RTDWD: Real-Time Distributed Wideband-Delphi for user stories estimation

Giovanni Aiello[1], Marco Alessi[1,2], Massimo Cossentino[3], Alfonso Urso[3], Giuseppe Vella[2]

[1] Engisud S.p.A. - Research and Development Lab. - Palermo, Italy
[2] Engineering Ingegneria Informatica S.p.A. - Research and Development Lab. - Palermo, Italy
[3] ICAR-CNR Istituto di Calcolo e Reti ad Alte Prestazioni Consiglio Nazionale delle Ricerche, Palermo, Italy
{giovanni.aiello, marco.alessi, giuseppe.vella}@eng.it
{cossentino, urso}@pa.icar.cnr.it

Abstract. This paper proposes the RTDWD (Real-time Distributed Wideband-Delphi) tool, a real-time collaborative web application for user stories estimation through the Wideband-Delphi method. RTDWD realizes, in a lightweight way, virtual meetings for a critical phase of the requirements management in distributed Agile development processes, such as Distributed eXtreme Programming. The web 2.0-based nature of RTDWD adds new communication modes to a distributed Agile development process, where a close real-time collaboration is needed but difficult to realize due to the geographic dislocation of team members. Features of RTDWD allow to take into consideration several scenarios where mobile devices (i.e. Pocket PCs and Smartphones) well substitute desktop and laptop computers. We present our experience in order to point out to the researcher community the usefulness of the RTDWD and, generally, of the lightweight real-time collaboration underlining the need to introduce new technologies on practices of distributed Agile processes.

1 Introduction

In the last years, the research area on software engineering aimed at Agile development processes causing great interest both of academic and industrial companies. Moreover, recently, also the embedded software market seems to be interested in Agile methodologies, because they propose lightweight development processes aiming to carry out a logarithmic trend of the requirements change cost according to the project duration [1].
The evaluation both in academic and industrial areas of the Agile methodologies has shown very good results if applied to small/medium co-localized working groups. Moreover a common principle of every agile framework is the continuous collaboration and communication among all team members and the customer, preferring face-to-face conversations[4]. These considerations help to underline

[4] Manifesto for Agile Software Development, http://agilemanifesto.org/

the difficulties to apply agile methodologies in contexts where team members are geographically distributed, and to highlight the need to create tools able to support agile processes even in distributed contexts. In [2] authors propose practices and values of DXP, a distributed version of eXtreme Programming (XP). DXP examines XP practices involved when team members of a software project are geographically distributed, giving importance to the communication. DXP assumes as available certain important conditions enabling a reliable communication among distributed team members; for instance the *application sharing* imposes synchronous communication among team members. In fact, in [3] the synchronous communication is considered as a way to improve the work process of distributed teams. DXP also proposes some challenges related to the communication, highlighting benefits of web technologies in terms of low costs and close involvement of team members. In [4] the importance of having a close communication within the team and tools supporting specific Agile practices is highlighted. Agile methodologies emphasize the direct communication between customers and developers, so that the percentage of information loss, due to the lack of long communication chains within the team, is minimal. Consequently distributed Agile processes emphasize the importance of close communication and collaboration *realizing lightweight techniques for a reliable communication and distributed collaboration.*

Literature presents several works dealing with distributed versions of Agile development processes (i.e. eXtreme Programming [1]). In [6] several patterns supporting a distributed eXtreme Programming are proposed. Two of these patterns (*virtual shared location* and *multiple communication modes*) are particularly important for the communication issue. The *virtual shared location* pattern deals with the need to use collaboration software in order to asynchronously post persistent information and ideas shared among distributed team members. Nevertheless, the virtual shared location pattern does not deal with the real-time communication, on the contrary with our experience where a real-time collaboration was necessary in order to realize reliable synchronous communications between team members and customers for user stories estimation. The multiple communication modes pattern suggests making available as many communication channels as possible, in order to replace, in the best way possible, the face-to-face communication and to maintain tacit knowledge, trust and shared understanding among remote team members.

In this paper we propose RTDWD (Real-time Distributed Wideband-Delphi), a fully web based tool to effectively perform Wideband-Delphi virtual meetings between several team members and the customer. As both customers and developers can participate to an estimation virtual meeting, their direct interaction minimizes the usual information loss in plan-based development processes, in fact they collaborate in real-time in a work context (a synchronous virtual shared location related to the user stories estimation practice is identified and common useful information is shared by each user) avoiding general misunderstanding. RTDWD realizes also asynchronous collaborations because the results of each virtual meeting are stored in a database. RTDWD features to support the use of

mobile devices (such as Pocket PCs and Smartphones) to participate to virtual meetings, very useful in contexts where the use of desktop or laptop computer is not possible. The deployment environment of RTDWD is CONDIVISA, an architecture providing knowledge sharing among several nodes forming a network. The remaining part of the paper is structured as follows: section 2 presents related works in this area, section 3 describes the context of the user stories estimation practice in Agile requirements management, section 4 introduces the deployment environment of RTDWD, section 5 describes RTDWD and its usefulness to perform real-time collaborations for the user stories estimation, section 6 presents the validation of RTDWD within Engisud and a comparison with other tools, finally section 7 traces some conclusions and future works.

2 Related works

The literature presents several works concerning the distributed collaboration in software engineering context. CAISE [5] is a valid architecture for the rapid development of CSE (Real-time Collaborative Software Engineering) tools but it does not refer to a web based collaborative systems.
The existing web based tools which support distributed Agile processes include XPlanner [7], VersionOne [8], MILOS [3] and MASE [4]. XPlanner is a project planning and tracking tool for eXtreme Programming (XP) teams which provides asynchronous communications among team members. It is a good management tool but it does not deal with the real-time collaboration among distributed team members and provides the customer off-line documentation only. VersionOne is an Agile software management tool supporting multiple Agile processes. MILOS and MASE are collaboration and knowledge sharing tools for Agile teams supporting both asynchronous and synchronous (real-time) collaborations. Precisely, MILOS and MASE realizes the real-time collaboration through their integration with Microsoft NetMeeting. Although the usefulness of Microsoft NetMeeting for synchronous collaboration, it has been a heavy-solution for us, due to the necessity of installing stand-alone applications, its operative system dependence, and its lack of supporting specific Agile practices. Since we have used user stories for the requirements elicitation phases, we needed a lightweight way to perform Wideband-Delphi virtual meetings for their estimation. Because of the above mentioned motivations, cited tools have not been the best solution due to their lack of supporting specific Agile practices where the real-time collaboration among team members and the customer is necessary.

3 The estimation phase in Agile software development

Agile methodologies, because of their emphasis on customer satisfaction through continuous delivery of valuable software, suggest to adopt the user story practice in order to gather functional requirements from the customer's point of view [1]. Moreover, Agile methodologies assume that the requirement elicitation and analysis does not take a single time window on the whole development

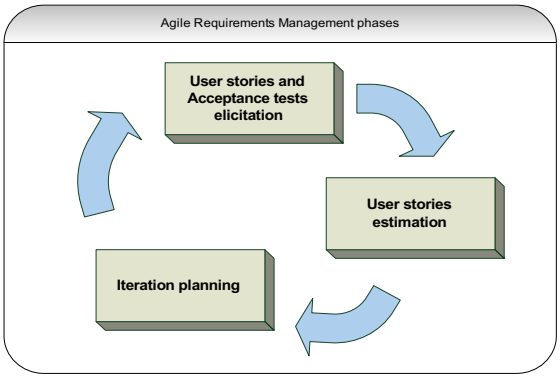


Fig. 1. Agile requirements management phases.

process, but their iterative nature requires periodic meetings with the customer in order to perform an incremental gathering of functional requirements and to always be ready to software requirement changes. This point is of great interest for IT companies because it minimizes several misunderstandings caused by inconstant communications. From this point of view, we share our experience with authors of [9].

When user stories and acceptance tests are gathered, the former have to be estimated from developers through meetings of user stories estimation. Estimates of user stories are necessary in order to realize reliable iteration plannings. The process is shown in figure 1. User stories estimation meetings are easily performable when involved team members (i.e. developers and project managers) and customers are co-localized. The co-localization is very important for these meetings because a continuous interaction related to technical and management issues have to be tackled. Nevertheless, when the team members are geographically distributed, problems concerning the lack of direct and synchronous communication precludes plans, several Agile principles (i.e. Customer On-Site) and values (i.e. Communication).

We use the Wideband-Delphi method to estimate a set of customer user stories. This method, as described in [10], is widely used because it represents a good chance for a close exchange of ideas between developers and the customer.

Agile methodologies, for user stories estimation, use the *ideal time* concept referring to a full devotion of a programmer to the code development during working hours. The unit of effort is the *story point*. The development team and project managers have to agree on the story point meaning taking into consideration a common unit of measure. For instance, an *ideal day* for an archetypal senior programmer defined specifically for the project, or an *ideal week* can represent a single story point. We preferred the former because a senior programmer has a large experience, she/he can debate the estimates with an high level of sureness and, therefore, she/he avoids problems to add calendar days from two or more developers with different levels of proficiency. In our case, the senior programmer

is a project manager.
It is a good practice to provide ranges of estimation through two estimates covering different levels of proficiency for each user story. We consider 50% and 90% levels of confidence in respect to the total confidence estimate.

3.1 The Wideband-Delphi method for user stories estimation

The Wideband-Delphi method, as described in [10], is an iterative and collaborative process that a team uses to estimate the effort that will be spent to realize a set of customer user stories. Generally, actors involved in a Wideband-Delphi meeting are developers, project managers and customers. Few days before the beginning of the meeting, project managers provide, to involved developers, user stories which will be estimated so that they can think about them. Therefore, estimators can be both developers and project managers. The participation of the customer is very useful because she/he can directly interact with developers and share opinions concerning user stories. In a Wideband-Delphi meeting a moderator (generally a project manager) manages the order of user stories to be estimated based on their coherence. When the meeting starts, each estimator read the user story and, in private, provides her/his 50% and 90% levels of confidence estimates in story points (iteration 1). When all estimators are ready, they publish their estimates. Initially, the estimates will be divergent, the estimators will therefore debate on the reasons of their estimates. It represents a good chance for knowledge sharing among several team members and to clear up misunderstandings, involving also the customer. After the group has discussed the story the moderator asks everyone to estimate again, keeping old estimates (iteration 2). The estimators erase their old estimates and write new ones on their cards. When all estimators are ready, the estimates are published again. In many cases the estimates will already converge by the second round, otherwise high and low estimators will explain the thinking behind their estimates. Iterations go on until estimate will converge, obtaining the final 50% and 90% levels of confidence estimates. The process is repeated for the following user story until all user stories for the iteration are estimated.
Therefore, the Wideband-Delphi method requires a close communication among all involved actors. If the development team is geographically distributed, a Wideband-Delphi meeting is very difficult to realize. For this reason, a specific tool to perform synchronous communications in a *real-time virtual shared location* for user stories estimation is needed. Tools cited in section 1 do not realize this specific virtual shared location. These reasons have provided us with motivations to create RTDWD.

4 CONDIVISA: the deployment environment of RTDWD

In this section we show CONDIVISA, a deployment environment of real-time collaborative web applications (i.e. RTDWD). CONDIVISA (Collaborative Object

shariNg on Distributed enVIronments based on SOAP and Ajax) is an architecture enabling the sharing of information, represented by XML documents, among several agents. In the CONDIVISA architecture two main agents are identified:

1. *CONDIVISA node*: it publishes several *services* represented by real-time collaborative web applications (i.e. RTDWD) and plays the role of *servent* with other CONDIVISA nodes. For this reason, a CONDIVISA node is both a web server for any web user participating to a virtual meeting and a component of a peer-to-peer network, called CONDIVISA network.
2. *Web user*: It is represented by a desktop/laptop computer or mobile device running a web browser (i.e. Microsoft Internet Explorer and Mozilla Firefox). It communicates with a CONDIVISA node deploying the real-time collaborative web application (i.e. RTDWD) which represents a shared repository for any web user connected.

CONDIVISA nodes exchange SOAP messages with attachments[5] containing XML documents related to the current state of a virtual meeting and several data of common interest for CONDIVISA nodes. For this reason, the communication among several CONDIVISA nodes is performed by web services interactions. Precisely, the notification pattern[6] is used so that an information update in a CONDIVISA node is immediately notified and shared with the remaining nodes without a specific request from closest nodes, minimizing the network traffic (the network traffic is generated when updates have to be sent and shared among CONDIVISA nodes).
The interaction between a web user and a CONDIVISA node is performed using the Ajax web technology[7], so that rich client applications are available to each web user, providing very similar functional features to stand-alone applications, but using http as application protocol and a simple web browser.
The CONDIVISA architecture, shown in figure 2, is a Service Oriented Architecture [12]. In a CONDIVISA network, the *interest* level of a node related some information available on the network identifies two data types:

- *Common interest data*: information shared by all CONDIVISA nodes.
- *Partial interest data*: information shared by a subset of all CONDIVISA nodes.

The CONDIVISA network topology is performed using the Distributed Network Knowledge (DNK) Heuristic [13]. DNK is a lightweight (low computational cost) Heuristic for a simple and robust knowledge distribution among several computers. At this level, the knowledge represents both information concerning a piece of network topology and information made available from each node. All information related to the network topology is represented by all *knowledge spikes*

[5] W3C SOAP protocol specification, http://www.w3.org/TR/soap/
[6] Technical works about Web Services Notification (WSN) have been provided by the OASIS technical committee, http://www.oasis-open.org/committees/wsn/
[7] W3C XMLHttpRequest object specification, http://www.w3.org/TR/XMLHttpRequest

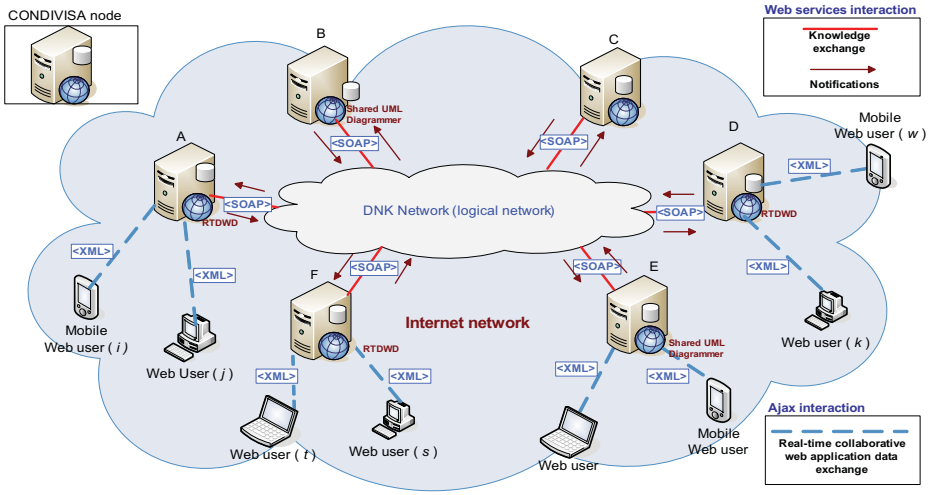


Fig. 2. CONDIVISA architecture according to SOA and Ajax interactions.

in the network. Each CONDIVISA node holds several knowledge spikes representing its closest nodes. An effective distribution of knowledge spikes, among CONDIVISA nodes, performed by the DNK Heuristic assures the connection of the network graph and an implicit full knowledge to each CONDIVISA node. Therefore, a CONDIVISA node directly communicates with its knowledge spikes through notifications of XML documents concerning both common interest data and partial interest data.

In figure 2 a scenario where nodes A-D-F and B-E have common interests respectively related to two real-time collaborative web applications (RTDWD and Shared UML Diagrammer). If the web user j performs an interaction with the shared Wideband-Delphi environment, several XML documents are notified from the CONDIVISA node A to D and F through the DNK network. Note that the mobile web user i will receive XML documents from the same CONDIVISA node, through Ajax interactions. The notification to nodes D and F involves the synchronization of the environment state visible to web users k, s ,t and mobile web user w. Communication types both among CONDIVISA nodes and web user-CONDIVISA node are based on XML document exchange using http, and two levels of knowledge sharing are identified: intra-CONDIVISA node and inter-CONDIVISA node sharing levels. The former deals with the sharing of XML documents related to deployed real-time collaborative web applications, the latter deals with the sharing of common interest data and partial interest data. An example of common interest data is the list of real-time collaborative web applications available in the CONDIVISA network. The list contains all knowledge fragment of CONDIVISA nodes but it is shared by all. Consequently, a web user communicating with a CONDIVISA node may have access to all public information.

Each real-time collaborative web application realizes a virtual shared location for all participant to a virtual meeting through the sharing of XML documents representing information related to the current environment (i.e. the shared web page) state and objects within the environment. For example, in RTDWD the environment is the shared web page containing several user stories and other data related to contents of each user story (description, notes, tasks, current estimates) and the current state of the Wideband-Delphi virtual meeting.

5 RTDWD: a tool for Distributed Wideband-Delphi virtual meetings

As described in section 3, the user stories estimation phase is performed for each iteration because it provides estimates to be considered during the iteration planning. Nevertheless, Agile methodologies assume as normal requirements changes, also related to user stories of the current iteration, involving the need to perform a Wideband-Delphi meeting to estimate again user stories. Therefore, requirements changes make unpredictable the number of estimation phases within a single iteration. When Agile team members are geographically distributed, the need to perform unpredictable lightweight virtual meetings for user stories estimation become a critical issue. The lightness of virtual meetings is very important because the fully web based solution allows each web user to always be ready to participate to the virtual meeting.
Real-time Distributed Wideband-Delphi (RTDWD) has been integrated with our Agile project management tool, called *eXtreme Project Manager* [11], which supports the Agile development process shown in section 6. RTDWD is a real-time collaborative web application realizing distributed Wideband-Delphi virtual meetings for user stories estimation, realizing a Real-time virtual shared location, and satisfying requirements related to the distributed communication [2]. Although RTDWD was tested for our Agile development process, it can easily be used in any distributed Agile development process, such as in [9], using the user stories estimation practice. RTDWD is not bound to the CONDIVISA architecture because it is an independent real-time collaborative web application performing XML documents sharing among several web users. XML documents describe the current state of the shared Wideband-Delphi virtual location so that a sharing, through Ajax interactions, assures a continuous synchronization of the environment state. Ajax refers to a set of techniques which rely on a layer added between browser and server. Instead of submitting a full page of data to the server, receiving a full page back, an application which uses Ajax techniques can send an individual field value and receive information to update only a single portion of the page. Ajax engine relies its power on JavaScript and CSS, but mainly on *Document Object Model (DOM)* and *XMLHttpRequest* object. DOM enables to modify the user interface on the fly, effectively redrawing parts of the page, while XMLHttpRequest object allows to perform asynchronous requests to the server and manipulate responses as a background activity.
RTDWD, due to its web based features, can be used from mobile devices (i.e.

Pocket PCs and Smartphones) with a web browser. This feature is very useful when some potential participant to the virtual meeting, such as the customer, cannot use a desktop or laptop computer. RTDWD realizes a virtual close communication between geographically distributed team members (i.e. developers and project managers) and the customer, in order to provide reliable estimates to customer user stories to be developed in the current iteration.
In order to perform Distributed Wideband-Delphi virtual meetings, RTDWD recognizes the following user types:

- *Moderator*: The moderator holds full ownership to move user stories within the environment, to confirm final estimates and initialize the estimation phase of a customer user story. These features represent the *external state* of user stories. The initialization of the estimation phase of a customer user story will create a shared dynamic table where estimators and generic users can interact in real time. Generally, the moderator does not estimate customer user stories, because she/he is represented by a project manager.
- *Estimator*: The estimator takes part to the iterative estimation phase of a user story in order to perform a real planning for the implementation of the selected customer user story. The estimator can see, in real time, the user story movement within the environment performed by a moderator and she/he can interact both with user stories and a dynamic table where she/he can insert her/his own min and max estimates (in story points) for the current iteration. Moreover, she/he can see estimates published by other estimators in real time. The estimators are, generally, the developers that will have the responsibility to realize customer user stories.
- *Generic user*: A generic user is, for instance, a customer. Customers can easily take part in the Wideband-Delphi session from any geographic location, in a non-invasive way. This user type cannot move on the shared user stories within the environment for order reasons. On the contrary she/he will see actions performed by other user types within the environment at real-time. If a participant associated with this user type wants to communicate with other participants, a chat room with all connected users is available.

RTDWD service is linked to a database containing all defined accounts for each Agile project in progress. As above mentioned, according to a specific account entered during the login phase, RTDWD recognizes the user type and projects where she/he works. Of course, any account can be assigned to multiple projects. The user type establishes the interaction policy that a single participant has on shared objects (e.g. user stories). The logged user can choose the room related to the project in which a Wideband-Delphi session is needed, as shown in figure 3. Figure 3 shows both the desktop/laptop computer and mobile device versions. Figure 4(a) and figure 4(b) show a shared Wideband-Delphi-session web page with customer user stories to be estimated, respectively for desktop/laptop computer and mobile device. Ajax capabilities allow to manage, on the client side, shared XML documents in order to update, “on the fly”, the local shared web page. According to the user type, it is possible to change or not the external state of the single user story. If user type is *moderator* she/he can drag and drop a

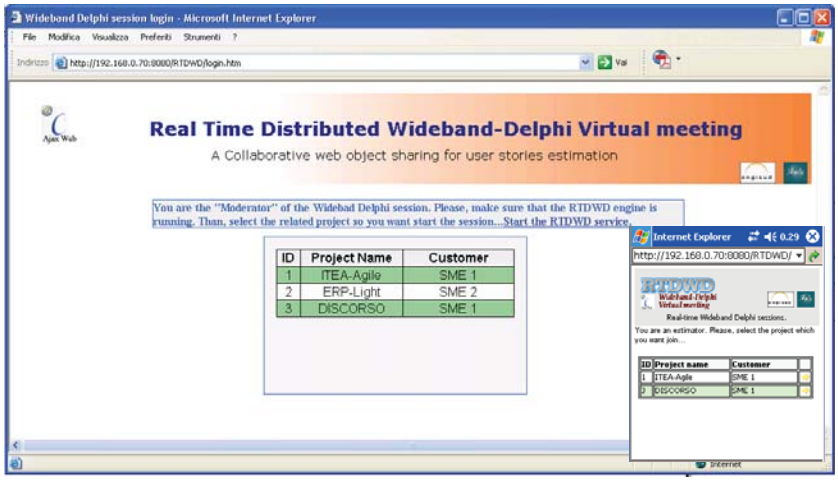
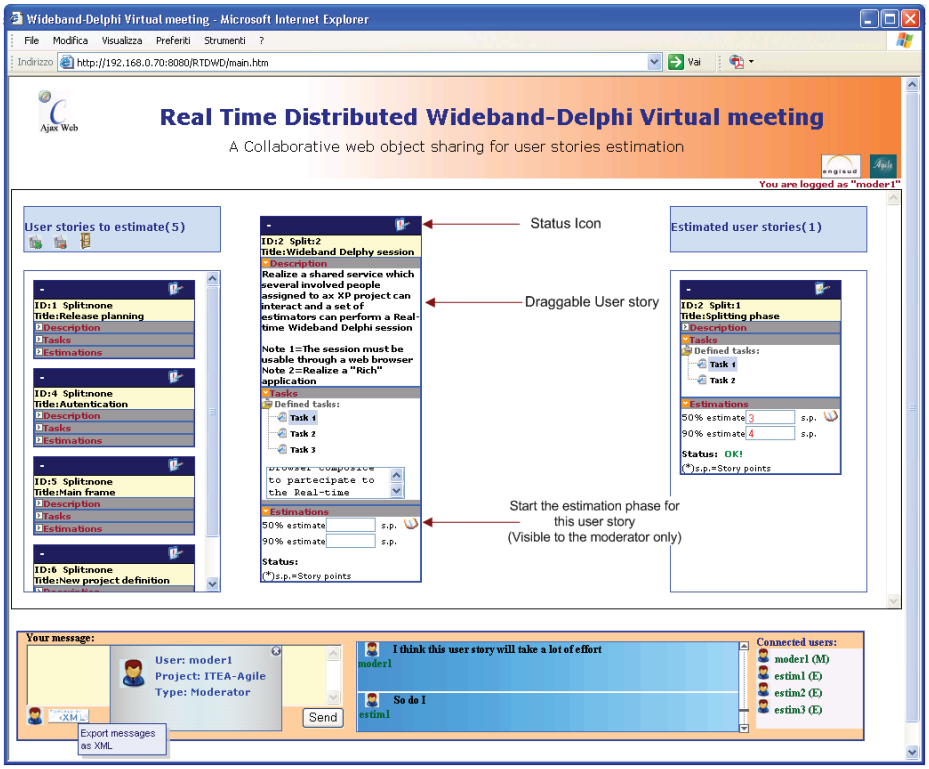
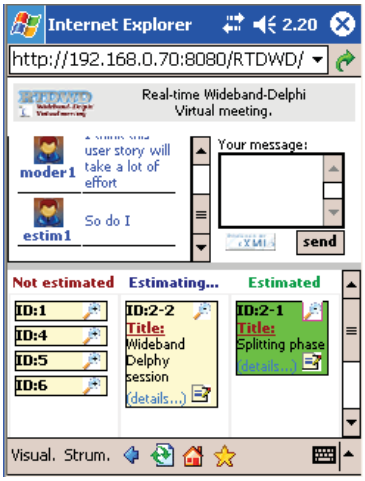


Fig. 3. The logged user can choose the room related to an assigned project.

user story wherever she/he wants within the environment, causing the real-time information propagation, through the notification of XML documents, to others participants which can instantaneously see the state change. Of course, multiple moderators can join a single Real-time Wideband-Delphi session, so they will have full ownership of external user stories states. All user types hold the *internal state* of user stories, so that each participant is owner of her/his local objects, and therefore can explode or collapse internal information sections such as user story description area, tasks area, estimates area (see figure 4(a)). Furthermore, each participant can export messages exchanged in XML format which can be used as documentation support, according to the Agile methodologies principles. RTDWD has been developed considering the situation in which Real-time Wideband-Delphi sessions related to different Agile projects run simultaneously. Each RTDWD node implements all software packages shown in figure 5. The *Security manager* package manages the user login phase through filtering and dispatching of ajax requests to session manager, communication manager and sharing manager packages. The *Session manager* realizes the multi-session capability allowing to simultaneously execute multiple wideband-delphi virtual meetings related to different Agile projects. The *Communication manager* package implements the data access policy based on the user type and communicates with the *Sharing manager* package that manages XML documents related to the shared wideband-delphi environment. The *Wideband-Delphi manager* is the core package of RTDWD because it implements the business logic needed for the user stories estimation phase and provides XML documents shared by web users. The data storage and interactions with the database is accomplished by the *Data persistence manager* package performing the persistence of user stories estimates .
RTDWD can be distributed on multiple CONDIVISA nodes, related to projects



(a)



(b)

Fig. 4. A Real-time Wideband-Delphi virtual meeting for a) a moderator and b) a mobile estimator.

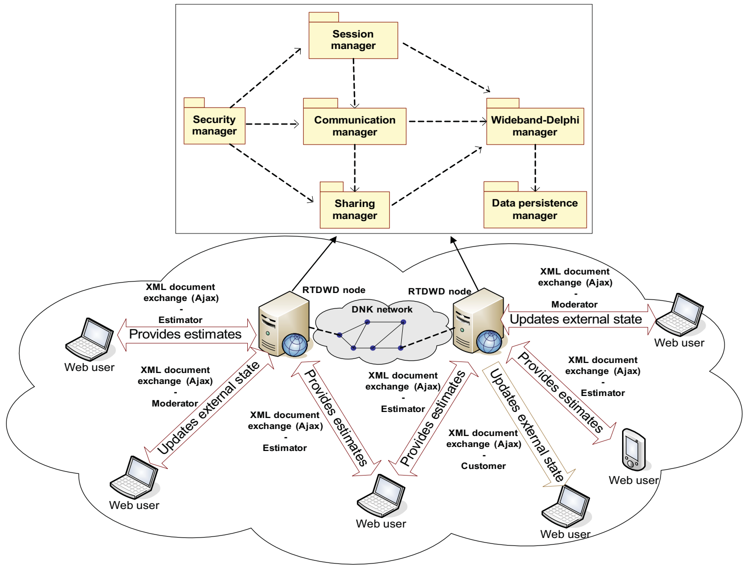


Fig. 5. RTDWD architecture, communication model and server side software packages.

of each company branch. Moreover, a single web user can join multiple Real-time Wideband-Delphi virtual meetings published on different CONDIVISA nodes. Figure 5 shows that exchanged documents are XML based and the direction is full duplex for moderator and estimator types and half-duplex (from server to client) for generic users. Server side packages are J2EE based and provide, for each session, the persistence and the management of XML documents exchange. As described in section 3.1, when a moderator begins an estimation phase of a user story, the wideband delphi process starts. For each iteration each estimator provides her/his estimates (50% and 90% levels of confidence estimates) debating her/his reasons with other estimators. If estimates conflict, the moderator will start the next iteration and each estimator provides estimates again, keeping in mind the debate of the previous iteration. The moderator coordinates the several iterations and confirms final estimates when estimators ideas converge. When a moderator starts the estimation phase of a user story using RTDWD, all participants will see a red title background of the user story. RTDWD supports iterative estimation phases putting a shared dynamic table at the participants' disposal. The dynamic table shows all estimation status of user stories and all estimates provided by estimators during the whole real-time Wideband-Delphi session. The dynamic table is built by the Wideband-Delphi Manager software component. For each user type an interaction level with the table exists:

– *Moderator*: The moderator sees the dynamic table containing all estimates submitted by estimators in previous and current iterations. The dynamic table is automatically updated when an estimator submits estimates. This

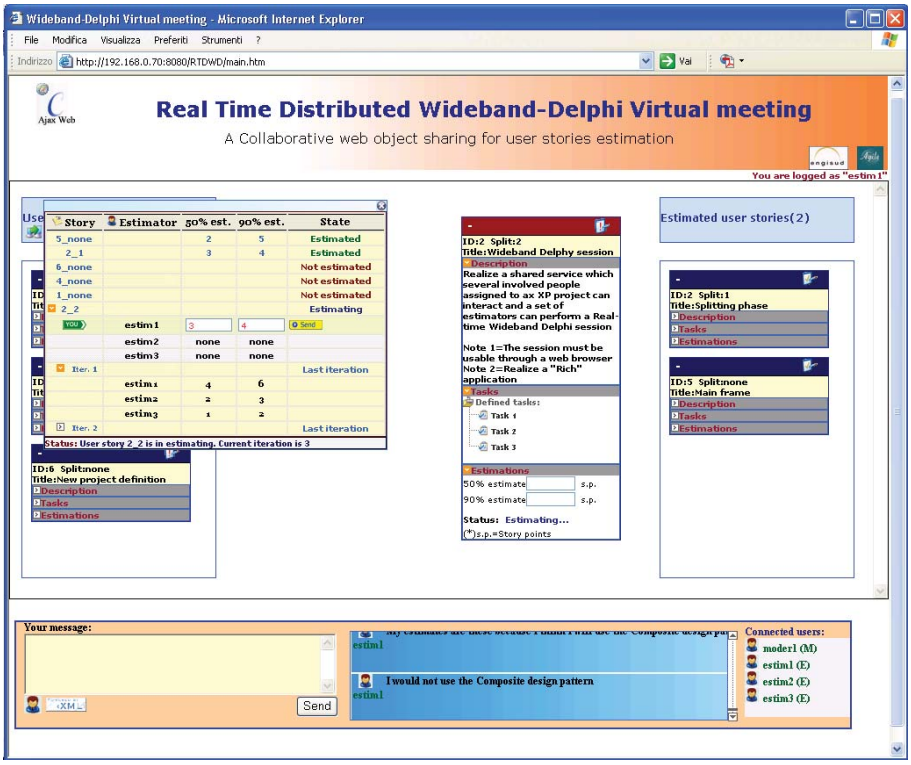


Fig. 6. Iterative estimation phase of a customer user story.

feature is provided through Ajax interactions. The dynamic table is non-invasive within the page because any user can drag, drop and close it whenever she/he wants.

- *Estimator*: The estimator provides her/his estimates for each iteration. She/he can interact with the dynamic table and insert her/his estimate only in the row related to her/him. The Wideband-Delphi manager package, shown in figure 5, recognizes connected users (interacting with the session manager) package, and builds the dynamic table.
- *Generic user*: The generic user (e.g. a customer) can see all submitted estimates by estimators. She/he cannot insert estimates in the table because she/he is not an estimator. The generic user can interact with estimators in order to participate to the estimation phase and debate the estimates.

Figure 6 and figure 7 show the iterative estimation phase of a customer user story respectively using a desktop/laptop computer and a mobile device.

Figure 6 shows the dynamic table of *estim1* estimator. She/he can see estimates of the previous iterations. Figure 7 refers to the participation of *estim3* estimator, using her/his mobile device. Figure 8 shows a fragment of the shared XML document containing current estimates related to the figure 6. When esti-

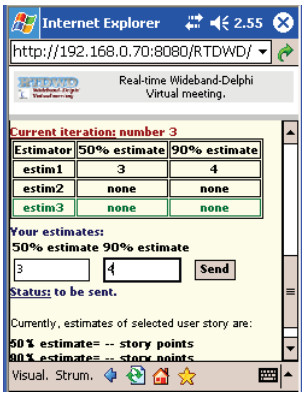


Fig. 7. Iterative estimation phase of a customer user story using a mobile device.

Fig. 8. Shared XML document containing current estimates.

mates for a single user story converge, the moderator will close the estimation phase of the selected user story and the dynamic table will be automatically updated with final estimates. This process goes on until all customer user stories of the iteration have been estimated. A logout of the moderator will cause the persistence of all estimates in the database of the CONDIVISA node. Furthermore RTDWD asynchronously stores documentation, in XML format, containing messages exchanged during the virtual meeting.

6 Tool validation

We have validated the RTDWD tool on the ITEA-AGILE project [14], during the Agile assessment within Engisud that usually outsources its projects to three geographically distributed development laboratories. The assessment has carried out an Agile development process, shown in figure 9. The iterative and collaborative nature of the Wideband-Delphi method makes difficult to provide reliable estimation sessions in a distributed environment because the use of asynchronous communication channels such as e-mails and phone calls involve frequent changes in iteration plannings that make short term plans unreliable. Since a reliable user stories estimation is a critical success factor during the Agile requirements management, a close communication and interaction support among several team members has to be assured.
A comparison of RTDWD with tools described in section 2 is shown in table 1. The comparison underlines the full web based nature of RTDWD for both synchronous and asynchronous communication. MILOS and MASE, although support the synchronous communication, depend on a stand-alone application.

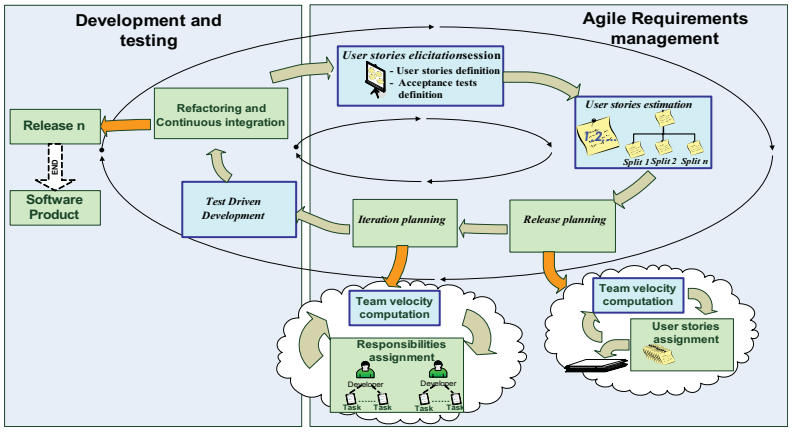


Fig. 9. Agile development process related to the Agile assessment in Engisud.

Tool	Communication type	Users interaction	Deployment	Supported practices	Synchronization support
CAISE tools	synch./asynch.	text/voice	stand-alone	generic	-
XPlanner	asynchronous	text	web-based	XP phases	-
VersionOne	asynchronous	text	web-based	several Agile methods	-
MILOS	synch./asynch.	text/voice	web-based	several Agile methods	stand-alone (MS NetMeeting)
MASE	synch./asynch.	text/voice	web-based	several Agile methods	stand-alone (MS NetMeeting)
RTDWD	synch./asynch.	text	web-based	user stories estimation	web-based (Ajax technology)

Table 1. Comparison of RTDWD with other collaborative tools

RTDWD allowed the distributed team and customer to realize a lightweight real-time Wideband-Delphi virtual shared location that enables a close communication between team members and the customer. During our trials RTDWD has shown to be a concrete approach for distributed Agile environment. RTDWD was a good way to considerably reduce travel costs and times needed to reach a collective sharing of ideas among several team members and customers. The proposed tool allowed us to minimize the use of tools for asynchronous communication to provide user stories estimates, such as e-mails. RTDWD provides a support to a critical phase of the Agile requirements management keeping simple for the customer to actively take part in the real-time Wideband-Delphi virtual location, using any web enabled device.
RTDWD can be adopted independently of its deployment environment due to its versatility, therefore it can support any tool for distributed Agile project management adding a reliable synchronous communication way for a specific Agile

practice. RTDWD well meets the DXP assumptions related to the communication.

7 Conclusions and future work

In this paper we have proposed RTDWD, a real-time collaborative web application realizing Wideband-Delphi virtual meetings for the Agile requirements management in a distributed environment. We have used RTDWD in our Agile development process providing a suitable communication mode among geographically distributed team members and customers for user stories estimation. Results of RTDWD motivates us to continue our research in real time collaborative environments for distributed Agile development processes. RTDWD will be extended to the face-to-face communication improving human interaction through vocal and visual supports. RTDWD is part of a real-time collaborative tool suite which aims to effectively support all critical phases of a distributed Agile development process related to synchronous and asynchronous communications. All results will enhance the CONDIVISA architecture in order to take advantage from lightweight real-time collaborative supports.

References

1. Kent Beck: Extreme programming explained. Addison-Wesley (2000).
2. Michael Kicher, Prashant Jain, Angelo Corsaro, David Levine: Distributed eXtreme Programming. XP International Conference (2001).
3. Frank Maurer: Supporting Distributed Extreme Programming. Proceedings of XP/Agile Universe Lecture Notes in Computer Science **2418** (2002).
4. Thomas Chau, Frank Maurer: Knowledge Sharing in Agile Software Teams. Logic versus Approximation (2004) 173–183.
5. Carl Cook, Neville Churcher: Constructing Real-Time Collaborative Software Engineering Tools Using CAISE, an Architecture for Supporting Tool Development. Twenty-Ninth Australasian Computer Science Conference **48** (2006) 267–276.
6. Keith Braithwaite, Tim Joyce: XP Expanded: Extreme Programming and Agile Procesees in Software Engineering. XP International Conference (2005).
7. XPlanner software, http://www.extremeplanner.org
8. VersionOne software, http://www.versionone.net/
9. Paul Prior, Frank Keenan: Requirements Management in a Distributed Agile Environment. Transaction on engineering, computing and technology **4**, (2005).
10. Mike Cohn: User Stories Applied - For Agile Software Development. Addison Wesley (2004).
11. Engisud: eXtreme Project Manager: a tool for Agile project management. Internal document for the ITEA-Agile project (2006).
12. Randall Perrey, Mark Lycett: Service-Oriented Architecture. Proceedings of the Symposium on Applications and the Internet Workshops (SAINT'03) (2003).
13. Engisud: DNK Heuristic for a lightweight knowledge distribution on P2P computer networks. Internal document for the ITEA-Agile project (2006).
14. ITEA-AGILE project web site. http://agile-itea.org
15. Engisud: An Agile development process for Engisud. Internal document for the ITEA-Agile project (2006).