

## Technologies and Tools for Distributed Teams

**Javier Portillo Rodríguez, Christof Ebert, and Aurora Vizcaino**

*Software development today is typically a team effort with team members in different geographical places. You might regret that teamwork isn't what it used to be, but you also might want to look toward technologies and tools that support distributed teamwork. Authors Javier Portillo Rodríguez, Aurora Vizcaino, and I provide an overview of such technologies with many tools examples, starting with technologies such as Jazz and then showing how users can orchestrate individual tools in a distributed context.*

*I look forward to hearing from both readers and prospective column authors about this column and the technologies you want to know more about. — Christof Ebert*

**S**everal technologies support communication, control, and coordination problems in a distributed development environment. We'll discuss such collaboration and integration technologies and take a close look at their tools for particular software processes, as the technologies often have a general purpose. To describe the processes, we'll follow the ISO/IEC 12207 standard. See our previous work for a more specific look at tools for distributed teams.<sup>1,2</sup>



### Technologies for Distributed Teams

Nowadays, there are many technologies for distributed teams. However, we selected four of the most widely used ones: IBM Jazz, Microsoft SharePoint, Google Apps, and Lotus Notes.

*IBM Jazz* ([www-01.ibm.com/software/rational/jazz](http://www-01.ibm.com/software/rational/jazz)) is a collaborative technology platform for software delivery. An extensible framework, IBM Jazz can integrate and synchronize people, processes, and assets related to software projects. The technology's main features include

- team awareness through automatic notifications (team members can see the work of the other members),
- collaboration features such as project-integrated presence and messaging,
- extensible infrastructure through open Web standards for including new plug-ins,
- process awareness guiding team workflow and automating process steps,
- flexible process definition by specifying and modifying process rules for different subjects or projects, and
- a connector framework for integrating other source control and change management systems, such as subversion.

*Microsoft SharePoint* is a business collaboration platform that enables collaboration on setting up websites, sharing information, managing documents, and publishing reports. SharePoint includes the following set of tools or capabilities to support collaboration:

- *Sites* provides a single infrastructure for all the business websites that allows sharing documents or managing projects.
- *Composites* offers tools and components for creating no-code business solutions.



- *Insights* helps team members access information in databases, reports, and business applications.
- *Communities* delivers collaboration tools to share ideas and work.
- *Content* represents SharePoint's content management system with features such as document types, retention policies, and automatic content sorting.
- *Search* helps people find information and contacts.

*Google Apps* offers Web-based office tools for any size business. This technology includes a set of tools to

- manage documents or multimedia information such as videos,
- collaboratively create and manage business sites,
- communicate through chat or email features, and
- manage work groups, tasks, activities, and so on, including email notifications and calendar management.

*IBM Lotus* software lets people connect, collaborate, and innovate. It offers Web-based tools for collaborating and connecting applications for creating, view-

ing, and exchanging information. The tools offer the following features:

- secure online spaces to share content and collaborate on team projects;
- presence awareness, instant messaging, and Web conferencing;
- Web-based email, calendar, and contact management;
- ability to manage, run, and analyze Web conferencing events;
- workflow and project management; and
- collaborative development and deployment of business applications.

Table 1 summarizes important aspects of each technology, such as the tools they offer, how they integrate external data, which tools are compatible or integrate with the technology, which Web 2.0 applications it includes, and which security controls or protocols the technology implements.

Companies developed most of these technologies to support communication, control, and coordination problems in distributed systems development; they didn't develop them to cover all software lifecycle processes (SLPs). Because of this, we want

to complement this description with tools that support some software processes.

### Distributed Development Tools

ISO/IEC 12207 defines taxonomy for SLPs. The aim of this standard is to be flexible, modulated, and adaptable, letting software constructors adapt the standard to their needs.

The SLPs defined by the standard are divided into tasks and activities and fall into primary, supporting, and organizational processes. Another way to group them is by system context processes and software-specific processes (see Figure 1). We use this last group classification for this work. In Figure 1, we show all the processes the ISO/IEC 12207 defines, and we highlight those processes we're focused on: project, implementation, and support processes. Because of space limitations, we deal with some of those most relevant to distributed teams. For more details on how tools support collaboration processes, we suggest reading our previous work.<sup>1,2</sup>

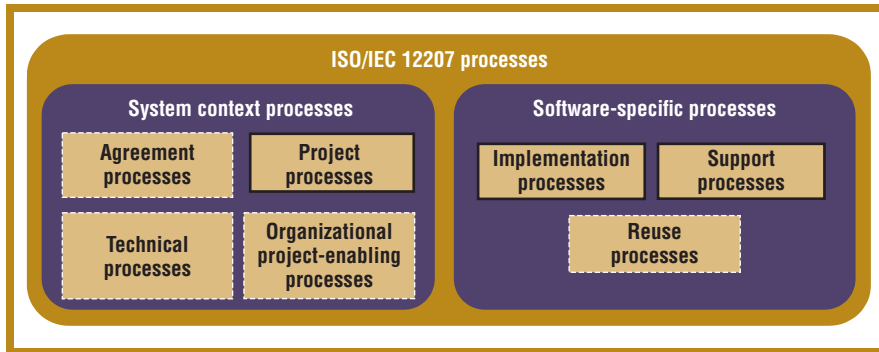
### Project Processes

The following software supports project processes (see Figure 2):

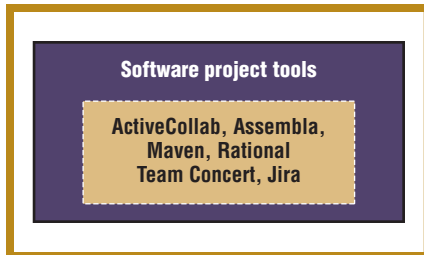
**Table 1**

**Collaborative technologies**

Collaborative technology	Technology tools	Data backbone	Compatible tools	Web 2.0 apps	Security
IBM Jazz ( <a href="http://www.01.ibm.com/software/rational/jazz">www.01.ibm.com/software/rational/jazz</a> )	Rational Team Concert, Rational Quality Manager, Rational Requirements Composer, Rational Asset Manager, Rational Insight	Integration based on Open Services for Lifecycle Collaboration using Jazz Foundation Services across independent databases that use Internet protocols	Rational ClearQuest, Rational ClearCase and Subversion, Lotus Connections, Microsoft SharePoint	blog	Lightweight Directory Access Protocol (LDAP)
Microsoft SharePoint ( <a href="http://SharePoint.microsoft.com">http://SharePoint.microsoft.com</a> )	SharePoint Sites, SharePoint Communities, SharePoint Content, SharePoint Search, SharePoint Insights, SharePoint Composites, SharePoint Designer, SharePoint Foundation, SharePoint Workspace	Windows Share-point Services Data Integration through platforms such as Mamboo MashPoint or Mainsoft Share-Point Integrator	Microsoft Office, Lotus Notes, Domino	wiki, blog	Secure Sockets Layer (SSL), IP Security, Windows Security Account
Google Apps ( <a href="http://www.google.com/apps">www.google.com/apps</a> )	Google Calendar, Google Docs, Google Groups, Gmail, Google Sites, Google Videos	Data Integration through Google Secure Data Connector	Microsoft Outlook, Lotus Notes	wiki, blog, video	HTTPS, SSL, Message Security through Postini technology
IBM Lotus ( <a href="http://www-01.ibm.com/software/lotus">www-01.ibm.com/software/lotus</a> )	Lotus Notes, Lotus Domino, Lotus SameTime, Lotus Live, Lotus Quickr, Lotus Connections	Data Integration through Lotus Enterprise Integrator	Rational Team Concert	blog, wiki, microblog	SSL, Access Control List (ACL)



**Figure 1. ISO/IEC 12207 processes. ISO/IEC 12207 defines a taxonomy for software life-cycle processes grouped into system-context and software-specific processes. Solid lines represent those processes we highlight in this article.**



**Figure 2. Project process tools. These tools focus on areas such as project planning, management, and tracking.**

- *ActiveCollab* ([www.activecollab.com](http://www.activecollab.com)) is a commercial software tool that supports activities related to project management, such as planning or progress tracking from a Web-based platform. It offers events email notifications and allows email integration into a project's workflow.
- *Assembla* ([www.assembla.com](http://www.assembla.com)) is a Web-based tool that lets the developer control and coordinate a project's activity stream. It enables collaboration among team members through Web-based features, such as wiki pages, and the ability to view each team member's work. Assembla enables integrating collaboration tools depending on the project team's need to perform project activities. It also lets the team share select project information with clients.
- *Maven* (<http://maven.apache.org>) provides a unified build system and quality project information. Maven is a free tool that developers can integrate into IDEs such as Eclipse or Netbeans and use to create and integrate plug-ins for adding functionalities.
- *Rational Team Concert* ([www-01.ibm.com/software/awdtools/rtc](http://www-01.ibm.com/software/awdtools/rtc)) is an IBM Jazz tool that provides planning, source code, work item, building, and project health management. To carry out these tasks, this tool integrates modules to support process and team awareness, work item tracking, and

agile planning, providing tools to create plans for teams and continuous build integration.

- *Jira* ([www.atlassian.com/software/jira](http://www.atlassian.com/software/jira)) is a tool with a high level of integration through plug-ins, offering issue tracking and agile project management. Jira offers easy bug tracking and full integration with popular IDEs such as Eclipse.

### Implementation Processes

As Figure 3 shows, the processes that support implementation include requirements analysis, detailed design, and integration.

**Requirements Analysis Tools.** This group includes tools that let distributed team members manage software requirements:

- *IBM Rational Doors* ([www-01.ibm.com/software/awdtools/doors](http://www-01.ibm.com/software/awdtools/doors)) permits collaboration among team members, letting them manage requirements, and can integrate with other tools to provide a comprehensive traceability solution. It supports the Requirements Interchange Format, which lets suppliers and development partners get directly involved in development. IBM Rational Doors provides full traceability through the entire life cycle by integrating other tools such as IBM Rational Tau.
- *eRequirements* ([www.erequirements.com/app](http://www.erequirements.com/app)) is a noncommercial, Web-based tool that allows teams to manage functional and nonfunctional requirements. It includes a PDF document generator and lets teams capture

a project's high-level details or collaboratively explore and build the use cases that will define the system. eRequirements also includes an automatic requirement specification document generator.

- *Rational Requirements Composer* (<https://jazz.net/projects/rational-requirements-composer>) is another IBM Jazz tool for collaboratively creating requirements through a Web-based interface. It lets users communicate through inserting comments as well as link requirements with work items in Rational Team Concert and tests in Rational Quality Manager. The tool supports collections, shared filters, tags, attributes, hyperlinks, and advanced searches to help find and organize requirements and related information.

**Detailed Design Tools.** This group includes tools that let teams design activities to perform collaboratively in a distributed team:

- *Gliffy* ([www.gliffy.com](http://www.gliffy.com)) permits collaboration among distributed team members. Users can invite team members to a session to view or edit diagrams. It includes email notifications, a revision control system, and an access control system. The tool also includes a document manager to organize diagrams, allowing tagging.
- *Camel* is a research tool in which users can sketch and discuss multiple diagrams (UML and freestyle).<sup>3</sup> It also supports mechanisms for managing the dialog's focus during meetings and provides mechanisms for capturing all the relevant information from design meetings.
- *Rational Tau* ([www.ibm.com/software/awdtools/tau](http://www.ibm.com/software/awdtools/tau)) is a UML 2.1-based environment that enables collaboration among teams and includes automatic document genera-

tion and test case management. It integrates requirements modeling and embedded application development.

**Integration Tools.** The standard defines the integration process's main activity as being the combination of the software units and software components, thus producing integrated software items. We have therefore included the following tools:

- **Hudson** (<http://hudson.dev.java.net>) is a Web-based, continuous integration system mainly focused on building and continuously testing software projects and monitoring executions of externally run jobs.
- **Merlin ToolChain** (<http://merlintoolchain.sourceforge.net>) offers a free solution for integrating project management, requirements management, configuration management, and test management tools.

This group includes tools that facilitate the integration of project changes.

### Software Support Processes

As Figure 4 shows, the third group comprises software tools for support processes. In this group, we specifically included tools to support documentation management and configuration management.

**Documentation Management Tools.** This group includes tools that permit documentation or document management for distributed teams:

- **TWiki** ([www.twiki.org](http://www.twiki.org)) is a Web application platform that uses wiki technology and is configurable as a document management system.
- **Google Docs** (<http://docs.google.com>) is a Web-based platform that lets teams create, edit, and share documents. They can also edit documents simultaneously.
- **Lotus Notes** ([www.ibm.com/software/lotus](http://www.ibm.com/software/lotus)) is a client-server collaboration system that lets users share, for instance, documents that different people must manage. Moreover, it enables the sharing of databases, calendars or agendas, and email management.

**Review or Quality Assurance Tools.** This

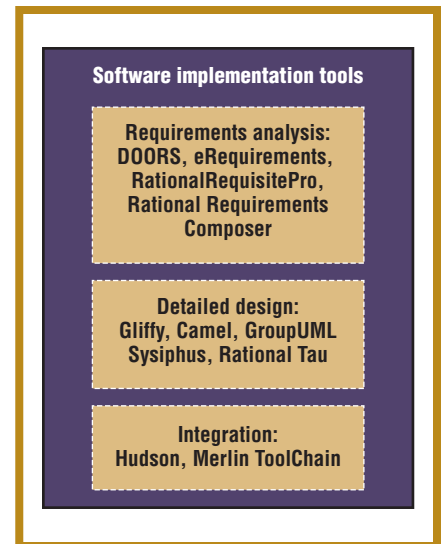
group includes tools related to the review and quality assurance processes, also known as software inspection tools (SITs). Related to the review process because software inspections are a type of peer review for detecting and removing defects in earlier development stages. They are also related to the quality assurance process because software inspections are the best practice for improving software quality.<sup>4</sup> The SITs selected are the following:

- **Internet-Based Inspection System (IBIS)** is a Web-based tool for supporting geographically dispersed inspection teams. IBIS minimizes synchronous activities and coordination problems. The tool includes automatic email notifications and forum discussion threads.
- **Rational Quality Manager** ([www.ibm.com/software/awdtools/rqm](http://www.ibm.com/software/awdtools/rqm)) is an IBM Jazz tool that offers Web-based reporting capable of quantifying project decisions' impact on business objectives for decision makers and quality professionals. It supports life-cycle quality management from requirements to build to test cases and defects.

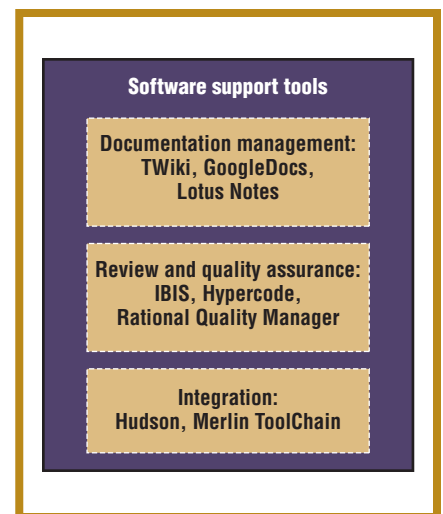
**A**fter studying the technologies and tools, we can extract some conclusions. To reduce the loss of communication's effect on distributed teams, technologies and tools should include communication features such as chats or forums for simultaneous synchronous and asynchronous communication. The majority of the tools we studied use asynchronous communication features such as threaded forums because distributed team members often can't meet at the same time. Only those tools that support synchronous collaborative activities, such as Camel, include communication features such as chats to support synchronous communication.

The technologies provide solid support for communication, coordination, and control aspects by including project management features such as tracking systems, automatic notifications, calendar management, and so on, but these technologies don't cover all the phases of the software life cycle.

Another problem is the lack of con-



**Figure 3. Software implementation tools.** These tools focus on areas such as traceability, specification, and organization.



**Figure 4. Software support tools.** These tools focus on areas such as document management, editing, and sharing, as well as inspections.

nection among tools. Selecting a set of tools to support the complete software life cycle is difficult because each tool has been developed without taking the next phase into account, and users must adapt information from one tool for use in another tool. Only in the case of Jazz tools, such as Rational Requirements Composer, does this connection exist with other IBM tools. Be aware, however, that close integration often builds on proprietary data models that lock in users with one supplier. We therefore recommend using Eclipse-based plug-ins

or open data backbones such as eASEE and not to embark on a single vendor-driven tool integration. ☞

### Acknowledgments

The Pegaso/MAGO project (Ministerio de Ciencia e Innovación MICINN and Fondo Europeo de Desarrollo Regional Feder, TIN2009-13718-C02-01) funded this work. Also, the Englobas (PII2I09-0147-8235) and Melisa (PAC08-0142-3315) projects, Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia and the Fabrum project (PPT-430000-2008) supported this work.

### References

1. F. Lanubile et al., "Collaboration Tools for Global Software Engineering," *IEEE Software*, vol. 27, no. 2, 2010, pp. 52–55.
2. C. Ebert, *Global Software Engineering: Distributed Development, Outsourcing, and Supplier Management*, Wiley, IEEE CS Press, 2010.
3. M. Cataldo et al., "Camel: A Tool for Collaborative Distributed Software Design," *Proc. IEEE Int'l Conf. Global Software Eng. (ICGSE 09)*, IEEE Press, 2009, pp. 83–92.
4. F. Lanubile, T. Mallardo, and F. Calefato, "Tool Support for Geographically Dispersed Inspection Teams," *Software Process: Improvement and Practice*, vol. 8, no. 4, 2003, pp. 217–231.

**Javier Portillo Rodriguez** is a PhD student and researcher at Alarcos Research Group, Institute of Information Technologies & Systems, Escuela Superior de Informática, University of Castilla-La Mancha, Spain. Contact him at javier.portillo@uclm.es.

**Christof Ebert** is a managing director and partner at Vector Consulting Services. Contact him at christof.ebert@vector.com.

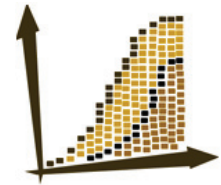
**Aurora Vizcaino** is an associate professor at Alarcos Research Group, Institute of Information Technologies & Systems, Escuela Superior de Informática, University of Castilla-La Mancha, Spain. Contact her at aurora.vizcaino@uclm.es.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.



## International Conference on Lean Enterprise Software and Systems



Lean Software and  
Systems Consortium

October 17-20, 2010  
Helsinki, Finland

## Call for Participation

<http://less2010.leanssc.org/>

### Tracks:

- Lean Product Development and Innovation
- Scaling Agile to Lean
- Beyond Budgeting
- Executive Track

**Registration:** open now

### Speakers include:

- Deborah Nightingale
- Martin Curley
- Duc Truong Pham
- Bjarte Bogsnes
- David Joyce
- Alan Shalloway
- David Anderson
- Ken Power
- Kati Vilkki
- Alisson Vale
- Marko Taipale  
*and many more*