Analysis & Recommendations for the Management of COTS - Computer Off The Shelf - Software Projects

MANSOOR AL-A'ALI[1] & SEEMA AL-MAHMOOD[2]

1. Department of Computer Science
College of Information Technology
University of Bahrain
PO Box 32038
KINGDOM OF BAHRAIN

2. Bahrain Telecommunication Company
KINGDOM OF BAHRAIN

Abstract: - The objectives of software project management are usually to minimize the project duration, to minimize the project cost, and to maximize the software quality. This paper presents the findings of an investigation into project management issues pertaining to COTS - Commercial Off-The-Shelf Software - projects. The paper discusses these findings and how they can be solved and what lessons can be learnt from this challenging process. The results of this investigation relating to three major COTS projects indicate that the obstacles faced by information systems project managers are related to COTS vendors and the consumers' problems. Based on these results we developed a number of recommendations which could be followed to ensure the success in the management of information systems projects.

Keywords: Project Management, Information systems, COTS systems, user requirements, software management

1 Introduction

Current software projects usually demand complex management involving scheduling, planning and monitoring tasks. There is a need to control people and processes, and to efficiently allocate resources in order to achieve specific objectives while satisfying a variety of constraints [1]. The objectives are usually to minimize the project duration, to minimize the project cost, and to maximize the product quality [12] & {3]. The problems of software project management are so complex that many researchers have resorted to using even genetic algorithms for help [16]. Too often software projects or product releases fail due to misunderstanding or misinterpreting customer needs and later on coping with associated changes in the designated time period or within the budget [9].

The success of any software project depends on the skills and competences of its project manager. The project manager holds responsible for software requirements, release definition, software release lifecycles, creating an effective multifunctional project introduction team and – above all – preparing and implementing the business case [5]. Yet, software project management is complex: there are many stakeholders, many responsibilities and no formalized education or body of knowledge. The project manager aims at having the required system mix and the right implementation strategy. The project manager evaluates the products or product releases with respect to their overall contribution to business success. He makes use of the product life cycle to revisit assumptions and implement decisions [5]. The product manager leads and manages one or several products from the inception to the phase-out in order to maximize business value [6] & [10]. Product management is closely linked to requirements engineering, because it is up to the product manager to plan and prioritize requirements into roadmaps, releases and projects [12], [6] & [4].

Project risk management is an importance issue in software project management. Uncertainties faced by software projects should be taken into account when planning and controlling the development of software systems [8]. Project risk level is the probability of a project's failure in achieving its proposed goals [8]. The risk level summarizes, in a single number, how risky a project is. By quantifying the risks and highlighting which project aspects may be more risk-prone, managers can better identify where to apply their limited resources in an attempt to reach projects goals.

The term COTS is the acronym generally used to describe commercial products. It commonly refers to things that one can buy, ready-made, from some manufacturer's store shelf. COTS products are sold, leased or licensed; offered by a vendor trying to profit from it; supported and evolved by the vendor, which retains the intellectual property rights; available in multiple identical copies and used without internal modification by a consumer [13].

COTS - Commercial Off-the-shelf- software systems have been widely used by many organizations as a means of developing Information Systems (IS) with lower risks, lower costs and shorter development schedule while increasing the functionalities and the capabilities of the system. The management expectation is that COTS system components are easy to implement to work in different environments and to customize to the organization's local requirements though this is not entirely true in real life situations. IS project managers find that managing this type of software acquisition projects is a challenge and carries many risks. IS project managers face many problems during the life cycle of the project, some of these problems may cause the project to fail, to increase the project costs significantly - unexpected costs, and not to meet their special requirements. Starting with these facts, investigating problems that IS project managers' face in managing software acquisition projects represents a substantial issue for study and research.

COTS projects normally go through a number of phases: contracting and procurement issues, product and vendor (evaluation, selection, testing and support), product modifications and tailoring, product evolution, vendor relationship, project management, software acquisition methodology, product familiarization and business processes.

The move to COTS products caused a fundamental change in the way organizations do business. This change can be described as "paradigm shift". The essence of the paradigm shift is that organizations will change from a producer to a consumer. In the case of a producer, developers create the implementation of the component or system. It can be viewed from every aspect because they have control over all its features and functions. For example, the code written to implement a software component is known. In the case of a consumer, the organization purchases a COTS product and in this case, developers only view the product's interfaces. In fact, the product is much like a black box because the developer cannot see how the box functions, but they can see what it produces and receives. The producer approach consists of: (identify requirements, define unique interfaces, develop custom implementations, integrate custom implementations, and use and support system of developed implementations). The consumer approach consists of: (identify requirements, adopt standards interfaces based on market research, procure implementations based on standards, integrate procured implementations, and use and support systems of procured implementations).

The most visible difference lies in stage 2 and 3. Instead of defining unique interface specifications, the consumer adopts standard interface specifications. The specifications will be selected as a result of market research. Instead of developing implementations, the consumer procures implementations that are based on standard interfaces. These implementations will be standard-based COTS products. The fourth and fifth stages manifest differences resulting from the shift to a consumer approach and may include the following:

- Integration of product is different, and possibly more difficult, because the consumer has less insight into the product. In some cases, a vendor may not want its products to integrate with others. The vendor may do this to retain market dominance.
- Support of the product is different because the consumer may use commercial sources to support the system.

The following COTS process was summarized from [14]:

Requirements Analysis, System Requirement Review, Package Identification Evaluation/Selection, Identify Glueware and Integration Requirements, System Design Review, Non COTS Development – Write Glueware and Interfaces, Integration and Testing, Target System Installation and Acceptance Test, Discrepancy resolution, and Sustaining Engineering.

In his study about the process of COTS software development, Morisio [14] discussed some project management issues like project budget and schedule. Project estimation and tracking both have to consider new activities. Estimating their duration is currently a complex task due to the limited amount of existing experience and the few estimation models available, but budget estimation is easier to estimate based on modifications to the effort accounting procedures. In his study Morisio [14] also discussed the new skills that COTS project activities introduced. This means, for example, that employees must be trained in the administrative, commercial, and other non-technical issues that come up during vendor interaction or be given the necessary support to handle these concerns with the vendor. Other such new skills include COTS evaluation and integration. In addition, the study mentioned a suggestion for projects that face problems in this area and the suggestion is to develop a COTS team, this team can be a group or a person, depending on the size of the organization, should concentrate on the COTS-related skills and activities. Single projects cannot afford to build these skills individually. The team acts as a repository of history, knowledge and skills about COTS, and offers them to projects as a consulting activity. Examples of skills are: evaluation and selection of COTS, history of COTS evaluations, COTS usage, and procurement skills.

Another important issue in project management planning is contingency planning in schedule and for budget. The budget should include contingency for other costs such as additional work needed, consulting services or training. All of these should be allowed for by the contingency plan [7].

Pressman [15] also discussed the importance of planning in the software engineering process. The software project plan is produced as the culmination of the planning tasks. It provides the baseline cost and scheduling information that will be used throughout the software process. The software plan is a relatively brief document that is addressed to a diverse of audience. It must (1) communicate scope and resources to software management, technical staff and the customer; (2) define risks and suggest risk aversion techniques; (3) define cost and schedule for management review; (4) provide an overall approach to software development for all people associated with the project, and (5) outline how quality will be ensured and change will be managed. It is important to note that software project plan is not a static document. That means it should be updated regularly – updating risks, estimates, schedules and related information as the project proceeds.

2 Factors in COTS Project Management

Based on the literature survey and our own experience, we have put together a number of factors which should be considered in software project management, especially in CTOS projects. For any project manager embarking on a COTS software project, he/she must consider these factors and measure the potential success of these factors before embarkation and after software delivery and live run, see Fig. 1. Further, we recommend that the project manager should decide on weighting systems for each of these factors so that he/she can measure the influence/success of each of these factors and consequently the success level of the overall project. In a scale from 0 to 1, the project manager can give a weighting to each factor, eg, 0.2 for factor 1.1, 0,2 for item 1.2, 0.1 for item 2, and so on.

3 Discussion

The three projects analyzed were successful in terms of not exceeding the budget allocated to the project. This indicates that properly planning and controlling the project budget as stated had positive impact in not exceeding the budget allocated to the project. This finding meets Morisio [14] opinion in that the project budget tracking is not complicated to accomplish in software acquisition projects and also reflects the ability of project managers to maintain project expenses within the budget allocated.

The findings also show that projects P1 and P3 were behind the planned schedule. This can be explained that estimating the project duration is a complex task due to the limited amount of existing experience and the few estimation models available (Morisio' et al. (2002)).

On the other hand, Hallow [7] mentioned the importance of maintaining a contingency plan for the project, which the three projects failed to meet. The three projects managers did not have a budget and schedule contingency plans for the project. Lack of budget contingency plan was not considered a critical issue in the project planning phase because the three project budgets were sufficient and managed properly. This was clear in that the three projects did not exceed the allocated budget. But lack of contingency plan if the project off- schedule reflects the lack of formal software acquisition project methodology on the consumer side. Pressman [15] insists on the importance of the software project plan to include: risk plan, estimates and schedules plans and quality assurance (QA) plan. Which were identified in the three projects.

With regard to the project plan, projects P1 and P2 maintained an updated project plan that was agreed on with the vendor's project manager. In project P3 case, the IS project manager created and maintained the project plan. However, in the beginning of the project,

1. ***Success of project in terms of:***
    1. *Within budget vs. over budget*
    2. *Within schedule vs. behind schedule?*
2. ***The contingency plans in case the project is off schedule or off budget***
3. ***Risk management plan***
    1. *Does the project plan articulate a list of current risks to the project?*
    2. *Is the list updated during the life cycle of the project?*
    3. *Is there a process in place for reviewing and updating project risks?*
    4. *Did the project have a risk manager who was responsible for identifying emerging risks to the project?*
5. ***People management***
    1. *Does the project team have all the technical expertise needed to complete the project?*
    2. *Does the project team have expertise with the business environment in which the software will operate?*
    3. *Does the vendor team have all the technical expertise (and business expertise) needed to complete the project?*
    4. *Are there enough people (manpower resources) to do all the work required by the consumer?*
    5. *Is there a training plan to meet the training requirements of those involved within the project? Was it sufficient?*
    6. *Does the project manager have experience and knowledge in managing acquisition software projects?*
    7. *Does the project manager attend training courses specialized in managing software acquisition projects?*
6. ***QA and Project Plan***
    1. *Is there a quality assurance (QA) plan maintained and followed? Which QA method was followed e.g. ISO, in-house developed?*
    2. *Does the project have a detailed, staged software delivery plan, which describes the stages in which the software will be designed, implemented and delivered? Is it updated regularly?*
    3. *Does the project plan include time for public holidays, annual leave, sick days, and ongoing training, and were resources allocated at less than 100% (were they involved in other projects)?*
    4. *Is the project plan and schedule approved by the project team? Key business stakeholders?*
    5. *Which parts of the project plan (planning process) is :*
        - *The weakest.*
        - *The strongest.*

Fig. 1: Proposed successful COTS project management factors

the manager did not agree with the vendor about the plan because the vendor's project manager did not have any project management knowledge, they did not know how to create a basic project plan. This led to create incomplete plan that was only a list of tasks without mentioning important phases of the project (e.g. missing testing plan). In addition, because the project plan did not include the training courses required for the project team, this caused the project to be behind the planned schedule. In project P1, the plan was changing very frequently because the vendor project team allocation was not planned properly and vendor underestimated duration required to compete project tasks (e.g. a task that required 7 working days was underestimated to 3 working days only!). Another reason was that some vendor's staff was removed from the project without replacement and without also notifying Customer's IS project manager.

QA Plan:

Pressman [15] discussed the importance of software quality assurance in identifying the conformance of requirements; this point was mentioned by the project managers during the interviews, where they said that the QA had an impact on the requirements conformance. The findings of projects P1 and P2 show that the project managers benefited from the QA plan in identifying the percentage of user requirements compliance as mentioned in section 3. Project P3 did not include the QA plan as part of the project plan, and this may lead to some requirements being not met during the project but had to be accommodated after the project completion. In addition, lack of QA could be referred to lack of Customer formal software acquisition project methodology.

Risk Plan:

The three project managers have included risk management plan as part of their project plan, they were maintained in a list or issue log forms. In project P2 case, the project manager linked updating the QA plan with the risk management plan to control the risks that may threaten the quality of the software.

Project Team Members:

1. Customer's Project teams

In projects P1 and P3, Customer's IS project teams had a good technical and business experience required to complete the project. In case of project P2, the team did not have the technical expertise required in SAP R/3 for the project, this was built during the project, but they had background in the business area. The unavailability of technical Customer IS project team was due to the fact that SAP R/3 was a new system that requires a very specialized technical background that was only provided by the product owner, this background was provided to the project team during the project as mentioned below in the training section.

Both projects P1 and P2 were staffed enough and key business users were also involved in the project. Project P3 had shortage in IS project team and shortage and business users. The reason for having this problem in project P3 is that the project manager did not plan well for the project staff required for the project both in technical and business areas; which forced the project manager to utilize other IS projects team members. The reason for not having enough business users was due to their reluctance to commit resources to the project.

2. Vendor Project Teams

Concerning the vendors' project team, in project P1, vendor A provided sufficient number of staff but they were not qualified and lacked the required technical background to complete the project. But in vendor B case, a few staff were provided for the project but they were qualified and able to complete the project. In vendor A case, the following problems indicate the lack of the vendor's experience and expertise:

- The vendor resisted to make tailoring to the product to meet some requirements and in some cases provided weak solutions with many gaps which clearly indicate a lack of technical expertise.
- The huge number of incidents that were reported during.
- The resolution of the acceptance incidents for phase 1 of the project was slow thus hindered the Customer from conducting the re-test for ultimate cut-over to the new system.

In project P2, the vendor project team was sufficient and qualified in technical and system functionalities areas, which was evident in the ability of the vendor to make the required modifications and handle the errors created easily.

Project P3 vendor's team was not qualified; they did not have technical background and knowledge in business area that were required for working with the project.

- The vendor did not understand the Customer's requirements well (Product-Vendor evaluation and selection)
- IS project team had to conduct testing and provide data input for testing without the vendor support (Product-Vendor evaluation and selection)
- Many errors were generated due to vendor's mistakes while trying to make some modification to the product (Modifications and Customizations)

Customer Project Team Training: The three projects had a training plan to meet the training requirements of the project team. In project P1, sufficient in-house training covering functional and technical courses were provided to the project team. In project P2, as mentioned above, the lack of technical background was solved by providing informal training (in-house) to project team during the project life cycle. But the project manager attended formal specialized training courses about SAP R/3. The project manager said "The training plan was insufficient for the project team, if we planned it properly, we would progress faster in the project." While in project P3, external training was provided to the project team. The project manager pointed that it was sufficient at that stage, but because it was not included in the project plan, the training caused a delay in the project.

Project Managers Skills: Project managers for P1 and P2 did not attend training courses specialized in managing software acquisition projects; only general project management courses were provided, however, project P3 manager attended a course that focused on managing software acquisition projects. This illustrates that three project managers were not trained in new professional skills imposed by software acquisition project management like vendor management and commercial and other non-technical issues [14].

4. Recommendations:

1. Develop a standard software acquisition methodology and communicate it to the IS projects teams.
2. Develop detailed contingency plans in case the project was behind schedule.
3. Both projects managers (vendor and Customer) should participate in preparing the project plan (schedule – including QA and risk plan) and both parties should agree and follow this plan.
4. The project manager should evaluate and select the vendor's team to ensure that they have the necessary technical and business skills required for the project.
5. Develop COTS team. Develop a group that should concentrate on the COTS-related skills and activities. This group acts as a repository

of history, knowledge and skills about COTS, and offers them to projects as a consulting activity. Examples of skills are: evaluation and selection of COTS, history of COTS evaluations, COTS usage, and procurement skills [14].

6. Provide necessary training related to COTS products and software acquisition project management.
7. Apply the successful proposed project management factors and establish appropriate weightings in order to measure the level of success of the project management process.

4 Conclusion

The research presented in this paper demonstrated that managing COTS software projects is a series complex and interrelated tasks and require knowledge, commitment, planning, budgeting and allocating responsibilities. The paper presented a number of factors which must be considered in order to properly manage a COT project. A number of findings were identified and presented as a result of researching the management of three major projects in Bahrain. Based on these findings, we presented a number of recommendations which if followed; we believe will eliminate any problems in a future major COTS project. Our future research will be to apply these findings and recommendations to a new COTS project in order to measure the success which could result from these finding and recommendations and will enable us to further tune them.

References:-

1. Alba Enrique, J. Chicano Francisco, Software project management with Gas, *Information Sciences*, 177, 2007, pp. 2380–2401

2. Assmann, D., Punter T., Towards partnership in software subcontracting. *Computers in Industry Journal* 2003.

3. Chang C.K., Christensen M.J., Zhang T., Genetic algorithms for project management, *Annals of Software Engineering*, 11,2001, pp. 107–139.

4. Davis, A.M., 2005. Just Enough Requirements Management, Dorset House, New York.

5. Ebert Christof, The impacts of software product management, *The Journal of Systems and Software*, 80, 2007, pp. 850–861

6. Gorchels, L., The Product Manager's Handbook: The Complete Product Management Resource, third ed., McGraw-Hill, New York, 2006.

7. Hallows, J, Information Systems Project Management, 1998, USA, AMACOM

8. He´lio R. Costa a, Marcio de, Barros b, Guilherme H. Travassos, Evaluating software project portfolio risks, *The Journal of Systems and Software 80*, 2007, pp. 16–31

9. Heckman, R, Organizing and managing supplier relationships in information technology procurement, *International Journal of Information Management*, 19, 1999.

10. IEEE Std 1220–2005, IEEE Standard for Application and Management of the Systems Engineering Process. IEEE New York, NY, USA.

11. Lander, M, Trust-building mechanisms utilized in outsourced IS development projects: a case study, *The Journal of Information and Management*, 41, 2004.

12. Martinich, J., Production Operations Management, 1997, USA, John Wiley & Sons, Inc

13. Meyers, B, Oberndorf, P., Managing software acquisition: Open systems and COTS products, 2001, USA, Addison Wesley.

14. Morisio, M., COTS-based software development: Processes and open issues, *The Journal of Systems and Software*, 61, 2002.

15. Pressman, R (), Software Engineering, 2000, UK, McGraw-ill Publishing Company

16. Ruhe G., Greer D., Quantitative studies in software release planning under risk and resource constraints, in: Proceedings of the International Symposium on Empirical Software Engineering, IEEE Computer Society, Roman Castles, Rome, Italy, 2003, pp. 262– 270.

17. Yardley, D, (2002), Successful IT Project Delivery: Learning the Lessons of Project Failure, UK, Addison Wesley.