

How to get mature global virtual teams: a framework to improve team process management in distributed software teams

Javier García Guzmán · Javier Saldaña Ramos · Antonio Amescua Seco · Ana Sanz Esteban

Published online: 3 April 2010
© Springer Science+Business Media, LLC 2010

Abstract Managing global software development teams is not an easy task because of the additional problems and complexities that have to be taken into account. This paper defines VTManager, a methodology that provides a set of efficient practices for global virtual team management in software development projects. These practices integrate software development techniques in global environments with others such as explicit practices for global virtual team management, definition of skills and abilities needed to work in these teams, availability of collaborative work environments and shared knowledge management practices. The results obtained and the lessons learned from implementing VTManager in a pilot project to develop software tools for collaborative work in rural environments are also presented. This project was carried out by geographically distributed teams involving people from seven countries with a high level of virtualness.

Keywords Geographically distributed software development · Management of IT function · Global distributed teams · Virtual team efficacy · Team processes maturity · Software team management

1 Introduction

Industrial globalization and outsourcing activities are on the increase because current markets are highly globalized (Braun 2007). In this environment, outsourcing software

J. García Guzmán · J. Saldaña Ramos (✉) · A. Amescua Seco · A. Sanz Esteban
Computer Science Department, University Carlos III of Madrid, Avda. Universidad, 30, 28911
Leganés, Madrid, Spain
e-mail: jsaldana@inf.uc3m.es

J. García Guzmán
e-mail: jgarcia@inf.uc3m.es

A. Amescua Seco
e-mail: amescua@inf.uc3m.es

A. Sanz Esteban
e-mail: asanz@inf.uc3m.es

activities (development, testing, maintenance, programming and incidence management) are indispensable if organizations want to maintain the required levels of competitiveness and productivity (Herbsleb and Moitra 2001; Sengupta et al. 2006). To create outsourcing environments, people in different locations have to work in a single software development team.

There are several strategies to configure this type of team depending on the existing interdependence among team members. In some projects, management, requirements specification and architectural design are done in one location and the development or modification of software components by software factories set up in separate geographical locations (Edwards and Sridhar 2003; Smite 2004).

However, other projects require the creation of several coordinated multidisciplinary teams bringing together highly skilled individuals working in dispersed geographical locations. This type of team is called a *virtual team* (Cohen and Gibson 2003). The work presented in this paper is especially centered on the second type of team.

The specific characteristics that differentiate virtual and traditional teams are:

- Each team member can work in a different location; members carry out independent tasks across locations (Hyrkkänen et al. 2007).
- Team members come from several cultures and have different teamwork practices (Powell et al. 2004). There is a great potential for conflict in global virtual teams as members work across cultural, geographical and time boundaries (Kankanhalli et al. 2007).
- Team members speak several languages. Although English is commonly used, there are different levels and flavors (Durnell and Webber 2005; Gaudes et al. 2007).
- Knowledge management is different as knowledge is distributed among the different sites (Ramasubbu and Krishnan 2005; Striukova and Rayna 2008).
- Communication activities and meetings rely strongly on technologies. So, more time is needed to obtain a common view of the goal and how to achieve it (Petkovic et al. 2006).
- The time difference makes agenda management more difficult; team work dynamics and the different location work dynamics need to be combined (Lee-Kelley and Sankey 2007).

Many organizations that have implemented Global Software Development have found that the geographically distributed environments are highly complex (Herbsleb and Moitra 2001). This complexity comes from the additional challenges they face:

- (a) Lack of common understanding of goals and requirements assigned to the team. This makes team members feel isolated and they are reluctant to collaborate, share and work together (Lee-Kelley and Sankey 2007; Striukova and Rayna 2008);
- (b) Difficulties in communication among different team members who are geographically distributed (Fuller et al. 2007; Rosen et al. 2007);
- (c) Differences between processes, management mechanisms and the associated skills and abilities produce problems and bottlenecks in the project execution. (Lee-Kelley and Sankey 2007; IPRC 2007);
- (d) Ineffective management of shared knowledge among different team members causes duplication, inconsistency and lack of knowledge of project assets (IPRC 2007; Rosen et al. 2007).

In order to deal with the previously mentioned problems, it is necessary to facilitate the effective integration of software development techniques in global environments with

other practices addressed to manage the high complexity of this type of environment, for example explicit practices for global virtual team management (Powell et al. 2004), definition of skills and abilities needed to work in these teams (Sengupta et al. 2006; Petkovic et al. 2006), availability of collaborative work environments (Gaudes et al. 2007; Herbsleb 2007) and shared knowledge management practices (Thomas et al. 2007).

This research defines and validates a set of practices to facilitate the effective integration of software development techniques with others to solve the specific problems of globally distributed teams.

2 Literature review

This section presents the relevant practices used in the software industry for global software team management. Hertel’s Five Phases Model (Hertel et al. 2005) was selected to structure the information already available in the literature because it provides a more structured view of the lifecycle of a global software team. This model distinguishes five main phases with specific management tasks that have to be addressed in the course of virtual teamwork.

Figure 1 summarizes and organizes the proposals considered for this literature review using Hertel’s Five Phases Model.

a) Preparation: The objectives of this phase are to define the team structure and goals, recruit team members based on the skills identified and the technological support needed to work in a geographically distributed environment.

When organizing teams, their organizational structure, the type of tasks to be assigned to each and the performance management rules should be contemplated (Ortiz de Guinea et al. 2005). In structuring global software teams, other papers mention the need to have effective infrastructure and processes in these environments and to define the requirements to establish effective teams (Carmel 1999; Karolak 1999).

The goal associated with every subteam in Global Software Development (GSD) is established in this phase. Requirements lists are used to facilitate better control of the goal assignments for each sub-team. In general, these are not as formal as the lists proposed by the IEEE 830 standard (IEEE 1993), but they are simpler. Other approaches highlight the importance of establishing the shared business goals and how to adapt them to each environment in this phase (Ramasubbu and Krishnan 2005). Individual goals, the reward systems to be implemented when the goals are achieved (Hertel et al. 2005; Matloff 2005), the technological infrastructure required for effective management of a global software development team should also be established. Depending on the purpose, different types of synchronous and asynchronous tools can be used. Nevertheless, it is considered more

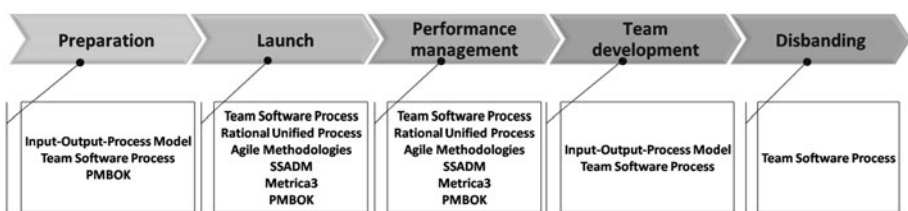


Fig. 1 Relationship between the current proposals

effective in this task to identify the collaboration patterns (Schümmer 2003) that make communication and collaboration easier among different people at different levels.

With regard to the recruitment of team members, it is necessary to consider the Team Software Process (Humphrey 2000) that establishes a set of activities for team formation covering several topics such as selection process, inherit formed teams, selection criteria, training, team players and potential leaders.

b) Launch: The objectives of this phase are to develop general rules for teamwork, facilitate communication among team members, define tasks based on a high-level definition of software architecture, assign roles and responsibilities to team members and define the strategy for monitoring and tracking (Hertel et al. 2005; Karolak 1999; Carmel 1999).

The methods used in global software development projects: the Rational Unified Process (RUP) (Kruchten 2004), Métrica V3 (MAP 2009), SSADM (Eva 1994), SCRUM (Schwaber and Beedle 2001) or PMBOK (PMI 2004), also contain activities to achieve the objectives stated for this phase.

There are cases of successful application of these methodologies when work is distributed among different teams and locations but the members of each team work in the same place (Battin et al. 2001; Ebert and De Neve 2001). However, several authors (Kruchten 2004; Larman 2004; Schwaber and Beedle 2001) indicate that these methodologies are not suitable when each team member is located in different geographical places. This is due in part to the fact that these approaches do not identify the specific rules that must be followed to ensure trust and effective communication, nor do they enhance the sense of teamwork, including aspects such as motivation and fear. (Casey and Richardson 2008; Hertel et al. 2005; Powell et al. 2004).

c) Performance management: This phase comprises the period from start to finish of the technical work to be developed. The main goal in this phase is to work efficiently while maintaining the atmosphere of a constructive team. In global virtual teams, leadership issues (Lee-Kelley and Sankey 2007), regulation of communication (Hertel et al. 2005), shared knowledge management (Prikladnicki et al. 2003) and tracking progress against the established plan (Ramasubbu and Krishnan 2005) need to be considered.

Leadership is particularly difficult in geographically displaced environments. Leadership is based on management by objectives, feedback to help every team member manage his work (Hertel et al. 2005) and management through delegation to get team members to do their work with little support from the leader (Jarvenpaa et al. 1998).

Communication within this type of team is based on electronic media and software collaborative tools. So, communication must focus on selecting the right technology for the most effective communication among team members (Powell et al. 2004; Prikladnicki et al. 2003).

Because of the special features of geographically distributed teams, shared knowledge management is an important issue (Prikladnicki et al. 2003). Without effective information and knowledge-sharing mechanisms, team performance is reduced. To prevent assumptions and ambiguity and to support maintainability, documentation and knowledge must reflect what every team member is using and working on (Herbsleb and Moitra 2001).

Tracking progress against a plan allows team leaders to control the project status as well as team performance. When team members are located in different places, this task is more difficult because collaborative tools are used for tracking (Ramasubbu and Krishnan 2005) and face-to-face meetings are not frequent.

d) Team development: The objectives of this phase are to execute team assessment and training activities. Team assessment activities determine whether the team has a common

sense of membership, its own processes and plans, the appropriate skills and is dedicated to doing an excellent job (Humphrey 2000). There are several approaches that provide specific indicators for virtual team management grouped into individual, team, leader, organization, project and technology (Gaudes et al. 2007).

In addition to carrying out methodological or technical training activities, team members have to be provided with periodic back-up activities on the methods and tools they need to use throughout the project in order to work in a distributed environment. After the training program is executed, it should be assessed to determine how it could be improved in future (Herbsleb and Moitra 2001; Hertel et al. 2005).

e) Disbanding: The goals of this phase are to disband the global virtual team and decide how to re-integrate the members in other teams. Team disbanding has to be done quickly, carefully and constructively to maintain a high level of motivation and satisfaction among employees and facilitate their rapid integration in other teams (Hertel et al. 2005). Moreover, in this phase, team achievements are recognized collectively and individually and team members are rewarded according to the system initially established.

Normally, software development methods only include recommendations to document experiences and efficient practices extracted from the project. TSP also contemplates these practices and includes others specifically addressed to improving the team management process in the team-member postmortem (Humphrey 2006). In TSP, the team manager gathers and analyzes data on different issues, for instance assess the adequacy of personal data-gathering practices or size and time estimating performance (Humphrey 2006).

To conclude, it can be stated that global virtual team management and traditional team management are different because of the nature of the teams. Consequently, traditional methods and techniques alone cannot be used to manage global virtual teams since they do not support issues such as collaboration or communication in geographically dispersed environments.

To manage this type of team efficiently, additional elements that influence their success also need to be considered, for example:

- Team management practices which are implicit in traditional teams. In this type of team with so many problems and different cultures, these practices should be defined and published for team members (Powell et al. 2004).
- An explicit approach to facilitating process integration during the project is needed because of the different processes each participant uses and the skills and abilities related to these processes (Sengupta et al. 2006; Petkovic et al. 2006).
- Collaborative work environments have to be available in geographically distributed teams since communication and collaboration rely heavily on technology in these environments (Gaudes et al. 2007; Herbsleb 2007).
- Efficient practices for shared knowledge management should be incorporated in distributed environments (Prikladnicki et al. 2003; Thomas et al. 2007).

Integration between the previous elements and technical software development activities is necessary to manage the global software team properly (Prikladnicki et al. 2003; Sengupta et al. 2006).

3 Research objectives

The aim of this research work is to define and validate a set of effective practices to facilitate the effective integration of software development techniques in global

environments with others addressed at managing the high complexity of global software development teams.

The specific subgoals associated with this aim are:

- (a) define a new methodology (hereafter, VTManager) to manage global software teams in multi-place and multi-project environments;
- (b) assess VTManager in a real project to determine whether this method contributes to increasing the global software team's productivity. More specifically, the following validation goals were put forward:
 - (1) To check the effectiveness and efficiency of the global software team management, provided by VTManager in each phase of global software team management.
 - (2) To evaluate the difficulties of applying VTManager to manage different types of global software teams.
 - (3) To evaluate VTManager's adaptability to new tools and paradigms of collaborative work.

4 Research method

4.1 Procedure

This research work was structured into two phases: definition of a methodology to manage global software teams and validation of VTManager contribution to increasing the productivity of a global software development team.

First, in order to define VTManager, several activities were carried out: analysis of the current problems of global software development teams, identification of the key needs for VTManager, identification of current best practices to consider and VTManager formalization. This formalization is based on process orientation, including the definition of processes and their activities. For each activity, we define the effective practices (called *specific recommendations*); the collaboration patterns that can be implemented for the specific recommendations; and the skills and abilities that global team members should have to implement effectively the specific recommendations.

Second, the authors decided on a field experiment as the best way (Juristo and Moreno 2001) to evaluate VTManager's contribution to increasing the productivity of a global software development team. The main reason for this choice was that a field experiment applies the scientific method to examine an intervention (coaching in VTManager activities) in the real world (the four teams selected).

The activities carried out to apply the research method, based on field experiments in this project, are described below.

1. Selection of the geographically distributed teams in the C@R project for this validation.
2. Coaching and training in VTManager activities.
3. Periodic reviews of team performance through meetings and interviews.
4. Observation of the results each team obtained. As the authors of this paper were able to observe teams directly, team evolution and information on validation goals and evidences found are discussed. Evidence is analyzed for each validation goal.

- Finally, the definition of VTManager was refined and enriched with specific recommendations obtained from the lessons learned.

4.2 Participants and setting

The pilot implementation of VTManager activities involved coaching different software teams created in the C@R project (C@R 2005). C@R was a huge project, with a budget of €12 million, which lasted 36 months and involved people from 30 different countries. The objective of C@R was to develop software tools (SCTs) to solve problems in rural environments. These problems were related to collaborative logistics, procurement, selling, geographic information systems applied to business incubation, process improvement, rural tourism and territorial planning. Figure 2 shows the structure of C@R project teams. From an organizational point of view, this work team structure is representative of software development projects (Wallace 2007), but in the C@R project, it was conditioned by the geographically dispersed team members.

Each team was composed of groups of people from different countries and the members could participate in different teams. The SCTs implementation team was a different case. This team was made up of other six sub-teams working independently to develop different SCTs, but under the same architecture. For this validation, only the Spanish Users and Developers Teams were selected because the authors of this paper could not participate actively in the others. Despite the fact that the architecture validation and the Software Quality Assurance (SQA) teams were different, they were considered a single team for validation purposes because the SQA team was made up of members of the architecture validation team. Several characteristics of the different teams that affected the validation were: types of goal(s) (technical, scientific or administrative), size of team, time needed to perform the tasks, experience and previous training in this type of distributed environment, how critical the success of the work performed was and cultural and linguistic diversity of team members.

In the C@R project, four global software teams, having different characteristics, implemented VTManager: the fifth team, Project Coordination Board, did not perform

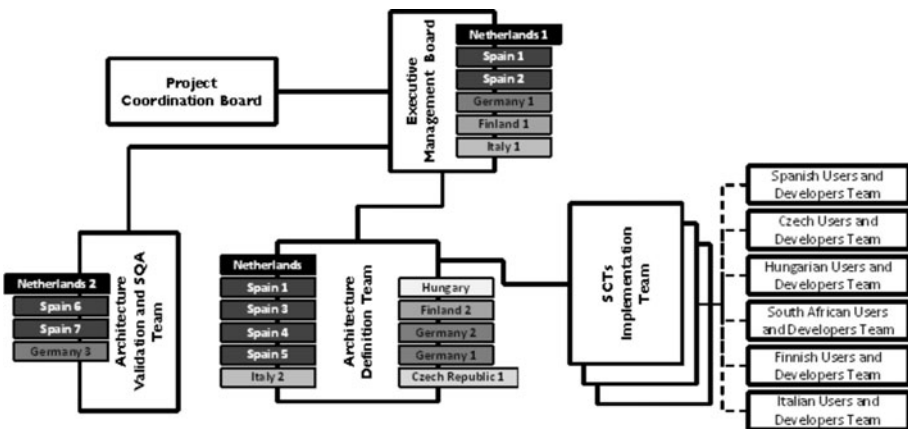


Fig. 2 Team organization in the pilot project

most of their tasks in a dispersed way. Table 1 shows the characteristics of each team considered for this pilot application and the type of work each did.

4.3 Research instruments

This section presents the information that was considered to evaluate VTManager contribution to increasing the productivity of a global software development team and the information gathering and analysis methods and techniques used. This information is discussed independently for each validation goal.

4.3.1 Validation goal 1: effectiveness and efficiency of global software team management

To achieve this validation goal, different efficiency and effectiveness indicators were analyzed. The indicators selected by the authors were: time needed to prepare and launch the teams, delays in submission of deliverables, and collaborative tools to support the team. The indicator called ‘time for preparation and launch’ is the time needed to complete phases 1 and 2 of VTManager, whereas ‘delays in submission of deliverables’ represents the longest delay.

The authors of this paper were able to register information on these indicators in the following ways:

- Time to prepare and launch teams: The authors could register this value as a result of their participation in training and coaching the different teams.
- Delays in submission of deliverables: The authors could register this value because one of the authors was in charge of the SQA group, so he had to check the delays in deliverables submission.

The authors also observed the application of VTManager and final interviews were conducted to determine lessons learned.

4.3.2 Validation goal 2: evaluation of the difficulty of implementing VTManager in different types of teams

To achieve this validation goal, it was necessary to assess how VTManager activities were implemented in the pilot project. This assessment was carried out in a planned way and included the following phases: data collection, data validation and results identification. VTManager implementation in the pilot project was assessed on the basis of objective evidence that demonstrated the execution of each VTManager activity—identification of the results obtained as a consequence of each activity, e-mails, meeting minutes and logs of the supporting tools used to implement VTManager. After gathering the data, annual interviews were carried out to confirm that the evidence collected was objective and to ensure that the evidence was representative of the degree of execution of an activity and that the complete information was consistent.

As a result of this assessment, the degree of execution of each VTManager activity was determined and classified using the following categories: (a) the activity was not fully performed; (b) the activity was performed without using collaborative patterns and/or technology support; and (c) the activity was performed using collaborative patterns and/or technology support.

Table 1 Characteristics of C@R teams

Team	Purpose	Staff involved	Number of members	Duration of the work (months)	Experience	Degree of criticality	Cultural diversity
1. Project Executive Management Board	Carry out the administrative and technical management of the project	Project Managers, Block Leaders, Deputies and Project Quality Engineers	12	36	Project Management: very experienced Collaboration Tools: High level	High	People from 5 countries
2. Interdisciplinary Group for CWE R&D in Spanish settings	Design collaborative work tools integrated with the project general architecture	Project Managers, Software Developers, CWE Researchers, End-users and Methodological Facilitators	22	24	Technical: High level Collaboration Tools: High level	High	People from 1 country
3. Definition of architectures for heterogeneous SCTs interoperability	Design the architecture that facilitates the integration, reuse and interoperability of CSS made by different manufacturers	CWE and SQA Researchers and several Developers	6	12	Technical: High level Collaboration Tools: Medium	High	People from 7 countries
4. Validation of the architecture for heterogeneous SCTs interoperability and assessment and evaluation of project quality (SQA)	Validate the architecture of interoperability among the different individual scenarios in the project Assess the quality of the project results	Architecture Team Representatives, Project Managers, Block Leaders and Project Quality Engineers	14	12	Technical: High level Collaboration Tools: Medium	Medium	People from 3 countries
5. Project Coordination Board	Take decisions about the project	One representative of each partner	30	36	Collaboration Tools: Diverse	Medium	People from 15 countries

4.3.3 Validation goal 3: degree of adaptability of VTManager to new tools and collaboration patterns

To achieve this validation goal, it was necessary to identify the supporting tools used to manage each of the globally distributed groups considered for validation purposes. The use of these tools was monitored annually throughout the pilot project, retrieving evidence of the tools used such as registries, minutes, logs of the tools and sets of messages interchanged using these tools. After the data gathering, annual interviews were carried out to confirm that the evidence collected was objective and representative.

5 VTManager definition

VTManager provides a set of effective guides and activities for training, developing and managing global software teams, from the earliest phases of construction to its disbandment. It also identifies some tools that support the execution of each activity as well as the competencies needed to develop the activity properly.

The activities VTManager proposes are grouped into five main phases (Fig. 3) according to the purpose of the activities to be developed. A new lifecycle for distributed teams was also defined because of the special features of global software teams. First, team training is executed throughout the lifecycle of the global software team. The team development phase in the state of the art was eliminated because the periodic assessment of team behavior is carried out in the team performance phase. Other performance management-related activities were introduced in the team training phase. This organization was the result of implementing the practices identified in the state of the art section in real projects.

The phases and their objectives are explained below. Each one is represented by a four-field figure. Field one shows the name of each activity, field two its description, field three

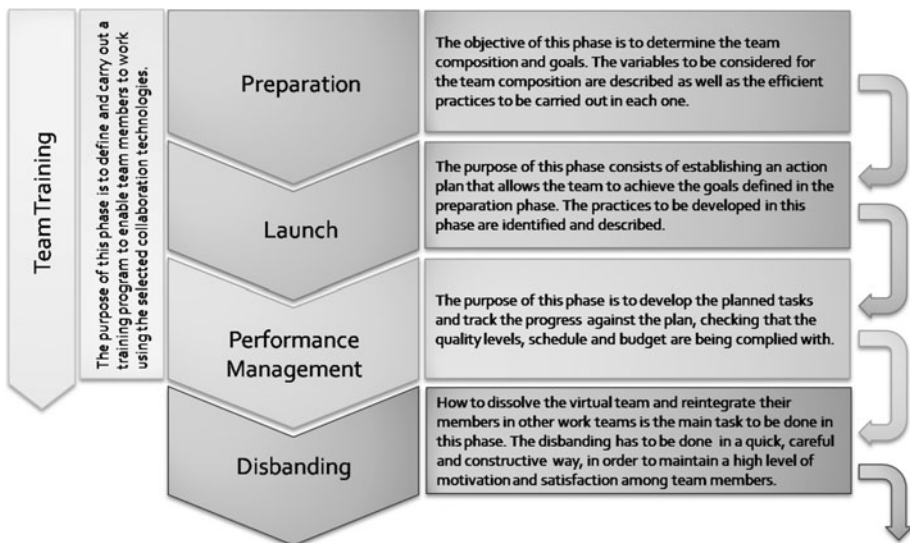


Fig. 3 Phases of VTManager

	Activities	Description	Skills and abilities	Tools
Preparation	Mission statement	Define the team mission in the project.	Meeting management; identification, estimation, prioritization and management of requirements, organization and planning.	Generating and structuring ideas, videoconference, teleconference, application sharing, document sharing, monitoring progress and feedback, work according to a common plan.
	Skill and ability definition	Skills and abilities needed to do the work have to be defined. Considering the application domain, product technology, tools and methods to be used.	Identification of skills, estimation and prioritization of requirements, organization and planning.	Videoconference, teleconference, application sharing, document sharing, monitoring progress and feedback, work according to a common plan, asynchronous and distributed brainstorming.
	Get necessary resources to build the team	Resources needed to build the work team should be got. The skills and abilities of every potential team member need to be analyzed.	Interpretation of CVs.	Visualization networks of people and competences, putting people in touch with others that can help them, looking for workers.
	Technology selection	Determine what technology will be used to facilitate communication among team members.	Synchronous and asynchronous communication management in distributed environments.	Asynchronous and distributed brainstorming, videoconference, teleconference, application sharing.
	Develop Intra-team rules	Intra-team rules have to be established as well as the way communication technologies should be used to do the work efficiently.	Interpersonal conflict resolution adapted to distributed environments, organization and planning, establishment of rules to work with shared information.	Videoconference, teleconference, application sharing, document sharing, monitoring progress and feedback, work according to a common plan, asynchronous and distributed brainstorming.

Fig. 4 Preparation activities

the competences that team members involved in the activity should have to perform the activity in the best way, and field four, tools categories that can be used to support each activity and facilitate achievement of the goals of the activity.

A. Preparation: The objective of this phase is to determine the team composition and goals. Next, the variables to be considered for the team composition are described as well as the activities performed in each (Fig. 4).

The first goal of this phase is to determine the team’s mission. In order to do so, VTManager recommends creating and managing a distributed requirements list using discussion and vote tools, both synchronous and asynchronous, for estimating and prioritizing each item. Moreover, to determine the team’s mission, VTManager also considers the application of collaboration patterns such as the multimedia meeting room, which provides team members with videoconference and application sharing, and the distributed vote tool.

The second goal is to select personnel with the appropriate skills and abilities. At the beginning of this stage, VTManager suggests developing a competence tree. A competence tree is a tool that visualizes operational and core competences available in an organization in a hierarchical connection with the corporate vision and the success factors in a competitive domain (Comi and Eppler 2009). It also allows us to combine allocation of individual responsibilities with the skills and abilities available in the team (operational and core competences). To identify personnel, VTManager proposes shared blackboard mechanisms in which the team leader defines the skills and abilities each potential team member requires. In this way, the personnel are selected according to these required responsibilities and competencies and the objective evidences of their curriculum vitae that prove they have the right skills or abilities. VTManager also considers the application of videoconference tools and shared desktop computers that allow team members to hold virtual meetings in which responsibilities are assigned.

For efficient global software team management, VTManager recommends ‘technology selection’ and ‘define intra-team rules’. First, the tools team members need to work efficiently must be selected. These tools should provide an environment of trust and dynamic communication, and mechanisms that support knowledge management. Conventional software engineering and programming tools needed to develop specific technical work are also selected because they strengthen collaboration in distributed development (Sengupta et al. 2006). These tools have to be easy to integrate into the technological infrastructure of the organization where team members work (Thomas et al. 2007). Any problem concerning the availability of the tools should be solved before the launch phase to avoid lack of synergies and communication in the early phases of team building.

Finally, the activity ‘establish intra-team rules’ includes the definition of communication and participation guidelines for distributed meetings’ management, interaction among team members and how to criticize the technical contributions from others in order to avoid personal conflicts.

B. Launch: This phase consists of establishing an action plan that allows teams to achieve the goals defined in the preparation phase. A set of activities has been defined to achieve this objective (Fig. 5).

The first activity states that a high-level design of the software architecture has to be done. In software engineering, this task is usually performed by means of face-to-face meetings in which design diagrams are prepared among team members (Dekel 2005). In the case of software teams where team members are also geographically dispersed, it is necessary to define effective ways to share design information dynamically. VTManager suggests a combination of synchronous and asynchronous tools such as videoconference, collaborative workspaces, wikis and software configuration management tools (SCMTs). In order to prevent the loss and duplication of knowledge, VTManagers indicates that these tools have to provide warnings of when an asset or artifact has been modified by other team members and what has been modified. It also recommends that team members communicate with each other frequently to maintain fluid communication, a constructive working atmosphere and to coordinate the work among members of a geographically distributed team. Many tools can be used, for example, e-mail, chat, instant messaging or teleconference (Thomas et al. 2007).

The second activity is to identify the development strategy. This consists of determining the best technical approach to implementing the previously defined architecture. Although this activity requires several preparatory tasks, VTManager considers it very important to complete this activity through meetings that normally cannot be done face-to-face, so the efficient use of a multimedia meeting room is important. The application of structured meeting techniques is also important to assure the productivity of these meetings. The Nominal Group Technique could be useful for this purpose (Dowling and St. Louis 2000).

In relation to task identification, responsibility allocation and task estimation, VTManager recommends creating and managing a distributed task list including the task purpose, the person in charge, the estimated total effort, the current state (not started/in progress/completed) and an estimation of the remaining effort to complete the task. In order to manage this artifact efficiently in a geographically distributed context, VTManager suggests introducing specific practices that permit shared editing of the items included to define specific procedures for registering the current status of each task. These procedures usually change due to the need for compatibility with the management procedures established in each team location, and the support tools considered.

Finally, plans are developed. They should be detailed enough to make it easier to track and evaluate achievements.

	Activities	Description	Skills and abilities	Tools
Launch	Conceptual design	A high-level preliminary design has to be done. It should allow to estimate product size and the time needed to develop it.	Technical problem resolution, shared knowledge management, interpersonal conflict resolution adapted to distributed environments, brainstorming session management.	Document sharing, monitoring progress and feedback, work according to a common plan, change and configuration management, supporting the development lifecycle and modeling, videoconference, teleconference, application sharing.
	Identify development strategy	The strategy to develop the software product is defined. The strategy definition includes how the product will be developed and how it will be integrated.	Technical problem resolution, shared knowledge management, interpersonal conflict resolution adapted to distributed environments, brainstorming session management.	Document sharing, monitoring progress and feedback, work according to a common plan, change and configuration management, videoconference, teleconference, application sharing.
	Identify an d design task	Identify the tasks needed to carry out the project and design the WBS with the subtask to perform in each one.	Shared knowledge management, interpersonal conflict resolution adapted to distributed environments, organization and planning, brainstorming session management.	Videoconference, teleconference, application sharing, asynchronous and distributed brainstorming, document sharing, monitoring progress and feedback, work according to a common plan, videoconference, teleconference, application sharing.
	Responsibilities allocation	Every team member has to know his/her role and responsibilities. Team roles should be clarified and responsibilities associated with them described.	Interpersonal conflict resolution adapted to distributed environments, organization and planning.	Generating and structuring ideas, videoconference, teleconference, application sharing, document sharing, monitoring progress and feedback, work according to a common plan, supporting the development lifecycle and modeling.
	Task estimation	The length of each task should be estimated according to the resources previously assigned.	Shared knowledge management.	Document sharing, monitoring progress and feedback, work according to a common plan, supporting the development lifecycle and modeling, change and configuration management, videoconference, teleconference, application sharing.
	Risk assessment	Risks are identified and assessed. Moreover the impact of every risk should be prioritized.	Shared knowledge management.	Videoconference, teleconference, application sharing document sharing, monitoring progress and feedback, work according to a common plan, asynchronous and distributed brainstorming, supporting the development lifecycle and modeling, change and configuration management.
	Plan delivery	Plans have to be developed. They should be made in detail to make it easier to track and evaluate achievements.	Shared knowledge management, establishment of rules to work with shared information, organization and planning.	Document sharing, monitoring progress and feedback, work according to a common plan, change and configuration management, videoconference, teleconference, application sharing.

Fig. 5 Launch activities

C. Performance management: The purpose of this phase is to develop the planned tasks and track progress against the plan, checking that the quality levels, schedule and budget are being complied with. Figure 6 shows the activities to be performed in this phase.

The first challenge for performance management in a global software team is effective leadership. With VTManager, the team leader has to motivate and help team members to achieve their goals. Motivation and goal identification make the team work efficiently. In this context, leadership refers to the responsibility of decision-making as regards project and team management; such decisions are made according to the current state of both project and team. VTManager suggests intensively gathering and disseminating relevant information on current performance and synergies. In relation to synergies, real and updated information on social processes established among team members must be obtained. The more fluent the personal relationships among team members, the better they communicate, resulting in greater success. There are many tools that allow registration of

	Activities	Description	Skills and abilities	Tools
Performance management	Leadership	The leader has to motivate and help virtual team members achieve their goals. Motivation and goal identification will make the team work efficiently.	Ability to use advanced techniques of shared communication, ability to solve conflicts among team members, positive attitude and motivational skills, initiative and leadership.	Videoconference, teleconference, application sharing, document sharing, monitoring progress and feedback, work according to a common plan, change and configuration management.
	Regulation of communication	It is important to choose the best suited medium for the type of information to be exchanged. This decision should be based on the type of action to be performed.	Collection, interpretation and analysis of statistical information, analysis of synergies using information collected for the social networks.	Videoconference, teleconference, application sharing, document sharing, monitoring progress and feedback, work according to a common plan, wikis, blogs.
	Knowledge management	This task should be performed to increase mutual knowledge about the individual working, enhance equal information distribution, systematic processing of unshared information and delete repeated information.	Shared knowledge management, ability to use advanced techniques of shared communication.	Document sharing, monitoring progress and feedback, work according to a common plan, videoconference, teleconference, application sharing, change and configuration management, wikis, blogs, data base.
	Task development	Software project should be developed in its different stages: requirements, design, implementation, integration and system tests and documentation.	Shared knowledge management, ability to use advanced techniques of shared communication, technical problem resolution, positive attitude and motivational skills, identification, estimation, prioritization and management of requirements.	Document sharing, monitoring progress and feedback, work according to a common plan, videoconference, teleconference, application sharing, change and configuration management, supporting the development lifecycle and modeling, integrated development environments.
	Tracking progress against the plan	At the same time that activities associated with the other stages are being developed, they should be tracked against the plan.	Shared knowledge management, collection, interpretation and analysis of statistical information, technical problem resolution, ability to use advanced techniques of shared communication.	Document sharing, monitoring progress and feedback, work according to a common plan, videoconference, teleconference, application sharing, change and configuration management.

Fig. 6 Performance management activities

this information: the challenge is to integrate them correctly with the team processes and the technological environments implemented in organizations.

One source of team inefficiency is mismanagement of the knowledge to be shared by team members (Thomas et al. 2007). In order to prevent this undesirable situation, the specific practices included in VTManager contribute to increasing mutual knowledge of individual working, enhancing equal information distribution and systematic processing of unshared information. Therefore, it should be noted that periodic (virtual and/or face-to-face) meetings among team members strengthen commitment (Powell et al. 2004). Finally, VTManager also recommends that each team member take charge of several team management responsibilities, thereby increasing trust among team members and motivating them.

For efficient implementation of these practices, VTManager states that it is appropriate to use tools for shared knowledge management (enabling mechanisms for asset linkage and shared access and use of project assets); and collaborative team and individual progress tracking. In addition, a multimedia meeting room should be available for virtual meetings for enabling decision-making processes at task level.

D. Disbanding: The purpose of this phase is to decide on the disbanding of the geographically distributed software team as well as how to re-integrate their members into other teams.

Team disbanding has to be done quickly, carefully and constructively to maintain a high level of motivation and satisfaction among employees and to facilitate their rapid integration in other work teams. Figure 7 shows the activities to be performed in this phase.

	Activities	Description	Skills and abilities	Tools
Disbanding	Revise work to ensure it is finished	All work performed by the team has to be revised to determine if the work is finished. Team members have to examine the work done to determine what issues could be improved in future.	Critical analysis of information.	Document sharing, monitoring progress and feedback, work according to a common plan, change and configuration management.
	Register data as organizational assets	Any document or information which can be useful for future projects have to be registered in the organizational asset library.	Shared knowledge management, ability to use advanced techniques of shared collaboration.	Data base, wikis, document sharing.
	Acknowledge the achievements of the team work	Individual and team achievements have to be recognized both collectively and individually. Moreover, team members will be rewarded according to the reward system initially established.	Collection, interpretation and analysis of statistical information.	Change and configuration management, document sharing, monitoring progress and feedback, work according to a common plan.
	Reintegrate team members	Team members have to be reintegrated with other work teams. These teams can either be existing or new .	Meeting management.	Videoconference, teleconference, application sharing.
	Register lessons learned	Experiences and best practices extracted from the project have to be documented.	Shared knowledge management, ability to use advanced techniques of shared collaboration.	Data base, wikis, document sharing, diary manager.

Fig. 7 Disbanding activities

In a global software team, these activities should be performed but several conditions should be imposed, for example registered knowledge must be available, complying with the rules on intellectual and industrial property of all organizations whose employees participate in the team. The necessary security measures to avoid fraudulent or inappropriate use of the assets must be implemented as well.

‘Lessons learned’ mechanisms must also be available from the beginning of the project to facilitate continuous improvement of the team. Besides these tools, wikis and team diary management tools, which can help to identify practices to be used in other projects and those that must be rejected, should be available to team members.

E. Team training: The purpose of this phase is to define and carry out a training program to enable team members to work with the collaboration technologies selected. To achieve this goal, a set of activities has to be performed (Fig. 8).

As team members usually have different levels of skills and experience in the collaborative working environment selected, the training program should include communication elements of electronic media, clarification of goals, roles and working in culturally diverse environments. This training program is usually in the form of training in the first stage and coaching in the second, but it can be adapted to the specific characteristics of the project.

Several activities in this phase must be carried out before the global software team is launched. These are to identify the needs and/or deficits of the team and its members. It should be based on an empirical assessment and the experience of the personnel in charge of this task. The results of this assessment determine the most suitable strategy to implement the required training.

Other activities of the training phase must be carried out throughout the life of the software team in order to execute the training program. There should be suitable descriptions of the methods and tools team members have to use, and the effects of training

	Activities	Description	Skills and abilities	Tools
Team training	Identification of needs and/or deficits of team and its members	Identification of the needs and/or deficits of the team and its members. This identification should be based on an empirical assessment and the experience of the personnel in charge of this task.	Ability to use advanced techniques of shared collaboration, critical analysis of information	Videoconference, teleconference, application sharing, document sharing, monitoring progress and feedback, work according to a common plan, change and configuration management.
	Determine the most suitable type of training	The type of training is defined. This training provides the team members with the competences needed to work in a virtual team.	Adaptation of types of training according to knowledge deficits.	Generating and structuring ideas, videoconference, teleconference, application sharing.
	Execution of training program	Execute the training program for the work team members who had deficits. A good training program should have a suitable description of the methods and tools that team members have to use during the project.	Self-learning capability, ability to use advanced techniques of shared collaboration.	Learning management systems, electronic process guides, wikis, document sharing.
	Assess the effects of previous training	The effectiveness of the previous training the virtual team members received have to be assessed. It is usually done empirically.	Critical analysis of information.	Wikis, learning management systems, document sharing, monitoring progress and feedback, work according to a common plan change and configuration management.

Fig. 8 Team training activities

should be assessed throughout the life of the team to prevent ineffectiveness through misuse of collaboration patterns and tools.

On completion of the initial training, the team leader must carry out periodic assessments to detect problems that team members experience in applying VTManager. If members run into difficulties, the team leader has to establish back-up training activities to reduce these deficiencies. These activities can be carried out in different ways using VTManager. First, an expert coaches team members on the application of one specific technique or tool. Second, guides with back-up training material are hosted in a wiki so that they can be consulted periodically, facilitating the inclusion of tips and actions not to be repeated.

6 Results and discussion

6.1 Validation goal 1: effectiveness and efficiency of global software team management

The effectiveness and efficiency of a global software team using VTManager were determined by gathering and analyzing data on time needed for preparation and launch, delays in submission of deliverables, and collaborative tools selected to support the team. The values that teams obtained for these indicators are shown in Table 2.

The values registered for time to prepare and launch can be considered positive because the values (1 week or month) are low in comparison with the duration of the pilot project (36 months). Similarly, the values registered can also be considered low (1 week–2 months in a 36 months project).

During the final review the practitioners indicated that the effective practices contemplated constitute a good instrument to check the correct management of the global software team and to identify areas where the team needs to improve. Team leaders used the information VTManager provided to select the most appropriate tools to manage the

Table 2 Effectiveness results of VTManager pilot application

Team	Time to prepare and launch	Delays in submission of deliverables	Tools selected to support the team
1. Project Executive Management Board.	1 week	1 week	Project Coordinator, e-mail and audio conferences.
2. Interdisciplinary Group for CWE R&D in Spanish Settings.	1 month	2 weeks	Project Coordinator, e-mail, audio conferences, wiki, blog and Moodle.
3. Definition of architectures for heterogeneous SCTs interoperability.	1 month	2 months	Project Coordinator, e-mail, Isabel and audio conferences.
4. Validation of the architecture for heterogeneous SCTs interoperability.	1 week	1 month	Project Coordinator, e-mail and audio conferences.

team. This selection was adapted to the different purposes and duration of each team and the tools that were available.

Finally, it is important to point out that the tools all the teams selected supported synchronous communication (audio/video conference and chat) and asynchronous communication (e-mails, forums and sharing document facilities). The same applied to other more sophisticated tools shown in Sect. 6.3.

6.2 Validation goal 2: evaluation of the difficulty of applying VTManager to different types of teams

Most of the VTManager activities were carried out by the teams involved in the pilot project. In this section, we present the degree of correctness found in the application of VTManager activities and discuss the problems identified.

Table 3 shows the degree of correctness in the application of VTManager activities in the pilot project. These activities are organized in the same order they were defined in Sect. 5 to facilitate analysis. Each column in Table 3 represents one team in this research and each row an activity of VTManager. Each cell between one activity and one team may contain the following information:

- Activity the team did not perform.
- Activity the team performed. In this case, the authors of this paper studied whether it had been carried out with collaborative support. An activity is performed with collaborative support when any practice or technology was used to support collaboration among team members who are dispersed in different locations. In this case, there are two options:
 - Activity performed with collaborative support.
 - Activity performed without collaborative support.

From the results shown in Table 3, we can draw some conclusions by phase for each team.

Preparation: All the activities were performed but ‘skill and ability’ definition and ‘get necessary resources to build the team’ were carried out without collaborative support

Table 3 Implementation degree of VTManager in the pilot project

Phase	Activity	Team 1	Team 2	Team 3	Team 4
Preparation	Skill and ability definition	■ ○	■ ○	■ ○	■ ○
	Get necessary resources	■ ○	■ ○	■ ○	■ ○
	Mission statement	■ ●	■ ●	■ ●	■ ●
	Technology selection	■ ●	■ ●	■ ●	■ ●
	Develop intra-team rules	■ ●	■ ●	■ ●	■ ●
Launch	Conceptual design	□	■ ●	■ ●	□
	Identify development strategy	□	■ ●	■ ●	□
	Identify and design tasks	■ ●	■ ●	■ ●	■ ●
	Responsibilities allocation	■ ●	■ ●	■ ●	■ ●
	Task estimation	□	■ ●	■ ●	□
	Risk assessment	■ ●	□	□	□
Performance management	Plan delivery	■ ●	■ ●	■ ●	■ ●
	Leadership	■ ●	■ ●	■ ●	■ ●
	Regulation of communication.	■ ●	■ ●	■ ●	■ ●
	Knowledge management	■ ●	■ ●	■ ●	■ ●
	Task development	■ ○	■ ●	■ ●	■ ○
Disbanding	Tracking progress against the plan	■ ●	■ ●	■ ●	■ ●
	Revise work to ensure it is finished	■ ●	■ ●	■ ●	■ ●
	Register data as organizational assets	■ ●	■ ●	■ ●	■ ●
	Acknowledge the achievements of the team work	■ ○	■ ○	■ ○	■ ○
Team training	Register lessons learned	■ ●	■ ●	■ ●	■ ●
	Identification of needs and/or deficits	■ ○	■ ○	■ ○	■ ○
	Determine the most suitable type of training	■ ○	■ ○	■ ○	■ ○
	Execute the training program	■ ○	■ ○	■ ○	■ ○
	Assess the effects of previous training	■ ○	■ ○	■ ○	■ ○

because they had been performed prior to the team mission statement. Coaching activities began shortly after the teams started their work, so some activities were not carried out in the right order. As a result, in some cases, ‘personnel selection’ had to be repeated because ‘people profile’ did not correspond to the team objective. This situation reinforces the order established in VTManager for the preparation phase.

The mission statement was prepared as a distributed requirements lists. This type of instrument was evaluated very positively because it contributed to defining team goals in terms of specific objectives and to managing the team work efficiently. During this phase, the technology to manage team activity was selected but the restrictions of all sites where team members work could not be addressed. As a result, several functionalities were unavailable to members working in some locations, disrupting communication in a few cases. Finally, it is relevant to mention that the rules established prevented personal conflicts among team members.

Therefore, we can conclude that the selected technology allowed the activities VTManager defines for this phase to be implemented.

Launch: During this phase, teams 2 and 3 did a high-level design of the software architecture to identify the development strategies and to determine specific development tasks. In this case, team members experienced more difficulties with fluent meetings due to their lack of previous experience in the protocols used to create and modify diagrams collaboratively during the virtual meetings. In order to ensure that agreements among team members were based on a correctly shared vision and to prevent future problems, a face-to-face meeting was held for teams 2, 3 and 4. The conclusion was that this face-to-face meeting demonstrated the appropriateness of the results obtained during the virtual collaborative work. These meetings helped to find and fix several missing details.

Teams 1 and 4 did not require a high-level design of the software architecture to define their strategy and tasks because these tasks were specified in a distributed task list. Face-to-face meetings were not required to complete this phase for these two teams; they did not perform the conceptual design activities or identify development strategy as their goal was not to develop a software product. Task estimation was not carried out because the teams executed management-related tasks and these tasks did not require the type of estimation proposed by VTManager.

The use of ‘distributed task lists’ VTManager proposed was considered positively by team members. These distributed task lists were used as planning elements and their publication meant that the plan delivery activity was carried out. In the pilot project, only team 1 performed the risk assessment activity because of the nature of the project. It was decided that all teams reported their initial state to the Project Management Board (team 1), which centralized all risk management activities. This allows us to establish that the activities VTManager proposed can be adapted to different circumstances that usually arise in global software projects.

Performance management: During this phase, leadership was based on management by objectives and feedback. Team members knew their responsibilities and communicated their progress to other team members and the leader. The leader periodically revised the work done by team members using electronic media to check the state of the project and task progress percentage. These tasks could be done collaboratively using the collaborative planning and tracking tool and workspaces.

Tools for analyzing synergies were not available during the project, the social processes and interpersonal relationship could not be tracked closely, virtual meetings and teleconferences only helped team members and leaders to know the degree of motivation and state of mind of the others.

Shared knowledge management was based on a shared repository that supported software configuration management. Team members managed it efficiently, periodically updating contents and sharing the information and assets that each team member was working on.

In the case of teams 2 and 3, task development was performed by team members using a repository with control of versions to share code and software development tools. E-mail, chat and teleconferences were used to solve problems during the software development and to take decisions. Nevertheless, face-to-face sessions were necessary to integrate the different developed components.

One of the most highly regarded aspects of the task allocation performed in the preparation phase was individual task allocation, so teams 1 and 4 did not need collaborative support to carry out their tasks.

Finally, assigning team management responsibilities to most team members was perceived as positive because they became more committed to the team and the project and were more motivated to develop their work.

Disbanding: In order to enhance knowledge sharing among different project teams, the intermediate results were stored in public repositories. ‘Registered experiences and organizational assets’ allowed team members to use them in subsequent project stages and know what should be improved. For both practices, an information repository was available and any team member could introduce experiences and assets that were considered useful for the future and access them whenever necessary. However, one team member managed each repository in order to avoid introducing repeated or weak experiences or assets.

Team training: Team members were trained in VTManager activities, which consisted of coaching during the project stages. The coaching program helped team members to develop the right skills and abilities to work in global software teams. After the coaching sessions, the program was assessed for possible future improvement based on these first four experiences. The program execution can be regarded as successful because most of the members of the four teams developed the right competencies to work on this type of team.

From a global point of view, team members considered that following the efficient practices reduced the time needed to perform team management activities in each phase; they had a specific list of practices for team management and how to perform them in a global context. The team members also indicated that the practices reduced of communication, collaboration and knowledge management problems through improved communication and coordination of the global software team, but they thought more powerful tools should be used to maximize the team performance.

6.3 Validation goal 3: degree of adaptability of VTManager to new tools and collaboration patterns

During the project, the coaches recommended the activities defined in VTManager. However, their correct implementation depended on the capability of the technological solutions used to support the activities and restrictions on installation and use imposed by the work environment. So, in order to analyze the success of the efficient practices of VTManager, it is very important to analyze the support capability of the technologies used for the collaboration patterns. Table 4 provides information to show that the technology used in this case study was suitable for implementing the collaborative situations stated in the VTManager definition.

Team members used audio conference, chat and e-mail for their non-face-to-face communication. The audio conferences supported team meetings when the team could not use the videoconferencing tool. The teams also used audio conference, chat and e-mail to solve problems, and Project Coordinator to share information.

Isabel was the tool considered for implementing the required multimedia room functionality. All the teams used this tool for videoconferencing, solving problems and holding global software team meetings and audio conferences.

Only team 2 used Wikis, blogs, IDEs and CVS tools to facilitate the daily work of team members.

Considering the tools that the different team members used and the results obtained in the pilot project, the authors of this paper have identified several issues relating to human aspects and productivity that should be contemplated before introducing a new technology to a software team.

New collaboration technologies require a change in human interaction. If users do not experience immediate personal benefit, new applications should not be applied even if the new tools are properly introduced and initial training provided. Early adoption of new

Table 4 Tools used in the pilot project to support each collaborative situation

Collaborative situation	Phase	Tool used in the pilot project
Facilitate access and use of shared data to team members	Preparation	Project coordinator
	Launch	Project coordinator
	Performance management	Project coordinator, Wikis, CVS, IDEs, Blogs
	Team training	Moodle, Wikis, CVS
Know why team member is interested in the work of another and what he/she wants it for	Launch	Project Coordinator, Isabel, audio conference, e-mail
	Performance management	Project Coordinator, Isabel, audio conference, chat, e-mail, Skype
Register the activities that team members have done with the shared information	Launch	Project coordinator, CVS
	Team training	Project coordinator, CVS, Moodle, Wikis
Avoid the conflicts that appear when team members work at the same time on different copies of the same document	Launch	Project coordinator
	Performance management	Project coordinator, CVS, IDEs
	Disbanding	Project coordinator, CVS
	Team training	Project coordinator, CVS
Negotiate the form of collaboration with other team members who are working on the same information item	Preparation	Isabel, e-mail, audio conference
	Launch	Isabel, e-mail, audio conference, Skype
	Performance management	Isabel, e-mail, audio conference, chat, Skype
	Disbanding	E-mail, audio conference, chat, Skype
Facilitate team members working on the same work items at the same time	Preparation	Project coordinator
	Launch	Project coordinator
	Performance management	Project coordinator, CVS, IDEs
Report to control and track the work performed by the different members	Launch	Project coordinator
	Performance management	Project coordinator, CVS
	Disbanding	Project coordinator, CVS
	Team training	Moodle, CVS
Allows users to attach their comments to artifacts	Preparation	Project coordinator
	Launch	Project coordinator
	Performance management	Project coordinator, Wikis, blogs, IDEs
	Disbanding	Project coordinator, Wikis
	Team training	Moodle, Wikis
Negotiate and establish agreements with other team members	Preparation	Isabel, audio conference, e-mail, Skype
	Launch	Isabel, audio conference, e-mail
	Performance management	Isabel, audio conference, e-mail, chat, Skype
	Disbanding	Isabel, audio conference, e-mail
	Team training	Chat, Moodle, e-mail

collaboration technologies (i.e.: video conference or presence awareness) often occurs in families or communities. Since companies are more reluctant to introduce these tools for various reasons (i.e., security, IPR, etc.), professionals are often frustrated because they cannot share as much information as they would like to.

It is a common understanding among the members of the pilot project that today's collaboration tools have already increased productivity but, on the other hand, they have also increased the workload. When users first begin to use the types of tools VTManager recommends, they often feel like a "rat in the wheel". This is in line with a change in the expectation of response time.

Collaboration technologies often result in an interrupt-driven work environment, making interruptions the normal behavior of the day. Some people cannot adapt to this new way of working and this results in a reduction in quality due to stress and interruptions.

To become useful and increase productivity, VTManager-recommended tools that must quickly reach a critical mass in an organization. Therefore, old systems or procedures must be switched off soon after the introduction of a new tool for managing virtual teams to get all the users onto the new systems. When collaboration tools are used among organizations, different work cultures sometimes lead to communication problems.

The experience gathered during the application of VTManager in the pilot project indicates that electronic meetings using audio and video communication are common nowadays. The essential prerequisite is, however, a high quality audio and video link. Users also found that multipoint meetings are less tiring than point-to-point meetings. The growing use of presence tools (MS-Messenger, Skype, Yahoo Messenger), in combination with face-to-face meetings, often has a positive effect on team building.

Today's collaboration technologies are still prescriptive and often designed to support the traditional way of working. Therefore, tools must become more flexible and should perhaps even be designed for unforeseen use.

New collaboration tools often only focus on a specific solution or collaboration task, without considering the integration of this process into the larger environment and process. In most cases, this creates incompatibilities among applications, data and usage patterns. As a result, the refined tools are not applied and there is no uptake of the collaborative tools.

7 Conclusions

The management of globally distributed software teams is complex because of problems of linguistic differences, geographical dispersion, different time zones and the cultural diversity of the team members. These problems are amplified when a single software development team is composed of highly skilled individuals working in dispersed geographical locations.

We have studied how traditional approaches to managing on-site software development teams have been adapted to manage globally distributed software teams. These initiatives have not been completely successful due to the lack of integration of explicit practices for global virtual team management, definition of skills and abilities needed to work in these teams, availability of collaborative work environments and shared knowledge management practices.

In this paper, the authors define VTManager, an approach that includes a set of effective practices that complement already existing methods to manage geographically distributed software teams. VTManager provides the effective practices structured into processes and

activities. These practices integrate the technical activities of software development with others, enabling effective management of geographically distributed teams. These practices are oriented toward:

- Managing effective communication among team members, avoiding problems related to language, culture and time differences.
- Facilitating the integration of the skills and abilities needed to manage the team during the project.
- Providing effective mechanisms to introduce technology support essential to the success of global software development teams.
- Incorporating effective mechanisms for shared knowledge management in distributed environments.

The paper presents a case study where VTManager was applied to guide the management of several virtual teams. This case study has permitted us to validate that the application of the effective practices proposed contributes to managing and controlling properly the performance of a global software team.

The future research on VTManager is centered on the definition of a model to assess and determine improvement action lines for global software teams launched and managed in a specific organization.

Acknowledgments This contract is partially supported by the Integrated Research Project C@R—Collaboration at Rural (IST-2006-034921), funded by the European Commission, specifically the Directorate-General Information Society Technologies and Media (DG-INFO). The research work for this paper has been also funded by TRAGSA under the projects UC3M/2008/00259/001 and UC3M/2007/04245/001.

Appendix: Software tools used in C@R project for coordination and collaboration

This appendix describes briefly tools recommended in VTManager. So the main functionalities to be available in tools for facilitating collaboration and communication can be identified (Table 5).

Table 5 Tools used in C@R project for coordination and collaboration

Tool	Description	Reference
CVS	The Concurrent Versioning System (CVS) is a software revision control system in the field of software development. Version control system software keeps track of all work and all changes in a set of files, and allows several developers to collaborate	http://www.nongnu.org/cvs/
Isabel	The ISABEL CSCW application is a group collaboration tool for the Internet based on advanced videoconferencing features. It provides a set of shared applications, such as: built in notepad, built in whiteboard, with connection to hardware blackboards; linux application sharing and windows application sharing	http://www.agora-2000.com/pdfs/Isabel-4.9_Introduction.pdf
Moodle	Moodle is a Course Management System (CMS), also known as a Learning Management System (LMS). It is a Free web application that educators can use to create effective online learning sites	http://moodle.org/

Table 5 continued

Tool	Description	Reference
NetBeans	NetBeans refers to both a platform framework for Java desktop applications, and an integrated development environment (IDE) for developing with Java, JavaScript, PHP, Python, Ruby, Groovy, C, C++, and Clojure	http://netbeans.org/
Project Coordinator	The main functionalities of Project Coordinator are access the updated information; share a common document, archive and create a knowledge bank for the workgroup or project; work according to a common project plan; share project news, developments and results; monitor and give feedback on the progress of the project; make the results achieved public and manage the co-workers' input and participation in different project phases or activities	http://www.designtech.se/projectweb/portalproject/Overview-PCX.html
Skype	Skype is software that enables the world's conversations. It is to make free video and voice calls, send instant messages and share files with other Skype users	http://share.skype.com/sites/garage/2009/10/12/ReleaseNotes_4.1.x.179_hotfix_3.pdf
Visual Studio 2008	Microsoft Visual Studio is an Integrated Development Environment from Microsoft. It can be used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight	http://msdn.microsoft.com/en-gb/vstudio/default.aspx

References

- Battin, R. D., Crocker, R., Kreidler, J., & Subramanian, K. (2001). Leveraging resources in global software development. *IEEE Software*, 18(2), 70–77.
- Braun, A. (2007). A framework to enable offshore outsourcing. In *Proceedings of the 2nd international conference on global software engineering* (pp. 125–129). IEEE Computer Society Press.
- C@R Consortium. (2005). *C@R: A collaboration platform for working and living in rural areas*. Retrieved from EU project with reference FP6-2005-IST-5-034921, <http://www.c-rural.eu>. Accessed May 25, 2008.
- Carmel, E. (1999). *Global software teams: Collaboration across borders and time zones*. Saddle River: Prentice Hall.
- Casey, V., & Richardson, I. (2008). Virtual teams: understanding the impact of fear. *Software Process: Improvement and Practice*, 13(6), 511–526.
- Cohen, S. G., & Gibson, C. B. (2003). Putting the team back in virtual teams. In *The 18th annual conference of the society for industrial/organizational psychology, Orlando*.
- Comi, A., & Eppler, M. J. (2009). Visualizing organizational competences: Problems, practices, perspectives. In *Proceedings of I-KNOW'09 and I-SEMANTICS'09* (pp. 115–127). Semantic Web Company.

- Dekel, U. (2005). Supporting distributed software design meetings: What can we learn from co-located meetings? In *Proceedings of the 2005 workshop on human and social factors of software engineering*. ACM.
- Dowling, K. L., & St. Louis, R. D. (2000). Asynchronous implementation of the nominal group technique: Is it effective? *Decision Support Systems*, 29(3), 229–248.
- Durnell, C., & Webber, S. (2005). Relationships among geographic dispersion, team processes, and effectiveness in software development work teams. *Journal of Business Research*, 58(6), 758–765.
- Ebert, C., & De Neve, P. (2001). Surviving global software development. *IEEE Software*, 18(2), 62–69.
- Edwards, H. K., & Sridhar, V. (2003). Analysis of the effectiveness of global virtual teams in software engineering projects. In *Proceedings of the 36th Hawaii international conference on system sciences*. IEEE Computer Society Press.
- Eva, M. (1994). *SSADM version 4: A user's guide*. New York: McGraw-Hill.
- Fuller, M., Hardin, A., & Davison, R. (2007). Efficacy in technology-mediated distributed teams. *Journal of Management Information Systems*, 23(3), 209–235.
- Gaudes, A., Hamilton-Bogart, B., Marsh, S., & Robinson, H. (2007). A framework for constructing effective virtual teams. *The Journal of E-working*, 1(2), 83–97.
- Herbsleb, J. D. (2007). Global software engineering: The future of socio-technical coordination. In *The Proceedings of the future of software engineering conference* (pp. 188–198). IEEE Computer Society Press.
- Herbsleb, J., & Moitra, D. (2001). Global software development. *IEEE Software*, 18(2), 16–20.
- Hertel, G., Geister, S., & Konradt, U. (2005). Managing virtual teams: A review of current empirical research. *Human Resource Management Review*, 15(1), 69–95.
- Humphrey, W. S. (2000). *Introduction to the team software process*. Massachusetts: Addison Wesley Longman.
- Humphrey, W. S. (2006). *TSPSM—coaching development teams*. Upper Saddle River: Addison-Wesley Professional.
- Hyrkkänen, U., Putkonen, A., & Vartiainen, M. (2007). Complexity and workload factors in virtual work environments of mobile work. *Ergonomics and Health Aspects*, 4566, 85–94.
- Institute of Electrical and Electronics Engineers (IEEE). (1993). *Recommended practice for software requirements specifications* (December 2, 1993), IEEE Std 830–1993.
- IPRC. (2007). *A process research framework*. Pittsburgh: Software Engineering Institute.
- Jarvenpaa, S. L., Knoll, K., & Leidner, D. E. (1998). Is anybody out there?: Antecedents of trust in global virtual teams. *Journal of Management Information Systems*, 14(4), 29–64.
- Juristo, N., & Moreno, A. M. (2001). *Basics of software engineering experimentation*. The Netherlands: Kluwer.
- Kankanhalli, A., Tan, B., & Wei, W. (2007). Conflict and performance in global virtual teams. *Journal of Management Information Systems*, 23(3), 237–274.
- Karolak, D. W. (1999). *Global software development: Managing virtual teams and environments*. Los Alamitos: IEEE Computer Society Press.
- Kruchten, P. (2004). *The rational unified process an introduction*. Boston: Pearson Education.
- Larman, C. (2004). *Agile and iterative development: A manager's guide*. Boston: Addison-Wesley.
- Lee-Kelley, L., & Sankey, T. (2007). Global virtual teams for value creation and project success: A case study. *International Journal of Project Management*, 26(1), 51–62.
- Matloff, N. (2005). Off-shoring: What can go wrong? *IT Professional*, 7(4), 39–45.
- Ministerio de Administraciones Públicas (MAP). (2009). *Metodología MÉTRICA versión 3*. <http://www.csae.map.es/csi/metrica3/index.html>. Accessed June 25, 2009.
- Ortiz de Guinea, A., Webster, J., & Staples, S. (2005). A meta-analysis of the virtual teams. In *The symposium on high performance professional teams*, Queen's University, Kingston, Canada.
- Petkovic, D., Todtenhoefer, R., & Thompson, G. (2006). Teaching practical software engineering and global software engineering: Case study and recommendations. In *Proceedings of the 36th ASEE/IEEE frontiers in education conference* (pp. 19–24). IEEE Computer Society Press.
- Powell, A., Piccoli, G., & Ives, B. (2004). Virtual teams: A review of current literature and directions for future research. *ACM SIGMIS*, 35(1), 6–36.
- Prikladnicki, P., Nicolas, J. L., & Evaristo, R. (2003). Global software development in practice. *Software Process Improvement and Practice*, 8(4), 267–281.
- Project Management Institute (PMI). (2004). *PMBOK: A guide to the project management body of knowledge* (3rd ed.). Newtown Square: Project Management Institute.
- Ramasubbu, N., & Krishnan, M. S. (2005). A process maturity framework for managing distributed development. *IEEE Software*, May/June 2005, 80–86.

- Rosen, B., Furst, S., & Blackburn, R. (2007). Overcoming barriers to knowledge sharing in virtual teams. *Organizational Dynamics*, 36(3), 259–273.
- Schümmer, T. (2003). GAMA: A pattern language for computer supported dynamic collaboration. In *Proceedings of the 8th European conference on pattern languages of programs* (pp. 53–113). Springer.
- Schwaber, K., & Beedle, M. (2001). *Agile software development with scrum*. Upper Saddle River: Prentice Hall.
- Sengupta, B., Chandra, S., & Sinha, V. (2006). A research agenda for distributed software development. In *The proceedings of the 28th international conference on software engineering* (pp. 73–740). ACM Press.
- Smite, D. (2004). Global software development project management-distance overcoming. In *Proceedings of 11th European conference on Software process improvement* (pp. 23–33). Springer.
- Striukova, L., & Rayna, T. (2008). The role of social capital in virtual teams and organisations: corporate value creation. *International Journal of Networking and Virtual Organisations*, 5(1), 103–119.
- Thomas, D. M., Bostrom, R. P., & Gouge, M. (2007). Making knowledge works in virtual teams. *Communications of the ACM*, 50(11), 85–90.
- Wallace, S. (2007). *The ePMbook*. <http://www.epmbook.com/> Accessed September 17, 2007.

Author Biographies



Javier García Guzmán is an associate professor in the Computer Science Department at Carlos III University of Madrid. He is a software process improvement consultant in Progesion. He has 10-year experience as a software engineer and consultant in public and private companies. He received his PhD in computer science from the Carlos III University of Madrid. His current research interests are related to SPI in small settings, measurement of software processes, ISO 15504 assessments, software capacity quick audits, evaluations for pre-diagnosis according to ISO 15504 and CMMI and management of knowledge related to software engineering.



Javier Saldaña Ramos received an Engineering degree in Computer Science at Carlos III University of Madrid, and Msc in Computer Science and Technology by the same university. He is a PhD student and associate professor in the Computer Science Department at the Carlos III University of Madrid. His research on software engineering is focused on software process improvement through teamwork process improvement. It involves team management improvement, virtual team management improvement and the use of TSP (Team Software Process) in teamwork. He has several years of experience as a software engineer.



Antonio Amescua Seco is a full professor in the Computer Science Department at Carlos III University of Madrid. He has been working as software engineer in a public company (Iberia Airlines) and in a private company (Novotec Consultores) as a software engineering consultant. He founded Progesion, a spin-off company, in order to offer advanced Software Process Improvement Services. He received his PhD in computer science from the Universidad Politécnica of Madrid. His research interests include Software Process Improvement, Software Reuse, Software Project Management.



Ana Sanz Esteban received an Engineering degree in Computer Science at Carlos III University of Madrid, and Msc in Computer Science and Technology by the same university. She is a PhD student associate professor in the Computer Science Department at the Carlos III University of Madrid. Her research on software engineering is focused on software process improvement, especially test process improvement. It involves different maturity models related with organization in general like CMMI as well as related only with test process like TMM or TPI. She has several years of experience as a software engineer.