A PRACTICAL MEASUREMENT CAPABILITY MODEL FOR SOFTWARE ORGANIZATIONS UTILIZING AGILE APPROACHES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

MURAT SALMANOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

APRIL 2018

A PRACTICAL MEASUREMENT CAPABILITY MODEL FOR SOFTWARE ORGANIZATIONS UTILIZING AGILE APPROACHES

Submitted by **MURAT SALMANOĞLU** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in the Department of Information Systems, Middle East Technical University** by,

Prof. Dr. Deniz Zeyrek Bozşahin
Dean, **Graduate School of Informatics** ____________________

Prof. Dr. Yasemin Yardımcı Çetin
Head of Department, **Information Systems** ____________________

Prof. Dr. Onur Demirörs
Supervisor, **Information Systems, METU & IZTECH** ____________________

Examining Committee Members:

Prof. Dr. Ferda Nur Alpaslan
Computer Engineering, METU ____________________

Prof. Dr. Onur Demirörs
Information Systems, METU & IZTECH ____________________

Assoc. Prof. Dr. Altan Koçyiğit
Information Systems, METU ____________________

Asst. Prof. Dr. Ömer Özgür Tanrıöver
Computer Engineering, Ankara University ____________________

Asst. Prof. Dr. Murat Yılmaz
Computer Engineering, Çankaya University ____________________

Date: 16/04/2018

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this wok.

Name, Last name : Murat SALMANOĞLU

Signature :

ABSTRACT

A PRACTICAL MEASUREMENT CAPABILITY MODEL FOR SOFTWARE ORGANIZATIONS UTILIZING AGILE APPROACHES

Salmanoğlu, Murat
Ph.D., Department of Information Systems
Supervisor: Prof. Dr. Onur Demirörs

April 2018, 147 pages

Measurement is the foundation for successful software management. However, it is not easy for software organizations to evaluate their measurement practices and to determine what they should do to improve them. There are models to evaluate capability and maturity of measurement processes. However, they frequently focus on the measurement process with a guidance from a well-defined capability model, like CMMI or SPICE. Many of the software organizations following agile methodologies do not prefer to apply these process-centric maturity models. This study presents a model to assess measurement capability of software organizations by inspecting individual measures, independent from the software development approach and the process architecture organizations use. The model exemplifies measures for aspects and defines generic practices for three capability levels. Organizations can use the model to determine and improve their measurement capability. This research includes action research and exploratory case studies conducted during the development of the model, an explanatory case study conducted to implement the model, and its results. Case studies demonstrate that the model provides additional benefits to organizations utilizing agile approaches while providing similar results with the process-centric models. The findings of the explanatory case study indicate that the results of the model are accepted and found beneficial by the employees from small-, medium- and large-scale organizations that participate in the study.

Keywords: Measurement Capability, Software Development, Software Measurement, Software Development Processes, Agile Software Development.

ÖZ

ÇEVİK YAKLAŞIMLAR KULLANAN YAZILIM ORGANİZASYONLARI İÇİN PRATİK BİR ÖLÇÜM YETENEĞİ MODELİ

Salmanoglu, Murat
Doktora, Bilişim Sistemleri Bölümü
Tez Yöneticisi: Prof. Dr. Onur Demirörs

Nisan 2018, 147 sayfa

Ölçüm, başarılı yazılım yönetimi için temeldir. Ancak, yazılım organizasyonları için ölçüm aktivitelerini değerlendirmek ve bunları iyileştirmek için yapılacakları belirlemek kolay değildir. Ölçüm süreçlerinin yetenek ve olgunluğunu değerlendiren modeller bulunmaktadır, ancak bunlar sıklıkla CMMI ve SPICE gibi iyi tanımlanmış yetenek modellerini temel alarak ölçüm süreçlerine odaklanırlar. Çevik yöntemleri takip eden organizasyonların çoğu bu süreç odaklı olgunluk modellerini uygulamayı tercih etmiyor. Bu çalışma, organizasyonların kullandığı yazılım geliştirme yaklaşımından ve süreç mimarisinden bağımsız olarak, tekil ölçümleri inceleyerek yazılım organizasyonlarının ölçüm olgunluğunu değerlendiren bir yöntem sunmaktadır. Yöntem farklı bakış açıları için ölçümleri örneklemekte ve üç yetenek seviyesi için genel uygulamalar tanımlamaktadır. Organizasyonlar bu yöntemi ölçüm olgunluklarını anlamak ve iyileştirmek için kullanabilirler. Bu araştırma; modelin geliştirilmesi sırasında yürütülen eylem araştırması ve keşfedici vaka çalışmalarını, modelin uygulanması için yürütülen açıklayıcı vaka çalışmasını ve onun sonuçlarını içermektedir. Vaka çalışmaları yöntemin çevik yaklaşımlar kullanan organizasyonlara süreç odaklı yöntemlerle benzer sonuçlar sunmasına rağmen ek faydalar sunduğunu göstermiştir. Açıklayıcı vaka çalışmasının bulguları yöntemin sonuçlarının çalışmaya katılan küçük, orta ve büyük çaplı organizasyonların çalışanları tarafından kabul edildiğini ve faydalı bulunduğunu göstermiştir.

Anahtar Kelimeler: Ölçüm Yeteneği, Yazılım Geliştirme, Yazılım Ölçümü, Yazılım Geliştirme Süreçleri, Çevik Yazılım Geliştirme.

for my father

ACKNOWLEDGEMENTS

First of all, I express my sincere gratitude to my supervisor Onur Demirörs for his insight and guidance during my study. I am thankful to him for his continuous stimulation, suggestions, and support. I learned a lot from him not only in academic area but also about life. He has been an excellent mentor.

I would like to thank my committee members Ferda Nur Alpaslan and Altan Koçyiğit for their ideas and support throughout the thesis study.

I also thank to my friends Ali Yıldız and Ahmet Coşkunçay for their contributions to this work and being good friends.

I would like to thank my wife, Gökçen, for her invaluable support and endless patience. I am lucky to have her in my life with her limitless love.

Lastly, I am grateful to my parents. They taught me to be curious about everything, always question the why, and work persistently for my goals. Without their love, guidance, and trust in me I would not be who I am. [Son olarak, her şeyi merak etmeyi, her zaman nedenleri sorgulamayı, ve hedeflerim için inatla çalışmayı bana öğrettikleri için ebeveynlerime minnettarım. Onların sevgisi, rehberliği ve bana olan güvenleri olmadan şu an olduğum kişi olamazdım.]

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AC	Academic Collaboration
AI	Action Item
BPMM	Business Process Maturity Model
CFP	COSMIC Function Point
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
COSMIC	The Common Software Measurement International Consortium
DevOps	Development and Operations
DoD	Department of Defense
DS	Decision Support
EC	External Collaboration
FSM	Functional Size Measurement
FT	Fast Track
GP	Generic Practice
GQM	Goal question Metric
ICT	Information–Communication Technologies
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IFPUG	International Function Point Users Group
ISBSG	International Software Benchmarking Standards Group
ISO	International Organization for Standardization
IT	Information Technologies
JTC	Joint Technical Committee
MCL	Measurement Capability Level
MCM	Measurement Capability Model
MCMM	Measurement Capability Maturity Model
MeSRAM	Measurement System Robustness Assessment Method
METU	Middle East Technical University
MI	Metrics Infrastructure
MIS–PyME	Marco metodológico para la definición de Indicadores de Software orientado a PyME (La pequeña y mediana empresa) – Methodological framework for the definition of Software Indicators oriented to SMEs
MO	Metrics Organization
MPC	Message passing coupling
MU	Metrics Used
NS	Not Satisfied
OM	Organizational metrics
OPM3	Organizational Project Management Maturity Model

PhD	Philosophiae Doctor (Doctor of Philosophy)
PSM	Practical Software and Systems Measurement
QMMG	Quality Management Maturity Grid
RCI	Ratio of Cohesive Interactions
RQ	Research Question
SCAMPI	Standard CMMI Appraisal Method for Process Improvement
SME	Small and Medium–Sized Enterprise
SPICE	Software Process Improvement and Capability Determination
VPOS	Virtual Point of Sale

CHAPTER 1

INTRODUCTION

Organizational improvement towards more productive development activities requires objective evaluation, and measures are important components of any organizational evaluation. Measurement in a software developing organization includes definition and collection of measures in process, project, and product levels. Continuous measurement is an invaluable input to understand performance, which is required for any improvement activity.

Organizations need to keep their measures and measurement practices up-to-date and make sure to use them at the right context to obtain optimal benefit from them. It is a demanding task to analyze and improve measurement activities. Software organizations frequently fail to cope with this demand because of the inherent difficulties of their products and processes (Ayca Tarhan & Demirors, 2012).

In software projects, most of the time, produced software products are unique, the processes are not standardized, and practices require adaptation to fast-paced technological advancements. To be able to adapt this always-changing environment, organizations are required to tailor their measures, measurement methods, and ways of using these measures according to their own needs.

Organizations, which do not want to implement process centric assessment and improvement frameworks, may benefit from a practical approach that guides them through possible measures and practices. That approach can relieve them from the workload required with a process centric framework. In this study, we present a measurement capability model (MCM), which guides organizations through measures for different activities of development lifecycle. Organizations can focus only on the activities that require improvement. By using different levels of practices listed in the model, they can assess their capability for selected measures and create an improvement plan accordingly.

In the following parts of this chapter, first, the background of the problem is explained, then, the purpose and significance of the study is given, which is followed by the research strategy, and lastly the structure of the study is explained.

1.1 Background of the Problem

Process capability and maturity models like CMMI (CMMI Product Team, 2010) and ISO 33000 (ISO/IEC 33001, 2015)(former ISO 15504 (ISO/IEC 15504-5, 2012)) help organizations to assess and improve every aspects of their development processes in all process areas. They define model specific assessment methods such as SCAMPI (SCAMPI Upgrade Team, 2011). These models cover measurement processes as well in addition to other processes. Similar to these process centric models, there are assessment models focusing on measurement processes and aiming to help organizations on how to assess and improve their measurement activities. Available models in the literature on measurement maturity and capability can be broadly divided into two categories. First category includes approaches that deal with the utilization of measures and with supporting components of measurement processes. Approaches in second category aim to assess maturity of organizations' measurement process.

For the first group of works, we can identify four models that can be used to evaluate the capabilities of supporting components of measurement in an organization. Daskalantonakis' model (Daskalantonakis & Yacobellis, 1990) is the first model of the first category. It aims to assess maturity of measurement technology used in an organization. This model focuses on the technology used for software measurement and does so by using CMM for software (Paulk, Curtis, Chrissis, & Weber, 1993) as its foundation. An assessment guide for this model is prepared by Budlong and Peterson (Budlong & Peterson, 1995).

Tarhan and Demirörs (Ayça Tarhan & Demirors, 2008) define an assessment process, model, and tool (Kırbaş, 2007), to evaluate usability of collected metrics for statistical process control in software organizations. This is a significant study focusing on usage of the metrics rather than capabilities of measurement processes.

An approach to improve existing measurement framework of an organization is proposed by Mendonca et al. (Mendonca, Basili, Bhandari, & Dawson, 1998). They aim to use data mining techniques to identify whether current metrics collected in an organization include meaningful information that the user currently does not use. This methodology is mainly about the capability of metrics to fulfill organizational needs. It does not focus on organization's measurement capability.

Staron and Medig (Staron & Meding, 2015) propose Measurement System Robustness Assessment Method (MeSRAM), a model aiming to assess robustness of organizational measurement programs. Consisting of a robustness model and an assessment method, MeSRAM evaluates robustness, or continuity, of a measurement program.

The four approach given above aim to evaluate measurement activities from different perspectives, but they do not cover assessment of organizational measurement capability. The two models given below focus on the assessment of organizational measurement capability.

First model aiming to evaluate organizational measurement program's capability is Measurement-CMM (Niessink & Van Vliet, 1998). It tries to give answers to the questions: "How to introduce measurement in a software organization? What are the necessary steps to set up a measurement program and in which order should they be per-formed? How can existing measurement programs be enhanced?" It provides a 5 level maturity scale similar to software CMM (Paulk et al., 1993) and suggests that organizations should adopt software CMM together with measurement CMM.

Second model focusing on measurement capability is Methodological framework for the definition of Software Indicators oriented to SMEs (MIS-PyME - Marco metodológico para la definición de Indicadores de Software orientado a PyME (La pequeña y mediana empresa)) (M Díaz-Ley, Garcí, & Piattini, 2008) (María Díaz-Ley, García, & Piattini, 2010). It focuses on small-medium enterprises and includes two parts: measurement program definition methodology and measurement capability maturity model. Its capability assessment model aligns with ISO 15504 (ISO/IEC 15504-5, 2012) and includes three components: maturity levels and attributes that need to be fulfilled, assessment process that aims to determine the capability, and an interface with MIS-PyME methodology to define measurement programs.

The two models, Measurement CMM and MIS-PyME, aim to provide guidance to software developing organization to assess their measurement processes. Even though they have some gaps, like the lack of a defined assessment model in measurement CMM, or MIS-PyME's focus on small-medium enterprises, they are successful models in the domain they are focusing on. However, they do not answer the needs of all organizations and leave a gap in the literature. Both of the models are based on traditional process centric improvement models: Software CMM and ISO 15504. Organizations utilizing dynamic software development approaches, like agile methodologies (Beck et al., 2001), are drawn away from these traditional models. These organizations demand dynamic models that can help them to improve specific points in their lifecycles as needed. Therefore, instead of process centric approaches, they require adaptive, flexible approaches, which help them focus on specific points they want to assess.

Organizations working with dynamic development models cannot receive intended benefits from available measurement capability models as they refrain from a process centric improvement initiative. Lack of an up-to-date measurement capability model answering the needs and demands of current dynamic organizations creates a gap in the literature.

1.2 Purpose of the Study

This study aims to fill a gap in the available literature by establishing a measurement capability model (MCM), which can satisfy the needs of organizations requiring practical measurement assessment approaches. During the study, a model will be developed and it will be applied in software developing organization to evaluate its applicability. The model is expected to have some key properties, which are listed below:

- The model should provide a practical approach for software organizations for assessing their measurement activities and for determining their measurement capability.

- With the help of this model, organizations should be able to focus on any aspect of their development lifecycle to improve related measures and practices to achieve higher capability levels. Aspects represent “sets of interrelated and interacting activities” in a software development lifecycle (Ozcan-Top & Demirörs, 2015).
- The model should define measurement practices related todifferent capability levels.
- The model should enable the organizations to identify the gaps of their current measurement capability with the desired level.

The study aims to define main parts of the model version one, to work with software organizations to develop a practical model, to apply the model in software organizations to observe difficulties and benefits of the model. Exploratory studies conducted in real organizations are used to explore, test, improve, and create the model version two. The model version two is implemented in several organizations as a part of an explanatory case study to observe the difficulties and benefits of the model in practice.

The model is expected to help organizations requiring a practical approach to assess and improve their measures and measurement activities without forcing them to implement an organization wide process improvement approach.

1.3 Research Strategy

To reach the goal, the study follows a design-science research approach. It starts with an action research phase in which the identification of main components of a possible model version one is studied. After this phase, there are four exploratory case studies to build the model version two. In the last phase, there is an explanatory case study to apply the model version two in real settings and observe its results.

The design-science approach used in this study follows the guidelines for design-science in information systems presented by Hevner et al. (Hevner, March, Park, & Ram, 2004).

The supporting qualities of our research for design science can be summarized with respect to the following properties:

- Design as artifact: The result of research is a purposeful Information Technologies (IT) artifact, a measurement capability model. The model addresses an important organizational problem. The research also describes its domain and usage.
- Problem Relevance: The model solves an important business problem in addition to filling a gap in the literature.
- Design Evaluation: The model is evaluated in different case studies conducted with several organizations and professionals, and the results are demonstrated.
- Research Contributions: This research produces a verifiable artifact as a contribution, a model to assess measurement capability in organizations.

- Research Rigor: During the construction of the model, in the first phase of the research, action research method is used. After that, the model is finalized with the use of four exploratory case studies and at the end, it is tested with an explanatory case study.
- Design as a Search Process: We propose the solution by conducting research in the problem domain. We also test the proposed solution in the problem domain until we reach a satisfactory result.
- Communication of Research: The resulting model answers the needs of the industry, as demonstrated during the case studies. Understanding and improving measurement capabilities of an organization can help to decrease technological risks and increase quality of the products. The model also fills a gap in literature therefore addresses academic audiences.

As a first step in this research, we implement action research to solve a real problem in an organization. Action research is a useful technique for the complex social settings, where it is hard to study a problem independent from the context (Baskerville, 1999). After the action research, we conduct four exploratory and one explanatory case studies to improve, finalize, and apply the model (Figure 1).

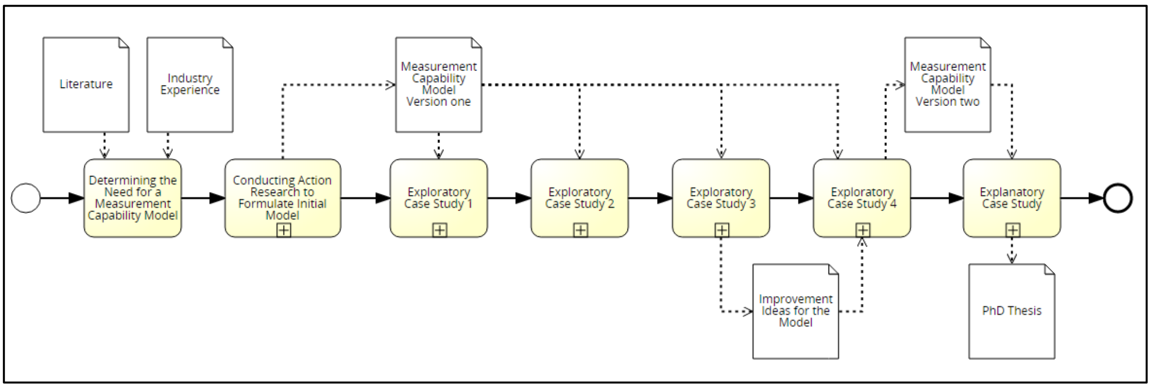

Figure 1 Overview of the research process followed

Before starting the first phase with the action research, we identify the research questions (RQ) guiding the research.

RQ-1. What are the main components to be used in a model to analyze software measurement infrastructure and identification of improvement opportunities?

During action research, an answer for the first research question is sought and main components of a measurement capability model version one is structured. Current literature and available best practices are used during the research. When the knowledge is combined with the academic and industrial experiences gained during the action research, the problem context provides main output. The preliminary components of the proposed model provide an answer for the first research question.

RQ-2. How well do the identified main components work to evaluate measures in software development process?

After the model version one is identified in the action research, an exploratory study with a case from a software developing organization is planned, designed and performed to answer second research question. Exploratory studies aim to explore ideas related with the phenomenon interested by the researcher. They can help the researcher to understand how to proceed with the initial ideas. The study is conducted by using retrospective data from the organization both before and after a measurement improvement initiative. The results show that the measurement capability model has the potential to help organizations to assess their measurement capability and provides answer for the second research question (Salmanoğlu, Coşkunçay, Yıldız, & Demirörs, 2018).

RQ-3. How useful is the model version one for software organizations?

RQ-3.1. How clearly does the model version one reflect the state of the organization with respect to their measurement capability?

RQ-3.2. What differences do results of the model version one include when compared with available measurement capability evaluation methods?

After the potential of the model version one is observed in the first exploratory case study, a second exploratory study is conducted with three cases, an organization's three independent divisions as three different cases (İnce, 2016). For each case, measurement capability model is applied along with two other measurement assessment methods. The study is carried out by an independent researcher who is an employee of the organization. This study aims to find answers for the research questions 3, 3.1, and 3.2. Application of the model by an employee of the organization also provides some initial ideas about the potential of the model as a self-evaluation tool. The results of three methods are used to evaluate the consistency between the results of our model and other available methods. The results are also evaluated to identify any additional benefits or difficulties of our model, if there is any. The results of the case study are used to evaluate and refine the model.

RQ-3.3. What benefits and shortcomings does the model version one provide to software organizations?

Third exploratory case study is planned to answer the research question 3.3. Its main aim is to use the model in a large organization and gather data about the perceived benefits and shortcomings of the model in the organization. Three branches of a large software development organization are used as three different cases. After the application of the model in the organization, the results are shared with senior management and their opinions about the model are noted.

RQ-4. Which aspects and core measures should be added to the model version one to update it to be applied in organization utilizing agile software development approaches?

After answering first three main research questions and having evidence about the applicability of the model in real organizations, a fourth exploratory study is planned to provide answer to the fourth research question. This study aims to improve the set of core measures in the model and be sure about their suitability for organizations applying agile development methodologies. Four experts with extensive agile method experiences attend the study; two participants are from the industry and two participants are working as researcher/consultant. During the study, interviews are conducted by using pre-defined questions about the applicability of the identified measures for agile settings. At the end of this study, the measure set is updated and the model version two is finalized.

RQ-5. How useful is the model for software organizations, specifically for those using agile software development approaches?

RQ-5.1. How clearly does the model reflect the state of the organization with respect to their measurement capability?

RQ-5.2. How fully do the identified improvement opportunities capture organization's potential?

RQ-5.3. How much new information does the model provide to organizations?

An explanatory case study is planned after the finalization of the model to answer research questions 5, 5.1, 5.2, 5.3. Explanatory studies aim to explore the data in a deeper level when than exploratory studies in order to explain the phenomenon interested by the researcher. During this study, the model version two is applied in four organizations utilizing agile development approaches: two small, one medium, and one large organizations. After the application of the model and identification of the measurement capabilities, interviews are conducted with participants from each of these organizations. During the interview, their opinions about the usefulness of the model, especially for agile organizations, are discussed and noted.

1.4 Significance of the Study

This study presents a practical measurement capability model for organizations to assess and improve their measurement activities without a need to follow a process centric process improvement approach. It is the first measurement capability model aiming agile organizations that is built by working closely with the needs of agile software organizations through systematic case studies to provide a practical.

It is the first measurement capability model with agile focus that enables organizations to choose improvement aspects and lists measures under different aspects. According to their needs, organizations can choose any combination of these aspects for evaluation. Having the opportunity to choose necessary aspects, organizations can select them according to their changing goals. This provides the opportunity to identify improvement points according to the goals of the organizations.

The model provides a granulated structure with its list of aspects, entities, attributes, and measures. Organizations can examine their practices in detail by using this detailed structure.

Agile methods gain popularity among software organizations with their fast-paced approach to software development. It is the first measurement capability model aiming to answer the needs of organizations preferring agile methodologies. Organizations that prefer to use agile techniques demand adaptable solutions to their problems. The model developed in this study aims to answer their needs to analyze and improve their measurement activities.

The model provides to organizations a list of sample measures classified according to aspects. It also includes the entities and attributes that these measures aim to quantify. Even though these measures are available in the literature, the model uniquely organizes and provides them to the organizations in a structural manner. Organizations can choose, evaluate, and implement them according to their needs.

The study provides the results of the application of model during the case studies. These results and lessons learned from the case studies can guide organizations for the application of the model. Case studies can also be used by organizations to compare their performances with the organizations described in the studies.

The model guides the organizations through the process of building a measurement baseline with the defined aspects, entities, attributes, core measures, their descriptions, and general practices. By using the components of the model, they can analyze and understand their needs, determine their measurement capability goals, assess their current situation, determine improvement points and conduct them.

During the research model is applied in real settings and the results demonstrate that organizations utilizing agile approaches can receive significant benefits from this model. One action research, four exploratory case studies and one explanatory case study is conducted during this research. 11 different cases are used from 7 different organizations. All of these organizations receive some benefits.

Measurement capability model is developed by using action research to define initial components; therefore, is takes its roots from practical needs of organizations. It is also compared with to other models from literature and provided additional benefits when compared with them.

1.5 Structure of the Thesis

This thesis includes five more chapters.

Second chapter gives a discussion about the literature on measurement in software development, general maturity and capability models, and measurement capability and maturity.

Third chapter explains the details of the research method conducted in this study. It starts with the action research conducted to create initial ideas and continues with the exploratory studies that helped to build the model.

Fourth chapter provides the details of the measurement capability model version two that is developed to help organizations with their measurement practices.

Fifth chapter includes the explanatory study conducted to evaluate the results and effects of measurement capability model version two in organizations utilizing agile software development approaches.

At the end, the sixth chapter discusses the contributions and limitations of this study and provides future work regarding with this thesis.

CHAPTER 2

STATE OF THE ART

This chapter includes four sections. Chapter starts with a section giving the description of software measurement and related literature. Second section summarizes available common capability maturity models related tosoftware engineering. Current measurement capability and maturity methods are given in third section. Last section discusses related studies and their comparison with the model suggested in this thesis.

2.1 Software Measurement

Measurement is an important part of any scientific or engineering activity. In 1883 Lord Kelvin says: "*I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely, in your thoughts, advanced to the stage of science, whatever the matter may be.*" (Thomson, 1883). Software engineering discipline is not an exception. From the first days of the foundation of software engineering area (Naur & Randell, 1969) measurement was an important part.

With the introduction of the term software engineering, research on metrics, measures, and measurement is started. In 1974 Wolverton mentions about measuring and estimation software productivity by using code written in a person-month (Wolverton, 1974). In 1976 McCabe uses cyclomatic complexity to measure the complexity of a software (McCabe, 1976). Albrecht presented the idea of measuring the functionality of a software in 1979 and created the idea of function points (Albrecht, 1979). From the initial steps of software metric and measurement in late 70's to today, this area grows rapidly. Today measures and metrics are a must-have part of producing software.

Fenton and Bieman define measurement as "the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way so as to describe them according to clearly defined rules." (Fenton & Bieman, 2014) Therefore, to measure any entity, one first needs to decide which attribute to measure about this entity. Then, define clear rules to describe this attribute.

With the increasing number of suggested metrics and measures, in the last two decades of the 20th century, studies in measurement methodologies have shown major progress. Measurement methodologies aim to provide practitioners a guideline while implementing their measurement strategies. Among these, Balanced Scorecard (Robert S Kaplan & Norton, 1992) is a well-established framework for supporting managers with financial and operational measures. Balanced Scorecard has four perspectives, which are financial, customer, internal business, and innovation and learning. Ensuring managers to have a complete view of their organization by suggesting measures to be grouped under these perspectives, Balanced Scorecard provides limited guidance on how to design a measurement system and how to identify metrics (R S Kaplan & Norton, 1993). Ibanez (Ibáñez, 1998) provides a tailoring to the perspectives of Balanced Scorecard in order to achieve fitness of the methodology to the IT domain.

The Goal Question Metric (GQM) (van Solingen et al., 2002) is a measurement approach that is introduced for software organizations. It is structured around the idea that a software measurement program can be established by starting with goal definitions and then mapping these goals to data and metrics. This top-down approach is based on identifying the goals, the questions that break down the goals to characterize objects of measurement, and metrics that are intended to explain the questions.

2.2 Capability and Maturity Models

In process management terms, maturity represents an organization's ability to improve continuously a particular part of its processes. Maturity and capability models aim to assist organizations to assess their level of maturity and draw a framework to improve. Higher the maturity level, higher the organization's ability to learn from its measures, outputs, errors and continuously improve its processes by using them. There are capability and maturity models for nearly every aspect of all organizational processes. Some of the capability maturity models can be listed as: Capability maturity model integration (CMMI Product Team, 2010) for processes of organizations developing software or systems. Organizational Project Management Maturity Model (OPM3) (Project Management Institute, 2003) for project management. Quality Management Maturity Grid (QMMG) (Crosby, 1979) for quality. Business Process Maturity Model (BPMM) (Curtis & Alden, 2007) for business process management. Open Source Maturity Model (Petrinja, Nambakam, & Sillitti, 2009) for open source software development. Service Integration Maturity Model (The Open Group, 2011) for service oriented architecture. E-Learning Maturity Model (Marshall & Mitchell, 2002) for e-learning. Data Management Maturity Model (CMMI Institute, 2014) for managing data. People Capability Maturity Model (Curtis, Hefley, & Miller Sally A., 2001) for managing human resources. Test Maturity Model (Burnstein, Homyen, Grom, & Carlson, 1998) for test processes.

One of the first introduced capability maturity models is Software Capability Maturity Model (Software-CMM) (Paulk et al., 1993), which takes its roots form the process maturity framework defined by Humphrey (Humphrey, 1988). CMM later evolved into Capability Maturity Model Integration (CMMI) (CMMI Product Team, 2010), which not only focus on software development processes, but also includes system development.

CMMI is one of the most commonly used maturity model in the software industry. Organizations may be requested to have a certain CMMI level capability to have a government contract, especially for software development contracts in the USA. CMMI defines 5 maturity levels: initial, managed, defined, quantitatively managed, and optimizing. It includes 24 process areas distributed under each maturity level. Standard CMMI Appraisal Method for Process Improvement (SCAMPI) (SCAMPI Upgrade Team, 2011) is used to appraise an organization and award a maturity level or capability level for its process areas.

Another common model used for capability determination is ISO/IEC 33001 (ISO/IEC 33001, 2015), formerly ISO/IEC 15504 (ISO/IEC 15504-5, 2012). Which was also termed as Software Process Improvement and Capability Determination (SPICE). It includes set of standards aiming to guide and assess organizations developing software. It is initially derived from "ISO/IEC 12207- Systems and software engineering – Software life cycle processes" (ISO/IEC 12207, 2008), which aims to define all tasks required for development and maintaining software.

ISO/IEC 33001 defines six capability levels: incomplete, performed, managed, established, predictable, optimizing. There are around 40 processes in the standard and it includes a guide for the assessment of organizations. The results of the assessment can be used to determine capability of the processes and to identify improvement opportunities.

2.3 Measurement Maturity and Capability Models

Common process improvement frameworks (ISO /IEC 33000 –formerly ISO/IEC 15504 (ISO/IEC 15504-5, 2012; ISO/IEC 33001, 2015), CMMI (CMMI Product Team, 2010)) provide guidance to assess several process areas and define practices for them. They also include measurement among these process areas. For their assessment processes, these frameworks have specialized methods, like SCAMPI (SCAMPI Upgrade Team, 2011) for CMMI. In the literature there are also several models focusing specifically on software measurement practices.

Tow broad categories can be used to classify methods aiming the measurement processes. First category includes studies related to the utilization of the measures and supportive components of measurement process. Second category includes studies directly related to the maturity or capability of software measurement processes. The first seven approaches in the following paragraphs are from the first category and the next two are from the second category.

The first approach of the first category is Daskalantonakis' (Daskalantonakis & Yacobellis, 1990) approach for software measurement technology assessment. They define a maturity model focusing on the measurement processes. Software CMM (Paulk et al., 1993) is the foundation of this method. Method includes five maturity levels similar to CMM. These levels are Initial, Repeatable, Defined, Managed, and Optimized. Method includes ten themes and each theme has definitions for maturity levels on a scale of one to five. Measurement technology maturity level is determined by assessing the conformance of the organization to these themes. Assessment tool of the model includes yes-no questions for each maturity level. A guide for evaluation for this maturity model is described by Budlong & Peterson (Budlong & Peterson, 1995). Focus of this model is the technology used in software measurement, it does not directly deal with measurement practices or processes.

The second approach is introduced by Tarhan and Demirörs (Ayça Tarhan & Demirors, 2008). It provides an assessment process, model and tool (Kırbaş, 2007) to evaluate software metrics to understand their usability for statistical analysis. It defines a standard set of usability attributes. These attributes are used to evaluate metrics with respect to the ratings given for the attributes in four scales: fully usable, largely usable, partially usable, and not usable. There are four assets for assessment: process execution record, process similarity matrix, process execution questionnaire, and metric usability questionnaire. This approach focuses on evaluating usability of the metrics; it cannot be classified as a measurement capability assessment approach.

Mendonca's (Mendonca et al., 1998) approach aims to improve existing measurement frameworks. It uses data mining to understand whether organizational metrics include meaningful data that the users are not aware. By comparing existing measures with the organizational goals, they try to determine whether the measures are necessary or whether there are enough metrics. This approach helps to understand the metrics and their capability to fulfil the needs.

Berry and Vandenbroek (Berry & Vandenbroek, 2001) suggests a targeted assessment of the software measurement process by aiming target specific software processes and by assessing the relationship between measurement and other processes. Their approach targets people who involve in the processes and includes surveys to understand their satisfaction with measures, processes, and their relations.

Another approach is the Measurement System Robustness Assessment Method (MeSRAM). It aims to assess the robustness of measurement programs (Staron & Meding, 2015). MeSRAM consists of a robustness model and an assessment method to measure the robustness of the measurement programs. It defines a robust measurement program as a program that is able to incorporate a broad set of measures and that has a support organization and a solid infrastructure. It focuses on the type of entities measured and whether they have proper definition. MeSRAM examines robustness according to seven main categories; metrics used, decision support, metric infrastructure, organizational metric maturity, metrics organization, external collaboration, collaboration with academia. There are also five subcategories under category metrics used; business metrics, product metrics, design metrics, organizational performance metrics, and project metrics. For each category and subcategory, there is a questionnaire for the organizations to answer. Model evaluates the answers to understand robustness for each category. MeSRAM cannot be classified as a measurement capability method, as it aims to understand the robustness or continuity of a measurement program despite the changes.

Another approach is practical software and systems measurement (PSM) (DoD and U.S. Army 2000). It includes a set of principles, best practices, and techniques that take part in tailoring, applying, implementing, and evaluating activities of a project measurement process. PSM also provides case studies, measurement tables, and indicator examples that supplement the process. It mainly helps to define a measurement process but does not include a component for capability assessment.

ISO/IEC 15939 (ISO/IEC 2017) is a standard providing a framework for measurement processes. It defines activities and tasks of systems and software measurement process. A measurement information model and criteria for selecting measures also accompanies the process model definition.

The seven approaches and models described for the first category do not directly aim to determine the maturity or capability of a measurement process. They respectively focus on maturity of measurement technology, usability of metrics for statistical analysis, capability of metrics to satisfy organizational goals, relation between measures and processes, robustness of the measurement program, and defining measurement process. In the following part, two models focusing on measurement maturity are explained.

Measurement-CMM (Niessink & Van Vliet, 1998) is the first method of the second category. It aims to assess the maturity of measurement programs in an organization. It starts with a similar motivation to ours: to assess organizational measurement capability and to provide a roadmap to improve measurement capability. They ask the questions: "How to introduce measurement in a software organization? What are the necessary steps to set up a measurement program and in which order should they be performed? How can existing measurement programs be enhanced?" To answer these questions they suggests a 5 level maturity scale, in parallel with Software CMM (Paulk et al., 1993). Model includes key process areas related to each of the maturity levels. Organizations are expected to implement these processes to reach a level. They suggest to the organizations that want to use this method to adopt Software-CMM.

Second method of the second category is MIS-PyME (M Díaz-Ley et al., 2008) (María Díaz-Ley et al., 2010), which is a software measurement capability maturity model focused on small-medium enterprises. There are two main parts defined in the method: measurement program definition methodology and measurement capability maturity model. This method aligns with ISO 15504 (ISO/IEC, 2006) and aims SMEs. It includes a set of generic goals and indicators, a reference measurement process definition, a maturity assessment process definition and questionnaire, and a tool for supporting maturity assessments (Maria Díaz-Ley, 2009). In the measurement program definition methodology, work products are provided to organizations. These work products are; measurement goals table (i.e. process improvement goals required to implement improvement activities), indicator templates (i.e. guide user defining indicators for each goal), and indicator database (i.e. successfully implemented indicator database). MIS-PyME measurement capability maturity model (MCMM) includes three main components: maturity levels and attributes that need to be fulfilled by measurement processes, assessment process aiming to determine the capability, and an interface with MIS-PyME methodology to define measurement programs.

There are three main conditions defined by MCMM to reach higher maturity; a better established and performed measurement process, more ambitious goals to be measured, and establishing better support tools, procedures, and resources. It utilizes an ISO/IEC 15504 (ISO/IEC 15504-5, 2012) based assessment methodology to determine the maturity. For the assessment, method uses the attributes defined for each level. A questionnaire is used to determine satisfaction level of each attribute. In the questioner, there are set of questions for each attribute. The percentage of positive answers in the questionnaire is calculated to determine the level of achievement.

Measurement CMM and MIS-PyME are the two method in the second category. They aim to understand measurement maturity of the organizations. Even though they define sound models for their purposes, there are some significant gaps in these models according to our focus. Next section provides a brief discussion about the available literature and its gaps that we aim to fill with the proposed method.

2.4 A Brief Discussion of the State of the Art

The methods described in the first group of the previous sections do not directly focus on the maturity or capability of measurement or measurement process. Instead, they evaluate supporting components of measurement or utilization of measures. The two models explained in the second group aim to assess organizations measurement capability.

Measurement CMM suggests adopting software CMM together with measurement CMM. However, traditional process based improvement models draw away organizations that prefer agile approaches over traditional process based approaches. Instead of process-based approaches, these organizations demand practical models to improve specific points in their lifecycles according to their needs. Measurement CMM also does not explicitly provide the methods for assessment and the practices that belong to the key processes.

MIS-PyME uses ISO 15504 (ISO/IEC 15504-5, 2012) as its base to a reference measurement process definition. Aiming small and medium scale enterprises, it expects the organizations to implement defined attributes in their measurement processes. With the inclusion of a process centric evaluation in its core, MIS-PyME fails to satisfy the demands of organizations requiring a flexible approach that can be used by dynamically adapting the needs of organizations utilizing agile approaches.

Considering the available literature, current methods mainly focus to improve measurement processes of the organizations by using common process-centric improvement models as guidelines. It is hard for the organizations that do not want to implement process-based approaches in their development lifecycle to receive the intended benefits of these models. Lack of an up-to-date measurement capability model answering the needs and demands of current development lifecycles creates a gap in the literature. The next chapter includes conducted research to answer this demand.

CHAPTER 3

DEVELOPMENT OF MEASUREMENT CAPABILITY MODEL

This chapter explains the research methodology used during the development of the software measurement capability model.

Conducted design-science research aims to fill a gap in literature by proposing a model for software organizations to evaluate their measurement capabilities. The model is expected to be useful especially for organizations following agile development principles. The research starts with an action research aiming to solve a problem from software industry. During the action research, initial components of the measurement capability model version one is defined. After the action research stage is completed and the initial idea of the model is structured, two exploratory case studies are conducted in organizations with the aim of refining the model. After these two exploratory case studies, version one of the model is implemented in a large organization within the scope of a third exploratory study including three cases. The fourth exploratory study aims to finalize the set of core measures provided in the model. During this study, interviews are conducted with four agile development specialists from different organizations to decide final measure set. With the completion of the measure set, version two of the model is completed. To apply the model and observe its results, an explanatory case study is conducted four organizations utilizing agile approaches. During this study, first the model is applied to the organizations and then an interview is conducted with the representatives of these organizations to collect their opinions about the usability of the model. The overview of the research process is given in Figure 2.

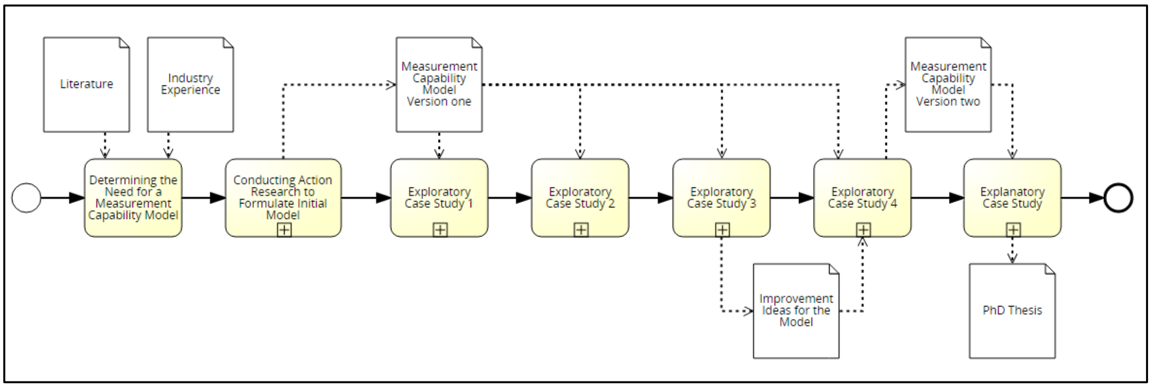

Figure 2 Overview of the research process followed

In this chapter, action research and exploratory case studies are explained. Explanatory case study is explained in chapter 5. First sections of this chapter includes the descriptions of the conducted work in each step of the action research. Later sections of the chapter investigate exploratory case studies in the order of conduct.

3.1 Defining the First Version of the Model

3.1.1 Planning the Action Research

Action research is a successful research technique especially for complex social settings, which are hard to study independent from their context (Baskerville, 1999). The context of action research includes a collaborative environment with actors from the problem domain in addition to the researcher. In this domain, the actors are on the receiving end of the research and the researchers are on the creative end, this is called client-system infrastructure in action research (Figure 3). Research happens in this infrastructure following five steps in a cycle: diagnosing, action planning, action taking, evaluating, and specifying learning (Baskerville, 1999).

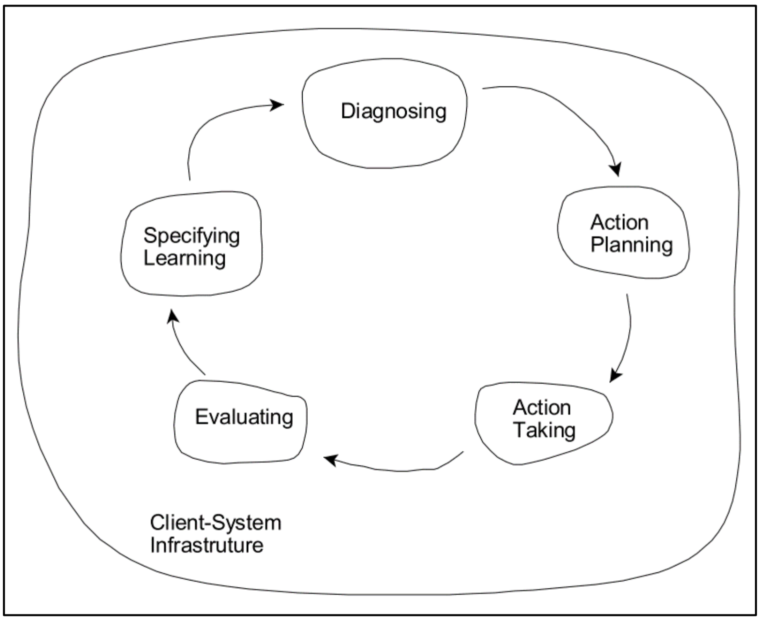

Figure 3 Action Research Cycle (Baskerville, 1999)

3.1.2 Mitigation of Threats to Validity in Action Research

We identified the main threats to the validity of our action research cycle and then created mitigation plans for them.

One of the most important threats for validity in our research setting is that it can be mistaken as consulting if necessary precautions are not taken. If the research orientation is not carefully communicated with the clients, they may expect a consulting performance, which in return may create an ethical problem about the consent of the other party. Even though action research and consulting have similar benefits for the organizations, there are a few properties distinguishing two of them. Firstly, main motivation for the researcher in the action research is the scientific output; as a result, the commitment is not only to the client but also to the scientific output. Secondly, action research requires more collaboration, and unlike consulting where the benefits mainly come from the experience of the consultant, in action research the benefits for the client comes from scientific theory.

During this research, to overcome these issues and be sure about the expectations of the client, the purpose and the method of the research are openly stated to the client at the start of the study. An agreement is settled with the client, which states that we are conducting a research where mutual commitment is required and the output will be based on the success of the hypotheses.

Another threat for the validity of action research is the danger of submerging into the practical problems and neglecting scientific responsibilities. To overcome this threat, at the beginning of the project, the main research questions and propositions are defined and shared with the client. The researchers strictly followed these questions and propositions in order not to lose their way. The research flow is also recorded in line with the research questions.

3.1.3 Conducted Action Research in the Organization

3.1.3.1 Client-System Infrastructure

In this action research project, information-communication technologies (ICT) division of a GSM service provider is the client in the client-system infrastructure. The client supports the research by providing the researcher access to their related processes, assets, products, and key employees and in return aims to benefit from the outputs of the research. Its goal is to improve its measurement capability, especially for their procurement processes. The details of the client's goals are given in diagnosing section.

The researcher supports this research by using literature and client's assets to create solutions to fill the gap in the literature. Main aim of the researcher is to find an answer to the starting research question and to build initial components of a model.

The infrastructure between the client and the researcher connects their goals on a common ground. According to their agreement, the researcher will use the assets of the organization, develop and test a measurement capability model to fill a gap in the current literature. The client will benefit from the interim and final outputs of researcher's work by understanding and improving its measurement capabilities.

The client utilizes a mixed development lifecycle for their software needs. As long as their requirements permit; they follow main agile principles like; valuing individuals and interactions over processes, working software over comprehensive documentation, customer collaboration over contract negotiations, and responding change over following a plan. However, they also utilize vendors to supply some of their software needs. For this software they share analysis documents with the client and share a final user tested software. Because of the nature of these type of business model, comprehensive documents should be shared with the vendors.

The client has trouble to measure the efficiency of its vendors who are supplying software. In their business model, they have two main types of agreements with the vendors; in the first type, they sign contracts for the delivery of specific software products. In the second type, they sign contracts for the development of any required software for a predetermined duration. These types of contracts usually signed annually and they pay to the vendors for the total effort spent during the contract period.

Using effort as a basis for payment is not a reliable method as it is impossible to know whether the amount of effort spent is justifiable. Moreover, it is not possible to compare costs of software from different domains, with different infrastructures, or with different technologies.

They want to be able to measure the outputs of the vendors objectively to analyze their output and to use the output to calculate unit cost of software. They also want the ability to compare productivities of different vendors with each other to determine which ones are more efficient.

3.1.3.2 Diagnosing

As a first step, we diagnose underlying causes of main problem of the client. Their request was to introduce objective criteria to analyze the works of different vendors. To understand their current processes, we conduct extensive meetings. Current procurement process starts with the preparation of an internal analysis document aiming to describe business needs. This document is shared with the vendor and upon receiving this analysis document, the vendor starts design and development activities. If there are points requiring a discussion, they contact to the analyst and clarify the points. After the completion of the development, unit acceptance tests are conducted together with the representatives of the analysis team and vendor. If the software satisfies all requirements, it is accepted.

Vendor is billed for the total effort spent for design, development, and test. The effort is reported by the vendor. This method creates two main problems, first one is that the organization cannot be confident about the required amount of effort to develop a specific software, therefore they have to trust the vendor about the necessity of the reported effort data. Second problem is that the organization cannot objectively compare efficiencies of different vendors. They need an objective measure to represent the work completed by the vendors in addition to effort.

A measure should be defined and used for the size of the software to represent the vendor's work objectively. As the vendors receive an analysis document to start the development, this size metric should be measurable from analysis documents early in the development life cycle. Considering these requirements, using functional size in their processes can solve organization's problem. Functional size can be successfully measured in analysis phase; it is objective, comparable, and applicable for all the software domains of the client.

Among the available functional size measurement methods, COSMIC (Symons & Lesterhuis, 2015) and IFPUG (ISO/IEC 20926, 2009) are the most commonly used standardized options. Considering the expertise of the researcher, the domain in which the client operates, and possible automatization possibilities COSMIC Functional Size Measurement method is chosen as the most suitable option for the situation.

After deciding the measure, it needs to be integrated into the processes of the organization. For the integration plan, it is crucial to understand current process artifacts of the client and their compatibility with functional size measurement. To be able to help the client to analyze and improve its overall measurement infrastructure, the researcher needs to find a method. The literature review shows that currently there is not a measurement capability model, which can help this organization, as they do not want to follow a standard process centric assessment or improvement approach. They want to focus just the implementation of this specific measure. They require assessing and improving a focused field in their development lifecycle. To help them analyze and improve a niche field, we need to create a specialized approach. To identify the main components of this approach to be applied in the organization we start with the first research question of this study:

RQ-1. What are the main components to be used in a model to analyze software measurement infrastructure and identify improvement opportunities?

Diagnosing step is completed after the problem is diagnosed and the research question is shared with the client. Next step explains necessary actions to answer the research question.

3.1.3.3 Action Planning

The planning task aims to plan required actions to be applied on the client organization to find an answer to the research question.

RQ-1. What are the main components to be used in a model, to analyze software measurement infrastructure and identify improvement opportunities?

In the client organization, our aim is to implement a size measure to improve their procurement processes; however, to be able to introduce the capability of size measurement, current measurement capabilities of the organization need to be assessed. By knowing current capabilities, it would be possible to define a roadmap to implement the new measure, as we should understand mutual interactions and effects between current and new processes and measures. A starting point to understand measurement capability is to know what the organization currently measures. Therefore, the first action item (AI) to conduct is:

AI-1.1. Identify current measures in the organization

After current measures of the organization are identified, we need to learn how these measures are used. This will help us to understand: first, the interaction between the measures and organizational processes; second, utilization of collected measures. Utilization of the measures can express the organization's capability of using these measures in their activities. This information can help the organization to identify where to improve to gain most benefit from the mew measure. To understand how these measures are used, we need to identify the practices related to the current and possible future measures in the organization.

AI-1.2. Identify practices related to the collected measures in the organization

First two action items are expected to provide an idea about the measurement capability of the organization. Next, we want to understand from where these measures are collected. Identified measures and the artifacts from which the measures are conducted should be examined. The examination can provide information about the quality of the artifacts and quality of the measures. It can also exhibit whether the artifacts include enough data to measure functional software size. If there are some gaps in the artifacts, these need to be identified as improvement points. We planned the third action item as:

AI-1.3. Analyze current measurement artifacts to understand their quality and suitability for new measures and identify improvement points

After the artifacts are analyzed and their quality is understood, if necessary they need to be improved according to measurement needs. In the client organization, as they aim to implement a functional size measurement, functional user requirements and functional processes need to be easily identified from the artifacts.

In addition to implementing the ability of size measurement, we require to collect historical size data from past projects. Vendors' efficiency needs to be compared with organizational efficiency figures and these figures should be calculated from completed projects. Similar requirements may be valid for different measures as well, when an organization is introducing a new measure, historical data related to this specific measure can be used to conduct an initial analysis to increase the benefit gained from them. Collection of historical data requires measurements from the past project artifacts. While conducting action item 1.3, researcher can also use proper artifacts to measure and collect past data. Actually measuring the artifacts also provides clear information about their sufficiency for measurement.

AI-1.3.1. Use artifacts that include sufficient data to collect historical data for the new measures

Action items identified up until this point are required to analyze and improve measurement capabilities. If successfully applied, they can provide an answer for the research question. Even though it is not a part of the research question, these improvements need to be implemented in the organization. It is expected by the client and it is also necessary for the researcher to observe the effectiveness of the model. Therefore, after the improvement opportunities are identified and approved by the client, improvement points can be applied and tested with a pilot group to observe their results and effectiveness.

AI-1.4. Implement and observe improvement points with a pilot group

After the pilot implementation, the results should be evaluated. If there are problems, another action cycle can be planned and conducted to identify and overcome problems. If the results are positive, then new processes can be applied organization wide.

After the action items are planned, action-taking phase begins. Conducted actions in this phase are given in the next section with their results.

3.1.3.4 Action Taking

After the action items are planned in previous step, organization and researcher approved to start of the implementation phase. There are 4 action items defined in the planning phase as listed below:

- AI-1.1. Identify current measures in the organization
- AI-1.2. Identify practices related to the collected measures in the organization
- AI-1.3. Analyze current measurement artifacts to understand their quality and suitability for new measures and identify improvement points
  - AI-1.3.1. Use artifacts that include sufficient data to collect historical data for the new measures
- AI-1.4. Implement and observe improvement points with a pilot group

The actions and their immediate results are explained in their own sections.

Identify current measures in the organization

Identification of the measures collected and used in the organization is first step to understand their measurement capabilities. To extract a list of measures used in all of the phases of their software development lifecycle, we used two main approaches: meetings and document reviews. Meetings with the employees participating in different phases of the lifecycle are the main source of information. Analysts, developers, testers, quality specialists, team leaders and managers attended in many meetings conducted during this phase. Main information gathered from them is; what measures they are collecting for the activities that they are conducting, how they collect them, and how the measures are used. Then this information is crosschecked by examining process documents, which include process definition documents, process outputs, and organizational measurement databases.

The resulting list of measures used during the lifecycle can be listed mainly as effort, duration, cost, risks, changes, and defects. For most of these measures, the organization makes an estimate before the project start date and collects an actual value after the measurement conducted. They also calculate their estimation efficiency and use it as a derived measure to understand their estimation accuracy.

After the identification of the measures, we wanted to compare them with a reference measure set to understand whether the organization is currently collecting at least a bare minimum set of measures. As given in the literature section, current available methodologies aim to help organizations to evaluate their measurement capabilities by using a process centric approach. The client organization does not wish to follow a comprehensive approach as they have confidence in their own way of doing business. Therefore, they require a flexible approach to be used wherever they wish to improve. We build a list of core measures to be used as a benchmark set as a part of this action item. This list can let the organization focus any area they want to evaluate or improve. The core measures included in the list are a result of combination of knowledge gained from the researchers' experience from both academic and industrial works and the client's experience in the industry. The list is given in Table 1. Identified core measures are also grouped according to their time and place in the development lifecycle.

Table 1 List of core measures

Measure Group	Core Measure
Project	Planned effort
	Actual effort
	Percentage of effort estimation efficiency
	Percentage of actual effort (at a specific time)
	Cost estimation
	Duration estimation
	Actual duration
	Percentage of duration estimation efficiency
	Percentage of actual duration (at a specific time)
	Planned cost
	Actual Cost
	Percentage of cost estimation efficiency
	Percentage of actual cost (at a specific time)
Risk	Number of anticipated risks
	Number of occurred risk
	Percentage of risk identification efficiency
	Percentage of undefined risk efficiency
Quality	Number of non-conformance
	Costs of corrective actions
	Cost of preventive actions
Configuration	Configuration changes
	Configuration change rate
Change	Number of proposed change
	Number of accepted change
	Cost of change

Measure Group	Core Measure
	Change density
Procurement	Procurement contract changes
	Quality of supplied work product
	Supplier productivity
Requirement	Number of requirement change
	Cost of Requirement change
	Requirement change density
Solution	Number of design change
	Cost of design change
	Design change density
Test	Number of defects
	Defect density
	Internal failure cost
	External failure cost
Integration	Integration errors
	Integration error density
Training	Training quality
Process improvement	Number of process improvement proposals
	Cost of quality
Size	Size
	Productivity

The client collects most of these core measures. Missing measures are related to the lack of a size measure, which is also identified early in the action research. The missing measures are; software size, productivity and measures that are related to density, like defect density. As the client does not have any measure representing the size of the software objectively, they cannot calculate the density of any other measure, which prevents them to compare the results of their measurement between projects, between teams and with the measures from other organizations. They lack the ability to compare because of the lack of a size measure. This ability of comparison is an important point that needs to be reflected in the model.

After the identification of the measures collected in the organization and comparing it with a list of core measures, we continued with the next action item, which is explained in following section.

Identify practices related to the collected measures in the organization

Identifying the measures and determining a set of required core measures is an important step; however, without knowing how these measures are used in the organization we cannot evaluate them. We need to know how the client collects, uses, evaluates, and benefits from these measures. Similar to the previous step, in addition to identifying the practices of the client, we also need to determine a set of base practices for an organization to be classified as capable.

Practices defined in available measurement capability models are not directly applicable for the client organization. To analyze measurement capability for an organization, we defined three main levels of practices together with the client organization. The reason for a three level evaluation is the aim of building a practical and agile approach for organizations and it needs to be as simple as possible. Three levels should provide enough perspective to organizations to understand their positions.

First level should represent the most basic level for the collected measures. In this level, the measures are collected but their usage and storage are mostly limited for project or team usage. If the measures are collected, stored, and used in organizational level, then the organization's measurement capability is expected to be mediocre. To classify an organization as highly capable, the organization needs to have the ability to use the measures as quantitative control parameters to identify and apply improvement opportunities for their processes.

Measures of the client organization are evaluated to classify them according to the three level approach. During the evaluation, several practices are determined to be able to classify the measures objectively. The practices defined for each level are given in Table 2

Table 2 Practices related to capability levels

Capability Level	**Practices**
Level 1	Identify measures
	Collect and store measures
	Analyze measures
	Communicate measurement to relevant stakeholders
Level 2	Plan and perform measurements according to a policy
	Use measurement and estimation methods suitable for organizational needs
	Define required sources and make them available to perform measurements
	Assign responsibility to perform the measurements
	Control products of the measurements
	Identify the relevant stakeholders of the measurements
	Monitor and control the measurements against the plan for performing measurements and take appropriate action
	Evaluate adherence of the measurements against defined measurement descriptions.
	Collect and store measurement related experience to support the future use
Level 3	Determine factors effecting measurement
	Use organizational tailored estimation models
	Use control charts to evaluate measurement activities
	Use statistical evaluation to improve measures

The analysis conducted in the organization by using the defined practices shows that there are generally accepted definitions for the collected measures across the organization. Participants are collecting measures as expected from them, and organizational audits inspect the collection of the measures. However, missing core measures, identified in the previous step, reduces the overall measurement capability of the organization.

The organization effectively collects actualized data for the projects and makes estimation at the beginning of the projects. However, organization does not have any tailored estimation models; therefore, estimations are based on expert judgement. As a result, meaningful comparisons of actual and planned measures for different projects or teams are not possible. Organization also fails other level 3 practices for almost all their measures.

Conclusion of this action item is that the organization has awareness about the importance of the measurements, they define and collect measures in organizational level. If the size measure is implemented and used to build estimation models, the general capability level can easily rise. Another important benefit of a size measurement implementation is the ability to measure the density of current measures. This capability can improve not only the procurement process but also other organizational processes.

After the identification of the practices, the next action item is the analysis of the measurement artifacts, which explained in the next section.

Analyze current measurement artifacts to understand their quality and suitability for new measures and identify improvement points

After the current measurement capabilities of the organization are analyzed and measures that can be improved are identified, analysis of the measurement artifacts is started. First measure to be implemented is the size of the software. The organization decides to use functional software size as it is possible to measure it at the beginning of development lifecycle. Among the common functional size measurement methods, the most up-to-date and suitable method for the organization's domain is COSMIC (ISO/IEC 19761, 2011). Therefore, the researcher and the client agree on implementing COSMIC Functional Size Measurement as the size measure. Functional size will be measured from the analysis documents, which is used by the organization to describe software requirements to internal and external developers.

In the client organization there are eight main development teams producing software for different domains with various technologies. To be able to cover all types of software, sample analysis documents are requested from the client. The sample is expected to include documents from all groups and teams. Organization divides projects into two according to the duration of the projects. Projects that do not take more than three days to complete are called as fast tracks (FT). The limit of three days does not include the time spent for the analysis. The ones taking longer than three days are called projects. These two types of projects should also be represented in the sample document package.

The client provides analysis documents of 530 projects, 442 of which are fast tracks, 88 of which are regular projects. This distribution is determined by the client by considering general distribution of FTs and projects. After documents are received, they are reviewed by expert COSMIC measurers to evaluate their fitness for COSMIC functional size measurement. From the 530 projects, 231 found as unfit for measurement and classified as level C, 62 found as fit and classified as level A, and remaining 237 are classified as possible to measure but require some improvements and named as level B.

Projects in level A and B are measured during their evaluation and their data is used to conduct analyses as explained in the next section. Projects in level C and level B help us to understand the required improvements on the analysis documents to make all documents fit for functional size measurement.

First outcome from the evaluation is that the documents are written in different level of details. Some of them include high-level description of the requirements whereas others describe the requirements in low-levels. Some samples even include detailed software design and code. Second outcome is related to the different application services used in the organization that are affected from the developed software. Most of the new software requests require several application services form the organizational inventory to communicate with each other to satisfy the requirements. Any software put on one of the service may have cascading effects on other services, which need to be known to be able to measure total size of the software correctly. However, old analysis documents do not include detailed information about the effects of the proposed software on all of the services.

The first improvement point determined for the analysis documents is adhering to a predetermined level of detail while describing the requirements. COSMIC already defines a level of decomposition to be able to measure functional size correctly. This decomposition should include enough detail to help the measurer identify functional processes. The analysis documents should be written as use cases where functional processes can easily be determined. A template analysis document is created where the analyst can enter as many use cases as necessary, and while writing use cases, analyst can select related data movements for each step in the use case and determine the number of related objects of interests. This document lets the analyst to see total size of the software as soon as all use cases are recorded into the document.

Second improvement point is including a use case diagram into the analysis document before each use case. By the help of these diagrams, data movements can easily be visualized by the analyst to make it easier to understand each step in the use case. This diagram also makes it easier for all parties who will be using this document to understand the requirements. Understanding the requirements means less discussion between the developer and analyst in future phases.

Third improvement point is the inclusion of affected application services in the analysis document. COSMIC dictates that a measured software can only work in one peer, meaning that a software effecting two different application services needs to be divided into two and each part on each application service should be counted as an individual software. If the analyst lists all of the effected application services in the analysis document, correct size of the software can be identified by counting each module in each service separately. Specifying all effected services also helps the organization to trace the size changes in each service. As different services utilizing different technologies may have different unit cost, this information is invaluable for the organization while building their cost models.

After the improvement opportunities are identified, they are applied with a pilot group in the organization. This pilot study is explained in the last action item. Next section gives the details about measurements conducted during the examination of the documents that are classified as level A, and B.

Use artifacts that include sufficient data to collect historical data for the new measures

During the evaluation of the analysis documents by expert measurers, documents classified as level A and B are measured to calculate functional size of them to build a historical data set. The findings are then reported to the organization and the organization provides spent effort values for these projects. In addition to building a historical size set, using software size and spent effort, several statistical analyses are conducted.

We start the analysis with the average sizes of fast tracks and projects. The organization used to use estimated development duration to determine whether the projects should be classified as a fast track or not. By using functional sizes of the projects, we can calculate the average size for past fast tracks. Using the size of the software provides the ability to classify projects with an objective measure instead of a duration estimation. We also calculate an average for each development team. The technology and domain of the teams can differ from each other; therefore, different criteria should be used for each team. Average sizes for fast tracks for each team are represented in the bar graph in Figure 4. The values are concealed because of the confidentiality agreement with the organization.

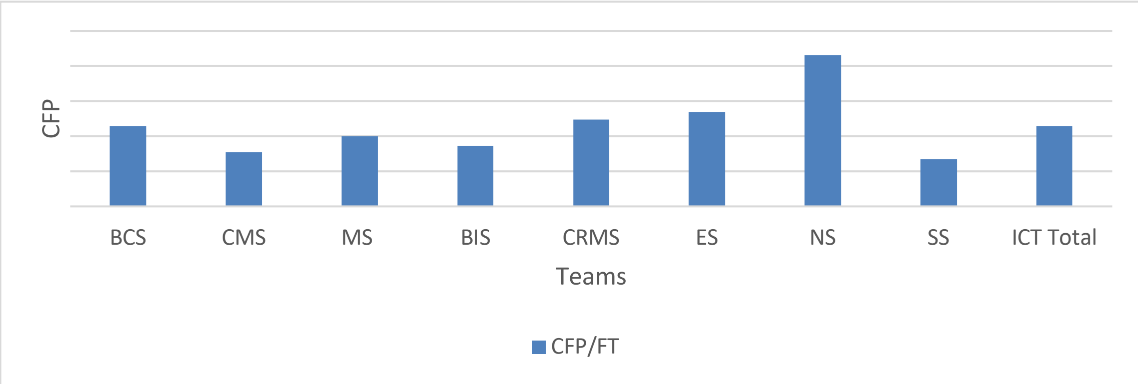

Figure 4 CFP/FT Distribution among teams

As seen in Figure 4, even though the fast track is defined as projects taking less than three days to complete, average sizes for each team are different. The reason is that all teams are working with different productivity rates, as they work with different technologies on different domains. To see the differences between their productivity rates we calculate required effort to produce one COSMIC Function Point (CFP) for each team. The results can be seen in the bar graph given in Figure 5 with the values removed. The red line on the graph shows an international sector productivity average calculated form ISBSG 2009 data set ("ISBSG Dataset Release 9," 2009). This rate is calculated by using software sizes and spent efforts from telecommunications, finance, banking, and information technology services projects in ISBSG data set. All selected projects are smaller than 100 CFP.

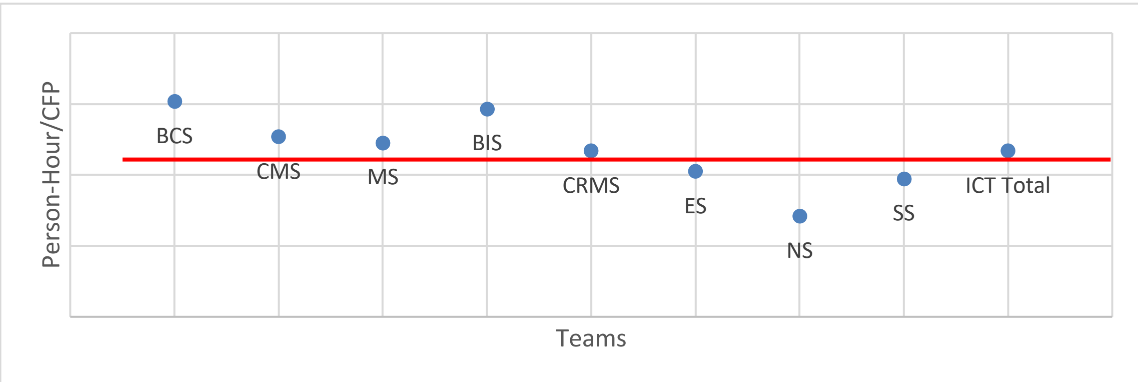

Figure 5 Productivity (Person-Hour/CFP) of fast tracks among teams

As expected, all teams have different average effort to produce 1 CFP. With the information provided in Figure 5, organization can observe average production rate for each team, average production rate of the ICT department, and an international benchmark value to compare their performance. It is possible to compare these values as they are all based on an objective size measure of the software.

Similar values are also calculated for projects and similar graphs are drawn. Average sizes for the projects are given in Figure 6 and productivity rates are given in Figure 7. ISBSG average for the projects larger than 100 CFP is also drawn in that graphs. In these figures, there are no values for the teams NS and SS, as they do not have any large projects in the measured set.

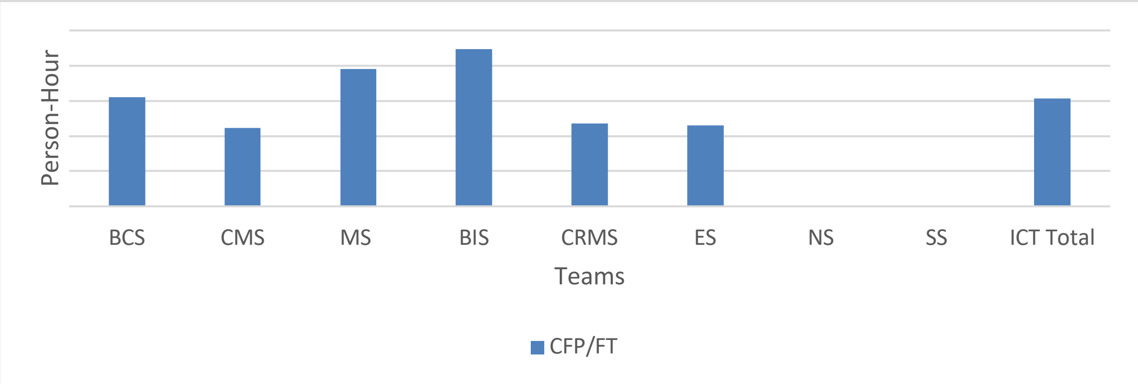

Figure 6 CFP/Project Distribution among teams

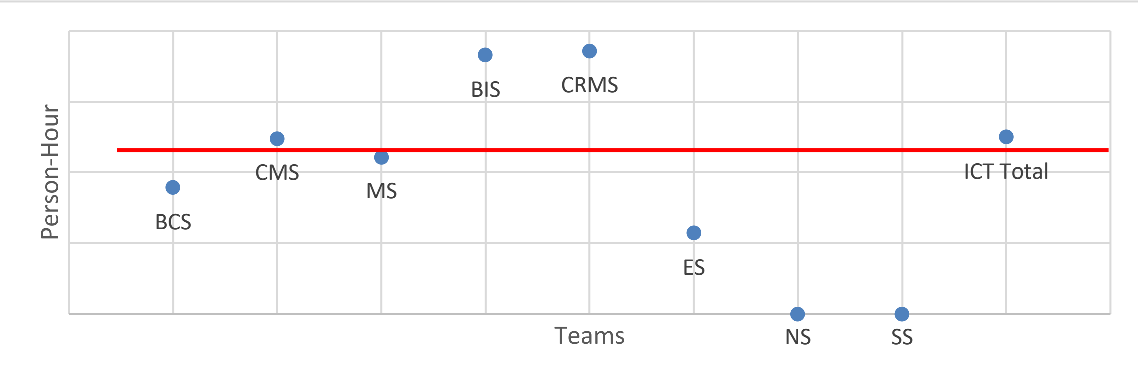

Figure 7 Productivity (person-hour/CFP) of projects among teams

Calculation of the average sizes for the project types is helpful, but we need to determine an optimal size to use to decide project type. To find out whether there is an optimal size for the projects we plot productivity rates for each project against project sizes. With the help of these plots drown for each team; we aim to determine an economy area. A sample graph is given in Figure 8, which shows the economy area for BIS team's fast track projects.

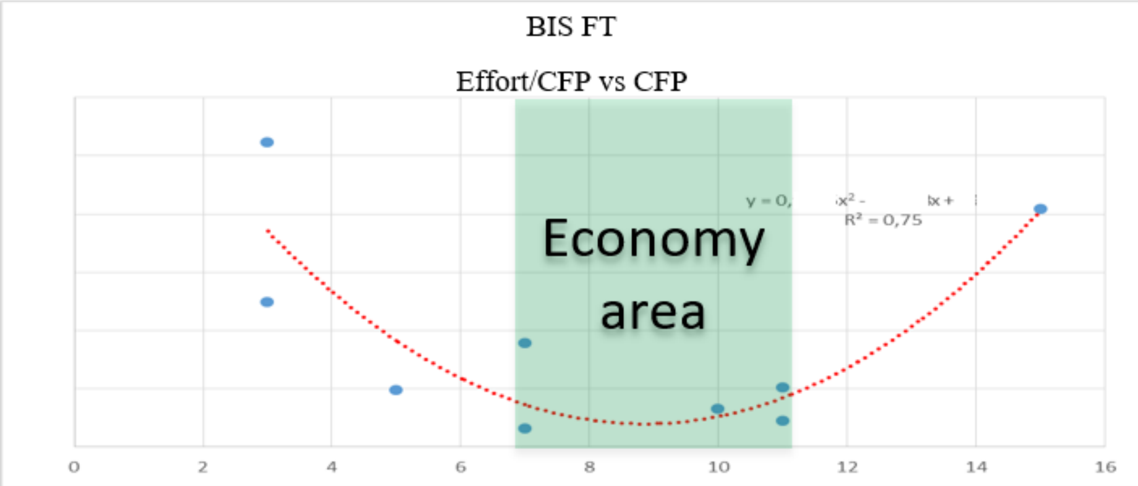

Figure 8 Effort/CFP vs CFP for a sample team

Economy areas represent the range of project size where effort is the lowest for unit CFP. If the organization can keep its projects' sizes in this area, they will have lowest effort per CFP. To be able to do that, they can combine several smaller projects into one large project or they can divide larger projects into several smaller projects to make them fall into this range.

As a part of the analysis activity, we create effort models for the projects. Main purpose of the organization is to implement a unit cost based contract with their vendors. To perform this successfully they need to know possible unit cost of production. We use size measurements from the past projects and their effort values to create an effort estimation model for each team, both for projects and fast tracks. One of the models is given in Figure 9 with its graph for FT's of BCS team. These models will be used as baseline while determining efforts and costs of future projects to negotiate with vendors. The models need to be updated periodically to respond changes in the organizational processes and improvements. Therefore, the organization will update all their estimation models in certain periods by including data from new projects.

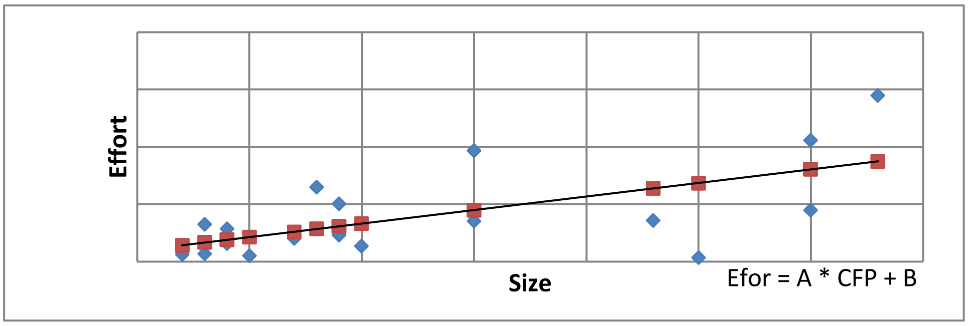

Figure 9 Effort model for BCS FT's

Base analyses demonstrate potential benefits of having size measure implemented into the processes. To reach the potential benefits of this improvement initiative, the organization needs to integrate this new measure into its processes. Before a full integration, we want to observe a pilot implementation to understand the effects of the changes on the employees.

Implement and observe improvement points with a pilot group

After the identification of the improvements and evaluation of the possible benefits from size measurement, the improvements are implemented with a pilot team. The purpose of the pilot application is to observe responses from the employees and to understand potential resistance to change. If the pilot team can adopt and implement changes with ease and without resistance, new processes can be brought into effect in the whole organization.

Identified changes for the analysis document are implemented and put into use of the pilot team. First, the changes on the analysis documents are implemented on a document template to be used by the team. Then the method to collect and store the new measures is determined. Before putting these changes into implementation, a comprehensive COSMIC training is provided to the analysts and developers in the organization. Training also covers the changes in the organization and expected benefits of using functional size. In the pilot phase, around 60 people are trained. After the trainings, additional workshops are conducted with the participants to conduct sample measurements and to answer their questions about the application. Workshops and trainings continue during the project life as demand arises from the participants.

New analysis document template includes a use cases section where the analyst writes the use case step-by-step. Analyst determines the size of related data movement for each step. At the end, functional size of the software is equal to the aggregation of these movements. After completion, analysis documents are sent to vendor's developer for detailed design. During the design phase, vendor should measure the size of the software again and if they doubt the measurements of the analyst. Developers can update the size if there are changes on the requirements after the analysis phase. Implementation and user acceptance tests are conducted after developers complete design and coding. After user acceptance tests, software will be measured again by the test team to understand possible changes during implementation. The three measurements collected during analysis, design, and user acceptance test are recorded. As long as there are not any changes in the software requirements, measured size in these three phases should be the same. If there are differences in these measures, the projects will be audited by analysts and developers to understand the reason of the difference.

Organization collects effort data for the projects with an organizational project planning software. This software remains as the effort collection medium and as an addition, a new field is implemented in the system to collect and store functional size of the software. Size values from all three phases are registered and stored for each project. This development enables organization to report and analyze size and effort data easily. It also enables them to update effort estimation models.

After the new analysis document template put into use and the infrastructure to collect project sizes is completed, analysts start to measure the software. At this point, the researcher and the organization agree on putting a control mechanism to audit the correctness of the measures. Even though all employees receive training, they are inexperienced and their first measurements expected to include minor errors. In addition, participants of pilot group may resist or protest the new method and specify the sizes without proper application of the rules. To prevent these issues, an audit process is planned. The organization selects sample projects to be audited. This sample covers at least 10% of the total size of software for the period and includes at least one document from each analyst preparing an analysis document. Selected samples are audited by the researcher to note the errors.

Results of the audits are shared with the participants who measure them. Participants are asked to review the audits and object if they observe a problem. This objection process is designed to eliminate any possible misunderstanding. When the researcher and the analyst agree on the final size, the result of the audit is reported to senior management. To analyze the effectiveness of the measures and audits, two metrics are determined and included in the report: measurement quality and commercial reliability. Measurement quality aims to represent how well the analyst applies the measurement rules. It is calculated by using total number of misidentified and forgotten data movements. Number of total errors is divided to real size of the software to calculate error rate. Measurement quality rate is then calculated by distracting the error rate from 100%. If the real size of the software is 100 CFP and the measurer measured it as 100 CFP by omitting 5 data movements and including an additional 5 data movement, error rate should be 10% and measurement quality rate should be 90%.

Commercial reliability rate shows the deviation between the measured size and the real size. It is called commercial reliability because the organization aims to pay its vendors according to unit size and this measure shows how close the measured size is to the real size. For the example given above, commercial reliability should be 100% as the measurer measured the software as 100 CFP, same as the real size.

The organization uses measurement quality metric to evaluate how well the participants can measure, whereas commercial reliability metric is used to understand how accurate the payments to the vendors are. For the pilot application, the results of the audits reveal the measurement quality metric around 80% and commercial reliability metric around 90%. Client evaluates these values as satisfactory for the pilot application and approves dissemination of the improvements across organization. Organizations aim to increase measurement quality and commercial reliability metrics to 90% and 95% respectively after 1 year of organization-wide application.

After the implementation with the pilot team, all planned actions are completed for the action-taking step of the action research. In the next part, evaluations of the taken actions are explained.

3.1.3.5 Evaluating

After all the action items are completed in the client organization, researcher and the client evaluate the actions and their results. Evaluation results for each individual action item are given below. Action items are built to find an answer to the research question and the evaluation of the research question is given in specifying learning section.

AI-1.1. Identify current measures in the organization

Item is successfully completed by listing the measures used in the organization. In addition, we identify a set of core measures to be used as a benchmark set for future evaluations. This set is one of the main components of the measurement capability model version one.

Researcher reaches to his goal by formulating first input to build a measurement capability model and by using it in the organization. Client reaches his goal by identifying current measures and by learning what measures it can introduce to improve its measurement structure.

AI-1.2. Identify practices related to the collected measures in the organization

Item is completed by identifying a set of practices to evaluate utilization of the measures in the organization. The practices are used to evaluate client organization's current measures and to identify improvement opportunities to increase the benefit gained from the measures.

Researcher identifies a preliminary set of practices. These practices can be used as a component for the measurement capability model version one. Researcher also tests these practices in the organization. Client gets its measures evaluated with the set of practices and learned its capabilities for each measurement. Improvements to increase the capabilities are also deducted from the determined practices. This action item is evaluated as successful by both the researcher and the client.

AI-1.3. Analyze current measurement artifacts to understand their quality and suitability for new measures and identify improvement points

Analyses of the artifacts are successfully completed by producing helpful outputs about how to improve them for a size measurement implementation in the organization. Client's goal of creating improvement ideas for the artifacts is successfully reached. Researcher has similar goal with the client, improving artifacts to improve measurement infrastructure successfully.

AI-1.3.1. Use artifacts that include sufficient data to collect historical data for the new measures

This item is part of the previous action item. During evaluation of the artifacts, researcher collects historical data. In addition to collecting size measures, basic statistical analyses are also conducted in this step to show the potential benefits of having accurate size measures.

Researcher reached his goal by conducting analyses and proving to the client the benefits of the measurement evaluation and improvement activities. Client reached its goal by observing the benefits of the work conducted up until this point.

AI-1.4. Implement and observe improvement points with a pilot group

Last action item is about the application of all the findings of the previous action items with a pilot group. New improvements are successfully implemented into the daily work routines of the pilot group. There is not any active resistance against the changes, and all participants actively engage during trainings. Improved documents are used and size measurements are collected. An audit process is implemented to evaluate measurements of the participants and audits result with acceptable success rate.

The organization reaches its aim of analysis and improvement of their measurement activities; they identify and apply necessary improvements with a pilot group. The researcher reaches his aim by observing successful application of proposed improvement opportunities that are identified by using main components of the measurement capability model version one.

Overall, actions taken during this step are evaluated as success by both the researcher and the client. Observing the success of the improvements in the pilot group, client decides to disseminate the changes in all teams. The researcher plans to test the measurement capability model version one with an exploratory case study and test the model with other organizations.

Dissemination of the improvements across the organization is started after the completion of the action research. Even though the researcher is involved during that phase by training all employees of the organization for size measurements and by auditing collected measures, dissemination process is not explained as a part of this research. In the first exploratory study, measurement infrastructure of the client before and after this dissemination process are evaluated with the model version one to test the method.

Next part summarizes the outputs of the action research cycle by specifying the learnings.

3.1.3.6 Specifying Learning

This action research cycle helps the organization to overcome their main problem by improving their processes and including new measures. They learn the necessary skills to analyze their measures, identify improvement opportunities, and conduct the new measures. They decide to follow this path to include these skills into their organizational capability pool by the means of trainings and workshops. They continuously monitor and improve their measures (Salmanoğlu, Öztürk, Bağrıyanık, Ungan, & Demirörs, 2015).

This action research helps the researcher to formulate main components for a measurement capability model version one to fill a gap in the literature. Available methods aiming to assess measurement capabilities do so by putting a process-based assessment approach in their core. The client does not want to change or assess their whole process structures, its aim is to use a practical approach to evaluate their measures and suggest improvements. Contemporary software organizations have similar opinions; they apply a mix of different development life-cycle models and require a practical approach to solve their measurement related problems. This action research cycle showed that such an approach might be possible.

For the researcher, this action research cycle provides two main components: a set of core measures and a set of practices related to the capability levels. Software engineering field includes numerous measure suggestions for the development process, for the development team, and for the software itself. Organizations may have difficulties to determine what to measure for their specific needs. A list of core measures distributed among different work areas can help them to evaluate themselves and understand what they need to improve. Second output, the list of practices, can help organizations to evaluate how efficiently they are using the measures and how they can increase their efficiency.

These outputs provide an answer to our first research question:

> RQ-1. What are the main components to be used in a model, to analyze software measurement infrastructure and identify improvement opportunities?
>
> Answer-1: *A list of core measures and a list of measurement practices can be used as main components of a measurement capability model. Organizations can use list of core measures distributed among different work areas to evaluate the measures they are using and to find out what they need to implement as a core measure. With the help of a set of practices divided into capability levels, organizations can evaluate their capabilities and create an improvement plan to increase their capability levels, which in return may help them to gain more benefits from the measures.*

This action research helps the researcher to formulate initial components for a measurement capability model; however, to be able to know the effectiveness of these ideas, further tests are required. In the next section, the first exploratory study that is conducted in the same client organization is given.

3.2 Exploratory Study 1 – Exploration of Initial Ideas

After the completion of the action research, client organization continue with the implementation of the improvements in the organization. Researcher also involves in the dissemination activities by providing training and audits independent from the research activity.

Nearly one year after the initial pilot study, organization completes dissemination activities and starts using the new measure actively in all teams. Researcher conducts an exploratory study with the organization to apply set of core measures and practices identified in the action research to assess organizations current measurement capability and to compare the results with the beginning of the action research.

3.2.1 Research Question of the Exploratory Study 1

At the beginning of the study, we define a research question and create a proposition aiming to answer that research question. We also structure a validation method to test the proposition, which is the starting point of the exploratory study.

RQ-2. How well do the identified main components work to evaluate measures in software development process?

Proposition 2: Core measure set and capability practices defined in the action research can be used as a measurement capability model. If this model works well, it can assess an organization measurement capability and can reflect conducted measurement related improvements.

Validation method for proposition 2: The model version one can be applied with the data collected in the organization before and after their improvement initiative. If the model has the ability to reflect organization's capability, the results should reflect the improvements that the organization implemented during the action research.

The study starts with planning the activities and then continues with the selection of the case. Case selection step also includes identification of information sources and environment. After the selection is complete, we conduct the case and analyze the results. The steps are explained in the following sections.

3.2.2 Activity Planning in the Exploratory Study 1

For the first exploratory case study, following activities are planned:

- Case selection: The case to be used the first exploratory case study expected to satisfy four criteria. First, the case should represent real-life context. Meaning that the case should be conducted with a real organization using its real software development process information. Second, there should be an improvement initiative in the organization, so that before and after this initiative can be compared. Third, process information and available measures should be available to the researcher. Final criterion is that the case should include all steps of development to observe all possible measures during a software development lifecycle.

- Establishing case study environment: If the selected case requires the researcher to work closely with the organization to collect data, the study may be conducted in the organization's premises. If the data is already available, the location and infrastructure is not relevant.

- Identifying information sources: Main information source for the study is the organization. Depending on the selected case, method for collecting information sources need to be defined.

- Applying Measurement Capability Model: The model version one should be used with the data collected from the information sources.

- Analyzing the conduct of the case studies: Findings of the case study will be inspected and the results will be evaluated to discuss the validity of the proposition 2 and respectively the answer for the research question 2.

3.2.3 Mitigation of Threats to Validity in the Exploratory Study 1

Threats are categorized as internal validity, construct validity, external validity, and reliability. These threats are discussed below with the help of a checklist from Wohlin et al. (Wohlin et al., 2012).

Threats to internal validity: For this study, main threat to the internal validity is possible positive inclination of the participant of the study. This study is carried out by the researcher by using retrospective data from the organization. As the researcher is also the participant in the study, inclination towards a positive result may be possible. Even though the researcher is following strict guidelines, to mitigate this threat a second case study is also planned. The second study will be conducted by a third party in his own organization.

External validity: Representation of population is an important threat to external validity of this study. As there is only one organization participating in this study, it is not possible to generalize the results. Therefore, several other case studies are also planned with different organizations to observe the effects of the model in different settings and samples. This study gives a picture in one organization to provide enough evidence to continue with the following cases.

Construct validity: There might be a mono-method bias in this study, as it only includes a single type of observation. There are two points related to the mitigation of this threat: first, the researcher is familiar with the subject organization and the model applied in the research. Therefore, he has a good basis of evaluation. Second and more important point is that there are other cases conducted in the research life cycle conducted with different methods of evaluation.

Reliability: This exploratory study is conducted by the researcher himself; therefore, operational judgment is an important threat to the validity. To mitigate this threat, after the completion of the study, the conduct and results of the study are evaluated by other researchers and industry participants (Salmanoğlu et al., 2018). In addition, during the research there are other exploratory studies including other researchers in different organizations.

3.2.4 Design and Execution of the Exploratory Study 1

Case Selection in the Exploratory Study 1: The aim of the first exploratory case study is to understand whether measurement capability of an organization can be successfully identified by using the proposed set of basic measures and practices. Therefore, while deciding the case to be used in the study the researcher considers four criteria; first criterion is that the case should belong to a real organization conducting development activities. Second, there should be a previous improvement initiative related to measurement capability. Third, this organization should voluntarily share its data related to the measurement activities. Final criterion is that the all steps of the development should be included.

The organization where the action research is conducted satisfies all selection criteria. As the organization is already evaluated during the identification of the core measures and practices, the researcher also plans to apply the model version one to the organization after the organization wide implementation for the measurement improvements. The researcher can compare the findings of before and after evaluations with the results of the action research. This comparison will provide data to observe the effectiveness of the model in assessing the change in the organization's measurement capability. As this organization does not utilize a fully agile approach but rather a mixed software development lifecycle, further test with other organizations may be required after this case study.

For the first exploratory study, two cases are used, the case including the beginning situation of the organization, before case, and the one including the measurement infrastructure of the organization after the improvements, after case.

Case Study Environment in the Exploratory Study 1: Exploratory study is conducted with the data gathered from the client organization in the action research cycle. The researcher works with the organization through the improvement initiative; therefore, he has the necessary data from the organization, for both before and after the initiative. He also has the possibility to retrieve any additional data from the organization upon request. The environment where the research is conducted does not necessarily have to be the organizations premises. Researcher can conduct the evaluation in a personal computer by using the data collected from the organization.

Information Sources in the Exploratory Study 1: Necessary information required for the conduct of the case study is acquired from organization. Researcher has the knowledge of the organization's measurement practices as he works with them during their measurement improvement initiative. Knowledge gained that time is used retrospectively to apply the model with the before and after cases. If a need for new data or information is arises, the researcher has direct connections to the organization and key personnel.

Application of the Model in the Exploratory Study 1: Exploratory study includes two cases, before and after the organization implemented measurement improvements. Application of the model is conducted retrospectively with the organizational measurement data and measurement capability level is determined for each core measure.

To apply the model criteria for each measurement practice are determined to evaluate each core measure. The criteria for the practices are given in Table 3. Practices are listed in a progressive approach, in the first level they require the measures to be defined and conducted but it may in project or team level. In the second level, there need to be guidelines about what to measure, how to measure, whom to distribute. In the third level, organization need to follow the results of the measure to understand its progress.

Table 3 Criteria for each practice for the model version one

Practice	Criteria
Identify measures	The measure should be identified, responsible body should know what to measure
Collect and store measures	The measure should be collected and stored, storage may be in project or team level
Analyze measures	The measure should be used in an analysis
Communicate measurement to relevant stakeholders	Collected measure and result of analysis should be shared with any relevant stakeholder. The stakeholder may be limited with project or team personnel
Plan and perform measurements according to a policy	Measurement should be planned and conducted according to an organizational policy
Use measurement and estimation methods suitable for organizational needs	Measurement or estimation should be used in organizational level
Define required sources and make them available to perform measurements	Organization should identified where and how to conduct the measure
Assign responsibility to perform the measurements	Organization should define the responsible body for the measurement
Control products of the measurements	Collection of the measurement should be controlled in organizational level
Identify the relevant stakeholders of the measurements	Relevant stakeholders, with whom the measures should be shared, should be identified by the organization
Monitor and control the measurements against the plan for performing measurements and take appropriate action	Organization should monitor the measurements and make sure responsible bodies are collecting them
Evaluate adherence of the measurements against defined measurement descriptions.	Organization should monitor measurements are conducted in line with the organizational description
Collect and store measurement related experience to support the future use	Lessons about the conduction and use of the measurements should be collected in organizational level
Determine factors effecting measurement	Organization should be able to determine which factors in project environment has positive or negative effect over the measure
Use organizational tailored estimation models	Organization should create organizational estimation models with its historical data and update accordingly with new developments
Use control charts to evaluate measurement activities	Organization should define and use control charts for the measure, evaluate progress, and take action accordingly
Use statistical evaluation to improve measures	Organization need to use statistical tools to evaluate the measures and improve measures and processes accordingly

The practices and their criteria are used for each core measure to evaluate organizations capability for that measure. Measures that are collected at some point in the organization are at least in basic capability, which is called as level 1 in the case study. To be able to increase a capability level, a measure should satisfy all practices defined for the first level and satisfy at least one practice from the second level. For this measure, organization knows conducting some practices in the organizational level; therefore, the capability is moderate, which is called as level 2. If organization performs all practices listed for level 2 and at least one practice from level 3, then for that specific measure the capability is level 3. Being level 3 means that this measure is managed in organizational level. In addition, collected measures may be used to improve organization's performance in some way.

When the current model applied to the organizational data from the start of the project, results are obtained as listed Table 4. If a measure satisfies all generic practices of a capability level and at least one practice of an upper level, it is graded as the upper level. If the measure is not implemented or it does not satisfy any practices of level 1, it is marked as not satisfied (NS).

While evaluating the capability levels for each measure, we also decided to assign a capability value for each measure group. This level represents measurement capability of that group of measurement and organization can use this value to have a higher-level conceptual understanding. That capability level should represent the majority of the measures under a specific measure group; therefore, we calculate the median value of the practice levels of the measures under that group. These measurement capability levels for each measure group are also listed in Table 4 in the column titled MCL, representing measurement capability level.

The output in Table 4 reveals that the organization is in capability level 2 for most of its measure groups with some 1's. Being in capability level 1 for most of technical aspects is not surprising as they are outsourcing most of their development needs. Their focus is procurement process, being in level 2 in this aspect shows that they are using related core measures in organization. However, they do not have the capability to compare these measures. This situation is also reflected in the size group.

The findings of the model are parallel with the results of the conducted gap analysis during the action research; if the organization implements a size measure that can be used as a comparability measure, it can use it to measure productivity for both project and procurement aspects. When they gain the comparison capability and conduct statistical analyses with their measures to improve their processes, they can satisfy practices of level 3 and improve their overall measurement capability.

These results are parallel to the results we reach during the action research. After the improvement opportunities identified in the action research, they are implemented first with a pilot group and then organization wide. The organization starts using COSMIC functional size actively in its projects to determine in-house productivity, vendor productivity, procurement cost comparisons, and effects of requirement changes in the projects. They built their own organizational estimation models, determine goals for their measures, conduct statistical analyses with the collected data, compare normalized measures of different projects, and use measurement results to improve their processes. These activities help them to increase some of their capability levels by satisfying some practices from level 3.

Table 4 Measurement Capability Levels of Aspects at the Beginning

Measure Group	Core Measures	Satisfied Practice Levels	MCL
Project	- Planned effort	2	2
	- Actual effort	3	
	- % of effort estimation efficiency	2	
	- % of actual effort (at a specific time)	2	
	- Planned duration	2	
	- Actual duration	3	
	- % of duration estimation efficiency	2	
	- % of actual duration (at a specific time)	2	
	- Planned cost	2	
	- Actual Cost	3	
	- % of cost estimation efficiency	2	
	- % of actual cost (at a specific time)	2	
Risk	- Anticipated risks	2	2
	- Occurred risks	3	
	- Unidentified risks	2	
	- Risk identification efficiency	2	
Quality	- Number of non-conformance	2	2
	- Costs of corrective actions	2	
	- Cost of preventive actions	2	
Configuration	- Configuration changes	2	2
Change	- Proposed changes	2	2
	- Accepted changes	2	
	- Cost of change	1	
Procurement	- Procurement contract changes	2	2
	- Quality of supplied work product	2	
Requirement	- Number of requirement change	2	2
	- Cost of Requirement change	1	
Solution	- Number of design change	1	1
	- Cost of design change	1	
Test	- Number of defects	2	1
	- Internal failure cost	1	
	- External failure cost	1	
Integration	- Integration errors	1	1
Process improvement	- Process improvement proposals	2	2
	- Cost of quality	1	
Size	- Size	NS	NS
	- Normalized measures, rates	NS	

Capability levels of the organization are measured again after one year of organization wide application of the identified improvement opportunities. If the model has the ability to reflect measurement capability, we need to see the effects of the improvements in the new capability levels. The results are given in Table 5. According to the results of this assessment, capability levels for project, procurement, and size measure groups are increased to level 3, mainly because of organization specific estimation models, implementing a size measure, and gaining the ability to normalize and compare different measures. They also improve capability level of nearly all of their measure groups from level 1 to level 2.

Table 5 Measurement capability levels of aspects at the end

Measure Groups	Core Measures	Satisfied Practice Levels	MCL
Project	- Planned effort	3	3
	- Actual effort	3	
	- % of effort estimation efficiency	3	
	- % of actual effort (at a specific time)	3	
	- Planned duration	3	
	- Actual duration	3	
	- % of duration estimation efficiency	3	
	- % of actual duration (at a specific time)	3	
	- Planned cost	3	
	- Actual Cost	3	
	- % of cost estimation efficiency	3	
	- % of actual cost (at a specific time)	3	
Risk	- Anticipated risks	2	2
	- Occurred risks	3	
	- Unidentified risks	2	
	- % of risk identification efficiency	2	
Quality	- Number of non-conformance	2	2
	- Costs of corrective actions	2	
	- Cost of preventive actions	2	
Configuration	- Configuration changes	2	2
Change	- Proposed changes	2	2
	- Accepted changes	2	
	- Cost of change	1	
Procurement	- Procurement contract changes	3	3
	- Quality of supplied work product	2	
Requirement	- Number of requirement change	2	2
	- Cost of Requirement change	1	
Solution	- Number of design change	2	2
	- Cost of design change	1	
Test	- Number of defects	2	2
	- Internal failure cost	2	
	- External failure cost	2	
Integration	- Integration errors	1	1
Process improvement	- Process improvement proposals	2	2
	- Cost of quality	2	
Size	- Size	3	3
	- Normalized measures, rates	3	

As of this thesis is being written, the organization still continuously collects data from the measurement activities, continuously monitors the data and use it to improve their productivity. They also get external auditors to test the correctness of the measured functional size of the software for both verification of the measures and resolution of possible conflicts with their vendors.

3.2.5 Analysis and Results of the Exploratory Study 1

After the application the model in the first exploratory study, we need to infer an answer to the research question.

RQ-2. How well do the identified main components work to evaluate measures in software development process?

Proposition 2: Core measure set and capability practices defined in the action research can be used as a measurement capability model. If this model works well, it can assess an organization measurement capability and can reflect conducted measurement related improvements.

Validation method for proposition 2: The model version one can be applied with the data collected in the organization before and after their improvement initiative. If the model has the ability to reflect organization's capability, the results should reflect the improvements that the organization implemented during the action research.

The validation method is satisfied with the results of two application. The results reflect the improvement conducted in the organization for their measurements. Therefore, we can accept the proposition defined for the research question. As a result, the answer for the research question is:

Identified components work well to enable organizations to identify and evaluate measures in its software development process. These components may be used in a model for organizational measurement capability.

The first exploratory case study successfully answers the second research question. In addition to receiving answers for the research question, some improvements in the model is identified. First one, which is also explained during the conduct of the study, is the calculation of capability level for each measure group. In addition to that, two naming changes do not affect the work of the model. The name "measure group" may be improved by using a name that represents the actions conducted in a specific phase of development lifecycle. We decide to use the term from the work of Ozcan and Demirors (Ozcan-Top & Demirörs, 2015); aspects. It represents "sets of interrelated and interacting activities" in software development lifecycle.

In addition to changing the naming of measure group, we realize that project aspect includes a larger set of core measures than other aspects and these measures can be grouped in three parts: effort, duration, and cost. We decide to use this grouping in the next exploratory study.

We also decide to assign names to the different capability levels to make it easier to differentiate them and understand their main differences. First level of capability in a conceptual level represents organizations that conduct measurements in an ad-hoc fashion. This level may be called as "Measured" to reflect that the activities conducted do not go beyond being just measurements. Second level of capability represents organizations which define their measures in an organizational level; therefore, they are "Institutionalized". Third level represents organization that uses the results of the measurements to improve themselves; therefore, they are "Improved".

These changes are reflected to the model starting with second exploratory study, which is given in the next section.

3.3 Exploratory Study 2 – Comparison with Other Approaches

After the promising results of the first exploratory case study, we conduct a second study to find answers for the next set of research questions. To overcome some of the identified validity threats described in the first exploratory study and test the self-application possibilities of the model, this case study aims to test the model when used in a real organization by an insider. With this case study, we also aim to compare the results of the model with the available alternatives.

3.3.1 Research Questions of the Exploratory Study 2

At the beginning of the study, we define research questions and create propositions aiming to answer the question. We also structure validation methods to test the proposition that constitute the starting point of the exploratory study. Action research and the first exploratory case study provide answers for the first two research questions. In this exploratory case study, we aim to find answers for two sub-questions under the third research question, 3.1 and 3.2.

RQ-3. How useful is the model version one for software organizations?

RQ-3.1. How clearly does the model version one reflect the state of the organization with respect to their measurement capability?

RQ-3.2. What differences do results of the model version one include when compared with available measurement capability evaluation methods?

Proposition 3: Successful application of measurement capability model can help organizations to evaluate their measurement capabilities and plan improvements. The usefulness of the model can be determined with the responses to research questions 3.1, 3.2, and 3.3.

Proposition 3.1: To be able reflect an organization's state clearly; measurement capability model version one should demonstrate similar results as the alternative methods currently available in the literature.

Proposition 3.2: When applied parallel with other available methods, measurement capability model provides more benefits than its drawbacks when compared to other methods.

Validation method for proposition 3: This exploratory case study helps to answer research questions 3.1 and 3.2. Next exploratory case study answers research question 3.3. If all questions have positive results, we can combine them to identify usefulness of the model.

Validation method for proposition 3.1: Model is applied in an organization together with two available methods and their results are compared. If the results are similar, proposition 3.1 should be accepted.

Validation method for proposition 3.2: At the end of the application of the all three methods, results are evaluated together with the advantages and disadvantages of all three methods. If there are more additional benefits of the new model compared with the others than its drawbacks, we can accept proposition 3.2.

This study uses the model version one with the improvements identified at the end of the first exploratory case study. We start by planning the activities in the study, then select the case and identify information sources in the case. After that step, the case is conducted and at the last step, the results are analyzed to find the answers for the research questions.

3.3.2 Activity Planning in the Exploratory Study 2

Planned activities for the case study are listed below:

- Case selection: To test the propositions during the case study, four main criteria are determined for the selection of the case. First criterion is that the case should be conducted in a real software developing organization to observe the results of the evaluations in a real-life context. Second criterion is that the case study should be conducted by an internal party from the organization who can access necessary data. Using an internal party can help us to observe the applicability of the model without an external expert. Third criterion is that alternative measurement capability methods can also be applied to the case to compare their results with the proposed measurement capability model. All of the methods are expected to be applied by the same participant by using available organization data. Fourth criterion is that there should be more than one division in the organization to be able to use a multiple case design to test the model with multiple cases.

- Establishing case study environment: Case study is conducted by an employee of the target organization with the organizational data; therefore, it is conducted in the organizational premises.

- Identifying information sources: Information is collected from the organizational documents, systems, and key personnel related to measurement by the conductor of the study. As the conductor is the part of the organization, data collection is expected to be unproblematic. The conductor uses available scientific publications to understand and apply different assessment models and methods.

- Application of the methods: Assessments are performed by the conductor and assessment results and processes are recorded for evaluation

- Analyzing the conduct of the case studies: The data collected during the case study is evaluated by the conductor to prepare the results. After the results are prepared, the researcher and the conductor control the data and the results to be sure that the deductions are accurate.

3.3.3 Mitigation of Threats to Validity in the Exploratory Study 2

Threats to internal validity: There is an important threat to internal validity in this study, which is related to the collection of the data. As the conductor of the study is an insider of the organization, he already has knowledge and opinion about the measurement capability. This situation is both an advantage and a threat. It is an advantage because the assessment process is expected to be easier having the knowledge already available. It is a threat because he may overlook some points during the assessment process by thinking that he already has the information. To mitigate this threat, the conductor receives two main instructions: first to examine evidences about all measures and second to crosscheck the findings with other participants in the organization.

Threats to external validity: Generalizability is a threat to external validity of this study as it is applied in one organization. To mitigate this threat three different divisions of the organization, which are following independent processes, are assessed as three cases. In addition, there are additional case studies in the research to mitigate this threat for all case studies.

Threats to construct validity: A threat to the construct of the study is that the study is carried out by a third party who does not involve in the development of the model. As the model is in its initial development phases, for a third party it might be hard to understand how to apply and evaluate the model. To mitigate this threat, the researcher works closely with the conductor before the study, and he is available during the study for any possible question about the model.

Threats to reliability: There might be an operational judgment towards the study from the conductor as he is a part of the organization where the model is applied. To mitigate this threat, the researcher provides clear instructions about the assessment approach and dictates to collect necessary evidences about the measure. The conductor also shares his findings with his colleagues from the other divisions to validate his findings. At the end of the study, the results and related evidences are also shared with the researcher by the conductor.

3.3.4 Design and Execution of the Exploratory Study 2

Case Selection in the Exploratory Study 2: The purpose of this case study is to compare measurement capability model version one and other available measurement maturity models. Before deciding case selection criteria, we have to decide which models to test beside measurement capability model. There are two models explained in the state of the art section that have similar purposes like our methods: Measurement-CMM (Niessink & Van Vliet, 1998) and MIS-PyME (M Díaz-Ley et al., 2008). However, assessment method of Measurement-CMM is not provided and it is an aged method. Instead of Measurement-CMM, we decide to use a method from other set of maturity models. Among them we choose MeSRAM (Staron & Meding, 2015) as it is the most up-to-date model. It aims to measure the robustness of a measurement system, which may provide comparable results with a capability assessment.

After deciding the methods to be used, we continue with the selection of the case. First criterion is that it needs to be a real-life case. It is necessary to understand the differences of applying all methods on a real-life case. Second, the case should be applied by someone other than the researcher to mitigate some of the validity threats with the application of the model. If the study can be conducted with an internal party from the organization, we can also understand the effectiveness of the model when it is applied by the organization itself rather than a third-party analyst. To satisfy both of the options, we decide to conduct the study with an internal party from the organization. Third, there should be available data to apply different methods. As the conductor is a part of the organization, he does not have any difficulty to reach organizational data. Forth criterion is to apply the methods in different divisions or departments of the organization to create a multiple case study and to observe the results of the methods with different cases.

The cases are selected from a large IT solutions organization located in Ankara. The conductor of the case study is an employee of this organization and a student from the Informatics Institute of METU. Before deciding the case study, the conductor examines the methods and carries out a short analysis to understand whether he can access necessary information from the organization. He clarifies any questions he has about the model with the researcher.

After the organization is decided, three different divisions in the organization are selected for the case study. These divisions are working independently from each other. They all utilize waterfall software development lifecycle but each of them uses different processes. Therefore, they can be considered as three different organizations and three different cases.

Case one is conducted in division A, which is developing collection systems, VPOS (Virtual Point of Sale) applications, loyalty systems and dealer management systems with its 135 employees. Case two is conducted in division B, which focuses on telecommunication systems and developing field operation systems, switchboard systems, and workforce management systems with 235 employees. Last case is conducted in division C, which is developing health information management systems with its 35 employees.

Case Study Environment in the Exploratory Study 2: The study is conducted in the organization of interest. As each division has their own data sources and repositories, the conductor visits each of them to use process documentation and process products. The conductor also interviews with related personnel whenever necessary. The hardware and software used for the analysis is the same for each division. Hardware is the portable computer of the conductor and as software Microsoft Word and Microsoft Excel is used to collect, analyze and report data.

The three methods used in the divisions are in the order of execution MeSRAM, MIS-PyME, and Measurement Capability Model Version one (MCM v1). In each division, the methods are applied with the given order and their results are recorded.

Information Sources in the Exploratory Study 2: Information about the measurements conducted in the organization are supplied from different sources depending on the organizational processes. Main sources of information are; process participants, process definitions, process products, organizational databases, and organizational information systems. During the case study, all of the possible sources of information are used. The conductor first analyzes organization information systems and databases to reach process definitions and products. After the available information is extracted, if there is still a gap in the required information it is obtained from process participants of related division. Moreover, for all divisions, the results of the methods are shared with the process participants and they are asked to review the results for possible misrepresentations. Reviews are evaluated by the conductor and discussions are held with the participants if necessary to reach a decision about the result.

Application of the Methods in the Exploratory Study 2: In this section, there will be brief explanations of the application of the methods in the organization. Detailed explanation is published in a project report (İnce, 2016).

- **Application of MeSRAM:** MeSRAM aims to evaluate robustness of the measurement program of an organization. It examines the robustness under seven main categories; metrics used, decision support, metric infrastructure, organizational metric maturity, metrics organization, external collaboration, collaboration with academia. The category ‘metrics used’ has its own five subcategories; business metrics, product metrics, design metrics, organizational performance metrics, and project metrics.

  Organizations require answering a questionnaire for each category and subcategory. The answers may be “Y” for yes, “N” for no and “?” for unknown answers. “?” is evaluated as “N”. Detailed results of the each questionnaire for each category and for each division are given in “APPENDIX A – MeSRAM Questionnaire Answers”. Summary of the answers are given in Table 6. For each category and for each division, number of questions, number of positive answers, number of negative answers, and ratio of positive answers are given respectively.

Table 6 Summary of questionnaire results for each category

Category	Division	Number of questions	Number of "Y"	Number of "N"	Ratio of "Y"
Metrics Used	**A**	37	5	32	13%
	B	37	14	23	37%
	C	37	14	23	37%
Decision Support	**A**	6	0	5	0%
	B	6	3	3	50%
	C	6	3	3	50%
Organizational Metrics Maturity	**A**	8	2	6	25%
	B	8	3	5	38%
	C	8	3	5	38%
Metrics Organization	**A**	21	0	21	0%
	B	21	5	16	24%
	C	21	5	16	24%
Metrics Infrastructure	**A**	6	0	6	0%
	B	6	0	6	0%
	C	6	0	6	0%
External Collaboration	**A**	2	0	2	0%
	B	2	0	2	0%
	C	2	0	2	0%
Collaboration with Academia	**A**	5	0	5	0%
	B	5	0	5	0%
	C	5	0	5	0%

After the ratio of positive answers are calculated for each division, these values are used to draw radar diagrams for each division to visualize the results of each category. These diagrams are given in Figure 10, Figure 11, and Figure 12. According to the results, all divisions need improvements for their relations with other organizations and academia. They all need to implement a metric infrastructure and metric team to increase their robustness for the categories "Metrics Organization" and "Metrics Infrastructure". Division A need to improve its overall robustness for all categories to catch other divisions. It mainly requires an improvement for the decision support category. That category mainly shows the organization's ability to use the metrics in their decision processes, or it measures the usage of the metrics in the processes.

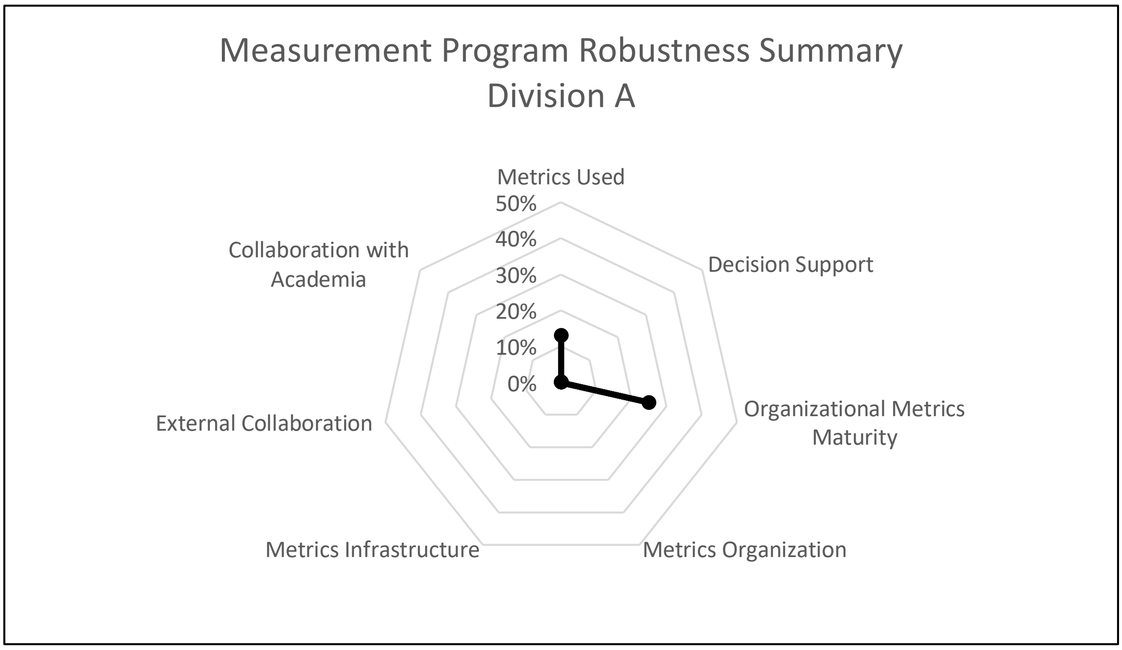

Figure 10 Measurement program robustness summary for Division A

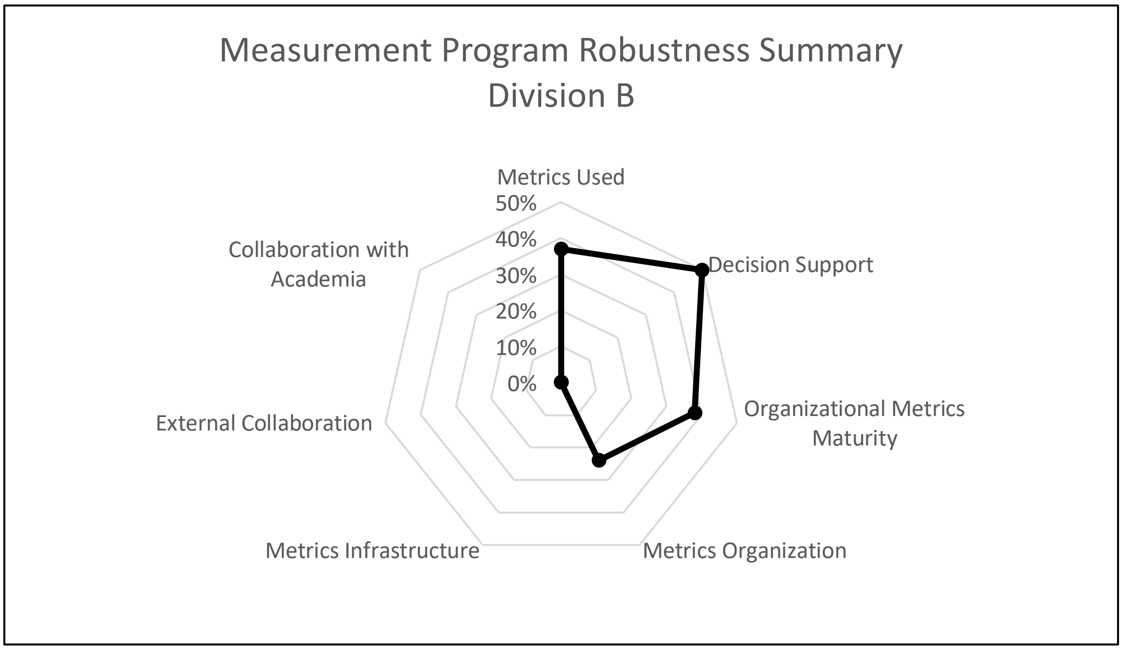

Figure 11 Measurement program robustness summary for Division B

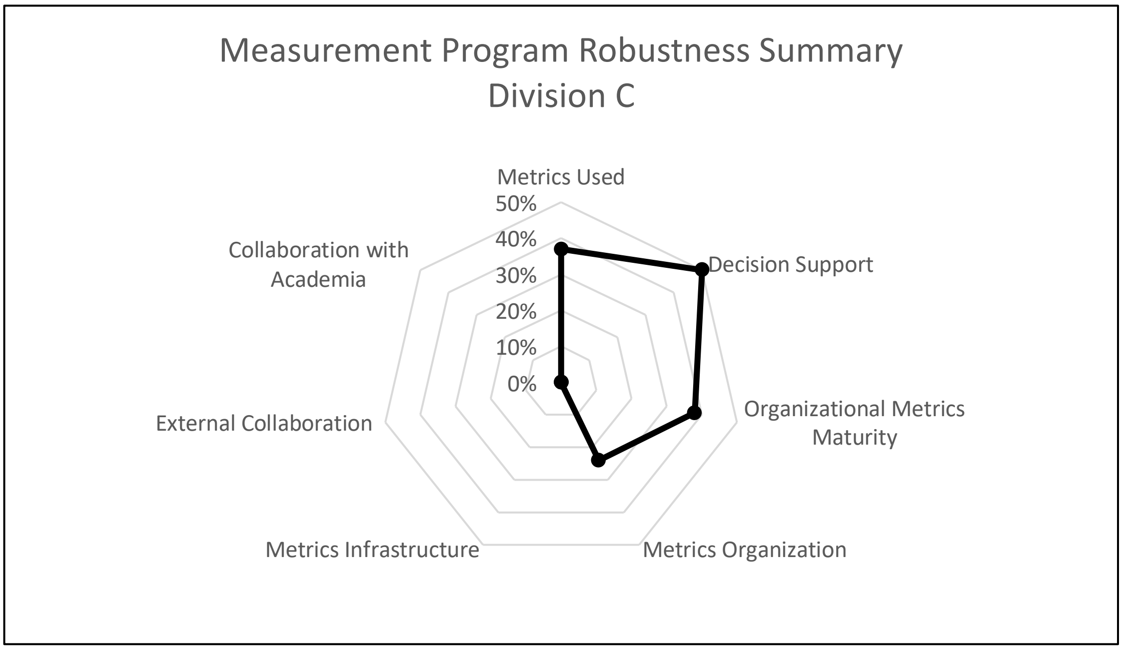

Figure 12 Measurement program robustness summary for Division C

- **Application of MIS-PyME**: MIS-PyME is a framework aiming to help measurement program definitions in small and medium enterprises. It includes a Measurement Capability Maturity Model (MCMM) to evaluate measurement maturity of organizations, it is used in this exploratory case study. It defines 6 maturity levels: level 0 – Incomplete, level 1 – performed process, level 2 – managed process, level 3 – established process, level 4 – predictable process, level 5 – optimizing process. Each maturity level has its own set of process attributes, and each attribute includes a set of questions (María Díaz-Ley et al., 2010). To determine the maturity level, the questions should be answered by "yes" or "no". The percentage of "yes" answers gives an achievement level for each attribute. The levels of achievement are given in Table 7.

Table 7 Achievement levels for proecess attributes

Percentage of "Yes"	Level of Achievement
%0 - % 15	Not Achieved
%15 - % 50	Partially Achieved
%50 - % 85	Largely Achieved
%85 - % 100	Fully Achieved

The divisions are evaluated by using the attributes and the questions. The summary of the results is given in Table 8. Maturity level for division A is 1, performed, as it doesn't satisfy most of the attributes for level 2. Being level 2 means that, although still not fully successful, division A conducts basic measurements. It conducts basic organization-wide management for the measures by the means of tools, but requires serious improvements to be able to be in managed maturity level. Division B and C mostly satisfy level 1 and 2 attributes. They also satisfy some of the level 3 attributes; however not enough to be awarded with level 3. They need some improvement for their organization-wide definitions for the measures and require conducting an improvement initiative for their usage of the measures to satisfy level 3 attributes.

Table 8 Brief results of MIS-PyME for all divisions

Attribute	Division A	Division B	Division C
Level 1: Performed Process			
PA 1.1: Process performance attribute.	Partially	Fully	Largely
PA 1.2: Basic project and product focus performance attribute	Fully	Largely	Largely
PA1.3: Basic management tools implemented attribute	Fully	Largely	Partially
Level 2: Managed Process			
PA 2.1: Performance management attribute	Not	Largely	Largely
PA 2.2: Work product management attribute	Not	Largely	Largely
PA 2.3: Basic project and product focus management attribute	Not	Partially	Partially
PA 2.4: Management and development tools implemented attribute	Largely	Largely	Partially
Level 3: Established Process			
PA 3.1: Process Definition Attribute	Not	Largely	Largely
PA 3.2: Process Deployment Attribute	Not	Fully	Fully
PA 3.3: Advanced Project Tracking Attribute	Not	Not	Not
PA 3.4: Advanced product tracking attribute	Not	Not	Not
PA 3.5: Process tracking attribute	Not	Not	Not
PA 3.6: Resources Deployment Attribute	Not	Partially	Partially

- **Application of MCM v1**: Last method to be used in this case is measurement capability model. MCM v1 provides an opportunity to the organization to assess the aspects they want to evaluate and aim to improve. Before starting the case, 6 aspects from the defined 12 aspects are chosen to be applied in the organization. Chosen aspects are project, change, requirement, solution, test, and integration. Other aspects, risk, quality, configuration, procurement, process improvement and size are not chosen. The conductor, being a part of the organization, decides that these aspects are not represented in the organization. Meaning that the organization does not practice any measurement activity for these aspects.

  For each measure under the aspects, capability levels are examined by using the practices of the model and capabilities are determined for each measure. Capabilities of each measure are given in Table 9 for the three divisions.

Table 9 MCM v1 capabilities for each division

Aspect	Core Measure	Capability for Div. A	Capability for Div. B	Capability for Div. C
Project	Planned Effort	2	2	1
	Actual Effort	2	2	1
	% of Effort Estimation Efficiency	1	1	1
	% of Actual Effort	1	1	1

Aspect	Core Measure	Capability for Div. A	Capability for Div. B	Capability for Div. C
	Planned duration	1	1	NS
	Actual Duration	1	1	NS
	% of Duration Estimation Efficiency	1	1	NS
	% of Actual Duration	1	1	NS
	Planned Cost	1	1	NS
	Actual Cost	1	1	NS
	% of Cost Estimation Efficiency	1	1	NS
	% of Actual Cost	1	1	NS
Change	Proposed Change	NS	1	1
	Accepted Change	1	1	1
	Cost Of Change	NS	NS	1
Requirement	Number Of Requirement Change	1	2	2
	Cost Of Requirement Change	1	1	1
Solution	Number Of Design Change	1	1	1
	Cost Of Design Change	1	1	1
Test	Number Of Defects	2	2	2
	Internal Failure Cost	NS	NS	NS
	External Failure Cost	NS	NS	NS
Integration	Integration Error	NS	1	1

Table 9 provides insights for each measure for all division. A brief table aggregating results for each aspect is also given in Table 10. For the project aspect, division A and B are in capability level 1, whereas division C does not satisfy practices for the most of measures. Division C requires improving itself for the measures related to cost and duration. It is probably related to relatively small size of division C and working with hospital information management systems, which are mostly dealing with change requests and small improvements. For the change aspect, division B and C are in level 1, but division A does not measure the effects of change proposal and does not actively measure the number of proposals, the teams deal only with the accepted change requests. For the requirement aspect they have the same capability, however, B and C are in capability level 2 for the number of requirement change, meaning that they are following the changes in requirement in an organizational level. For the solution and test aspects, they have same level of capability. For the test aspect, they are following defects in an organizational level; however, they do not use them to calculate failure cost. Which in return reduces their overall capability. For the integration aspect, division A does not follow integration errors; B and C follow them in a basic level.

Table 10 MCM v1 capabilities for each aspect- aggregated

Aspect	Div. A	Div. B	Div. C
Project	1	1	NS
Change	NS	1	1
Requirement	1	1	1

Aspect	Div. A	Div. B	Div. C
Solution	1	1	1
Test	NS	NS	NS
Integration	NS	1	1

After the application of the models in three divisions, overall results are analyzed and research questions are evaluated by considering the results. Next section evaluates each division considering the results of the models and then evaluates the research questions in their own subtitle.

3.3.5 Analysis and Results of the Exploratory Study 2

Analysis of the results of the application of each method is given for each division. Their results are compared to identify differences, commonalities, advantages and disadvantages. After that in the last part, the research questions are discussed considering the analyses of the results.

Analysis and Results for the Division A: According to the result of MeSRAM division A has the lowest overall robustness. It needs improvement in all areas examined by MeSRAM. Most related category with our purposes is Metrics Used category, which tries to understand robustness of used metrics in the organization. In this area division A is rated as 13%, which is quite low and requires improvement for the metrics used in the organization. When the details of the questionnaire examined, we can observe which areas require further metrics; however, the method does not provide information which metrics or measures are required. In addition, this measure evaluates the organization as a whole in a process centric way. Therefore, even though organization may not require the evaluation of all metric areas, it is graded from all areas.

Results of MIS-PyME show that division A has a measurement process with a maturity level 1, meaning that it performs the process but does not manage organization-wide. Similar to MeSRAM this method evaluates measurement process as a whole. We can only debate about the application of the practices in the organization. Division A needs to improve its process performance. It has basic project, product and management focus in place, but it does not manage them.

The results of MCM v1 are parallel with other two methods, division A uses most of the measures under the selected aspects, but most of them are not defined organization-wide. It requires improvement for all aspects and all measures. The main difference of this model is that it is easier to observe which aspects and measures require improvements and which additional measures are required in the measurement process. We can also identify the required practices to improve the capabilities. This model also lets the organization choose which aspects to evaluate; as a result, they can save their effort for the parts of process that they are not interested.

Analysis and Results for the Division B: According to the results of MeSRAM, division B has a more robust measurement process compared to division A; however, it still needs quite a bit of improvements. Similar to A, it does not have activities related to collaboration and infrastructure categories. It has some activity for metrics organization and decision support categories, and better results in metric maturity and metric used categories. Compared to division A, B can be classified as more robust, and it collects and evaluates more metrics.

Similar to MeSRAM, MIS-PyME also classifies division B as a more mature organization from the measurement perspective. Its maturity level is 2, managed process, which means that in addition to collecting the measurements the measurements are also managed in an organizational level. It even has some activities in the third level. If it implements some improvements to tracking the measurements and their results, it may increase its maturity level.

MCM v1 also evaluates division B as slightly more capable than division A for some aspects and measures. When the measures are evaluated individually, it is observed that even though they both require improvements to increase overall capability, results of divisions A and B are less distinct compared to their results with other methods. The main reason of this difference from other methods is probably MCM v1's flexible approach that lets the organizations choose the aspects that they want to evaluate.

Analysis and Results for the Division C: MeSRAM provides same results for division C and B, therefore same comments are also valid for division C. MIS-PyME also provides mostly similar results. The results of MCM v1 also similar for them; however, there is an important distinction in project aspect. Division C is evaluated at NS for duration and cost measures whereas division B is evaluated as level 1 and for effort estimation and collection C is level 1 whereas B is level 2. The reason of this difference may be that division C is smaller with 35 employees when compared to 235 employees of division B. Being a smaller organization, C may manage projects without formal estimation and data collection activity definitions. Another point to consider is that C develops information systems for hospitals and these development activities are mostly carried out by change requests and error fixes.

MCM v1 is a flexible approach when compared with the other methods. It provides an important advantage; we can observe the difference between B and C in project aspects. As other methods evaluate the organizations from a process centric view, they miss the difference in practices for individual measures and evaluate them nearly same. With a measure-oriented approach, MCM v1 identifies the difference between the two divisions.

Evaluation of Research Questions in the Exploratory Study 2: Second exploratory study is conducted to find answers for three research questions, one primary and two secondary questions. Below, the answers for the research questions are discussed starting with the secondary questions.

RQ-3.1. How clearly does the model version one reflect the state of the organization with respect to their measurement capability?

To be able to answer this question we apply the model with two other methods aiming to assess measurement processes of an organization. To be able to give an answer to this question, assessment results of the methods should be compared with each other. The results that are discussed in previous sections show that they provide similar results while evaluating organization's measurement practices. Even though they have different focus, all methods identify division A as the least capable division and they identify B and C with similar capabilities. Considering these results, we can say, "*Proposed measurement capability model quite clearly provides similar results about the state of an organization's measurement capability when compared with the results of available measurement capability evaluation methods.*"

RQ-3.2. What differences do results of the model version one include when compared with available measurement capability evaluation methods?

The results of the application demonstrate that even though the proposed model provides similar results with the two other methods applied in the organization, it has additional benefits and one main drawback. First benefit is that organizations can choose which aspects to include into the analysis, and do not waste their effort to analyze components that they do not want to evaluate.

Second benefit is that MCM v1 helps organizations to assess and evaluate individual aspects and measures instead of the complete measurement process. That approach helps to identify which measure from which activity needs to be improved and helps organizations to draw a more direct route for their improvement activities.

Third benefit is the measurement-oriented approach of MCM v1, which helps organizations to observe nuances between the measures. In the case study, other methods rate division B and C nearly in the same maturity level. MCM v1 demonstrates that even though they have similar capabilities for most aspects, there is an important distinction in project aspect. Using a model with more granularity in the focus areas helps organizations to observe their weak and strong points with more precision.

Fourth benefit is that MCM v1 aims to evaluate capabilities of measures instead of processes. Therefore, when compared with the other applied methods, if an organization does not want to analyze its processes, or does not want to follow a process-centric improvement approach, MCM v1 is the best available choice.

Fifth benefit is related to the required effort for the application of the methods. Even though it is only one case study and the effort values are small and close to each other, in this case study MCM v1 requires the least amount of effort. Being able to select aspects for evaluation is an important factor reducing required effort.

The drawback of the model is that, organizations should choose the aspects they want evaluate with the model; therefore, they are unable to know their status regarding with the aspects that are not selected. However, not choosing an aspect already reflects that the organization has some degree of awareness that they may have weakness about that aspect.

Considering the identified differences of the results of MCM v1, we can state, "*proposed measurement capability model version one provides five benefits and one drawback when compared with the results of available measurement capability evaluation methods*".

RQ-3. How useful is the model version one for software organizations?

To answer the research question 3, we need to have the answers for research questions 3.1, 3.2, and 3.3. In this exploratory case study, we observe the answers for 3.1 and 3.2, which demonstrates that the model provides additional benefits as listed previously while providing similar results. Research question 3.3 is tested in the next exploratory case study. After the evaluation of research question 3.3, we can provide a full answer for research question 3.

After the completion of the second exploratory case study, the model version one is applied in a large organization with three divisions to observe its results in a large-scale setting. Next chapter explains the third exploratory case study.

3.4 Exploratory Study 3 – Application in a Large Organization

After the second exploratory case study, a third exploratory case study is performed to apply the model version one in a large-size organization and investigate the last research question. This case study is conducted in three divisions of a large software developing organization as three cases. After its completion, the results are presented to the senior managers of the organization in a meeting. In this meeting, the model and its results are discussed to understand their opinions about the correctness of the results and effectiveness of the method.

3.4.1 Research Questions of the Exploratory Study 3

Third exploratory study aims to provide an answer mainly to the research question 3.3. If we can provide an answer for that question, we would also be able to answer third research question, as we already provides the answers for the questions 3.1 and 3.2.

RQ-3. How useful is the model version one for software organizations?

Proposition 3: Successful application of measurement capability model can help organizations to evaluate their measurement capabilities and plan improvements. The usefulness of the model can be determined with the responses to research questions 3.1, 3.2, and 3.3.

Validation method for proposition 3: Exploratory case study 2 helps to answer research questions 3.1 and 3.2. Exploratory case study 3 can provide an answer for the research question 3.3. If all questions have positive results, we can combine them to identify usefulness of the model.

RQ-3.3. What benefits and shortcomings does the model version one provide to software organizations?

Proposition 3.3: Representatives from an organization in which the model is applied to assess measurement capability can state the benefits and shortcomings of the model.

Validation method for proposition 3.3: Model is applied in an organization to assess its measurement capabilities. The results are presented to the senior management. The responses of the senior management should provide answers of research question 3.3.

3.4.2 Activity Planning in the Exploratory Study 3

The activities that are planned for the third exploratory case study are listed below:

- Case selection: Main aim of the third exploratory case study is to provide an answer for the research question 3.3, therefore we need a real organization to apply the model and then present the findings to the upper management. Meaning that we need a case from an organization developing software and we need upper management to cooperate with us to evaluate the findings. Application of the model in a large-sized organization can help us to observe its applicability in a large organization. If possible, the model should be applied in more than one cases to observe multiple results.

- Establishing case study environment: Case study should be conducted in the organizations premises and then a meeting should be conducted with the upper management in the organization.

- Identifying information sources: Depending on the organization in which the study is conducted, main information sources should include process products, projects, project documents, organizational information systems and databases, and key personnel participating in measurement activities.

- Application of the model: The model should be applied in the organization with the selected cases. Assessment report should be prepared and shared with the organization, and then an evaluation meeting should be conducted with the upper management.

- Analyzing the conduct of the case study: Findings of the case study should be inspected and the results should be evaluated to discuss the validity of the proposition 3.3 and 3, and the answers for the research questions 3.3 and 3 are finalized.

3.4.3 Mitigation of Threats to Validity in the Exploratory Study 3

Threats to internal validity: Data collection and assessment for the case study follow a tight schedule because of the constraints from the organization. Because of the tight schedule, a threat for the internal validity is the loss of interest and focus from the participants during lengthy and intense sessions. To mitigate this risk, case sessions are decided to be shorter than 75 minutes with 30 minutes breaks between them and a 60 minutes break after two sessions. With this schedule, 4 sessions are conducted in a workday. In addition to this precaution, the researcher requests from the organization to work with at least two set of different personnel for each case. One group of personnel should attend the sessions before the noon and one group of personnel after the noon.

Threats to external validity: A threat to external validity of a case study is the representation of the setting. To mitigate the threat and collect realistic data, a real-world setting is used with the data from several real, completed software projects. In the organizations, upper management also reminds to the participants the importance of collecting real data. The anonymity of the collected data and reports are also stressed to the participants to remove any doubts about using this data to evaluate project or employee performance.

Threats to construct validity: To mitigate possible threats to the construct validity, more than one project from each division are evaluated during the study. In addition, to overcome a mono-method bias, a second researcher accompanies the researcher during the conduct. This additional researcher is an expert in process analysis, process improvement and software measurement with a PhD degree and 10 years of experience in the industry. Two researchers evaluate the data simultaneously and later compare their results to overcome any discrepancies.

Threats to reliability: To ensure the reliability of the results, another researcher, who does not have a direct interest with the results of this study, also attends the conduct of the study. That researcher follows through the meetings and data collection, and reviews the assessment results and reports. In addition to the additional researcher, a participant from the organization always attends the meetings and data collection to ensure that the meetings are conducted according to the determined agenda. At the end of the study, results are shared with the organization for their review and for possible objections to the results.

3.4.4 Design and Execution of the Exploratory Study 3

Case Selection in the Exploratory Study 3: For the selection of the case, we determine three criteria: a large software developing organization, cooperation of the upper management, and more than one division to apply the model. The organization in which the study is applied is a large software organization with more than one thousand employees. They are developing defense and civil software both for public and private sectors. Their processes are officially assessed as CMMI level 3 complied. The study is conducted in three divisions of the organization as three different cases. The process guidelines that these divisions are following are determined by a central quality department and these CMMI level 3 compliant guidelines follow waterfall development lifecycle; however, the divisions have independence on how to carry out them and some teams try to utilize agile principles in their activities. Even though some of them try to follow agile principles they cannot be classified as agile cases as the central guidelines they follow force them to prepare all necessary documentation.

Three divisions have their specialized areas that they are focusing on; they are producing different types of software for their specialized domains. The details of these domains, software types, name of the organization, and the division details are not provided in this thesis as long as they are not related to the context because of the confidentiality agreement with the organization.

Case Study Environment in the Exploratory Study 3: After the organization is determined, an agreement is reached with them related to the collection of the data. According to the agreement, the study is conducted in the organization, in the offices of related divisions. There will be at least four sessions with each division and two groups of key employees from each divisions are expected to attend to the meetings. In addition to the employees from the divisions, a representative of the central quality department also attends the meetings together with the two researchers.

In addition to the office space and attending personnel, the study environment also includes personal computers of the researchers and organizational information systems. Information systems are only used by the employees of the organization to search and show data requested by the researchers.

Information Sources in the Exploratory Study 3: Main information sources during the case study are the key personnel. They provided the information about the way of doing business in each division, the processes they follow. They showed projects artifacts about the collected measures. Process definition documents are used to learn defined and collected measures, and reporting activities. Process outputs, project artifacts, and reports provide the information about conducted measurements, measurement repositories, and the usage of collected measures.

Application of the Model in the Exploratory Study 3: During the application of the model, first the aspects are selected for evaluation. Being a large organization all aspects are in use and they are all included in the study. To assess the capabilities of the divisions, meetings with each division are conducted in the organization. During the meetings, each aspect and their core measures are evaluated by using organizational data with the help of the personnel attending to the meeting. The evaluations are conducted with the help of the generic practices defined in the model.

Total effort spent for the conduct of the case study is recorded for future reference. For the first division 5 sessions are conducted. Three division representatives, two researchers, and one representative of the quality department are attended in the meetings. Each session were approximately 75 minutes, in total 37,5 person-hours spent for the first division, which is close to 4,5 person-day. For the second division there were two division representatives in four sessions making 25 person-hours and close to 3 person-days. For the last division, two division representatives are attended to the four sessions making 25 person-hours and 3 person-days similar to the second division. In total 10,5 person days are spent to assess the measurement capabilities of three divisions of this large software developing organization. Main reason of the lengthy meetings is that the researchers are examined all measurement evidences for all core measures and for all general practices defined in the model version one.

The results of the analysis exhibited a uniform result for all three divisions. This result may be related to the centrally defined processes of the organization by the quality department. Divisions have their differences in the method of application of the rules and they use different technological infrastructures; however, they all collect, store, and use the measures defined in the central processes.

Detailed result matrix for the measures and their satisfaction for the generic practices are given in “APPENDIX B – MCM v1 Results in Exploratory Study 3”. Briefly, except the aspects under strategic group, all aspects and measures satisfy all practices of capability level 1 and capability level 2. In strategic group, cost of quality measure under process improvement aspect is not used in the organization. However, all the required measures to calculate this derived measure is available. Therefore, this aspect is assessed as capability level 2. Under the size aspect, even though size measure is widely used, normalization of the measures is not carried out in the organization, as most of the projects use incomparable size measures, like line of code or number of requirements. Therefore, this aspect is assessed as capability level 1.

The result shows that in overall the organization is in level 2. They can advance to level 3 if they implement a few improvements; a comparability measure, establishing organizational estimation methods, and statistical control of already collected measures. Being in level 2 is also consistent with organization’s current CMMI level, which is level 3. In this level, organizations are expected to define and follow their processes centrally, which is consistent with the expectations of level 2 practices of MCM v1.

This results and improvement opportunities are collected in a report and shared with the organization for their review. After the review of the report is completed, a meeting is planned with the attendance of vice president responsible from quality, the director and managers from the quality department, representatives of the divisions, key personnel from the divisions, and key personnel from the quality department. In the meeting, the results are presented to the participants and discussions are held about the application process of the model, correctness of the results, possibility of implementing identified improvement opportunities, and cost benefit analysis of the implementing improvements. Overall, the senior management and the participants state positive opinions about the application and results of model. They agree on the benefits of improvement opportunities exhibited by the model; however, before implementing them they want to carry out their own feasibility studies to make sure that the benefits compensate the effort required to implement them.

In the next section, the findings of the explanatory case study are examined by considering the research questions.

3.4.5 Analysis and Results of the Exploratory Study 3

After the completion of the study, the results are evaluated to find answers to the related research questions. To determine the answers, the experiences during the conduct of the study, the results of the application, and the feedback received from the participants in post-study meeting are used.

RQ-3.3. What benefits and shortcomings does the model version one provide to software organizations?

The proposition for the research 3.3 requires a response from the organization for the application and result of the case study. During the meeting with the participant from the organization, we receive positive responses. These responses can be outlined in several points as listed below. In the responses, the views of the applicants are given as ideas instead of exact statements, because of the security restrictions of the organization. The organization has high security clearance and it is not allowed to record any voice or video inside the premises. Therefore, the comments are noted by the researchers for later use during the meeting.

Application effort required for the model: The application of the model took 10,5 person-day effort. Considering the size of the organization, different types of products they are producing, different types of customer, different types of technology, and different types of development life cycles, organization finds this effort as acceptable considering the results and benefits of the model. Organization already has experience with extensive assessments, as they are a CMMI level 3 organization and they go through assessment for all related process areas. Therefore, they have a benchmark to compare and evaluate required effort for the provided benefit.

Flexibility of the model: The organization is satisfied with the direct focus on the measures of the model. In future, they want to implement an initiative to gain the ability to benchmark themselves with other organizations in the industry. Benchmarking the measurements can provide great benefits if the organization can improve their capability and conduct comparable measures. Observing the improvement opportunities as action items for each individual measure to reach their goal of benchmarking is appreciated by the organization.

Applicability with different development lifecycles: The organization follows a process centric approach to assess and improve their development activities. However, the divisions are free to use different approaches as long as they do not conflict with the organizational process definitions. Some teams follow strict waterfall processes while some apply agile principles into their workload; therefore, organization experiments with a composite development approach. The model is successfully applied with all the different approaches. However, there is not enough evidence to claim that this study proves that the model is applicable for all development lifecycles. A later study focusing on agile development approaches can provide a definitive answer for this point.

Correctness of the results: The model assesses the organization in capability level 2 nearly for all aspects and measures. A comparable size measure is required to be level 2 for all measures. In the meeting, participants state that this is an expected situation and known by the quality department; however, they do not have any evidence supporting this. Their intention to conduct benchmark with other organizations could not be started because of the missing denominator, which can make it possible to compare different projects and organizations with each other. They are also aware that they do not satisfy the practices of level 3, as they do not use collected measurement data to conduct organizational analyses and improve their processes. The senior management states that even though most of the problems are known in the organization, the results help them to see themselves from a higher perspective. The model helps them to quantify the problems that cannot be communicated with the senior management easily.

Benefits of the results: In addition to the comments given above related to the correctness of the results, senior management also comments about the identified improvement opportunities. The discussion mainly focuses on the implementation of an objective size measure to gain the ability to compare measurements. An objective size measure is important for the organization to be level 2 in all aspects and it is also necessary to gain the benchmarking capability. They state that it is an important item and they will be focusing on implementation of a new size measure. As a shortcoming of the model, they state that the introduction of a new measure into organizational processes is not an easy job. To start a possibly costly project of introduction of a new measure, they want to conduct a small feasibility study themselves. To overcome this shortcoming of the method, the researcher decided to introduce detailed descriptions of measures in the model version two, which can help organizations while introducing new measures into their processes. These descriptions are mentioned in "Chapter 5: MEASUREMENT CAPABILITY MODEL" and also given in "APPENDIX E – Sample Measures and Related Information".

Another improvement idea that senior management focuses on is related to the usage of the measures. Even though organization collects the measures and store them in the enterprise resource planning software, they do not actively use historical measurement data to conduct statistical analyses and improve their estimation models. This gap is detected with the practices of capability level 3. Senior management agrees that this is an important point to improve and provides great benefits when compared with the effort required for implementation.

Considering the positive comments of the participants from the organization about the conduct of the model and provided results, we can give an answer to the research question: *Senior management and key personnel of a large software developing organization state benefits of using the proposed software measurement capability model in five main topics and point out a shortcoming*

After finding an answer for the research question 3.3 we can look at the research question 3.

RQ-3. How useful is the model version one for software organizations?

The proposition to answer the research question 3 is that research questions 3.1, 3.2, and 3.3 should be answered to determine the usefulness of the model version one. Second exploratory case study provides answers for the first two questions by showing that the model provides similar results as available methods and provides additional benefits compared to them with some drawbacks. Third exploratory study shows that a large organization states the benefits of the model. Considering the results of these two case studies, we can give an answer the third research question.

Application of measurement capability model version one helps organizations to evaluate their measurement capabilities and plan improvements. Usefulness of the model can be determined with the responses to research questions 3.1, 3.2, and 3.3. We can say that *measurement capability model version one is quite useful for organizations in reflecting current measurement capabilities of them and guiding them to identify improvement opportunities. It provides the opportunity to evaluate measurement practices without an in depth process analysis, it lets the organizations to select the aspects that they want to focus on, it provides improvement opportunities for the selected aspects.*

3.5 Exploratory Study 4 – Finalizing the List of Core Measurements

After the completion of the first three exploratory case studies and finding answers for the first three main research questions, we want to be sure that the model can also be successfully applied in organizations utilizing agile software development approaches. Aspect oriented approach is used in the model successfully used in the previous case studies. During this case study, without changing the fundamental premise of the model, the list of measures and general practices used to evaluate the measures are updated. The list of measures is discussed with agile methodology experts and by considering their opinions, the list is finalized.

3.5.1 Research Questions of the Exploratory Study 4

This study aims to provide answer to the research question 4.

RQ-4. Which aspects and core measures should be added to the model version one to update it to be applied in organization utilizing agile software development approaches?

Proposition 4: Agile software development methodologies may require a different set of measures when compared to the traditional development approaches. As the model aims to be also used by agile organizations, the measure list should be arranged accordingly. This list of aspects and core measure can best be identified experts working with agile approaches.

Validation method for proposition 4: The aspects and the list of measures is updated considering the base practices of agile approaches and then this list is discussed with agile experts to collect their opinions. Considering their comments, this list is reviewed. The final list with the aspects and measures are the answer for the research question 4.

3.5.2 Activity Planning in the Exploratory Study 4

The activities of the exploratory study 4 are listed below:

- Creating the measure list: The aspects and measures of the model is updated by considering the experiences gained from the conducted case studies. Some measures related to the agile approaches are also added into the list. Before the discussion with the experts, details of the measures are also added into the list: aspects, measured entities, measured attributes, and sample measures.

- Identifying information sources: The list of measures is discussed with experts; the source of information is the experts and their experience with agile methodologies. Identification of the experts is important as they directly affect the outcome of this case study.

- Establishing case study environment: Case study is conducted as online interviews with the experts. Prior to the interviews the list of the measures are shared with the experts to examine. A form prepared for interviews is used during all interviews to collect the comments.

- Conducting the case study: Interviews are conducted with the experts and comment forms are completed according to their comments.

- Analyzing the results of the case study: The comments for each aspect and measure are evaluated and an evaluation table is built combining all comments in brief forms. After that, necessary changes are reflected on the final measure list. General practices are also updated with the gained insight from the experts to reflect agile perspectives.

3.5.3 Mitigation of Threats to Validity in the Exploratory Study 4

Threats to internal validity: Main validity threats for this study are related to the instrumentation and the selection of the participants. Instrumentation is related to the instruments to use data. For this case, they are the data collection forms used to collect opinions of the participants. If the documents are prepared poorly, they may affect the result of the study. Therefore, special care is given to these documents to mitigate this threat; they are prepared before the interviews including all questions the experts need to consider. Moreover, before the interviews, detailed explanation documents are shared with the experts. These explanation documents include aspects groups, aspects, entities and their descriptions, attributes, sample measures, measure justifications, measure purpose, measure application method, measure formula and measure scale.

Selection of the participants is another threat to the internal validity. If the experts participating in the study are selected poorly, the results may not reflect the real situation. To mitigate this risk, selected experts aim to have different point of views for the agile development approach. From four experts: one of them is working in large-scale technology organization in a senior position, one of them is working in a small-scale organization as an agile project manager. One of them works as a researcher with a PhD degree specialized in agile approaches, one of them works as a consultant and researcher specialized in agile approaches with industry experience and again with a PhD degree.

Threats to external validity: Main threat for the external validity of this study, similar to the internal validity threats, is related to the selection of the participants. If we select wrong participants for the study, the results may not represent the reality. To mitigate this risk we pay special attention to select experts with experience in agile and measurement. In addition, to increase selected experts' power of representing the reality, we aim them to have different perspectives about agile approaches. Therefore, we select two participants from industry with different experience levels, one researchers and one consultant.

Threats to construct validity: A threat to the construct validity is miscommunication of the purpose of the study to the participants. The aim of the study is to collect the opinions of the experts about the fitness of the list of aspects and core measures to agile approaches. If this purpose is not communicated well, the topic may skew towards their personal experiences about the measures rather than a general evaluation. To prevent this and mitigate the threat, a detailed list of the measures and their explanations are shared with the experts before the study along with the purpose of the study. During the study, a pre-prepared questionnaire is used including questions to guide the participants towards the main goal of the study.

Threats to reliability: The main threat to the reliability of the study is the reliability of the evaluation of the responses of the participants. The responses from the four experts are evaluated to decide the necessary changes on the model. If this evaluation is conducted without proper controls, the results may not reflect the real situation. To prevent this, first the responses from individual participants are evaluated to create a table with codes for each measure: not applicable, beneficial, common and beneficial. Then responses from four participants are combined and evaluated together. Measures mostly rated as common and beneficial do not require a change, whereas measures mostly evaluated as beneficial may require some changes in their definition. These changes should be applied considering the comments of the experts. Lastly, the measures evaluated mostly as not applicable need to be marked as optional for agile methods.

3.5.4 Design and Execution of the Exploratory Study 4

Case Selection in the Exploratory Study 4: The main case of the study is the list of measures structured in a hierarchical way starting from aspect group and continuing with aspect, entity, attribute, and sample measure. In addition to this list, a table including the detailed explanations of the measures is also provided to the participants. This list is given in the fourth chapter where the model is explained. The list of measures is constructed to include measures that can be beneficial for agile approaches in addition to measures used in previous case studies.

Case Study Environment in the Exploratory Study 4: The participants of the case study are from different cities; therefore, a face-to-face discussion is hard to schedule. To overcome this problem, online discussions are conducted with each of the participants. Necessary documents are shared with the participants before the discussions and an online meeting software is used to conduct the discussions. Data is collected with the prepared data collection templates.

Information Sources in the Exploratory Study 4: Information sources of the case study are the participating experts. During the discussions, we also ask questions about the information about the experiences and education levels of the participants. These are listed in Table 11.

Table 11 Information about the participants of exploratory study 4

Participant	Current Title	Years of Experience in Software Industry	Years of Experience with Agile Approaches	Level of Education	Graduation Department of Latest Degree
1	Project Manager	8	8	Master's Degree	Computer Engineering
2	Digital Learning Solutions Technology Manager	21	10	PhD Candidate	Computer Engineering
3	Researcher	15	8	PhD	Information Systems
4	Post-Doctoral Researcher	10	6	PhD	Information Systems

Application of Exploratory Study 4: Before the discussions with the experts, the list of measures and practices are updated by considering agile principles and common measures used in agile approaches. Scrum is the most common agile method (Collab.net & VersionOne, 2018), therefore; common measures used in scrum are examined (Ifra & Bajwa, 2016). Crucial measures that can be used by organizations utilizing all development lifecycles are included in the list. These measures are mainly aim to quantify blockage during the sprints, efficiency of development flow, velocity of the team, burn rate of backlog, code churn, release frequency, involvement in reflection meetings. In addition to including measures related with agile approaches, some measures are deleted, updated, or included. For example measures related with the changes in project artifacts are combined in one measure for the sake of simplicity, several measures are included to reflect deigns complexity, and measures related to the quality of supplied software are increased and clarified. The entities and their attributes that the measures aim to quantify are also included in the list of measures.

Considering main principles of agile manifesto (Beck et al., 2001) general practices are also updated. Agile approaches value individuals and interactions over processes and tools, working software over comprehensive documentation, and responding change over following a plan. Therefore, practices related with using a process or plan are removed from the list. Also, the practices of capability level 3 that are related to statistical evaluations and control charts are excluded from the list as they are not suitable with agile principles. Instead practices related to communication and evaluation in organizational level are added.

After the measure list is updated it is shared with experts and experts are asked several questions about aspects and measures included in the model. These questions are listed below:

- Questions regarding the aspects:
- Do you think it is important to measure this aspect?

- If not, why?
- Do you have any other measure suggestion for this aspect?

- Questions regarding the measures:
- Do you use this measure?
- If yes why?
- If you use it:
- Do you record measurement data?
- Do you act on measurement data?
- If you don't use it:
- What do you think about the possible benefits of this measure?
- Will the benefit of using it be more than the cost of implementation for your environment? Why?
- Will it be beneficial for agile teams in general? Why?
- What do you think about the collection cost of this measure?
- Can it be easily adapted to your organization? If not why?
- Can it be easily adapted by agile teams in general? How / Why?

The question about the aspects does not require a detailed analysis, as the responses are fairly straightforward; the experts find measured aspects beneficial and provide one main suggestion as a new aspect suggestion: customer satisfaction. However, this aspect is not integrated into the model. Firstly, it may require a dedicated research to identify correct measures successfully analyze customer satisfaction. Secondly, this model aims to suggest a list of core measures to all type of software organizations but measuring customer satisfaction may require specialized measures according to the type and domain of customers. Therefore, this aspect is not included in the model. A future research may be conducted for the customer satisfaction aspect.

Responses for the questions about measures are divided into three main groups: First, the measures that the experts find unrelated to agile approaches, they neither use these, nor find them beneficial. Second, the measures that the experts do not use, but think that they might provide higher benefit then their costs, if implemented. Third, the measures that they actively use and find beneficial. First group of measures are labeled as “Not applicable”, second group is labeled as “Beneficial”, third group is labeled as “Common and beneficial”. After all the responses from all participants are evaluated, they are combined in a single table to evaluate the possible changes in the measures. This table is given in “APPENDIX C – Results and Evaluations of Exploratory Study 4”. If the results of a measure show mostly “Not applicable”, this measure is classified as optional for agile organizations. If one is beneficial but not commonly used, its description is adapted to reflect possible benefits and measuring methods.

3.5.5 Analysis and Results of the Exploratory Study 4

After the completion of the case study and analysis of the results, the answer for the research question 4 can be given.

RQ-4. Which aspects and core measures should be added to the model version one to update it to be applied in organization utilizing agile software development approaches?

The proposition for the research question 4 is that agile approaches might require a different set of measure. Validation method for this proposition is collecting opinions from domain experts. During the study, discussions with domain experts are held to find the answer for the research question 4. The final list of measures is given in Table 12 with the information about the aspects groups, aspects, entities, and attributes that the measures are aiming to quantify. The details about all the measures are also given in "Chapter 4: MEASUREMENT CAPABILITY ".

Table 12 List of measures after the exploratory study 4

Aspect Group	Aspect	Entity	Attribute	Sample Measures
Management	Planning	Planned work unit	Effort	Planned person-months from start to finish
			Duration	Planned days from start to finish
			Cost	Planned cost to be spent for the unit
			Risks	Weighted impact of anticipated risks
	Monitoring	Completed work unit	Effort	Person-months from start to finish
			Duration	Days from start to finish
			Cost	Cost to be spent for the unit
			Risks	Impact of occurred risks
		Product delivery cycle	Blockage	Days of blockage
			Flow efficiency	Percentage of actual work duration over total duration
		Work unit development process	Earned effort	Effort actualization percentage
			Earned duration	Duration actualization percentage
			Earned cost	Cost actualization percentage
		Project Artifacts (Optional for agile methods)	Conformance	Number of non-conformance
			Change	Number of proposed changes
				Number of accepted changes
		Corrective actions (Optional for agile methods)	Cost	Cost of correcting errors in artifacts
		Preventive actions	Cost	Cost of preventing errors in artifacts
		Artifact changes (Optional for agile methods)	Cost	Cost of implementing change
		Backlog burndown	Velocity	Team velocity
			Rate	Sprint burn rate
Technical	Procurement	Vendor Supplied Software	Size	Size of supplied product
			Duration	Time to delivery
			Unit cost	Cost of product per unit size
			Quality	Defects density
		Vendor Corrective Actions	Duration	Duration of defect resolution
	Production	Testing	Efficiency	Number of escaped defects
				Number of identified defects
			Coverage	Percentage of covered criteria over all criteria
		Defect Resolution	Effort	Defect resolution effort
			Cost	Internal failure cost
				External failure cost
		Code	Churn	Number of change in the code
			Reuse	Percentage of software code reused
		Product delivery cycle	Frequency	Time between each release
		Design	Complexity	Cyclomatic complexity

Aspect Group	Aspect	Entity	Attribute	Sample Measures
		Design component	Cohesion	Ratio of cohesive interactions (RCI) (Briand & Daly, 1998)
			Coupling	Message passing coupling (MPC) (Briand, Daly, & Wüst, 1999; Li & Henry, 1993)
Strategic	Improvement	Improvement process	Propositions	Number of proposed improvements per employee
			Involvement	Reflection meeting attendance over total employees
		Effort estimation process for work units	Efficiency	Percentage of actual effort over planned effort
		Duration estimation process for work units	Efficiency	Percentage of actual duration over planned duration
		Cost estimation process for work units	Efficiency	Percentage of actual cost over planned cost
		Non-value-added tasks	Cost	Cost of quality
	Foundation	Software component	Size	COSMIC FSM
		Measure	Normalization	Defect density

In addition to the identification of the measures, this case study also helps the researchers to understand how agile professionals evaluate distribution of different development life-cycle actions and what they understand from higher capability for measurement activities. Considering this insight, there are several changes implemented into the model in addition to the measure list. Naming of the aspects and aspect groups are changed to reflect agile perspectives. After the discussions, the generic practices are also updated by removing some of the practices aiming traditional software development approaches. In addition, the names of the different capability levels are updated accordingly. New capability level names and generic practice list is given in Table 13.

Table 13 Updated capability levels and generic practices

Measurement Capability Level	Generic Practices
MCL1 Measured	MCL1.GP1 Measures are identified during project lifecycle MCL1.GP2 Measures are collected during project lifecycle
MCL2 Coherent	MCL2.GP1 Collected measures are analyzed and evaluated during project lifecycle MCL2.GP2 Measures are communicated with the relevant stakeholders MCL2.GP3 Project stakeholders take necessary actions related to the measures
MCL3 Innovated	MCL3.GP1 Determine factors effecting measurement MCL3.GP2 Communicate applied measures with project participants and identify improvement opportunities related to them MCL3.GP3 Evaluate projects in organizational level by using collected measures

CHAPTER 4

MEASUREMENT CAPABILITY MODEL

Measurement Capability Model aims to present a practical approach to organizations to evaluate their measurement capabilities. The model aims to fill the gap in the literature by providing a flexible approach focusing on measures instead of processes. With this model, organizations utilizing agile lifecycles instead of process-based development can focus on specific measures and specific areas of their lifecycles according to their needs.

The model guides its users by grouping the measures according to the areas they are related in development lifecycle. The building blocks of the model are the areas that are called as aspects, which include core measures to be used in these areas. Organizations are expected to work on all core measures under an aspect to reach a certain capability for that aspect. The term aspect is used by Ozcan-Top and Demirörs as “sets of interrelated and interacting activities” to represent traditional processes in an agile approach (Ozcan-Top & Demirörs, 2015). The model includes three groups of aspects: Management aspects, Technical aspects, and Strategic aspects. The purpose of this grouping is to provide a basic guidance to the organizations while deciding which aspects they want to assess.

Organizations should evaluate their measures by using general practices defined in the model. A capability level is determined for each measure and then measurement capability level of an aspect is calculated from the levels of the measures under that aspect. Capability levels are determined according to the satisfaction level of the general practices. There are three capability levels: measured, coherent, and innovated. These levels help organizations to understand current situation of their measures. Organizations can also use the practices as a path to improve their measures.

An organization can choose to focus on any aspect by working on its defined core measures. For example, if project management division in a company want to use the model they may only focus on core measures under aspects of management group. Although the different core measures defined in different aspects may have commonalities, organizations may have requirements to handle the measures independently.

Two main components of the model are explained in the remaining parts of this chapter. The first component includes aspects, and second one includes capability levels and related practices. Organizations first need to determine which aspects they aim to assess and then proceed by checking the practices for each measure under this aspect. As a result, corresponding capability levels for the aspects can be calculated.

To be on a capability level, a measure needs to satisfy at least one practice of that level, and all practices of the levels below.

In addition to the two main components of the model, there are two other concepts related to the model that are explained in the following parts; one of them is measurement comparability. Organizations require a denominator measure to be used with other measures to gain the ability of comparing them with other projects, teams, or organizations. Developers of the model suggest using COSMIC functional size as denominator, which is used in the first exploratory case study of this research. However, any proven method may be used for this purpose as long as it represents the size of the software objectively. Even though software size is included in the model as a foundation measure, it is crucial to mention it as an additional concept to express its importance. Second concept is related to using the results of the model to improve organizational measurement capability. Even though the model does not include an assessment process, organizations can use it to evaluate their processes and identify improvement opportunities to improve their measures using the results of the model.

4.1 Aspects and core measures:

The model lists the aspects under three groups according to their purposes in the software development lifecycle:

Management: This group includes aspects related to the managerial parts of a software project: planning and monitoring.

Technical: This group includes technical aspects that add value to the resulting products within a project scope: procurement and production.

Strategic: This group includes aspects that effects strategic workings of organizations from software development point of view and are usually managed centrally: improvement and foundation.

Aspects include different entities and related attributes that need to be measured. There are also sample measures that provide guidance to organizations to determine what they need to measure for selected aspects. All aspect groups, aspects, entities, attributes, and sample measures are given in Table 14. Another detailed list is given in "APPENDIX D –List of Aspects, Entities, Attributes, and Sample Measures" with the definitions of entities. In "APPENDIX E – Sample Measures and Related Information" the measures are explained with their justifications, purpose, measurement method, formula, and measure scale. Aspects that will be used to assess organizations measurement capability can be selected according to the organization's domains, needs, and goals. Total number and type of measures used in an organization may be quite large; in the model a set of sample measures are suggested. Additional measures can be determined according to the organizations needs and requirements. In addition, some entities are noted as "optional for agile methods". Most agile approaches do not utilize these entities; therefore, agile organizations may omit them if they do not use them. It should also be noted that because of the nature of agile approaches it may not be possible to measure some attributes for the duration of the whole life of a project.

For example, estimating total effort required for a project may be hard as project scope may change during the lifecycle. Therefore, agile organizations may use defined development cycles, like sprints, as work units for related measures.

Table 14 Aspects and core measures

Aspect Group	Aspect	Entity	Attribute	Sample Measures
Management	Planning	Planned work unit	Effort	Planned person-months from start to finish
			Duration	Planned days from start to finish
			Cost	Planned cost to be spent for the unit
			Risks	Weighted impact of anticipated risks
	Monitoring	Completed work unit	Effort	Person-months from start to finish
			Duration	Days from start to finish
			Cost	Cost to be spent for the unit
			Risks	Impact of occurred risks
		Product delivery cycle	Blockage	Days of blockage
			Flow efficiency	Percentage of actual work duration over total duration
		Work unit development process	Earned effort	Effort actualization percentage
			Earned duration	Duration actualization percentage
			Earned cost	Cost actualization percentage
		Project Artifacts (Optional for agile methods)	Conformance	Number of non-conformance
			Change	Number of proposed changes
				Number of accepted changes
		Corrective actions (Optional for agile methods)	Cost	Cost of correcting errors in artifacts
		Preventive actions	Cost	Cost of preventing errors in artifacts
		Artifact changes (Optional for agile methods)	Cost	Cost of implementing change
		Backlog burndown	Velocity	Team velocity
			Rate	Sprint burn rate
Technical	Procurement	Vendor Supplied Software	Size	Size of supplied product
			Duration	Time to delivery
			Unit cost	Cost of product per unit size
			Quality	Defects density

Aspect Group	Aspect	Entity	Attribute	Sample Measures
		Vendor Corrective Actions	Duration	Duration of defect resolution
	Production	Testing	Efficiency	Number of escaped defects
				Number of identified defects
			Coverage	Percentage of covered criteria over all criteria
		Defect Resolution	Effort	Defect resolution effort
			Cost	Internal failure cost
				External failure cost
		Code	Churn	Number of change in the code
			Reuse	Percentage of software code reused
		Product delivery cycle	Frequency	Time between each release
		Design	Complexity	Cyclomatic complexity
		Design component	Cohesion	Ratio of cohesive interactions (RCI) (Briand et al. (1998))
			Coupling	Message passing coupling (MPC) (Li and Henry 1993, Briand et al. 1999b)
Strategic	Improvement	Improvement process	Propositions	Number of proposed improvements per employee
			Involvement	Reflection meeting attendance over total employees
		Effort estimation process for work units	Efficiency	Percentage of actual effort over planned effort
		Duration estimation process for work units	Efficiency	Percentage of actual duration over planned duration
		Cost estimation process for work units	Efficiency	Percentage of actual cost over planned cost
		Non-value-added tasks	Cost	Cost of quality
	Foundation	Software component	Size	COSMIC FSM
		Measure	Normalizatio n	Defect density

Measures include estimations and measurements. Estimations are measures that are conducted before occurrence of the real value. Organizations need to use one of the several available estimation methods to estimate a value for that measure to be used in their planning activities. Measurements represent occurred real values that are collected from occurred events. Estimations and measurements need to be used together to understand estimation efficiency.

Organizations can select any number of aspects from any of the three groups according to their needs. Capability levels are determined for the measures in the selected aspects using general practices defined under each capability level. Capability levels and general practices are explained in the next section.

4.2 Capability levels

Model defines a three-level capability scale. The reason of defining three levels instead of traditional five levels is, first to increase the practicality of the model by reducing number of practices, and second to provide a more direct dynamic approach with lesser levels than process-centric approaches. Also, higher maturity levels defined in models like CMMI and ISO 33000 aim measuring and statistically improving processes. These concepts are not directly applicable for agile organization and it is not meaningful to expect them from agile approaches. A similar three level structure is also used in AgilityMod (Top, 2014). AgilityMod defines four levels including a level 0- not implemented; level 1 – Ad-Hoc, level 2 – Lean, and level 3 – Effective. Three levels of the measurement capability model are explained respectively in the following parts.

Measurement Capability Level 1 - MCL1 - Measured: In the first level of capability, organizations conduct measurements but not necessarily uniquely in all units. Measurements are usually conducted as ad-hoc analysis. As the collections methods and rules are not common across the organization, the results might not be comparable among different units and projects. Rules and applications usually stay in the specific project's or team's boundaries. In this level, measures are identified and collected.

Measurement Capability Level 2 - MCL2 - Coherent: In this level, organizations collect the measures as in the first level, however; the measures are analyzed, evaluated, communicated, and necessary actions about them are taken; rules and applications. The results of measurements are actively used as feedback in the project or team level.

Measurement Capability Level 3 - MCL3 - Improved: In the highest level, collected measures are evaluated and used in the organizational level. The factors effecting the measures are identified, the results of the measurement analyses are used to identify improvement opportunities related to them. In this level the measures go beyond the project borders and they are evaluated in the organizational level.

To determine the capability level of a measure, the model defines generic practices for each measurement capability level. These practices are listed in Table 15. In traditional process centric improvement models, mainly based on CMMI (CMMI Product Team, 2010) and ISO 33000 (ISO/IEC 33001, 2015), there are also specific practices defined for each process area. This model does not have them because of its measure-focused approach. It is not possible to define specific practices for each measure, especially as the model lets the organizations adapt the list of measures according to their needs, goals, and practices if necessary. This adoption also includes adding new measures into the aspects. Defining specific practices for each measure will lessen the effectiveness and practicality of the model by preventing organizations to add new measures. Defined generic practices for each capability level are applicable to all measures examined during an assessment with this model.

Table 15 Measurement capability levels and generic practices

Measurement Capability Level	Generic Practices
MCL1 Measured	MCL1.GP1 Measures are identified during project lifecycle MCL1.GP2 Measures are collected during project lifecycle
MCL2 Coherent	MCL2.GP1 Collected measures are analyzed and evaluated during project lifecycle MCL2.GP2 Measures are communicated with the relevant stakeholders MCL2.GP3 Project stakeholders take necessary actions related to the measures
MCL3 Innovated	MCL3.GP1 Determine factors effecting measurement MCL3.GP2 Communicate applied measures with project participants and identify improvement opportunities related to them MCL3.GP3 Evaluate projects in organizational level by using collected measures

To determine capability of a measure by using generic practices, a measure should be evaluated according to the practices. If a measure is not used, therefore does not satisfy any practice defined in level 1, it should be classified as NS – Not satisfied. It is named as NS instead of level 0 as it is not possible to assign a level if a measure does not exist. If a measure satisfies only some practices in level 1, it is classified as capability level 1. If a measure satisfies all practices in level 1 and some practices in level 2, it is classified as level 2. If a measure satisfies all practices in level 1 and 2, and satisfies some practices in level 3, then it is classified as level 3.

To determine capability of an aspect, at first capabilities of all the measures under that aspect should be determined. After that following the rule set defined in Table 16 for all the measures under that aspect, the capability level of the aspect can be determined.

Table 16 Determining capability level of an aspect

Situation	Result	Comment
All measures are NS	Aspect is NS	
All measures are level 1 or NA, at least one measure is level 1	Aspect is level 1	
At least one measure is level 2 or 3, and at least one is level 1 or NS	Aspect is level 1 or level 2	If there is potential for organization to easily improve lower capabilities, level 2 should be assigned
All measures are level 2 or 3 with at least one level 2	Aspect is level 2	
All measures are level 3	Aspect is level 3	

4.3 Measurement Comparability

Normalization of measures is an important activity for the organizations, as it lets them have the ability to compare their measures between different projects or with different organizations. Being able to benchmark their measures, they can easily identify improvement opportunities.

The model lists normalized measures as an attribute of any measure under the foundation aspect. It is also an important component for measurement activities in an organization; therefore, comparability of the measures is especially emphasized in the model.

Most measures cannot be compared directly as they depend to the project from which they are collected. Organizations need to use a metric that reflects the size of the project objectively. This metric then can be used as a denominator for the other measures, to make them comparable. Although there are various alternatives for software size such as the functional size, lines of code, number of requirements, story points, and total effort; most of them are not objective measures. An objective representation of the software should be used as a size measure to be used for normalization. Suggested measure is functional size of the software (ISO/IEC 15939, 2017).

4.4 Measurement Capability Assessment and Improvement:

Main aim of the measurement capability model is to provide a practical model for the organizations requiring a flexible approach. To remain true to this aim, other than the core measures under the aspects and general practices, the model does not define a clear-cut assessment process. Instead, there is a guideline organizations can follow to apply the model. If necessary, this process can be modified to suit the needs of the organizations. The results of an assessment should demonstrate the organizations their position in the capability scale. Using this result and main components of the model, measures and practices, an organization can define an improvement map to follow to increase its measurement capability.

An assessment may include introduction to the model, selection of aspects, assessing organization with general practices, validation of findings, and reporting findings. After the assessment, if needed, an improvement initiative may be started. The improvement initiative can include an improvement plan that is structured with the help of general practices, implementation of the improvement plan, pilot application of the improvements, dissemination of the improvements, and independent audits. Both assessment and improvement processes are applied in the first exploratory case study with the version one of the model; however, the scope of this research does not include implementation process of improvements. Suggested steps for the assessment process are explained below.

Introduction to the model: This step should be conducted if the model is new for the organization or for the key personnel who should be involved during the assessment.

Depending on the purpose of the assessment, size of the organization, and organization's history with the model, the scope of this step may range from short meetings with a few key personnel to large training seminars to introduce the model to the organization. Main aims of this step are:

- To familiarize the organization with the model
- To determine organization's motivation
- To determine purpose and scope of the assessment
- To decide necessary commitments and required data for the assessment

Selection of the aspects: With the help of organization's motivation, purpose, and scope, targeted aspects should be identified. Organizations usually start an improvement initiative to handle a specific problem or to reach a specific goal. Aspects should be identified considering this motivation. If the organization wants to assess its general measurement capability, then all aspects should be included in the assessment. When an aspect is selected, organization needs to use all core measures listed under that aspect; however, they can also include any additional measures according to their needs.

Assessing organization: Aspects selected in the previous step should be examined to find out their capability level by the guidance of generic practices. Examination can be carried out by conducting interviews with practitioners of the processes and examining supporting proofs. By examining the proofs, evaluation is carried out according to the rules defined in the model and capability levels of the assets are assigned.

Validation of findings: After the assessment, initial findings may be communicated with the key personnel in the organization and opinions on the findings may be requested from the stakeholders for evaluation. If there are disagreements on some points, they need to be resolved and related changes should be reflected on the assessment results.

Reporting findings: After the validation step, final report may be prepared reflecting current capability levels of the aspects. It may be shared with the project sponsor from the organization. Considering the findings of the report, organization may decide to conduct an improvement initiative. Core measures and generic practices can be used to determine necessary improvement steps.

Improvement plan: If required by the organization, an improvement plan can be prepared by using the results of the assessment and components of the model. Sample measures given in the model can be used to implement new measures into the processes if necessity is observed. Generic practices can also help to understand how the measures can be used to increase the benefit provided by them. After the identification of the improvements, a plan should be prepared to integrate them into organizational processes.

Implementation of the improvement plan: After the plan is prepared, it should be implemented by updating necessary processes, by updating necessary software if necessary, and providing related training to the participants. Implementation does not include putting the measure into application; it is only related to preparing the organization for the application.

Pilot application of the improvements: Before applying the new rules and processes in the whole organization, a pilot application may be necessary, especially for larger organizations. Pilot application may be conducted in a selected project or with a selected team. The purpose is to observe possible shortcomings of the improvement and make sure that they really provide additional benefit to the status-quo. If the results are successful, organization-wide application can be started.

Dissemination of the improvements: If pilot application provides positive evidence about the success of the improvements, changes can be disseminated through the whole organization.

Independent audits: After the implementation of the improvements, inspecting the application of the changes may be necessary to detect possible miss-applications. Conducting these inspections independently is preferable, as a third party can objectively evaluate the participants.

CHAPTER 5

EXPLANATORY STUDY

After the model version two is structured, an explanatory case study is conducted to apply the model in real organizations to answer the last main research question. The explanatory case study is conducted with four different organizations that are utilizing agile development approaches. After the application of the model, the results are discussed with the participants from these organizations during an interview to understand their opinions about the usefulness of the results and effectiveness of the model.

5.1 Research Questions in Explanatory Study

Explanatory study aims to provide answers to research questions 5, 5.1, 5.2, 5.3.

RQ-5. How useful is the model for software organizations, specifically for those using agile software development approaches?

Proposition 5: Main motivation for the development of the model is to help organization utilizing agile development approaches. Through the development of the model several case studies conducted, some of them included agile methodologies; however, in none of them the focus was directly related to agile approaches. To understand the effectiveness of the method with these organizations, we need to find out how useful it is when applied in these organizations.

Validation method for proposition 5: Research questions 5.1, 5.2, and 5.3 examine respectively; whether the model can clearly reflect current measurement capabilities of participant organizations, whether it can successfully identify improvement opportunities, and whether it can provide new information to the organization. If, we can observe positive answers to all of these questions after the study, we can conclude that the model is useful for organizations using agile development approaches as it can help them to understand their status regarding measurement capability. It can help them to identify improvement opportunities and provides them new information that they did not know before the application of the model.

RQ-5.1. How clearly does the model reflect the state of the organization with respect to their measurement capability?

Proposition 5.1: Measurement capability model is expected to help organizations to evaluate their current measurement capability. This evaluation should reflect the real situation in the organization utilizing agile approaches.

Validation method for proposition 5.1: After using the model in the case study with the participant organizations and assessing their current state, an interview is conducted with them to understand their opinions about the results. To be able to accept proposition 5.1, during the interview the participants should agree the assessment results of the model and state their opinion about the degree of clarity of the results.

RQ-5.2. How fully do the identified improvement opportunities capture organization's potential?

Proposition 5.2: Measurement capability model is expected to help organizations to identify opportunities to improve their current measurement capability. These improvement opportunities should be consistent with the observed potential of the organization.

Validation method for proposition 5.2: After using the model in the case study with the participant organizations and determining improvement opportunities, an interview is conducted with them to understand their opinions about the results. To be able to accept proposition 5.2, during the interview the participants should state how much the identified improvement opportunities are consistent with their observations about the organization's potential.

RQ-5.3. How much new information does the model provide to organizations?

Proposition 5.3: Measurement capability model is expected to provide information to organizations that they were not aware of before.

Validation method for proposition 5.3: After using the model in the case study with the participant organizations and reporting the findings, an interview is conducted with them to understand their opinions about the results. To be able to accept proposition 5.3, during the interview the participants should identify new information that they gain by the use of the model.

5.2 Activity Planning in Explanatory Study

The activities that are planned for the explanatory case study are listed below:

- Case selection: Aim of the explanatory case study is to provide an answer for the research questions 5, 5.1, 5.2, and 5.3, therefore we need organizations using agile development approaches to apply the method. After the application there need to be interviews to discuss the findings with the representatives of the organizations. Meaning that we need several organizations developing software with agile approaches and we need participants to cooperate with us to evaluate the findings. It would provide an advantage to apply the model in organizations with different sizes to observe its applicability with changing size. To represent different sizes, we aim to find at least one small-sized, one medium-sized, and one large-sized organization.

- Establishing case study environment: Case study is expected to be conducted with the representatives of the organization. If the organization is located in another city, then the meetings could be conducted by using online meeting software.

- Identifying information sources: Information sources of the case study are the participants representing the organizations, who are expected to have the knowledge about the measurement activities and practices of the organization in managerial level.

- Application of the method: The method should be applied in the participating organizations with the help of participants. Then organizations' current measurement capabilities and improvement opportunities are identified and shared with the participants. After that, there is an interview with the participants by using pre-prepared list of questions to discuss the findings. The responses of the participants are recorded onto discussion sheets.

- Analyzing the conduct of the case study: Findings of the case study will be inspected and the results will be evaluated to discuss the validity of the proposition 5, 5.1, 5.2, and 5.3 to find answers for the related research questions.

5.3 Mitigation of Threats to Validity in Explanatory Study

Threats to internal validity: Three main threats to the internal validity of the case study are identified and related mitigatory actions are planned before the start of the study. These threats are related to the selection of the participants, positive inclination of the participants, and collection of data.

There is a threat about the selection of the participants as the knowledge and opinions of the participants, who are helping to assess the capability, can greatly affect the results. To mitigate this threat, the researcher aims to conduct the study with participants having managerial roles in the organizations. It may be a project manager or s quality manager. Specialized employees of organizations, like developers or analysts, may have biased opinions or they may not know about the measurement activities, whereas, project or quality managers expected to have a command over whole processes.

Positive inclination of the participants is another important threat for the study. Participants may incline to give wrong answers to represent themselves as more capable than they already are. The researcher provides a list of improvement ideas to the participants by using the results of the assessment to increase their measurement capability. Communicating this output of the study with the participants is expected to motivate them to provide realistic responses during the case study.

Collection of data is another factor for the internal validity of this case study. The data collection process should be carefully designed to reach an objective conclusion after the study. To ensure the quality of the data, the researcher uses the general practices defined in the model to evaluate all the measures without any subjective input. In addition, the results are discussed with the participants by using pre-determined guideline questions to test that the results represent real situation in the organization.

Threats to external validity: Two main threats to the external validity of this study are related to the representation of the population and representation of the setting. To prevent the first threat and to represent the population as comprehensive as possible, the researcher aims to represent the population from the size perspective. It is aimed to find at least one participant from each of the three main size groups; a small organization, a medium size organization, and a large organization. To prevent the threat of misrepresentation of the application setting, the researcher uses real organizational data supplied by the participants, instead of using a toy setting.

Threats to construct validity: Explanation of the construct and overlooked factors are the threats to the validity of the construct of this study. Explanation of the construct represents the risk of using wrongful measure definitions while evaluating the results of the study. To prevent this threat, the researcher uses the results of post-discussions conducted with the participants. Pre-determined discussion questions will be used in these discussions. These questions aim to evaluate the results with the participants. Explanations from the participants' perspective should help to clarify the explanation of the construct of the study.

Overlooked factors can also affect the results of the case study. Even though before the explanatory case study, there were other studies aiming to determine organizational measurement capability, there might be some missing factors. To eliminate this threat, a question is added into the list of post-discussion questions: "What else would you consider as important while determining measurement capability?" If the participants think or know another factor that can affect the results, they have the opportunity to share it with this question. The researcher will evaluate suggested factors at the end of the study if there is any.

Threats to the reliability: Operational judgement and reliability of implementation are two threats to the reliability of the study. Operational judgement represents the subjectivity of the researcher who is conducting the study. It is related whether two different researchers can reach to the same conclusion in the same situation or not. The researcher aims to mitigate this by the help of post-discussion sessions. During the discussion, the participants should examine the results and state their opinions about them. If there is a subjectivity introduced by the researcher into the assessment, participants should be able to identify them in this part.

Reliability of the implementation is related to the implementation differences among the cases in the study. To prevent this threat the researcher follows strict guidelines with all cases; he uses same questions during the assessment and utilizes same discussion questions during post discussions.

5.4 Explanatory Study Design and Execution

Case Selection in Explanatory Study: For the selection of the case, we have three criteria: organizations using agile development approaches, cooperation of the management, and at least three organizations with different sizes. To find the organizations we use our network and make contacts with a few suitable organizations. Four organizations accept our proposal and they are convinced that the findings will be beneficial for them. They all utilize agile approaches by using scrum framework.

- First organization is small sized with around 20 employees. They develop medical information systems and recently assessed as level 2 according to ISO 15504. We conduct the study with someone from the quality department who has involved in all process development and improvement activities in the organization and has extensive knowledge about the used measures.

- Second organization is another small sized organization with around 15 employees. The participant we conducted the case study with is the project manager responsible from a team of 5 employees. She currently aims to assess and improve measurement processes; therefore, she is motivated to participate and use the results of the study in the organization. Her responses mostly include measures related to this team. This organization does not have any official quality assessment.

- Third organization is a medium seized organization working in industrial software and automation sector with 50 employees in total. The participant who provides measurement data is the quality manager of the organization. The responses provided by the participant represent works and processes of teams totaling around 40 software developers. Remaining 10 employees are working in industrial automation. This organization recently acquired a level 2 certificate according to ISO 15504.

- Fourth organization is a large organization operating in telecommunication sector with over 4000 employees in their technology division. From those 4000, around 1000 employees are developing software and around 350 of them using agile approaches. The participant representing this organization is a team leader managing two different teams with a total of 13 team members.

Case Study Environment in Explanatory Study: The organizations that accept to participate in the study are located in different cities. The method to apply this study does not require examining detailed evidence, especially because the organizations are applying agile methodologies and they do not keep much documentation. All the data is collected from the participants. Therefore, the study is conducted as online meetings utilizing online meeting software. There are two sessions conducted with the participants, in the first session the model is applied to the organization and capability levels are determined for all available aspects. These sessions take approximately one hour each. In the second session, the results are shared with the participants and post-study interviews are conducted. These sessions take approximately 45 minutes each. For the second and fourth organizations, sessions are conducted in separate dates, whereas for the first and third organization's sessions are sequential. The results of measurement capability evaluations and notes of the interviews are recorded in the corresponding forms.

Information Sources in Explanatory Study: Information sources for the case study are the participants form the organization. They provide information about their organizations during the evaluation of their measurement capability by replying the questions regarding the measures and generic practices. They also provide information during the post-study interviews about their opinions about the model, its results, and its compatibility to their organizations.

Application of the Method in Explanatory Study: Explanatory study is started by the evaluations of measurement capabilities of the organizations. The results of the evaluations for each organization are given in "APPENDIX F – Evaluation Results in the Explanatory Study". Then these results are shared with the participants and a post-study interview is conducted with each other. The questions used in the interview are given below.

- Do you think the results reflect the measurement capability of your organization?
- Do the results of assessment correspond to your actual observations in the organization? What is the deviation and/or what is the correspondence?
- Do the provided improvement opportunities correspond to your actual observations in the organization?
- Do you find the improvement opportunities beneficial? If so, which results are most beneficial?
- Are the improvement opportunities feasible? If so, will you plan to change your practices accordingly in the near future? If not, why not?
- Have the results provided new information about your organization, anything you did not know before? If so, what is it?
- Are there any aspects that are not covered by the model but are important for your organization to determine the measurement capability and if so what are they?
- Are there any additional measures that are not covered by the model but are important for your organization to determine measurement capability and if so what are they?
- What are your opinions about the generalizability of the assessment and identification of improvement opportunities provided by the method for agile organizations in general?
- Are there any other points you would like to add about the method and its applications?

The responses collected from the participants for each question are given and examined in the next section.

5.5 Explanatory Study Analysis and Results

After the completion of the explanatory case study, the results are evaluated to find answers to the research question determined at the beginning of the study. To determine the answers, responses of the participants in post-study interview are used. All questions and answers are given in "APPENDIX G – Answers for the Discussion Questions in Explanatory Study". Related questions are also given under each research question together with tables that include representative answers given by the participants.

RQ-5.1. How clearly does the model reflect the state of the organization with respect to their measurement capability?

To provide answer for this research question we need to analyze the responses of the participants during the interview about the results of the evaluation conducted by using the model. There are two questions aiming to find out the answer. These questions and representative responses provided for them are given in Table 17.

Table 17 Representative answers for RQ-5.1

Do you think the results reflect the measurement capability of your organization?	
Organization 1	Accurately reflects the situation
Organization 2	It reflects correctly
Organization 3	It reflects my organization correctly
Organization 4	Results are correct
Do the results of assessment correspond to your actual observations in the organization? What is the deviation and/or what is the correspondence?	
Organization 1	Our organization is level 2 according to ISO 15504... results are consistent...at this level…also consistent with my personal observations
Organization 2	Results are consistent with my observations
Organization 3	My observations are represented by the results
Organization 4	Results are consistent with my experience

Considering the positive responses from all the participants, we can state that the participants evaluate the results as a clear representation of the measurement capabilities of their organizations. Organization 1 has a level 2 ISO 15504 certificate, they can use this as a benchmark and state that the results are compatible with each other. For the second question, the participants make general statements about the correspondence of the evaluation results to their status; seemingly, they agree on the evaluations results in general.

By using the positive responses from the participants, we can state that "*the model quite clearly reflects the state of the organization with respect to their measurement capability; its evaluation results are parallel with the opinions of the participants.*"

RQ-5.2. How fully do the identified improvement opportunities capture organization's potential?

In the interview conducted with the participants three questions aimed to find out the answer for this research question. The questions and provided responses are given in Table 18.

Table 18 Representative answers for RQ-5.2

Do the provided improvement opportunities correspond to your actual observations in the organization?	
Organization 1	The suggestions reflect my thoughts...some suggestions that we did not think about before
Organization 2	Suggestions correspond to our observations on base level...we need to investigate the feasibility of new methods to understand whether they worth the required effort to implement. If the investigation provides positive results, we want to implement the suggestions
Organization 3	Some of the suggested measures were already in our agenda...there are suggested measures that we do not practice and they seem applicable...generic practices of level 3, I don't know whether they worth the required effort
Organization 4	Suggested improvements are applicable
Do you find the improvement opportunities beneficial? If so, which results are most beneficial?	
Organization 1	There are beneficial suggestions among them...there are also one or two that I do not think can help us
Organization 2	I find them beneficial...I find the measures that are related to production most beneficial, we would like to adopt them
Organization 3	Especially the measures related to test coverage are the points that we lack...We need to improve ourselves by considering these suggestions
Organization 4	It would be beneficial to select most applicable ones and to start with them
Are the improvement opportunities feasible? If so, will you plan to change your practices accordingly in the near future? If not, why not?	
Organization 1	All of the suggestions are feasible
Organization 2	I plan to implement the suggestions
Organization 3	We wish to implement the measures related to code and test; we think they can be useful...There are also some measures, which we do not plan to implement...Maybe after implementing other measures we may feel the need to change this situation
Organization 4	I definitely think to implement the measures that I stated as applicable in the previous question...I may not implement some of the suggested improvement ideas in the first step

These responses show us that the suggestions generally correspond with the participants observations and they find them useful. Some suggestions are found as useful by one participant but same suggestion is found as not important by another participant. For example, the participant from Organization 3 has doubts about applying the generic practices of level 3, whereas, the participant from Organization 1 especially emphasizes their value. This can be a sign that organizations with different needs and experiences can gain benefit from different outputs of the model.

For the last question, participants state that they find the improvement suggestions mostly feasible. They prioritize the suggestions and want to apply some of them immediately while waiting for some organizational change for others. The participant from Organization 4 states that there are some suggestions that are not applicable. Organization 4 is a large-sized organization and its participant is the only one finding a measure non-applicable. Even though its overall measurement capability is similar to other organizations according to the model, a larger organization may have more experience with measures. The statement of the participant: “they [some measures] are not meaningful for our organization” may be a result of this experience.

By considering all responses, we can state that “*identified improvement opportunities generally correspond with the observations of the organizational participants; they agree that the model provides beneficial and applicable opportunities to the organizations. These benefits may change according to organization’s needs and experiences; however, all participants of this study find something valuable among them. Identified improvement opportunities capture organization’s potential according to their needs and experiences.*”

RQ-5.3. How much new information does the model provide to organizations?

One question aiming to answer this research question is given in Table 19 together with the received responses.

Table 19 Representative answers for RQ-5.3

Have the results provided new information about your organization, anything you did not know before? If so, what is it?	
Organization 1	We learned new measures and new ways to examine them. We learned how to evaluate ourselves
Organization 2	We observed the points of improvements. Other than the improvements, there are some measures in the provided list that I did not know before
Organization 3	It is useful as it shows our shortcomings in a systematic way...I did not see any new measure that I personally did not know before...these results should be helpful while communicating with senior management.
Organization 4	Other than [Cost of quality], there are not any new measure that I did not know before...the model helped us to realize some points to improve ourselves

When the responses are examined, all participants learn something new from the model, in addition to the suggested improvements. Participants 3 and 4 state that they knew all the measures before, but gained new insights from the model. Participant 3 finds the model especially useful for communicating with upper management. Participant 4 claims that it is the first time that she comes across cost of quality measure as an applicable option.

Participants 1 and 2 state that in addition to the improvements suggested by the model, they learned new measures. When we take the sizes of the organizations into account, we can say that “*while the model can provide new information to all organizations during the case study, smaller organizations may gain more benefit as they may have less experience with measurement.*”

RQ-5. How useful is the model for software organizations, specifically for those using agile software development approaches?

The answers to the previous research questions provide information for this research question. Moreover, the answers given to the last questions during the interviews are also related to this research question. The last questions and representative answers are given in Table 20.

Table 20 Representative answers for RQ-5

Are there any aspects that are not covered by the model but are important for your organization to determine the measurement capability and if so what are they?	
Organization 1	There may be measures related to customer satisfaction...there might be an aspect to reflect...competitiveness
Organization 2	There might be measures related to the personnel
Organization 3	I find the scope quite adequate and operable...there might be additional components for them, like training or human resources, but current version is adequate for essential needs
Organization 4	There might be an aspect related to organizational agility
Are there any additional measures that are not covered by the model but are important for your organization to determine measurement capability and if so what are they?	
Organization 1	While analyzing the backlog, understandability of requirements and refactoring time can be added
Organization 2	I would like to measure cross-functionality
Organization 3	current context is adequate for fundamental needs
Organization 4	Measures related to how defect resolutions reach to customer, how fast defect resolutions are, and automatic testing can be included...Measures focusing on extreme programming and DevOps can also be included
What are your opinions about the generalizability of the assessment and identification of improvement opportunities provided by the method for agile organizations in general?	
Organization 1	It can be generalized for agile organizations. Especially small-sized organizations or organizations just starting to focus on maturity can gain quite benefit
Organization 2	Software organizations utilizing agile approaches and do not exceed 50 people can use this model...In higher levels I am not sure...but I only worked on smaller organizations
Organization 3	This model lists fundamental works...If organizations start with a more detailed process improvement model they cannot move forward...with this model...they can identify the problems. This model can be used as a regular feedback tool.
Organization 4	Model may be useful in general...in addition, it can be used as a checklist by applying at certain times and regularly evaluating the results
Are there any other points you would like to add about the method and its applications?	
Organization 1	The definitions may be improved
Organization 2	Currently there is nothing I would like to add
Organization 3	The model has 3 levels, this is a realistic approach for me...3 level approach of this model is more operable...for the maintainability and dissemination of this model some processes may be defined
Organization 4	I think model includes necessary components for its purpose and adequate for its aimed scope

Considering the answers of the previous research questions and the responses given for the last four interview questions, we can conclude that participants of the case study agree that the model is useful for the organizations utilizing agile approaches. Small and middle-sized organizations in the study find the model quite beneficial and want to apply the findings and improve their practices according to the generic practices of the model. The participant of the large organization finds the model beneficial; however, she is not as enthusiastic as the other participants are.

Considering the responses, we can answer the research question as "*The model is useful for software organizations utilizing agile approaches; it evaluates their current situation, suggest feasible improvement opportunities, and provides new information. However, it should be noted that among the participants of the case study, small and medium sized organizations seem to receive more benefit from the model.*"

In addition to help to answer the research questions, the interview also provides us improvement ideas for the model. When the answers are combined, the suggestions can be listed as below.

- Additional aspects
- Customer satisfaction
- Competitiveness
- Employee performance
- Training
- Agility
- Additional measure
- Understandability of requirements
- Refactoring time
- Cross-functionality
- Defect resolution duration
- Automatic testing
- Extreme programming related measures
- DevOps related measures
- Others
- Best practices
- Maintainability process for the model
- Dissemination process for the model

This list of improvement ideas can be used to improve the process in future research. Especially a process to disseminate the findings and improvements of the model, and a process to maintain organizational measurement capability may be useful to improve the benefits of the model. Suggested aspects need to be evaluated to decide whether they fit into the scope of the model or not. Suggested measures can also be evaluated; however, the model suggest the organizations to add their own measures in the list and evaluate them with the generic practices, even though it is not applied during the case studies.

CHAPTER 6

CONCLUSION

A software measurement capability model is described in this study. The model is investigated and tested with exploratory and explanatory case studies to understand its usage and results. The process of the research is given in Figure 13. At the beginning of the research, planned process was to conduct exploratory case studies to formulate the model and apply it in an explanatory case study. Number of exploratory studies were not certain, as according to the findings new studies may be required. At the end, all research questions are answered as a result of one action research cycle, four exploratory case studies, and one explanatory case study. Initial components of model version one is structured during the action research, then they are applied in real life cases during exploratory case studies one, two, and three to find out answers for related research questions. During exploratory study four, the model is updated to version two, a model that is applicable for agile organizations. Finally, it is applied in agile organizations during the explanatory case study.

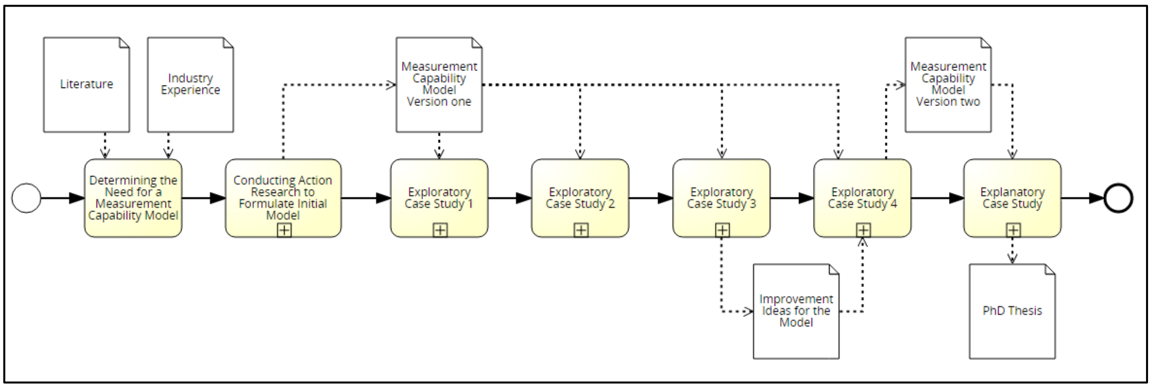

Figure 13 Overview of the research process followed

This chapter includes discussion about the findings of the research, explains the limitations of the study, provides lessons learned during the conduct of case studies, and suggests future research ideas to improve the contributions to the literature.

6.1 Contributions

During this research, we observe that using a measurement capability model can provide great benefit for the organization utilizing agile approaches. Organizations can use this model to analyze their measurement practices to determine their capabilities and identify improvement points to increase it. The case studies also provide us a snapchat of the measurement activities in the participating organizations. Some of these organizations are in great need of a measurement capability analysis. During the studies, they also observed this need and stated their satisfaction about the results.

The results of case studies show some measures commonly lacking in the subject organizations. Specifically one of the main problems is observed as the lack of and need for a ratio scale size measures that can be used to objectively measure the size of the software and provide the organization the ability to normalize and compare measures between projects, teams, or organizations. The results of the first exploratory case studies show that implementing this measure can increase organizational measurement capability significantly.

During this research, we develop a first measurement capability model in the literature, which can be used with agile approaches. To understand the needs and requirements of agile organizations from the measurement viewpoint we conduct a case study with highly experienced practitioners of agile approaches. The findings of this study is integrated to the model to build the MCM v2. During the explanatory case study, the model is applied in four organizations utilizing agile approaches and the findings are discussed with the experienced participants from these organizations. The results of this study demonstrate that organizations with agile perspectives can receive significant benefit from this model. Especially small and medium sized organizations can use it as a road map to identify possible measures and implement them into their practices.

We apply the model in real life settings and observe the results. In the five independent case studies, we examine 11 different cases in 7 different organizations. All organizations, which the results are discussed with, agree the results of the model and find them beneficial. In exploratory case study one, we observe the effects of the applied improvements in the organization with the results of the model. In explanatory case study, we observe that organizations are glad to participate the study as they have the chance to evaluate their measurement capabilities and find the results useful.

By using action research, we develop a measurement capability model that takes its roots from practical needs of organizations first time in the literature. This research method provides the researcher the opportunity to observe the problem setting in detail and understand the requirements of the solution from the perspective of the subject of the study. The steps conducted during the action research reveal that using a list of core measures for each aspect in software development lifecycle could help organizations to understand the required core measure set for their measurement needs. In addition to the list of core measures, a set of general practices evaluating used measures in three capability levels are identified.

We compare the model with two other models from the literature with similar purposes, to observe its differences from them. The model provides similar results when compared with other methods; however, there are some important differences with the measurement capability model. Most of these differences are advantages of the model. First advantage is that organizations can choose the aspects that they need to focus; therefore, they do not have to analyze components that they do not want to evaluate. Second, the model lets organizations focus on individual aspects instead of complete measurement process. This approach helps them to identify focused improvement activities. Third benefit is that the proposed model helps organizations to evaluate the measures with more precision. Using a model with a finer granularity helps organization see their weak and strong points with more precision. Fourth, the model evaluates capabilities of measures instead of processes. Therefore, if an organization does not want to analyze its process, or follow a process-centric improvement approach, MCM v1 is a more beneficial choice when compared with the other applied methods. Lastly, even though all efforts are close to each other, in the case study MCM v1 requires the least amount of effort. Providing the opportunity to select required aspects for the evaluation is an important factor reducing required effort. There is also one drawback of the model, which is that the organization cannot evaluate the aspects that are not selected. However, not selecting an aspect reflects the organizational awareness about the lack of capability in this area.

This research proposes a measurement capability model for software organizations. The model is proven to have additional benefits when compared with the available alternative methods in the literature. These benefits can be listed as:

- The model uses a measure oriented dynamic approach, instead of a process-centric approach like its alternatives, allowing organizations to focus on specific aspects instead of whole processes.
- Focusing on specific aspects helps organizations to identify goal oriented improvement points.
- With its granulated structure, this model helps organizations to examine their measures in detail.
- Organizations utilizing agile methodologies can benefit from this model to identify their current situations and to identify improvement opportunities.
- Organizations that do not have measurement expertise can benefit from listed core measures and practices in the model.
- Organizations can use the model definitions and case studies given in this research to apply the model. To use the self-contained nature of the model, they may not have to work with an outside consultant.

6.2 Limitations, Lessons Learned, and Future Work

Threats to the validities of individual case studies and action research are discussed in the related sections together with the methods to mitigate them. The mitigatory actions are applied as described in the related sections to overcome those threats. To mitigate some of the threats additional case studies are used. For example, one case study is conducted by another researcher to mitigate possible positive inclination, to prevent mono-method bias several case studies are conducted with different methods, and to increase generalizability case studies are conducted in different organizations.

One of the limitations of this research is related to the generalizability of the domains where the model can be used. Because of the costly nature of conducting a case study in a real setting, it is not feasible to test the model in all the possible domains. Even though the model is successfully applied in eleven different cases in seven different software organizations, expected benefits of the model would not be validated enough to assure generalizability. As it is expected from a case study based research, to validate all the expected benefits of the model and to improve generalizability, the number and diversity of case studies should be increased.

Another limitation is related to one of the main principles of the model. As organizations are free to choose any aspects to evaluate and improve themselves, overall capabilities of two different organizations may not be comparable. If there is a need of comparison, instead of overall capabilities of selected aspects should be used. This is especially important if this model is used to evaluate and compare competing organizations.

During the conduct of the exploratory and explanatory case studies, in addition to the findings of the studies, researcher also gains valuable information related with the conduct of case studies. First, working with real organizations provides a challenge. Organizations help academic research as long as they also gain some kind of benefit. During this research, while communicating with the organizations before the studies, special care is given to emphasize benefits that will be provided to them at the end. For this research this benefit is the resulting evaluation of measurement practices and improvement ideas. As the results of the explanatory study also demonstrate, organizations are satisfied with this benefit. Second important lesson learned from the experiences is related to the importance of communication medium. As some of the organizations participating in the research are located in other cities and there are some scheduling conflicts, online meeting tools are used for communication. Even though these technological advancements are quite helpful, they do not provide the same advantages as face-to-face communication. If possible, face-to-face communications should be preferred as it provides a more intimate environment for communication and also permits non-verbal communication.

In the future, additional research might be conducted to improve the model and to test its benefits that are not tested during this study. Below are some of the ideas that can be used to form an agenda for the future work:

- The model is expected to self-contained, meaning that organizations can easily apply it without a third party support. This feature should be tested in software developing organization with case studies.
- The model has the potential to increase response times of organizations to the environmental changes. This should be tested with a comparative case study with additional methods, similar to the second exploratory case study conducted in this research, but with an increased domain focus.
- The model should be applied in organizations utilizing different software development lifecycles to understand the domains that the model is optimally effective. It is designed to answer demands of organizations preferring dynamic approaches; however, case studies present a potential for all types of development lifecycles.

- The model can be used to assess measurement capability; however, a formal assessment method is not available. A model specific assessment method can be developed.
- To help the organizations that are new to process improvement, to disseminate and maintain the model, processes can be developed.
- Best practices for the suggested measures can be included in the model to help smaller organizations that wish to implement new measures.
- Additional aspects like; customer satisfaction, competitiveness, employee performance, training, and agility can be evaluated according the scope of the model to be included as new aspects.
- The research provides all components of the model to apply in a real life setting. A software application can be developed to ease the use of the model. The application can communicate with organizational databases and collect data from them by utilizing measurement ontologies. It can also guide organizations through assessment and improvement opportunity identification.
- To integrate into a possible software application, a measurement ontology that is compatible with this model and popular organizational measurement databases can be developed. An ontology can potentially reduce required data collection effort for organizations.

REFERENCES

Albrecht, A. J. (1979). Measuring Application Development Productivity. In *Proceedings of IBM Application Development Symposium* (pp. 83–92).

Baskerville, R. (1999). INVESTIGATING INFORMATION SYSTEMS WITH ACTION RESEARCH. *Communications of AIS*, *2*(19), 1–32.

Beck, K., Beedle, M., Arie van Bennekum, Cockburn, A., Cunningham, W., Fowler, M., … Thomas, D. (2001). Manifesto for Agile Software Development. Retrieved November 13, 2017, from http://agilemanifesto.org/

Berry, M., & Vandenbroek, M. F. (2001). A Targeted Assessment of the Software Measurement Process. In *Software Metrics Symposium, 2001. METRICS 2001* (pp. 222–235). London.

Briand, L. C., & Daly, J. W. (1998). A Unified Framework for Cohesion Measurement in Object-Oriented Systems. *Empirical Software Engineering*, *3*, 65–117.

Briand, L. C., Daly, J. W., & Wüst, J. K. (1999). A Unified Framework for Coupling Measurement in Object-Oriented Systems. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, *25*(1).

Budlong, F., & Peterson, J. (1995). *SOFTWARE METRICS CAPABILITY EVALUATION GUIDE*.

Burnstein, I., Homyen, A., Grom, R., & Carlson, C. R. (1998). A Model to Assess Testing Process Maturity. *The Journal of Defense Software Engineering*.

CMMI Institute. (2014). Data Management Maturity Model V1.0. Retrieved November 23, 2017, from http://cmmiinstitute.com/resources/data-management-maturity-model-v10

CMMI Product Team. (2010). *CMMI® for Development, Version 1.3 CMMI-DEV, V1.3*.

Collab.net, & VersionOne. (2018). *The 12th Annual State of the Agile Report*. Retrieved from https://explore.versionone.com/state-of-agile

Crosby, P. B. (1979). *Quality is Free: The Art of Making Quality Certain*. McGraw-Hill.

Curtis, B., & Alden, J. (2007). The Business Process Maturity Model (BPMM): What, Why and How. *BPTrends*.

Curtis, B., Hefley, W. E., & Miller Sally A. (2001). *People Capability Maturity Model (P-CMM) Version 2.0*. Pittsburgh: Carnegie Mellon Software Engineering Institute.

Daskalantonakis, M. K., & Yacobellis, R. H. (1990). A Method for Assesing Software Mesurement Technology. *Quality Engineering*, *3*(1), 27–40.

Díaz-Ley, M. (2009). *Measurement framework for the definition of sotware measurenment programs in smes: mis-pyme*. Universidad de Castilla-La Mancha.

Díaz-Ley, M., Garcí, F., & Piattini, M. (2008). Implementing a software measurement program in small and medium enterprises: a suitable framework.

Díaz-Ley, M., García, F., & Piattini, M. (2010). MIS-PyME software measurement capability maturity model - Supporting the definition of software measurement

programs and capability determination. *Advances in Engineering Software*, *41*, 1223–1237.
Efe, P., & Demirors, O. (2013). Applying EVM in a Software Company: Benefits and Difficulties. In *2013 39th Euromicro Conference on Software Engineering and Advanced Applications* (pp. 333–340). IEEE. Retrieved from http://ieeexplore.ieee.org/document/6619530/
Fenton, N., & Bieman, J. (2014). *Software Metrics A Rigorous and Practical Approach* (Third). Boca Raton, FL, USA. https://doi.org/10.1201/b17461
Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *Design Science in IS Research MIS Quarterly*, *28*(1), 75–105.
Humphrey, W. S. (1988). Characterizing the Software Process: A Maturity Framework. *IEEE Software*, 73–79.
Ifra, & Bajwa, J. K. (2016). Metrics of Scrum Methodology. *International Journal of Computer Applications*, *149*(2), 975–8887.
ISBSG Dataset Release 9. (2009). Retrieved from www.ISBSG.org
ISO/IEC 12207. (2008). Systems and software engineering -- Software life cycle processes. Geneva, Switzerland: International Organization for Standardization/International Electrotechnical Commission.
ISO/IEC 15504-5. (2012). Information technology -- Process assessment -- Part 5: An exemplar software life cycle process assessment model. Geneva, Switzerland: International Organization for Standardization/International Electrotechnical Commission.
ISO/IEC 15939. (2017). Systems and software engineering -- Measurement process. Geneva, Switzerland: International Organization for Standardization/International Electrotechnical Commission/The Institute of Electrical and Electronics Engineers.
ISO/IEC 19761. (2011). Software engineering -- COSMIC: a functional size measurement method. Geneva, Switzerland: International Organization Standardization/International Electrotechnical Commission.
ISO/IEC 20926. (2009). Software and systems engineering -- Software measurement -- IFPUG functional size measurement method. Geneva, Switzerland: International Organization for Standardization/International Electrotechnical Commission.
ISO/IEC 33001. (2015). Information technology -- Process assessment -- Concepts and terminology. Geneva, Switzerland: International Organization for Standardization/International Electrotechnical Commission.
İnce, A. (2016). *Comparison of Three Measurement Maturity Models: Case Study in a Software Organization*. Ankara.
Kırbaş, S. (2007). *An Assessment and Analysis Tool for Statistical Process Control of Software Processes a Thesis Submitted To the Graduate School of Natural and Applied Sciences of the Middle East Technical University By Serkan Kirbaş in Partial Fulfillment of the Requirem*. MIDDLE EAST TECHNICAL UNIVERSITY.
Li, W., & Henry, S. (1993). Object-Oriented Metrics that Predict Maintainability. *Systems Software*, *23*, 111–122.

Marshall, S., & Mitchell, G. (2002). An E-Learning Maturity Model. *InProceedings of the 19th Annual Conference of the Australian Society for Computers in Learning in Tertiary Education, Auckland, New Zealand*, (May). https://doi.org/10.1.1.464.7579

McCabe, T. J. (1976). A Complexity Measure. *IEEE Transactions on Software Engineering*, *SE-2*(4), 308–320.

Mendonca, M. G., Basili, V. R., Bhandari, I. S., & Dawson, J. (1998). An approach to improving existing measurement frameworks. *IBM SYSTEMS JOURNAL*, *37*(4), 484–501.

Naur, P., & Randell, B. (1969). Software Engineering. In *NATO Software Engineering Conference*. Garmisch.

Niessink, F., & Van Vliet, H. (1998). Towards Mature Measurement Programs. In P. Nesi & F. Lehner (Eds.), *Proceedings of the Second Euromicro Conference on Software Maintenance and Reengineering* (pp. 82–88). Florence, Italy: IEEE.

Ozcan-Top, O., & Demirörs, O. (2015). A Reference Model for Software Agility Assessment: AgilityMod. In *SPICE 2015* (Vol. 526, pp. 145–158). https://doi.org/10.1007/978-3-319-19860-6

Paulk, M. C., Curtis, B., Chrissis, M. B., & Weber, C. V. (1993). *Capability Maturity Model for Software, Version 1.1*. Pittsburgh.

Petrinja, E., Nambakam, R., & Sillitti, A. (2009). FLOSS: Introducing the OpenSource Maturity Model. In *FLOSS'09* (pp. 37–41). Vancouver.

Project Management Institute. (2003). *Organizational Project Management Maturity Model (OPM3)*. Project Management Institute.

Salmanoğlu, M., Coşkunçay, A., Yıldız, A., & Demirörs, O. (2018). An Exploratory Case Study for Assessing the Measurement Capability of an Agile Organization. *Software Quality Professional*, *20*(2), 36–47.

Salmanoğlu, M., Öztürk, K., Bağrıyanık, S., Ungan, E., & Demirörs, O. (2015). Benefits and Challenges of Measuring Software Size: Early Results in a Large Organization. In *25th International Workshop on Software Measurement and 10th International Conference on Software Process and Product Measurement, IWSM-Mensura 2015,*. Kraków, Poland.

SCAMPI Upgrade Team. (2011). *Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A, Version 1.3: Method Definition Document*. Software Engineering Institute, Carnegie Mellon University. Retrieved from http://www.sei.cmu.edu

Staron, M., & Meding, W. (2015). MeSRAM – A method for assessing robustness of measurement programs in large software development organizations and its industrial evaluation. *The Journal of Systems and Software*, *113*, 76–100. https://doi.org/10.1016/j.jss.2015.10.051

Symons, C., & Lesterhuis, A. (2015). *The COSMIC Functional Size Measurement Method Version 4.0.1 Measurement Manual*.

Tarhan, A., & Demirors, O. (2008). Assessment of Software Process and Metrics to Support Quantitative Understanding. In J.J. Cuadrado-Gallego et al. (Ed.), *IWSM-Mensura 2007* (Vol. 4895, pp. 102–113).

Tarhan, A., & Demirors, O. (2012). Apply Quantitative Management Now. *IEEE*

Software, *29*(3), 77–85. https://doi.org/10.1109/MS.2011.91
The Open Group. (2011). The Open Group Service Integration Maturity Model (OSIMM) Version 2. Retrieved November 23, 2017, from http://www.opengroup.org/soa/source-book/osimmv2/p1.htm
Thomson, W. (1883). Lecture on "Electrical Units of Measurement." Popular Lectures Vol. I.
Top, Ö. Ö. (2014). *AgilityMod: A Software Agility Reference Model for Agility Assessment*. MIDDLE EAST TECHNICAL UNIVERSITY. Retrieved from http://etd.lib.metu.edu.tr/upload/12618213/index.pdf
Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in Software Engineering*. Berlin: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-29044-2
Wolverton, R. W. (1974). The Cost of Developing Large-Scale Software. *IEEE Transactions on Computers*, *c-23*(6), 615–636.

APPENDICES

APPENDIX A – MeSRAM Questionnaire Answers

Table 21 Assessment of Business Metrics

ID	Explanation	Assessment for Div. A	Assessment for Div. B	Assessment for Div. C
MU-1	Customer	N	Y	Y
MU-2	Value	N	N	N
MU-3	Financial Perspective	Y	Y	Y
MU-4	Product Delivery	N	Y	Y
MU-5	Defects in Production	N	Y	Y
MU-6	Product Backlog	N	Y	Y

Table 22 Assessment of Product Metrics

ID	Explanation	Assessment for Div. A	Assessment for Div. B	Assessment for Div. C
MU-6	Product Backlog	N	Y	Y
MU-7	Readiness	N	Y	Y
MU-8	Defects	Y	Y	Y
MU-9	Product Properties	Y	Y	Y
MU-10	Product Performance	Y	Y	Y
MU-11	Product Management	N	N	N
MU-12	Maintenance	Y	Y	Y

Table 23 Assessment of Design Metrics

ID	Explanation	Assessment for Div. A	Assessment for Div. B	Assessment for Div. C
MU-13	Design Stability	N	N	N
MU-14	Product/Code Stability	N	N	N
MU-15	Design Debt	N	N	N
MU-16	Defects	N	N	N
MU-17	Size	N	N	N

Table 24 Assessment of Organizational Performance Metrics

ID	Explanation	Assessment for Div. A	Assessment for Div. B	Assessment for Div. C
MU-18	Velocity	N	Y	Y
MU-19	Throughput/Efficiency	N	N	N
MU-20	Customer Perspective	N	Y	Y
MU-21	Financial Perspective	N	N	N
MU-22	Internal Business Process Perspective	N	N	N
MU-23	Delivery Precision	N	N	N
MU-24	Innovation and Learning growth	N	N	N
MU-25	Employee assets	N	N	N
MU-26	Ways of Working	N	N	N

Table 25 Assessment of Project Metrics

ID	Explanation	Assessment for Div. A	Assessment for Div. B	Assessment for Div. C
MU-27	Status	N	Y	Y
MU-28	Progress	N	Y	Y
MU-29	Release Readiness	N	N	N
MU-30	Quality	N	N	N
MU-31	System Management	N	N	N
MU-32	Design	N	N	N
MU-33	Integration	N	N	N
MU-34	Test	N	N	N
MU-35	Prediction	N	N	N
MU-36	Team	N	N	N
MU-37	Legacy	N	N	N

Table 26 Assessment of Decision Support Category

ID	Explanation	Assessment for Div. A	Assessment for Div. B	Assessment for Div. C
DS-1	It is clear who is interested in the metrics data	N	Y	Y
DS-2	Meaning/interpretations of metrics are defined	N	Y	Y
DS-3	Metrics are used for analyses of problems/root causes	N	N	N
DS-4	Metrics and indicators are used to formulate decisions	N	N	N
DS-5	Metrics and indicators are used to monitor implementation of decisions	N	N	N
DS-6	It is clear which metrics and indicators are used for technical and managerial areas respectively	N	Y	Y

Table 27 Assessment of Organizational Metrics Maturity Category

ID	Explanation	Assessment for Div. A	Assessment for Div. B	Assessment for Div. C
OM-1	There is a prioritized list of defects per product	Y	N	N
OM-2	There is a list over the most complex SW modules	Y	N	N
OM-3	Metrics and indicators are collected/calculated using documents and repeatable algorithms	N	N	N
OM-4	Metrics and indicators are collected/calculated manually	N	Y	Y
OM-5	Metrics and indicators are collected/calculated automatically	N	Y	Y
OM-6	Metrics and indicators are visualized with decision criteria	N	N	N
OM-7	Metrics and indicators are accompanied with information quality/reliability evaluation	N	N	N
OM-8	Metrics and indicators available in standard tools (Eclipse, MS Excel) are used to understood in the organization	N	Y	Y

Table 28 Assessment of Metrics Organization Category

ID	Explanation	Assessment for Div. A	Assessment for Div. B	Assessment for Div. C
MO-1	There is a metrics organization (Role, Team, Group)	N	Y	Y
MO-2	The metrics organization has sufficient resources	N	N	N
MO-3	There is metrics organization that maintains existing metrics	N	Y	Y
MO-4	There is metrics organization that supports the organization with metrics related issues (what to measure, etc.)	N	N	N
MO-5	The metrics organization has good knowledge of the company's products	N	N	N
MO-6	It is well defined and transparent how the metrics organization prioritizes its assignment	N	N	N
MO-7	The metrics organization can handle emergencies	N	N	N
MO-8	There exist a role: metrics champion	N	N	N
MO-9	There exist a role: measurement sponsor	N	N	N
MO-10	There exist a role: measurement analyst	N	N	N
MO-11	There exist a role: metrics designer	N	N	N
MO-12	There exist a role: measurement librarian	N	N	N
MO-13	There exist a role: metrics team leader	N	Y	Y
MO-14	There exist a document describing how the metrics organization works	N	N	N
MO-15	There exist a strategy plan	N	Y	Y
MO-16	There exist a contingency plan	N	N	N
MO-17	The metrics organization gives presentations/seminars/courses	N	Y	Y
MO-18	There exist a Statement of Compliance for ISO/IEC/IEEE 15939	N	N	N
MO-19	There exist a Statement of Compliance for ISO/IEC/IEEE 12207	N	N	N
MO-20	There exist a Statement of Compliance for IEEE Std 1061	N	N	N
MO-21	There exist a Statement of Compliance for ISO/IEC/2502x family	N	N	N

Table 29 Metrics Infrastructure Category

ID	Explanation	Assessment for Div. A	Assessment for Div. B	Assessment for Div. C
MI-1	There exist a structure that contains all/the most important metrics	N	N	N
MI-2	The infrastructure is secure	N	N	N
MI-3	The infrastructure is built up so that is supports automation	N	N	N

ID	Explanation	Assessment for Div. A	Assessment for Div. B	Assessment for Div. C
MI-4	All measurement systems include Information Quality	N	N	N
MI-5	The infrastructure supports/enables dissemination of information products	N	N	N
MI-6	There exist naming rules for folders and files	N	N	N

Table 30 Assessment of External Collaboration Category

ID	Explanation	Assessment for Div. A	Assessment for Div. B	Assessment for Div. C
EC-1	MO has collaboration within company	N	N	N
EC-2	MO has collaboration with other companies	N	N	N

Table 31 Assessment of Collaboration with Academia Category

ID	Explanation	Assessment for Div. A	Assessment for Div. B	Assessment for Div. C
AC-1	MO has collaboration with academia	N	N	N
AC-2	MO executes research projects	N	N	N
AC-3	MO publishes papers	N	N	N
AC-4	MO has students on site	N	N	N
AC-5	MO supervises BSc/MSc thesis	N	N	N

APPENDIX B – MCM v1 Results in Exploratory Study 3

Table 32 MCM v1 Results in Exploratory Study 3

				MCL1				MCL2										MCL1			
	Aspect	**Core Measure**	**Short Explanation**	GP1	GP2	GP3	GP4	GP1	GP2	GP3	GP4	GP5	GP6	GP7	GP8	GP9	GP10	GP1	GP2	GP3	GP4
Operational	Project	Planned effort	Carried out according to the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Actual effort	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		% of effort estimation efficiency	Included in the closing report	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		% of actual effort (at a specific time)	Monitored quarterly	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Planned duration	Carried out according to the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Actual duration	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		% of duration estimation efficiency	Included in the closing report	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		% of actual duration (at a specific time)	Monitored quarterly	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Planned cost	Carried out according to the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Actual cost	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		% of cost estimation efficiency	Included in the closing report	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		% of actual cost (at a specific time)	Monitored quarterly	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
	Risk	Anticipated risks	Included in risk plan	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Occurred risks	Monitored	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Unidentified risks	Included in the closing report	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Risk identification efficiency	Included in the closing report	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
	Quality	Number of non-conformance	Collected during peer-reviews	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Costs of corrective actions	Available on effort reports	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Cost of preventive actions	Available on effort reports	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
	Configuration	Configuration changes	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
	Change	Proposed changes	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Accepted changes	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Cost of change	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
	Procurement	Procurement contract changes	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Quality of supplied work product	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
Technical	Requirements	Number of requirement change	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Cost of requirement change	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
	Solution	Number of design change	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Cost of design change	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				

				MCL1				MCL2										MCL1			
	Aspect	**Core Measure**	**Short Explanation**	GP1	GP2	GP3	GP4	GP1	GP2	GP3	GP4	GP5	GP6	GP7	GP8	GP9	GP10	GP1	GP2	GP3	GP4
	Test	Number of defects	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Internal failure cost	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		External failure cost	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
	Integration	Integration errors	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
Strategic	Process Improvement	Process improvement proposals	Collection and monitoring are defined in the process	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Cost of quality	All inputs are available, can be calculated																		
	Size	Size	In use, but require improvements	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
		Normalized measures, rates	Size is available but not suitable for normalization																		

APPENDIX C – Results and Evaluations of Exploratory Study 4

Table 33 Results and Evaluations of Exploratory Study 4

Aspect Group	Aspect	Entity	Attribute	Sample Measures	Expert 1	Expert 2	Expert 3	Expert 4	Evaluation
Management	Planning	Planned work unit	Effort	Planned person-months from start to finish	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
			Duration	Planned days from start to finish	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
			Cost	Planned cost to be spent for the unit	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
			Risks	Weighted impact of anticipated risks	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
	Monitoring	Completed work unit	Effort	Person-months from start to finish	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
			Duration	Days from start to finish	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
			Cost	Cost to be spent for the unit	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
			Risks	Impact of occurred risks	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
		Product delivery cycle	Blockage	Days of blockage	Beneficial	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
			Flow efficiency	Percentage of actual work duration over total duration	Beneficial	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
		Work unit development process	Earned effort	Effort actualization percentage	Common and beneficial	Beneficial	Beneficial	Common and beneficial	Common and beneficial
			Earned duration	Duration actualization percentage	Common and beneficial	Beneficial	Not applicable	Common and beneficial	Beneficial
			Earned cost	Cost actualization percentage	Common and beneficial	Beneficial	Common and beneficial	Common and beneficial	Common and beneficial
		Project Artifacts	Conformance	Number of non-conformance	Not applicable	Common and beneficial	Not applicable	Not applicable	Not applicable
			Change	Number of proposed changes	Beneficial	Not applicable	Beneficial	Beneficial	Beneficial
				Number of accepted changes	Not applicable	Not applicable	Beneficial	Beneficial	Not applicable

Aspect Group	Aspect	Entity	Attribute	Sample Measures	Expert 1	Expert 2	Expert 3	Expert 4	Evaluation
		Corrective actions	Cost	Cost of correcting errors in artifacts	Beneficial	Beneficial	Not applicable	Common and beneficial	Beneficial
		Preventive actions	Cost	Cost of preventing errors in artifacts	Beneficial	Beneficial	Beneficial	Common and beneficial	Beneficial
		Artifact changes	Cost	Cost of implementing change	Beneficial	Not applicable	Not applicable	Common and beneficial	Beneficial
		Backlog burndown	Velocity	Team velocity	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
			Rate	Sprint burn rate	Common and beneficial	Common and beneficial	Not applicable	An important metric	Common and beneficial
Technical	Procurement	Vendor Supplied Software	Size	Size of supplied product	-	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
			Duration	Time to delivery	-	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
			Unit cost	Cost of product per unit size	-	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
			Quality	Defects density	-	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
		Vendor Corrective Actions	Duration	Duration of defect resolution	-	Common and beneficial	Common and beneficial	Beneficial	Common and beneficial
	Production	Testing	Efficiency	Number of escaped defects	Beneficial	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
				Number of identified defects	Beneficial	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
			Coverage	Percentage of covered criteria over all criteria	Beneficial	Common and beneficial	Beneficial	Beneficial	Beneficial
		Defect Resolution	Effort	Defect resolution effort	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial
			Cost	Internal failure cost	Beneficial	Common and beneficial	Beneficial	Common and beneficial	Common and beneficial
				External failure cost	Beneficial	Common and beneficial	Beneficial	Common and beneficial	Common and beneficial
		Coding	Churn	Number of change in the code	Beneficial	Beneficial	Not applicable	Beneficial	Beneficial
			Reuse	Percentage of software code reused	Beneficial	Beneficial	Beneficial	Beneficial	Beneficial
		Product delivery cycle	Frequency	Time between each release	Not applicable	Common and beneficial	Beneficial	Common and beneficial	Beneficial

Aspect Group	Aspect	Entity	Attribute	Sample Measures	Expert 1	Expert 2	Expert 3	Expert 4	Evaluation
		Design	Complexity	Cyclomatic complexity	Beneficial	Beneficial	Beneficial	Common and beneficial	Beneficial
		Design component	Cohesion	Ratio of cohesive interactions (RCI) (Briand et al. (1998))	Beneficial	Beneficial	Beneficial	Common and beneficial	Beneficial
			Coupling	Message passing coupling (MPC) (Li and Henry 1993, Briand et al. 1999b)	Beneficial	Beneficial	Beneficial	Common and beneficial	Beneficial
Strategic	Improvement	Improvement process	Propositions	Number of proposed improvements per employee	Beneficial	Common and beneficial	Beneficial	Common and beneficial	Common and beneficial
			Involvement	Reflection meeting attendance over total employees	Not applicable	Common and beneficial	Beneficial	Common and beneficial	Beneficial
		Effort estimation process for work units	Efficiency	Percentage of actual effort over planned effort	Beneficial	Common and beneficial	Common and beneficial	Not applicable	Beneficial
		Duration estimation process for work units	Efficiency	Percentage of actual duration over planned duration	Beneficial	Common and beneficial	Common and beneficial	Not applicable	Beneficial
		Cost estimation process for work units	Efficiency	Percentage of actual cost over planned cost	Beneficial	Common and beneficial	Common and beneficial	Not applicable	Beneficial
		Non-value-added tasks	Cost	Cost of quality	Beneficial	Beneficial	Not applicable	Common and beneficial	Beneficial
	Foundation	Software component	Functional Size	COSMIC FSM	Beneficial	Common and beneficial	Beneficial	Not applicable	Beneficial
		Measure	Normalization	Defect density	Beneficial	Common and beneficial	Common and beneficial	Common and beneficial	Common and beneficial

APPENDIX D –List of Aspects, Entities, Attributes, and Sample Measures

Table 34 List of Aspects, Entities, Attributes, and Sample Measures

Aspect Group	Aspect	Entity	Entity Definition	Attribute	Sample Measures
Management	Planning	Planned work unit	Each planned base unit of the project	Effort	Planned person-months from start to finish
				Duration	Planned days from start to finish
				Cost	Planned cost to be spent for the unit
				Risks	Weighted impact of anticipated risks
	Monitoring	Completed work unit	Each completed base unit of the project	Effort	Person-months from start to finish
				Duration	Days from start to finish
				Cost	Cost to be spent for the unit
				Risks	Impact of occurred risks
		Product delivery cycle	Time between two consecutive deliveries, for each delivery cycle	Blockage	Days of blockage
				Flow efficiency	Percentage of actual work duration over total duration
		Work unit development process	Up until the point of measurement in the process of developing, creating, or finalizing work units	Earned effort	Effort actualization percentage
				Earned duration	Duration actualization percentage
				Earned cost	Cost actualization percentage
		Project Artifacts (Optional for agile methods)	Deliverables or outputs of any step in the development lifecycle	Conformance	Number of non-conformance
				Change	Number of proposed changes
					Number of accepted changes
		Corrective actions (Optional for agile methods)	Any corrective action carried out in development lifecycle	Cost	Cost of correcting errors in artifacts
		Preventive actions	Any preventive action carried out in development lifecycle	Cost	Cost of preventing errors in artifacts
		Artifact changes (Optional for agile methods)	Any change in any project artifact	Cost	Cost of implementing change
		Backlog burndown	The process of completing project requirements wafting for development	Velocity	Team velocity
				Rate	Sprint burn rate
Technical	Procurement	Vendor Supplied Software	The software product supplied by the vendor	Size	Size of supplied product
				Duration	Time to delivery
				Unit cost	Cost of product per unit size
				Quality	Defects density
		Vendor Corrective Actions	Any corrective action carried out by the vendor to fix identified errors after the delivery	Duration	Duration of defect resolution
	Production	Testing	Testing phase or process during development lifecycle	Efficiency	Number of escaped defects
					Number of identified defects

Aspect Group	Aspect	Entity	Entity Definition	Attribute	Sample Measures
				Coverage	Percentage of covered criteria over all criteria
		Defect Resolution	The process of resolving defied defects	Effort	Defect resolution effort
				Cost	Internal failure cost
					External failure cost
		Code	Coding phase or process during development lifecycle	Churn	Number of change in the code
				Reuse	Percentage of software code reused
		Product delivery cycle	Time between two consecutive deliveries, for each delivery cycle	Frequency	Time between each release
		Design	The plan of the software solution conceptualizing the final product	Complexity	Cyclomatic complexity
		Design component	Design unit	Cohesion	Ratio of cohesive interactions (RCI) (Briand et al. (1998))
				Coupling	Message passing coupling (MPC) (Li and Henry 1993, Briand et al. 1999b)
Strategic	Improvement	Improvement process	The process of organizational improvement	Propositions	Number of proposed improvements per employee
				Involvement	Reflection meeting attendance over total employees
		Effort estimation process for work units	Estimation method of effort for work units	Efficiency	Percentage of actual effort over planned effort
		Duration estimation process for work units	Estimation method of duration for work units	Efficiency	Percentage of actual duration over planned duration
		Cost estimation process for work units	Estimation method of cost for work units	Efficiency	Percentage of actual cost over planned cost
		Non-value-added tasks	Tasks that are not directly adding value. Mainly includes cost of control and cost to fix	Cost	Cost of quality
	Foundation	Software component	Main components of software; specifications, requirements, design, code	Size	COSMIC FSM
		Measure	Any measure used in the organization	Normalization	Defect density

APPENDIX E – Sample Measures and Related Information

The sample measures suggestion in the model are explained below with their justifications, purposes, method of application, formulas, and measurement scales.

Measure: Planned person-months from start to finish
Justification: A core measure for Software Engineering as it identifies the amount of most critical resource for software engineering tasks.
Agile methods creates a fixed effort boundary by determining time (e.g. 1 week) and dedicated personnel for specified work units (mostly called sprint) and estimates amount of works going into these time slots.
Purpose: To understand required effort for the identified work unit and the project.
How to apply: Should be estimated during planning phase of the project or specified work unit. It is also used as an input for duration and cost planning.
Formula: A suitable estimation technique
Scale: Ratio

Measure: Planned days from start to finish
Justification: A core measure for any Engineering Project.
Agile methods create a fixed work unit duration period (sprint duration) and decompose projects into these work units by estimating amount of work for each period.
Purpose: To understand required time for the identified work unit and the project.
How to apply: Should be estimated during planning phase of the project or specified work unit and should be used to determine project schedule.
Formula: A suitable estimation technique
Scale: Ratio

Measure: Planned cost to be spent for the unit
Justification: A core measure for any project. In software projects usually calculated by effort. If there are additional costs, they can be added.
Agile methods has fixed efforts for each sprints and estimate work units to assign into them. Other costs can also be considered.
Purpose: To understand required budget for the work unit and the project.
How to apply: Should be estimated during planning phase of the project or specified work unit by using effort and duration estimations. Possible procurement activities and development tools should also be taken into account.
Formula: Estimated cost of labor + other estimated costs
Scale: Ratio

Measure: Weighted impact of anticipated risks
Justification: A core measure for the success of a project to foresee and mitigate possible drawbacks.

The collaborative and involving structure of agile methods make it somewhat easier to identify and respond risks in time. However, there is still the need of identifying and prioritizing the risks, which usually happen at planning, review and retrospective meetings.
Purpose: To be prepared for and to mitigate possible problems during project lifecycle.
How to apply: Project teams should determine possibility of happening and possible impacts of anticipated risks for the work units and the project by taking into account project properties and environmental factors.
Formula: possibility of the risk x impact of the risk
Scale: Ordinal

Measure: Person-months from start to finish
Justification: A core measure for Software Engineering. It is required to analyze the use of available resources and calculate cost of work units.
Agile methods have definitive measures for spent effort, as they use teams with fixed sizes and sprints with fixed durations. They select available work units to fit into sprints and control their completion at the end. Knowing spent efforts for specific work units is necessary to improve their estimation techniques for future sprints.
Purpose: To evaluate effort spent for work units and project, to use for the calculation of derived measures and for benchmarking purposes.
How to apply: Effort data should be collected for all the steps of the projects from all participants
Formula: Spent effort for all project steps for all participants
Scale: Ratio

Measure: Days from start to finish
Justification: A core measure for any Engineering Project. It is important as it determines the date of releasing the product. Organizations need to compare occurred duration with their estimations to improve their future estimations.
Agile methods create a fixed duration period (Sprint duration) for work units, therefore, it is important to understand occurred durations for work units to improve future distribution of work. It is also important as it determines total duration of projects and helps future estimations.
Purpose: To evaluate project schedule and duration, to use for the calculation of derived measures and for benchmarking purposes
How to apply: Start and finish dates of all project steps should be recorded to build overall occurred schedule and total duration of work units and project.
Formula: Start and finish dates of all project steps should be recorded in addition to the project start and finish date
Scale: Ratio

Measure: Cost to be spent for the unit
Justification: A core measure for any project. It is required to conduct financial analyses and to understand utilization of organizational resources.
Agile methods calculates occurred cost by total number of sprints and team size and adding additional spending. The distribution of cost is also available by the examination of work units in each sprint and assigned roles for them in the team.

Purpose: To evaluate project budget, to use for the calculation of derived measures and for benchmarking purposes.
How to apply: Cost can be calculated by using effort data and additional spending.
Formula: Cost of labor + other costs
Scale: Ratio

Measure: Impact of occurred risks
Justification: This measure is important to understand the impact of not mitigating a possible risk. If organization cannot mitigate a risk even though it is anticipated at the beginning of the project, the issue need to be addressed for future projects.
Agile methods have review and retrospective meetings for the team to discuss critical issues. Anticipated but not mitigated risks are discussed in these meetings to assess their effects and determine action items for future implementations.
Purpose: To understand reasons of not mitigating an identified risk and identify lessons learned about the issue.
How to apply: If one of the anticipated risks occurs, record its effects and the response of the team.
Formula: Impact of the risk
Scale: Ordinal

Measure: Days of blockage
Justification: A measure to understand the duration spent without producing an output. With the help of it, organization can understand the lost value resulting from blocking items and create a strategy to eliminate them if necessary according to their effects for future projects.
Agile methods value the interaction and blockage that is mostly the results from an obstacle in this interaction. It is important to identify the cost of the obstacles to analyze the feasibility of efforts required to eliminate them.
Purpose: To understand total duration spent by waiting a required input during the work of a unit.
How to apply: Record time spent without production while awaiting a response from the customer or another entity.
Formula: Total time spent without any output during the work on a unit
Scale: Ratio

Measure: Percentage of actual work duration over total duration
Justification: A measure derived from the blockage, aiming similar results. In addition, this measure shows the gap in the total time. This knowledge can be used as an improvement opportunity to eliminate blockages.
Agile methods value continuous delivery of products. It is important to understand ways to improve delivery time and understanding the percentage of real work during the duration of work unit an important measure for improvement.
Purpose: To understand how long a unit is worked on during the time allocated to work on it.
How to apply: Use blockage time spent to work on a unit and how much time total spent to complete that unit
Formula: (Total time - Blockage) / Total time
Scale: Ratio

Measure: Effort actualization percentage
Justification: A measure that help to understand the percentage of predicted effort spent up until a point in the project lifecycle or for a defined work unit. Knowing the actualized effort provides the organization the possibility to understand its position compared with the planned effort and provide the chance to update project plan for future work items if there is a considerable gap between plans and current effort spending. (Efe & Demirors, 2013)
Agile methods value changing requirements, and change in the requirements usually mean change in the required effort. For agile methods, it means the number of sprints planned for a set of work unit may change. Current actualized effort or percentage of completed work in a sprint provide information for the planning of next iterations.
Purpose: To understand how much of the estimated effort for the project or work unit is already spent.
How to apply: Use the value of actual effort until any point in the work unit or project lifecycle and total estimated effort for the work unit or project.
Formula: Actual effort at any point / total estimated effort
Scale: Ratio

Measure: Duration actualization percentage
Justification: A measure that help to understand the percentage of predicted duration up until a point in the project lifecycle or for a defined work unit. Knowing the actualized duration gives the organization to understand its position compared with the planned duration and provide an opportunity to update project plan for future work items if there is a considerable gap between plans and current spent time. (Efe & Demirors, 2013)
Agile methods value changing requirements and changes in the requirements usually mean changes in the overall duration. For agile methods, it means the number of sprints planned for a set of work unit may change. Current duration for a set of work units compared to planned number of sprints provides information to be used in planning meetings of next iterations. As the duration of the sprints and size of teams are usually fixed, earned duration and earned effort sometimes can be used interchangeably.
Purpose: To understand how much of the estimated duration is already spent for the work unit.
How to apply: Use the value of actual duration until any point in the work unit or project lifecycle and total estimated duration for it.
Formula: Actual duration at any point / total estimated duration
Scale: Ratio

Measure: Cost actualization percentage
Justification: A measure that help to understand the percentage of predicted cost spent up until a point in the project lifecycle or for a defined work unit. Knowing the actualized cost gives the organization to understand its position compared with the planned budget and provide an opportunity to update project plan for future work items if there is a considerable gap between budget and current spending. (Efe & Demirors, 2013)

Agile methods value changing requirements and changes in the requirements usually mean changes in the required work and implicitly required cost. For agile methods, it means the number of sprints planned for a set of work unit may change. Current spending for a work unit compared to planned budget provides information to be used in planning meetings of next iterations. As the cost of unit usually calculated only by the effort, earned cost and earned effort sometimes can be used interchangeably.
Purpose: To understand how much of the estimated cost is already spent for the work unit.
How to apply: Use the value of actual cost until any point in the work unit lifecycle and total estimated cost.
Formula: Actual cost at any point / total estimated cost
Scale: Ratio

Measure: Number of non-conformance
Justification: This measure helps the organization to understand the deviation from the expected criteria for the project artifacts. This can later be used to help the organization identify the criticality of the situation and take action if necessary.
Agile methods value quality, therefore identification of non-conformances and tracking their number is crucial to understand the criticality of the situation and resolve problems with creation of valuable products if necessary. However, organizations utilizing agile methods most of the time do not produce any project artifacts other than software itself. For those organizations, this is an optional measure.
Purpose: To understand how many errors are present in different project artifacts (documents and products).
How to apply: Use all non-conformances in the artifacts reported at the end of different audits
Formula: Number of non-conformances
Scale: Absolute

Measure: Number of proposed changes
Justification: The number of change requests help organization to conduct additional analyses by combining this measure with the number of accepted measures. They can understand the satisfaction rate of proposed changes from the customer.
Responding to change is one of the main components of Agile Manifesto. By knowing the number of change requests organization can calculate their rate of satisfying customer requests. However, some organizations may choose not to measure the changes, as it is an expected property of agile methods. For those organizations, this is an optional measure.
Purpose: To evaluate possible changes received after the completion of project artifacts
How to apply: Collect all change proposals for each project artifacts
Formula: Total number of change proposals for project artifacts
Scale: Absolute

Measure: Number of accepted changes
Justification: Knowing the number of accepted changes in the deliverables help organizations to calculate the rate of accepting change requests. In addition, they can

also conduct additional analyses if they require calculating additional effort spent for the change.
Responding to change is one of the main components of Agile Manifesto. By knowing the number of change requests organization can understand how much additional requests of the customer is satisfied. However, some organizations may choose not to measure the changes, as it is an expected property of agile methods. For those organizations, this is an optional measure.
Purpose: To evaluate validity of versions of project artifacts
How to apply: Collect all accepted change proposals for each project artifacts
Formula: Total number of accepted change proposals for project artifacts
Scale: Absolute

Measure: Cost of correcting errors in artifacts
Justification: This measure is similarly important as the number of non-conformances. This measure provides an input for the organizations to understand the cost of not producing conforming artifacts in the first try and provide necessary motivation to change their processes if necessary.
Agile methods aim to produce best quality. The cost spent for non-conformances is an important measure of the cost of not producing best quality in the first try. If an agile utilizing organization choose not to evaluate number of non-conformances because of lack of project artifacts, it may also omit this measure.
Purpose: To understand how much cost is spent to correct the errors in project artifacts
How to apply: Collect data of cost spent for the actions conducted to fix non-conformances.
Formula: Cost of actions to fix non-conformances
Scale: Ratio

Measure: Cost of preventing errors in artifacts
Justification: This measure relates with the measure of cost of correcting errors. It is expected that if this measure increases, it creates a higher decrease in the other measure, as preventing errors is less costly than fixing them. Organizations need to use both measures to understand the effects of their preventative actions.
Agile manifesto aims to deliver valuable products continuously; therefore, it is more crucial to prevent errors than fixing them. Similar to traditional methods, to compare the effectiveness of prevention both measure need to be collected continuously.
Purpose: To understand how much cost is spent to prevent errors in project artifacts
How to apply: Collect data of cost spent for the actions conducted to prevent errors. Cost may include effort and any possible procurement
Formula: Cost of actions to prevent non-conformances
Scale: Ratio

Measure: Cost of implementing change
Justification: Similar to measuring the number of changes, measuring the cost is important to understand the effects of changing needs of customer. Also with the help of other measures, organization can differentiate additional costs occurring as a result of internal effects and occurring as a result of customer requests.
Responding to change is one of the main components of Agile Manifesto. The cost of implementing this change shows the cost of satisfying customers' additional needs. If

an agile utilizing organization choose not to evaluate number of changes, it may also omit this measure."
Purpose: To evaluate cost of adding additional requests to the project artifacts
How to apply: Collect the cost of implementing a change into a project artifact by considering its effect on all project lifecycle
Formula: Total cost of implementing an accepted change into project artifacts
Scale: Ratio

Measure: Team velocity
Justification: This measure helps organizations to understand the change of team productivity over time and they can use this information on future planning activities.
Agile methods use sprints to develop a pre-determined amount of work units. Knowing the velocity of a team over time helps to plan workload of future sprints more efficiently.
Purpose: To understand the rate of completing requirements for each development cycle
How to apply: Calculate total size of completed requirements in each cycle
Formula: Size of completed requirements in each cycle
Scale: Interval or Ratio (depending on the size measure)

Measure: Sprint burn rate
Justification: "This measure helps organizations to plan required number of sprints to develop waiting requirements by using number of sprints used to develop past requirements.
Agile methods assign a pre-determined size of work-units for each sprint. By knowing past burn rates, future sprints can be planned accordingly. "
Purpose: To understand the speed and rate of completing waiting requirements in each sprint
How to apply: Track how much the team is developing in each sprint
Formula: Developed requirements / number of sprints of these requirements
Scale: Ratio

Measure: Size of supplied product
Justification: The size of the product submitted by the vendor is a core measure to compensate the vendors fairly for their work. This measure is also used to derive other measures to understand and compare effectiveness of vendors.
Agile methods may also require procurement of software. This measure is required for procurement activities independent from the lifecycle used in customer organization.
Purpose: To track production performance of the vendor
How to apply: Track the size of submitted work by the vendor and compare it with the expected work
Formula: Total size of submitted work by vendor
Scale: Interval or Ratio (depending on the size measure)

Measure: Time to delivery
Justification: This measure is important to evaluate the speed of delivery of the procured software to evaluate vendors according to the contract.
In Agile methods same measure is required if a procurement activity is required.

Purpose: To track duration of completing works submitted to the vendor
How to apply: Track the duration to complete the work by the vendor
Formula: Total duration to complete work by vendor
Scale: Ratio

Measure: Cost of product per unit size
Justification: This derived measure is important to evaluate the productivity of the vendor against organizations own productivity rate and against the rates of other vendors.
In Agile methods same measure is required if a procurement activity is required.
Purpose: To evaluate productivity of a vendor
How to apply: Use size and cost of the product completed by the vendor to calculate its unit cost
Formula: Cost of completed software / size of completed software
Scale: Ratio

Measure: Defects density
Justification: This measure is important to evaluate the quality of the product supplied by the vendor. It can be used to control the satisfaction levels of contract requirements and it can be used as an evaluation factor for future contracts.
In Agile methods same measure is required if a procurement activity is required.
Purpose: To evaluate the quality of the software produced by the vendor
How to apply: Test the submitted software and report number of defects
Formula: Number of identified defects / size of completed software
Scale: Ratio

Measure: Duration of defect resolution
Justification: This measure is used to evaluate vendors' effectiveness to resolve the problems of the supplied product. It can be used to control the satisfaction levels of contract requirements and it can be used as an evaluation factor for future contracts.
In Agile methods same measure is required if a procurement activity is required.
Purpose: To evaluate the performance of the vendor to solve the defects
How to apply: Calculate the time spent between submitting a defect and receiving corrected software
Formula: Time of resolving a defect
Scale: Ratio

Measure: Number of escaped defects
Justification: This measure is important to understand the effectiveness of quality control activities in the project by calculating number of defects that reached to the customer.
Agile manifesto defines working software as the primary measure of the progress. Number of escaped defects are an important measure to evaluate the software.
Purpose: To understand the effectiveness of testing before the shipment of the software
How to apply: Calculate number of defects identified after the shipment of the software
Formula: Number of defects identified after shipment

Scale: Absolute

Measure: Number of identified defects
Justification: This measure is important to understand the effectiveness of quality control activities in the project and success of the tests conducted in the project. This method, together with the previous measure, can be used to evaluate effectiveness of the test activities and to identify improvement activities.
Agile manifesto values continuous attention to technical excellence, and identified defects are a measure of the excellence of the test activities in the project.
Purpose: To understand the quality of development
How to apply: Record number of defects identified before shipment
Formula: Number of defects identified before shipment
Scale: Absolute

Measure: Percentage of covered criteria over all criteria
Justification: This measure helps to understand the coverage of the test methods applied in the project. This measure can be used together with the efficiency measure to evaluate and improve testing methodologies.
Agile methods value quality products and technical excellence, evaluating testing activities and improving them benefit these main criteria of agile manifesto.
Purpose: To understand how well the software is tested
How to apply: Determine coverage criteria and calculate the rate of coverage of tests conducted on the software according to these criteria (functions, statements, branches, conditions...)
Formula: Covered number of defined test criteria / total number of defined test criteria
Scale: Ratio

Measure: Defect resolution effort
Justification: This measure is used to calculate the cost of defects in the product. The cost data is an important input to evaluate and improve defect prevention and detection activities in the project.
Quality focus in the Agile Methods require improved defect prevention and detection activities. Effort and cost of resolving the defects crucial in creating motivation to improve those activities.
Purpose: To understand organizations ability to solve identified defects
How to apply: Record effort spent to resolve each defect
Formula: Cumulative effort of resolving each defect
Scale: Ratio

Measure: Internal failure cost
Justification: The differentiation of the cost of resolving internal and external failures is important for the organization. Usually it is expected that catching a defect internally and fixing it cost less than fixing it after discovered by the customer. Using these measures help organization to follow their change during a period and help to structure a plan to decrease overall costs.
Customer satisfaction is crucial for Agile Methods. Measuring the cost of letting an error reach to the customer helps to visualize the effects more clearly. Being able to

track the change in these two measures also helps to track the improvement in producing better products.
Purpose: To understand cost of fixing failures identified before the shipment
How to apply: Record total cost to fix defects identified before shipment
Formula: Cost of fixing defects identified before shipment
Scale: Ratio

Measure: External failure cost
Justification: The differentiation of the cost of resolving internal and external failures is important for the organization. Usually it is expected that catching a defect internally and fixing it cost less than fixing it after discovered by the customer. Using these measures help organization to follow their change during a period and help to structure a plan to decrease overall costs.
Customer satisfaction is crucial for Agile Methods. Measuring the cost of letting an error reach to the customer helps to visualize the effects more clearly. Being able to track the change in these two measures also helps to track the improvement in producing better products.
Purpose: To understand total cost of fixing failures identified after the shipment
How to apply: Record total cost to fix defects identified after shipment and cost of their long-term effects
Formula: Cost of fixing defects identified after shipment and cost of their effects
Scale: Ratio

Measure: Number of change in the code
Justification: This measure aims to show the rework on the code. Rework on the code is considered less productive then creating new code. This measure helps to take actions if the amount of rework is higher that organization's threshold.
Agile methods expects a constant pace from the developers, rework on the code is an obstacle on a constant pace.
Purpose: To understand in which rate the code is evolving
How to apply: Identify added, modified or deleted lines through the lifecycle of the code
Formula: Number of change in the code
Scale: Absolute

Measure: Percentage of software code reused
Justification: This is a core measure to derive several other measures to differentiate newly created software vs reused software. It is especially important for the project management activities.
Iterative nature of agile methods requires reuse of previously developed components. While distributing work units for the sprints it is important to consider reuse possibilities. The effect of this consideration can be decided by using past data about reuse.
Purpose: To understand how much of the code is reused from previous projects and how much is newly developed
How to apply: Determine the size of the reused code in the software and size of total code in the software with the size measure utilized in the organization, if there is not any use line of code

Formula: Size of code reused / size of total code
Scale: Ratio

Measure: Time between each release
Justification: This measure especially important for iterative development activities, as customer expected to receive a working product at the end of iterations. The frequency of the releases should be followed to ensure that customer receives them with an acceptable rate.
Agile methods aims continuous delivery, as often as possible. Measuring the frequency helps to track release performance.
Purpose: To understand how often customer receives a working product
How to apply: The time between each release of a working software to the customer should be measured and observed continuously for all releases.
Formula: Time between each release
Scale: Ratio

Measure: Cyclomatic complexity
Justification: This is an important input to calculate satisfaction level of several non-functional requirements, like usability or reliability.
Simplicity is essential for Agile Methods. Therefore tracking the complexity of a software is an important measure.
Purpose: To understand complexity of the software design
How to apply: Count the number of linearly independent paths through the source code using edges and nodes in a control flow graph
Formula: the number of edges of the control flow graph - the number of nodes of the control flow graph + 2(the number of connected components in control flow graph)
Scale: Interval

Measure: Ratio of cohesive interactions (RCI) (Briand & Daly, 1998)
Justification: This is an important input to calculate satisfaction level of several non-functional requirements, like usability or reliability. Cohesion in a design unit expected to be higher and this measure help the organization to evaluate its design to achieve a better design.
Agile methods value simplicity and technical excellence. Measuring cohesion for design units provides insights to satisfy these principles.
Purpose: To understand strength of relationship between the methods and data of a class
How to apply: Cohesive interactions (CIs), which include interactions between public data declarations and interactions between method parameters and return types in public method interfaces
Formula: RCI is the relative number of CIs: RCI(C) = NCI(C)/NPCI(C) where NCI(C) is the number of actual CIs in class C and NPCI(C) is the maximum possible number of CIs.
Scale: Ratio

Measure: Message passing coupling (MPC) (Briand et al., 1999; Li & Henry, 1993)
Justification: This is an important input to calculate satisfaction level of several non-functional requirements, like usability or reliability. Coupling between design units

expected to be lower and this measure help the organization to evaluate its design to achieve a better design.
Agile methods value simplicity and technical excellence. Measuring coupling for the design units provides insights to satisfy these principles.
Purpose: To understand the degree of interdependence between software modules; how closely connected two routines or modules are, or the strength of the relationships between modules
How to apply: The MPC value of a class is a count of the number of static invocations (call statements) of methods that are external to the class
Formula: count of the number of static invocations of methods that are external to the class
Scale: Absolute

Measure: Number of proposed improvements per employee
Justification: The measure provides an understanding of how much the employees are involved in the organizational improvement.
Agile manifesto states that team should regular reflect on how to become more effective. This measure track the involvement of team members on increasing the effectiveness.
Purpose: To understand how much participants are involved for idea generation to improve organizational way of doing things
How to apply: Count the ideas to improve processes coming from the participants of the processes
Formula: Number of ideas for improvement / number of employees
Scale: Ratio

Measure: Reflection meeting attendance over total employees
Justification: The measure provides an understanding of how much the employees are involved in the evaluation of previously conducted work in the development cycle.
In Agile methods team should come together regularly and reflect their opinions to increase effectiveness. This measure tracks teams' performance in attending these regular meetings.
Purpose: To understand how many participants are involved with the evaluations of the completed work.
How to apply: Count the people actively attending the reflection meetings
Formula: Number of active involvements to reflection meetings / number of employees
Scale: Ratio

Measure: Percentage of actual effort over planned effort
Justification: Effort estimation is a core measure for the project management activities. Measuring the efficiency of the estimation method helps the organization to track the efficiency of estimation methods and improve them.
In agile methods better estimations results with better distribution of work units for each sprint and successful completion of the sprints.
Purpose: To understand how accurate the estimation method is
How to apply: Calculate how accurate the estimation is compared with the actual value

Formula: Actual total effort / total estimated effort
Scale: Ratio

Measure: Percentage of actual duration over planned duration
Justification: Duration estimation is a core measure for the project management activities. Measuring the efficiency of the estimation method helps the organization to improve estimation methods.
In agile methods to make better plans for sprint size, release dates and team sizes better duration estimations are requires. To satisfy this, the efficiency of the estimation method should be measured.
Purpose: To understand how accurate the estimation method is
How to apply: Calculate how accurate the estimation is compared with the actual value
Formula: Actual total duration / total estimated duration
Scale: Ratio

Measure: Percentage of actual cost over planned cost
Justification: Cost estimation is a core measure for the project management activities. Measuring the efficiency of the estimation method helps the organization to improve estimation methods.
In agile methods, cost estimations are related to team assignments and sprint planning. Understanding the efficiency of cost estimation methods and improving them helps to make better team assignments and sprint planning.
Purpose: To understand how accurate the estimation method is
How to apply: Calculate how accurate the estimation is compared with the actual value
Formula: Actual total cost / total estimated cost
Scale: Ratio

Measure: Cost of quality
Justification: This derived measure helps the organization understand how much of the project cost is spent on assuring the quality and how much on fixing problems. In the long term this measure help to reduce total cost by guiding organization to eliminate defects.
Agile methods values technical excellence and good design. Knowing the cost of quality provide a way to reduce problems and increase the overall quality of the product.
Purpose: To understand the cost the organization spent to ensure its current level of quality
How to apply: Combined the costs of activities related to the quality of the final product
Formula: Cost of corrective actions + cost of preventive actions + cost of internal and external failures
Scale: Ratio

Measure: COSMIC FSM
Justification: Size is a core measure for a software project, which can be used as an input for all planning and monitoring activities.

For agile methods COSMIC is more efficient in comparison to traditional Story Point (Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement)
Being an comparable measure, it can be used for normalization of other measures and allow organizations to benchmark measurement results
Purpose: To understand total size of the software and use it to calculate normalized values to be used in benchmarking activities
How to apply: Calculate data movements for all of the functional processes in the software
Formula: Total number of Entry, eXit, Read, Write data movements in all functional processes
Scale: Ratio

Measure: Defect density
Justification: Measures themselves have meaning in the project; however, they can only be compared with other projects or organizations if they are normalized with another objective measure, like functional size.
As a sample measure, number of defects may be important for project management. If the organization requires comparing different projects, defect density is a better measure as it regards the size of the software.
Most common size measure in agile methods is story points. It is also used to normalize some measures; however, as it is a subjective estimation instead of an objective measure. To satisfy the need of normalization an objective measure need to be used.

Purpose: To have a comparable defect rate that can be used in benchmarking activities
How to apply: Calculate the number of defects that need to be compared with other projects and divide it with the size of the software
Formula: Number of defects / software size
Scale: Ratio

APPENDIX F – Evaluation Results in the Explanatory Study

Table 35 Evaluation results of organization 1

Aspect Group	Aspect	Entity	Attribute	Sample Measures	Measure level	Aspect level
Management	Planning	Planned work unit	Effort	Planned person-months from start to finish	2	2
			Duration	Planned days from start to finish	2	
			Cost	Planned cost to be spent for the unit	2	
			Risks	Weighted impact of anticipated risks	2	
	Monitoring	Completed work unit	Effort	Person-months from start to finish	2	2
			Duration	Days from start to finish	2	
			Cost	Cost to be spent for the unit	2	
			Risks	Impact of occurred risks	2	
		Product delivery cycle	Blockage	Days of blockage	NA	
			Flow efficiency	Percentage of actual work duration over total duration	NA	
		Work unit development process	Earned effort	Effort actualization percentage	2	
			Earned duration	Duration actualization percentage	2	
			Earned cost	Cost actualization percentage	2	
		Project Artifacts (Optional for agile methods)	Conformance	Number of non-conformance	2	
			Change	Number of proposed changes	2	
				Number of accepted changes	2	
		Corrective actions (Optional for agile methods)	Cost	Cost of correcting errors in artifacts	2	
		Preventive actions	Cost	Cost of preventing errors in artifacts	NA	
		Artifact changes (Optional for agile methods)	Cost	Cost of implementing change	NA	
		Backlog burndown	Velocity	Team velocity	2	
			Rate	Sprint burn rate	2	
Technical	Procurement	Vendor Supplied Software	Size	Size of supplied product	NA	NA
			Duration	Time to delivery	NA	
			Unit cost	Cost of product per unit size	NA	
			Quality	Defects density	NA	
		Vendor Corrective Actions	Duration	Duration of defect resolution	NA	

Aspect Group	Aspect	Entity	Attribute	Sample Measures	Measure level	Aspect level
	Production	Testing	Efficiency	Number of escaped defects	NA	1
				Number of identified defects	2	
			Coverage	Percentage of covered criteria over all criteria	2	
		Defect Resolution	Effort	Defect resolution effort	NA	
			Cost	Internal failure cost	NA	
				External failure cost	NA	
		Code	Churn	Number of change in the code	NA	
			Reuse	Percentage of software code reused	2	
		Product delivery cycle	Frequency	Time between each release	2	
		Design	Complexity	Cyclomatic complexity	NA	
		Design component	Cohesion	Ratio of cohesive interactions (RCI) (Briand et al. (1998))	NA	
			Coupling	Message passing coupling (MPC) (Li and Henry 1993, Briand et al. 1999b)	NA	
Strategic	Improvement	Improvement process	Propositions	Number of proposed improvements per employee	NA	1
			Involvement	Reflection meeting attendance over total employees	NA	
		Effort estimation process for work units	Efficiency	Percentage of actual effort over planned effort	2	
		Duration estimation process for work units	Efficiency	Percentage of actual duration over planned duration	2	
		Cost estimation process for work units	Efficiency	Percentage of actual cost over planned cost	2	
		Non-value-added tasks	Cost	Cost of quality	NA	
	Foundation	Software component	Size	COSMIC FSM	1	1
		Measure	Normalization	Defect density	NA	

Table 36 Evaluation results of organization 2

Aspect Group	Aspect	Entity	Attribute	Sample Measures	Measure level	Aspect level
Management	Planning	Planned work unit	Effort	Planned person-months from start to finish	2	2
			Duration	Planned days from start to finish	2	
			Cost	Planned cost to be spent for the unit	2	
			Risks	Weighted impact of anticipated risks	2	

Aspect Group	Aspect	Entity	Attribute	Sample Measures	Measure level	Aspect level
	Monitoring	Completed work unit	Effort	Person-months from start to finish	2	2
			Duration	Days from start to finish	2	
			Cost	Cost to be spent for the unit	2	
			Risks	Impact of occurred risks	2	
		Product delivery cycle	Blockage	Days of blockage	1	
			Flow efficiency	Percentage of actual work duration over total duration	1	
		Work unit development process	Earned effort	Effort actualization percentage	2	
			Earned duration	Duration actualization percentage	2	
			Earned cost	Cost actualization percentage	2	
		Project Artifacts (Optional for agile methods)	Conformance	Number of non-conformance	NA	
			Change	Number of proposed changes	NA	
				Number of accepted changes	NA	
		Corrective actions (Optional for agile methods)	Cost	Cost of correcting errors in artifacts	NA	
		Preventive actions	Cost	Cost of preventing errors in artifacts	NA	
		Artifact changes (Optional for agile methods)	Cost	Cost of implementing change	NA	
		Backlog burndown	Velocity	Team velocity	2	
			Rate	Sprint burn rate	2	
Technical	Procurement	Vendor Supplied Software	Size	Size of supplied product	NA	NA
			Duration	Time to delivery	NA	
			Unit cost	Cost of product per unit size	NA	
			Quality	Defects density	NA	
		Vendor Corrective Actions	Duration	Duration of defect resolution	NA	
	Production	Testing	Efficiency	Number of escaped defects	NA	1
				Number of identified defects	NA	
			Coverage	Percentage of covered criteria over all criteria	1	
		Defect Resolution	Effort	Defect resolution effort	1	
			Cost	Internal failure cost	NA	
				External failure cost	NA	
		Code	Churn	Number of change in the code	NA	
			Reuse	Percentage of software code reused	NA	
		Product delivery cycle	Frequency	Time between each release	1	
		Design	Complexity	Cyclomatic complexity	NA	

Aspect Group	Aspect	Entity	Attribute	Sample Measures	Measure level	Aspect level
		Design component	Cohesion	Ratio of cohesive interactions (RCI) (Briand et al. (1998))	NA	
			Coupling	Message passing coupling (MPC) (Li and Henry 1993, Briand et al. 1999b)	NA	
Strategic	Improvement	Improvement process	Propositions	Number of proposed improvements per employee	1	1
			Involvement	Reflection meeting attendance over total employees	1	
		Effort estimation process for work units	Efficiency	Percentage of actual effort over planned effort	1	
		Duration estimation process for work units	Efficiency	Percentage of actual duration over planned duration	1	
		Cost estimation process for work units	Efficiency	Percentage of actual cost over planned cost	1	
		Non-value-added tasks	Cost	Cost of quality	NA	
	Foundation	Software component	Size	COSMIC FSM	1	1
		Measure	Normalization	Defect density	NA	

Table 37 Evaluation results of organization 3

Aspect Group	Aspect	Entity	Attribute	Sample Measures	Measure level	Aspect level
Management	Planning	Planned work unit	Effort	Planned person-months from start to finish	3	3
			Duration	Planned days from start to finish	3	
			Cost	Planned cost to be spent for the unit	3	
			Risks	Weighted impact of anticipated risks	3	
	Monitoring	Completed work unit	Effort	Person-months from start to finish	3	2
			Duration	Days from start to finish	3	
			Cost	Cost to be spent for the unit	3	
			Risks	Impact of occurred risks	3	
		Product delivery cycle	Blockage	Days of blockage	1	
			Flow efficiency	Percentage of actual work duration over total duration	1	
		Work unit development process	Earned effort	Effort actualization percentage	3	
			Earned duration	Duration actualization percentage	3	
			Earned cost	Cost actualization percentage	3	
		Project Artifacts (Optional for agile methods)	Conformance	Number of non-conformance	2	
			Change	Number of proposed changes	2	
				Number of accepted changes	2	

Aspect Group	Aspect	Entity	Attribute	Sample Measures	Measure level	Aspect level
		Corrective actions (Optional for agile methods)	Cost	Cost of correcting errors in artifacts	2	
		Preventive actions	Cost	Cost of preventing errors in artifacts	NA	
		Artifact changes (Optional for agile methods)	Cost	Cost of implementing change	2	
		Backlog burndown	Velocity	Team velocity	NA	
			Rate	Sprint burn rate	NA	
Technical	Procurement	Vendor Supplied Software	Size	Size of supplied product	NA	NA
			Duration	Time to delivery	NA	
			Unit cost	Cost of product per unit size	NA	
			Quality	Defects density	NA	
		Vendor Corrective Actions	Duration	Duration of defect resolution	NA	
	Production	Testing	Efficiency	Number of escaped defects	3	2
				Number of identified defects	3	
			Coverage	Percentage of covered criteria over all criteria	1	
		Defect Resolution	Effort	Defect resolution effort	3	
			Cost	Internal failure cost	3	
				External failure cost	3	
		Code	Churn	Number of change in the code	NA	
			Reuse	Percentage of software code reused	NA	
		Product delivery cycle	Frequency	Time between each release	2	
		Design	Complexity	Cyclomatic complexity	2	
		Design component	Cohesion	Ratio of cohesive interactions (RCI) (Briand et al. (1998))	2	
			Coupling	Message passing coupling (MPC) (Li and Henry 1993, Briand et al. 1999b)	2	
Strategic	Improvement	Improvement process	Propositions	Number of proposed improvements per employee	NA	2
			Involvement	Reflection meeting attendance over total employees	NA	
		Effort estimation process for work units	Efficiency	Percentage of actual effort over planned effort	3	
		Duration estimation process for work units	Efficiency	Percentage of actual duration over planned duration	3	
		Cost estimation process for work units	Efficiency	Percentage of actual cost over planned cost	3	
		Non-value-added tasks	Cost	Cost of quality	1	
	Foundation	Software component	Size	COSMIC FSM	NA	NA

Aspect Group	Aspect	Entity	Attribute	Sample Measures	Measure level	Aspect level
		Measure	Normalization	Defect density	NA	

Table 38 Evaluation results of organization 4

Aspect Group	Aspect	Entity	Attribute	Sample Measures	Measure level	Aspect level
Management	Planning	Planned work unit	Effort	Planned person-months from start to finish	2	2
			Duration	Planned days from start to finish	2	
			Cost	Planned cost to be spent for the unit	2	
			Risks	Weighted impact of anticipated risks	1	
	Monitoring	Completed work unit	Effort	Person-months from start to finish	2	2
			Duration	Days from start to finish	2	
			Cost	Cost to be spent for the unit	2	
			Risks	Impact of occurred risks	1	
		Product delivery cycle	Blockage	Days of blockage	NA	
			Flow efficiency	Percentage of actual work duration over total duration	NA	
		Work unit development process	Earned effort	Effort actualization percentage	1	
			Earned duration	Duration actualization percentage	1	
			Earned cost	Cost actualization percentage	1	
		Project Artifacts (Optional for agile methods)	Conformance	Number of non-conformance	2	
			Change	Number of proposed changes	2	
				Number of accepted changes	2	
		Corrective actions (Optional for agile methods)	Cost	Cost of correcting errors in artifacts	1	
		Preventive actions	Cost	Cost of preventing errors in artifacts	NA	
		Artifact changes (Optional for agile methods)	Cost	Cost of implementing change	NA	
		Backlog burndown	Velocity	Team velocity	2	
			Rate	Sprint burn rate	2	
Technical	Procurement	Vendor Supplied Software	Size	Size of supplied product	3	3
			Duration	Time to delivery	3	
			Unit cost	Cost of product per unit size	3	
			Quality	Defects density	3	
		Vendor Corrective Actions	Duration	Duration of defect resolution	3	
	Production	Testing	Efficiency	Number of escaped defects	2	2
				Number of identified defects	2	
			Coverage	Percentage of covered criteria over all criteria	NA	
		Defect Resolution	Effort	Defect resolution effort	1	
			Cost	Internal failure cost	1	
				External failure cost	1	
		Code	Churn	Number of change in the code	NA	
			Reuse	Percentage of software code reused	NA	

Aspect Group	Aspect	Entity	Attribute	Sample Measures	Measure level	Aspect level
		Product delivery cycle	Frequency	Time between each release	2	
		Design	Complexity	Cyclomatic complexity	NA	
		Design component	Cohesion	Ratio of cohesive interactions (RCI) (Briand et al. (1998))	NA	
			Coupling	Message passing coupling (MPC) (Li and Henry 1993, Briand et al. 1999b)	NA	
Strategic	Improvement	Improvement process	Propositions	Number of proposed improvements per employee	NA	2
			Involvement	Reflection meeting attendance over total employees	1	
		Effort estimation process for work units	Efficiency	Percentage of actual effort over planned effort	2	
		Duration estimation process for work units	Efficiency	Percentage of actual duration over planned duration	2	
		Cost estimation process for work units	Efficiency	Percentage of actual cost over planned cost	2	
		Non-value-added tasks	Cost	Cost of quality	NA	
	Foundation	Software component	Size	COSMIC FSM	2	1
		Measure	Normalization	Defect density	NA	

APPENDIX G – Answers for the Discussion Questions in Explanatory Study

Do you think the results reflect the measurement capability of your organization?

- **Organization 1:** The final result accurately reflects the situation; the measurements are compatible with second-level ISO 15504.
- **Organization 2:** Yes, I think it reflects correctly, the result is consistent with my point of view.
- **Organization 3:** Yes, I think it reflects my organization correctly.
- **Organization 4:** The results are correct when we look at them with agile perspective.

Do the results of assessment correspond to your actual observations in the organization? What is the deviation and/or what is the correspondence?

- **Organization 1:** Our organization is level 2 according to ISO 15504 evaluation. The results are consistent with the expectancies from an organization at this level. They are also consistent with my personal observations.
- **Organization 2:** Yes, these results are consistent with my observations.
- **Organization 3:** My observations are represented by the results; they do not deviate from our current status.
- **Organization 4:** Overall, the results are consistent with my experience.

Do the provided improvement opportunities correspond to your actual observations in the organization?

- **Organization 1:** In general, the suggestions reflect my thoughts. There are also some suggestions that we did not think about before.
- **Organization 2:** Suggestions correspond to our observations on base level; however, we need to investigate the feasibility of new methods to understand whether they worth the required effort to implement for our organization. We can initiate an action in the organization. If the investigation provides positive results, we want to implement the suggestions.
- **Organization 3:** Some of the suggested measures were already in our agenda. For example, we aim to implement test coverage measures in 2018. Other than that, there are suggested measures that we do not practice and they seem applicable. However; for the generic practices of level 3, I don't know whether they worth the required effort. We need to work on it.
- **Organization 4:** Suggested improvements are applicable. I think they can provide benefit and they are consistent with my observations.

Do you find the improvement opportunities beneficial? If so, which results are most beneficial?

- **Organization 1:** There are beneficial suggestions among them, I believe in general they are helpful, but there are also one or two that I do not think can help us. For me, the most beneficial suggestions are the information provided by practices about what we should pay attention during measurements and what we should do to gain benefit from the measures. Especially the generic practices of level 3 provide valuable information on what we should do to improve ourselves.

- **Organization 2:** I find them beneficial. Especially among the suggested measures, we can evaluate adoption of some measures. I find the measures that are related to production most beneficial, we would like to adopt them. We do not currently use some measures under technical, I plan to investigate ways to measure them and integrate them into our processes.
- **Organization 3:** Especially the measures related to test coverage are the points that we lack. We are actively trying to improve them. We believe our test practices but we also need to measure how much we cover the software that we develop. In addition, I also believe that the measures related to the code are also important. We need to improve ourselves by considering these suggestions.
- **Organization 4:** We cannot implement all of the suggestions at the same time. It would be beneficial to select most applicable ones and to start with them. I am especially interested in the measures related to defects; there may be improvements about them in organizational level while considering the generic practices. It is crucial to investigate the reasons of the defects and how we solve them, they are absolutely necessary measures.

Are the improvement opportunities feasible? If so, will you plan to change your practices accordingly in the near future? If not, why not?

- **Organization 1:** All of the suggestions are feasible, for most of them our infrastructure is ready, we can immediately start to measure. For example, cost of defect resolution can easily be calculated and it can be quite useful. As I said, quite beneficial outputs and they provide us quite a benefit. We already had an idea that we can improve our measures. We received feedback during our SPICE evaluations and we were planning to think about them and improve. This model provided us important ideas about how to improve.
- **Organization 2:** I plan to implement the suggestions. For us, the main reason of participation in this study is that we want to learn the measures, which we are not aware of but can be helpful. For the strategic measures, we need some time, but we want to implement the measures other than them as soon as possible. Especially the ones related to technical subjects.
- **Organization 3:** We wish to implement the measures related to code and test; we think they can be useful. There are also some measures, which we do not plan to implement. Especially measures that depend on individual employees, like blockage and flow efficiency, do not suit our organizational structure; therefore, we do not wish to implement them. We work as team focused units instead of task based; therefore, we do not have organizational personnel pool, only the teams have their pools. Maybe after implementing other measures we may feel the need to change this situation and update it.
- **Organization 4:** I definitely think to implement the measures that I stated as applicable in the previous question [measures related to defects]. I may not

implement some of the suggested improvement ideas in the first step, because they are not meaningful for our organization.

Have the results provided new information about your organization, anything you did not know before? If so, what is it?

- **Organization 1:** They did provide some new information; we learned new measures and new ways to examine them. We learned how to evaluate ourselves. Aspect oriented approach provided us ideas about how to evaluate us. There are measures that we did not think before but may be beneficial for us. If we use the provided feedback and implement improvements, we can improve our performance. Especially if we use cost of quality, we may gain valuable knowledge about our quality costs, which we can use to provide valuable feedback to senior management.
- **Organization 2:** We observed the points of improvements. Other than the improvements, there are some measures in the provided list that I did not know before. We need to work on them, especially the ones related to strategic issues.
- **Organization 3:** It is useful as it shows our shortcomings in a systematic way. Most of the measures and practices are familiar for us because of our ISO 15504 efforts. Therefore, I did not see any new measure that I personally did not know before. However, as the model assessed the points that we knew but we could not define before, these results should be helpful while communicating with senior management.
- **Organization 4:** I did not think about cost of quality measure before, it may be useful. Other than that, there are not any new measure that I did not know before. As I mentioned in other questions, the model helped us to realize some points to improve ourselves.

Are there any aspects that are not covered by the model but are important for your organization to determine the measurement capability and if so what are they?

- **Organization 1:** There may be measures related to customer satisfaction, it is important for us. In addition, we want to introduce new ideas into the market before our competitors; there might be an aspect to reflect that, like an aspect of competitiveness. In that aspect there might be measures focusing on how many different feature we offer, how successful are our research and development activities, do we innovate or do we imitate. I am not sure whether these are in the scope of the model, but they might be useful for us.
- **Organization 2:** There might be measures related to the personnel, for example how well do the developers improve themselves, can they work multidisciplinary? It might be beneficial to measure the level of their work elasticity, their capabilities to work in different fields.
- **Organization 3:** I find the scope quite adequate and operable. Of course, there might be many additions; however, current context consists of fundamental points that require awareness. For me, models that are more detailed are utopic; but this model is not utopic, it is applicable. If an organization completes current context, there might be additional components for them, like training or human resources, but current version is adequate for essential needs.

- **Organization 4:** Current version of the model seems to be focused on project management. There might be aspects related to the results of the conducted work. In addition, there might be an aspect related to organizational agility. The term value is very important for agile, there might be an aspect measuring the value provided to the customer.

Are there any additional measures that are not covered by the model but are important for your organization to determine measurement capability and if so what are they?

- **Organization 1:** Under production aspects there are measures that we use but model does not include, we could not evaluate them. For example while analyzing the backlog, understandability of requirements and refactoring time can be added.
- **Organization 2:** Some employees are expected to specialized, but on the other hand working in a cross-functional fashion is also important. I would like to measure this cross-functionality; for example, can everybody understand every aspect of this software project, what is their level of understanding? Our organization used to include very specialized people; however, nowadays we aim to make them cross functional. I do not know what kind of a measure can be used, but it will be useful for us.
- **Organization 3:** As I stated in previous question, current context is adequate for fundamental needs.
- **Organization 4:** Measures related to how defect resolutions reach to customer, how fast defect resolutions are, and automatic testing can be included in the technical parts. Measures focusing on extreme programming and DevOps can also be included in the model.

What are your opinions about the generalizability of the assessment and identification of improvement opportunities provided by the method for agile organizations in general?

- **Organization 1:** I think it can be generalized for agile organizations. Especially small-sized organizations or organizations just starting to focus on maturity can gain quite benefit.
- **Organization 2:** Real software organizations utilizing agile approaches and do not exceed 50 people can use this model. In higher levels I am not sure, larger organizations may not able to apply. To apply successfully in larger organizations dedicated employees can be assigned, but I only worked on smaller organizations, I do not have experience in larger organizations.
- **Organization 3:** I am not just thinking about my current organizations, I am also considering my previous experiences; it is hard to manage a process in Turkey. When I first started my career it was much harder, but nowadays process improvement perspective is evolved. This model lists fundamental works. Organizations successfully implementing this model probably wish to continue with improvements; however if they start with a more detailed process improvement model they cannot move forward. If they start with this model, define where they observe hardship, and report them, they can identify the problems. This model can be used as a regular feedback tool.

- **Organization 4:** Model may be useful in general. Its context provides benefits as a minimum measure set; in addition, it can be used as a checklist by applying at certain times and regularly evaluating the results.

Are there any other points you would like to add about the method and its applications?

- **Organization 1:** I required explanations for some measures and practices, the definitions may be improved.
- **Organization 2:** Currently there is nothing I would like to add. I plan to apply identified improvements; I might have additional feedback after the application. As a first step, I will choose two or three measures and collect them. My first goal is to bring our missing measures to level 2. I will start to work with my team; we just do not know how to apply certain measures. We have theoretical knowledge but we do not have practical experience. Information like best practices about the measures can be added to the model.
- **Organization 3:** The model has 3 levels, this is a realistic approach for me. Previously I worked with CMMI and ISO 15504, both of them include heavy requirements that an organization that is new to the process improvements cannot easily meet. These models include 5 levels. 5 is in the extreme, 1 and 2 are too close to each other, but it is hard to transition to level 3. Organizations need to have higher volume of business for level 3. I think 3 level approach of this model is more operable. Other models include too many process breakdown, the grouping of this model is quite ideal. For many organizations, only this model should suffice. As an addition, for the maintainability and dissemination of this model some processes may be defined. There would be organizations that require assistance for these.
- **Organization 4:** I think model includes necessary components for its purpose and adequate for its aimed scope. I do not have anything to add other than my previous comments.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Salmanoğlu, Murat
Nationality: Turkey
Date of Birth: August 12th 1985
Place of Birth: Adana, Turkey
e-mail: muratsalman@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
MS	METU, Information Systems	2012
BS	METU, Business Administration	2008

WORK EXPERIENCE

Year	Place	Enrollment
2012-Present	METU, Informatics Institute, Ankara Turkey	Research Assistant
2013-2017	Bilgi Grubu Ltd., Ankara Turkey	Software Measurement & Process Management & Software Quality Consultant, and Researcher
2008-2012	ASELSAN Inc.	Engineering Management and Analysis specialist

FOREIGN LANGUAGE

English Fluent
German Beginner

PUBLICATIONS

1. M. Salmanoğlu, A. Coskuncay, A. Yıldız, O. Demirörs (2018) "An Exploratory Case Study for Assessing the Measurement Capability of an Agile Organization". In: Software Quality Professional Magazine, Volume:20, Issue:2, pp. 36-47, March 2018

2. M. Salmanoğlu, T. Hacaloğlu, O. Demirörs (2017) "Efort Estimation for Agile Sofware Development: Comparative Case Studies Using COSMIC Functional Size Measurement and Story Points". In: IWSM/Mensura '17, October 25–27, 2017, Gothenburg. Association for Computing Machinery
3. M. Salmanoğlu, O. Demirörs, A. Coşkunçay, A. Yıldız (2017) "Exploration of a Practical Approach for Assessing the Measurement Capability of Software Organizations". In: In: Mas A., Mesquida A., O'Connor R., Rout T., Dorling A. (eds) Software Process Improvement and Capability Determination. SPICE 2017. Communications in Computer and Information Science, vol 770. Springer, Cham
4. M. Salmanoğlu, A. Yıldız and O. Demirörs, "COSMIC İşlevsel Yazılım Büyüklüğü Ölçüm Yönteminin Kurumlarda Uygulanmasında Dikkat Edilmesi Gereken Noktalar," in 10. Ulusal Yazılım Mühendisliği Sempozyumu - UYMS'16, 2016.
5. M. Salmanoglu, K. Öztürk, S. Bağrıyanık, E. Ungan and O. Demirörs, "Benefits and Challenges of Measuring Software Size: Early Results in a Large Organization," in The Joint Conference of the 25nd International Workshop on Software Measurement (IWSM) and the 7th International Conference on Software Process and Product Measurement (Mensura), 2015.
6. M. Salmanoğlu, B. D. Yanik, F. N. Demir, Z. Gürel, and O. Demirörs, "Yazılım Kalite Maliyetleri Üzerine Bir Çalışma – Farklı Sektörlerden Proje İncelemeleri," in 9. Ulusal Yazılım Mühendisliği Sempozyumu - UYMS'15, 2015.
7. M. Salmanoglu, O. Demirörs and O. Türetken, "Exploration of a Method for COSMIC Size Estimation from S-BPM," in Proceedings of 6th International Conference, S-BPM ONE 2014, 2014, pp. 55–66.
8. A. M. Ertugrul, G. Yilmaz, M. Salmanoglu and O. Demirors, "The Effect of Highlighting Error Categories in FSM Training on the Accuracy of Measurement," in 2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement, 2014, pp. 152–156.
9. M. Salmanoğlu and O. Demirörs, "COSMIC Fonksiyonel Büyüklük Ölçüm Yöntemi için Bir Hata Önleme Modeli," in 8. Ulusal Yazılım Mühendisliği Sempozyumu - UYMS'14, 2014.
10. M. Salmanoglu and O. Demirörs, "Exploration of an Error Prevention Model for COSMIC Functional Size Measurement Method," in The Joint Conference of the 22nd International Workshop on Software Measurement (IWSM) and the 7th International Conference on Software Process and Product Measurement (Mensura), 2012, pp. 198 – 204.

CERTIFICATES

1. COSMIC Functional Size Measurer, Certificate, Common Software Measurement International Consortium, April 2011.
2. Introduction to CMMI for development v1.3, Software Engineering Institute (SEI), 2011.
3. CBAP international business analyst training, BA Works, 2010

4. Fundamentals of IBM rational CLEARCASE,Ayrotek,2012
5. Patent training, Ankara Patent, 2011
6. EU funding & programmes for Turkish organizations seminar, Europe Media, 2011
7. Process management and improvement, KalDer, 2010
8. Certificate international programme, School of Economics, Hogeschool INHOLLAND, 2008
9. Nyenrode/INHOLLAND business programme, Nyenrode Business Universiteit, 2007
10. Zentrale Deutschprüfung zum schulabschluss, German Embassy, 2003