QEVM: QUALITY INTEGRATED EARNED VALUE MANAGEMENT

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

BY

PINAR EFE

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

AUGUST 2015

QEVM: QUALITY INTEGRATED EARNED VALUE MANAGEMENT

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

Prof. Dr. Nazife BAYKAL ____________
Director, **Informatics Institute**

Prof. Dr. Yasemin YARDIMCI ÇETİN ____________
Head of Department, **Information Systems**

Prof. Dr. Onur DEMİRÖRS ____________
Supervisor, **Information Systems, METU**

Examining Committee Members:

Prof. Dr. Semih BİLGEN ____________
Computer Engineering Dept., Yeditepe University

Prof. Dr. Onur DEMİRÖRS ____________
Information Systems Dept., METU

Assoc. Prof. Dr. Aysu BETİN CAN ____________
Information Systems Dept., METU

Assoc. Prof. Dr. Altan KOÇYİĞİT ____________
Information Systems Dept., METU

Prof. Dr. Ali YAZICI ____________
Software Engineering Dept., Atilim University

Date: 24.08.2015

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 work.

Name, Last Name : Pınar Efe

Signature : ____________________

ABSTRACT

QEVM: QUALITY INTEGRATED EARNED VALUE MANAGEMENT

Efe, Pınar
Ph. D., Department of Information Systems
Supervisor: Prof. Dr. Onur Demirörs

August 2015, 152 pages

Software project management discipline is in need of valuable tools and techniques to complete projects in agreed scope, on time and within budget according to customer needs. Earned Value Management (EVM) is a simple and powerful performance management and feedback tool, which is widely used in project management with an increasing interest particularly in last decade. EVM clearly displays the project progress in terms of scope, cost and schedule and enables predicting future of the project based on the trends and patterns in the past. Even though EVM addresses and integrates three main elements of a project- scope, schedule and cost- , the quality aspect is not explicitly considered in EVM. Instead it is supposed to be implicitly included in the scope. This approach might work for several project management disciplines, but software projects require special attention at this point. They have substantial rework effort due to its essential characteristics. By defining quality, doing things right first time, these reworking costs are directly related with the quality. Quality factors are affecting project progress and future significantly. In this study, we propose a new EVM model, Quality Integrated Earned Value Management (QEVM), which integrates quality element into traditional EVM to monitor and control software projects better and more accurately. The model utilizes the concepts of the Cost of Quality (CoQ). We performed an initial exploratory case study to identify the shortages of EVM for software projects before developing the model. Later, we performed a multiple case study including six cases in six different companies to explore the applicability of the model.

Keywords: Earned Value Management, Software Project Management, Reworking in Software Projects, Cost of Quality, Failure Costs

ÖZ

KKDY: KALİTE ENTEGRE EDİLMİŞ KAZANILMIŞ DEĞER YÖNETİMİ

Efe, Pınar
Doktora, Bilişim Sistemleri
Tez Yöneticisi: Prof. Dr. Onur Demirörs

Ağustos 2015, 152 sayfa

Yazılım proje yönetimi, projeleri planlanmış içerik, zaman ve maliyetle, müşteri ihtiyaçlarına uygun bir şekilde tamamlamak için faydalı araç ve tekniklere ihtiyaç duymaktadır. Kazanılmış Değer Yönetimi (KDY), proje yönetimi alanında yaygın bir şekilde kullanılan ve özellikle son on yılda popülerliği artmış, basit ve güçlü bir performans yönetimi ve geri bildirim aracıdır. KDY, içerik, zaman ve maliyet açısından projenin ilerlemesini net bir şekilde gösterir ve geçmiş eğilim ve örüntülere dayanarak projenin geleceğini tahminleme imkanı sunmaktadır. Projelerin üç ana unsuru olan içerik, zaman ve maliyet boyutlarına odaklanmasına rağmen, kalite boyutu açık bir şekilde bu model içerisinde dikkate alınmamaktadır. Bunun yerine kalite boyutunun içerik boyutunun bir parçası olduğu varsayılmıştır. Bu yaklaşım birçok proje yönetimi disiplininde doğru olsa da, yazılım projeleri bu konuda özel ilgiye ihtiyaç duymaktadır. Yazılım projeleri doğası gereği ciddi bir şekilde yeniden yapma faaliyetleri ile karşı karşıya kalmakta olup, bununla ilişkili olarak kalite faktörleri projenin ilerlemesini sonradan önemli ölçüde etkilemektedir. Bu tez çalışmasında, yazılım projelerini daha iyi ve doğru izlemek ve kontrol etmek için, kalite boyutunu geleneksel KDY yöntemine entegre eden, KKDY adını verdiğimiz yeni bir KDY modeli geliştirilmiştir. Model geliştirilirken kalite maliyetleri kavramlarından yararlanmıştır. Modeli geliştirmeden önce KDY'nin yazılım projeleri için eksikliklerini belirlemek üzere iki projeyi kapsayan çoklu bir araştırma durum çalışması gerçekleştirilmiştir. Modeli doğrulama amacıyla da altı farklı şirkette altı projeyi içeren çoklu bir durum çalışması gerçekleştirilmiştir.

Anahtar Kelimeler: Kazanılmış Değer Yönetimi, Yazılım Proje Yönetimi, Yazılım Projelerinde Yeniden Yapma Faaliyetleri, Kalitenin Maliyeti, Hata Maliyetleri

To Cemile, Ramazan and Işıl Efe

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor Dr. Onur Demirörs for his guidance, encouragement, endless patience and continuous support throughout the years of my study. I could not accomplish this study without his support and inspiration.

I wish to express my appreciation to Dr. Semih Bilgen and Dr. Altan Koçyiğit for their support, critical suggestions and valuable comments throughout the steering meetings of this study.

My studies would not be complete without the collaboration of colleagues and friends from Siemens, IC Terra and Türk Telekom. I want to present my sincere thanks to my managers, Vedat Uslu from IC Terra and Dr. Selçuk Alimdar from Türk Telekom for their support.

I would like to thank the project managers, quality managers and all the other staff of the companies who contributed this research by their voluntarily support.

I am thankful to my friends, Dr. Nurcan Alkış and Dr. Özden Özcan Top for their friendship and close collaboration.

Special thanks go to Dr. Ayça Müge Sönmez for her encouragements, support and friendship.

My heartfelt thanks to my dear friends, Asena Şanlı, Berfin Aktaş, Burçay Erçetin, Gonca Gedik, Muharrem Görkem, Ömer Sarıkoç and Süheyla İkiz for their continuous encouragement, infinite tolerance and moral support in my never ending studies during years. I am very lucky to have them in my life.

A very special gratitude goes to my mother Cemile Efe, my father Ramazan Efe and my lovely sister Işıl Efe for their endless love and for being there anytime I needed

I was supported by TÜBİTAK BİDEB 2211 scholarship during my PhD study; thereby I would like to thank TÜBİTAK to giving me this opportunity.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AC	: Actual Cost
ACF	: Actual Cost with Failure
ACWP	: Actual Cost of Work Performed
APC	: Appraisal Cost
API	: Appraisal Cost Index
ANSI	: American National Standards Institute
BAC	: Budget at Completion
BCWP	: Budgeted Cost of Work Performed
BCWS	: Budgeted Cost of Work Scheduled
CA	: Control Accounts
CAP	: Control Account Plan
CC	: Conformance Cost
CCI	: Conformance Cost Index
CCPM	: Critical Chain Project Management
CMMI	: Capability Maturity Model
CoQ	: Cost of Quality
CoSQ	: Cost of Software Quality
CP	: Critical Path
CPI	: Cost Performance Index
CV	: Cost Variance
C/SCSC	: Cost/Schedule Control Systems Criteria
DoD	: Department of Defense
EAC	: Estimate at Completion
EIA	: Electronic Industries Alliance Association
EFQM	: European Foundation for Quality Management
ES	: Earned Schedule
EV	: Earned Value
EVM	: Earned Value Management
EVMS	: Earned Value Management System
EVM3	: Earned Value Management Maturity Model
ETC	: Estimate To Complete
ETFC	: Estimated Total Failure Cost
EFC	: External Failure Cost
FC	: Failure Cost
FP	: Function Point
LOC	: Lines of Code
IEEE	: Institute of Electrical and Electronics Engineers
IFC	: Internal Failure Cost
IS	: Information Systems
NASA	: National Aeronautics and Space Administration
PBEV	: Performance Based Earned Value
PC	: Prevention Cost

PCI : Prevention Cost Index
PMBOK : Project Management Body of Knowledge
PMI : Project Management Institute
PMO : Project Management Office
PV : Planned Value
SOA : Service Oriented Architecture
SPI : Schedule Performance Index
SV : Schedule Variance
TCFC : To Complete Failure Cost
TQM : Total Quality Management
UCP : Use Case Point
QCF : Quality Cost Factor
QEVM : Quality Integrated Earned Value Management
QPI : Quality Performance Index
VAC : Variance at Completion
WBS : Work Breakdown Structure
WP : Work packages

CHAPTER 1

INTRODUCTION

Software development industry is suffering from unsuccessful projects with exceeded budget, late in delivery and with low quality. Every year billion dollars are wasted on failed software development projects. According to CHAOS Manifesto of 2013 by Standish Group [1], only 39% of software development projects are delivered on time, on budget, with required features and quality goals, while 18% are totally failed by being canceled prior to completion or never used after delivery. The remaining 43% are counted as challenged being late, over budget and with less features than planned.

Poor project management is addressed as the main reason causing project failures and overruns. It is said that “Good project management does not guarantee the success of every project, but poor project management usually leads to failure”. Therefore, improving software project management provides invaluable benefits to the software industry.

The main objective of software project management is to deliver the project successfully, in other words, in agreed scope, on time and within budget according to customer requirements [2]. Hence, the project management success basically depends on integrating four dimensions of projects, scope, time, cost and quality.

There are several tools and techniques used in the project management to achieve project objectives successfully. The significance of these tools and techniques in the organizations is undisputable. Their value and benefits are widely accepted and well recognized [3]. In the scope of this study, we focus on one of the most simple and powerful project management tool that is basically used during project controlling: EVM.

EVM has been widely and successfully used in project management for more than 40 years and could be applied to any project at any industry. It objectively measures the project progress and performance and estimates the future of the project. In spite of its extensive use and success in project management, software projects do not utilize it much to its full potential compared to the other industries.

In this research, we mainly explore the applications and weaknesses of EVM for software projects and propose an improved EVM model to contribute its spread in software projects by considering quality costs and their trends.

This chapter initially presents the background of the problem with the problem statement. After the purpose of the study is described, research strategy is introduced. Finally, the last section presents the organization of the thesis.

1.1. **Background of the Problem**

Project management is defined as "the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements" in Project Management Body of Knowledge (PMBOK) Guide of Project Management Institute (PMI) [4]. The primary goal of project management is to complete the project within the agreed scope, time and budget. Project management life cycle basically consists of initiating, planning, execution, controlling and closing processes.

The principles of project management could be applied to any industries for any type of projects but those industries may have variations of the processes and terminology. Software project management is accepted as a sub-discipline of project management as well and the principles of project management have been applied to software projects since the 60s. However, diverse characteristics of software and software projects compared to other engineering disciplines make software projects more challenging to manage and to complete successfully. Among distinct characteristics of software, complexity, invisibility and changeability and their consequences are quite significant in project management perspective [5].

Software entities are remarkably complex compared to any other construct since there are no two parts alike in general [5]. Software cannot ignore or simplify details of the real world. Complexity grows exponentially as the size of the system increases. Since the software has no physical reality, it is not accurately modeled as in the case of, for example, construction. The software is continuously subject to change even after being completed. In contrast, manufactured things, such as buildings, cars, and computers are rarely changed after manufacturing. They are simply outdated by later models. For example, callbacks of automobiles occur reasonably infrequent.

In particular, because of the changeability, software projects are subject to continuous change and this makes planning and controlling very hard. Additionally, complexity and invisibility bring other challenges into management. In the light of these essential difficulties, the new concept that does not exist in traditional projects comes out as an inevitable part of a software project: reworking [6] [7]. It represents all the changes to the existing system as well as corrective actions of defective, failed or non-conforming items. Reworking itself introduces further complexity in terms of planning, estimating, monitoring and controlling. It could also cause further rework in a recursive cycle that can affect the project timeline. In a study performed at Raytheon, Dion reported that approximately 40% of the total software project budget was spent for reworking [8]. The studies show that the cost of rework can approach 50% of the project budget for the large software projects [6] [9] [10]. Reworking impacts the entire software development process from definition to implementation and testing.

Due to these characteristics of software projects and the lack of proper tools and techniques to handle these characteristics, software projects become very complicated in planning and monitoring and so more difficult to manage and complete successfully. They require particular attention. Even though the tools, techniques and methods used in the traditional project management have been utilized in software projects for years, in general, these traditional project management approaches cannot be sufficient for software projects in their traditional forms without adapting according to the difficulties of software projects [11]. They may not be so powerful without considering the challenges of software projects and may not meet the needs of software projects adequately. Therefore, it is necessary to adapt the tools and techniques according to the difficulties and particular needs of software projects.

EVM is a major and widely used quantitative project management tool to measure and track the progress and performance of a project objectively. It integrates three critical elements of project management success; project scope, schedule and cost and enables tracking schedule and cost variance between planned and accomplished work of a given project at any time [12]. EVM additionally allows management to predict the future of the project with the estimates including total cost at completion and the possible completion date based on the patterns and trends in the past. It is called as "Management with the lights on" since it sheds light on where the project is now and where it is going comparing to where it was supposed to be and supposed to be going [13]. It basically compares the planned work and accomplished work at a specific time in a project and calculates the value of this accomplished work. It is a forewarning system against the possible cost overruns and delays in the schedule and so provides a valuable opportunity to take necessary actions in order to calibrate the cost and schedule plans.

1.2. Statement of the Problem

Even though EVM is commonly accepted in project management and has been employed to a wide variety of projects of different sizes and complexities around the world, this powerful technique is still little known in the software industry [14].

Project management aims to deliver projects on time, with the agreed scope according to specified requirements, within the planned budget. This scope, schedule and cost constraints are called as triple constraint or project triangle. From the perspective of this study, the key point here is "according to specified requirements", which corresponds to the definition of quality. Thus, the project triangle implicitly covers the quality and assumes that scope means scope with quality. However, in software project management, it is not so appropriate to bundle scope with quality unlike the other types of projects. Even the task is completed, and scope target is achieved, it might require a few iterations on the task to fix the bugs, improve the features and so on. They are tightly related but not exactly together. In this research, we inspired from Crosby's quality approach and defined quality as doing it right the first time and measure it by the cost of nonconformance.

Software projects are suffering a lot from reworking [6] [8] [9] [15]. Based on the essential characteristics of the software projects and the quality related issues, reworking is accepted as a natural consequence and an indispensable part of the software projects. For the projects of the other industries like construction projects, reworking is not very common or acceptable particularly after some milestones. As a result, the consequences of reworking are not so visible for those projects in most cases.

In parallel to targets of project management, EVM in a traditional form focuses on the three main elements of software project success: scope, schedule and cost. It essentially emphases the quantity of the work performed. It does not deal with the quality explicitly.

EVM assumes quality a part of scope element. It means when the task is completed, it is supposed to be completed without any quality deficiency. Therefore, the quality shortages or issues are not considered explicitly in EVM. In traditional project management, those three elements would be enough by considering quality dimension already in the scope since there is no significant changeability as in software projects. Even though this approach might work for many project management disciplines, software projects could not utilize EVM properly because reworking is an inevitable part of software projects.

Software projects have significant rework effort directly influenced by quality factors. The quality of the work matters and affects the quantity as well in terms of reworking. A research shows that software specialists spend about 40 to 50 percent of their time on avoidable rework rather than on work that's done right in the first time [16]. Another study states that software vendors typically spend 30% to 50% of their development budget on detecting and fixing errors [17].

The main issue of EVM here is the volatility of the value earned. As we already experienced in the exploratory case studies, at a given time in the project, we calculated the earned value (EV), but after some time, it is seen that EV is less than formerly calculated one or just by spending more effort it reaches the previously calculated one. The value specified as EV is changing in time and previously calculated ones may not be definitely correct. It gives an incorrect picture to project managers. Any kind of reworking i.e. unpredictable changes, requirement and design errors, software bugs affect the EV. If we would do it 100% correct in every aspect for the first time, we would not have such an issue and we would have exactly the same EV in every calculation. As an example, the following is a very ordinary scenario in a software project: At a given time in the project, the task is completed and the scope is achieved, but after some time, it is changed due to defects, and more effort is spent. The scope is still the same but cost is more spent, it is not the cost of scope, it is the cost of quality or scope was not complete before. The EV calculated at the beginning would not be correct or some other values should not be correct to explain this situation.

EV is particularly significant and key data of EVM in order to reveal the current status as well as predicting the future of the project. It is vital to have EV as accurate as possible. For that reason, we need an improved EVM approach for software projects to calculate more accurate EV, to provide enhanced current and future estimates of the projects and to have an idea about the project quality status.

EVM does not represent the quality status of a project in any way. Since EVM does not consider the quality dimension explicitly, it does not measure anything related to quality and therefore does not give any clues to the project managers regarding the quality perspective and progress of the project. Quality is a vital success factor and the fourth dimension of a project and also affects the other dimensions, scope, schedule and cost in due course.

Applying traditional EVM could give software project managers incorrect information. Even at a specific time the project is supposed to be on track, the additional cost/effort would still be needed for the features that are already completed. Since these later costs may approach 50% of the total software project cost, this fact should not be ignored and needs to be carefully considered [16] [17] [18].

EVM is a powerful technique and could be improved considering the needs of software projects. The quality dimension should be taken into consideration separately for software projects. Additionally, in EVM, there are metrics covering scope, schedule and cost. As expected, there is no clue about the quality status of the project. There is nothing about failure costs (FC) or reworking percentages, nothing special of any quality assuring efforts including prevention, detection, fixing. It would be valuable if there exists such measures and metrics for project managers since it already affects the future of the project. This quality dimension of EVM would provide those quality measures. With all these backgrounds, adapting EVM according to the quality needs of software projects is seen mandatory. Therefore, the quality integrated version of EVM will generate more meaningful performance measures and successful future estimates for software projects in terms of cost, schedule and quality constraints.

1.3. **Purpose of the Study**

The purpose of this study is to improve EVM in order to make it more usable and valuable for software projects. Therefore, this study develops a new EVM model, which is called QEVM, as an extension to traditional EVM that incorporates the quality aspect by means of CoQ and provides quality related measures during tracking project progress and performance.

Such a model shall enable measuring project performance by means of quality aspect in addition to existing scope, schedule and cost. We design the QEVM model in such a way that it is compatible with PMI's project management principles and go further including quality costs explicitly.

The quality approach of the model is inspired from Crosby's quality approach: Quality is doing it right for the first time and is measured by the cost of failures. QEVM does not aim to evaluate the product quality. It does not propose any new metrics indicating product quality. QEVM is neither a method for Quality Measurement, intending to bring about the desired level of quality nor Quality Assessment, controlling and assessing the level of quality. Instead QEVM quantifies the quality status by means of quality costs. It only aims to use historical quality cost data and measure the current quality related performance with these quality costs. It proposes quality metrics indicating the quality status of the project.

The model describes the new quality related concepts and presents calculations subsequently. It introduces new quality metrics for performance analysis and forecasting. In addition to that, it updates the existing EV calculations with more accurate ones. The model provides a complete application guideline to set up quality measurements in the related processes to demonstrate big picture.

QEVM delivers valuable information to project managers by means of current quality effectiveness status according to the historical failure data using CoQ concepts. QEVM enables not only more accurate performance evaluations in terms of cost, schedule and quality, but better future forecasts for all of them.

1.4. **Research Method**

The literature review and qualitative research based on case studies constitute the research method applied in this study, which is presented in detail in Chapter 4.1 and Figure 8 presents the data flow diagram of research method.

This study consists of two main phases; development of the model and application of the model.

The first phase includes the literature review, exploratory case studies and model development. We initially started the research activities with an extensive literature review, including the researches on EVM, using EVM in software projects, extensions to EVM, software quality measurement, CoQ and CoQ applications for software projects. Afterwards, we conducted an exploratory, multi-case study with two cases in order to investigate the difficulties and shortages of EVM for software projects with our own hands-on experience and so observed the problems on application. Based on the findings of the literature review and the results of the exploratory case studies performed, we identified improvement opportunities for EVM in particular for software projects and formulated research questions. Afterwards, we developed QEVM considering all these improvement opportunities with our direct observation based on previous experiences.

In the second phase of the study, we followed multi-case study research strategy. We refined and improved the initial model after the initial qualitative experiment conducted. In total, we perform six case studies on six different companies to explore the applicability of the new QEVM model and answer the research questions. Every case project is selected carefully with different characteristics to evaluate the model on different project types. We collected metrics, evaluated direct and participant observations as multiple sources of evidences. Toward the end, we evaluated the results of the case studies, answered the research questions and provided the discussions.

1.5. **Organization of the Thesis**

The remainder of the thesis is organized as follows.

Chapter 2 provides a review of literature on EVM, EVM applications in software projects, extensions of EVM, software quality measurement, CoQ and CoQ for software projects. It also gives an overview of the method.

Chapter 3 describes QEVM. At first, it gives an overview of QEVM and defines the relation with the project management processes. Then, it presents the conceptual model in detail with the new concepts and metrics.

Chapter 4 provides a detailed description of the research approach and the research design in the first part. Afterwards, it presents an exploratory case study with two cases to observe EVM applications on software projects. In the last part, it provides the multi-case study with six cases in order to validate QEVM model. The descriptions of the case projects, the conduct and the findings are given in detail and the results are discussed.

Chapter 5 discusses the conclusions and summarizes the contribution of this research. Suggestions for further research are also provided in this chapter.

CHAPTER 2

RELATED RESEARCH

This chapter includes the results of the relevant literature review on Earned Value Management including its history, the overview of the method, the extensions, the drawbacks addressed and EVM usage in software projects. Additionally, it presents the review of literature on cost of quality concept and its usage in software projects too.

2.1. Brief History of EVM

The EV concept is not new. It has been used in industrial manufacturing since the late 1800s [19] [20]. In the early American factories, the industrial engineers measure the performance of "planned standards" using "earned standards" gained against "actual expenses". EV concept is used there in its most fundamental form.

EV was formally introduced as a project management tool by the US Navy, as part of the PERT/Cost methodology in 1962. Later in 1967, the US Department of Defense (DoD) formally issued Cost/Schedule Control Systems Criteria (C/SCSC), which incorporates the EV concept with thirty-five criteria and mandated their use on systems developed for DoD. On the other hand, the private industry did not accept this formal criteria concept and EVM as a part of C/SCSC was never adopted for use in the industrial projects except governmental contracts due to the complexity till the mid of the 90s. There were some concerns of the private industry and DoD to make EV more user-friendly and simple. Therefore, in April 1995, the group has been set-up for re-examining and re-writing of DoD's formal EV criteria with the target of making EV more useful project management tool. The group has been reviewed this formal guide and worked on the more lightweight EVM system to encourage wider use of EVM. In early 1997, the DoD revised and accepted thirty-two industry earned value criteria, known as the Earned Value Management System (EVMS) and added into DoD Instruction 5000.2R (DoD Directive 5000.2-R). Subsequently, these thirty-two criteria are gathered as a guideline under American National Standard Institute/Electronic Industry Association (ANSI/EIA). In 1998, ANSI/EIA-748 Guide was officially issued to the public [21]. Therefore EVMS became the private industry method as well as government obligation and the private industry started to maintain EVM.

After simplification of EVM, it was quickly expanded beyond the DoD and adopted by many governmental organizations, like National Aeronautics and Space Administration (NASA), United States Department of Energy and also the private industry. The method had become understandable and practical by project managers in addition to the EVM specialists. The construction industry was an early user of EVM in the private industry [22].

PMI included an overview of EVM concept in the first version of the PMBOK Guide and broadened in subsequent versions [4] [22] [13]. In 2005, PMI has published "Practice Standard of Earned Value Management" as a supplement to the PMBOK Guide to facilitate its role in effective project management [12]. This guideline is revised afterwards and the second edition is published in 2011 [23]. PMI's practice standard provides project managers simple and practical way for measuring project performance and projecting final results rather than the DoD's formal EVMS guideline.

The usage of the method has already been spread out to the other governmental agencies and the other nations such as Australia, Canada, Sweden [24]. In Australia, EVM has been released as standards, AS 4817-2003 and AS 4817-2006 [25] [26].

Table 1 EVM History and Progress Timeline

Year	Event
late 1800s	first use of EV concept in American factories
1962	initial introduction of EVM as a project management tool as a part of the PERT/Cost methodology by US DoD
1967	formal introduction of EVM in C/SCSC by US DoD
1997	revised and simplified EVMS by DoD, draft industry guideline
1998	ANSI/EIA-748-1998, formal industry guideline for EVMS
2000	simplified EVM terminology published by PMI in PMBOK
2005	Practice Standard for Earned Value Management by the PMI
2011	Practice Standard for Earned Value Management by the PMI, 2nd edition

A maturity model, called as Earned Value Management Maturity Model (EVM3), is proposed to assess the capability of an organization in applying EVM in 2000 [27]. EVM3 is a staged, 5-level maturity model for organizations to use in implementing and improving their EVMS. The organizations with an ANSI/EIA-748 compliant EVMS can use the EVM3 to establish EVM metrics and create EVMS improvement plans.

2.2. **Method Overview**

This section presents EVM overview mostly utilizing PMI's PMBOK and EVM Practice Standard [13] [4] [12] [23]. PMI defines EVM in its most fundamental and easy-to-use form. This standard gives project managers usable and easily implementable method. EVM aims to answer the following very critical management questions according to this standard [12]:

- "Are we ahead or behind the schedule?
- How efficiently are we using our time?
- When is the project likely to be completed?
- Are we under budget or over budget?
- How efficiently are we using our resources?
- What is the remaining work likely to cost?
- What is the entire project likely to cost?"

EVM has two major key practices, which are establishing a Performance Measurement Baseline (PMB) and measuring the performance against the baseline. The first one is basically related to the planning process group and the second one is for controlling process group. Briefly, EVM is all about planning and controlling/measuring progress and performance according to this plan.

In establishing PMB in the planning process, the complete work is decomposed to a manageable level, Work Breakdown Structure (WBS) is established, and responsibilities, resources and time estimates are assigned to the tasks. So, PMB is the initial complete plan of the project and maintained throughout the project. During controlling, resource usage and physical work progress are measured, EVM key elements are calculated and so with the EV metrics of variances and indices, performance and progress are measured against the baseline. Afterwards, cost and schedule performances are analyzed, future estimates performed according to the current performances using EV prediction metrics.

EVM has three key data elements, namely, EV, Planned Value (PV) and Actual Cost (AC).

PV is the sum of all the budgets for all planned work at any given time in the project schedule. It represents also established PMB, and known as Budgeted Cost of Work Scheduled (BCWS) in earlier versions. The project performance is measured against PV. PV is typically plotted with an S-shaped curve as cost versus time (see Figure 1).

EV is the value of the work progress at a given point in time, also known as Budgeted Cost of Work Performed (BCWP). EV is expressed in terms of PV, representing the amount of work accomplished. Figure 1 shows the graphical representation of EV versus PV, AC.

AC is the summation of the resources expended in accomplishing all work performed for the time phase. It is also known as Actual Cost of Work Performed (ACWP) in earlier EVM versions. Figure 1 shows the graphical representation of AC versus PV, EV.

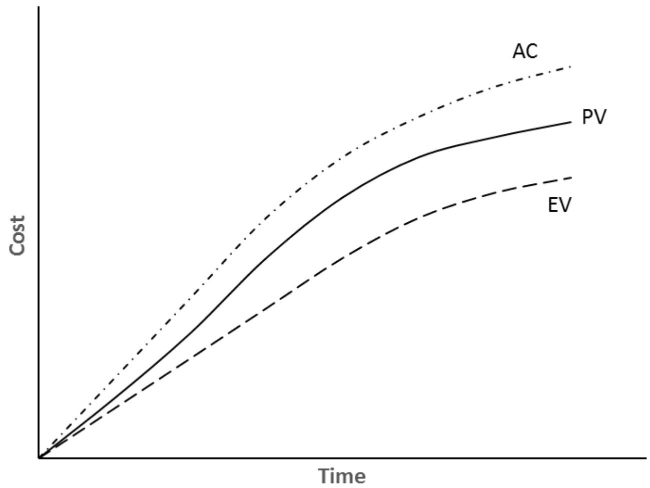

Figure 1 EVM Key Data Elements

These three key data elements basically ground a base for EVM. All the other following EVM metrics, including variances, indices and forecasts are derived from these three basic elements. The variances show the project current status clearly comparing these data elements. The indices are the indicators of how cost and schedule are efficiently used and represent the trends of the progress in the project. Based on the fundamental principle that trends and patterns in the past determine the future, the indices are used to predict the future of the project and project completion metrics are predicted. The project managers take necessary actions based on these metrics during project controlling. Figure 2 shows the graphical representation of EVM and variances.

Schedule Variance (SV) is the difference between the planned value of the work scheduled and the value of the work accomplished for the same time phase. It displays objectively how much the project is ahead or behind schedule.

SV= EV – PV (1)
SV > 0, ahead of schedule
SV < 0, behind of schedule

Cost Variance (CV) is defined as the difference between the values of the work accomplished and the actual cost incurred to perform the work; and utilizing this parameter, the percentage of cost overrun or underrun can be calculated.

CV= EV – AC (2)
CV > 0, under budget
CV < 0, over budget

Schedule Performance Index (SPI) is an index showing the efficiency of the time utilized on the project.

SPI = EV/PV (3)
SPI > 1, efficiency in utilizing the time allocated to the project is good
SPI < 1, efficiency in utilizing the time allocated to the project is poor

Cost Performance Index (CPI) is an index showing the efficiency of the utilization of the resources allocated to the project.

CPI = EV/AC (4)
CPI > 1, efficiency in utilizing the resources allocated to the project is good
CPI < 1, efficiency in utilizing the resources allocated to the project is poor

Budget at Completion (BAC) is the cost of total estimated work in the plan, located at the end of the PV curve.

Estimate to Complete (ETC) is the estimated cost required to finish all the remaining work, calculated when the past estimating assumptions become invalid and a need revised estimates.

ETC = (BAC – EV) / CPI (5)

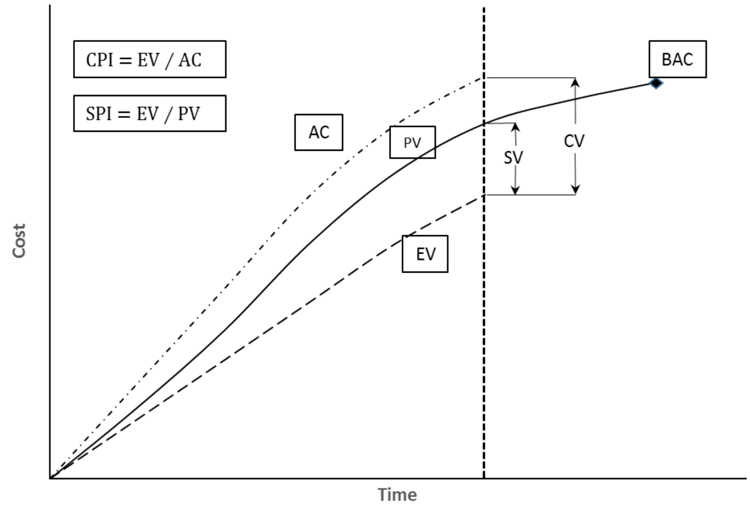

Figure 2 EVM Graphical Representation

Estimate at Completion (EAC) is the projected final cost required to finish the complete work and based on a statistical prediction using the performance indices.

EAC = BAC / CPI
= AC + ETC
= AC + ((BAC – EV) / CPI) (6)

Variance at Completion (VAC) is the variance on the total budget at the end of the project. It is the difference between what the project was originally planned to cost, versus what it is now estimated to cost.

VAC = BAC – EAC (7)

2.3. **EVM Limitations**

This section presents the literature review on the limitations and problems of EVM. Initially, the general problems and limitations of EVM are given as well as the solutions of the problems if proposed by the authors. Afterwards, the problems specific to software projects are explored.

One of the major drawbacks discussed in the literature is that EVM schedule indicators are reported in units of cost rather than time. In his famous article in 2003, Lipke criticizes traditional EVM both on measuring schedule performance, not in units of time, but rather in cost and giving unreliable time estimates and schedule forecasts through the end of the project [28] [29]. EVM measures schedule performance by means of cost units. Schedule variance, if greater than 0, does not reflect the variance of schedule instead it shows that there is a variance. Even we ignore this logical problem, EVM has another deficiency in terms of schedule indicators. EVM fails when the project is still ongoing after the planned completion date. For the project behind schedule, SPI converges and concludes at value 1 at the completion. SV behaves in the same way and it converges and concludes at value 0. This weakness of schedule metrics shows the progress and current status is incorrect using SV and also makes predictions unreliable using SPI.

Even we know that the project is completed late, the indicator cannot reflect it and shows that the project has perfect schedule performance. In order to overcome these deficiencies of EVM, Lipke proposes a new extension to EVM, called Earned Schedule (ES) that is detailed in the next section. ES provides time based measures that are based on the similar principles of EVM but using time metrics for schedule dimension [30] [31].

Similar to Lipke, Henderson states that while EVM has significant achievements in quantitatively expressing and analyzing project cost performance, the method is not so successful in terms of schedule performance [32]. He claims that since the cost is the units of schedule measures, the schedule indicators are not intuitive and EVM users need a period of adaptation before becoming familiar with the metrics. Furthermore, depending on the schedule indicators in units of cost, it is very difficult to compare with the time-based network schedule indicators like Critical Path (CP). He also stressed out the incorrect results of schedule indicators as a much more serious issue of EVM. He states that EVM schedule indicators always return to unity at project completion. EV equals the final PV and the BAC. Hence, the SV always equals to 0 and SPI always returns to 1 regardless of duration based delay. Additionally, the schedule indicators are not successful for projects passing the planned completion date.

In one of the earlier study, Brandon argues the reasons why EVM is not used more in the industry and proposes four main reasons [33]. First, he mentions that the awareness of EVM is minimal. It is not widely known in the corporate world and there are relatively little commercial books and articles on the subject. The second reason is the high cost of EVM in data gathering especially for estimating percent complete and obtaining actual costs. Another reason is the difficulty of EV reporting. Finally, fourth reason is the significant resistance of the employees and contractors when trying to put EV into practice. After defining these reasons, he proposes alternatives for overcoming these problems: appropriate work package sizing, simple yet accurate estimation of work package percent complete, simple spreadsheet reporting methods, and focus on the use of EVM for big project picture instead of individual evaluations. Brandon claims that if an organization effectively integrates EVM into their project management properly as described, then it will be the best single method for measuring and reporting project performance accurately and predicting project future reliably.

Lukas thinks that even though EVM is the most effective tool for the project performance, there are various errors in implementation [34]. He claims that a complete and integrated project plan is a foundation of successful EVM implementation. Based on his experiences, he listed the top 10 reasons why EVM does not work as the following:

- Not properly documented requirements
- Incomplete requirements
- Unused or unaccepted WBS
- Incomplete WBS
- Integration problems of plan (WBS-Schedule-Budget)
- Incorrect schedule and/or budget
- Ineffective change management
- Inadequate cost collection system
- Incorrect progress
- Management influence and/or control

He defines the reasons in detail and provides suggestions on how to overcome each specific problem.

Kim et al. [35] explores the ways of better EVM implementation in different types of organizations and projects (e.g. public and private, large and small projects), and develops a model for this purpose. They initially expose the following problems of EVM according to the source by means of six on-site case studies (see Table 2).

As a result of this research, they found out overall organizational approach, e.g. culture, top-level management support, organizational integrating mechanisms, effective training and so on, is needed for successful implementation of EVM. They propose a broader approach including four-factor groups, which are EVM users, EVM methodology, project environment and implementation process that together could significantly increase the acceptance, use and performance of EVM in different types of organizations and projects.

Table 2 EVM Problems by Kim et al.

Problems	Problem Source
Optimistic view of users in planning	User
Inaccurate assessment of EVM	User
Lack of understanding of EVM	User
Culture such as distrust	Culture
Poor image of EVM	Culture
Takes long time to train	System
Too much paperwork	Implementation
Lots of jargon	Implementation
Inaccuracy in high-tech projects	System
Too much rules	Implementation
Lack of user participation in designing EVM	Implementation
Projection based on historical data	System
Lots of costs	Implementation
Inconsistency between WBS and Org.	Implementation
Use of deterministic scheduling tech.	System
Need additional scheduling systems	System
Detailed WBS	Implementation
EVM weakens management power	Culture

The practical usage of EVM in software projects and its limitations have been studied as well in the following publications.

Fleming and Koppelman, who are the authors of the famous EVM book, “Earned Value Project Management” [19], state that EVM could be applied to software projects just as it is applied to the other projects [24]. They do not emphasize any variations in the method for software projects. Instead, they present a guideline to apply EV successfully on all projects and advise “Ten Musts to Implement Earned Value on All Projects”. They claim that if these following ten “musts” followed, the critical fundamentals of the EV concept are captured and the management of all projects from any industry is succeeded:

- Define work scope
- Create an integrated bottom-up plan

- Formally schedule control account plans (CAP)
- Assign each CAP to an executive for performance
- Establish a baseline that summarizes CAPs
- Measure performance against schedule
- Measure cost efficiency against the costs incurred
- Forecast final costs based on performance
- Manage remaining work
- Manage baseline changes

In her study of EVM application on IT projects, Ferle (2006) first focuses on software project management and highlights the factors affecting the project success [36]. She also points out the difficulties specific to software project management. She states that implementing EVM on software projects is not trivial and explores the complexities in planning, monitoring and reporting processes from the view of EVM. She also claims that effort estimation is extremely difficult, even impossible in certain situations and explains various reasons for this uncertainty and points out that being totally human oriented is one of the significant difficulties. The effort relies on the skills of the developers. Also, the significance of initial estimate, no matter how accurate or inaccurate, is emphasized. Last but not the least, during monitoring, the subjective assessment of task progress makes the visibility of progress challenging.

Hanna (2009) presents the challenges specific to software management and their effects on EVM [37]. He categorizes these challenges under three groups as "Innovation and Prototypes", "Error Discovery and Resolution" and "Architectural Changes". He proposes several approaches to make a software project more appropriate for EVM and then explains how these solutions could be applied. The first challenge represents the innovative characteristics of software projects that are in general never developed before and it brings the large uncertainty to the estimates of cost and schedule. This uncertainty might result in many iterations and prototypes and it is not known how many iterations are needed as well as how much effort each prototype takes. He mentions that these iterations and prototypes result in unplanned rework and it adds to the project new scope and needs re-planning effort. So, it makes EVM results inaccurate. The second challenge is the defects of software and their resolution timeline. He states that all software has defects and development process requires fixing defects of the previous releases in addition to developing the features of the new release. Depending on the maturity of the development process and the project complexity, the number of these defects discovered in the different phases would have large fluctuations. Therefore, the cost and schedule of the defects is not easy to predict, may result in a large amount of volatility and long delays with no positive progress. That makes performance indices incorrect and results in inaccurate forecasts. The last challenge is the architectural changes made during time progress and after more technical knowledge gained. This architectural change can again result in unplanned rework and inaccurate estimations. It also brings new defects being introduced into the software and so the challenges related with defect discovery are also included in this challenge. After presenting comprehensive overview of those challenges, he proposes the following approaches to overcome these challenges:

- Measuring volatility through metrics
- Improving task definitions by means of PERT analysis, confidence intervals, short duration tasks

- Improving scheduling techniques by means of resource loading, weighted milestones, task queuing technique
- Improving cost account structure including deliverables based WBS, separating repair and maintenance cost in WBS and also separating management cost
- Using incremental implementation approach with prioritizing high risk tasks and high value tasks
- Identifying and calculating bias including estimation bias and financial bias
- Measuring component volatility

Solomon criticizes traditional EVM in several respects [38]. Firstly, even though EVM was not designed to manage risk and does not even mention the risk subject, it is perceived and used to be a risk management tool. Secondly, EVM covers only the project work scope and disregards the product scope and requirements. Thirdly, EV is a derived measure. Subsequently, its effectiveness depends on reliability and accuracy of its base measures. Fourthly, EVM does not require precise, quantifiable measures. Objective EV methods are preferred, but management assessment may be used to determine the percentage of work completed. Finally, EV is a measurement of quantity, not quality, of work completed. It is the responsibility of the project manager to ensure that EV also measures the quality and technical performance of work products instead of just the quantity of work. He claims that EVM is only reliable and accurate if the right base measures of technical performance are selected and if the progress is objectively measured. Based on these limitations, he proposes an extension method to EVM, Performance Based Earned Value (PBEV), which we described in the next section.

In an earlier study, Christensen states that it is difficult to use EV methods on software development projects since the models that estimate cost and schedule and the metrics for measuring work progress are insufficient [39]. He criticizes the lack of standardized metrics of software development processes in those areas and so the managers need to decide appropriate metrics for each phase. Therefore, he proposes the following seven metrics that help to apply EVM to the software projects better and easier:

- Requirements and design progress; the number of requirements determined
- Code and testing progress; the number of components designed, coded and tested
- Person-months of effort; the effort of the project in person-months (the cost of software development is almost entirely labor-related)
- Software size; the size of software (e.g. Lines Of Code(LOC))
- Computer resource utilization; the measure of the available computer hardware timing, memory, input/output resources consumed by the software
- Requirements stability; the number of changes made to the requirements
- Design stability; the number of changes to the detailed design

In his another study, Brandon discusses the integrating EV into the management of software development projects [40]. He initially states the reasons of difficulties in software project management as the following:

- Most of the project cost is typically labor
- This labor is usually varying productivity, even within the same job category
- Quantitative methods of measuring task progress are immature
- The technology and accordingly associated tools are changing rapidly

- Applications are developed in new environments where prior estimations are bare, so estimations become less reliable
- There are unrealistic goals and pressures on the project teams to deliver software better, faster and cheaper.

After giving these reasons, he proposes best practices on work package sizing, defining complete percentages, reporting the results for software projects in particular labor perspective.

Agile software development methodologies attract much the software community's attention since their initial introduction and have gained significant acceptance in the software industry. Since the characteristics of agile development methodologies are quite different than the traditional ones, applying traditional EVM may not be so convenient for agile projects. With the increasing interest and acceptance both in agile methodologies and EVM in last decade, using them together is also the interest of researchers.

Wu (2011) discusses the limitations of EVM and compares the processes of traditional EVM and agile EVM [41]. He points out the key success criteria of traditional EVM as the quality of baseline plan, tracking actual performance carefully, re-planning the baseline and taking corrective actions based on the EVM performance results. He also criticizes EVM that it has no native quality related metrics, and so it cannot measure quality objectively. By means of EVM, the project can be on time and under budget but still might be low quality if it does not satisfy customer requirements. Furthermore, he puts emphasis on determining objectively the percentage complete and states that it is a difficult but vital task of EVM.

In his study in 2009, Rusk emphasizes that agile and EVM are a natural fit for each other and EVM implementations can be radically simplified for agile projects [42]. Moreover, the techniques used in agile project management such as burn charts provide status and progress information quite similar to what EVM provides [43]. In order to adjust EVM to the agile projects, Agile EVM that is introduced in the next section is proposed.

2.4. **EVM Extensions**

The use of EVM has significantly increased in last fifteen years with the simplification of the method by PMI. Consequently, a considerable amount of research has been conducted to utilize it more effectively for specific situations and to overcome the related difficulties identified in the previous section [44]. As a result of these researches, various extensions of the method are introduced. This section presents the bunch of major EVM extensions proposed in the literature.

ES is an extension technique to EVM, which is introduced in 2003 by Lipke in order to overcome schedule related deficiencies in traditional EVM method [28]. The technique defines a new metric that is called as ES based on the idea of tracking project schedule in units of time rather than traditional EVM units of cost. Therefore, ES is an advanced technique resolving the problems of EVM in schedule analysis. It is completely derived from EVM and requires no additional data for measures. Since the ES schedule performance indicators are in units of time, they are easier to understand and interpret.

The second version of PMI Practice Standard for EVM contains ES as the method in the appendix as “A method for extracting time-based schedule information from EVM data” [23].

PMI defines ES as the time duration where EV equals PV [23]. It measures the scheduled work completed, expressing in the time based unit of measure being utilized (e.g., week, month). The ES metrics reflect the progress clearly for schedule and enable predictions accurately, as EVM provides them for the cost. The related schedule variance and schedule performance indices are calculated based on ES. ES defines SV and SPI in two separate dimensions: cost and time. The traditional ones are called as SV($) and SPI($), and the new time related ones are called as SV(t) and SPI(t).

Agile EVM is a light-weight adaption of EVM for agile project management. The idea of using EVM together with the agile approach was first proposed by Lett from Lockheed-Martin in 1998 [45]. Afterwards, Agile EVM has been evolved with the several contributions [46] [47] [48] [49] . Agile EVM does not have a target to replace current agile metrics. It is just an additional one to existing others to increase the visibility of the project status and to support decision making. It is light-weight and easy to implement. Since it uses previously existing metrics of the agile projects and puts very little effort to the current process, it is quite cost effective to apply.

Agile EVM particularly uses the values defined in Scrum and contains a simplified set of EV calculations adapted from traditional EVM. Agile EVM uses the product backlog instead of PMB and tracks the progress with respect to the releases. A release corresponds to a phase applying Agile EVM and may include iterations or sprints or might be a time-interval, weeks or month. Agile EVM uses the story point in most cases to estimate the size of backlog items but any other consistent estimate of size can be used like hours, days. Total story points are estimated at the beginning of the releases and tracked accordingly. It combines PV, AC and EV on Release Burndown Chart and Release Burn up Charts instead of EVM graph and estimates velocity by means of EV. One of the main differences of Agile EVM comparing to traditional EVM is to embrace the changes in parallel to the characteristics of agile development principles. Agile EVM puts the changes into the releases as both added to or removed features.

In a publication in 2006, the use of Agile EVM is empirically validated by demonstrating the mathematical relationship between EVM calculations and Scrum metrics [49]. The publication initially describes Agile EVM in detail, compares the traditional EVM applications with Agile EVM applications and then presents the results of empirical validity test of the Agile EVM by applying it on two projects.

PBEV is again an extension to traditional EVM particularly focusing on software projects. It is introduced by Solomon in 2001 [50]. He initially concentrates on the existing measure shortcomings of EVM and possible EVM improvement opportunities based on the best practices and lessons learned by the Northrop Grumman team in developing weapon system software. The specific measures for requirements, requirement status, component status, test status, increment content-components, and increment content-functions are defined in three categories, which are functional size and stability, work unit progress, and incremental capability, to track progress against a plan using EVM. The data items and completion criteria are described for those measures.

Solomon published the details of PBEV in a book and in the subsequent publications [38] [51] [52] [53]. PBEV is proposed to overcome shortcomings of EVM related to measuring technical performance considering customer requirements. The method is developed based on standards and models for systems engineering, software engineering, and project management.

PBEV aims to handle the shortcomings of traditional EVM that are described in the previous section. Using CMMI, PMBOK Guide, EIA 632, IEEE 1120 and author's experiences, the PBEV guideline defining step-by-step activities is established. The distinguishing feature of PBEV is its focus on the customer requirements. PBEV includes the product scope and product quality requirements. The performance-based measures of progress for satisfying product quality requirements are specified. The progress is measured using those performance-based measures [52] [54].

In addition to these mature and well-established methods, there are further researches on applying EVM with other methodologies.

Using Critical Chain and EVM together has been the topic of a few researches [55] [56] [57] [58]. Critical Chain Project Management (CCPM) basically uses buffer management while EVM assesses the performance of a project accurately. Although it seems that Critical Chain and EVM are in conflict [58] and could not be applied in a project together, those researches particularly focus on combining Critical Chain with EVM and so getting the advantages of both separate methods. Leach states that a successful integration of CCPM and EVM is possible and it provides a complete solution for the project management [55]. He claims that they are complementary and suggests using CC to manage timelines and EVM to manage cost. Baker states that combining CCPM and EVM gives the project manager the ability of the dealing with the uncertainty [57]. The other study shows that EVM provides early warning of feature creep and subsequent buffer penetration in CCPM [58].

Incorporating Function Points (FP) into EVM is also the interest to the researchers. In this approach, the researchers use FP as means of quantifying the work and calculate cost and schedule performances accordingly [22] [59] [60] [61]. In a similar way, tracking software projects with EVM and Use Case Points (UCP) is the topic of another research performed in 2008 [62] [63] [64]. The idea here is to use more objective measures for software projects to represent the baseline consistently and measure the progress more accurately by means of FPs or UCPs.

Stratton proposed Earned Value Management Maturity Model (EVM3) to assess the capability of an organization in applying EVM [27]. EVM3 is a staged, 5-level maturity model for organizations to use in implementing and improving their EVMS. The organizations with an ANSI/EIA-748 compliant EVMS can use the EVM3 to establish EVM metrics and create EVMS improvement plans. In his book, he explains fundamental tools required for an effective EVMS and defines the maturity model describing metrics to measure the status and efficiency of EVMS implementation. He also gave practical examples through a comprehensive case study that includes a fictional company and a project manager.

2.5. **Cost of Quality**

CoQ is a concept used essentially in quality management and process improvement and basically covers all the cost spent for quality related activities. It represents the sum of both conformance costs (CC) and non-conformance costs (NCC), which are total costs spent on a project to have quality and to remove quality problems. Especially in manufacturing industry, CoQ is the initial step of process improvement activities and widely used on different quality management systems like Total Quality Management (TQM), European Foundation for Quality Management (EFQM), Six Sigma and so on. CoQ also provides vital metrics reflecting quality investment on a project and besides it clearly displays the possible improvement areas on the topic of quality.

PMI defines CoQ in PMBOK as "a method of determining the costs incurred to ensure quality" [13]. Initially, Juran mentions the CoQ analysis in 1951 in his book, Quality Control Handbook [65]. Juran defines quality as "fitness for use" and underlines the balance between product features and being free from deficiencies [66]. Moreover, he claims that quality is characterized by means of quality of design, quality of conformance, availability and full service. Juran defines the cost of poor quality as the sum of all costs that would disappear if there were no quality problems.

Later in 1956, Feigenbaum describes CoQ under four main cost categories as Prevention Costs (PCs), Appraisal Costs (APCs), Internal Defect Costs and External Defect Costs [67]. Afterwards the concept has been used and further improved by their followers.

With Crosby's book "Quality is Free", CoQ concept becomes more popular. Crosby defines quality as "conformance to requirements". He states that it is necessary to define measurable product characteristics based on the requirements and to use numerical specifications to quantify these characteristics [68] [69]. Crosby introduced the concept of "zero defects", which states that things should be done right the first time. Crosby opted for the prevention of errors rather than correcting them since he claims that the cost of prevention is lower than costs of detection, failure and correction [68]. His famous phrase "quality is free" represents this preventive approach and accordingly the measurement of quality is the price of non-conformance.

Crosby's CoQ approach is very similar to the approach of Juran but he defines four categories under two main categories, Cost of Conformance and Cost of Non-Conformance instead of cost of good quality and cost of poor quality [68].

Cost of conformance targets to achieve quality and consists of prevention costs and appraisal costs. PCs are the costs of activities planned to prevent poor quality and help to avoid future APCs and FCs. APCs are the cost of activities assessing and measuring the level of quality achieved by the process. There is a strong relation with PCs and FCs. In general, preventing errors will reduce FCs. PCs and APCs have the similar correlation. The more PCs result in the less APCs and FCs.

Cost of non-conformance occurs due to lack of quality and contains internal failure costs (IFC) and external failure costs (EFC). IFCs are the costs of correcting activities of the errors before delivery to the customer. These errors might be caused by both errors in development and inefficiencies in the processes and would have led to the customer not being satisfied unless fixed. They might occur in any phases of the project. EFCs are the costs of correcting activities of the errors after delivery to the customer. So, they additionally might damage the reputation and cause loss of trust.

The sum of PCs, APCs and FCs constitutes the total CoQ. The APC and PCs increase when investments are made to improve product quality and hence it decreases the FCs [70] [71].

2.6. **Cost of Quality for Software Projects**

Cost of Software Quality (CoSQ) is the term used for software projects in place of CoQ. The concept of CoQ has been adapted for the software projects. CoQ and CoSQ terms are used interchangeably through the thesis.

In manufacturing industries, CoQ is a well-recognized technique in process improvement activities but software industry does not utilize it as much as the manufacturing industry in the past [72]. CoSQ is becoming increasingly important as a driver in software quality in recent years [73].

Just like the four categories of CoQ defined in the previous section, CoSQ has the same categories and the following specific activities under these categories:

- Prevention costs: The cost of all activities regarding avoiding quality problems including quality planning, process improvement, training, standards and policies, prototyping, process improvements, documentation, metrics collection and analysis
- Appraisal costs: The cost of all validation and verification activities including reviews, walkthroughs, code reviews, inspections, testing and audits
- Internal failure costs: The cost of failure analysis, re-designing, rework, fixing errors, updating documents due to the changes, re-integration, re-testing and downgrading
- External failure costs: The cost of all activities regarding complaints from customer, including technical support, rework, retest, production, upgrade, document update, penalties and warrantee work

Even though the costs of software quality has been a topic of interest for over 30 years, there is limited research available in particular focusing on the empirical results. In 2011, the systematic literature review on software quality cost summarizes the latest research results by analyzing 87 articles published between 1980 and 2009 in 60 journals [74]. Based on the review results, the authors stated that software quality cost research is more focused on the model building and theory generation. Only one-third of the analyzed articles presents empirical study. They conclude that it is not insufficient for software quality cost research, since software domain needs quantitative data to generate new findings.

The empirical studies measuring CoSQ shows that how quality costs are significant in overall costing and how the maturity of an organization affect the quality. One of the earliest research investigating the relation between CoSQ and CMM levels by Knox presents the prediction model about the quality cost and rework expectations based on the maturity of the software organization leveled by SEI CMM [70] (see Figure 3). Knox utilized CoQ model developed in the manufacturing and extend his experience to software organizations in order to develop this prediction model. The model maps four components of CoQ with the CMM maturity models and the total CoSQ has a decreasing trend as the organization becomes more mature. The model represents that even CoSQ increases 60% of total development cost for a CMM-1 level organization, it reduces to 50% of total cost for a Level 3 organization and almost to 20% for a Level 5 organization.

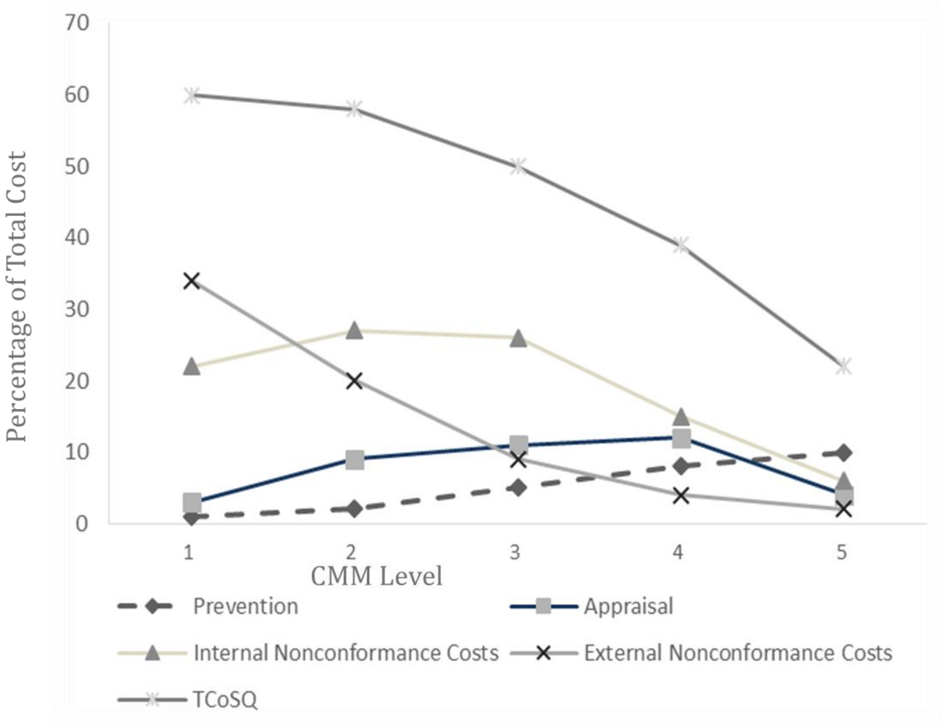

Figure 3 CoSQ as a percent of development and CMM Levels

In another study, Krasner et al. discusses the rationale of CoSQ after analyzing the nature of software [72]. Then he propose a CoSQ model with three cost components; (1) the efforts to prevent poor quality, (2) appraisal efforts performed for the detection of the poor quality and so the achievement of acceptable quality, (3) the efforts for handling non-conformances due to a lack of quality. Krasner points out the significant difference between the values of typical manufacturing CoQ and CoSQ. While typical manufacturing CoQ is ranging from 5% to 25% of company sales, CoSQ is varying from 10% to 70% of development costs [72].

The studies in Raytheon Electronic Systems in 1993 and 1996 successfully proves CoSQ model is a very good estimator of cost elements and correlates well with CMM maturity based on the results of a multi-year process improvement program from 1987 through 1996 [75] [8]. The study shows that the CoSQ is up to 67% when the company was at CMMI level 1 and it is around 40% when the company was at CMMI level 3 [75].

In 2000, Demirors et al. conducted a case study with two projects in a public software development organization in order to utilize CoSQ as an initiator for software process improvement activities [76]. External failure cost of the first project was about 6% while it was calculated around 21% of total cost for the second one in their study.

The more recent paper in 2012 presents the results of a quality cost measurement conducted in a software development department of a railway company [18]. In the study, CoSQ measurement approach is also proposed as the following and detailed explanation is given accordingly:

- Identification of the CoSQ related tasks
- Development of a list of CoSQ related tasks
- Categorization of those tasks
- Introduction and application of weight factors
- Determination of the accuracy of the weighting rules

This case study shows that the CoSQ represents 33% of the overall cost. In a detailed way, 10% of development cost is correction costs, 2% of that is prevention costs and 21% is the appraisal costs. The author compares the results with the previous studies performed by Dion and Krasner, and states that they confirm and validate their data model introduced in the study [8] [72].

CHAPTER 3

THE PROPOSED MODEL: QUALITY INTEGRATED EVM

This chapter presents the proposed Quality Integrated Earned Value Management model, QEVM. The main purpose of this model is to provide an infrastructure to monitor and control software projects better and more accurately by means of simple and powerful EVM additionally considering the specific needs of software projects. It incorporates quality element into traditional EVM and measures progress and performance of the project explicitly considering quality together with scope, schedule and cost. QEVM introduces new quality related measures in addition to existing ones provided by traditional EVM. Measuring quality by means of QEVM is based on the concepts of CoQ. The model evolved into this latest version through one major update after the review of thesis committee and the first application on a case project.

This chapter initially starts with QEVM overview and continues with the relations with the project management process. Then the measures of the QEVM are defined and the method for performance analysis and forecasting are described.

3.1. Overview

QEVM is an extension model to traditional EVM that is used in project management for 40 years. Using the core concepts and the fundamental approach of EVM, QEVM adds a new dimension, quality, to the method so that it brings new performance measures, improves existing ones, and enhances related future estimations. This extension is not only an EVM parameters improvement, but also a complete management system model integrating the quality costs.

The objectives of QEVM are summarized as the following:

- to provide a model, which is integrating quality element into traditional EVM
- to discern the quality status of the project by means of EVM in addition to scope, schedule and cost
- to estimate project progress more accurately at any given time using past quality cost data
- to offer more realistic future forecasts

Having incorporated quality dimension into traditional EVM, QEVM is based on four essential constituents of a project: scope, schedule, cost and quality (see Figure 4). It adds a new element, quality, and defines quality related concepts, measures and analysis. The quality element improves the calculation of EV and makes quality costs visible in project management so it affects cost measures significantly and reflects the project cost status more accurately. It provides improvement in schedule measures by a more realistic EV calculation. Additionally, it modifies future- related estimates again by means of improved EV calculation and reworking costs.

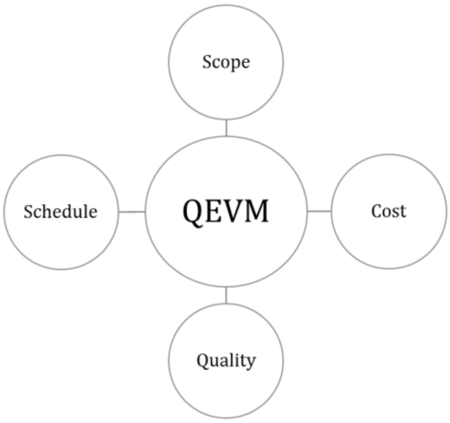

Figure 4 QEVM Elements

The quality element makes use of the concepts of CoQ. CoQ has four parts under two main categories, conformance costs and non-conformance costs, as explained in Chapter 2 in detail. Here in this model, non-conformance costs, which include IFCs and EFCs, are the main focus to estimate reworking cost and integrate quality dimension into the new model. On the other hand, CCs are also the interest of the model, especially in terms of providing benchmarking. QEVM uses the term “failure costs” for non-conformance costs through this thesis. The list of main terms used and definitions in QEVM is given in Table 3. QEVM by itself does not introduce new metrics regarding the product quality. QEVM is neither a method for quality measurement, intending to bring about the desired level of quality, nor quality assessment, controlling and assessing the level of quality. Instead, QEVM quantifies the quality status by means of quality costs. It only aims to use historical quality cost data including quality investment and reworking costs and so measure the current quality related performance using these quality costs.

Traditional EVM estimates the future of the project based on the fundamental principle that trends and patterns in the past determine the future. It uses schedule and cost indices to predict possible completion dates and eventual budget at completion. QEVM uses the same approach and so makes use of that fundamental principle not only for future estimates but also for current calculations. QEVM assumes that the calculated EV at a given time may not be strictly correct and would change after some time as a result of, possibly, reworking of the completed tasks; it analyzes the past trend and pattern of this change to calibrate current EV.

QEVM is neither a method for quality measurement, intending to bring about the desired level of quality, nor quality assessment, controlling and assessing the level of quality. Instead, QEVM quantifies the quality status by means of quality costs. It only aims to use historical quality cost data including quality investment and reworking costs and so measure the current quality related performance using these quality costs.

Table 3 QEVM Terms

QEVM Element	Explanation
Task	The manageable units of work, WBS item, corresponding to Control Accounts of EVM
Phase	The period of QEVM application (typically 1-2 weeks or month)
Planned Value	Planned effort of the task
Actual Cost	Actual effort spent for the task
Failure Costs	Reworking effort resulting from the tasks not conforming to requirements, happens due to lack of quality, total failure cost of CoQ
Reworking Cost	Same as failure costs, interchangeably used in QEVM
Internal Failure Cost	The failure cost which occurs during development, especially before customer delivery, but QEVM enables to define it for any level or milestone, depending on the project, for example, before system test or before field trial and so on.
External Failure Cost	The failure cost which occurs after customer delivery or any other pre-defined milestone
Conformance Costs	Total effort spent to have better quality, achieved with prevention and detection activities, Conformance Costs of CoQ
Prevention Costs	Efforts spent to prevent the failures and so reworking, Prevention Costs of CoQ
Appraisal Costs	Efforts associated with measuring, evaluating or assessing to assure conformance to requirements and quality standards, Appraisal Costs of CoQ
Total Cost of Quality	Total effort spent for having quality, including both investment and failure costs
Quality Cost	Used for referring failure costs

QEVM suggests utilizing the trends and patterns of quality costs. Trends and patterns are typically specific to a particular project's earlier calculations. Organizational trends or industrial benchmarking values could also be used to spot trends and patterns. The most convenient one is apparently project specific values whenever possible. However, depending on the project type, the trends and patterns may not be visible till the mid of the project. In such cases, again depending on the project characteristics, the organizational or industrial values could be useful.

QEVM uses effort instead of money as the cost unit since it focuses on the software development projects. Since the main cost driver of a software project is the effort of the project organization, it is very common to track software projects and even apply EVM by means of effort. The effort is addressed entirely through the study by "cost" terminology.

QEVM has three different approaches depending on the relation of reworking effort to the origins, which are task-based, phase-based and project based:

- Task-based approach tracks the FCs for every task. This approach might be too detailed and a bit difficult to apply comparing to simple EVM. It enables to spot problematic tasks and may lead the project manager to take preventive actions for those specific tasks.

- Phase-based approach requires the FCs of the tasks that are completed in the specific phase. This approach provides more detailed view compared to project-based approach. It is possible with this one to track the quality trends of specific phases such as requirements analysis, design and so on. Those trends would give insight to the project managers for the later applications. It may also let problematic tasks/phases be identified effectively and easily and would lead the project manager to take preventive actions accordingly.
- Project-based approach needs the FCs of the whole project. This is the most common usage in the projects since the origin of the errors is not tracked and reflected. It is easier to implement than the other approaches and gives a general idea about the project in a broader sense. It could be applied in any case if we have FCs somehow. It is similar to the approach that CoQ follows. The project-based approach is more suitable with cumulative basis application. It does not make sense to apply it phase-by-phase.

The project manager has to decide the application approach at the beginning of the project and collect data accordingly. It is possible to apply the project-based approach in any case but the phase-based and task-based approaches require the link of the reworking with the origin phases or tasks.

By means of QEVM, it is possible to monitor a project on both phase-by-phase basis and/or cumulative basis. If the project manager wants to particularly measure the performance of the phases or to measure the performance of the phases implemented by different teams, QEVM should be applied phase-by-phase. In this implementation, we will have different qcf values for every phase. The mathematical models or regression techniques could be used to select qcf. On the other hand, to measure the overall progress of the project, the cumulative application is required. It is possible to use both types of application together to provide more insights to project managers. Even for cumulative application type, it is possible to use phase-based or task-based approach.

QEVM is designed to overcome the shortcomings of EVM with regard to quality cost, especially neglected high reworking efforts. The distinguishing feature of QEVM is its focus on the quality costs and tracking and utilizing the trends of quality costs in project management. QEVM provides principles and guidance for better and more accurate measurement of cost, schedule and quality performance.

3.2. QEVM and Project Management Process

QEVM requires the application of project management principles outlined in PMBOK® Guide [4]. The process of QEVM is very similar to EVM and it adds new activities and practices to existing ones. Project management primarily consists of planning, executing and controlling work. As a performance management method, QEVM brings new practices especially into project planning and controlling areas with the aim of measuring, analyzing, forecasting and reporting cost, schedule and quality performance and taking actions by project managers (see Figure 5).

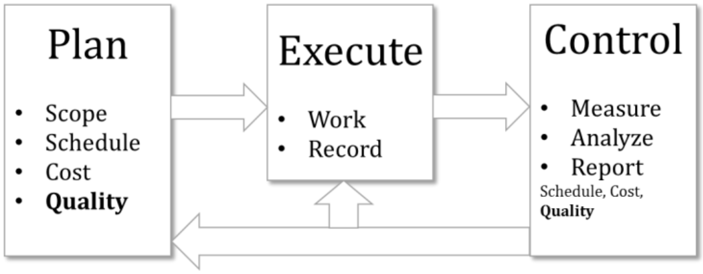

Figure 5 QEVM and Basic Project Management Process

In project management cycle, QEVM has two main key practices [23]: Establishing a PMB in the planning process and measuring and analyzing performance against the baseline in executing and controlling processes. Figure 6 illustrates an overview of QEVM activities.

In project planning process, EVM needs planning all the work and establishing PMB. Initially, project work is decomposed into executable and manageable tasks using work break down structure and resources are assigned to the tasks. After all the project work is scheduled, the project scope, schedule and cost are integrated into a time-phased budget, which is PMB.

QEVM adds three new activities to the planning process:

- Identifying the tasks related with prevention and appraisal activities
- Calculating and benchmarking quality investment cost
- Planning rework explicitly at the beginning based on the historical data in order to have more realistic plans

At the end, PMB is established with the all work scope including reworking and marking quality investment related tasks as preventive and detective ones.

During project execution process, EVM requires recording all execution data of planned work including actual start and finish dates, efforts and resource utilization for the performed work. QEVM adds two new activities to project execution process:

- Recording the data of internal and external failures including their occurrence and fix dates, the origin of the failure, efforts of fixing
- Gathering the actual investment costs as well as actual costs

In project controlling process, EVM assesses performed work based on the completion rates and calculates EV. Consequently, EV analysis is performed with this EV data, planned value data from PMB of the planning process and actual cost data from execution process. QEVM focuses on the FCs here and recalculates past EVs based on the FCs gathered in execution. QEVM observes EV changes according to FCs and analyzes their trend. It assumes this change will occur for the calculated current EV and predicts its value based on the potential FCs. QEVM adds five new quality performance related activities to controlling process and updates all schedule and cost related ones incorporating quality metrics into EV:

- Calculating quality performance index (QPI) and quality cost factor (qcf), based on the FCs
- Recalculating past EV_s retroactively using qcf
- Analyzing quality performance based on EV deviations in time
- Estimating current EV (EV_{est}) based on quality performance trends
- Analyzing and forecasting cost/schedule performance
- Reporting performance problems and taking action
- Benchmarking quality CC
- Benchmarking FCs

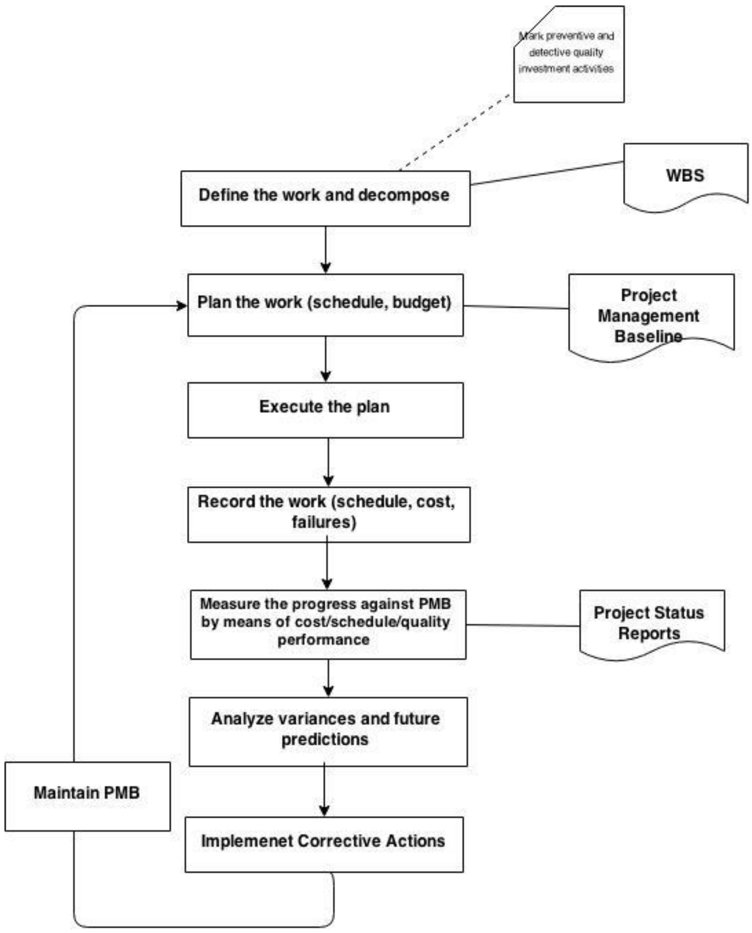

Figure 6 Overview of QEVM

The recommended metrics for scope, schedule and cost are also the basis for QEVM with additional quality cost metrics in controlling process. QEVM brings quality performance aspect and also improves the effectiveness of schedule and cost measures by means of quality cost and provides more accurate progress measurement and future estimates. The key practices and activities of QEVM and existing traditional EVM are given in the following Table 4:

Table 4 Key Practices of EVM and QEVM

Item	QEVM Changes	PM Process
Establish a performance baseline		
Decompose project work into manageable tasks		Planning
Identify conformance related (preventive and detective) tasks	New	Planning
Assign management responsibility		Planning
Plan the activities (Estimate the effort and resource)		Planning
Plan the rework intentionally during activity planning	New	Planning
Develop a time based budget		Planning
Collect planned conformance cost and benchmark	New	Planning
Establish a PMB		Planning
Measure and analyze performance against baseline		
Record resource usage during project execution, AC		Execution
Record failure costs occurred in execution and keep traceability with the origins, FC	New	Execution
Measure the physical work progress, % complete		Execution
Calculate EV according to EV techniques		Execution
Calculate quality performance index and quality cost factor, based on the failure costs	New	Controlling
Recalculate past EVs according qcf	New	Controlling
Analyze quality performance based on EV deviations in time	New	Controlling
Estimate current EV (EV_{est}) based on quality performance trends	New	Controlling
Analyze quality performance trends	New	Controlling
Analyze and forecast cost/schedule performance	Updated	Controlling
Report performance problems and take action	Updated	Controlling

3.3. New Measures of QEVM

Having introduced QEVM, this section goes through the details of the method. The three key data points of traditional EVM, which are PV, EV and AC, are also the key data elements for QEVM since this new model is an extension to traditional one by improving it with quality aspect. In addition to these three key data points, QEVM defines the following key elements and measures:

- Failure Cost
- Actual Cost with Failure Cost (ACF)
- Estimated EV (EV_{est})
- Conformance Cost

Detailed description of each element are presented in next sections.

3.3.1. Failure Cost (FC)

The main target of collecting FCs in QEVM is to make reworking costs visible and include them in measuring the progress of the project. The distinguishing feature of this model from the traditional EVM is to take the reworking costs into consideration and to calibrate EV based on this cost and its trend in time.

FCs are the total effort spent for correcting errors after activities are completed. It covers all reworking efforts. It is an important marker of quality maturity of both process and project. As soon as it is estimated regularly combined with EVM metrics, it offers to track the trend of reworking. FCs and reworking costs are the same and used interchangeably through the document, however, we use the term failure costs in QEVM templates and case studies.

FC concept is borrowed from the CoQ and entirely corresponds to "Cost of Non-Conformance" category which is the sum of IFCs and EFCs. Just as CoQ, QEVM has two types of failure cost: internal and external costs:

- IFCs cover the reworking costs before delivery to the customer
- EFCs represent the reworking costs after delivery to the customer

Here QEVM offers flexibility in terms of defining delivery to the customer. It does not have to be the customer delivery of the project; any milestone would be defined and their trends would be tracked separately. For example, for a development team, releasing a test version would be specified as a milestone and all the efforts for bug-fixing of the errors coming from test team would be external failure cost.

Additionally, QEVM offers defining different levels in internal and external costs if needed. For example, in the previous example if the project has different levels of testing and the project manager wants to track the failures from different levels for the purpose of estimating their future trend, different EFCs would be defined and differentiated by distinct symbols, like EFC1 and EFC2. Depending on FCs, ACF, qcf and all the other metrics should be separately calculated for those levels.

FC is the most significant base measure of QEVM and the success of this new method heavily depends on carefully and correctly gathered FC data. FC data is gathered in parallel to collecting main EVM measures. During project execution wherever AC data is gathered for EVM calculation, FCs are gathered as well at that point if any failure happens. It covers all correcting activities in the project. The list of these activities is given in Table 5.

Fundamentally, QEVM follows this principle: If there is a failure cost regarding a task completed in an earlier phase, it means that EV is changing somehow and should be recalculated retroactively considering this cost as described in the next section. As an example, even though a task, T1, was completed two months ago with 100h, if there are four errors regarding this task and afterwards 40h was spent additionally for these errors, it means that task was not entirely completed at that time and the EV previously calculated was not so true.

Failure cost obviously shows that a project has quality issues. It would be calculated for the tasks separately to check the tasks and spot the ones suffering from quality issues. It might be the signal of significant quality problems in the project and lead to further investigations.

Table 5 List of Failure Cost Activities

Failure Cost Items
Internal Failure Cost Items
(prior to delivery to the customer/or milestone)
Any development rework; errors, changes, improvements
Failure analysis
Re-designing
Fixing errors
Improving and changing features
Updating documents due to the changes
Re-integration
Re-testing
…
External Failure Cost Items
(after the delivery to the customer/or milestone)
All IFC items mentioned above
Activities regarding issues from customer
Technical support
Re-integration
Re-production
Upgrade
Updating documents due to the changes
…

QEVM suggests three alternative ways for FC calculation depending on the QEVM application approach described in Section 3.1:

- The task-based needs the FCs for every task
- The phase-based needs the failure cost for every phase that corresponds to QEVM application time period
- The project-based approach just needs all the FCs. For project-based approach, it is possible to link FC with the phases of software development life cycle or the high-level components of the project.

FC can increase through the time as soon as a new failure occurs, so we called the initial effort to complete a task as AC and all the additional effort is FC.

We could summarize the steps regarding FC collection in QEVM as the following:

1. Select QEVM application approach: either task or phase or project based
2. During analysis, design, development, testing and maintenance phases and afterwards, keep all the errors and changes in issue tracking tool and relate it with the origin tasks/phases. It is also possible to calculate FC in a cumulative way according to past iterations if there is no relation kept with the origin tasks/phases. So, it is assumed that, in the project-based application, all the FCs in total do belong to the past.

3. Save all the related effort of correcting activities from re-analysis to documentation update in the issue tracking tool.
4. During QEVM application, gather FC for the new phase just in parallel to AC and keep the relations of the issues with the tasks/phases.
5. Collect FC for the tasks/phases/project accordingly.

3.3.2. Actual Cost with Failure (ACF)

ACF is the total effort of a task, phase or project at a specific time, which is the sum of the initial cost for completion (AC) and reworking cost needed afterwards for fixing defects or improving it (FC) till that specific time. After collecting failure cost, it is straightforward to estimate it with the following formula:

$$ACF = AC + FC \qquad (8)$$

ACF could increase through the time as soon as reworking occurs, so it is possible to display its status as the following:

Table 6 ACF Changes in Time

Task/Phase	**1**	**2**	**3**	**...**	**n**
T1	ACF_1	ACF_2	ACF_3	...	ACF_n
...					
Phase1	ACF_{11}	ACF_{12}	ACF_{13}	...	ACF_{1n}
...					
Phase2		ACF_{22}	ACF_{23}	...	ACF_{2n}
...					

The ACF of a phase or project is the sum of all the tasks and their reworking efforts in that phase or the project.

For example, we completed a task, T1, in phase1 with 100h, we fixed four errors regarding this task in phase3 and phase4 and afterwards spent 40h additionally (25h in phase3, 15h in phase4) for these errors, here is the ACF for T1:

Table 7 Sample ACF for a Task

ACF	**Phase1**	**Phase2**	**Phase3**	**Phase4**	**...**
T1	100	100	125	140	...

3.3.3. Estimated EV (EV_{est})

The aim of calculating EV_{est} is to adjust current EV by utilizing reworking trends of the project. As explained in detail thorough the thesis, the reworking in software projects is inevitable but the traditional project management techniques are not embracing these characteristics. QEVM accepts the fact that EV might be changing in time by the effect of quality factors and reworking. EV_{est} aims to represent the actual EV based on the assumption that quality cost trends in the past will occur in the future. QEVM uses two significant concepts during EV_{est} calculation: Retroactive EV, EV_r, and qcf, which are explained in the following sections in detail.

EV_{est} is the projected value of EV by means of EV_rs. In other words, it corresponds to the future retroactive EV of the current calculated EV. The idea is to track EV and EV_r differences for earlier phases and then apply this pattern to the current EV to estimate EV_r of current EV, which is EV_{est}. This approach is the same as the way that traditional EVM calculates schedule and cost projections for the future. EVM calculates cost and schedule indices and variances and utilizes them in the future predictions. In this way, we apply this approach to estimate current EV.

EV_{est} is only meaningful for the current application phase. It is the specialized form of EV_r at the present time. EV_rs uses the occurred reworking related with the specific task or phase but EV_{est} is a prediction using the past reworking trends that are not happened yet for the new ones. qcf is the depiction of this reworking trend. EV_{est} does not make sense for the tasks. After EV calculated for the tasks and total EV aggregated, EV_{est} is calculated using the factor, qcf.

QEVM calculates qcf only for past phases and based on QEVM application approach, qcf value could be selected differently as the following:

- If the task-based approach is followed, qcf and EV_r values for the tasks are calculated separately. Afterwards, total EV_r is calculated summing up EV_rs of all the tasks. qcf is calculated by dividing total EV_r to total EV.
- If the phase-based approach is followed, qcf is calculated based for the phase using the failure cost of the phases. Regression models could be used to find the appropriate qcf and qcf of the phase with the similar characteristics could be chosen.
- If the project-based is followed, only one qcf is available and used for EV_{est} calculation.

QEVM favors EV_{est} and assumes that it gives a better, clearer and more accurate status of the project than EV does and so uses EV_{est} for all future estimates.

In brief, the steps for EV_{est} are described in the following:

1. Measure current EV in the traditional way as in EVM at a specific time t.
2. Based on the FC occurred in the past, calculate qcf and then recalculate earlier EV_{t-1} values for the earlier phases/projects
3. Select appropriate qcf as described before.
4. Multiply qcf with the measured current EV.

$$EV_{est} = EV^* (1 - qcf) \quad (9)$$

3.3.3.1. **Quality Cost Factor (qcf)**

qcf is the reworking factor of a specific task, phase or project. qcf represents the ratio of the tasks that cannot be completed right at the first time and earned the claimed value considering the work and the ratio of rework for that particular task or phase. It is calculated by means of FC.

Depending on the collection of FC, qcf could be defined and calculated per task, per phase or per project. qcf may change through the time as soon as the new failure occurs and FC and ACF changes. Thus it is possible to have different qcf values just as ACF values in Table 6. qcf would be calculated for the different phases of software development life cycle or any other component that are linked with the errors during issue tracking.

qcf is used to calculate the previous EV in the past, which we call retroactive EV, EV_r and defined in the following section. qcf is also utilized to calculate adjusted value of current EV, which we call estimated EV, EV_{est} since it is the projection of EV_r to present time. It is assumed that the efficiency will be similar for the current EV and it will be reworked with the same ratio. Therefore, based on this factor and current EV; EV_{est} , which is assumed to be actual EV, would be calculated.

The difference between qcf and QPI is that QPI includes all the tasks even ongoing ones during its calculation. QPI is dealing with the whole reworking, not bounded to any specific task and its reworking ratios. However, qcf checks the trends of the reworking for specific task or phase and is limited to their own reworking. Whenever qcf is calculated, it is not the qcf of that period, instead it belongs to previous periods since reworking occurs afterwards. For example, in the middle of an iteration, QPI counts the costs of the tasks of that iteration but qcf only use the values of the previous one since the trend for that iteration is not visible yet. Additionally, QPI is for the project or for phase but qcf could be calculated for the task, phase or project depending on the QEVM application. QPI is much easier to calculate than qcf. There is no need to know any relation between FCs and tasks for QPI. At the end of the project, QPI is equal to (1- qcf) of the project.

It is possible to have organizational qcf database for different project characteristics of an organization. Especially, more detailed qcf values of each phase or phases of software development cycles (analysis, design, coding, testing and so on.) for the different project types would be very beneficial during planning. It will give the insight to project managers during project management in different granularities.

qcf is different than other indices of EVM, that's why we call it factor instead of index. It is calculated backward for the previous phase and used to estimate adjusted current EV (EV_{est}), so provides estimating current progress accurately, not just the future as the indices do.

The steps for qcf are described as the following:

1. Collect AC of the completed tasks, all the FCs for the completed tasks and keep the relation of the completed tasks and their reworking effort
2. Calculate qcf with the following formula (at a specific time, t):

   $qcf_t = FC_{t-1}/ACF_{t-1}$ (10)

   **ACF represents the total cost including all the reworking through the time. Here AC, FC and AFC efforts are exactly for the specific same task or phase.*

If qcf is bigger than 0, it means that FCs exist and the tasks have been reworked. It indicates how much work has been reworked and not achieved successfully at first time for this specific task or phase or complete project.

3.3.3.2. **Retroactive EV (EV_r)**

Retroactive EV, EV_r, is a significant concept of QEVM originated from the finding that EV is changing over time. It represents the recalculated value of EV retroactively at a specific time by means of incorporating FCs that have occurred after the initial EV was calculated. It could be calculated for tasks, phases or project depending on the QEVM application approach.

Retroactive EV is calculated using qcf since this factor, qcf, symbolizes the reworking trend in the past. As soon as FC, ACF and qcf change, retroactive EV changes too.

The main assumption here is that the actual value of EV, EV_r, is found after some time passed, errors fixed and changes happened. During project execution, we could re-estimate the previous EV_rs to observe and analyze how much it changed comparing the value in the past and the retroactive one. Obviously there may exist a gap between EV and EV_r of a project in the course of time.

Its value at a time, n, is represented as EV_{mn}, which shows retroactive EV at a specific time, m, in the past depending on the total FCs occurred afterwards between m and n. Therefore, retroactive EV may constitute a matrix as seen in Table 8. For every phase, it is possible to have a changing EV_m value later. The latest value at a specific time is considered at the end.

Table 8 Retroactive EV Matrix

	EV	1	2	3	4	…
Phase1	EV_1	EV_{11}	EV_{12}	EV_{13}	EV_{14}	…
Phase2	EV_2		EV_{22}	EV_{23}	EV_{24}	…
Phase3	EV_3			EV_{33}	EV_{34}	…
Phase4	EV_4				EV_{44}	…
..	…	…	…	…	…	…

The main issue in this concept is changing the value of completion percentage. EV is just an expression of PV in terms of completion ratio. However, QEVM does not focus on the details of completion percentage, it just uses it to calculate EV_r properly as EVM uses in EV calculation. Instead, QEVM concentrates on the relation of EV_rs for the same time points in the past. The similar trend is assumed to be determinant for the current EV.

The calculation of EV_r is an important step of QEVM. As a prerequisite item, FC and ACF need to be gathered and qcf should be calculated before EV_r calculation. The steps for EV_r are described in the following:

1. Collect AC, FC, ACF and then calculate qcf.
   - If QEVM is applied task-based, FC is needed to be related with the tasks. If QEVM is applied as phase-based, FC is needed to be related with the phases.
   - Otherwise, if QEVM is applied project-based, total FC is enough and EV_r of the project is calculated. qcf is calculated for the previous period.

2. Calculate EV_r with the following formula:

$EV_r = EV * (1\text{-}qcf)$ (11)

3. If the task-based approach followed, we have qcf and EV_r values for the tasks. For the application period of QEV, total EV_r is found by summing all the tasks. In that case, total qcf is calculated using this total EV_r by the following formula and this final one is used calculating EV_{est}.

   $qcf = 1 - (\sum EV_r / \sum EV)$ (12)

4. If the phase-based approach followed, we have qcf value for the phases obtained from FCs. EV_r is found by using qcf.

   $qcf = 1 - (\sum EV_r / \sum EV)$ (13)

The following example is given to explain the concepts numerically:

Two tasks Task1, Task2 in phase1 and two tasks Task3, Task4 in phase2. Here are the numbers at the end of phase1:

Table 9 Example Tasks Data – phase 1

	PV	AC	Completion %	EV
Task1	100h	100h	100%	100
Task2	50h	50h	100%	50

At the end of phase1, we have EV=150. During phase2, new tasks are completed and also new errors occurred related Task1 and so 30h more spent for Task1. Here are the numbers at the end of phase2:

Table 10 Example Tasks Data – phase 2

	PV	AC	Completion %	EV
Task3	100h	100h	100%	100
Task4	100h	100h	100%	100

From QEVM perspective, at the end of phase2, FC happened for Task1 and AFC, qcf and EV_r could be calculated for tasks as in Table 11. We selected the phase-based QEVM approach and apply it phase-by-phase.

Table 11 Example QEVM Table - 1

	PV	AC	EV	FC	ACF	qcf	EV_r
Phase1	150h	150h	150	30	180	0.16	126
Phase2	200h	200h	200				

At the end of phase2:

For phase1: $qcf = FC/ACF = 0.16$ and $EV_r = EV* (1\text{-}qcf) = 126$

We can estimate EV_{est} for the phase 2: $EV_{est} = EV* (1\text{-}qcf) = 200* (1\text{-}0.16) = 168$

3.3.4. **Conformance Cost (CC)**

QEVM includes CC concept to give insight to the project manager at the beginning of the project about their quality investment and benchmark this cost with the other projects. CCs represents the total cost that is planned to achieve good quality in a project by means of preventing failures or detecting them. It is the concept of CoQ.

Two main components of CCs are PCs and APCs as CoQ proposes:

- PCs are the total costs of the activities aiming to prevent failures.
- APCs are the total costs of the activities targeting to detect failures.

PC and APC have pretty tight relation. Obviously the less preventive activities bring the more detective activities to achieve the good quality or vice versa.

QEVM suggests to mark the related activities of prevention and appraisal during project planning or establishing PMB and to sum their efforts accordingly.

By means of CC in QEVM, the project manager becomes explicitly aware of the cost reserved to have a good quality. QEVM suggests having an organizational database keeping those values based on the project types and characteristics for benchmarking. Also, industrial statistics would be helpful.

Additionally, project manager finds an opportunity to see the relation of the quality investment cost versus FCs. In that respect, it is possible to extend CCs via PV and ACs and observe and compare their relations through the time during execution and controlling of the project. Since those concepts are the main interest of CoQ and this study focus on the reworking issues, we limit the CC related contribution with the CC and Conformance Cost Index (CCI). Possibly, further study would investigate these relations.

We present the summary of the steps regarding CC in the following:

1. Mark preventive and detective tasks during project planning or PMB generation.
2. Calculate PCs, APCs and so CCs and benchmark CC/PV with the organizational /industry values by means of CCI.

3.4. **QEVM Performance Analysis and Forecasting**

This section explains how the measures of QEVM defined in the previous section are used to analyze the current status of a project and predict its future.

As mentioned in Section 3.1, QEVM measures project performance phase-by-phase and/or cumulatively to a particular date. Here it is described and illustrated in terms of cumulative data. Phase-by-phase application is adjusted accordingly considering the data of specific phases.

The performance analysis and forecasting of QEVM involves three parts: Cost Analysis & Forecasting, Schedule Analysis & Forecasting and Quality Analysis & Forecasting. The first two analyses are already coming from traditional EVM but QEVM improves them with the updated measures in their calculations. QEVM proposes to use EV_{est} in the estimations of cost and schedule and includes FCs in cost estimations.

The metrics of performance analysis and forecasting consists of variances, indices and forecasts. QEVM includes the new and updated following metrics:

- Variances: SV, CV, VAC
- Indices: SPI, CPI, QPI, CCI
- Forecasts: EAC, ETC, Estimated Total Failure Cost (EFC), To Complete Failure Cost (TCFC)

EVM metrics of performance analysis and forecasting represent the status of the projects providing answers to the key management questions. QEVM additionally responses new project management questions regarding the quality and give more clear and more accurate answers to the existing ones that EVM already provides the answers (see Table 12). The following sections describe those metrics in detail.

3.4.1. Quality Performance Analysis and Forecasting

Quality performance measures are the main contribution of this thesis. QEVM aims to give more accurate information about the project by utilizing quality cost trends.

FC is the base measure showing the reworking that is the effort of the tasks not doing right first time and so total cost that quality problems caused. It is already defined in the previous section in detail. Its existence indicates the quality issues and reworking. It directly corresponds to the effort spent for correcting activities occurred due to not doing right first time. It is not possible to have a negative FC.

QPI is the index representing the quality efficiency of the project by means of FCs. At a specific time, QPI shows the ratio of the tasks done right for the first time considering all the work and rework till that time.

QPI is calculated for the project or the phases. There are two main usages of QPI. Firstly, it is used to give a quick impression about the efficiency of project quality during project execution and control. The second usage is for benchmarking and consequently it is the input of planning rework activity in the planning process. Organizations would have their historical QPI database according to project characteristics and use this input in the planning of similar projects or even phases.

QPI uses FC and cumulative AC during its calculation. It does not keep the relations, instead it gives the general view about the status. The errors do not have to belong to the completed tasks or any other relation needed with the tasks and origins, instead the timeline of their correcting activities are considered here. Its formula is given in the following:

Table 12 QEVM and Basic Project Management Questions

Project Management Questions	**QEVM Performance Measures**		
How are we doing time-wise?	**Schedule Analysis & Forecasting**		
Are we ahead or behind the schedule?	SV	Schedule Variance	From EVM
Are we ahead or behind the schedule considering reworking?	SV_{est}	Estimated Schedule Variance	New
How efficiently are we using time?	SPI	Schedule Performance Index	From EVM
How efficiently are we using time considering reworking?	SPI_{est}	Estimated Schedule Performance Index	New
How are we doing cost-wise?	**Cost Analysis & Forecasting**		
Are we under budget or over budget?	CV	Cost Variance	From EVM
Are we under budget or over budget considering reworking?	CV_{est}	Estimated Cost Variance	New
How efficiently are we using our resources?	CPI	Cost Performance Index	From EVM
How efficiently are we using our resources considering reworking?	CPI_{est}	Estimated Cost Performance Index	New
What is the entire project likely to cost?	EAC	Estimate at Completion	From EVM
What is the entire project likely to cost considering reworking?	EAC_{est}	Estimate at Completion	New
Will we be under budget or over budget?	VAC	Variance At Completion	From EVM
Will we be under budget or over budget considering reworking?	VAC_{est}	Variance At Completion	New
What is the remaining work likely to cost considering reworking?	ETC	Estimate to Complete	From EVM
What is the remaining work likely to cost?	ETC_{est}	Estimate to Complete	New
How are we doing quality-wise?	**Quality Performance Analysis & Forecasting**		
What is the total cost of quality problems (reworking)? Do we have quality problems?	FC	Failure Costs	New
How efficient is our quality performance? How efficient are we doing right first time?	QPI	Quality Performance Index	New
How much is planned for achieving good quality?	CCI	Conformance Cost Index	New
What is the reworking of the entire project likely to cost?	EFC	Estimated Total Failure Cost	New
What is the reworking of remaining work likely to cost?	TCFC	To Complete Failure Cost	New

QPI is better and shows good performance of quality when it is close to one. If the failure cost is zero, it equals to one and means no reworking at all.

There is no QPI for a specific task. QPI of the phases shows the reworking percentage for that specific phase.

CCI represents the ratio of CCs to total planned costs. Its main target is to measure the total effort planned for achieving good quality in the project by means of preventing failures or detecting them comparing to total planned effort. CCI is used for benchmarking and gives the idea to the project manager about their investment to quality comparing to other projects or earlier projects.

$CCI = CC/PV$ (15)

EFC represents the estimated FCs of the total project by means of FC projections through time. At the end of the project, it is equal to total quality cost variance. It gives an idea to the project manager about the effort to be spent for quality problems if goes with this way. The project manager would take some precautions to avoid these costs accordingly. If it is too high, it might trigger to increase quality CCs especially PCs.

$EFC = qcf * (BAC - EV_{est}) + FC_t$ (16)

TCFC represents the FCs to be spent for the remaining work of the project. It indicates how much more effort will be spent based on the past FC projections.

$TCFC = qcf * (BAC - EV_{est})$ (17)

3.4.2. Cost Analysis and Forecasts

In addition to new quality performance measures, QEVM updates cost analysis and forecasts significantly with quality costs and adds new measures to those analyses. QEVM improves EV by estimating EV_{est} which is supposed to be the actual current EV. Additionally, QEVM improves AC with ACF and it makes the significant change in cost variances and indices.

CV_{est} (Estimated Cost Variance) shows whether a project under or over budget. It is an improved version of CV of traditional EVM. Although CV is very useful to reflect cost overruns during new feature implementation, if there is reworking for the earlier tasks, it might skip this type of cost overrun. CV_{est} is defined in QEVM to consider FCs and calculate cost variances based on EV_{est} and ACF rather than EV and AC in order to decide whether the project is under or over budget.

$CV_{est} = EV_{est} - ACF$ (18)
$CV_{est} > 0$, under budget
$CV_{est} < 0$, over budget

CPI_{est} (Estimated Cost Performance Index) is an index showing the efficiency of the utilization of the resources allocated to the project. It is an improved version of CPI of traditional EVM. CPI_{est} is defined to consider FCs during calculating the efficiency of cost performance.

$CPI_{est} = EV_{est}/ACF$ (19)

$CPI_{est} > 1$, efficiency in utilizing the resources allocated to the project is good

$CPI_{est} < 1$, efficiency in utilizing the resources allocated to the project is poor

ETC_{est} (Estimate to Complete) is the estimated effort required to finish all the remaining work, calculated when the initial assumptions are not valid anymore and revised estimates are needed. It is an improved version of ETC of traditional EVM. ETC_{est} is defined to consider FCs during remaining work calculation.

$ETC_{est} = (BAC - EV_{est}) / CPI_{est}$ (20)

EAC_{est} (Estimate at Completion) is the estimated final effort of the project that is required to finish all the work. It is calculated using the performance indices. It is an improved version of EAC of traditional EVM including FCs.

$EAC_{est} = BAC / CPI_{est}$

$= ACF + ETC_{est}$

$= ACF + ((BAC - EV_{est}) / CPI_{est})$ (21)

VAC_{est} (Variance at Completion) represents the variance on the total budget at the end of the project. It is the difference between the cost that is initially planned and the cost that is now estimated. It is an improved version of VAC of traditional EVM including FCs.

$VAC_{est} = BAC - EAC_{est}$ (22)

3.4.3. Schedule Analysis and Forecasts

This section describes the updated and new schedule analysis and forecasts of QEVM by estimating EV_{est}. As in the previous section, here we do not give the measures already described in Section 2.2, but just describe the new ones.

SV_{est} (Estimated Schedule Variance) shows whether the actual costs of the work accomplished exceed the initially planned costs. It is important to identify the significant variances to take action.

$SV_{est} = EV_{est} - PV$ (23)

$SV_{est} > 0$, ahead of schedule

$SV_{est} < 0$, behind of schedule

SV_{est} is the improved version of traditional SV. Even SV displays objectively how much the project is ahead or behind schedule for new features, SV_{est} improves it including reworking costs.

SPI_{est} (Estimated Schedule Performance Index) is an index showing the efficiency of the time utilized on the project.

$SPI_{est} = EV_{est}/AC$ (24)

$SPI_{est} > 1$, efficiency in utilizing the time allocated to the project is good

$SPI_{est} < 1$, efficiency in utilizing the time allocated to the project is poor

3.5. **Putting All Together**

QEVM elements and performance measures are already defined in previous sections. The complete idea and metrics are summarized graphically in the chart below and demonstrated in Table 13. Additionally, Table 14 presents all the QEVM elements in the table. The results of QEVM application is represented by means of

- Tables
- S-curves
- Bar charts

We used all these representations during case studies in the next chapter.

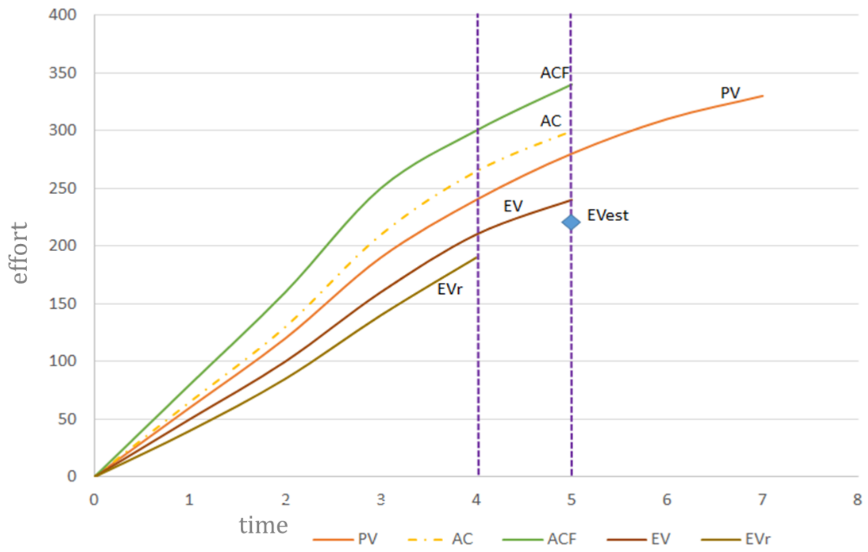

Figure 7 QEVM Graphical Representation

To give a brief summary, we gathered all the concepts and measures in the following tables.

Table 13 QEVM All Together

	QEVM
DATA	PV AC EV FC ACF
	EV_r EV_{est} CC PC APC
VARIANCES	FC SV CV VAC EFC
INDICES	SPI CPI QPI CCI ACI PCI
FORECAST	EAC ETC EFC TCFC
FORMULAS	$ACF = AC + FC$ $QPI = (1 - FC_t / ACF_t)$ $qcf_t = FC_{t-1}/ACF_{t-1}$ $CC = PC + APC$ $EV_r = EV * (1-qcf)$ $EV_{est} = EV * (1- qcf)$ $CCI = CC/PV$ $EFC = qcf *(BAC - EV_{est}) + FC_t$ $TCFC = qcf *(BAC - EV_{est})$ $CV_{est} = EV_{est} - ACF$ $CPI_{est} = EV_{est}/ACF$ $ETC_{est} = (BAC - EV_{est}) / CPI_{est}$ $EAC_{est} = BAC / CPI_{est}$ $EAC_{est} = ACF + ETC_{est}$ $EAC_{est} = ACF + ((BAC - EV_{est}) / CPI_{est})$ $VAC_{est} = BAC - EAC_{est}$ $SV_{est} = EV_{est} - PV$ $SPI_{est} = EV_{est}/AC$

Table 14 QEVM Summary Table

Metric	Name	Definition	Equation	Notes
FC	Failure Costs	total effort spent for correcting activities of the errors and changes occurred after completed, covers all reworking efforts for a task, a phase or a project, shows the effort not doing right first time		Quality problems exists and reworking happens, and equals to how much effort spent for quality problems
ACF	Actual Cost with Failure	total effort spent to complete and improve for a task, phase or project at a specific time, the sum of the initial cost for completion (AC) and the reworking cost needed afterwards for fixing defects or improving it (FC) till that specific time	AC + FC	ACF could change in time as soon as failures occurs
qcf	Quality Cost Factor	index showing the reworking ratio of a specific task, phase or project that cannot be completed right and earned the claimed value at the first time	FC/ACF $1 - (\sum EV_r / \sum EV)$	Reworking occurs with this ratio, qcf
EV_r	Retroactive Earned Value	recalculated value of EV retroactively at a specific time in the past by means of failure cost occurred after the initial EV calculated, could be calculated for tasks, phases or project	EV * (1-qcf)	EV retroactively recalculated for a specific time in the past
EV_{est}	Estimated Earned Value	projected value of EV based on the assumption that quality cost trends in the past will occur in the future by means of EV_rs and qcf, specialized EV_r at the present time	EV* (1-qcf)	current EV with the calibrations
CC	Conformance Cost	represents the total cost that is planned to achieve good quality in a project by means of preventing failures or detecting them, corresponds to the cost of conformance concept	PC + APC	To prevent failures in the future and detect them, we plan this effort
QPI	Quality Performance Index	index representing total reworking status of a project or phase. At a specific time, QPI represents the ratio of the tasks done right for the first time considering all the work and rework till that time	$(1 - FC_t / ACF_t)$	gives a quick idea about the quality performance status

CCI	Conformance Cost Index	index representing quality conformance ratio of a project or phase. At a specific time, CCI represents the ratio of total effort spent for achieving good quality in the project by means of preventing failures or detecting them.	CC/PV	gives a quick idea about the quality investment and provides benchmarking opportunity
ETFC	Estimated Total Failure Cost	total effort to be spent for failures through the project, calculated by means of FC projections	qcf *(BAC - EV_{est}) +FC_t	shows estimated quality cost variance at the end of the project
TCFC	To Complete Failure Cost	remaining effort to be spent for failures in the project, calculated by means of FC projections	qcf *(BAC - EV_{est})	shows estimated quality cost variance for the remaining part of the project
SV_{est}	Schedule Variance	improved SV of traditional EVM by means of failure costs	EV_{est} – PV	
SPI_{est}	Schedule Performance Index	improved SPI of traditional EVM by means of failure costs	EV_{est}/PV	
CV_{est}	Cost Variance	improved CV of traditional EVM by means of failure costs	EV_{est} - ACF	
CPI_{est}	Cost Performance Index	improved CPI of traditional EVM by means of failure costs	EV_{est}/ACF	
VAC_{est}	Variance at Completion	improved VAC of traditional EVM by means of failure costs	BAC - EAC_{est}	
ETC_{est}	Estimate To Complete	improved ETC of traditional EVM by means of failure costs	(BAC – EV_{est}) / CPI_{est}	
EAC_{est}	Estimate at Completion	improved EAC of traditional EVM by means of failure costs	ACF + ((BAC – EV_{est}) / CPI_{est}) ACF + ETC_{est} BAC / CPI_{est}	

CHAPTER 4

CASE STUDIES

This chapter presents the research methodology and the case studies performed for exploratory and validity purposes. Section 4.1 explains the research methodology that we follow through this thesis. Then, Section 4.2 describes the exploratory case study that we conducted to investigate the shortages of traditional EVM on software projects. Finally, Section 4.3 presents the multiple case study performed to validate the applicability of QEVM.

4.1. Research Methodology

There are three common approaches to conducting research: quantitative research, qualitative research and mixed methods which include both quantitative and qualitative [77] [78].

Creswell [79] defines that quantitative research is the process of collecting, analyzing, interpreting, and writing the results of a study. It is a systematic investigation of quantitative properties, their observable occurrence and relationship. Quantitative research aims to develop and employ mathematical models, theories and hypothesis relating to the phenomena by collecting numerical data and using statistics. Quantitative research methods were originally developed in natural sciences to investigate natural phenomena and lately are widely used in social sciences such as psychology, sociology, political science and economics and so on. Surveys, formal methods and laboratory experiments are examples of quantitative methods [80].

Qualitative research is a subjective research type, which is depending on the analysis of controlled observations of the researcher. In qualitative research, data is obtained from a relatively small group of subjects comparing to quantitative research. Statistical techniques are not used to analyze data. Typically, in qualitative research, narrative data is gathered via interviews and observations. Qualitative research methods were originally applied in the social sciences to explore social and cultural phenomena. Its target is to offer an in-depth understanding of people and the social and cultural contexts within which they live. Action research, grounded theory, case study research and ethnography are examples of qualitative methods. In qualitative research, the researcher collects data in the natural settings through the overview of the documents, observing the behavior or interviewing the participants [81].

While the quantitative method offers an objective measure of reality, the qualitative method provides to explore and better apprehend the complexity of a phenomenon. Williams [78] noted that researchers typically select the quantitative approach in order to respond to the research questions requiring numerical data, the qualitative approach for research questions requiring textual data, and the mixed methods approach for research questions requiring both numerical and textual data.

The research approach followed in this study is qualitative research because we need to explore and better understand a phenomenon as well as collect data in its natural setting over documents.

Case study is selected as the qualitative method in this study. The case study research is the most common qualitative method used in information systems (IS) [82]. Yin [83] defines that "a case study is an empirical inquiry that investigates a contemporary phenomenon within its real-life context especially when the boundaries between phenomenon and context are not clearly evident". Yin argues that a case study is the preferred research method when contextual factors are believed to be highly relevant to the subject of study. A case study has a distinct advantage as a research strategy in situations when "a 'how' or 'why' question is being asked about a contemporary set of events, over which the investigator has little or no control" [83].

The case studies vary according to purpose and scope. In terms of purpose, Yin distinguishes between exploratory, descriptive, and explanatory case studies [83]. In terms of scope, there are two types; single-case design strategy and multiple-case design strategy. Despite their differences, case studies share a number of common features, including research design components, a case study protocol, principles of data collection, and quality criteria. Single cases are a common design for doing case studies while multiple-case design strategy involves more than one case. The evidence from the multiple-case study is often considered more compelling and the overall study is regarded as being more robust [83].

In IS, the application of qualitative research methods are increasing since recently there is a shift in IS research from technological to organizational issues [80]. The case study research method is particularly well-suited to IS research [84].

The literature review, qualitative research based on case studies with both exploratory and explanatory research approach are the foundations of this research. This study contains two main phases; development of the conceptual model and application of the conceptual model as seen in the following figure, Figure 8.

At the initial phase of the study, we performed extensive literature review on the researches on EVM, using EVM in software projects, shortages of EVM, extensions of EVM, software quality measurement and CoQ. The literature review helps us to understand the gaps and shortages of EVM for software projects.

Afterwards, we utilized exploratory research approach with qualitative methods to comprehend the difficulties and shortages of EVM and conducted an exploratory case study with two cases on two different software projects in a company, Company A. Exploratory research is used when problems are in a preliminary stage and helps to define a problem [85]. It is often carried out qualitatively. It is conducted when the researcher does not know about the details of the problem and needs additional information. These two cases provide us an opportunity to inspect the difficulties and shortages of EVM for software projects as well as the possible improvement opportunities and made the problem more visible.

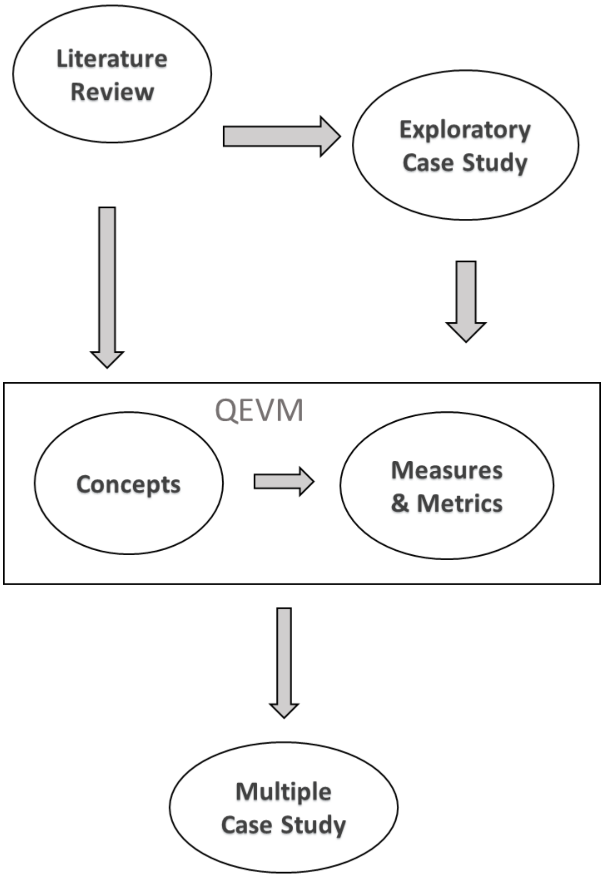

Figure 8 Research Method of the thesis

The results of extensive literature review, the findings of exploratory case study and direct observation of the author, who is working for software projects for more than 13 years, are consolidated and hence the conceptual model, QEVM, is established.

The second phase of the study includes the application of the conceptual model by means of qualitative experiments conducted. The explanatory case study is used here as a qualitative research method. We applied multiple case study research to observe the applicability of new QEVM model that we established in this study, to answer the research questions, and subsequently to validate the conceptual model based on the results of these case studies. When the research goal is the description, theory building or theory testing, multiple case designs are more appropriate [84]. Additionally, multiple-case studies enable cross-checking between different cases and provide the extension of theory. Since the objective is to validate a new conceptual EVM model in this part, we conducted six case studies in six different companies in a multiple-case design.

There are three ways of data collection in qualitative analysis: in-depth, open-ended interviews, direct observation and written documents [86]. In interviews, people express their experience, opinions, feelings and knowledge directly. Interviews may be conducted unstructured, semi-structured or structured. However, in direct observation, observations depend more on the evaluator. It requires the researcher to observe the people's activities. Written documents are based on the inspection of the existing written documents, reports, and other paper material. It should be the first consideration since existing information is cost-effective and not corrupted by the participants [87].

In the exploratory case study, we used written documents and semi-structured interview methods to collect data. The interview with the project manager was necessary to gather data and clarify conflicts in written documents, which are the project plans, progress reports, and error reports.

During the development of the conceptual model, we used direct observation as an instrumentation. We evaluated literature review and the findings of exploratory case studies and combine them with author's own 12 years professional experience and observation on software projects.

In the case studies conducted to validate the conceptual model, we utilized written documents and interviews again. Written documents include project documents, plans, reports, error reports, change request reports and so on. We also carried out the semi-structured interview with the project manager and team leader. Before the interviews, all parties were informed about the research details to establish initial trust, to avoid unethical issues and to let them get prepared. We prepared interview forms beforehand and we were not allowed to use a voice recorder by the interviewees.

In the data analysis phase of the exploratory case studies, project data gathered from written documents and interviews are analyzed together based on comparisons and calculations. The analysis did not use any statistical methods. In the exploratory case study, additionally direct observation of the researcher were considered during data analysis and the most common qualitative data analysis technique, observer impression, used during conceptual model development.

A set of improvement areas about the effective use of EVM in software projects is produced in the conceptual model by interpreting the collected data, direct observation and analyzing the most important issues in the field.

During the application of the model, again the interpretation of the researcher with the project document's inspection, calculation of project data based on the new conceptual model, actual data from the written documents are used. Comparison between the calculated data from the conceptual and the actual data from the inspected documents are the indicator of the success of the conceptual model. Additionally, MMRE is utilized for future estimates and comparison of EAC of EVM and EAC of QEVM.

The conceptual model is revisited during the research, as new information emerge or our understanding of the current information evolve.

4.2. Exploratory Multiple Case Study on Implementation of EVM

The first case study is a multiple-case study which was conducted to explore the applicability of traditional EVM on software projects, to figure out its difficulties and weaknesses for software projects, to assess the usefulness of EVM in software projects and to observe possible improvement areas. This case study has been performed as a first step just after literature review.

We asked the following research questions during exploratory case study in accordance with these objectives given above:

RQ1: What are the benefits of using EVM in software projects?

RQ2: What are the difficulties and weaknesses of EVM application for software projects?

We present the design, conduct and results of this case study in the following sections.

4.2.1. Multiple Case Study Design

Case Selection Strategy: Our strategy is to apply the initial case study on two software projects applying different development methodologies in order to practice EVM on software projects and to look into answers of the defined research questions. We want to investigate and evaluate the applicability of the EVM by our own hands on experience and explore the difficulties and shortages.

Data Collection Strategy: We plan to use written documents and semi-structured interview methods to collect data in this study. The interview with the project manager and team leader is necessary to get the brief project info, to gather data and to clarify conflicts in written documents, which are the project plans, progress reports and error reports.

Validation Strategy: We aim to analyze the results of EVM calculations comparing with the actual project data from the documents. We do not plan to use any statistical methods in analysis. Calculations and comparisons are the main methods answering the research questions.

4.2.2. Case Study Conduct

We performed this case study with two case projects applying different development methodologies with different characteristics in a software company, Company A, which is Turkish subsidiary of a multi-national company. It has ISO 9001:2000 certification and is employing nearly hundred software engineers. The company is mainly developing various in-house software projects with different sizes and its annual revenue is around 7 million Euros.

The first project is a software product development project ongoing for years with iterative approach while the second one is an application development project with waterfall methodology and has relatively short development time. The reason behind selecting projects with different characteristics is to observe various challenges, to reflect different drawbacks in applying EVM and so to provide an extensive look to the possible issues. Even they are from the same company since the company is a part of a global organization and the projects are from irrelevant business domains under different departments, their project management approaches are quite unalike. This situation lets us detect the various points in EVM implementation as if they are from the different companies.

For all the organizations and projects that we contact and conduct the case studies, we keep their names private for confidentiality issues upon their request.

We prepared case study statement prior to data gathering to inform the project managers about the study and then sent them via e-mail. Afterwards, we scheduled an interview to get the project info and the written documents.

For both projects, we performed initial interviews with the project manager taking approximately 1 hour. We inform them about the study briefly, get the relevant project information and discuss our needs. They provided us the written documents via e-mail later. The details of each case and conduct are given in their own sections.

4.2.2.1. 1st Case Study: Project I, Company A

Project Context

The first case project, Project I, is the development of a sale management tool that enables selling exceedingly complex hardware systems with many peripherals. It is a product development project, started in October 2003 and is still on-going. It follows iterative and incremental development approach.

Every major release of the product is managed like a new project. The case study has conducted on the latest major release on the date it was conducted, from May 2011 to April 2012. In a major release, there are monthly-based minor releases, lasting approximately four or five weeks. In addition to the high-level coarse-grained major release planning, there is a fine-grained iteration planning for minor releases.

There are two main teams in the project, definition team and development team. The definition team is located in another region. Hence, the case study here only covers the project management of development team and shows their performance. The development team consisted of 7 people, including 1 project manager, 1 part-time team leader attending development activities part-time as well, 1 senior software engineer, 2 software engineers, 1 test engineer and part-time configuration manager. The total development effort is given in the following table, Table 15.

Table 15 Case I Total Effort

Item	Effort (person-hour)
Total Planned Effort	6341
Actual Effort spent for the new features	6665
Actual Effort spent for the errors and changes	1704

The project is developed using organization-specific, high-level modeling tool. The requirements specified by definition team are kept in a requirements database, developed by the organization itself. The changes, improvements and errors are stored in an organizational ticketing system. The development team has a close collaboration with definition team.

The project management manages the project under two separate lines: the first line is for new features and the other is for errors and changes. Even the budgets of those two lines are separate. They assume that the first line is the project management part and they apply the techniques of project management just on this line. On the contrary, the other line, called sustaining, has buffers and is not managed in a special way. These buffers including all the errors and changes are paid under sustaining budget. All the project info given above was gathered during an initial interview with the project manager.

EVM Application

The case study conducted in May 2012 after a major release completed. The project related data, basically project plan, timesheets, error reports in excel sheets were provided by the project manager in two parts. The first part including the planning and realization data of the new features, including the short definitions, the responsible, planned and actual efforts, planned and actual realization dates. The second part was a list of errors, changes and improvements including the short definitions, the responsible, detection and resolution dates and versions and efforts spent to resolution.

The necessary data were mainly collected from these excel sheets and subsequently the conflicts in the documents were discussed and clarified with the project manager and team leader in another session to gather data as accurately as possible.

During the application of EVM, the effort (person-hour) is used as cost unit since it is in-house software development project and the main budget consideration is the effort of the developers. All projects are tracked in an effort-driven way in the organization. We measured the project progress on every minor release using EVM, so the month is used as the time unit.

First, we implemented EVM on the project according to their project plan including new tasks in every minor release and their completion (see Figure 9) since the project manager tracks the project in this way. The EVM application data is given in Table 43.

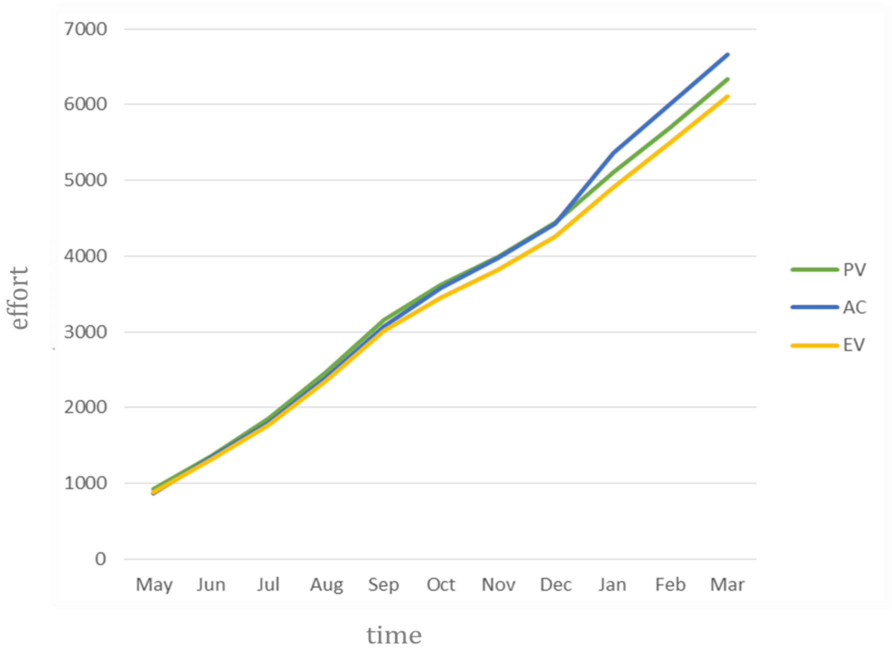

Figure 9 Case I EVM results for the new features

As a result of first EVM implementation, the picture for new tasks was looking good. The tasks were mostly implemented on time by the developers assigned to them. The deadlines almost met without delay. The assigned tasks were in general implemented in the planned time interval. All minor releases were published on time including the planned features nearly done.

Nevertheless, after we checked the error reports, we discovered that these tasks had been significantly reworked again and again with the errors coming from system test and the other parties. In their process, during a minor release, system testers are already testing the completed features and change the task status to "completed" later. After release, system testers perform the release test, and additionally another test group performs an additional level of testing in a more integrated way. The errors found before release were counted in the iteration and the fix effort by the developer was already counted in the feature implementation effort. Only the efforts found after release is considered as rework effort in this study.

Then, we categorized the errors and added their efforts for the releases to see how much effort actually spent for the tasks in the releases including later bug-fixing efforts.

Based on this data, we applied EVM in a different way and we add the effort of bug fixing to the effort of the related period (see Table 44). It is not exact implementation of EVM and does not directly reflect the actual effort at this period since the errors have been performed later than the releases completed but we just want to have an idea on reworking effects of the releases. Therefore, in the second figure, we put the phase names instead of months and the horizontal line represents the phases. The results are displayed in Figure 10.

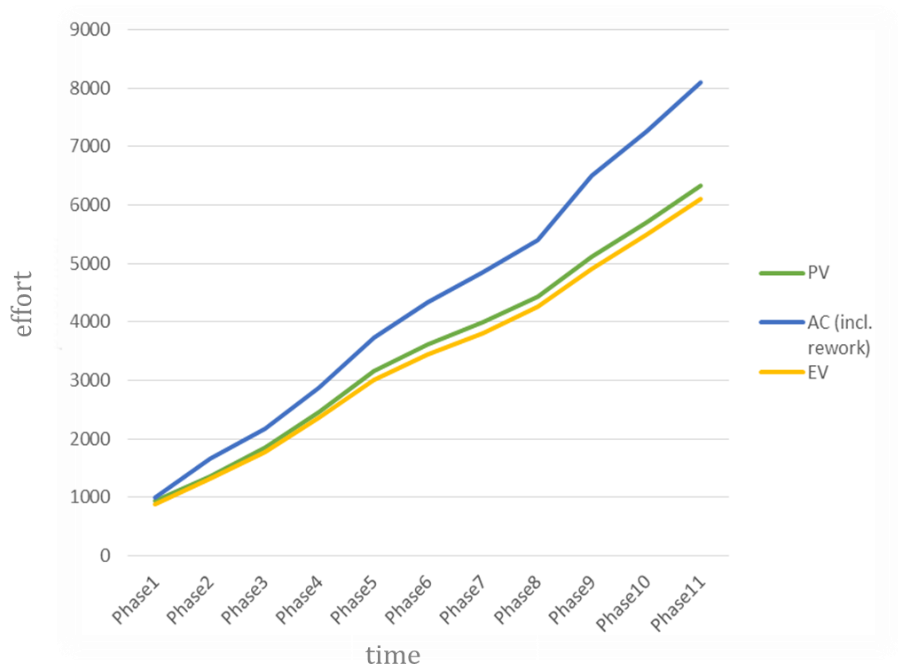

Figure 10 Case I EVM results including reworking

As seen in the figure, there is a major effort gap between planned value and actual cost in this case. We can interpret these two figures as the following by choosing a month randomly; according to first EVM results before reworking, at the end of December, the 8th minor release of this major release was available and the project seemed almost on track with a little cost overrun and schedule delay. However, in a few months, some changes occurred and errors found and it is understood that the earned value calculated at that time was not precisely correct; cost and schedule variances were significant. Even though for a specific task in this release, project manager thought it was completed at that time, afterwards it is observed that it was not completed and additional hours, corresponding to %20 of the initial effort, spent for this task later.

4.2.2.2. 2nd Case Study: Project II, Company A

Project Context

The case project is a middleware application which enables data collection from the various products in the product portfolio of the company in order to serve data for the different management applications. These management applications are web-based high-level applications used by technicians in the service department to configure these products.

The project development started in January 2012 and completed at the end of July 2012. Before January, the product management already prepared product backlog and system architects shaped the architecture. The project followed waterfall development methodology. The case study only covers the project management of development team till the end of system test phase since the organization was responsible for these phases. The development team staff consisted of 8 people, including 1 project manager, 1 team leader, 1 senior software engineer, 3 software engineers, 1 test engineer and 1 part-time configuration manager. Table 16 shows total effort for the project during development and testing phases.

Table 16 Case II Total Effort

Item	Effort (person-day)
Total Planned Effort	408
Actual Effort spent for the new features	470
Actual Effort spent for the errors and changes	136

During the development of the project, Service Oriented Architecture (SOA) and Java - based, organizational web application development framework were utilized. The requirements were kept in requirements database as use cases and wireframes. The errors and improvements were stored in an industrial ticketing system, Clear Quest.

EVM Application

The case study conducted in December, 2012. The project plans with Microsoft Project and the excel sheets displaying progress were the main source of EVM application. Also, the error reports exported from the ClearQuest was utilized to observe the later efforts. The conflicts in the documents were discussed with the project manager.

During the application of EVM, the effort (person-day) is used as cost unit since the project manager already tracked the project in this way.

The monthly project progress reports were used basically during EVM implementation. Therefore, the month is used as the time unit.

At the beginning of the project, 4-months development phase planned to complete the features of the application and during 2 months-system test phase, two additional developers were allocated to support bug-fixing and reworking. In this project, rework for errors and changes was already estimated and included in the project plan by the project manager.

First, EVM was implemented monthly on the project excluding system test phase; just concentrated on development phase. The development phase completion delayed one month and completed at the end of May. Figure 11 displays the picture of the progress in terms of time and the EV key elements and the details of the EVM application is given in Table 45.

In the first three months, there is a considerable gap between the planned and earned values. The project manager explained that the main reason for this gap is the infrastructural problems, having a high learning curve. Additionally, they encountered many problems regarding framework infrastructure. Due to these unexpected infrastructural problems, the project was very behind the schedule at that time and the estimations have become unrealistic. Also, there were some uncertainties in some work packages and their estimations were not so correct. They were in general underestimated.

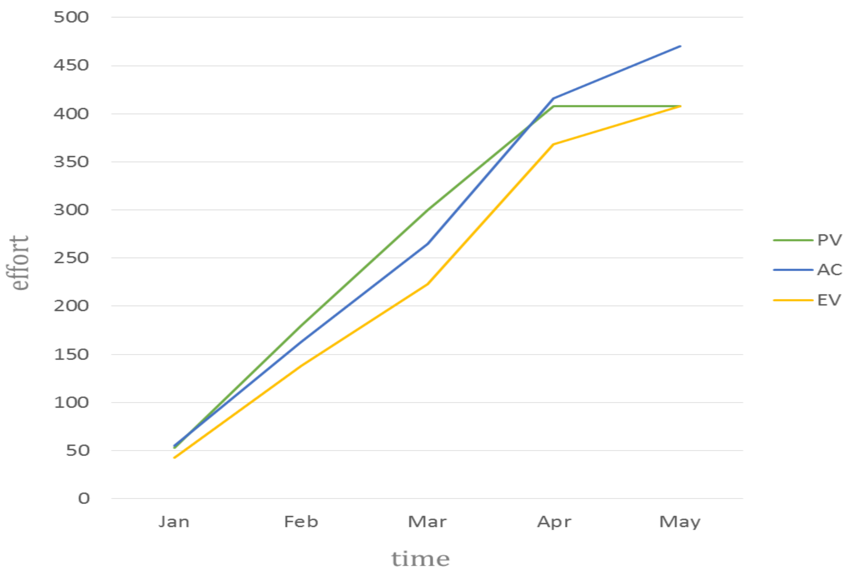

Figure 11 Case II EVM results for development phase

It is apparently seen from the Table 45 and Figure 11 that the earned value velocity is quite high during last two months. The project manager explained that it is the result of high overtime during April and May to catch the milestone.

At the end of the belated development phase, the project seems that it is over-budget but on-schedule based on the EV metrics.

After development phase, additionally EVM has been applied to the development team activities during 2 months system test phase (see Table 46 and Figure 12).

In system test phase, there was a task scheduled for bug-fixing and simply two developers were assigned to this task initially. The project manager stated that during system test phase, they needed to assign more developers than estimated and five developers worked on bug-fixing. The error ratio was much higher than estimated.

During system test phase, EVM has nothing to reflect correctly. The features already implemented had been reworked during 2 months with more developers than estimated and no metric gives us any clue about the real progress or the status objectively other than low cost performance index. The low quality of the development phase misled project manager at the beginning of the system test phase and prevented to see the possible quality issues.

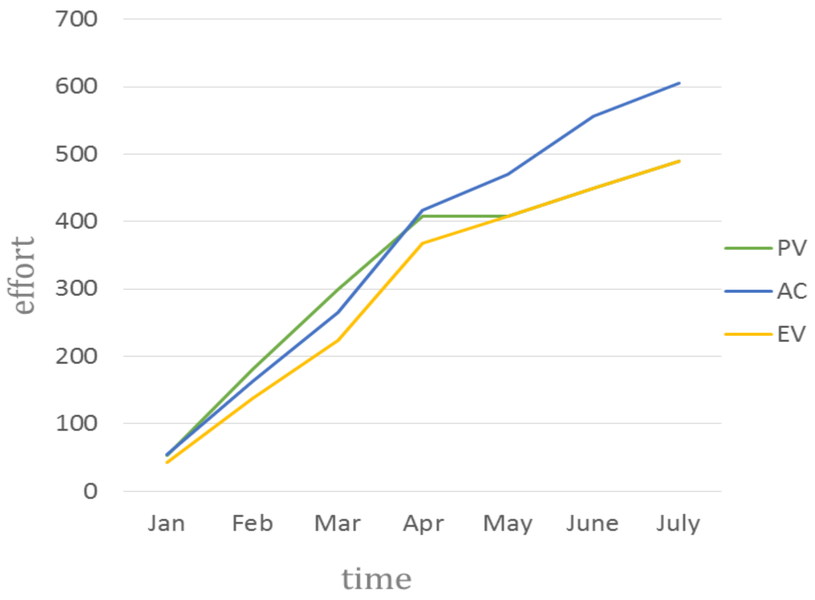

Figure 12 Case II EVM results for development and system test phase

4.2.3. **Results**

This section summarizes the findings of EVM applications corresponding to research questions defined for the exploratory case study. We initially start with answering the first question explaining benefits of the method. Afterwards, we present the difficulties encountered and the deficiencies identified during EVM applications to these case projects.

RQ1: What are the benefits of using EVM in software projects?

EVM is useful for software projects like the other projects since it reflects the project status at a specific time point. For both case studies, in particular for new features, it demonstrates how the project is progressing. In 2nd Case Study, during the development phase, it obviously reflects the latency in the project beginning from early months and so gives an opportunity to project manager to estimate this one-month milestone delay. However, the reworking of the implemented features are not represented here and EVM does not give any clue about these failures and reworking.

RQ2: What are the difficulties and weaknesses during EVM application to software projects?

The first difficulty was related with data gathering for EVM. It is very important to have a mature project management process to apply EVM correctly. Since both case projects were already completed, the only sources were the available documents and the support of project managers. Therefore, the backward data gathering for the actual efforts was difficult and not so obvious. The efforts for implementing new features, changes to the existing features and errors were kept in different sources and not so up-to-date. Sometimes, the documents were inconsistent with each other. Hence, it was not easy to obtain actual implementation efforts precisely.

If the project management process is more mature and EVM is applied during project execution, the case studies would have generated better results and may let us apprehend other difficulties of applying EVM.

The second deficiency, which is the most significant finding of this study, is the effect of rework effort on EVM in the software projects. Reworking is a major issue and indispensable part of software projects. Based on the essential characteristics of the software projects and the quality related issues, reworking is in general accepted as a natural consequence.

In the 1st case study, in every release, there are some changes and errors for the features implemented in the previous releases. Those errors are coming from testers, external testers, sales groups, definition team, and product management and so on. Therefore, the significant effort is spent on errors on the previous features but it is counted in the current release. In that case, the gap between earned value and actual cost seems huge (see Figure 12). However, if we exclude rework effort of the existing features, the figure is quite reasonable (Figure 11) to observe the progress of the new features. In this project, reworking is not calculated or planned systematically, only some reserves are put for that effort since it is already accepted. However, this approach makes EVM results unacceptable, and the correct value earned at some specific time cannot be calculated.

In other words, at a specific time, even though it seems 100% of a task completed, after some time, additional effort is needed for the reworking of this task. Therefore previously calculated earned value is not correct or something wrong here. EVM does not represent the figure and the status of the project correctly.

In the second case study, rework is already accepted at the beginning of the project, based on the past experiences and visibly put into the project plan, as bug-fixing. When the development phase is completed with 1-month delay, EVM shows us that the project is over budget but on time. This is also another problem of the EVM for late projects which are already observed before by Lipke and ES method is proposed to resolve this drawback of EVM [28]. During system test phase, much more effort is needed for the bug-fixing comparing to the original plan but based on the EVM metrics, we could not observe any sign before that happens. There are major quality problems during development phase causing significant reworking. It considerably affects project's progress but there is no systematic way to plan and measure it in the software projects.

We observe that applying EVM as it is could give software project managers wrong direction in both case studies. Even at a specific time the project is supposed to be on track, additional effort would be needed for completing features. Since it is known as rework costs may approach 50 % of the total software project cost, this fact should not be ignored and carefully considered [9] [8].

The third difficulty encountered during EVM application is the ambiguity of effort and value relation of a software task. In both case studies, EVM cost metric is considered as effort, which is person-hour and person-day since the organization tracks all their projects based on effort. It is also very common in software organizations to track the projects, based on effort as the main cost driver is the effort. However, it is very subjective to quantify the value created for specific effort. It is more or less known in the projects of other industries, for example, it is very clear in construction projects how much value can be created in an hour for a worker. In software projects, the effort and size prediction models are not enough to state precisely the effort and value relation. There is no standard size measure used as a base to progress measurement in the industry yet. There are many measures i.e. LOC, FP, UCP but none of them is accepted and used widely. Furthermore, the effort estimates are still based on the expert opinions and usually are not directly related with the size that progress can be measured. Both case projects used expert opinions during effort estimation. Since the first one is already ongoing for years, the estimations are not problematic but Case II suffers a lot from underestimation, some tasks double the effort comparing to initial estimations. As a result, the 4-months development phase is extended one month more. Additionally for software projects, it is unclear how to state the value of an ongoing task objectively and precisely and give an exact statement about that considering the fact that there is no objective and concrete size/effort measure. In Case II, it is observed that especially the huge tasks spanning more than a month may get lower complete percentage than the previous month. It means after one month work, the completeness percentage is decreased. It basically points to the incorrect and subjective measurement of physical work progress. Additionally, it also brings us the problem of managing huge work packages. It is quite subjective and hard to decide work progress ratio in particular for such a big work packages and may lead to incorrect earned value calculations.

4.2.4. **Discussion**

In the exploratory case study, we identified three significant problems while applying EVM on these two software projects.

First of all, the mature project management process is fundamental during EVM application; otherwise EVM would not be applied properly and produces incorrect values. Secondly, it is observed that the most considerable reason of the gap between planned and actual is reworking, which is an inevitable part of software projects is considering the fact that the reworking cost may approach 50% of total cost. EVM is not dealing with any reworking specific issues. It must be taken into consideration in the software projects in order to see the project progress objectively. The last but not least issue is the uncertainty of effort and value relation of a software task. There is no standard size measure of software projects. It is very subjective and people dependent. So, incorrect effort estimation is a fundamental problem. Also, measuring/estimating the physical progress of the tasks properly and objectively is another challenge of the software project managers.

This study shows us that EVM is a powerful method to reflect project progress in terms of scope, time and cost. It focuses on the quantity of the tasks rather than the quality. The quality is supposed to be implicitly included in the scope dimension and once the task is completed there is no further work regarding this task again. Even though it might be correct for the traditional project management, for software projects the status is a bit different. Implementing EVM as it is without considering the characteristics and challenges of software projects might mislead software project managers at a specific time. Though project manager thinks the project is on track, this quality issues and subsequent reworking efforts make project over budget and delayed in an unexpected way using EVM. Software projects have substantial rework and quality factors are affecting project progress significantly. To demonstrate the project progress correctly and to provide project managers more accurate estimation, EVM needs to consider quality factors in the method and so could tell more about the progress and performance of the project accurately.

4.2.5. **Validity Threats**

We aim to observe the difficulties and shortages of EVM application on software projects in this study.

The internal validity threat was the occurrence of the problems within the project management process rather than the deficiencies within the application of EVM. To avoid this limitation, our main selection criterion for the cases is to pick the projects having as mature as project management process. The accurate project management data is crucial to reflect the possible problems of EVM instead of the problems of the process itself. Additionally, we set up meetings with the project managers for clarification of the inconsistencies in the documents.

The threat regarding external validity was to generalize the results of the case study to the software projects. In order to minimize this threat, we designed multiple case study instead of the single case study. Additionally, we particularly select the project applying different development methodology, including iterative and waterfall.

4.3. **Multiple Case Study on Implementation of QEVM**

The second case study is a multiple-case study which involves six different cases. In this multiple case study, our objective was to explore the applicability of the new EVM model that we introduced in Chapter 3, QEVM, and to validate the applicability of the model. Furthermore, this case study explores whether QEVM can be applied properly to different software projects by integrating quality aspect by overcoming the problems that we identified during this research and whether QEVM could be used for tracking the progress of software projects more accurately and predicts project future better compared to traditional EVM. This case study has been performed as the last step after model developed and refined.

We asked the following research questions during this case study in accordance with the objectives given above:

RQ1: Is QEVM helping project managers to see the project current status more clearly and to estimate project future more accurately?

RQ2: What are the benefits of QEVM for software projects comparing to EVM?

The QEVM application templates used in these case studies are given in APPENDIX B. We present the design, conduct and results of this case study in the following sections.

4.3.1. **Multiple Case Study Design**

Case Selection Strategy: We plan to conduct case studies on six different projects from different organizations. During case study design, we pay attention to select cases utilizing different software development methodologies, from different business domains, with different sizes and from different organizations in order to observe QEVM applicability, benefits and possible problems on different circumstances.

Software development methodology especially affects reworking and their occurrence timeline, so we aim to observe if QEVM applies better depending on the methodology.

We target to select the projects using EVM and also the organizations applying EVM on their projects.

We also attend the maturity levels of the organizations to avoid the internal process problems of the organizations and projects and focus QEVM issues clearly. Mature project management process is a must for EVM success and so for QEVM.

Data Collection Strategy: As in the exploratory case study, we aim to use written documents and semi-structured interview methods to collect data as well in this study. The interviews with the project managers are planned to conduct to get the brief project info, to gather data and to clarify conflicts in written documents, which are mainly the project plans, progress reports, error reports and other project management documents.

Validation Strategy: Our strategy for validation is to compare the results of QEVM calculations with the actual project data from the documents and EVM data that we calculated in parallel. For the analysis of current estimates, we do not plan to use any statistical methods. The QEVM calculations and comparisons with the EVM calculations and actual data will be the indicators of the QEVM success.

For future estimates, we target to utilize a statistical method, MMRE (Mean Magnitude of Relative Error), which is the most commonly used measure of the average estimation accuracy [88]. We plan to estimate the EAC with both EVM and QEVM and compare their results using MMRE. Even it has been criticized, MMRE has become de facto standard for evaluating software effort estimation and prediction models [89] [90].

MMRE is defined as [91]:

MMRE = $\frac{1}{n}\sum_{i=1}^{i=n}\left(\frac{|Xi-X|}{Xi}\right)$

where x_i is the actual value and x is the estimated value of a variable of interest.

MRE of the each estimate is defined as:

MRE = $\frac{|actual\ effort - estimated\ effort|}{actual\ effort}$

The usability of the model is increasing if MMRE is getting close to zero. Low MMRE indicates that the model is successful and the models with MMRE value lower than 0.25 is accepted as an applicable estimation method [88].

For validation purpose, we will calculate MRE for EAC of every iteration both by EVM and QEVM for each case. Therefore, we will analyze how the methods predict the future.

4.3.2. **Case Study Conduct**

We applied QEVM on six different projects in six different organizations as planned.

According to these objectives, we initially selected twenty organizations and contacted the quality manager or the head of Project Management Office (PMO) or the project manager. Fifteen organizations responded positively and ask about the details of the study while two of them never responded and three organizations state that they cannot support such a study due to their workload or policies.

We prepared case study statement (see APPENDIX B) during initial contact with the companies to inform the project managers about our research and this study and then distributed this statement to the organizations that are willing to collaborate. For all the projects, we planned to schedule an interview to get the project info and the written documents.

Afterwards, two of organizations do not want to share the effort details and stopped collaboration. One of the organizations requested us to sign NDA for data protection. Finally, we agreed with six organizations and they provide the details of projects for the case studies.

These initial interviews are supposed to be performed with the project manager and estimated around an hour. The checklist given in APPENDIX B has been used during the interview. Our purpose was to inform them about the study briefly, to get the relevant project information and to discuss our needs before retrieving the written documents.

At the beginning, we aim to give a priority to the organizations applying EVM on their projects. However, we observed that it is not very often in the software development organizations in Turkey. Only one company among them is applying EVM on a pilot project and very interested in our model, however, they could not provide us their project data due to the privacy level of the project.

During this study, for each case, we conducted both EVM and QEVM application sequentially to compare their applications and results to observe the benefits and difficulties.

Additionally, for all the case studies, since the projects are already completed or the companies allow the application for completed parts of the project, we could not start QEVM application in the planning process.

Since QEVM uses effort as cost unit, in all case studies we use effort even we apply EVM as well to compare the application results.

All the case studies are conducted by the author of the thesis. She is experienced in EVM and a PMP certified project manager for 7 years. The details of organizations, cases and their conduct are given in following subsections.

4.3.2.1. 1st Case Study: Project III, Company A

Organizational and Project Context

The initial case study is conducted in Company A which we already introduced in Section 4.2.2. The case project is the latest major release of the first exploratory case project, Project I, which is a development of a sales management tool that enables selling exceedingly complex hardware systems with many peripherals.

We contacted the same project manager again by sending our new case study statement. We performed short interview to get the following project info and status plus to explain the new model and the data requested for the study.

In this release, the development team includes 8 people profiled as the following:

- 1 project manager
- 1 team leader
- 4 software engineers
- 1 test engineer
- 1 part-time configuration manager

As we described already in the exploratory case project description in Section 4.2.2.1, the project is a product development project, initially started in October 2003 and is still on-going. It follows iterative and incremental development approach. Every major release of the product is managed like a new project. In 2012, an exploratory case study has been performed on a major release. This case is latest major release in 2014, lasting from May 2013 to January 2014, and two major release after then we performed the exploratory case study. The total development effort is given in Table 17.

The development tools and the project team structure are same with the exploratory case study.

Table 17 Case III Total Effort

Item	Effort (person-hour)
Total Planned Effort	7230
Actual Effort	7600
Rework Effort	979

The project management approach to reworking is exactly same with the previous study in two lines of project management and two separate budgets as the new line and sustaining line. The same team is working for both lines. They have a buffer for the sustaining line based on expert judgment estimates. The project manager traces the source of the errors and changes and prepares reports for the definition team.

Conduct

The case study was conducted in February 2014 after the latest major release completed in January. The project manager delivered us the written project documents in excel sheets in two parts again two weeks after our initial interview.

- The first part including the planning and realization data of the new line, including short definitions, planned and actual efforts, responsible and planned and real timelines of the new features. The project manager is already marked the conformance related tasks upon our request in the first part.
- The second part was a list of errors and improvements including the short definitions, the responsible, detection and resolution dates and versions, efforts spent to resolution and the origin features.

All the necessary data were mainly composed of these sheets. Subsequently, to resolve the conflicts in the document, two semi-structured interviews were conducted with the project manager and team leader separately.

The origin tasks of the errors and changes are kept in this project. They also use this info for the performance evaluations of the developers. Accordingly, we followed task-based QEVM approach.

Both traditional EVM and QEVM have been applied by the author of this thesis in 4 weeks.

We applied EVM and QEVM on every minor release to track the project progress, so the month is used as the time unit during QEVM application. The effort, in person-hour, is used as cost unit.

<u>Results of traditional EVM Application</u>

First, we applied EVM on the project according to the documents provided in the first part, which are the project plan based on the planned features and their realization in the new line with 9 minor releases and the new features in these releases. Table 18 provides the EVM application results and Figure 13 shows the graphical EVM representation for these new features. Moreover, Table 47 presents the performance analysis and forecasting values.

Table 18 Case III EVM Results

	Iteration	PV	AC	EV
May	**1**	904	963	884
June	**2**	1660	1734	1600
July	**3**	2527	2513	2307
Aug	**4**	3194	3332	3089
Sep	**5**	4034	4176	3874
Oct	**6**	4981	5198	4783
Nov	**7**	5915	6253	5676
Dec	**8**	6600	6957	6327
Jan	**9**	7230	7600	6877

This project tracks failures under the sustaining project and EVM application does not include any reworking effort of errors and changes.

Following failures in the scope of another project than development itself makes development project progress perfectly on-time in spite of cost overruns. There are just some minor delays in the schedule as well as minor but increasing cost overrun with the approximate index of 0.90.

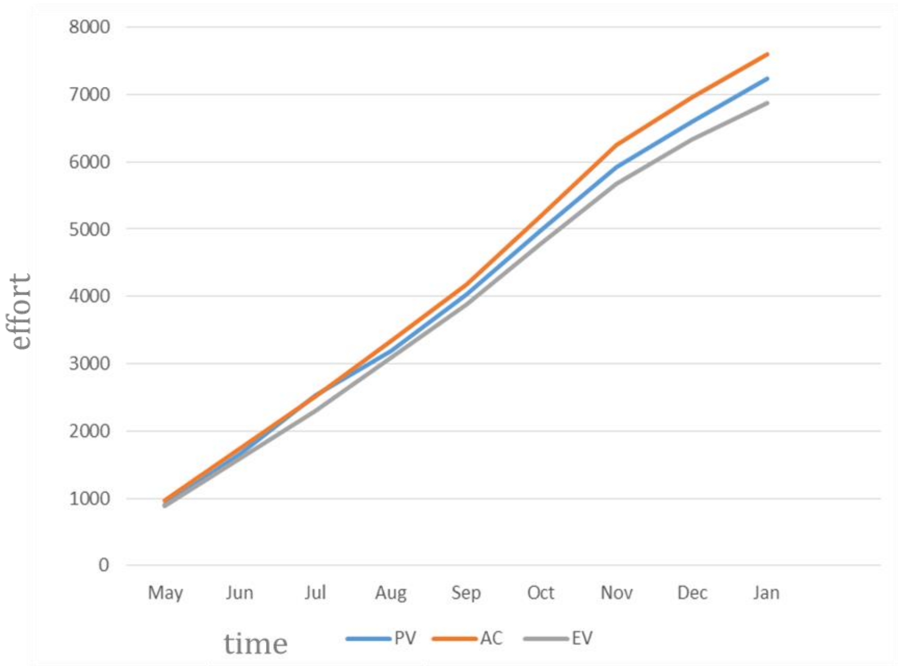

Figure 13 Case III EVM Application Results

Results of QEVM Application

After traditional EVM, we applied QEVM to the project. Initially, we gathered quality costs. FCs are already available in the error list and tracked already associating with the original tasks. We categorized FC according to the tasks and so iterations. The sum of the

FC data based on the iterations as well as the cumulative one is given in Table 48 and Table 49.

First, based on the FCs, we calculated ACF and qcf values for the tasks and calculate retroactive EVs of the tasks. Afterwards, we estimate the retroactive EVs of the iterations through the time and based on that we calculate qcf of these iterations, which shows us the trend of the iterations through the time (see Table 50, Figure 27 and Table 51). Here the calculation of retroactive EV for each task might be a bit complex for the project manager. Then the median of the qcf values of the iterations is counted as qcf. The different statistical models could be applied to find qcf value.

Accordingly, we obtained EV_{est} using qcf of the iterations. We did not calculate the retroactive EVs of the iterations here since we already obtained qcf. The application results are given in Table 19. Besides, Figure 14 shows the graphical representation of QEVM application.

Table 19 Case III QEVM Results

							considering previous iteration, t-1		
	Iteration	**PV**	**AC**	**EV**	**FC**	**ACF**	**qcf_{t-1}**	**EV_{est}**	**QPI**
May	**1**	904	963	884	0	963	0.00	884	1.00
June	**2**	1660	1734	1600	106	1840	0.09	1536	0.94
July	**3**	2527	2513	2307	220	2733	0.09	2243	0.92
Aug	**4**	3194	3332	3089	341	3673	0.09	3019	0.91
Sep	**5**	4034	4176	3874	444	4620	0.08	3811	0.90
Oct	**6**	4981	5198	4783	594	5792	0.09	4701	0.90
Nov	**7**	5915	6253	5676	704	6957	0.08	5605	0.90
Dec	**8**	6600	6957	6327	848	7805	0.09	6268	0.89
Jan	**9**	7230	7600	6877	979	8579	0.09	6828	0.89
			7966	7200	1080	9046	0.10		0.88

Subsequently, we calculate all QEVM performance analysis and forecast metrics and the results are given in Table 52.

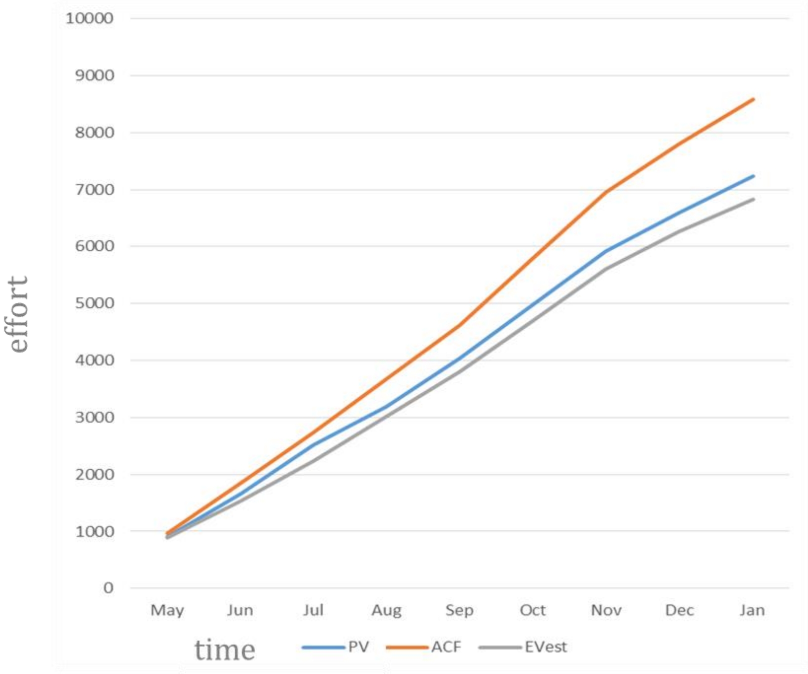

Figure 14 Case III QEVM Graph

The prevention costs and appraisal costs are gathered from project plan going through the tasks with the project manager during the second semi-structured interview. The project does not have much prevention activities other than regular feature reviews with the product manager (see Table 53). Appraisal costs are much higher and driven by a test engineer in the team. Table 20 presents the overall CC plan at the beginning of the project. It is the summary of the quality investment that could be used for comparisons and benchmarking.

Table 20 Case III QEVM Conformance Costs Summary

	CC	**PV**	**CCI**	**PCI**	**ACI**
Case III – Total	1170	7230	0.16	0.01	0.16

Findings

The initial EVM application results show that the project is almost on time till last two months but later minor delay becomes more visible (see Figure 13). The figure shows insignificant cost overrun with the approximate index of 0.90.

The cost overrun starts with 79 person-hours and increases later till 723 person-hours with almost linear CPI. The schedule seems on track with SPI index around 0.96. Based on that, the tasks were mostly implemented on time and the deadlines almost met without any major delay.

This project has another separate line and budget for the issues related with these implemented features. The additional rework efforts including errors and changes are handled under this budget with the same project staff. They designed such a mechanism since traditional project management does not have a standard way of handling reworking. Neither traditional EVM does. Thus, initial EVM implementation does not give any information about the second line.

QEVM implementation combined the efforts coming from reworking and feature implementation. Therefore, it represents all the cost spent for the project more accurately and gives higher cost variances and lower cost effectiveness. The actual costs of EVM and QEVM are presented in Table 54, which shows that QEVM counts approximately 1100 person-hours more than EVM, QEVM makes the failures and their effort visible.

The new quality effectiveness index of QEVM, QPI, is around 0.90, indicating the ratio of the tasks completed right first time in the project (see Figure 28). QPI shows that 88% of the tasks are done right first time at the end of the project. QPI trend is linear during the project.

Table 55 presents the EV and adjusted EV_{est}. Here QEVM gives not so many differences in values since reworking already provides regaining the values. Still through the project it is a better estimation of the gained value comparing to EV. Figure 27 and Table 50 shows how the retroactive EVs change and the project manager analyze the trends through time.

CPI of QEVM is around 0.80 and that is lower than CPI of EVM, which is 0.90. Having considered the failures, QEVM indicates more cost problems. SPI is seen slightly better with QEVM but it still does not solve its essential problems of EVM that Lipke spotted [28]. The comparisons of indices are given in Table 56 and Table 65.

QEVM calculates that the project will be completed around 9000 person-hour, which is almost 1100 person-hours more than the estimate of EVM. The difference of total cost variance and other forecasts are compared in Table 57. QEVM further predicts total FCs almost 1000 person-hours.

The results of MREs that are calculated for EAC by means of both EVM and QEVM are given in Table 58. MRE of EVM is changing between 0.12 and 0.14 while QEVM gives MRE values between 0.004 to 0.04. QEVM predicts EAC considerably better than EVM as seen in the table since the initial iterations,

Furthermore, QEVM shows that the project invests 16% of its budget in ensuring good quality (see Table 53). 1% of this CCs devoted to the activities that prevent quality problems while 15% of the total budget reserved for the appraisal activities detecting the quality problems. At the beginning of the project, the project manager would benchmark these percentages and could evaluate the investment on good quality. In this project, it is much preferred detecting and fixing errors rather than preventing them.

In summary, throughout the execution of the project, the project manager realizes more clearly the cost problems and expects more cost overruns utilizing QEVM. Even though this project does not suffer high reworking efforts, it is still important to make all the costs and the effort of people visible. Besides, the project manager gets the awareness of their quality investment with CCI index and the status of failures and their cost with QPI index. Therefore, he would get some precautions after understanding the root cause of the failures or would increase preventive activities depending on the situation.

4.3.2.2. 2nd Case Study: Project IV, Company B

Organizational and Project Context

Project IV is the maintenance of a web-based procurement tool that provides managing complex tenders, bids and contracts for a large amount of audiences.

We performed the second case study in a software development organization that we called Company B through the thesis. Company B is a software development company employing approximately 60 software engineers. All the employees in the company are working for various e-government projects for a specific government organization. The company has ISO/IEC 20000 IT Service Management and ISO/IEC 27001 Information Security Certifications and was preparing for CMMI-level 3 certification at the time that the case study conducted.

There is one scrum team in the project that is responsible for implementation of the features and testing of them. This team includes 6 staff profiled as the following:

- 1 part-time project manager
- 1 team leader also working as software engineer
- 2 software engineers
- 1 senior test engineer
- 1 part-time quality manager

The maintenance project was started in December 2012 and planned to be completed at the end of December 2013. However, the case study has been conducted on 10 iterations of the project, from February 2013 to December 2013 depending on the data provided us. This project follows iterative development approach with Scrum practices. The total development effort is given in Table 21.

Table 21 Case IV Total Effort

Item	Effort (person-hour)
Total Planned Effort	4162
Actual Effort	4553
Rework Effort	758

The project is developed using .NET Framework. The new features coming from the field are kept in the feature list and the details are stored in Microsoft Team Foundation Server (TFS). The related tasks for those features are created and tracked the project in terms of those tasks. The changes, improvements and errors found by the tester are again stored in TFS. MS Project is utilized for project management and MS Word and MS Excel are used for documentation of requirements and testing.

Conduct

The case study conducted in May 2014. Initially, we sent the case study statement to the project manager via e-mail and then conducted an initial meeting with the project manager and quality manager to discuss the details of the study. Afterwards, we scheduled an additional meeting as a semi-structured interview with the quality manager and obtain the brief overview of the project as well as explain the needs of a case study in detail. Finally, the quality manager provided the project data, basically sprint plans, resource utilization reports, error reports and exported all to the excel sheets hiding the domain related contents due to the confidentiality.

The document that they provide includes the bugs fixed and new features from the feature list as distributed into the releases. We combine those releases as one month periods and apply EVM accordingly.

All EVM and QEVM applications are conducted by the author of this thesis accordingly in the following 3 weeks.

As the time unit, we used one month phases in which the releases are combined. The effort, in person-hour, is used as cost unit.

In this project, the relation between the errors/changes and the initial tasks are not kept, so it is not possible to track the origin. In such a case, we assume that the errors and changes belong to the past. Consequently, we followed project-based QEVM approach described in Section 3.2 as QEVM suggests.

Results of traditional EVM Application

We initially applied EVM on the project considering the high-level project plan including planned features and their realizations. Table 22 shows the EVM application data and Figure 15 is the graphical representation of application for the project.

The overall status of the project from this initial EVM implementation demonstrates that the tasks are completed on time with a cost overrun. Table 59 shows the performance of the schedule and cost in addition to variances and indices.

Table 22 Case IV EVM Results

	Iteration	**PV**	**AC**	**EV**
Feb	1	101	96	86
Mar	2	750	831	752
Apr	3	1619	1712	1560
May	4	2207	2392	2185
Jun	5	2558	2814	2537
Jul	6	2970	3179	2878
Aug	7	3203	3480	3160
Sep	8	3460	3716	3378
Nov	9	3703	4036	3667
Dec	10	4162	4553	4151

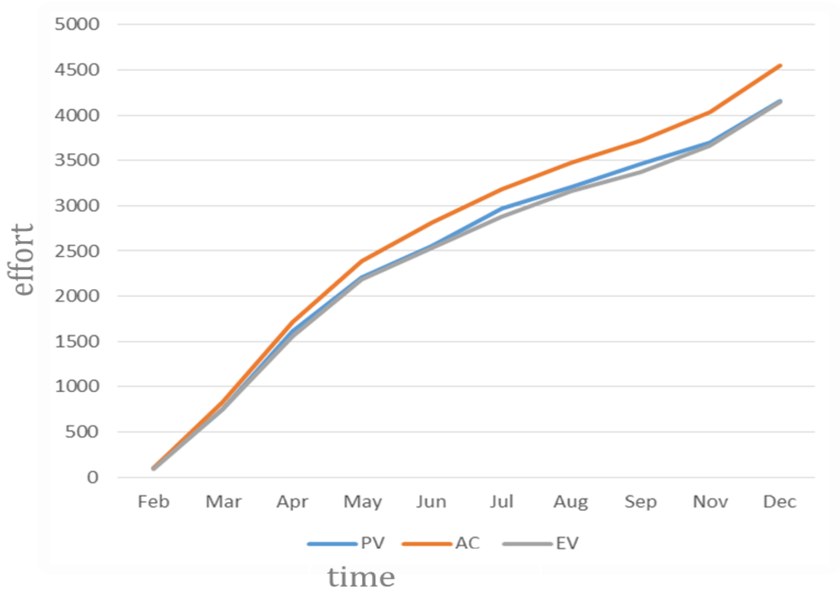

Figure 15 Case IV EVM Graph

<u>Results of QEVM Application</u>

We applied QEVM just after EVM implementation on the project. The application is started with the collecting FCs. The project documents provided includes the release versions of tasks completed and errors found and completed. The original task of an error is not tracked instead only the origin version of the error is kept. We only know in which version this error has detected. Therefore, we collect all the items for errors and changes as failure cost.

Then we grouped them as internal failures and external failures according to the releases and phases (see Table 60). Some of the errors occurred were fixed in the one month phase, in one of the releases. We assume that those errors are internal failures and the others are external failures. In other words, the failure is internal if it is related with the current phase and it is external if the failure belongs to the release of the earlier phase. IFCs cover the efforts of development team after once completed but before release. EFCs are the ones after the release. The reason behind we differentiate IFC and EFC here is to estimate future based on the EFC since IFC are already spent when we control the status. However, we prefer to keep it separate than initial AC because it is already detached and it happens after some time. The Figure 29 shows the FCs graphically, here AC already includes IFC.

ACF and ACF for the previous phase, ACF(t-1), are calculated. ACF (t-1) represents the actual cost of the previous phase, t-1, including the FCs spent during the recent phase. ACF for the phase changes through the time whenever reworking occurs.

Next, the qcf calculation is accomplished considering EFC. Accordingly, EV_r and EV_{est} are calculated. The application results are given in Table 23. Figure 16 shows the graphical representation of QEVM application. Additionally, Table 61 presents the performance analysis according to QEVM.

Table 23 Case IV QEVM Results

							considering previous iteration, t-1				
	Iteration	**PV**	**AC**	**EV**	**EFC**	**ACF**	**ACF_{t-1}**	**qcf_{t-1}**	**EV_r**	**EV_{est}**	**QPI**
Feb	1	101	96	86	0	96	96	0	0	86	0.9
Mar	2	750	831	752	25	856	121	0.26	64	614	0.88
Apr	3	1619	1712	1560	93	1805	924	0.11	579	1479	0.86
May	4	2207	2392	2185	341	2733	2053	0.20	1470	2081	0.8
Jun	5	2558	2814	2537	418	3232	2810	0.17	2061	2485	0.78
Jul	6	2970	3179	2878	491	3670	3305	0.17	2475	2827	0.78
Aug	7	3203	3480	3160	550	4030	3729	0.17	2819	3118	0.78
Sep	8	3460	3716	3378	579	4295	4059	0.17	3111	3347	0.79
Nov	9	3703	4036	3667	639	4675	4355	0.17	3342	3625	0.78
Dec	10	4162	4553	4151	758	5311	4794	0.19	3617	4074	0.78
later					955		5508	0.21	4060		0.75

Furthermore, the quality investment status is analyzed according to project data as QEVM suggests. In that case project, not much preventive activities planned but more detective activities devoted to quality investment. The quality investment data is given in Table 62 and Table 24 shows the summary of the quality investment that could be used for comparisons and benchmarking.

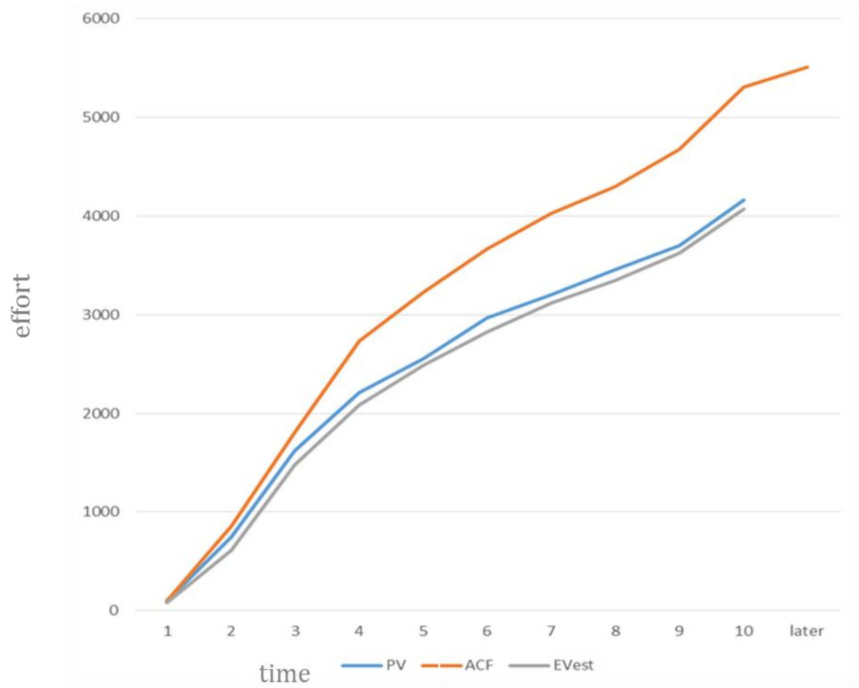

Figure 16 Case IV QEVM Graph

Table 24 Case IV QEVM Conformance Costs Summary

	CC	PV	CCI	PCI	ACI
Case IV - Total	1150	4162	0.28	0.01	0.26

Findings

The initial EVM application results show that the project has a cost overrun but almost on time (see Figure 15).

The cost overrun starts with 10 person-hour and increases later till 400 person-hours with the CPI index changing around 0.90. This linear increasing trend of cost variance and the stable cost performance index gives an alarm about the cost problem of the project. The future estimation metrics show that the final project estimation will be around 4580 ±20 person-hours.

The schedule seems on track with SPI index between 0.97 to 1 after the initial phase. Based on that, the tasks were mostly implemented on time and the deadlines almost met without any delay.

During execution of the project, there are additional rework efforts including errors and changes related with the existing features in the current phase. They are already recorded in the issue tracking tool as development errors/changes that we call internal failure in QEVM. AC of EVM already includes the efforts for these internal failures. Additional reworking efforts after the features released by the development team are not included in AC.

Therefore, EVM implementation does not give any information about the efforts spent for errors and changes after feature completed. Initial project plan does not include any task related with errors or changes. EVM could not track them after once completed because it is comparing what we plan and what we achieved.

Figure 15 includes the reworking inside the releases but not after reworking efforts. The actual cost of EVM is not correct anymore since more effort needed after it is calculated and not updated. The figure shows that the project is on time but there is a cost overrun that might be caused by the internal failures. EVM does not differentiate these FCs from the first time implementation costs.

QEVM application adds the EFCs to the actual costs. Therefore, it represents all the costs more accurately by including the EFCs and so results in higher cost variances and lower cost effectiveness. The actual costs of EVM and QEVM are given in the table, Table 63. At the end, QEVM considers more than 1000 person-hours due to EFCs. Starting from iteration 2, QEVM makes the reworking effort spent for failures visible. The total effort of project team becomes more solid and their real performance is seen.

The new quality effectiveness index of QEVM, QPI indicates the ratio of the tasks completed right first time in the project. Starting with the first appraisal activities, QPI index is decreasing from 0.9 to 0.78 during planned iterations, and the value is 0.75 at the end (see Figure 30). Therefore, at the end of the project, only 75% of the tasks are done right first time. QPI trend is linear during the project.

QEVM presents adjusted EV_{est} values as seen in Table 64. Through the project, it is a better indication of the gained value comparing to EV.

The cost performance index of QEVM is around 0.77 and that is lower than EVM's CPI. Since QEVM spots more serious cost problems considering FCs, it provides us to estimate more cost overrun than EVM. SPI is seen relatively better with QEVM but it still does not solve its essential problems of EVM. The comparisons of indices are given in Table 65.

QEVM calculates that the project will be completed around 5400 person-hour, which is almost 900 person-hour more than EVM's estimate. The difference of total cost variance and other forecasts are compared in Table 66. QEVM further predicts total FCs more than 700 person-hours.

Table 67 presents the results of MREs that are calculated for EAC by means of both EVM and QEVM. While MRE of EVM is changing between 0.16 and 0.17, QEVM gives MRE values between 0.01 and 0.03. Starting from the initial iterations, QEVM predicts EAC much better as seen in the table.

Additionally, at the beginning of the project, CCI of QEVM shows that this project invests 28% of its budget to ensure good quality. 1% of this CCs devoted to the activities that provide good quality while 26% of that is reserved for the appraisal activities detecting the quality problems. The project manager would benchmark these percentages and could evaluate the investment on good quality at the beginning. It is obvious that the company prefers detecting and fixing errors rather than preventing them.

In summary, during execution of the project, the project manager realizes more clearly the cost problems and expects more cost overruns utilizing QEVM. Also, the project manager has the awareness of the high and increasing failures and their cost with QPI index. Therefore, he would get some precautions after understanding the root cause of the failures or would increase preventive activities and so on. However, in the real case, they just focused on the finish them as soon as possible with night and weekend overtimes. The failures are perceived as troubles that need to be fixed immediately and so they are not measured and managed. QEVM also gives much better future estimates and allows the project manager to re-plan the activities or budget or scope based on this fact.

4.3.2.3. 3rd Case Study: Project V, Company C

Organizational and Project Context

Project V is the software development project of a new product, which is an innovation management tool supporting innovation management processes effectively in an organization from idea gathering to project realizations.

We performed this study in a Turkish solution provider for information and communication technologies in the local and global market, called Company C, employing approximately 90 software engineers in Ankara. Company C is both developing their own various products in their product portfolio and delivering turn-key projects to their local and global customers from different sectors. It has ISO 9001:2008 certification and develops software projects compatible with PMI standards and agile practices.

The project team includes 10 staff including:

- 1 project manager
- 1 product manager
- 1 business analyst
- 3 software engineers
- 2 part-time senior software engineers
- 1 senior test engineer
- 1 part-time configuration manager

The team is all located in the same room except the configuration manager. The development team has a daily close collaboration with the product manager.

The project started in December 2012 and planned to be completed at the end of April 2013. However, it could be completed with 2 months delay at the end of June 2013. The project follows iterative and incremental development approach. Since the product manager wants to release the tool into the market as soon as possible, they divide the project timeline into the monthly iterations that produce working software. Therefore, every iteration includes development, testing and reworking activities. The summary of total effort is given in Table 25.

Table 25 Case V Total Effort

Item	Effort (person-hour)
Total Planned Effort	5984
Actual Effort	7016
Rework Effort	3204

The project is developed using the following software products and programming languages:

- Java technologies including Hibernate and JSF frameworks, with a combination of RichFaces and PrimeFaces component libraries
- JBoss Seam framework on Eclipse IDE
- SVN for configuration management
- Atlassian Confluence for the requirements database
- Atlassian Jira for managing all the project tasks, changes, improvements and errors
- MS Project and MS Excel for project management

In a general way, the company manages the reworking by means of contingency buffers planned after the testing activities. The duration of the buffers generally depends on their own previous experiences of the project managers. However, since this project has a very tight schedule, no buffer planned at the beginning.

Conduct

The case study was initially conducted in September 2013 but revised in July 2014 based on the major change on QEVM.

Initial contact with the project manager was a semi-structure interview containing the questions about the brief project information in January 2013. The project just started a month ago at that time being. We got the initially released project plan, first progress report and the error report in the following week. We plan to get the progress reports and error reports monthly in order to apply the model ongoing project but the project manager could not send the project data as often as planned due to their tight schedule and troubles in the project. Eventually, he shared all the progress and error reports at the end of the project and we apply the model backward just as the other case studies.

The project manager delivered us the project data in two parts:

- The first part contains three documents: project plan baseline, updated project plan and monthly progress reports. The plan includes short definitions, planned and actual efforts, planned and actual dates of the tasks. Monthly progress reports including charts for every iteration are provided in Excel sheets. The project manager is marked the tasks related with CCs in the project plan baseline upon our request.
- The second part is a list of errors and changes exported from Jira including the short definitions, the responsible, detection and resolution dates, detected and resolved versions and efforts spent for resolution.

All the necessary data were mainly composed of these documents. Afterwards, another semi-structured interview were conducted with the project manager to resolve the conflicts in the documents.

The progress reports are based on the completed features and reflect their progress. The reworking effort is not tracked in the progress reports but the report only includes how many errors are fixed and how many changes are performed. On the other hand, this project uses issue tracking tool very frequently. They record all kinds of errors and changes in the tool and make the changes afterwards.

Both traditional EVM and QEVM have been applied by the author of this thesis in 4 weeks. We applied EVM and QEVM on every iteration in parallel to project iterations, so the month is used as the time unit during QEVM application. The effort, in person-hour, is used as cost unit.

In the project, the relation between the errors/changes and the initial tasks or phases is not tracked, so it is not possible to understand the origin of the errors and changes. Hence, we use project-based QEVM approach, which assumes that errors and changes belong to the past as a whole.

Results of traditional EVM Application

EVM application is mainly based on the project plan baseline and their realization data retrieved from updated project plan and progress reports. In the following, Table 26 provides EVM application data and Figure 17 shows the graphical EVM representation for the new features that are tracked by means of the project plan.

The EVM graph shows that there is a minor delay in the schedule and cost overrun starting from iteration 2.

Table 26 Case V EVM Application Results

	Iteration	PV	AC	EV
Dec	**1**	1304	1280	1240
Jan	**2**	2544	2760	2464
Feb	**3**	3824	4240	3624
Mar	**4**	4984	5480	4824
Apr	**5**	5984	6556	5824
May	**6 (late)**	5984	6936	5904
June	**7 (late)**	5984	7016	5984

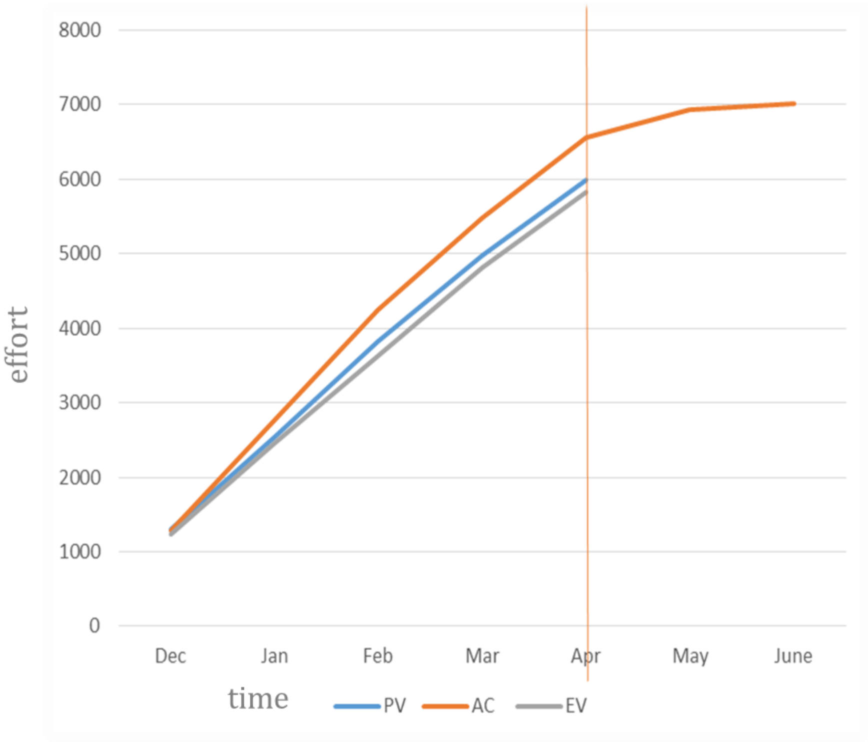

Figure 17 Case V EVM Graph

Results of QEVM Application

As soon as applied EVM, we started QEVM application with collecting FCs. The error list document exported from Jira is the main source of FC data. In the document, the origin of the issue is not recorded. The detection and resolution date and efforts are available. So based on the resolution dates, we grouped the errors and improvements and match them with the iterations (see Table 69, Table 70 and Figure 31).

Since there is no relation with the errors and the origins, we do not have an idea about the owner of the failure and accordingly regained earned value. In such a case, QEVM suggests that all the failures found belong to the past and apply project-based QEVM approach in a cumulative way.

Then, we go on with ACF calculations using FC and its changing value in course of time as seen in Table 71. Similarly, in Table 27 ACF for the previous period, ACF(t-1), representing the actual cost of the previous iteration (t-1), including the FCs spent during the recent iteration. The qcf calculation is accomplished as suggested for project-based QEVM.

Accordingly, we obtained EV_r and EV_{est}. The application results are given in Table 27. Besides, Figure 18 shows the graphical representation of QEVM application.

Table 27 Case V QEVM Results

							considering previous iteration, t-1				
Iteration		**PV**	**AC**	**EV**	**FC**	**ACF**	**ACF_{t-1}**	**qcf_{t-1}**	**EV_r**	**EV_{est}**	**QPI**
Dec	**1**	1304	1280	1240	0	1280	0	0	0	1240	1
Jan	**2**	2544	2760	2464	296	3056	1576	0.19	1007	2234	0.89
Feb	**3**	3824	4240	3624	696	4936	3456	0.20	2218	3390	0.84
Mar	**4**	4984	5480	4824	1128	6608	5368	0.21	3380	4572	0.79
Apr	**5**	5984	6556	5824	1648	8204	7128	0.23	4547	5593	0.75
May	**6 (late)**	5984	6936	5904	2408	9344	8964	0.27	5555	5883	0.65
June	**7 (late)**	5984	7016	5984	3224	10240	10160	0.32	5879	5904	0.54

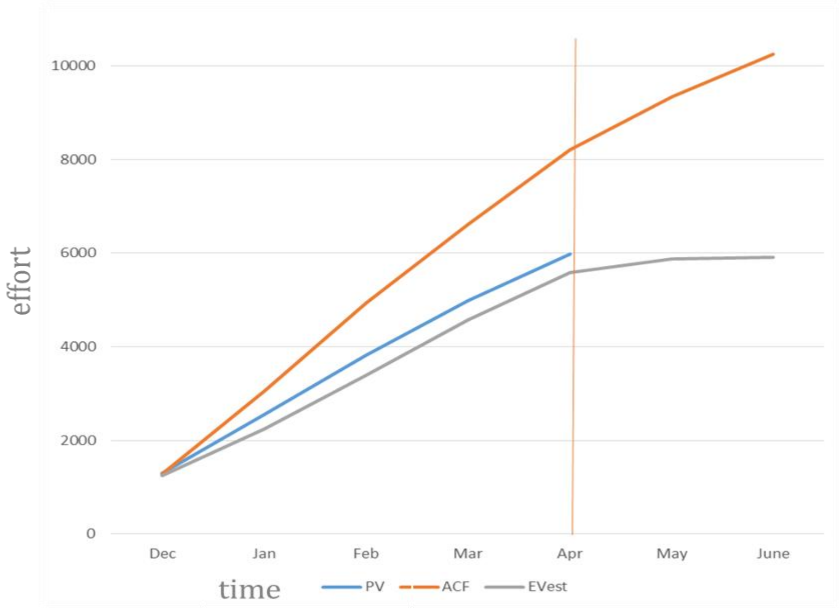

Figure 18 Case V QEVM Graph

Afterwards, we calculate all QEVM performance analysis and forecast metrics and the results are given in Table 73.

Furthermore, the quality CCs are gathered according to project data as QEVM suggests. In this case project, there are some prevention and appraisal activities planned. The planned conformance cost data for the iterations is given in Table 78. The following table presents the overall conformance cost status. It shows the summary of the quality investment that could be used for comparisons and benchmarking (see Table 28).

Table 28 Case V QEVM Conformance Costs Summary

	CC	PV	CCI	PCI	ACI
Case V - Total	1080	5984	0.18	0.07	0.11

Findings

The initial EVM application results give us information about the overall status of the tasks scheduled in the project plan. The results show that there is a minor delay in the schedule and cost overrun starting from iteration 2 (see Figure 17).

The cost overrun starts with 40 person-hour and increases later with the CPI index changing between 0.97 to 0.85. This increasing trend of cost variance and the change of index gives us an alarm about the cost problem of the project. The numbers show that there will be cost overrun and at the end the final project estimation will be around 7000 person-hour at most.

The delay in the schedule starts with 64 person-hour and almost stays at the same ratio with the SPI index between 0.97 to 0.95. Based on that schedule metrics, the tasks were mostly implemented on time and the deadlines almost met without major delay. The tasks were in general implemented in the planned time interval. All minor releases were published on time including the planned features almost done.

In summary, during application from December to May, EVM shows that the project is almost on time and cost effectiveness is about 85%.

QEVM application presents fairly different picture and more additional information by taking the quality costs into consideration in addition to all EVM data.

At the beginning of the project, CCI of QEVM shows that this project invests 18% of its budget in ensuring good quality. 7% of this CCs devoted to the activities that provide good quality while 11% of that are reserved for the testing activities detecting the quality problems. The project manager would benchmark these percentages and could evaluate its investment on good quality at the beginning.

The new quality effectiveness index of QEVM, QPI, shows the effectiveness of the quality, which corresponds to doing things right first time. Starting with the first appraisal activities, QPI index is decreasing from 0.89 to 0.75 during planned five months, and the trend continues till 0.51 at the end (Figure 32). It means at the end of the project, only 51% of the tasks are done right first time. The exponential decreasing trend of QPI also shows that more tasks are reworked during the course of the project.

Considering all the later FCs as well as actual implementation costs of tasks, QEVM results reflect the real cost spent for the iterations (ACF), bigger cost variances and lower cost effectiveness (see Figure 18 and Table 27). The comparison of EVM and QEVM actual costs is given in the following table, Table 72. At the end, QEVM calculates more than 3000 person-hours due to FCs. QEVM represents all the costs spent more accurately by showing up the ignored FCs.

Starting from iteration 2, QEVM makes the failures and the reworking effort spent for failures visible. The effort of project team becomes more accurate and their overtimes become remarkable. The project manager can investigate the causes of reworking and take the necessary actions to reduce them.

EV_{est} values of QEVM reflects the current earned value more accurately comparing to EV of EVM (see Table 74). EV_{init} shows the gained EV on the condition that no rework has done. Although there is a high reworking effort, the gap between the EVM and QEVM values is not so high since the EV of the previous iterations are regained by means of reworking, fixing the failures. The gap only reflects the potential decrease in the EV of the current phase. EV_{init} shows that how EV would look like if there is no reworking at all till that time.

QEVM results spot more serious cost problems considering ACF and EV_{est} and so expect more cost overrun than EVM.

Table 75 compares the performance indices of EVM and QEVM. In QEVM, CPI is decreasing till 0.68 during the planned time interval and 0.58 at the completion time. CV reaches more than 4300 person-hours. Although QEVM has relatively reduced SPI and SV values than EVM, it still has essential problems of EVM mentioned by Lipke [28].

In addition to below better estimates of future forecasts, QEVM presents the metrics regarding expected FCs. QEVM expects the total cost around 9000 person-hour during planned iterations and calculates total cost variance around 3000 person-hours (see Table 76). The MREs for EAC also shows that how better QEVM predicts the future since the initial phases comparing to EVM in Table 77. QEVM calculates EAC between 0.20 to 0.07 just after collecting initial failures in phase 2 but EVM gives much higher error rates changing between 0.40 to 0.31.

4.3.2.4. 4th Case Study: Project VI, Company D

Organizational and Project Context

Project VI is the software development of Tactical Shooting Simulator system, which includes developing simulation software and visual database for Turkish Special Forces for shooting training simulator.

Company D is a software company operating mainly in the defense sector. They mostly focus on R&D projects to develop high technology products and engineering solutions for the local and global market. The company is located in Ankara and employing almost 140 employees, including engineers and researchers. The company has ISO 9001:2008 certification. They use IEEE/EIA 12207 and MIL-STD-498 standards for software development and also develop projects compatible with AQAP 160 and AQAP 2110 standard. They follow PMI standards in project management and set up PMO a year ago in order to manage the projects in a standardized way.

The project team consists of 9 staff with the following roles:

- 1 project manager
- 1 part-time quality manager
- 3 senior software engineers
- 2 system engineers
- 1 modeling expert
- 1 test engineer

The project started in August 2014 and is still going on with the expected completion date of 2016, March. It follows waterfall development methodology. The project was at the beginning of the development at the time being that case study conducted. The summary of total effort is given in Table 29.

Table 29 Case VI Total Effort

Item	**Effort (person-day)**
Total Planned Effort (till 2016)	1225
Total Planned Effort (as of March 2015)	536
Actual Effort (as of March 2015)	576
Rework Effort (as of March 2015)	74

The requirements are kept in an industrial tool, DOORS. For project management, MS Project and Redmine are utilized. Issue tracking is already included in Redmine.

The company manages the reworking by means of contingency buffers and decide the duration of the buffers depending on the previous experiences. No historical data kept specific to reworking in the organization.

The project data could not be continuously retrieved as initially planned after March 2015 due to the workload of the contact person in the company and we only perform analysis on the data till March 2015.

Conduct

The case study has been started in December 2014 and completed in March 2015. The initial contact person was the head of PMO, to whom we explained the purpose and details of the study. Later he selected the appropriate project from their portfolio and we scheduled a semi-structured interview to get the brief project information.

After this initial interview, the head of PMO provided the project data in two excel sheets that we provided to him:

- The first sheet includes the planning and realization data of the project, including task numbers, short definitions of the tasks, planned and actual efforts, planned and actual timelines of the tasks. He already marked the CCs related tasks in the first part as a preventive or detective one or none.

- The second part contains the list of errors and improvements including the short definitions, detection and resolution dates and efforts spent to resolution, related task numbers.

Both traditional EVM and QEVM have been applied by the author of this thesis in 2 weeks. We applied EVM and QEVM on every phase of the project, so six weeks period are used as the time unit during QEVM application. The effort, in person-day, is used as cost unit.

In the project, the relation between the errors/changes and the initial tasks or phases is tracked for development tasks. For the others, analysis and design tasks, they are only marked as analysis or design issue. Therefore, we apply phase-based QEVM approach.

Results of traditional EVM Application

We conducted EVM application based on the project plan that the project manager provided. Table 34 shows EVM application data and Table 79 presents its performance analysis. The graphical EVM representation till March 2015 is given in Figure 20 while Figure 19 shows the planned data of the complete project in an EVM graph.

Table 30 Case VI EVM Application Results

Phase		**PV**	**AC**	**EV**
Phase 1 (initiation)	1	122	122	122
Phase 2 (analysis & design)	2	295	317	288
Phase 3 (design)	3	386	416	373
Phase 4 (dev1)	4	536	576	509
March 2015				
Phase 5 (dev2)	5	214	-	-
Phase 6 (dev3)	6	106	-	-
Phase 7 (dev4)	7	208	-	-
Phase 8 (test)	8	161	-	-

The EVM graph displays that there is a minor delay in the schedule starting from phase 3 and a cost overrun starting from phase 2.

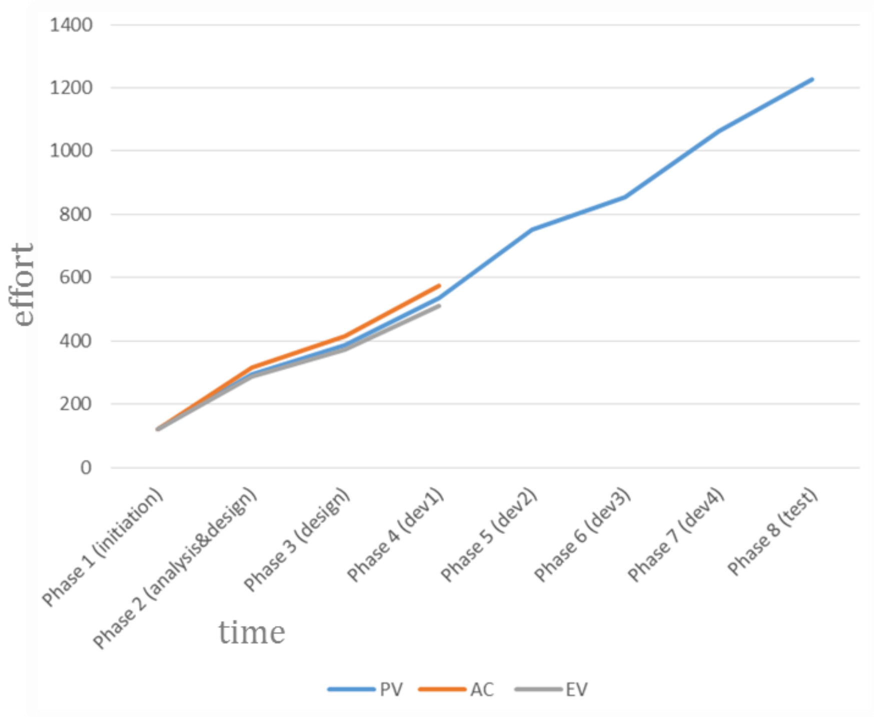

Figure 19 Case VI EVM Graph including the whole project

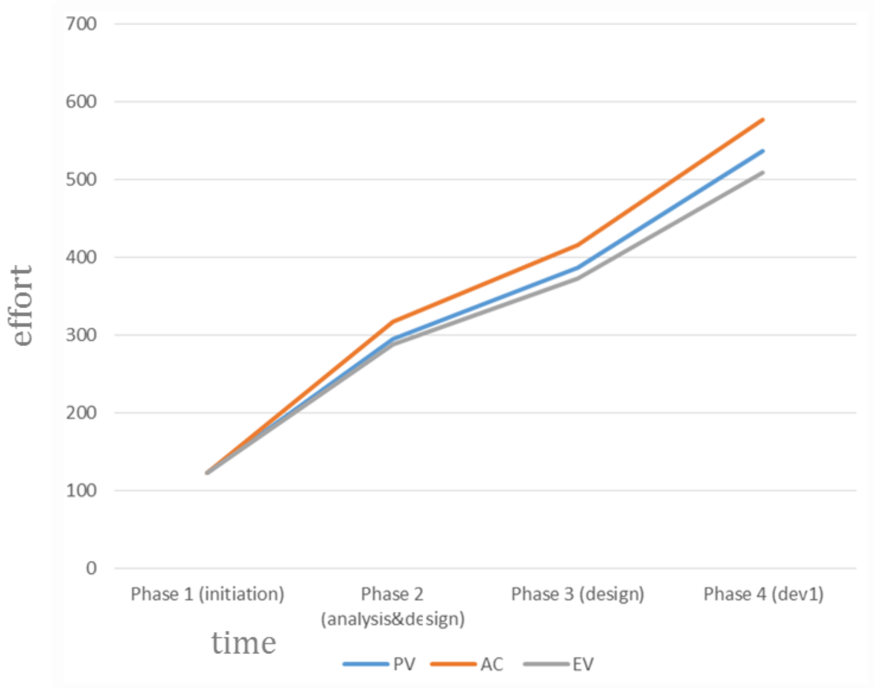

Figure 20 Case VI EVM Graph

Results of QEVM Application

The issue list document provided us is the main source of FC data. In the document, the origin of the issue is recorded if it is development related. For the analysis and design issues, only their type is recorded. First, we grouped the errors and improvements according to the timeline of the phases (see Table 80 and Figure 33).

Since the development methodology is waterfall and the origin phase of the issues is known, we apply phase-based QEVM approach on requirements, design and development phases. The test phase and development phases, phase 1, phase 2, phase 3 and phase 4 are also considered separately. Then we grouped the issues according to the phases and calculated qcf for the separate phases (see Table 31). FCs and qcf are increasing through the time as seen in Table 81 and Table 82. So, QEVM allows us to keep the qcf for these separate phases.

Table 31 Case VI FCs and qcf grouped for the Phases

	FC	**qcf**
Initiation	0	0
Analysis & Pr. Design	29	0.15
Design	28	0.28
Dev1	17	0.11

Afterwards, we obtained EV_r based on the qcf and so EV_r is changing based on the changing values of qcf (see Table 83 and Figure 34). It indicates that as much as the failure occurs EV is calculated retroactively and might have a different value. Figure 34 shows how the different phases have the retroactive EV in a bar chart. This figure could be extended and new bars would be available next to the existing ones through the time. The figure might help the project manager to comprehend the status of the project better and obtain more realistic predictions.

After this detailed phase investigation, the QEVM metrics are calculated the project-based way to look the overall status. Here we utilized qcf values obtained from the phases and get their averages in the previous phases. Hence, EV_{est} is calculated and QEVM application results are given in Table 32. Besides, Figure 18 shows the graphical representation of QEVM application.

Table 32 Case VI QEVM Results

Phase	**PV**	**AC**	**EV**	**FC**	**ACF**	**qcf**	**EV_{est}**	**QPI**
Initiation	122	122	122	0	122	0.00	122	1.00
Analysis & Pr. Design	295	317	288	0	317	0.00	288	1.00
Design	386	416	373	22	438	0.08	366	0.95
Dev1	536	576	509	49	625	0.17	486	0.92
later				25	650	0.18		0.89

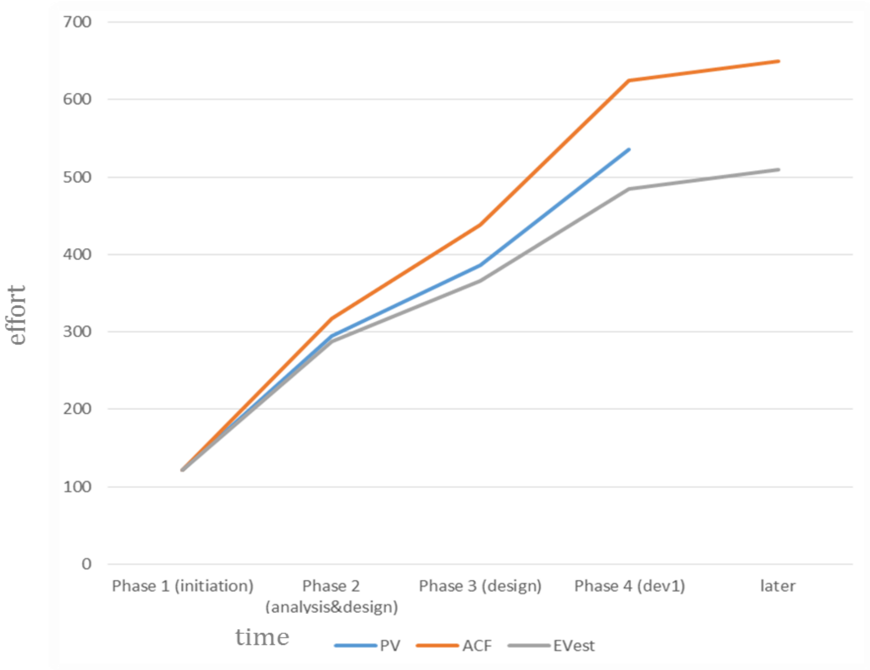

Figure 21 Case VI QEVM Graph

Afterwards, we calculate QEVM performance analysis and forecast metrics just for four periods and the results are given in Table 84.

Furthermore, the planned quality CCs are gathered from the project plan. In this project, there are some prevention and appraisal activities planned depending on the phases. The prevention activities are mainly planned initial phases while the appraisal activities are planned through the end of the project. The planned CC data for the iterations is given Table 89 and the following table, Table 33, presents the CC summary. It could be used for comparisons and benchmarking.

Table 33 Case VI QEVM Conformance Costs Summary

	CC	PV	CCI	PCI	ACI
Case VI - Total	152	1225	0.12	0.004	0.12

Findings

Initial EVM application gives us the overall status of the tasks scheduled in the project plan. The results show that there is a minor delay in the schedule starting from Design phase and minor cost overrun starting from analysis and preliminary design phase (see Figure 20).

The cost overrun starts with 29 person-hour and increases later with the CPI index changing between 0.91 to 0.88. This increasing trend of cost variance and the change of index might be the signal of the cost problem. The numbers show that there will be cost overrun and at the end the final project estimation will be around 1386 person-hour versus 1225 person-hour planned.

The delay in the schedule starts with 7 person-hour and increases with the SPI index between 0.98 to 0.95. Based on that schedule metrics, the tasks were mostly implemented on time and the deadlines almost met without major delay. However, the increasing index might be the indicator for further delays.

In summary, for the early four phases of the project, EVM shows that the project is slightly behind schedule and has a cost overrun with 88% cost effectiveness.

QEVM application presents more information from the quality cost perspective and gives more detailed results and accurate results including these costs. Since the project applies waterfall development methodology, it is also very helpful for the later similar projects.

First, CCI of QEVM shows that this project reserves 12% of its budget to ensure good quality at the beginning of the project. This project dedicated very few hours to preventive activities. Only 0.5% of the total cost is devoted to the activities that provide good quality while 11.9% of total cost is reserved for the testing and review activities to identify the quality problems (see Table 89). The project manager would benchmark these values and evaluate the quality investment at the beginning of the project.

QPI is decreasing from 0.95 to 0.89 during performed four phases, and the trend is decreasing (see Figure 35). It means that 89% of the tasks are done right first time. Since the project applies waterfall development methodology, few appraisal activities are performed at the initial phases. They only include review activities and corresponding fixes as a result of reviews. The FCs are mainly composed of those activities till the end of first development phase. That's the reason why the ratio of the QPI is not so high.

The phase-based application shows us the changing FCs, qcf values and retroactive EVs in detail (see Table 81, Table 82, Table 83 and Figure 34). These tables show that whenever failure cost spent, EV is calculated retroactively and may have a different value. Figure 34 shows the retroactive EVs of the phases in a bar chart. This figure would have new bars next to existing ones through the time. This qcf values could be used for the later projects with the similar characteristics. In particular, since the project is waterfall, the FCs might happen late and in earlier phases it might not be so easy to catch the trends. For that type of the projects, the data of the earlier projects are extremely valuable. During the execution of phase 5, we calculated the qcf values: 0.15 for the requirements, 0.28 for design and 0.11 for development. Those values indicate that these phases have that ratio of reworking.

QEVM reflects bigger cost variances and lower cost effectiveness (see Table 32 and Figure 21). The comparison of EVM and QEVM actual costs is given in the following table, Table 85. Till phase 5, QEVM has more than 74 person-hours actual cost due to failures. QEVM represents all the costs spent more accurately by showing up the ignored FCs.

EV_{est} indicates the current EV adjusting based on the past failures and the comparison is given in Table 86. The difference is not so high comparing the cumulative EVs. It is is 5% of the cumulative one but around 25% of the current phase. EV_{init} shows how EV would look like if there is no reworking at all till that time.

By including quality costs, QEVM spots cost problems better and points more cost overrun than EVM.

Table 75 shows that CPI is decreasing till 0.70 in phase 5 based on QEVM results but EVM gives 0.88 at the same point. SPI of QEVM is slightly different from the SPI of EVM, which are 0.95 vs 0.91.

Accordingly, QEVM expects the total cost around 1800 person-hours and gives total cost variance around 550 person-hours (see Table 88) while EVM calculates 1400 person-hours and 160 person-hours in sequence. Furthermore, QEVM present two more metrics related with FCs and

Since the project is still ongoing, we do not have an actual costs and so cannot apply MMRE analysis for this project.

4.3.2.5. 5th Case Study: Project VII, Company E

Organizational and Project Context

Project VII is an ongoing civil aviation project which develops Electronic Flight Bag system. The system contains main management station and mobile application. It is composed of nine sub-modules that provide "Paperless Cockpit" managing whole flight life cycle through mobile devices rather than printed papers.

Company E is the leading company in the fields of consultancy services and software development in Turkey. They develop software projects for military and civil systems in the national and international platforms as well as provides system engineering, professional services and technical support services. The company is located in Ankara and employs nearly 500 engineers.

Company E holds SEI CMMI Level-3 and has also ISO 9001:2008, NATO AQAP-160, ISO/IEC 27001 and NATO & National Secret Level Facility Clearance certifications. They develop projects in accordance with the following standards; PMI Standard Practices in Project Management, EIA/IS-632 and IEEE 1220 in System Engineering, IEEE/EIA 12207, MIL-STD 498 in Software Engineering, MIL-STD-973, ANSI/IEEE 1042, IEC/ISO 15846 in Configuration Management. The company set up PMO a year ago in order to manage the projects in a standardized way.

The project team includes 7 staff profiled as the following:

- 1 part-time project manager
- 2 part-time senior software engineers
- 1 part-time senior software engineers
- 1 part-time system engineer
- 1 part-time quality manager
- 1 part-time configuration manager

The software engineers and the other team members are working for other projects in parallel depending on the planning and prioritization of the tasks. In the company, they frequently apply multi-project management. There is no resource 100% dedicated this project. They generally work for this project around 30% of their time.

This system contains two main phases and the first phase already completed last year. The second phase of this system is managed as a separate project and development of the project started in January 2014. The project activities completed at the end of December. The project has a major release in October 2014. In the scope of this study, we include the second phase of the system spanning the period from January 2014 to December 2014. After the release in October and during May, depending on the project needs and the availability of the team, no activities has been performed and these sprints is reserved for the testing activities of the customer test team. Scrum is being used for project management activities with monthly sprints. They perform monthly sprint review meeting and weekly status meetings. The total development effort is given in Table 34.

Table 34 Case VII Total Effort

Item	Effort (person-hour)
Total Planned Effort	1117
Actual Effort	1009
Rework Effort	666

All the project tasks, changes, improvements and errors are stored and managed using an industrial tool, Redmine. MS Project is also utilized at the beginning for the project plan. MS Word and MS Excel are used for project documents. No info provided about the programming languages.

Almost every release of the sprints is installed on the customer environment and they also perform their tests with 160 pilots. The responsible tester from customer site gathers all the test results from the pilots and opens issues. They estimate and plan the timeline of all the errors at the beginning of every sprint as well as features. They do root-cause analysis about the errors but do not document them.

They do not plan the reworking in advance. The availability of the project staff may change based on the need of the other projects and the prioritizations. Every sprint is planned at the beginning of the sprint.

Conduct

The case study has been started in February 2015. We initially contacted the head of PMO and he invited us one-hour meeting with the project manager of this project. We explained the study and discuss the needs.

After the initial interview with PMO Head and project manager, the project manager delivered us the related project documents via e-mail. The documents include the following:

- The product backlog and its initial high-level planning
- The project management plan, risk management plan
- The issue list exported from Redmine, which is planning and realization data of the new features and errors, including title, planned and actual effort, % complete, and start date.

The sprint plans are not provided. All the required data were primarily collected from these documents. The document does not contain any info about CC and the project manager could not provide us this data afterwards since they do not keep them. As soon as analyzed the documents, we performed a semi-structure interview lasting half an hour via phone call for the project details, conflicts and resolution of the problems.

Both traditional EVM and QEVM have been applied by the author of this thesis in 2 weeks.

We applied EVM and QEVM on every sprint, so the month is used as the time unit. The effort, in person-hour, is used as cost unit.

Since the project does not record the origin of the errors/changes, we apply project-based QEVM approach.

Results of traditional EVM Application

We conducted EVM application based on the data collected from product backlog plan and the issue lists. Table 35 shows EVM application data and its performance analysis is given additionally in Table 90. The graphical EVM graph in Figure 22 shows us application results graphically.

Table 35 Case VII EVM Application Results

	Sprint	PV	AC	EV
Jan	**1**	174	167	162
Feb	**2**	253	242	237
Mar	**3**	357	325	321
Apr	**4**	605	526	551
May	**-**	-	-	-
Jun	**5**	735	656	665
Jul	**6**	797	702	717
Aug	**7**	903	831	808
Sep	**8**	996	914	901
Oct	**-**	-	-	-
Nov	**9**	1025	937	923
Dec	**10**	1117	1009	1001

The EVM figure and the application results show that there is a considerable delay in the schedule but almost no cost overrun starting from the second iteration through the end.

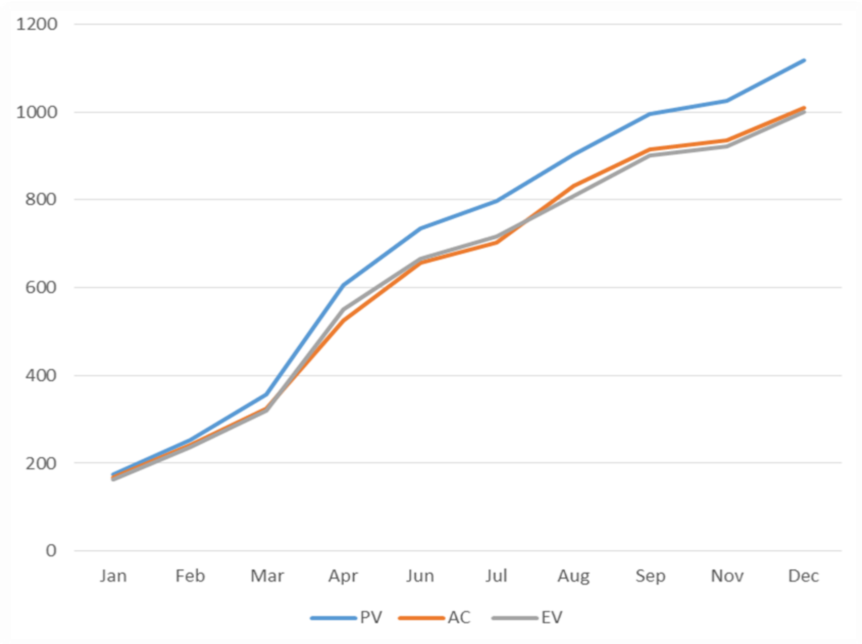

Figure 22 Case VII EVM Graph

Results of QEVM Application

QEVM application is started with the collecting FCs from the issue list exported from Redmine. In this project, the errors and later changes are planned as new feature backlog items at the beginning of every sprint and they are included in sprint backlogs. The issue list provided us only include fix date, planned and actual efforts information for the errors and changes. There is no field regarding the origin of the error.

We gathered FCs assuming they belong to the previous sprints and then calculated ACF and ACF(t-1) (see Table 91 and Figure 36).

Accordingly, qcf, EV_r and EV_{est} calculations are accomplished sequentially. The QEVM results are given in Table 36 and Figure 16 in addition to the performance analyses and future forecasts in Table 61.

The quality investment status cannot be analyzed since they do not keep track of the prevention and appraisal activities and their costs and so the project manager could not provide us the quality cost data.

Table 36 Case VII QEVM Results

							considering previous iteration, t-1				
	Iteration	**PV**	**AC**	**EV**	**FC**	**ACF**	**ACF_{t-1}**	**qcf_{t-1}**	**EV_{r-1}**	**EV_{est}**	**QPI**
Jan	1	174	167	162	82	249	0	0.00	0	162	0.67
Feb	2	253	242	237	104	346	271	0.38	100	208	0.70
Mar	3	357	325	321	121	446	363	0.33	212	293	0.73
Apr	4	605	526	551	179	705	504	0.36	291	469	0.75
Jun	5	735	656	665	231	887	757	0.31	481	630	0.74
Jul	6	797	702	717	278	980	934	0.30	631	702	0.72
Aug	7	903	831	808	390	1221	1092	0.36	698	776	0.68
Sep	8	996	914	901	456	1370	1287	0.35	776	868	0.67
Nov	9	1025	937	923	612	1549	1526	0.40	864	914	0.60
Dec	10	1117	1009	1001	666	1675	1603	0.42	914	969	0.60
later			1129	1952	823						0.58

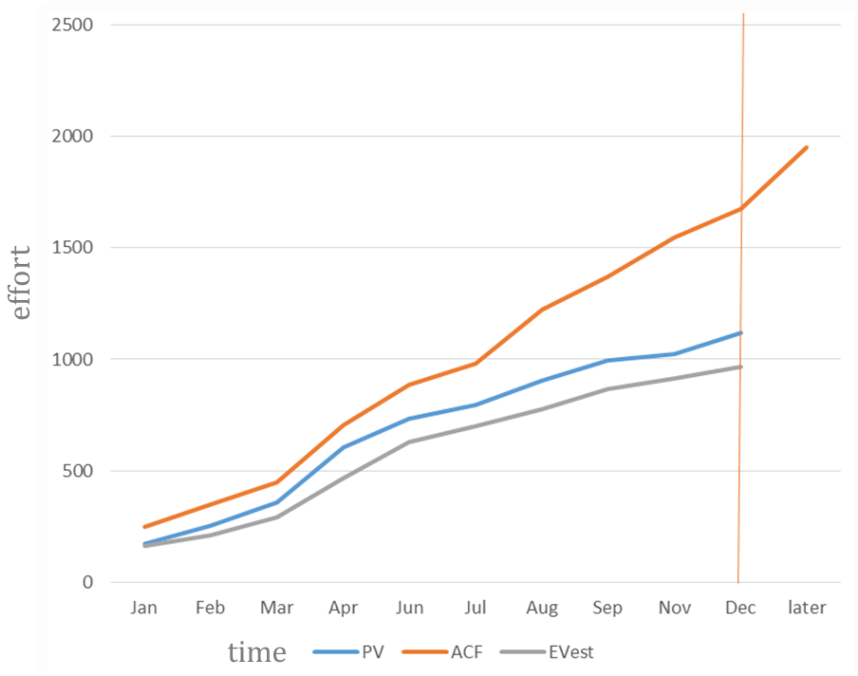

Figure 23 Case VII QEVM Graph

Findings

EVM application results show that the project has a delay starting from the earlier phases but no cost overrun (see Figure 22).

The cost status seems very on track with CPI changing between 0.97 to 1.05. Between the 4th and 7th sprints, it is already more effective than planned budget. This optimistic budget status result that the future forecast for the final budget will be around 1100±30 person-hours similar to planned value of 1117.

The schedule is consistently behind of the planned one with the SPI value 0.90. Based on that, the tasks were not implemented on time and the latency is ongoing through the project.

In summary, from Jan to Dec, EVM shows that the project is behind the schedule 90% but has almost no cost overrun with 99% cost effectiveness.

QEVM application results add the quality costs to the actual cost as well as all EVM data.

There is no data available regarding CCs and so we could not accomplish the analysis and benchmark about the planning of appraisal and prevention activities.

The FCs start with 82 person-hours and increases up to 823 person-hours. The actual costs of EVM and QEVM are given in the following table, Table 93. At the end, QEVM calculates more than 800 person-hours total cost due to the FCs.

EV_{est} values of QEVM reflects the current earned value more accurately comparing to EVM (see Table 94).

QEVM results spot more serious cost problems considering ACF and EV_{est} and so expect more cost overrun than EVM. CPI is changing between 0.72 to 0.58 (see Table 95). CV reaches more than 700 person-hours. Also, QEVM has relatively reduced SPI, changing between 0.89 to 0.93, and better SV values than EVM, which is more than 100.

The new quality effectiveness index of QEVM indicates that only 60% of the tasks are done right first time. QPI trend is initially increasing and then decreasing during the project.

Additionally, QEVM presents the metrics regarding expected FCs. The project manager expects further failure cost (EFC) around 400-700 person-hours as seen in Table 96. This table also shows that QEVM expects the total cost at the end (EAC) around 1700 person-hours and consequently calculates total cost variance around 600 person-hours. The MREs for EAC also shows that how better QEVM predicts the future since the initial phases comparing to EVM in Table 97. QEVM calculates MRE for EAC around 0.40 while QEVM gives much lower error rates changing between 0.20 to 0.01.

4.3.2.6. 6th Case Study: Project VIII, Company F

Organizational and Project Context

Project VIII is the Command and Control System development which integrates emergency management solution with fifteen different application as a part of Integrated City Surveillance System.

Company F is the Turkish subsidiary of a global company serving Consultancy and Systems Integration services on different business sectors including Financial Services, Health, Public Sector, Retail, Telecommunications and Transportation. The company is located in Ankara and Istanbul and employs nearly seven hundred people, including two hundred software engineers in Turkey. Company F holds ISO 9001:2008 and ISO/IEC 27001 certifications.

The project team includes 15 full-time software developers and a project manager. There is no specific analyst, developer or test engineer roles in the team, all engineers are doing all the tasks depending on the needs.

The project started in March 2011 and completed in September 2011. Even though the project is a large scaled integration project combining many different projects, we focus on the software development project since our aim is to apply QEVM on software projects. The software development methodology followed in the project is Waterfall tailored according to the project needs. Each of five development phases following the first analysis and design phase includes the testing activities in it. At the end, there is a deployment and customer training phase, which includes the release with the some missing features. Afterwards, they go on with the project for an additional month to complete the features. The summary of total effort is given in Table 37.

Table 37 Case VIII Total Effort

Item	Effort (person-hour)
Total Planned Effort	12843
Actual Effort	17793
Rework Effort	5445

The project is developed using the following software products and programming languages:

- Java technologies together with Oracle Fusion and TCL/TK scripting
- MSSQL is used for database management system
- Enterprise Architect for software design
- SVN for configuration management
- MS Excel for storing the requirements and test cases specified by the development team
- MS Project and MS Excel for project management
- Bugzilla for the errors and changes

In the project, the project manager plans the project at the beginning and establish their change management process against the changes. For the changes requested by the customer, if it takes more than 1 week, they execute change management and plan those changes separately. The project plan does not include reworking planning. They do not track and explore the causes of the errors and reworking explicitly but the project manager states that the employee turnover, misunderstandings of the requirements and changing requirements are three top causes of the reworking.

Conduct

The case study has been conducted in January 2015.

We initially contacted the project manager via e-mail. Then we explained the study and discuss the needs in a semi-structured interview. After the initial interview with the project manager, she delivered us the project data in MS Project and Excel sheets. The documents include:

- The released project plan and realization data of the plan
- The error reports exported from Bugzilla to an excel sheet

All the necessary data for the EVM and QEVM application were gathered from these documents. The project plan does not contain any info about CC and the project manager could not provide us this data afterwards since they do not keep them. We discuss the inconsistencies and get project details by means of a semi-structure interview in a face-to-face meeting that lasted an hour and resolve the problems.

We applied EVM and QEVM every four weeks, so the month is used as the time unit during applications. The effort, in person-hour, is used as cost unit.

The origin of the errors and changes are not tracked in task level. Although for some errors, the software component is kept, it was not available for all and we follow project-based QEVM approach, which assumes that errors and changes belong to the past as a whole.

Results of traditional EVM Application

Initial EVM application are conducted based on the project plan that the project manager provided. Table 38 shows EVM application results and Figure 24 shows the results graphically.

Table 38 Case VIII EVM Application Results

	Phase	**PV**	**AC**	**EV**
Mar	**Analysis& Design**	2034	2340	1836
Apr	**Development**	4284	5490	3726
May	**Development**	6174	7740	5436
June	**Development**	7974	10053	6984
July	**Development**	10134	12618	8829
Aug	**Development**	12474	15480	10719
Sep	**Deployment& Training**	12843	15975	11088
Oct			17793	12843

This EVM application graph gives a clue about the delay in the schedule and cost overrun from the beginning to the end of the project.

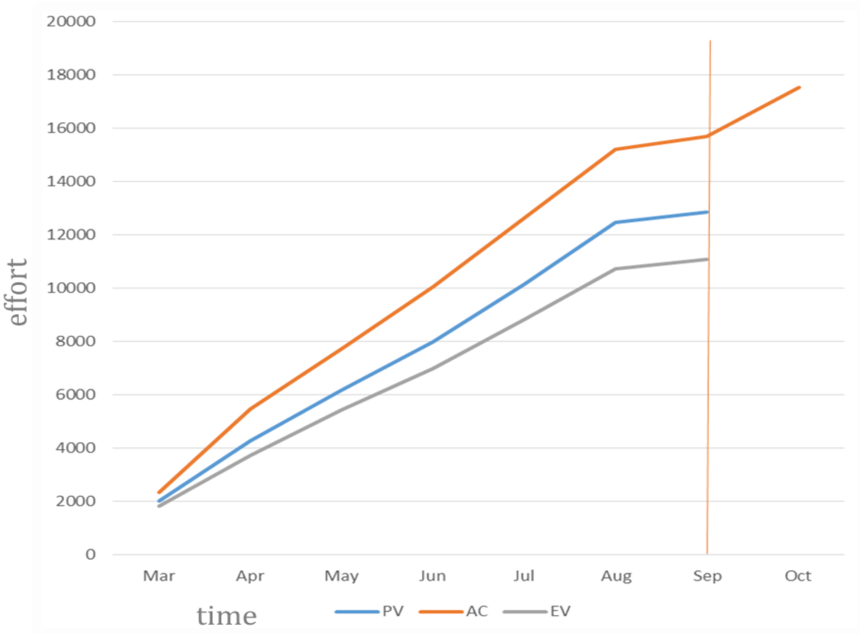

Figure 24 Case VIII EVM Graph

Results of QEVM Application

QEVM application is started with collecting FCs from the error reports. Since we followed project-based QEVM approach, we gathered FC as a cumulative way assuming belong to the past and then calculated ACF and ACF(t-1) (see Table 99 and Figure 38).

Afterwards, qcf, EV_r and EV_{est} calculations are achieved sequentially. The QEVM results are presented in Table 39 and Figure 25 in addition to the performance analyses and future forecasts in Table 100.

Table 39 Case VIII QEVM Results

							considering previous iteration, t-1				
	Phase	**PV**	**AC**	**EV**	**FC**	**ACF**	**ACF_{t-1}**	**qcf_{t-1}**	**EV_r**	**EV_{est}**	**QPI**
Mar	**Analysis& Design**	2034	2340	1836	0	2340	0	0.00	0	1836	1.00
Apr	**Dev**	4284	5490	3726	45	5535	2385	0.02	1801	3690	0.99
May	**Dev**	6174	7740	5436	684	8424	6174	0.11	3517	5247	0.92
Jun	**Dev**	7974	10053	6984	1242	11295	8982	0.14	5200	6770	0.89
Jul	**Dev**	10134	12618	8829	2196	14814	12249	0.18	6706	8498	0.85
Aug	**Dev**	12474	15210	10719	3249	18459	15867	0.20	8451	10332	0.82
Sep	**Training**	12843	15705	11088	4077	19782	19287	0.21	10319	11010	0.79
Oct			17523	12843	4698	22221	20403	0.23	11003	12439	0.79
later					5535	23058	23058	0.24	12422		0.76

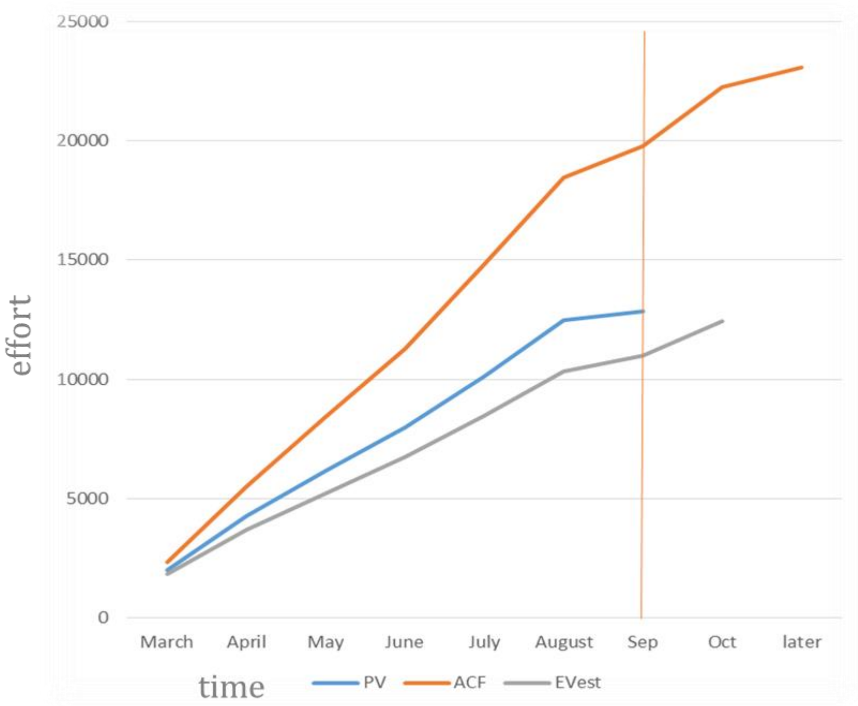

Figure 25 Case VIII QEVM Graph

Since the CC info is not available in the project, we could not perform analysis on the investment status of the project.

Findings

First EVM application results show that the project has a cost overrun and a delay in the schedule (see Figure 24).

The cost overrun starts with 500 person-hour and increases later till 4680 person-hour with CPI changing between 0.78 to 0.69. This stable low cost performance index gives an alarm about the cost problem of the project. Subsequently, the value of EAC is calculated around more than 18000 person-hours, which costs more than 5000 person-hours than planned value.

The delay in the schedule is also considerable since the second phase. SPI is changing between 0.90 to 0.86. The tasks are not implemented on time and the project is behind the schedule with the percentage of 86%.

Although EVM reflects this latency and cost overrun, QEVM presents improved and more accurate numbers, which includes much higher cost variances and lower cost effectiveness.

There is quite a low failure cost in the second period of QEVM, 45 person-hours after the analysis and design phase. It climbs dramatically on the third period and keeps its increasing trend till 4000 person-hours through the end of September, when the project is planned to be completed (see Table 99). The project has total 5500 person-hours costs spent for failures when it is completed with almost two months delay.

FCs constitute the significant part of the total costs. QEVM presents the actual cost with the more accurate numbers. Especially from the 3th period, QEVM makes the reworking effort spent for failures visible. The comparison of ACs is given in Table 101. ACF of QEVM shows that the final cost is almost double of the planned one.

The new quality effectiveness index of QEVM, QPI indicates that around 80% of the tasks completed right first time in the project (Figure 39). At the end of the project, this ratio decreases to 76%.

EV_{est} values of QEVM presents the current earned value more accurately considering possible failures of implemented features (see Table 102). In the table, EV_{init} shows the considerable amount of earned value gained by means of reworking.

Consequently, QEVM results spot serious cost problems considering ACF and EV_{est} and so expect more cost overrun than EVM. CPI is changing between and CV is 0.78 to 0.54. The cost variance exceeds 9000 person-hours. Although QEVM has relatively better SPI, from 0.90 to 0.83, and SV values, from -200 to -20000, than EVM.

Based on the improved CPI, QEVM estimates completion budget more accurately. The project manager expects the final budget about 20000 person hours during the project execution and variance about 10000 person hours (see Table 104). Additionally, the project manager can be prepared with the total failure cost about 5000 person-hours. Although EVM gives MREs for EAC around 0.20, QEVM presents better and improving EAC, with MRE changing 0.16 to 0.005 (see Table 105). QEVM predicts the future just after collecting initial failures and provides a better estimation model.

4.3.3. **Discussion**

We applied the model in six different organizations, in six different projects with different characteristics. Demographics information about the case studies is given in two different tables. Table 40 shows the organization related information covering the domain of the organization, maturity level of the organization, certifications, standards used in the organization, whether the organization has a PMO, whether the organization has ever used EVM, whether the organization has ever used CoQ. Table 41 presents the project related information including project description, team size, team location, programming languages, software development methodology, project management tool, issue tracking tool and project effort. Furthermore, the brief overview is given in Table 42, including planned effort, actual project effort, reworking effort, QPI, MMRE values of EVM and QEVM for. EAC, CC indices; CCI, PCI and ACI.

We applied the model on different software projects in various organizations, which are from different domains like Telecommunication, Defense, Consultancy, E-Government Systems, Information and Communication Technologies and Software Development. Three of the organizations hold CMMI-Level 3 certifications and have a PMO to standardize their project management process.

Table 40 Demographics of the Cases – 1

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
	Project III, Company A	**Project IV, Company B**	**Project V, Company C**	**Project VI, Company D**	**Project VII, Company E**	**Project VIII, Company F**
Domain of the Organization	Tele-communication Systems	E-Government Systems	Information and Communication Technologies	Defense Systems	Software Development and Consultancy Services	Consultancy and Systems Integration services
Organization Maturity Level	CMMI-Level3 compliant according to internal assessment	Preparing for CMMI-Level3	Not available	CMMI-Level3	CMMI-Level3	CMMI-Level3
Organization certifications	ISO 9001:2008	ISO/IEC 20000 IT Service Management, ISO/IEC 27001 Information	ISO 9001:2008	ISO 9001:2008	ISO 9001:2008, NATO AQAP-160, ISO/IEC 27001, NATO & National Secret Level Facility Clearance	ISO 9001:2008 and ISO/IEC 27001
Standards used in the Organization	Organizational Product Development Standards	PMI standards, Scrum practices	PMI standards, Agile Practices	PMI standards, IEEE/EIA 12207, MIL-STD-498, AQAP 160 and AQAP 2110	PMI Standards, EIA/IS-632 and IEEE 1220, IEEE/EIA 12207, MIL-STD 498, MIL-STD-973, ANSI/IEEE 1042, IEC/ISO 15846	PMI Standards
Does the organization has a PMO?	No	No	No	Yes	Yes	Yes
Is EVM ever used in the organization?	No	No	No	No	No	No
Is CoQ ever used in the organization	No	No	No	No	No	No

Table 41 Demographics of the Cases – 2

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
	Project III	**Project IV**	**Project V**	**Project VI**	**Project VII**	**Project VIII**
With whom case study performed?	Project Manager	Quality Manager	Project Manager	Head of PMO	Project Manager	Project Manager
Project Description	Development of Sales Management Product	Maintenance of a Web-Based Procurement Tool	Development of Innovation Management Product	Development of Tactical Shooting Simulator System	Development of Electronic Flight Bag System	Development of a Command and Control System
Project Team Size	7 full-time, 1 part-time	4 full-time, 2 part-time	7 full-time, 3 part-time	8 full-time, 1 part-time	7 part-time	15 full time
Team Location	Co-located in Ankara and Germany	Local	Local	Local	Local	Local
Programming Language	Organization-specific high-level modeling tool	.Net	Java	No info provided	No info provided	Java, TCL/TK scripting
Software Development Methodology	Iterative incremental	Iterative development combined with Scrum practices	Iterative incremental	Waterfall	Scrum	Waterfall and iterative combined
Project Management Tool	MS Excel	MS Project, MS Team Foundation Server	MS Project, MS Excel, Atlassian Jira	MS Project, Redmine	Redmine	MS Project and MS Excel
Issue Tracking Tool	Organizational ticketing system	Microsoft Team Foundation Server	Atlassian Jira	Redmine	Redmine	Bugzilla
Total Project Effort (person-hour)	7600	4553	7016	5184	1009	17793

None of them utilizes ever EVM and CoQ in their projects. The team sizes of the case projects change between 7 part-time to 15 full-time staff. Two of the projects apply iterative and incremental approach, one project follows iterative approach combining them with Scrum practices, one project utilizes Scrum practices and two of them apply waterfall development methodology in their software development processes. The actual effort spent changes between 1009 to 17793 person-hours.

QEVM model applied monthly for all the projects. The project with iterative and incremental approach already has 4-5 weeks length iterations. The one applying scrum has 4 weeks sprints and for the others we set the monthly periods for QEVM application based on the availability of the data. The number of EVM and QEVM application periods of the projects is changing between 4 to 10 iterations. We apply the task-based QEVM on one of the projects and the phase-based approach of QEVM on another one depending on the available data. For the other four projects, we implemented the project-based approach.

In the design of this multiple-case study, our aim was to answer the following research questions and so the results allow us to discuss the answers.

RQ1: Is QEVM helping project managers to see the project current status more clearly and to estimate project future more accurately?

Yes, QEVM provides the more accurate status of the project and so it offers more clear visibility of the project to the project manager. The most significant contribution of QEVM is to include FCs to the estimations and so make them visible. Although there is a huge amount of reworking effort caused by failures and changes in software projects, there are no well-structured standardized solutions to manage those efforts. The most common approach is to fix the failures as soon as possible after noticed. So, this approach makes the projects in low quality and long hours working resulted in overtime of the development team. Since their data is not tracked, it may not be easy to see how big effort spent on the project and how the team is worked too much. Therefore, including the reworking effort during project execution is vital to improve the process and measure the actual project data.

Incorporating FCs into the project tracking results in more accurate actual costs, more effective cost and schedule indices and so healthier future estimates. With "estimated EV" concept, QEVM provides more correct earned value. In the results of the case studies, at first sight, it might be perceived as EV_{est} is not so different than EVM. Since we apply QEVM cumulative and directly checks the values at that specific time, reworking already provided to gain the value for the previous phases, so the difference is in general reflects the differences belongs to latest application period.

Based on the more accurate actual costs and the more correct earned value, it is apparent that QEVM provides much better evaluations of cost status. As we seen in the results of case studies, CPI and CV values are quite enhanced comparing to the ones that calculated with EVM particularly for the high reworking cases. Schedule evaluations slightly better due to better EV estimation but are not dramatically improved by QEVM because EVM has some problems with the schedule estimation spotted before by Lipke [28] like the unit of schedule indicators and behavior of EVM for the projects past the planned date.

QPI is an easy indicator of reworking as well. The project manager comprehends how much of the things are done right first time just checking QPI.

Also, QEVM returned much better future estimates in the case studies. As seen in the QEVM applications of the case projects, QEVM estimates the completed budget quite better than EVM. MMRE values for QEVM is in between 0.02 to 0.12 while EVM gives the values changing 0.13 to 0.42 for the same projects (see Table 42). MMRE also supports that QEVM provides better estimation model.

Quality cost investments is also another concept that QEVM brought and the metrics increase the insight of the project manager at the beginning of the project.

All these benefits of EVM containing more accurate estimates, revealing unhidden costs of failures and better future estimates, QEVM helps project managers to see the project current status more clearly and to estimate project future more accurately.

Table 42 Brief Overview of QEVM Results

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
	Project III	**Project IV**	**Project V**	**Project VI**	**Project VII**	**Project VIII**
Planned Effort	7230	4162	5984	536/1225 (in person days)	1117	12843
Actual Project Effort	7600	4553	7016	576 (in person days)	1009	17793
Reworking Effort	1080	758	3204	74 (in person days)	666	5445
# of EVM Application period	9	10	7	4	10	8
QPI	0.88	0.75	0.54	0.89	0.60	0.76
MMRE-EVM	0.13	0.17	0.34	Data Not Available	0.42	0.22
MMRE-QEVM	0.02	0.03	0.14	Data Not Available	0.11	0.05
CCI	0.16	0.28	0.18	0.12	Data Not Available	Data Not Available
PCI	0.01	0.01	0.07	0.004	Data Not Available	Data Not Available
ACI	0.15	0.26	0.11	0.12	Data Not Available	Data Not Available

RQ2: What are the benefits of QEVM for software projects comparing to EVM?

The benefits of QEVM is tightly related with the previous answer. We could group the main benefits that QEVM provides under three category: visibility, accuracy and predictability

Visibility of the projects is increased by defining the new dimension of quality. QEVM brings the new point of views and so these new viewpoints enlighten some dark corners of the project.

The huge but unhidden costs spent for failures are the major part that delivers more visibility. Although the failures including defects and changes are in general recorded, their management is not integrated into project management. They are more likely treated as a quality issues and just keep as a quality metrics. It is important to integrate them into the project planning to get them under control.

The quality investment of the project is visible by QEVM and the project manager could see them at the beginning of the project. Benchmarking the current investment with the previous projects might give a quick and brilliant idea to the project manager.

The total efforts and cost spent for the project becomes more visible and the later estimations based on the history become more accurate.

The efforts of the project team come to be more visible and their performance could be measured better.

Accuracy is a significant metric that a project manager needs to track a project successfully. Accuracy is considerably related with the visibility. The more visibility brings more accuracy to the projects.

The project progress is measured more accurately comparing to EVM. Especially increased the visibility of FCs greatly affects the calculations of the total cost spent and earned value. The case study results already show the superiority of QEVM in accuracy of the calculations.

QEVM particularly provides more accurate cost metrics including CPI and CV.

The earned value is calibrated based on the previous failures costs and so more accurate EV is estimated.

More accurate progress metrics results in more accurate and realistic future estimates that increases the predictability.

Predictability is vital for a project manager to get the precautions as soon as possible. It is strongly related with the accuracy and so visibility. The more visibility and accuracy result in better predictions for projects.

Based on the accurate project progress and performance metrics, QEVM provides very realistic forecasts. The case study results show us MMRE index is quite low. Also during project execution, MRE of every phase is improving seriously.

The forecasts of estimated completion budget are quite similar to actual ones.

QEVM also offers failure cost related predictions to estimate if more failures will happen and the cost of these failures.

Additionally, we would like to mention a few observations that we obtained during this interviews. We observed that there is an increasing trend establishing PMOs in the software organizations in Turkey recently. Especially we observed that the enterprises realize its significance and already set up in recent years. On the other hand, the maturity of the project management processes is very low in the industry even in CMM certified ones. It was quite difficult to find these case projects and then sometimes collect the data from their side.

In the industry, the people are not so willing to collaborate. Main concerns of the companies are data security. I already signed NDA with one of the companies but I could not reach the project manager later. The other impediment in collaboration is the suspicion of the industry about the results. They already supported similar studies but did not get enough benefits for their side.

4.3.4. **Validity Threats**

The case study research presents threats to validity, which are basically the concerns with the question of how the conclusions might be wrong [92] [93] [94]. It is important to identify and address these threats and so various models have been presented to help researchers in analyzing the validity and mitigate threats [94] [95]. In this section, we discuss the possible limitations and internal and external validity threats of this case study and present actions conducted to mitigate them.

The major threat to internal validity for this study has been the quality and reliability of the data provided and collected.

To improve trustworthiness of data quality, we focus on the selecting the projects from higher maturity levels, CMMI-3 maturity level or having PMO in the organization, since the success of EVM implementation heavily depends on the maturity of the project management process and the accuracy of the collected metrics. Additionally, for all the case projects, we observed many inconsistencies in the project documents and plans and needed to clarify with the project managers afterwards.

Furthermore, in order to increase the quality of the data collected, at the beginning of the study, the study has been explained in detail to the project manager/PMO head/quality manager, the templates for the related data have been provided and the needs have been communicated. Additionally, for all case studies, after data provided, additional sessions have been conducted with the project manager or the people who are the responsible for data to clarify the inconsistencies in the project management document. The issue lists including errors and changes are revisited during these semi-structured interviews.

External validity is our concern since we make generalizations on the results of the study by means of applying QEVM. To mitigate this threat, we conduct multiple case study including six case projects from different organizations and increased diversity of the projects. The organizations were from different business domains including Telecommunications, Government, ICT, Defense, Software Development and Consultancy, as given in Table 40.

Additionally, we selected the projects with different development methodologies and different sizes to increase the diversity as well.

We did not encounter specific difference or difficulty for the application of QEVM in these business and project domains. Hence, we conclude that the model is applicable to any business and project domains.

CHAPTER 5

CONCLUSIONS

EVM is a proven, simple and valuable project management tool in order to track the progress of the project as well as to estimate the future status. It brings together three fundamental elements of the project: scope, schedule and cost. EVM evaluates their progress against initially planned values in order to give a clear picture of the project progress and performance.

Software projects do not utilize EVM as frequently as traditional projects. Even though the traditional EVM is a method integrating scope, schedule and cost for measuring project progress and performance, it does not consider the quality concepts explicitly. Quality is another major factor for the software project success and predominantly affects later progress by means of failures and reworking. In this study, we introduced a new EVM model, QEVM, improving the traditional one in terms of quality concepts.

QEVM is compatible with PMI's project management principles and additionally includes quality costs explicitly. Inspiring from Crosby's quality approach, QEVM defines the quality as doing the things right for the first time and measures the quality of the project by means of cost of quality.

This chapter presents the summary and contributions of this study and finally gives the suggestions for future work.

5.1. Contributions

The major contribution accomplished by this research is QEVM, which is the extension of traditional EVM by incorporating quality cost metrics. QEVM provides the usable and valuable model for software projects since it takes the significant quality costs into consideration. Even though the software projects suffer from a lot of reworking, those costs are not incorporated into traditional EVM. QEVM provides the quality related metrics to the project manager in order to not only track the quality status but also integrate the cost of quality with the project cost status.

Another contribution of this study is that QEVM enables calculation of planned CC. By means of CC and CCI, the project manager initially benchmarks the quality investment status of the project with the similar ones in the company or in the industry. It is important to quantify the quality investment of the project at the beginning. The separate PCI and API could give a chance to review the benchmarking in a separate way, from the costs dedicated to prevention and appraisal activities. In the case studies, we observed that the projects focus more on the APCs than the PCs and so it leads to increasing FCs. By QEVM, the project manager could evaluate the status at the beginning and revise their distributions. For example, if the PCs are really low for a mission critical system, the project manager may increase prevention activities and may try to reduce the failures. This CCI gives a valuable information to the project manager before starting the project.

Visible FCs and their incorporation into EVM are the other main contributions that QEVM provides. Since the reworking occurs a lot in software projects, EVM could not reflect the accurate ACs and cost indices without incorporating them into EVM. The project is always perceived more optimistically than the actual status. If there is no distinct planning for the reworking, the project team would be overloaded but this effort would not be seen clearly. Revealing FCs provides the visibility on the actual costs of the project as well as the performance of the project team. Additionally, QEVM provides more accuracy to the project manager in terms of cost and schedule metrics thanks to more accurate measurement of ACs of the project. Since it explicitly considers and calculates FCs, it affects the actual tracking status of the project completely. QEVM presents more accurate current progress, in addition to accurate cost indices. Schedule indicators are also better with respect to EVM but do not reflect dramatic change due to the essential problems of EVM schedule metrics.

Tracking project quality status at any time in a project execution by means of the QPI is another contribution of this study. The project manager monitors the quality status through QPI by measuring how much of the work accomplished right first time. QPI gives significant information about the quality of the project. If it gets too high during the course of project, the project manager could investigate the reasons and trying to find out the root cause. There might be several possible reasons of failures indicated by the high QPI value. The initial estimations and planned values might be wrong or underestimated and so the tasks are implemented so quickly and in a chaotic way. The team might be under qualified and/or forced to perform the tasks in less time than estimated duration. The project manager gets the necessary actions having identified the problem. For example, if it is related to the insufficient quality investment, the project manager would plan more preventive and detective activities instead of corrective ones and so on.

In addition to these primary contributions, QEVM delivers more visibility to effort and costs, more accurate forecasts and better predictions of future. Including FCs into total costs increases the visibility of the project aspects, quality status and effort become visible. The revealed FCs result in more accurate total cost, schedule and cost indices and so this improves the accuracy of the project. Accuracy in current progress information enables more accurate estimations of future values of project. In the case studies, the major EAC metric of QEVM is calculated for the every phase and the results demonstrates significant improvement in the cost forecasts. The accurate progress information and forecasts are the main targets of project management since they allow the project manager to understand the present clearly getting him necessary actions. Depending on the status of the project, the project manager could get different actions like informing the stakeholders about these trends and forecasts or calibrating the project budget and schedule, or investigating the reasons behind and so taking the necessary actions to make the project on track.

The initial exploratory case study shows that even EVM is a powerful method to reflect project progress in terms of scope, time and cost, especially for the first time implementation of the tasks, it could not represent the later reworking and could not incorporate the reworking costs and its effects into the method. Although the project manager observes that the project is on track by EVM at a given time during project execution, there could be some cost and schedule problems due to the quality issues and subsequent reworking efforts. By providing accurate project progress, QEVM removes this complication.

For the purpose of validating applicability of QEVM, a multiple-case study with six cases from six different organizations was planned and performed. These cases provided differing perspectives to validate the methodology for projects with distinct characteristics. For all the case projects, QEVM provides substantial improvement in estimation of future costs, i.e. EAC, comparing to traditional EVM based on the realized total cost at the end.

Two main objectives of traditional EVM is to measure the progress clearly and to estimate future correctly. QEVM improves both for the software projects with high FCs. If there were no reworking for the software projects, QEVM would be same as the traditional EVM. We would not need such an improvement.

Additionally, the projects using the agile methods make use of QEVM very well. It provides an improvement of Agile EVM by dint of quality concepts. Agile EVM is an adapted implementation of traditional EVM based on the Scrum framework. Instead of planning the whole project at the beginning as in EVM, Agile EVM plans the iterations and so EVM is executed on the releases. Since the agile methods embrace the change and encourage the refactoring, it is expected to have a high amount of FCs in such projects. QEVM could provide the measurements regarding these FCs and quality related metrics. Additionally, further study may improve QEVM by means of the significant concept of agile methods, technical debt.

Main benefits that QEVM provides to software projects are summarized in the followings:

- Providing CC indices and benchmarking opportunity at the beginning of the project
- Revealing hidden FCs and integrating them into project management and performance management - more visibility
- Measuring the quality status of a project at a given time in addition to schedule and cost – more visibility
- Estimating the project progress more accurately at any given time using past quality cost data –more accuracy
- Estimating project future more realistically – more predictability

5.2. Future Work

This section presents further research suggestions and future improvement opportunities discovered during the course of this study.

The proposed QEVM model can be extended in several ways. In the scope of this study, we incorporate CC into QEVM only for planning and benchmarking. However, it would be good to incorporate the performance of prevention and appraisal activities as well as planning. The correlations of prevention and appraisal activities with the failures can be also included in QEVM. Such metrics would demonstrate the occurrence of the failures and their relations with the prevention and appraisal activities. Therefore, the project manager would adapt further prevention and appraisal activities accordingly. It might be more detailed and complex application in terms of quality costs but absolutely gives project managers more insight in order to take precautions.

The forecasts of QEVM give better estimations than EVM but it seems there is still an improvement opportunity. The exponential increasing trend of failures is not so captured in QEVM. Here we suggest using prediction models for qcf estimation. Evaluating the trends of qcf of the previous phases, estimating the new qcf based on the trends would give more accurate results especially in exponential increasing cases. Currently, we calculate reworking trends linearly and take the latest one into consideration as EVM does. However, the behavior of the reworking would be linear or exponential depending on the project. For the projects having exponential reworking trends, it would give more accurate results to estimate qcf value by means of exponential distribution functions. Also, the different types of prediction models could be applied and their usability as well as applicability could be discussed according to the results.

In this study, during EV_{est} calculation for project-based approach, we accept that the reworking belongs to the latest application phase and the earned value at the previous phase is already gained with the existing reworking. The distribution of the reworking according to the phases can be the topic of another research. Therefore, the previous EV is not assumed totally gained and based on the distribution of the reworking EV_{est} can be calculated more reliably.

QEVM does not focus on the details of the performance measurement method and uses "Percent Complete". The other type of measurement methods, including fixed formula, weighted milestone and so on can be studied further and more case studies can be performed accordingly.

Additionally, more case studies can be performed to reinforce the validation of the model for the projects with the different characteristics (large-scale projects, small projects, complex algorithmic systems, data strong systems and so on).

New case studies on ongoing projects can improve the results of the model and strengthen the validation. The application for ongoing projects will reduce the difficulties regarding data collection and so the actual shortages of the model can be visible.

The tool supporting QEVM can be developed or plugin modules can be developed for well-known existing project management tools for QEVM. It would make the implementation of the model easier, would standardize the application process and would provide descriptive charts and graphs.

REFERENCES

[1] The Standish Group, "Standish Store," The Standish Group International, 2013. [Online]. Available: https://secure.standishgroup.com/reports/reports.php. [Accessed 01 08 2014].

[2] D. Van Der Westhuizen and E. P. Fitzgerald, "Defining and measuring project success," in *In: European Conference on IS Management, Leadership and Governance* , Reading, United Kingdom, 2005.

[3] S. Lipovetsky, A. Tishler, D. Dvir and A. Shenhar, "The relative importance of project success dimensions," *R&D Management,* vol. 27, no. 2, pp. 97-106, 2002.

[4] Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK® Guide), 5th ed., Newton Square, PA, USA: Project Management Institute, 2013.

[5] F. P. Brooks, "No Silver Bullet Essence and Accidents of Software Engineering," *IEEE Computer,* vol. 20, no. 4, pp. 10-19, 1987.

[6] A. Cass, S. M. Sutton Jr and L. J. Osterweil, "Formalizing Rework in Software Processes," in *Software Process Technology*, Heidelberg, Springer Berlin, 2003, pp. 16-31

[7] R. Fairley and M. J. Wilshire, "Iterative rework: The good, the bad and the ugly," *IEEE Computer,* vol. 38, no. 9, pp. 34-41, 2005.

[8] R. Dion, "Process Improvement and the Corporate Balance Sheet," *IEEE Software,* vol. 10, no. 4, pp. 28-35, 1993.

[9] B. Boehm and C. Papaccio, "Understanding and Controlling Software Costs," *IEEE Transactions of Software Engineering,* vol. 14, no. 10, pp. 1462 - 1477, 1988.

[10] K. Butler and W. Lipke, "Software process achievement at Tinker Air Force Base," Carnegie-Mellon Software Engineering Institute, 2000.

[11] V. Plekhanova, "On Project Management Scheduling where Human Resource is a Critical Variable," in *Proceedings of the 6th European Workshop on Software Process Technology (EWSPT-6)*, London, 1998.

[12] Project Management Institute, Practice standard for Earned Value Management, Newtown Square, PA, USA: Project Management Institute, 2005.

[13] Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK® Guide), 4th ed., Newtown Square, PA, USA: Project Management Institute, 2008.

[14] H. Erdogmus, "Tracking Progress through Earned Value," *IEEE Software,* vol. 27, no. 5, pp. 2-7, 2010.

[15] J. Twentyman, "The crippling costs of IT project rework," *Inside Knowledge,* 15 Jun 2005.

[16] R. N. Charette, "Why Software Fails," *IEEE Spectrum,* vol. 42, no. 9, pp. 42-49, 2005.

[17] C. Ebert and R. Dumke, Software Measurement: Establish - Extract -Evaluate - Execute, Berlin: Springer, 2010.

[18] C. Y. Laporte, N. Berrhouma, M. Doucet and E. Palza-Vargas, "Measuring the Cost of Software Quality of a Large Software Project at Bombardier Transportation," *Software Quality Professional Journal: American Society for Quality,* vol. 14, no. 3, pp. 14-31, 2012.

[19] Q. W. Koppelman and J. M. Flemimg, Earned Value Management, 3rd ed., Newtown Square, PA, USA: Project Management Institute, 2005.

[20] F. Anbari, "Earned Value Project Management Method and Extensions," *Project Management Journal,* vol. 34, no. 4, pp. 12-13, 2003.

[21] *"ANSI/EIA -748A: American National Standard Institute / Electronic Industries Alliance/ Standard for Earned Value Management Systems",* 1998.

[22] P. Solanki, Earned Value Management: Integrated View of Cost and Schedule Performance, New Delhi: Global India Publications Pvt Ltd., 2011.

[23] Project Management Institute, Practice standard for Earned Value Management, Newtown Square, PA, USA: Project Management Institute, 2011.

[24] Q. W. Koppelman and J. M. Fleming, "Earned Value Project Management: A Powerful Tool for Software Projects," *STSC CrossTalk: The Journal of Defense Software Engineering,* no. 4, pp. 19-23, 1998.

[25] *"AS 4817-2003: Project performance measurement using Earned Value",* 2003.

[26] *“AS 4817-2006: Project performance measurement using Earned Value”,* 2006.

[27] R. Stratton, The Earned Value Management Maturity Model, Vienna, VA: Management Concepts, 2006.

[28] W. Lipke, " Schedule is Different," *The Measurable News,* pp. 10-15, March 2003.

[29] W. Lipke, O. Zwikael, K. Henderson and F. Anbari, "Prediction of Project Outcome: The application of Statistical Methods to Earned Value Management and Earned Schedule Performance Indexes," *International Journal of Project Management,* vol. 4, no. 27, pp. 400-407, 2009.

[30] W. Lipke, "Earned Schedule Contribution to Project Management," *PM World Journal,* vol. I, no. II, pp. 5-19, 2012.

[31] W. Lipke, "An Emerging Enhancement to Earned Value Management," *CrossTalk,* pp. 26-30, November 2006.

[32] K. Henderson, "Earned Schedule: A Breakthrough Extension to Earned Value Management," in *Proceedings of PMI Global Congress Asia Pacific*, Hong Kong, 2007.

[33] D. Brandon, "Implementing Earned Value Easily and Effectively," *Project Management Journal,* vol. 29, no. 2, pp. 11-18, 1998.

[34] J. A. Lukas, "Earned Value Analysis - Why it does not work," in *AACE International Transactions*, 2008.

[35] E. Kim, G. W. Wells and M. R. Duffey, "A model for effective implementation of Earned Value Management methodology," *International Journal of Project Management,* vol. 21, no. 5, pp. 375-382, 2003.

[36] M. Ferle, "Implementing Earned Value Management on IT Projects," in *19th International Cost Engineering Congress*, Ljubljana, Slovenia, 2006.

[37] R. A. Hanna, "Earned Value Management Software Projects," in *Proceedings of the 3rd IEEE International Conference on Space Mission Challenges for Information Technology*, Washington, DC, 2009.

[38] P. J. Solomon, "Integrating Systems Engineering with Earned Value Management," *Defense AT&L,* vol. 33, pp. 42-26, 2004.

[39] D. S. Christensen and D. V. Ferens, "Using earned value on software development projects," *Acquisition Review Quarterly,* no. 2, p. 155–171, 1995.

[40] D. Brandon, "Integrating Earned Value into the Management of Software Development Projects," in *Proceedings of the IRMA International Conference*, Hersey, PA, USA, 1999.

[41] S. Wu, "Su-Cheng Wu," 20 June 2011. [Online]. Available: http://www.pmroi.com/storage/1_WuS_TEVMAEVMProcess_20110620_pmroi_web.pdf. [Accessed 01 March 2015].

[42] J. Rusk, "Earned Value for Agile Development," *DoD Software Tech News,* vol. 12, no. 1, 2009.

[43] A. Cabri and M. Griffiths, "Earned Value and Agile Reporting," in *AGILE '06: Proceedings of the conference on AGILE 2006*, Washington, DC, 2006.

[44] J. I. Hernández, J. R. O. Olaso and J. R. Gomez, "Technical Performance Based Earned Value as a Management Tool for Engineering Projects," in *Engineering Management*, InTech, 2013, pp. 143-166.

[45] S. H. Lett, "An Earned Value Tracking System for Self-Directed Software Teams," in *Proceedings of European SEPG*, London, 1998.

[46] A. Cockburn, Crystal Clear: A Human-Powered Methodology for Small Teams, Upper Saddle River, NJ, USA: Pearson Education Inc, 2005.

[47] G. B. Alleman, "Project Management = Herding Cats, A Field Report, Agile Project Management," PMFORUM, 2003.

[48] G. B. Alleman, M. Henderson and R. Seggelke, "Making Agile Development Work in a Government Contracting Environment. Measuring velocity with Earned Value," in *Proceedings of the Agile Development Conference*, Salt Lake City, Utah, 2003.

[49] T. Sulaiman, B. Barton and T. Blackburn, "AgileEVM – Earned Value Management in Scrum Projects," in *Proceedings of AGILE 2006 Conference*, Minneapolis, Minnesota, USA, 2006.

[50] P. J. Solomon, "Practical Software Measurement, Performance-Based Earned Value," *CrossTalk,* no. September, 2001.

[51] P. J. Solomon, "Performance-Based Earned Value," *CrossTalk,* no. August, pp. 22-26, 2005.

[52] P. J. Solomon, "Practical Performance-Based Earned Value," *Crosstalk,* no. May, pp. 20-24, 2006.

[53] P. J. Solomon and R. Young, Performance-Based Earned Value, Hoboken, NJ: Wiley & Sons, 2007.

[54] P. Solomon, "Basing Earned Value on Technical Performance," *CrossTalk,* Vols. 25-28, pp. 25-28, January/February 2013.

[55] L. P. Leach, Critical Chain Project Management, Boston: Artech House, 2004.

[56] Realization Technologies, Inc, "http://www.realization.com/," 2012. [Online]. Available: http://www.realization.com/pdf/whitepapers/Integrating_Earned_Value_Management_ with_CC_Execution.pdf. [Accessed 08 03 2015].

[57] C. Baker, "Association for Project Management," 8 November 2013. [Online]. Available: https://www.apm.org.uk/news/earned-value-and-critical-chain-project-management-ccpm-method. [Accessed 10 01 2015].

[58] G. Silber and A. Hussey, " Using earned value management to predict buffer penetration in critical chain project management," in *PMI Seminars and Symposium Proceedings*, 2002.

[59] H. Raju and Y. Krishnegowda, "Software Sizing and Productivity with Function Points," *Lecture Notes on Software Engineering,* vol. 1, no. 2, pp. 204-208, 2013.

[60] I. Brown and B. A. Hamilton, "Earned Value Management: Integrated View of Cost and Schedule Performance," in *Better Software Conference & EXPO*, San Jose, CA, 2004.

[61] W. Roetzheim, "Incorporating Function Points Into Earned Value Management," in *23rd Systems and Software Technology Conference (SSTC),* Salt Lake City, 2011

[62] J. Li, Z. Ma and H. Dong, "Monitoring Software Projects with Earned Value Analysis and Use Case Point," in *IEEE/ACIS International Conference*, Portland, 2008.

[63] C. A. L. Garcia and C. M. Hirata, "Integrating functional metrics, COCOMO II and earned value analysis for software projects using PMBoK," in *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC)*, Fortaleza, Ceara, Brazil, 2008

[64] G. Liu, "Tracking Software Development Progress with Earned Value and Use Case Point," in *Proceedings of the International Workshop on Information Security and Application*, Qingdao, China, 2009.

[65] J. M. Juran, Quality Control Handbook, 1st ed., New York: McGraw-Hill, 1951.

[66] J. G. Suarez, Three Experts on Quality Management, Arlington, VA: TQL Office, 1992.

[67] A. V. Feigenbaum, "Total Quality Control," *Harvard Business Review,* vol. 34, no. 6, pp. 93-101, 1956.

[68] P. B. Crosby, Quality is Free: The Art of Making Quality Certain, Mentor , 1980.

[69] M. Lidia, P. Lucian, C. C. Georgeta, P. Aurelian and B. Ovidiu, "Paradigms of Total Quality Management," in *WSEAS International ISI Conference on Manufacturing Engineering, Quality and Production Systems*, Braşov, Romania, 2011.

[70] S. T. Knox, "Modeling the cost of software quality," *Digital Technical Journal,* vol. V, no. 4, p. 9–17, 1993.

[71] A. Schiffauerova and V. Thomson, " A review of research on cost of quality models and best practices," *International Journal of Quality & Reliability Management,* vol. 23, no. 4, p. 647–669, 2006.

[72] H. Krasner and D. Houston, "Using the cost of quality approach for software," *STSC CrossTalk:The Journal of Defense Software Engineering,* no. November, pp. 1-11, 1998.

[73] S. P. Vedula, "Benchmarking Software Quality With Applied Cost of Quality," 01 09 2012. [Online]. Available: www.tcs.com/SiteCollectionDocuments/White %20Papers/Benchmarking_software_Quality.pdf. [Accessed 01 02 2015].

[74] L. M. Karg, M. Grottke and A. Beckhaus, "A systematic literature review of software quality cost research," *Journal of Systems and Software ,* vol. 84, no. 3, pp. 415-427, 2011.

[75] T. J. Haley, "Software Process Improvement at Raytheon," *IEEE Software,* vol. 13, no. 6, pp. 33-41, 1996.

[76] O. Demirors, Ö. Yildiz and A. S. Güceglioglu, "Using Cost of Software Quality for a Process Improvement Initiative," in *Proceedings of the 26th Euromicro Conference*, Maastricht, The Netherlands, 2000.

[77] J. Creswell, Research design: Qualitative, quantitative and mixed methods approaches, 2nd ed., Thousand Oaks, CA: SAGE Publications, 2003.

[78] C. Williams, "Research Methods," *Journal of Business & Economic Research ,* vol. 5, no. 3, pp. 65-72, 2007 .

[79] J. Creswell, Educational research: Planning, conducting, and evaluating quantitative and qualitative, Upper Saddle River, NJ: Merrill Prentice Hall, 2002.

[80] M. Myers, "Qualitative research in information systems," *MIS Quarterly,* vol. 21, no. 2, pp. 241-242, 1997.

[81] J. W. Creswell, Research design: Qualitative, quantitative, and mixed methods approaches, Sage Publications, Inc, 2009.

[82] W. J. Orlikowski and J. J. Baroudi, "Studying Information Technology in Organizations: Research Approaches and Assumptions," *Information Systems Research (2),* pp. 1-28, 1991.

[83] R. K. Yin, Case study research: Design and methods, 5th ed., Thousand Oaks, CA: SAGE Publications, 2014.

[84] I. Benbasat , D. Goldstein and M. Mead, "The case research strategy in studies of information systems," *MIS Quarterly,* pp. 369-386, 1987.

[85] E. Babbie, The practice of social research, Belmont, CA: Wadsworth, 1989.

[86] M. Q. Patton, Qualitative research and evaluation methods, 3rd ed., Thousand Oaks, CA: SAGE Publications, 2002.

[87] J. Fitzpatrick, J. R. Sanders and B. Worthen, Program evaluation: Alternative approaches and practical guidelines, Boston: Pearson Education, 2004.

[88] T. Foss, E. Stensrud, B. Kitchenham and I. Myrtveit, "A Simulation Study of the Model Evaluation Criterion MMREF," *IEEE Transactions on Software Engineering,* vol. 29, no. 11, pp. 985-995 , 2003.

[89] B. Kitchenham, L. Pickard and M. Sheppard, "What accuracy statistics really measure," *IEE Proceedings Software,* vol. 148, no. 3, pp. 81-85, 2001.

[90] D. Port and M. Korte, "Comparative Studies of the Model Evaluation Criterions MMRE and PRED in Software Cost Estimation Research," in *Proceedings of the Second International Symposium on Empirical Software Engineering and Measurement*, Kaiserslautern, Germany, 2008.

[91] S. D. Conte, H. E. Dunsmore and V. Y. Shen, Software engineering metrics and models, Redwood City, CA, USA: Benjamin-Cummings Publishing Co, 1986.

[92] M. Gibbert, W. Ruigrok and B. Wicki, "What passes as a rigorous case study?," *Strategic Management Journal,* vol. 29, no. 13, pp. 1465-1474, 2008.

[93] J. Maxwell, Qualitative research design: An interactive approach, Sage Publications Inc, 2004.

[94] R. Feldt and A. Magazinius, "Validity Threats in Empirical Software Engineering Research - An Initial Survey," in *Proceedings of the 22nd International Conference on Software Engineering and Knowledge Engineering* , 2010.

[95] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell and A. Wesslén, Experimentation in Software Engineering: An Introduction, Norwell, MA,: Kluwer Academic Publishers, 2000.

[96] P. Efe and O. Demirors, "Applying EVM in a Software Company: Benefits and Difficulties," in *Proceedings of the 39th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)*, Santander, 2013.

APPENDIX A

A.1. Case I

Table 43 Case I Project Data for New Features

Time	PV	AC	EV	CV	CPI	SV	SPI
May	928	872	882	10	1.01	-46	0.95
Jun	1360	1349	1314	-35	0.97	-46	0.97
Jul	1852	1792	1764	-28	0.98	-88	0.95
Aug	2465	2395	2354	-41	0.98	-111	0.95
Sep	3152	3071	3014	-57	0.98	-138	0.96
Oct	3617	3579	3454	-125	0.97	-163	0.95
Nov	3997	3974	3814	-160	0.96	-183	0.95
Dec	4437	4434	4254	-180	0.96	-183	0.96
Jan	5109	5364	4914	-450	0.92	-195	0.96
Feb	5698	6010	5503	-507	0.92	-195	0.97
Mar	6341	6665	6113	-552	0.92	-228	0.96

Table 44 Case I Project Data including both New Features and Rework

Time	PV	AC	Rework	AC (incl. rework)	EV	CV	CPI	SV	SPI
May	928	872	123	995	882	-113	0.89	-46	0.95
Jun	1360	1349	318	1667	1314	-353	0.79	-46	0.97
Jul	1852	1792	384	2176	1764	-412	0.81	-88	0.95
Aug	2465	2395	483	2878	2354	-524	0.82	-111	0.95
Sep	3152	3071	657	3728	3014	-714	0.81	-138	0.96
Oct	3617	3579	753	4332	3454	-878	0.80	-163	0.95
Nov	3997	3974	879	4853	3814	-1039	0.79	-183	0.95
Dec	4437	4434	969	5403	4254	-1149	0.79	-183	0.96
Jan	5109	5364	1134	6498	4914	-1584	0.76	-195	0.96
Feb	5698	6010	1261	7271	5503	-1768	0.76	-195	0.97
Mar	6341	6665	1444	8109	6113	-1996	0.75	-228	0.96

A.2. Case II

Table 45 Case II Project Data for Development Phase

Time	PV	AC	EV	CV	CPI	SV	SPI
Jan	53	55	43	-12	0.78	-10	0.81
Feb	180	163	138	-25	0.85	-42	0.77
Mar	300	265	223	-42	0.84	-77	0.74
Apr	408	416	368	-48	0.88	-40	0.90
May	408	470	408	-62	0.87	0	1.00

Table 46 Case II Project Data for Development and Test Phase

Time	PV	AC	Rework	AC (incl. Rework)	EV	CV	CPI	SV	SPI
Jan	53	55	0	55	43	-12	0.78	-10	0.81
Feb	180	163	0	163	138	-25	0.85	-42	0.77
Mar	300	265	0	265	223	-42	0.84	-77	0.74
Apr	408	416	0	416	368	-48	0.88	-40	0.90
May	408	470	0	470	408	-62	0.87	0	1.00
June	450	556	86	556	450	-106	0.81	0	1.00
July	490	606	50	606	490	-116	0.81	0	1.00

A.3. Case III

Table 47 Case III EVM Application – Performance Analysis and Forecast Metrics

Iteration		PV	AC	EV	CV	CPI	SV	SPI	EAC	VAC	ETC
May	**1**	904	963	884	-79	0.92	-20	0.98	7876	-646	6913
Jun	**2**	1660	1734	1600	-134	0.92	-60	0.96	7836	-606	6102
Jul	**3**	2527	2513	2307	-206	0.92	-220	0.91	7876	-646	5363
Aug	**4**	3194	3332	3089	-243	0.93	-105	0.97	7799	-569	4467
Sep	**5**	4034	4176	3874	-302	0.93	-160	0.96	7794	-564	3618
Oct	**6**	4981	5198	4783	-415	0.92	-198	0.96	7857	-627	2659
Nov	**7**	5915	6253	5676	-577	0.91	-239	0.96	7965	-735	1712
Dec	**8**	6600	6957	6327	-630	0.91	-273	0.96	7950	-720	993
Jan	**9**	7230	7600	6877	-723	0.90	-353	0.95	7990	-760	390
later			7966								

Table 48 Case III Failure Cost Data for Iterations

Iteration		AC	FC
May	**1**	963	0
Jun	**2**	771	106
Jul	**3**	779	114
Aug	**4**	819	121
Sep	**5**	844	103
Oct	**6**	1022	150
Nov	**7**	1055	110
Dec	**8**	704	144
Jan	**9**	643	131
later		183	101

Table 49 Case III Cumulative Failure Cost Data

Iteration		AC	FC
May	**1**	963	0
Jun	**2**	1734	106
Jul	**3**	2513	220
Aug	**4**	3332	341
Sep	**5**	4176	444
Oct	**6**	5198	594
Nov	**7**	6253	704
Dec	**8**	6957	848
Jan	**9**	7600	979
later		7966	1080

Table 50 Case III Retroactive EV Matrix

Iteration		EV	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	later
May	**1**	884	884	801	793	789						
Jun	**2**	716	-	716	662	655	651					
Jul	**3**	707	-	-	707	661	661	650				
Aug	**4**	782	-	-	-	782	720	703				
Sep	**5**	785	-	-	-	-	785	733	721	711	693	
Oct	**6**	909	-	-	-	-	-	909	877	837	828	820
Nov	**7**	893	-	-	-	-	-	-	893	828	808	
Dec	**8**	651	-	-	-	-	-	-	-	651	625	585
Jan	**9**	550	-	-	-	-	-	-	-	-	550	498

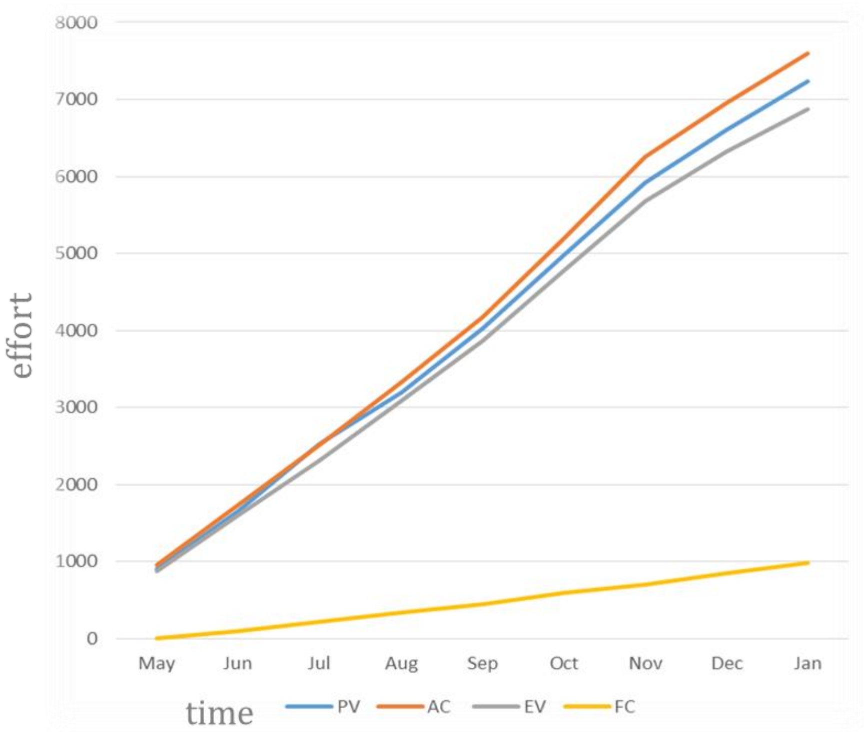

Figure 26 Case III Failure Cost Graph

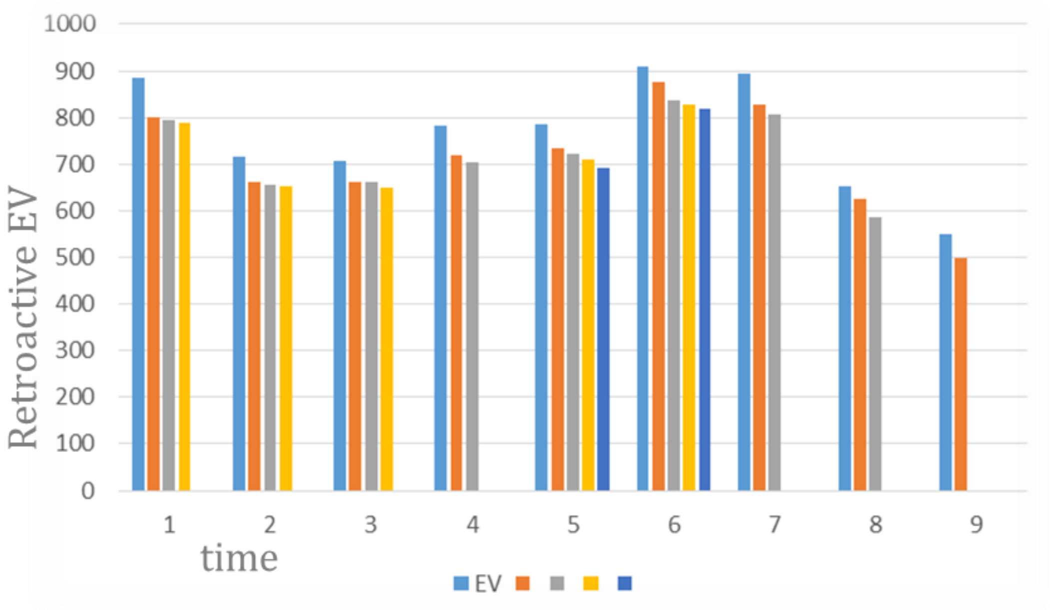

Figure 27 Case III EV_r change for the phases

Table 51 Case III qcf of the Phases in Time

Iteration		May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Later
May	**1**	0	0.09	0.10	0.11						
Jun	**2**	-	0	0.08	0.09	0.09					
Jul	**3**	-	-	0	0.07	0.07	0.08				
Aug	**4**	-	-	-	0	0.08	0.10				
Sep	**5**	-	-	-	-	0	0.07	0.08	0.09	0.12	
Oct	**6**	-	-	-	-	-	0	0.04	0.08	0.09	0.10
Nov	**7**	-	-	-	-	-	-	0	0.07	0.10	
Dec	**8**	-	-	-	-	-	-	-	0	0.04	0.10
Jan	**9**	-	-	-	-	-	-	-	-	0	0.09

Table 52 Case III QEVM Application-Performance Analysis and Forecast Metrics

Iteration		PV	ACF	EV_{est}	CV_{est}	CPI_{est}	SV_{est}	SPI_{est}	QPI	EAC	ETC	VAC	EFC	TCFC
May	**1**	904	963	884	-79	0.92	-20	0.98	1.00	7876	6913	-646	0	0
Jun	**2**	1660	1840	1536	-304	0.83	-124	0.93	0.94	8663	6823	-1433	618	512
Jul	**3**	2527	2733	2243	-490	0.82	-284	0.89	0.92	8808	6075	-1578	669	449
Aug	**4**	3194	3673	3019	-654	0.82	-175	0.95	0.91	8797	5124	-1567	720	379
Sep	**5**	4034	4620	3811	-809	0.82	-223	0.94	0.90	8764	4144	-1534	718	274
Oct	**6**	4981	5792	4701	-1091	0.81	-280	0.94	0.90	8908	3116	-1678	822	228
Nov	**7**	5915	6957	5605	-1352	0.81	-310	0.95	0.90	8975	2018	-1745	834	130
Dec	**8**	6600	7805	6268	-1537	0.80	-332	0.95	0.89	9002	1197	-1772	935	87
Jan	**9**	7230	8579	6828	-1752	0.80	-403	0.94	0.89	9085	506	-1855	1015	36
Later			9046						0.88					

Table 53 Case III QEVM Conformance Cost Data

Iteration		Conformance Costs			Planned Value	Conformance Cost Indices		
		Prevention Costs	Appraisal Costs	**Conformance Costs**		Prevention Cost Index	Appraisal Cost Index	**Conformance Cost Index**
May	**1**	10	120	130	904	0.01	0.13	0.14
Jun	**2**	20	240	260	1660	0.01	0.14	0.16
Jul	**3**	30	360	390	2527	0.01	0.14	0.15
Aug	**4**	40	480	520	3194	0.01	0.15	0.16
Sep	**5**	50	600	650	4034	0.01	0.15	0.16
Oct	**6**	60	720	780	4981	0.01	0.14	0.16
Nov	**7**	70	840	910	5915	0.01	0.14	0.15
Dec	**8**	80	960	1040	6600	0.01	0.15	0.16
Jan	**9**	90	1080	1170	7230	0.01	0.15	0.16

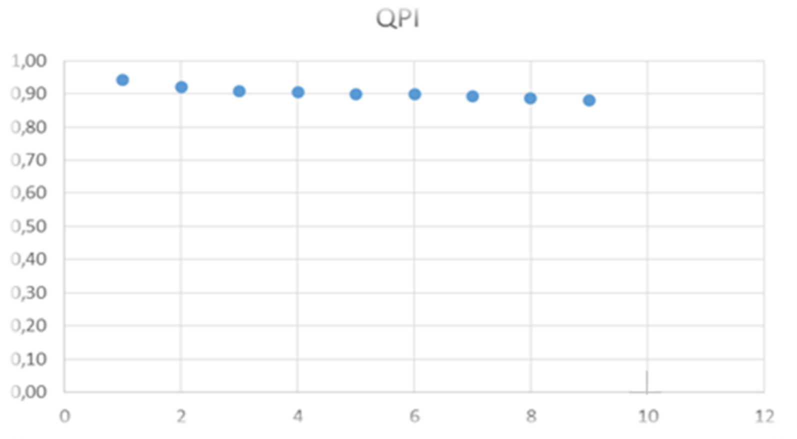

Figure 28 Case III QPI Graph

Table 54 Case III Comparison of ACs

Iteration		**EVM**	**QEVM**
		AC	**ACF**
May	**1**	963	963
Jun	**2**	1734	1840
Jul	**3**	2513	2733
Aug	**4**	3332	3673
Sep	**5**	4176	4620
Oct	**6**	5198	5792
Nov	**7**	6253	6957
Dec	**8**	6957	7805
Jan	**9**	7600	8579
later		7966	9046

Table 55 Case III Comparison of EVs

Iteration		**EVM**	**QEVM**
		EV	EV_{est}
May	**1**	884	884
Jun	**2**	1600	1536
Jul	**3**	2307	2243
Aug	**4**	3089	3019
Sep	**5**	3874	3811
Oct	**6**	4783	4701
Nov	**7**	5676	5605
Dec	**8**	6327	6268
Jan	**9**	6877	6828

Table 56 Case III Comparison of Performance Indices

Iteration		EVM SPI	QEVM SPI_{est}	EVM CPI	QEVM CPI_{est}
May	1	0.98	0.98	0.92	0.92
Jun	2	0.96	0.93	0.92	0.83
Jul	3	0.91	0.89	0.92	0.82
Aug	4	0.97	0.95	0.93	0.82
Sep	5	0.96	0.94	0.93	0.82
Oct	6	0.96	0.94	0.92	0.81
Nov	7	0.96	0.95	0.91	0.81
Dec	8	0.96	0.95	0.91	0.80
Jan	9	0.95	0.94	0.90	0.80

Table 57 Case III Comparison of Forecasts

Iteration		EVM EAC	QEVM EAC	EVM VAC	QEVM VAC	EVM ETC	QEVM ETC	QEVM EFC	TCFC
May	1	7876	7876	-646	-646	6913	6913	0	0
Jun	2	7836	8663	-606	-1433	6102	6823	618	512
Jul	3	7876	8808	-646	-1578	5363	6075	669	449
Aug	4	7799	8797	-569	-1567	4467	5124	720	379
Sep	5	7794	8764	-564	-1534	3618	4144	718	274
Oct	6	7857	8908	-627	-1678	2659	3116	822	228
Nov	7	7965	8975	-735	-1745	1712	2018	834	130
Dec	8	7950	9002	-720	-1772	993	1197	935	87
Jan	9	7990	9085	-760	-1855	390	506	1015	36

Table 58 Case III Comparison of MREs for EAC

Iteration		EVM MRE	QEVM MRE
May	1	0.13	0.13
Jun	2	0.13	0.04
Jul	3	0.13	0.03
Aug	4	0.14	0.03
Sep	5	0.14	0.03
Oct	6	0.13	0.02
Nov	7	0.12	0.01
Dec	8	0.12	0.005
Jan	9	0.12	-0.004
MMRE		0.13	0.02

A.4. Case IV

Table 59 Case IV EVM Application – Performance Analysis and Forecast Metrics

Iteration		PV	AC	EV	CV	CPI	SV	SPI	EAC	VAC	ETC
Feb	**1**	101	96	86	-10	0.90	-15	0.85	4646	-484	4550
Mar	**2**	750	831	752	-79	0.90	2	1.00	4599	-437	3768
Apr	**3**	1619	1712	1560	-152	0.91	-59	0.96	4568	-406	2856
May	**4**	2207	2392	2185	-207	0.91	-22	0.99	4556	-394	2164
Jun	**5**	2558	2814	2537	-277	0.90	-21	0.99	4616	-454	1802
Jul	**6**	2970	3179	2878	-301	0.91	-92	0.97	4597	-435	1418
Aug	**7**	3203	3480	3160	-320	0.91	-43	0.99	4583	-421	1103
Sep	**8**	3460	3716	3378	-338	0.91	-82	0.98	4578	-416	862
Oct	**9**	3703	4036	3667	-369	0.91	-36	0.99	4581	-419	545
Nov	**10**	4162	4553	4151	-402	0.91	-11	1.00	4565	-403	12

Table 60 Case IV Failure Cost Data

Iteration		AC	IFC	EFC
Feb	**1**	96	10	0
Mar	**2**	831	79	25
Apr	**3**	1712	152	93
May	**4**	2392	207	341
Jun	**5**	2814	277	418
Jul	**6**	3179	301	491
Aug	**7**	3480	320	550
Sep	**8**	3716	338	579
Oct	**9**	4036	369	639
Nov	**10**	4553	402	758
later				955

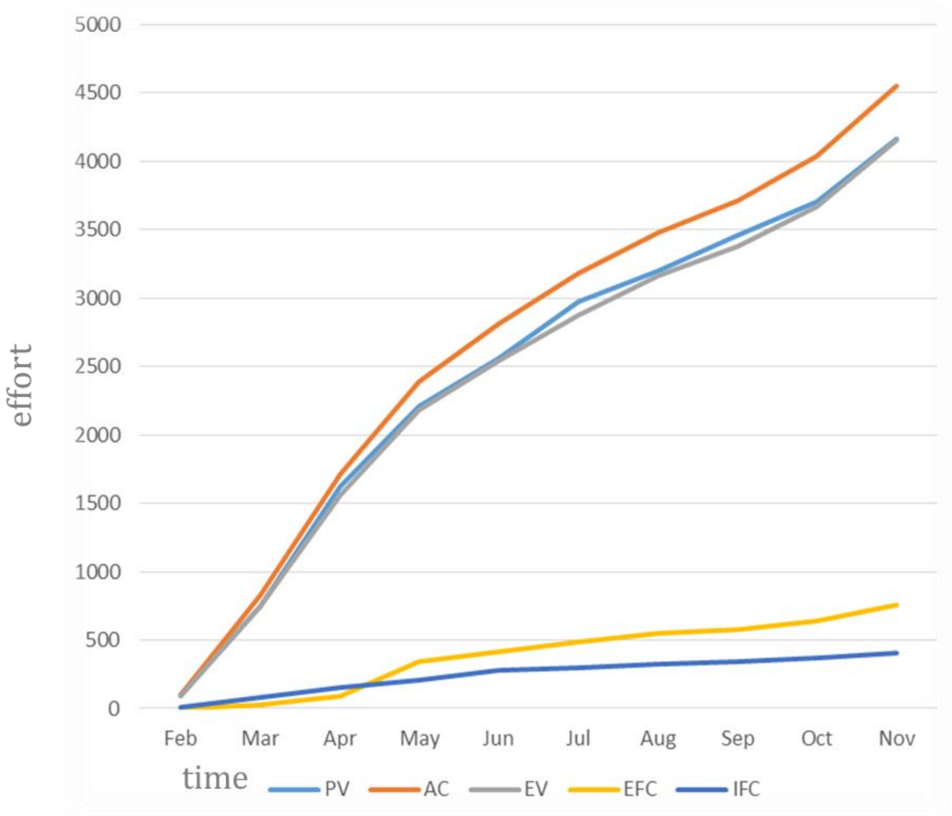

Figure 29 Case IV Failure Cost Graph

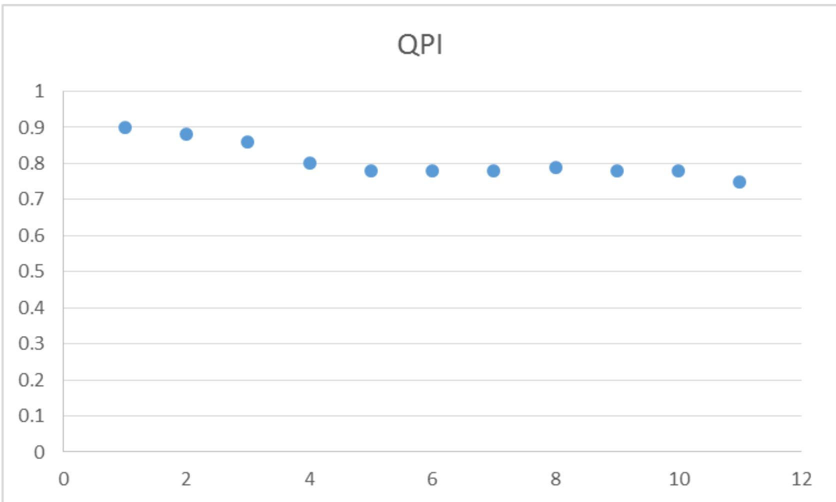

Figure 30 Case IV QPI Graph

Table 61 Case IV QEVM Application – Performance Analysis and Forecast Metrics

Iteration		PV	ACF	EV_{est}	CV_{est}	CPI_{est}	SV_{est}	SPI_{est}	QPI	EAC	ETC	VAC	EFC	TCFC
Feb	**1**	101	96	86	-10	0.90	-15	0.85	0.9	4646	4550	-484	0	0
Mar	**2**	750	856	614	-242	0.72	-136	0.82	0.88	5802	4946	-1640	949	924
Apr	**3**	1619	1805	1479	-326	0.82	-140	0.91	0.86	5079	3274	-917	393	300
May	**4**	2207	2733	2081	-652	0.76	-126	0.94	0.8	5466	2733	-1304	755	414
Jun	**5**	2558	3232	2485	-747	0.77	-73	0.97	0.78	5413	2181	-1251	711	293
Jul	**6**	2970	3670	2827	-843	0.77	-143	0.95	0.78	5403	1733	-1241	724	233
Aug	**7**	3203	4030	3118	-912	0.77	-85	0.97	0.78	5379	1349	-1217	731	181
Sep	**8**	3460	4295	3347	-948	0.78	-113	0.97	0.79	5341	1046	-1179	715	136
Oct	**9**	3703	4675	3625	-1050	0.78	-78	0.98	0.78	5368	693	-1206	731	92
Nov	**10**	4162	5311	4074	-1237	0.77	-88	0.98	0.78	5426	115	-1264	775	17
later			5508						0.75					

Table 62 Case IV QEVM Conformance Cost Data

Iteration		Conformance Costs			Planned Value	Conformance Cost Indices		
		Prevention Costs	Appraisal Costs	Conformance Costs		Prevention Cost Index	Appraisal Cost Index	Conformance Cost Index
Feb	**1**	5	40	45	101	0.05	0.40	0.45
Mar	**2**	10	160	170	750	0.01	0.21	0.23
Apr	**3**	15	360	375	1619	0.01	0.22	0.23
May	**4**	20	520	540	2207	0.01	0.24	0.24
Jun	**5**	25	680	705	2558	0.01	0.27	0.28
Jul	**6**	30	700	730	2970	0.01	0.24	0.25
Aug	**7**	35	820	855	3203	0.01	0.26	0.27
Sep	**8**	40	900	940	3460	0.01	0.26	0.27
Oct	**9**	45	980	1025	3703	0.01	0.26	0.28
Nov	**10**	50	1100	1150	4162	0.01	0.26	0.28

Table 63 Case IV Comparison of ACs

Iteration		EVM	QEVM
		AC	ACF
Feb	1	96	96
Mar	2	831	856
Apr	3	1712	1805
May	4	2392	2733
Jun	5	2814	3232
Jul	6	3179	3670
Aug	7	3480	4030
Sep	8	3716	4295
Oct	9	4036	4675
Nov	10	4553	5311
later			5508

Table 64 Case IV Comparison of EVs

Iteration		EVM	QEVM	
		EV	EV_{est}	EV_{init}
Feb	1	86	86	86
Mar	2	752	614	556
Apr	3	1560	1479	1385
May	4	2185	2081	1750
Jun	5	2537	2485	2094
Jul	6	2878	2827	2376
Aug	7	3160	3118	2613
Sep	8	3378	3347	2816
Oct	9	3667	3625	3036
Nov	10	4151	4074	3371

Table 65 Case IV Comparison of Performance Indices

Iteration		EVM	QEVM	EVM	QEVM
		SPI	SPI_{est}	CPI	CPI_{est}
Feb	**1**	0.85	0.85	0.90	0.90
Mar	**2**	1.00	0.82	0.90	0.72
Apr	**3**	0.96	0.91	0.91	0.82
May	**4**	0.99	0.94	0.91	0.76
Jun	**5**	0.99	0.97	0.90	0.77
Jul	**6**	0.97	0.95	0.91	0.77
Aug	**7**	0.99	0.97	0.91	0.77
Sep	**8**	0.98	0.97	0.91	0.78
Oct	**9**	0.99	0.98	0.91	0.78
Nov	**10**	1.00	0.98	0.91	0.77

Table 66 Case IV Comparison of Forecasts

Iteration		EVM	QEVM	EVM	QEVM	EVM	QEVM	QEVM	
		EAC	EAC	VAC	VAC	ETC	ETC	EFC	TCFC
Feb	**1**	4646	4646	-484	-484	4550	4550	0	0
Mar	**2**	4599	5802	-437	-1640	3768	4946	949	924
Apr	**3**	4568	5079	-406	-917	2856	3274	393	300
May	**4**	4556	5466	-394	-1304	2164	2733	755	414
Jun	**5**	4616	5413	-454	-1251	1802	2181	711	293
Jul	**6**	4597	5403	-435	-1241	1418	1733	724	233
Aug	**7**	4583	5379	-421	-1217	1103	1349	731	181
Sep	**8**	4578	5341	-416	-1179	862	1046	715	136
Oct	**9**	4581	5368	-419	-1206	545	693	731	92
Nov	**10**	4565	5426	-403	-1264	12	115	775	17

Table 67 Case IV Comparison of MREs for EAC

Iteration		EVM	QEVM
		MRE	MRE
Feb	1	0.16	0.16
Mar	2	0.16	0.05
Apr	3	0.17	0.08
May	4	0.17	0.01
Jun	5	0.16	0.02
Jul	6	0.17	0.02
Aug	7	0.17	0.02
Sep	8	0.17	0.03
Oct	9	0.17	0.02
Nov	10	0.17	0.01
MMRE		0.17	0.03

A.5. Case V

Table 68 Case V EVM Application Performance Analysis and Forecast Metrics

Iteration		PV	AC	EV	CV	CPI	SV	SPI	EAC	VAC	ETC
Dec	1	1304	1280	1240	-40	0.97	-64	0.95	6177	-193	4897
Jan	2	2544	2760	2464	-296	0.89	-80	0.97	6703	-719	3943
Feb	3	3824	4240	3624	-616	0.85	-200	0.95	7001	-1017	2761
Mar	4	4984	5480	4824	-656	0.88	-160	0.97	6798	-814	1318
Apr	5	5984	6556	5824	-732	0.89	-160	0.97	6736	-752	180
May	6 (late)	5984	6936	5904	-1032	0.85	-80	0.99	7030	-1046	94
June	7 (late)	5984	7016	5984	-1032	0.85	0	1.00			

Table 69 Case V Failure Cost Data for Iterations

Iteration		AC	FC
Dec	1	1280	0
Jan	2	1480	296
Feb	3	1480	376
Mar	4	1240	432
Apr	5	1176	520
May	6 (late)	280	760
June	7 (late)	80	816

Table 70 Case V Cumulative Failure Cost Data

	Iteration	AC	FC
Dec	**1**	1280	0
Jan	**2**	2760	296
Feb	**3**	4240	696
Mar	**4**	5480	1128
Apr	**5**	6656	1648
May	**6 (late)**	6936	2408
June	**7 (late)**	7016	3224

Table 71 Case V ACF Calculations

ACF for Iterations	Dec	Jan	Feb	Mar	Apr	May	Jun
1	1280	1600					
2		3080	3456				
3			4936	5368			
4				6608	7128		
5					8204	8964	
6 (late)						9344	10160
7 (late)							10240

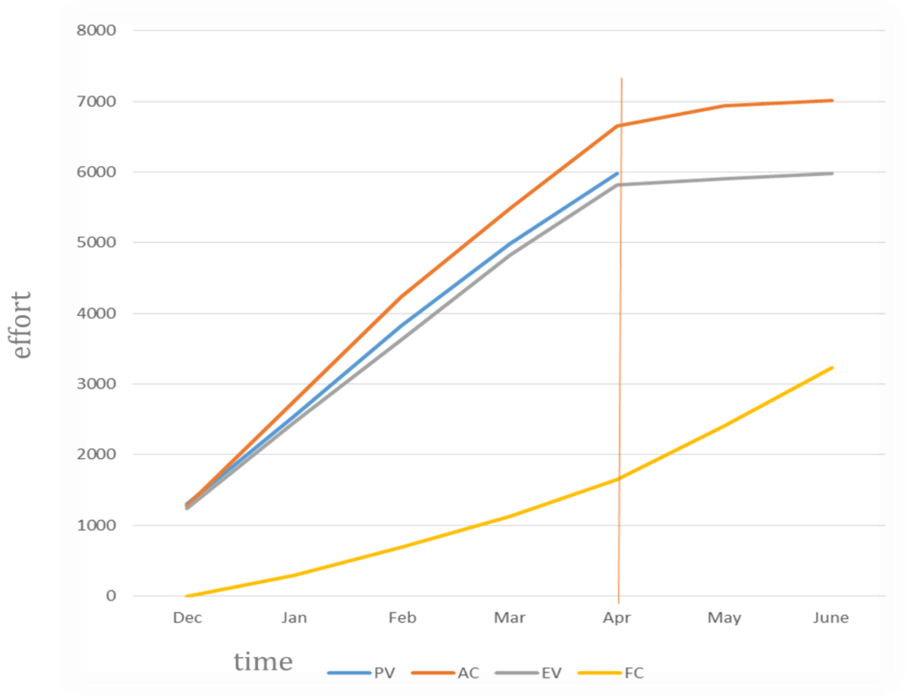

Figure 31 Case V Failure Cost Graph

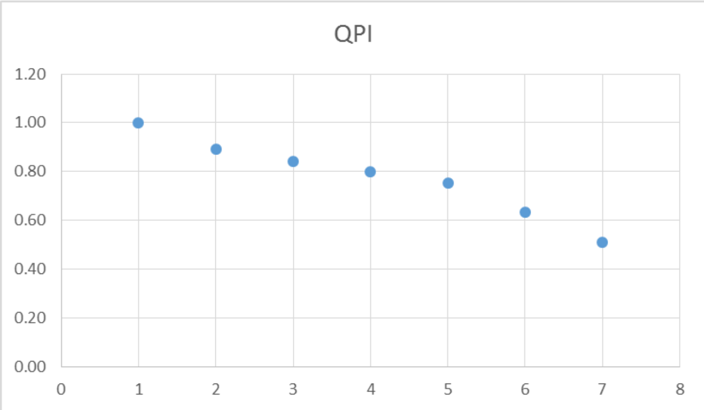

Figure 32 Case V QPI Graph

Table 72 Case V Comparison of ACs

Iteration		EVM	QEVM
		AC	ACF
Dec	**1**	1280	1280
Jan	**2**	2760	3056
Feb	**3**	4240	4936
Mar	**4**	5480	6608
Apr	**5**	6656	8204
May	**6 (late)**	6936	9344
Jun	**7 (late)**	7016	10240

Table 73 Case V QEVM Application Performance Analysis and Forecast Metrics

	PV	ACF	EV_{est}	CV_{est}	CPI_{est}	SV_{est}	SPI_{est}	QPI	EAC	ETC	VAC	EFC	TCFC
Dec	1304	1280	1240	-40	0.97	-64	0.95	1	6177	4897	-193	0	0
Jan	2544	3056	2234	-824	0.73	-310	0.88	0.89	8191	5133	-2207	1010	712
Feb	3824	4936	3390	-1546	0.69	-434	0.89	0.84	8712	3776	-2728	1241	545
Mar	4984	6608	4572	-2036	0.69	-412	0.92	0.79	8649	2041	-2665	1467	339
Apr	5984	8204	5593	-2611	0.68	-391	0.93	0.75	8778	574	-2794	1754	106
May	5984	9344	5883	-3461	0.63	-101	0.98	0.65	9505	161	-3521	2439	31
June	5984	10240	5904	-4336	0.58	-80	0.99	0.54	10379	139	-4395	3251	27

Table 74 Case V Comparison of EVs

Iteration		EVM	QEVM	
		EV	EV_{est}	EV_{init}
Dec	**1**	1240	1240	1240
Jan	**2**	2464	2234	2001
Feb	**3**	3624	3390	2894
Mar	**4**	4824	4572	3810
Apr	**5**	5774	5593	4477
May	**6 (late)**	5902	5883	4318
Jun	**7 (late)**	5984	5904	4085

Table 75 Case V Comparison of Performance Indices

Iteration		EVM	QEVM	EVM	QEVM
		SPI	SPI_{est}	CPI	CPI_{est}
Dec	**1**	0.95	0.95	0.97	0.97
Jan	**2**	0.97	0.88	0.89	0.73
Feb	**3**	0.95	0.89	0.85	0.69
Mar	**4**	0.97	0.92	0.88	0.69
Apr	**5**	0.97	0.93	0.89	0.68
May	**6 (late)**	0.99	0.98	0.85	0.63
Jun	**7 (late)**	1.00	0.99	0.85	0.58

Table 76 Case V Comparison of Forecasts

Iteration		EVM	QEVM	EVM	QEVM	EVM	QEVM	QEVM	
		EAC	EAC	VAC	VAC	ETC	ETC	EFC	TCFC
Dec	**1**	6177	6177	-193	-193	4897	4897	0	0
Jan	**2**	6703	8191	-719	-2207	3943	5133	1010	712
Feb	**3**	7001	8712	-1017	-2728	2761	3776	1241	545
Mar	**4**	6798	8649	-814	-2665	1318	2041	1467	339
Apr	**5**	6736	8778	-752	-2794	180	574	1754	106
May	**6**	7030	9505	-1046	-3521	94	161	2439	31

Table 77 Case V Comparison of MREs for EAC

Iteration		EVM	QEVM
		MRE	MRE
Dec	1	0.40	0.40
Jan	2	0.35	0.20
Feb	3	0.32	0.15
Mar	4	0.34	0.16
Apr	5	0.34	0.14
May	6	0.31	0.07
MMRE		0.34	0.14

Table 78 Case V QEVM Conformance Cost Data

Iteration		Conformance Costs			Planned Value	Conformance Cost Indices		
		Prevention Costs	Appraisal Costs	**Conformance Costs**		Prevention Cost Index	Appraisal Cost Index	**Conformance Cost Index**
Dec	1	80	80	160	1304	0.06	0.06	0.12
Jan	2	96	144	240	1240	0.08	0.12	0.19
Feb	3	96	160	256	1280	0.08	0.13	0.20
Mar	4	80	128	208	1160	0.07	0.11	0.18
Apr	5	64	152	216	1000	0.06	0.15	0.22

A.6. Case VI

Table 79 Case VI EVM Application Performance Analysis and Forecast Metrics

Phase		PV	AC	EV	CV	CPI	SV	SPI	EAC	VAC	ETC
Initiation	1	122	122	122	0	1.00	0	1.00	1225	0	1103
Analysis & Pr. Design	2	295	317	288	-29	0.91	-7	0.98	1348	-123	1031
Design	3	386	416	373	-43	0.90	-13	0.97	1366	-141	950
Development1	4	536	576	509	-67	0.88	-27	0.95	1386	-161	810
March 2015											
Development2	5	750									
Development3	6	856									
Development4	7	1064									
Test	8	1225									

Table 80 Case VI Failure Cost Data

Phase		AC	FC
Initiation	**1**	122	0
Analysis & Pr. Design	**2**	317	0
Design	**3**	416	22
Development1	**4**	576	49
Later			74

Table 81 Case VI Failure Costs of the Phases through the time

Total FC	**End of 1st phase**	**End of 2nd phase**	**End of 3rd Phase**	**End of 4th phase**	**Later**
Initiation	0	0	0	0	0
Analysis & Pr. Design	0	0	15	23	29
Design	0	0	7	21	28
Dev1	0	0	0	5	17

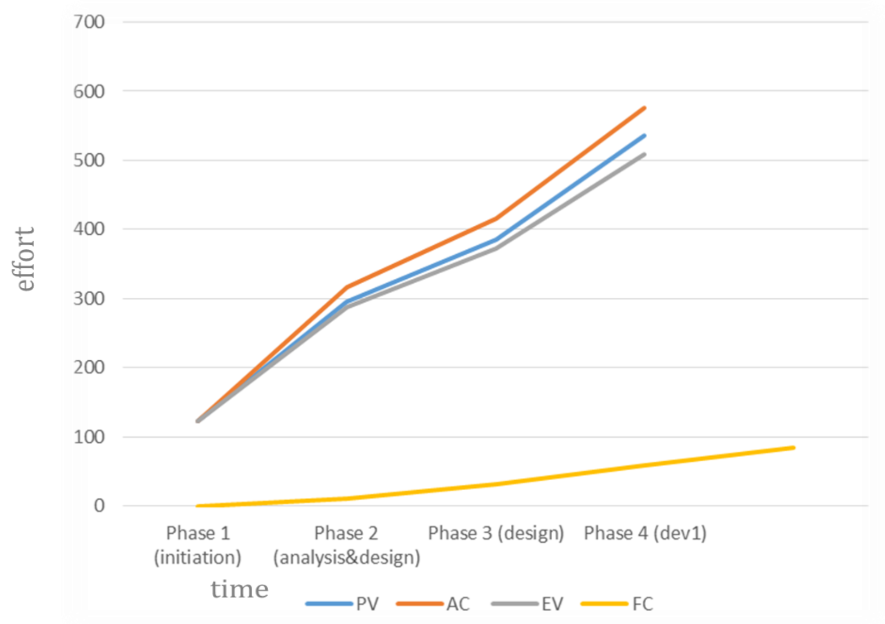

Figure 33 Case VI Failure Cost Graph

Table 82 Case VI qcf of the phases through the time

qcf	End of 1st phase	End of 2nd phase	End of 3rd phase	End of 4th phase	Final
Initiation	0	0	0	0	0
Analysis & Pr. Design	0	0	0.08	0.12	0.15
Design	0	0	0.07	0.21	0.28
Dev1	0	0	0.00	0.03	0.11

Table 83 Case VI Retroactive EV Matrix

EV_r	EV	End of Initiation	End of Analysis	End of Design	End of Dev1	later
Initiation	122	122				
Analysis & Pr. Design	166	-	153	146	141	
Design	85	-	-	79	62	45
Dev1	136	-	-	-	132	122

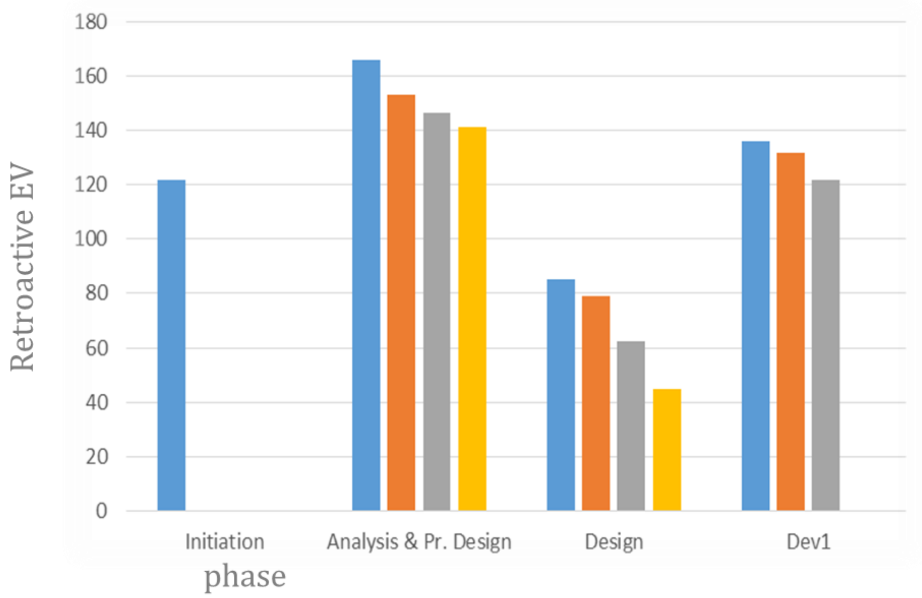

Figure 34 Case VI EVr change for the phases

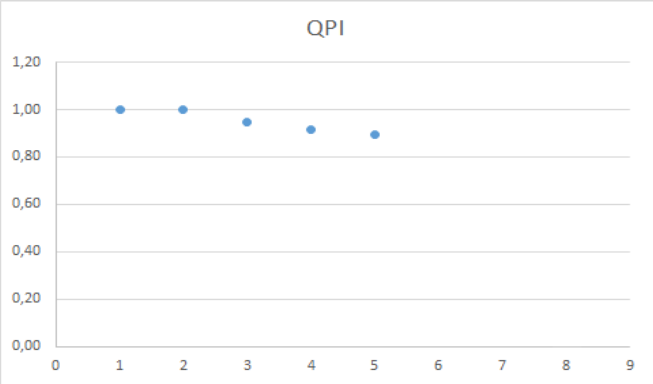

Figure 35 Case VI QPI Graph

Table 84 Case VI QEVM Application Performance Analysis and Forecast Metrics

Phase		PV	ACF	EV_{est}	CV_{est}	CPI_{est}	SV_{est}	SPI_{est}	QPI	EAC	ETC	VAC	EFC	TCFC
Initiation	**1**	122	122	122	0	1.00	0	1.00	1.00	1225	1103	0	0	0
Analysis &Pr. Design	**2**	295	317	288	-29	0.91	-7	0.98	1.00	1348	1031	-123	0	0
Design	**3**	386	438	366	-72	0.84	-20	0.95	0.95	1465	1027	-240	91	69
Dev1	**4**	536	625	486	-139	0.78	-50	0.91	0.92	1576	951	-351	175	126
later			699		-213	0.70			0.89	1762	1063	-537	207	133

Table 85 Case VI Comparison of ACs

Phase		EVM	QEVM
		AC	**ACF**
Initiation	**1**	122	122
Analysis & Pr. Design	**2**	317	317
Design	**3**	416	438
Development1	**4**	576	625
			699

Table 86 Case VI Comparison of EVs

Phase		EVM	QEVM	
		EV	EV_{est}	EV_{init}
Initiation	**1**	122	122	122
Analysis & Pr. Design	**2**	288	288	288
Design	**3**	373	366	343
Development1	**4**	509	486	422

Table 87 Case VI Comparison of Performance Indices

Phase		EVM	QEVM	EVM	QEVM
		SPI	**SPI_{est}**	**CPI**	**CPI_{est}**
Initiation	**1**	1.00	1.00	1.00	1.00
Analysis & Pr. Design	**2**	0.98	0.98	0.91	0.91
Design	**3**	0.97	0.95	0.90	0.84
Development1	**4**	0.95	0.91	0.88	0.78
later		0.95	0.91	0.88	0.70

Table 88 Case VI Comparison of Forecasts

Phase		EVM	QEVM	EVM	QEVM	EVM	QEVM	QEVM	
		EAC	**EAC**	**VAC**	**VAC**	**ETC**	**ETC**	**EFC**	**TCFC**
Initiation	**1**	1225	1225	0	0	1103	1103	0	0
Analysis & Pr. Design	**2**	1348	1348	-123	-123	1031	1031	0	0
Design	**3**	1366	1465	-141	-240	950	1027	91	69
Development1	**4**	1386	1576	-161	-351	810	951	175	126
later		1386	1762	-161	-537	810	1063	175	126

Table 89 Case VI QEVM Conformance Cost Data – phase-based

Phase	Conformance Costs			Planned Value	Conformance Cost Indices		
	Prevention Costs	Appraisal Costs	Conformance Costs		Prevention Cost Index	Appraisal Cost Index	Conformance Cost Index
Initiation	0	0	0	122	0	0	0
Analysis& Pr. Design	2	21	23	173	0.01	0.12	0.13
Design	4	25	29	91	0.04	0.27	0.32
Dev1	0	0	0	150	0	0	0
Dev2	0	0	0	214	0	0	0
Dev3	0	0	0	106	0	0	0
Dev4	0	0	0	208	0	0	0
Testing	0	100	100	161	0	0.62	0.62
Total	6	146	152	1225	0.004	0.12	0.12

A.7. Case VII

Table 90 Case VII EVM Application Performance Analysis and Forecast Metrics

Sprint		PV	AC	EV	CV	CPI	SV	SPI	EAC	VAC	ETC
Jan	**1**	174	167	162	-5	0.97	-12	0.93	1151	155	984
Feb	**2**	253	242	237	-5	0.98	-16	0.94	1141	145	899
Mar	**3**	357	325	321	-4	0.99	-36	0.90	1131	135	806
Apr	**4**	605	526	551	-25	1.05	-54	0.91	1066	70	540
Jun	**5**	735	656	665	-25	1.05	-54	0.91	1102	106	446
Jul	**6**	797	702	717	-9	1.01	-70	0.90	1094	98	392
Aug	**7**	903	831	808	15	1.02	-80	0.90	1149	153	318
Sep	**8**	996	914	901	-23	0.97	95	0.89	1133	137	219
Nov	**9**	1025	937	923	-13	0.99	95	0.90	1134	529	197
Dec	**10**	1117	1009	1001	-13	0.99	95	0.90	1126	391	117

Table 91 Case VII Failure Cost Data

Sprint		AC	FC
Jan	**1**	167	82
Feb	**2**	242	104
Mar	**3**	325	121
Apr	**4**	526	179
Jun	**5**	656	231
Jul	**6**	702	278
Aug	**7**	831	390
Sep	**8**	914	456
Nov	**9**	937	612
Dec	**10**	1009	666
later		1129	823

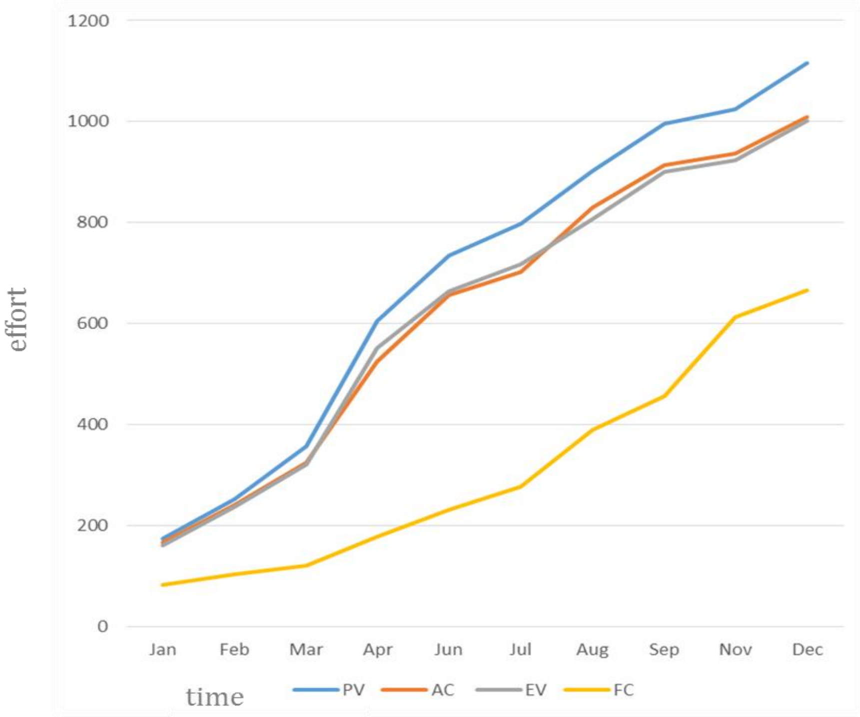

Figure 36 Case VII Failure Cost Graph

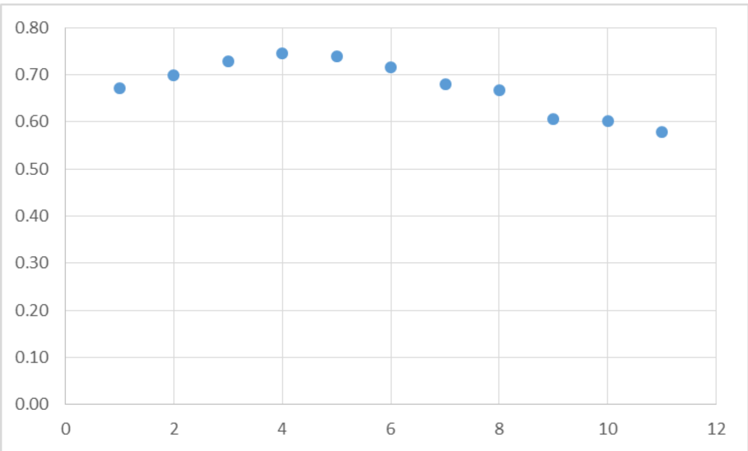

Figure 37 Case VII QPI Graph

Table 92 Case VII QEVM Application Performance Analysis and Forecast Metrics

Sprint		PV	ACF	EV_{est}	CV_{est}	CPI_{est}	SV_{est}	SPI_{est}	QPI	EAC	ETC	VAC	EFC	TCFC
Jan	**1**	174	249	162	-87	1.00	-12	0.93	0.67	1717	1468	-600	82	0
Feb	**2**	253	346	208	-138	0.60	-45	0.82	0.70	1856	1510	-739	453	349
Mar	**3**	357	446	293	-153	0.66	-64	0.82	0.73	1700	1254	-583	396	275
Apr	**4**	605	705	469	-236	0.67	-136	0.78	0.75	1678	973	-561	409	230
Jun	**5**	735	887	630	-257	0.71	-105	0.86	0.74	1572	685	-455	380	149
Jul	**6**	797	980	702	-278	0.72	-95	0.88	0.72	1560	580	-443	402	124
Aug	**7**	903	1221	776	-446	0.64	-128	0.86	0.68	1759	538	-642	512	122
Sep	**8**	996	1370	868	-502	0.63	-128	0.87	0.67	1763	393	-646	544	88
Nov	**9**	1025	1549	914	-635	0.59	-111	0.89	0.60	1893	344	-893	693	81
Dec	**10**	1117	1675	969	-706	0.58	-148	0.87	0.60	1932	257	-815	728	62
later			1952						0.58					

Table 93 Case VII Comparison of ACs

Sprint		EVM	QEVM
		AC	**ACF**
Jan	**1**	167	249
Feb	**2**	242	346
Mar	**3**	325	446
Apr	**4**	526	705
Jun	**5**	656	887
Jul	**6**	702	980
Aug	**7**	831	1221
Sep	**8**	914	1370
Nov	**9**	937	1549
Dec	**10**	1009	1675
later		1129	1952

Table 94 Case VII Comparison of EVs

Sprint		EVM	QEVM	
		EV	EV_{est}	EV_{init}
Jan	**1**	162	162	100
Feb	**2**	237	208	158
Mar	**3**	321	293	207
Apr	**4**	551	469	383
Jun	**5**	665	630	467
Jul	**6**	717	702	461
Aug	**7**	808	776	522
Sep	**8**	901	868	540
Nov	**9**	923	914	540
Dec	**10**	1001	969	551

Table 95 Case VII Comparison of Performance Indices

Sprint		EVM	QEVM	EVM	QEVM
		SPI	SPI_{est}	CPI	CPI_{est}
Jan	**1**	0.93	0.93	0.97	0.65
Feb	**2**	0.94	0.82	0.98	0.60
Mar	**3**	0.90	0.82	0.99	0.66
Apr	**4**	0.91	0.78	1.05	0.67
Jun	**5**	0.91	0.86	1.05	0.71
Jul	**6**	0.90	0.88	1.01	0.72
Aug	**7**	0.90	0.86	1.02	0.64
Sep	**8**	0.89	0.87	0.97	0.63
Nov	**9**	0.90	0.89	0.99	0.59
Dec	**10**	0.90	0.87	0.99	0.58

Table 96 Case VII Comparison of Forecasts

Sprint		EVM	QEVM	EVM	QEVM	EVM	QEVM	QEVM	
		EAC	EAC	VAC	VAC	ETC	ETC	EFC	TCFC
Jan	**1**	1151	1117	-155	-600	984	1468	82	0
Feb	**2**	1141	1856	-145	-739	899	1510	453	349
Mar	**3**	1131	1700	-135	-583	806	1254	396	275
Apr	**4**	1066	1678	-70	-561	540	973	409	230
Jun	**5**	1102	1572	-106	-455	446	685	380	149
Jul	**6**	1094	1560	-98	-443	392	580	402	124
Aug	**7**	1149	1759	-153	-642	318	538	512	122

Sep	**8**	1133	1763	-137	-646	219	393	544	88
Nov	**9**	1134	1893	-529	-776	197	344	693	81
Dec	**10**	1126	1932	-391	-815	117	257	728	62

Table 97 Case VII Comparison of MREs for EAC

Sprint		**EVM**	**QEVM**
		MRE	**MRE**
Jan	**1**	0.41	0.43
Feb	**2**	0.42	0.05
Mar	**3**	0.42	0.13
Apr	**4**	0.45	0.14
Jun	**5**	0.44	0.19
Jul	**6**	0.44	0.20
Aug	**7**	0.41	0.10
Sep	**8**	0.42	0.10
Nov	**9**	0.42	0.03
Dec	**10**	0.42	0.01
MMRE		0.42	0.11

A.8. Case VIII

Table 98 Case VIII EVM Application Performance Analysis and Forecast Metrics

	PV	**AC**	**EV**	**CV**	**CPI**	**SV**	**SPI**	**EAC**	**VAC**	**ETC**
Mar	2034	2340	1836	-504	0.78	-198	0.90	16369	-3526	14029
Apr	4284	5490	3726	-1764	0.68	-558	0.87	18923	-6080	13433
May	6174	7740	5436	-2304	0.70	-738	0.88	18286	-5443	10546
Jun	7974	10053	6984	-3069	0.69	-990	0.88	18487	-5644	8434
Jul	10134	12618	8829	-3789	0.70	-1305	0.87	18355	-5512	5737
Aug	12474	15210	10719	-4491	0.70	-1755	0.86	18224	-5381	3014
Sep	12843	15705	11088	-4617	0.71	-1755	0.86	18191	-5348	2486
Oct	12843	17523	12843	-4680	0.73		1.00			

Table 99 Case VIII Failure Cost Data

	Phase	AC	FC
Mar	**Analysis&Design**	2340	0
Apr	**Development**	5490	45
May	**Development**	7740	684
Jun	**Development**	10053	1242
Jul	**Development**	12618	2196
Aug	**Development**	15210	3249
Sep	**Training**	15705	4077
Oct		17523	4698
later			5535

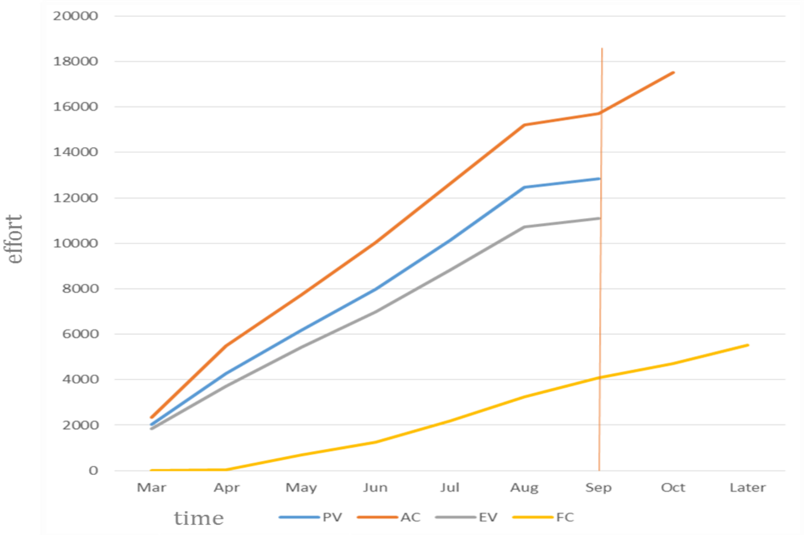

Figure 38 Case VIII Failure Cost Graph

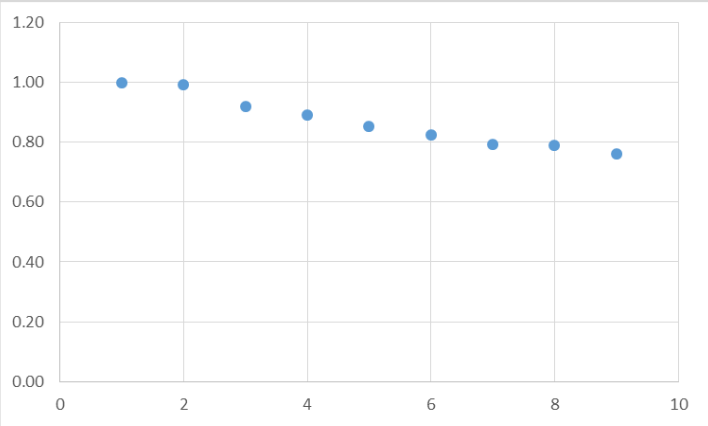

Figure 39 Case VIII QPI Graph

Table 100 Case VIII QEVM Application Performance Analysis and Forecast Metrics

	PV	ACF	EV_{est}	CV_{est}	CPI_{est}	SV_{est}	SPI_{est}	QPI	EAC	ETC	VAC	EFC	TCFC
Mar	2034	2340	1836	-504	0.78	-198	0.90	1.00	16369	14029	-3526	0	0
Apr	4284	5535	3690	-1845	0.67	-594	0.86	0.99	19263	13728	-6420	221	176
May	6174	8424	5247	-3177	0.62	-927	0.85	0.92	20621	12197	-7778	1630	946
Jun	7974	11295	6770	-4525	0.60	-1204	0.85	0.89	21427	10132	-8584	2217	975
Jul	10134	14814	8498	-6316	0.57	-1636	0.84	0.85	22388	7574	-9545	3145	949
Aug	12474	18459	10332	-8127	0.56	-2142	0.83	0.82	22945	4486	-10102	3896	647
Sep	12843	19782	11010	-8772	0.56	-1833	0.86	0.79	23075	3293	-10232	4568	491
Oct	12843	22221	12439	-9782	0.56	-494	0.97	0.79	22943	722	-10100		
late	12843	23058	12439		0.54		0.97	0.76					

Table 101 Case VIII Comparison of ACs

		EVM	**QEVM**
	Phase	**AC**	**ACF**
Mar	**Analysis&Design**	2340	2340
Apr	**Development**	5490	5535
May	**Development**	7740	8424
Jun	**Development**	10053	11295
Jul	**Development**	12618	14814
Aug	**Development**	15210	18459
Sep	**Training**	15705	19782
Oct		17523	22221
later			23058

Table 102 Case VIII Comparison of EVs

		EVM	QEVM	
	Phase	**EV**	**EV_{est}**	**EV_{init}**
Mar	**Analysis&Design**	1836	1836	1801
Apr	**Development**	3726	3690	3313
May	**Development**	5436	5247	4684
Jun	**Development**	6984	6770	5732
Jul	**Development**	8829	8498	7021
Aug	**Development**	10719	10332	8453
Sep	**Training**	11088	11010	8535
Oct		12843	12439	9760
later				23058

Table 103 Case VIII Comparison of Performance Indices

		EVM	QEVM	EVM	QEVM
	Phase	**SPI**	**SPI_{est}**	**CPI**	**CPI_{est}**
Mar	**Analysis&Design**	0.90	0.90	0.78	0.78
Apr	**Development**	0.87	0.86	0.68	0.67
May	**Development**	0.88	0.85	0.70	0.62
Jun	**Development**	0.88	0.85	0.69	0.60
Jul	**Development**	0.87	0.84	0.70	0.57
Aug	**Development**	0.86	0.83	0.70	0.56
Sep	**Training**	0.86	0.86	0.71	0.56
Oct			0.97	0.73	0.56
later					0.54

Table 104 Case VIII Comparison of Forecasts

		EVM	QEVM	EVM	QEVM	EVM	QEVM	QEVM	
Phase		**EAC**	**EAC**	**VAC**	**VAC**	**ETC**	**ETC**	**EFC**	**TCFC**
Mar	**Analysis&Design**	16369	16369	-3526	-3526	14029	14029	0	0
Apr	**Development**	18923	19263	-6080	-6420	13433	13728	221	176
May	**Development**	18286	20621	-5443	-7778	10546	12197	1630	946
Jun	**Development**	18487	21427	-5644	-8584	8434	10132	2217	975
Jul	**Development**	18355	22388	-5512	-9545	5737	7574	3145	949
Aug	**Development**	18224	22945	-5381	-10102	3014	4486	3896	647
Sep	**Training**	18191	23075	-5348	-10232	2486	3293	4568	491
Oct		17523	22943	-4680		0	722	0	0

Table 105 Case VIII Comparison of MREs for EAC

Phase		EVM	QEVM
		MRE	MRE
Mar	Analysis&Design	0.29	0.29
Apr	Development	0.18	0.16
May	Development	0.21	0.11
Jun	Development	0.20	0.07
Jul	Development	0.20	0.03
Aug	Development	0.21	0.005
Sep	Training	0.21	-0.001
Oct		0.24	0.005

APPENDIX B

B.1 - Case Study Statement sent to the companies in Turkish

Doktora Tezi Durum Çalışması

ODTÜ Enformatik Enstitüsü Bilişim Sistemleri Bölümü

Pınar Efe, Danışman: Prof. Dr. Onur Demirörs

Doktora tezim kapsamında proje yönetiminde sıklıkla kullanılan Kazanılmış Değer Yönetimi (Earned Value Management) yönteminin yazılım projelerinde daha etkin ve doğru kullanılabilmesini sağlayacak QEVM (Quality Integrated Earned Value Management) adını verdiğimiz bir model geliştirdim. Geliştirdiğimiz model, yazılım projelerinde sıklıkla karşılaştığımız değişiklik ve tekrarlama (reworking) eforlarının Kazanılmış Değer Analizi yöntemine ve dolayısıyla proje yönetimine entegre edilmesini amaçlamıştır.

QEVM

Kazanılmış Değer Yönetimi proje yönetimi alanında dünyada yaygın olarak kullanılan, PMI'ın desteklediği, etkin bir performans izleme ve geri besleme aracıdır. Proje yöneticileri bu yöntemle projelerin ilerlemesini net bir şekilde görebilir ve buna ek olarak proje gidişatı ile ilgili tahminlemeler yapar. Kazanılmış Değer Yönetimi performans izlemesini içerik, takvim ve maliyet üzerinden yapar, yapılan işin kalitesini açık bir şekilde izlemeye almaz, yapılan bir işin istenildiği gibi gerçekleştirildiğini var sayar. Bu yaklaşım bazı alanlardaki projelere uygun olsa da, yazılım projeleri gibi kimi zaman proje bütçesinin %40-50 değerlerine ulaşan yüksek tekrarlama maliyeti içeren projeler için çok uygulanabilir değildir, proje yöneticilerini yanlış yönlendirebilmektedir.

Geliştirdiğimiz modelde (QEVM) kalite maliyetleri (Cost of Quality) kavramlarından yararlanılarak Kazanılmış Değer Yönetimi yönteminde iyileştirmeler yapılmış, projelerin içerik, takvim ve maliyetlerinin yanı sıra kalitesinin de izlenmesi sağlanmıştır.

Bu aşamada model uzmanlar tarafından gözden geçirilmiş olup hali hazırda 3 projede durum çalışmaları ile uygulanmıştır. Amacımız daha fazla organizasyonda farklı özelliklerdeki projelerde durum çalışmaları ile uygulamalar yaparak modelin gerçekten amacına uygun şekilde çalışabilirliğini değerlendirmek ve varsa iyileştirme noktalarını belirleyip iyileştirmeler yapmaktır. Bu çalışmaların modeli uyguladığımız organizasyonlar için de önemli geri bildirimler üreteceğine inanıyoruz.

Projenizde Durum Çalışmaları

Durum Çalışması için modelin ihtiyacı olan veri; proje planlanma ve izlenme verileri ile ortaya çıkan hata ve değişikliklerin izlenme verileri. Bu verileri size uyan bir formatta (excel, MS Project.vs) sağladıktan sonra, ilgili olarak proje yöneticisi ile görüşme yapmam yeterli olacaktır. Proje hakkında içeriğine girmeden genel bilgileri öğrenmem (büyüklük, metodoloji, çalışan kişi sayısı..vs) raporlama yapabilmek açısından anlamlı olacaktır. Projelerinizde Kazanılmış Değer Yönetimi uyguluyor olmasanız da proje planlama ve izleme verileriyle öncelikle klasik Kazanılmış Değer Yönetimi uygulayıp, daha sonra geliştirdiğimiz QEVM modeli uygulayacağım. Değerlendirme bittikten kısa bir görüşme ile sonuçları birlikte değerlendirebiliriz.

B.2 -Case Study Statement sent to the companies in English

Case Study

in the scope of PhD Thesis of Pınar Efe, Advisor: Prof.Dr.Onur Demirörs

METU, Informatics Institute

In the scope of my PhD study, I have developed a model, QEVM, which extends Earned Value Management and enables more accurate and efficient project tracking for software projects. The model aims to integrate quality dimension to the EVM and provides more effective management considering reworking efforts.

QEVM

EVM is a widely used simple and quantitative method in project management. It is quite simple and gives a general idea about the project progress and its future seamlessly. It brings together three fundamental elements of the project; scope, schedule and cost and evaluates their progress comparing to plans at the beginning in order to give the clear picture of project progress and performance. The fourth fundamental element of a project, quality, is not directly included in traditional EVM. Quality is assumed to be a part of scope element. It means when the task is completed, it is supposed to be completed without any quality deficiency here and therefore quality shortages or issues are not considered explicitly in this method. Even though this approach might work for many project management disciplines, the software projects could not utilize EVM properly due to the high reworking efforts.

Our model, QEVM, takes quality dimension into consideration and so provides more accurate current progress as well as more reliable future estimates. Recently just after we developed the model, we already applied it on three projects in two different organizations. Our goal is to apply it on more projects with differents specifications in different organizations to truly assess the applicability, to detect deficiencies if any and to make improvements on the model with the feedbacks. We believe that the case studies will generate significant results and feedbacks for the organizations.

Case Study Requirements

For the case study, I need project planning and monitoring data in addition to the data associated with errors and changes (WBS items, WBS items tracking data, progress reports, timeline and effort for errors and changes, the tracebility of the errors and WBS items if possible). I do not need any content info other than general project information (size, methodology, number of employees...etc), which could be significant in terms of reporting.

It is sufficient to do an interview with the project manager. Even if you do not implement Earned Value Management in your projects, after you provided planning and monitoring data, first I'll apply classical Earned Value Management on your project. Afterwards I'll apply QEVM model and prepare a report containing findings after evaluation. We can evaluate the results together with a short interview.

B.3 -Questions of semi-structured interview

Item	Explanation
Project name	
Project description	
Organization name	
Organization description	
Project planned start date	
Project planned end date	
Project actual start date	
Project actual end date	
Development methodology	
Project organization	
Project team size	
Team staff and profiles	
Programming languages	
Technologies used	
Tools used	
Project management tool	
Issue tracking tool	
Requirements management tool	
Test management tool	
Total planned effort	
Actual effort	
Effort spent for the reworking	
Reworking	
How do you manage reworking? Errors and changes?	
Are you tracking the efforts for the later issues?	
How do you plan reworking initially?	
Are you exploring the causes of reworking?	
Are you updating PMB? in which cases?	

B.4-Progress Report Template

	Planned			Actual				
WBS Item	Effort	Start Date	End Date	Effort	Completion %	Start Date	End Date	Earned Value

B.5 Issue List Template

Issue No	
Problem Description	
Solution	
Priority	
Effort (hour)	
Detected at	
Detected in	
Detected by	
Fixed in	
Related Task/Component/	
Issue Type	
Failure Type (int/ext)	

CURRICULUM VITAE

PERSONAL INFORMATION

Pınar Efe was born in Denizli on June 13, 1981. She received her bachelor degree from Computer Engineering in Middle East Technical University in 2002. She received her M.S. degree from Information Systems in Informatics Institute of Middle East Technical University in 2006. She has been working in software projects more than 13 years. Her research interests involve software project management, earned value management, people management, software size estimation and measurement, agile software development. You can contact her at pinar.efe@gmail.com

WORK EXPERIENCE

July 2015 – Present, Türk Telekom Grubu, Agile Coach

2013 –2015, ICTerra, Ankara, Software Consultant

2007 –2013, Siemens EC, Project Manager&Group Manager

2004 –2007, Siemens, Software Engineer

2004 –2007, TÜBİTAK UEKAE, Researcher, Software Developer

2002 –2003, Havelsan, Software Engineer

EDUCATION

2009 – Present, PhD, Information Systems at the Informatics Institute, Middle East Technical University

2002 – 2006, MSc, Information Systems at the Informatics Institute, Middle East Technical University

Thesis Title: "A Unification Model and Tool Support for Functional Size Measurement"

1997 – 2002, BS, Computer Engineering Department at Middle East Technical University

PUBLICATIONS

Efe P., Demirörs, O., Yazılım Projelerinde Kazanılmış Değer Yönetimi Kullanımı, Ulusal Yazılım Mühendisliği Sempozyumu, İzmir, Türkiye, September 2426, 2013

Efe P., Demirörs, O., Applying EVM in a Software Company: Benefits & Difficulties, 39th EUROMICRO Conference on Software Engineering and Advanced Applications, Santander, Spain, September 46, 2013.

Efe, P., Demirors, O., Gencel, C., "Mapping Concepts of Functional Size Measurement Methods", Book chapter in COSMIC Function Points: Theory and Advanced Practices, Editors: Dumke, R. and Abran, A., CRC Press, 2010.

Desharnais, J.M., Abran, A., Efe, P. Ilis, M.C, Karaca, I.N., Functional Size of a Real Time System. Software Process and Product Measurement, International Conferences IWSM 2009 and Mensura 2009, Amsterdam, Netherlands, November 46, 2009.

Efe P., Demirors O., Gencel C., Mapping Concepts of Functional Size Measurement Methods, Proceedings of IWSM/MetriKon 2006, Potsdam (Germany), November 24, 2006

Buglione L., Gencel C., Efe P., Suggestions for Improving Measurement Plans: a BMP application in Turkey, Proceedings of IWSM/MetriKon 2006, Potsdam (Germany), November 24, 2006, pp. 203-227

Gencel C., Buglione L., Demirors O., Efe P., A Case Study on the Evaluation of COSMICFFP and Use Case Points, Proceedings of SMEF 2006, 3rd Software Measurement European Forum, Rome (Italy), May 1012, 2006, pp.121-140

CERTIFICATES

2007 – IT Infrastructure Library (ITIL) Foundation Level

2008 – COSMIC Software Functional Size Measurer

2009 – Project Management Professional, PMP

2009 – Microsoft Certified Professional Developer, MCPD

2011 – Professional Scrum Master, PSM I

INTERESTS

Travelling, cinema, photography, running, yoga and psychology.