International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-6, Issue-11)

Research Article | **November 2017**

Controlling Task Dependency as a Risk Mitigation Process

Mohamed S M Hasni*
Kastamonu University, Institute of Science,
Dept. of Material Science and Engineering, Turkey
Email: mhasni@ogr.kastamonu.edu.tr

Turhan Köprübaşı
Kastamonu University, Institute of Science,
Department of Mathematics, Turkey
Email: tkoprubasi@kastamonu.edu.tr

Abstract—

The construction of highly reliable software projects requires that these projects meet the objectives for which they have been set, and therefore failure to meet any of these objectives and needs will make the software project prone to failure. Thus, ignoring strong needs and not including users adequately especially in identifying their needs and tracking their dependencies clearly can be considered one of the main risks of failure. In this paper, we present a systematic approach to addressing this phenomenon and its mitigation mechanism.

Keywords— Risks, Risk mitigation, UML, Dependency, Software requirements.

I. INTRODUCTION

A lot of projects fail because of inaccurate planning and management, in addition to project complexity which is considered considered as a major factor for failure due to budget and schedule issues. Therefore, identifying the major risks and their root causes is an important process to mitigate their effects for a project to be successful. Examples of these risks are: planning, poor project definition and not recognizing the existing problems, poor or missing resources, interdependencies between projects, technical, commercial, unstable regulations and rules, as well as customers change of requirements [1].

In this paper, a systematic approach is proposed to mitigate such risks especially in the work team organizations.

A. Project risks

All projects are exposed to risks, which may arise from the multiplicity of resources, change in management and environment. Thus, risk mitigation requires a clear understanding of its causes and extensive assessment for the impacts [2].

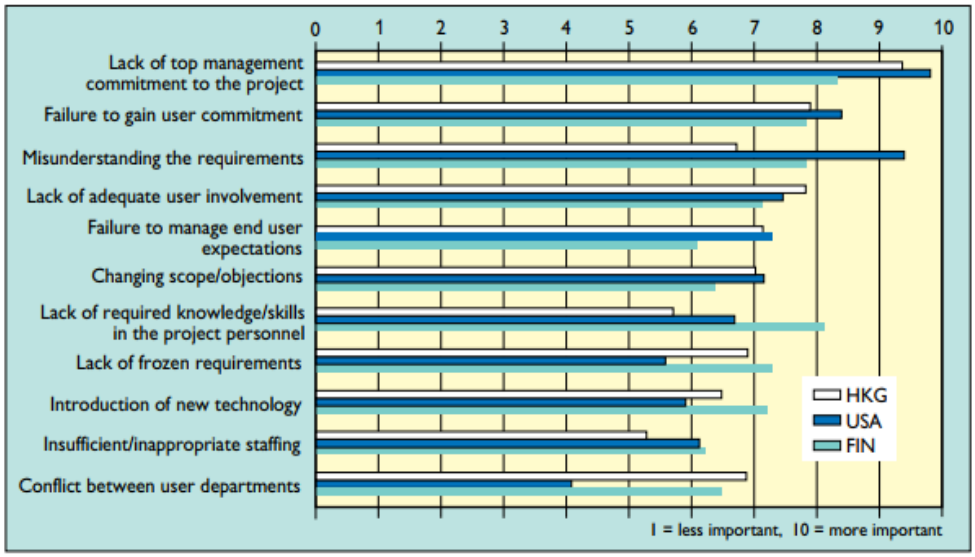

Figure1. Most 11 risk factors and causes [6]

Number of efforts are put to mitigate and deal with risks, and most of them are focusing on the sources of which risks arise [3, 4].

Software risk management was presented by Boehm in the spiral model [5], where the process of risk handling consist of: (1) identifying, analysing priorities of risks, and (2) planning and controlling these risks.

In addition to, most 11 risk factors are identified in [6], figure 1 illustrates these factors.

From the above figure, it is clear that lack of management commitment and misunderstanding of project requirements are the most important factors that lead a project to be in risk, affecting its acceptability and operations. Therefore, users' involvement may contribute to treating these issues by increasing top management commitment through derived requirements that follow a project. This means, a clear approach to improve the degree of project success and to mitigate these risks as essential requirement for a project's success.

II. TASK DEPENDENCY AS A RISK MITIGATION PROCESS

In order for any project to be successful, it should satisfy the needs of its users. Furthermore, the project must respond to its users' demands promptly. So, a thorough study of the needs of users becomes necessary. These needs represent the backbone of building functional requirements that must be met by the software system to be developed. However, these needs, even if they are gathered clearly, the dependency among these needs are considered as an obsession that must be taken into account in the case that the construction of the software system requires several individuals (team members). Therefore, we intend to focus on teams distribution to be based on the process of tasks dependency. Fig 1 illustrates the conceptual overview of the proposed approach.

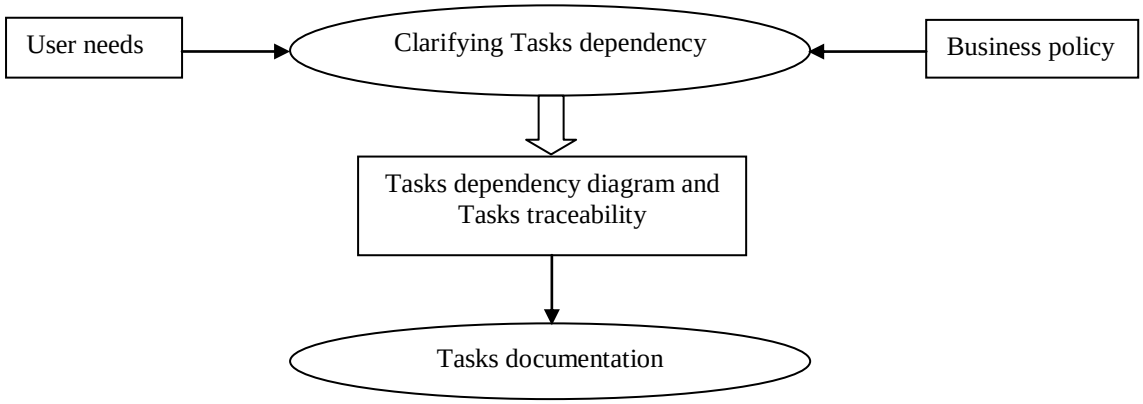

Fig. 1 Conceptual overview of controlling exceptions dependency

A. Clarifying Tasks dependency

The goal of this stage is to identify the needs of users by studying the nature of their work and the duties they perform, in addition to a deep knowledge of the rules of work that govern each task and its relationship to other tasks.

To facilitate the traceability process, a visual description will lead to the success of that process and create opportunities for negotiation between the system's stakeholders and the development team, and then resolve any ambiguities that may arise.

The process of distributing development teams is a technical process that is not primarily of interest to the stakeholders, who are interested in obtaining a system that fulfills all their needs according to their identified budget and time constraints. Therefore, the project manager must seek these needs and put into account the process of dependency among these tasks for the optimal distribution to the members of his team, which are developing the system.

- ***Task dependency model***

Based on the purpose of clarifying tasks dependency in a visual manner, we suggest using the UML use case diagram to explain these dependencies, thus, these users are reassured that the development process can follow the needs they wish to be included in the project, and for the project manager who can distribute his team easily according to these relations. Figure 2 illustrates an example of the task dependency model.

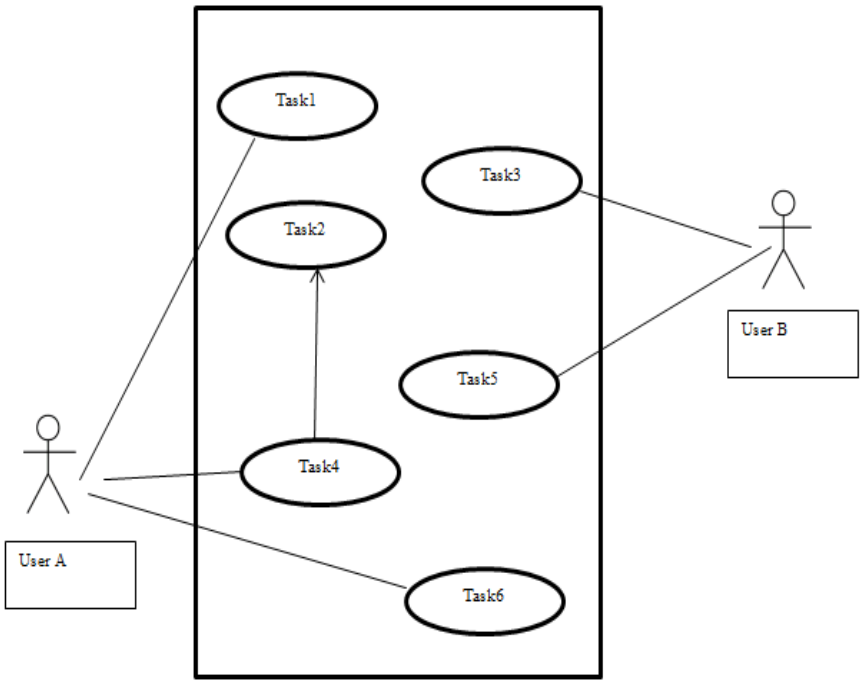

Fig2. An example of Task dependency using use case diagram

Form the above figure; it is clear that two users are considered. Each of them has their responsibilities. Dependency between these uses cases are shown throw the arrows directions. Firstly, we have adopted the concepts of this diagram with some adjustments, so that our focus here on the overlapping of roles of users and dependency between these tasks in order to emphasize the interest in the reliability of the tasks, which will lead to two main things: (1) The roles of users can be seen and subsequently verified, and (2) This dependency is clear for the project manager and also will help him to distribute the work among the project team.

- ***Tasks traceability***

The second and basic thing after understanding the users' needs in the form of tasks dependency, the process of monitoring the development of these tasks becomes necessary, and this can be controlled only through the tracking of the important tasks, the dependencies and arrange priority among them. Therefore, tracing tasks means that any shortage or change in any of them may affect the tasks on which they depend on, and therefore may require modifications on the affected task or addition of new user needs that was ignored or uncaptured before. Table 1 illustrates an example of the tasks dependency that is captured from the Figure 1.

Table1. Task dependency table

	Task1	Task2	Task3	Task4	Task5	Task6
Task1						
Task2						
Task3						
Task4		√				
Task5				√		
Task6						

From this simplified table, that shows the dependency of the tasks described in Figure 1. It is noted that tasks 1,3 and 6 are independent tasks, this gives project manager the opportunity to propose any development mechanism he deems appropriate such as parallel development. While it is more complicated in other tasks 2,4 and 5 where it is noted that there are dependencies that restrict the freedom of developing them without taking into considerations these dependencies.

B. Task documentation

To complement tasks tracking process, documenting these tasks are an important process to serve as a starting point for documentation of functional requirements later. Table 2 illustrates an example of the document format.

Table2. An example of Task documentation form

Task number		**Task name**	
List of dependant tasks			
Task scenario			
1.			
2.			
3.			
4.			
5.			
.			
.			
.			

III. CONCLUSION

The development of any successful software system depends on the good management of the stages of its development, especially if there are several members in a working team involved. Failure to track the dependency of the tasks required to be accomplished and the impact of the change between them is an important concern that may lead to the failure of the development process, especially in the absence of a mechanism to track the roles of the team and the tasks they play.

In this paper, we have presented a systematic approach that considers neglecting task dependencies as a risk that must be treated. We have adopted the UML diagram to clarify the tasks of the team members, and then the task traceability table to show dependency among those tasks. This approach helps project managers in distributing or redistributing tasks among their team members in accordance with those dependencies. In addition, the adoption of this approach can contribute in the future to the development of a mechanism for the distribution of teams, even if they are geographically distributed.

As a future vision, we seek to enhance this approach that take into account geographical distributed teams and how to improve their performance, taking into consideration risk-mitigation mechanisms that may arise also when the teams are geographically distributed.

REFERENCES

[1] Thamhain, H., Managing risks in complex projects. Project Management Journal, 2013. 44(2): p. 20-35.
[2] Hillson, D., Using Risk Management for Strategic Advantage. 2003, Retrieved from the Project Management Institute Website: http://www. pmi. org/learning/risk-management-strategic-advantage-tactics-7727.
[3] Akintoye, A.S. and M.J. MacLeod, Risk analysis and management in construction. International journal of project management, 1997. 15(1): p. 31-38.
[4] Kwak, Y.H. and S. Dewan, Risk management in international development projects. The George Washington University, 2001.
[5] Boehm, B.W., A spiral model of software development and enhancement. Computer, 1988. 21(5): p. 61-72.
[6] Keil, M., et al., A framework for identifying software project risks. Communications of the ACM, 1998. 41(11): p. 76-83.

