Chapter 54
Learning Software Engineering With Global Teams

Markus Ende
Nuremberg Institute of Technology, Germany

Ralf Lämmermann
Nuremberg Institute of Technology, Germany

Patricia Brockmann
Nuremberg Institute of Technology, Germany

Jesús-Manuel Olivares-Ceja
National Polytechnic Institute, Mexico

ABSTRACT

Global software engineering requires the coordination of team participants around the world, mainly in large software projects. How can computer science students learn the organizational and intercultural skills required to guide and participate in global distributed projects? To answer this question, this paper analyzes international virtual team teaching with the use of software engineering. Experiences and lessons learned are presented based on the results of a joint Mongolian-German team project. The obtained results with the Mongolian team encourage the project to include students and researchers from the National Polytechnic Institute in Mexico.

1. INTRODUCTION

Global software engineering is a discipline devoted to improving the systematic development of international software projects (Herbsleb & Moitra, 2001). Highly complex, large software projects are often broken down into a number of smaller components, which are then assigned to different teams, often located in different countries. Although personnel costs were the initial incentive for global outsourcing, today the scarcity of qualified IT professionals is one of the main reasons why software is developed globally (Braun, 2007).

DOI: 10.4018/978-1-5225-3923-0.ch054

Global software engineering presents a number of new challenges:

- **Geographic Distance:** Distributed teams often take twice as long to complete a task as teams working in the same location (Herbsleb & Mockus, 2003).
- **Different Time Zones:** Although globally distributed teams could theoretically work around the clock ("follow the sun" organization), real-time communication between groups is more difficult. Due to different time zones, there is often only a small period of time available when the different groups are all awake. When it's 8 a.m. in Mexico, in Germany it's 3 p.m. and in Mongolia it's 10 p.m. (Sosa, M., Eppinger, S., Pich, M., McKendrick, D., & Stout, 2002).
- **Language Differences:** Spoken and written communication between teams with different native languages can often prove difficult. When none of the team members share a common language, English is often used as a third language. Non-native speakers are often reluctant to communicate via telephone or video conference, because they don't have adequate time to translate their thoughts. In this case, asynchronous written communication, such as e-mail, is often preferred (Sangwan, R., Bass, M., Mullick, N., Paulish, D., & Kazmeier, J., 2007).
- **Trust:** Members of different teams who have never met personally often have difficulty establishing informal communication (Nguyen, P., Babar, M., & Verner, J., 2006). Due to economic constraints and visa requirements, it is often not feasible to fly team members to one location for a face-to-face meeting.
- **Cultural Differences:** The most difficult challenges faced by global software engineering result from misunderstandings due to cultural differences (Kruchten, 2004; Brockmann & Thaumueller, 2009; Paasivaara & Lassenius, 2006; Huang & Trauth, 2007).

In order to deal with these new challenges, computer programming skills alone are no longer sufficient. IT professionals today need to learn a new set of skills to cope with these new challenges. According to Romero (Romero, Vizcaíno, & Piattini, 2009), software professionals need the following skills to work effectively in global software engineering:

- English language skills,
- Virtual team skills,
- Appreciation of multicultural diversity,
- Understanding cultures and customs of other countries,
- Computer mediated communication protocols,
- Communication skills,
- Ability to resolve conflicts,
- Critical and self-critical abilities,
- Ability to deal with uncertainty and ambiguity,
- Teamwork skills.

The goal of this joint Mongolian-German team-teaching project is to investigate problem-based methods to effectively teach these skills which are necessary for global software engineering. The rest of the paper is organized as follows: Section 2 relates to the teaching global software engineering. Section 3 explains the Mongolian-German team teaching project. Section 4 describes the experiences obtained from the Mongolian-German project. Section 5 presents the lessons learned, which should be taken into