Guest Editors’ Introduction: Special Section on the Socio-Technical Environment of Software Development Projects

Marcelo Cataldo, Kate Ehrlich, and Audris Mockus

1 INTRODUCTION

Software engineering has long been recognized as a human activity that is managed through a system of methods, tools, processes, and work practices. This interaction between the nature of the product, the work, and the people doing it represents the socio-technical environment of a software development project. Software projects that fail to adequately consider the people’s social behavior in the context of the product and the work are less likely to be successful [2], [3]. The negative consequences of a neglected socio-technical environment are even more evident in geographically distributed software development, a project setting that has become commonplace in the last few decades.

A growing body of research has adopted the socio-technical perspective. However, the research is currently dispersed across multiple venues, including computer-supported cooperative work (CSCW), computer-human interaction (CHI), software engineering, and organizational theory and behavior. As a first attempt to integrate these strands of research for software engineering, we organized workshops at ICSE 2008 and ICSE 2009. This special section builds on the success of those workshops and highlights emerging and promising lines of research that have the potential to shape future research and practice in software engineering.

The socio-technical perspective has proven especially enlightening in revealing issues that affect coordination, awareness, and their consequences within software teams [1], [3], [4]. In particular, socio-technical congruence examines the match between the coordination requirements established by the dependencies among tasks and the actual coordination activities carried out by the engineers. Socio-technical congruence provides the theoretical grounding and a methodological framework for studying a range of questions related to communication, coordination, awareness, and collaboration in a wide range of software development settings. In the first paper, entitled “Does Socio-Technical Congruence Have an Effect on Software Builds Success? A Study of Coordination in a Software Project,” Irwin Kwan, Adrian Schröter, and Daniela Damian extend the socio-technical congruence approach to study the implications of coordination gaps on the success of software integration tasks. The authors studied the IBM Rational Jazz collaboration platform, which involved 151 individuals distributed across seven development sites. Their analyses examined the mapping between the technical dependencies associated with development tasks and the communication patterns among project members. They established that the link between socio-technical congruence and lack of failures in software builds was contingent on the type of software build task under consideration. In addition, their results provide insight into the complexity of the theoretical relationship between dependencies, coordination, and software development outcomes. For example, in this particular setting and under particular situations, higher levels of congruence between dependencies and coordinative action led to lower build success rates. The results reported by Kwan et al. highlight the complexity associated with understanding the theoretical linkage between dependencies, coordination, and software development outcomes.

The study of work practices in software development, particularly through data-rich qualitative approaches, has also been particularly valuable to the study of the socio-technical environment of software development projects. This line of work provides the basis for significantly enhancing our understanding of the relationship between tools, methods, processes, and individual actions so critical to the knowledge-intensive activities in software development. In the second paper, “The Awareness Network: To Whom Should I Display My Actions? And, Whose Actions Should I Monitor?” Cleidson R.B. de Souza and David F. Redmiles used established qualitative methods to articulate the expression and implications of awareness in three distinct software teams. They make an important theoretical distinction between general awareness and monitoring of individuals on a team and the overt signaling of behavior to specific others through direct and indirect software processes. This contribution is of particular value in a software...
development setting because dependencies among engineers are quite dynamic over the life cycle of a software development project. The paper further describes actual work practices used to manage task dependencies, which provides critical insight into our understanding of coordination in the context of large-scale geographically distributed projects.

Answering research questions in those two distinct but complementary lines of research has the potential to influence several subareas of software engineering, in particular, software architecting and design, software processes, and empirical software engineering. The papers included in this special section are excellent representatives of advances in the last two areas, software processes and empirical software engineering.

2 Paper Selection Procedure
An open call for papers was distributed in several venues, including the CHI and ICSE conferences and the well-established mailing lists for those communities during 2009. The call was also posted on the IEEE Transactions of Software Engineering’s website. A total of 12 submissions were received and a total of three rounds of full reviews resulted in the two papers presented in this special section.

Acknowledgments
This special section was made possible by a great deal of effort by many people, in particular, a group of very professional and committed reviewers with strong background in a diverse set of research areas such as software engineering, product development, information systems, CSCW/CHI, and organizational theory and behavior. The guest editors would like to thank those reviewers for their detailed and constructive reviews that led to significant improvements in the quality of the manuscripts. They would also like to thank the editorial board of the IEEE Transactions on Software Engineering and its editor-in-chief (Bashar Nuseibeh) for supporting this special section.

Marcelo Cataldo
Kate Ehrlich
Audris Mockus
Guest Editors

References