The ISC Int'l Journal of Information Security
ISeCure
August 2019, Volume 11, Number 3 (pp. 161–172)
http://www.isecure-journal.org

SELECTED PAPER AT THE ICCMIT'19 IN VIENNA, AUSTRIA

Towards Measuring the Project Management Process During Large Scale Software System Implementation Phase☆

Wajdi Aljedaibi [1,*], and Sufian Khamis [2]

[1] *Department of Computer Science, Faculty of Computing and Information technology, King Abdulaziz University, Jeddah, Saudi Arabia*
[2] *Deanship of Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia*

ARTICLE INFO.

Keywords:
Enterprise Resource Planning Systems (ERPs), Critical Success Factors (CSF), Measurement, Goal/Question/Metric paradigm (GQM), Software Project Management.

Abstract

Project management is an important factor to accomplish the decision to implement large-scale software systems (LSS) in a successful manner. The effective project management comes into play to plan, coordinate and control such a complex project. Project management factor has been argued as one of the important Critical Success Factor (CSF), which need to be measured and monitored carefully during the implementation of Enterprise Resource Planning (ERP) systems. The goal of this article is to develop âĂIJCSF-Live!âĂİ which is a method for measuring, monitoring, and controlling critical success factors of large-scale software systems. To achieve such goal, we apply CSF-Live for the project management CSF. The CSF-Live uses the Goal/Question/Metric paradigm (GQM) to yield a flexible framework containing several metrics that we used it to develop a formulation to enable the measurement of the project management CSF. The formulation that we developed for the project management CSF implies that the significance of having proper project management when conducting an ERP system implementation, since it is positively associated with the success of the ERP.

1 Introduction

According to Dennis Lock (1996), "project management has evolved in order to plan, coordinate and control the complex and diverse activities of modern industrial and commercial projects" [1]. Typical tasks of a project manager include: completion of a project plan items within the specified time limits, actual cost equal of planned value of project phases, ensuring of the project implementation quality [2], work within the goals and activities of the organization, the ability to manage resources efficiently (e.g., equipment, software and human) [2], abilities level and proficiency, sufficient knowledge and expertise of implementation large-scale software system [3], conduct meetings with top management and project team members constantly [4]. Some of the disadvantages of a project manager are: lack of scientific knowledge and technologies [2], not organizing training for em-

☆ The ICCMIT'19 program committee effort is highly acknowledged for reviewing this paper.

* Corresponding author.
Email addresses: waljedaibi@kau.edu.sa, smkhamis@kau.edu.sa
ISSN: 2008-2045

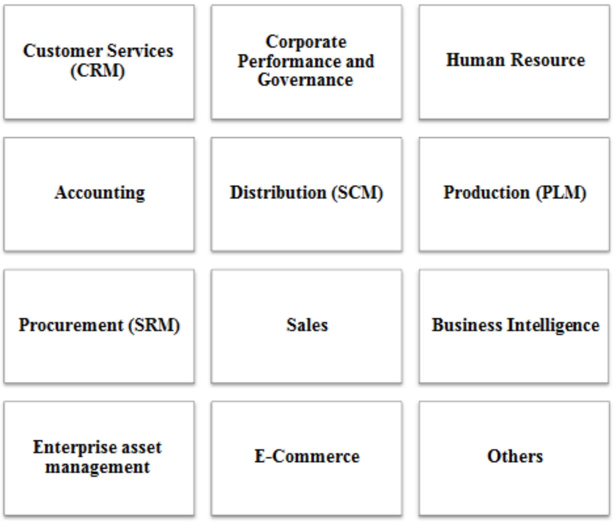

Figure 1. ERP Modules

ployees especially for those who have shortage in experience in the ERP project implementation [3]. As a result, strong leadership and cooperation are major causes for successful implementations of large-scale software systems [5].

Despite the importance of project management of software, there were no attempts to measure it using numerical values. However, it was measured using descriptive measures, e.g. high, medium and low. Here in this article, we proposed a novel quantification method based of the GQM paradigm for the project management factor. Such method provides an accurate monitoring framework for the project management factor. This paper is organized as follows: in Section 2 2 presents a short a background, while paper design and methodology is discussed in Section 3. Section 4 shows CSF-Live Method then Section 5 shows measure of project management. Conclusion are presented in Section 6.

2 Background

Large scale software systems (LSS) are characterized by huge size as well as difficulty in dealing with software engineering aspects represented by project management, requirement analysis, design, implementation, testing and maintenance [6]. Each step of the above tasks need to handle in separate and different manner through competent persons. The ERP is considered business management software that a company can use it to collect, store, manage and interpret data from various business activities covering administrative, functional, financial management, procurement and warehouses in different institutions and companies as shown in Figure 1 [7]. There are different studies and research [8–10] shed the light

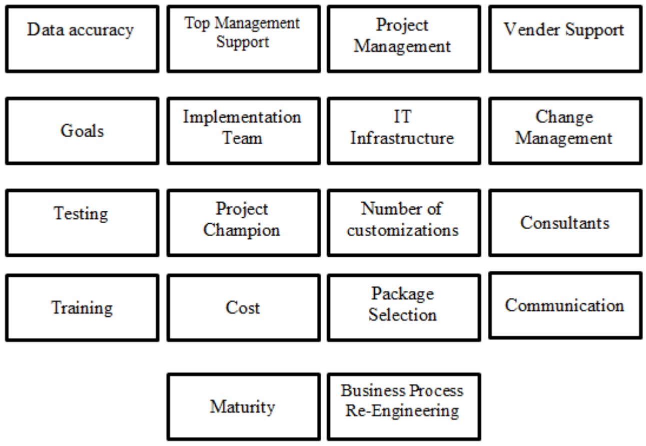

Figure 2. Set of Critical Success Factors

about and discuss how many ERP implementations have failed or faced serious delays. In [11], it was clear that there are several problems and obstacles appeared in the performance of these tasks within the ERPs. Also among the main findings of [12], it was observed that during such projects there were various factors that led to such final results and that gave rise to what is known today as the critical success factors (CSFs) of large-scale software systems. Author of [7] illustrates that there are more than 66 critical success factors , which were viewed to effect on the implementation of ERP. As shown in Figure 2, there are 18 factors effect on CSF. There were no previous attempts to measure these factors which we believe are important to assess the status of each and its subsequent impact on the success or failure of the program. Basili *et al.* [13] introduced the Goal/Question/Metric paradigm (GQM) to address measurement of some goal, which maybe an object as well, according to the following approach:

- Identification of (a) goal(s) of the project.
- Ask questions with respect how the goal can be achieved.
- Identify metrics.

GQM consists of three levels [13]:

(1) Conceptual level (Goal)
This parameter is defined as a goal for a particular object in a specific environment, using various quality models and for a variety of reasons from different view points.

(2) Operational level (Question)
It represent the usage of a set of questions determining the goal of the project as well as determining the characteristics of the evaluation or accomplish a specific goal.

(3) Quantitative level (Metric)
A set of metrics, based on the models, is associated with every question in order to answer it

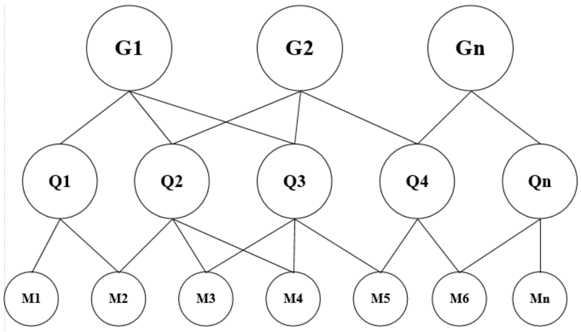

Figure 3. GQM Model Hierarchical Structure

Table 1. GQM Goal Definition Template

Analysis	Measurement object as (product, process, other experience models).
Purpose (Why)	foretelling, Description, amelioration, valuation, controlling, stimulus, realization or improving the object.
Consideration	The quality focus of the object that the measurement focuses on (fault removal, cost, modification, correctness, accuracy and user friendlinessâĂę.).
From the opinion of (Who)	The people that measure the object (client, user, administrator, developer and company âĂę.).
Environment	The environment in which measurement tasks place (issue, factors, people factors, resource factors and process factorsâĂę.).

in a measurable way.

The Goals is the top of GQM model and it is refined to many questions. Answers of these questions called "metrics". The same metric can be the answer for more than one question as shown in Figure 3. Differing viewpoints in answering some of the questions affect the determination of the metrics. Basili et.al. Described his six-step GQM process as follows [13]:

(1) Establish a set of goals and objectives for the project associated with the measurement of productivity and quality.
(2) Ask questions to define those goals clearly.
(3) Determine measurements to be collected, which will help you get answers.
(4) Develop data collection methods.
(5) Collect and validate data on time.
(6) Collect and validate data on time.

Measurement goals should be defined in an understandable way and should be clearly structure [13]. The goal is defined by filling in a set of values for the various parameters in the template, it includes purpose (what object and why), perspective (what aspect and who) and the environmental characteristics (where) see more Table 1.

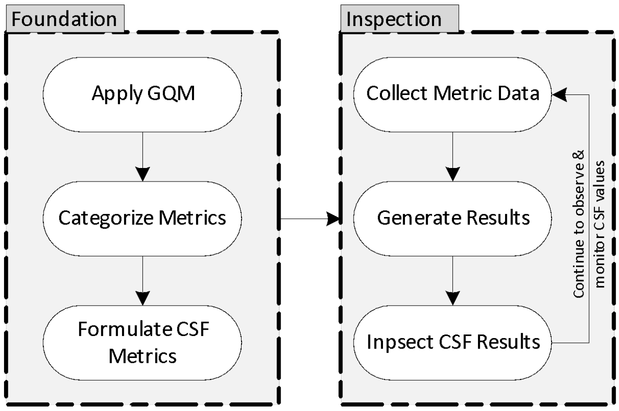

Figure 4. CSF-Live! Method

3 Paper Design & Methodology

To achieve the goals of this article, the following steps were followed which were applied on project management factor:

(1) Study of CSFs for LSS
We present a study of the previous research that focus on the CSFs for implementing LSS (e.g. ERP systems) and from which we selected project management factor of these factors to be studied in the framework
(2) Apply GQM-analysis
To measure the impact of the project management to the success/failure of the project of implementing large-scale software system, we used GQM to reach a group of metrics directly linked to project management factor to enable monitoring and observation capabilities.
(3) Measurement Formulation
Using GQM analysis, a formulation of the metric is presented as part of the measurement model for project management factor.

4 CSF-Live Method

In this work we used a method (CSF-Live) [14] that represent our proposed framework for measuring project management factor. The purpose of the CSF-Live method is to measure, track, monitor, and control the critical success factors during the implementation of LSS by using the Goal/Question/Metric (GQM) paradigm. The CSF-live method has six steps as shown in Figure 4.

5 Measure of Project Management

5.1 Project Management as a Numeric Value

Despite the importance of project management of software, there were no attempts to measure it using numerical values. However, it was measured using descriptive measures, e.g. high, medium and low [15].

Here in this article, we proposed a novel quantification method based of the GQM paradigm for the project management factor. Such method provides an accurate monitoring framework for the project management factor.

As we see from Table 2, we have formulated a goal to measure software project management and from the workshop with the graduate students and some staff at King Abdulaziz University (KAU), we generated a set of questions and metrics during the discussion which helped us to measure the goal. The generation of questions and metrics were driven by the actual formulation of the goal of the software project management. In addition, metrics must be represented numerically so that we can quantify the performance of the goal. A formulation of the derived metrics will yield a single number that represents the goal, through which goal achievement can be monitored. It should be noted, that our solution to CSF measurement does show an accurate indication of current status of a single CSF quantified numerically. Here we do not evaluate the efficiency of the project itself and number of errors but evaluate the efficiency of the project manager in how to manage the project. The project should be successful, but the efficiency of the project manager is weak and vice versa. As mentioned earlier, for data collection we did not have currently a large-scale software system, consequently, we have created hypothetical data that we summed more relevant to real projects. The following sections explain the details of each metric related to project management.

The metrics provide the means to measure the software project management. If the value of any of these metrics increased during the large-scale software system implementation, this provides a positive indication about the project management. This is applicable for all the metrics except the following metrics (Number of Unsuccessful Project Plan Items Metric, Number of Unaccomplished Tasks Metric, Number of Delay Days per Phase Metric, Budget Deficit Metric and Number of Unsolved Risks Metric). In these metrics, however, the lower value they have the positive indication they tell about the project management. The data that we created for these metrics is a hypothetical data which we assumed to be more relevant to real projects. This data represents 11 weeks such that we measure the value of metrics in each week independently until the 11th week by studying the metrics, we categorize metrics into three groups, and each group has different way to calculate measures.
Group I:

(1) # Successful / Unsuccessful Project Plan Items Metrics.
(2) # Accomplished / Unaccomplished Tasks Metrics.

Group II:

(1) # Completed Features in the New System Metric.
(2) # Completed Milestones Metric.
(3) # Reports Metric.
(4) # Weekly Meetings with Project Team Members Metric.
(5) Degree of Communication Skills Metric.

Group III:

(1) # Delay Days per Phase Metric.

Table 2. GQM for Software Project Management

Goal
To analyze project management for the purpose of evaluation with respect to effectiveness from the point view of project sponsor in the context of new large-scale software systems development.
Questions
How many successful project plan items?
How many unsuccessful project plan items?
How many tasks that have been implemented?
How many tasks that have been non-implemented?
How many new system features completed?
How many milestones completed?
How many reports completed?
How many meetings with project team members?
What is the tool used for communication?
What is the degree of communication skills?
How many delay days in the project?
What is the deficit cost in the budget?
How many mitigated risks?
How many unsolved risks?
Metrics
Successful Project Plan Items (As Prepared by The PM)
Unsuccessful Project Plan Items (As Prepared by The PM)
Accomplished Tasks
Unaccomplished Tasks
Completed Features in the New System
Completed Milestones
Reports
Weekly Meetings with Project Team Members
Degree of Communication Skills
Delay Days per Phase Budget Deficit
Solved Risks
Unsolved Risks

(2) Budget Deficit Metric.
(3) # Solved / Unsolved Risks Metrics.

Group I, we calculated the following measures:

(1) Values of each metric in each week.
(2) Average daily values of each metric in each week calculated by as values of each metric (step 1) divided by the number of working days (5 days) per week.
(3) Cumulated average daily values of each metric since project initiation which is calculated by adding the average daily values of each metric per week (step 2) to cumulated average daily values of each metric of the previous weeks since project initiation.
(4) Average daily values of each metric since project initiation calculated as the cumulated average daily values of each metric (step 3) divided by number of weeks.
(5) Average daily difference between values of metric2 and values of metric3 since project initiation (step 4). In Number of Successful / Unsuccessful Project Plan Items Metrics we called Project Completion Index (PCI) while in Number of Accomplished /Unaccomplished Tasks Metrics we called Project Accomplishment Index (PAI).

Group II, we calculated the following measures:

(1) Values of each metric in each week.
(2) Average daily values of metric in each week calculated as the values of metric (step 1) divided by the number of working days (5 days) per week.
(3) Cumulated average daily values of metric since project initiation which is calculated by adding the average daily values of metric per week (step 2) to cumulated average daily values of metric of the previous weeks since project initiation.
(4) Average daily values of metric since project initiation calculated as the cumulated average daily values of metric (step 3) divided by number of weeks. In Number of Completed Features in the New System Metric we called System Features Index (SFI), in Number of Completed Milestones Metric we called Completed Milestones Index (CMI), in Number of Reports Metric we called Report Index (RI), Number of Weekly Meetings with Project Team Members Metric we called Weekly Meetings Index (WMI) and in Degree of Communication Skills Metric we called Communication Skills Index (CSI).

Group III, each one of them has different way in calculate the measure (details in section 5.8, 5.9 and 5.10).

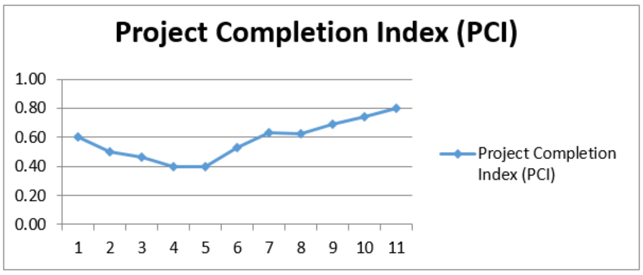

Figure 5. Project Completion Index (PCI)

5.2 Number of Successful / Unsuccessful Project Plan Items Metrics

The number of successful / unsuccessful project plan items metrics are defined as: a numerical count of the number of successful / unsuccessful project plan items while we implement the large-scale software system.

By calculating the previous measures on the assumed data that we created, we notice Project Completion Index (PCI) is important number that help to compare between the number of successful / unsuccessful project plan items because it represents average daily difference between number of successful and number of unsuccessful project plan items since project initiation as shown in Figure 5. By analyzing the number of successful / unsuccessful project plan items in over the 11 weeks that we studied, we notice the number of successful project plan items was bigger than the number of unsuccessful project plan items in each week. The largest number of PCI was in the last week. It was 0.80 because the number of successful project plan items was 7 while we had zero rejected unsuccessful project plan items. The smallest number of PCI was in the fourth, fifth weeks. They were 0.40 in each because increased unsuccessful project plan items comparison successful project plan items. Finally, the higher value for the PCI, the higher good impact on the project management of large-scale software system and vice versa.

5.3 Number of Accomplished / Unaccomplished Tasks Metrics

The number of accomplished / unaccomplished tasks metrics are defined as: a numerical count of the number of accomplished / unaccomplished tasks while we implement the large-scale software system. We notice Project Accomplishment Index (PAI) is important number that help to compare between the number of accomplished / unaccomplished tasks because it represents average daily difference between number of accomplished and number of unaccomplished tasks since project initiation as shown in Figure 6. By analyzing the number of accomplished / unaccomplished tasks in over the 11 weeks that we studied, we notice

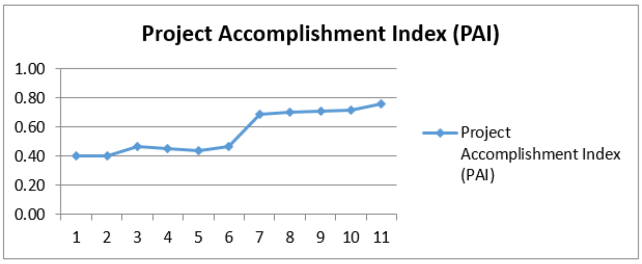

Figure 6. Project Accomplishment Index (PAI)

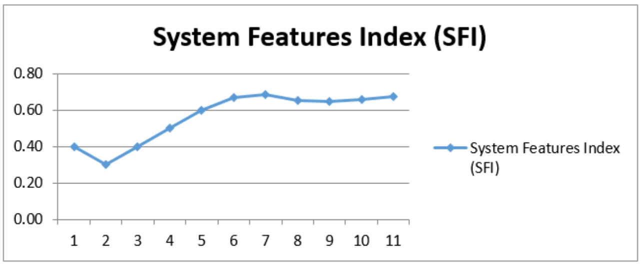

Figure 7. System Features Index (SFI)

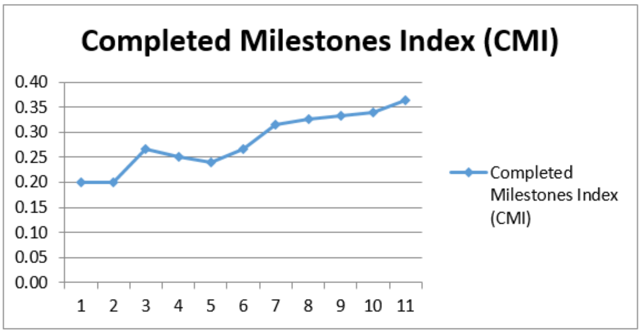

Figure 8. Completed Milestones Index (CMI)

the number of accomplished tasks was bigger than the number of unaccomplished tasks in each week. We notice that the largest number of accomplished tasks was in the seventh week (10 accomplished tasks) while number of unaccomplished tasks was zero unaccomplished tasks. As a result, this increased the PAI for the past 7 weeks to become 0.69. We note PAI was big increased from seventh to last week because big different between accomplished and unaccomplished tasks especially, last week PAI have highest number. It was 0.76 because the number of accomplished tasks was 6 while we had zero unaccomplished tasks. Finally, the higher value for the PAI, the higher good impact on the project management of large-scale software system and vice versa.

5.4 Number of Completed Features in the New System Metric

The number of completed features in the new system metric is defined as: a numerical count of the number of completed features in the new system while we implement the large-scale software system. We notice that the System Features Index (SFI) is an important number that help to measure the completed features in the new system because it represents average daily number of completed features in the new system since project initiation as shown in Figure 7. We note that the number of completed features in the new system appeared since the beginning of the first week until the last week. We notice that the largest number of completed features in the new system was in the fifth, six weeks (5 completed features in each). As a result, this increased the SFI for the past 5 and 6 weeks to become 0.60 and 0.67, respectively. The smallest number of completed features in the new system was in the second week. It was one so decreased the SFI for the past 2 weeks to become 0.30. Finally, the higher value for the SFI has the higher positive impact on the project management of large-scale software system and vice versa.

5.5 Number of Completed Milestones Metric

The number of completed milestones metric is defined as: a numerical count of the number of completed milestones while we implement the large-scale software system. We notice that the Completed Milestones Index (CMI) is an important number that help to measure the completed milestones because it represents average daily number of completed milestones since project initiation as shown in Figure 8. We note that the number of completed milestones appeared since the beginning of the first week until the last week. We notice that the largest number of completed milestones was in the seventh, eleventh week (3 completed milestones in each). As a result, this increased the CMI for the past 7 and 11 weeks to become 0.31 and 0.36, respectively. The smallest number of completed milestones was in the first and second week (1 completed milestone in each). As a result, this decreased the CMI for the past 1and 2 weeks to become 0.20 in each. Finally, the higher value for the CMI has the higher positive impact on the project management of large-scale software system and vice versa.

5.6 Number of Reports Metric

The number of reports metric is defined as: a numerical count of the number of reports delivered for project sponsor while we implement the large-scale software system. We notice that the Report Index (RI) is an important number that help to measure the reports because it represents average daily number

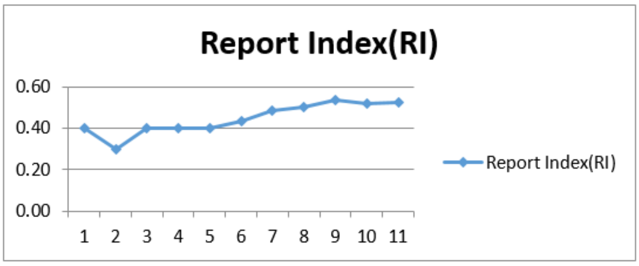

Figure 9. Report Index (RI)

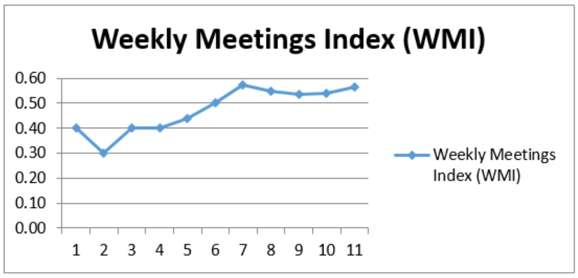

Figure 10. Weekly Meetings Index (WMI)

of reports since project initiation as shown in Figure 9. We note that the reports delivered for project sponsor since the beginning of the first week until the last week. We notice that the largest number of reports was in the seventh, ninth week (4 reports in each). As a result, this increased the RI for the past 7 and 9 weeks to become 0.49 and 0.53, respectively. The smallest number of reports was in the second week (one report only). As a result, this decreased the RI for the past 2 weeks to become 0.30. Finally, the higher value for the RI has the higher positive impact on the project management of large-scale software system and vice versa.

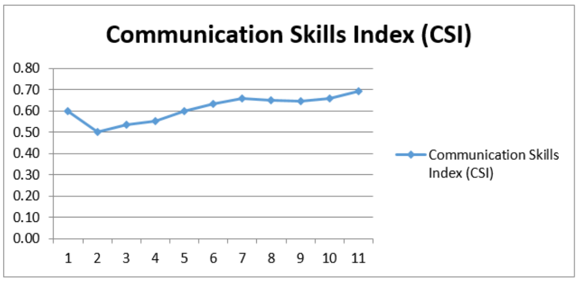

Figure 11. Communication Skills Index (CSI)

5.7 Number of Weekly Meetings with Project Team Members Metric

The number of weekly meetings with project team members metric is defined as: a numerical count of the number of weekly meetings of project manager with project team members while we implement the large-scale software system.

We notice that the Weekly Meetings Index (WMI) is an important number that help to measure the weekly meetings with project team members because it represents average daily number of weekly meetings with project team members since project initiation as shown in Figure 10. We note that the weekly meetings with project team members since the beginning of the first week until the last week. We notice that the largest number of weekly meetings with project team members was in the seventh week (5 meetings). As a result, this increased the WMI for the past 7 weeks to become 0.57, respectively. The smallest number of weekly meetings with project team members was in the second week (one meeting only). As a result, this decreased the WMI for the past 2 weeks to become 0.30. Finally, the higher value for the WMI has the higher positive impact on the project management of large-scale software system and vice versa.

5.8 Degree of Communication Skills Metric

The degree of communication skills metric is defined as: a numerical count of the degree of communication skills of project manager with project team member and project sponsor while we implement the large-scale software system.

We can measure the degree of communication skills of project manager weekly by a questionnaire form or an application tool from the point of view project team member and project sponsor. We can make the value 5 to indicate to an excellent degree of communication skills while 0 indicates a very bad degree of communication skills. By calculating the previous measures on the assumed data that we created, we notice that the Communication Skills Index (CSI) is an important number that help to measure the degree of communication skills because it represents average daily degree of communication skills since project initiation as shown in Figure 11. We note the highest degree of communication skills was in the last week with values 5. As a result, this increased CSI for the past 11 weeks to become 0.69. Also, we notice that the smallest degree of communication skills was in the second weeks and it was equal 2. As a result, this decreased CSI to become 0.50. Figure 12 shows bounds of the CSI so if the CSI is one or close to one so that will help to the project management factor to reach in its high level and if CSI is closer or equal to zero so that will help to that the project management factor to reach in its low level. We suggest that bad degree of degree of communication skills is not a recommended practice and may lead to a decline in the project management as will be shown later.

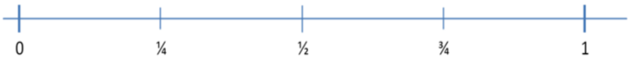

Figure 12. Bounds of CSI

5.9 Number of Delay Days per Phase Metric

The number of delay days per phase metric is defined as: a numerical count of the number of delay days per phase while we implement the large-scale software system. We calculated the following measures:

(1) Number of days per phase.
(2) Adjusted weeks per phase calculated as the number of days per phase (step 1) divided by the number of working days (5 days) per week.
(3) Cumulated weeks which is calculated by adding the adjusted weeks per phase (step 2) to cumulated weeks of the previous weeks since project initiation.
(4) Number of delay days per phase.
(5) Average daily number of delay days per phase calculated as the number of delay days per phase (step 4) divided by the number of days per phase (step 1).
(6) Cumulated average daily number of delay days per phase since project initiation which is calculated by adding the average daily number of delay days per phase (step 5) to cumulated average daily number of delay days per phase of the previous phases since project initiation.
(7) Project Delay Index (PDI) is average daily number of delay days per phase since project initiation calculated as the cumulated average daily number of delay days per phase (step 6) divided by cumulated weeks (step 3).

We notice that the PDI is an important number that help to measure the delay days per phase because it represents average daily number of delay days per phase since project initiation as shown in Figure 13. We notice that the largest number of delay days was in the third phase (3 days). As a result, this increased the PDI for the past 3 phases to become 0.34. The smallest number of delay days per phase was in the first, second, fifth, seventh and eleventh phases (zero days). As a result, this decreased the PDI for the past 1, 2, 5, 7 and 11 phases to become 0, 0, 0.18, 0.23 and 0.23, respectively. Finally, the higher value for the PDI has the higher negative impact on the project management of large-scale software system and vice versa.

5.10 Budget Deficit Metric

The budget deficit metric is defined as: a numerical count of the budget deficit while we implement the large-scale software system. The Budget deficit happens when a person, government or company budget more outlay than there is income available to pay for the spending, through limited time [16]. Then, we calculated the following measures:

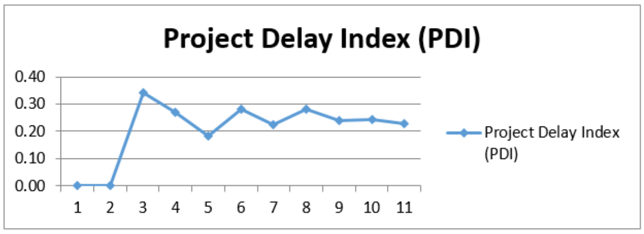

Figure 13. Project Delay Index (PDI)

(1) Planned value (PV) ("the confirmed budget for the work scheduled to be achieved through limited time " [16]) in each week.
(2) Actual cost (AC) ("the costs indeed incurred for the work achieved through limited time " [17]) in each week.
(3) Budget deficit in each week.
(4) Average daily budget deficit in each week calculated as the budget deficit (step 3) divided by the number of working days (5 days) per week.
(5) Cumulated average daily budget deficit since project initiation which is calculated by adding the average daily budget deficit per week (step 4) to cumulated average daily budget deficit of the previous weeks since project initiation [18].
(6) Budget Deficit Index (BCI) is average daily budget deficit since project initiation calculated as the cumulated average daily budget deficit (step 5) divided by number of weeks.
(7) Value of BCI is big so we divided by 100.

From the above results, it was shown that the BCI is an important number that help to measure the budget deficit because it represents average daily budget deficit since project initiation as shown in Figure 14. We notice that the largest budget deficit was in the first week (200). As a result, this increased the BCI for the first weeks to become 0.40. Some weeks, there was no budget deficit as in the second, fourth and sixth weeks. As a result, this decreased the BCI for the past 2, 4 and 6 weeks to become 0.2, 0 and 0.03 respectively. Some weeks have been provided part of the budget as in the third, eighth, ninth and tenth weeks so this decreased the BCI for the past 3, 8, 9 and 10 weeks to become 0, 0.3, 0 and 0.04 respectively. This helps to compensate for the budget deficit in some weeks so the result of the BCI in last week was zero. Finally, the higher value for the BCI has the higher negative impact on the project management of large-scale software system and vice versa.

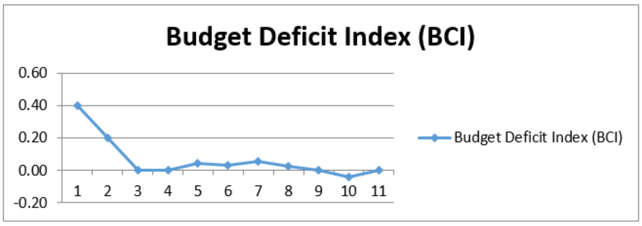

Figure 14. Budget Deficit Index (BCI)

5.11 Number of Solved / Unsolved Risks Metrics

The number of solved / unsolved risks metrics are defined as: a numerical count of the number of solved / unsolved risks while we implement the large-scale software system. Then, we calculated the following measures:

(1) Number of days per risk.
(2) Adjusted weeks of per risk calculated number of days per risk (step 1) divided by the number of working days (5 days) per week.
(3) Cumulated weeks which is calculated by adding the adjusted weeks of per risk (step 2) to cumulated weeks of the previous weeks since project initiation.
(4) # risks.
(5) # inactive risks.
(6) # solved risks.
(7) Number of unsolved risks.
(8) Average daily number of risks in each phase calculated by as the number of risks (step 4) divided by number of days per risk (step 1).
(9) Average daily number of inactive risks in each phase calculated by as the number of inactive risks (step 5) divided by number of days per risk (step 1).
(10) Average daily number of solved risks in each phase calculated by as the number of solved risks (step 6) divided by number of days per risk (step 1).
(11) Average daily number of unsolved risks in each phase calculated by as the number of unsolved risks (step 7) divided by number of days per risk (step 1).
(12) Cumulated average daily number of risks since project initiation which is calculated by adding the average daily average daily number of risks in each phase (step 8) to cumulated average daily number of risks of the previous days since project initiation.
(13) Cumulated average daily number of inactive risks since project initiation which is calculated by adding the average daily average daily number of inactive risks in each phase (step 9) to cumulated average daily number of inactive risks of the previous days since project initiation.
(14) Cumulated average daily number of solved risks since project initiation which is calculated by adding the average daily average daily number of solved risks in each phase (step 10) to cumulated average daily number of solved risks of the previous days since project initiation.
(15) Cumulated average daily number of unsolved risks since project initiation which is calculated by adding the average daily average daily number of unsolved risks in each phase (step 11) to cumulated average daily number of unsolved risks of the previous days since project initiation.
(16) Average daily number of risk since project initiation calculated as the cumulated average daily number of risks (step 12) divided by cumulated weeks (step 3).
(17) Average daily number of inactive risks since project initiation calculated as the cumulated average daily number of inactive risks (step 13) divided by cumulated weeks (step 3).
(18) Average daily number of solved risks since project initiation calculated as the cumulated average daily number of solved risks (step 14) divided by cumulated weeks (step 3).
(19) Average daily number of unsolved risks since project initiation calculated as the cumulated average daily number of unsolved risks (step 15) divided by cumulated weeks (step 3).
(20) Risk Resolution Index (RRI) is average daily difference between number of solved risk and number of unsolved risks since project initiation (step 18 and 19).

As we run this test, we notice that the risks were continuous since the beginning of the first week until the last week. RRI is an important number that help to compare between the solved risks and unsolved risks because it represents average daily difference between solved and unsolved risks since project initiation as shown in Figure 15. The largest number of RRI was in the first week. It was 0.31 because there was one solved risk while we had zero unsolved risks were in the first 5 days of the project. The smallest number of RRI was in the third week. It was 0 because increased unsolved risks comparison solved risks. Finally, the higher value for the of RRI, the higher good impact on the project management of large-scale software system and vice versa.

5.12 Formulation of Project Management (PM) Metric

We formulated project management as the summation of PCI, PAI, SFI, CMI, RI, WMI, CSI and RRI and subtract from this total the PDI and BCI that we

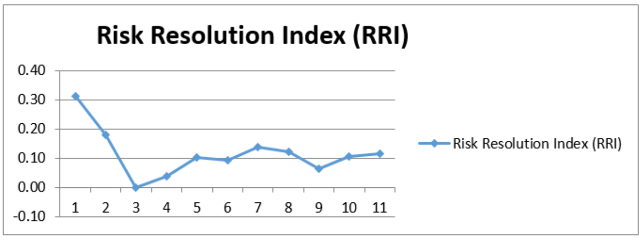

Figure 15. Risk Resolution Index (RRI)

Table 3. Measurement of the Project Management

weeks	PCI	PAI	SFI	CMI	RI	WMI	CSI	RRI	PDI	BCI	PM	Normalization map Minimum Maximum to 0 and 10 (PM)
1	0.60	0.40	0.40	0.20	0.40	0.40	0.60	0.31	0.00	0.40	2.91	2.43
2	0.50	0.40	0.30	0.20	0.30	0.30	0.50	0.18	0.00	0.20	2.48	0.00
3	0.47	0.47	0.40	0.27	0.40	0.40	0.53	0.00	0.34	0.00	2.59	0.62
4	0.40	0.45	0.50	0.25	0.40	0.40	0.55	0.04	0.27	0.00	2.72	1.33
5	0.40	0.44	0.60	0.24	0.40	0.44	0.60	0.10	0.18	0.04	3.00	2.92
6	0.53	0.47	0.67	0.27	0.43	0.50	0.63	0.09	0.28	0.03	3.28	4.47
7	0.63	0.69	0.69	0.31	0.49	0.57	0.66	0.14	0.23	0.06	3.88	7.85
8	0.63	0.70	0.65	0.33	0.50	0.55	0.65	0.12	0.28	0.03	3.82	7.48
9	0.69	0.71	0.64	0.33	0.53	0.53	0.64	0.06	0.24	0.00	3.91	8.03
10	0.74	0.72	0.66	0.34	0.52	0.54	0.66	0.11	0.24	-0.04	4.08	8.97
11	0.80	0.76	0.67	0.36	0.53	0.56	0.69	0.11	0.23	0.00	4.27	10.00

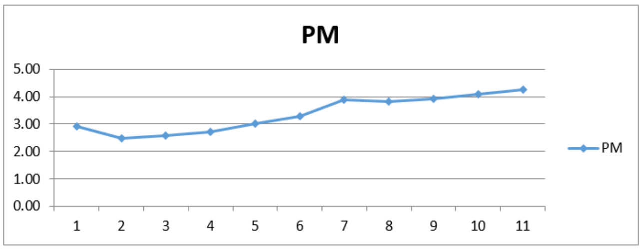

Figure 16. Measurement of the Software Project Management

described in the previous sections shown in following: PM=PCI+PAI+SFI+CMI+RI+WMI+CSI+RRI-PDI-BCI

Table 3 shows important values of metrics that help to measure the project management and values of the project management.

We calculate software project management in each week independently until the 11th week. Based on the results of the hypothetical data that we assumed to be more relevant to real projects, we notice that the PCI, PAI, SFI and CSI have the highest influence on the performance of project management in the new large-scale software system. In addition, we notice that the values of the project management changed through 11th weeks because of the change in some values of the numbers of metrics as shown in Figure 16. According to the above results, we see that the largest value of

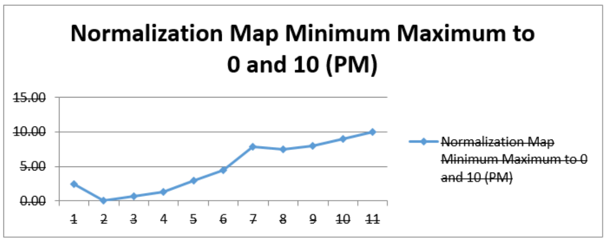

Figure 17. Measurement of the Project Management Using Normalization Map

Figure 18. Bounds of the Normalization Map for Software Project Management

project management was in the eleventh week:

$$\text{Project Management} = 4.27$$

Because of the increased values of the PCI, PAI, CMI and CSI, which were:

$$PCI = 0.8$$
$$PAI = 0.76$$
$$CMI = 0.36$$
$$CSI = 0.69$$

The smallest number of project management - was in the second week:

$$\text{Project Management} = 2.48$$

Because of the decreased values for the PAI, SFI, CMI, RI, WMI and CSI in that week to:

$$PAI = 0.4$$
$$SFI = 0.3$$
$$CMI = 0.2$$
$$RI = 0.3$$
$$WMI = 0.3$$
$$CSI = 0.5$$

We used normalization of the result project management between 0 and 10 as shown in Table 3 and Figure 17. We notice the similarity of results of project management before and after normalization as in Figure 17 and Figure 18. Figure 18 shows bounds of normalization map of project management so if the project management is ten or close to ten then it means that the project management is high and if the project management is closer to zero then it means that the project management is low.

6 Conclusion and Future Works

Large-scale software systems (LSS) are complicated because of their size, amounts of hardware, lines of source code, numbers of users, volumes of data, diversity of services and applications they provide. Several factors play role in achieving a successful LSS implementation which are known as the critical success factors (CSFs) of large-scale software systems. In this work we selected a project management CSF to be studied by measuring its impact on the implementation of a large-scale software system. We applied CSF-Live! method for measuring and monitoring project management factor that may affect the implementation of large-scale software.

We generated set of metrics which were represented numerically to enable monitoring and controlling of goals and collect data to reflect metrics. Finally, we generated formulas representing project management factor, collected data, and presented a case study that explores and explains the results. The article that we conducted in this study can be extended in the following way: include more CSFs: in this study we focused on three CSFs, however, other factors are also important, and we aim to study more CSFs and make proper formulation for each of them and focus on other project types: in this study we focused on large-scale software systems. In future we can develop some other CSFs for small and medium software systems.

References

[1] Zhang Liang. Critical success factors of enterprise resource planning systems implementation success in china. In *Proceedings of the 36th Annual Hawaii International Conference on. IEEE*, 2003.

[2] PabedinskaitÄŮ Arnoldina. Factors of successful implementation of erp systems. In *The Economics and Management Conference*, 2010.

[3] Wong Ada. Critical failure factors in erp implementation. *PACIS*, 40, 2005.

[4] Dezdar Shahin and Sulaiman Ainin. Critical success factors for erp implementation: insights from a middle-eastern country. *Middle-East Journal of Scientific Research*, 10:798–808, 2011.

[5] Malik F. Saleh, Muneer Abbad, and Mohammed Al-Shehri. Erp implementation success factors in saudi arabia. *International Journal of Computer Science and Security (IJCSS)*, 7:1–15, 2013.

[6] Bar Adnan Al. An analysis of the contracting process for an erp system. In *SEAS-2013, Computer Science Conference Proceedings in Computer Science & Information Technology (CS & IT) series*, 2013.

[7] Ahmad Mutahar. An empirical study of critical success factors for erp software system implementations in saudi arabia. *Master thesis*, 2016.

[8] Rich. Bauer. Just the facts101 e-study guide for: Introduction to chemistry. *Cram101 Textbook Reviews*, 2012.

[9] Fang Li and Sylvia Patrecia. Critical success factors in erp implementation. In *IEEE 6th International Conference*, 2005.

[10] David Allen, Thomas Kern, and Mark Havenhand. Erp critical success factors: an exploration of the contextual factors in public sector institutions. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on. IEEE*, 2002.

[11] Adnan Al Bar, Victor Basili, Wajdi Al Jedaibi, and Abdul Jawad Chaudhry. An experience based evaluation process for erp bids. *Journal of Software Engineering*, 29:1–14, 2013.

[12] Melia Detta. Critical success factors and performance management and measurement: a hospitality context. In *the 6th annual âĂŸTourism and Hospitality Research in Ireland Conference, Shannon College of Management, Shannon*, 2010.

[13] Esteves Jose, Joan Pastor-Collado, and Josep Casanovas. Monitoring business process redesign in erp implementation projects. In *the AMCIS 2002 Proceedings Conference*, 2002.

[14] Sufian Khamis and Wajdi Aljedaibi. A framework for measuring critical success factors of large-scale software systems. *International Journal of Computer Engineering and Technology*, 7:71–82, 2016.

[15] Osman M. R. Erp systems implementation in malaysia: the importance of critical success factors. *International Journal of Engineering and Technology*, 3:125–131, 2006.

[16] Mark Senn. *Using LATEX for Your Thesis*, (accessed February, 2019). http://study.com/academy/lesson/what-is-a-budget-deficit-definition-causes-history.html.

[17] Mark Senn. *Using LATEX for Your Thesis*, (accessed February, 2019). http://www.dummies.com/how-to/content/earned-value-management-terms-and-formulas-for-pro.html.

[18] Caldiera Victor, R. Basili-Gianluigi, and H. Dieter Rombach. Goal question metric paradigm. *International Journal of Encyclopedia of Software Engineering*, 1:528–532, 1994.

Wajdi Aljedaibi is an associate professor in the department of Computer Science at Faculty of Computing & Information Technology, King Abdulaziz University (KAU). He is currently the Dean of Information Technology at KAU, which provide IT services to over 200k beneficiaries. He was awarded the Ph.D in Information Technology âĂŞ Software Engineering and MSc. in Software Systems Engineering both from George Mason University. His current research interests are: software measurements, software architecture, and Open Source ERP systems.

Sufian Khamis have a BSc and MSc in Computer Science with Software Engineering specialization. I got my BSc in 2011 with GPA 4.79 of 5 (Excellent with First Honors) and I got my MSc in 2016 with GPA 4.69 of 5 (Excellent). Also, I have 9 years experience of professional work, as I am working as a software developer in the Deanship of Information Technology in King Abdulaziz University since 2011.