Supporting Collaboration between Customers and Developers: A Framework for Distributed, Agile Software Development

Francesco Bergadano, Dipartimento di Informatica, Università di Torino, Torino, Italy
Gianni Bosio, Dipartimento di Informatica, Università di Torino, Torino, Italy
Stefano Spagnolo, Dipartimento di Informatica, Università di Torino, Torino, Italy

ABSTRACT

The traditional, in-house software development process is progressively losing its appeal in favor of distributed, multi-site development: this is mainly due to the well-known advantages of the latter, such as higher productivity and lower costs. However, this practice has also some documented disadvantages that are inherent to distance: geographical, temporal and socio-cultural distances between stakeholders can affect communication, coordination and control activities, thus making collaboration very difficult. This would not immediately look as the ideal scenario for applying Agile methodologies, which definitely rely on continuous collaboration between all stakeholders, including (with a very important role) the customers. This paper analyzes issues related to collaboration between customers and developers in a distributed, Agile setting and proposes a framework that defines practices and tools for handling project information and communication activities.

Keywords: Collaborative working environments, Communications, Control Activities, Coordination, Customer-Developer Collaboration, Distributed Software Development

INTRODUCTION

A great part of the activities that happen in software development projects are usually enacted between people that are responsible of the development phase of the project itself: developers, analysts, testers, managers. However, a project features other stakeholders that can be involved in collaborative activities, covering one or more phases of the project, such as customers and final users.

Collaboration between the development team and customers can have several positive effects: in fact, correct requirement analysis can lead to a better comprehension of the customers’
real needs, in order to develop a final product which is better responding to such needs and, therefore, fully appreciated (Tseng & Du, 1998). Collaborating with customers can, especially during the early phases of a project, foster an effective exchange of information, which can lead to identifying and correcting errors and defects in the product (Saiedian & Dale, 2000).

Agile development methodologies propose a different paradigm that regulates interaction between customers and developers: specifically, they acknowledge the importance of a continuous flow of communication between them; such flow should comprehend the full duration of the project and should not be limited to the early phases. However, the maintenance of a communication flow can be difficult when, as it happens in Distributed Software Development, different people can be physically located very far from each other: while this is a condition that affects all kinds of distributed development projects, it is particularly severe in Agile ones, as they strongly rely on informal communication and continuous collaboration.

This paper presents the SCoAP (Support for Communication and Agile Project management) framework, a set of collaborative practices and tools that addresses communication, coordination and control issues between customers and developers in Distributed, Agile Software Development projects. Our target is to facilitate situated and contextual communication between developers and customers across remote sites.

The framework is composed of the following three layers:

- **User Definition Layer**: an informative space which collects and presents information about users, allowing to track their personal information, role and current activities.
- **Project layer**: practices and tools which facilitate awareness regarding project and tasks information and enables stakeholders to exploit it as a context for coordination and control strategies.
- **Communication layer**: practices and tools which leverage both formal and informal communication processes in order to promote the sense of proximity between stakeholders.

The article extends our previous work on the subject (Bergadano & Bosio, 2013) by describing in detail our prototype implementation of SCoAP.

**DISTRIBUTED SOFTWARE DEVELOPMENT**

In Distributed Software Development (DSD), also known as Global Software Development (GSD), the activity of developing the same software product is scattered between geographically distributed locations (Prikладnicki et al., 2003; Damian & Moitra, 2006; Carmel, 1999). The industry has been pushed to transform software development into a “multisite, multicultural, globally distributed undertaking” (Herbsleb, & Moitra, 2001) by several factors:

- Cost reduction is possible, if activities are outsourced to countries whose labor is cheaper.
- It can be possible to reach skilled workers and involve them in a project, no matter how distant they are.
- People with different backgrounds can share their ideas and solutions, thus promoting innovative problem solving.

Considering these factors, it is easy to understand why DSD has reached such a considerable level of diffusion. However, along with its advantages, DSD also poses several problems and constraints, that can hamper the final result in various forms (Ågerfalk, 2005; Conchuir, 2006):

- Long travels are needed in order to compensate for the obvious fewer opportunities for in person, face-to-face interaction, which is fundamental for building trust (Geographical distance).
Teaching Principles of Petri Nets in Hardware Courses and Students Projects
www.igi-global.com/chapter/teaching-principles-petri-nets-hardware/38261?camid=4v1a

Maintaining User Control While Storing and Processing Sensor Data in the Cloud
www.igi-global.com/article/maintaining-user-control-while-storing-and-processing-sensor-data-in-the-cloud/102759?camid=4v1a