

Requirements Engineering for Social Software

Anna Glukhova

Information Systems, RWTH Aachen, 52056 Aachen, Germany
glukhova@dbis.rwth-aachen.de

1. Introduction

The modern Internet world is enriched with social software, often referred to as Web2.0 [O'Rel05]: YouTube, Flickr and FaceBook are some of the most famous examples. Social software bears some special characteristics. One is the shift of the focus from technology to the user. Users increasingly play the central role in the design of such systems. A second characteristic is the self-organization of Web 2.0 technologies. Thereby a “community-software evolution cycle” emerges: A certain amount of people start using a provided social platform, which will initiate the building of a community with certain rules and constraints. The community, in turn, will attract new users to the system and, therefore, change its size and structure. Furthermore, the community members get more experienced in the system over time. These changes result in the establishment of new requirements to the platform from the community side.

For a successful community organization, an undefined critical mass of community is always needed. Reaching a critical mass, however, is not a lifetime guarantee – badly managed organizations may chase comers away again, leading to the death of the community. A third special characteristic is the voluntary user participation in his/her community and his/her contribution to it. Here, the question comes up what can attract the user to participate voluntarily and why he/she should be interested in doing so. No concrete formalization of requirements engineering process for social software is established so far. There are numerous emerging Web 2.0 applications, from which only a few successfully managed to build up a community and expand extremely. Unfortunately, the majority of this new generation software never achieves its critical mass and, therefore, dies out sooner or later.

The cluster project “Context adaptive interaction in cooperative knowledge processes” (CONTici, www.contici.org) focuses on the design and development of the contextadaptive, cooperative systems, with special support for the groups with dynamically changing needs. In the project part “Traceable Requirements Engineering for Communities of Practice (CoP, [Wen98])” of RWTH Aachen University the processes of requirements identification, traceability and realization for cooperative systems have to be defined. Thereby, the system needs to be adapted to the socio-psychological characteristics and learning processes of the user group. In our research we concentrate on the social software. Social software brings enormous potential for cooperative knowledge processes.

Based on the observations stated above regarding user-centeredness, self-organization and voluntarism, it becomes clear that the community and its state are

important context aspects that have to be considered during the whole software evolution. Community rules, constrains and interconnections play an important role for the satisfaction of the community needs: Do people already know each other? Do they have anything in common? Is there already a community and, if yes, is it tight or loose, is it new or already well established? There are many more questions, one can think about. Web 2.0 applications are per se highly community dependent, more concrete on the social and organizational state of the targeted community. For example, systems which target at teenagers will be different from the systems developed to support researchers. The members of these two groups not only differ from each other in their age, but more importantly in their communication and interaction styles, interests and goals, amount of time, which they can invest in the system usage and so on. However, due to our previous findings “the community context [...] are not well taken into consideration“ [CKHJ08]. Obviously, all these aspects connected with a certain state of the target group need to be considered. The RE team can use the experience from other successful projects targeting the same community. Since social aspects evolve over time, these analyses have to be carried out continuously, leading to an evolving design.

The users of social software form communities wherein continuous learning and knowledge exchange emerges. Therefore, the community is subject to an ongoing evolution process [RE02]. The assumption about stability of community structure and needs cannot be applied here. The stated evolution of the community has to be considered in the system design in terms of flexibility. Beginning with a group of users with no experience in the system, the key design principle would be easiness and intuitiveness of the usage. During the time people get experienced with the system and can advance their requirements to the systems.

Not only the community experience changes over the time, but also the community structure evolves: a community can either expand or shrink, get more loose or more tight, there can be many or no newcomers. Although almost none of the social factors can be assumed to be steady over time, the system has to be adapted to this kind of community restructuring. This requires continuous monitoring of the community [Rob02]. Due to the stated observations the need for new approaches for RE of social software which will consider community and its evolution become obvious.

Acknowledgements

This work is supported by the DFG within the research cluster project CONTiCi.

References

- [CKHJ08] Y. Cao, R. Klamma, M. Hou, M. Jarke: *Follow Me, Follow You - Spatiotemporal Community Context Modeling and Adaptation for Mobile Information Systems*, Proc. of the 9th International Conference on Mobile Data Management, April 27-30, 2008, Beijing, China, pp. 108- 115
- [O'Rel05] T. O'Reilly. "What Is Web 2.0 Design Patterns and Business Models for the Next Generation of Software", <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-isweb-20.html>, 2005, last visit 30.09.08

- [Rob02] W. N. Robinson: *Monitoring Software Requirements using Instrumented Code*, IEEE, Proceedings of the 35th Annual Hawaii International Conference on Systems Sciences, January 7-10 2002
- [Wen98] E. Wenger. *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press, Cambridge, UK, 1998