

A Web Platform for Social Requirements Engineering

Steffen Lohmann¹, Sebastian Dietzold², Philipp Heim¹, Norman Heino²

¹Interactive Systems and Interaction Design
University of Duisburg-Essen
Lotharstrasse 65, 45057 Duisburg, Germany
{lohmann, heim}@interactivesystems.info

²Agile Knowledge Engineering and Semantic Web
University of Leipzig
Johannisgasse 26, 04009 Leipzig, Germany
{dietzold, heino}@informatik.uni-leipzig.de

Abstract: This paper presents a web platform that applies concepts from the domain of Social Software to requirements engineering. The platform implements several community-oriented features that support collaboration and knowledge sharing and aim to foster the engagement of larger groups of stakeholders in the collection, discussion, development, and structuring of software requirements.

1 Motivation

Today's requirements engineering (RE) tools are primarily designed to support a relatively small group of experts in the capturing, structuring, development, and management of software requirements. The direct engagement of larger groups of stakeholders (e.g., end-users) is usually not an issue as commercial tools such as DOORS, RequisitePRO, or CaliberRM are too heavyweight in order to be used by untrained stakeholders that have only little expertise in requirements engineering [DG07, NSK00]. Furthermore, existing tool solutions provide only limited support for interaction and collaboration among a large number of diverse stakeholders [Wh07, De07]. For these reasons, participants are often forced to switch to additional tools (e.g., e-mail, instant messaging) or face-to-face contact for communication and collaboration in the requirements engineering process.

However, if participants are geographically distributed, face-to-face contact is rather limited. On the other hand, using additional collaboration tools requires further installations and a change of the application environment resulting in an increased effort and participation barrier for the stakeholders. But the main problem of all collaboration among stakeholders that happens independently of RE tools is that it cannot be fully tracked by the RE tools and therefore provides little to no transparency and traceability: For instance, the collaboration process that takes place before a requirement is entered in the RE tool is often insufficiently documented and therefore hard to retrace.

In order to overcome these drawbacks of tool-independent collaboration, RE solutions must offer integrated features that better support collaboration among stakeholders. These features should be easy-to-use and should foster stakeholder engagement but must also allow to track and trace the collaboration activities. The advent and great success of a new generation of web-based, community-centered applications – often subsumed under the label *Social Software* – might stimulate the integration of these types of collaboration features in RE environments. Some typical characteristics of Social Software are community- and user-orientation, simplicity, quick collaboration, self-organization, social feedback, transparency, and emerging structures (cp. Hi06]) – these are also central issues for any RE solution that supports social requirements engineering. In summary, *Social Software Engineering (SSE)* can be defined as “the application of processes, methods, and tools to enable community-driven creation, management, deployment, and use of software in online environments” [Ha08, p. 531].

Based on these ideas, we developed a web platform within the SoftWiki project [SW09] that offers lightweight support for social requirements engineering. The platform enables a large number of geographically distributed stakeholders to collaboratively collect, discuss, semantically enrich, and classify software requirements. This paper presents selected features of the web platform that illustrate how Social Software concepts might fruitfully complement RE practices in an integrated fashion and might encourage stakeholder groups that would not use conventional requirements engineering tools to actively participate and collaborate in software development.

2 Web Platform for Social Requirements Engineering

Fulfilling an essential prerequisite of all Social Software, the RE platform can be easily installed on a server in the internet (or intranet) allowing it to be used from any location with internet access. Since the platform is completely web-based only a web browser is required in order to take part, without any need for further application or plug-in installations. This helps to reduce the participation barrier and fosters distributed collaboration.

2.1 Balancing Self-Organization and Moderation

The web platform follows the Wiki philosophy [LC01] as it is driven by the ideas of quick collaboration and little regulation: In general, every registered user is allowed to edit and discuss all existing requirements, to enter new ones, or to define relations between requirements. Similar to Wikis, every change is logged in a revision history along with the author’s name allowing to track, review, and selectively rollback changes, which guarantees transparency as a key issue in RE. Though not implemented in the current version, sophisticated visualizations based on the data of the revision history might additionally be offered to support traceability as another key issue in RE, for instance, by visualizing the traces in the change history of requirements [DC06].

Though the web platform is designed in a way that allows a high degree of self-organization, it cannot be expected that untrained stakeholders are able to create a comprehensive collection of high-quality requirements. Thus, the supervision and moderation by experienced requirements experts remains crucial for a project’s success. However, similar to Wikis the moderation process is intended to be rather unobtrusive by not forcing participants to fulfill certain tasks or activities. In accordance with the principles of Social Software, the single user should always commit herself to the overall goals of the community, in this case, the successful collection of requirements for a software product.

2.2 Combining Top-down and Bottom-up Classification

Figure 1 shows a screenshot of a test installation of the web platform. After login, the user gets an overview of the requirements that have already been entered (Fig 1 C0). The left sidebar offers different types of navigation: A tree navigation enables the exploration of the requirements collection along a hierarchical classification structure (Fig. 1 A0/1). As common in RE, this taxonomy is pre-defined by the project managers and is usually based on Best Practices and experiences from prior projects. If a user defines a new requirement she must decide in which class it fits best (Fig. 1 C1.1).

In addition to this pre-defined classification, the web platform adopts a type of classification that is well-known from Social Software: Users can collaboratively assign freely chosen keywords (so-called *tags*) to requirements (Fig. 1 C1.1). These tags are also presented in the sidebar and can be used for navigation. They are visualized as an alphabetical tag cloud where a tag’s font size represents its popularity (Fig. 1 B0/1).

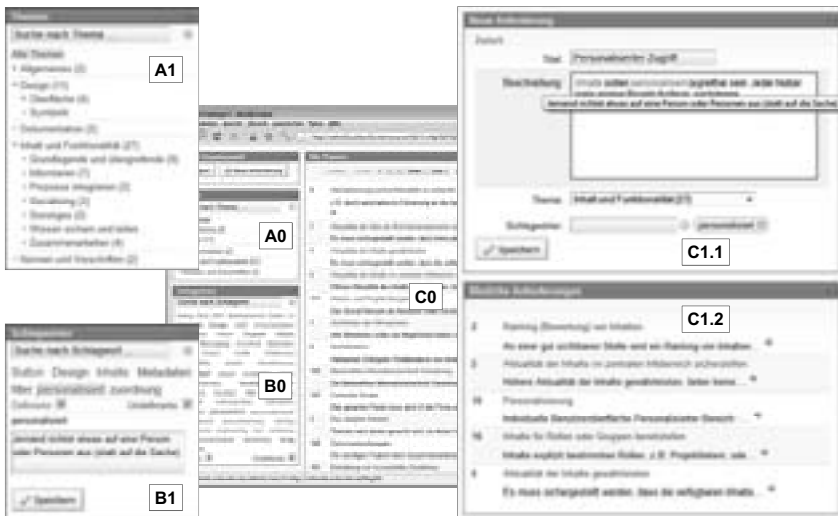


Figure 1: Requirements are classified in both a taxonomic (A0/1) and folksonomic (B0/1) way allowing for a combined top-down and bottom-up exploration. The requirements are collaboratively edited (C 1.1). Editing is differently supported, for instance, existing requirements that are detected to be similar to the one entered are displayed while typing (C1.2).

Thus, the web platform offers both a top-down (taxonomic) and bottom-up (so-called folksonomic) approach of classification, which can be used in combination when navigating in the requirements collection. For instance, a set of requirements resulting from a selection of a class in the taxonomy tree can be further filtered by selecting tags from the tag cloud and vice versa. To ease combined navigation, at each time only tags are shown in the cloud that enable additional filtering.

2.3 Maturing Vocabulary: From Tags to Glossary Terms

Even though tagging is a popular and successful concept in Social Software, it is also very ambiguous as freely chosen keywords of users can have several meanings. However, ambiguity is normally to be avoided in the definition of software requirements. Requirements should generally be formulated in a way that fosters shared understanding and leaves minimal room for divergent interpretations. Ambiguous, uncommon or technical terms need to be further defined. Therefore, the web platform implements an advanced form of tagging by allowing to add definitions to tags: If a tag is selected from the tag cloud, the user can enter a description that explains its meaning in a textbox below the cloud (Fig. 1 B1). In doing so, the user transforms an *undefined tag* in a *defined tag* that is visually distinguished in the cloud. That way, the participants successively create a glossary for central terms in the requirements collection.

In addition, the platform assists in the correct interpretation of a requirement's meaning by highlighting all defined tags in the text and showing their definitions in tooltips (Fig. 1 C1.1). That way, users can easily lookup an ambiguous, technical or unknown term's meaning if this meaning has already been defined by other users. Since the highlighting is already provided while a stakeholder enters her requirement, it also advises the stakeholders to check for existing definitions and hence prevents the misuse of already defined terms when expressing requirements.

The user-assigned tags can moreover be used to update a project's taxonomy from time to time: For instance, popular tags (i.e., tags with a large font size in the cloud) might be considered as valuable to be integrated in the taxonomy at some stage of development or in a subsequent project.

2.4 Social Feedback and Prioritization of Requirements

In addition, the web platform offers discussion and rating features as they are well-known from other Social Software contexts (see Fig. 2). These social feedback mechanisms might be very helpful for authors of requirements as they give some hints where a requirement needs to be improved or more precisely defined. They can furthermore provide a valuable starting point for prioritization of requirements.

In general, we distinguish three types of social feedback: *commenting*, *rating*, and *voting*. Comments can be used to discuss requirements and help to improve their quality. Ratings are similar to comments but additionally allow to judge a requirement's quality on a five point scale. Voting enables users to express their agreement or disagreement

with a requirement, i.e., if they would like to see a requirement realized in the software product or not. However, we experienced that – similar to other Social Software contexts – the meaning of *rating* is differently interpreted: some rate the quality of the requirement description, others the quality of the requirement itself; some argue with personal experiences or opinions, others try to be objective in their judgments. Therefore, developers must be very careful when analyzing the social feedback and basing their decisions on it or using it for prioritization of requirements. Apart from that, the users' *votes* might be a valuable measure when prioritizing requirements. In accordance with the original idea of Social Software, the priority of a requirement could be calculated as the average of all votes. However, depending on the use case, also other models for weighting the stakeholder votes (e.g., by the stakeholders' roles) are imaginable and can easily be implemented accordingly.



Figure 2: Requirements can be discussed and rated by the stakeholders

2.5 Additional Features

Besides the described functionality, the web platform offers further support that might be helpful in social requirements engineering but is outside the scope of this paper, such as features for interlinking requirements or adding files that contain illustrations and diagrams. Furthermore, an export in the Requirements Interchange Format (RIF) [RIF07] is currently implemented to better support the subsequent processing of the requirements data.

Another novelty is the technical backend of the web platform that builds on Semantic Web standards. It is based on OntoWiki [ADR06], a tool for distributed knowledge engineering, and uses several approved vocabularies (e.g., SKOS, FOAF, SIOC). The conceptual structure of the web platform itself is also described in ontological form [Ri07]. This allows for a high semantic interoperability and enables easy import and export of parts of the platform's contents in RDF format, for instance, the interlinked requirements, the taxonomy and folksonomy, or the users' social network structure. The configuration of the platform, such as the modification of the taxonomy or the administration of users, is also realized in the OntoWiki backend.

3 Discussion and Conclusion

In this paper, we have presented a web platform that applies several concepts of Social Software to requirements engineering. As has become apparent, a central challenge consists in balancing conflicting demands: On the one hand, an adequate solution should follow Social Software principles, such as simplicity, community-orientation, quick collaboration, and social feedback, in order to activate larger groups of stakeholders to directly participate in requirements engineering. On the other hand, sufficient formality must be provided in order to serve typical demands of requirements engineering, such as structured access or efficient analysis and post-processing of the collected requirements.

The presented approach aims particularly at supporting early phases of requirements engineering with many distributed participants and much informal collaboration. It focuses on simplicity and ease of participation instead of advanced and powerful requirements management features. It emphasizes the social experience of developing requirements for a software system: Diverse stakeholders are enabled to collaboratively collect, discuss, improve, and structure requirements, even without training and experience. Under the supervision of experts in the field, the requirements are formulated in natural language and are successively improved by all participants. These goals do explicitly not exclude the possibility of later 'cleaning' and refinement of the requirements by experienced engineers and in established requirements management tools.

The presented web platform should not be regarded as a comprehensive solution but rather as a starting point and first step towards better support for integrated collaboration. To further examine the feasibility of this approach, much more work is needed. So far, we cannot make any reliable statement on how well the web platform performs in real world contexts. Evaluations within use cases of industry partners from the project are in preparation and are expected to provide some valuable insights regarding the general acceptance and experienced benefits of a web platform for social requirements engineering.

References

- [ADR06] Auer, S.; Dietzold, S.; Riechert, T.: OntoWiki – A Tool for Social, Semantic Collaboration. In: Proceedings of the 5th International Semantic Web Conference (ISWC 2006), LNCS 4273. Springer, Berlin, Heidelberg, 2006; pp. 736-749.
- [De07] Decker, B.; Ras, E.; Rech, J.; Jaubert, P.; Rieth, M.: Wiki-Based Stakeholder Participation in Requirements Engineering. In: IEEE Software, 24(2), 2007; pp. 28-35
- [DG07] Delgadillo, L.; Gotel, O.: Story-Wall: A Concept for Lightweight Requirements Management. In: Proceedings of the 15th International Requirements Engineering Conference (RE'07). IEEE; pp. 377-378.
- [DC06] Duan, C.; Cleland-Huang, J.: Visualization and Analysis in Automated Trace Retrieval. In: Proceedings of the 1st International Workshop on Requirements Engineering Visualization (REV'06). IEEE, 2006.
- [Ha08] Hammouda, I.; Bosch, J.; Jazayeri, M.; Mikkonen, T.: 1st International Workshop on Social Software Engineering and Applications (SoSEA 2008). In: Proceedings of the 23rd IEEE/ACM International Conference on Automated Software Engineering (ASE 2008). IEEE, 2008; pp. 531-532.
- [Hi06] Hippner, Hajo: Bedeutung, Anwendungen und Einsatzpotenziale von Social Software. In: HMD – Praxis der Wirtschaftsinformatik, 252, 2006; pp. 6-16.
- [LC01] Leuf, B.; Cunningham, W.: The Wiki Way: Quick Collaboration on the Web. Addison-Wesley Longman, Amsterdam, 2001.
- [NSK00] Nikula, U.; Sajaniemi, J.; Kälviäinen, H.: A State-of-the-Practice Survey on Requirements Engineering in Small- and Medium-Sized Enterprises. Research report, Telecom Business Research Center Lappeenranta, 2000.
- [RIF07] Requirements Interchange Format (RIF), 1.1a, 2007; www.automotive-his.de/rif/
- [Ri07] Riechert, T.; Lauenroth, K.; Lehmann, J.; Auer, S.: Towards Semantic based Requirements Engineering. In: Proceedings of the 7th International Conference on Knowledge Management. J.UCS, 2007; pp. 144-151
- [SW09] SoftWiki – Cooperative research project funded by the BMBF, <http://softwiki.de>
- [Wh07] Whitehead, J.: Collaboration in Software Engineering: A Roadmap. In: 2007 Future of Software Engineering. IEEE, 2007; pp. 214-225

