Chapter 11

Team Software Process in GSD Teams: A Study of New Work Practices and Models

Adrián Hernández-López
Universidad Carlos III de Madrid, Spain

Ricardo Colomo-Palacios
Universidad Carlos III de Madrid, Spain

Ángel García-Crespo
Universidad Carlos III de Madrid, Spain

Pedro Soto-Acosta
University of Murcia, Spain

ABSTRACT

Distributed software development is becoming the norm for the software industry today as an organizational response to globalization and outsourcing tendencies. In this new environment, centralized models for software development team building models have to be reanalyzed. Team Software Process (TSP) guides engineering teams in developing software-intensive products and is intended to improve the levels of quality and productivity of a team’s software development project. In this paper, the authors assess the difficulty of using TSP in distributed software development environments. The objective of this assessment is twofold: firstly, know the general difficulty for using TSP in these environments, and secondly, know the caveats to be addressed in future software development team building models designed specifically for distributed environments.

INTRODUCTION

Software development is a collaborative and knowledge intense process where success depends on the ability to create, share and integrate information (Walz et al., 1993). This process has been evolving since its origins in the NATO Software Engineering Conference at end of the 60’s. During this evolution, new forms of software development have appeared, i.e., open-source software development (Lerner & Tirole, 2002) or outsourcing development (Sahay et al., 2003). Along with the
evolution of software development, the research about it has evolved and branched too. Moreover, an external but nearly related factor to software development process and its characteristics is still a top issue since, which are people (Gannon, 1979; Laughery, Jr., & Laughery, Sr., 1985); “People factors” seem to dominate “tools and techniques” as Blackburn et al. (1996) pointed in their study. Within people issue, other issues such as teams (Krishnan, 1998), people’s factors (Blackburn et al., 1996), and human resources management issues such as assigning roles (Acuña & Juristo, 2004), productivity management of IT projects (Mahmood et al., 1996) or skills identification (Colomo et al., 2010; Trigo et al., 2010) as top issues in software development research.

As a key issue for software development process, quality improvement within an organization is founded on three basic pillars: processes, technologies and people. These pillars are interconnected with each other, forming an inseparable triangle on which organizations operates. In turn, software process improvement requires an effort on three levels: organization, team, people (Humphrey, 2002). For each one of these levels there are well-recognized models for its continuous improvement and quality. At organizational level, models such as CMMI (Capability Maturity Model Improvement), which based on the premise that management is convinced of the need to improve processes, facilitates the integration of traditional functions, establishes objectives and priorities for processes improvement in the guide to quality processes, and evaluates current processes; at team level, models such as TSP (Team Software Process), which guides engineers in the creation and maintenance of self-managed software teams, which plan and monitor their work, set their goals and commit to compliance with the them (Humphrey, 2000); and at individual level, models such as PSP (Personal Software Process), which shows software engineers to manage the quality of their projects, agreeing that can meet commitments to improve their estimates and schedules and reduces defect rates in their products (Humphrey, 1995).

A software development team is defined as a team that has the implementation of a software project as its mission (Sawyer, 2004). Starting from the basic conditions presented above, Humphrey (2000) highlighted some necessary characteristics for a team to be considered an effective team: the team must be cohesive, it must have some ambitious goals, frequently receives comments on work done and should have a working framework for the tasks. In addition, globalization has led to the evolution of software development teams, and has created several types of teams with similar characteristics. Among these types are the GSD (Global Software Development) teams, geographically distributed teams, which use virtual communication channels such as email, and whose activity is focused on software development (Herbsleb & Moitra, 2001). These teams can be considered as a specification of the virtual teams (Martins et al., 2004), and are further encouraged by the relationships between customers of software development outsourcing organizations and developers (Heeks et al., 2001). GSD teams have a number of critical factors which must be considered before using a model for the management of teams in such environment. These factors are mainly the difficulty in communicating due to the geographical distances (Cramton, 2001), the cultural differences among team members (Ali Barbar et al., 2007), and the level of trust among team members (Jarvenpaa et al., 1998).

Finally, reports of TSP utilization are slim; see Callison and MacDonald (2009) for a complete bibliography. Specifically, reports of TSP utilization in distributed teams are limited to few cases; see Callison and MacDonald (2009). Distributed teams are a working style in outsourcing environments. In one of the reports of TSP utilization, Serrano and Oca (2004) assert that TSP is flexible enough to be used in outsourcing environments making some tunings but these changes were not