Awareness meets requirements management: awareness needs in global software development

Daniela Damian, James Chisan, Polly Allen

Dept. of Computer Science
University of Victoria, BC
PO Box 3055, BC V8W 3P6, Canada
{danielad,chisan,allenp}@uvic.ca

Brian Corrie

New Media Innovation Centre
600-515 West Hastings Street
Vancouver, B.C., Canada, V6B 5K3
Brian.Corrie@newmic.com

Abstract

There has been growing attention to awareness issues in group collaborative processes. In this paper we address workspace awareness in requirements management processes in global software development endeavors.

When working on a software project, developers, system analysts, testers, and managers make use of information about current sets of requirements, design artifacts and relationships to customer requirements or test cases, as well as roles and responsibilities assigned to particular work artifacts. Co-located teams benefit from social mechanisms and processes that naturally facilitate the work practice and diminish the perceived need for explicit workspace awareness support. However, limited access to informal communication in geographically distributed teams make problems caused by reduced workspace awareness more acute.

We propose a set of features for awareness support in geographically distributed requirements management activities and outline our first step in researching such awareness issues. In finding usable and sensible solutions to this problem, we may be creating solutions long overdue in requirements engineering in general.

Introduction

In this paper we address the issue of awareness in requirements management processes of global software development (GSD) teams. We believe that awareness needs are inherent in all collaborative processes of software development but particularly critical in requirements management activities carried throughout the software development life-cycle. When working on a software project, developers, system analysts, testers, and managers make use of information about sets of outstanding requirements, design artifacts and relationships to customer requirements, test cases and traceability links to system requirements, and roles and responsibilities assigned to work on particular work artifacts. We will refer to the availability of this information as workspace awareness.

Co-located teams benefit from social mechanisms and processes that naturally facilitate the work practice and diminish the perceived need for explicit workspace awareness support. However, access to informal communication in geographically distributed teams is significantly limited. Therefore, problems caused by reduced workspace awareness are more acute.

After discussing the need to address awareness in requirements management in GSD, this paper proposes a preliminary list of features of awareness support in requirements management, and discusses our immediate practical approach to research such awareness support in global software projects.

The need to address awareness needs in requirements management in GSD

The issue of awareness in requirements management in software development requires more adequate research treatment. Requirements as expressions of interests, goals and needs in software development reside with the stakeholders, who are rarely all physically co-located. Requirements are thus naturally distributed and we believe that the need for workspace awareness in requirements engineering has always existed. Even in ideal circumstances, it is extremely unlikely that all stakeholders of a software project would share the same workspace. Consider, for example, a classic three-stakeholder software project consisting of customer, user and developer. If these stakeholders inhabited the same physical space, they would all have an intimate appreciation of the day-to-day dilemmas they each face. In reality these roles are separated organizationally, politically and especially, physically. Knowing the source or rationale of a particular requirement, who is working on implementing a particular requirement, or who has intimate knowledge about the rationale for a test scenario, is often not a trivial task. Although the requirements process might explicitly require the dissemination of such information during the

development process, this requirements is, unfortunately, rarely in place.

Research into the tasks and communication patterns that occur in traditional large software projects (Curtis, Krasner and Iscoe, 1988) finds that in co-located environments information about design problems is often propagated informally, through people meeting in the hallway, chatting over coffee, during design or code review meetings. Although companies establish formal processes for making and reviewing decisions about the design of large systems, these structures are often ineffective for communicating design problems that arise in sections of the organization that are not part of the formal process. Rather, informal personal contacts are frequently the most effective way to communicate messages across organizational boundaries.

Awareness in collaborative activities

Awareness simply refers to knowledge one has of the environment in which one finds oneself, essentially, *knowing what is going on* (Endsley, 1995). Awareness is thought to have two basic characteristics. First, awareness constitutes accurate knowledge about the state of a dynamic, changing environment. Second, through normal interactions with the environment one is able to maintain one's awareness as a side effect of that interaction (Gutwin and Greenberg, 2001). This suggests awareness is naturally tacit, and that the means for acquiring this knowledge must be low-cost or even no cost for the recipient. Awareness serves to provide understanding of what is happening, who is affecting change, and where it is happening (Kobylinsky, Creighton, Dutoit and Bruegge, 2002). While there has been much attention paid to addressing general awareness in the physical environment, there is significant opportunity to support awareness of the workspace environment. In software development, this environment consists primarily of documents and code. Capitalizing on this opportunity requires comprehending what aspects of this environment may be of some value to stakeholders, how to capture those aspects and how to deliver awareness.

A number of awareness issues have been identified with respect to requirements engineering, in particular document change and contact identification (Herbsleb, Mockus, Finholt and Grinter, 2000). It has been found that prescribed apparatus to propagate information updates (typically formal documentation) is untimely, if not ineffective. Furthermore, coordination and conflict resolution depends on knowing whom to contact about what, a task commonly impeded by the unavailability of such information. Progress often suffers from 'organizational amnesia' where issues that have already been discussed and seemingly resolved are rehashed, reflecting limited collective memory (Catledge and Potts, 1996). Attempts have been made to address these issues with tools such as instant messaging, shared calendars, and web boards. It is our belief that reliance on informal communication still prevails in spite of these attempts.

GSD removes the informal interactions that normally promote awareness, so their effects need to be replicated by some kind of technology. While awareness about requirements information could be maintained in organizational and project documents, they have been shown to be very poor communication media (Curtis *et al.*, 1988; Al-Rawas and Easterbrook, 1996). In particular, when requirements change, formal mechanisms such as documents do not react quickly enough and often news is typically propagated informally (Herbsleb *et al*, 2000)

A strong motivator for this research is an earlier focused investigation of requirements engineering challenges in globally distributed software development organizations (Damian and Zowghi, 2002). This work clearly emphasizes the need for mechanisms to support awareness of requirements and related artefacts in their management in GSD. In software projects where stakeholders worked on several continents, there was incomplete and unreliable information about incoming requests, or priorities and issues related to particular software requirements. Not only did decision makers have difficulties keeping up to date on the latest decisions made on outstanding or emergent issues, but they also had difficulties obtaining a snapshot of the current version of features being implemented and related information such as decisions about change request approvals. Similarly, software developers reported difficulties in contacting originators of requirements and related issues, for clarification or elaboration, and in determining the responsible stakeholders for a particular decision.

Although we are witnessing rapid advances in collaboration technologies that potentially provide support for software development in global settings, studies of global teams (e.g. (Damian and Zowghi, 2002), (Herbsleb *et al*, 2000)) identified that (1) collaborative technologies for global collaboration are used in an ad-hoc and inadequate manner and (2) we lack an understanding of appropriate technological capabilities necessary to overcome challenges due to geographical distribution.

The main barrier to using collaborative tools more effectively is the lack of understanding we have of the tasks being performed and the information that is pertinent to completing those tasks successfully. In requirements management it is necessary to understand not only the formal process in great depth, but also to understand the informal processes that are used to facilitate the formal process. Generic awareness tools that provide information on the physical environment, although useful, do not provide specific information that is relevant to the requirements management process. It is our belief that tools that provide awareness about the

artefacts involved in requirements management are required to deliver a successful distributed requirements management environment.

Our Research approach

In this paper we propose a set of features for awareness support in geographically distributed requirements management. We then discuss a research project in which we are attempting to study the features of such awareness support. We welcome feedback from discussions in the workshop. At the workshop we intend to report on our insights into refining these requirements from the experience of implementing a system to meet these requirements.

In discussing the features of an awareness system in requirements management, we consider the support it should provide to tasks team members perform during requirement management activities, particularly tasks which require requirements awareness information. Features are highlighted in bold font while data to be tracked by the system is underlined.

1. **Allow project stakeholders to seek existing information**

    In general, this is the simplest though potentially most frequent task that project team members perform during requirements management: finding information about requirements and the status of their implementation. The awareness system should allow team members to query or browse information about:

    - Requirements: This includes a description of the requirement itself, meta-information about the requirement (such as rationale, priority, stability level) as well as inter-dependencies with other requirements and software development artefacts (e.g. design, code or test). Another piece of information that often resides with the team members and is rarely well documented are the issues and decisions related to particular requirements. The ability to access this rich information about requirements contributes to alleviating problems of "organizational amnesia" (Catlege and Potts, 2000).
    - People: This includes the roles and responsibilities for each team member related to the implementation of particular requirement(s), including personal contact information. This contact information can be as simple as a phone number or an e-mail address, but in geographically distributed settings it may be important to include at a minimum, some indication of the team member's office location and time zone. More elaborately, it may even include physical awareness cues, giving system users indications of where team members can be found and how best to contact them.

    - Specific relationships between People and Requirements: This information would identify initiators, decision makers, analysts, estimators and those currently responsible for the realization of a requirement. These relationships can be stored by tracking the initiators of a requirement, issues (including who initiated the issue, the resolution status of the issue, and any decisions made because of the issue), meetings (including the date, time and location of the meeting, the stakeholders involved, the issues discussed and decisions made), and change requests (including who initiated the change request, the status of a decision regarding the change request, who was involved in that decision and the decision itself).

2. **Support stakeholders in decision making**

    Besides seeking information about requirements, for purely informative purposes, project members need the information outlined above to make decisions. Decisions with respect to the implementation of requirements are needed throughout the project lifetime. Project managers may try to decide how to assign resources to a project, whether it is feasible to implement a feature with the given resources, or whether the project plan is proceeding smoothly. Team members may try to decide if their understanding of the current requirements baseline is correct, whether their implementation of a feature fits with the rationale behind a requirement, or who to contact for more information regarding a previously made decision. All decisions need to be recorded so that they can feed back into the system; in this way, all team members benefit from the group's experience.

    The high-level task of decision making may involve any of the following subtasks:

    - **Assigning responsibilities and managing a project**
      At the beginning of a project, people or teams are made responsible for the implementation of each requirement or feature. If the information about responsibilities is captured, it is possible to track the particular states of requirements with respect to their progress. This enables project members to easily become aware of who is working on a particular requirement and query the current state of requirements. Similarly, the project manager can easily identify assigned people responsible for the requirements from the assigned people and/or dynamically delegate the implementation of particular requirements to available personnel.

    - **Gathering and managing estimation data**
      Decision-makers often need to involve other stakeholders during estimation gathering for the

purposes of negotiation and project management. This may be thought of as an iterative propagation of <u>requests-for-information</u>, <u>clarifications</u> and <u>estimations</u> between stakeholders. Tracking information about the interactions that occurred, the history of estimates as clarifications were exchanged, and the eventual estimates on which decisions were based aids in future decision-making. It also allows users to decide how the project is proceeding according to original, or amended, estimates.

- **Impact analysis**
  To support decision-making, users need to analyze the impacts of potential decisions. Again, awareness of the <u>relationships between requirements, and between requirements and other artefacts</u> of the software lifecycle, can help users understand the extent of changes being proposed. Further, the awareness of <u>relationships between requirements and personnel responsible for the implementation of requirements</u> is needed in the change notification process. Decisions made as a result of change requests need be made available to the affected stakeholders

3.Decision notification
For future decision-making activities, the <u>decision</u> must be documented. This documentation should include references to the information listed above under "Seeking existing information", as well as <u>rationale</u> for the final decision. .An awareness tool must provide a means to decide which stakeholders are affected. To this end, the <u>responsibilities</u> of each team member must be tracked as mentioned above. Finally, the decision must be made available to all stakeholders to support coordination.

Design and evaluation of awareness support in requirements management of global projects

Our current endeavors in researching such awareness support for requirements management is a design and development effort for a tool that meets the criteria outlined above. A first step is being taken in a graduate course in the department of Computer Science at the University of Victoria, between January and April 2003. This research effort will serve as a case study for requirements engineering, and indeed software system engineering, in a geographically distributed setting. Our goal in this study is to discover if the requirements outlined above are complete or if other needs arise during the development of the tool. It is our hope that the workshop will present the authors with the opportunity to discuss the case study and our experience of designing and using awareness tools for requirements management in geographically distributed software development.

In this course, the authors themselves design and develop awareness tools in a geographically distributed software development environment. The course provides the opportunity for a case study in which the features outlined above are considered as a starting point in the development. The activities of distributed requirements management are experienced during this development effort.

Group face-to-face meetings will be held at the University of Victoria once every four weeks. Except for these meetings, one participant will be physically separated from the University, working in Vancouver and participating in formal group meetings via audio or videoconference. During these formal meetings, documents and applications will be shared between locations using distributed application sharing software, and interactions will be facilitated and recorded using an electronic whiteboard. Microsoft's NetMeeting software was chosen due to its ubiquitous availability on all Windows platforms. Requirements will be managed using the Rational RequistePro requirement management tool. The two other participants, both working on the University campus, will be co-located for formal meetings, but will attempt to avoid informal communication regarding the project outside of the computer-supported environment. Informal communication between all three participants will be limited to electronic mail, instant-message and chat programs, voice-over-IP or telephone conversations, or videoconference tools. The group uses the Groove collaborative software (http://www.groove.net) to provide both formal collaboration capability (shared documents) as well as informal communication (instant messaging, threaded discussion groups, and physical awareness).

The tools used for this project are a selection of widely available collaboration technologies that have been coalesced into a suite of tools directed at accomplishing the requirements management task. The tools being used for this project were selected based on several requirements:

- They provided the collaboration capabilities required by the project
- They provided adequate quality (audio, video, etc.) for the distributed collaboration
- They were easily available to all collaborators

Discussion
We expect that our experience in developing the awareness tools using this computer-supported collaboration infrastructure will be invaluable. Our goal is to take a first step toward the development of an awareness tool that is consistent with the requirements described in this paper. By engaging in this exercise within a geographically distributed environment we expect to experience, first-hand, many of the same awareness problems that we wish to address. We hope

that both the development and this experience will equip us with improved understanding of the problem and indications for future direction of research and exploration.

We are aware that this research project represents only a first step in studying the issue of awareness in requirements management for global software teams. We acknowledge a number of potential limitations of the study:

- Due to the size of the group, the study is not fully representative of a global software development environment.
- Due to the fact that the study is offered as part of a course, the project environment is controlled.
- Due to the fact that the project only simulates global distribution, issues of cultural, organizational, and environmental factors are minimal.

Conclusion

Requirements management is gaining increased interest and appreciation as a difficult task in software development. Much of the information about project status and change management is typically propagated though informal communication in an organization. Perhaps these channels have always been inefficient; many system errors can still be traced to requirements misunderstandings, miscommunications or mismanagement. Only since the large-scale adoption of GSD, and the removal of these informal mechanisms, has the problem of lack of awareness been highlighted. In finding usable and sensible solutions to this problem, we may be creating solutions long overdue in requirements engineering in general.

References:

1. Al-Rawas, A. and Easterbrook, S. Communication problems in requirements engineering: a field study, Proc of the First Westminster Conference on Professional Awareness in Software Engineering, Londong, 1996, 47-60

2. Catledge, L. and Potts, C., "Collaboration During Conceptual Design," in Proc. 2nd Intl. Conf. on Requirements Engineering, (Colorado Springs, Colorado, USA), pp. 182--189, 1996

3. Curtis, B., Krasner, H. and Iscoe, N. A field study of the software design process for large systems, Communication of ACM, 31(11), 1988, 1268-1287

4. Damian, D. and Zowghi, D. The impact of stakeholders' geographical distribution on requirements engineering in a multi-site development organization, Proc. of the 10th IEEE Int'l Conference on Requirements Engineering (RE'02), Essen, Germany, 2002, 319-328

5. Endsley M.R. Toward a Theory of Situation Awareness in Dynamic Systems, Human Factors, 37(1), pp.65-84, 1995

6. Gutwin, C., and Greenberg, S. (In Press) The Importance of Awareness for Team Cognition in Distributed Collaboration. In E. Salas, S. M. Fiore and J. A. Cannon-Bowers (Editors) Team Cognition: Process and Performance at the Inter- and Intra-individual Level. APA Press. 2001

7. Herbsleb, J.D., Mockus, A., Finholt, T.A., & Grinter, R.E. Distance, Dependencies, and Delay in a Global Collaboration. Proceedings of CSCW 2000, Philadelphia, PA, Dec. 2-7, 2000.

8. Kobylinski, R., Creighton, O. Dutoit, A.H. and Bruegge, B. Building Awareness in Global Software Engineering Projects: Using Issues as Context, International Workshop on Global Software Development (co-located with ICSE '02), Orlando, FL, May 21, 2002.