Knowledge of IT Project Success and Failure Factors:
Towards an Integration into the SDLC

Walid Al-Ahmad, Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

ABSTRACT

Traditionally, project success/failure is considered only after the project is completed or cancelled. Integrating project success and failure factors knowledge and software engineering activities would result in a situation where project success/failure is considered as part of the development process, leading to more successful software projects. This article aims to identify the common issues responsible for IT projects' success/failure to develop a deeper understanding of these root causes. Knowledge about success can be used to understand failure and vice versa. Therefore, generic taxonomies of the root causes are developed for that purpose. Knowledge of these taxonomies is integrated into software development and management activities to help software developers and project managers complete projects successfully.

Keywords: IT Project Management, Knowledge Management, Project Auditing, Project Success/Failure Factors, Software Development Life Cycle (SDLC)

INTRODUCTION

Failures of IT projects have been discussed for several years and many studies concerning failure/success have been conducted. There are studies that have tried to categorize the success or failure factors (Heeks, 1999; Lytyinen & Hirschheim, 1987; Standish Group, 1995). However, there are no studies that address the issue of incorporating these factors into the Software Development Life Cycle (SDLC). A reason for the lack of such studies is due to the fact that software development and project management developed independently.

A comprehensive study conducted by Standish Group (1995) addressed the IT project success/failure issue. The study included several thousand IT projects and revealed that only 16% of those projects were finished on time and within the estimated budget, and 32% were terminated before they were completed. The remaining 52% of the projects involved costs higher than the original estimates and were completed behind their schedule. Another study conducted in the UK by Oxford University and Computer Weekly in 2003 reported surprisingly that only 16% of the IT projects reviewed were considered successful (Sauer & Cuthbertson, 2003). The Standish Group study was repeated in 2000 and 2003 (Standish Group, 2001). There was a significant improvement in project success
rates (28% in 2000 and 34% in 2003). Success has been attributed, among others, to better software development processes, better project management, and better user involvement.

The literature review indicates clearly that failure of IT projects can be attributed to some generic root causes, the main categories of the project failure taxonomy. It also indicates that the success of IT projects can be attributed to generic root causes, the main categories of the project success taxonomy. This study reveals that any reason for IT projects’ success or failure should belong to one of the categories in the corresponding taxonomy independent of the application domain (banking, telecommunication, healthcare, human resources, etc.) Thus, creating taxonomies of the root causes for project failure and project success is a key objective of this research work.

IT projects have been considered a tough undertaking and have certain characteristics that make them different from the rest of engineering projects, increasing the chances of their failure. Such characteristics are classified in seven categories (Peffers et al., 2003; Salmeron & Herrero, 2005):

1. Abstract constraints which generate unrealistic expectations and overambitious projects.
2. Difficulty of visualization which has been attributed to senior management asking for overambitious or impossible functions and the late detection of problems (intangible product).
3. Excessive perception of flexibility which contributes to time and budget overrun and frequent requests of changes by the users.
4. Hidden complexity which involves the difficulty to be estimated at the project outset and the interface with the reliability and efficiency of the system.
5. Uncertainty which causes difficulty in specifying requirements and problems in implementation of the specified system.
6. A tendency to software failure which is due to assumptions that are not thought of during the development process and the difficulty of anticipating the effects of small changes in software.
7. The goal to change existing business processes which requires IT practitioners understanding of the business and processes concerned in the IT system and good processes to automate and make them quicker.

The rapid pace of technological progress in software engineering hinders the expertise in a particular technique, tool, or methodology and creates a culture where the use of tools not completely tested is commonplace (Mahaney & Lederer, 2010). IT projects contain a greater degree of novelty than other engineering projects. In particular, IT projects related to product innovation development are extremely complex, risky and expensive endeavors (Cormican & O’Sullivan, 2004).

Capturing knowledge of IT project success/failure factors in generic taxonomies and integrating the taxonomies into software development lifecycles enables software developers and project managers to better relate these factors to the unique characteristics of IT projects. It also enables them to better understand how these characteristics influence project success/failure. Consequently, software developers and project managers will be able to implement successful IT projects. This is a kind of design for success approach to software development, i.e. a proactive approach instead of a reactive one. How to reach success and prevent failure should be built in software development activities.

The literature review indicates that IT project failure/success factors are similar in various IT domains: traditional information systems (database-oriented), Web-based systems, healthcare informatics systems, and so forth. Therefore, IT projects from the different domains can benefit from the taxonomies developed by this research.

It should be emphasized that the taxonomies presented in this article consider success/failure factors during project development rather than during the use of the software system. The focus is on the first four phases of the software life cycle (concept, development, implementa-
A Lag Effect of IT Investment on Firm Performance
[www.igi-global.com/article/lag-effect-investment-firm-performance/1285?camid=4v1a](www.igi-global.com/article/lag-effect-investment-firm-performance/1285?camid=4v1a)

Pareto Artificial Life Algorithm for Multi-Objective Optimization
Jin-Dae Song and Bo-Suk Yang (2011). *Journal of Information Technology Research* (pp. 43-60).
[www.igi-global.com/article/pareto-artificial-life-algorithm-multi/52817?camid=4v1a](www.igi-global.com/article/pareto-artificial-life-algorithm-multi/52817?camid=4v1a)