

Mechanism to Ensure Success in Software Projects

R. Udayakumar, K.P. Kaliyamurthie, Khanaa and A.V. Allin Geo

School of Computing Science, Bharath University, Chennai-73, India

Abstract: Projects have a high rate of failure. More specifically, software projects often do not comply with the traditional standard measurements of success, namely time, cost and specifications. The questions that are discussed in the paper are related to software project management, the kind of management system to be use in the project, manager and developer communication, project planning, management system changes and management of team performance & Soft Skills in Software Project Management Such skills, acquired through experience, are concerned with managing and working with people, effective co-ordination, ensuring customer satisfaction and creating a conducive environment for the project team to deliver high quality products within budget and on time and exceeding expectations.

Key words: Project risks • Project management • Managerial roles • Technical issues and risk control

INTRODUCTION

The failure rate of software projects has proven to be very high and the incidence of failure has been getting worse as more and more companies venture into software development. The Failure factors work as risk to the project. Risk management is a collection of methods aimed at minimizing or reducing the effects of project failure [1, 2]. The aim of Risk management is to take counter measures to prevent risks from affecting the project or to reduce their impact. Software project risk has been defined as the product of uncertainty associated with project risk factors and the magnitude of potential loss due to project failure [3, 4]. Thus, the key elements that need to be controlled are the project risk factors. The relative importance of risk factors needs to be established in order to reduce the failure rate in software projects [5].

A project is usually deemed as successful if it meets requirements of measures such as functionality, reliability, maintainability, portability, efficiency, integration and operability [6]. It needs to be delivered on time and within budget. Research proves that only one-sixth of all projects were completed on time and within budget, one-third of all projects were cancelled and over half were considered 'challenged'. Reasons for the high failure rate

are many but the main reason is on account of managers not taking measures to assess and manage the risks involved in software projects [7].

The aim of this report is to highlight software project risks and measure of controls to be adapted to reduce risk. The aim is to study the significant risks and controls that can be applied to reduce their occurrence and minimise their impact on projects [8]. The effectiveness of different controls to reduce the occurrence of risk factors was also identified and studied.

MATERIALS AND METHODS

Efforts have been made to incorporates techniques and guidelines to identify, analyse and control software risk [9]. Project Risk Management should be viewed as a fundamental component of the project management process By including risk management in a project the exposure to software risk can be reduced and can thereby increase software quality and improve software development. Identifying of project risks has been done with the help of interacting with project managers who have dealt with projects faced several risks.

Research Objectives: Understanding of the factors that contribute to software project risk

Effect of the size, complexity and strategic importance of software projects currently being developed.

Effect of managerial in competency on projects failure and Whether technological in competency plays a bigger role or managerial in competency.

Types of Software Project Risks: The risks listed below have been identified by many researchers and regularly occur in software projects [10]. These risks are considered to be issues that need to be addressed and thereafter controlled.

Top Ten list of the software risk item.

- Unclear or misunderstood scope/objectives
- Misunderstanding the requirements
- Failure to gain user involvement
- Lack of senior management commitment
- Developing the wrong software functions
- Unrealistic schedules and budgets
- Continuous requirement changes
- Inadequate knowledge/skills
- Lack of effective project management methodology
- Gold plating

Unclear or Misunderstood Scope/objectives: The different stakeholders in a software development project have their own objectives, which often conflict with the objectives of another stakeholder [11]. For instance, users require a robust, user friendly system with many functions that can support their tasks while development team members on the other hand hope to encounter interesting technical challenges. These differing expectations, create fundamental conflicts when simultaneously approached, resulting in unclear or misunderstood scope/objectives of the project.

Unrealistic Schedules and Budgets: The 'scheduling and timing' risk is the major complicating factor as it is difficult to estimate schedules with acceptable accuracy and consistency. Very often, organizations embark on a large project having underestimated its size and complexity. This risk leads to the difficulties in scheduling the project correctly. The performance with scheduling and timing risks improves with project experience. A fixed schedule may lead to schedule pressures and people under pressure do not necessarily work better, resulting in the inability to produce satisfactory results or deliver any software at all [12].

Lack of Senior Management Commitment to the Project:

The lack of senior management commitment is most critical risk. A project succeeds only when senior management makes it a top priority as management support influences a project's process and progress [13].

Failure to Gain User Involvement: If users are not involved, there is the risk that developers may assume detailed functionality and business requirements, leading to project objectives not being achieved. It also means laying the blame for 'lack of client responsibility' on the project manager rather than on the users.

Inadequate Knowledge/Skills: Project personnel may not have adequate knowledge of the technology, or the business, or may just not have the experience to handle the project. 'People' risk arises from inadequate skills (both technical and managerial) as well as level of experience. The lack of experience with technology also increases the likelihood of this risk occurring Unrealistic expectations of personnel's abilities may affect the project's outcome.

Lack of Effective Project Management Methodology:

Those software projects that are implemented and do reach the operational mode are often poorly rated by the users. The main reason for this is the developers not using a good methodology [14].

Misunderstanding the Requirements: It may be time consuming and difficult to collect and record all of the required details from all prospective users, resulting in the project team not knowing enough about what is required to complete the project successfully This may lead to the possibility of developing a system that cannot be used mainly because a proper systems analysis to develop a complete and accurate set of requirements has not been performed [15].

Gold Plating: Often developers and analysts think of additional capabilities or changes, known as gold plating, which they think would make the system better and more attractive in their view. These deviations may result in unsatisfied users and unnecessary costs.

Continuous Requirement Changes A 'continuous stream of requirement changes' was identified as a significant risk. As the users' needs change, so do the requirements of the project. By freezing a part of the functionality and

delivery date, completion of the system is enabled. But requirements should not be frozen because in today's fast moving business environment, a frozen design does not accommodate changes in the business practices. With a frozen design, the developer has little flexibility in changing the specifications. Continuous and uncontrolled changes in requirements, however will inevitably lead to a delay in the project schedule.

Developing the Wrong Software Functions: This risk is less likely to arise when project managers have higher experience and level of education. This risk occurs for the reason that users make changes or improvements to incorporate new factors that may arise and this may not be communicated well to the project team.

Subcontracting: This risk is important because of the consequences in managing contracted tasks and the shortfalls in externally developed components.

Resource Usage and Performance: Incorrect evaluation of performance requirements can result in an inability to implement the system as a result of inappropriate technical solutions and computing power. Larger organizations tend to experience resource usage and performance risk more often since they develop complex systems that can lead to poor performance of the resulting system.

Introduction of New Technology: This risk occurs by using new or 'leading edge', technology that has not been used successfully at other companies. This risk may also increased further if there is a shift in technology during the project.

Failure to Manage End User Expectations: Expectations determine the success or failure of a project. If expectations are mismatched with the deliverable, problems are created. Expectations should be correctly identified and constantly reinforced in order to avoid failure.

Software Project Controls: Software project management has become a critical task in many organizations. With new risks occurring whenever a project is implemented, there needs to be a formal method of controlling them. A control system for software projects needs a measuring method to detect the activities performed; a mechanism for comparing activities

performed with a standard or expectation of what should be carried out; a procedure for changing behavior if there is a need; and a feedback method or mechanism. If control systems such as this are not in place, people are inclined to report favorable information and to withhold unfavorable information. Development efforts have been cancelled not for inadequate programming or for lack of technology but from the sheer frustration of all concerned in attempting to determine when, if ever, they would be completed.'

The controls below have been identified and are considered important in reducing the project risks.

Developing and Adhering to a Software Development Plan: A list of all known potential and relevant risks should be included as a section in all software project plans. The goals should be set for determining when each risk item has been satisfactorily controlled. Each mitigation action should be assigned to an individual for implementation and monitoring of effectiveness, together with a target date for completion. The most visible top ten risks should be maintained and regularly tracked for effectiveness of the mitigation approaches.

Combine Internal Evaluations with External Reviews: Internal evaluations combined with external reviews should be performed to keep the project on the right track. This control applies to risks of inappropriate or insufficient staffing, lack of effective development process methodology, poor estimation and improper definition of roles and responsibilities. This step should include reviews of the strategic business plan of the company as well as a technology assessment. Along with the external analysis, a review of the internal strengths and weaknesses of the IS department should also be conducted to keep the project on the right track. The key is to recognize the risk areas early enough so that project managers can take appropriate preventative actions.

Involve Management During the Entire Project Life Cycle: Project managers should be able to create and maintain long-term relationships with users and promote user commitment to the project. Project managers should periodically determine the level of commitment from senior management and the user community so that support for the project does not evaporate. The primary job of the software project manager is to structure the project to meet the 'win' conditions of various stakeholders.

Involve Users During the Entire Project Lifecycle: Managers should not expand the scope of the project without consulting the users and revising the project budget and schedule.

Ensure There Is a Steering Committee in Place: An executive steering committee can be an effective means to ensure that management processes are established and maintained. To be effective, the steering committee should therefore have the ultimate authority on the approval of projects. Regular meetings should be held to report on the status of ongoing projects and communicate concerns, issues and possible remedies.

Assign Responsibilities to Team Members: Clearly define roles and responsibilities for each team member to contribute fully to the development of the project. In this way the manageable unit of work can be decreased and more tasks identified, resulting in improved supervision of individual activities.

Develop Contingency Plans to Cope with Staffing Problems: The accurate projection of required staff levels has proven to be an absolute critical function in software development. Overstaffing may lead to higher communication and coordination overheads, which translates into lower unit productivity. On the other hand, under staffing often leads to project delays, volatile priorities and inadequate testing. As a project progresses through its life cycle, the number of people assigned typically increases.

Include a Formal and Periodic Risk Assessment: Risk assessment consists of three activities: identifying the risks, analyzing them and assigning priorities to each of them. By analyzing the risks, it is easier to understand when, why and where the risks might occur. Assigning priorities enables the project team to devote their limited resources only to the most threatening risks. For most IT projects, risk assessments are usually conducted on an infrequent and informal basis, very few organisations perform any formal risk assessment to see if risks can be controlled, reduced or managed. The management of the implementation process is often complicated by the time and effort that practitioners can devote to it. Since most organisations follow some type of software development methodology, it would be much easier to include a formal and periodic risk assessment as part of the methodology for developing systems. In this way, it is not seen as another process that has to be accomplished but instead

as part of the methodology. Risks should be assessed at the beginning of the project and at the review of each phase, which should result in a decreasing exposure since they will be identified and resolved.

Divide the Project into Controllable Portions: While most managers know that large projects must be broken down into incrementally smaller work elements, few actually dissect them to the level necessary to effectively manage them in sufficiently distinct work elements.

Educate Users on the Impact of Changes During the Project: While implementing project plans, they often fail to view, listen and communicate effectively. To ensure that problems, issues and risks are brought to the appropriate managerial levels, the project manager should ensure that open communication is maintained with both the users and senior management.

Assess Cost and Schedule Impact of Each Change to Requirements and Specifications: To avoid the problem of scope creep, project managers should inform users of the impact of scope changes in terms of project cost and schedule. Project managers should be able to distinguish between desirable and absolutely necessary functionality.

Stabilize Requirements and Specifications as Early as Possible: Many projects are faced with uncertainty when requirements are first stated. Requirement issues need to be resolved as the project progresses. The project manager should provide the leadership and vision and make sure that the users of the system are involved in defining and stabilizing its detailed requirements as early as possible.

Avoid Having Too Many New Functions on Software Projects: Stress the need to implement new functionality on an incremental basis rather than multiple new functions being incorporated simultaneously.

Review Progress to Date and Set Objectives for the next Phase: Project's progress should be frequently compared to the project plan. Knowledge of the risks of failure inherent in a particular situation can enable practitioners to develop strategies (based on organizational experience and relevant theories) to reduce or overcome the influence of setbacks. From the above, it is clear that very few control factors relate to technical issues. A majority of the issues are concerned with communications and commitments (or rather the lack thereof) among people.

CONCLUSION

It has been seen that experienced project managers use certain controls more than inexperienced project managers, particularly 'assign responsibilities to team members' and 'stabilize requirements and specifications'. This has helped to control project risk to a very great extent. Therefore, following of procedural norms and methods may result in lesser project risk.

Managerial competencies are most important as explained during the course of the paper and competent project managers have had lesser failure in their projects as compared to managers with less experience and competency.

The Role of the top management is very crucial. The support of the management to the project manager and his decisions is a must. The project manager need to be given the authority to take decisions on critical issues during the course of the project in order to avoid delays.

Managerial competency plays a bigger role than technological competency in successful project management.

REFERENCES

1. David J. Bryde, 2005. Method for Managing Different Perspectives of Project Success _ British Journal of Management, 16: 119-131.
2. Lynn Crawford, 2006. Developing Organisational Project Management Capabilities-Theory and Practice-Project Management Journal-p74-86
3. Smith, D., M. Eastcroft, N. Mahmood and H. Rode, 2006. Risk Factors Effecting Soft ware Projects in South Africa-Soutyh African Journal of Business Management, 37(2).
4. Mary Sumner, 1999. Critical Success Factors in Enterprise Wise Information Management System Project- ACM, pp: 1-58113-063.
5. Software Engineering-Rodger Pressman-Tata Mc Graw
6. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. Application of Soft Computing Techniques in weather forecasting : Ann Approach, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1845-1850.
7. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. Improving Web Information gathering for personalised ontology in user profiles, Middle-East Journal of Scientific Research, ISSN:1990-923315 (12): 1675-1679.
8. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. Detection of Material hardness using tactile sensor, Middle-East Journal of Scientific Research, ISSN:1990-9233 15(12): 1713-1718.
9. Saravanan, T. and R. Udayakumar, 2013. Comparision of Different Digital Image watermarking techniques, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1684-1690.
10. Saravanan, T. and R. Udayakumar, 2013. Optimization of Machining Hybrid Metal matrix Composites using desirability analysis, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1691-1697.
11. Saravanan, T., V. Srinivasan and R. Udayakumar, 2013. Images segmentation via Gradient watershed hierarchies and Fast region merging, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1680-1683.
12. Udayakumar, R., V. Khanna, T. Saravanan and G. Saritha, 2013. Retinal Image Analysis Using Curvelet Transform and Multistructure Elements Morphology by Reconstruction, Middle-East Journal of Scientific Research, ISSN:1990-9233, 16(12): 1798-1800.
13. Udayakumar, R., V. Khanna, T. Saravanan and G. Saritha, 2013. Cross Layer Optimization For Wireless Network (Wimax), Middle-East Journal of Scientific Research, ISSN:1990-9233, 16(12): 1786-1789.
14. Udayakumar, R. and Kumaravel A. Rangarajan, 2013. Introducing an Efficient Programming Paradigm for Object-oriented Distributed Systems, Indian Journal of Science and Technology, ISSN: 0974-6846, 6(5S): 4596-4603.
15. Udayakumar, R., V. Khanaa and K.P. Kaliyamurthie, 2013. Performance Analysis of Resilient FTTH Architecture with Protection Mechanism, Indian Journal of Science and Technology, ISSN: 0974-6846, 6(6): 4737-4741.