Computer Standards & Interfaces 66 (2019) 103353

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

A change management model and its application in software development projects

Pinar Efe[*], Onur Demirors

Bilgi Grubu, Ankara, Turkey, Department of Computer Engineering, Izmir Institute of Technology, Izmir, Turkey

ARTICLE INFO

Keywords:
Earned Value Management
Software project management
Performance measurement
Reworking
Change management

ABSTRACT

Change is inevitable in software projects and software engineers strive to find ways to manage changes. A complete task could be easily in a team`s agenda sometime later due to change demands. Change demands are caused by failures and/or improvements and require additional effort which in most cases have not been planned upfront and affect project progress significantly. Earned Value Management (EVM) is a powerful performance management and feedback tool for project management. EVM depicts the project progress in terms of scope, cost, and schedule and provides future predictions based on trends and patterns of the past. Even though EVM works quite well and widely used in disciplines like construction and mining, it is not the case for software discipline. Software projects require special attention and adoption for change. In this study, we present a model to measure change and subsequent rework and evolution costs to monitor software projects accurately. We have performed five case studies in five different companies to explore the usability of the proposed model. This paper depicts the proposed model and discusses the results of the case studies.

1. Introduction

Project management aims to deliver projects on time, with the agreed scope and quality according to specified requirements and within the planned budget. The achievement of project management is perceived as planning the project accurately at the beginning and then executing the project according to this plan.

Execution of the projects requires monitoring progress. EVM is commonly used performance management tool to measure project progress objectively in terms of scope, cost and schedule. It basically compares the planned work and accomplished work in a project and calculates the value of this accomplished work. EVM is called as “Management with the lights on” [1] since it clearly identifies where the project is at a specific time and where it is going based on the trends. EVM is widely used in numerous industries such as construction and mining. Though, it is still little known and utilized in the software industry.

The tools, techniques, and methods used in the traditional project management have been applied to software projects for years. However, in the field of software engineering, there is an inevitable factor of change that makes the majority of tools, methods, and techniques unusable as they are. The earlier approach in the industry was to minimize changes by making better analysis, better plans and preventing changes. The products as well as processes have been the target for stabilization without producing desired effects [2–5] but it was never enough to prevent or avoid change [6–8].

Research studies show that software specialists spend about 30%–50% of their time on rework rather than on work done right in the first time [7,9]. In traditional fields such as construction and mining, rework is neither very common and acceptable nor physically possible, particularly after certain milestones. Once a task is completed, it is assumed that there will be no further related work. As a result, the methods, tools and techniques built on the assumption that clear milestones cannot be used as is, in software projects.

Software industry has been searching for different solutions for the substantial change issue. Starting from 1980s prototyping methods and iterative life cycle models are developed [10] and later from the 2000s, agile approaches are in focus to manage changes [11]. Today, agile approaches are used by a prominent percentage of software organizations [12–16]. Agile approaches embrace the change and use it as an opportunity but do not have the variety of the tools that exist for plan-driven project management specifically for depicting the status quantitatively and establishing future estimates. To provide best of both worlds traditional project management methods, tools, and techniques could be adapted or replaced by more effective ones for software engineering projects.

[*] Corresponding author.
E-mail addresses: pinar@bg.com.tr (P. Efe), onurdemirors@iyte.edu.tr (O. Demirors).

https://doi.org/10.1016/j.csi.2019.04.012
Received 23 January 2019; Received in revised form 9 April 2019; Accepted 26 April 2019
Available online 29 April 2019

EVM is tightly connected with the traditional project management methods and as such it assumes that the plan will be stable over the life of the project. Initial planning and baselines are significant for EVM, as it basically depicts how much the project is aligned with the initial plan. EVM does not offer any special treatment for later changes. In other words, there is a conflict between the foundational assumptions of EVM and the facts of software projects. Apparently, “done” for a task does not always mean that this task was done completely in software projects. It might be reworked after some time during the project due to various reasons like defects and improvements. In most cases, the requirements also change, and systems evolve as they are built. Evolution and rework result in unplanned effort that requires special attention for monitoring the project.

The main problem of EVM for software projects is the calculation of the value earned. As presented in [17], at a given time in the project, the calculated earned value (*EV*), can be less than the formerly calculated one. The effort and cost spent later do not increase the value earned, it goes for the change. The scope is still the same but costs more. It is not the cost for scope but for reworking. As a result, it depicts an incorrect picture to project managers about the progress and the future of the project. The effort spent for rework and evolution including unpredictable changes, requirement and design changes, defects, technical debts is ignored. If we did it 100% correct in every aspect for the first time and there were no later changes, we would have the correct *EV* in every calculation. Therefore, to be able to utilize EVM better in software projects, cost and effort measures related to change are required since they already affect the time and budget calculations of the project.

The studies on EVM and software projects in the literature address the uncertainty and changeability in software projects, by focusing on preventing and/or avoiding the uncertainty and change [18–22]. However, they do not propose any solution that incorporates change as a significant parameter in the calculations. Their approach is quite consistent with the approach of traditional project management.

In this study, we propose a change management model integrated with EVM. This model accepts change as a reality and a characteristic of a software project and aims to manage it rather than preventing and/or avoiding it. The model will be referred as CMOD in the rest of the paper. CMOD enables change aspect to be quantified and depicted. It is based on representing change effort of any kind such as rework and evolution and relates them to scope, schedule and cost aspects to enable better visualization of software projects progress. Additionally, it provides more precise future estimates based on these more accurate progress and trend metrics. CMOD presents new concepts and metrics related with change. These metrics help project managers monitoring and controlling change efforts.

We have conducted five case studies in five diverse companies to investigate the usability of the model by exploring its precision for depicting software project progress and future estimates. To do so, we applied CMOD as well as traditional EVM to these five case projects for the same intervals and compared the results. The planned efforts for the case projects are 1117 to 12,843 person-hours and the planned durations are 5 months–10 months. The project organizations are from different domains: Telecommunication, E-Government Systems, Information and Communication Technologies, Software Development and Consultancy Services in Defense Industry, and Systems Integration Services.

We have obtained promising results in these studies. In all projects, we were able to calculate the progress as well as future estimates more precisely by CMOD. We have also faced with new challenges that require further studies.

The paper is structured as follows. Section 2 summarizes the background and research performed on EVM. Section 3 proposes the change oriented EVM model that we proposed. Section 4 presents all the case studies, with the background of the projects and the results of the execution. Section 5 discusses the findings and draws the conclusion.

2. Background

2.1. EVM method overview

The Earned Value concept has been used in industrial manufacturing since the late 1800s [23]. In the early American factories, it was used in its most fundamental form by measuring the performance of “planned standards” using “earned standards” gained against “actual expenses”.

EVM formally introduced as a project management tool by the US Navy in 1962. Since then, it has been evolved and spread out to the private industry, other governmental agencies and the other countries [1,24,25]. In 1998, EVM was formally issued as a standard by American National Standards Institute (ANSI) [24]. Project Management Institute (PMI) first included EVM into its PMBOK and later updated in every new version. Also, in 2007, PMI published a separate guideline “Practice Standard of Earned Value Management” as a supplement to PMBOK Guide to facilitate its role in effective project management [1,26].

EVM has two major key practices, which are establishing a performance measurement baseline and measuring the performance against this baseline [1]. It has three key data elements: (1) Planned Value (PV) is the sum of all the budgets for all planned work in the schedule. (2) Earned Value (EV) is the value of the work accomplished at a given point in time. (3) Actual Cost (AC) is the summation of the resources expended for the time period. These three data elements are the basics of EVM and could be easily understood graphically, as shown in Fig. 1. All the other EVM performance metrics including variances, indices, which are depicted in Fig. 1 as well, and forecasts are derived from these elements.

The complete EVM metrics are presented in Table 1 [1,26]. The variances (SV and CV) show the project's current status. If SV is less than zero, it means that the project is behind the schedule. If CV is less than zero, it means that the project is behind over budget. The indices (SPI and CPI) are the indicators of how cost and schedule efficiently used and also represent the trends of the progress. If SPI is less than one, it means that the project is behind the schedule and the project works are not effectively completed on their planned time. If CPI is less than one, it means that the project is over budget and the project works are not completed on their planned budget. The indices are used to predict the future of the project based on the fundamental principle that trends and patterns in the past determine the future [26].

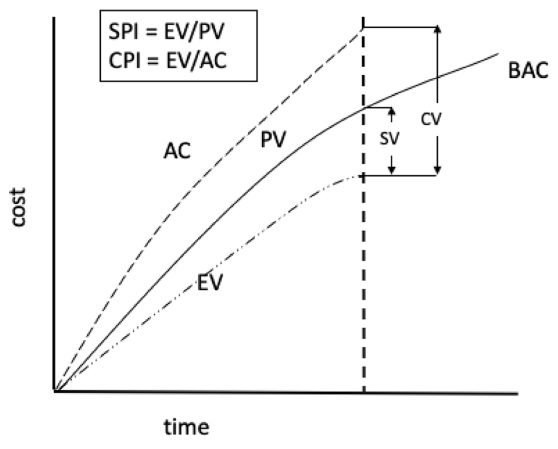


Fig. 1. EV metrics.

Table 1
EVM metrics.

Metric	Equation	Description
Schedule Variance (SV)	EV − PV	the difference between the planned value of the work scheduled and the value of the work accomplished
Schedule Performance Index (SPI)	EV /PV	index showing the efficiency of the time utilized
Cost Variance (CV)	EV − AC	the difference between the values of the work accomplished and the actual cost incurred
Cost Performance Index (CPI)	EV/AC	index showing the efficiency of the utilization of the resources allocated
Budget at Completion (BAC)	Total PV	cost of total estimated work in the plan
Estimate to Complete (ETC)	(BAC − EV) / CPI	estimated cost required to finish all the remaining work
Estimate at Completion (EAC)	AC + ((BAC − EV)/CPI), AC + ETC, BAC/CPI	projected final cost required to finish complete work and based on a statistical prediction using the performance indexes
Variance at Completion (VAC)	BAC − EAC	the difference between what the project was originally baselined to cost, versus what it is now estimated to cost

2.2. *EVM in software projects*

The practice of EVM in software projects has been the subject of the researches in the literature due to the special characteristics of software projects.

In an early study, Christensen et al. state that it is difficult to use Earned Value on software projects since the models that estimate cost and schedule and the metrics for measuring work progress are insufficient [18]. He criticizes the lack of standardized metrics of software development processes and proposes the following metrics to apply EVM to the software projects better and easier:

- # of requirements determined
- # of components designed, coded and tested, effort of the project in person-months
- size of software (e.g., Lines of Code),
- measure of the available computer hardware timing, memory, input/output resources consumed
- # of changes made to the requirements
- # of changes to the detailed design

Fleming and Koppelman state that EVM could be applied to software projects just as it is applied to the other projects [27]. They do not emphasize any variations in the method. Instead, they present a guideline to apply EV successfully on all projects and advise ten musts to implement EVM on all projects.

Brandon discusses the integrating EV into the management of software projects and states five difficulties [19]:

- Most of the project cost is labor, usually varying productivity, even within the same job category
- Quantitative methods of measuring task progress are immature
- The technology and 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.

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

Solomon criticizes EVM for software projects in several aspects [20]. First, EVM was not designed to manage risk but perceived and used to be a risk management tool. Second, EVM covers only the project scope and disregards the product scope. Third, *EV* is a derived measure and so its effectiveness depends on the reliability and accuracy of its base measures. Fourth, EVM does not require precise, quantifiable measures. Fifth, *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 performance of work. Based on these limitations, Solomon proposed an extension, Performance Based Earned Value (PBEV), which particularly aims to overcome the shortcomings of EVM related to measuring technical performance considering customer requirements. Using CMMI, PMBOK, EIA 632, IEEE 1120, and his experiences, he defines PBEV guideline including step-by-step activities. PBEV focuses on the customer requirements and underlines the product scope and quality rather than project scope and quality. It specifies the performance-based measures of progress for satisfying product quality requirements [28,29].

Ferle, in her study of EVM application on IT projects, points out the following difficulties specific to software projects [21]: EVM application for software projects is not trivial and explores the complexities in planning, monitoring and reporting processes. Effort estimation is extremely difficult, even impossible in certain situations. The effort relies on the skills of the developers. Initial estimate, even in this uncertainty, is quite significant in EVM. During monitoring, the subjective assessment of task progress makes the visibility of progress challenging.

Hanna (2009) presents the challenges specific to software management on EVM [22]. He categorizes these challenges under three groups: (1) Innovative characteristics of software projects bring the large uncertainty to the estimates of cost and schedule, which might result in unplanned rework and incorrect estimations. (2) The cost and schedule of the defects is not easy to predict and it may result in long delays with no positive progress, which makes performance indices incorrect. (3) The architectural changes during time progress after more technical knowledge gained can again result in unplanned rework and inaccurate estimations. It also brings new defects and so the challenges related with defect discovery are also included in this challenge. Furthermore, he proposes several approaches to make a software project more appropriate for EVM: measuring volatility through metrics, improving task definitions, improving scheduling techniques, improving cost account, using incremental approach with prioritizing high risk and high value tasks, identifying and calculating bias and measuring component volatility.

There are also various studies on applications of EVM on software projects applying agile approaches.

Agile EVM is a light-weight adaption of EVM for agile project management [30]. The idea of using EVM together with the agile approach was first proposed by Lett from Lockheed–Martin in [31]. Afterwards, Agile EVM has been evolved with several contributions [32–35]. Agile EVM does not aim to replace current agile metrics. It brings additional metrics to increase the visibility of the project status and to support decision making.

Agile EVM particularly uses the values defined in Scrum and contains a simplified set of EVM calculations. It uses the product backlog instead of PMB and tracks the progress with respect to the releases, which corresponds to a phase applying Agile EVM and may include iterations or sprints or might be a time-interval, weeks or month. It uses the story points 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*.

In 2009, Rusk emphasizes that agile and EVM are a natural fit for each other and EVM can be radically simplified for agile projects [36]. The techniques used in agile methods such as burn charts provide status and progress information quite similar to what EVM provides.

Wu discusses the limitations of EVM and compares the processes of traditional EVM and Agile EVM [37]. 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, so it cannot measure quality objectively.

There are further studies on applying EVM together with other methodologies. By incorporating Function Points (FP) into EVM, the researchers use FP as a means of quantifying the work and calculate cost and schedule performances accordingly [38–41]. In a similar way, tracking software projects with EVM and Use Case Points (UCP) is the topic of another research [42–44].

The most prominent solution in these studies 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. PBEV perceives later changes as quality problems and proposes defining quality requirements better to reduce later change. Although Agile EVM does not aim to control change, it is fundamentally different than traditional project management and EVM by rejecting upfront planning since the main idea of traditional EVM is to measure how much the progress is aligned with the initial plan.

Based on the literature that we presented in this section, we can state that software projects have fundamental difficulties in terms of application of EVM. These studies look from different perspectives to EVM. The studies focusing on traditional EVM struggle to control change while agile related ones accept change and attempt to adapt EVM to agile approach. The common findings in these studies could be summarized as uncertainty, architectural and technical changes and lack of reliable metrics for productivity, progress measures, schedule and cost estimations. The researchers proposed various solutions to these issues in the scope of their studies by preventing uncertainty and change or defining objective measures like FP or UCP. However, these studies do not propose any solution for managing the change rather than preventing or avoiding it. Also, they do not suggest such an extension to EVM with the concept of change management.

3. CMOD – a change management model for software project monitoring

CMOD establishes a change management model for software project monitoring and performance measurement by extending traditional EVM. CMOD accepts that change is inevitable for software projects due to various reasons and adjusts EVM according to this fact rather than preventing or avoiding change. The objectives of the model can be summarized as follows:

- to integrate change and subsequent reworking and evolution effort into EVM
- to reflect project change status in addition to scope, schedule and cost
- to estimate project progress more accurately at any given time by considering past change data
- to offer more realistic forecasts

CMOD provides better visibility of scope, schedule and cost dimensions in the traditional EVM by integrating the change aspect into calculations. To be able to do so, we define change related concepts, elements and measures. In addition, we reformulate the future related estimates. Traditional EVM estimates the future of the project based on the fundamental principle that trends and patterns in the past determine the future. CMOD uses the same approach for future estimates, but integrates change into the equations. It assumes that the change trend occurred in the past will be similar in the future.

The foundational concept of CMOD is the rework and evolution costs incurred by change. Software projects encounter the following situation regularly: at a specific time, a task is closed since it is assumed to be completed but after some time, rework is required due to a defect or an improvement coming from the testers, field trials…etc. and more effort spent on the same task. If it is completed at first time, what happened to effort spent later? CMOD assumes that completed tasks is under effect of a change factor and might be changing over time as EV is. It proposes a new concept, estimated *EV*, which is the calibrated value of current *EV* based on estimated rework and evolution efforts.

The distinguishing feature of CMOD from the traditional EVM is to incorporate rework and evolution costs into actual costs and to calibrate *EV* based on the total cost and its trends in time. In addition to three key data points of traditional EVM, CMOD defines Reworking and Evolution Cost (*REC*), Total Cost (*TC*), Cost Factor (*cf*), First-time Completion Efficiency(*ftce*), Expected Reworking and Evolution Cost (*RECexp*), and Estimated *EV* (*EVest*) as described in Table 2.

Furthermore, CMOD improves the performance metrics of traditional EVM as seen in Table 3.

REC is the cost of any change happening after the work completed. The collected *REC* is expected to relate with the previous release in the project since the work are in-progress in the current application period and *REC* is not meaningful before completing the work once. *REC* is quite critical data for CMOD and the success of the method heavily depends on correctly gathered data of rework and evolution activities. For that purpose, the organization should have records of changes and their associated efforts. It is significant to keep and track these efforts and tailor issue tracking activities of the project accordingly.

Table 2
CMOD additional core measures.

Metric	Equation	Description
Reworking and Evolution Cost (*REC*)	–	rework and evolution cost occurring once the work completed, might be caused by bugs, defects, and improvements
Total Cost (*TC*)	REC + AC	total cost of project summing up rework and evolution cost with actual cost
Change Factor (*cf*)	REC/ACt-1	index showing the change ratio of the work completed
First-time Completion Efficiency (*ftce)*	1 – cf	index showing what percentage of the work done right at first time
Expected Reworking and Evolution Cost (*RECexp*)	AC * cf	total expected rework and evolution cost for the completed work according to the change factor
Estimated Earned Value (*EVest*)	EV * TC/(AC + RECexp)	calibrated EV according to the change trends

TC represents total effort including rework and evolution. In traditional EVM, *AC* only covers the initial completion cost, but *TC* takes later change costs into consideration. *REC* and *TC* are gathered in parallel as EVM measures.

cf is the ratio of change and shows us the change percentage of the completed work. It is calculated by means of division of *REC* to related *AC*, represented as *ACt-1*. While by definition [1], *AC* is the summation of the costs in accomplishing all work performed for the time period, *ACt-1* is the summation of the costs belongs to previous EVM period. The main idea here is that *REC* gathered in the current EVM period belongs to *AC* of previous one since we aim to find how much completed work changed.

RECexp represents the projection of rework and evolution costs at a specific time. The change trend in the past, *cf*, is used to calculate it. Thus, we would have an idea about how much change may be needed to earn this value.

EVest represents the actual *EV* considering prospective rework and evolution effects. CMOD proposes to calibrate *EV*. The value earned at some time earlier might have been reworked until the time being, more effort spent after the first completion, but the earned value is not changing. It means that we may have similar deviations in the current value earned after some time passed. In order to handle the *EV* deviation and reflect the project status more precisely, the most convenient way is to approximate the value based on the past trends. As a result, we propose a new metric, *EVest*, which is calibrated value of current *EV* by means of potential reworking and evolution costs, *RECexp* that we calculated based on the change factor, *cf*. By means of basic projection, *EV* is multiplied by the ratio of total cost over expected total cost.

ftce shows us "doing it right first time" efficiency. It provides visibility for the development process from the quality perspective if changes are assumed to be exceptions.

CMOD proposes to have organizational *cf* database for different project types for an organization in addition to tracking project *cf*. It is quite informative for organizations as well as project managers, if it is tracked properly. From the organizational perspective, it might give an opportunity to compare the projects. Additionally, industrial *cf* benchmarking values could also be collected. In some cases, depending on the project characteristics, the trends would not be visible until the midst of the project and organizational or industrial values would be helpful there.

CMOD uses effort instead of money as the cost unit since the main cost driver of a software project is the effort. It is very common to track software projects and even apply traditional EVM by using effort as cost. However, it will be a trivial process to change it to monetary units if needed.

4. Case studies

A multiple case was designed in order to explore the applicability of CMOD and to compare the benefits with respect to the traditional EVM. We asked the following research questions during this case study in accordance with these objectives:

RQ1: How/If does CMOD depict project`s current status and estimate project`s future?
RQ2: What are the benefits and difficulties of CMOD for software projects?

Initially, we applied the model on two projects [45]. These projects were applying different software development methodologies, which are waterfall and iterative. In waterfall project, the change factor measured around 30% and in iterative one it was around 20%. After these first trials of the model have been found successful, we have extended its application in new domains by means of three additional projects. Furthermore, we have basically improved definitions of CMOD elements based on the feedbacks and our insights. As a result, we conducted case studies on five different projects from five different organizations. During case study design, our case selection strategy was to select cases utilizing different software development methodologies, from different business domains, with different sizes and from different organizations to observe CMOD applicability on different circumstances. In addition, we selected organizations that track issues and record effort data as these are essential parameters for CMOD calculations.

The dataset for the case studies basically include all project documents related with project planning and monitoring. We used written documents and semi-structured interview methods to collect data. The documents were mainly retrieved from project management and issue tracking tools. For each case study, we initially contacted the project managers via e-mail. In a first semi-structured interview, we explained

Table 3
CMOD performance metrics.

Metric	Equation
Estimated Schedule Variance (SVest)	EVest – PV
Estimated Schedule Performance Index (SPIest)	EVest/PV
Estimated Cost Variance (CVest)	EVest – TC
Estimated Cost Performance Index (CPIest)	EVest / TC
Estimated Variance at Completion (VACest)	BAC – EACest
Estimate to Complete - Estimated (ETCest)	(BAC – EVest) / CPIest
Estimate at Completion - Estimated (EACest)	TC + ((BAC – EVest) / CPIest), TC + ETCest, BAC / CPIest
Estimated Total Reworking and Evolution Cost (ETREC)	cf * (BAC – EVest) + REC
Estimate to Complete Reworking and Evolution Cost (TCRECest)	cf *(BAC – EVest)

the study, obtained general info about the project, and described the further needs. In two weeks, the project managers delivered us the project documents, including the project plans, progress reports, and issue/error reports as we requested. All the necessary data for the EVM and CMOD applications were gathered from these documents by the authors of this paper.

Direct observation of the authors has been utilized for data analysis. The progress reports involve planned effort, planned start date, planned end date, effort, completion%, actual start date, actual end date per work item. The issue/error reports contain the following fields: issue no, priority, effort (hour), detection date, detected version, fixed date, fixed version, related work item, issue type. The issues were categorized mainly according to related work item and fixed date and related efforts were collected from the documents. We resolved the conflicts and got more project details by means of second semi-structured interview in a face-to-face meeting. In these meetings, we spent long hours for the missing information, especially for effort data of issues.

Our strategy for validation was to compare the results of CMOD calculations and the EVM calculations with the actual project data from the documents. For the analysis of current estimates, we did not use any statistical methods. The CMOD calculations and comparisons with the EVM calculations and actual data are the indicators of the CMOD performance. For future estimates, we utilized a statistical method, *MMRE* (Mean Magnitude of Relative Error), since it is the most commonly used measure of the average estimation accuracy [46]. We estimated the *EAC* with both EVM and CMOD and compare their results using *MMRE,* mean of the *MRE*'s. *MRE* of each estimate is defined as:

$$\text{MRE} = |\textit{total cost} - \textit{estimate at completion}|/\textit{total cost}$$

The usability of the model increase as *MMRE* gets close to zero. Low *MMRE* indicates that the model is successful and the models with *MMRE* values lower than 0.25 is accepted as an applicable estimation method [46]. Our evaluation criteria were consistent with that. For validation purpose, we calculated *MRE* for *EAC* of every iteration both by EVM and CMOD. Therefore, we analyzed how the methods predict the future in comparison to actual data.

4.1. Characteristics of the cases

The case projects are selected from five organizations and the brief info about the projects and organizations are given in this section.

Organization-A develops Telecommunication Systems. It is CMMI-L3 compliant in internal assessment but has no Project Management Office (PMO). Organizational development standards are utilized in their development processes. Project 1 is a development of Sales Management Product. The project team is co-located internationally and includes 8 engineers in total. Organization- specific tool is used in development. The iterative and incremental methodology is followed. MS Excel is the tool for project management and internal ticketing system is utilized for issue tracking.

Organization-B specializes in E-government Systems. It does not have PMO but uses PMI standards and Scrum in the projects. Project 2 is a maintenance project of a Web-Based Procurement Tool. The team with 6 engineers follows iterative methodology combined with Scrum practices and uses .Net technologies. The tools are MS Project and MS Team Foundation Server for project management and Microsoft Team Foundation Server for issue tracking.

Table 4
EVM and CMOD application overview.

	Case1 - Proj1	Case2 - Proj2	Case3 - Proj3	Case4 - Proj4	Case5 - Proj5
PV	7230	4162	1117	5984	12,843
AC	7600	4553	1129	7016	17,523
REC	1080	955	823	3204	5535
TC	8680	5508	1952	10,240	23,058
cf	0.14	0.21	0.73	0.46	0.32
ftce	0.86	0.79	0.27	0.54	0.68
MMRE	0.09	0.17	0.42	0.34	0.22
MMRE–CMOD	0.03	0.04	0.11	0.16	0.08

Organization-C is an expert on Software Development and Consultancy Services in Defense industry. It is CMMI-Level3 and has PMO. It uses PMI Standards, ANSI/IEEE 1042, IEC/ISO 15,846, MIL-STD 498 and MIL-STD-973 in the projects. Project 3 is an Electronic Flight Bag System development. The team involves 7 part-time engineers and follows waterfall methodology. Redmine is used for project management and issue tracking.

Organization-D develops Information and Communication Technologies. There is no PMO in the organization but PMI standards are utilized with a few agile practices. Project 4 is the development of Innovation Management System. The team with 10 engineers uses iterative and incremental approach and Java technologies. The tools for project management and issue tracking are MS Project, MS Excel, Atlassian Jira.

Organization-E is a CMMI-L3 Consultancy and Systems Integration Services company, which has a dedicated PMO. PMI standards are followed. Project 5 is a Command and Control System. The technologies are Java and TCL/TK scripting and methodology is Waterfall combined with iterative. The team includes 15 engineers. MS Project, MS Excel, and Bugzilla are the tools for project management and issue tracking.

4.2. Applications of CMOD

EVM and CMOD were applied consecutively for each project by one of the authors of the paper and reviewed by the other one. EVM and CMOD have been applied for monthly periods (4 weeks) based on the available data. The number of EVM and CMOD application periods of the projects has been changing between 4 and 10 iterations. The planned effort is between 1117 and 12,843 person-hours while total effort spent varies between 1952 and 23,058 person-hours. Table 4 depicts the summary of the results.

4.3. Results and discussion

Detailed results regarding EVM and CMOD applications are given in Table 5. Due to space restrictions, only selected measures are included here. At first glance, *MMRE* values, which changes from 0.03 to 0.16, depict that CMOD is an appropriate estimation method. It also achieves better future estimates comparing to EVM for all cases.

The graphs for EVM and CMOD applications of Project I are given in Fig. 2.

Reflecting change efforts, CMOD signifies all the costs spent more accurately in the second graph and so presents higher total cost and cost variance, *CV* and lower cost effectiveness, *CPI* (see Fig. 2.). The progress seems more optimistic in the first graph since PV, AC and EV values are rather close and it does not give much clue about the possible cost overruns and delays that is seen better in the difference between PV, TC and EVest.

The first project has comparatively lower rework and evolution efforts, with%14 *cf*. The *cf* values in time for project I, II and III are depicted in Fig. 3. The change trend for Project I during project execution is quite linear. Based on the *MRE* values in Table 5, CMOD predicts completion budget 3 times better than EVM on average with *MMRE* values 0.03 vs. 0.09. After change trends observed in the 3rd period, it almost predicts precisely.

The EVM and CMOD graphs for the second project are given in Fig. 4. CMOD depicts project status more realistically by underlining the gap between *PV, EVest* and *TC*. This project has similar change trends with Project I but has some fluctuations (see Fig. 3). The total change ratio is 21%. CMOD spots cost overruns much better as depicted in Fig. 4 and estimates completion to budget 4 times better on average with MMRE values 0.04 vs. 0.17. Starting from the early phases, CMOD

Table 5
Case study results.

Case 1															
	PV	AC	EV	REC	cf	EVest	TC	SPI	SPIest	CPI	CPIest	EAC	EACest	MRE	MREest
1	904	963	884	0	–	884	963	0.98	0.98	0.92	0.92	7876	7876	0.09	0.09
2	1660	1734	1600	106	0.11	1529	1840	0.96	0.92	0.92	0.83	7836	8698	0.10	0.0021
3	2527	2513	2307	220	0.13	2226	2733	0.91	0.88	0.92	0.81	7876	8875	0.09	0.02
4	3194	3332	3089	341	0.14	2998	3673	0.97	0.94	0.93	0.82	7799	8857	0.10	0.02
5	4034	4176	3874	444	0.13	3782	4620	0.96	0.94	0.93	0.82	7794	8832	0.10	0.02
6	4981	5198	4783	594	0.14	4666	5792	0.96	0.94	0.92	0.81	7857	8975	0.09	0.03
7	5915	6253	5676	704	0.14	5562	6957	0.96	0.94	0.91	0.80	7965	9044	0.08	0.04
8	6600	6957	6327	848	0.14	6251	7805	0.96	0.95	0.91	0.80	7950	9028	0.08	0.04
9	7230	7600	6877	979	0.14	6805	8579	0.95	0.94	0.90	0.79	7990	9114	0.08	0.05
				1080	**0.14**		**8680**						**MMRE**	**0.09**	**0.03**
Case 2															
1	101	96	86	0	–	86	96	0.85	0.85	0.90	0.90	4646	4646	0.16	0.16
2	750	831	752	25	0.26	615	856	1.00	0.82	0.90	0.72	4599	5797	0.16	0.05
3	1619	1712	1560	93	0.11	1479	1805	0.96	0.91	0.91	0.82	4568	5079	0.17	0.08
4	2207	2392	2185	341	0.20	2082	2733	0.99	0.94	0.91	0.76	4556	5464	0.17	0.01
5	2558	2814	2537	418	0.17	2480	3232	0.99	0.97	0.90	0.77	4616	5423	0.16	0.02
6	2970	3179	2878	491	0.17	2829	3670	0.97	0.95	0.91	0.77	4597	5399	0.17	0.02
7	3203	3480	3160	550	0.17	3120	4030	0.99	0.97	0.91	0.77	4583	5376	0.17	0.02
8	3460	3716	3378	579	0.17	3347	4295	0.98	0.97	0.91	0.78	4578	5340	0.17	0.03
9	3703	4036	3667	639	0.17	3624	4675	0.99	0.98	0.91	0.78	4581	5369	0.17	0.03
10	4162	4553	4151	758	0.19	4076	5311	1.00	0.98	0.91	0.77	4565	5422	0.17	0.02
				955	**0.21**		**5508**						**MMRE**	**0.17**	**0.04**
Case 3															
1	174	167	162	82	0.49	162	249	0.93	0.93	0.97	0.65	1151	1717	0.41	0.12
2	253	242	237	104	0.62	209	346	0.94	0.83	0.98	0.60	1141	1851	0.42	0.05
3	357	325	321	121	0.50	294	446	0.90	0.82	0.99	0.66	1131	1696	0.42	0.13
4	605	526	551	179	0.55	476	705	0.91	0.79	1.05	0.68	1066	1654	0.45	0.15
5	735	656	665	231	0.44	625	887	0.90	0.85	1.01	0.70	1102	1586	0.44	0.19
6	797	702	717	278	0.42	703	980	0.90	0.88	1.02	0.72	1094	1557	0.44	0.20
7	903	831	808	390	0.56	763	1221	0.89	0.85	0.97	0.63	1149	1787	0.41	0.08
8	996	914	901	456	0.55	872	1370	0.90	0.88	0.99	0.64	1133	1755	0.42	0.10
9	1025	937	923	612	0.67	914	1549	0.90	0.89	0.99	0.59	1134	1893	0.42	0.03
10	1117	1009	1001	666	0.71	971	1675	0.90	0.87	0.99	0.58	1126	1926	0.42	0.01
		1129		**823**	**0.73**		**1952**						**MMRE**	**0.42**	**0.11**
Case 4															
1	1304	1280	1240	0	–	1240	1280	0.95	0.95	0.97	0.97	6177	6177	0.40	0.40
2	2544	2760	2464	296	0.23	2216	3056	0.97	0.87	0.89	0.73	6703	8253	0.35	0.19
3	3824	4240	3624	696	0.25	3369	4936	0.95	0.88	0.85	0.68	7001	8767	0.32	0.14
4	4984	5480	4824	1128	0.27	4595	6608	0.97	0.92	0.88	0.70	6798	8606	0.34	0.16
5	5984	6556	5824	1648	0.30	5603	8204	0.97	0.94	0.89	0.68	6736	8762	0.34	0.14
6-l[*]		6936	5904	2408	0.37	5817	9344	0.99	0.97	0.85	0.62	7030	9612	0.31	0.06
7-l[*]		**7016**	5984	3224	0.46	5962	10,240	1.00	1.00	0.85	0.58	7016	10,277		0.003
				3224	**0.46**		**10,240**						**MMRE**	**0.34**	**0.16**
Case 5															
1	2034	2340	1836	0	–	1836	2340	0.90	0.90	0.78	0.78	16,369	16,369	0.29	0.29
2	4284	5490	3726	45	0.02	3686	5535	0.87	0.86	0.68	0.67	18,923	19,287	0.18	0.16
3	6174	7740	5436	684	0.12	5261	8424	0.88	0.85	0.70	0.62	18,286	20,565	0.21	0.11
4	7974	10,053	6984	1242	0.16	6762	11,295	0.88	0.85	0.69	0.60	18,487	21,453	0.20	0.07
5	10,134	12,618	8829	2196	0.22	8507	14,814	0.87	0.84	0.70	0.57	18,355	22,364	0.20	0.03
6	12,474	15,210	10,719	3249	0.26	10,345	18,459	0.86	0.83	0.70	0.56	18,224	22,916	0.21	0.01
7	12,843	15,705	11,088	4077	0.27	11,014	19,782	0.86	0.86	0.71	0.56	18,191	23,067	0.21	0.0004
8-l[*]		**17,523**	12,843	4698	0.30	12,536	22,221	1.00	0.98	0.73	0.56	17,523	22,765	0.24	0.01
				5535	**0.32**		**23,058**						**MMRE**	**0.22**	**0.08**

[*] these periods shown by "-l" are late phases. They are not planned at the beginning but executed due to delay. For that reason, there is no PV values assigned.

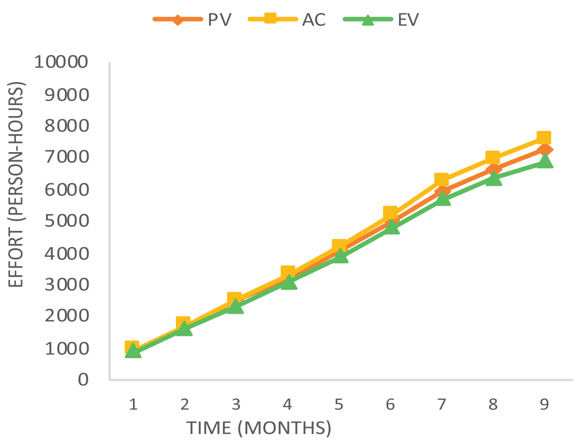
makes cost and schedule variance more visible. The progress seems more optimistic than reality in EVM results, and the minor delay happened at the end is observed better in CMOD results in Fig. 4.

The third project has significant rework and evolution efforts, with %73 *cf* at the end as seen in Fig. 3. The following EVM graph in Fig. 5., and EVM metrics in estimations do not reveal any concern about the progress other than minor cost problem with 0.9 *CPI*. CMOD again reflects higher cost issue, with *CPI* changing between 0.58 and 0.72 and highlights rework and evolution costs much better. In addition, the change trend is not linear, decreasing and increasing with time (see Fig. 3). CMOD predicts completion budget almost 4 times better on average with *MMRE* values 0.11 vs 0.42.

The *cf* values for Project IV and V are depicted in Fig. 6.

The application result for Project IV is given in Fig. 7. EVM graph and metrics do not spot any major issue other than a little cost overrun but CMOD metrics warn the project manager about the progress with high total cost, significant *cf* between 0.23 and 0.46, high expected *REC,* increasing from 296 to 3224 person-hours and rather low *CPI,* decreasing 0.97–0.58.

As seen in Fig. 6, there are significant reworking and evolution cost and cost overruns for the project, with 46% cf. The future predictions of CMOD are worse than the other cases with 0.16 *MMRE* but it can still be



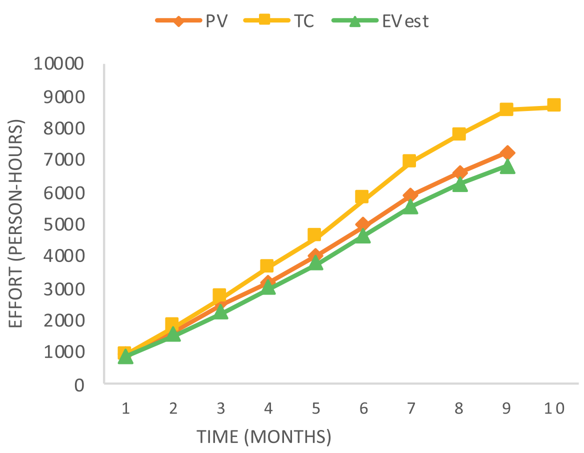


Fig. 2. Project I - EVM and CMOD application results.

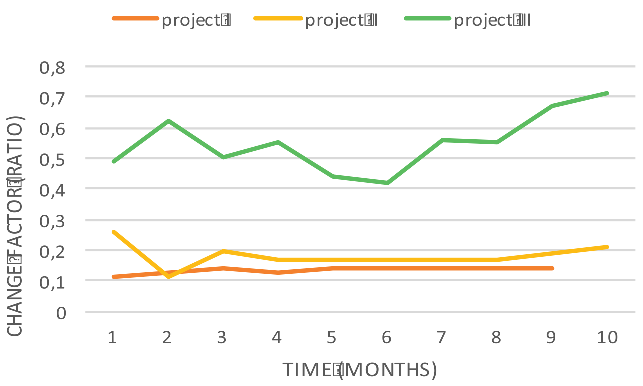


Fig. 3. Change trends for Project I, Project II and Project III.

considered acceptable as it is below 0.25 and it is 2 times better than EVM based prediction, with 0.34 *MMRE*. Since the trend of change is not linear and cf is increasing during project execution, the prediction is not very precise. This project was planned for 5 months but it is completed in 7 months with almost 40% delay in the schedule.

Project V has 32% change ratio, with the similar trend as of the 4th case (see Fig. 6). It has planned to be completed in 7 months but actually completed in close to 9 months. In the graphs in Fig. 8, CMOD results spot serious cost problems better. Also, the potential delay in the schedule is seen in CMOD much clearer. The future estimates are almost 3 times better than EVM with *MMRE* values 0.08 vs 0.22.

Under the light of this results, we can answer research questions by considering all the cases as follows:

RQ1: How/If does CMOD depict project`s current status and estimate project's future?

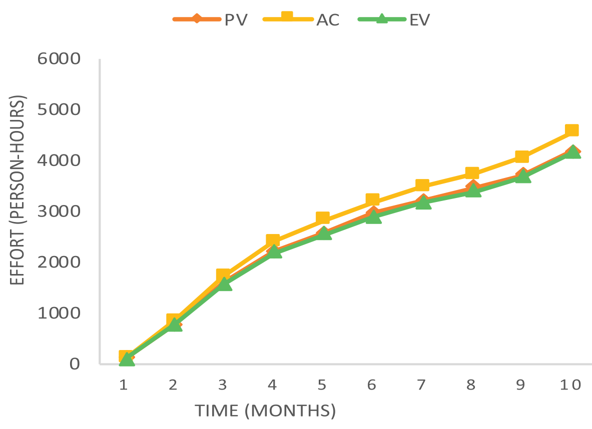
Based on the case studies we observe that the project status is better visualized by CMOD with respect to EVM from precision, effort and cost tracking, as well as the end of the project estimation perspectives. CMOD helps project managers to visualize the project's current status clearly by making change costs visible and to estimate project future more accurately.

Incorporating change costs into the project tracking results in the more accurate estimation of actual costs. With more accurate estimations of actual costs and the calibrated earned value, it is apparent that CMOD provides much better evaluation of the budget versus the value. As depicted in the results of case studies, *CPI* values are significantly better comparing to the ones that are calculated with traditional EVM methods, particularly for the cases with high rework and evolution effort. Schedule evaluations are also better due to better *EV* estimation but are not dramatically improved by CMOD.

Also, CMOD returned much better future estimates in the case studies. As seen in the results, CMOD estimates the completion budget much better than EVM.

RQ2: What are the benefits and difficulties of CMOD for software projects comparing to EVM?

We can group the main benefits that CMOD provides under three categories: visibility, accuracy and predictability.



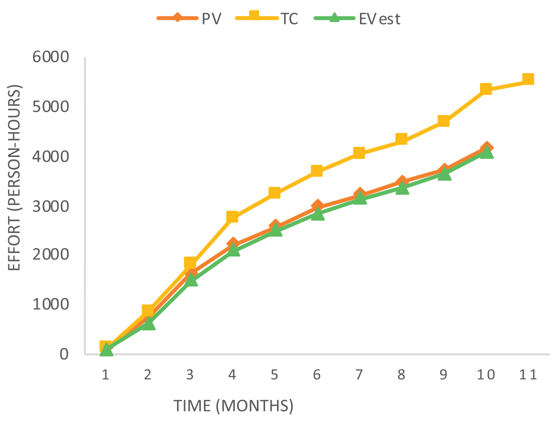


Fig. 4. Project II - EVM and CMOD application results.

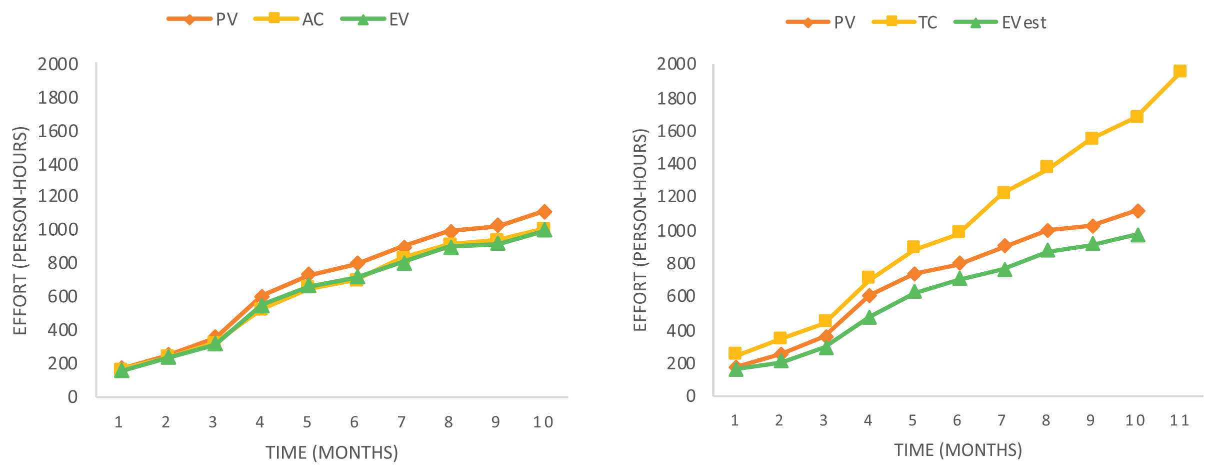


Fig. 5. Project III - EVM and CMOD application results.

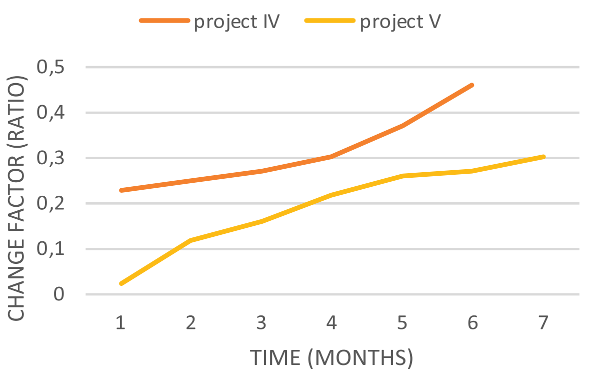


Fig. 6. Change trends for Project IV and Project V.

CMOD helps to increase project visibility by means of revealing hidden but huge rework and evolution cost. Frequently in software projects, the rework and evolution including failures and changes are in general recorded in issue tracking tools but their administration is not integrated into project management. At best, they are treated as quality issues and just kept as quality metrics. As such, the focus shifts to the number of issues rather than effort spent to resolve them. So, it may not be easy to realize the amount of rework and evolution effort spent on the project. CMOD makes those efforts explicit, integrates them into the project planning and monitoring since they significantly affect the progress and future of the project. As a result, the total effort and cost spent for change become visible.

Increased visibility brings increased accuracy for the projects. Increasing the visibility of change especially affects the calculations of the total costs and the earned value. Including rework and evolution costs into actual cost, the project progress is measured more accurately. The case study results demonstrated the superiority of CMOD in accuracy of the calculations. In particular, CMOD provides significant improvements on cost performance metrics. More accurate progress metrics lead to more accurate and realistic future estimates that increase the predictability. As seen in the case studies, CMOD provides more accurate actual cost, more accurate performance metrics and more accurate future estimates.

Predictability is also strongly related with the accuracy and visibility. As the visibility and accuracy increase for the current phase, the predictions for the future improves as well. Based on the accurate project progress and performance metrics, CMOD provides more realistic cost forecasts. Furthermore, we observed that during project execution, *MRE* improves significantly. CMOD also offers reworking and evolution cost related predictions to estimate whether more change will happen and the cost of these failures.

We also observed that *cf*, as defined in CMOD by itself, is a simple and an effective indicator of change. The project managers could track *cf* trends during project as in Figs. 3 and 6 and make root cause analysis when required. The project manager could utilize *cf* in managing contingency budget. The organizational or industrial *cf* database would be

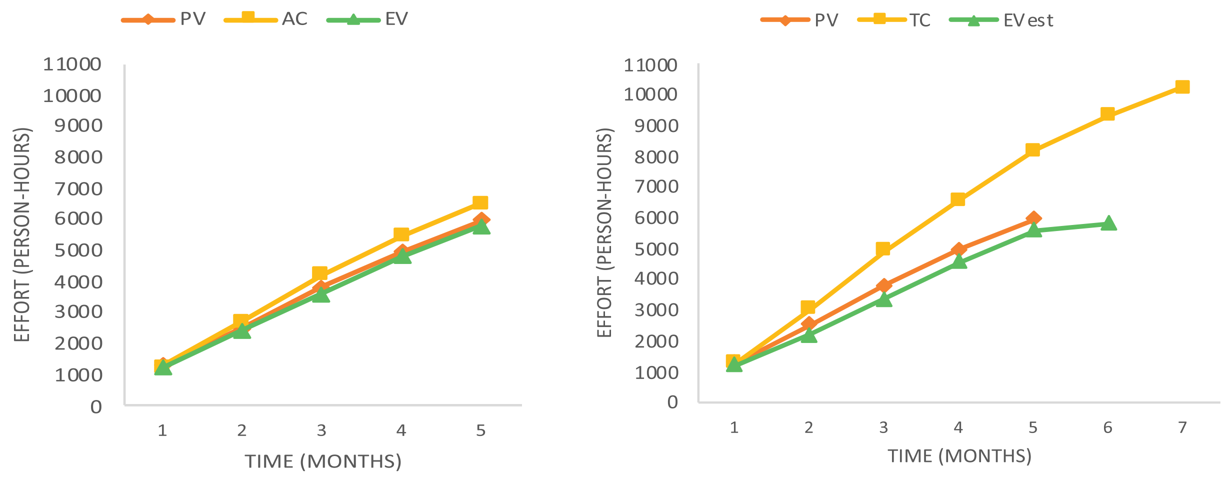


Fig. 7. Project IV - EVM and CMOD application results.

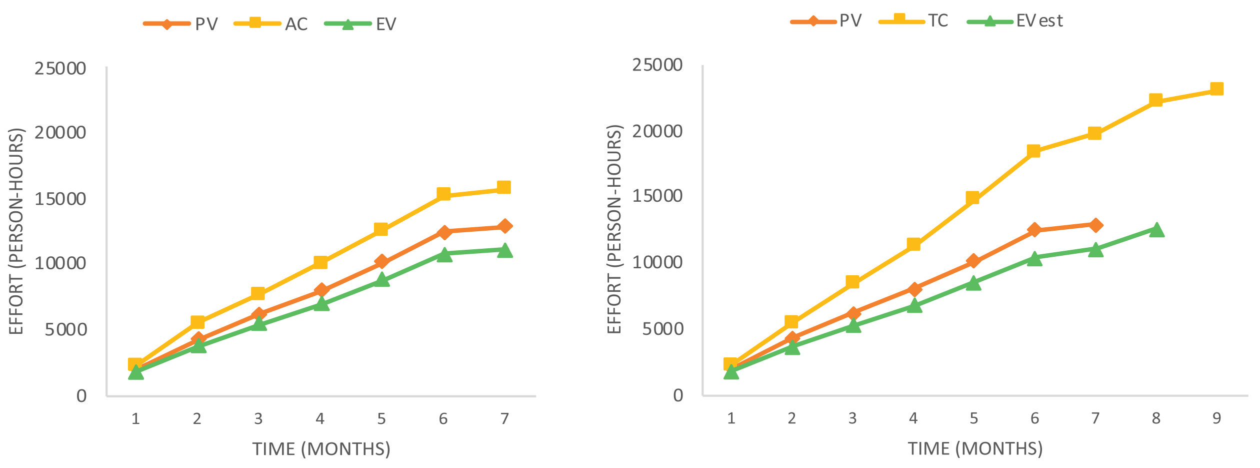


Fig. 8. Project V - EVM and CMOD application results.

useful in estimating and planning of contingency budget as well.

The most significant difficulty we observed is the availability and the validity of the change and related effort data. For many software organizations, the effort data is not collected and tracked properly [47]. Organizations to adopt CMOD need to collect effort data and make issue tracking more systematic. Especially, in the organizations where team members work on the variety of task per work period, precise effort collection can be a real challenge. In most cases, although such data exist, they are not as reliable as we need it to be. Issue tracking is not as challenging since most software organizations use specialized tools for this purpose.

4.4. Validity threats

The main threats to internal and external validity of this study have been determined and mitigatory actions have been identified at the beginning of the study.

The major threat to internal validity is the quality and reliability of the data provided and collected from the organizations. To improve the reliability of the data, we selected projects from organizations with well-defined data collection practices. We considered organizations with higher maturity levels, CMMI-3 or above or having PMO in the organization as an evidence for the data quality. Furthermore, in order to increase the reliability of the data collected, at the beginning of the study, it has been explained in detail to the project managers, the templates for the related data have been provided and the needs have been communicated. For all case studies, after data provided, additional interviews have been conducted with the project manager to clarify the identified issues with the data and records provided. The issue lists including errors and changes have been revisited during these semi-structured interviews.

Additional threat to internal validity is the data analysis procedure. We have chosen MRE for analysis since it is objective and so allows comparing the results of EVM and CMOD. Both EVM and CMOD enable future predictions during project execution by means of EAC. We already collected actual cost data from the project documents. Therefore, MRE based on EAC and realized cost data ensures the reliable results.

The main threat to external validity is the representation of the setting. To mitigate this threat, we have conducted case studies in organizations with different characteristics. The organizations are selected from different business domains including Telecommunications, Government, ICT, Defense, Software Development and Consultancy. Additionally, we preferred the projects with different development methodologies, which are waterfall, iterative and incremental, waterfall combined with Scrum practices and in different sizes to increase the diversity.

5. Conclusions and future work

In the scope of study, we presented a change management model, called CMOD, to monitor project progress better. Software projects are subject to change due to various reasons like its essential characteristics, technical challenges, changing customer needs. In this study, we accept change as a reality of software projects and develop a method for managing the change rather than preventing or avoiding. We know change will happen regularly in the life cycle of a software project for several reasons. We make it visible, measure it, observe its trend and adapt our project controlling metrics accordingly.

CMOD introduces new concepts, metrics and calculations by extending traditional EVM. To explore its usage, we applied CMOD on five projects from industry and depicted and discussed the results. Based on the case study results, the main benefits that CMOD provides can be summarized as follows:

- Revealing hidden change related costs and integrating them into project and performance management
- Measuring the change status of a project at a given time in addition to schedule and cost
- Estimating the project progress more accurately in comparison to traditional methods at any given time by including change data
- Estimating project future more realistically in comparison to traditional methods

Fundamentally, CMOD underlines rework and evolution efforts and makes them visible. Therefore, using CMOD project managers can obtain more visible and correct cost data and so calculate more accurate performance metrics. Without change costs, the progress might be perceived more optimistically than the reality dictates. If there is no distinct planning for rework and evolution in the project, the project team could easily be overloaded which cannot easily be perceived by traditional methods. Revealing the costs spent for change provides the visibility on the actual costs of the project as well as the performance of the project team. CMOD presents more precise cost indices and schedule indicators are also better than those of EVM.

CMOD offers change related metrics in addition to rework and evolution costs. Tracking project change status at any time during project execution by means of the change factor, *cf*, provides valuable information about the change status of the project. If it gets unexpectedly high during the course of a project, the project manager could investigate the reasons and try to find out the root cause. There might be several possible reasons for change as indicated by the high *cf* value. The initial estimations and planned values might be wrong or underestimated and so the tasks might be implemented in a chaotic way. The team might be underqualified and/or forced to perform the tasks in less time than the estimated duration. The customer requirements might change frequently. The project manager could take necessary actions after having identified the problem. Furthermore, the project manager monitors how much of the work is accomplished right the first time by means of the ftce. That enables calculation and visibility for the cost of quality.

CMOD delivers more accurate forecasts and more precise future predictions comparing to traditional EVM based on the case studies we performed. Including rework and evolution costs into total cost increases the visibility of the project costs. The revealed reworking and evolution costs result in more accurate total cost, schedule and cost indices. Better current progress and performance metrics enable more accurate future estimations. In the case studies, the key future estimate, *EAC*, is calculated for every phase and the results demonstrated significant improvement in the cost forecasts. The accurate progress information and forecasts are the main targets of project management since they allow project managers to visualize the present status clearly.

CMOD relies on accurate change data, and the key to success for organizations applying CMOD is adopting the change management process accordingly.

For future work, we plan to improve CMOD in several ways. Although the forecasts of CMOD give better estimations than EVM, there is still a room for improvement. Currently, we calculate rework and evolution trends linearly and take the latest one into consideration as EVM does. However, the behavior of the rework and evolution could change depending on the project characteristics. The exponentially increasing trend of change is not well captured in CMOD. We plan to study various prediction models for better *cf* estimation. For the projects having exponential change trends, it would give more accurate results to estimate *cf* value by means of exponential distribution functions. The different types of prediction models could be applied and their usability as well as applicability could be discussed according to the results.

We also plan to perform further case studies to better understand the usability of the model for the projects with different characteristics (e.g., large-scale projects, complex algorithmic systems, micro services, embedded systems). Additionally, we plan to apply CMOD on a software subsystem that is part of a larger deliverable consisting mechanical, electronic, structural and procedural deliverables as well as software deliverables to examine the usefulness of CMOD and to realize how schedule and cost integration occurs at higher levels. Furthermore, new case studies on ongoing projects can provide insight to improve the model by reducing the difficulties regarding data collection and application of CMOD during project execution should be observed for better insight.

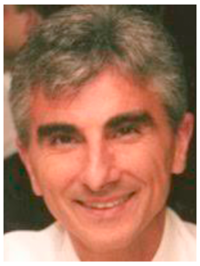
In summary, CMOD delivers significant benefits to software project monitoring by incorporating rework and evolution costs. CMOD provides significant improvements over traditional EVM in projects with change by measuring the progress clearly and estimating the future correctly. Change is inevitable for software projects and so CMOD embraces it.

References

[1] Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK® Guide), sixth ed., Project Management Institute, Newtown Square, PA, USA, 2017.
[2] A. Tarhan, O. Demirors, Apply Quantitative Management Now, IEEE Softw. 29 (3) (May-June 2012) 77–85.
[3] A. Uskarci, O. Demirors, Do staged maturity models result in organization-wide continuous process improvement? Insight from employees, Comput. Stand. Interfaces 52 (2017) 25–40.
[4] O. Tanriover, O. Demiros, A process capability based assessment model for software workforce in emergent software organizations, Comput. Stand. Interfaces 37 (2015) 29–40.
[5] M. Sulayman, E. Mendes, An extended systematic review of software process improvement in small and medium web companies, Proceedings of the Conference on Evaluation and Assessment in Software Engineering, London, 2011.
[6] A. Cass, S.M. Sutton Jr, L.J. Osterweil, Formalizing rework in software processes, Software Process Technology, Springer Berlin, Heidelberg, 2003, pp. 16–31.
[7] B. Boehm, C. Papaccio, Understanding and controlling software costs, IEEE Trans. Softw. Eng. 14 (10) (1988) 1462–1477.
[8] J. Twentyman, The crippling costs of IT project rework, Inside Knowledge (15 Jun 2005).
[9] R.N. Charette, Why Software Fails, IEEE Spectr. 42 (9) (2005) 42–49.
[10] A.M. Davis, E.H. Bersoff, E.R. Comer, A strategy for comparing alternative software development life cycle models, IEEE Trans. Softw. Eng. 14 (10) (Oct. 1988) 1453–1461.
[11] Agile Manifesto, 2001, [Online]. Available:http://agilemanifesto.org[Accessed 01 03 2018].
[12] VersionOne, "11th Annual State of Agile", https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2, 2017.
[13] S.W. Ambler, M. Lines, Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise, IBM Press, 2012.
[14] Serrador, J.K. Pinto, Does agile work? - a quantitative analysis of agile project success, Int. J. Project Manage. 33 (5) (2015) 1040–1051.
[15] V. Garousia, A. Coşkunçay, A. Betin-Can, O. Demirors, A survey of software engineering practices in Turkey, J. Syst. Softw. 108 (2015) 148–177.
[16] O. Ozcan-Top, O. Demirors, Application of a software agility assessment model - agilitymod in the field, Comput. Stand. Interfaces 62 (2019) 1–16.
[17] P. Efe, O. Demirors, Applying EVM in a software company: benefits and difficulties, Proceedings of the 39th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), Santander, 2013.
[18] D.S. Christensen, D.V. Ferens, Using earned value on software development projects, Acquis. Rev. Q. 2 (1995) 155–171.
[19] D. Brandon, Integrating earned value into the management of software development projects, Proceedings of the IRMA International Conference, Hersey, PA, USA, 1999.
[20] P.J. Solomon, Practical software measurement, performance-based earned value, CrossTalk (2001) September.
[21] M. Ferle, Implementing earned value management on IT projects, 19th International Cost Engineering Congress, Ljubljana, Slovenia, 2006.
[22] R.A. Hanna, Earned value management software projects, Proceedings of the 3rd IEEE International Conference on Space Mission Challenges for Information Technology, Washington, DC, 2009.
[23] F. Anbari, Earned value project management method and extensions, Project Manag. J. 34 (4) (2003) 12–13.
[24] ANSI/EIA -748A: American National Standard Institute / Electronic Industries Alliance/ Standard for Earned Value Management Systems, 1998.
[25] AS 4817-2006: Project performance measurement using Earned Value, 2006.
[26] Project Management Institute, Practice Standard for Earned Value Management, Newtown Square, Project Management Institute, PA, USA, 2005.
[27] Q.W. Koppelman, J.M. Fleming, Earned value project management: a powerful tool for software projects, STSC CrossTalk 4 (1998) 19–23.
[28] P.J. Solomon, R. Young, Performance-Based Earned Value, Wiley & Sons, Hoboken, NJ, 2007.
[29] P.J. Solomon, Basing Earned Value on Technical Performance, CrossTalk 25-28 (January/February) (2013) 25–28.
[30] A. Cabri, M. Griffiths, Earned value and agile reporting, AGILE'06: Proceedings of the Conference on AGILE 2006, Washington, DC, 2006.
[31] S.H. Lett, An earned value tracking system for self-directed software teams, Proceedings of European SEPG, 98 1998.
[32] A. Cockburn, Crystal Clear: A Human-Powered Methodology for Small Teams, Pearson Education Inc, Upper Saddle River, NJ, USA, 2005.
[33] T. Sulaiman, B. Barton, T. Blackburn, AgileEVM – earned value management in scrum projects, Proceedings of AGILE 2006 Conference, Minneapolis, Minnesota, USA, 2006.
[34] G.B. Alleman, Project Management = Herding Cats, Agile Project Management, PMFORUM, 2003 A Field Report.
[35] G.B. Alleman, M. Henderson, R. Seggelke, Making agile development work in a government contracting environment. Measuring velocity with earned value, Proceedings of the Agile Development Conference, Salt Lake City, Utah, 2003.
[36] J. Rusk, Earned value for agile development, DoD Software Tech News 12 (1) (2009).
[37] 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].
[38] P. Solanki, Earned Value Management: Integrated View of Cost and Schedule Performance, Global India Publications Pvt Ltd., New Delhi, 2011.
[39] H. Raju, Y. Krishnegowda, Software Sizing and Productivity with Function Points, Lecture Notes Softw. Eng. 1 (2) (2013) 204–208.
[40] W. Roetzheim, Incorporating function points into earned value management, 23rd Systems and Software Technology Conference (SSTC), Salt Lake City, 2011.
[41] S. Di Martino, F. Ferrucci, C. Gravino, F. Sarro, Web effort estimation: function point analysis vs. COSMIC, Inf. Softw. Technol. 72 (2016) 90–109.
[42] J. Li, Z. Ma, H. Dong, Monitoring software projects with earned value analysis and use case point, IEEE/ACIS International Conference, Portland, 2008.
[43] C.A.L. Garcia, C.M. Hirata, Integrating functional metrics, COCOMO II and earned value analysis for software projects using PMBoK, Proceedings of the 2008 ACM Symposium on Applied Computing (SAC), Fortaleza,Ceara, Brazil, 2008.
[44] G. Liu, Tracking software development progress with earned value and use case point, Proceedings of the International Workshop on Information Security and Application, Qingdao, China, 2009.
[45] P. Efe, O. Demirors, B. Benetallah, Measuring change in software projects through

an earned value lens, Proceedings of the 18th International Conference on Software Process Improvement and Capability Determination, SPICE 2018, Thessaloniki, Greece, 2018.
[46] T. Foss, E. Stensrud, B. Kitchenham, I. Myrtveit, A simulation study of the model evaluation criterion MMREF, IEEE Trans. Softw. Eng. 29 (11) (2003) 985–995.
[47] A. Özkaya, E. Ungan, O. Demirors, Common practices and problems in effort data collection in the software industry, Joint Conference of the 21st International Workshop on and 6th International Conference on Software Process and Product Measurement (IWSM-MENSURA), Nara, Japan, 2011.

P. Efe is a researcher and consultant at Bilgi Grubu Ltd. She completed her Ph.D. and M.Sc. in Information Systems in Middle East Technical University. She has managed numerous software development projects in large enterprises during her professional career. Her research interests are software project management, software measurement, human factors in software projects.



O. Demirors is a Professor of Computer Engineering at Izmir Institute of Technology and the strategy director of Bilgi Grubu Ltd. He has his Ph.D. and M.Sc. from Southern Methodist University. His current research focuses on decentralized modeling, software measurement, organizational change and software analysis and design methods. He has leaded major research and consultancy projects on developing and applying improvement and modeling techniques, on establishing and implementing organizational modeling approaches and on establishing software measurement infrastructures. He continues to teach on decentralized modeling and event based systems, software project and quality management, software measurement and innovative software development approaches.