Exploring the Emergence of Social Networks in Collaborative Software Development

Abhishek P.S, Madhuri Katabathuni, G. Sudhakar

Abstract : Collaborative software development has been around as it has been made possible with groupware software. In such scenario, it is important to have coordination and planning among the team members involved in collaborative work.In such development tasks are divided into work items while sharing and tracking the work done. Exploration of tagging for resolving communication concerns in collaborative software development environment is the problem to be addressed. The existing system explores tagging for resolving communication concerns in collaborative software development environment is the problem to be addressed where different kinds of tags are used by various stakeholders to categorize and organize the work items. The tags that will be generated are used to support finding of different tasks, the articulation work, and the information exchange.different type of mechanisms like Implicit, explicit mechanisms have come up to manage the tag vocabulary. The Software that is used in the current day environment of collaborative software development focuses on individual development but not the group work.Most of the software that are used by the developers that are used in their daily work are tailored towards the work performed by an individual but hardly supports team work. The aim of the proposed system is that to explore which social networks emerge in the software development between the author of the work items, owners of the work items and tag authors . This will increase our basic understanding of the team dynamics in the software development cycle and may ultimately result in a better collaborative software tool support for a productive outcome of the product being developed . Thus the social network can provide you an insight of how the communication patterns plays a major role in preventing build failures in a system .

Index Terms – Exploring , Social Networking , Collaboration , Software Development , Task Management , Work Items , Tag Management

1 Introduction

Collaborative software development has been around as it has been made possible with groupware software. In such scenario, it is important to have coordination and planning among the team members involved in collaborative work. In such development tasks are divided into work items while sharing and tracking the work done. Software is rarely developed by individuals and the success of the project is depends on the communication , co ordination within the teams.As a human being when we start writing a document, design software product,write code and test cases we are extremely slow and error prone. As a result we work in teams to complete the correct product in time.

Software engineering projects are naturally cooperative that requires many software engineers to work in co-ordination , their efforts to produce a large enterprise application. As a individual human being, when we start writing software requirements specifications, designing a software model, writing code for that model and creating test cases—we are extremely slow and error-prone. As a result, we should work together in a team to boost the performance and to complete the large projects in reasonable time. Software is rarely developed by individuals and the success of software projects largely depends on the effectiveness of communication and coordination within teams.

In order to understand the communication concerns that are involved in the development of the software product , there should be some communication environment that the development team (software development team) can communicate with different teams spread geographically for a smooth software development process . This project focuses on designing a small environment that supports a collaborative software development which is spread geographically .

With the movement of software development to a team based software development there is a need of articulation work that playes a major role in the development of a software product. They may be used to organize, manage and categorize software artifacts in general in an informal and collaborative way. The work over here lies in the benefits of social networking areas in social computing mechanism .Thus leads to an understanding of the drawbacks in the communication environment of a collaborative software development process.

2 Background & Related Work

2.1 Supporting Work Activities

The development of tools to support synchronous communications between non-collocated colleagues has received considerable attention in recent years. Much of work done was focused on increasing a sense of co-presence between interlocutors by supporting aspects of face-to-face conversations that go beyond mere words (e.g. gaze, postural shifts). Thus a design goal for many environments is the provision of as much media-richness as possible to support non-collocated communication. In this paper we bring to you the results from our most recent interviews.

Studying the use of a text-based virtual environment to support work collaborations. We describe environments like this , though lacking almost all the visual and auditory cues known to be important in face-to-face communication, has played a vital role in everyday communication. We provuide a set of characteristics that we feel are important to the success of this text-only tool and discuss issues emerging from its long-term use [1].

One clearly-stated belief is that interactions can be facilitated by structuring the virtual work space/environment to create a shared "landscape of work artifacts". Application based projects therefore provide support for the management of personal and shared workspaces and the sharing of work-related artifacts. A notion of shared spaces or "locales" is also used to structure interactions between interlocutors. A second belief is that for tightly coupled collaborations at least, the creation of a feeling of co-presence between non collocated collaborators is crucial if computer mediated communication is to be successful. With reference to such communication technologies, Lombard and Ditton state "an enhanced sense of presence is central to the use, and therefore the sense of usefulness and profitability of new technologies". Their discussion suggests that a greate sense of presence is correlated with "a feeling of non mediation", and this will result in a greater quality of interaction [1].

2.2 Collaborative Tagging

Marking content with descriptive terms, which are called keywords or tags, is a common way of organizing content for future navigation, filtering or search. Though organizing electronic content this way is not new, a collaborative form of this process, which has been given the name "tagging" by its proponents, is gaining popularity on the web. Document repositories or digital libraries often allow documents in their collections to be organized by assigned keywords. However, traditionally such categorizing or indexing is either performed by some authority, such as a librarian, or else as mentioned from the material provided by the authors of the documents (Rowley 1995). In contrast, collaborative tagging is the practice of allowing anyone – especially consumers – to freely attach keywords or tags to content. Collaborative tagging is most useful

When there is nobody in the "librarian" role or there is simply too much content for a single authority to classify; both of these traits are true of the web, where collaborative tagging has grown popular [2].

2.3 Breaking the Code

Software development is typically cooperative endeavor where a group of engineers need to work together to achieve a common and coordinated result. As a cooperative effort, it is difficult because of many interdependencies amongst the artifacts created during this process. This has lead engineers to create tools like configuration management tools, that seperates developers from the effects of each other's work. In doing so , these tools create a difference between private and public aspects of work of the developer. Technical support is widely provided to these aspects as well as for transitions between them. However, we provide the empirical material collected from a software development team that suggests that the transition from private to public work needs to be more carefully handled. Indeed, analysis of the material suggests that different formal and informal work practices are adopted by the developers to allow a delicate transition, where developers of the softeare are not largely affected by the emergent public work. Finally, we talk about how groupware tools might support this transition [3].

2.4 Distributed Software Problem Management

Software problems, or bugs, are errors or mistakes of commission or omission inadvertently introduced to software during the software development process. A recent study by the National Institute for Standards and Technology on software testing infrastructure estimates that $22-60 billion dollars are lost in the U.S. economy each year due to inadequate infrastructure for detecting and correcting software problems, and the subsequent release of bug-ridden software to users and consumers (p.ES-3).1 This cost estimate includes both the costs to developers to repair problems and the costs to user organizations to mitigate their effects. Software problems can also have significant social and human impacts [4].

2.5 Coordination in Large-Scale Software Development

Large-scale software development requires coordination within and between very large engineering teams which may be located in deferent buildings, on different company campuses, and in di_erent time zones. At Microsoft Corporation, we studied a 3-year-old, 300-person software application team based in Redmond, WA to learn how they coordinate work within ther intra-organization, physically distributed dependencies that are dependent on each other : a platform library team also in Redmond; a team three time zones away in Boston, MA; and also a team in Hyderabad, India. Thirty-one interviews have been conducted with 26 team members revealed that coordination was most impacted by issues of communication, capacity and cooperation. Distributed teams faced additional challenges due to time zone and cultural differences between the team members. We support all our findings with a survey conducted among 775 engineers across Microsoft who described their experiences managing coordination in their own software products. We reomend new processes and tools to improve coordination between teams [5].

Viewing the coordination between teams through this framework leads us to ask several research questions:

1. What kinds of behaviors are associated with being helpful or unhelpful to others?
2. How do individuals on a software team communicate to get work done?
3. How do software teams manage dependencies on a personal level?

To understand inter- and intra-team dependencies in large scale software development, we conducted a large interview-based study of a 300-person Microsoft software development group, out of them two of its teams are distributed globally.

We collected 31 hours of interviews from 26 engineers to learn with whom these individuals collaborated and which actions the individuals considered helpful or unhelpful for coordination. The interviews were audio recorded, transcribed, and coded. We corroborated our interview data with a follow-up survey of 775 Microsoft engineers [5]

2.6 Tagging Human Knowledge

Fundamental premise of tagging systems is that regular users can organize large collections for browsing and other tasks using uncontrolled vocabularies. Tagging approach to organizing a collection. We found out that tagging systems have three major large scale organizational features like consistency, quality, and completeness. In addition to testing these features, we provide results suggesting that users produce tags similar to the topics designed by experts, that tagging can effectively supplement tags in a tagging system, information integration may be possible across tagging systems. web sites are turning to the "tagging approach" rather than the "library approach" for organizing the content generated by their users. This is both by necessity and by choice. For example, the photo tagging site Flickr has thousands of photos uploaded each second, an untenable amount to have labeled by experts [6].

3 Methodology

In this project we are going to develop a tool that is mainly useful for the collaboration between the software engineers. And also it focuses on the following aspects.

1. Team Awareness. This is about the people involved in development of the project , what their actions , roles , positions are in the development of the project.

2. Connects all the intended people to coordinate with each other.

3. Manage dependencies among activities and organizations. Dependencies over here talk about the interrelation of work from one activity to another ie the dependencies of one work on another.

4. Coordination through Team Knowledge. This point mainly focuses on the co ordination among much similar work-related activities and the people involved in that particular work activity .

5. Reduce dependencies among engineers .

6. Identify and resolve errors.

7. Social Aspects in Software Development.

4 Findings

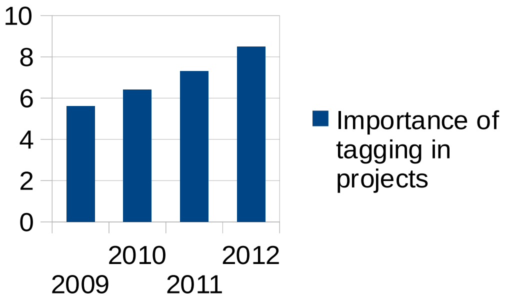
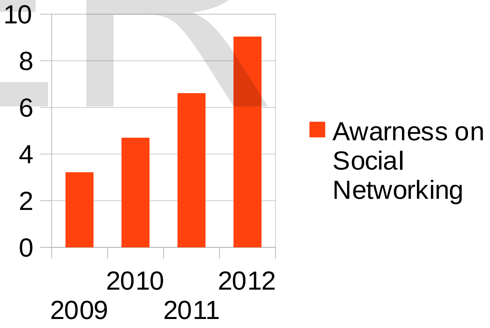


Fig a.Tags



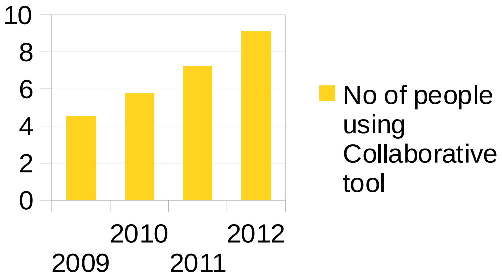


Fig b. Social Network

Fig c. Collaboration Tool

From the above graphs it is clearly visible that the number of people using tags , awareness of social networking , use of the collaborative tool are gradually increasing year by year . Thus by implementing the this paper (the ideology behind the paper) the productivity of the project may increase in the coming years .

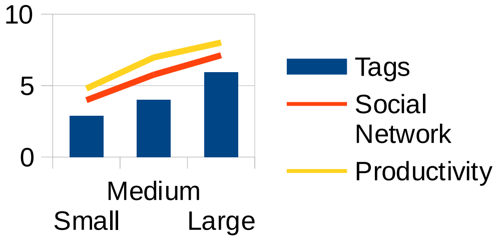


Fig d. Produtivity

Figure d is about the productivity that may increase on the implementation of this research.

5 Conclusion

In this paper we have presented observations that tagging work activities along with social networking for communicating matters of concern with the team and higher officials within the company, when deployed for the use within an organization would increase productivity and improve team dynamics.

Acknowledgment

First of all, I am grateful to The Almighty God for establishing me to complete this research.

I wish to express my sincere thanks to Mrs. Chepure Srilatha, Deputy Director, Aurora's Scientific Technological and Research Academy

I place on record, my sincere gratitude to Mrs. Bemula Aruna, HOD, Department of Computer Science Engineering, for her constant encouragement extended to me.

I also thank Mrs. K. Madhuri Katabathuni, Senior Assistant Professor, Department of Computer Science Engineering, Aurora's Scientific Technological and Research Academy and Mr. G. Sudhakar, Academic Associate (Lecturer), Department of Computer Science Engineering , School of Information Technology, JNTU - Hyderabad.

I take this opportunity to record my sincere thanks to all the faculty members of the Department of Computer Science Engineering from Aurora's Scientific Technological and Research Academy for their help and encouragement. I also thank my parents for their unceasing encouragement and support.

I also place on record, my sense of gratitude to one and all who, directly or indirectly, have lent their helping hand in this project.

References

[1] Elizabeth F. Churchill, Sara Bly, 1999. "It's all in the words: Supporting work activities with lightweight tools".

[2] Scott A. Golder and Bernardo A. Huberman, 2002. "The Structure of Collaborative Tagging Systems".

[3] Celadon R. B. de Souza, David Red miles, Paul Dourish, 2003. "Breaking the Code", Moving between Private and Public CollaborativeSoftwaDevelopment

[4] Robert J. Sandusky, Les Gasser, 2005. "Negotiation and the Coordination of Information and Activity in Distributed Software Problem Management"

[5] Christopher Poile, Andrew Begel, Nachiappan Nagappan, 2009 "Coordination in Large-Scale Software Development: Helpful and Unhelpful Behaviors".

[6] Paul Heymann, Andreas Paepcke, and Hector Garcia-Molina, 2010 "Tagging Human Knowledge".

[7] Marcelo Cataldo, Matthew Bass, James D. Herbsleb, Len Bass, 2011. "On Coordination Mechanisms in Global Software Development"