

Chapter 5.18

Globalising Software Development in the Local Classroom

Ita Richardson

University of Limerick, Ireland

Sarah Moore

University of Limerick, Ireland

Alan Malone

Siemens Corporate Research, USA

Valentine Casey

University of Limerick, Ireland

Dolores Zage

Ball State University, USA

ABSTRACT

In the dynamic global economy that exists today the operation and structure of organisations have had to adapt to the reality of the information revolution which has taken place. This has been the case within the software industry where global software development (GSD) has become a popular strategy and software development has become a globally sourced commodity. Given

the requirement for graduates to operate in this type of environment, we as educators considered how our teaching methods could be developed and enhanced to instil GSD competencies within our graduates. We provided students with the opportunity to take part in a learning experience that transcended geographical and institutional boundaries, giving them first-hand experience of working within globally distributed software teams. Two separate projects were undertaken.

One was with Siemens Corporate Research which was part of a larger project. The focus of this project was the shadowing of the development of an actual geographically distributed software product. The second project was carried out in collaboration with Ball State University, and the focus of this endeavour was virtual team software testing. Extensive qualitative research was undertaken on the data provided by the students. We identified three specific forms of learning which had taken place: (1) pedagogical, (2) pragmatic, and (3) the achievement of specific globally distributed competencies. Our findings would confirm that mimicking real-work settings creates the possibility of giving rise to the range of learning benefits that are associated with truly problem-based learning environments.

INTRODUCTION

This chapter explores the reality of the software industry today, which is becoming more virtual and globally distributed in its methods of operation. It discusses the educational implications of these strategies and how they impact graduates. It looks at what measures can be taken to prepare students to operate in this dynamic and virtual environment. It outlines two projects in which masters students participated in that transcended geographical and institutional boundaries. The projects and the students' experiences were researched and analysed. The results, which are presented here, demonstrate the benefits associated with utilising a hands-on, truly problem-based learning environment.

GLOBAL SOFTWARE DEVELOPMENT

GSD has given rise to the implementation of new types of development teams and project structures within organisations. In many software develop-

ment organisations, teams are no longer local, but operate within a virtual team environment. As a result they are fundamentally different in their structure and modus operandi to those of a single site team. For educationalists, the emergence of new team and organisational structures require that graduates from software engineering courses be made familiar with these new methods of operation.

The number of organisations employing virtual,¹ team-based globally distributed software development strategies continues to increase (Powell, Piccoli, & Ives, 2004). GSD in essence allows distributed teams to split up the tasks of a project and distribute them as separate jobs (Grinter, Herbsleb, & Perry, 1999). This allows development decisions about each project task to be made with a degree of independence (Herbsleb & Grinter, 1999). However, managing this type of team is not a straightforward endeavour. Some of the difficulties encountered include the problems of understanding requirements and the testing of systems (Toaff, 2002). These difficulties are compounded by cultural and language differences, lack of communication, distance from the customer, different process maturity levels, testing tools, standards, technical ability, and experience. These issues are further augmented by the lack of "trust-building" communication techniques. Trust is important for software development teams to work together successfully, and it is harder to establish trust within virtual teams than it is with local teams (Robey, Khoo, & Powers, 2000), rising from the fact that face-to-face communication methods that can build trust are generally not present in a distributed team (Pyysiainen, 2003). Equally, established trust gained from co-located experiences can deteriorate over time in a distributed setting (Casey, 2007). To address these substantial issues, project management must change from the traditional to the virtual for a GSD strategy to be successfully implemented.

21 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/globalising-software-development-local-classroom/29496

Related Content

Smart Device Authentication Based on Online Handwritten Script Identification and Word Recognition in Indic Scripts Using Zone-Wise Features

Rajib Ghosh, Partha Pratim Roy and Prabhat Kumar (2018). *International Journal of Information System Modeling and Design* (pp. 21-55).

www.irma-international.org/article/smart-device-authentication-based-on-online-handwritten-script-identification-and-word-recognition-in-indic-scripts-using-zone-wise-features/208638

Determinants of Intention to Use Ride-Hailing Services in Vietnam: An Integrated Model of Perceived Value and Trust Transfer Theory

Thao Thi Thanh Vo and Trong Hung Van (2022). *International Journal of Software Innovation* (pp. 1-10).

www.irma-international.org/article/determinants-of-intention-to-use-ride-hailing-services-in-vietnam/313382

A Model-Driven Approach to Service Composition with Security Properties

Stéphanie Chollet and Philippe Lalanda (2014). *Software Design and Development: Concepts, Methodologies, Tools, and Applications* (pp. 2051-2071).

www.irma-international.org/chapter/model-driven-approach-service-composition/77790

Lean, Light, Adaptive, Agile and Appropriate Software Development: The Case for a Less Methodical Methodology

John Mendonça and Jeff Brewer (2003). *Practicing Software Engineering in the 21st Century* (pp. 42-52).

www.irma-international.org/chapter/lean-light-adaptive-agile-appropriate/28109

Usability Engineering Methods and Tools

Amandeep Kaur (2013). *Designing, Engineering, and Analyzing Reliable and Efficient Software* (pp. 202-216).

www.irma-international.org/chapter/usability-engineering-methods-tools/74882