Chapter 76
Managing Software Projects with Team Software Process (TSP)

Salmiza Saul Hamid
Two Sigma Technologies, Malaysia & University of Malaya, Malaysia

Mohd Hairul Nizam Md Nasir
University of Malaya, Malaysia

Shamsul Sahibuddin
Universiti Teknologi Malaysia, Malaysia

Mustaffa Kamal Mohd Nor
University of Malaya, Malaysia

ABSTRACT

Despite the widespread use of sound project management practices and process improvement models over the last several years, the failure of software projects remains a challenge to organisations. As part of the attempt to address software industry challenges, several models, frameworks, and methods have been developed that are intended to improve software processes to produce quality software on time, under budget, and in accordance with previously stipulated functionalities. One of the most widely practised methods is the Team Software Process (TSP). The TSP was designed to provide an operational framework for establishing an effective team environment and guiding engineering teams in their work. This chapter provides an overview of the TSP and its associated structures and processes. It also highlights how the TSP operational framework can assist project manager and software development team to deliver successful projects by controlling and minimizing the most common software failure factors. Comparative analysis between the TSP and conventional project management has also been presented. Additionally, the results of TSP implementation in industrial settings are highlighted with particular reference to scheduling, quality, and productivity. The last section indicates additional advantages of TSP and comments on the future of TSP in the global software development project.

DOI: 10.4018/978-1-4666-4301-7.ch076

INTRODUCTION

In this day and age, many government organizations and information technology based companies develop and maintain software to support their daily operations. The software turns out to be their business product as well. The need for complex software products to support businesses operations are a very important issue nowadays. As projected by Boehm (2006), between now and 2025, the sustainability of the organizations and their products, systems and services are much depending heavily on software and this ever-increasing demands will cause major differences in the processes currently used to define, design, develop, deploy, and evolve a diverse variety of software-intensive systems. The statistics on software projects are discouraging, as there is high percentage of projects that fail, thereby not conforming to the requirements and causing deviations in time and cost. These result in poor quality products that lead to customer dissatisfaction.

In striving to address the software industries challenges, several frameworks and methods have been developed covering all aspects of improving project management practices and software processes purposely to produce quality software on time, under budget and within pre-agreed functionalities. One of the most widely practiced methods is Team Software Process (TSP), which has been implemented in wide range of organizations worldwide and gained positive results (Davis & Mullaney, 2003).

This chapter provides an overview of the TSP and its associated structures and processes. It also highlights how the TSP operational framework can assist project manager and software development team to deliver successful projects by controlling and minimizing the most common software failure factors. Comparative analysis between the TSP and conventional project management is also been presented. Additionally, the results of TSP implementation in industrial settings are highlighted with particular reference to scheduling, quality, and productivity. The last section indicates additional advantages of TSP and comments on the future of TSP in the global software development project.

BACKGROUND

Software Crisis

The term "software engineering" was coined at the first NATO Software Engineering Conference in Germany in 1968 (Naur & Randell, 1969) amid widespread consensus that there were problems with software development and maintenance. These problems were later discussed by Brooks (1975, 1987, 1995), and the term "software crisis" emerged to describe the software industry's inability to provide customers with high-quality products within schedule and under budget. Brooks concluded that there is no silver bullet to overcome this problem. Hardware costs were dropping, while software costs were rising rapidly. Major computer system projects were sometimes years late, and the resulting software was unreliable, hard to maintain and performed poorly.

Since the 1980s, in the medical field, for example, computers have been designed to help people, and most of the time, they do. However, in the case of Therac-25, computer errors could be fatal. Between 1985 and 1987, two people died and four others were seriously injured after they received massive radiation beamed via Therac-25 radiation therapy machines. Investigations revealed that defective software was among the various factors leading to this accident (Leveson & Turner, 1993). Another example is the delay of over 16 months in the opening of Denver International Airport, in addition to construction costs of over 100 million dollars in excess of the budget (Swartz, 1996). Indeed, *one main reason for the delay and overrun was the presence of major bugs in the baggage handling control software* (Glass, 1998). The explosion of the European Space Agency rocket Ariane 5 40 seconds after