DOCTORAL DISSERTATION

Design and Evaluation of a CMMI conformant Light-Weight Project Management Approach

MR. CHAYAKORN PIYABUNDITKUL

SUBMITTED IN PARTIAL FULLFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF ENGINEERING IN ELECTRICAL ENGINEERING
THE SIRINDHORN INTERNATIONAL THAI-GERMAN GRADUATE SCHOOL OF ENGINEERING
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK
ACADEMIC YEAR 2013

Author

Name	Mr. Chayakorn PIYABUNDITKUL
Title	Design and Evaluation of a conformant Light-Weight Project Management Approach
Major Field	Electrical and Software Systems Engineering
Advisor	Prof. Dr. rer. nat. Horst Lichter Assoc. Prof. Dr. rer. nat. Toni Anwar
Academic Year	2013

Acknowledgement

I would like to acknowledge the contributions of the following people, without their help and guidance this dissertation would not have reached completion. Firstly, I acknowledge the counsel and support of my supervisor, Prof. Dr. rer. nat. Horst Lichter and my co-supervisor, Assoc. Prof. Dr. Toni Anwar. Not only their valuable advices guided me through the research, but also they provided me an opportunity for doing the research in RWTH Aachen, Germany in the first year. I received lots of experiences on working in a team with my colleague both scientific staff and also secretary in Software Construction (SWC) group. I am indeed proud and fortunate to be supervised by them.

Secondly, my heartfelt gratitude goes to Dr.Chumphol Krootkaew and Dr.Apinporn Methawachananont who have always support my thesis related to Software Process Improvement project of NECTEC to CMMI maturity level 3. Thanks to Mrs. Tachanun Kangwantrakool who is NECTEC-CMMI consultant for her supports both valuable advices and information. Lastly, thanks to Mrs. Nithipat Wongchingchai and Mrs. Pawarat Nontasil for their guiding in Statistical which give more complete trustworthy through the data of writing the dissertation.

Thirdly, I would like to thank my parents, my family includes Dr.-Ing. Chayakorn Netramai and Dr. Sansiri Tanachutiwat, who always give me a moral support during several years of studies leading to the degree of Doctor of Engineering in Electrical Engineering. I also gratefully acknowledge Institute of Software Engineering improvement and quality Management (ISEM), mimacom group (Cornelia Gilgen and Raphael Ernst), Switzerland and my colleague from Software Engineering Laboratory (SWE) for providing the data for this work.

Chayakorn PIYABUNDITKUL

Abstract

The aim of this dissertation is to develop a new perspective framework which is standardized and offers best practices for very small or small and medium enterprises (VSEs/SMEs), and to gain very challenge admittance to the software market by companies who represent their accepted software capability/maturity quality standard. In the first part, we present an introduction, some definitions and the motivation for this work. It shows that there is a growing demand for software process improvement in VSEs/SMEs software development companies, but at present there is a very little guidance in this matter. The dissertation then identifies a combination of the Capability Maturity Model Integration (CMMI) and Agile Software Development Approaches. It provides a new approach to blend CMMI and Agile Approaches for VSEs/SMEs, focusing on project management in CMMI and Scrum. To this end, the dissertation introduces a number of practices/artifacts rating as well as conformance rating. Furthermore it presents a solution aiming to overcome the gap between CMMI and Scum by proposing CMMISF which is a development framework for VSEs/SMEs using a Light-Weight Project Management SCAMPI Assessment Model (LWPM-SAM) that is implemented by means of the Software Process Improvement Adaptive Learning System (SPIALS).

Finally we present an evaluation of the proposed solution. It shows that the performance of agile-Driven (CMMIbyScrum) approaches is better than plan-driven approaches regarding the aspects: controlling the project effort, software budgeting, and project duration. However, the project manager effort and software process improvement budgeting are better controlled by plan-driven approaches. This dissertation offers a framework and a self-assessment web-based tool to VSEs/SMEs software development companies, and thus makes a small contribution to improving the general standard of software process improvement in VSEs/SMEs environment.

Table of Content

List of figure

List of table

Chapter 1 Introduction

In this dissertation, we present a step by step to design and evaluate Capability Mutuality Model Integration (CMMI) with light-weight software development life cycle (SDLC) focusing on project management perspective.

Firstly, we start with a question of “How can we accelerate adoption CMMI by Agile methodologies?” which we assume that Agile is one of the representatives for light-weight SDLC. Based on our first question, there are many software organizations invested a lot of resources and budget to reach the target of high-quality by implementing high-weight organization plans and processes likes CMMI, however, Agile practices are one of the potential alternatives for accelerating a successful of developed organization to overcome the better result with more light-weight practices. The balance between CMMI and Agile practices is expected to increase the organization productivity. This is useful for plan-driven organizations that based on CMMI model to improve their processes concurrently with Agile practices by defining process framework based on rapid practices.

High-performance organization is the purpose of software development companies. CMMI is a plan-driven model that focuses on organization level for knowing "What to do?” On the other hand, Agile is an adaptive methodology that focuses on project or team level for learning "How to do?" The synergy from both approaches, aim to improve organizations that have been deploying CMMI, and plan to optimize their software development processes towards agility.

Secondly, after we realize that Scrum is better appropriate for adopting with project management practices then we step further to next question as “How can we step forward CMMI-Project Management by optimized Scrum?” Regarding to second question, there is a lot of success software development project based on CMMI have been broadly accepted throughout the world. One of the sparking points is deploying CMMI focused on project management category combining Agile method as Scrum practice which is expected to bring balance of high quality and optimal cost for VSEs/SMEs (Very Small Enterprises/Small Medium Enterprises) in software industry. Previous research question has shown an adoption of the synergy between CMMI model and Agile practices leading to satisfy for lightweight Software Process Improvement (SPI) procedure. However, this research is to specific focus on the gaps and the strengths/weakness between an iterative-driven (Agile-Scrum) and a plan-driven (Project management category in CMMI).

Finally, as we known that there are some process areas in project management which are appropriate for deploying Scrum practices then we would like to understand the comparison between Waterfall model and our CMMIbyScrum Framework

(CMMISF) based on topic namely, “Design and Evaluation of a CMMI Conformant Light-Weight Project Management Approach” to illustrate the difference of our investigation to propose our CMMISF for deploying with VSEs/SMEs from presented experiment. As we know, CMMI is one of the well-known and accepted maturity models that many software organizations have implemented for its quality processes which are expected to bring a good quality for their software products. However, traditional software process models become too heavy-weight to be deployed. The aim of this dissertation is to design the Light-Weight Project Management (LWPM) approach to implement CMMI by mapping between CMMI goals and Agile-Scrum based on defined artifacts and to indicate the differences in applying LWPM and the traditional Waterfall model. Our approach focuses on the Project Management category which composes Project Planning (PP), Project Monitoring and Control (PMC) and Integrated Project Management (IPM).

In order to compare both models we collected relevant data by using questionnaire and also the dedicated tool as namely, “Software Process Improvement Adaptive Learning System or SPIALS” for VSEs/SMEs. Because these resources are limited to these organizations for preparing the SPI self-assessment to start a continuous SPI initiative, in this case, we hope that our SPIALS can learn variety of qualitative and quantitative critical success factors and extract empirical data from experiences database. The following details are present our vocabulary definition of our research work.

1.1 Software Development Life Cycle (SDLC)

The Systems Development Life Cycle (SDLC) is a process definition which can be models serve as a high-level definition of the highlight key activities and their interdependencies as phases that occur during development. Examples of software life cycle models are the waterfall model, the throwaway prototyping model, evolutionary development, incremental/iterative delivery, the spiral model, the reusable software model, and automated software synthesis [1.1]. The standard on developing life cycle processes also provides a list of processes and activities for software development and maintenance [1.2].

SDLC typically consists of seven phases which are initiation of the project, definition requirements, functional design, built the system, verification, operation all establishments, and finalize with maintenance and review activities [1.3]. In general, SDLC is methodologies or process frameworks to develop an information system, and training usage for stakeholder. The SDLC aims to produce a high quality system that works effectively and efficiently as planned within cost-effective to enhancement.

The most important of SDLC model is waterfall for top-down implemented strategy which is the classic waterfall model was defined in early 1970 by Royce [1.4] and later refined by Boehm [1.5] in 1976 to help cope with the growing complexity of the software projects being tackled. On the other hand, Agile is the new continues SDLC which rises to change a dramatically increasing pace of today’s turbulent business and technology environment [1.6].

1.1.1 Waterfall model

The Waterfall process model is viewed as progressing linearly from conception, through requirements, design, code, and test. The Waterfall model shows a process, where developers can follow in each phase as in order; requirements specification and requirements analysis, software design, implementation and integration, validate test, and finally, deployment maintenance [1.4]. In a strict Waterfall model, after each phase is finished, it proceeds to the next one. Reviews may occur before moving to the next phase which allows for the possibility of changes as a formal change control process. Waterfall discourages revisiting and revising any prior phase once it's complete so that this brings inflexibility for purring Waterfall model [1.7].

1.1.2 Scrum model

In recognition of these ideas, in February 2001, the original of Agile community is "The Agile Alliance", this group of independent thinkers about software development, and sometimes competitors to each other, agreed on the manifesto for Agile software development as “Individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following plan” [1.6].

Agile methods stress two concepts which are the unforgiving honesty of working code and the effectiveness of people working together with goodwill. Using Agile development methods requires close customer partnerships. [1.8] Agility is dynamic, context-specific, aggressively change embracing, and growth-oriented. It is not about improving efficiency, cutting costs, or battening down the business hatches to ride out fearsome competitive “storms” [1.9]. Agile software development is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing and cross-functional teams. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change [1.10]. Agile has variety methodology. This research focuses on “Scrum” which is a simple process for managing complex projects [1.11].

Scrum is an Agile framework for completing complex projects. Scrum originally was formalized for software development projects, but works well for any complex, innovative scope of work [1.12]. Scrum is an iterative and incremental Agile software development framework. Its focus is on "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal" as opposed to a "traditional, sequential approach". Scrum is for achieving results in complex situations. Using practices such as the Product Backlog, the results can be optimized to the situation. But Scrum is also very much about people. Scrum Masters become dedicated to their teams because teams are neighborhoods that people, including the Scrum Master, live within [1.11].

1.2 Software Process Improvement (SPI)

Software process improvement is an action which is taken to change an organization's software processes. They have to meet the organization's business needs and help it to achieve its business goals more effectively [1.14]. SPI requires management to take an action in active role and also the participated workers in defining and implementing usable and effective processes. SPI has been practiced when the Software Engineering Institute (SEI) in Pittsburgh first lunched its Capability Maturity Model for Software or the SW-CMM. Software Process Improvement is based on software process assessment (SPA) which is concerned with assessing a software process against a process standard or framework such as Software Process Improvement and Capability determination (SPICE) to support the development of an International Standard for Software Process Assessment [ISO/IEC Std 15504] [1.13] or Standard CMMI Appraisal Method for Process Improvement (SCAMPI) to provide benchmark-quality ratings relative to Capability Maturity Model Integration (CMMI) models [1.15].

1.3 Project Management in Capability Maturity Model Integration (CMMI)

The original version of CMMI is the Capability Maturity Model for Software (SW-CMM) is a framework that demonstrates the key elements of an effective software process. The CMM describes an evolutionary improvement path for software development from an ad hoc, immature process to a mature, disciplined process, in a path laid out in five levels. [1.16] Then the CMMI enhancement describes the stages through which software organizations evolve as they define, implement, measure, control, and improve their software processes. These models developed by the Software Engineering Institute (SEI). It provides a guide for selecting process improvement strategies by facilitating the determination of current process capabilities and the identification of the issues most critical to software quality and process improvement. [1.17]

The CMMI has category in four groups and twenty-two process areas, in project management category, there activities related to planning, monitoring, and controlling, In CMMI-DEV version 1.3, the PM category compose of seven PM process areas which are Integrated Project Management (IPM), Project Monitoring and Control (PMC), Project Planning (PP), Quantitative Project Management (QPM), Requirements Management (REQM), Risk Management (RSKM), and Supplier Agreement Management (SAM). In this dissertation we focus only "IPM, PMC and PP" [1.17].

1.3.1 Project Planning (PP)

The project plan covers the various project management and development activities performed by the project. The purpose of PP is to establish and maintain plans that define project activities which are based on their specific goal for instance establish estimates of the project scope, effort, cost, work product and task attributes, to develop a project data management, and stakeholder plan based on obtain commitments.

1.3.2 Project Monitoring and Control (PMC)

The project monitoring and control process area contains practices for monitoring and controlling activities and taking corrective action which is specifies the frequency of progress reviews and the measures used to monitor progress. The purpose of PMC is to provide an understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan by monitoring the project against the plan, and managing corrective action to closure the project.

1.3.3 Integrated Project Management (IPM)

The integrated project management process area establishes and maintains the project's defined process that is tailored from the organization's set of standard processes or organizational process definition. The purpose of IPM is to establish and manage the project and the involvement of relevant stakeholders according to an integrated and defined process that is tailored from the organization's set of standard processes by using the project's defined process, coordinating and collaborating with relevant stakeholders.

1.4 Standard CMMI Appraisal Method for Process Improvement (SCAMPI)

The standard CMMI appraisal method for process improvement (SCAMPI) is designed to provide benchmark-quality ratings relative to CMMI models. SCAMPI A satisfies all of the appraisal requirements for CMMI (ARC) requirements for a Class A appraisal method. Although designed for conducting appraisals against CMMI-based reference models, the SCAMPI A method can also be applied for conducting appraisals against the People CMM and other reference models.

The SCAMPI A appraisal method is data-oriented which is used to identify strengths, weaknesses, and ratings relative to appraisal reference models. It is decisions on practice implementation and goal rating are made based on the aggregate of objective evidence available to the appraisal team. The SCAMPI A Phase for Conducting Appraisal is composed of plan and prepared for appraisal, conduct appraisal, and report results.

Chapter 2 Motivation

Nowadays, there are numerous competitions in software development industry which means the software development companies across the world have recognized the need to focus on their core business competencies by efficient working to develop and upgrade their products and services. Consequently, these organizations need to outstanding from other competitors in the market. It is important for them to concentrate on their core business with high performance.

The concept of software development outsourcing is growing dramatically, particularly in the Asia. Once in the past, Thailand is the fourth ranking among the top outsourcing destinations of the world next to India, China, and Malaysia. [BusinessWeek: July 2009] This trend not only raises new service contract in the software industry but also brings new opportunities. Therefore, many very small enterprises or small medium enterprises (VSEs/SMEs) software development companies are unaware of this, or do not even know that how to keep their abilities to compete for surviving in this business. However, some of VSEs/SMEs firms are concern about particularly vulnerable to their developments on what can they do to avoid the threats? And how can they exploit the new opportunities?

Nevertheless, this is challenge to the software companies to gain admittance to the market; however, there are some requirements from international buyers/employers who do not much know about Thai software development companies. In this case, they prefer these companies to represent their capability or maturity likes standard to be accepted. Many of the Thai firms possess quality accreditation by reaching international standard for instance International Organization for Standardization (ISO) or Capability Maturity Model Integration (CMMI). Furthermore, in order to maintain their competitiveness in this battle, they not only have to retain the necessary skill set based on their limitation of resources, but also they have to concern in various topics; e.g. prepare a good request for proposal, set a guaranteed delivery date, consider cost and value and get stakeholder buy-in. Presently, requirements of software development environment often change during the product development life cycle to meet shifting business demands.

In order that, Agile development is propose to solve this issue. This Agile approach based on iterative development, frequent inspection and adaptation, and incremental delivery, in which requirements and solutions evolve through collaboration in cross-functional teams and through continuous stakeholder feedback. Scrum is one of the Agile popular methods that use to address these concerns. Scrum is basically an iterative project management framework used in Agile development, in which a team agrees on development items from these required backlogs and produces them within a short duration in a month. Thus, Scrum is appropriate for projects where requirements up front cannot be defined and chaotic conditions are anticipated throughout the product

development life cycle model. All in all, this is motivating reasons to find a new solution for designing the light-weight framework based on Agile approach and CMMI model. And also present the evidence of an empirical to show how its efficiency.

Therefore, VSEs/SMEs have a very challenge goal to compliance and synergize to use different combination of both approaches. Establishing an organization process is accepted by the world standard for instance CMMI for process control, quality improvement, and capability/maturity evaluation. On the other hand, the process has to compete in price and performance competition by combining Agile development likes Scrum in their software project environment. However, there are some reasons that may be obstacles to VSEs/SMEs development teams; e.g. CMMI is too expensive to use and a lot of overhead and documents. Hence, it is interesting for VSEs/SMEs to have a new perspective framework which is standardize and optimize the best practices of their organization. There by, VSEs/SMEs need to modify and tailoring their own frameworks to get additional values or benefit to its practices. From this notice, VSEs/SMEs should define the combination of CMMI and Agile to complete each other which evoke to enhance visible performance, long-term benefits and affordable cost. Thus, the concept of “CMMI-by-Agile” in light-weight approach instead of selection between CMMI and Agile are occurrence.

2.1 Current Status

In the world of software competition, a large software development company has provides access to their huge available resources. However, a requirement change in a large company is often making slow, cumbersome and incredible frustrating times. Next, the communication and coordination overhead rises dramatically with large size team. This seems to always be systematic inefficiencies or roadblocks that prevent the necessary amount of resource allocation.

A small software development company is faced not only with a lack of resources and funds which are required to implement in their practices, but also with process improvement programs based on the assessment model likes CMMI which is not fit with small organization. Nevertheless, a small company will have more responsibility, respect, familiarity and trust. Employee in small firms probably wears multiple hats and it goes beyond the shared focus on a single goal, or the ability to influence overall direction. Once process improvement is accepted as a course of action by the management, it should much easier to change the corporate culture and steer the organization toward improvement goals in a small company than in a large company because of less inertia and less bureaucracy.

2.2 Challenges Questions & Goals

Regarding to these current situations of blending CMMI and Agile approach, it is significant initialization to start with harmony elements from both sides. Based on previous works, the definition of an Agile approach on Scrum and CMMI is significantly usefulness for organizations that have been trying to improve their processes between

agility and maturity models [2.5]. Scrum is shown satisfactory percentage of conformity with CMMI in project management category [2.19]. Scrum practice is very close to Project Planning (PP), Integrated Project Management (IPM) and Project Monitoring and Control (PMC) and the success of perfect conformity with CMMI can be fulfilling your implemented with other Scrum practices [2.20]. Therewith, next step is to set the context and objectives related to the dissertation approach to focuses on Scrum and project management as ordered for Agile approach and CMMI model which the main objectives of the challenge questions are as follow;

- To identify the questions related to the Agile approach and Scrum based on project management category in CMMI assessment models

- To develop a summary result of the related previous works with identified their studies based on the research questions.

- To define the scope to verify the concerned research questions.

As the strategy, the research questions are proposed below;

RQ1. Are Agile and CMMI mutually exclusives approaches for process improvement program?

This question tried to figure out whether Agile approach & Scrum and Project Management in CMMI model are synergized. This question assesses compatibility dimension.

RQ2. What is a gap between Scrum and the project management requirements of CMMI?

This question is based on how much compatible and gap items of Agile approach & Scrum and Project Management in CMMI model are. This question assesses compliance dimension.

RQ3. How to combine Scrum and the project management, CMMI to close the gap?

This question tried to find out to identify on how can we fulfill the CMMI goals via Agile approach & Scrum the different or lacks for implementing CMMI when develop by using Agile approach & Scrum. This question assesses capability/maturity dimension.

RQ4. What is the benefit to synergize Scrum and the project management requirements of CMMI?

This question tried to identify empirical experiences to prove that a development by Agile approach & Scrum can be useful to gain for reaching the CMMI goal. This question assesses profitability dimension.

The dissertation proposes a summary of related works based on the research questions which are explained in literature review session.

The goals of this work are to analyze the feasibility, to design the combining framework of the CMMI model and Agile approach with Scrum, and to evaluate the profitability of this synergize works. The CMMI-SCAMPI scope is the Project Management category which contains PP, PMC and IPM. The main reason for selecting the project management category, because it is very important to the successful of the project in overall perspective based on its activities.

2.3 Related work

Currently, there are many works related to the motivation of this propose framework. Regarding to the mutually exclusives approaches between Agile and Scrum (see RQ1), Jeff Sutherland and et al. [2.12] argued that as Scrum and CMMI together bring a more powerful combination of adaptability and predictability than either one alone like propose by Carsten Ruseng Jakobsen, Kent Aaron Johnson [2.15] and Hillel Glazer [2.18]. Moreover, Armin Preis [2.2] commented this opinion; Scrum and CMMI match in large parts and the integration of both concepts can involve strong synergetic effects; e.g. reducing complexity in CMMI and expanding Scrum's processes towards higher process quality. Nevertheless, organizations must be aware that it usually takes long, to combine both concepts and to improve them over the lifecycle of the maturity process. And there is a need to extend Agile methods for covering all the mandatory goals and expected practices for CMMI level 2 and 3.

Related to RQ2, Ana Sofia C. Marcal, Bruno Celso C. and et al. [2.4] stated that Scrum does not cover all the specific practices of the project management process area, but it could be tailored to be more compliant with CMMI. On the other hand, we can conclude that a plan-driven process based on CMMI model can be improved by adding some Scrum Agile practices to their activities. As well as the Scrum project management processes and practices satisfy the CMMI project management requirements only fully at Maturity Level 2 and at least partially Level 3 [2.2].

To overcome these problems (see RQ3), some authors found that it is possible for small software development organizations or small co-located projects [2.6] to achieve a CMMI certification implementing Agile methods [2.16], [2.13]. Likewise, Martin Fritzsche, Patrick Keil [2.11] and Neil Potter and Mary Sakry [2.9] stated that Agile methods can be applied without any major adaptations up to level 2 and up to 3 with some minor changes, however, some process areas, mainly those of the maturity levels 4 and 5, are in conflict with Agile principles. The gap can close by adding an effort for the success of CMMI’s perfect conformant. It can be closed by implementing with other Agile practices [2.20] However, Armin Preis [2.2] argues that CMMI maturity level 4 is reachable by the introduction of Agile metrics. And CMMI Maturity Level 5 and Scrum seem to be ultimate partners, as the goal of Maturity Level 5 is to reduce complexity and increase organizational and process efficiency.

Therefore (see RQ4), the benefits of synergizing both Scrum and the project management requirements of CMMI, are presented by many other works. Scrum and CMMI are significantly useful for organizations [2.5], [2.1], [2.14]. Software development project can benefit from them [2.3] and the return on investment (ROI) is

high for deploying the Agile methods [2.8], [2.17], however, there is a conflict result in P.J. Rundle, R.G. Dewar [2.7] group experiment. Moreover, M. A. Awad said that in general, there are some aspects of software development project that can benefit from an Agile approach and others can benefit from a more predictive traditional approach. When it comes to methods, each project is different. However, one thing is clear: that there is no "one-size-fits all" solution [2.3]. Hillel Glazer [2.18] also argued that Agile helps to improve many operational and transactional activities but was not intended to provide higher levels of organizational constructs to facilitate long-term process evolution.

Furthermore, Minna Pikkarainen and Annukka Mantyniemi [2.10] concluded that Agile software development using CMMI produces useful results for starting Agile based improvement efforts as same as comments from Jeff Sutherland and et al. [2.12] stated that Scrum shows many optimistic deploy practices for less high-weight SPI procedure, both are strength in Engineering and Project Management categories, however, they still have some weaknesses in term of Process Management and Support categories [2.19].

And it is focuses on how to implement a CMMI model for process improvement program based on Agile development. This framework aims to close the gap and some conflicts between Scrum approach and CMMI model. And also to propose efficient framework to produce better performance and less effort as CMMI Appraisal Method for Process Improvement (SCAMPI)'s needed. Moreover, we analyzed and proved propose framework by requesting questionnaires to software development industry regarding to the related practice information between Agile approach & Scrum and Project Management in CMMI model.

2.4 Structure of the dissertation

In Chapter 1 is an introduction, the first chapter describes basic theories for Software Development Life Cycle (SDLC), Software Process Improvement (SPI), Project Management in Capability Maturity Model Integration (PM in CMMI) and Standard CMMI Appraisal Method for Process Improvement (SCAMPI).

Chapter 2 is about motivation, this chapter introduces background, motivation and challenges for this dissertation.

Chapter 3 is related to combine CMMI and Agile Software Development Approaches, this chapter describes the synergy of CMMI and Agile in small setting environment likes Very Small/Small Medium Enterprise (VSEs/SMEs) and how to overcome the Gap in CMMI and Agile by Scrum. Finally, we propose the proposal of CMMISF; the development framework for small setting environment (VSEs/SMEs).

Chapter 4 is explain Light-Weight SCAMPI Assessment Model (LW-SAM), this chapter describes Light-Weight SCAMPI Assessment Model (LW-SAM) and SPIALS; the tool to support LW-SAM.

Chapter 5 is about an evaluation, this chapter describes design and set up of the evaluation, defined questionnaire, data collection and data analysis.

Finally, Chapter 6 is a conclusion and future research work, in the last chapter describes answers to research questions, implications, limitation of the study and open questions and future research, the overall procedures in this dissertation are concluded.

Chapter 3 Combining CMMI and Agile Software Development Approaches

Many software organizations are invest a lot of resources and budgets to reach the target of high-quality by implementing high-weight organization plans and processes likes CMMI (Capability Mutuality Model Integration). However, small software development companies like VSEs/SMEs, Agile practices are one of the potential alternatives for responding a successful of developed organization to overcome the better result with more light-weight practices. The balance between CMMI and Agile practices is expected to increase the organization productivity.

Ita Richardson reported about small software development companies in Software Journal that “Developers around the world are working on adapting software engineering solutions for small organizations, and the number of experience reports on such applications is increasing. Customized approaches will likely become more available. Furthermore, interest in research in small software companies seems to be increasing, so researchers' skills and experience are becoming more available in those settings. These factors will contribute to supporting small organizations as they apply software engineering solutions and help them operate more effectively and efficiently. Small software companies need efficient, effective software engineering solutions.”

Moreover, she said about the team and organization that “People often believe that good practices and solutions are expensive, time consuming, and targeted more toward large organizations, and therefore difficult to apply in small companies. Large and small software development companies face similar software engineering challenges. They need to manage and improve their software processes, deal with rapid technology advances, maintain their products, operate in a global software environment, and sustain their organizations through growth. However, they often require different approaches because of specific business models and goals, market niche, size, availability of (financial and human) resources, process and management capability, and organizational differences, among other things. Small companies are not just scaled-down versions of large firms. Generally, they are extremely responsive and flexible, because that is their advertised competitive advantage”.

In general, high-performance organization is the target of VSEs/SMES software development companies. On one hand, CMMI is an international quality standard assessment model that focuses on organization capability or maturity level for knowing "What to do for keeping software development quality standard?” On the other hand, Agile is an adaptive method that focuses on project or team level for learning "How to do the best practice for their organization?" The synergy from both approaches, aim to

improve organizations that have been deploying CMMI, and plan to optimize their software development processes towards agility.

The purposes of this chapter is to explore the possibility for these VSEs/SMEs CMMIer organizations, firstly, to apply the best fit Agile practices by emphasize on small or medium size organization. Secondly, it is also to find the best category in CMMI that fit for Agile practices. Consequently, Agile practices have a promising capability to fulfill the optimized resources for accomplishing of the SPI organizations.

The combining of CMMI and Agile in Table 3.1 shows the similarities and dissimilarities of both two approaches as shows on Table 1. CMMI is an organizational process with plan-driven approach for determining organizational maturity and process capability. It provides high-weight documentation, procedural, extensive planning, predictability, and stability through discipline which make all functions and capabilities contributing to the development of products and process as the process improvement effort. Many organizations are adopted CMMI to build software process improvement (SPI) framework for high level of maturity standard in software industry. On the other side, Agile practices are iterative approach for minimize time and process, creativity, responsiveness to change and continuous releases through customer respond and customer change requirement. It provides light-weight documentation, and incremental life-cycle through customer respond [3.1] [3.2].

Table 3.1 Similarities and dissimilarities of CMMI and Agile Practices

No.	Properties	CMMI	Agile
Similarities			
1.	Goal	High Performance Organization	High Performance Organization
2.	Process Reference	Based on Best Practices	Based on Experience
Dissimilarities			
1.	Approach	Process	Customer Responses
2.	Purpose	Improve and measure process	Minimize process and short time
3.	Communication	Macro in Organizational	Micro for Person to Person

No.	Properties	CMMI	Agile
4.	Knowledge Management	Process assets	People assets
5.	Type of People	Disciplined, Follow Rules, Risk Averse	Comfortable, Creative. Risk Takers
6.	Characteristics	Carefully Changes, High-weight documentation	Rapid Changes, Light-Weight documentation
7.	Life-cycle	Procedural, Extensive Planning	Iterative, Incremental
8.	Improvement Level	Organizational Level	Project Level
9.	Capability/ Maturity	Success by Predictability	Success by Realizing Opportunities
10.	Working group	Committees	Individuals
11.	Customer Trust	In Process Infrastructure	Working SW, Participants
12.	Team Trust	Low-Trust Environment	High-Trust Environment
13.	Risk Management	Proactive	Reactive
14.	Business Focus	Internal Rules, Stability	External Innovation, Speed
15.	Management Style	Ordering	Coaching
16.	Planning Style	Project-level	Product-level
17.	Marketed Customer	Mature and process user	Emergent and not well-understood
18.	Learning	Multi-level	Macro-level
19.	Appraisal	Processes against the practices	Results against the customer satisfaction
20.	Cost of Failure	High cost	Low cost

Moreover, CMMI management style plays very important roles for the success of the project by ensuring the plan, managing dependencies and performing risk management. The CMMI team is disciplined and followed rules based on risk adverse via

processes against the practices as SCAMPI. CMMI has multi-level of team learning; training class is establishes as an organization level. In additional, technical training and self-learning are issues from the development activities, as a project level. While, Agile management style is focus on coaching function, recognized the team work with high-trust on individual operating based on comfortable and creative via results against the customer satisfaction. Learning in Agile is mostly happens at project levels. Finally, CMMI is extremely high cost of failure when a plan is crashed; Agile is in a domain of low cost of failure or linear incremental cost of failure.

Although, there are many of dissimilarities, CMMI and Agile practices have the same main goal; its aim to be high performance organization based on practices and the best experiences. The balances of both approaches are challenge. And also the synergy of CMMI and Agile are purposely identified in this research.

3.1 CMMI – What to do for small setting environment (VSEs/SMEs)

On October 19 and 20, 2005, the first International Research Workshop (IRW) for Process Improvement in Small Settings was held in Pittsburgh, Pennsylvania, USA. The goal of the workshop focused on research from the world-wide community addressing the unique issues of process improvement in small settings, including small teams, small projects, small organizations, and small businesses. The workshop was the result of two synergistic forces; firstly, the SEI'S Applying CMMI in Small Settings (ACSS) project was to foster communication and collaboration among worldwide researchers to leverage learning related to applying CMMI and other process improvement techniques in small settings (projects, organizations, and companies) and secondly, the International Process Research Consortium (IPRC) identified implementing process improvement in small settings as one of the early high-priority topics needing primary transition research rather than technology research.

The term small setting has been defined as an organization or company of fewer than approximately 100 people, and a project of fewer than approximately 20 people [SEI 04], with less than 25 people and a project of fewer than approximately 6 people as a very small setting. The small are include both small projects in large companies, and small projects in small organizations. A major aspect to be considered in these environments is that the amount of resources used to support a process improvement effort would be a large percentage of an organizations operating budget, [SEI 04]. In addition, there are three more aspects about the environment that I consider relevant in very small settings; the cash flow of the company, the people skills, and the project size.

Moreover, the definition of small and very small enterprises is challenging. To take a legalistic perspective the European Commission defines three levels of small to medium-sized enterprise (SME) as being: Small to medium – "employ fewer than 250 persons and which have an annual turnover not exceeding 50 million Euro, and/or an annual balance sheet total not exceeding 43 million Euro. The term "Very Small Entity" (VSE) had been defined by ISO/IEC 29110 as being "An entity (enterprise, organization, department or project) is having up to 25 people". (Laporte, C.Y. et.al.)

The result performed that a process improvement (PI) project based on a comprehensive reference model such as the Capability Maturity Model Integration (CMMI) requires additional effort and time to interpret the model. It is common for small companies to have budget, schedule, and organizational resources constraints. Activities such as planning, task assignment, training, and developing schedules are also needed with further requires sponsorship from top executives and a good communication scheme to motivate the individuals involved in this continuous endeavor, the CMMI in its current format and packaging is not feasible for SMEs to adopt and implement. The challenges to successfully carry out a PI project based on CMMI are considerable. In some cases from small and medium Brazilian companies, the implementation of SW-CMM level 2 into the company was not an easy task. Many obstacles had to be overcome, not only on the technical and relationship sides, but also in the supply of financial resources and staff. The success of the project improvement depended on the ability and the involvement of its professionals, the relation of co-responsibility between the staff and the company then if a team is well prepared, motivated, well-coordinated and uses a tool that supports the implementation of the model, implementing process improvement, and achieving the desired maturity level which can be quick and successful.

In CMMI based on Agile practice, the results of the application for project management in improvement projects for small settings have lead us to think that is feasible to use the CMMI model to strengthen defined Agile practices. These practices allow an organization to improve its project management activities significantly and at the same time, to be compliant with the requirements of a high capability level (using the continuous representation) for project management activities.

Some recognized authors have declared that a balance between Agile methods and the CMMI are feasible [Paulk 2002, Boehm 2003]. They believe that declaration is the basis for this work. They find the current results promising. The work performed so far with the Project Planning (PP) and Project Monitoring and Control (PMC) process areas lets them state that these organizations are able to achieve CMMI capability level 2 for these process areas and keep the process Agile and adequate for the organizations. The next step is to try to articulate Risk Management (RSKM) and Integrated Project Management (IPM) development to support capability level 3 for PP, PMC, and RSKM. In China and Hong Kong SME Software Companies, most of the SMEs also have staff that evolves to the project management/sales offices rather than software development [3.3].

SEI revealed that, there are challenges to successful in CMMI best practices as well. No development approach or methodology can be effectively addressed all difficult challenges or situations [Elm 2007]. An organization has been appraised at a particular CMMI maturity level. There is no guarantee that a particular project in the organization will succeed. However, organizations using CMMI can fail because they misuse the model or pursue process improvement and subsequent appraisal with misguided motivation or with imprudent leadership. CMMI and Agile each brings something to the table on how to run the business that the other side should listen to and learn from healthy and beneficial dialogue based on users, the paradigms, and the broader community.

Finally, the relative importance of these critical success factors varies across different types of projects. Therefore, almost of the primary sources of the generic CSFs is related to "project management". So that, in our study, we focus on the first most influence in the VSEs/SMEs successful CSFs with "Project Management in CMMI" and Scrum which is also starting to focus on project management approach based on 3 process areas which are Project Planning (PP), Project Monitoring and Control (PMC) and Integrated Project Management (IPM). The challenges when using CMMI today shows CMMI based organizational process improvement approaches cannot rely executively on the traditional project management approaches likes Waterfall software development life cycle project and also other heavyweight methods.

3.2 Agile – How to do for small setting environment (VSEs/SMEs)

There are many researches show a successful of Agile methods by implementing in small setting environment. Agile development methods are gaining popularity for small programming projects with tight deadlines; however, they are some points that Agile has to fulfill some weaknesses. B.Fitzgerald et.al. reported that, Agile methods were seen initially as the best suited to small, co-located teams and developing non-critical systems. This is an important issue as research suggests that Agile methods are developer-centric and are typically enthusiastic embraced by developers, but management require the actual business benefits of Agile methods. Moreover, T. Dingsoyr et.al. revealed that, the benefit of the scrum software development process in small cross-organizational development project is flexibility and motivation. In addition, the estimation can be challenging due to the customer-provider relationship between the participating organizations.

Scott W Ambler said in the Agile Scaling Model (ASM) that, Agile is not only suitable for small setting environment but also capable for Scaling Agile strategies at the project level. It is a contextual framework for effective adoption and tailoring of Agile practices to meet the unique challenges faced by a system delivery team of any size. The ASM is depicting to distinguish between three scaling categories:

1. The core Agile development composes of value driven life cycle, self-organizing teams and focus on construction.

2. The disciplined Agile delivery related to risk & value driven life cycle, self-organizing teams and full delivery life cycle.

3. The Agility at scale which is composes of disciplined Agile delivery when one or more scaling factors apply: large team size, geographic distribution, regulatory compliance, domain complexity, organization distribution, technical complexity, organizational complexity and enterprise discipline as presented in Figure 3.1. The distinct definition between simple and complex in term of agility scale are shown in Figure 3.2.

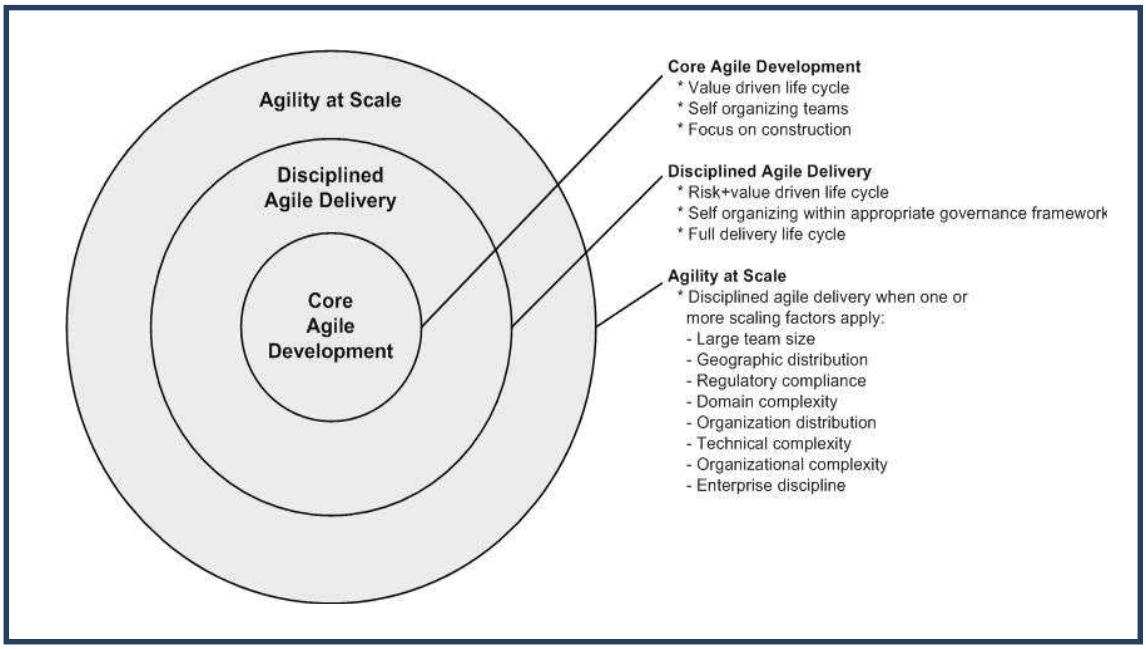


Figure 3.1 The core relationship in Agile

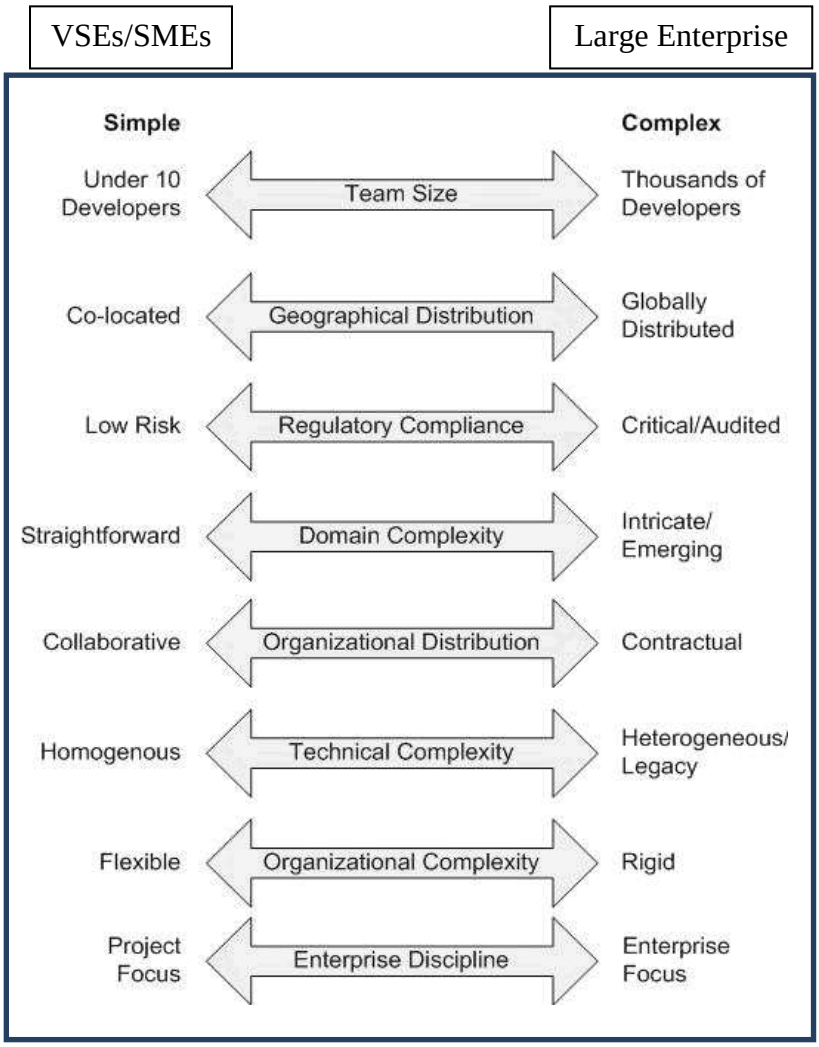


Figure 3.2 The distinct Agility Scale between Simplicity and Complexity

The ability of Agile practices to scale to “large” software development efforts has been widely debated in recent years. When considering a large-scale development effort, ensuring that there is a sufficient number of people with the desired behaviors and the appropriate infrastructure and processes. (E.Moore, J.Spens) Using Agile methods to develop large systems presents a thorny set of issues. If large teams are quickly produce lots of software functionality, the Agile methods involved must scale to meet the task. After that, a small team could create the software if the functionality to be delivered was small and, conversely, could be delivered given we had the time. Scaling Agile teams thus becomes an issue if the only option for meeting a system delivery deadline is to have many developers working concurrently (Donald J. Reifer et.al.).

Lastly, the major challenge when using an Agile approach in a large project is keeping the small teams aligned and coordinated for the duration of the project to ensure its success while adhering to Agile team-focused principles and values. Maintaining alignment and coordination across a distributed project that require someone (possibly a team) or a mechanism maintain coherence (i.e., unity, logic, and consistency) of the following: overall system capabilities to be developed, including non-technical requirements scope, quality, schedule, cost, and risk tradeoffs product (or service).

The challenges when using Agile approach is keeping the small team to ensure the success of the project based on Agile principle and values. Agile methods are generally lack of practices for implementing and supporting an Agile approach across the organization. Blending a top-down approach likes CMMI and a bottom-up approach likes Agile are concerned to improve the management effort and performance.

3.3 The synergize of CMMI and Agile in VSEs/SMEs

CMMI and Agile can complement and synergies each other. CMMI provide the best practices of engineering and the organizational process management. Agile methods provide on how to manage software development which is missing from CMMI.

From Figure 3.3, CMMI model describes the three critical dimensions which are identified by the Software Engineering Institute (SEI). The three aspects of development projects are compose of 1) Processes-Procedures and methods defining the relationship of tasks 2) Technology-Tools and equipment and finally, 3) People-People with skills, training, and motivation.

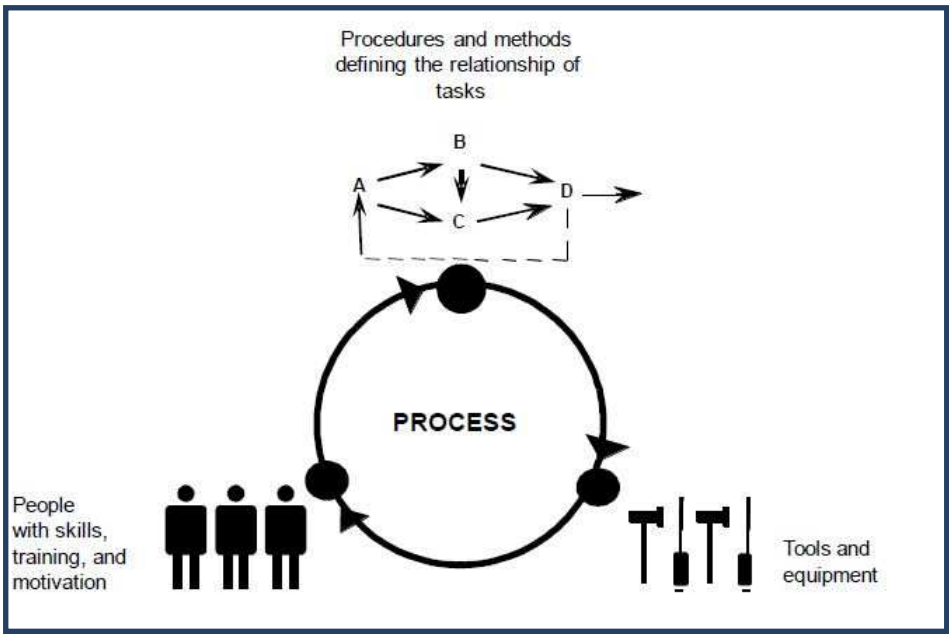


Figure 3.3 The core relationship in CMMI

The established project which is designed CMMI practices by implementing Agile principles can create more acceptable in process definition activities. Moreover, in case of implementing CMMI practices in an Agile organization, the combination of CMMI goals into the Agile project activities can make these teams more mature and/or capable at managing the continuity of projects. Whereas, CMMI organization who is adding a faithful to Agile principles. It maintain continually a lean, interactive activities, customer collaboration and responding to change over for developing the product. This synergy is clearly to bring more efficiency software development team. Thus, CMMI focuses on processes. Meanwhile, Agile methods focus on people who determine technology and its procedures.

3.3.1 The relationship between Project Management in CMMI and Scrum

In order to select an appropriate priority of CMMI category to implement with suitable Agile methods, this research reviews from the Critical Success Factors (CSFs) from Managing Offshore Software Development (OSD) Projects [3.4] which is presented identification and structure of the critical success factors (CSFs) for software development project based on implementing practices. For instance, the "definition of clear project goals" represents the most relevant CSFs for VSEs/SMEs. The other CSFs rated as significantly relevant for the success of an OSD project is "continuous controlling of project results", "ensuring of a continuous communication flow", "high quality of offshore employees", "good language abilities of the offshore employees", "composition of an appropriate project team", and "preparation of a detailed project specification".

Moreover, there are supported evidences which help project managers focus on areas that can make difference between success and failure in software development projects such as the "clear defines objectives", "top management support", "adequate budget", "realistic schedule", "client/user participation", "project leadership", "project reviews", "change control/management", "communication", and "problem solving" [3.5].

The relative importance of these critical success factors varies across different types of projects. Therefore, almost of the primary sources of the generic CSFs is related to "project management". In this study, we focus in the first most influence in the VSEs/SMEs successful CSFs with Project Management in CMMI and Scrum which is also focused on project management approach.

The core relationship between CMMI and Agile by Scrum are shown an example of Scrum framework that conforms to Specific Goals (SG) in Project Planning area in Table 3.2.

Table 3.2 The conformation between Specific Goals (SG)
in Project Planning (PP) area with Scrum framework

CMMI	**Agile method**
PP: Project Planning	**Scrum framework**
SG 1 Establish Estimates	• sprint planning • tasks and iterative effort estimations (4 weeks) • self-organizing teams
SG 2 Develop a Project Plan	• sprint planning • sprints, product backlog • sprint backlog • daily meetings • tasks and iterative effort estimations (4 weeks) • self-organizing teams estimations
SG 3 Obtain Commitment to the Plan	• sprint planning, sprint review • self-organizing teams

The mapping analysis for all process areas are shown in Table 3.3 to Table 3.25. The relationship of Project Planning (PP) area and Scrum practices and artifacts to determine the Specific Goal (SG), Specific Practices (SP) which can benefit from Scrum framework are shown in Table 3.3. In other words, it is present how much Agile can complete the implementation of define process area in CMMI. The criteria meaning for all mapping analysis tables below are reference from;

1. Criteria for Practices and Artifacts rating;
   Conformance of Scrum regarding to process areas related to project management
   1.1 Scrum Conformance

Let PMCMMI be the set of all CMMI process areas related to project management.
For all pa ∈ PMCMMI let SPpa be the set of all specific practices of process area pa.
Let SP be the union of all SPpa, pa ∈ PMCMMI
Let SC = {U, PS, S} be the set of conformance coverage values where

U: The practice is not addressed by Scrum.
PS: There is some evidence of the practice being addressed by Scrum; however, the practice is not fully addressed.
S: The practice is fully addressed by Scrum.

For all sp∈ SP the metric Conf (sp) → SC determines the Scrum coverage of sp.

** Source of defined Criteria: Based on several papers 21.[2.4], 22.[2.5] which are related to CMMI and Scrum conformance.*

1. 2 Conformance Value

For all pa ∈ PMCMMI let ConfSpa = {sp ∈ SPpa | Conf (sp) = S} the set of all sp that are fully addressed by Scrum.
For all pa ∈ PMCMMI let ConfPSpa = {sp ∈ SPpa | Conf (sp) = PS} the set of all sp that are not fully addressed by Scrum.
For all pa ∈ PMCMMI the metric ConfValue (pa) → [0,100] determines the conformance value of the process area pa to Srcum.
It is defined as follows:

ConfValue (pa) = ((| ConfSpa | + (| ConfPSpa | * 0.5)) / | SPpa |) * 100

2. Criteria for Conformance rating

For all pa ∈ PMCMMI the metric ConfRating (pa) → [+++,++,+,-] determines the conformance rating of the process area pa to Srcum where

+++*Fully addressed in Scrum*
++ *Largely addressed in Scrum*
+ *Partially addressed in Scrum*
- *Not addressed in Scrum*

It is defined as follows:

ConfRating (pa) = { +++ : if ConfValue (pa) >= 80
++ : if 80 > ConfValue (pa) >= 60
+ : if 60 > ConfValue (pa) >= 40
- : if ConfValue (pa) < 40}

*** The format that presents in table 3.3 to table 3.25 of "criteria for conformance rating is ConfRating (ConfValue)".*

**** The "U" is identified that Specific Practices (SP) in CMMI are not addressed by Scrum practices (DSM/SPM/SRM/SR) and Scrum artifacts (PB/SB/BDC).*

3. The criteria for conformance rating are come from 100% Satisfied, 50% Partially Satisfied and 0% Unsatisfied.
4. The Scrum framework has 4 practices; DSM (Daily Scrum Meeting), SPM (Sprint Planning Meeting), SRM (Sprint Review Meeting), SR (Sprint Retrospective) and Scrum Artifact has 3 artifacts; PB (Product backlog), SB (Sprint Backlog), BDC (Burndown Chart).

***** Source of conformity percentage of process Area in table 3.3 to table 3.25: Supported by EPG and CMMI Lead Appraisal from 3 companies with 10 participants (1st step is operated by Delphi method then 2nd step by statistical process)*

Table 3.3 Conformity percentage of Project Planning (PP) with Scrum framework

PP: Project Planning		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	Description (WP)	DSM	SPM	SRM	SR	U	PB	SB	BDC	U
PP.SG 1	**Establish Estimates**									
PP.SP 1.1	Estimate the Scope of the Project		S							
	Task descriptions Work package descriptions WBS						S			
PP.SP 1.2	Establish Estimates of Work Product and Task Attributes		S							
	Size and complexity of tasks and work products Estimating models Attribute estimates Technical approach							S		
PP.SP 1.3	Define Project Lifecycle Phases		S							
	Project lifecycle phases							S		
PP.SP 1.4	Estimate Effort and Cost		PS							
	Estimation rationale Project effort estimates Project cost estimates							PS		
PP.SG 2	**Develop a Project Plan**									
PP.SP 2.1	Establish the Budget and Schedule		S							
	Project schedules Schedule dependencies Project budget							S		
PP.SP 2.2	Identify Project Risks	PS								
	Identified risks Risk impacts and probability of occurrence Risk priorities								PS	
PP.SP 2.3	Plan Data Management		PS					PS		
	Data management plan Master list of managed data Data content and format description Lists of data requirements for acquirers and suppliers Privacy requirements Security requirements Security procedures Mechanisms for data retrieval, reproduction, and distribution Schedule for the collection of project data Listing of project data to be collected									
PP.SP 2.4	Plan the Project's Resources		S							
	Work packages WBS task dictionary Staffing requirements based on project size and scope Critical facilities and equipment list Process and workflow definitions and diagrams Project administration requirements list Status reports							S		
PP.SP 2.5	Plan Needed Knowledge and Skills					U				
	Inventory of skill needs Staffing and new hire plans Databases (e.g., skills and training) Training plans									U
PP.SP 2.6	Plan Stakeholder Involvement		S							
	Stakeholder involvement plan							S		
PP.SP 2.7	Establish the Project Plan		S							
	Overall project plan							S		
PP.SG 3	**Obtain Commitment to the Plan**									
PP.SP 3.1	Review Plans That Affect the Project		S							
	Record of the reviews of plans that affect the project								S	
PP.SP 3.2	Reconcile Work and Resource Levels		S							
	Revised methods and corresponding estimating parameters (e.g., better tools, the use of off-the-shelf components) Renegotiated budgets Revised schedules Revised requirements list Renegotiated stakeholder agreements							S		
PP.SP 3.3	Obtain Plan Commitment		S							
	Documented requests for commitments Documented commitments							S		
		1	12	0	0	1	1	10	2	1
Criteria for Conformance rating is 82 (+++)		**7**	**86**	**0**	**0**	**7**	**7**	**71**	**14**	**7**

In Project Planning (PP) Area, PP SG1 Establish estimates; Scrum methods as Sprint Planning Meeting (SPM) addresses the similar practices to all specific practices in

SG1 as estimate the scope of the project (PP.SP 1.1), establish estimates of work product and task attributes (PP.SP 1.2), define project lifecycle phases (PP.SP 1.3), and it is partially similarity to estimate effort and cost (PP.SP 1.4) because it is not follow a formal method or they are not derived from absolute size or complexity as required by CMMI model. On the other hand, the Sprint Backlog (SB) shows the well-matched artifacts to PP.SP 1.2, 1.3, 1.4 (partially) and the Product Backlog (PB) to PP.SP 1.1.

However, it is impossible to know realistic estimates of all effort and cost from the beginning of defining project lifecycle in Scrum. Therefore, we can have relative estimation effort and cost from Planning Poker cards which are a good way to force people to share and voice their opinions as a team consensus without spending too much time on any one topic, each estimator is given a deck of Planning Poker cards. Each card has one of the valid estimates on it, for example: 0, 1, 2, 3, 5, 8, 13, 20, 40 and 100. Team estimates the project effort and cost through Product Backlog (PB) and Sprint Backlog (SB). The more iterative time-boxes increase the precise estimation due to next iterations scope. This would improve accuracy over the pure guess. Scrum's estimation is indicated on 2 levels (Product backlog and Sprint Backlog). The estimation of Product Backlog (PB) is high level estimation which shows less accurate and often ordered by value, risk, priority, and necessity. Then, Sprint Backlog (SB)’s estimation is more accurate by Team based on sprints, and the relative complexity of the required tasks to deliver the Sprint Goal.

PP SG2 Develop a project plan; Establish the budget and schedule (PP.SP 2.1), Plan the project’s resources (PP.SP2.4), Plan stakeholder involvement (PP.SP 2.6), Establish the project plan (PP.SP 2.7) and Plan data management (PP.SP 2.3) are related to Sprint Planning Meeting (SPM) in Scrum practice with small iteration plans evolve throughout the project. Identify project risks (PP.SP 2.2) is partially satisfied via Daily Scrum Meeting (DSM) by monitoring through Burndown Chart (BDC).

In term of develop a project plan, the Sprint Backlog (SB) has a short-iterative plan for delivering the product increment and realizing the sprint goal to fulfill required CMMI's artifacts for instance; project schedules, project budget and project resources (with velocity information from BDC), stakeholder involvement plan. However, the data management plan is also partially satisfied. Nevertheless, the plan needed knowledge and skills (PP.SP 2.5) are unsatisfied by implementing Scrum framework.

PP SG3 Obtain commits to the plan; is satisfied by Scrum via Sprint Planning Meeting (SPM), and then team reconciles work, resource levels and review the iterative plan through Sprint Backlog (SB). Scrum, plan continuously revised at the beginning of each sprint and change accordance to requirements and technologies. Product owner and the team define the revised priorities of the task in SB. The Product Owner can decide to remove some items which are too much loaded work and less priority from the tasks.

Table 3.4 Conformity percentage of Project Monitoring and Control (PMC) with Scrum framework

PMC: Project Monitoring and Control		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	**Description (WP)**	**DSM**	**SPM**	**SRM**	**SR**	**U**	**PB**	**SB**	**BDC**	**U**
PMC.SG 1	**Monitor the Project Against the Plan**									
PMC.SP 1.1	Monitor Project Planning Parameters	S								
	Records of project performance Records of significant deviations Cost performance reports								S	
PMC.SP 1.2	Monitor Commitments		S							
	Records of commitment reviews								S	
PMC.SP 1.3	Monitor Project Risks	S								
	Records of project risk monitoring								S	
PMC.SP 1.4	Monitor Data Management			PS						
	Records of data management								PS	
PMC.SP 1.5	Monitor Stakeholder Involvement			S						
	Records of stakeholder involvement								S	
PMC.SP 1.6	Conduct Progress Reviews	S								
	Documented project review results								S	
PMC.SP 1.7	Conduct Milestone Reviews			S						
	Documented milestone review results								S	
PMC.SG 2	**Manage Corrective Action to Closure**									
PMC.SP 2.1	Analyze Issues	S								
	List of issues requiring corrective actions							S		
PMC.SP 2.2	Take Corrective Action	S								
	Corrective action plans							S		
PMC.SP 2.3	Manage Corrective Actions	S								
	Corrective action results								S	
		6	1	3	0	0	0	2	8	0
Criteria for Conformance rating is 95 (+++)		**60**	**10**	**30**	**0**	**0**	**0**	**20**	**80**	**0**

From Table 3.4, Project Monitoring and Control (PMC) Area, PMC SG1; Monitor the project against the plan; Daily Scrum Meeting (DSM) is supported to monitor project parameters (PMC.SP 1.1), project risks (PMC.SP 1.3) and progress review (PMC.SP 1.6) against the plan including PMC.SG2 also has an iteration of smaller and more frequent releases to analyze, take and manage corrective action (PMC.SP 2.1,2.2,2.3) when team reports all drives against expected quality or performance levels via Burndown chart (BDC) and Sprint Backlog (SB). The Monitor commitment (PMC.SP 1.2) is conduct by Sprint Planning Meeting (SPM). Besides that, Sprint Review Meeting (SRM) is fulfillment to stakeholder involvement (PMC.SP 1.5) to assure that all stakeholders understand the rules and practices defined in SCRUM by Scrum master, and Milestone review (PMC.SP 1.7). However, monitor data management (PMC.SP 1.4) is only partially support through Sprint panning meeting (SPM).

Table 3.5 Conformity percentage of Integrated Project Management (IPM) with Scrum framework

IPM: Integrated Project Management		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	Description (WP)	DSM	SPM	SRM	SR	U	PB	SB	BDC	U
IPM.SG 1	**Use the Project's Defined Process**									
IPM.SP 1.1	Establish the Project's Defined Process		S							
	The project's defined process							S		
IPM.SP 1.2	Use Organizational Process Assets for Planning Project Activities		PS							
	Project estimates Project plans							PS		
IPM.SP 1.3	Establish the Projects Work Environment		PS							
	Equipment and tools for the project Installation, operation, and maintenance manuals for the project work environment User surveys and results Usage, performance, and maintenance records Support services for the project's work environment							PS		
IPM.SP 1.4	Integrate Plans		S							
	Integrated plans							S		
IPM.SP 1.5	Manage the Project Using Integrated Plans		S							
	Work products created by performing the project's defined process Collected measures (i.e., actuals) and status records or reports Revised requirements, plans, and commitments Integrated plans								S	
IPM.SP 1.6	Establish Teams		S							
	Documented shared vision List of members assigned to each team Team charters Periodic team status reports							S		
IPM.SP 1.7	Contribute to Organizational Process Assets				PS					
	Proposed improvements Actual process and product measures collected from the project Documentation (e.g., checklists, lessons learned) Process artifacts associated with tailoring and implementing the OSSP							PS		
IPM.SG 2	**Coordinate and Collaborate with Relevant Stakeholders**									
IPM.SP 2.1	Manage Stakeholder Involvement		S							
	Agendas and schedules for collaborative activities Recommendations for resolving relevant stakeholder issues Documented issues							S		
IPM.SP 2.2	Manage Dependencies		S							
	Defects, issues, and action items from reviews with relevant stakeholders Critical dependencies Commitments to address critical dependencies Status of critical dependencies							S		
IPM.SP 2.3	Resolve Coordination Issues		S							
	Relevant stakeholder coordination issues Status of relevant stakeholder coordination issues							S		
		0	9	0	1	0	0	9	1	0
Criteria for Conformance rating is 85 (+++)		**0**	**90**	**0**	**10**	**0**	**0**	**90**	**10**	**0**

From Table 3.5, Integrated Project Management (IPM) area, Scrum has Sprint Planning Meeting (SPM) to fulfill IPM SG1; Use the Project's Defined Process; establish the Project's Defined Process (IPM.SP 1.1), integrate Plans (IPM.SP 1.4), manage the project using integrated plans (IPM.SP 1.5), establish team (IPM.SP 1.6), and IPM SG2; Coordinate and Collaborate with Relevant Stakeholders; manage stakeholder involvement (IPM.SP 2.1), manage dependencies (IPM.SP 2.2), resolve coordination issues (IPM.SP 2.3) through Sprint Backlog (SB) except using Burndown chart (BDC) for IPM.SP 1.5. In addition, IPM.SP 1.2 (Use Organizational Process Assets for Planning

Project Activities) and IPM.SP 1.3 (Establish the Projects Work Environment) are only partially support by Scrum because the usage and contribution to the organizational process assets (OPA) is not fully implemented in Scrum.

Table 3.6 Conformity percentage of Requirement Management (REQM) with Scrum framework

REQM: Requirements Management		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	**Description (WP)**	**DSM**	**SPM**	**SRM**	**SR**	**U**	**PB**	**SB**	**BDC**	**U**
REQM.SG 1	**Manage Requirements**									
REQM.SP 1.1	Understand Requirements		S							
	Lists of criteria for distinguishing appropriate requirements Criteria for evaluation and acceptance of requirements Results of analyses against criteria A set of approved requirements						S			
REQM.SP 1.2	Obtain Commitment to Requirements		S							
	Requirements impact assessments Documented commitments to requirements and req. changes							S		
REQM.SP 1.3	Manage Requirements Changes	S								
	Requirements change requests Requirements change impact reports Requirements status Requirements database						S			
REQM.SP 1.4	Maintain Bidirectional Traceability of Requirements					U				U
	Requirements traceability matrix Requirements tracking system									
REQM.SP 1.5	Ensure Alignment Between Project Work and Requirements			PS						
	Review project plans, activities, and work products for consistency with requirements and changes made to them. Identify the source of the inconsistency (if any). Identify any changes to the requirements baseline. Initiate any necessary corrective actions.								PS	
		1	2	1	0	1	2	1	1	1
Criteria for Conformance rating is 70 (++)		**20**	**40**	**20**	**0**	**20**	**40**	**20**	**20**	**20**

From Table 3.6, Requirement Management (REQM), there is only one specific goal which is REQM SG1 (manage requirements); Daily Scrum Meeting (DSM) and Sprint Planning Meeting (SPM) are support REQM.SP 1.1 (understand requirements), REQM.SP 1.2 (obtain commitment to requirements) and REQM.SP 1.3 (Manage Requirements Changes) is fully welcomed as "Welcome changing requirements" as mentioned in the Agile principles via Product Backlog (PB) and Sprint Backlog (SB). Sprint Review Meeting (SRM) is partially support to review project plans, activities and work products for consistency with requirements as defined in REQM.SP 1.5 (ensure alignment between project work and requirements) through Burndown Chart (BDC). Furthermore, REQM.SP1.4 (maintain bidirectional traceability of requirements) is not compliance to Scrum framework because Scrum focuses to get requirements just enough to deliver quality working software to customers.

Table 3.7 Conformity percentage of Supplier Agreement Management (SAM) with Scrum framework

SAM: Supplier Agreement Management		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	**Description (WP)**	**DSM**	**SPM**	**SRM**	**SR**	**U**	**PB**	**SB**	**BDC**	**U**
SAM.SG 1	**Establish Supplier Agreements**									
SAM.SP 1.1	Determine Acquisition Type					U				
	List of the acquisition types									U
SAM.SP 1.2	Select Suppliers					U				
	Market studies List of candidate suppliers Preferred supplier list Trade study or other record of evaluation criteria for selection of suppliers Solicitation materials and requirements									U
SAM.SP 1.3	Establish Supplier Agreements					U				
	Statements of work Contracts Memoranda of agreement Licensing agreement									U
SAM.SG 2	**Satisfy Supplier Agreements**									
SAM.SP 2.1	Execute the Supplier Agreement					U				
	Supplier progress reports and performance measures Supplier review materials and reports Action items tracked to closure Product and documentation deliveries									U
SAM.SP 2.2	Accept the Acquired Product					U				
	Acceptance procedures Acceptance reviews or test results Discrepancy reports or corrective action plans									U
SAM.SP 2.3	Ensure Transition of Products					U				
	Transition plans Training reports Support and maintenance reports									U
		0	0	0	0	6	0	0	0	6
Criteria for Conformance rating is 0 (-)		**0**	**0**	**0**	**0**	**100**	**0**	**0**	**0**	**100**

From Table 3.7, Supplier Agreement Management (SAM), there are 2 specific goals which are establish supplier agreements (SAM.SG 1); determine acquisition type (SAM.SP 1.1), select suppliers (SAM.SP 1.2), establish supplier agreements (SAM.SP 1.3) and satisfy supplier agreements (SAM.SG 2); execute the supplier agreement (SAM.SP 2.1), accept the acquired product (SAM.SP 2.2), ensure transition of products (SAM.SP 2.3).

Scrum framework is not addressed by Scrum framework within these two specific goals. Scrum only mention about development processes. However, to complete “CMMI-by-Agile”, it is necessary to establish additional supplier agreement management (SAM) processes and artifacts for implementing within projects in large organizations.

Table 3.8 Conformity percentage of Risk Management (RSKM) with Scrum framework

RSKM: Risk Management		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	**Description (WP)**	**DSM**	**SPM**	**SRM**	**SR**	**U**	**PB**	**SB**	**BDC**	**U**
RSKM.SG 1	**Prepare for Risk Management**									
RSKM.SP 1.1	Determine Risk Sources and Categories	PS								
	Risk source lists (external and internal) Risk categories list							PS		
RSKM.SP 1.2	Define Risk Parameters	S								
	Risk evaluation, categorization, and prioritization criteria Risk management requirements (e.g., control and approval levels)							S		
RSKM.SP 1.3	Establish a Risk Management Strategy		PS							
	Project risk management strategy							PS		
RSKM.SG 2	**Identify and Analyze Risks**									
RSKM.SP 2.1	Identify Risks	PS								
	List of identified risks, including the context, conditions, and consequences of risk occurrence							PS		
RSKM.SP 2.2	Evaluate, Categorize, and Prioritize Risks	PS								
	List of risks and their assigned priority							PS		
RSKM.SG 3	**Mitigate Risks**									
RSKM.SP 3.1	Develop Risk Mitigation Plans					U				
	Documented handling options for each identified risk Risk mitigation plans Contingency plans List of those who are responsible for tracking and addressing									U
RSKM.SP 3.2	Implement Risk Mitigation Plans					U				
	Updated lists of risk status Updated assessments of risk likelihood, consequence, thresholds Updated list of risk handling options Updated list of actions taken to handle risks Risk mitigation plans of risk handling options									U
		4	1	0	0	2	0	5	0	2
Criteria for Conformance rating is 43 (+)		**57**	**14**	**0**	**0**	**29**	**0**	**71**	**0**	**29**

From Table 3.8, Risk Management (RSKM), most of Daily Scrum Meeting (DSM) is partially support RSKM.SG 1 and RSKM.SG 2 specific goals (Prepare for Risk Management-RSKM.SG 1; identify determine risk sources and categories (RSKM.SP 1.1), Analyze Risks-RSKM.SG 2; identify risks (RSKM.SP 2.1) and Evaluate, Categorize, and Prioritize Risks (RSKM.SP 2.2) except in define risk parameters (RSKM.SP 1.2), it is fully support by implementing Sprint Backlog (SB).

Another Sprint Planning Meeting (SPM) is established to respond a risk management strategy (RSKM.SP 1.3). However, develop risk mitigation plans (RSKM.SP 3.1) and implement risk mitigation plans (RSKM.SP 3.2) are unsatisfied Scrum based on RSKM.SG 3 (Identify and Analyze Risks) due to Scrum has no strategies to establish and deployment risk mitigation plan based on historical information.

Table 3.9 Conformity percentage of Quantitative Project Management (QPM) with Scrum framework

QPM: Quantitative Project Management		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	Description (WP)	DSM	SPM	SRM	SR	U	PB	SB	BDC	U
QPM.SG 1	**Prepare for Quantitative Management**									
QPM.SP 1.1	Establish the Project's Objectives		S							
	The project's quality and process performance objectives Assessment of the risk of not achieving the project's objectives							S		
QPM.SP 1.2	Compose the Defined Process					U				
	Criteria used to evaluate alternatives for the project Alternative subprocesses Subprocesses to be included in the project's defined process Assessment of risk of not achieving the project's objectives									U
QPM.SP 1.3	Select Subprocesses and Attributes					U				U
	Criteria used to select subprocesses to achieving the project's objectives Selected subprocesses Attributes that help in predicting future project performance									
QPM.SP 1.4	Select Measures and Analytic Techniques					U				
	Definitions of measures and analytic techniquesin quantitative management Traceability of measures to the project's quality and performance objectives Quality and process performance objectives Process performance baselines and models for use by the project									U
QPM.SG 2	**Quantitatively Manage the Project**									
QPM.SP 2.1	Monitor the Performance of Selected Subprocesses		PS							
	Natural bounds of process performance for each selected subprocess attribute The actions needed to address deficiencies in the process stability								PS	
QPM.SP 2.2	Manage Project Performance		S							
	Predictions to the project's quality and process performance objectives Graphical displays and data tabulations for support quantitative management Assessment of risks of not achieving the project's quality Actions needed to address deficiencies in achieving project objectives								S	
QPM.SP 2.3	Perform Root Cause Analysis					U				
	Subprocess and project performance measurements and analyses recorded Graphical displays of data used to understand project performance and trends Identified root causes and potential actions to take									U
		0	3	0	0	4	0	1	2	4
Criteria for Conformance rating is 36 (-)		**0**	**43**	**0**	**0**	**57**	**0**	**14**	**29**	**57**

From Table 3.9, Quantitative Project Management (QPM), Scrum framework have statistical focus in process performance objectives assessment of the risk (QPM.SP 1.1- establish the project's objective) via sprint backlog (SB) and partially support monitor the performance (QPM.SP 2.1) to monitor process performance and process stability via Burndown chart (BDC). Moreover, manage project performance (QPM.SP2.2) to address deficiencies in achieving project objectives is fully support. However, Scrum is not provide method to compose the define process (QPM.SP 1.2), select sub-process and attributes (QPM.SP 1.3), select sub and attributes (QPM.SP 1.4), and it is also not perform root cause analysis (QPM.SP 2.3).

3.3.2 The relationship between CMMI and Scrum in Process Management Category

From Table 3.10, Organizational Process Focus (OPF) compose three specific goals; Determine process improvement opportunities (OPF.SG 1), Plan and implement process actions (OPF.SG 2), Deploy organizational process assets and incorporate experiences (OPF.SG 3).

Table 3.10 Conformity percentage of Organizational Process Focus (OPF) with Scrum framework

OPF: Organizational Process Focus		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	**Description (WP)**	**DSM**	**SPM**	**SRM**	**SR**	**U**	**PB**	**SB**	**BDC**	**U**
OPF.SG 1	**Determine Process Improvement Opportunities**									
OPF.SP 1.1	Establish Organizational Process Needs					U				
	The organization's process needs and objectives									U
OPF.SP 1.2	Appraise the Organization's Processes					U				
	Plans for the organization's process appraisals Appraisal findings strengths and weaknesses of the organization Improvement recommendations for the organization's processes									U
OPF.SP 1.3	Identify the Organization's Process Improvements				PS					
	Analysis of candidate process improvements Identification of improvements for the organization's processes							PS		
OPF.SG 2	**Plan and Implement Process Actions**									
OPF.SP 2.1	Establish Process Action Plans					U				
	Organization's approved process action plans									U
OPF.SP 2.2	Implement Process Action Plans					U				
	Commitments among process action teams Status and results of implementing process action plans Plans for pilots									U
OPF.SG 3	**Deploy Organizational Process Assets and Incorporate Experiences**									
OPF.SP 3.1	Deploy Organizational Process Assets					U				
	Plans for deploying organizational process assets (OPAs) and changes Training materials for deploying OPAs and changes to them Documentation of changes to organizational process assets Support materials for deploying OPAs and changes to them									U
OPF.SP 3.2	Deploy Standard Processes					U				
	The organization's list of projects and the status of process deployment Guidelines for deploying the organization's set of standard processes Records of tailoring and implementing the OSSP									U
OPF.SP 3.3	Monitor the Implementation					U				
	Results of monitoring process implementation on projects Status and results of process compliance audits Results of reviewing selected process artifacts created									U
OPF.SP 3.4	Incorporate Experiences into Organizational Process Assets					U				
	Process improvement proposals Process lessons learned Measurements of organizational process assets Improvement recommendations for organizational process assets Records of the organization's process improvement activities Information on organizational process assets and improvements to them									U
		0	0	0	1	8	0	1	0	8
Criteria for Conformance rating is 6 (-)		**0**	**0**	**0**	**11**	**89**	**0**	**11**	**0**	**89**

Scrum method is not addressed this process area because it supports to implement for the organizational level. However, OPF.SP 1.3 which is identify the organization's Process Improvements by analysis of candidate process improvements and identification

of improvements is partially support because there is a review process in Sprint Review (SR) practice.

Table 3.11 Conformity percentage of Organizational Training (OT) with Scrum framework

OT: Organizational Training		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	**Description (WP)**	**DSM**	**SPM**	**SRM**	**SR**	**U**	**PB**	**SB**	**BDC**	**U**
OT.SG 1	**Establish an Organizational Training Capability**									
OT.SP 1.1	Establish Strategic Training Needs					U				
	Training needs Assessment analysis									U
OT.SP 1.2	Determine Which Training Needs Are the Responsibility of the Organization					U				
	Common project and support group training needs Training commitments									U
OT.SP 1.3	Establish an Organizational Training Tactical Plan					U				
	Organizational training tactical plan									U
OT.SP 1.4	Establish a Training Capability		PS							
	Training materials and supporting artifacts							PS		
OT.SG 2	**Provide Training**									
OT.SP 2.1	Deliver Training					U				
	Delivered training course									U
OT.SP 2.2	Establish Training Records					U				
	Training records Training updates to the organizational repository									U
OT.SP 2.3	Assess Training Effectiveness					U				
	Testing in the training context Post-training surveys of training participants Surveys of manager satisfaction with post-training effects Assessment mechanisms embedded in courseware									U
		0	1	0	0	6	0	1	0	6
Criteria for Conformance rating is 7 (-)		**0**	**14**	**0**	**0**	**86**	**0**	**14**	**0**	**86**

From Table 3.11, Organizational Training (OT), there are two specific goals which are establishing an organizational training capability (OT.SG 1), and provide training (OT.SG 2). Scrum is not support the purpose to develop the skills and knowledge. Nevertheless, establish a training capability (OT.SP 1.4) is partially support via team work and working software in Sprint Planning Meeting (SPM).

Table 3.12 Conformity percentage of Organizational Process Performance (OPP) with Scrum framework

OPP: Organizational Process Performance		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	**Description (WP)**	**DSM**	**SPM**	**SRM**	**SR**	**U**	**PB**	**SB**	**BDC**	**U**
OPP.SG 1	**Establish Performance Baselines and Models**									
OPP.SP 1.1	Establish Quality and Process Performance Objectives				PS					
	Organization's quality and process performance objectives								PS	
OPP.SP 1.2	Select Processes					U				
	List of processes identified for process performance analyses									U
OPP.SP 1.3	Establish Process Performance Measures					U				
	Definitions of selected measures of process performance with rationale									U
OPP.SP 1.4	Analyze Process Performance and Establish Process Performance Baselines					U				
	Analysis of process performance data Baseline data on the organization's process performance									U
OPP.SP 1.5	Establish Process Performance Models					U				
	Process performance models									U
		0	0	0	1	4	0	0	1	4
Criteria for Conformance rating is 10 (-)		**0**	**0**	**0**	**20**	**80**	**0**	**0**	**20**	**80**

From Table 3.12, Organizational Process Performance (OPP), establish performance baselines and models (OPP.SG 1); Scrum framework is not compliance to select processes (OPP.SP 1.2), establish process performance measures (OPP.SP 1.3), analyze Process performance and establish process performance baselines (OPP.SP 1.4), establish process performance models (OPP.SP 1.5) because OPP is process oriented and it applies to the organizational level while Scrum focuses on the Agile Manifesto as "Individuals and interactions over processes and tools" at project level.

However, OPP.SP 1.1 is partially support to Scrum which has measures for instance; time, resource and progress to establish quality and process performance objectives (OPP.SP 1.1) in Sprint Review (SR) practice via Burndown Chart (BDC).

Table 3.13 Conformity percentage of Organizational Performance Management (OPM) with Scrum framework

OPM: Organizational Performance Management		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	**Description (WP)**	**DSM**	**SPM**	**SRM**	**SR**	**U**	**PB**	**SB**	**BDC**	**U**
OPM.SG 1	**Manage Business Performance**									
OPM.SP 1.1	Maintain Business Objectives				PS					
	Revised business objectives Revised quality and process performance objectives Senior management approval of revised business and quality objectives Communication of all revised objectives Updated process performance measures								PS	
OPM.SP 1.2	Analyze Process Performance Data				PS					
	Analysis of current capability vs. business objectives Process performance shortfalls Risks associated with meeting business objectives								PS	
OPM.SP 1.3	Identify Potential Areas for Improvement				PS					
	Potential areas for improvement								PS	
OPM.SG 2	**Select Improvements**									
OPM.SP 2.1	Elicit Suggested Improvements				S					
	Suggested incremental improvements Suggested innovative improvements								S	
OPM.SP 2.2	Analyze Suggested Improvements				S					
	Suggested improvement proposals Selected improvements to be validated								S	
OPM.SP 2.3	Validate Improvements				PS					
	Validation plans Validation evaluation reports Documented lessons learned from validation								PS	
OPM.SP 2.4	Select and Implement Improvements for Deployment				PS					
	Improvements selected for deployment Updated process documentation and training								PS	
OPM.SG 3	**Deploy Improvements**									
OPM.SP 3.1	Plan the Deployment				PS					
	Deployment plans for selected improvements							PS		
OPM.SP 3.2	Manage the Deployment					U				
	Updated training materials (to reflect deployed improvements) Documented results of improvement deployment activities Revised improvement measures, objectives, priorities, and deployment plans									U
OPM.SP 3.3	Evaluate Improvement Effects					U				
	Documented measures of the effects resulting from deployed improvements									U
		0	0	0	8	2	0	1	7	2
Criteria for Conformance rating is 50 (+)		**0**	**0**	**0**	**80**	**20**	**0**	**10**	**70**	**20**

From Table 3.13, Organizational Performance Management (OPM), Scrum framework is mostly partially compliances OPM.SG 1 (manage business performance), OPM.SG 2 (select improvements) and OPM.SG 3 (deploy improvements) except in OPM.SP 3.2 (manage the deployment) and OPM.SP 3.2 (evaluate Improvement effects) is not compliance in Scrum implementation.

However, OPM.SP 2.1 (elicit suggested improvements) and OPM.SP 2.2 (analyze suggested improvements) perform fully support to elicit and analyze suggested improvements. The OPM supports another Agile manifesto which is “The organization’s performance to meet its business objectives” in Sprint Retrospective (SR) based on Burndown Chart (BDC) information. SR is continually improving processes in sprint which is provides an opportunity for team members to reflect, tune and identify practices that possible to improve in the next sprint.

Table 3.14 Conformity percentage of Organizational Process Definition (OPD) with Scrum framework

OPD: Organizational Process Definition		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	**Description (WP)**	**DSM**	**SPM**	**SRM**	**SR**	**U**	**PB**	**SB**	**BDC**	**U**
OPD.SG 1	**Establish Organizational Process Assets**									
OPD.SP 1.1	Establish Standard Processes					U				
	Organization's set of standard processes									U
OPD.SP 1.2	Establish Lifecycle Model Descriptions		PS							
	Descriptions of lifecycle models							PS		
OPD.SP 1.3	Establish Tailoring Criteria and Guidelines					U				
	Tailoring guidelines for the organization's set of standard processes									U
OPD.SP 1.4	Establish the Organization's Measurement Repository		PS							
	Definition of the common set of product and process measures for the OSSP Design of the organization's measurement repository Organization's measurement repository (i.e., the repository structure) Organization's measurement data								PS	
OPD.SP 1.5	Establish the Organization's Process Asset Library					U				
	Design of the organization's process asset library The organization's process asset library Selected items to be included in the organization's process asset library The catalog of items in the organization's process asset library									U
OPD.SP 1.6	Establish Work Environment Standards					U				
	Work environment standards									U
OPD.SP 1.7	Establish Rules and Guidelines for Teams					U				
	Rules and guidelines for structuring and forming teams Operating rules for teams									U
		0	2	0	0	5	0	1	1	5
Criteria for Conformance rating is 14 (-)		**0**	**29**	**0**	**0**	**71**	**0**	**14**	**14**	**71**

From Table 3.14, Organizational Process Definition (OPD), there is only one goal which is established organizational process assets (OPD.SG 1). Scrum framework is not support the goal of OPD, in term of establish standard processes (OPD.SP 1.1), establish tailoring criteria and guidelines (OPD.SP 1.3), establish the organization's process asset library (OPD.SP 1.5), establish work environment standards (OPD.SP 1.6) and establish rules and guidelines for teams (OPD.SP 1.7).

Nevertheless, establish lifecycle model descriptions (OPD.SP 1.2) and establish the organization's measurement repository (OPD.SP 1.4) are partially satisfied via Sprint Plan Meeting (SPM).

3.3.3 The relationship between CMMI and Scrum in Engineering Category

From Table 3.15, Requirement Development (RD), develop customer requirements (RD.SG 1) is compose of two specific practices which are elicit needs (RD.SP 1.1) and transform stakeholder needs into customer requirements (RD.SP 1.2). Scrum is satisfied by Sprint Planning Meeting (SPM) through Product Backlog (PB). RD.SG 2 (develop product requirements) and RD.SG 3 (analyze and validate requirements) are mostly partially support by SPM and Daily Scrum Meeting (DSM) which are use Sprint Backlog (SB).

Table 3.15 Conformity percentage of Requirement Development (RD) with Scrum framework

RD: Requirements Development		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	Description (WP)	DSM	SPM	SRM	SR	U	PB	SB	BDC	U
RD.SG 1	**Develop Customer Requirements**									
RD.SP 1.1	Elicit Needs		S							
	Results of requirements elicitation activities						S			
RD.SP 1.2	Transform Stakeholder Needs into Customer Requirements		S							
	Prioritized customer requirements Customer constraints on the conduct of verification Customer constraints on the conduct of validation						S			
RD.SG 2	**Develop Product Requirements**									
RD.SP 2.1	Establish Product and Product Component Requirements		S							
	Derived requirements Product requirements Product component requirements Architectural requirements							S		
RD.SP 2.2	Allocate Product Component Requirements		PS							
	Requirement allocation sheets Provisional requirement allocations Design constraints Derived requirements Relationships among derived requirements							PS		
RD.SP 2.3	Identify Interface Requirements		PS							
	Interface requirements							PS		
RD.SG 3	**Analyze and Validate Requirements**									
RD.SP 3.1	Establish Operational Concepts and Scenarios		PS							
	Operational concept Product or product component Disposal concepts Use cases Timeline scenarios New requirements							PS		
RD.SP 3.2	Establish a Definition of Required Functionality and Quality Attributes					U				
	Definition of required functionality and quality attributes Functional architecture Activity diagrams and use cases Object oriented analysis with services or methods identified Architecturally significant quality attribute requirements									U
RD.SP 3.3	Analyze Requirements	PS								
	Requirements defects reports Proposed requirements changes to resolve defects Key requirements Technical performance measures							PS		
RD.SP 3.4	Analyze Requirements to Achieve Balance	PS								
	Assessment of risks related to requirements							PS		
RD.SP 3.5	Validate Requirements		PS							
	Record of analysis methods and results							PS		
		2	7	0	0	1	2	7	0	1
Criteria for Conformance rating is 60 (++)		**20**	**70**	**0**	**0**	**10**	**20**	**70**	**0**	**10**

However, establish a definition of required functionality and quality attributes is not support by Scrum because Agile principles emphasize building working software that people can get hands on quickly, instead of spending a lot of time writing specifications up front but unit test or test driven development in small iterative can help to clearly understand the customer requirements and also validate them.

Table 3.16 Conformity percentage of Technical Solution (TS)
with Scrum framework

TS: Technical Solution		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	Description (WP)	DSM	SPM	SRM	SR	U	PB	SB	BDC	U
TS.SG 1	**Select Product Component Solutions**									
TS.SP 1.1	Develop Alternative Solutions and Selection Criteria		S							
	Alternative solution screening criteria									
	Evaluation reports of new technologies									
	Alternative solutions							S		
	Selection criteria for final selection									
	Evaluation reports of COTS product									
TS.SP 1.2	Select Product Component Solutions		PS							
	Product component selection decisions and rationale									
	Documented relationships between requirements and product components							PS		
	Documented solutions, evaluations, and rationale									
TS.SG 2	**Develop the Design**									
TS.SP 2.1	Design the Product or Product Component		S							
	Product architecture									
	Product component designs							S		
TS.SP 2.2	Establish a Technical Data Package		PS							
	Technical data package							PS		
TS.SP 2.3	Design Interfaces Using Criteria		PS							
	Interface design specifications									
	Interface control documents									
	Interface specification criteria							PS		
	Rationale for selected interface design									
TS.SP 2.4	Perform Make, Buy, or Reuse Analyses		PS							
	Criteria for design and product component reuse									
	Make-or-buy analyses							PS		
	Guidelines for choosing COTS product components									
TS.SG 3	**Implement the Product Design**									
TS.SP 3.1	Implement the Design		S							
	Implemented design							S		
TS.SP 3.2	Develop Product Support Documentation					U				
	End-user training materials									
	User's manual									
	Operator's manual									U
	Maintenance manual									
	Online help									
		0	7	0	0	1	0	7	0	1
Criteria for Conformance rating is 63 (++)		**0**	**88**	**0**	**0**	**13**	**0**	**88**	**0**	**13**

From Table 3.16, Technical Solution (TS), Scrum mostly supports this process area. Develop alternative solutions and selection criteria (TS.SP 1.1), design the product or product component (TS.SP 2.1) and implement the design (TS.SP 3.1) are fully support via Sprint Planning Meeting (SPM) and Sprint Backlog (SB).

Additionally, to select product component solutions (TS.SP1.2), establish a technical data package (TS.SP2.2), design interfaces using criteria (TS.SP2.3), perform make, buy, or reuse analyses (TS.SP2.4) are partially satisfied. And, lastly, develop product support documentation (TS.SP3.2) is unsatisfied by implementing Scrum framework because Scrum has small iterative development to respond customer feedback with more flexible and accept changes based on Agile practices for instance code re-factoring, pair programming, coding standard, or test driven development. However, the documentations for product development or product design are not including in Scrum's continuous improvements by working software concept.

Table 3.17 Conformity percentage of Product Integration (PI) with Scrum framework

PI: Product Integration		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	**Description (WP)**	**DSM**	**SPM**	**SRM**	**SR**	**U**	**PB**	**SB**	**BDC**	**U**
PI SG 1	**Prepare for Product Integration**									
PI.SP 1.1	Establish an Integration Strategy		S							
	Product integration strategy Rationale for selecting or rejecting alternative product integration strategies						S			
PI.SP 1.2	Establish the Product Integration Environment		PS							
	Verified environment for product integration Support documentation for the product integration environment						PS			
PI.SP 1.3	Establish Product Integration Procedures and Criteria		PS							
	Product integration procedures Product integration criteria						PS			
PI SG 2	**Ensure Interface Compatibility**									
PI.SP 2.1	Review Interface Descriptions for Completeness		S							
	Categories of interfaces List of interfaces per category Mapping of the interfaces to the product components and the PI environment						S			
PI.SP 2.2	Manage Interfaces		PS							
	Table of relationships among the product components and the environment Table of relationships among the different product components List of agreed-to interfaces defined for each pair of product components Reports from the interface control working group meetings Action items for updating interfaces Application program interface (API) Updated interface description or agreement						PS			
PI SG 3	**Assemble Product Components and Deliver the Product**									
PI.SP 3.1	Confirm Readiness of Product Components for Integration		S							
	Acceptance documents for the received product components Delivery receipts Checked packing lists Exception reports Waivers							S		
PI.SP 3.2	Assemble Product Components		S							
	Assembled product or product components							S		
PI.SP 3.3	Evaluate Assembled Product Components		PS							
	Exception reports Interface evaluation reports Product integration summary reports							PS		
PI.SP 3.4	Package and Deliver the Product or Product Component		PS							
	Packaged product or product components Delivery documentation							PS		
		0	9	0	0	0	5	4	0	0
Criteria for Conformance rating is 72 (++)		**0**	**100**	**0**	**0**	**0**	**56**	**44**	**0**	**0**

From Table 3.17, Product Integration (PI), in general, Scrum framework supports PI process area for example; establish an integration strategy (PI.SP 1.1), review interface descriptions for completeness (PI.SP 2.1), confirm readiness of product components for integration (PI.SP 3.1) and assemble product components (PI.SP 3.2). Some specific practices are partially support which are establish the product integration environment (PI.SP 1.2), establish product integration procedures and criteria (PI.SP 1.3), manage interfaces (PI.SP 2.2), evaluate assembled product components (PI.SP 3.3) and package and deliver the product or product component (PI.SP 3.4). There is no unsatisfied gap between Scrum practice and PI process area because to integrated product is a main objective of working software over comprehensive documentation.

Table 3.18 Conformity percentage of Verification (VER) with Scrum framework

VER: Verification		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	**Description (WP)**	**DSM**	**SPM**	**SRM**	**SR**	**U**	**PB**	**SB**	**BDC**	**U**
VER.SG 1	**Prepare for Verification**									
VER.SP 1.1	Establish Standard Processes		PS							
	Lists of work products selected for verification Verification methods for each selected work product							PS		
VER.SP 1.2	Establish the Verification Environment		S							
	Identify verification environment requirements. Identify verification resources for reuse or modification. Identify verification equipment and tools. Acquire verification support equipment and an environment							S		
VER.SP 1.3	Establish Verification Procedures and Criteria		PS							
	Verification procedures Verification criteria							PS		
VER.SG 2	**Perform Peer Reviews**									
VER.SP 2.1	Prepare for Peer Reviews		S							
	Peer review schedule Peer review checklist Entry and exit criteria for work products Criteria for requiring another peer review Peer review training material Selected work products to be reviewed							S		
VER.SP 2.2	Conduct Peer Reviews		S							
	Peer review results Peer review issues Peer review data							S		
VER.SP 2.3	Analyze Peer Review Data		PS							
	Peer review data Peer review action items							PS		
VER.SG 3	**Verify Selected Work Products**									
VER.SP 3.1	Perform Verification		PS							
	Verification results Verification reports Demonstrations As-run procedures log							PS		
VER.SP 3.2	Analyze Verification Results			PS						
	Analysis report Trouble reports Change requests for verification methods, criteria, and the environment							PS		
		0	7	1	0	0	0	8	0	0
Criteria for Conformance rating is 69 (++)		**0**	**88**	**13**	**0**	**0**	**0**	**100**	**0**	**0**

From Table 3.18, Verification (VER), Scrum framework based on Agile practice likes unit testing or test driven development by team are compliance with Establish the Verification Environment (VER.SP 1.2), and partially support to establish standard processes (VER.SP 1.1), establish verification procedures and criteria (VER.SP 1.3). Perform peer reviews (VER.SG2) and verify selected work products (VER.SG3) are supported via Sprint Review Meeting (SRM) through Sprint Backlog (SB).

Table 3.19 Conformity percentage of Validation (VAL) with Scrum framework

VAL: Validation		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	**Description (WP)**	**DSM**	**SPM**	**SRM**	**SR**	**U**	**PB**	**SB**	**BDC**	**U**
VAL.SG 1	**Prepare for Validation**									
VAL.SP 1.1	Select Products for Validation		S							
	Lists of products and product components selected for validation Validation methods for each product or product component Requirements for performing validation for each product Validation constraints for each product or product component						S			
VAL.SP 1.2	Establish the Validation Environment		PS							
	Validation environment						PS			
VAL.SP 1.3	Establish Validation Procedures and Criteria		PS							
	Validation procedures Validation criteria Test and evaluation procedures for maintenance, training						PS			
VAL.SG 2	**Validate Product or Product Components**									
VAL.SP 2.1	Perform Validation			S						
	Validation reports Validation results Validation cross-reference matrix As-run procedures log Operational demonstrations							S		
VAL.SP 2.2	Analyze Validation Results			PS						
	Validation deficiency reports Validation issues Procedure change request							PS		
		0	3	2	0	0	3	2	0	0
Criteria for Conformance rating is 70 (++)		**0**	**60**	**40**	**0**	**0**	**60**	**40**	**0**	**0**

From Table 3.19, Validation (VAL), Scrum framework supports the spirit of customer collaboration so that in Sprint Planning Meeting (SPM) is compliance VAL.SG 1 (prepare for validation) through Product Backlog (PB) and VAL.SG 2 (validate product or product components) is compliance Sprint Review Meeting (SRM) via Sprint Backlog (SB) by demonstrating of the product to product owner or the stakeholders to approve the right product.

3.3.4 The relationship between CMMI and Scrum in Support Category

From Table 3.20, Configuration Management (CM), Scrum has Dairy Scrum Meeting (DSM) to support track change requests (CM.SP 2.1) and a part of control configuration items (CM.SP 2.2). In additional, Sprint Planning Meeting (SPM) is partially support identify configuration items (CM.SP 1.1), establish configuration management records (CM.SP 3.1), and perform configuration audits (CM.SP 3.2) by deploying Sprint Backlog (SB) and Burndown Chart (BDC).

Table 3.20 Conformity percentage of Configuration Management (CM) with Scrum framework

CM: Configuration Management		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	**Description (WP)**	**DSM**	**SPM**	**SRM**	**SR**	**U**	**PB**	**SB**	**BDC**	**U**
CM.SG 1	**Establish Baselines**									
CM.SP 1.1	Identify Configuration Items		PS							
	Identified configuration items								PS	
CM.SP 1.2	Establish a Configuration Management System					U				
	Configuration management system with controlled work products									
	Configuration management system access control procedures									U
	Change request database									
CM.SP 1.3	Create or Release Baselines					U				
	Baselines									
	Description of baselines									U
CM.SG 2	**Track and Control Changes**									
CM.SP 2.1	Track Change Requests	S								
	Change requests							S		
CM.SP 2.2	Control Configuration Items	PS								
	Revision history of configuration items									
	Archives of baselines								PS	
CM.SG 3	**Establish Integrity**									
CM.SP 3.1	Establish Configuration Management Records		PS							
	Revision history of configuration items									
	Change log									
	Copy of the change requests							PS		
	Status of configuration items									
	Differences between baselines									
CM.SP 3.2	Perform Configuration Audits		PS							
	Configuration audit results									
	Action items							PS		
		2	3	0	0	2	0	3	2	2
Criteria for Conformance rating is 43 (++)		**29**	**43**	**0**	**0**	**29**	**0**	**43**	**29**	**29**

However, while implementing Scrum, it is automated configuration these items; code, design, tests due to the frequently changing environment.

Table 3.21 Conformity percentage of Process and Product Quality Assurance (PPQA) with Scrum framework

PPQA: Process and Product Quality Assurance		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	**Description (WP)**	**DSM**	**SPM**	**SRM**	**SR**	**U**	**PB**	**SB**	**BDC**	**U**
PPQA.SG 1	**Objectively Evaluate Processes and Work Products**									
PPQA.SP 1.1	Objectively Evaluate Processes		PS							
	Evaluation reports Noncompliance reports Corrective actions							PS		
PPQA.SP 1.2	Objectively Evaluate Work Products				PS					
	Evaluation reports Noncompliance reports Corrective actions							PS		
PPQA.SG 2	**Provide Objective Insight**									
PPQA.SP 2.1	Communicate and Resolve Noncompliance Issues				PS					
	Corrective action reports Evaluation reports Quality trends							PS		
PPQA.SP 2.2	Establish Records					U				
	Evaluation logs Quality assurance reports Status reports of corrective actions Reports of quality trends									U
		0	1	0	2	1	0	3	0	1
Criteria for Conformance rating is 38 (-)		**0**	**25**	**0**	**50**	**25**	**0**	**75**	**0**	**25**

From Table 3.21, Process and Product Quality Assurance (PPQA), This process area objectively evaluate processes and work products (PPQA.SG 1) and provide objective insight (PPQA.SG 1) against the standards process descriptions and procedures instead of the end product so that, this PPQA process area is not completely and automatically applicable to Scrum framework.

Therefore, some Scrum's activities likes refinements in Sprint Planning Meeting (SPM) or Sprint Retrospective (SR) are partially compliance with objectively evaluate processes (PPQA.SP 1.1), Objectively evaluate work products (PPQA.SP 1.2) and communicate and resolve noncompliance issues (PPQA.SP 2.1).

Table 3.22 Conformity percentage of Measurement and Analysis (MA) with Scrum framework

MA: Measurement and Analysis		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	**Description (WP)**	**DSM**	**SPM**	**SRM**	**SR**	**U**	**PB**	**SB**	**BDC**	**U**
MA.SG 1	**Align Measurement and Analysis Activities**									
MA.SP 1.1	Establish Measurement Objectives		PS							
	Measurement objectives								PS	
MA.SP 1.2	Specify Measures		PS							
	Specifications of base and derived measures								PS	
MA.SP 1.3	Specify Data Collection and Storage Procedures					U				
	Data collection and storage procedures Data collection tools									U
MA.SP 1.4	Specify Analysis Procedures		PS							
	Analysis specifications and procedures Data analysis tools								PS	
MA.SG 2	**Provide Measurement Results**									
MA.SP 2.1	Obtain Measurement Data	S								
	Base and derived measurement data sets Results of data integrity tests								S	
MA.SP 2.2	Analyze Measurement Data	PS								
	Analysis results and draft reports								PS	
MA.SP 2.3	Store Data and Results					U				
	Stored data inventory									U
MA.SP 2.4	Communicate Results	S								
	Delivered reports and related analysis results Contextual information to help interpret analysis results								S	
		3	3	0	0	2	0	0	6	2
Criteria for Conformance rating is 50 (+)		**38**	**38**	**0**	**0**	**25**	**0**	**0**	**75**	**25**

From Table 3.22, Measurement and Analysis (MA), Scrum framework is perform Dairy Scrum Meeting (DSM) and Sprint Planning Meeting (SPM) via Burndown Chart (BDC) which is align "Frequent intensive communication between team members" to support align measurement and analysis activities (MA.SG 1) and provide measurement results (MA.SG 2) for developing and sustaining a measurement capability.

However, specify data collection and storage procedures (MA.SP 1.3) and store data and results (MA.SP 2.3) are unsatisfied based on MA specific practices' objectives to collect and store data inventory.

Table 3.23 Conformity percentage of Decision Analysis and Resolution (DAR) with Scrum framework

DAR: Decision Analysis and Resolution		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	Description (WP)	DSM	SPM	SRM	SR	U	PB	SB	BDC	U
DAR.SG 1	**Evaluate Alternatives**									
DAR.SP 1.1	Establish Guidelines for Decision Analysis					U				
	Guidelines for when to apply a formal evaluation process									U
DAR.SP 1.2	Establish Evaluation Criteria	PS								
	Documented evaluation criteria Rankings of criteria importance							PS		
DAR.SP 1.3	Identify Alternative Solutions	PS								
	Identified alternatives							PS		
DAR.SP 1.4	Select Evaluation Methods					U				
	Selected evaluation methods									U
DAR.SP 1.5	Evaluate Alternatives Solutions					U				
	Evaluation results									U
DAR.SP 1.6	Select Solutions	PS								
	Recommended solutions to address significant issues							PS		
		3	0	0	0	3	0	3	0	3
Criteria for Conformance rating is 25 (-)		**50**	**0**	**0**	**0**	**50**	**0**	**50**	**0**	**50**

From Table 3.23, Decision Analysis and Resolution (DAR), there is only one specific goal which is evaluate alternatives (DAR.SG 1), it is purposely to analyze possible decisions using a formal evaluation process that against established criteria. Scrum framework is not support establish guidelines for decision analysis (DAR.SP 1.1), Select Evaluation Methods (DAR.SP 1.4) and Evaluate Alternatives Solutions (DAR.SP 1.5). It is also has conflict the spirit of “Individuals and interactions over processes and tools”.

However, Dairy Scrum Meeting (DSM) is partially support to establish evaluation criteria (DAR.SP 1.2), identify alternative solutions (DAR.SP 1.3), and select solutions (DAR.SP 1.6) through Sprint Backlog (SB) because it is support another spirit of Agile likes "collaborative process".

Table 3.24 Conformity percentage of Causal Analysis and Resolution (CAR) with Scrum framework

CAR: Causal Analysis and Resolution		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
SG/SP	**Description (WP)**	**DSM**	**SPM**	**SRM**	**SR**	**U**	**PB**	**SB**	**BDC**	**U**
CAR.SG 1	**Determine Causes of Defects**									
CAR.SP 1.1	Select Outcomes for Analysis				PS					
	Data to be used in the initial analysis Initial analysis results data Outcomes selected for further analysis								PS	
CAR.SP 1.2	Analyze Causes				PS					
	Root cause analysis results Action proposal								PS	
CAR.SG 2	**Address Causes of Selected Outcomes**									
CAR.SP 2.1	Implement Action Proposals		PS							
	Action proposals selected for implementation Action plans							PS		
CAR.SP 2.2	Evaluate the Effect of Implemented Actions					U				
	Analysis of process performance and change in process performance									U
CAR.SP 2.3	Record Causal Analysis Data					U				
	Causal analysis and resolution records Organizational improvement proposals									U
		0	1	0	2	2	0	1	2	2
Criteria for Conformance rating is 30 (+)		**0**	**20**	**0**	**40**	**40**	**0**	**20**	**40**	**40**

From Table 3.24, Causal Analysis and Resolution (CAR), determine causes of defects (CAR.SG 1) by select outcomes for analysis (CAR.SP 1.1), analyze causes (CAR.SP 1.2) and implement action proposals (CAR.SP 2.1) are perform partially though Sprint Retrospective (SR) practice which team members reflect on the questions of what we did well in the last sprint based on Burndown Chart (BDC) information and what could be improved in the next sprint via Sprint Planning Meeting (SPM) by using Sprint Backlog (SB) to implement action to resolve the cause.

Therefore, there are no causal analysis and resolution records, there is only informally identify and evaluate alternatives decisions in Scrum framework.

3.3.5 The relationship between CMMI and Scrum in Generic Goal

From Table 3.25, Generic Goal (GG), Sprint Planning Meeting (SPM) supports various generic goals likes; plan the process (GG.GP 2.2), provide resources (GG.GP 2.3), assign responsibility (GG.GP 2.4), identify and involve relevant stakeholders (GG.GP 2.7), and establish a defined process (GG.GP 3.1) through Sprint Backlog (SB). Sprint Review Meting (SRM) is also support control work products (GG.GP 2.6), objectively evaluate adherence (GG.GP 2.9) via Product Backlog (PB) and Burndown Chart (BDC).

Table 3.25 Conformity percentage of Generic Goal (GG) with Scrum framework

GG: Generic Goal		Scrum Practice (S/PS/U)					Scrum Artifact (S/PS/U)			
GG/GP	Description (WP)	DSM	SPM	SRM	SR	U	PB	SB	BDC	U
GG 1	Achieve Specific Goals									
GG.GP 1.1	Perform Specific Practices									
	The work products and deliver the services based on each PA									
GG 2	Institutionalize a Managed Process									
GG.GP 2.1	Establish an Organizational Policy					U				
	Policy based on each PA									U
GG.GP 2.2	Plan the Process		PS							
	Process based on each PA							PS		
GG.GP 2.3	Provide Resources		S							
	DBMS, Tools, Statistical packages							S		
GG.GP 2.4	Assign Responsibility		PS							
	List of assigned responsibility and authority for performing the process							PS		
GG.GP 2.5	Train People					U				
	Method of providing training									U
GG.GP 2.6	Control Work Products			S						
	work products placed under control ex. Action proposals, Action plans, List of products						S			
GG.GP 2.7	Identify and Involve Relevant Stakeholders		PS							
	Activities for stakeholder involvement ex. Defining the team structure for the project							PS		
GG.GP 2.8	Monitor and Control the Process				S					
	Measures and work products used in monitoring and controlling ex. Schedule with status, Number of open and closed corrective actions								S	
GG.GP 2.9	Objectively Evaluate Adherence			PS						
	Work products and activities reviewed ex. Process improvement plans, Monitoring project progress and performance against the project plan								PS	
GG.GP 2.10	Review Status with Higher Level Management				PS					
	Presentation topics ex. Status of improvements being developed by process action teams, Results from validation activities							PS		
GG 3	Institutionalize a Defined Process									
GG.GP 3.1	Establish a Defined Process		PS							
	The organization's set of standard processes, Tailoring process							PS		
GG.GP 3.2	Collect Process Related Experiences				PS					
	Process related experiences ex. Action proposals, Data analysis reports, Number of product defects								PS	
		0	5	2	3	2	1	6	3	2
Criteria for Conformance rating is 54 (+)		**0**	**42**	**17**	**25**	**17**	**8**	**50**	**25**	**17**

Finally, Sprint Retrospective (SR) is compliance to monitor and control the process (GG.GP 2.8), review status with higher level management (GG.GP 2.10) and collect process related experiences (GG.GP 3.2). However, establish an organizational policy (GG.GP 2.1) and train people (GG.GP 2.5) are not support Scrum framework.

3.3.6 The summary of relationship between CMMI and Scrum

The following Table 3.26 shows CMMI 22 process areas (PA) in capability level to four process categories; Project Management, Process Management, Engineering and Support. The number shows conformity percentage which Scrum framework support SG/SP in each process area. Regarding to this information, the most top-three conformity are Project Monitoring and Control (PMC), Integrated Project Management (IPM) and Project Planning (PP) which shows 95%, 85% and 82%, respectively.

Project Planning (PP), Project Monitoring and Control (PMC) and Integrated Project Management (IPM) are shown significantly conformance value to Scrum practices and artifacts along with Requirement Management (REQM) and Engineering category. However, Process Management category likes; Organizational Training (OT), Organizational process Focus (OPF) show the lowest conformance value especially, Supplier Agreement Management (SAM) is not addressed at all by Scrum.

Table 3.26 Conformity percentage of Process Area (PA) category with Scrum framework

Process Area	ML	Process Area Category	Conformance (%)	Conformance rating
PP: Project Planning	2	Project Management	82	+++
PMC: Project Monitoring and Control	2	Project Management	95	+++
IPM: Integrated Project Management	3	Project Management	85	+++
REQM: Requirements Management	2	Project Management	70	++
SAM: Supplier Agreement Management	2	Project Management	0	-
RSKM: Risk Management	3	Project Management	43	+
QPM: Quantitative Project Management	4	Project Management	36	-
OPF: Organizational Process Focus	3	Process Management	6	-
OT: Organizational Training	3	Process Management	7	-
OPP: Organizational Process Performance	4	Process Management	10	-
OPM: Organizational Performance Management	5	Process Management	50	+
OPD: Organizational Process Definition	3	Process Management	14	-
RD: Requirements Development	3	Engineering	60	++
TS: Technical Solution	3	Engineering	63	++
PI: Product Integration	3	Engineering	72	++
VER: Verification	3	Engineering	69	++
VAL: Validation	3	Engineering	70	++
CM: Configuration Management	2	Support	43	+
PPQA: Process and Product Quality Assurance	2	Support	38	-
MA: Measurement and Analysis	2	Support	50	+
DAR: Decision Analysis and Resolution	3	Support	25	-
CAR: Causal Analysis and Resolution	5	Support	30	-

In term of average value from each category, Scrum performs to compliance process area in average consecutively; Project Management category 68%, Engineering category 67%, Support category 37% and Process Management category 17%. In overall perspective number of all categories shows 48% in total.

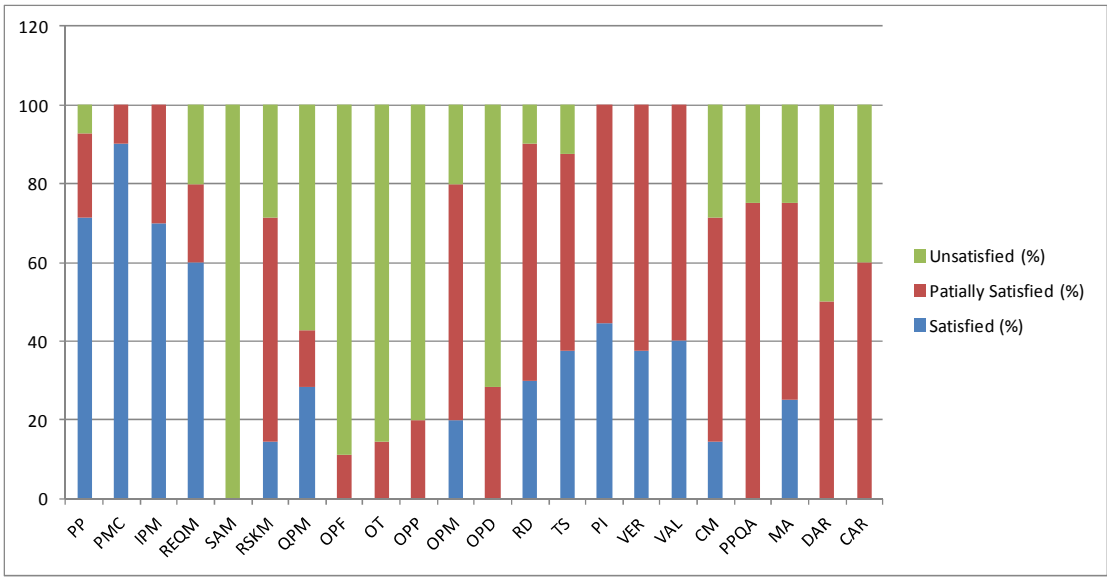


Figure 3.4 The conformity satisfied percentage of CMMI process area with Scrum framework

Figure 3.4 shows the proportion of conformity satisfied percentage from each process area with Scrum framework. The blue color presents satisfied percentage, brown color presents partially satisfied percentage and green presents unsatisfied percentage. Project Planning (PP), Project Monitoring and Control (PMC) and Integrated Project Management (IPM) are the most process areas which perform significantly conformance value to Scrum practices and artifacts. The Process Management category shows majority in unsatisfied values, Scrum is not address Supplier Agreement Management (SAM) in every aspect.

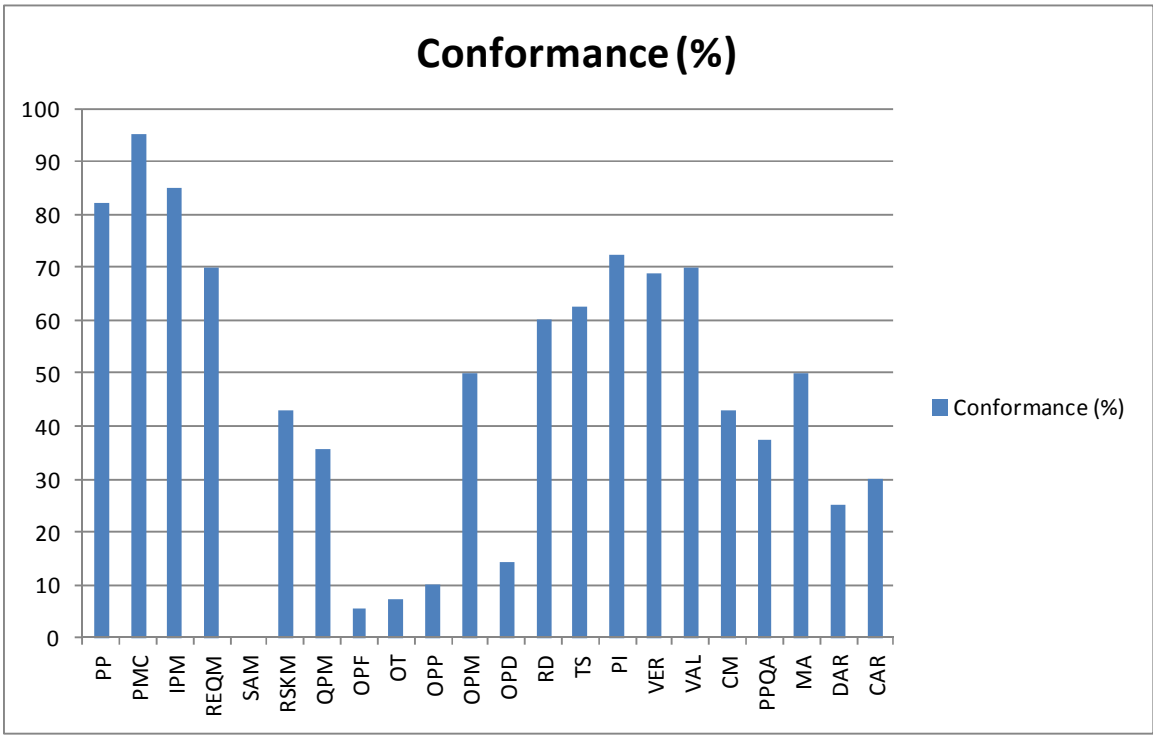


Figure 3.5 The conformity percentage of CMMI process area with Scrum framework

In order to show the represented value of the conformity, we propose the formulation as follow; % Conformance of Process Area = (% Satisfied) + (½ *(% Partially Satisfied)). Finally, the final value of % conformity of CMMI and Scrum are presented in Figure 3.5.

Table 3.27 Conformity percentage in maturity level classification with Scrum framework

Process Area	ML	Process Area Category	Conformance (%)	Conformance rating
PP: Project Planning	2	Project Management	82	+++
PMC: Project Monitoring and Control	2	Project Management	95	+++
REQM: Requirements Management	2	Project Management	70	++
SAM: Supplier Agreement Management	2	Project Management	0	-
CM: Configuration Management	2	Support	43	+
PPQA: Process and Product Quality Assurance	2	Support	38	-
MA: Measurement and Analysis	2	Support	50	+
IPM: Integrated Project Management	3	Project Management	85	+++
RSKM: Risk Management	3	Project Management	43	+
OPF: Organizational Process Focus	3	Process Management	6	-
OT: Organizational Training	3	Process Management	7	-
OPD: Organizational Process Definition	3	Process Management	14	-
RD: Requirements Development	3	Engineering	60	++
TS: Technical Solution	3	Engineering	63	++
PI: Product Integration	3	Engineering	72	++
VER: Verification	3	Engineering	69	++
VAL: Validation	3	Engineering	70	++
DAR: Decision Analysis and Resolution	3	Support	25	-
QPM: Quantitative Project Management	4	Project Management	36	-
OPP: Organizational Process Performance	4	Process Management	10	-
OPM: Organizational Performance Management	5	Process Management	50	+
CAR: Causal Analysis and Resolution	5	Support	30	-

From Table 3.27 shows CMMI in 22 process areas (PA) are classified into four maturity level; Managed (ML2), Defined (ML3), Quantitatively Managed (ML4), Optimizing (ML5). This is not including Initial which is meaning to maturity level 1 (ML1). The number shows % of conformity which Scrum framework supports SG/SP in each level.

Regarding to Scrum, its practices and artifacts are cover 63%, 47%, 23% and 40% of conformance value respectively from ML2, ML3, ML4, and ML5 to SG/SP in CMMI.

Table 3.28 Conformity percentage in process area category to Scrum practices

Process Area	ML	Process Area Category	DSM	SPM	SRM	SR	U
PP: Project Planning	2	Project Management	7	86	0	0	7
PMC: Project Monitoring and Control	2	Project Management	60	10	30	0	0
IPM: Integrated Project Management	3	Project Management	0	90	0	10	0
REQM: Requirements Management	2	Project Management	20	40	20	0	20
SAM: Supplier Agreement Management	2	Project Management	0	0	0	0	100
RSKM: Risk Management	3	Project Management	57	14	0	0	29
QPM: Quantitative Project Management	4	Project Management	0	43	0	0	57
OPF: Organizational Process Focus	3	Process Management	0	0	0	11	89
OT: Organizational Training	3	Process Management	0	14	0	0	86
OPP: Organizational Process Performance	4	Process Management	0	0	0	20	80
OPM: Organizational Performance Management	5	Process Management	0	0	0	80	20
OPD: Organizational Process Definition	3	Process Management	0	29	0	0	71
RD: Requirements Development	3	Engineering	20	70	0	0	10
TS: Technical Solution	3	Engineering	0	88	0	0	13
PI: Product Integration	3	Engineering	0	100	0	0	0
VER: Verification	3	Engineering	0	88	13	0	0
VAL: Validation	3	Engineering	0	60	40	0	0
CM: Configuration Management	2	Support	29	43	0	0	29
PPQA: Process and Product Quality Assurance	2	Support	0	25	0	50	25
MA: Measurement and Analysis	2	Support	38	38	0	0	25
DAR: Decision Analysis and Resolution	3	Support	50	0	0	0	50
CAR: Causal Analysis and Resolution	5	Support	0	20	0	40	40
			13	**39**	**5**	**10**	**34**

From Table 3.28 shows conformity percentage in process area category with Scrum practices; DSM (Daily Scrum Meeting), SPM (Sprint Planning Meeting), SRM (Sprint Review Meeting) and SR (Sprint Retrospective). DSM (Daily Scrum Meeting) is mainly support PMC and RSKM. SPM (Sprint Planning Meeting) is mainly support to PP, IPM, REQM, RD, TS, PI, VER, VAL, CM and SR (Sprint Retrospective) is mainly support to OPM and PPQA. Therefore, SAM, QPM, OPF, OT, OPP, OPD are majority to unsatisfied by implementing Scrum practices.

Totally, these Scrum practices are conforming to CMMI by SPM, DSM, SR and SRM as 39%, 13%, 10% and 5%, respectively.

Table 3.29 Conformity percentage of process area to Scrum artifacts

Process Area	ML	Process Area Category	PB	SB	BDC	U
PP: Project Planning	2	Project Management	7	71	14	7
PMC: Project Monitoring and Control	2	Project Management	0	20	80	0
IPM: Integrated Project Management	3	Project Management	0	90	10	0
REQM: Requirements Management	2	Project Management	40	20	20	20
SAM: Supplier Agreement Management	2	Project Management	0	0	0	100
RSKM: Risk Management	3	Project Management	0	71	0	29
QPM: Quantitative Project Management	4	Project Management	0	14	29	57
OPF: Organizational Process Focus	3	Process Management	0	11	0	89
OT: Organizational Training	3	Process Management	0	14	0	86
OPP: Organizational Process Performance	4	Process Management	0	0	20	80
OPM: Organizational Performance Management	5	Process Management	0	10	70	20
OPD: Organizational Process Definition	3	Process Management	0	14	14	71
RD: Requirements Development	3	Engineering	20	70	0	10
TS: Technical Solution	3	Engineering	0	88	0	13
PI: Product Integration	3	Engineering	56	44	0	0
VER: Verification	3	Engineering	0	100	0	0
VAL: Validation	3	Engineering	60	40	0	0
CM: Configuration Management	2	Support	0	43	29	29
PPQA: Process and Product Quality Assurance	2	Support	0	75	0	25
MA: Measurement and Analysis	2	Support	0	0	75	25
DAR: Decision Analysis and Resolution	3	Support	0	50	0	50
CAR: Causal Analysis and Resolution	5	Support	0	20	40	40
			8	**39**	**18**	**34**

From Table 3.29, shows conformity percentage of process area to Scrum artifacts likes PB (Product backlog), SB (Sprint Backlog), and BDC (Burndown Chart). PB (Product backlog) is mainly support REQM, PI and VAL. SB (Sprint Backlog) is mainly support to PP, IPM, RSKM, RD, TS, VER, CM and PPQA. Lastly, BDC (Burndown Chart) is mainly support to PMC, OPM, and MA. However, SAM, QPM, OPF, OT, OPP and OPD are majority to unsatisfy by implementing Scrum artifacts.

Totally, these Scrum artifacts are conforming to CMMI by SB, BDC, and PB as 39%, 18% and 8%, respectively.

The next part presents on how to overcome the gap for the Synergize in CMMI and Agile by Scrum which describe the fulfillment to cover all requirements from specific goals (SG) and specific practices (SP) in CMMI (Capability Maturity Model Integration) when applying CMMI by Scrum. (From Table 25-26, and Figure 3.4, the conformity of CMMI process area with Scrum framework) However, this research focuses on PMC, IPM and PP from Project Management categories to explore the best process category that is the best fit with Agile by Scrum framework.

3.4 How to overcome the Gap for the Synergize of Project Management in CMMI and Agile by Scrum

Scrum framework has 4 practices; DSM (Daily Scrum Meeting), SPM (Sprint Planning Meeting), SRM (Sprint Review Meeting), SR (Sprint Retrospective) which is include Scrum Artifact has 3 artifacts; PB (Product backlog), SB (Sprint Backlog), BDC (Burndown Chart). However, to cover 100% satisfy to fulfill Scrum based on Standard CMMI Appraisal Method for Process Improvement (SCAMPI) assessment model. There are needed more practices and artifacts to close the gap to overcome these synergize.

Regarding to this research which focuses on Project Management category, the information on Table 30 to Table 33 below show the improvement recommendations which are needed to improve on how to fulfilling the gap for Project Planning (PP), Project Monitoring and Control (PMC), Integrated Project Management (IPM) and Generic Goal (GG) in Scrum Framework.

Table 3.30 How to fulfilling the gap of Project Planning (PP) in Scrum Framework

PP: Project Management					
CMMI Practice	**Description (WP)**	**Scrum Mapping Result (Only PS and U)**			**Improvement Recommedation**
SG/SP		**Conformance**	**Practice**	**Artifact**	
PP.SG 1	**Establish Estimates**				
PP SP 1.4	Estimate Effort and Cost	PS	SPM	SB	Scrum has estimate effort and cost based on team judgement by using relatively estimate approach with tool likes; Poker card in Sprint Planning Meeting (SPM) through Sprint Backlog (SB). However, to satisfy the purpose of SG/SP of CMMI practice, then the following tasks could be undertaken; 1. Establish estimation effort and cost models and its rationale 2. Establish procedure and condition to select suitable estimation models 3. Establish organizational standard values to support effort and cost models
PP.SG 2	**Develop a Project Plan**				
PP SP 2.2	Identify Project Risks	PS	DSM	BDC	Scrum identifies risks in Daily Scrum Meeting (DSM) through Burndown Chart (BDC) and also Sprint Backlog (SB). However, to satisfy the purpose of SG/SP of CMMI practice, then the following tasks could be undertaken; 1. Monitor and document risks 2. Estimate risk impacts and probability/likelihood of occurrence (For instance calculating by an operation of risk level-A=high, B=medium, C=low) 3. Priorities risk and establish risk plan or/and mitigation plan
PP SP 2.3	Plan Data Management	PS	SPM	SB	Scrum has small part of plan for data management in Sprint Planning Meeting (SPM) through Sprint Backlog (SB). However, to satisfy the purpose of SG/SP of CMMI practice, then the following tasks could be undertaken; 1. Establish specific Data management plan with master list of managed data, data content and format description 2. Establish security/Privacy requirement and its procedure 3. Establish mechanisms for data retrieval, reproduction, and distribution 4. Listing and Schedule for the collection of project data
PP.SP 2.5	Plan Needed Knowledge and Skills	U	-	-	Scrum does not address any plan for knowledge and skills which are needed from the project. So that, to satisfy the purpose of SG/SP of CMMI practice, then the following tasks could be undertaken; 1. Establish an inventory of skill needs 2. Establish databases to support needed skill and training schedule 3. Establish a procedure of managing staffing and new hire plans

Table 3.31 How to fulfilling the gap of Project Monitoring and Control (PMC) in Scrum Framework

PMC: Project Monitoring and Control					
CMMI Practice	**Description (WP)**	**Scrum Mapping Result (Only PS and U)**			**Improvement Recommedation**
SG/SP		**Conformance**	**Practice**	**Artifact**	
PMC.SG 1	**Monitor the Project Against the Plan**				
PMC.SP 1.4	Monitor Data Management	PS	SRM	BDC	Regarding to PP.SP 2.3 (Plan data management) Scrum involes its purpose in Sprint Planning Meeting (SPM) through Sprint Backlog (SB). However, in PMC.SP 1.4 to satisfy the purpose of SG/SP of CMMI practice, team should be establish records of data management in Sprint Review Meeting (SRM) through Burndown Chart (BDC) and use an authorization and permission system with periodically backup all project data

Table 3.32 How to fulfilling the gap of Integrated Project Management (IPM) in Scrum Framework

IPM: Integrated Project Management					
CMMI Practice	**Description (WP)**	**Scrum Mapping Result (Only PS and U)**			**Improvement Recommedation**
SG/SP		**Conformance**	**Practice**	**Artifact**	
IPM.SG 1	**Use the Project's Defined Process**				
IPM.SP 1.2	Use Organizational Process Assets for Planning Project Activities	PS	SPM	SB	Scrum addresses use organizational process assets for planning project activities, therefore, it is only Scrum parctices which is Sprint Planning Meeting (SPM) through Sprint Backlog (SB) based on Agile methods. However, to satisfy the purpose of SG/SP of CMMI practice, then the organization could be provide more organizational process assets likes project plans policy and project estimates model.
IPM.SP 1.3	Establish the Projects Work Environment	PS	SPM	SB	Scrum has establish the projects work environment, therefore, it is miss some support services for the project's work environment. Sprint Planning Meeting (SPM) through Sprint Backlog (SB) based on Agile methods is also applicable, however, to satisfy the purpose of SG/SP of CMMI practice, then the following tasks could be undertaken; 1. Identify and provide equipment and tools for the project 2. Establish an operation manuals for installation, operation, and maintenance 3. Create a records of usage, performance, and maintenance
IPM.SP 1.7	Contribute to Organizational Process Assets	PS	SR	SB	Scrum contributes some proposed improvements by Sprint Retrospective (SR) through Sprint Backlog (SB) However, to satisfy the purpose of SG/SP of CMMI practice, then the following tasks could be undertaken; 1. Establish actual process and product measures collected 2. Produce some essential documentation for instance checklists and lessons learned 3. Establish a process guidelines associated with tailoring and implementing the OSSP (Organisations Standard Software Process).

Table 3.33 How to fulfilling the gap of Generic Goal (GG) in Scrum Framework

GG: Generic Goal					
CMMI Practice	**Description (WP)**	**Scrum Mapping Result (Only PS and U)**			**Improvement Recommedation**
SG/SP		**Conformance**	**Practice**	**Artifact**	
GG 2	Institutionalize a Managed Process				
GG.GP 2.1	Establish an Organizatio	U	-	-	Scrum not addresses practice to establish an organizational policy at all. However, to satisfy the purpose of GG/GP of CMMI practice, then he following steps could be undertaken in order to satisfy GG.GP2.1: 1. Define organizational policies, tools and guidelines which can be use in Scrum framework 2. Produce needed documents of all processes based on Scrum practices
GG.GP 2.2	Plan the Process	PS	SPM	SB	Scrum has define its own processes in Scrum practices. However, To standardize the project procedure and also to define and document the Scrum process and minimum requirements in a Scrum project are needed to fulfill and satisfy GG.GP2.1
GG.GP 2.4	Assign Responsibility	PS	SPM	SB	In Scrum framework, there ia only three roles are defined; Scrum Master, Product Owner and team. To satisfy GG.GP 2.4 then it necessary to establish and define the list of assigned responsibility and authority for performing the process
GG.GP 2.5	Train People	U	-	-	Scrum does not show any practices that provide training to the project. In order to satisfy GG.GP 2.5, so that, the training plan should be setting up for closing the gap of needed knowledge which need to use in project including Scrum knowledge.
GG.GP 2.7	Identify and Involve Rele	PS	SPM	SB	There are some activities in Sprint Planning Meeting (SPM) which identify involev relavant stakeholders. However, to fulfill GG.GP 2.7 it should be clearly define the team structure for the project
GG.GP 2.9	Objectively Evaluate Adh	PS	SRM	BDC	Scrum defines activities to review work products in Sprint Review Meeting (SRM) through Burndown Chart (BDC). However, to compleate GG.GP 2.9, project should has more concreate doucument likes; process improvement plans, monitoring project progress and performance against the project plan
GG.GP 2.10	Review Status with High	PS	SR	SB	Scrum has Sprint Retrospective (SR) to review project status to higher level management. However to satisfy GG.GP2.10, it should be establish more needed report likes; results from validation activities report or status of product/process improvements
GG 3	Institutionalize a Defined Process				
GG.GP 3.1	Establish a Defined Proc	PS	SPM	SB	Scrum has its own defined process. Therefore, to satisfy GG.GP 3.1, it is need to establish the organization's set of standard processes (OSSP), and process tailoring guideline
GG.GP 3.2	Collect Process Related	PS	SR	BDC	Scrum has collect process related experiences in Sprint Retrospective (SR), however, to fulfill the requirement from GP.GP 3.2, action proposals, data analysis reports and number of product defects are needed.

The following part present the proposed framework to synergize CMMI and Agile by Scrum for small setting environment likes VSEs/SMEs (Very Small Enterprises/Small and Medium Enterprise namely “CMMISF – CMMI by Scrum framework which is propose to be the development framework for small setting environment (VSEs/SMEs).

3.5 CMMISF – The development framework for small setting environment (VSEs/SMEs)

Software Engineering Institute (SEI) had research related CMMI and Agile in "CMMI or Agile: Why Not Embrace Both!" which presents a value in both paradigms as CMMI and Agile are compatible. At the project level, CMMI focuses at a high level of abstraction on what projects do, not on what development methodology is used, while Agile methods focus on how projects develop products. Therefore, CMMI and Agile methods can co-exist. There can be much value gained from Agile and CMMI synergies.

Today, many CMMI-adopting organizations have Agile development teams. Conversely, CMMI can be effectively introduced in an Agile setting where an iterative, time-boxed approach is used, which is perfectly compatible with CMMI. CMMI and Agile can complement each other by creating synergies that benefit the organization using them. Agile methods provide software development on how to that are missing from CMMI best practices that work well—especially with small, co-located project teams. CMMI provides the systems engineering practices that help enable an Agile approach on large projects. CMMI also provides the process management and support practices that help deploy, sustain, and continuously improve the deployment of an Agile approach in any organization.

CMMI and Agile are focus in difference perspective however, they are compatible. In general, CMMI focuses at a high level of abstraction on what projects do, while Agile methods focus on how projects develop products. Therefore, the CMMI and Agile methods can be synergizing. There can be much value gained from Agile and CMMI synergies in bi-direction. In adopted CMMI organizations can have Agile practices for development teams. On the other hand, CMMI can do more effective for introducing to an Agile environment where iterative and incremental, time-boxed approach are implemented, which is compatible to deployed CMMI.

The main propose of this research is to combine a development process model (Agile) and an improvement model (CMMI) to become on framework namely CMMISF or CMMIbyScrum Framework as shows in Figure 6.

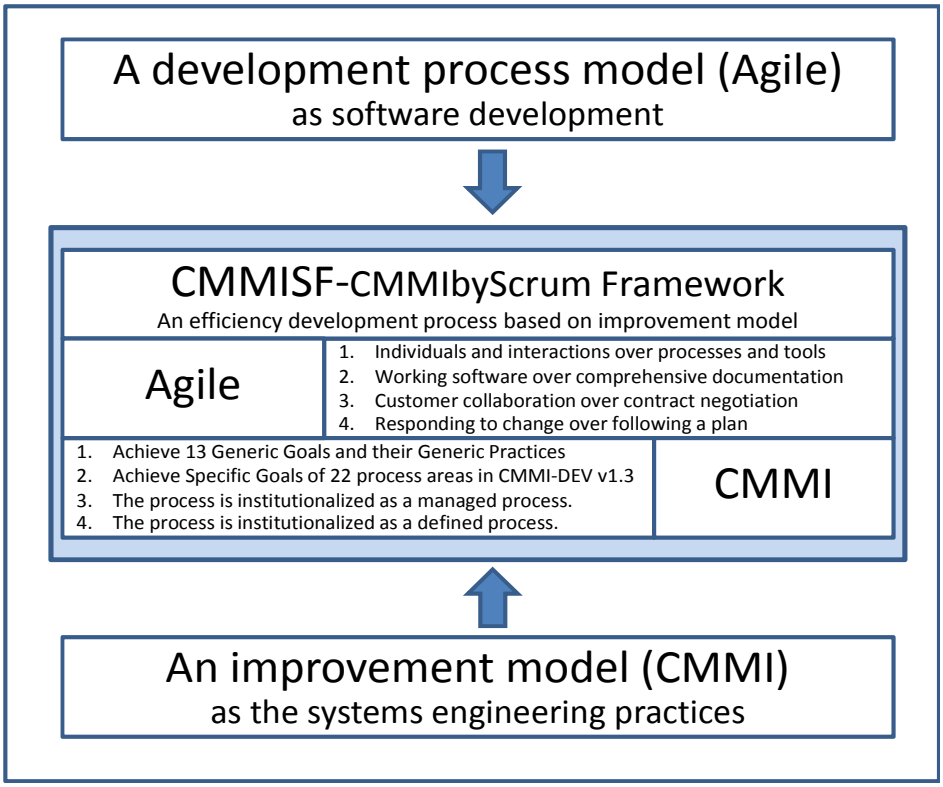


Figure 3.6 The idea of CMMIbyScrum Framework

CMMI and Agile can supplement each other by creating synergies that benefit to the deployed organization. Agile methods provide a development process model as software development on "how to" which are missing from CMMI. The best practices are work well especially with small project teams likes VSEs/SMEs. Agile manifesto for Agile Software Development value is indicated as individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan.

On the other hand, CMMI provides an improvement model in term of the systems engineering practices that help enable an Agile approach on large projects. CMMI also provides the process management, project management; engineering and support practices that help continuously improve the deployment of an Agile approach in any organization. The principle of CMMI is defined things to do as achieve 13 generic goals and their generic practices, achieve specific goals of 22 process areas in CMMI-DEV v1.3. The process is institutionalized as a managed process and a defined process.

The CMMIbyScrum Framework (CMMISF) is intently and purposely to design the Light-Weight Project Management (LWPM) approach to implement CMMI by mapping between CMMI-Project Management goals and Agile-Scrum based on enhanced artifacts and practices to be satisfied by Light-Weight Project Management SCAMPI Assessment Model (LWPM-SAM) which is designed for VSEs/SMES companies. Our approach focuses on the Project Management category which composes Project Planning (PP), Project Monitoring and Control (PMC) and Integrated Project Management (IPM). In order to enhance both additional artifacts and practices with

defined LWPM-SAM. Moreover, we collected relevant data by using questionnaire and also the dedicated tool SPIALS (Software Process Improvement Adaptive Learning System) as shown in Figure 7.

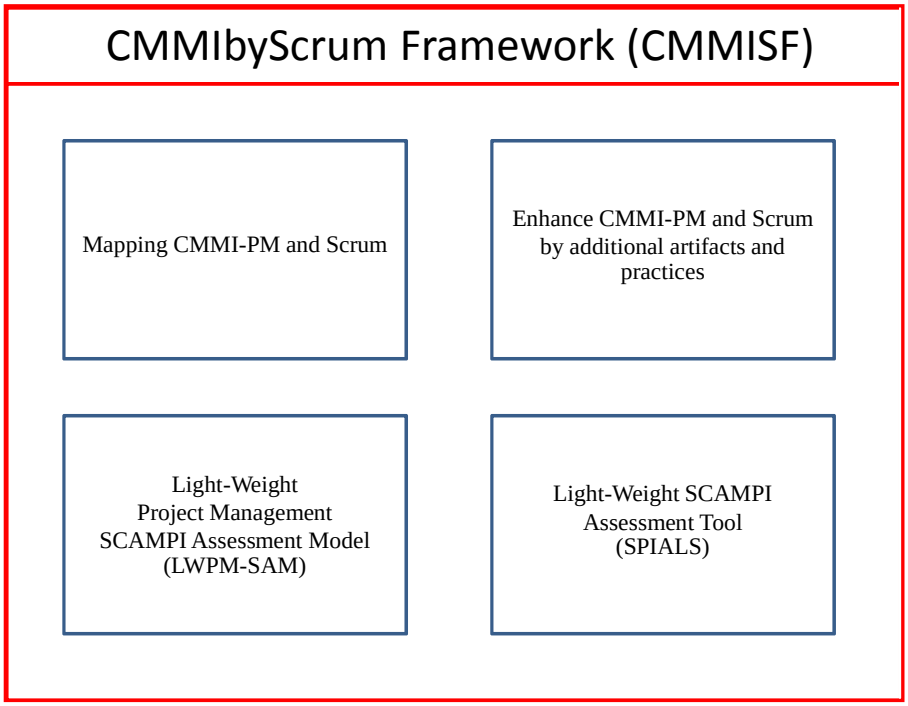


Figure 3.7 The component of CMMISF

CMMIbyScrum Framework (CMMISF) is an alternative approach to accelerate the transition process by using the Light-Weight Project Management (LWPM) organization framework. This intends to achieve better performance with less effort. However, the LWPM implementation should be done with enough quality of processes and needed products to be qualified by the Standard CMMI Appraisal Method for Process Improvement (SCAMPI).

The CMMISF is a conceptual framework for an effective practice. The CMMISF is composed of four parts as following;

1: Gap Analysis by mapping between CMMI-PM and Scrum in term of artifacts and practices, and then, complete the gap.

2: Enhance CMMI-PM and Scrum by additional artifacts and practices. Besides, there is specific light-weight CMMI assessment which is designed for VSEs/SMEs companies namely “Light-Weight Project Management SCAMPI Assessment Model (LWPM-SAM)”, come together with Light-Weight SCAMPI Assessment Tool (SPIALS). The detail of CMMISF is explained in next chapter.

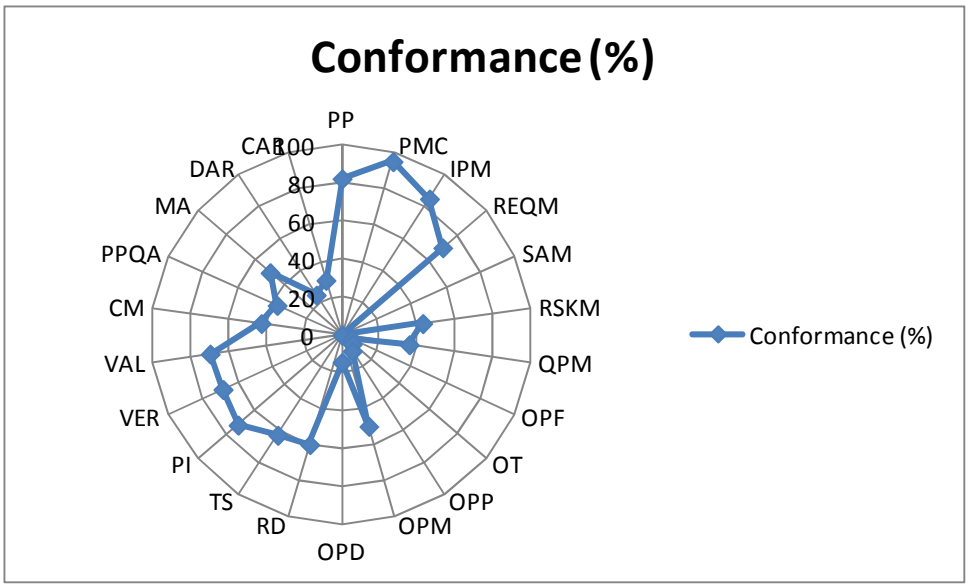


Figure 3.8 The conformance value of CMMISF

Figure 3.8 shows the conformance value of CMMISF in order to use as adaptive learning function through LWPM-SAM. In this research, the project management approach which comprise of PP, PMC and IPM is focused. In chapter 5, the evaluation data is presented the statistical in PP, PMC and IPM based on the most conformity value (%) among 22 process areas.

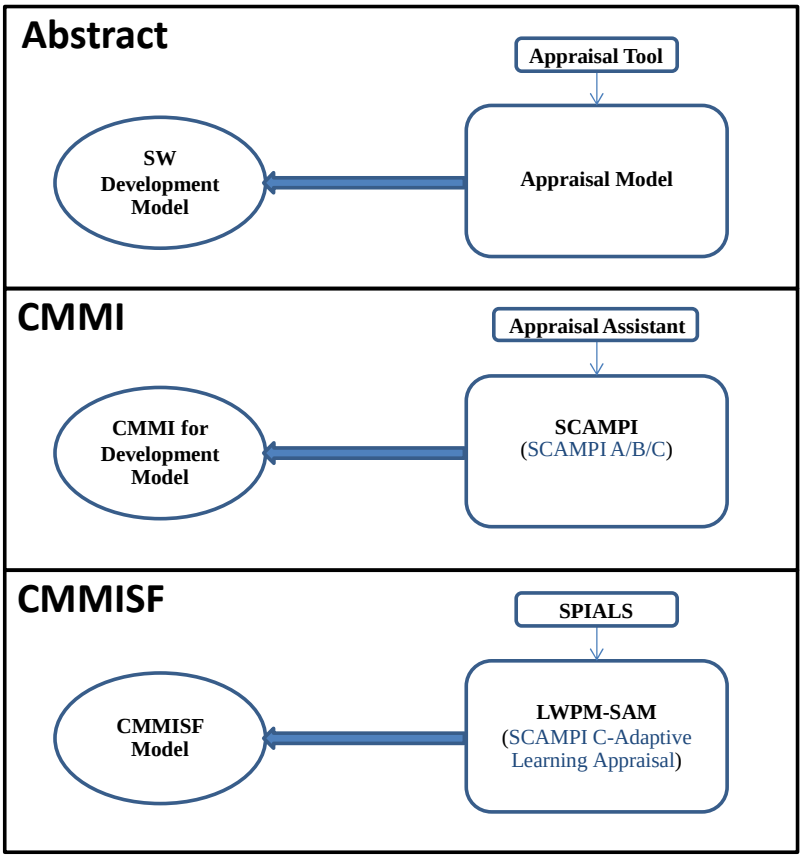


Figure 3.9 The comparison of Abstract model and CMMI/CMMISF

Figure 3.9 shows the comparison of Abstract model and CMMI/CMMISF. This figure presents basic element of software development abstract which compose of software development model and appraisal model that working by appraisal tool. For instance, CMMI has CMMI for development and SCAMPI which working by appraisal tool likes Appraisal Assistant that is developed by the Software Quality Institute, Griffith University. Finally, this dissertation proposes CMMISF and LWPM-SAM which working through SPIALS.

In chapter 4, the light-weight SCAMPI (Standard CMMI Appraisal Method for Process Improvement) assessment model or LWPM-SAM is presented. This LWPM-SAM has tool namely “SPIALS-Software Process Improvement Adaptive Learning System” which is a web-based tool to support the CMMI self-assessment and also presents their assessment result.

Chapter 4 Light-Weight Project Management SCAMPI Assessment Model (LWPM-SAM)

The SCAMPI stands for Standard CMMI Appraisal Method for Process Improvement. Assessments in CMMI are conducted using the SCAMPI Methodology which is ratings by an appraisal team to the process areas under assessment for the capability level (for process areas) or the maturity level (for an organizational unit) based on SCAMPI Method Definition Document (MDD).

From Table 4.2, SCAMPI has defined three classes namely A, B and C. There is a difference in term of definition as explained; SCAMPI A is the most rigorous method and the only method that can result in a rating. SCAMPI B provides options in model scope, but the characterization of practices is fixed to one scale and is performed on implemented practices. SCAMPI C provides a wide range of options, including characterization of planned approaches to process implementation according to a scale defined by the user.

Table 4.1 Characteristics of Appraisal Classes from Software Engineering Institute (SEI)

Characteristic	Class A	Class B	Class C
Amount of objective evidence	High	Medium	Low
Ratings generated	Yes	No	No
Resource needs	High	Medium	Low
Team size	Large	Medium	Small

SCAMPI C Method can be scoped at any level of granularity and the scale can be tailored to the appraisal objectives, which might include the fidelity of observed practices to model/goal achievement or the return on investment to the organization from implementing practices.

Characterization of examined CMMI components can be done using a different scale on the basis degree determined for implementation during appraisal. This can be

classified as Yellow, Red Green, where Red means low level of implementation and Green is a high level of implementation.

4.1 LWPM-SAM: The light-weight assessment for VSEs/SMEs

In this propose, Light-Weight Project Management SCAMPI Assessment Model (LWPM-SAM) is establish based on SCAMPI C appraisal by scaling down SCAMPI A for self-preparing.

Table 4.2 Process activities between SCAMPI A and LWPM-SAM processes

Phase	**SCAMPI A Process**	**LWPM-SAM Process**
1. Plan and Prepare for Appraisal	1.1 Analyze Requirements 1.2 Develop Appraisal Plan 1.3 Select and Prepare Team 1.4 Obtain and Inventory Initial Objective Evidence 1.5 Prepare for Appraisal Conduct	1.1 Self-Analyze Requirements 1.2 Self-Develop Appraisal Plan 1.3 Self-Select and Prepare Team 1.4 Obtain and Inventory Initial Objective Evidence based on SCAMPI C 1.5 Self-Prepare for Appraisal Conduct
2. Conduct Appraisal	2.1 Prepare Participants 2.2 Examine Objective Evidence 2.3 Document Objective Evidence 2.4 Verify Objective Evidence 2.5 Validate Preliminary Findings 2.6 Generate Appraisal Results	2.1 Prepare Participants by register in SPIALS 2.2 Examine Objective Evidence by SPIALS 2.3 Document Objective Evidence by SPIALS 2.4 Verify Objective Evidence by SPIALS 2.5 Validate Preliminary Findings by SPIALS 2.6 Generate Appraisal Results by SPIALS
3. Report Results	3.1 Deliver Appraisal Results 3.2 Package and Archive Appraisal Assets	3.1 Deliver Appraisal Results by SPIALS 3.2 Package and Archive Appraisal Assets by SPIALS

LWPM-SAM is defined via level of practice implementation indicator (Table 4.4) based on the conformance value of CMMISF (Figure 3.8) which can be adapt through SPIALS tool.

From Table 4.2, the SCAMPI has three phases of SCAMPI method which is compose of 1) plan and prepare for appraisal, 2) conduct appraisal, and 3) report results as shown to compare with LWPM-SAM.

Table 4.3 Process activities between SCAMPI A and LWPM-SAM and its outputs

Phase	**SCAMPI A Process**	**SCAMPI A- MDD Process (Activities)**	**SCAMPI A (Outputs)**	**LWPM-SAM (Activities)**	**LWPM-SAM (Outputs)**
1. Plan and Prepare for Appraisal	1.1 Analyze Requirements	1.1.1 Determine Appraisal Objectives 1.1.2 Determine Data Collection Strategy 1.1.3 Determine Appraisal Constraints 1.1.4 Determine Appraisal Scope 1.1.5 Determine Appraisal Outputs 1.1.6 Obtain Commitment to Initial Appraisal Plan	Initial Appraisal Plan	Self-analyze requirements	Self-appraisal plan (draft)
	1.2 Develop Appraisal Plan	1.2.1 Tailor Method 1.2.2 Identify Needed Resources 1.2.3 Develop Data Collection Plan 1.2.4 Determine Cost and Schedule 1.2.5 Plan and Manage Logistics 1.2.6 Document and Manage Risks 1.2.7 Obtain Commitment to Appraisal Plan	approved appraisal plan	Self-develop appraisal	Self-appraisal plan
	1.3 Select and Prepare Team	1.3.1 Identify Appraisal Team Leader 1.3.2 Select Team Members 1.3.3 Document and Manage Conflicts of Interest 1.3.4 Prepare Team	• training records • appraisal team member assignments and qualifications • identified and documented conflicts of interest • a prepared appraisal team that has completed − appraisal method training − appraisal reference model training − team-building activities − team orientation regarding	Self-select and prepare team	Self-training

			appraisal		
	1.4 Obtain and Inventory Initial Objective Evidence	1.4.1 Obtain Initial Objective Evidence 1.4.2 Inventory Objective Evidence	• data inventory results (e.g., data availability summaries) • identification of additional information needed • initial set of objective evidence	Self-initial objective evidence	Self-defined set of objective evidence based on SCAMPI C
	1.5 Prepare for Appraisal Conduct	1.5.1 Perform Readiness Review 1.5.2 Re-Plan Data Collection	• revised appraisal plan • updates to the data collection plan as required	Self-plan and prepare	Self-updated and revised plan
2. Conduct Appraisal	2.1 Prepare Participants	2.1.1 Conduct Participant Briefing	prepared appraisal participants	Registered Participants in SPIALS	Participants and related roles in SPIALS
	2.2 Examine Objective Evidence	2.2.1 Examine Objective Evidence from Artifacts 2.2.2 Examine Objective Evidence from Affirmations	• updated appraisal data • updated data collection plan	Artifacts mapping to Objectives by SPIALS	Questionnaires form by SPIALS
	2.3 Document Objective Evidence	2.3.1 Take/Review/Tag Notes 2.3.2 Record Presence/Absence of Objective Evidence 2.3.3 Document Practice Implementation 2.3.4 Review and Update the Data Collection Plan	• updated appraisal data − tagged notes − noted practice implementation gaps (if any) − noted exemplary implementation (if any) − revised data collection plan (if applicable) − annotated worksheets • requests for additional data (artifacts or affirmations)	Artifacts mapping to Objectives by SPIALS	Questionnaires form by SPIALS
	2.4 Verify Objective Evidence	2.4.1 Verify Objective Evidence 2.4.2 Characterize Implementation of Model Practices and Generate Preliminary Findings	• updated appraisal data − strength statements (if any) − weakness statements (if any) − annotated worksheets • updated appraisal artifacts − preliminary findings − revised data collection plan − requests for additional data • practice characterizations − instantiation level	Artifacts mapping to Objectives by SPIALS	Questionnaires form by SPIALS

			− organizational unit level		
	2.5 Validate Preliminary Findings	2.5.1 Validate Preliminary Findings	validated appraisal findings	Artifacts mapping to Objectives by SPIALS	Questionnaires form by SPIALS
	2.6 Generate Appraisal Results	2.6.1 Derive Findings and Rate Goals 2.6.2 Determine Process Area Ratings 2.6.3 Determine Process Area Profile 2.6.4 Determine Maturity Level 2.6.5 Document Appraisal Results	• final findings • recorded rating decisions	Artifacts mapping to Objectives by SPIALS	Questionnaires form by SPIALS
3. Report Results	3.1 Deliver Appraisal Results	3.1.1 Deliver Final Findings 3.1.2 Conduct Executive Session(s) 3.1.3 Plan for Next Steps	• documented final findings • final report (if requested) • recommendations report (if requested) • process improvement action plan (if requested)	Deliver Final Findings by SPIALS	Documented final findings and reports by SPIALS
	3.2 Package and Archive Appraisal Assets	3.2.1 Collect Lessons Learned 3.2.2 Generate Appraisal Record 3.2.3 Provide Appraisal Feedback to the SEI 3.2.4 Archive and/or Dispose of Key Artifacts	• appraisal data package • appraisal record • completed forms and checklists • sanitized data (as appropriate and agreed upon during planning) • lessons learned (appraisal team, organization)	Deliver Final Findings by SPIALS	Documented final findings and reports by SPIALS

From Table 4.3 shows the detiled of process activities between SCAMPI A and LWPM-SAM and its outputs. Table 4.3 also presents difference approaches of activities and outputs between SCAMPI and LWPM-SAM based on tool supporting “Software Process Improvement Adaptive Learning System-SPIALS”. To generate the assessment result by LWPM-SAM, many issues have to be concerned. According to SCAMPI, evidences are most important as the can imply the strength of a practice. Therefore we have precisely defined the meaning of the concepts Practice Implementation Indicator, Practice Characteristic, Goal and Process Area Satisfaction (see Tables 4.4-4.7).

The number of strength practices implies the quantity of implemented practices for a goal. The number of goals with high quantity of implemented practices can imply the satisfaction of a process area.

Table 4.4 The Relation of Evidence Selection to Determine Practice Implementation Indicator

Evidence Selection (for All Related Roles or Majority after an Affirmation)	**Practice Implementation Indicator**
All roles: Use	Strength
All roles: Not use	Weak
All roles: Not available to use	Not Rated
Majority: Combination of evidence selection result	Ask for affirmations for those evidences. Then assumes an indicator value from a majority. If there is no majority, uses Weak.

Based on the values of the Practice Implementation Indicators the respective Practice Characteristics can be rated. We define:

Let W: number of Weak indicators

S: number of Strength indicators

NR: number of Not Rated indicators

Table 4.5 The Relation of Practice Implementation Indicator to Determine Practice Characteristics

Practice Implementation Indicator Condition	**Practice Characteristics**
$W \geq NR$ and $S = 0$ and $W > 0$	Not implemented
$W > S$ and $W > NR$ and $S > 0$	Partially implemented
$S > W$ and $S > NR$ and $W > 0$	Largely implemented
$S \geq NR$ and $W = 0$ and $S > 0$	Fully implemented
$S = 0$ and $W = 0$ and $NR \geq 0$	Not Rated
$NR > W + S$ and $S \geq 0$ and $W \geq 0$	Not Capable

Now we can use the PC values to rate the satisfaction of associated goals based on the following definitions:

Let SI: sum of Largely and Fully Implemented PCs

NSI: sum of Partially and Not Implemented PCs

Table 4.6 The Relation of Practice Characteristics to Determine Goal Satisfaction

Practice Characteristics	**Goal Satisfaction**
SI > NSI	Satisfied
SI < NSI	Not Satisfied
SI = NSI	Not Capable

Finally, we determine the process area satisfaction from the goal satisfaction values. Again we define:

Let SG: number of Satisfied goals

NSG: number of Not Satisfied goals

NCG: number of Not Capable goals

Table 4.7 The Relation of Goal Satisfaction to Determine Process Area Satisfaction

Goal Satisfaction	**Process Area Satisfaction**
SG > (NSG + NCG) and SG ≥ NCG	Satisfied
SG < (NSG + NCG) and NSG ≥ NCG	Not Satisfied
NCG > SG + NSG	Not Capable

4.1.1 The conceptual design for LWPM-SAM

The necessary input data is entered by an organization representative who defines evidences for respective software development processes. The result is reported automatically indicating acceptance in the selected process areas. However, collecting documents is not a focus because every organization has its own document types and standards. Therefore, only evidence names are collected. The system reacts immediately if the user enters mismatching data. Fig.4.1 presents the LWPM-SAM by SPIALS work flow which is used as a filter to present only related evidence to the participants.

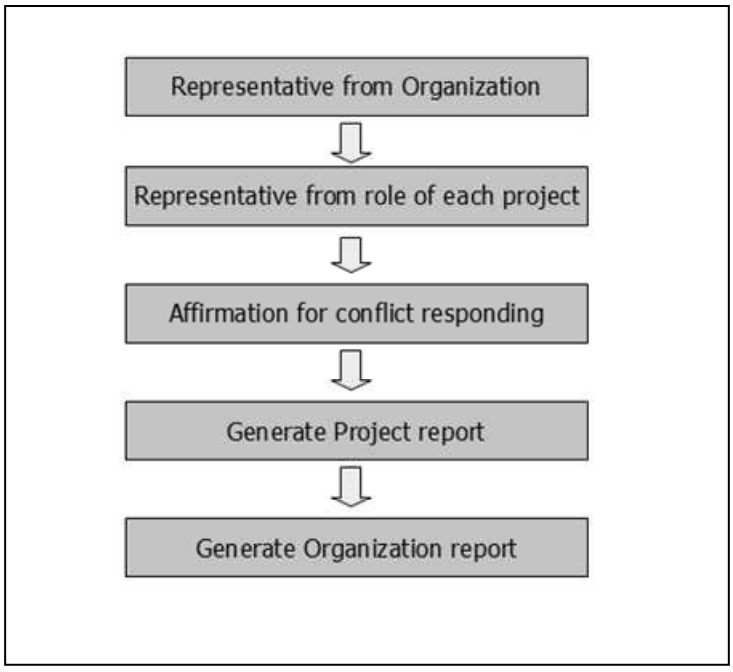


Figure 4.1 Light-Weight Project Management SCAMPI Assessment Model (LWPM-SAM) Workflow

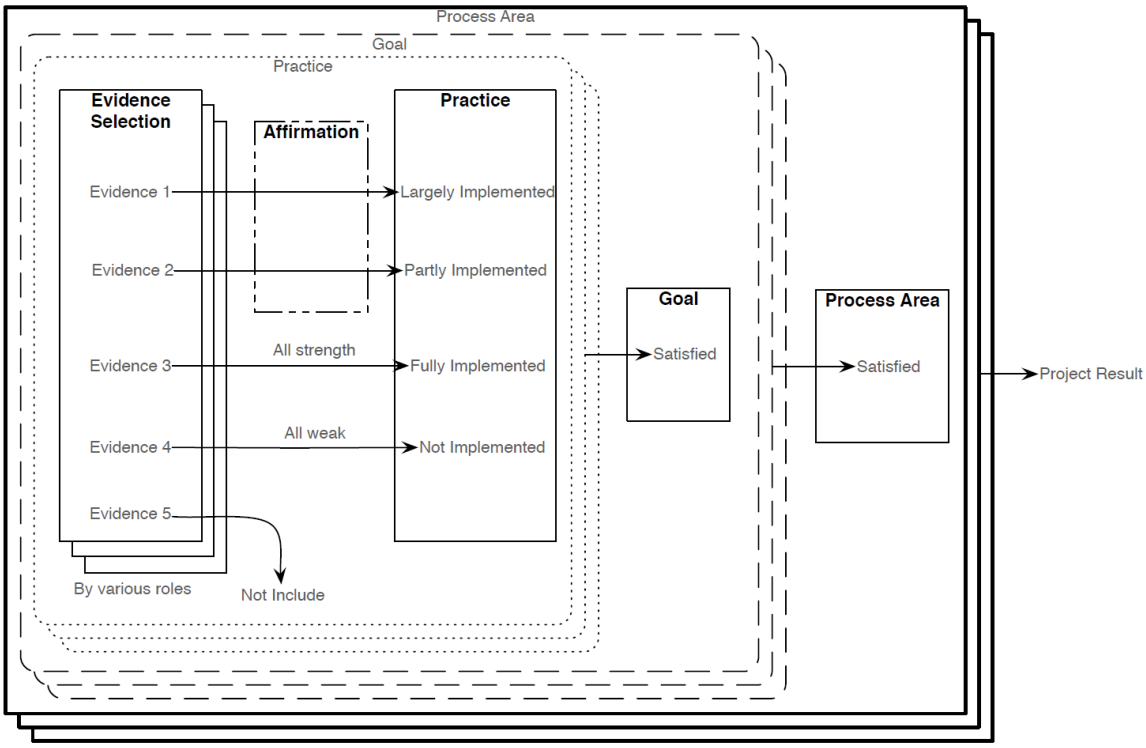


Figure 4.2 LWPM-SAM procedure of evidence selection to indicate project result

In summary, LWPM-SAM by SPIALS is automatically processes evidence usage for every role, and then generates assessment results for the project. The selected evidence is a representative for a practice implementation indicator. The LWPM-SAM approach is to evaluate goals and process areas which are presented in Fig.4.2.

4.2 SPIALS – The tool to support LWPM-SAM

In this chapter, we present how to apply the tool supporting LWPM-SAM Approach namely, “SPIALS” which is to perform a self-assessment model. SPIALS also produces a gap report analysis and SPI proposal report which can be used to start a process improvement program for VSEs/SMEs.

The SPIALS (Software Process Improvement Adaptive Learning System) is a web-based tool supporting the self-assessment and presenting the results based on a specific SCAMPI customization. The benefit of using SPIALS for VSEs/SMEs is to easily get an individual self-assessment result which reflects its process status and performance. The tool serves mainly to analyze potential weaknesses and to define and perform improvement measures before investing in a formal SCAMPI certification assessment. Because approved SPI best practices should be applicable especially for VSEs/SMEs, SPIALS tries to collect such SPI information given by VSEs/SMEs for further analysis to improve the system itself but also to propose SPI road maps for VSEs/SMEs to gain more quality improvements in software industry.

The organization status and needs are an initial and important input to classify an appropriate CMMISF approach which fits to the organization. As a result, SPIALS provides gap analysis information and also proposes individual SPI measures to achieve the VSEs/SMEs process improvement objectives. In the following sections we present some interesting aspects of SPIALS.

4.2.1 The conceptual design for SPIALS

SPIALS assists VSEs/SMEs to perform self-assessments. Its procedure is consistent to the SCAMPI principles including the three phases; Plan and Prepare for Appraisal, Conduct Appraisal, and Report Results. Fig. 4.3 depicts the main use cases identified for SPIALS.

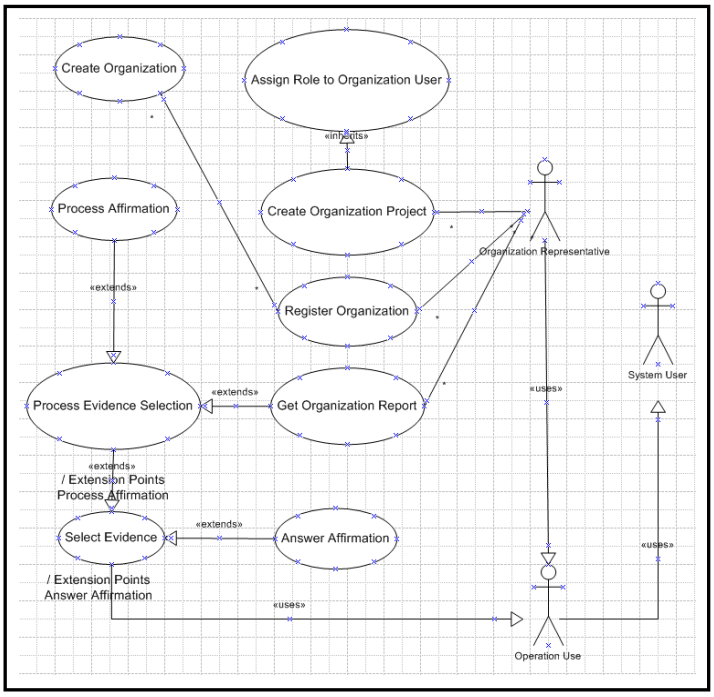


Figure 4.3 SPIALS Use Cases

SPAILS is assumed to be used by an organization's representative. At first projects to be evaluated have to be created. Then roles associated with the projects are assigned. Next systematic questions are defined and an evidence type is selected.

The core elements of the SPIALS self-assessment are questionnaires that have to be completed by the representatives of the organization. These questionnaires are generated based on the organization’s input. All questionnaires conform to a common underlying model, shown in Fig. 4.4.

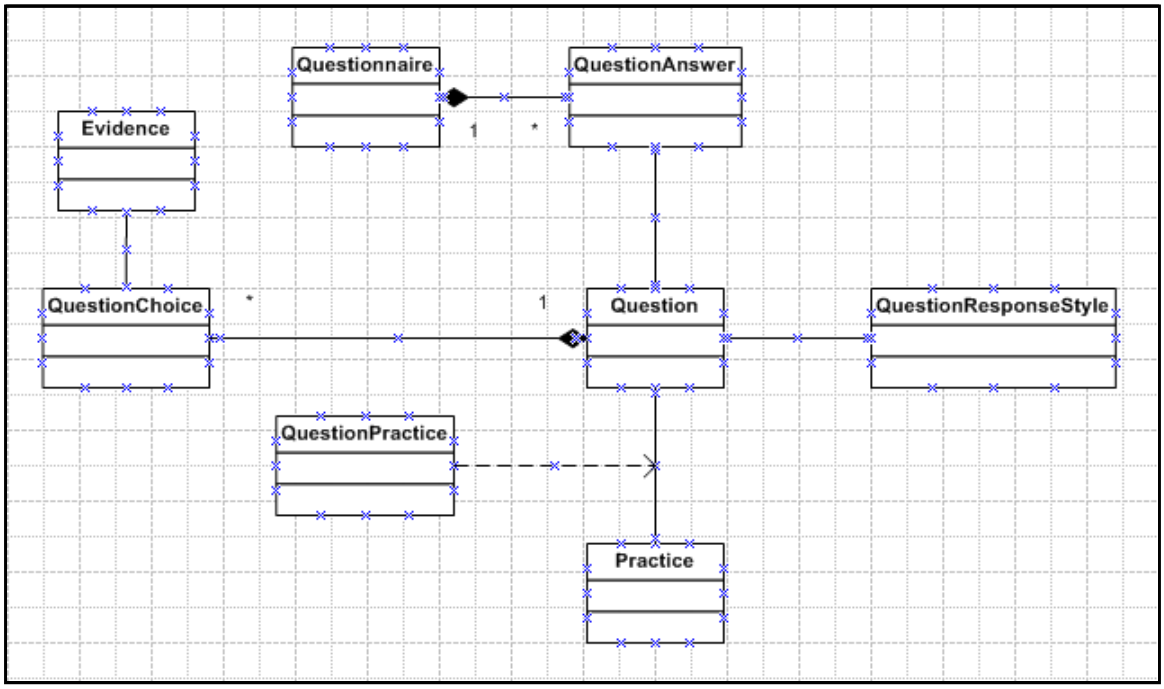


Figure 4.4 Questionnaire Model

4.2.2 The user interface for SPIALS

In this section, we present the structure of the Questionnaire Framework based on a light-weight for Software Process Improvement Self-Assessment Tool (QF-SPISAT) and descriptions of the SPIALS component. QF-SPISAT composes of five parts which show according A to E (see Fig.4.5); A. Organization's general information, B. Participant information, C. Project information, D. Assessment definition which include; D1.Assessment scope and D2.Participant's multi-roles in each project, and E. Appraisal's artifact confirmation which include; E1.Participant dashboard and E2.Artifact usage. The QF-SPISAT is based on SCAMPI appraisal theory with Scrum deployment. The confirmation of assessment use project's artifact as references.

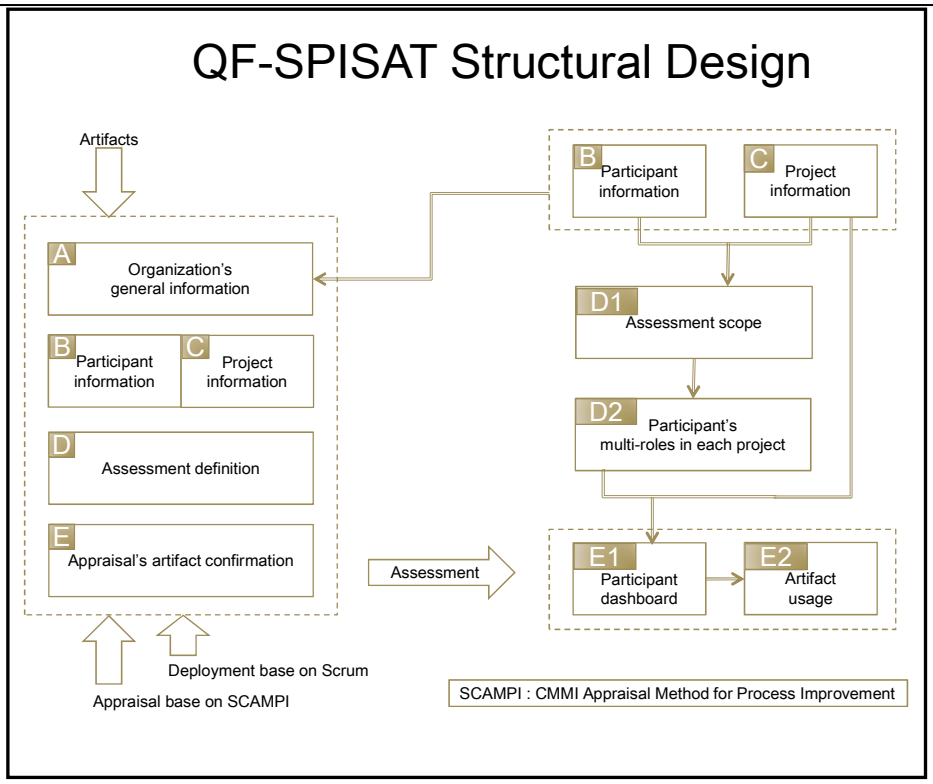


Figure 4.5 The QF-SPISAT Structure Design

Fig. 4.5 shows the steps of information preparing phase from A to C, then D1 to D2 are assessment defining phase for the assessment scope and participants' roles in each project. Finally, assessing phase in E (E1 to E2), each person has to answer questions base on their roles in every related projects. The questionnaire is answer in term of project artifacts as "Yes/No" questions order by process area which defines in D.Assessment definition. The result of appraisal is presented by organization's gap and software process improvement report.

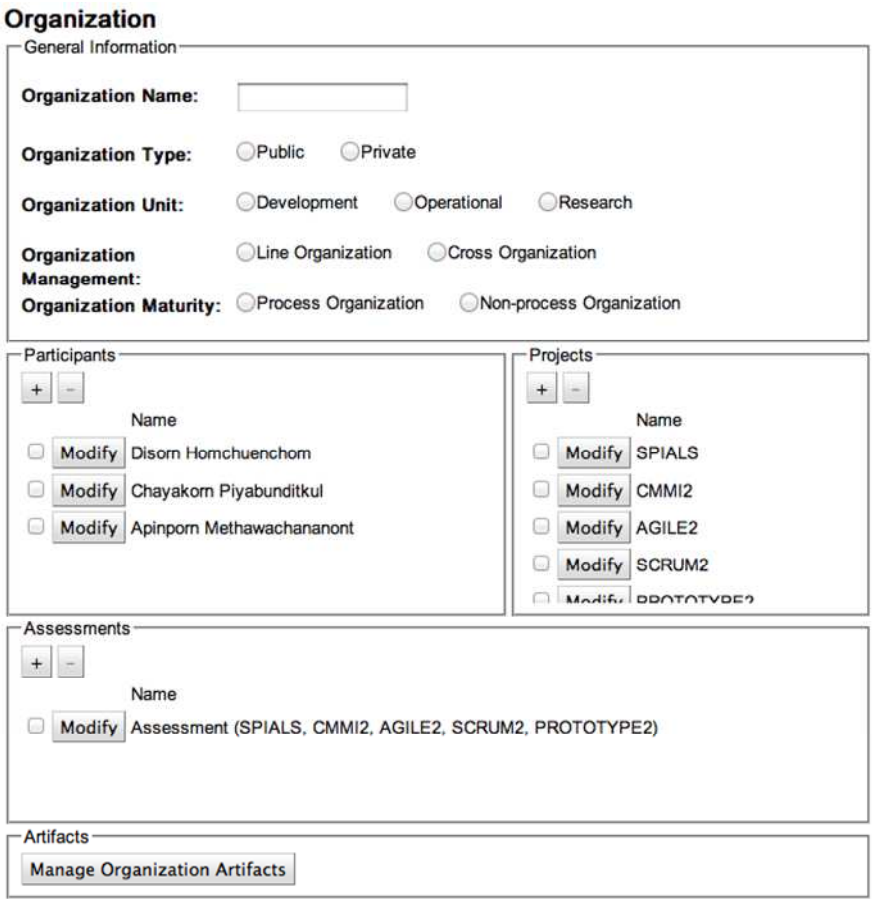


Figure 4.6 Organization General Information

We illustrate the information flow of SPIALS. Organization has to be entered organization data and details for instance (see Fig. 4.6); participant's role, project, and also the assessment definition (maturity or capability) to be a criterion of automate questionnaire system which is confirms via artifact. (See Fig. 4.7 and Fig. 4.8)

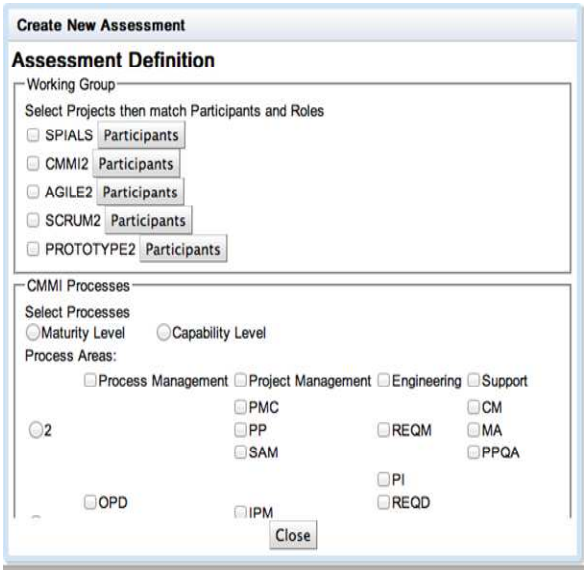


Figure 4.7 Assessment Definition

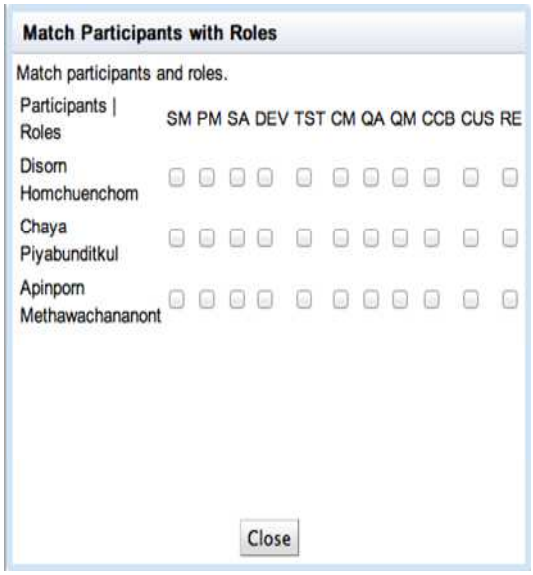


Figure 4.8 Match Participants with roles

We use the information above to create a set of question which participants have to answer based on their roles, step by each process area (PA) upon organization's assessment definition. (See Fig. 4.9 and Fig. 4.10) The definition of roles is based on visited 3 companies which are implemented CMMI ML 3 and 5 including NECTEC.

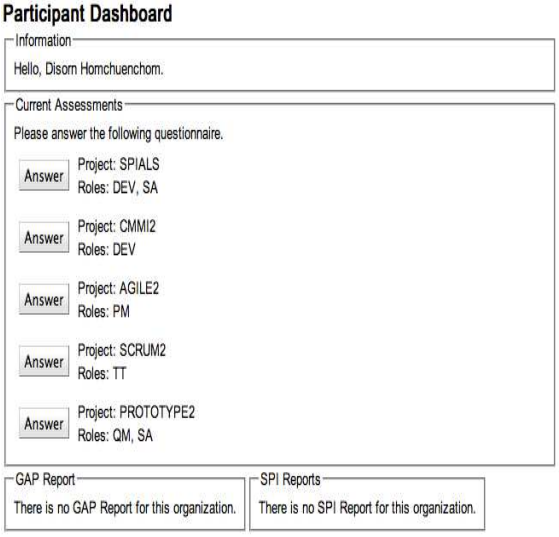


Figure 4.9 Participants Dashboard

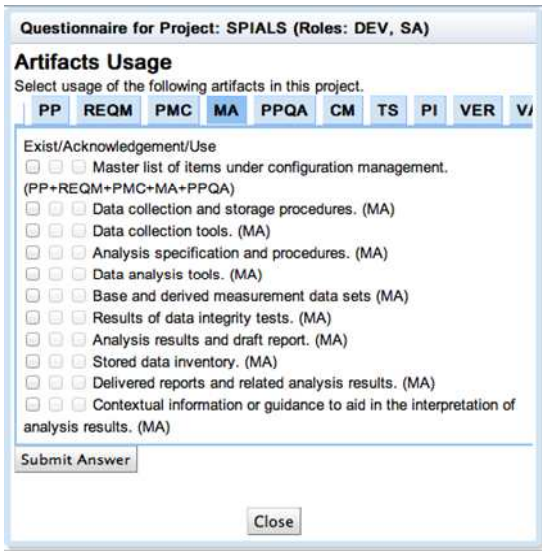


Figure 4.10 Questionnaire for Artifact Usage

4.2.3 The assessment report for SPIALS

In the last part, according to Fig. 4.11, SPIALS produces two reports. First, the gap analysis report contains an overall organization summary, gap analysis result, strength and weakness of organization and its project. Second, the SPI proposal report describes the details for a continuous process improvement. It explains how to fulfill organization weaknesses and shows the values of measurement comparison in terms of effort and User Acceptance Test (UAT) defects.

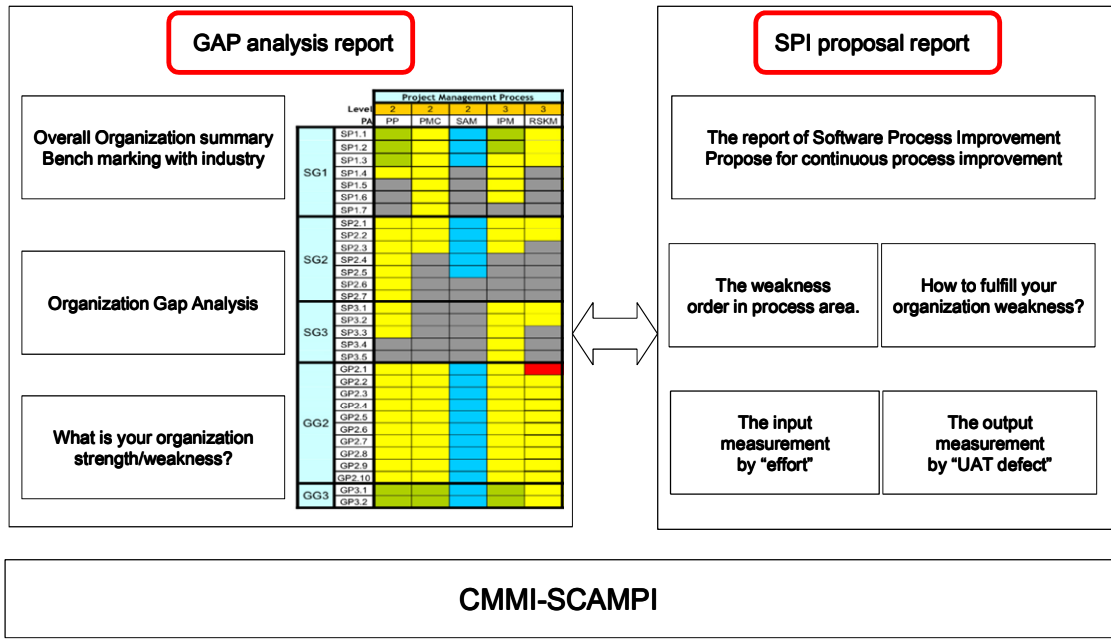
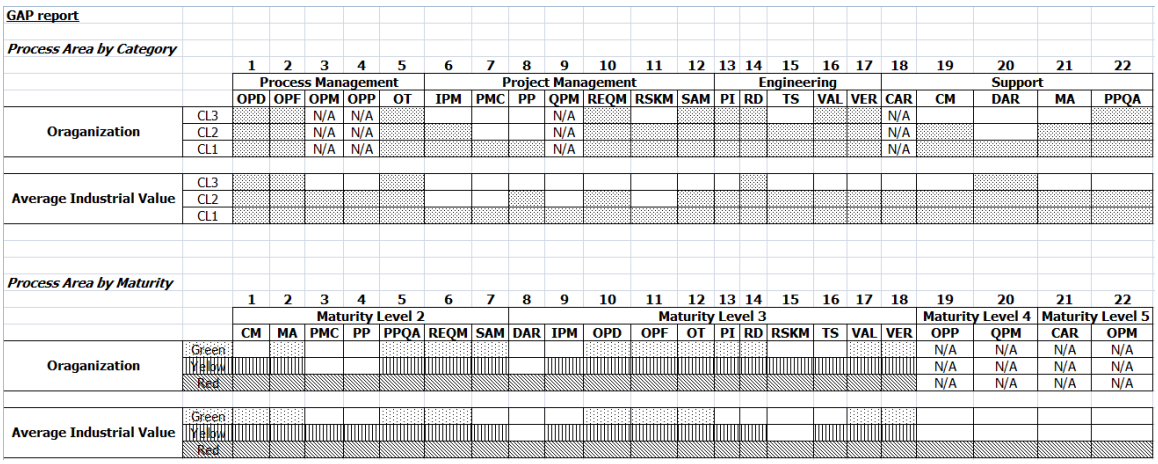


Figure 4.11 GUI of SPIALS for Gap Analysis and SPI Proposal Report

The relevant artifacts from those topics are mapped according to the mapping mechanism presented in chapter 3. SPIALS has an automated process to create a result such as the Gap Analysis Report that describes the gap between organizational SCAMPI targets and the classified current practices. The gap analysis also shows the comparison results with the industrial bench-marking systems. Beside the gap analysis reports, SPIALS also displays the Software Process Improvement report (SPI proposal) explaining how to fulfill those gaps.



GAP report

Process Area by Category

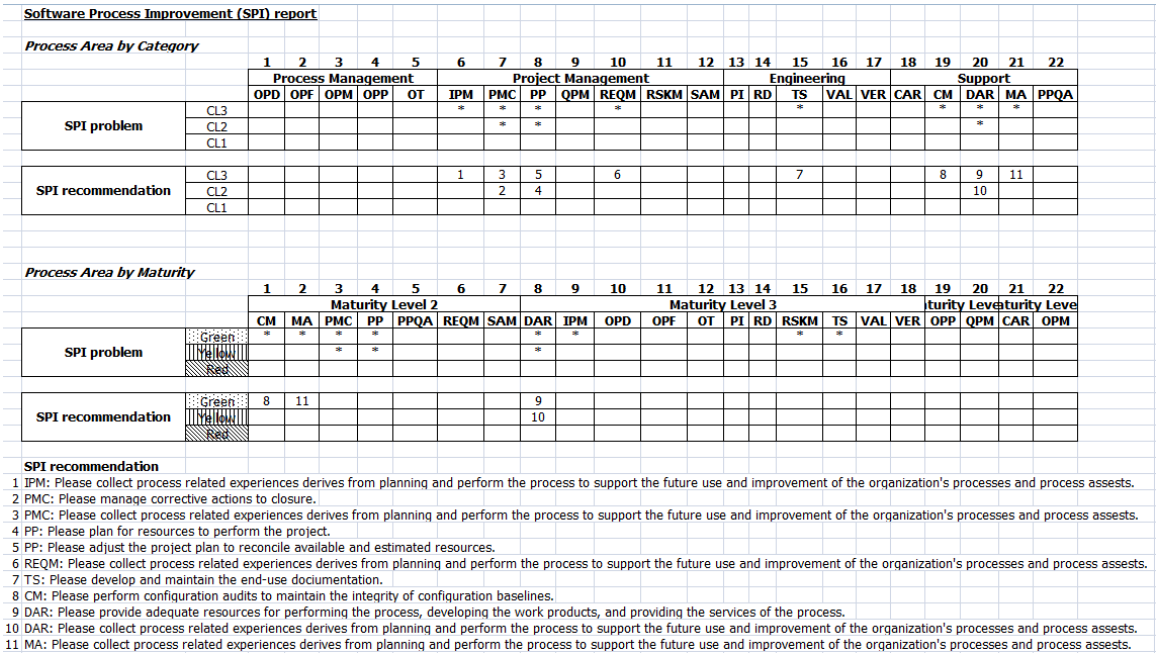
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
		Process Management					Project Management							Engineering						Support			
		OPD	OPF	OPM	OPP	OT	IPM	PMC	PP	QPM	REQM	RSKM	SAM	PI	RD	TS	VAL	VER	CAR	CM	DAR	MA	PPQA
Oraganization	CL3			N/A	N/A					N/A									N/A				
	CL2			N/A	N/A					N/A									N/A				
	CL1			N/A	N/A					N/A									N/A				
Average Industrial Value	CL3																						
	CL2																						
	CL1																						

Process Area by Maturity

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
		Maturity Level 2							Maturity Level 3											Maturity Level 4		Maturity Level 5	
		CM	MA	PMC	PP	PPQA	REQM	SAM	DAR	IPM	OPD	OPF	OT	PI	RD	RSKM	TS	VAL	VER	OPP	QPM	CAR	OPM
Oraganization	Green																			N/A	N/A	N/A	N/A
	Yellow																			N/A	N/A	N/A	N/A
	Red																			N/A	N/A	N/A	N/A
Average Industrial Value	Green																						
	Yellow																						
	Red																						

Figure 4.12 Gap report an evidence selection to indicate project result

Regarding to mention above, SPIALS determines according to the CMMISF to produce gap report and Software Process Improvement (SPI) proposal report which are present in 2 views; process area by category and process area by maturity. Gap report shows the organization's values and the comparison between organization and average industrial value based on its benchmark to show an organization point of reference based on industry database. (See Fig. 4.12) The benchmark reference here is the data collection from organizations that use this SPIALS.



Software Process Improvement (SPI) report

Process Area by Category

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
		Process Management					Project Management							Engineering						Support			
		OPD	OPF	OPM	OPP	OT	IPM	PMC	PP	QPM	REQM	RSKM	SAM	PI	RD	TS	VAL	VER	CAR	CM	DAR	MA	PPQA
SPI problem	CL3						*	*	*		*					*				*	*	*	
	CL2							*	*												*		
	CL1																						
SPI recommendation	CL3						1	3	5		6					7				8	9	11	
	CL2							2	4												10		
	CL1																						

Process Area by Maturity

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
		Maturity Level 2							Maturity Level 3											turity Leve		turity Leve	
		CM	MA	PMC	PP	PPQA	REQM	SAM	DAR	IPM	OPD	OPF	OT	PI	RD	RSKM	TS	VAL	VER	OPP	QPM	CAR	OPM
SPI problem	Green	*	*	*	*				*	*						*	*						
	Yellow			*	*				*														
	Red																						
SPI recommendation	Green	8	11						9														
	Yellow								10														
	Red																						

SPI recommendation

1 IPM: Please collect process related experiences derives from planning and perform the process to support the future use and improvement of the organization's processes and process assests.
2 PMC: Please manage corrective actions to closure.
3 PMC: Please collect process related experiences derives from planning and perform the process to support the future use and improvement of the organization's processes and process assests.
4 PP: Please plan for resources to perform the project.
5 PP: Please adjust the project plan to reconcile available and estimated resources.
6 REQM: Please collect process related experiences derives from planning and perform the process to support the future use and improvement of the organization's processes and process assests.
7 TS: Please develop and maintain the end-use dociumentation.
8 CM: Please perform configuration audits to maintain the integrity of configuration baselines.
9 DAR: Please provide adequate resources for performing the process, developing the work products, and providing the services of the process.
10 DAR: Please collect process related experiences derives from planning and perform the process to support the future use and improvement of the organization's processes and process assests.
11 MA: Please collect process related experiences derives from planning and perform the process to support the future use and improvement of the organization's processes and process assests.

Figure 4.13 Software Process Improvement (SPI) proposal report

Finally, SPI proposal report depicts to advise the possibility solution to solve the non-conformity which implements SPI program based on SPIALSs recommendation. The recommendation follows the SCAMPI appraisal requirements for CMMI. (See Fig. 4.13)

Chapter 5 Evaluation

5.1 Design and Set up of the evaluation

5.1.1 Methodology Framework

The target group in our scope is a special group of our focused companies which have CMMI experience in various methods of implementation. The companies are mostly site from Thailand, China and there are some representative from Switzerland, Vietnam, and Malaysia. The sampling number is 45 samples. We use a survey method by using questionnaire to participated team project based on their roles for instance, Software Engineering Process Group (SEPG), Project Management Officer (PMO), Project Manager (PM), System Analyst (SA) and the other roles with various years of experiences. The duration of this survey is five months between April 1 and 31 Aug 2012.

5.1.2 Literature review

We review many research proposals and results section based on an efficiency method in CMMI implementation. Its main goals are to situate the current study within the body of literature and to convey what knowledge and ideas have been established on related topic.

5.2 Defined Questionnaire

In our study, there are three main impact factors which are organization, project and integrated project planning and monitoring control characteristics. In each characteristic, we have their own objectives as presented as following.

1. **Part A: Organization Characteristic** is purposely to study the difference of geographic, size, team characteristics of an organization and the experienced CMMI.

2. **Part B: Project Characteristic** is aimed to study in each project based on; staff experiences, roles, software process improvement experiences, effort, duration, and software development process characteristic include customer's type and experiences.

3. **Part C: Integrated project planning and monitoring control characteristics** are deeply study the approximated values of the project both in planned and actual at the end. It is including the following topics like SW development project effort, cost, quality cost, duration, defects/reworks, and finally, its result and satisfaction rating.

The survey questions focus on comparing the conformance of Project Management between heavyweight and lightweight software development processes. The

success of both approaches in “Plan-driven” and “Agile-driven” are presented in terms of quality and performance.

The results will be used to provide guidance for future VSEs/SMEs’ Software Development Processes (SDP). It is very important that you return your completed questionnaire in order to produce representative and useful results.

Furthermore, we establish three types of questions (Multiple-Choice, Short-Answer and Numerical) based on above objectives by using questionnaire survey. The questionnaire is including instruction, condition, general information, and questionnaire. (See Table 5.1)

Table 5.1 The detail of questionnaire and Question type

No.	Detail of Information	Type of Question		
		Multiple -Choice	**Short-Answer**	**Numer -ical**
PART A: Organizational Characteristics				
1	Organizational Geographic		X	
2	Full time SW development employees			X
3	Active SW development projects			X
4	CMMI SW development projects			X
PART B: Project characteristics				
5	SW development project name		X	
6	SW development team experience			X
7	Major role in SW development team	X		
B1. Quality characteristics				
8	Software Process Improvement program	X		
9	Capability Maturity Model Integration (CMMI) program	X		
10	CMMI Target	X		
11	SW development project type	X		
B2. Customer characteristics				
12	Customer type	X		
13	Customer's domain software application	X		
B3. Project size characteristics				
14	Full time SW development team			X
15	SW development project effort (actual)			X
16	SW development project duration (actual)			X
17	SW development project effort (plan)			X

		Type of Question		
No.	**Detail of Information**	**Multiple -Choice**	**Short-Answer**	**Numer -ical**
B4. Software Development Process (SDP) characteristics				
18	SW development Process	X		
PART C: Integrated Project Planning and Monitoring Control				
19	SW development project effort (planned/actual-No.15)			X
20	Project manager effort (planned/actual)			X
21	SW development cost (planned/actual)			X
22	SW quality cost (planned/actual)			X
23	SW development project duration (plan/actual-No.16)			X
24	SW development defects/reworks (planned/actual)			X
25	SW development defects/reworks removal (planned/actual)			X
26	Understanding of Software Development Process (SDP)			X
27	SW development resources			X
28	SW development on-time deliverables			X
29	SW development in-budget deliverables			X
30	SW development establishes and maintains as planned			X
31	SW development monitor as planned			X
32	SW development achieve the corrective actions			X
33	Appropriate software process development (SDP) in a project			X
34	SW development establishes and manages stakeholders			X
35	SW development customer satisfaction rating			X
36	Employee satisfaction rating			X

PART A: Organizational Characteristics are purposely to understand the characteristic of an organization. There is including 4 questions;

1. “Where is your organization located? (Please name the country)”

Organizational geographic is namely the country which is located of its organization.

2. “How many “full-time equivalents” SW development employees are assigned to a project in your organization? (Please give the number)”

Full-time software developments employees are present a number of employees that assigned to software development project in each organization

3. “How many active SW development projects in your organization have been run since last year until now?”

Active software development projects are present current active software development projects which still running since last year in an organization.

4. “How many SW development projects have been run based on CMMI?”

Software development projects which is run together with CMMI in the organization.

PART B: Project Characteristics are purposely to understand the characteristic of project in an organization. There is including 4 parts as 14 questions;

5. “What is the name of the reference SW development project?”

Reference Software development projects is the project which is the project that is reference information for answering the questionnaire.

6. “What is the average team experience for this SW development project in Project management/ Software process improvement/ Domain expertise/ and Application language expertise? (Please specify the number of years)”

The average team experience is present year of experience in term of software development project in specific domains.

7. “Please specify your major role in this SW development project?”

The major role in software development project is present their specialist in specific roles in Software Engineering Process Group (SEPG), Project Management Officer (PMO), Project Manager (PM) and System Analyst (SA).

B1. Quality characteristics

8. “Did your project apply any software process improvement and/or quality management program?”

The question would like to know the current status of software process improvement and/or quality management program which is implemented in the organization.

9. “Did your project apply CMMI (Capability Maturity Model Integration)?”

The question would like to know the current status of CMMI (Capability Maturity Model Integration) which is implemented in the organization.

10. “What was your target to achieve for applying CMMI (Capability Maturity Model Integration)?”

The question would like to know type (ML: Maturity Level/ CL: Capability Level) and target level (ML1-Initial to ML5-Optimizing/CL0-Incomplete to CL5-Optimizing) of implemented CMMI (Capability Maturity Model Integration) or did not apply CMMI in the organization.

11. “What was your project type?”

The question would like to know the project type of its software development which is classified in development, maintenance or none of above.

B2. Customer characteristics

12. “Please specify the kind of customer from your reference project?”

The question would like to know the customer type of its software development which is classified in government or non-government organization.

13. “What was the domain of the customer’s software application?”

The domain of the customer’s software application is classified in Services (Business, Construction, Entertainment, Finance, Medical, Retail, Telecommunication), Manufacturing (Product, Oils-Energy), Education and training, Research and development or none of above.

B3. Project size characteristics

14. “How many employees have been in the SW development project team? (Please give the number)”

The question would like to know a number of employees which is working in software development project team.

15. “What was the approximate project effort? (Please give the number in Person)”

The question would like to know a number of efforts (Person-months) in term of person-month unit which is working in software development project team. The calculation is based on the following rules: “1 full-time employee working in 1 month is approximately equal to 1 staff * 8 hours * 22 days so that 1 person-month = 22 person-days = 176 person-hours”

16. “What was the approximate project duration? (Please give the number in Person-months)”

The question would like to know a number of project duration in term of month which is present the project duration in software development project team.

17. “Please enter the estimated size of the software based on the applied estimation model. (Please select only one model, numbers with only two decimal places)”

The question would like to know an estimated of the project size in the following items; the past experience estimations (Year), Source lines of code-SLOC (KLOCs), and Function points (FPs).

B4. Software Development Process (SDP) characteristics

18. “What was the major applied SDP in your SW development project? (Please select only one)”

The question would like to know a major type of applied software development process (SDP) characteristic in the software development project in the plan-driven (Waterfall, Prototype, Joint Application Development-JAD)/Rapid Application Development-RAD, Unified Process or Spiral) and also in Agile-driven (Crystal Clear, Extreme Programming-XP, Scrum, Feature Driven Development-FDD, Dynamic Systems Development Method-DSDM, Adaptive Software Development-ASD) or there is none of both SDP.

PART C: Integrated Project Planning and Monitoring Control are purposely to understand the planned and the actual result of software development project which is implement “CMMIbyScrum”. There is including 2 parts as 18 questions;

C1. The approximate values of the itemed project (as planned and actual values)

19. “What was the project effort? (In person-month)” [from question no. 15]

The question would like to know a number of efforts (man-hour) as planned and actual values in term of person-month unit which is working in software development project team. The calculation is based on the following rules: “1 full-time employee working in 1 month is approximately equal to 1 staff * 8 hours * 22 days so that 1 person-month = 22 person-days = 176 person-hours”

20. “What was the project manager effort? (In person-month)”

The question would like to know a number of efforts (man-hour) of the project manager as planned and actual values in term of person-month unit which is working in software development project team. The calculation is based on the following rules: “1 full-time employee working in 1 month is approximately equal to 1 staff * 8 hours * 22 days so that 1 person-month = 22 person-days = 176 person-hours”

21. “What was the approximate total SW development cost? (in $ USD)”

The question would like to know a number of software development cost as planned and actual values in US Dollar unit.

22. “What was the approximate total SW quality cost? (in $ USD)”

The question would like to know a number of quality cost which project mainly paid for preventing cost as planned and actual values in US Dollar unit.

23. “What was the project duration? (In month(s))” [from question no. 16]

The question would like to know a number of project duration as planned and actual values in month unit which is present the project duration in software development project team.

24. “What was the average number of defects/reworks for software product in the User Acceptance Test (UAT) phase?”

The question would like to know the defects/reworks per unit (DPU) for software product in the User Acceptance Test (UAT) phase which user mainly detective defects for reworks as planned and actual values in DPU unit.

25. “What was the average percentage of defects/reworks removal?”

The question would like to know the removal defects/reworks per unit (DPU) for software product in the User Acceptance Test (UAT) phase which project mainly collective defects as planned and actual values in DPU unit.

C2. The ordinary values of the itemed project (as planned and actual values)

It is defined as following values; 5-Definitely, 4-Probably, 3-Be unsure, 2-Probably not, 1-Definitely not and 0-Not applicable.

26. “Did you understand well to apply the SDP in plan-driven or agile-driven?”

The question would like to know the level of understanding on how to apply the software development process (SDP) in plan-driven or Agile-driven which is a body of knowledge to implement CMMIbyScrum based on these research questions (RQ).

27. “Did your project receive adequate resources both quality & quantity?”

The question would like to know the level of resources both in quality and quantity on the readiness to apply the software development process (SDP) in plan-driven or Agile-driven which are basic requirement resources to implement CMMIbyScrum based on these research questions (RQ).

28. “Did your project deliver the product on-time (+/- 10%)?”

The question would like to know the degree of on time deliver product when implementing the software development process (SDP) in plan-driven or Agile-driven or CMMIbyScrum based on the RQ1 which is related to the mutual exclusives approaches of Agile and CMMI for process improvement program.

29. “Did your project deliver the product in budget (+/- 5%)?”

The question would like to know the degree of exceeded budget in product when implementing the software development process (SDP) in plan-driven or Agile-driven or CMMIbyScrum based on the RQ1 which is related to the mutual exclusives approaches of Agile and CMMI for process improvement program.

30. “Did your project establish and maintain plans that defined the project activities?”

The question would like to know the degree of establish and maintain plans regarding to defined project activities when implementing the software development process (SDP) in plan-driven or Agile-driven or CMMIbyScrum based on the RQ2 which is related to the gap between Agile-Scrum and the project management of CMMI.

31. “Has your project been monitored periodically to follow the SDP and plans?”

The question would like to know the degree of periodic monitored plans regarding to defined project activities when implementing the software development process (SDP) in plan-driven or Agile-driven or CMMIbyScrum based on the RQ2 which is related to the gap between Agile-Scrum and the project management of CMMI.

32. “Has your project been achieved the corrective actions after monitored?”

The question would like to know the degree of the achievement of corrective actions regarding to monitored project activities when implementing the software development process (SDP) in plan-driven or Agile-driven or CMMIbyScrum based on the RQ3 which is related to the combination between Agile-Scrum and the project management of CMMI to close the gap.

33. “Did you appropriately select software process development (SDP) for your project?”

The question would like to know the degree of the appropriate selected software process development (SDP) when implementing the plan-driven or Agile-driven or CMMIbyScrum based on the RQ3 which is related to the combination between Agile-Scrum and the project management of CMMI to close the gap.

34. Did your project establish and manage the involvement of the relevant stakeholders?

The question would like to know the degree of the established and managed the involvement of the relevant stakeholders when implementing the software development process (SDP) in plan-driven or Agile-driven or CMMIbyScrum based on the RQ3 which is related to the combination between Agile-Scrum and the project management of CMMI to close the gap.

35. “Did your project meet customer expectation (customer satisfaction rating >80%)?”

The question would like to know the degree of customer satisfaction rating when implementing the software development process (SDP) in plan-driven or Agile-driven or CMMIbyScrum based on the RQ4 which is related to the benefit to synergize between Agile-Scrum and the project management of CMMI.

36. Did you have a good employee satisfaction rating (employee satisfaction rating >80%)?

The question would like to know the degree of employee satisfaction rating when implementing the software development process (SDP) in plan-driven or Agile-driven or CMMIbyScrum based on the RQ4 which is related to the benefit to synergize between Agile-Scrum and the project management of CMMI.

5.3 Data Collection

The result of the questionnaires based on the data collecting process from Table 5.1 above, we use the focus group of representative to send an invitation to respond the questionnaire. The source of the data is mainly from China 68% (32 papers), Thailand 23% (11 papers), Switzerland (2 papers), and Vietnam, Malaysia each 2% (2 papers) are show a number of located organizations respectively from all 47 questionnaires. The survey of the questionnaires is during July, 2012 to December, 2012. Moreover, we not only give one questionnaire for one organization, but also, we try to give to other specific roles for more informative dimension. However, the focused roles are Software Engineering Process Group (SEPG), Project Management Officer (PMO), Project Manager (PM), System Analyst (SA) because these roles are very crucial impact to organization processes and project management. The result of the questionnaire is present as following in Appendix B.

In summary, the majority information of the questionnaire in overall picture is show in section as follow;

PART A: Organizational Characteristics

The full-time equivalents of software development employees are between 10-15 persons.

PART B: Project Characteristics

The average team experience has difference based on roles as project manager has 5-10 years of experiences, Software Process Improvement specialist has 2-10 years of experiences, Application Domain Expert and Application Language Expert also has 2-5 years of experiences. Most of them are Project Manager (PM).

B1. Quality characteristics

The software development projects apply software process improvements and/or quality management programs likes ML 3 (Maturity Level) in CMMI (Capability Maturity Model Integration).

B2. Customer characteristics

The customers of the software development projects are both from non-government customer and government customer in service domains likes communication, resource management, and transportation.

B3. Project size characteristics

There are 5-10 employees in the SW development project team. The project efforts is approximately 2-5 person-months (1 person-month = 22 person-days = 176 person-hours). The software development project durations are less than 6 months. The software sizes in the software development projects are approximately 20-30 KLOCs.

B4. Software Development Process (SDP) characteristics

The majorities of Software Development Processes (SDP) in the software development projects are Plan-Driven (ex. Waterfall) and Agile-Driven (ex. Scrum).

PART C: Integrated Project Planning and Monitoring Control

The perspective for Agile-Driven is shown better performance to manage an effort than Plan-Driven based on the comparison of the of project effort usage between planned and actual at the end. However, in term of project manager effort, Plan-Driven is shown better performance than Agile-Driven. It may be because these project managers are new in deploying Agile-driven projects.

In term of managing software budget, Agile-Driven is shown better performance to manage cost than Plan-Driven. However, in term of managing software process improvement budget, Plan-Driven is shown better performance to manage cost than Agile-Driven.

The managing of software development project in term of time variation, a number of the defects/reworks per unit (DPU), and a number of the defects/reworks removal per unit (DPU), Agile-Driven is shown better performance to manage a schedule than Plan-Driven.

The majority of projects are satisfied in topic of receive adequate resources both quality and quantity, appropriately select software process development (SDP), establish and manage the involvement of the relevant stakeholders, deliver the product on-time, deliver the product in budget, establish and maintain plans that defined the project activities, monitored periodically to follow the SDP and plans, achieved the corrective actions after monitored, meet customer expectation and finally, get a good employee satisfaction rating.

Moreover, there is interesting information regarding to Question 26 to Question 36 by shading based on personal experiences which are related to plan-driven and Agile-driven from all 43 and 3 questionnaires which are defined as follow value; 5 is Definitely, 4 is Probably, 3 is Be unsure, 2 is Probably not, 1 is Definitely not and 0 is Not applicable.

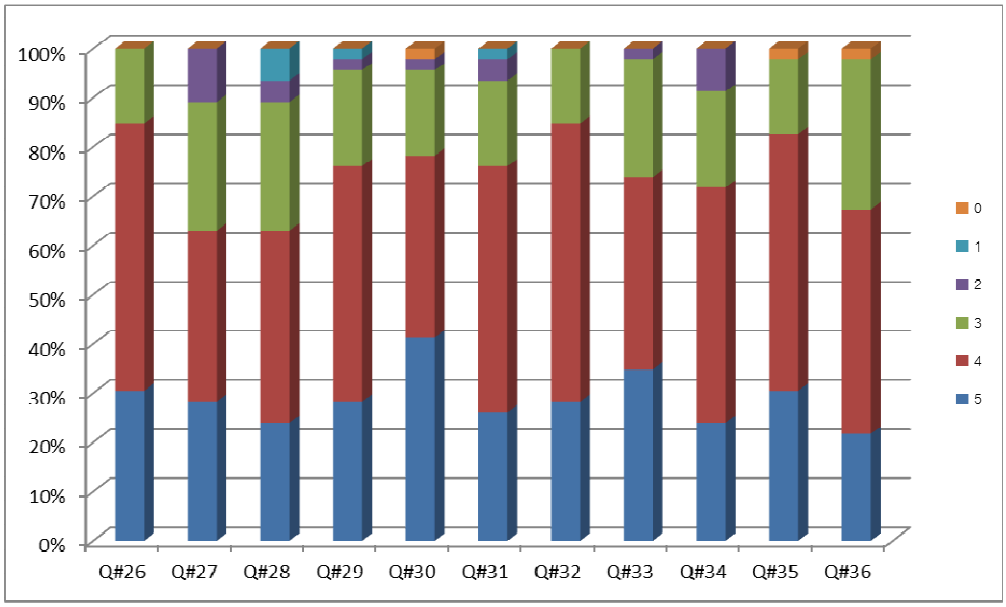


Figure 5.1 The level of defined value from Question 26 to 36

The conclusion of satisfaction values are show in Figure 5.1. The level of defined value is classified in 6 levels; Not applicable (0), Definitely not (1), Probably not (2), Be unsure (3), Probably (4), Definitely (5). On the other axle, The 11 questions from 26 to 36 are related to understand software development process-SDP (Q#26), resource (Q#27), time (Q#28), budget (Q#29), plan (Q#30), monitor (Q#31), corrective actions (Q#32), appropriately select software development process-SDP (Q#33), establish and manage stakeholder (Q#34), meet the expectation of customer satisfaction (Q#35) and employee satisfaction (Q#36) respectively.

5.4 Data Analysis

Data analyze is a systematic investigation of the data and their flow in a real or planned system. Our data analyze has two part; the first part present the hypothesis based on the information from the result of an empirical study which are related to specific goal (SG) in PM@CMMI which is including Integrated Project Planning and Monitoring Control (PP, PMC and IPM) to fulfill the Light-Weight Project Management SCAMPI Assessment Model (LWPM-SAM). The second part is presented the comparison value between Plan and Agile driven from question number 26 to 36 to support the research question that related to the CMMIbyScrum Framework (CMMISF). The results are presented in the following part.

The first part analyze by “Pearson” and “ANOVA” as a statistical measurement to show the linear relationship between paired data and to analyze the differences between group means and their associated procedures.

The Pearson’s correlation coefficient is a statistical measure of the strength of a linear relationship between paired data. It is a measure of the linear correlation (dependence) between two variables X and Y, giving a value between +1 and −1 inclusive, where 1 is total positive correlation, 0 is no correlation, and −1 is total negative correlation. It is widely used in the sciences as a measure of the degree of linear dependence between two variables. It was developed by Karl Pearson from a related idea introduced by Francis Galton in the 1880s. A value of 1 implies that a linear equation describes the relationship between X and Y perfectly, with all data points lying on a line for which Y increases as X increases. A value of −1 implies that all data points lie on a line for which Y decreases as X increases. A value of 0 implies that there is no linear correlation between the variables. Correlation is an effect size and so we can verbally describe the strength of the correlation using the guide that Evans (1996) suggests for the absolute value of r: 0.00-0.19 - “very weak”, 0.20-0.39 - “weak”, 0.40-0.59 - “moderate”, 0.60-0.79 - “strong”, and 0.80-1.00 - “very strong”.

To do this we test the null hypothesis, H_0, that there is no correlation in the population against the alternative hypothesis, H_1, that there is correlation; our data will indicate which of these opposing hypotheses is most likely to be true.

Regarding to specific goal (SG) in PM@CMMI, these Integrated Project Management (IPM) list is IPM, PP, and PMC as Specific Goal and Practice Summary are presented as following;

Integrated Project Management (IPM)

SG 1 Use the Project's Defined Process

SG 2 Coordinate and Collaborate with Relevant Stakeholders

Project Planning (PP)

SG 1 Establish Estimates

SG 2 Develop a Project Plan

SG 3 Obtain Commitment to the Plan

Project Monitoring and Control (PMC)

SG 1 Monitor the Project Against the Plan

SG 2 Manage Corrective Action to Closure

There are 7 hypothesizes; the hypothesis 1 (People size), the hypothesis 2 (Software size), the hypothesis 3 (Year of Experience) and the hypothesis 6 (Number of Project Team) based on Pearson's Correlation Analysis method. And the hypothesis 4 (CMMI Project), the hypothesis 5 (Domain of the customer's software application) and the hypothesis 7 (Software Development Process) are based on ANOVA method.

The hypothesis 1 is related to IPM.SG 2 (Coordinate and Collaborate with Relevant Stakeholders); People size depends on Integrated Project Planning and Monitoring Control from the question no. 2 and 26, 28, 29, 35, 36.

The hypothesis 2 is related to PMC.SG 1 (Monitor the Project Against the Plan); Software size depends on Integrated Project Planning and Monitoring Control from the question no. 3 and 27, 31, 34.

The hypothesis 3 is related to PP.SG 3 (Obtain Commitment to the Plan); Year of Experience depends on Integrated Project Planning and Monitoring Control from the question no. 6 and 22, 28, 29, 35, 36.

The hypothesis 4 is related to PMC.SG 2 (Manage Corrective Action to Closure); CMMI Project depends on Integrated Project Planning and Monitoring Control from the question no. 9.

The hypothesis 5 is related to PP.SG 2 (Develop a Project Plan); Domain of the customer's software application depends on Integrated Project Planning and Monitoring Control from the question no. 13.

The hypothesis 6 is related to PP.SG 1 (Establish Estimates); Number of Project Team depends on Integrated Project Planning and Monitoring Control from the question no. 14 and 26, 28, 29, 35, 36.

The hypothesis 7 is related to IPM.SG 1 (Use the Project's Defined Process); Software Development Process (SDP) depends on Integrated Project Planning and Monitoring Control from the question no. 18.

The results of all hypothesizes are present as following part.

Hypothesis 1

People size depends on Integrated Project Planning and Monitoring Control because a number of persons affect to IPM.SG 2 Coordinate and Collaborate with Relevant Stakeholders so that researcher assumes that;

H_0:	People size do not depend on Integrated Project Planning and Monitoring Control
H_1:	People size depends on Integrated Project Planning and Monitoring Control

In this hypothesis, the data analysis result between Q2 is related Q26, Q28, Q29, Q35 and Q36 based on the reason as following;

Q2	“How many “full-time equivalents” SW development employees are assigned to a project in your organization? (Please give the number)”
Q26	"Did you understand well to apply the SDP (plan- or agile-driven)?" : Coordinate and Collaborate with Relevant Stakeholders affect to an understanding of applying the performance SDP.
Q28	"Did your project deliver the product on-time (+/- 10%)?" : Coordinate and Collaborate with Relevant Stakeholders affect to the performance of delivering the product on-time.
Q29	"Did your project deliver the product in budget (+/- 5%)?" : Coordinate and Collaborate with Relevant Stakeholders affect to the performance of delivering the product in budget.
Q35	"Did your project meet customer expectation (customer satisfaction rating >80%)?" : Coordinate and Collaborate with Relevant Stakeholders affect to the customer expectation.
Q36	"Did you have a good employee satisfaction rating (employee satisfaction rating >80%)?" : Coordinate and Collaborate with Relevant Stakeholders affect to the employee expectation.

The following is data analysis data table between Q2 and Q26, Q28, Q29, Q35 and Q36;

Table 5.2 The hypothesis 1: People size is depend on Integrated Project Planning and Monitoring Control

		Accumulate of *Q26 Q28 Q29 Q35* and *Q36*	Q2
Accumulate of *Q26 Q28 Q29 Q35* and *Q36*	Pearson Correlation	1	.441
	Sig (1-tailed)		.001
Q2	Pearson Correlation	0.441	
	Sig (1-tailed)	.001	

The Significant value (Sig) is equal to 0.001 at 0.01 Significance level which Sig value is less than Significance level therefore this hypothesis is reject H_0 and accept H_1 then people size does depend on Integrated Project Planning and Monitoring Control which mean a number of persons affect to IPM.SG 2 Coordinate and Collaborate with Relevant Stakeholders.

Hypothesis 2

Software size depends on Integrated Project Planning and Monitoring Control because a size of software affects to PMC.SG 1 Monitor the Project Against the Plan so that researcher assumes that;

H_0:	Software size does not depend on Integrated Project Planning and Monitoring Control
H_1:	Software size depends on Integrated Project Planning and Monitoring Control

In this hypothesis, the data analysis result between Q3 and Q27, Q31 and Q34 based on the reason as following;

Q3	"How many active SW development projects in your organization have been run since last year until now?"
Q27	"Did your project receive adequate resources both quality & quantity?" : Monitor the Project Against the Plan affect to the receiving of adequate resources both quality & quantity.
Q31	Has your project been monitored periodically to follow the SDP and plans? : Monitor the Project Against the Plan affect to the periodical monitoring to follow the SDP and plans.
Q34	Did your project establish and manage the involvement of the relevant stakeholders? : Monitor the Project Against the Plan affect to the establishing and managing the involvement of the relevant stakeholders.

The following is data analysis data table between Q3 and Q27, Q31 and Q34;

Table 5.3 The hypothesis 2: Software size is depend on Integrated Project Planning and Monitoring Control

		Accumulate of Q27 Q31 and Q34	Q3
Accumulate of Q27 Q31 and Q34	Pearson Correlation	.001	1
	Sig (1-tailed)	.498	
Q3	Pearson Correlation	1	.001
	Sig (1-tailed)	.498	

The Significant value (Sig) is equal to 0.498 at 0.01 Significance level which Sig value is more than Significance level therefore this hypothesis is accept H_0 then software size does not depend on Integrated Project Planning and Monitoring Control which mean a size of software does not affect to PMC.SG 1 Monitor the Project Against the Plan.

Hypothesis 3

Year of Experience depends on Integrated Project Planning and Monitoring Control because experiences affect to PP.SG 3 Obtain Commitment to the Plan so that researcher assumes that;

H_0:	Year of Experience does not depend on Integrated Project Planning and Monitoring Control
H_1:	Year of Experience depends on Integrated Project Planning and Monitoring Control

In this hypothesis, the data analysis result between Q6 and Q26, Q28, Q29, Q35 and Q36 based on the reason as following;

Q6	"What is the average team experience for this SW development project in Project management, Software process improvement, Domain expertise, Application language expertise (Please specify the number of years)"
Q26	"Did you understand well to apply the SDP (plan- or agile-driven)?" : Obtain Commitment to the Plan affect to an understanding of applying the performance SDP.
Q28	"Did your project deliver the product on-time (+/- 10%)?" : Obtain Commitment to the Plan affect to the performance of delivering the product on-time.
Q29	"Did your project deliver the product in budget (+/- 5%)?" : Obtain Commitment to the Plan affect to the performance of delivering the product in budget.
Q35	"Did your project meet customer expectation (customer satisfaction rating >80%)?" : Obtain Commitment to the Plan affect to the customer expectation.
Q36	"Did you have a good employee satisfaction rating (employee satisfaction rating >80%)?" : Obtain Commitment to the Plan affect to the employee expectation.

The following is data analysis data table between Q6 and Q26, Q28, Q29, Q35 and Q36;

Table 5.4 The hypothesis 3: Year of Experience is depend on Integrated Project Planning and Monitoring Control

		Accumulate of Q26 Q28 Q29 Q35 and Q36	Q3
Accumulate of Q26 Q28 Q29 Q35 and Q36	Pearson Correlation	.124	1
	Sig (1-tailed)	.208	
Q3	Pearson Correlation	1	.124
	Sig (1-tailed)	.208	

The Significant value (Sig) is equal to 0.208 at 0.01 Significance level which Sig value is more than Significance level therefore this hypothesis is accept H_0 then a year of experience does not depend on Integrated Project Planning and Monitoring Control which mean experiences do not affect to PP.SG 3 Obtain Commitment to the Plan.

Hypothesis 4

CMMI Project depends on Integrated Project Planning and Monitoring Control because project that implementing CMMI affects to PMC.SG 2 Manage Corrective Action to Closure so that researcher assumes that;

H_0:	CMMI Project does not depend on Integrated Project Planning and Monitoring Control
H_1:	CMMI Project depends on Integrated Project Planning and Monitoring Control

In this hypothesis, the data analysis result of Q9 by ANOVA analysis based on the reason as following;

Q9	"Did your project apply CMMI (Capability Maturity Model Integration)?"

The following is data analysis data table between Q9 in ANOVA analysis;

Table 5.5 The hypothesis 4: CMMI Project is depend on Integrated Project Planning and Monitoring Control

	df	*F*	Sig.
Between Groups	14	.929	.541
Within Groups	30		
Total	44		

The Significant value (Sig) is equal to 0.541 at 0.05 Significance level which Sig value is more than Significance level therefore this hypothesis is accept H_0 then CMMI project does not depend on Integrated Project Planning and Monitoring Control which mean project that implementing CMMI does not affect to PMC.SG 2 Manage Corrective Action to Closure.

Hypothesis 5

Domain of the customer's software application depends on Integrated Project Planning and Monitoring Control because its affect to PP.SG 2 Develop a Project Plan so that researcher assumes that;

H_0:	Domain of the customer's software application does not depend on Integrated Project Planning and Monitoring Control
H_1:	Domain of the customer's software application depends on Integrated Project Planning and Monitoring Control

In this hypothesis, the data analysis result of Q13 by ANOVA analysis based on the reason as following;

Q13	"What is the domain of the customer's software application?"

The following is data analysis data table between Q13 in ANOVA analysis;

Table 5.6 The hypothesis 5: Domain of the customer's software application is depend on Integrated Project Planning and Monitoring Control

	df	*F*	Sig.
Between Groups	6	.523	.787
Within Groups	38		
Total	44		

The Significant value (Sig) is equal to 0.787 at 0.05 Significance level which Sig value is more than Significance level therefore this hypothesis is accept H_0 then Domain of the customer's software application does not depend on which means type of customer's software application does not affect to PP.SG 2 Develop a Project Plan.

Hypothesis 6

Number of Project Team depends on Integrated Project Planning and Monitoring Control because a number of project teams affect to PP.SG 1 Establish Estimates so researcher assumes that;

H_0:	Number of Project Team does not depend on Integrated Project Planning and Monitoring Control
H_1:	Number of Project Team is depends on Integrated Project Planning and Monitoring Control

In this hypothesis, the data analysis result between Q14 and Q26, Q28, Q29, Q35 and Q36 based on the reason as following;

Q14	“How many “full-time equivalents” SW development employees are assigned to a project in your organization? (Please give the number)”
Q26	"Did you understand well to apply the SDP (plan- or agile-driven)?" : Establish Estimates affect to an understanding of applying the performance SDP.
Q28	"Did your project deliver the product on-time (+/- 10%)?" : Establish Estimates affect to the performance of delivering the product on-time.
Q29	"Did your project deliver the product in budget (+/- 5%)?" : Establish Estimates affect to the performance of delivering the product in budget.
Q35	"Did your project meet customer expectation (customer satisfaction rating >80%)?" : Establish Estimates affect to the customer expectation.
Q36	"Did you have a good employee satisfaction rating (employee satisfaction rating >80%)?" : Establish Estimates affect to the employee expectation.

The following is data analysis data table between Q14 and Q26, Q28, Q29, Q35 and Q36;

Table 5.7 The hypothesis 6: Number of Project Team is depend on Integrated Project Planning and Monitoring Control

		Accumulate of Q26 Q28 Q29 Q35 and Q36	Q3
Accumulate of Q26 Q28 Q29 Q35 and Q36	Pearson Correlation	.532	1
	Sig (1-tailed)	.000	
Q14	Pearson Correlation	1	.000
	Sig (1-tailed)	.532	

The Significant value (Sig) is equal to 0.532 at 0.05 Significance level which Sig value is more than Significance level therefore this hypothesis is accept H_0 then a number of project team does not depend on Integrated Project Planning and Monitoring Control which mean a number of project team do not affect to PP.SG 1 Establish Estimates.

.

Hypothesis 7

Software Development Process (SDP) depends on Integrated Project Planning and Monitoring Control because SDP affects to IPM.SG 1 Use the Project's Defined Process so that researcher assumes that;

H_0:	Software Development Process (SDP) does not depend on Integrated Project Planning and Monitoring Control
H_1:	Software Development Process (SDP) depends on Integrated Project Planning and Monitoring Control

In this hypothesis, the data analysis result of Q18 by ANOVA analysis based on the reason as following;

Q18	"What is the domain of Software Development Process (SDP)?"

The following is data analysis data table between Q18 in ANOVA analysis;

Table 5.8 The hypothesis 7: Software Development Process (SDP) is depend on Integrated Project Planning and Monitoring Control

	df	*F*	Sig.
Between Groups	14	.599	.843
Within Groups	29		
Total	44		

The Significant value (Sig) is equal to 0.843 at 0.05 Significance level which Sig value is more than Significance level therefore this hypothesis is accept H_0 then Software Development Process (SDP) does not depend on Integrated Project Planning and Monitoring Control which mean SDP that deploying does not affect to IPM.SG 1Use the Project's Defined Process.

In summary, there is only a number of persons that affect to Coordinate and Collaborate with Relevant Stakeholders (IPM.SG 2). The others likes; a size of software and Monitor the Project Against the Plan (PMC.SG 1), experiences and Obtain Commitment to the Plan (PP.SG 3), implementing CMMI and Manage Corrective Action to Closure (PMC.SG 2), type of customer's software application and Develop a Project Plan (PP.SG 2), a number of project team and Establish Estimates (PP.SG 1) and SDP that deploying and Use the Project's Defined Process (IPM.SG 1) do not relevant and affect to Integrated Project Planning and Monitoring Control.

In this case, we conclude all hypotheses in table 5.9 (Correlation Analysis method) and 5.10 (ANOVA method) as following;

In term of Correlation Analysis method, the most familiar measure of dependence between two quantities is the Pearson product-moment correlation coefficient, or "Pearson's correlation coefficient", commonly called simply "the correlation coefficient". It is obtained by dividing the covariance of the two variables by the product of their standard deviations. Karl Pearson developed the coefficient from a similar but slightly different idea by Francis Galton. [J.L.Rodgers and W.A.Nicewander]

In additional, Analysis of variance (ANOVA) is a collection of statistical models used to analyze the differences between group means and their associated procedures. [R.A.Fisher] ANOVA provides a statistical test of whether or not the means of several groups are equal, and therefore generalizes the t-test to more than two groups. ANOVAs are useful in comparing (testing) three or more means (groups or variables) for statistical significance.

Table 5.9 The hypothesis 1, 2, 3 and 6 based on Correlation Analysis method

No.	Hypothesis	Result
Hypothesis 1	The achievement of the Coordinate and Collaborate with Relevant Stakeholders goal (IPM.SG 2) depends on People size	A number of persons affect to Coordinate and Collaborate with Relevant Stakeholders (IPM.SG 2)
Hypothesis 2	The achievement of the Monitor the Project Against the Plan goal (PMC.SG 1) depends on software size	A size of software does not affect to Monitor the Project Against the Plan (PMC.SG 1)
Hypothesis 3	The achievement of the Obtain Commitment to the Plan goal (PP.SG 3) depends on year of experience	Year of experiences do not affect to Obtain Commitment to the Plan (PP.SG 3)
Hypothesis 6	The achievement of the Establish Estimates goal (PP.SG 1) depends on number of project team	A number of project team does not affect to Establish Estimates. (PP.SG 1)

Table 5.10 The hypothesis 4, 5 and 7 based on ANOVA method

No.	Hypothesis	Result
Hypothesis 4	The achievement of the Manage Corrective Action to Closure goal (PMC.SG 2) depends on CMMI project	The project that implementing CMMI does not affect to Manage Corrective Action to Closure (PMC.SG 2)
Hypothesis 5	The achievement of the Develop a Project Plan goal (PP.SG 2) depends on domain of the customer's software application	A domain of the customer's software application does not affect to Develop a Project Plan (PP.SG 2)
Hypothesis 7	The achievement of the Domain of Use the Project's Defined Process (IPM .SG 1) depends on Software Development Process (SDP)	Software Development Process (SDP)'s software application does not affect to the Project's Defined Process (IPM .SG 1)

However, there are limited qualities of data from 45 questionnaires because projects which are implementing in Scrum are less in term of number so that the result of these hypotheses from table 5.9 and 5.10 are not show the result as expected [Scrum.org]. There is only hypothesis 1 that shows a number of persons affect to coordinate and collaborate with relevant stakeholders. The hypothesis 2, 3, 4, 5 and 6 are expected that the results of an achievement of their specific goals (SG) should affect related to software size, year of experience, project which is implement CMMI and number of project team. Moreover, the hypothesis 7 is open to understand the effect of the achievement of the domain of develop a project plan goal and type of customer's software application.

The second part is presented the comparison value between Plan and Agile driven from question number 26 to 36 and summarize on how the research questions (RQ) is related to the questionnaire. The results of the questionnaires are presented as followed.

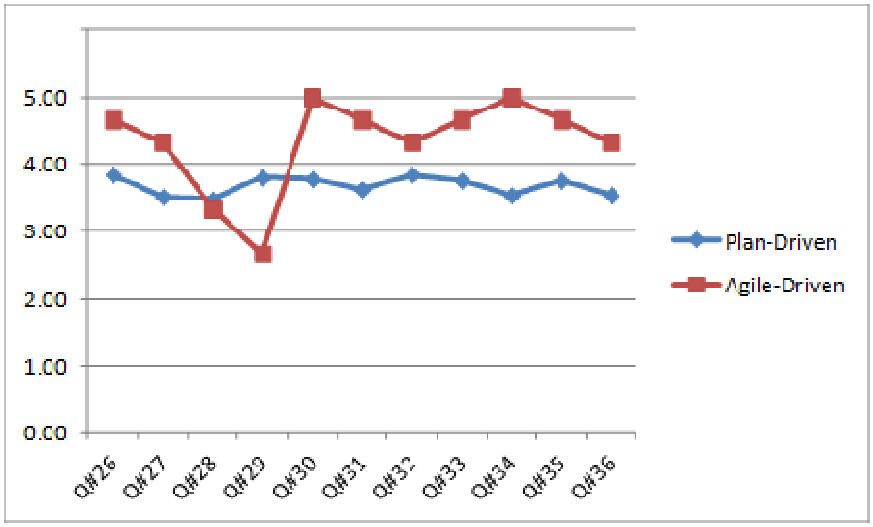


Figure 5.2 The comparison value between Plan and Agile driven related to question No.26 to 36

From Figure 5.2 presents the comparison between Plan and Agile driven related to question No.26 to 36 which are compose of the level of defined value in the level of defined value in applying the SDP (Q#26), receiving adequate resources both quality & quantity (Q#27), delivering the product on-time and in budget (Q#28 & Q#29), establishing and maintaining plans that defined the project activities (Q#30), periodical monitoring to follow the SDP and plans (Q#31), achieving the corrective actions after monitored (Q#32), selecting software process development (SDP) for your project (Q#33), establishing and managing the involvement of the relevant stakeholders (Q#34), meeting customer expectation (Q#35) and having a good employee satisfaction rating (Q#36).

However, delivering the product on-time and in budget (Q#28 & Q#29) of Agile driven are show unexpected result. [Scrum.org] There are some reasons why Agile-driven shows satisfied values less than Plan-driven for instance; unrealistic estimates of required time or resources based on Scrum method, and/or a project manager still makes the mistake of equating time on task to duration. This is precisely the danger for larger projects, when too much time has already elapsed before being terminated. Long time-spans can cause the continued supply of products or services that are no longer in use or of low priority. [Triodor Software, NL]

The challenge of scaling large is Scrum Scalable or Scrum of Scrums. The Scrum of Scrums meeting is an important technique in scaling Scrum to large project teams. These meetings allow clusters of teams to discuss their work, focusing especially on areas of overlap and integration. For a perfectly balanced project, each of the teams would conduct (simultaneously or sequentially) its own daily scrum meeting. Each team would then designate one person to also attend a scrum of scrums meeting. The decision of who to send should belong to the team. Usually the person chosen should be a technical contributor on the team likes a programmer, tester, database administrator, designer, and so on rather than a product owner or Scrum Master. [scrumalliance.org]

Chapter 6 Conclusion and Future Research

The target group in our scope is a special group of our focused companies which have CMMI experiences in various methods of implementation. The sampling number is 45 samples. We use a survey method by using questionnaire to participated team project based on their roles for instance, Software Engineering Process Group (SEPG), Project Management Officer (PMO), Project Manager (PM), System Analyst (SA) and the other roles.

In this chapter, the summary of research questions and the summary of evaluation are presented in Appendix C: The summary of the majority group from the questionnaire. Moreover, the implication for practice of CMMI and Scrum adoption and continuous SPI for VSEs/SMEs also are displayed. Open questions and future research and limitation of the study are provided in the last part. Moreover, the research questions (RQ) is related to the questionnaire which presents in term of satisfied percentage from Probably (4) and Definitely (5) groups based on the information from Appendix B as follow;

The questions which are related to the readiness to give some information for the questionnaire are from Question No. 26 and 27; there is 84% understand well to apply the SDP in plan-driven or agile-driven. And 63% of the project receive adequate resources both quality & quantity.

The questions which are related to the RQ1: Are Agile and CMMI mutually exclusive approaches for process improvement program? This question tried to figure out whether Agile approach & Scrum and Project Management in CMMI model are synergized from Question No. 28 and 29; there is 63% of the project which deliver the product on-time (+/- 10% from the planned). And 76% of the project that deliver the product in budget (+/- 5% from the planned).

The questions which are related to the RQ2: What is a gap between Scrum and the project management requirements of CMMI? This question is based on how much compatible and gap items of Agile approach & Scrum and Project Management in CMMI model are from Question No. 30 and 31; there is 78% of the project which establishes and maintain plan that defined the project activities. And 76% of the project been monitored periodically to follow the SDP and plans.

The questions which are related to the RQ3: How to combine Scrum and the project management, CMMI to close the gap? This question tried to find out to identify on how can we fulfill the CMMI goals via Agile approach & Scrum which different or lacks for implementing CMMI when develop by using Agile approach & Scrum from Question No. 32, 33 and 34; there is 85% of the project which achieved the corrective actions after monitored, 74% of the project select appropriately software process

development (SDP). And 72% of the project establish and manage the involvement of the relevant stakeholders.

Lastly, the questions which are related to the RQ4: What is the benefit to synergize Scrum and the project management requirements of CMMI? This question tried to identify empirical experiences to prove that a development by Agile approach & Scrum can be useful to gain for reaching the CMMI goal from Question No. 35 and 36; there is 82% of the project which meet customer expectation (customer satisfaction rating >80%). And 68% of the project have a good employee satisfaction rating (employee satisfaction rating >80%).

In overall picture, the research questions is affect to the CMMIbyScrum Framework (CMMISF) which is based on the CMMI model from Figure 3.9 (The comparison of Abstract model and CMMI/CMMISF). The CMMISF focuses on Integrated Project Planning and Monitoring Control (PP, PMC and IPM) and the conformance value which shown in Figure 3.8 are apply to implement in LWPM-SAM (Light-Weight Project Management SCAMPI Assessment Model). Finally, the self-assessment tool called "SPIALS" - Software Process Improvement Adaptive Learning System is bringing to use as a SCAMPI-C for adaptive learning appraisal tool.

6.1. Answers to Research Questions

6.1.1 The Summary of the Results Questions (RQ1/RQ2/RQ3/RQ4)

RQ1. Are Agile and CMMI mutually exclusives approaches for process improvement program?

Agile & Scrum approach with Project Management in CMMI model are synergized in term of compatibility dimension. The result of all mapping analysis of the relationship between Project Management (PP, PMC, IPM) area in CMMI and Scrum are performed the percentage of conformity as 82%, 95% and 85% respectively which is highest values among all process areas (PA).

RQ2. What is a gap between Scrum and the project management requirements of CMMI?

The compatible and gap items of Agile & Scrum approach with Project Management in CMMI model are mapped in term of compliance dimension. Regarding to Project Management (PP, PMC, IPM) area, Daily Scrum Meeting (DSM) performs 60% of practiced conformity to PMC activities. Sprint Planning Meeting performs 86% and 90% of practiced conformity to PP and IPM activities respectively. The rests of Scrum practices are Sprint Review Meeting (SRM) and Sprint Retrospective (SR) show 30% of practiced conformity to PMC, and 10% of practiced conformity to IPM respectively. However, there is 7% of gap that Scrum has to fulfill to complete SCAMPI.

On the other aspect, regarding to Project Management (PP, PMC, IPM) area, Sprint Backlog (SB) performs 71% and 90% of conformity to PP and IPM artifacts. Burn Down Chart (BDC) performs 80% of conformity to PMC artifacts.

RQ3. How to combine Scrum and the project management, CMMI to close the gap?

The identify on fulfill the CMMI goals via Agile & Scrum approach with Project Management in CMMI is shows the different or lacks for implementing CMMI when develop by using Agile & Scrum approach in term of capability/maturity dimension.

To fulfill the gap of Project Planning (PP), improvement recommendations are provided. "Estimate Effort and Cost" (SP1.4), "Identify Project Risks" (SP2.2), "Plan Data Management" (SP2.3) and "Plan Needed Knowledge and Skills" (SP2.5) are needed to fulfill some practices and artifacts for satisfying by SCAMPI.

To fulfill the gap of Project Monitoring and Control (PMC) in Scrum Framework, improvement recommendations are provided. "Monitor Data Management" (SP1.4) is needed to fulfill some practices and artifacts for satisfying by SCAMPI.

To fulfill the gap of Integrated Project Management (IPM) in Scrum Framework, improvement recommendations are provided. "Use Organizational Process Assets for Planning Project Activities" (SP1.2), "Establish the Projects Work Environment" (SP1.3) and "Contribute to Organizational Process Assets" (SP1.7) are needed to fulfill some practices and artifacts for satisfying by SCAMPI.

To fulfill the gap of Generic Goal (GG) in Scrum Framework, improvement recommendations are provided. "Establish an Organizational Policy" (GP2.1), "Plan the Process" (GP2.2), "Assign Responsibility" (SP2.4), "Train People" (GP2.5), "Identify and Involve Relevant Stakeholders" (SP2.7), "Objectively Evaluate Adherence" (GP2.9), Review Status with Higher Level Management" (SP2.10), "Establish a Defined Process" (GP3.1) and "Collect Process Related Experiences" (GP3.2) are needed to fulfill some practices and artifacts for satisfying by SCAMPI.

RQ4. What is the benefit to synergize Scrum and the project management requirements of CMMI?

The empirical experiences are proven that a development by Agile & Scrum approach with Project Management in CMMI can be useful to gain for reaching the CMMI and SCAMPI goal in term of profitability dimension.

The comparison value between Plan and Agile driven related to question No.26 to 36 shows the satisfaction of Agile-Driven (CMMIbyScrum) is higher than Plan-Driven in all dimensions (establish and maintain plan, periodical monitor, achieve the corrective actions, appropriately select SPD, establish and manage the involvement of the relevant stakeholders, meet the customer and employee expectation) except to deliver the product on time (+/- 10%) and deliver the product in time (+/- 5%). This is ambiguous information from our expectation.

The summary of the majority group from the questionnaire also shows the performance of Agile-Driven (CMMIbyScrum) is better than Plan-Driven in the angle of controlling the project effort, software budgeting, and project duration. However, the project manager effort and software process improvement budgeting are better controlled by Plan-Driven.

6.1.2 The summary of the Evaluation

In summary, the target group in our scope is a special group of our focused companies which have CMMI experience in various methods of implementation. However, a number of Agile-Driven organizations are few numbers. The total sampling number is 47 samples. We use a survey method by using questionnaire to participated team project based on their roles for instance, Software Engineering Process Group (SEPG), Project Management Officer (PMO), Project Manager (PM), System Analyst (SA) and the other roles. The summary of majority group is shown in Appendix C (The summary of the majority group from the questionnaire).

6.2. Implications

6.2.1 Implication for the Practice of CMMI and Scrum Adoption

Regarding to our literature and empirical study, CMMI and Agile are partially mutual exclusives approaches for process improvement program. However, in Project Management approach (PP, PMC and IPM), the conformity percentage is significantly upward compatibility for CMMI and Scrum adopting.

The conformance value of CMMISF (Especially for Project Management approach likes PP, PMC and IPM) shows highest values among all process areas. In conclusion, the practice of CMMI and Scrum adoption are moderately concinnity to practice based on appraisal model.

The purpose to create a Scrum base Project Management approach that conforms to CMMI (PM@CMMI) presents in Chapter 3 related to the conformity percentage of CMMI process area with Scrum framework. The Project Monitoring and Control (PMC), Integrated Project Management (IPM), Project Planning (PP) are show Conformance Rating (CR) at "+++" with Conformance Value (CV); 95, 85 and 82 respectively. These are the highest CR and CV values among all process areas. Regarding to this information, not only it is present that the Scrum base Project Management approach is supported by project management area in CMMI (PM@CMMI) but also it is an initiation point to design software development framework based on CMMI-Scrum namely “Light-Weight Project Management Approach (LWPM)”.

6.2.2 Implication for Continuous SPI for VSEs/SMEs

Regarding to this dissertation, Continuous SPI for Very Small Enterprises/ Small Medium Enterprises (VSEs/SMEs) based on CMMIbyScrum which is focusing in Project Management approach is satisfactory.

The comparison of Abstract model and CMMI/CMMISF shows software development model which is stand for “CMMI for Scrum Framework (CMMISF)”. The Light-Weight Project Management SCAMPI Assessment Model (LWPM-SAM) is proposed to deploy as appraisal model which practically fit for VSEs/SMEs organizations through Software Process Improvement Adaptive Learning System (SPIALS) which is the alternative tool that use for self-assessment in software process improvement program.

The purpose to design software development framework based on CMMI-Scrum namely “Light-Weight Project Management Approach (LWPM)” which is a component of “CMMI-by-Scrum” Framework (CMMISF) which is a conceptual framework for an effective practice which is needed to be qualified by the Standard CMMI Appraisal Method for Process Improvement (SCAMPI).

Moreover, the purpose to offer the cost/efficiency LWPM for VSEs/SMEs as a self-assessment model are propose based on values for CMMI appraisal to compete with other CMMI organizations is presented by “Light-Weight Project Management SCAMPI Assessment Model (LWPM-SAM)”. It is designed for VSEs/SMEs companies for their self-assessment as SCAMPI C level which can implement by Light-Weight SCAMPI Assessment Tool (SPIALS).

Finally, the purpose to evaluate LWPM based on a CMMI conformance focusing on Project management perspective need to have more data from companies which are deploy LWPM based on CMMISF in software industrial. It is to ensure the appropriate characteristics of this framework to introduce to VSEs/SMEs organizations for implementing in their software development procedure.

6.3. Limitation of the Study

Regarding to our literature and empirical study, there are limited not only number of data but also quality of data, too. In our evaluation, the total questionnaire is 47 sets. We have quite good data in term of multi-choice, however, in term of short-answer and numerical types, the quantity of these answers are rarely in a good mood. The important limitation is a number of projects that deploy by Agile or Scrum practices.

6.4. Open Questions and Future Research

The future research has two alternatives. Firstly, research can be focused on others categories of CMMI to literature for more mutually exclusives approaches for process improvement program. Secondly, the opportunities to blend CMMI with other Agile methods are interesting to benefit to not only VSEs/SMEs organizations but also for applying in the large organization. Lastly, in empirical study, the number of sampling should be enough regarding to statistical definition.

Moreover, CMMISF, Light-Weight Project Management SCAMPI Assessment Model (LWPM-SAM) and Software Process Improvement Adaptive Learning System (SPIALS) are well match and interesting to deploy for self-assessment appraisal in

VSEs/SMEs organization. However, the evaluation of LWPM based on a CMMI conformance focusing on Project management perspective is needed to implement and affirm in the future research. Then the value-added of deployment the LWPM-SAM for VSEs/SMEs is expected to be exhibit.

Bibliography

1. [1.1] **IEEE Computer Society**, *"Guide to the Software Engineering Body of Knowledge (SWEBOK-V3.0)"*, 2013.

2. [1.2] **IEEE Computer Society**, *"IEEE Standard for Developing Software Life Cycle Processes (IEEE 1074-97)"*, 1998.

3. [1.3] **Keri Pearlson and Carol Saunders**, *"Managing and Using Information Systems: A strategic approach"*, paperback ISBN 0-471-71538-7, John Wiley & Sons, Inc. Press (2006).

4. [1.4] **W. W. Royce**, *"Managing the development of large software systems: Concepts and techniques,"* in Proc. WESCON, Aug. 1970.

5. [1.5] **B. W. Boehm**, *"Software engineering,"* IEEE Trans. Comput., vol.C-25, pp. 1226-1241, Dec. 1976.

6. [1.6] **Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas**, *"agilemanifesto.org"*, 2001.

7. [1.7] **Highsmith, J. and Cockburn, A.**, *"Agile software development: the business of innovation"*, Computer Journal, vol.34, no.9, pp.120,127, Sep 2001.

8. [1.8] **Steven L. Goldman, Roger N.Nagel, and Kenneth Preiss**, *"Agile Competitors and Virtual Organizations: Strategies for Enriching the Customer"*, Van Nostrand Reinhold, October 25, 1994.

9. [1.9] **Beck, Kent; et al.**, *"Manifesto for Agile Software Development"*, Agile Alliance (2001) Retrieved 14 June 2010.

10. [1.10] **Ken Schwaber**, *"Agile Project Management with Scrum"*, ISBN 978-0-7356-1993-7, Microsoft Press, 2004.

11. [1.11] **scrumalliance**, *"Why Scrum?"*, "scrumalliance.org", 2014.

12. [1.12] **Schwaber Ken, Beedle Mike**, *"Agile software development with Scrum"*, O'Reilly Media, Inc., ISBN 0735637903, 9780735637900, 2009.

13. [1.13] **International Organization for Standardization**, *"SPICE (Software Process Improvement and Capability Determination)"*, ISO/IEC Std TR 15504-9: 1988.

14. [1.14] **Robin B. Hunter, Richard H. Thayer**, *"Software Process Improvement"*, IEEE Computer Society, ISBN 0-7695-0999-1, 2001.

15. [1.15] **PAULK Mark C., WEBER Charles V., CURTIS Bill, CHRISSIS Mary Beth**, *"The capability maturity model : guidelines for improving the software process"*, Software Engineering Institute (SEI), 1995.

16. [1.16] **Mark C. Paulk, Charles V. Weber, Suzanne M. Garcia, Mary Beth Chrissis, Marilyn Bush**, *"Key Practices of the Capability Maturity ModelSM, Version 1.1"* (CMU/SEI-93-TR-25), Carnegie Mellon University, 1993.

17. [1.17] S**oftware Engineering Process Management Program-Carnegie Mellon University**, *"CMMI for Development, Version 1.3"*, CMU/SEI-2010-TR-033, 2010.

18. [2.1] **Linda Rising and Norman S. Janoff**, *"The Scrum Software Development Process for Small Teams"*, IEEE SOFTWARE, July/August, 2000.

19. [2.2] **Armin Preis**, *"Integration Evaluation of Scrum and CMMI-Theoretical Survey at the Project-Management Level"*, arminpreis.at, 2012.

20. [2.3] **M. A. Awad**, *"A Comparison between Agile and Traditional Software Development Methodologies"*, The University of Western Australia, 2005.

21. [2.4] **Ana Sofia C. Marcal, Bruno Celso C. de Freitas, Felipe S. Furtado Soares, Arnaldo D. Belchior**, *"Mapping CMMI Project Management Process Areas to SCRUM Practices"*, 31st IEEE Software Engineering Workshop, pp. 13-22, 2007.

22. [2.5] **Ana Sofia C. Marcal, Bruno Celso C. de Freitas, Felipe S. Furtado Soares, Maria Elizabeth S. Furtado, Teresa M. Maciel, Arnaldo D. Belchior**, *"Blending Scrum practices and CMMI project management process areas"*, 4:17–29, Innovations Syst Softw Eng, Springer, 2008.

23. [2.6] **Hillel Glazer; et al.**, *"CMMI® or Agile: Why Not Embrace Both!"*, Technical Note, CMU/SEI-2008-TN-003, November 2008.

24. [2.7] **P.J. Rundle, R.G. Dewar**, *"Using Return on Investment to Compare Agile and Plan-Driven Practices in Undergraduate Group"*, ICSE 2006, May 20-28, 2006.

25. [2.8] **David F. Rico**, "What is the Return on Investment (ROI) of Agile Methods?", "davidfrico.com".

26. [2.9] **Neil Potter and Mary Sakry**, *"Implementing Scrum (Agile) and CMMI PLEMENTING SCRUM (AGILE) and CMMI® Together"*, "processgroup.com", Post, vol.16 no.2, March 2009.

27. [2.10] **Minna Pikkarainen and Annukka Mäntyniemi**, *"An Approach for Using CMMI in Agile Software Development Assessments: Experiences from Three Case Studies"*, SPICE 2006 conference, Luxemburg, May 2006.

28. [2.11] **Martin Fritzsche and Patrick Keil**, *"Agile Methods and CMMI: Compatibility or Conflict?"*, e-Informatica Software Engineering Journal, vol 1, issue 1, 2007.

29. [2.12] **Jeff Sutherland, Carsten Ruseng Jakobsen, Kent Johnson**, *"Scrum and CMMI Level 5: The Magic Potion for Code Warriors"*, ISBN 0-7695-2872-4, pp. 272-278, Agile Conference (AGILE), 2007.

30. [2.13] **Navarrete, F., P. Botella, and F. Xavier**, *"An Approach to Reconcile the Agile and CMMI Context in Product Line Development"*, Proceedings of the 1st International Workshop on Agile Product Line Engineering (APLE'06), Japan, 2006.

31. [2.14] **Viljan Mahnic, Natasa Zabkar**, *"Introducing CMMI Measurement and Analysis Practices into Scrum-based Software Development Process"*, International Journal of Mathematics and Computers in Simulation. , vol 1, issue 1, 2007.

32. [2.15] **Carsten Ruseng Jakobsen, Kent Aaron Johnson**, *"Mature Agile with a twist of CMMI"*, ISBN 978-0-7695-3321-6, pp. 212 - 217, Agile Conference (AGILE), 2008.

33. [2.16] **Julio Ariel Hurtado Alegrıa and Marıa Cecilia Bastarrica**, *"Implementing CMMI using a Combination of Agile Methods"*, CLEI Electronic Journal, vol.9 no.1, paper 7, June 2006.

34. [2.17] **Enrico Berardi and Luca Santillo**, *"COSMIC-based Project Management in Agile Software Development and Mapping onto related CMMI-Dev Process Areas"*, IWSM/MetriKon 2010.

35. [2.18] **Hillel Glazer**, *"Love and Marriage: CMMI and Agile Need Each Other"*, CROSSTALK The Journal of Defense Software Engineering, January/February 2010.

36. [2.19] **Chayakorn Piyabunditkul**, Apinporn Methawachananont, Sompol Chaimongkhon, Boonchai Charoendouysil, *"Accelerated Adoption CMMI by Agile Methodologies"*, TGGS-RWTH Aachen- NECTEC, In Proceeding(s) of the ProMAC Symposium 2009 (ProMAC2009), Bangkok, Thailand, October 28 – 30, 2009.

37. [2.20] **Chayakorn Piyabunditkul**, Nithipat Wongchingchai, Apinporn Methawachananont, *"Step forward CMMI-Project Management by optimized Scrum"*, TGGS-RWTH Aachen-NECTEC, In Proceeding(s) of the 5th International Conference on Project Management (ProMAC 2010), Makuhari Messe, Japan, October, 13-15, 2010.

38. [3.1] **M. Coram M.**, *"The impact of agile methods on software project management"*, Engineering of Computer-Based Systems on the Engineering of Computer-Based Systems (ECBS'05), 2005.

39. [3.2] **Boehm, B.**, *"Get Ready for Agile Methods, with Care"*, ISSN 0018-9162, IEEE Computer, vol.35 , issue 1, pp. 64-69, Jan 2002.

40. [3.3] **Mei-Chih HU**, *"Information Technology Service as a Key Supporting Industry of Hong Kong"*, Graduate Institute of Technology & Innovation Management, National Tsing Hua University, Taiwan, Hong Kong Innovation Project Conference, January 2009.

41. [3.4] **Ulrich Remus, Martin Wiener**, *"Critical Success Factors for Managing Offshore Software Development Projects"*, University of Canterbury.

42. [3.5] **J. Jurison**, *"Software Project Management: The Manager's View"*, Communications of AIS , vol. 2, article 17, September 1999.

Biography

Name-Surname (English)	Chayakorn Piyabunditkul, Mr. (CSPM)
Identification Number	3409900411375
Position	Engineer (Software Engineer Laboratory) National Electronics and Computer Technology Center (NECTEC) National Science and Technology Development Agency (NSTDA)
Organization Address	112 Thailand Science Park, Phahonyothin Road, Khlong Nueng, Khlong Luang, Pathum Thani, 12120, Thailand
Telephone	+66-2564-6900 Ext. 2422
Mobile	+66-81556-0117
Fax	+66-2564-6761
e-mail	chayakorn@nectec.or.th
Education	- M.SC. in Information Technology (Management Information System), Chulalongkorn University, Thailand, 1999. - B.ENG. in Survey Engineering, Chulalongkorn University, Thailand, 1996.
Special Expertise	- Software Project Management - Information Technology - Process Improvement
Experience related to Research Administration	1. JCSSE 2009 (The 6th International Joint Conference on Computer Science and Software Engineering) -Paper "Stepping up ISO 9001:2000 and ISO 90001:2008 to CMMI-DEV v1.2 at ML3 in Process Management Category", 13-15 May, 2009, Dusit Laguna Beach Resort, Phuket, Thailand. 2. SEPG Europe 2009 (Software Engineering Process Groups Europe 2009) -Poster "SPI with CMMI – Agile Implementation and Adaptive Tool (SCAAT)" SEPG Europe 2009 conference, Software Engineering Institute (SEI), Carnegie Mellon University (CMU), Jun 9-12, 2009, Prague, Czech Republic. 3. NECTEC ACE 2009 (NECTEC Annual Conference & Exhibitions 2009) -Poster "CMMI – Agility Based on SPI Advisory Tool (CAST)", Sep, 23-25, 2009, National Electronics and Computer Technology Center, Thailand Science Park, Pathumthani, Thailand. 4. ProMAC 2009 (Project Management Symposium 2009) -Paper "Accelerated Adoption CMMI by Agile Methodologies", Oct 28-30, 2009, The Society of Project Management (SPM), Japan, Bangkok, Thailand.

	5. SEPOW/APSEC 2009 (The 16th Asia-Pacific Software Engineering Conference) -Paper "Combining CMMI and Agile Approaches Based on a SPI Advisory Tool", Dec 1-3, 2009, Malaysian Software Engineering Interest Group and Malaysia Universities, Penang, Malaysia. 6. ProMAC 2010 (The 5th International Conference on Project Management) -Paper "Step forward CMMI-Project Management by optimized Scrum", Oct, 13-15, 2010, Makuhari Messe and Tokyo Disney Resort, Japan. 7. MYSEC 2011 (The 5th International Conference on Project Management) -Paper "SPIALS: A light-weight Software Process Improvement Self-Assessment Tool", 13–14 Dec 2011, The Puteri Pacific Johor, Malaysia. 8. EUROSPI 2012 (The 19th EuroSPI Conference) -Paper "SPI through Process Tailoring Framework with Objective-Based Scoring Metric", 25-27 Jun 2012, BENA Business Base Nineteen, Vienna, Austria. 9. EUROSPI 2013 (The 20th EuroSPI Conference) -Paper "Toward Systematic Approach for Objective Based Software Process Tailoring", 25-27 Jun 2013, Dundalk Institute of Technology, Ireland.

Appendix

Appendix A: Questionnaire

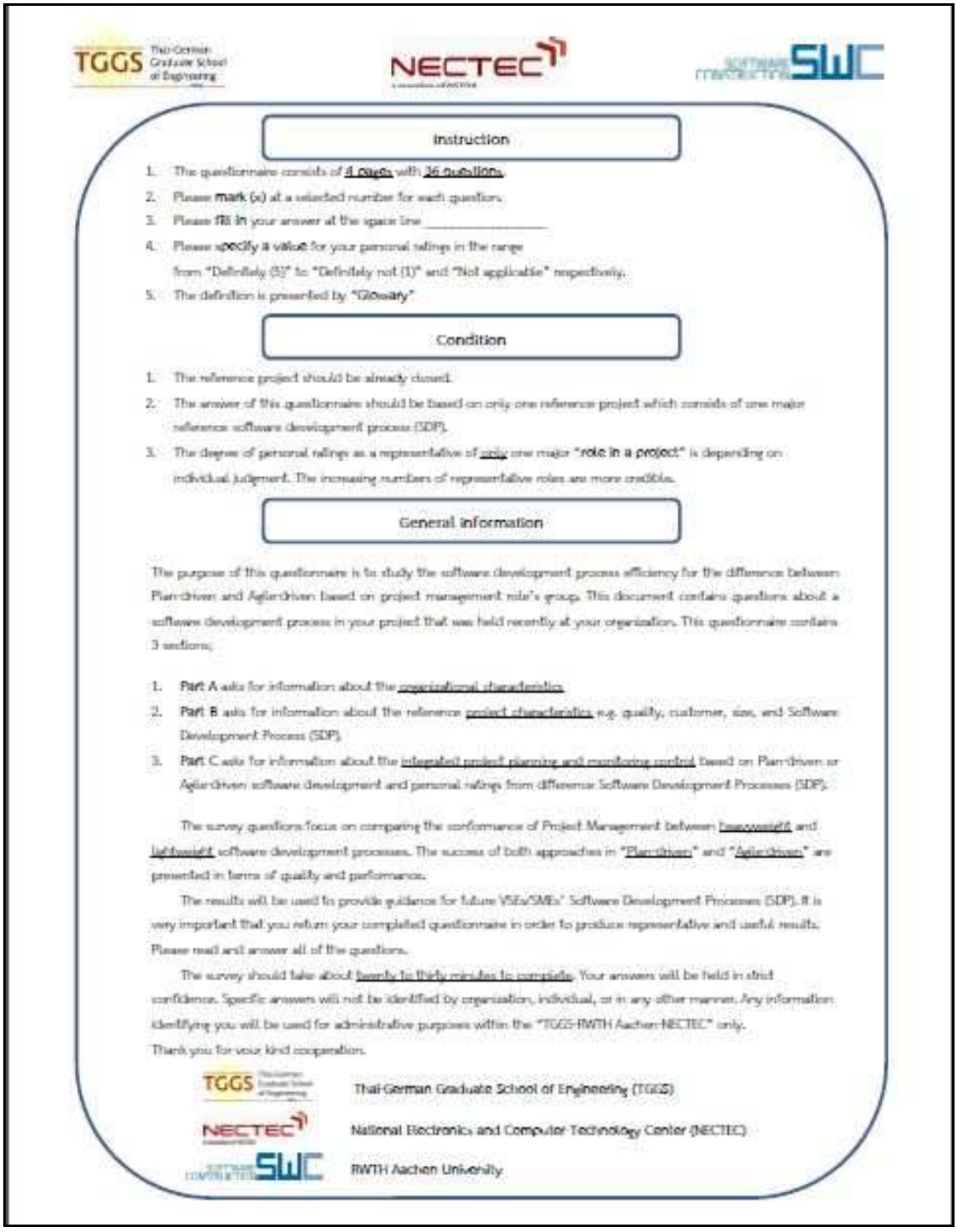
No. _ _ _

The survey of Design and Evaluation of CMMI conformance Lightweight Project Management Approach for VSEs/SMEs

Please return this form at your earliest convenience.
Use the enclosed envelop or send it to:

Chayakorn Piyabunditkul
Ph.D. candidate of TGGS-RWTH Aachen University-NECTEC

Software Engineering Laboratory: SWE
National Electronics and Computer Technology Center
112 Phahon Yothin Rd., Klong 1, Klong Luang, Pathumthani 12120, Thailand
Tel. 662-564-6900 ext. 2422 Fax. 662-5646761
chayakorn@nectec.or.th



TGGS Thai-German Graduate School of Engineering

NECTEC

SWC

Instruction

1. The questionnaire consists of 4 parts with 26 questions.
2. Please **mark (x)** at a selected number for each question.
3. Please **fill in** your answer at the space line _____
4. Please **specify a value** for your personal ratings in the range from "Definitely (5)" to "Definitely not (1)" and "Not applicable" respectively.
5. The definition is presented by "**Glossary**"

Condition

1. The reference project should be already closed.
2. The answer of this questionnaire should be based on only one reference project which consists of one major reference software development process (SDP).
3. The degree of personal ratings as a representative of only one major "**role in a project**" is depending on individual judgment. The increasing numbers of representative roles are more credible.

General information

The purpose of this questionnaire is to study the software development process efficiency for the difference between Plan-driven and Agile-driven based on project management role's group. This document contains questions about a software development process in your project that was held recently at your organization. This questionnaire contains 3 sections;

1. **Part A** asks for information about the organizational characteristics.
2. **Part B** asks for information about the reference project characteristics e.g. quality, customer, size, and Software Development Process (SDP).
3. **Part C** asks for information about the integrated project planning and monitoring control based on Plan-driven or Agile-driven software development and personal ratings from difference Software Development Processes (SDP).

The survey questions focus on comparing the conformance of Project Management between heavyweight and lightweight software development processes. The success of both approaches in "Plan-driven" and "Agile-driven" are presented in terms of quality and performance.

The results will be used to provide guidance for future VSEs/SMEs' Software Development Processes (SDP). It is very important that you return your completed questionnaire in order to produce representative and useful results. Please read and answer all of the questions.

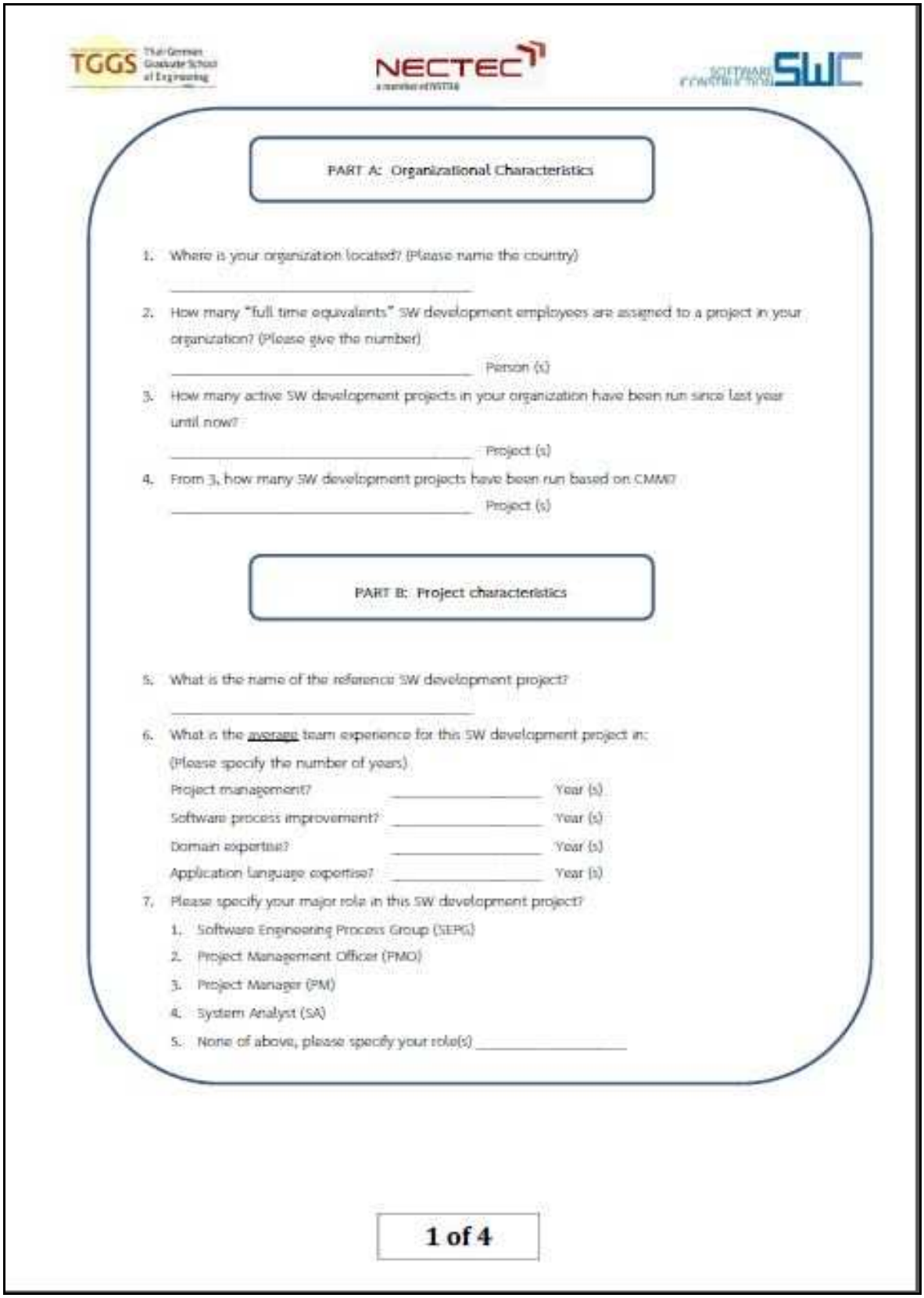
The survey should take about twenty to thirty minutes to complete. Your answers will be held in strict confidence. Specific answers will not be identified by organization, individual, or in any other manner. Any information identifying you will be used for administrative purposes within the "TGGS-RWTH Aachen-NECTEC" only.

Thank you for your kind cooperation.

TGGS Thai-German Graduate School of Engineering (TGGS)

NECTEC National Electronics and Computer Technology Center (NECTEC)

SWC RWTH Aachen University



PART A: Organizational Characteristics

1. Where is your organization located? (Please name the country)

   _____

2. How many "full time equivalents" SW development employees are assigned to a project in your organization? (Please give the number)

   _____ Person (s)

3. How many active SW development projects in your organization have been run since last year until now?

   _____ Project (s)

4. From 3, how many SW development projects have been run based on CMMI?

   _____ Project (s)

PART B: Project characteristics

5. What is the name of the reference SW development project?

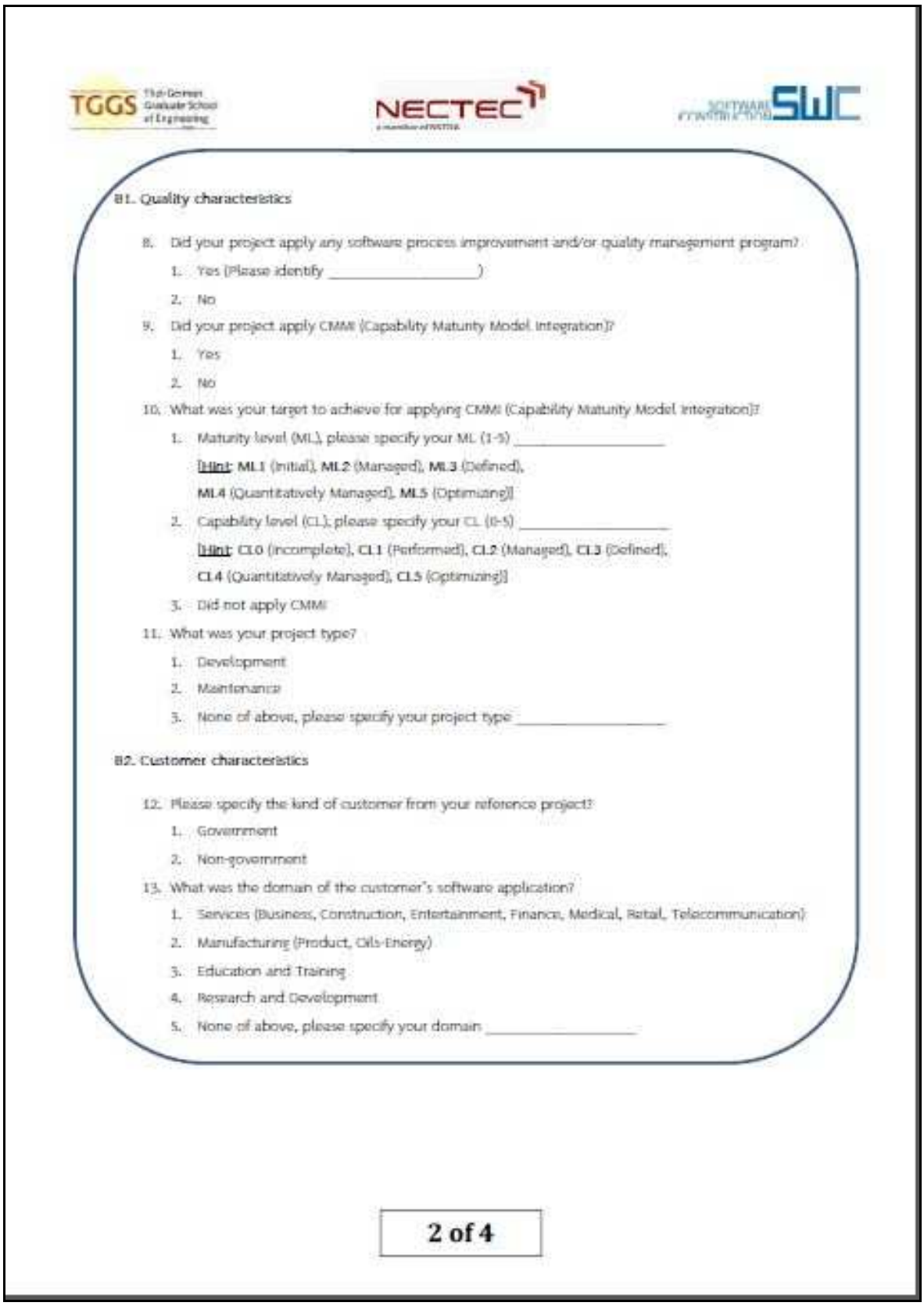
   _____

6. What is the average team experience for this SW development project in:
   (Please specify the number of years)

   Project management? _____ Year (s)
   Software process improvement? _____ Year (s)
   Domain expertise? _____ Year (s)
   Application language expertise? _____ Year (s)

7. Please specify your major role in this SW development project?
   1. Software Engineering Process Group (SEPG)
   2. Project Management Officer (PMO)
   3. Project Manager (PM)
   4. System Analyst (SA)
   5. None of above, please specify your role(s) _____

1 of 4



B1. Quality characteristics

8. Did your project apply any software process improvement and/or quality management program?
   1. Yes (Please identify _____)
   2. No
9. Did your project apply CMMI (Capability Maturity Model Integration)?
   1. Yes
   2. No
10. What was your target to achieve for applying CMMI (Capability Maturity Model Integration)?
   1. Maturity level (ML), please specify your ML (1-5) _____
      [Hint: **ML1** (Initial), **ML2** (Managed), **ML3** (Defined), **ML4** (Quantitatively Managed), **ML5** (Optimizing)]
   2. Capability level (CL), please specify your CL (0-5) _____
      [Hint: **CL0** (Incomplete), **CL1** (Performed), **CL2** (Managed), **CL3** (Defined), **CL4** (Quantitatively Managed), **CL5** (Optimizing)]
   3. Did not apply CMMI
11. What was your project type?
   1. Development
   2. Maintenance
   3. None of above, please specify your project type _____

B2. Customer characteristics

12. Please specify the kind of customer from your reference project?
   1. Government
   2. Non-government
13. What was the domain of the customer's software application?
   1. Services (Business, Construction, Entertainment, Finance, Medical, Retail, Telecommunication)
   2. Manufacturing (Product, Oils-Energy)
   3. Education and Training
   4. Research and Development
   5. None of above, please specify your domain _____

TGGS Thai-German Graduate School of Engineering NECTEC a member of NSTDA SWC Software Engineering

B3. Project size characteristics

14. How many employees have been in the SW development project team?
(Please give the number)

_____ Person (s)

15. What was the approximate project effort? (Please give the number)
[Hint: 1 full-time employee working in 1 month is approximately equal to
1 staff * 8 hours * 22 days so that 1 person-month = 22 person-days = 176 person-hours]

_____ Person-month

16. What was the approximate project duration? (Please give the number)

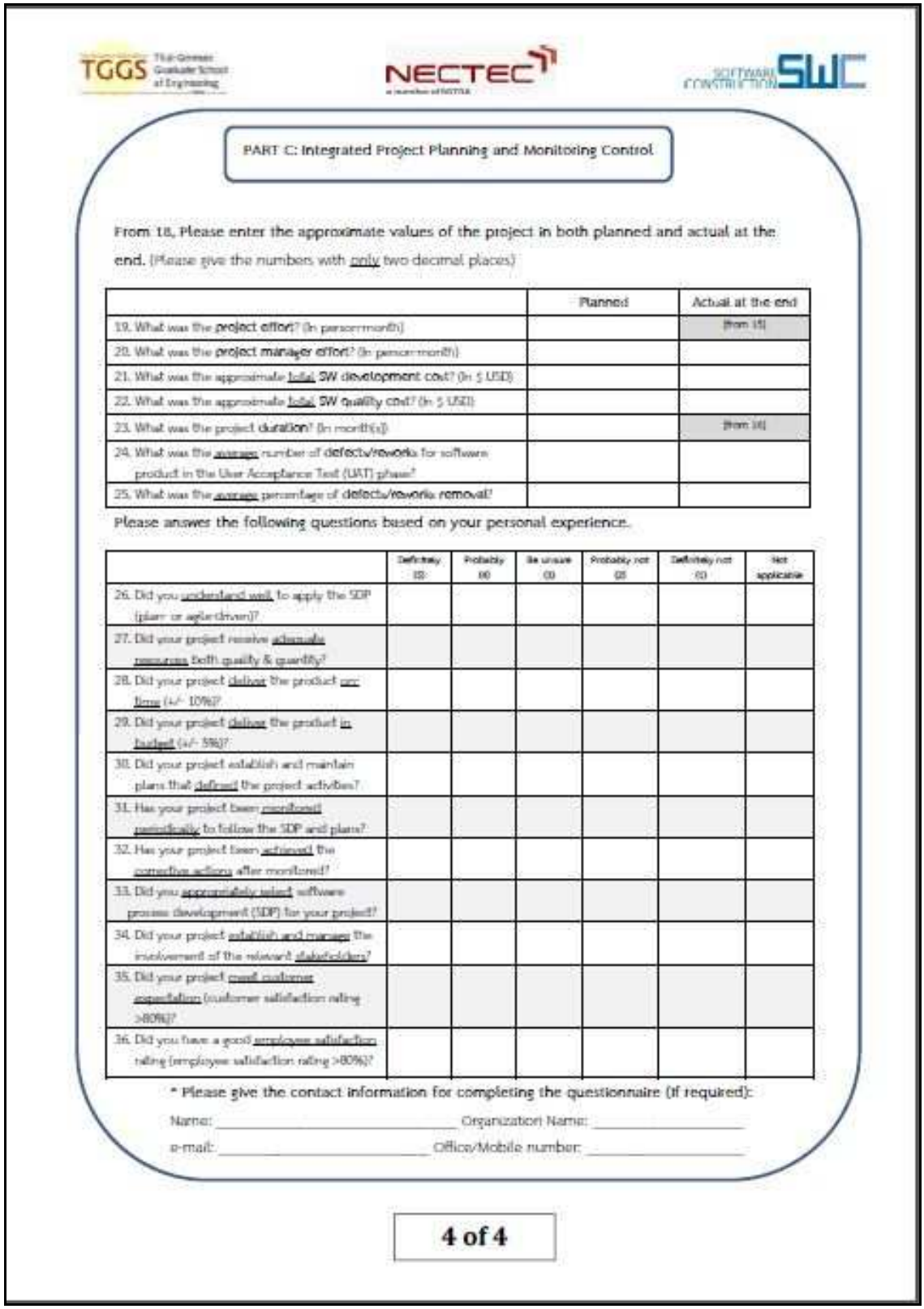
_____ Month (s)

17. Please enter the estimated size of the software based on the applied estimation model.
(Please select only one model, numbers with only two decimal places)

Estimation Model by	Unit (in person-month)
1. Past experience estimations _____	
2. Source lines of code (SLOC) _____ KLOCs	
3. Function points (FP) _____ FPs	
4. Other, please specify _____	
5. None	

B4. Software Development Process (SDP) characteristics

18. What was the major applied SDP in your SW development project? (Please select only one)
    1. Plan-driven - Waterfall
    2. Plan-driven - Prototype
    3. Plan-driven - Joint Application Development (JAD)/Rapid Application Development (RAD)
    4. Plan-driven - Unified Process
    5. Plan-driven - Spiral
    6. Agile-driven - Crystal Clear
    7. Agile-driven - Extreme Programming (XP)
    8. Agile-driven - Scrum
    9. Agile-driven - Feature Driven Development (FDD)
    10. Agile-driven - Dynamic Systems Development Method (DSDM)
    11. Agile-driven - Adaptive Software Development (ASD)
    12. None of above, please specify your major applied SDP _____



TGGS Thai-German Graduate School of Engineering

NECTEC a member of NSTDA

SOFTWARE CONSTRUCTION SWC

PART C: Integrated Project Planning and Monitoring Control

From 18, Please enter the approximate values of the project in both planned and actual at the end. (Please give the numbers with only two decimal places)

	Planned	Actual at the end
19. What was the **project effort**? (In person-month)		(from 15)
20. What was the **project manager effort**? (In person-month)		
21. What was the approximate total **SW development** cost? (In $ USD)		
22. What was the approximate total **SW quality** cost? (In $ USD)		
23. What was the project **duration**? (In month(s))		(from 16)
24. What was the average number of **defects/reworks** for software product in the User Acceptance Test (UAT) phase?		
25. What was the average percentage of **defects/reworks removal**?		

Please answer the following questions based on your personal experience.

	Definitely (5)	Probably (4)	Be unsure (3)	Probably not (2)	Definitely not (1)	Not applicable
26. Did you understand well to apply the SDP (plan- or agile-driven)?						
27. Did your project receive adequate resources both quality & quantity?						
28. Did your project deliver the product on-time (+/- 10%)?						
29. Did your project deliver the product in budget (+/- 5%)?						
30. Did your project establish and maintain plans that defined the project activities?						
31. Has your project been monitored periodically to follow the SDP and plans?						
32. Has your project been achieved the corrective actions after monitored?						
33. Did you appropriately select software process development (SDP) for your project?						
34. Did your project establish and manages the involvement of the relevant stakeholders?						
35. Did your project meet customer expectation (customer satisfaction rating >80%)?						
36. Did you have a good employee satisfaction rating (employee satisfaction rating >80%)?						

* Please give the contact information for completing the questionnaire (if required):

Name: _____ Organization Name: _____

e-mail: _____ Office/Mobile number: _____

4 of 4

TGGS Thai German Graduate School of Engineering

NECTEC a member of NSTDA

SOFTWARE CONSTRUCTION SWC

Glossary

1. **Agile-driven software development:** is a group of software development methodologies based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. These methodologies promote a disciplined project management process for software development that encourages frequent inspection and adaptation, teamwork, self-organization and accountability, a set of engineering best practices for rapid delivery of high-quality software. The Agile-driven methodologies are also known as "lightweight" methodologies. Early implementations of lightweight methods include Scrum (1995), Crystal Clear, Extreme Programming (1996), Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development Method (DSDM) (1995).
2. **ASD (Adaptive Software Development):** developed by James A. Highsmith, offers an agile and adaptive approach to high-speed and high-change software projects [19]. It is not possible to plan successfully in a fast moving and unpredictable business environment. In ASD, the static plan-design life cycle is replaced by a dynamic speculate-collaborate-learn life cycle.
3. **CMMI (Capability Maturity Model Integration):** is a process improvement approach that provides organizations with the essential elements of effective processes, which will improve their performance. CMMI-based process improvement includes identifying your organization's process strengths and weaknesses and making process changes to turn weaknesses into strengths.
4. **COCOMO (Constructive Cost Model):** is an algorithmic software cost estimation model developed by Barry W. Boehm. The model uses a basic regression formula with parameters that are derived from historical project data and current project characteristics.
5. **Crystal Clear:** is a member of the Crystal family of methodologies as described by Alistair Cockburn and is considered an example of an agile or lightweight methodology. Crystal Clear can be applied to teams of up to 6 or 8 co-located developers working on systems that are not life-critical. The Crystal family of methodologies focuses on efficiency and habitability as components of project safety. Crystal Clear focuses on people, not processes or artifacts.
6. **DSDM (Dynamic System Development Method):** was developed in the United Kingdom in the mid-1990. It is a blend of, and extension to, rapid application development and iterative development practices. Martin Fowler, one of the writers of Agile Manifesto, believes, "DSDM is notable for having much of the infrastructure of more mature traditional methodologies, while following the principles of the agile methods approach". The fundamental idea behind DSDM is to fix time and resources, and then adjust the amount of functionality accordingly rather than fixing the amount of functionality in a product, and then adjusting time and resources to reach that functionality.
7. **Extreme programming (XP):** has evolved from the problems caused by the long development cycles of traditional development models. The XP process can be characterized by short development cycles, incremental planning, continuous feedback, reliance on communication, and evolutionary design. With all the above qualities, XP programmers respond to changing environment with much more courage. XP team members spend few minutes on programming, few minutes on project management, few minutes on design, few minutes on feedback, and few minutes on team building many times each day. The term 'extreme' comes from taking these commonsense principles and practices to extreme levels.

8. FDD (Feature Driven Development): was used for the first time in the development of a large and complex banking application project in the late 90's.Unlike the other methodologies, the FDD approach does not cover the entire software development process but rather focuses on the design and building phases. The FDD approach includes frequent and tangible deliverables, along with accurate monitoring of the progress of the report.
9. Function Points (FP): are units of measure for functional size as defined within the IFPUG Functional Size Measurement (FSM) Method and it is the major global functional sizing methodology. FP is a unit of measurement to express the amount of business functionality an information system provides to a user. The cost (in dollars or hours) of a single unit is calculated from past projects.
10. Joint Application Development (JAD): is a technique for engaging a group or team of software developers, testers, customers, and prospective end-users in a collaborative requirements elicitation and prototyping effort (Wood and Silver 1995). JAD is quintessentially a technique for facilitating group interaction and collaboration.
11. Plan-driven software development: is a formal and well planned methodology for developing software applications. Plan-driven methodologies incorporate repeatability and predictability, well defined incremental process, extensive documentation, up front system architecture, detailed planning, process monitoring, resource controlling, risk management, verification and validation and user training. The Plan-driven methodologies are also known as "Heavyweight" methodologies or "Traditional" methodologies. These methodologies are based on a sequential series of steps, such as requirements definition, solution building, testing and deployment. Heavyweight methodologies require defining and documenting a stable set of requirements at the beginning of a project. There are many different heavyweight methodologies for instance: Waterfall, Prototyping, Spiral Model, Rapid Application Development and Unified Process.
12. Project Management Officer (PMO): is the department or group that defines and maintains the standards of process, generally related to project management, within the organization. The PMO strives to standardize and introduce economies of repetition in the execution of projects. The PMO is the source of documentation, guidance and metrics on the practice of project management and execution.
13. Project Manager (PM): is a professional in the field of project management. Project managers can have the responsibility of the planning, execution and closing of any project, typically relating to construction industry, architecture, Aerospace and Defense, computer networking, telecommunications or software development.
14. Prototyping: is a technique for providing a reduced functionality or a limited performance version of a software system early in its development (Balzer 1983, Budde 1984, Hekmatpour 1987). In contrast to the classic system life cycle, prototyping is an approach whereby more emphasis, activity, and processing are directed to the early stages of software development (requirements analysis and functional specification). Software prototypes come in different forms including throwaway prototypes, mockups, demonstration systems, quick-and-dirty prototypes, and incremental evolutionary prototypes (Hekmatpour 1987).

15. **Scrum:** is developed in the late 1980's and early 1990's primarily with object oriented development circles as a highly iterative development methodology. Scrum was developed by Ken Schwaber, Jeff Sutherland, and Mike Beedle (Beedle & Schwaber, 2001). Scrum is an iterative, incremental framework for project management and agile software development. Although Scrum was intended for management of software development projects, it can be used to run software maintenance teams, or as a general project/program management approach. Scrum concentrates on the management aspects of software development, dividing development into thirty day iterations (called 'sprints') and applying closer monitoring and control with daily scrum meetings.
16. **SDP (Software Development Process):** is a structure imposed on the development of a software product. Similar terms include software life cycle and software process. It is often considered a subset of system development life cycle. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process.
17. **Software Engineering Process Group (SEPG):** is an organization's focal point for software process improvement activities. These individuals perform assessments of organizational capability, develop plans to implement needed improvements, coordinate the implementation of those plans, and measure the effectiveness of these efforts.
18. **Software Process Improvement (SPI):** is a set of activities that will lead to a better software process, and thus higher quality software delivered in a more timely manner and it is a long-term investment. Clearly defined, measurable goals are necessary to provide guidance and to assist in developing tactics for improvement.
19. **Source lines of code (SLOC):** is software metric used to measure the size of a software program by counting the number of lines in the text of the program's source code. SLOC is typically used to predict the amount of effort that will be required to develop a program, as well as to estimate programming productivity or maintainability once the software is produced.
20. **The spiral:** has many cycles. The radial dimension represents the cumulative cost incurred in accomplishing the steps done so far and the angular dimension represents the progress made in completing each cycle of the spiral. Each cycle in the spiral begins with the identification of objectives for that cycle and the different alternatives are possible for achieving the objectives and the imposed constraints. The next step in the spiral life cycle model is to evaluate these different alternatives based on the objectives and constraints. This will also involve identifying uncertainties and risks involved. The next step is to develop strategies that resolve the uncertainties and risks. This step may involve activities such as benchmarking, simulation and prototyping. Next, the software is developed by keeping in mind the risks. Finally the next stage is planned.
21. **SW development project:** is Software development project.
22. **UAT (User Acceptance Test):** is a formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system. [ISTQB]
23. **Unified Process:** is a popular iterative and incremental software development process framework. The best-known and extensively documented refinement of the Unified Process is the Rational Unified Process (RUP). The Unified Process is not simply a process, but rather an extensible framework which should be customized for specific organizations or projects.
24. **VSEs/SMEs (Very Small Enterprises/Small Medium Enterprises):** are companies with up to 10 employees, small companies employ up to 50 workers, whilst medium-sized enterprises contain up to 250 employees.
25. **Waterfall:** is a sequential design process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance.

Appendix B: The result of the questionnaire

The result of the questionnaire is present as following;

PART A: **Organizational Characteristics** are purposely to understand the characteristic of an organization. There is including 4 questions;

1. "Where is your organization located? (Please name the country)"

China, Thailand, Switzerland, Vietnam and Malaysia are show a number of located organizations as 68%, 23%,4%,2% and 2% respectively from all 47 questionnaires.

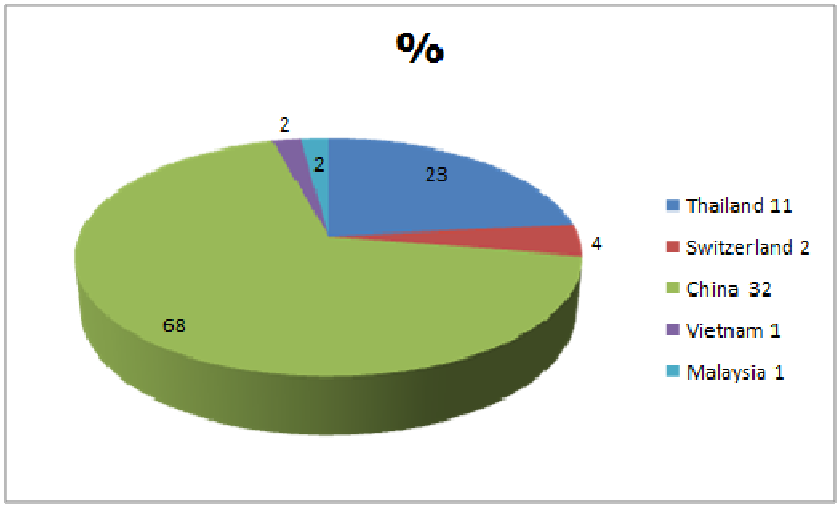


Figure B.1 The percentage of organizational Geographic

2. "How many "full-time equivalents" SW development employees are assigned to a project in your organization? (Please give the number)"

The full-time equivalents of software development employees are show a number of sizing in term of Medium Enterprise (ME), Very Small Enterprise (VSE), Small Enterprise (SE), Large Enterprise (LE) as 62%, 26%, 11% and 2% respectively from all 47 questionnaires which are defined as follow value; VSE <=5, 5>SE<=10, 10>ME<=15, and LE >15 in term of full-time software development employees.

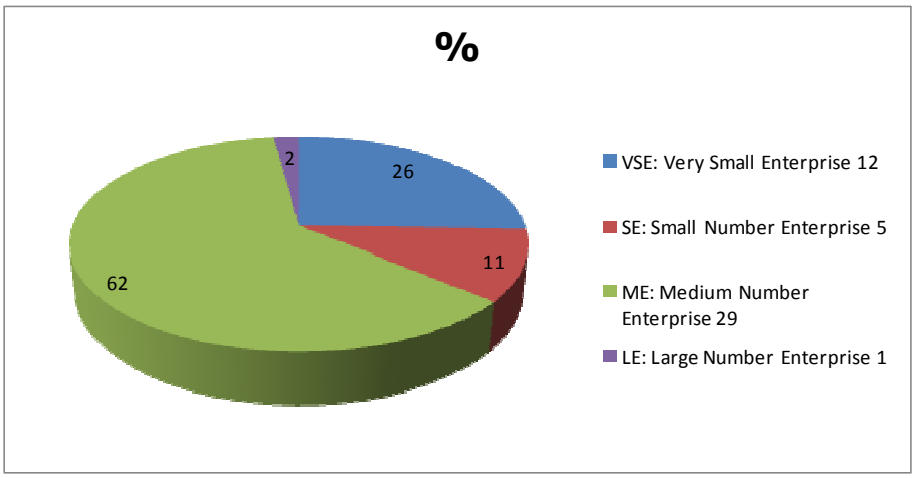


Figure B.2 The percentage of Full time SW development employees

3. "How many active SW development projects in your organization have been run since last year until now?"

A number of active software development projects are Medium Number, Small Number, Large Number and Very Small Number as 53%, 28%, 15% and 4% respectively from all 47 questionnaires which are defined as follow value; VS <=5, 5>S<=10, 10>M<=15, and L >15 projects.

The active software development projects show current active software development projects which still running since last year in an organization.

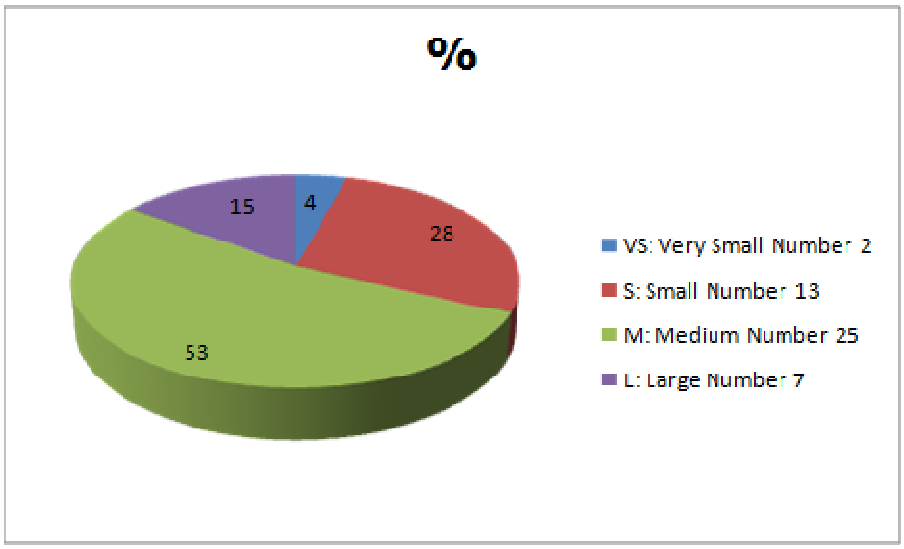


Figure B.3 The percentage of Active SW development projects

4. "How many SW development projects have been run based on CMMI?"

A number of CMMI software development projects are Small Number, Medium Number, Large Number and Very Small Number as 49%, 36%, 9% and 6% respectively from all 47 questionnaires which are defined as follow value; VS <=5, 5>S<=10, 10>M<=15, and L >15 CMMI software development projects.

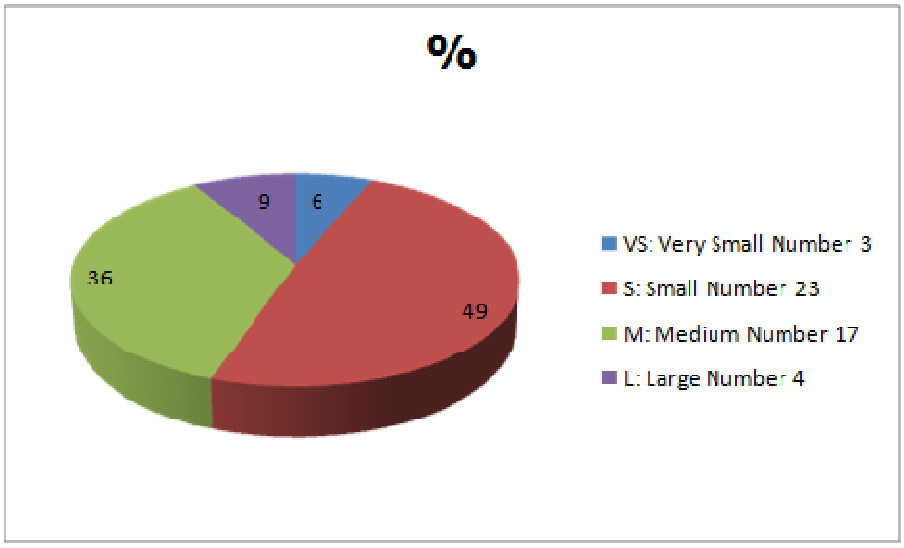


Figure B.4 The percentage of CMMI SW development projects

PART B: Project Characteristics are purposely to understand the characteristic of project in an organization. There is including 4 parts as 14 questions;

5. "What is the name of the reference SW development project?"

This question is reference to the name of the software development project's name which is referred information for answering the questionnaire.

6. "What is the average team experience for this SW development project in Project management/ Software process improvement/ Domain expertise/ and Application language expertise? (Please specify the number of years)"

The average team experience is present year of experience in term of software development project in specific domains. The Entry Level is defined when experiences is less than or equal to 2 years. The Intermediate Level is defined when experiences is during more than 2 and less than or equal to 5 years. The Master Level is defined when experiences is during more than 5 and less than or equal to 10 years. And finally, the Expert Level is defined when experiences is more than 10 years.

6.1 Project Manager

In term of Project Manager, a number of experiences are Master Level, Expert and Intermediate Level, and Entry Level as 55%, 19% and 6% respectively from all 47 questionnaires which mean the Master Level (5>Master Level<=10 years) is the majority group in experienced Project Manager.

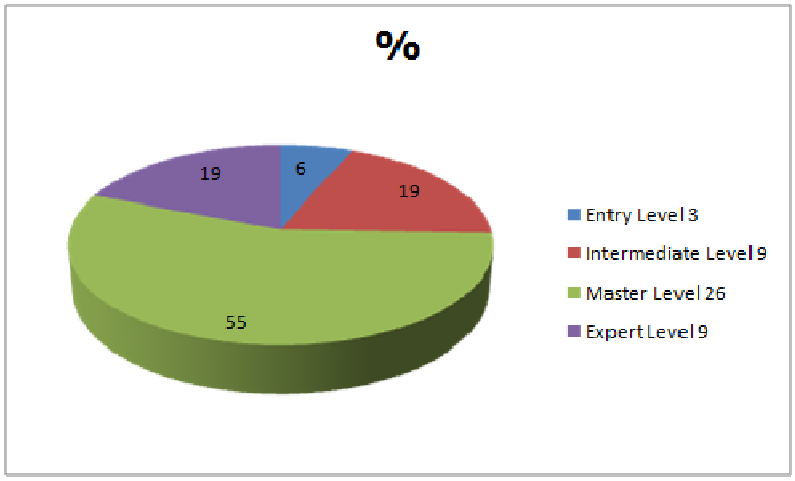


Figure B.5 The percentage of SW development experiences in Project Manager

6.2 Software Process Improvement

In term of Software Process Improvement, a number of experiences are Master Level and Intermediate Level, Expert Level and Entry Level as 36%, 15% and 13% respectively from all 47 questionnaires which mean the Intermediate Level and Master Level (2>Intermediate Level<=5 years and 5>Master Level<=10 years) are the majority group in experienced Software Process Improvement.

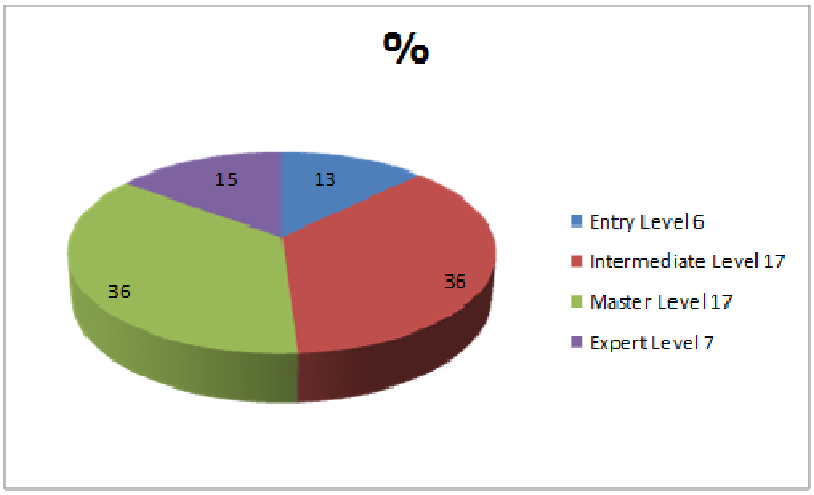


Figure B.6 The percentage of SW development experiences in Software Process Improvement

6.3 Application Domain Expertise

In term of Application Domain, a number of experiences are Intermediate Level, Master Level, Entry Level and Expert Level as 47%, 38%, 13% and 2% respectively from all 47 questionnaires which mean the Intermediate Level (2>Intermediate Level<=5 years) is the majority group in experienced Application Domain Expertise.

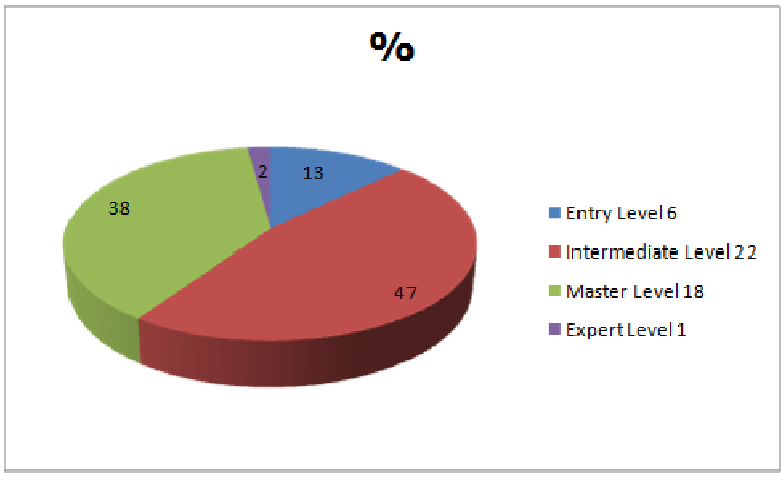


Figure B.7 The percentage of SW development experiences in Application Domain Expertise

6.4 Application Language Expertise

In term of Application Language Expertise, a number of experiences are Intermediate Level, Master Level, Entry Level and Expert Level as 60%, 21%, 11% and 9% respectively from all 47 questionnaires which mean the Intermediate Level (2>Intermediate Level<=5 years) is the majority group in experienced Application Language Expertise.

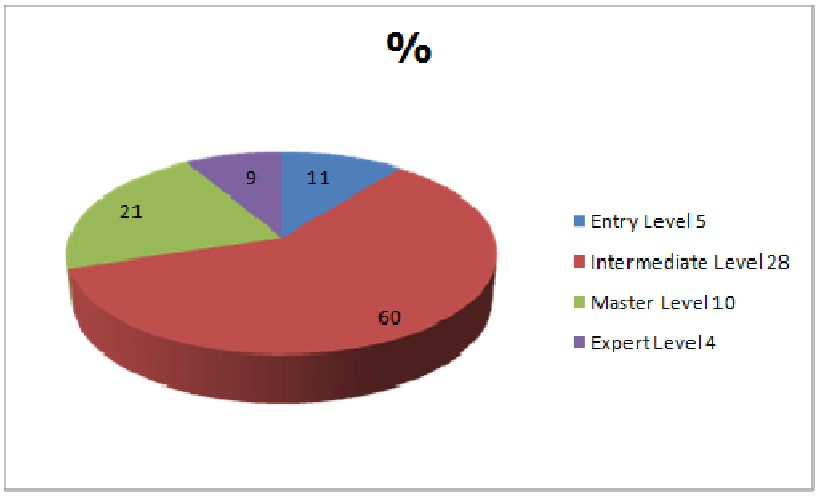


Figure B.8 The percentage of SW development experiences in Application Language Expertise

7. "Please specify your major role in this SW development project?"

The major role in software development project is present their specialist in specific roles as Software Engineering Process Group (SEPG), Project Management Officer (PMO), Project Manager (PM) and System Analyst (SA). In this regard, a number of Project Management is 67% which means it is a majority role. The other roles like SEPG, SA, other roles and PMO are 13%, 9%, 7% and 4% respectively from all 46 questionnaires.

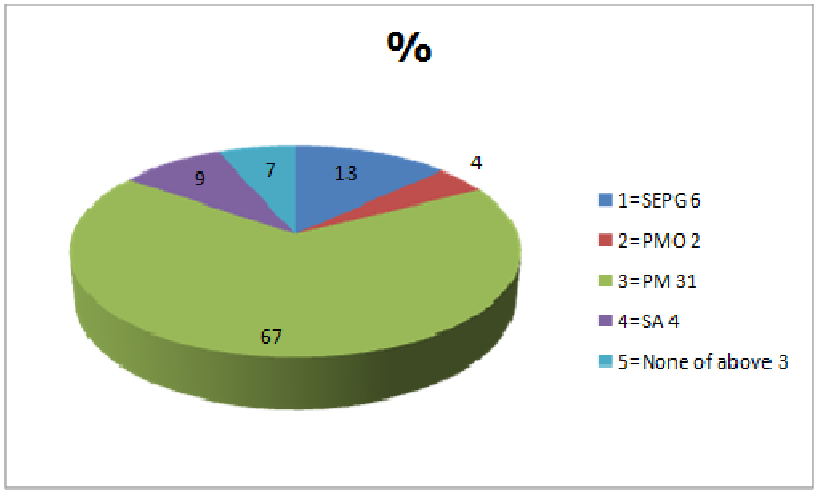


Figure B.9 The percentage of Major role in SW development team

B1. Quality characteristics

8. "Did your project apply any software process improvement and/or quality management program?"

A number of organizations are applying software process improvements and/or quality management programs are 96% from all 47 questionnaires.

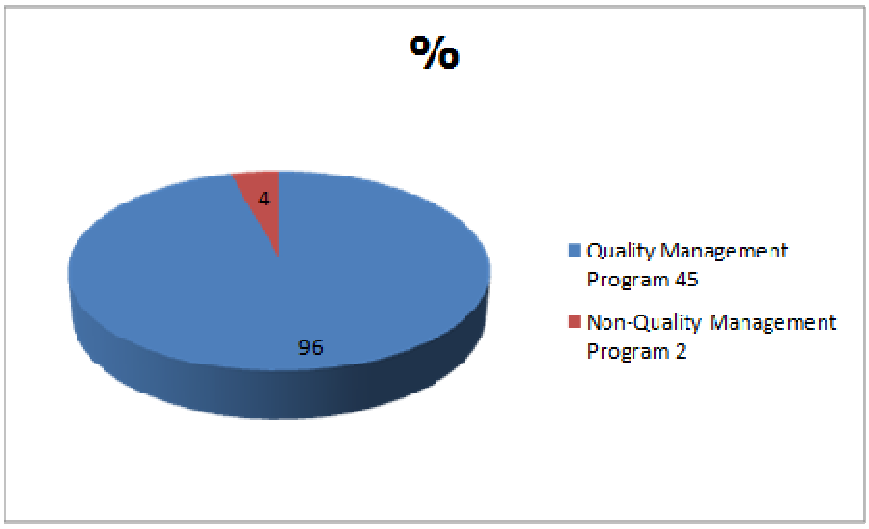


Figure B.10 The percentage of Software Process Improvement program

9. "Did your project apply CMMI (Capability Maturity Model Integration)?"

A number of organizations are applying CMMI (Capability Maturity Model Integration) is also 98% from all 47 questionnaires.

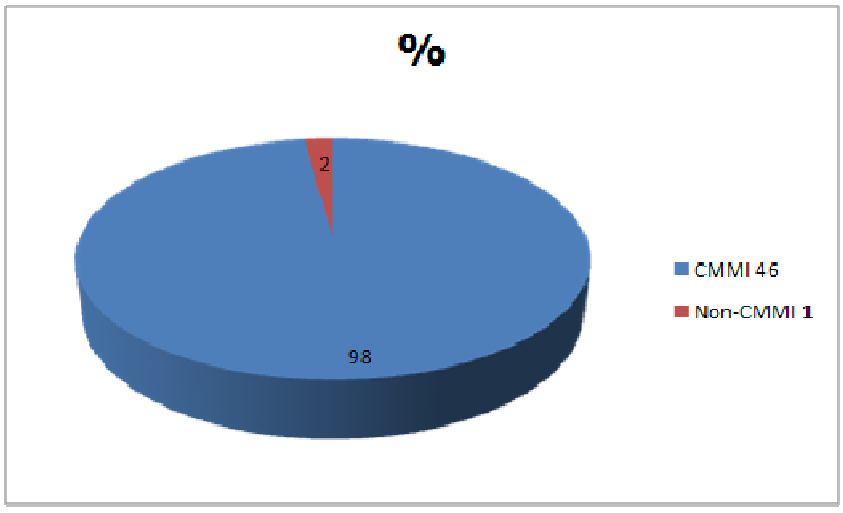


Figure B.11 The percentage of Capability Maturity Model Integration (CMMI) program

10. "What was your target to achieve for applying CMMI (Capability Maturity Model Integration)?"

10.1 ML/CL in CMMI

The majority of organization is ML (Maturity Level) in CMMI model which is 98% of implemented CMMI and 2% is not applying CMMI.

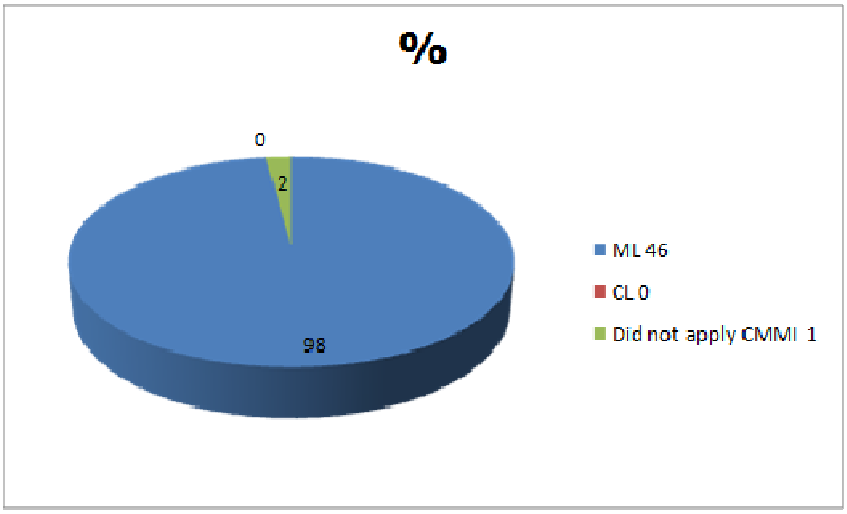


Figure B.12 The percentage of Maturity Level (2/3/4/5) in Capability Maturity Model Integration (CMMI)

10.2 Maturity Level (2/3/4/5) in CMMI

Regarding to ML (Maturity Level) which is applying in CMMI model, the majority is ML3 as 81% then, ML2, ML5 and ML4 are 13%, 4% and 2% respectively from all 47 questionnaires.

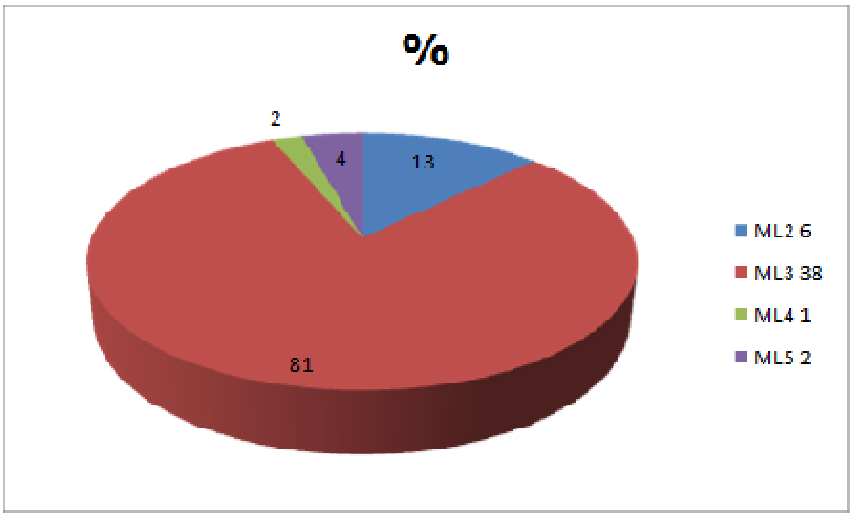


Figure B.13 The percentage of Maturity Level (2/3/4/5) in Capability Maturity Model Integration (CMMI)

11. “What was your project type?”

The project type of software development is 100% in development type from all 47 questionnaires.

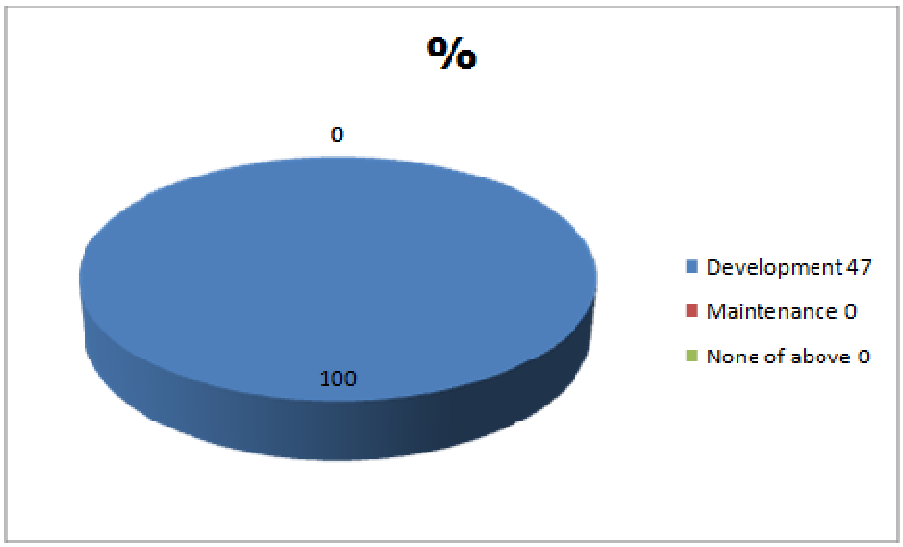


Figure B.14 The percentage of SW development project type

B2. Customer characteristics

12. “Please specify the kind of customer from your reference project?”

A number of customer types are non-government customer and government customer as 57% and 43% from all 47 questionnaires.

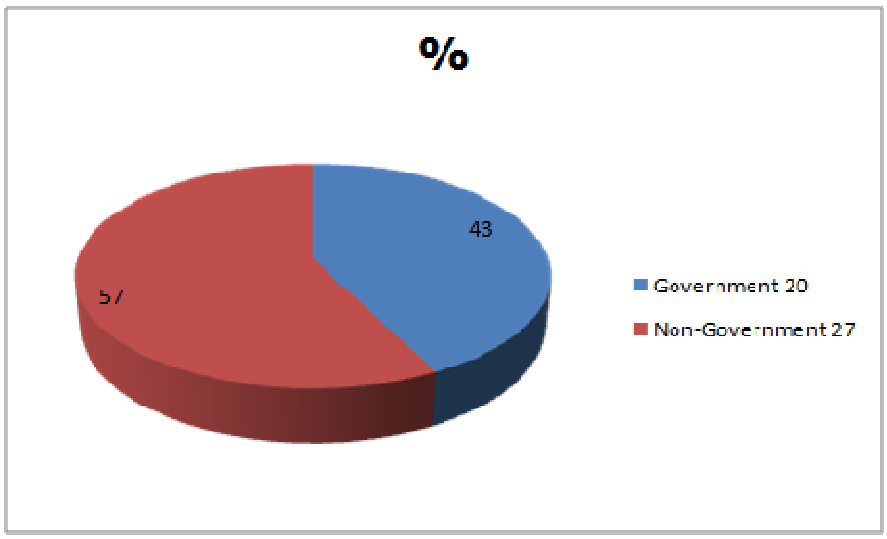


Figure B.15 The percentage of Customer type

13. "What was the domain of the customer's software application?"

The domain of the customer's software applications are classified in Services (Business, Construction, Entertainment, Finance, Medical, Retail, Telecommunication), Manufacturing (Product, Oils-Energy), the other domains and Research and development as 64%, 17%, 11% and 9% respectively, and there is none software application in Education and Training from all 46 questionnaires which mean the majority is software application in Services.

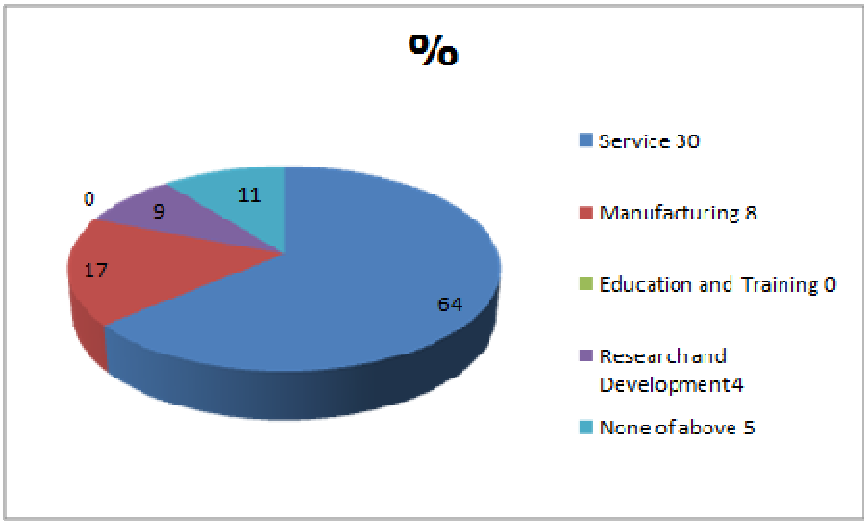


Figure B.16 The percentage of Customer's domain software application

In this regard, the detailed domain of the customer's software application, services likes Communication, Resource Management, Transportation are all the same as 11%, and Security, Enterprise management, Water-supply are 8% however, the majority is other domains likes Mobile application, Operational supporting, Network management, Website, Office automation network system, financial and customer service as 45% from all 47 questionnaires.

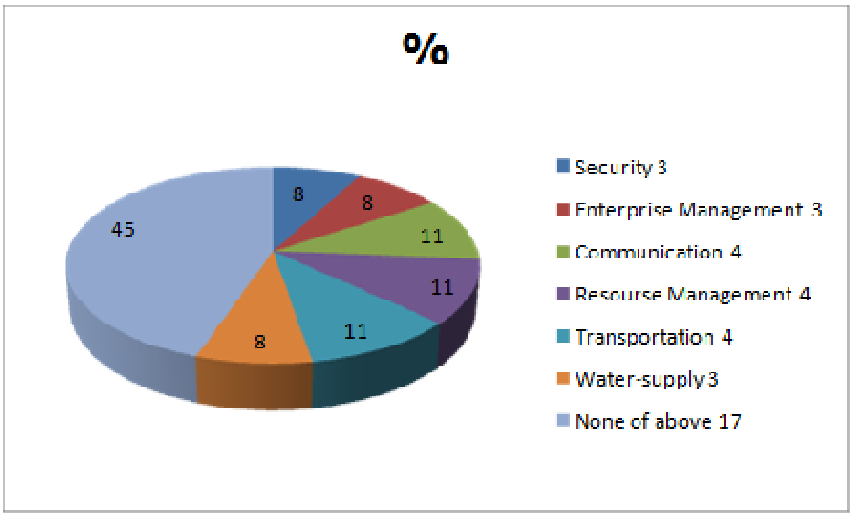


Figure B.17 The percentage of Detailed in Customer's domain software application

B3. Project size characteristics

14. "How many employees have been in the SW development project team? (Please give the number in Persons)"

A number of employees who have been in the software development projects are Medium Number, Large Number, Small Number and Very Small Number as 57%, 26%, 11% and 6% respectively from all 47 questionnaires which are classified as follow value; VS <=2, 2>S<=5, 5>M<=10, and L >10 persons.

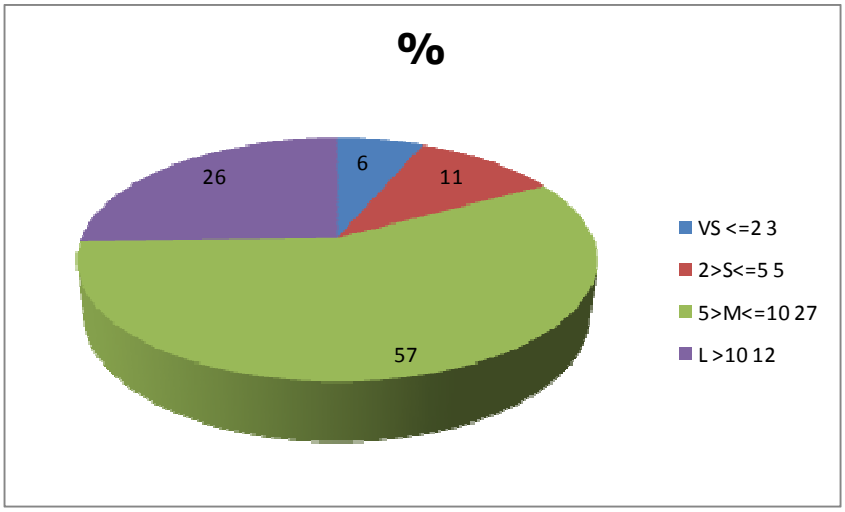


Figure B.18 The percentage of Full time SW development team

15. "What was the approximate project effort? (Please give the number in Person-months)"

A number of project efforts (Person-months) in the software development projects are Small Number, Very Small Number, Medium Number and Large Number are equality as 47%, 32% and 11% respectively from all 47 questionnaires which are defined as follow value; VS <=2, 2>S<=5, 5>M<=10, and L >10 person-months.

Hint: "1 full-time employee working in 1 month is approximately equal to 1 staff * 8 hours * 22 days so that 1 person-month = 22 person-days = 176 person-hours"

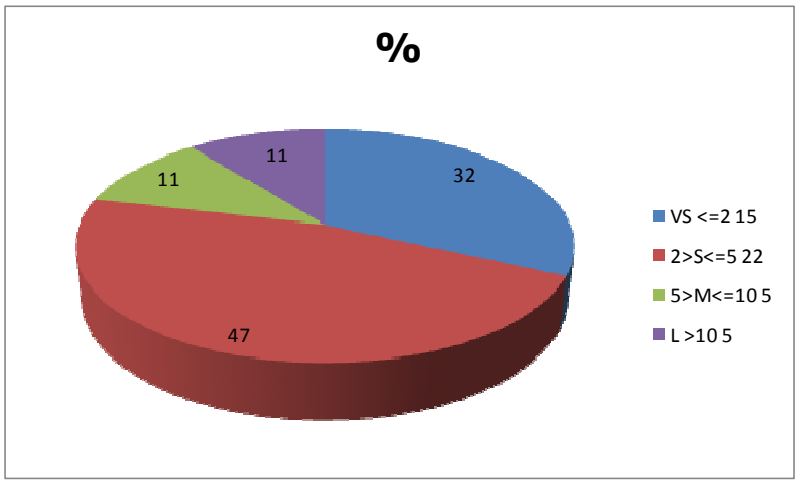


Figure B.19 The percentage of SW development project effort (actual)

16. "What was the approximate project duration? (Please give the number in month)"

A number of project durations (months) in the software development projects are Very Small Number, Small Number, Medium Number and Large Number as 79%, 13%, 6% and 2% respectively from all 47 questionnaires which are defined as follow value; 6 months (VS <=6), S is between 6 months to 12 months (6>SE<=12), M is between 12 to 24 (12>M<=24), and L is more than 24 months (L >24).

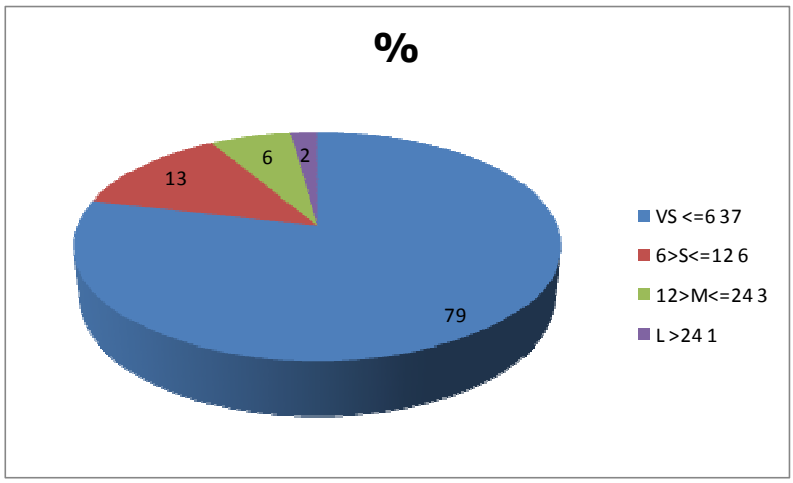


Figure B.20 The percentage of SW development project duration (actual)

17. "Please enter the estimated size of the software based on the applied estimation model. (Please select only one model, numbers with only two decimal places)"

17.1 Past experience estimations (Year)

A number of the past experience estimations (Year) in the software development projects are Expert Level, Master Level, Intermediate Level as 58%, 25% and 17% respectively and also there is none Entry Level from all 12 questionnaires which are defined as follow value; Entry Level <=2, 2>Intermediate Level<=5, 5>Master Level<=10, and Expert Level >10 years.

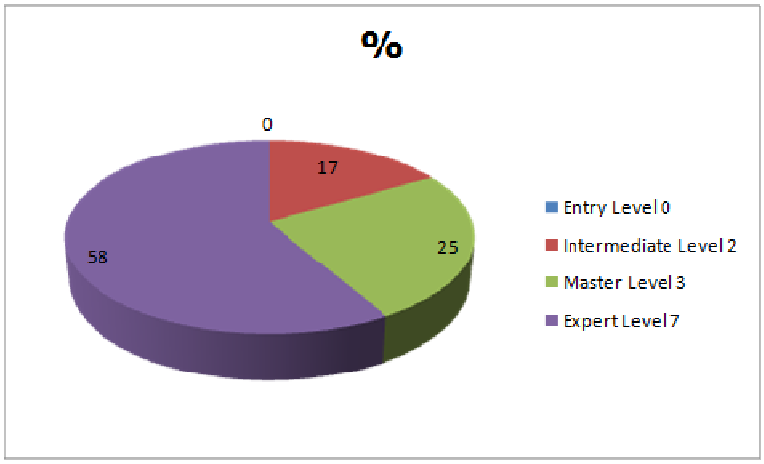
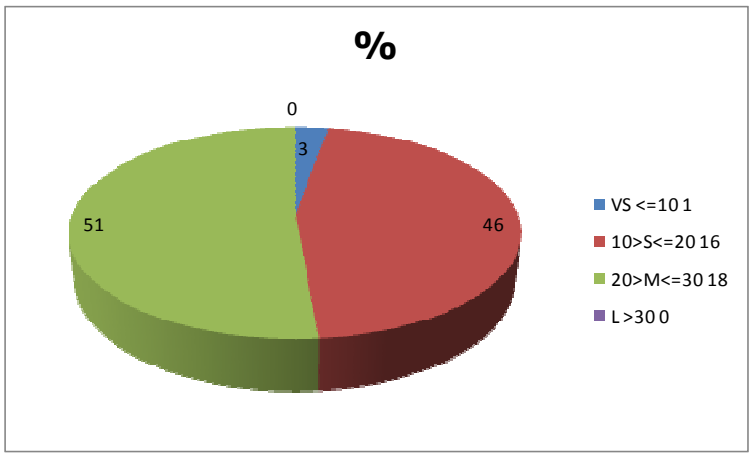


Figure B.21 The percentage of Past experience estimations

17.2 Source lines of codes (KLOCs)

A number of the estimated software sizes or software source lines of codes (KLOCs) in the software development projects are Medium Number, Small Number and Very Small Number as 51%, 46% and 3% respectively and also there is none Large Number from all 35 questionnaires which are defined as follow value; Very Small Number <=10, 10>Small Number<=20, 20>Medium Number<=30, and Large Number >30 KLOCs.



B.22 The percentage of Source Lines of Codes (KLOCs)

B4. Software Development Process (SDP) characteristics

18. "What was the major applied SDP in your SW development project? (Please select only one)"

The defined software development process (SDP) characteristic in the software development project in the plan-driven is including Waterfall, Prototype, Joint Application Development-JAD)/Rapid Application Development-RAD, Unified Process or Spiral). And Agile-driven is including Crystal Clear, Extreme Programming-XP, Scrum, Feature Driven Development-FDD, Dynamic Systems Development Method-DSDM, Adaptive Software Development-ASD) and finally there is none of above.

18.1 Software Development Process (SDP)

A number of Software Development Processes (SDP) in the software development projects are Plan-Driven and Agile-Driven as 85% and 7% respectively and also there is none of above defined answer as 9% from all 46 questionnaires.

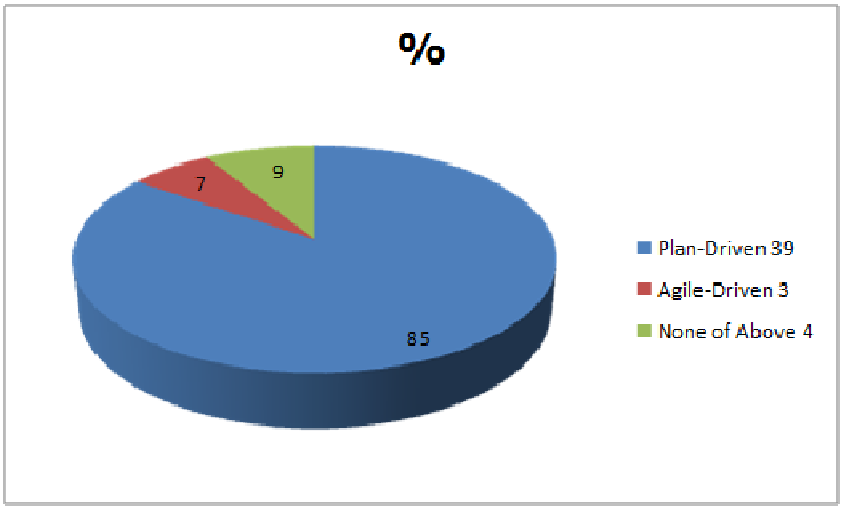


Figure B.23 The percentage of Software Development Process (SDP)

18.2 Plan-driven/Agile-driven

Regarding to 18.1, a number of Software Development Processes (SDP) in the software development projects based on Plan-driven/Agile-driven are "Waterfall and Waterfall & prototype", "Incremental and Incremental & prototype", "V-Shape and V-shape & Prototype" and Scrum as 48%, 28%, 14% and 9% respectively from all 35 questionnaires. However, there is 11 questionnaires that is not defined their SDP.

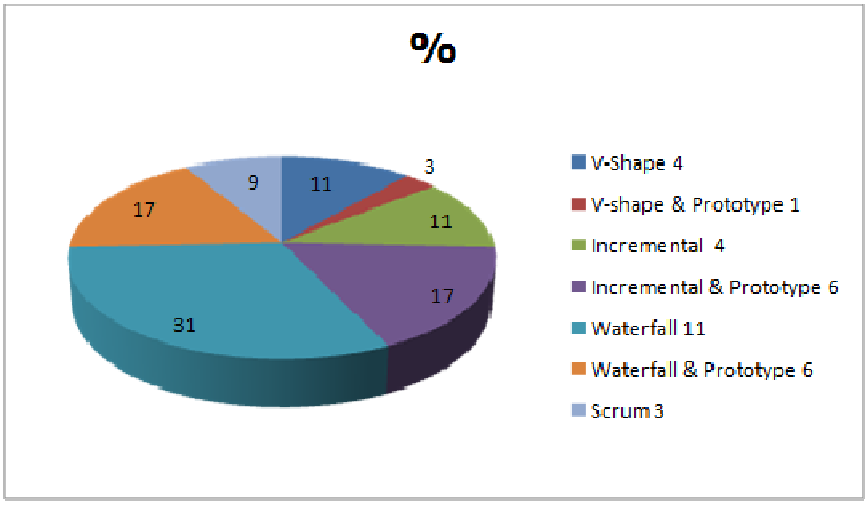


Figure B.24 The percentage of Plan-driven/Agile-driven method

PART C: Integrated Project Planning and Monitoring Control are purposely to understand the planned and the actual result of software development project which is implement "CMMIbyScrum". There is including two detailed part as 18 questions;

C1. The approximate values of the itemed project (as planned and actual values)

From the question number 19 to 25 present a value of project variable in both planned and actual at the end. The comparison between Plan-Driven and Agile-Driven are also presented. However, a number of Agile-Driven projects are significantly less than Plan-Driven; there are only three Agile-Driven projects in our investigation.

19. "What was the project effort? (In person-month)" [from question no. 15]

19.1 Project effort (person-month) as Planned

A number of the project efforts (person-month) as planned in the software development projects are Small Number, Medium Number, Large Number and Very Small Number as 53%, 28%, 12% and 7% respectively from all 43 questionnaires which are defined as follow value; Very Small Number <=2, 2>Small Number<=5, 5>Medium Number<=10, and Large Number >10 person-months.

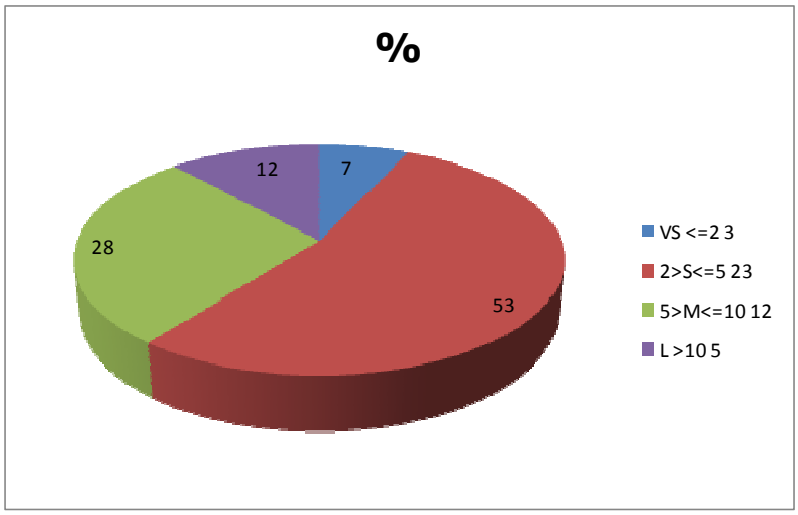


Figure B.25 The percentage of Project effort (person-month) as Planned

Remark: the calculation is based on the following rules: “1 full-time employee working in 1 month is approximately equal to 1 staff * 8 hours * 22 days so that 1 person-month = 22 person-days = 176 person-hours”.

19.2 Project effort (person-month) as Actual at the end

A number of the project efforts (person-month) as actual at the end in the software development projects are Medium Number, Small Number, Large Number as 51%, 33%, 16% respectively and there is none Very Small Number from all 43 questionnaires which are defined as follow value; Very Small Number <=2, 2>Small Number<=5, 5>Medium Number<=10, and Large Number >10 person-months.

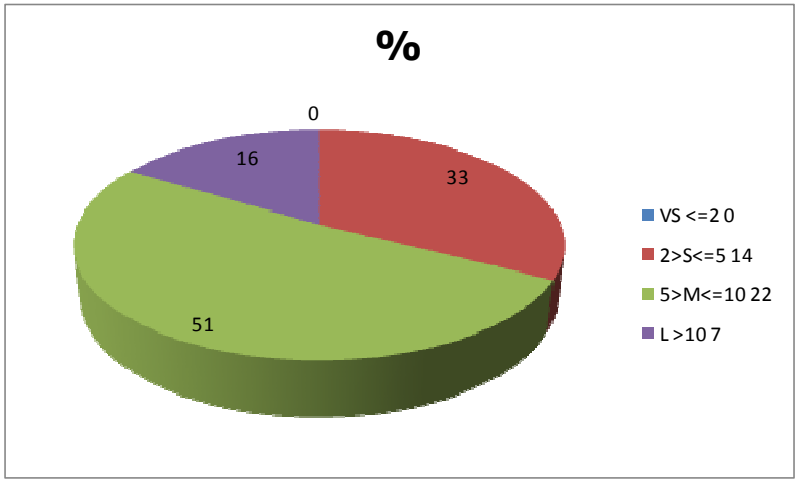


Figure B.26 The percentage of Project effort (person-month) as Actual at the end

Regarding to a difference between a number of the project efforts (person-month) as planned and actual at the end in the software development projects, a group of Medium Number and Large Number are getting larger from 28% to 51% and 12% to 16%. A group of Small Number and Very Small Number are smaller from 53% to 33% and 7% to 0%.

In term of differentiation, a number of the project effort (person-month) in actual at the end is higher value than expectation as a planned.

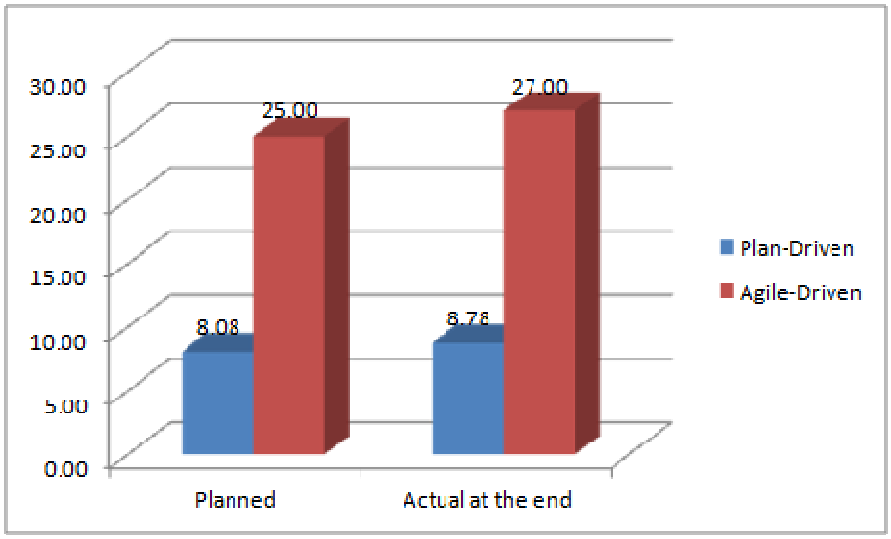


Figure B.27 The variation of Plan and Agile driven in Project effort (person-month) between Planned and Actual at the end

From above figure, both of Plan-Driven and Agile-Driven are show a number of the project effort (person-month) in actual at the end is higher value than expectation as a planned. However, the variation of Agile-Driven from the actual compare to the plan is lesser value at 7.41% and the variation of Plan-Driven from the actual compare to the plan is 7.95%. It is meaning that Agile-Driven is shown better performance to manage an effort than Plan-Driven.

20. “What was the project manager effort? (In person-month)”

20.1 Project Manager effort (person-month) as Planned

A number of the project manager efforts (person-month) as planned in the software development projects are Very Small Number, Small Number, Medium Number and Large Number as 80%, 14%, 5% and 2% respectively from all 44 questionnaires which are defined as follow value; Very Small Number <=2, 2>Small Number<=5, 5>Medium Number<=10, and Large Number >10 person-months.

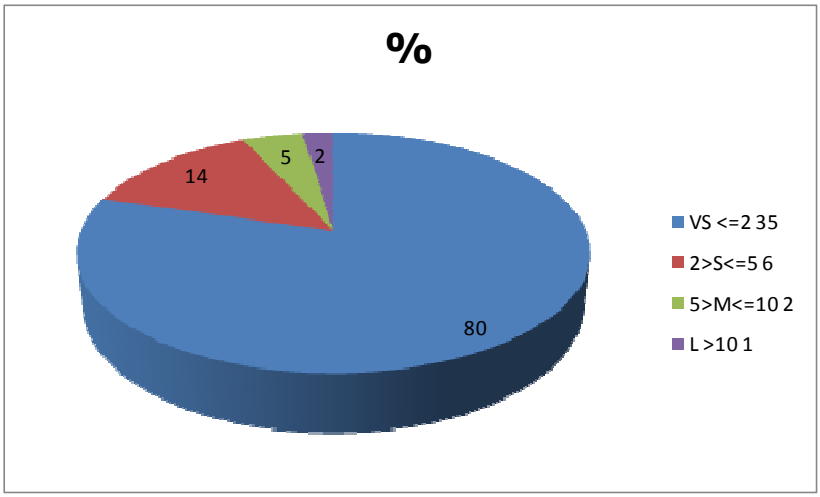


Figure B.28 The percentage of Project Manager effort (person-month) as Planned

20.2 Project Manager effort (person-month) as Actual at the end

A number of the project manager efforts (person-month) as actual at the end in the software development projects are as same as planned in term of order as Very Small Number, Small Number, Medium Number and Large Number, however, the percentages are change as 70%, 20%, 7% and 2% respectively from all 44 questionnaires which are defined as follow value; Very Small Number <=2, 2>Small Number<=5, 5>Medium Number<=10, and Large Number >10 person-months.

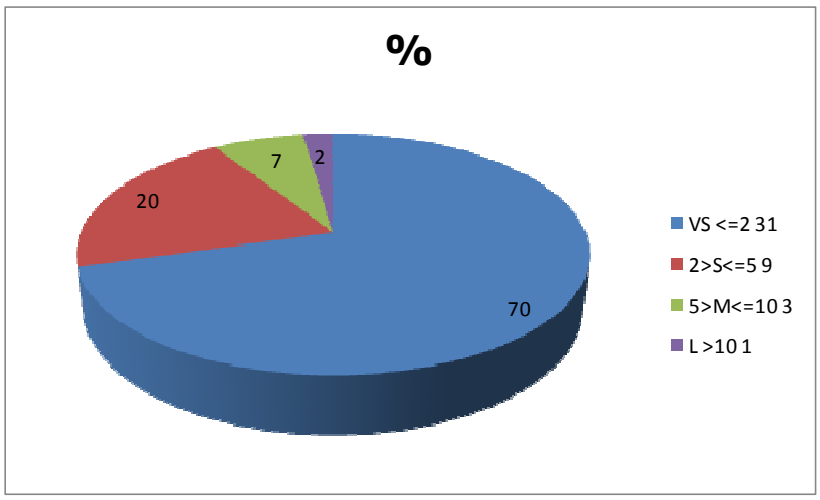


Figure B.29 The percentage of Project Manager effort (person-month) as Actual at the end

Regarding to a difference between a number of the project manager effort (person-month) as planned and actual at the end in the software development projects, a group of Small Number and Medium Number are getting larger from 14% to 20% and 5% to 7%. Group of Very Small Number are smaller from 80% to 70% and there is no change at Large Number at 2%.

In term of differentiation, a number of the project manager efforts (person-month) in actual at the end are higher value than expectation as a planned.

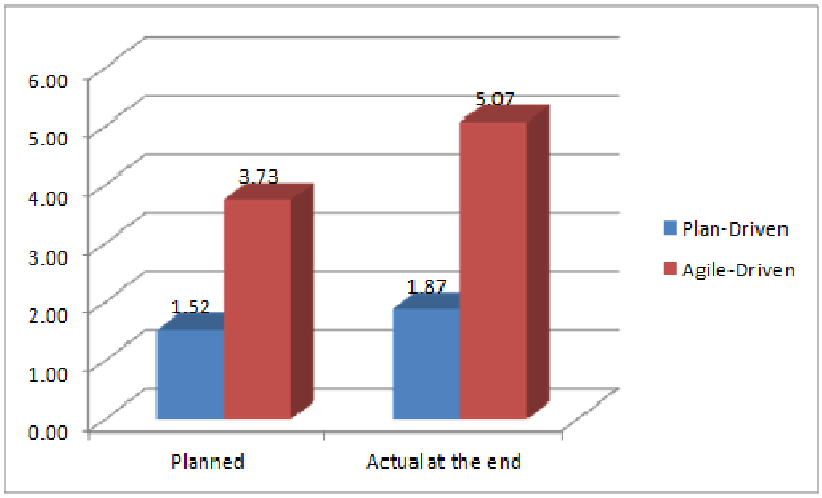


Figure B.30 The variation of Plan and Agile driven in Project Manager effort (person-month) between Planned and Actual at the end

From above figure, both of Plan-Driven and Agile-Driven are show a number of the project manager efforts (person-month) in actual at the end is higher value than expectation as a planned. However, the variation of Plan-Driven from the actual compare to the plan is lesser value at 18.90% and the variation of Agile-Driven from the actual compare to the plan is 26.32%. It is meaning that Plan-Driven is shown better performance to manage an effort than Agile-Driven.

21. "What was the approximate total SW development cost? (in $ USD)"

21.1 Software Budget ($ USD) as Planned

A number of the software development cost (USD) as planned in the software development projects are Medium Number, Small Number, Small Number, Very Small Number and Large Number as 46%, 36%, 13% and 5% respectively from all 39 questionnaires which are defined as follow value; Very Small Number <=20,000, 20,000>Small Number<=40,000, 40,000>Medium Number<=60,000, and Large Number >60,000 USD.

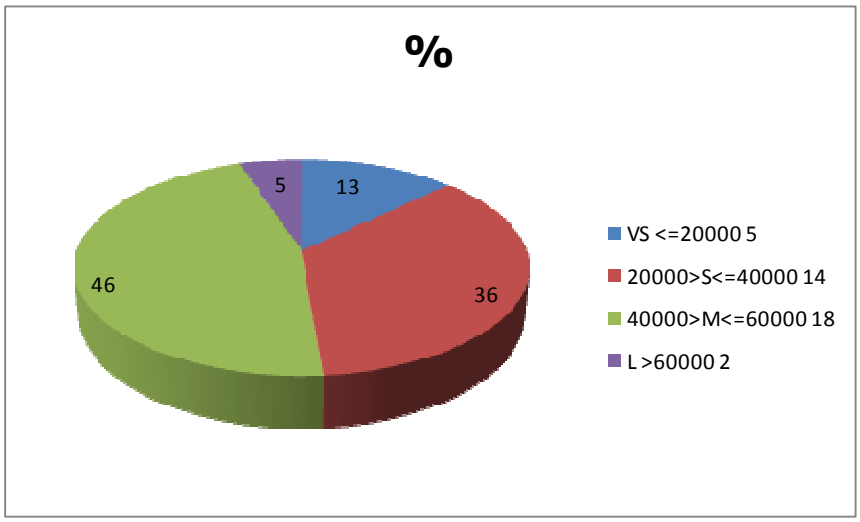


Figure B.31 The percentage of Software Budget ($ USD) as Planned

21.2 Software Budget ($ USD) as Actual at the end

A number of the software development cost (USD) as actual at the end in the software development projects are Medium Number, Small Number, Large Number and Very Small Number as 49%, 21%, 18% and 13% respectively from all 39 questionnaires which are defined as follow value; Very Small Number <=20,000, 20,000>Small Number<=40,000, 40,000>Medium Number<=60,000, and Large Number >60,000 USD.

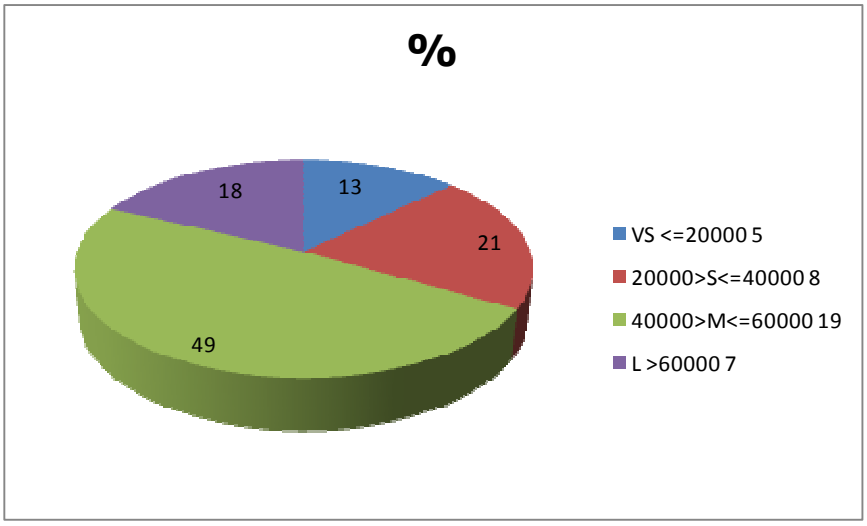


Figure B.32 The percentage of Software Budget ($ USD) as Actual at the end

Regarding to a difference between a number of the software development cost (USD) as planned and actual at the end in the software development projects, a group of Medium Number and Large Number are getting larger from 46% to 49% and 5% to 18%. A group of Small Number is smaller from 36% to 21% and there is no change at Very Small Number at 13%.

In term of differentiation, a number of the software development cost (USD) in actual at the end are a higher value than expectation as a planned.

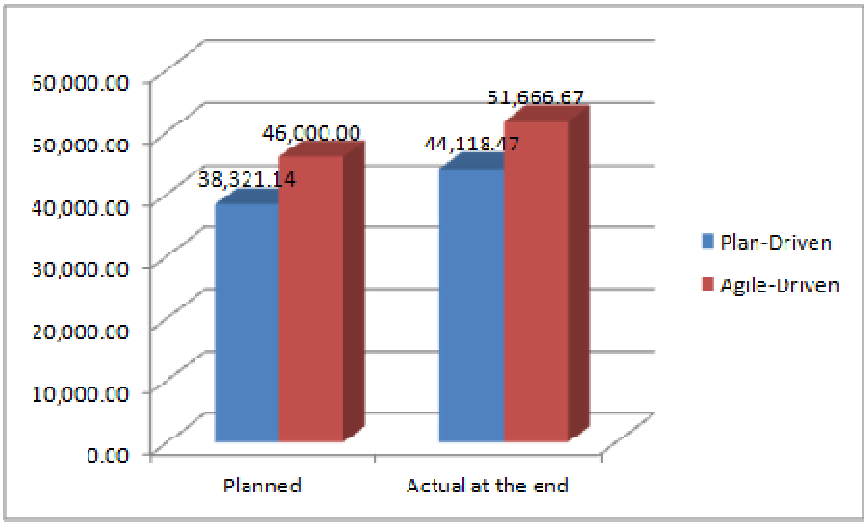


Figure B.33 The variation of Plan and Agile driven in Software Budget ($ USD) between Planned and Actual at the end

From above figure, both of Plan-Driven and Agile-Driven are show a number of software development cost (USD) in actual at the end are higher value than expectation as a planned. However, the variation of Agile-Driven from the actual compare to the plan is lesser value at 10.97% and the variation of Plan-Driven from the actual compare to the plan is 13.14%. It is meaning that Agile-Driven is shown better performance to manage cost than Plan-Driven.

22. "What was the approximate total SW quality cost? (in $ USD)"

The question would like to know a number of quality cost which project mainly paid for preventing cost as planned and actual values in US Dollar unit.

22.1 Software Process Improvement Budget ($ USD) as Planned

A number of the software quality cost (USD) as planned in the software development projects are Small Number, Medium Number, Large Number and Very small Number as 38%, 33%, 17% and 13% respectively from all 24 questionnaires which are defined as follow value; Very Small Number <=1,000, 1,000>Small Number<=2,000, 2,000>Medium Number<=3,000, and Large Number >3,000 USD.

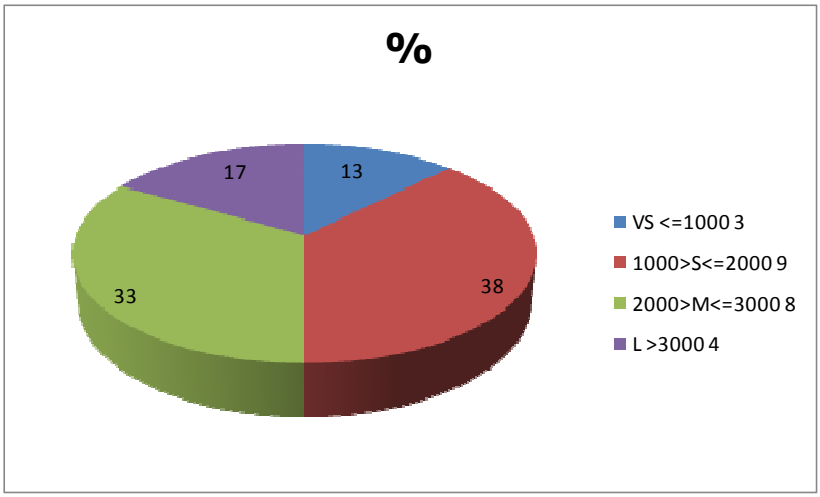


Figure B.34 The percentage of Software Process Improvement Budget ($ USD) as Planned

22.2 Software Process Improvement Budget ($ USD) as Actual at the end

A number of the software quality cost (USD) as actual at the end in the software development projects are Small Number, Medium Number and Vey Small Number/Large Number as 58%, 17% and 13% respectively from all 39 questionnaires which are defined as follow value; Very Small Number <=1,000, 1,000>Small Number<=2,000, 2,000>Medium Number<=3,000, and Large Number >3,000 USD.

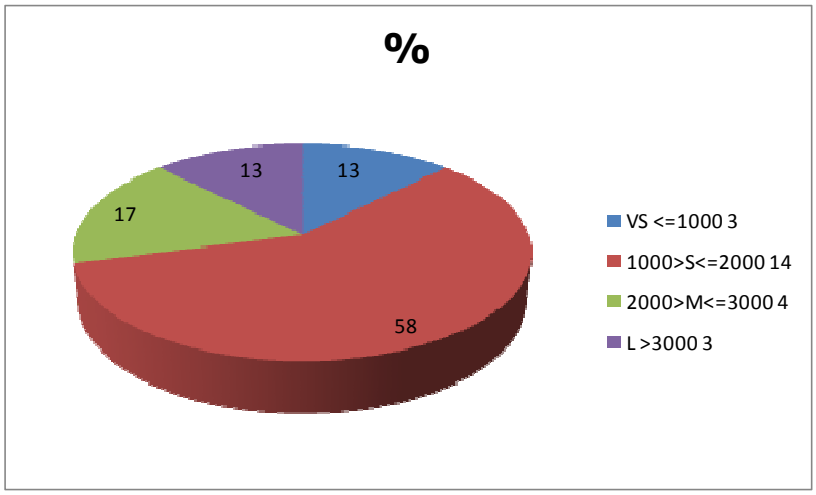


Figure B.35 The percentage of Software Process Improvement Budget ($ USD) as Actual at the end

Regarding to a difference between a number of the software quality cost (USD) as planned and actual at the end in the software development projects, a group of Small Number is getting larger from 38% to 58%. A group of Medium Number and Large Number are smaller from 33% to 17% and 17% to 13%. However, there is no change at Very Small Number at 13%.

In term of differentiation, a number of the software quality cost (USD) in actual at the end is a little lower value than expectation as a planned.

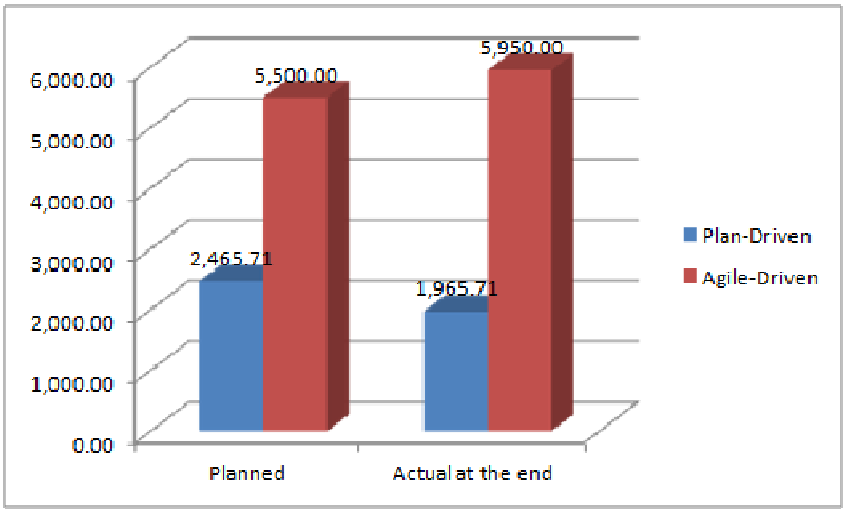


Figure B.36 The variation of Plan and Agile driven in Software Process Improvement Budget ($ USD) between Planned and Actual at the end

From above figure, Plan-Driven is shows a number of software quality cost (USD) in actual at the end is lower value than expectation as a planned. On the other hand, Agile-Driven shows higher value than expectation as a planned. The variation of Plan-Driven from the actual compare to the plan is lesser value at -25.44% and the variation of Agile-Driven from the actual compare to the plan is 7.56%. It is meaning that Plan-Driven is shown better performance to manage cost than Agile-Driven.

23. "What was the project duration? (In month(s)) [from question no. 16]"

23.1 Project duration (month(s)) as Planned

A number of the project durations (month) as planned in the software development projects are Very small Number, Small Number and Medium Number as 74%, 19% and 6% respectively and there is none Large Number from all 31 questionnaires which are defined as follow value; Very Small Number <=6, 6>Small Number<=12, 12>Medium Number<=24, and Large Number >24 months.

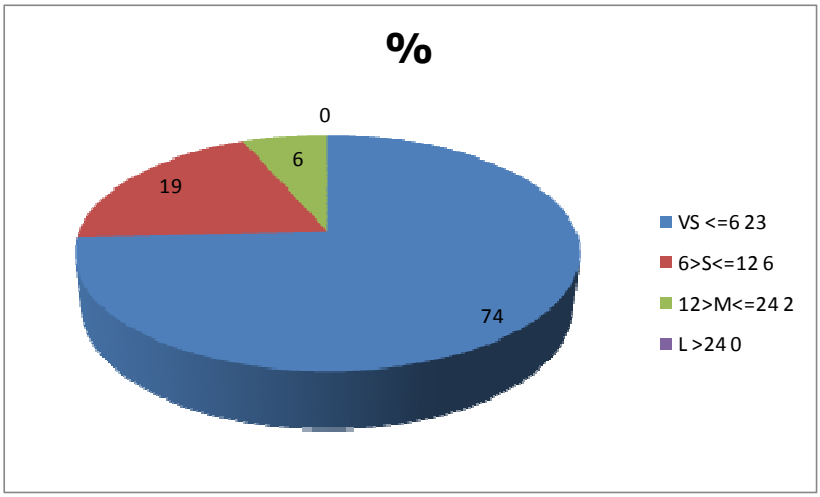


Figure B.37 The percentage of Project duration (month(s)) as Planned

23.2 Project duration (month(s)) as Actual at the end

A number of the project durations (month) as actual at the end in the software development projects are Very small Number, Small Number, Large Number and Medium Number as 74%, 16%, 6% and 13% respectively from all 31 questionnaires which are defined as follow value; Very Small Number <=6, 6>Small Number<=12, 12>Medium Number<=24, and Large Number >24 months.

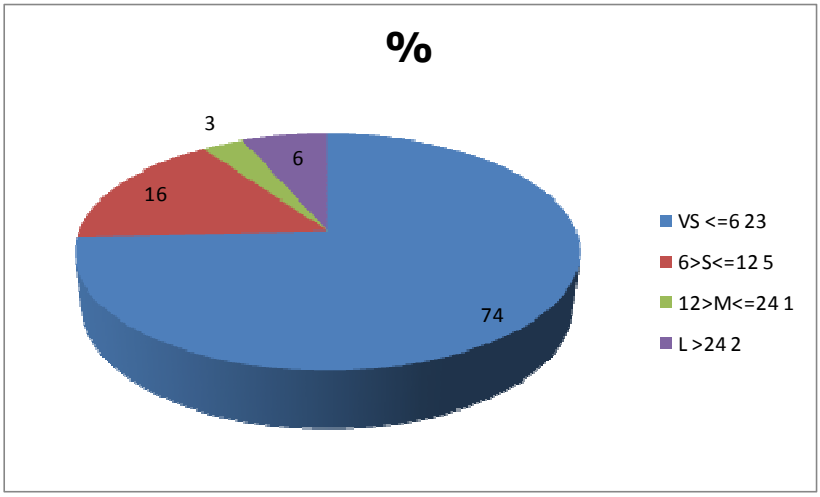


Figure B.38 The percentage of Project duration (month(s)) as Actual at the end

Regarding to a difference between a number of the project duration (month) as planned and actual at the end in the software development projects, a group of Large Number is getting larger from 0% to 6%. A group of Small Number and Medium Number are smaller from 19% to 16% and 6% to 3%. However, there is no change at Very Small Number at 74%.

In term of differentiation, a number of the project duration (month) in actual at the end is a little higher value than expectation as a planned.

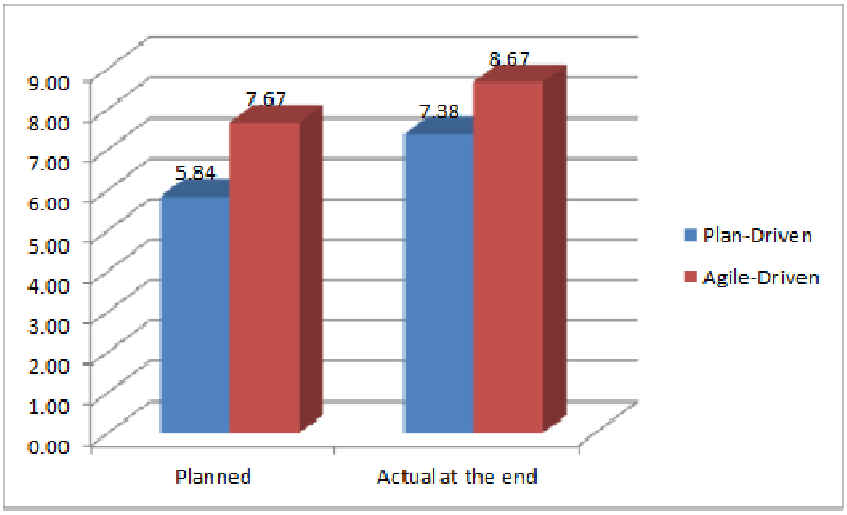


Figure B.39 The variation of Plan and Agile driven in Project duration (month(s)) between Planned and Actual at the end

From above figure, Agile-Driven is shows a number of project duration (month) in actual at the end is lower value than expectation as a planned. On the other hand, Plan-Driven shows higher value than expectation as a planned. The variation of Agile-Driven from the actual compare to the plan is lesser value at 11.54% and the variation of Plan-Driven from the actual compare to the plan is 20.80%. It is meaning that Agile-Driven is shown better performance to manage a schedule than Plan-Driven.

24. "What was the average number of defects/reworks for software product in the User Acceptance Test (UAT) phase?"

The defects/reworks per unit (DPU) for software product in the User Acceptance Test (UAT) phase which user mainly detective defects for reworks as planned and actual values in DPU unit.

The defects/reworks per unit (DPU) in the User Acceptance Test (UAT) phase of Plan-Driven and Agile-Driven as planned in the software development projects are 10.00 and 6.67 in DPU unit. The defects/reworks per unit (DPU) in the User Acceptance Test (UAT) phase of Plan-Driven and Agile-Driven as actual at the end in the software development projects are 14.67 and 13.33 in DPU unit.

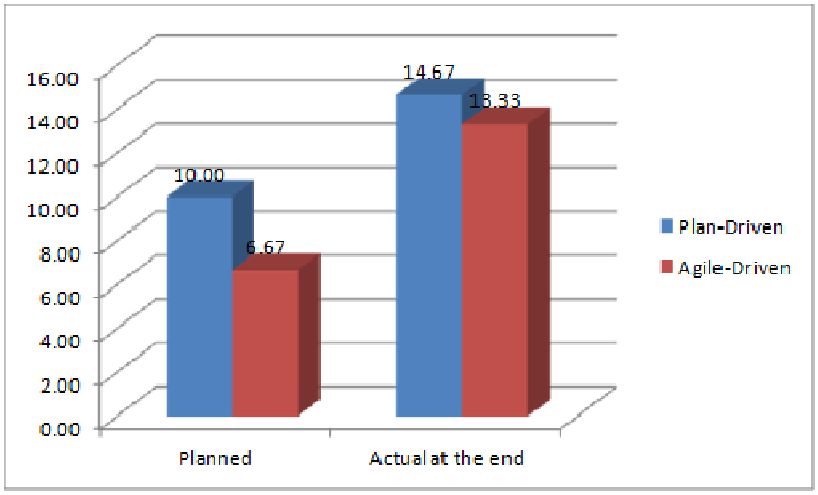


Figure B.40 The variation of Plan and Agile driven in SW development defects/reworks between Planned and Actual at the end

From above figure, Agile-Driven is shows a number of the defects/reworks per unit (DPU) in actual at the end is higher value than expectation as a planned. The variation of Agile-Driven from the actual compare to the plan is lesser value at 50.00% and the variation of Plan-Driven from the actual compare to the plan is 31.82%. It is meaning that Agile-Driven is shown better performance to manage defects/reworks than Plan-Driven.

25. "What was the average percentage of defects/reworks removal?"

The removal defects/reworks per unit (DPU) for software product in the User Acceptance Test (UAT) phase which project mainly collective defects as planned and actual values in DPU unit.

The removal defects/reworks per unit (DPU) in the User Acceptance Test (UAT) phase of both Plan-Driven and Agile-Driven as planned in the software development projects are 100.00% from DPU unit. The removal defects/reworks per unit (DPU) in the User Acceptance Test (UAT) phase of Plan-Driven and Agile-Driven as actual at the end in the software development projects are 53.33% and 85.00% from DPU unit.

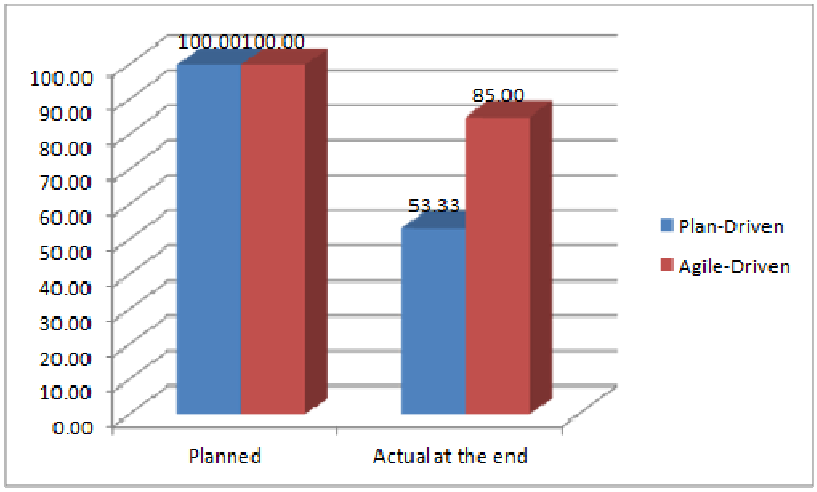


Figure B.41 The variation of Plan and Agile driven in SW development defects/reworks removal between Planned and Actual at the end

From above figure, Agile-Driven is shows a number of the defects/reworks removal per unit (DPU) in actual at the end is higher value than expectation as a planned. The variation of Agile-Driven from the actual compare to the plan is higher value at -17.65% and the variation of Plan-Driven from the actual compare to the plan is -87.50%. It is meaning that Agile-Driven is shown better performance to manage the removal defects/ reworks than Plan-Driven.

C2. The ordinary values of the itemed project (as planned and actual values)

It is defined as following values; 5-Definitely, 4-Probably, 3-Be unsure, 2-Probably not, 1-Definitely not and 0-Not applicable.

26. "Did you understand well to apply the SDP in plan-driven or agile-driven?"

Regarding to research questions (RQ), the level of understanding on how to apply the software development process (SDP) in plan-driven or Agile-driven are Probably (4), Definitely (5), and Be unsure (3) as 54%, 30% and 15% respectively and there is none Probably not (2), Definitely not (1) and Not applicable (0) from all 46 questionnaires which are defined as follow value; 5 is Definitely, 4 is Probably, 3 is Be unsure, 2 is Probably not, 1 is Definitely not and 0 is Not applicable. It is meaning that "Probably (4)" is the majority group at 54%.

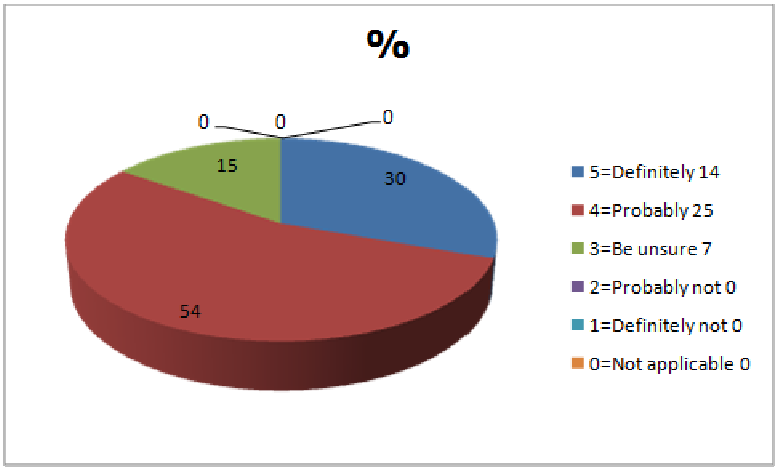


Figure B.42 The level of defined value in applying the SDP in plan-driven or agile-driven

27. "Did your project receive adequate resources both quality & quantity?"

Regarding to research questions (RQ), the level of resources both in quality and quantity on the readiness to apply the software development process (SDP) in plan-driven or Agile-driven are Probably (4), Definitely (5), Be unsure (3) and Probably not (2) as 35%, 28%, 26% and 11% respectively and there is none Definitely not (1) and Not applicable (0) from all 46 questionnaires which are defined as follow value; 5 is Definitely, 4 is Probably, 3 is Be unsure, 2 is Probably not, 1 is Definitely not and 0 is Not applicable. It is meaning that "Probably (4)" is the majority group at 35%.

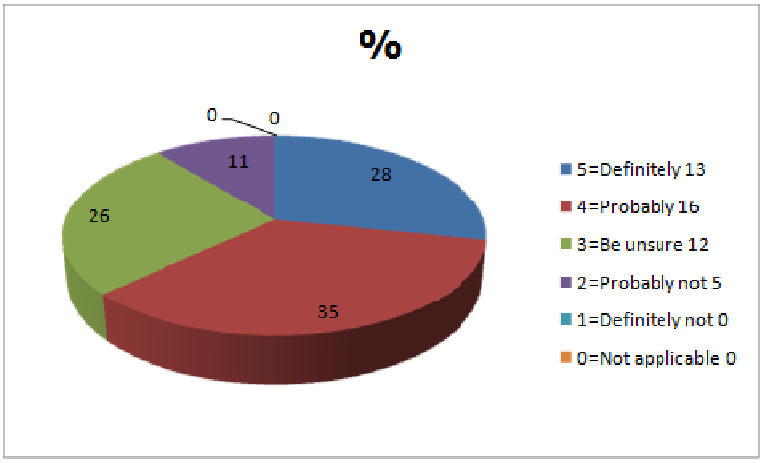


Figure B.43 The level of defined value in receiving adequate resources both quality & quantity

28. "Did your project deliver the product on-time (+/- 10%)?"

Regarding to RQ1 which is related to the mutual exclusives approaches of Agile and CMMI for process improvement program, the degree of on time deliver product when implementing the software development process (SDP) in plan-driven or Agile-driven are Probably (4), Be unsure (3), Definitely (5), Definitely not (1) and Probably not (2) as 39%, 26%, 24%, 7 and 4% respectively and there is none Not applicable (0) from all 46 questionnaires which are defined as follow value; 5 is Definitely, 4 is Probably, 3 is Be unsure, 2 is Probably not, 1 is Definitely not and 0 is Not applicable. It is meaning that "Probably (4)" is the majority group at 39%.

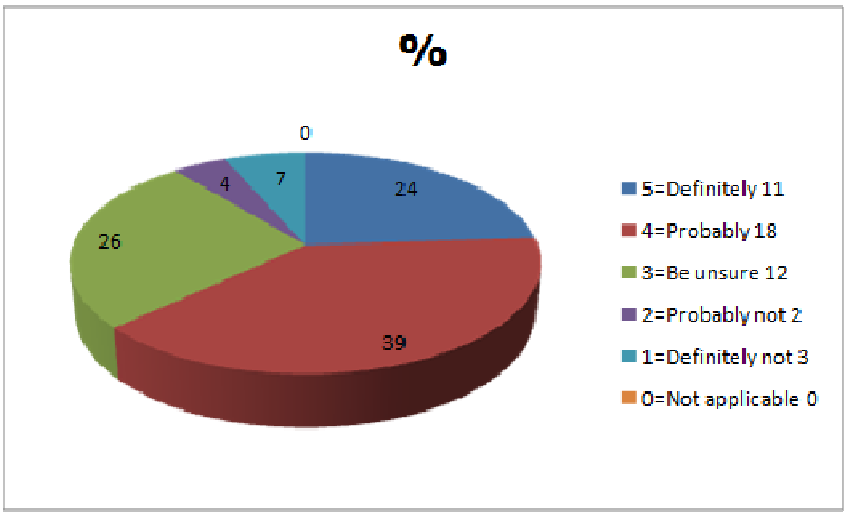


Figure B.44 The level of defined value in delivering the product on-time

29. "Did your project deliver the product in budget (+/- 5%)?"

Regarding to RQ1 which is related to the mutual exclusives approaches of Agile and CMMI for process improvement program, the degree of exceeded budget in product when implementing the software development process (SDP) in plan-driven or Agile-driven are Probably (4), Definitely (5), Be unsure (3), Probably not (2) and Definitely not (1) as 48%, 28%, 20%, 2% and 2% respectively and there is none Not applicable (0) from all 46 questionnaires which are defined as follow value; 5 is Definitely, 4 is Probably, 3 is Be unsure, 2 is Probably not, 1 is Definitely not and 0 is Not applicable. It is meaning that "Probably (4)" is the majority group at 48%.

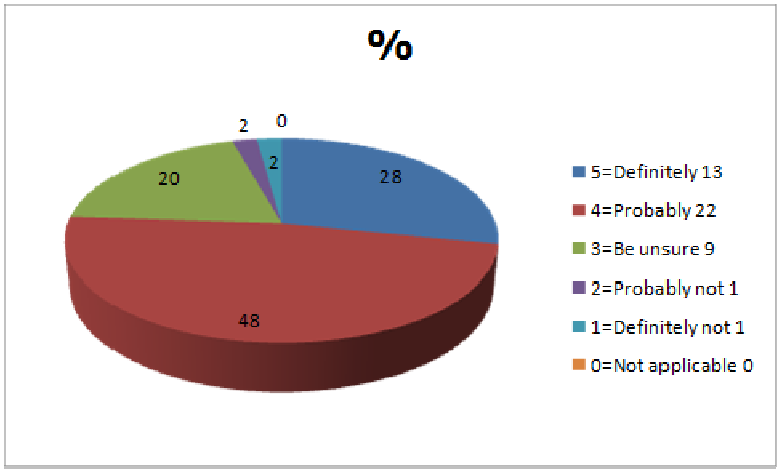


Figure B.45 The level of defined value in delivering the product in budget

30. "Did your project establish and maintain plans that defined the project activities?"

Regarding to RQ2 which is related to the gap between Agile-Scrum and the project management of CMMI, the degree of exceeded budget in product when implementing the software development process (SDP) in plan-driven or Agile-driven are Definitely (5), Probably (4), Be unsure (3), Probably not (2) and Not applicable (0) as 41%, 37%, 17%, 2% and 2% respectively and there is none Definitely not (1) from all 46 questionnaires which are defined as follow value; 5 is Definitely, 4 is Probably, 3 is Be unsure, 2 is Probably not, 1 is Definitely not and 0 is Not applicable. It is meaning that "Definitely (5)" is the majority group at 41%.

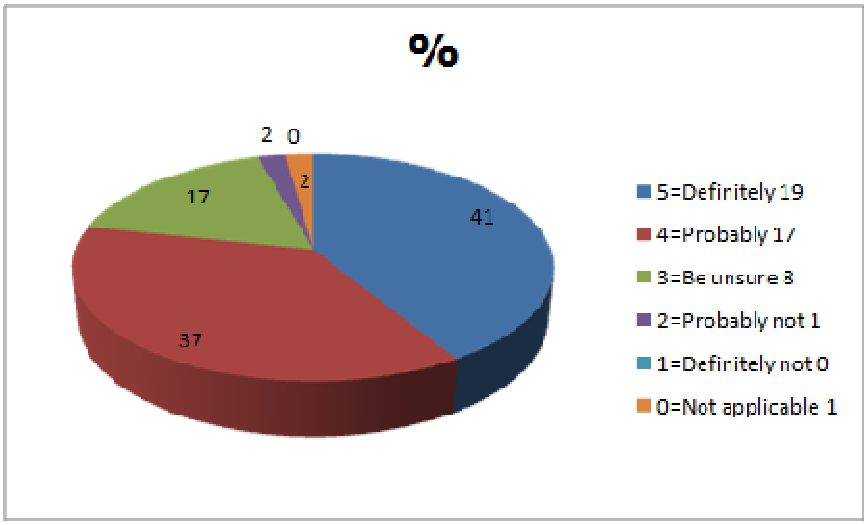


Figure B.46 The level of defined value in delivering the product in establishing and maintaining plans that defined the project activities

31. "Has your project been monitored periodically to follow the SDP and plans?"

Regarding to RQ2 which is related to the gap between Agile-Scrum and the project management of CMMI, the degree of periodic monitored plans regarding to defined project activities when implementing the software development process (SDP) in plan-driven or Agile-driven are Probably (4), Definitely (5), Be unsure (3), Probably not (2) and Definitely not (1) as 50%, 26%, 17%, 4% and 2% respectively and there is none Not applicable (0) from all 46 questionnaires which are defined as follow value; 5 is Definitely, 4 is Probably, 3 is Be unsure, 2 is Probably not, 1 is Definitely not and 0 is Not applicable. It is meaning that "Probably (4)" is the majority group at 50%.

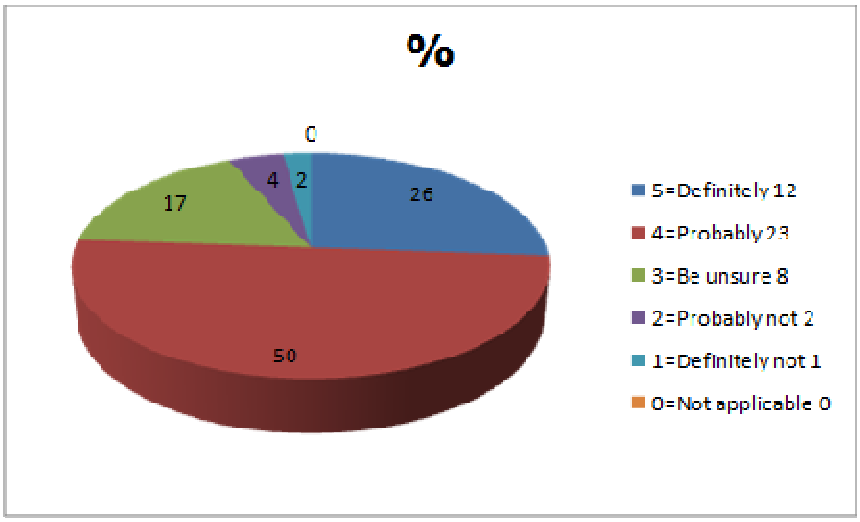


Figure B.47 The level of defined value in periodical monitoring to follow the SDP and plans

32. "Has your project been achieved the corrective actions after monitored?"

Regarding to RQ3 which is related to the combination between Agile-Scrum and the project management of CMMI to close the gap, the degree of the achievement of corrective actions regarding to monitored project activities when implementing the software development process (SDP) in plan-driven or Agile-driven are Probably (4), Definitely (5), and Be unsure (3) as 57%, 28% and 15% respectively and there is none Probably not (2), Definitely not (1) and Not applicable (0) from all 46 questionnaires which are defined as follow value; 5 is Definitely, 4 is Probably, 3 is Be unsure, 2 is Probably not, 1 is Definitely not and 0 is Not applicable. It is meaning that "Probably (4)" is the majority group at 57%.

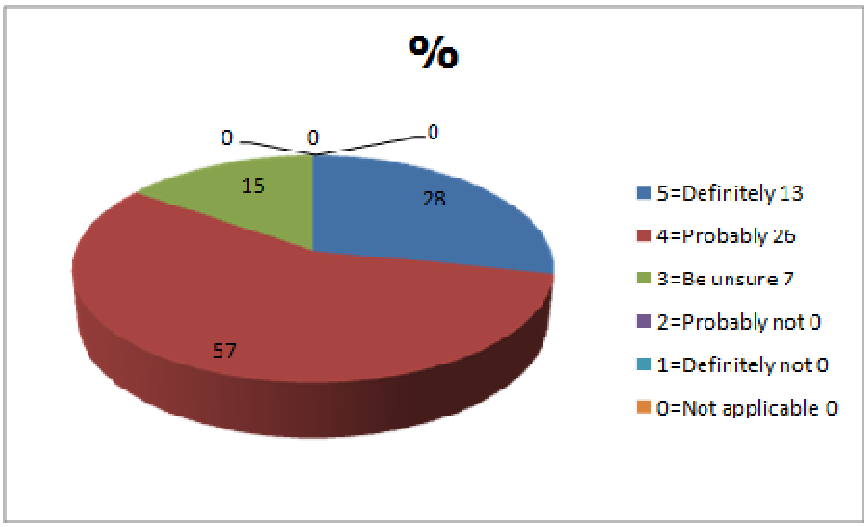


Figure B.48 The level of defined value in achieving the corrective actions after monitored

33. "Did you appropriately select software process development (SDP) for your project?"

Regarding to RQ3 which is related to the combination between Agile-Scrum and the project management of CMMI to close the gap, the degree of the appropriate selected software process development (SDP) when implementing the software development process (SDP) in plan-driven or Agile-driven are Probably (4), Definitely (5), Be unsure (3) and Probably not (2) as 39%, 35%, 24% and 2% respectively and there is none Definitely not (1) and Not applicable (0) from all 46 questionnaires which are defined as follow value; 5 is Definitely, 4 is Probably, 3 is Be unsure, 2 is Probably not, 1 is Definitely not and 0 is Not applicable. It is meaning that "Probably (4)" is the majority group at 39%.

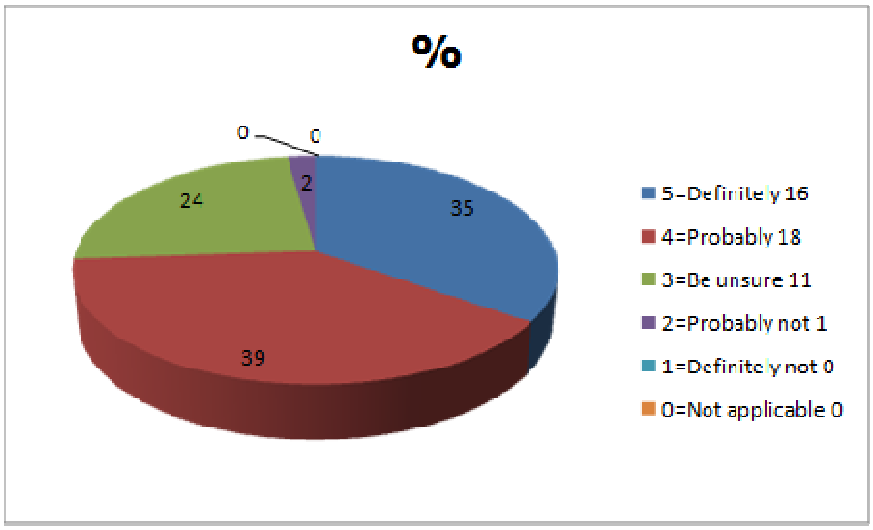


Figure B.49 The level of defined value in selecting software process development (SDP) for your project

34. "Did your project establish and manage the involvement of the relevant stakeholders?"

Regarding to RQ3 which is related to the combination between Agile-Scrum and the project management of CMMI to close the gap, the degree of the involvement of the relevant stakeholders when implementing the software development process (SDP) in plan-driven or Agile-driven are Probably (4), Definitely (5), Be unsure (3) and Probably not (2) as 48%, 24%, 20% and 9% respectively and there is none Definitely not (1) and Not applicable (0) from all 46 questionnaires which are defined as follow value; 5 is Definitely, 4 is Probably, 3 is Be unsure, 2 is Probably not, 1 is Definitely not and 0 is Not applicable. It is meaning that "Probably (4)" is the majority group at 48%.

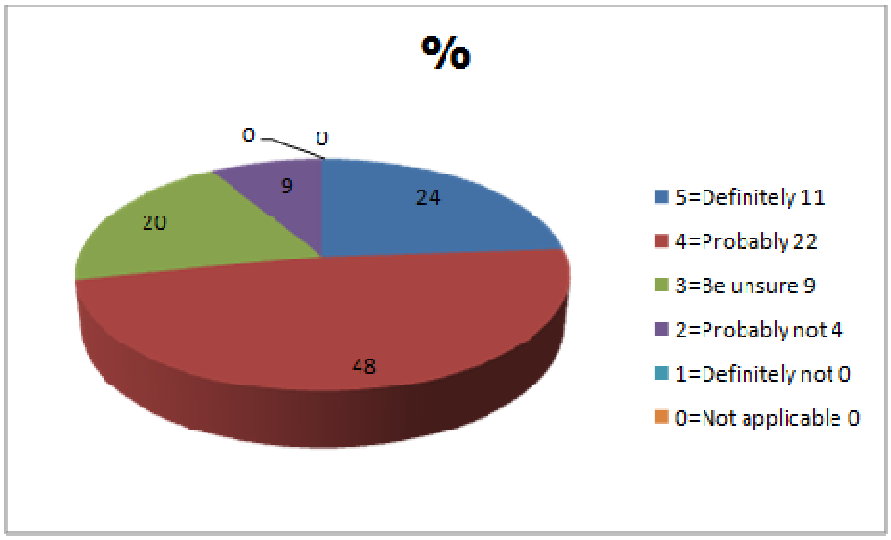


Figure B.50 The level of defined value in establishing and managing the involvement of the relevant stakeholders

35. "Did your project meet customer expectation (customer satisfaction rating >80%)?"

Regarding to RQ4 which is related to the benefit to synergize between Agile-Scrum and the project management of CMMI, the degree of customer satisfaction rating when implementing the software development process (SDP) in plan-driven or Agile-driven are Probably (4), Definitely (5), Be unsure (3) and Probably not (2), Not applicable (0) as 52%, 30%, 15% and 2% respectively and there is none Probably not (2) and Definitely not (1) from all 46 questionnaires which are defined as follow value; 5 is Definitely, 4 is Probably, 3 is Be unsure, 2 is Probably not, 1 is Definitely not and 0 is Not applicable. It is meaning that "Probably (4)" is the majority group at 52%.

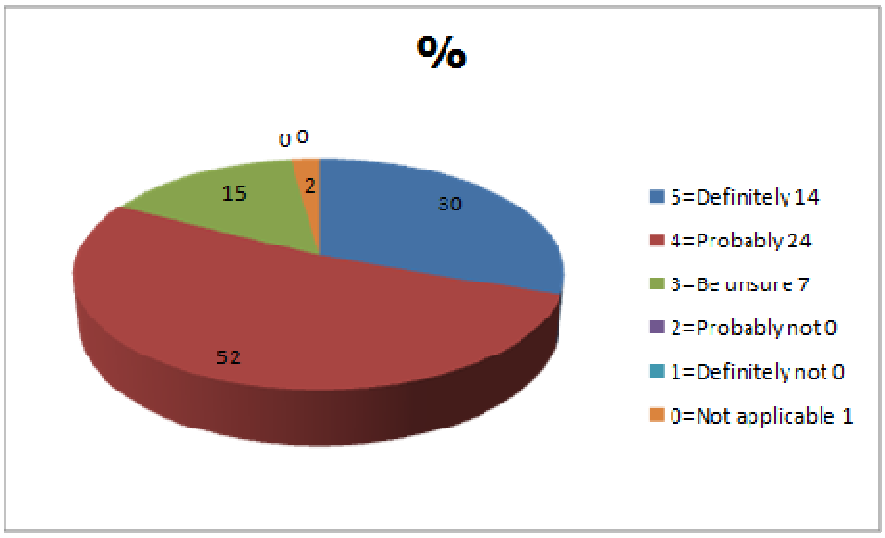


Figure B.51 The level of defined value in meeting customer expectation

36. "Did you have a good employee satisfaction rating (employee satisfaction rating >80%)?"

Regarding to RQ4 which is related to the benefit to synergize between Agile-Scrum and the project management of CMMI, the degree of customer satisfaction rating when implementing the software development process (SDP) in plan-driven or Agile-driven are Probably (4), Be unsure (3), Definitely (5) and Probably not (2), Not applicable (0) as 46%, 30%, 22% and 2% respectively and there is none Probably not (2) and Definitely not (1) from all 46 questionnaires which are defined as follow value; 5 is Definitely, 4 is Probably, 3 is Be unsure, 2 is Probably not, 1 is Definitely not and 0 is Not applicable. It is meaning that "Probably (4)" is the majority group at 46%.

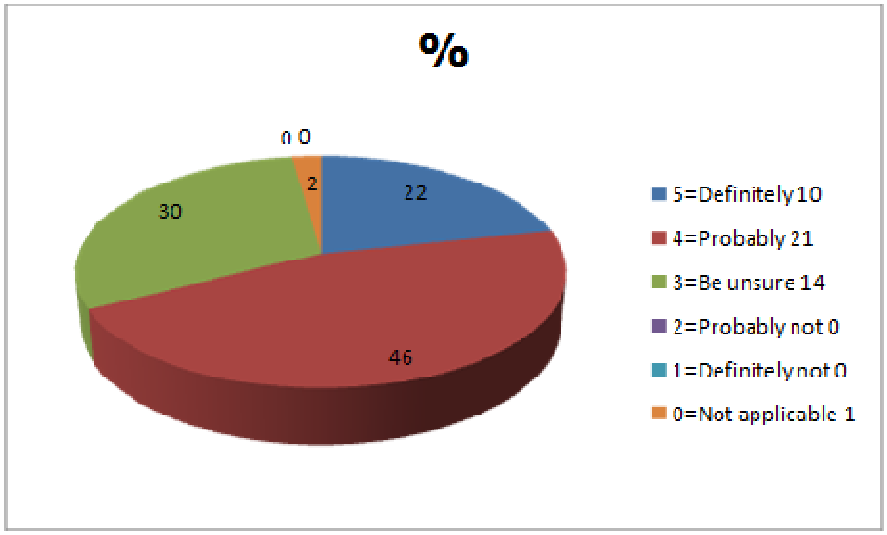


Figure B.52 The level of defined value in having
a good employee satisfaction rating

Appendix C: The summary of the majority group from the questionnaire

No.	Question topics	Majority Group
PART A: Organizational Characteristics		
1.	Organizational Geographic	China, Thailand, Switzerland, Vietnam and Malaysia
2.	Full time SW development employees	10>Medium Enterprise (ME)<=15 employees
3.	Active SW development projects	10>Medium Number (M)<=15 projects
4.	CMMI SW development projects	5>Small Number (S)<=10 projects
PART B: Project characteristics		
5.	SW development project name	software development project's name
6.	SW development team experience	
6.1	Project Manager	5>Master Level<=10 years
6.2	Software Process Improvement	2>Intermediate Level<=5 years 5>Master Level<=10 years
6.3	Application Domain Expertise	2>Intermediate Level<=5 years
6.4	Application Language Expertise	2>Intermediate Level<=5 years
7.	Major role in SW development team	Project Management
B1. Quality characteristics		
8.	Software Process Improvement program	software process improvements and/or quality management programs
9.	Capability Maturity Model Integration (CMMI) program	applying CMMI (Capability Maturity Model Integration)
10.	CMMI Target	
10.1	ML/CL in CMMI	ML (Maturity Level) in CMMI model
10.2	Maturity Level (2/3/4/5) in CMMI	ML3
11.	SW development project type	development
B2. Customer characteristics		
12.	Customer type	non-government customer

No.	Question topics	Majority Group
13.	Customer's domain software application	services
B3. Project size characteristics		
14.	Full time SW development team	5>Medium Number (M)<=10
15.	SW development project effort (actual) *1 person-month=176 person-hours	2>Small Number (S)<=5 person-months
16.	SW development project duration (actual)	Very Small Number (VS)<=6 months
17.	Estimated size of software	
17.1	Past experience estimations	Expert Level>10 years
17.2	Source Lines of Codes (KLOCs)	5>Medium Number<=10 KLOCs
B4. Software Development Process (SDP) characteristics		
18.	SW development Process	
18.1	Software Development Process (SDP)	Plan-Driven
18.2	Plan-driven/Agile-driven	Waterfall and Waterfall & prototype
PART C: Integrated Project Planning and Monitoring Control		
19.	SW development project effort (planned/actual-No.15)	
19.1	Project effort (person-month) as Planned *1 person-month=176 person-hours	2>Small Number<=5 person-months
19.2	Project effort (person-month) as Actual at the end	5>Medium Number<=10 person-months
19.3	Agile-Driven variation = +7.41%	Plan-Driven variation = +7.95%
20.	Project manager effort (planned/actual)	
20.1	Project Manager effort (person-month) as Planned *1 person-month=176 person-hours	Very Small Number <=2 person-months

No.	Question topics	Majority Group
20.2	Project Manager effort (person-month) as Actual at the end	Very Small Number <=2 person-months
20.3	Agile-Driven variation = +26.32%	Plan-Driven variation = +18.90%
21.	SW development cost (planned/actual)	
21.1	Software Budget ($ USD) as Planned	40,000>Medium Number<=60,000 USD
21.2	Software Budget ($ USD) as Actual at the end	40,000>Medium Number<=60,000 USD
21.3	Agile-Driven variation = +10.97%	Plan-Driven variation = +13.14%
22.	SW quality cost (planned/actual)	
22.1	Software Process Improvement Budget ($ USD) as Planned	1,000>Small Number<=2,000 USD
22.2	Software Process Improvement Budget ($ USD) as Actual at the end	1,000>Small Number<=2,000 USD
22.3	Agile-Driven variation = +7.56%	Plan-Driven variation = -25.44%
23.	SW development project duration (plan/actual-No.16)	
23.1	Project duration (month(s)) as Planned	Very Small Number <=6 months
23.2	Project duration (month(s)) as Actual at the end	Very Small Number <=6 months
23.3	Agile-Driven variation = +11.54%	Plan-Driven variation = +20.80%
24.	SW development defects/reworks (planned/actual)	
24.1	Agile-Driven variation = +50.00%	Plan-Driven variation = +31.82%.
25.	SW development defects/reworks removal (planned/actual)	
25.1	Agile-Driven variation = -17.65%	Plan-Driven variation = -87.50%
26.	Understanding of Software Development Process (SDP)	Probably (4) = 54%
27.	SW development resources	Probably (4) = 35%
28.	SW development on-time deliverables	Probably (4) = 39%

No.	Question topics	Majority Group
29.	SW development in-budget deliverables	Probably (4) = 48%
30.	SW development establishes and maintains as planned	Definitely (5) = 41%
31.	SW development monitor as planned	Probably (4) = 50%
32.	SW development achieve the corrective actions	Probably (4) = 57%
33.	Appropriate software process development (SDP) in a project	Probably (4) = 39%
34.	SW development establishes and manages stakeholders	Probably (4) = 48%
35.	Customer satisfaction rating	Probably (4) = 52%
36.	Employee satisfaction rating	Probably (4) = 46%