

Framework for Integrating Software Project Tasks and Change Requests

Amira M. Gaber
Department of Information
System,
Faculty of Computer and
Information Sciences, Cairo
University

Sherief Mazen
Department of Information
System,
Faculty of Computer and
Information Sciences, Cairo
University

Ehab E. Hassanei
Department of Information
System,
Faculty of Computer and
Information Sciences, Cairo
University

ABSTRACT

Most of the software projects may deviate from their original plans during the Software Life Cycle in the organization. This deviation typically results from the changes that happened and the project manager depends on his experience to estimate the effects of the change. This may affect the duration and budget of the project. It is important that the software project should not deviate or crash according to the estimated deadline. Thus any change should be estimated and discussed before implementation. In this paper will introduce a framework that integrates the Software Project Management (SPM) with Software Change Management (SCM) to achieve the following objectives: Firstly, optimizing the Change Management Process in a project and track the whole Change Requests. Secondly, monitoring if the software project finished early by revising the project plan w.r.t the actual schedule to shrink the project deadline. And execute the consequences processes according to finish early. Thirdly, the actual results with performance will be tracked against the project plan and any variances should be analyzed. In addition, facilitating distribution of tasks (Change Requests) and availability of information. Finally, providing more control over a project when the technical issues can be raised and turned to managerial issues to take corrective actions if: the project takes more time, the project takes more cost or both of them

General Terms

[Software Engineering]: [Software Project Management] and [Software Configuration Management]: [Software Change Management].

Keywords

Software Project Management (SPM), Software Change Management, Software Configuration Change Board (SCCB), Change Requests (CRs), Work Breakdown Structure (WBS), Change Request Breakdown Structure (CRBS).

1. INTRODUCTION

In general, when an organization introduces a new software project or introduces a new requirement or version or any change that occur in the project. *The first step* is the project manager will divide the software project into tasks. A *Task* it is a planned activity and it is an individual work item that defines a piece of work to complete a project. All tasks

contain a task id, name, a definite estimated duration to complete it and resources.

Each task in the project will be transformed to a change request. A *Change request* is a formally submits artifact associated with the task included in the project plan It can be Add Requirement, Update Requirement, Delete Requirement, New Version, New Release along the project lifecycle. [1]

The second step is the project manager will compose the tasks into smaller to construct the work breakdown structure (WBS). Then the change management team will transform the task into Change Request (CR) .they construct the Change Request Breakdown Structure (CRBS). A **CRBS** It is a graphical chart that despite detailed planning for a Change Request by subdividing the project Change Request into smaller, manageable CRs. The task will be transformed into Major Change Request (MCR) and each MCR spawns from it smaller Sub Change Request (SCRs) implemented on this Chart .These SCRs efforts can be mapped to the individual responsibilities.

The third step, the project manager will build up a network diagram for the tasks and tasks transformed into CRs

Each CR has the following [2]

- **ES - earliest start time:** the earliest time at which the CR can start given that its precedent CR must be completed first.
- **EF - earliest finish time:** equal to the earliest start time for the CR plus the time required completing the CR.
- **LF - latest finish time:** the latest time at which the CR can be completed without delaying the project.
- **LS - Latest start time:** equal to the latest finish time minus the time required to complete the CR.

The slack time for a CR is the time between its earliest and latest start time, or between its earliest and latest finish time. Slack is the amount of time that a CR can be delayed past its earliest start or earliest finish without delaying the project. The objective of this paper to put forward our personal thoughts on some of the best practices in Software Project Management integrated with Software Change Management to increase the software quality, minimize the project duration and cost.

2. PREVIOUS WORK

There are many change management models. Each one has its own procedure for how to treat with the change request through the project life cycle.

Olsen's Change Management Model [3] presented software development process in both development and maintenance phases. This model views all work done by software designers as change and it focuses on real things, such as paper and files. It couples the critical roles played by marketing, design, programming, training, documentation, support, testing, quality assurance, and manufacturing.

Ince's Change Process Model [4] covers change activities during both the software maintenance and development phases. And proposes variation according to the type of change, and suggests the size of the change to be the criterion for defining the validation strategy. It relates the Configuration Management activities, which are configuration identification, configuration control, status accounting and configuration auditing, to the change process.

The Spiral Change Process Model [4] divides the change management process into four cycles, in which the same main tasks are performed by each cycle, but the viewpoint is different in each cycle. It does not treat software change as post-delivery activity. It is relevant not only in the maintenance phase but also to handle changes during the development phase of life cycle. And it only addresses the action performed for changing existing pieces of work, not creating new artifacts.

The Olsen model treated all the works as a Change in the Model treats a task as a Major CR (MCR) to be started to implement and each MCR spawns from it Sub CRs (SCRs) Linked directly to the CRBS (MCR and SCR) for each task. So the change management team can trace every CR. Each hierarchy of CR has its access rights and deliverables. Ince model doesn't support all improvement ideas of software developers or all new ideas but in the proposed model all ideas are supported as a change request but need acceptance from SSCB to be within time and budget of the project. The spiral change model doesn't support Different types of changes in the proposed model all types are supported (add new requirement, update a requirement, delete a requirement, new version, new feature and new release) and it works on existing piece of work or create new.

3. THE PROPOSED FRAMEWORK

The framework will present all project processes integrated with the software change management process

3.1 Project Initiation Phase

First – initiate CR: the management will conduct the Software Configuration Control Board (SCCB) to check clarity of CR. If the CR is clear the SCCB will discuss CR. If not clear the SCCB will return the CR to the originator of the CR.

3.2 Project Planning Phase

Second – Evaluate CR

After dissection of the CR and the approval on it. The Project Manager will start creating the Project plan's tasks and will construct the work Breakdown Structure (WBS) The Change Management Team (CMT) will turned the major tasks into Major Change Request (MCR) and its Sub Change Request (SCR).and then construct the Change Breakdown Structure (CRBS). And the Project Manager Assigning the team of the CR according to their tools and experiences.

Once the CRBS had been created the next step is to construct the project network.

For Each CR

- Estimate the planned completion time for each CR.
- Identify for each CR (ES, EF, LS, LF, slack).
- Identify the critical path (longest path through the network).
- Sum all the slacks and put them in the Time Accumulator (\sum total slacks) = TA
- Estimate its Planned Cost (Estimation Cost).
- The Duration of MCR_i is equal to Summation of duration of SCR_i .

$$d(MCR_i) = \sum d(SCR_i)$$

3.3 Project Execution Phase

Third – Execute CR

Execute the CR and Unit test and turned to the tester to check validity if no returned it to the responsible if yes the CR will be submitted. The CR will be change its status, put into the change tracking sheet, the responsible will record the actual time of CR. Then calculate its time variance, cost variance and then put in the cost and time accumulators. The framework will demonstrate how to compute the time and cost variance in next section framework implementation.

3.4 Project Monitoring and Controlling Phase

Project Control function is involves comparing actual performance with planned performance and taking corrective action to get the desired outcome when there are significant differences. By monitoring and measuring progress regularly, identifying variances from plan, and taking corrective action if required, project control ensures that project objectives are met.

The Change Management Team helps to get opportunities for improvement. Improvements implements by using the Cost Accumulator (CA) and Time Accumulator (TA) to indicate the variances between planned and actual in the following cases.

3.4.1 Cases of the CR

Case 1: the CR finishes early

The Change Management Team will check the summation of duration of all smaller CRs (SCR_i) if it is less than or equal the Early Finish (EF) of the Major of CR (MCR).

$$\text{If } \sum d(SCR_i) \leq (MCR_i)_{EF}$$

The Change Management Team will Notify the PM the CR will finish early using time accumulator and calculate its saving cost and ask him to start early the next CR and if he need to update the Project Plan.

Case 2: the CR finishes late

The Change Management Team will check the summation of duration of SCR_i If it is greater than or equal the Actual Latest Finish of Major CR(MCR_i).

$$\text{If } \sum d(SCR_i) \geq (MCR_i)_{LF}$$

The Change Management Team will subtract the delay from Time Accumulator of slacks TA

- Check if there is value in the Time Accumulator there is no action needed.

- Check if there is No value or subtract in the Time Accumulator the change management team must report to the next level in the project hierarchy take actions.
 - Adding new project resources.
 - Multiple shift work.
 - Overtime/extended workdays.
 - Increasing # of workers.
 - Using more, larger or more efficient equipment.
 - Using Training courses.
 - Outsourcing.
 - Negotiate with sponsor of the project to exceed the Deadline.

The change management team will responsible to calculate the cost variance and subtract it from the cost accumulator and check if it is within the contingency plan then no action is need if no must report the cost Variance to the next level in the project hierarchy.

3.4.2 Cases of the project

Case1: if the project finishes early and there are saving in cost

$$TA = \sum TS_i$$

If $TA \geq \text{Deadline of the project}$, $CA > 0$

If the Time Accumulator (TA) is greater than the deadline of the project and Cost Accumulator (CA) is positive. The Change Management Team will report to the next level to take the true actions if there are some operational processes after completion of the project depend on the deadline of the project and report the saving cost by using the cost accumulator.

Case 2: if the project finishes late and there are no saving in cost

$$TA = \sum TS_i$$

If $TA \leq 0$ and $CA \leq 0$

If the Time Accumulator (TA) is less than or equal zero and the Cost Accumulator (CA) is less than or equal zero. The Change Management Team will report to the next level to take the true actions. if the project needs to use crashing Methods and also report the cost losses using cost variance.

Case 3: if the project finishes on time and there are saving in cost

If $TA = 0$ and $CA > 0$

If the Time Accumulator (TA) is equal to zero and the Cost Accumulator (CA) is greater than zero. The Change Management Team will report to the next level to take the true actions with saving money. If the project needs other action outside the project using the Cost Accumulator.

3.5 Project Closing Phase

In this phase the CR will be closed. For each Sub CR spawn from Major CR has a Team Leader (TL) for it. The Team Leader should encourage the team members to be finished his

Sub CR early because it will affect the duration of the Major CR to be delivered in Early Finish and awarded each team member who finished early and The Change Management Team will put him in the "Employee Performance Sheet" and punishments when they delays. The CR will be closed and update the "CR Tracking Sheet" to sustains the history of the CRs

If there is a problem in one CR. The responsible should change the CR status to work on it but there are several steps to that change

- **First:** reschedule of the project plan by adding parallel process and add more resources.
- **Second:** update time variance and cost variance.
- **Third:** execute the "Execute CR Block" and all rest blocks.
- **Fourth:** close the CR and then update the Tracking sheet.

4. SYSTEM ARCHITECTURE

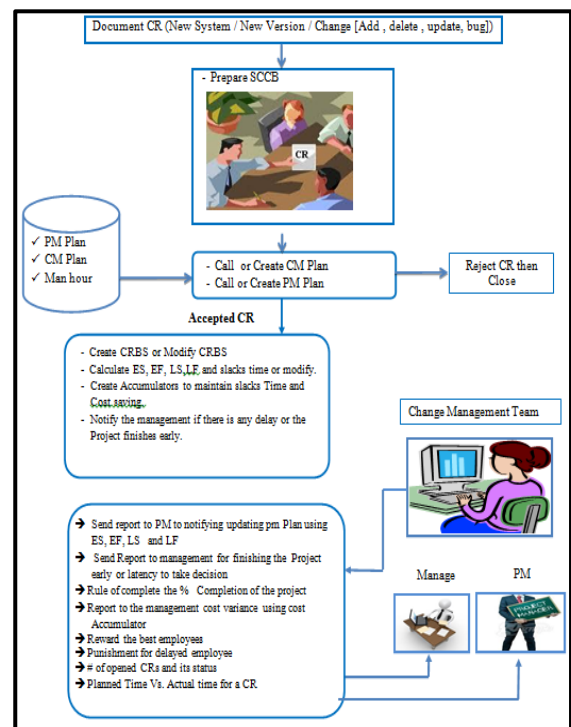


Fig. 1 system Architecture

5. FRAMEWORK IMPLEMENTATION

The following algorithm used to find the longest path of the network and calculate ES, EF, LS, and LF for each CR and add on it how to calculate the Time and Cost Accumulator. this algorithm based on the Dijkstra's Algorithm (Dijkstra [5] , Kelley, J.E. and Walker [6], R. Morovatdar, A. Aghaie, E. Roghanian, A. Asl Haddad [7]). Let us denote

- $p_i, p_i = \{p_i^l | l=1, \dots, |p_i|\}$ the set of longest paths from the CR up to i th event.
- CR_i has ES, EF, LS, LF, Duration (d_i), and Slack (S_i)
- $Predi$ is the set of active predecessors realizing the i th event

- Succ_i is the set of active successors realizing the *i*th event
- TA Time Accumulator Variable of all CR[i] ,
- CA Cost Accumulator Variable of all CR[i] ,
- N #number of activities

```

Read CRs and its duration, pred, suc
1: CR1 (ES) =1, And CR1 (EF) = CR1 (ES) + d1 -1
2: For CRi =2 to N
  forward pass      //
3:  CRiES ← max {CRprediEF + 1}
4:  CRiEF ← max {CRiES +di-1 }
5: next i
Backward Pass      //
6: CRLF [n] =CREF [n]
7: CRLS [n] =CRLF [n] - dn + 1
8: for N to 2
9:  CRi ← min {CRh + dhi | h ∈ predi}
10: succi* ← {h | (h ∈ succi) AND (CRi = CRh - dhi )}
11: next i
      // for calculating slack
12: Si = CRLS [i] - CRES [i]
13: If Si = 0
14: Pi ← CRi
15: For CRi:1 to N
16: CRTime Variance = CRPlanned Duration - CRActual Duration
17: if (CRTime Variance >=0 ) then TA += CRTime Variance
18: if (CRTime Variance < 0) then TA -= CRTime Variance
TA = Sum (CRTime Variance) 19:
20: If (TA <= Max (CRLF[i]))      // Deadline
21: Message to the management "the Project crashed and
"we need for Crashing Method
)22: Otherwise if (TA >= Max (CRLF[n])
23: Message "You Can Revise Project Plan PP" Alert to
PM
24: Recalculate CP[i]
25: CRCost Variance = CRPlanned Cost - CRActual Cost
26: if (CRCost Variance >=0 ) then CA += CRCost Variance

```

```

27: if (CRCost Variance < 0) then CA -= CRCost Variance
CA = Sum (CRCost Variance) 28:
29 : Message the management with the Cost
Accumulator value
30 : End for

```

As shown in the above algorithm the first step is to determine the set of tasks then. Draw network diagram that shows the CRs and its scheduling dependencies and determine for each CR ES, EF, LS, LF, and SLACK. Determine the deadline of the project and Find the critical path in step 13 and 14 then calculate Time Accumulator (TA) in step (15 to 19) If total slack near to be zero or equal zero the change management team will notify the Management the project will be crashed before crashing they should take corrective actions to use Crashing methods or adding resources step (20 to 21) If Time Accumulator greater than or equal Deadline of the project. The change management team reports the project manager to revise the project plan step (22 to 24) and then calculate cost accumulator (CA) in steps (25 to 29).

Finally, the change management team should continually evaluate all employees during the project. In other words. The employee who will deliver his work lately should be punished. On the other hand the employee who will deliver his work on time should be awarded.

6. EXPERIMENTAL RESULTS

the proposed framework tasted on real projects plans. The framework applied on five real project plans with actual times and costs are assumed. After that for each project do evaluation and analysis for such projects plans.

6.1 Case1

Fig.2 shows the actual and planned Time and cost for packages system [8].

After applying the framework on this case the figure illustrates the project manager will be notified to review his plan according to the time accumulator. The project will finish earlier than its deadline with 6 Days and with 430 saving money.

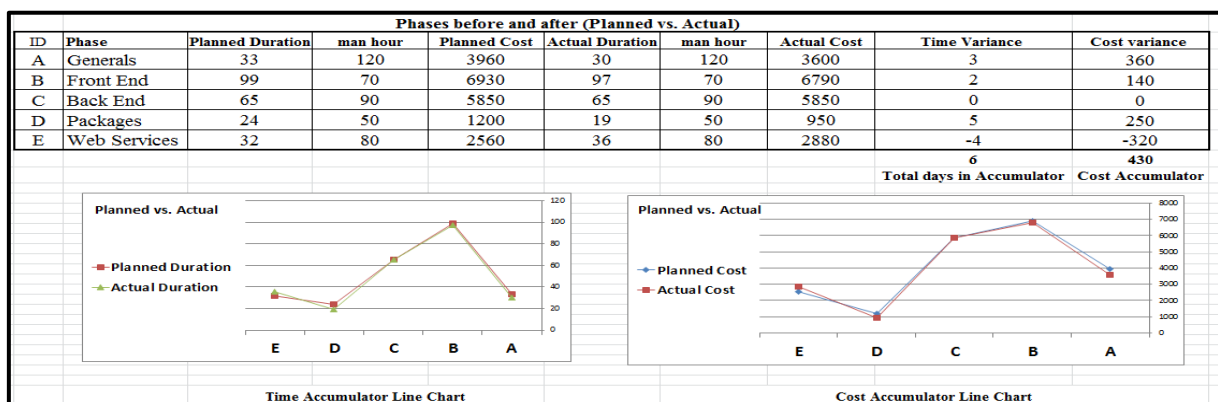


Figure 2: planned and actual time and cost for packages system

6.2 Case 2

Fig.3 shows planned and actual time and cost for an application software development for specific company. with 167.5 saving in money using the time accumulator and cost accumulator.

In this case here there are saving also in money and time the project will finish earlier than its deadline with 6.5 Days an

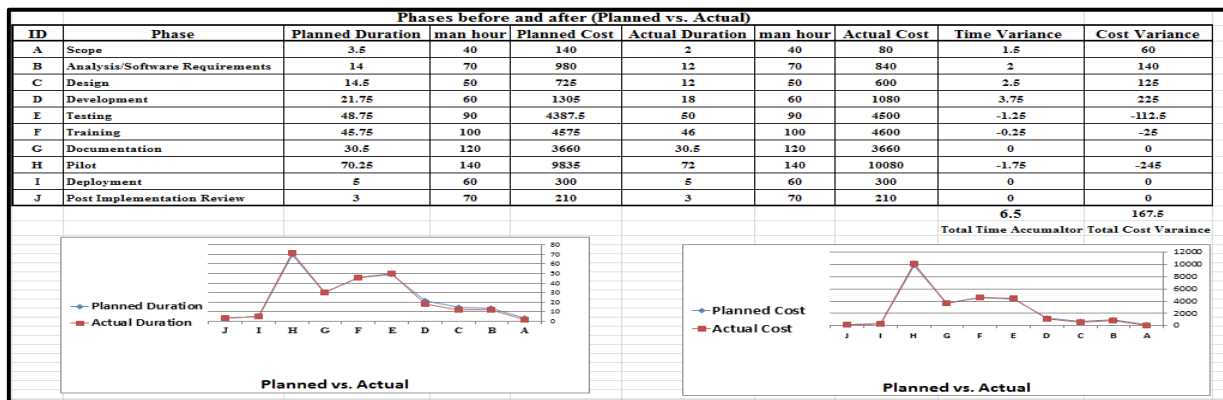


Figure 3: Planned and Actual Time and Cost for an application software development

6.3 Case 3

Fig.4 shows a planned project plan for developing a product

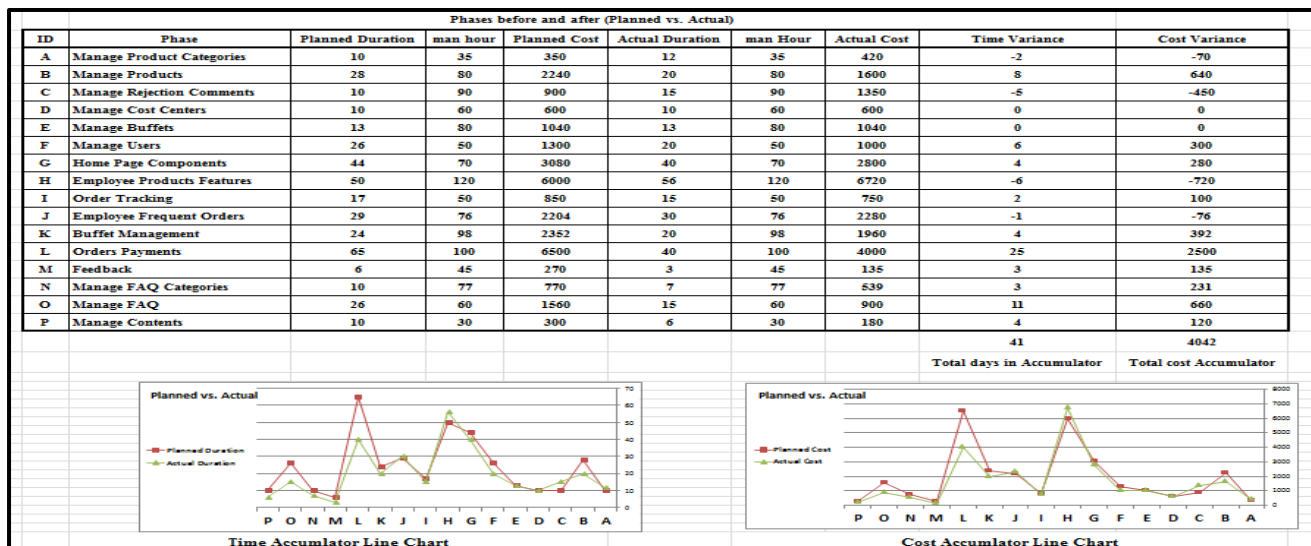


Figure 4: planned and actual time and cost for developing a product

After executing the framework on this case also the Time accumulator will guide the project to finish early with 41 days and saving money 4042 by using cost accumulator.

Actions can be

- Adding new project resources
- Multiple shift work
- Overtime/extended workdays
- Increasing # of workers
- using more, larger or more efficient equipment
- Using Training courses
- Outsourcing
- Negotiate with sponsor of the project to exceed the deadline

6.4 Case4

Fig.5 shows a planned and actual time and cost for transactional survey projects. After executing tasks with Actual Duration for CR. the time accumulator will guide that there are shortage in the first CRs in their duration (negative signs) then the change management team will notify the management to take actions to avoid increasing in deadline of the project with 66 days and loss in cost 6227.

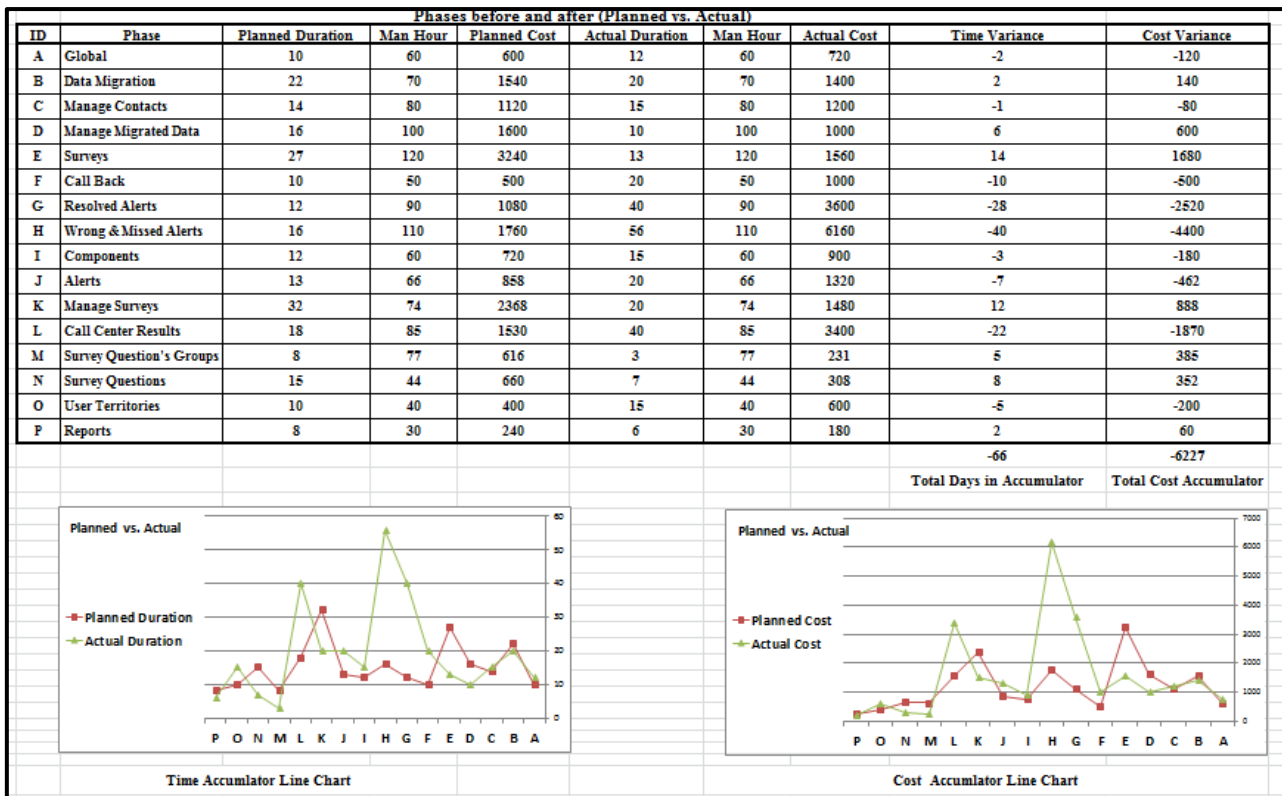


Fig.5 Planned and Actual Time and Cost for transactional survey projects

6.5 Case5

Fig.6 shows a planned project plan for a call center system[9].in this case after executing the framework will guide that there are shortage in the duration (negative signs)

60 days in the time and shortage in cost 270. So the change management team will notify the management to take actions to avoid increasing in deadline of the project and using crashing method.

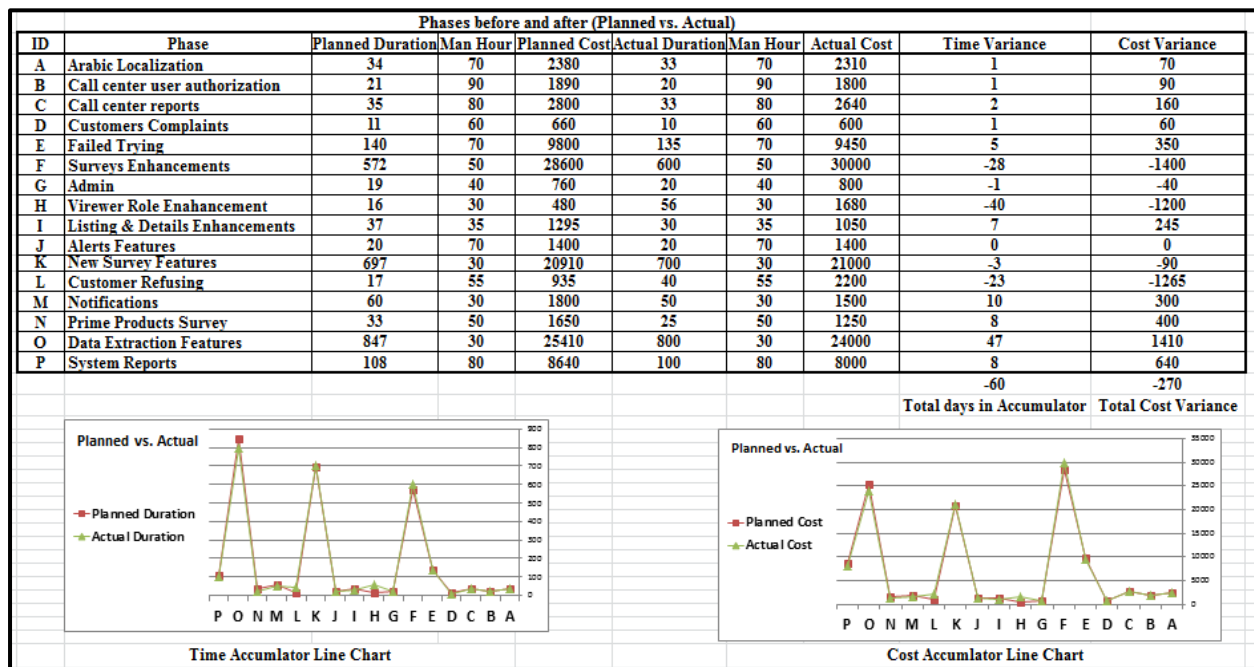


Fig.6 Planned and Actual Time and Cost for a call center system

The following Fig. 7, shows the histogram for the time accumulators and Cost accumulator for the above five project cases w.r.t its planned duration vs. actual and % complete for each project w.r.t. its planned duration

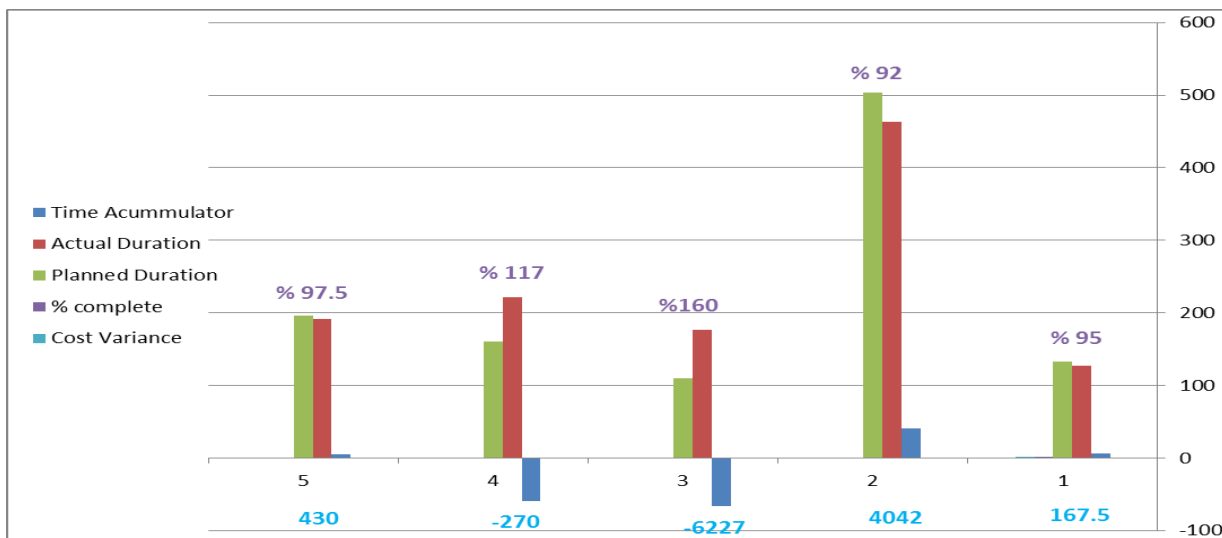


Fig.7: the histogram for the time accumulators and Cost accumulator for the above 5 project cases

7. CONCLUSION

As a result this paper presents the following As a results for implementing such proposed framework first, The change management team helps to monitor the activities of the project they Prepares and delivers reports to the next level of management on performance and improvements when they check each responsible if he/ she finished their CR within the EF Duration of the CR recorded this responsible in a sheet of awarded and punishment them if they late their CR. Second, By using the accumulator the change management team helps the project manager in monitoring tasks. They notify the project manager if any CR finished early revising the project plan and notifying him if there is Time extra or delay for the project and the cost variances for both cases. Third, the change management team responsible for notifying the next level if the project will finishes late using the time accumulator indicator to avoid increasing in deadline of the project. Finally, if there are corrective actions should be taken. The Change Management Team Notify the Management to take the True Actions if there are some operational processes after completion date of the project depend on the deadline of the project.

8. ACKNOWLEDGMENTS

I would like to express my deepest thanks and appreciation to all the people who helped me. First of all; I would like to thank god for supporting me and I would also like to thank my supervisors, Dr. Shereif Mazen and Dr. Ehab Ezzat, for coaching me throughout working on the paper.

9. REFERENCES

- [1] A. R. M. Nordin , S. Suhailan. 2009. Managing Software Change Request Process: Temporal Data Approach. International Journal of Computer Science and Security, (IJCSS) Volume (3) : Issue (3).
- [2] www.mosaicprojects.com.au
- [3] Olsen, N. 1993. The software rush hour. In IEEE Software Magazine. IEEE. Pp. 29 – 37. (September).
- [4] S. M. Ghosh , H. R. Sharma, V. Mohabay. 2011. Analysis and Modeling of Change Management Process Model.
- [5] Dijkstra, E.W. 1959. A note on two problems in connection with graphs, Numerische Mathematik, 1, 269-271.
- [6] Kelley, J.E. and Walker, M.R. (1959), Critical path planning and scheduling, Eastern Joint Computer Conference, 16, 160-172.
- [7] R. Morovatdar, A. Aghaie, E. Roghanian, A. Asl Haddad. 2013. An Algorithm to Obtain Possibly Critical Paths in Imprecise Project Networks", *Iranian Journal of Operations Research*, Vol. 4, No. 1, pp. 39-54
- [8] GET EGYPT, Inc. (2009) www.getgroup.com
- [9] Mantrac Unatrac Group, Inc. (2013). www.mantracgroup.com.