Supporting the Collaborative Collection of User's Requirements

Carlos E. Acosta and Luis A. Guerrero

Department of Computer Science, Universidad de Chile
{caacosta, luguerre}@dcc.uchile.cl

1 Introduction

Most of the software systems development processes carry high risks for technological companies and service organizations. Some of these risks may ultimately produce delayed or failed projects with low quality software products. The Standish Group in 1999 presented the “Chaos Report”[1], which determined through a large scale scan on IT projects in northern hemisphere corporations that the software development industry continued to be marked by low quality products and services. According to this report, the main causes of failed projects during 1994 were the following: incomplete requirements (13,1%), little users participation (12,4%), lack of resources (10,6%), non-realistic expectations (9,9%), insufficient executive support (12.4%), changes in requirements and specifications (8,7%), lack of planning (8,1%), little administration of information technology (6,2%), technological incomprehension (4,3%), others (9,9%). It is important to note that a high percentage of errors are related to definition and administration of requirements and little user’s participation. Even though this study was performed some years ago, most of the current software projects are still dealing with the same risks. It is clear that any improvement in requirement engineering may substantially reduce the high risk that menace the quality of decision taken during the negotiations of the user’s final requirements.

Requirement engineering includes the following steps (see e.g. Nuseibeh and Easterbrook, 2000):

- Collect requirements: identify the stakeholders and obtain their requirements.
- Model and analyze the requirements: build abstract descriptions for a better interpretation.
- Communicate the requirements: transfer the knowledge among the stakeholders.
- Come to an agreement on the requirements: negotiate the requirements with the stakeholders.
- Evolve the requirements: manage changes not only in the requirements but also in the environment, or changes generated by the stakeholders.

[1] See: http://standishgroup.com/sample_research/PDFpages/chaos1999.pdf

Requirements engineering is a highly collaborative process that involves many stakeholders: the customer who pays for the project, the users who interacts with it, the domain expert, the software developers, etc. Much research has been carried out on the use of Group Support Systems (GSS) to support and improve the process of requirements engineering (see e.g. Boehm et al., 2001; Grunbacker and Briggs, 2001).

Although eliciting requirements does not need a group setting, it can benefit from it. It is more time efficient, it has a higher flexibility (depending on the group, the steps can be adapted along the way), the output is very easy to understand, a higher richness of the information is gained, and information that would have stayed hidden emerges (see e.g. Nunamaker et al., 2001).

During the past few years, we have observed an increased importance in the role of collaboration for organizational value creation with a significant improvement in the collaborative work through the use of GSS tools. However, the success of collaborative efforts is not assured. Because of the limitations of the human mind, many collaborative efforts consume time and resources without creating substantial value (see e.g. Shepherd et al., 1996; de Vreede et al., 2003). Collaboration Engineering is a newly emerging field that seeks to bring the value of facilitated interventions to people who do not have access to facilitation (see e.g., Briggs et al., 1998).

Collaboration Engineering is an approach to the design of re-usable collaboration processes and technologies meant to engender predictable and success among practitioners of recurring mission-critical collaborative tasks. A mission-critical task is one which creates substantial value, or which reduces the risk of loss of substantial value for organizational stakeholders.

In this paper we propose a collaborative process to capture and analyze requirements including both the software engineers and the final users. The main objective of this process is refining and completing a previously defined requirement list. This previously requirement list was probably defined in a first phase of a Software Development Process, like in the Inception phase in a RUP based development process, or can be an empty list when starting from zero.

In section 2 we provide a revision of some of the actual techniques for Requirements Engineering. In section 3 the principles of Collaborative Engineering are shown. Finally, in section 4 we show our designed process based on a Collaborative Engineering technique and a Group Storytelling activity.

2 Requirements Engineering

A Requirement Engineering process comprehends all the required activities for creating and maintaining the set of requirements of a system. There are four high-level activities in this process: a feasibility study of the system, a process for collection and analysis of requirements, a process for specification and documentation of requirements, and a validation process (see e.g. Sommerville, 2002). In this work we are focused on the design of a process to support the requirements collection and analysis. However, we need to revise some techniques proposed regarding the changing nature of the user's requirements.

These techniques can be summarized into several categories (see e.g., Nuseibeh and Easterbrook, 2000):

- Traditional techniques that include questionnaires, surveys, and interviews with the individual stakeholders.
- Group elicitation techniques aim to foster stakeholder agreement and buy-in, while exploiting team dynamics to elicit a richer understanding of needs. For example, brainstorming, focus groups, and RAD/JAD2 workshops.
- Prototyping can be used when there is a great deal of uncertainty or when early feedback from stakeholders is required (see e.g., Davis 1992).
- Model-driven techniques provide a specific model of the type of information to be gathered and use this model to drive the elicitation process.
- Cognitive techniques include a series of techniques originally developed for knowledge acquisition, such as thinking aloud and card sorting (see e.g., Shaw and Gaines, 1996).
- Contextual techniques are an alternative to both traditional and cognitive techniques. These include techniques such as participant observation and conversation analysis (see e.g., Goguen and Linde, 1993)

Each technique has its advantages and disadvantages and there is no a best technique. The combination of several techniques can be very fruitful: combining prototyping with cognitive techniques, for example, allows stakeholders to think aloud while actually experiencing the new system. When taking a closer look at the categories above, we can distinguish between group techniques and individual techniques. Prototyping, model-driven techniques, cognitive and contextual techniques can all be used in a group setting or in an individual setting. Although eliciting requirements does not need a group setting, it can benefit from it. It is more time efficient, it has a higher flexibility (depending on the group, the steps can be adapted along the way), the output is very easy to understand, a higher richness of the information is gained, and information that would have stayed hidden emerges.

Some of the disadvantages are long preparation time, a group setting is more difficult to manage, participants could be very dominating or silent on the other hand, recruiting participants takes a lot of time and effort, the analysis of the results afterwards takes a lot of time, and the participants are not really representative for a larger group (see e.g., Langford and McDonagh, 2003).

The combination of techniques to be used and steps to be taken to elicit requirements depends greatly on the situation at hand. The stakeholders involved are an important indicator for this. Requirements elicitation involves "capturing" the requirements of different stakeholders, such as customers, users, and developers. Users play a central role in the elicitation process (see e.g., Nuseibeh and Easterbrook, 2000).

User requirements define what should be developed. The requirements of the other stakeholders mainly define the constraints to what has to be developed.

It is often the case that users find it difficult to articulate their requirements or to communicate knowledge about requirements. It is very unlikely that the users can tell what they want in the future. Communicating knowledge is a key activity for today's specialized workforce. The direct and effective transfer of experiences, insights, and know-how among different experts and decision makers is a prerequisite for efficient decision making and coordinated, organizational action (Straub and Karahanna, 1998). Situations of such deliberate knowledge transfer through interpersonal communication or group conversations (Gratton and Goshal, 2002) can be found in various business constellations, as the following typical examples illustrate: for example, experts from various domains need to share their views and insights regarding a common goal in order to agree on a common rating of risks, requirements (Bronne and Ramesh, 2002).

In the Collaborative Requirements Engineering context we found a collaborative methodology for requirements negotiation proposed by (see e.g. Boehm, et al., 2001). This methodology called EasyWinWin is based on the WinWin negotiation model (see e.g. Boehm, 1998).

EasyWinWin combines the WinWin Spiral Model of Software Engineering with collaborative knowledge techniques and automation of a Group Support System (GSS). The WinWin negotiation model defines a set of artifacts guiding stakeholders through a negotiation process. Objectives of stakeholders are captured as win conditions. Conflicts among win conditions are recorded as issues. Options are proposed to reconcile Issues. Agreements are developed out of winconditions and out of options by taking into account the preceding decision process and rationale (Boehm, 1988; Boehm et al., 1998).

In the real world requirements emerge in a process of collaborative learning and negotiation. Each step in EasyWinWin challenges the stakeholders to use their tacit knowledge in a different way. The group dynamics created by the methodology helps the users to access tacit knowledge that might otherwise go untapped. This surpasses traditional ways of gathering requirements like interview-based techniques, which lack the ability to surface tacit knowledge. Finally, notice that this methodology prevents people from arguing aloud about system requirements until the win conditions have been listed and explained, and assessed, until issues have been raised, and options have been proposed.

EasyWinWin is focused on supporting the requirement negotiation process in a group context. In our proposal we are focused on supporting the externalization and transferring of the knowledge generated in the requirement elicitation process in order to produce a more detailed and less conflicting requirement list. In this context, the *tacit knowledge problem* appears. The tacit knowledge problem refers to the phenomenon that people may know how to do something, without being able to articulate how they do it. Tacit knowledge is connected with perceptions, ideas, and experience and it is difficult to transfer and acquired through practice and experience (see e.g., Nonaka 1994).

Software Engineers must deal with this tacit knowledge problem when it concerns a proponed software system. Experience shows that simply asking stakeholders what they want often works poorly. They cannot accurately describe their needs. If tacit stakeholder knowledge remains hidden the following problems can occur:

- Incomplete requirements because "obvious" ideas are not captured.
- Reduced ability to identify conflicts because not all project-relevant knowledge is explicitly available.
- Conflicting interpretations due to terminology differences.
- Hidden stakeholder expectations and assumptions not available explicitly.
- The challenging issue is if elements of our tacit knowledge are inaccessible in principle (see e.g., Wittgenstein 1960), or just hard to verbalize, and remain either consciously or unconsciously suppressed (see e.g., Varela et al., 1992).

Next section describes the benefits for the organization and for the design of reusable process of using a Collaboration Engineering design approach.

3 Collaboration Engineering

Organizations exist to create value for their stakeholders that the stakeholders cannot create for themselves as individuals. Collaboration is a sine-qua-non to creation of value in organizations. Collaboration Engineering focuses on recurring rather than ad hoc processes for two reasons. First, the benefits of designing an effective ad hoc process accrue to the organization only once, while the benefits of designing a recurring process accrue each time the process is used. Second, participants in ad hoc processes have no need to learn the interventions they experience, because they are unlikely to repeat the process, whereas practitioners of recurring processes can learn to conduct its activities and even pass that skill on to others without learning to become general purpose facilitators.

Group Support Systems are designed to improve the efficiency and effectiveness of meetings by offering a variety of tools to assist the group in the structuring of activities, generating ideas, and improving group communications (see e.g., Nunamaker et al., 1991).

Previous studies on GSS have reported labor cost reductions averaging 50% and reductions of project calendar days averaging 90% (Grohowski et al., 1990; Post, 1992).

The success of GSS meetings is often attributed to specific GSS characteristics: anonymity, parallel input, and group memory. These characteristics also resolve many of the disadvantages of using groups mentioned in the previous section (see e.g., de Vreede et al., 1997).

- Anonymity: By being able to enter ideas and votes anonymously, silent or shy participants are more encouraged to participate, other group members cannot dominate. Ideas are, therefore, judged on their merit, not on the personality or position of the person that submitted it.
- Parallel input. By generating ideas and communicating those in parallel, participants get equal time, preventing production blocking, so that participants

can spend more time on generating new ideas. Also working in parallel allows groups to generate more ideas. It is as if all people in the meeting are talking at the same time.

- Group memory. During an electronic meeting, all ideas and votes are stored electronically. Hence, little time is needed to produce meeting minutes and previous meeting results are readily available in follow-up meetings. Moreover, the meeting record is untainted in nature and also describes the evolution of a group's position over time.

Although these factors are often reported as success factors, we can find some conflicting results when comparing the performance of GSS's in the literature. To counter this problem have been proposed another unit of analysis, labeled thinkLets as an approach to produce far more predictable and repeatable results (see e.g., Briggs et al., 2001). GSS is at too high a level of abstraction, while the thinkLets describe in detail how a certain activity can be realized. This has an added advantage that thinkLets can inform the design of sessions. ThinkLets can be used for collaboration engineering. Have been identified seven basic activities in a group process: divergence, convergence, organization, elaboration, abstraction, evaluation, and building consensus (see e.g., Briggs et al., 2003).

ThinkLets will be attached to these basic activities in order to create a successful repeatable process. ThinkLets must be defined at least in terms of the tool used, the configuration of this tool, and the facilitation script. The tool component describes the specific version of the specific hardware and software used. The configuration specifies how the hardware and software were configured. And the script describes the sequence of events and instructions given to the group.

We find in previous publications the design of a repeatable process based on the identification of thinkLets related to each of the activities mentioned previously as part of a group process (see e.g., den Hengst et. to., 2004).

The application of this alternative takes place in a context where the organization that must interpret and use the elements produced by the proposed repeatable process is a part of the same enterprise where the necessity originates. Due to the fact that most software programs are developed by teams of engineers who belong to companies providing computer services, an activity to obtain and analyse the requirements that receives a list of requirements with a general level of detail and that allows us to deliver a list of requirements with fewer conflicts and more specific requirements.

To achieve this objective we propose the application of a cooperative activity within the process, whose purpose consists in illustrating the knowledge of the members of a group by means of the telling of a story or events that re-create the actions of a user in a specific context. This kind of activity is called Group Storytelling.

4 Group Storytelling

Storytelling has been studied in many disciplines, e.g. linguistics, sociology, management, education and artificial intelligence. This technique is commonly used to elicit and communicate knowledge. Recently, group storytelling has been proposed within the community of Computer-Supported Cooperative Work (see e.g. Perret et al., 2004; Schäfer et al., 2004). Group storytelling is a collective activity of sense-building, with many individuals contributing with their recollections and interpretations about shared experiences.

The stories help to humanize the environment, and the narratives involve emotions, thus they provoke personal commitment and stimulate the externalization (see e.g., Lelic 2001). Besides, telling a story is also a way to explain things informally, because of the needs for contextual cues to underline, e.g. to explain how to ride a bike.

Methods, techniques and tools have been developed to capture, register and retrieve stories in many research fields such as education, knowledge management and artificial intelligence (Schank 1997; Stanton et al., 2004; Thomas et al., 2001).

In business, real stories are generally experienced by teams. Therefore, members of a team can contribute to create a story about a work performed or a situation experienced by them, jumping in with additions, questions, corrections, comments, protests. A negotiation process will generally take place. For instance, providing the context of a claim could help other members to accept it and change the collective story, or to reject the claim in an augmented way. Since each participant performed a role in the scenario, stories written by a team will probably contain more valuable details and everybody have the opportunity to present their viewpoint on what had happened.

Based on these conclusions, we claim that shared context of a task performed by a group can be elicited and represented through group storytelling because it helps to identify, represent and make explicit the contextual elements related to the events of the task told through the story in order to establish the right relationships among them.

In previous works we have applied Group Storytelling to re-create the implicit context in a shared story (see e.g., Acosta et al, 2004). The result of this activity has been the externalization of events or annotations (comments and discussions) related to the context in which the actions are carried out by each one of the group participants.

Some processes of software development like RUP or XP, use the settings or contexts of the users (RUP) and users' stories (XP) as input information to establish the requirements of future developments in further detail. Our hypothesis is that Group Storytelling within the context of each of these processes would help to make the context and the implicit internal conflicts of it explicit. These are difficulties that development processes face nowadays in the stages of acquisition and analysing requirements in the execution of the engineering requirement.

Next we will describe our proposal and the incorporation of Group Storytelling as a part of the actions that is carried out to obtain and analyse requirements in a process of Requirement Engineering designed by means of Collaborative Engineering paradigm.

5 Proposed process

Bearing in mind the principles of Collaborative Engineering's design (of Vreede and Briggs, 2005), any process that can be replicated by the same members involved must have the follow-up steps clearly identified to create the different patterns of interaction which will allow us to reach the aims of the process.

Our proposal is based in the context of the acquisition and the analysis of requirements as part of the development process of software carried out by a software manufacturing company as part of the application of Technologies of Information project by a client organization. As part of this process of development, beginning with a list of characteristics or requirements with a general level of detail, one of the different technologies to obtain and analyse requirements can be put into action.

Then, having as input information a previous list of requirements, a set of activities and steps to carry out can be identified to have an improved second list of requirements

Selecting and inviting the participants is the first step for analyzing user requirements (see Figure 1). The selection of participants for the requirements elicitation process is an important topic. The participants should be chosen carefully through purposive sampling (as opposed to random sampling for surveys) – by selecting participants belonging to specific user groups (see e.g., Morgan 1998).

The participants need to be reasonably knowledgeable about the topic and should be interested in talking about it. Ideally, the groups should not include too many different types of people (see e.g., Maguire 2003), whilst a certain amount of diversity may be useful to encourage contrasting opinions (see e.g., Bruseberg et al., 2003).

Participants need to be comfortable in talking to each other and should share a similar background to encourage a common understanding of more detailed insights. Preferably, requirements elicitation groups consist of participants who are not too familiar with each other: the more diverse the views that are represented, the more reliable or robust the results become.

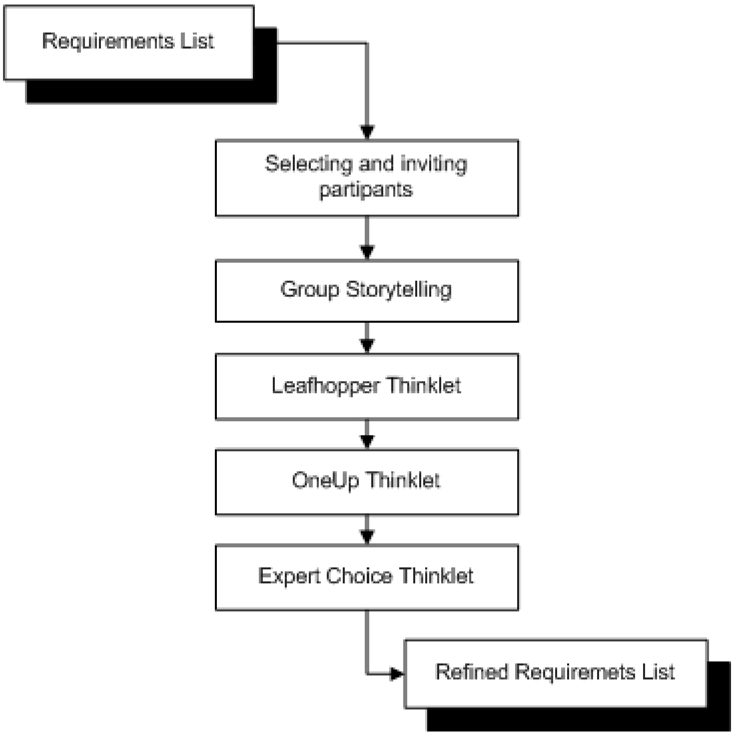

Figure 1: Process for improving requirement list

Group sessions that are not facilitated by GSS ideally have between five and twelve participants. GSS, however, have been found to effectively support large groups -more than 8 members (see e.g., Dennis et al., 1990).

Once the group is settled within the outline previously described, we proceed to the election of the general requirements of major priority according to the application complexity criteria, as it is developed inside the processes of software development established by RUP or XP. Accordingly, the input information to the following steps is conformed.

Then the Group Storytelling technique begins to support the implicit knowledge transference associated with the requirements of the stakeholders in the context of the organization. The different analysts and stakeholders that were selected to take part in the group session are involved in this activity. Once Group Storytelling session has finished and the result has been obtained, the refinement of the general requirements list comes, focusing the efforts to give solution to the concerns regarding the comments proposed by the members. To accomplish this, tasks or steps to be executed by the members are identified within the design of this flow of work in order to achieve a level of consensus that allows a clear and complete specification of requirements, with the maximum of requirements concerning the business domain that is being analyzed.

As shown in Figure 1, these steps consist of small necessary units of knowledge to create patterns of collaboration among the different members of the group session. As we have seen before, these steps form the ThinkLet.

The specific thinkLets uses in this document were adapted from two major publications on thinkLets (Briggs and de Vreede, 2003).

The following step in this process consists in the generation of solutions for the conflicts present in the commentaries given by the group members.

To accomplish this it is used the LeafHopper thinkLet (Vreede's Briggs and, 2003). Here the members that originate the comments are in charge of explaining the fundamental ideas that originated that commentary. Solutions are identified to the problems raised by means of the commentaries and a validation occurs.

In the following step, the approach of the group is focused on the solutions to unsolved problems. All likely solutions existing are displayed in a presentation to the complete group.

During this redefinition of solutions the participants will initiate the proposal of solutions including the valid viewpoints of the other participants, in order to create a solution with the venue of group. To carry out the set of actions previously described, it is used the thinkLet One Up (Vreede's Briggs and, 2003). In this thinkLet, the participants identify the ideas of high quality, while they explain why one idea is better than some of the previous proposals. The explanation of these ideas bears information of great value for the purge of the requirements exposed by the organization.

Finally with the incorporation of the thinkLet Expert Choice (Vreede's Briggs and, 2003), we look for a participant (Business Analyst), who is able to organize the set of ideas proposed in the previous step. Hereby we obtain information that is clear and consistent, which will allow the final improvement of the requirements and its specifications.

As we have mentioned previously, the Engineering Requirement in an activity that it originates in a group context and one sees highly benefited by the support of tools that promote the group collaboration. Bearing in mind these reasons a software tool was designed to generate propitious scenery for the proposed process.

6 Process Support Tool

This tool consists in the application incorporated on Microsoft platform .NET, which supports the previously described process of acquisition and analysis of requirements. One of the main goals of our proposal is the support of knowledge representation associated with the requirements of the stakeholders, which are generated through Group Storytelling by means of conceptual maps.

The construction of conceptual maps by means of a collaborative software tool can support the externalization of the knowledge when a technique of group storytelling is used to define and capture the requirements. In the designed tool we have proposed a user's interface with several views, which are described below.

Map of concepts and relations: here the concepts and relations are shown derived from the contextual knowledge obtained from the stakeholders through the proposed process. Besides we will find additional information of every concept, as the author, content, and relations to other concepts, annotations or related commentaries (See Figure 2).

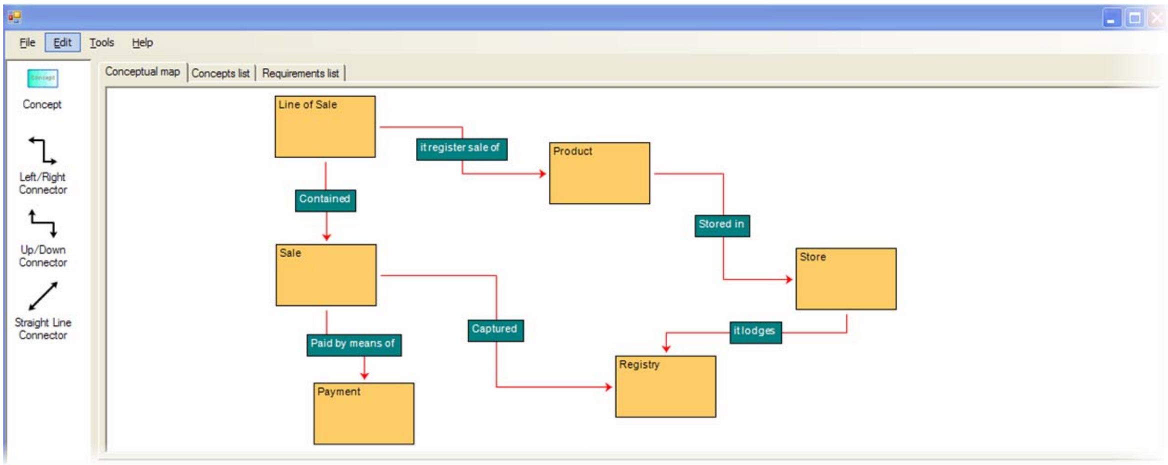

Figure 2: Conceptual Map View

List of concepts: shows only the concepts created in an instance of time with no relation to others; also, the additional information of every concept persists. (See Figure 3).

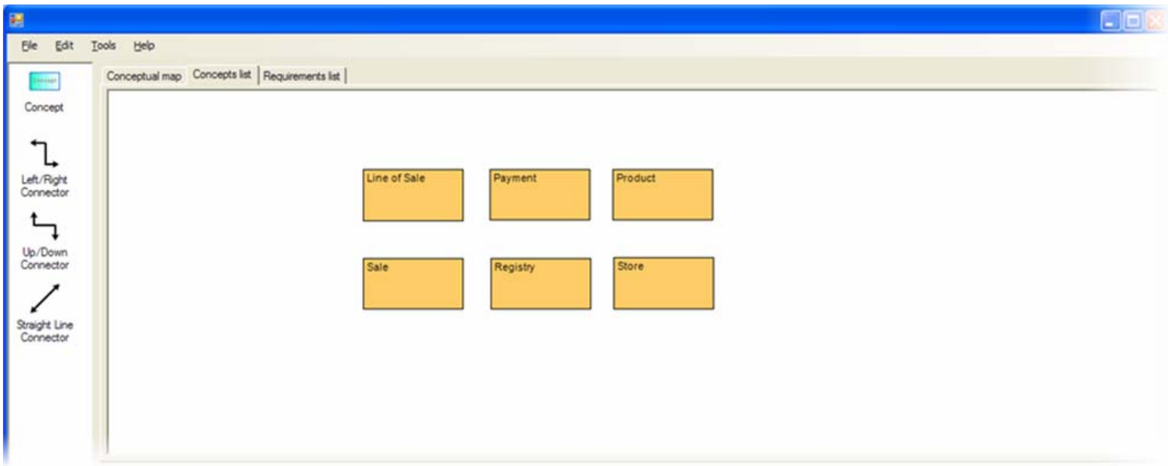

Figure 3: Concept list view

List of requirements: this view at the beginning of the process will show the general requirements presented by the organization, and will present the purified requirements after the execution of the proposed process.

7 Conclusions

Based on the Collaborative Engineering paradigm, we have designed a support collaborative process to obtain and analyse requirements in a software development process. This process implements Group Storytelling technology as an activity to support the transference of the implicit knowledge between the stakeholders and the analysts.

With the implementation of the previous activity and the identification of some thinkLets a repeatable process was designed to achieve the purge of requirements. To accomplish this goal, an initial list with little or no purge is taken as an entry, and a document of requirements is originated with a greater deal of details and a better understanding of the participants.

The stage of requirements analysis within the established processes of development known as RUP (Rational Unified Process) or XP (Extreme Programming) is the context in which the process execution designed as part of our proposal takes place. Even more, the cases of use in XP are called user's story, for that reason it makes sense to talk about group storytelling by the system analysts and the final users.

In this article, a support tool for the different activities that are executed through the Engineering requirement process has been proposed, which propitiates different patterns identified in the collaboration design of the proposal. This process is supported by a designed tool to implement the technique of group storytelling with analysts and final users. This tool uses conceptual maps for the externalization and representation of the knowledge generated by the construction of user's stories.

Acknowledgments

The first author would like to thank the Global Works S.A for financial support.

References

Acosta, C. E., Collazos C. A., Guerrero L. A., Pino J. A., Neyem H. A., Motelet O. (2004): "StoryMapper: A Multimedia Tool to Externalize Knowledge", XXIV International Conference of the Chilean Computer Science Society (SCCC'04), sccc, pp. 133-140.

Boehm B. (1988): "A spiral model of software development and enhancement". *IEEE Computer*, 21(5), pp. 61–72.

Boehm B., A. Egyed, J. Kwan, D. Port, A. Shah, R. Madachy (1998): "Using the WinWin Spiral Model: A Case Study". *IEEE Computer*, 7:33–44.

Boehm, B., P. Grunbacker, R.O. Briggs (2001): "Developing Groupware for Requirement Negotiation: Lessons Learned". *IEEE Software*, 18(3), pp. 46-55.

Briggs, R.O., M. Adkins, D. Mittleman, J. Kruse, S. Miller, J.A. Nunamaker (1998): "Technology Transition Model derived from field investigation of GSS: Use aboard the U.S.S. CORONADO", *Journal of MIS*, 15 (3), pp. 151-195.

Briggs, R.O., G.J. de Vreede, J.F. Nunamaker, and D. Tobey (2001): "ThinkLets: Achieving Predictable, Repeatable Patterns of Group Interaction with Group Support Systems (GSS)". Proc. of 34th Hawaii Int. Conf. on System Sciences, IEEE CS Press.

Briggs R.O., G.J. de Vreede. (2003): “ThinkLets: Building Blocks for Concerted Collaboration.” Second Edition,GroupSystems.com.

Briggs, R.O., G.J. de Vreede, and J.F. Nunamaker (2003): "Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems". *Journal of Management Information Systems*, 19(4), pp. 31-64.

Bronne, G.J. Ramesh, V. (2002): “Improving information requirements determination: a cognitive perspective”; Information & Management, 39, pp. 625-645.

Bruseberg, A. and D. McDonagh (2003): "Organizing and Conducting a Focus Group: The Logistics". In: Focus Groups, Supporting Effective Product Development, J. Langford and D. McDonagh (Eds.), Taylor & Francis, London, pp: 21-45.

Davis, A. (1992): "Operational Prototyping: A New Development Approach". *Software*, 9(5), pp. 70-78.

Dennis, A.R., A. Heminger, J. Nunamaker, and D. Vogel (1990): "Bringing automated support to large groups: The Burr-Brown Experience". *Information & Management*, 18(3), pp. 111-121.

den Hengst, M., E. van de Kar, J. Appelman (2004): "Designing Mobile Information Services: User Requirements Elicitation with GSS Design and Application of a Repeatable Process," hicss, p. 10018c, Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 1.

de Vreede, G.J., R. O. Briggs (2005) "Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks," hicss, p. 17c, Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 1

de Vreede, G.J., R. Davison, R.O. Briggs (2003): "How A Silver Bullet May Lose Its Shine", *Communications of the ACM*, 46(8), pp. 96-101.

de Vreede, G.J. and P. Muller (1997): "Why Some GSS Meetings Just Don't Work: Exploring Success Factors of Electronic Meetings". Proceedings of the 7th European Conference on Information Systems (ECIS), pp. 1266-1285.

Goguen, J. and C. Linde (1993): "Techniques for Requirements Elicitation". Proc. of 1st IEEE Int. Symposium on Requirements Engineering. San Diego, USA, pp. 152-164.

Gratton, L., Goshal, S. (2002): “Improving the Quality of Conversations”; Organizational Dynamics, 31(3), pp. 209-223.

Grohowski, R., C. McGoff, D. Vogel, B. Martz, and J.F. Nunamaker (1990): "Implementing Electronic Meeting Systems at IBM: Lessons Learned and Success Factors". *MIS Quarterly*, 14(4), pp. 327-345.

Grunbacker, P., R.O. Briggs (2001): "Surfacing Tacit Knowledge in Requirements Negotiation: Experiences using EasyWinWin". Proceedings of the 34th Hawaii International Conference on System Sciences, IEEE CS Press.

Langford, J. and D. McDonagh (2003): "Introduction on Focus Groups". In: *Focus Groups, Supporting Effective Product Development*. J. Langford and D. McDonagh (Eds.), Taylor & Francis, London, pp. 1-18.

Lelic, S. (2001): "Fuel Your Imagination - KM and the Art of Storytelling". Knowledge Management.

Maguire, M. (2003): "The Use of Focus Groups for User Requirement Analysis". In: Focus Groups, Supporting Effective Product Development. J. Langford and D. McDonagh (Eds.), Taylor & Francis, London, pp. 73-96.

Morgan, D.L. (1998): "*Planning Focus Groups*", *Sage Publication*, Thousand Oaks.

Nonaka I. (1994): "A dynamic theory of organizational knowledge creation", *Organization Science*, 5, pp. 14-37.

Nunamaker J.F., A.R. Dennis, J.S. Valacich, D.R. Vogel, J.F. George (1991): "Electronic Meeting Systems to Support Group Work". *Communications of the ACM*, 34(7).

Nunamaker, J.F., R.O. Briggs, G.J. de Vreede (2001): "From Information Technology to Value Creation Technology". In: Dickson, G.W., DeSanctis, G. (Eds). *Information Technology and the Future Enterprise*, New York: Prentice-Hall.

Nuseibeh, B., S. Easterbrook (2000): "Requirements Engineering: A Roadmap". Proceedings of International Conference on Software Engineering (ICSE'00), ACM Press, Limerick, Ireland.

Perret, R., M.R.S. Borges, F.M. Santoro (2004): "Applying Group Storytelling in Knowledge Management". Proceedings of 10th International Workshop on Groupware, (CRIWG'04), LNCS 3198, Springer Verlag.

Post, B.Q. (1992): "Building the Business Case for Group Support Technology". Proc. of the Hawaiian International Conference on System Sciences, IEEE CS Press.
Schäfer, L., C. Valle, W. Prinz (2004): "Group Storytelling for Team Awareness and Entertainment". Proceeding of ACM NordCHI, Tampere, Finland.
Schank, R. (1997): "*Virtual Learning: A Revolutionary Approach to Building a Highly Skilled Workforce*". *McGraw-Hill*, USA.
Shaw, M., and B. Gaines (1996): "Requirements Acquisition". *Software Engineering Journal*, 11(3), pp. 149-165.
Shepherd, M.M., R.O. Briggs, B.A. Reinig, J. Yen, J.F. Nunamaker (1996): "Social Comparison To Improve Electronic Brainstorming: Beyond Anonymity". *Journal of MIS*, 12(3), pp. 155-170.
Sommerville, I., (2002): *Ingeniería de Software*, Peason Education, Mexico, MX.
Stanton, D., V. Bayon, H. Neale, H. Ghali, S. Benford, S. Cobb, R. Ingram, C. O'Malley, J. Wilson, T. Pridmore (2001): "Classroom Collaboration in the Design of Tangible Interfaces for Storytelling". Computer-Human Interface.
Straub, D., Karahanna, E. (1998): "Knowledge Worker Communications and Recipient Availability: Toward a Task Closure Explanation of Media Choice"; Organization Science, Mar/Apr, 9(2), pp. 160-176.
Thomas, J.C., W.A. Kellogg, and T. Erickson (2001): "The Knowledge Management Puzzle: Human and Social Factors in Knowledge Management". *IBM Systems Journal*, 40(4).
Varela, F., E. Thompson, and E. Rosch (1992): "*The Embodied Mind: Cognitive Science and Human Experience*". *MIT Press*: Cambridge.
Wittgenstein, L. (1960): "*Schriften*". *Suhrkamp*: Frankfurt.