

Collaborative Development Environments



Javier Soriano

Universidad Politécnica de Madrid (UPM), Spain

Genoveva López

Universidad Politécnica de Madrid (UPM), Spain

Rafael Fernández

Universidad Politécnica de Madrid (UPM), Spain

INTRODUCTION

More and more often organizations tend to behave like dynamically reconfigurable networked structures that carry out their tasks by means of collaboration and teamwork. Effective teamwork is an essential part of any non-trivial engineering process, and collaborative capabilities are an essential support for these teams. Software development is no exception; it is in itself a collaborative team effort, which has its own peculiarities. Both in the context of open source software development projects and in organizations that develop corporate products, more and more developers need to communicate and liaise with colleagues in geographically distant areas about the software product that they are conceiving, designing, building, testing, debugging, deploying, and maintaining. In their work, these development teams face significant collaborative challenges motivated by barriers erected by geographic distances, time factors, number of participants, business units or differences in organizational hierarchy or culture that inhibit and constrain the natural flow of communication and collaboration. To successfully overcome these barriers, these teams need tools by means of which to communicate with each other and coordinate their work. These tools should also take into account the functional, organizational, temporal and spatial characteristics of this collaboration. Software product users are now becoming increasingly involved in this process, for which reason they should also be considered.

In the context of the software development process, then, a collaborative development environment (CDE) can be defined as a safe and centralized solution conceived to optimize collaborative and distributed software development generally based on Internet standards.

This chapter introduces and defines the concept of CDE, while stressing the role these environments play in setting up a virtual space for negotiation, brainstorming, discussion, information and knowledge sharing, cooperation, coordination, development and management in engineering projects generally and especially software development projects. It then analyzes the collaboration-related points of conflict in the software development process. This conflict is motivated by issues, such as the space-time distribution of resources, which have a negative impact on both individual and team effectiveness and efficiency. On the basis of this analysis, we describe what essential purposes a CDE should serve, including: (a) the holistic integration of disparate collaborative processes and tools through a collaborative environment that represents a Web-accessible virtual project space, (b) the expansion of visibility and change control, (c) the centralization and administration of resources, and (d) the reinforcement of collaboration, creativity and innovation. We also examine what features and services a CDE should provide.

Then, we introduce the chief classification frameworks, according to which collaborative tools can be ranked by the needs that they satisfy, each one from a different viewpoint. Knowing and considering these frameworks, a team can contextualize the range of collaborative tools available, and compare them from different viewpoints and on the basis of assembled criteria sets to be able to make a grounded decision on what collaborative tools best meet its needs.

Finally, the chapter will refer to how CDEs are related within open source software communities. These communities have led to a change in how software development is viewed, and both communities and CDEs have been clearly influenced each other. A number of software and open source software develop-

ment support web sites that use CDEs to achieve their goals will be presented.

WHAT IS A CDE AND WHERE DO THEY COME FROM?

The issue of CDEs was perhaps taken up for the first time back in 1984, when Iren Greif and Paul Cashman organized a workshop that brought together an influential group of people to examine how to apply technology within a collaborative work environment. This meeting was the source of the “computer-supported cooperative work (CSCW)” concept (Grudin, 1994), which aimed to find an answer to how computer systems can support and coordinate collaborative activities.

A few years later, after further researching the concept of CSCW, Malone and Crowston (1994) introduced *coordination theory*, conceived on the basis of research in several different disciplines like computer science, organization theory, management science, economics, linguistics, and psychology, and according to which they defined coordination as a way of managing dependencies between activities. By characterizing the different types of possible dependencies between task activities, Malone and Crowston were able to identify and, consequently, manage the so-called coordination processes. This investigation identified some of the problems that future CDEs would have to deal with, such as, for example, resources allocation, as well as possible solutions.

Years later, when the technology was far enough evolved and after the Internet had materialized, these coordination processes and all the years of CSCW research led to collaborative tools capable of improving not only the development of software applications, but also the networked exchange of information and ideas from different branches of knowledge, with users who had possibly never worked together before and did not even know each other, based at geographically distant places, even overcoming time differences. This then led to the concept of *groupware* (Baecker, 1993), that is computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment, thanks to the enabling technologies of computer networking, software and services, materializing the ideas emerged from CSCW research (Engelbart, 1992).

Predictably, this activity yielded the first tangible definitions of CDEs. For example, “a CDE is a virtual space wherein all the stakeholders of a project, even if distributed by time or distance, may negotiate, brainstorm, discuss, share knowledge, and generally labor together to carry out some task, most often to create an executable deliverable and its supporting artifacts” (Booch & Brown, 2003). In this definition, the authors establish the key aspects to be taken into account in any CDE. In view of the importance that these environments have gained both in the open source context and the corporate environment with the upsurge of virtual and networked enterprises though, we believe that the definition falls short, as it only states what a CDE is and not how it works. It fails to come up with solutions for the challenges to be met by any CDE concerning the space-time distribution of resources. Therefore, we can add to the definition by saying that a CDE holistically integrates multiple collaborative tools and resources, thanks to which it offers a set of services to aid all the stakeholders in the software development area, including managers, developers, users, commercial software manufacturers and software product support enterprises, to communicate, cooperate and liaise. CDEs consider software development’s social nature and assure that the people who design, produce, maintain, commercialize and use software are aware of and communicate about the activities of the others simply, efficiently and effectively, also encouraging creativity and driving innovation.

CHARACTERIZATION OF A CDE

Grady Booch and Alan W. Brown (2003) state that the purpose of a CDE is to create a foundation that minimizes the frictions that have an impact on the routine work of software developers, reducing both individual and group efficiency. The key points of friction are:

- **The cost of working space start-up and on-going organization.** At the start of a project or when a new member joins, there will be a period of adaptation until the team finds the best tools to use, who to ask, the project status, and so forth.
- **Inefficient work product collaboration.** More than one person sometimes needs to work on the same document at the same time. When this is a critical document, a change control log needs to

5 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/collaborative-development-environments/17616

Related Content

An Exploratory Study Examining Group Dynamics in a Hackathon

Alana Pulayand Tataleni I. Asino (2019). *International Journal of Virtual and Augmented Reality* (pp. 1-10).

www.irma-international.org/article/an-exploratory-study-examining-group-dynamics-in-a-hackathon/239894

Tacit Knowledge in Communities of Practice

Michele Suzanne Zappavigna (2006). *Encyclopedia of Communities of Practice in Information and Knowledge Management* (pp. 508-513).

www.irma-international.org/chapter/tacit-knowledge-communities-practice/10539

Tacit-Explicit and Specific-General Knowledge Interactions in CoPs

Tunç Medeni (2006). *Encyclopedia of Communities of Practice in Information and Knowledge Management* (pp. 514-522).

www.irma-international.org/chapter/tacit-explicit-specific-general-knowledge/10540

Clustering Analysis of Networked Organizations

Hakki Eraslan, Melih Buluand Metin Turkey (2008). *Encyclopedia of Networked and Virtual Organizations* (pp. 199-208).

www.irma-international.org/chapter/clustering-analysis-networked-organizations/17613

Insight About the Use of AR and VR for Trainees: The Case of Bosch Company

José Rodrigues Oliveira (2020). *Managerial Challenges and Social Impacts of Virtual and Augmented Reality* (pp. 178-192).

www.irma-international.org/chapter/insight-about-the-use-of-ar-and-vr-for-trainees/248303