ISSN 2355-0082

Low Cost Data Analytics Implementation on Project Management Process Automation

Cornelius Mellino Sarungu[1], Liliana[2]
Program Studi Sistem Informasi, Bina Nusantara University, Jakarta, Indonesia
Program Studi Teknik Informatika, Universitas Surabaya, Surabaya, Indonesia
cornelius.sarungu@binus.ac.id
lili@staff.ubaya.ac.id

Received on August 12th, 2018
Accepted on December 21st, 2018

***Abstract*— Project management practice used many tools to support the process of recording and tracking data generated along the whole project. Project analytics provide deeper insights to be used on decision making. To conduct project analytics, one should explore the tools and techniques required. The most common tool is Microsoft Excel. Its simplicity and flexibility make project manager or project team members can utilize it to do almost any kind of activities. We combine MS Excel with R Studio to brought data analytics into the project management process. While the data input process still using the old way that the project manager already familiar, the analytic engine could extract data from it and create visualization of needed parameters in a single output report file. This kind of approach deliver a low cost solution of project analytics for the organization. We can implement it with relatively low cost technology on one side, some of them are free, while maintaining the simple way of data generation process. This solution can also be proposed to improve project management process maturity level to the next stage, like CMMI level 4 that promote project analytics.**

***Index Terms*—project management, project analytics, data analytics.**

I. INTRODUCTION

Monitoring is an important project phase in project management processes. It consists of gathering information about current situation of the project and compare it to its baselines (Mahaney & Lederer, 2003). The result then used by the project manager, project sponsors or even top management to create strategic decision upon the project situation in order to maintain the project performance (Mahaney & Lederer, 2010). Feedback that given to the team by analyzing the progress report that is an output of monitoring is also an act of maintaining project performance (Akhavan Tabassi, Roufechaei, Bakar, & Nor'Aini, 2017). Monitoring believed could reduce performance deviation and set the declining performance back on track (Chang, 2017).

II. LITERATURE REVIEW

From the author's experience with the CMMI certification process at the software development company a few years ago, noticed that there are still many processes in project management that are done manually. Starting from data input to report generation.. There is nothing wrong with that. But when we talk about CMMI for development, at process maturity level 4 and above, then the criteria requirements are (1) quantitatively managed, (2) analyzed and (3) continuously improved (Chaudhary & Chopra, 2017, p. 13). We will discuss the first two points.

The meaning of the quantitatively managed is more or less the use of data in every process of planning, execution and monitoring and control of the project (Chaudhary & Chopra, 2017, p. 13). It does not matter whether data is collected manually or automatically, which is important when performing an action there is data that underlies its decision making.

In project management process usually there are activities of information distribution in a project which include making periodic report like: (1) Project status report; (2) Progress report; (3) Trend report; (4) Forecasting report; (5) Variance report (Mulcahy, 2013, p. 393).

Each report has a specific purpose. The status report contains explanation about the position of the current project. Progress report present the percentage of progress in project positions compared to previous period positions. Trend report present a significant trend toward the direction and velocity of project parameters such as time, scope, cost, quality and resources. Forecast report present the approximate direction and achievement of those parameters over the next several periods. While the variance report present deviations that occur from the

planned position at the beginning or last time, which is often called the baseline.

The making of such reports is usually based on data entered either daily or periodically by the entire project team and most likely stored in separate systems according to their respective functions. Quality related data for example, can be extracted from the issue management or bug tracker software application used by the team. The programmer and tester who is responsible for the process of input and update data.

Analytics can be defined as a method to use the results of analysis to better predict customer or stakeholder behaviors (Singh, 2016). Implementation of data analytic can start from this point. Project analytics provide deeper information and a lot of metrics useful for decision making (Stolovitsky, 2011). Current status information, in terms of coverage, cost, time, quality and others, which generated periodically (Mavenlink, 2013), are sources of data that can be used for analysis and generate new insights for the organization and control process of projects, depending on the type of analysis process used, whether descriptive, diagnostic, predictive, or prescriptive (Declues, 2017).

To combine the process of analytics with project management, the first thing we do is to make sure that the underlying processes run with high discipline. The process meant here is the process of data input and update. Make sure it works. In a software development environment that has been CMMI level 3 certified usually ensures the processes are running well by conducting periodic audits. This audit process is usually done by process quality assurance (PQA) or software quality assurance (SQA) (Chaudhary & Chopra, 2017, p. 76). There is a suite of tools for auditing that tracks for every change occur and ensures that all the necessary processes are executed in a disciplined manner.

Project management tools that are equipped with analytics module are rarely found in cheap price. Let's say SAP PPM, that has almost everything you need to run an enterprise scale program and project management, complete with its visual analytics dashboard, can bring high implementation cost or total cost of ownership to organization. Assumed that the implementation itself is successful, if not, it will bring another sunk cost. In this paper we propose a low cost solution that also quite common to the users. Microsoft Excel is a common tool for project managers, it is a basic handy swiss army knife for them to use. Relatively cheap, with only $8.25/month for Office 365 or $399 for one time purchase of Office 2016 [14].

III. METHODS

Because there are ten knowledge areas in project management as seen on table 1 (Project Management Institute, 2013, p. 61), we limit the scope of this experiment on Quality Management area only, with specialization on issue tracking which still related to task management topic.

Table 1 Project Management Knowledge Areas.

No.	Knowledge Area
1	Project Integration Management
2	Project Scope Management
3	Project Time Management
4	Project Cost Management
5	Project Quality Management
6	Project Human Resource Management
7	Project Communication Management
8	Project Risk Management
9	Project Procurement Management
10	Project Stakeholder Management

Issue data are collected and stored in MS Excel file that already has a preformatted table. The Excel file then served as input to our analytical engine build on R platform. R is chosen because of its flexibility and the ability to create customized analytical report using the code (RMarkdown). List of reasons why R is chosen could be described as below: (1) Flexible, easy, and friendly graphical capabilities that can be displayed on the video display of your computer or stored in different file formats; (2) Data storage facility to store large amounts of data effectively in the memory for data analysis; (3) Large number of free packages available for data analysis; (4) Provides all the capabilities of a programming language; (5) Supports getting data from a wide variety of sources, including text files, database management systems, web XML files, and other repositories; (6) Runs on a wide array of platforms, including Windows, Unix, and macOS; (7) R is free (Hodeghatta & Nayak, 2017, p. 21).

The architecture of the solution is presented in figure 1.

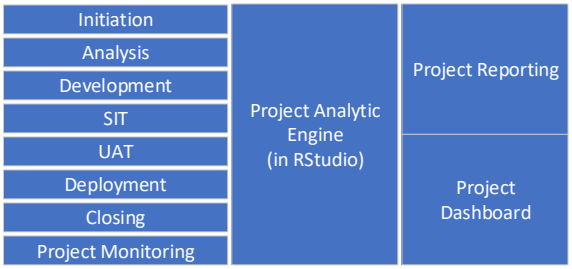

Figure 1 High Level Architecture Diagram.

The Excel file is acting as the data source and will be on its position in the left side of the architecture diagram. It is used by the project to store related data in simple manner. Each stage of the software development project has its own directory that store related Excel file. In example, the SIT Result is stored in SIT folder. The Task List can be stored in Project Monitoring directory. The project analytic engine will then open the file and read all data from selected table. The data will go through a transformation process if necessary and resulted in a *tidy data* ready to be processed.

The project analytic engine itself is a module consists of codes that doing the data analysis. The result of the data analysis process is a table filled with result data. Result data then visualized in charts as a project dashboard component or dumped in a conventional project report. All steps can be done in a single program file including the visualization and report formatting. The last result is an MS Word file contains the analysis descriptions, necessary tables and charts.

The solution model used in this experiment could be summarized with this block diagram:

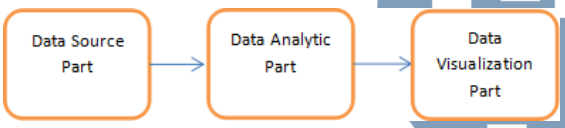

Figure 2 Block diagram of the solution.

There are three parts that comes into play. First part is the Data Source part, dealing with the data input and files that the project management process deal with every day. Second part is the Data Analytic part, dealing with the data extraction (ETL), data analysis processes, and result preparation. Third part is Data Visualization part that deals with the process of shaping the data into visual easy-to-read graphs.

All parts are actually code modules in reality. Data Source part consists of the data file itself and code module that open the file and extract its data into R table. In the real situation the file to be extracted could be in any format, whether it is .csv, MS Excel, or even XML file. As long as the format could be read by the R language program. The Data Source part function can also be extended to include the ETL (Extract, Transform and Load) process. Converting the raw data into tidy data that ready to be processed by the Data Analytic part, or we could say the Analytic Engine.

The Data Analytic part is a code module that contains algorithms or methods, whether it is statistical or not, that deals with the extracted data. From one Data Source part, we can build and connect to multiple Data Analytic part. Each of the module serves specific analytic process.

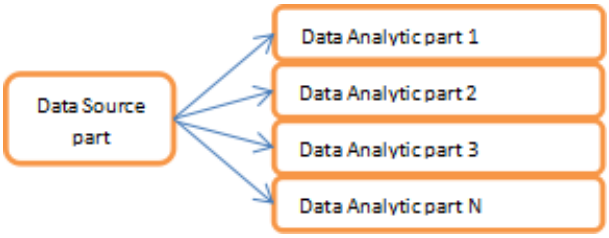

Figure 3 Data Source and Data Analytic part connection.

After the analytic process has been done in Data Analytic part, usually the result data is thrown into visualization module, and it is accommodated by the Data Visualization part. This part consists of codes that draw any model of chart, tables, dashboard components, or anything that can be seen on screen. The relationship between the Data Analytic and Data Visualization part are one-to-many relationship as illustrated on figure 4.

Figure 4 Data Analytic and Data Visualization part.

The connection illustrated above means that for each Data Analytic part that give a result data, we can build many visualization modules to visualize specific requirement (i.e. bar chart, pie chart, line chart, 3D chart, reports etc.) with the same result data. In real case, there can also happen only one Data Visualization part connected to each Data Analytic part. Just make sure to follow the requirements.

All this sequence of process is run in R Studio. We choose this tool and programming language because it is a quite handy tool and relatively easy to understand programming language. R is well known programming language that already used among statisticians. R Studio is used not only to develop the code modules but also to play with the data itself.

To deal with the process of development, we also offer steps to be followed on developing the solution. As illustrated on figure 5, the first step is Data Preparation. In this step, we design the data structure and fill it testing data or if we are in a real project, we can use real data.

The second step is Data Cleaning which consists of cleaning the garbage data that splattered across the data block. Removing data that do not fit to be included in analytic process will make our result more accurate.

The third step is Development of the Data Source part. Codes that functioned for accessing the MS Excel file (data source file) are built in this step. It is

including the transformation from raw data into tidy data. After the development is finished, then testing can be conducted on Data Source Testing step where we conduct the testing for Data Source code module.

The fourth step is Development of the Data Analytic part. Here we build the analytic engine for our solution. Algorithm, business rules and formulas are combined to compose an analytic engine that fit for our solution. This step is followed by Data Analytic Testing step which consists of testing the analytic engine using the testing data.

The fifth step is Development of the Data Visualization part. After the analytic engine finish the analytic process, the result data will be produced.

Data Visualization codes read that result data and create a chart or any other dashboard elements to make the data easily read and understand by business users. This step is also followed by its testing step pair.

Here, we could state that the development phase is done. Every steps of development should be paired with their related testing steps. In advanced software development life cycle used by Java, .NET or Python developer, there are unit tests to run along with the development of the code. In R we still don't have that kind of mechanism, but we can substitute it with other kind of test that show similar purpose.

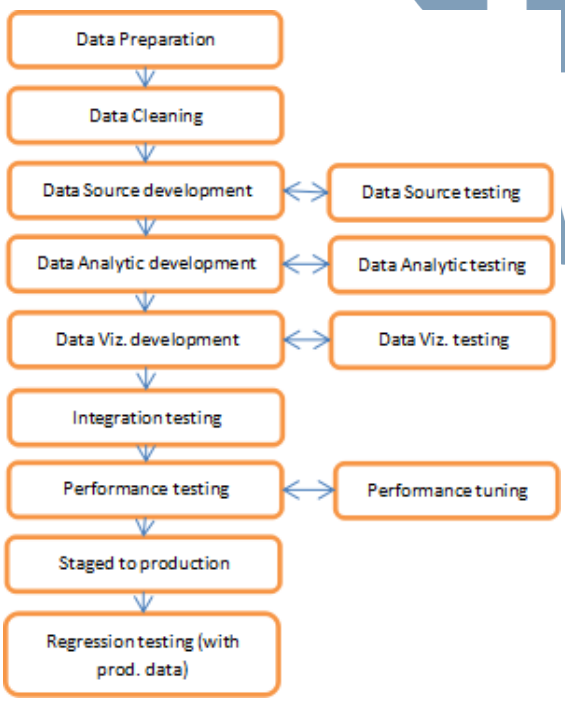

Figure 5 Suggested sequence of the implementation.

When all the code modules are integrated, integration testing is conducted to test the overall data processing and analytic process. In this step we run all integrated part from Data Source to Data Visualization and using testing data for that purpose.

After the integration testing is done, we move to the next step which is Performance Test. On this step we conduct a load and performance testing using a huge volume of testing data to see how our analytic engine and overall modules behave with that kind of situation. Stability is the key to pass this test, but of course a performance threshold should have been set as the expected minimum performance limit. Any instability at number below that minimum limit should be fixed by doing tune up to any of the modules, enhancing the algorithm or upgrading the hardware and network used. This all done in Performance Tuning step.

After the minimum performance limit is achieved, we promote our solution into the production server. Regression test is then conducted after the deployment process finished. Regression test means testing all of the features once again to confirm the readiness of the solution. Real data is used after the regression test finished.

Table 2 The pre-formatted Excel table for task management.

Field	Types	Description
NO	String	Sequence number
TASK ID	String	Task id
SUB PROJECT	String	Sub project / module name
PROJECT PHASE	String	Project phases number (phase 1, 2, ...)
SDLC PHASE	String	Phases of SDLC (analysis, development, etc.)
TASK NAME	String	Task name
DESCRIPTION	String	Task description
PIC	String	PIC responsible for the task
STATUS	String	Status of the task (open, in progress, closed)
PLANNED START	Date	Planned start date
PLANNED END	Date	Planned end date
ACTUAL START	Date	Actual start date
ACTUAL END	Date	Actual end date
LAST LIFE	Integer	Last modified date
LIFETIME	Integer	Difference between actual start and last life
PLANNED DURATION	Integer	Difference between planned start and end
ACTUAL DURATION	Integer	Difference between actual start and end
REMARK	String	Optional remark

We have discussed the input or data source part of the architecture. For the ETL section, analytic engine and data warehousing we can choose technology that is not too complicated and cheap. In this experiment, I chose to use a lightweight and very flexible R studio. One with little experience of R

programming skills will be able to create lightweight ETL modules and customized analytic engine. Another tool that also good to be used is Jasper ETL.

There will be some examples of the results of the analytic engine output encoded in R. Excel file contain related task tracking is used as input or data source. While the output is some graph indicators that can be used as a dashboard component, or just information for management.

The ETL process in R Language is described with modules of code. The first module is building *Task lifetime per module* data as follows:

```
library(xlsx)

project_data <- read.xlsx("pm-analytics/Task List.xlsx", sheetIndex=1, header=TRUE)

#lifetime per module

duration_per_module_max <- tapply(project_data$LIFETIME, project_data$SUBPROJECT, FUN=max)

duration_per_module_min <- tapply(project_data$LIFETIME, project_data$SUBPROJECT, FUN=min)

duration_per_module_avg <- tapply(project_data$LIFETIME, project_data$SUBPROJECT, FUN=mean)

m_duration <- as.matrix(duration_per_module_min)

m_duration <- cbind(m_duration, duration_per_module_avg, duration_per_module_max)

colnames(m_duration) <- c("min", "avg", "max")
m_duration <- t(m_duration)

colours <- c("green", "yellow", "red")
```

The analytics and custom report generation process is as follows:

```
## 1.Task Lifetime per Module
par(mar=c(8.75,4,1.5, 0.1))

barplot(m_duration, main="Task Lifetime per Module", ylab="Duration (days)", ylim=c(0,5+max(m_duration)), las=2, beside=TRUE, cex.names = 0.8, axis.lty = 1, col=colours)

legend("topleft", c("min","avg","max"), cex=0.7, bty="n", fill=colours)
```

The resulting graph is as follows:

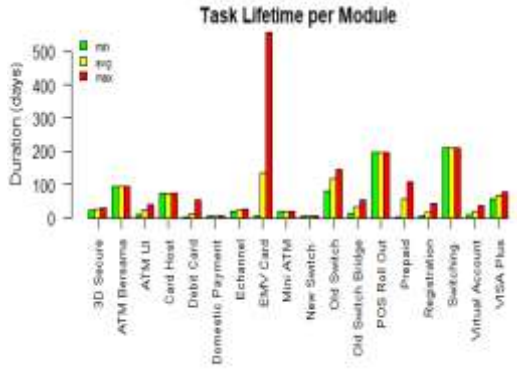

Figure 6 Task lifetime per module.

The second module is building *Task lifetime per phase* data as follows:

```
## 2. Task Lifetime per Phase

duration_per_phase_max <- tapply(project_data$LIFETIME, project_data$SDLC.PHASE, FUN=max)

duration_per_phase_min <- tapply(project_data$LIFETIME, project_data$SDLC.PHASE, FUN=min)

duration_per_phase_avg <- tapply(project_data$LIFETIME, project_data$SDLC.PHASE, FUN=mean)

m_duration <- as.matrix(duration_per_phase_min)
m_duration <- cbind(m_duration, duration_per_phase_avg, duration_per_phase_max)

colnames(m_duration) <- c("min", "avg", "max")

m_duration <- t(m_duration)
par(mar=c(6,4,1.5, 0.1))

barplot(m_duration, main="Task Lifetime per Phase", ylab="Duration (days)", ylim=c(0,5+max(m_duration)), las=2, beside=TRUE, cex.names = 0.8, axis.lty = 1, col=colours)

legend("topleft", c("min","avg","max"), cex=0.7, bty="n", fill=colours)
```

The resulting graph is as follows:

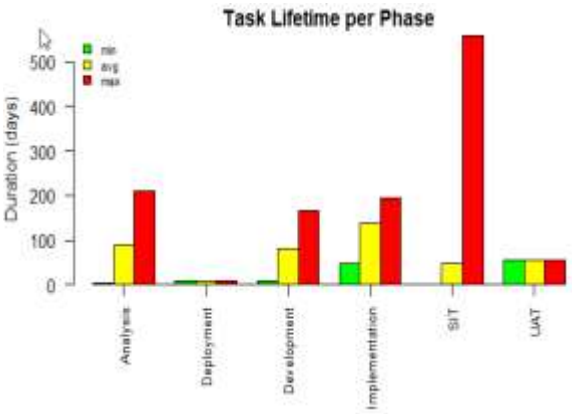

Figure 7 Task lifetime per phase.

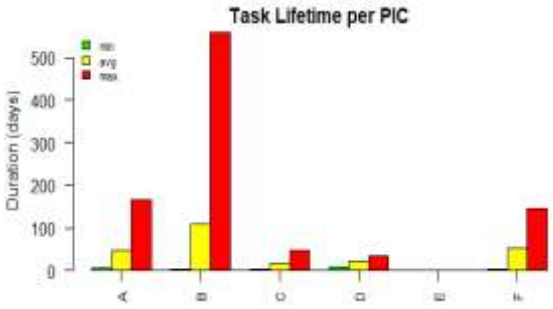

Figure 8 Task lifetime per phase.

In addition to those previous two graphs, we can also create your own other charts according to our own needs. Determine what information we want to dig from existing data, then write the code and run it. Or if we use tools like Jasper ETL, we can set the scheduler to do it periodically.

IV. Discussion of The Result

Experiment conducted with real project data of task tracking resulting in a quite satisfying result. A well formatted report in MS Word format is produced by the Data Visualization part, contains three charts: (1) Task lifetime per module; (2) Task lifetime per phase; (3) Task lifetime per PIC (Person in Charge).

More charts could easily be added into the report as required. The process of report generation also fast, below 0.5 seconds. User should trigger the process of report generation by selecting menu from inside R Studio.

The time taken to run the whole parts are listed below (experiment is repeated ten times) in Table 3:

Table 3 Time required for execution

Experiment	Time (Sec.)
1	0.4258
2	0.4358
3	0.3498
4	0.4188
5	0.4298
6	0.4198
7	0.4169
8	0.3570
9	0.4318
10	0.3510

This time required is directly proportional with the size of data being loaded, transformed and analyzed. The bigger the size of data, the longer the time required to process that data. Performance tuning is needed if huge data volume is involved. But for small or tiny volume, ranging from 10 KB to 1 MB the performance tuning step could be skipped.

The chart produced by the analytic process is useful to determine which module, phase, or even people in charge have a high contribution to a delay time or risk of delay of related project. From figure 6 there are three kind of information could be spotted: (1) The days spent to resolve an issue or finish a task (resolve time). We could see that days spent is very high, around 250 days maximum, and this shows us that vital problems are happening right now in the project and top levels management involvement is needed. (2) The module that has a maximum risk of delay. From the chart, which shows us that EMV (Europay, Mastercard Visa) Card module has the highest contribution to project delay, we could draw a conclusion that this module is quite complex, or has a complex level of coordination needed to develop it. (3) The rank of problem priority level, the top three are EMV Card module, Switching and POS Rollout related tasks. The longer the time for resolving an issue or finishing a task related to a module give us insight that the module probably still has lots of gray area on its specification.

The result of second module, Task Lifetime per Phase can be seen on figure 7. From the chart we can draw some conclusions: (1) The SIT (System and Integration) phase has the maximum resolve time, but its average is quite low compared to Development, Implementation and Analysis phase. The Project Manager should escalate the issue that has maximum resolve time to steering committee. (2) The Analysis, Development, and Implementation phase are definitely at high risk. Problems are happening at those phases. At Analysis phase some requirements or specifications are probably still unclear. At Development phase the problem could be waiting interrelated modules to be finished, dependency to external modules developed by other team, or unclear technical specification. At Implementation phase the problem is usually happen on the deployment preparation.

The third module chart on figure 8 shows us who is the most problematic person in the project. The longest time consecutively taken by D, A and T to finish their tasks. The management or resource manager should check the person whether he or she is overloaded with many tasks or there is a performance problem.

Suggestions for improvement: This experiment is only a small part of the overall project management process. If we want to apply it we can start with the following steps: (1) Standardize all the data input and update processes, along with the output format; (2) Create Extract-Transform-Load (ETL) scenario and analytics engine applications for each need, grouping

per knowledge area (Integration, Scope, Cost, Time, Quality, Communication, HR, Risk, Procurement, Stakeholder).

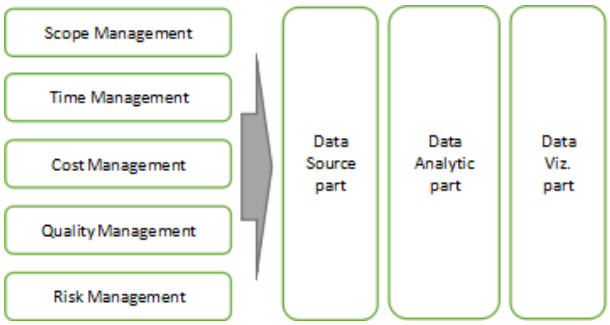

Figure 9 Block diagram of larger suggested architecture.

The suggested architecture shown in figure 9 consists of specific folders on left of the Data Source part. Each of this folder contains project management data file specific to its knowledge area. The Data Source part grab necessary related data file from those folders. The Data Analytic part analyze the data, then give the result to the Data Visualization part. The process of these three core modules could be in build as synchronous or asynchronous processes. The result of this larger solution is a global view of project dashboard that showing all aspects related to project management knowledge areas. Each knowledge area could have more than one measurement metric, so the global dashboard will be very rich with project indicators. This condition also applies to report generated by the solution. We could use the report as a basis for periodical project monitoring and control, project health check, and steering committee meeting; (3) Choose a tool that is good enough and can accommodate process automation.

V. CONCLUSION

Toward the achievement of CMMI level 4 or analytical project management process, we could combine the mainstream tools with any data analytic tool. R language and R Studio are chosen because of their familiarity to use, support statistical methods, and support highly customization upon Data Analytic and Data Visualization parts.

Before we implement the solution, it is suggested that we should design carefully the data structure used in the Data Source part. This is to avoid any problem related to the data structure in the analytic process.

The overall solution should be able to run automatically. This is important when we're dealing with periodic reporting, or dashboard visualization. We could list the application we have developed into the system scheduler to gain the automation trigger.

Implementation of this solution could be done with all knowledge areas of Project Management Process included as long as there are any data management involved in its process or sub-processes. A good and computerized model of project monitoring combined with good mental model will increase the project's probability of success (Abdel-Hamid, 2011).

REFERENCES

[1] Abdel-Hamid, T. K.. "Single-loop project controls: reigning paradigms or strait jackets?" Project Management Journal, 42(1), 17–30, 2011.

[2] Akhavan Tabassi, A., Roufechaei, K. M., Bakar, A. H. A., & Nor'Aini, Y. "Linking team condition and team performance: A transformational leadership approach." Project Management Journal, 48(2), 22–38, 2017.

[3] Chang, J. Y. T. "Mutual monitoring of resources in an enterprise systems program." Project Management Journal, 48(1), 100–114, 2017.

[4] Chaudary, M., & Chopra, A. "CMMI for Development: Implementation Guide". New York: Apress, 2017.

[5] Declues, J. "Four types of big data analytics and examples of their use.", from http://www.ingrammicroadvisor.com/data-center/four-types-of-big-data-analytics-and-examples-of-their-use. 2017, March 23. Retrieved October 10, 2017.

[6] Hodeghatta, U. R., & Nayak, U. "Business Analytics Using R - A Practical Approach." Berkeley, CA: Apress, 2017.

[7] Mahaney, R. C., & Lederer, A. L. "The role of monitoring and shirking in information systems project management." International Journal of Project Management, 28(1), 14–25, 2010.

[8] Mahaney, R. C., & Lederer, A. R. "Information systems project management: an agency theory interpretation." The Journal of Systems and Software, 68, 1–9, 2003.

[9] Mavenlink. "Using analytics for project management," from http://blog.mavenlink.com/using-analytics-for-project-management, 2017, February 20. Retrieved October 10, 2017.

[10] Mulcahey, R. "PMP exam prep: Accelerated learning to pass PMI's PMP exam (8th ed.)." Minnetonka : Minn: RMC Publications, 2013.

[11] Project Management Institute. "A guide to the Project Management Body of Knowledge (PMBOK guide), fifth edition." Newton Square, PA: PMI, 2013.

[12] Sing, H. "Project management analytics: A data-driven approach to making rational and effective project decisions (1st ed.)." Pearson FT Press, 2016.

[13] Stolovitsky, N. "Project performance management analytics," from http://www.projectperfect.com.au/white-paper-project-performance-management.php, 2011, September. Retrieved October 11, 2017.

[14] Foley, Mary J. "How much does Microsoft Office 2016 cost without a subscription?" from https://www.zdnet.com/article/how-much-does-microsoft-office-2016-cost-without-a-subscription/, 2015, September. Retrieved February 12, 2017.