

## Chapter 5.2

# Characterization and Classification of Collaborative Tools

**Javier Soriano**

*Universidad Politécnica de Madrid (UPM), Spain*

**Rafael Fernández**

*Universidad Politécnica de Madrid (UPM), Spain*

**Miguel Jiménez**

*Universidad Politécnica de Madrid (UPM), Spain*

### INTRODUCTION

Traditionally, collaboration has been a means for organizations to do their work. However, the context in which they do this work is changing, especially in regards to where the work is done, how the work is organized, who does the work, and with this the characteristics of collaboration. Software development is no exception; it is itself a collaborative effort that is likewise affected by these changes. In the context of both open source software development projects and communities and organizations that develop corporate products, more and more developers need to communicate and liaise with colleagues in geographically distant places about the software product they are con-

ceiving, designing, building, testing, debugging, deploying and maintaining. Thus, work teams face sizeable collaborative challenges, for which they have need of tools that they can use to communicate and coordinate their work efficiently.

The response is the collaborative development environment (CDE), a virtual space where all the software project stakeholders, possibly distributed in time and space, can negotiate, brainstorm, discuss, share knowledge and resources and, generally, labor together to carry out some task in the context of a software development process (Booch & Brown, 2003).

The collaborative needs of a team depend largely on factors related to the environment, such as the team's organizational structure, its

geographical and temporal distribution, the target software domain, the software product structure and the actual team members. Each of these factors highlights a different aspect of collaboration. All existing collaborative development tools have been conceived considering different aspects of collaboration, each one meeting particular sets of needs and considering the particular functional, organizational, temporal and spatial characteristics of this collaboration.

As part of Morfeo (2005), we conducted a survey of existing software products and collaborative sites. This survey suggests that the likelihood of being able to develop a CDE meeting all the collaborative needs of the software process is remote, especially taking into account that these needs change depending on contextual factors such as the above. In this respect, the “contextual” approach to collaboration is gaining strength. This approach enables the holistic integration and deployment of collaborative components and services in a CDE suited for a particular context, as opposed to a monolithic conception of such an environment. Therefore it is worth examining existing classification frameworks. There are different classification frameworks that order collaborative tools by the needs they satisfy, each from a different viewpoint. A team that is acquainted with these frameworks can contextualize the range of available collaborative tools, and compare them from different viewpoints and on the basis of assembled criteria sets. This way it can make a grounded decision on what collaborative tools best meet its needs.

This chapter starts with a definition and preliminary characterization of collaborative tools and CDEs. Without claiming to be exhaustive, it goes on to describe some of the most representative frameworks that have been developed to date. The chapter then shows the resulting categorization for each approach and presents some representative tools for each particular category. Finally, it suggests ideas on how to use these frameworks to select the best collaborative tools for a particular work team and development project.

## COLLABORATIVE TOOLS AND DEVELOPMENT ENVIRONMENTS

The issue of CDEs was taken up perhaps 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 research into the concept of CSCW, Malone and Crowston (1994) introduced *coordination theory*. This theory was conceived on the basis of research in several different disciplines like computer science, organization theory, management science, economics, linguistics, and psychology. The theory defined coordination as the 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 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. Such an exchange often involved users who had possibly never worked together before and did not even know each other based at geographically distant places and even had to overcome time differences. This, in turn, led to the concept of *groupware* (Baecker, 1993), that is computer-based systems that support groups of people engaged in a common task

8 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/characterization-classification-collaborative-tools/36220](http://www.igi-global.com/chapter/characterization-classification-collaborative-tools/36220)

## Related Content

---

### Knowledge Management Success: Roles of Management and Leadership

Vittal Anantatmula (2010). *IT Outsourcing: Concepts, Methodologies, Tools, and Applications* (pp. 1954-1965).

[www.irma-international.org/chapter/knowledge-management-success/36257](http://www.irma-international.org/chapter/knowledge-management-success/36257)

### Outsourcing in the Healthcare Industry: Information Technology, Intellectual Property and Allied Aspects

Amar Gupta, Raj K. Goyal, Keith A. Joiner and Sanjay Saini (2008). *Outsourcing and Offshoring of Professional Services: Business Optimization in a Global Economy* (pp. 115-141).

[www.irma-international.org/chapter/outsourcing-healthcare-industry/27964](http://www.irma-international.org/chapter/outsourcing-healthcare-industry/27964)

### Lessons Learned from Successes and Failures in Information Systems Outsourcing

Kathryn M. Zuckweiler (2007). *Outsourcing Management Information Systems* (pp. 153-175).

[www.irma-international.org/chapter/lessons-learned-successes-failures-information/27985](http://www.irma-international.org/chapter/lessons-learned-successes-failures-information/27985)

### Flexible Global Software Development(GSD): Antecedents of Success in Requirements Analysis

Vanita Yadav, Monica Adya, Varadharajan Sridhar and Dhruv Nath (2010). *IT Outsourcing: Concepts, Methodologies, Tools, and Applications* (pp. 2404-2436).

[www.irma-international.org/chapter/flexible-global-software-developmentgsd/36286](http://www.irma-international.org/chapter/flexible-global-software-developmentgsd/36286)

### Information Technology Offshoring Outsourcing: A Perspective of Advanced Countries

Smita Gupta and Narendra S. Chaudhari (2006). *Outsourcing and Offshoring in the 21st Century: A Socio-Economic Perspective* (pp. 122-139).

[www.irma-international.org/chapter/information-technology-offshoring-outsourcing/27944](http://www.irma-international.org/chapter/information-technology-offshoring-outsourcing/27944)