Can Collaborative Tagging Improve User Feedback? A Case Study

Rana Alkadhi
Technische Universität München
Munich, Germany
alkadhi@in.tum.de

Dennis Pagano
Technische Universität München
Munich, Germany
pagano@cs.tum.edu

Bernd Bruegge
Technische Universität München
Munich, Germany
bruegge@in.tum.de

King Saud University
Riyadh, Saudi Arabia
ralkadi@ksu.edu.sa

ABSTRACT

User feedback is a rich source of information which can help developers to improve software quality and identify missing features. However, developers need to analyze user feedback in order to assess its relevance and potential impact, which bears several challenges due to its quantity, quality, structure, and content, particularly when feedback volume is high. In this paper we present the results of a case study to explore which role collaborative tagging can play to improve user feedback, in particular, its impact on the navigation within, understandability, and structure of user feedback. Our results indicate that collaborative tagging might contribute to decrease the pain when analyzing and organizing user feedback.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications

General Terms

Human Factors, Documentation, Measurement

Keywords

Social software, user involvement, user feedback, collaborative tagging

1. INTRODUCTION

With the emergence of social media and Web 2.0, today's generation of software developers has access to a variety of novel tools to communicate with and learn from users [1, 24]. Likewise, software users of today can provide feedback and request new features with a variety of tools built on social media pillars [24]. This feedback is a rich source of information, which can help developers to improve software quality, identify missing features, indicate real software usage, and assess the acceptance of the software and even specific features in the user community [16]. However, developers need to analyze user feedback in order to assess its relevance and potential impact, and to create prioritized tasks which fit into their product roadmap, which typically requires high effort as it is mostly accomplished manually [16]. Moreover, users seem to give feedback just as their concern happens, typically without mediation, and often in the form of natural language, which might result in unstructured information with unpredictable content and quality [15,16]. These factors constitute the main problems of developers when trying to get answers from collected user feedback.

Collaborative tagging is one of the most popular social media features adopted by the software development environments. However, despite the widespread use of Web 2.0, only a few studies investigated the implications of using this technique in the modern software development environment [24]. Even less is known in the context of user feedback. Schneider et al. [19] let users themselves annotate their feedback with a selection regarding the type of feedback, e.g. complaint, and the object where the context of the feedback applies, e.g. subsystem, but with predefined categories and context objects. Maalej and Pagano [13] propose a context-aware framework that supports the social development process, where users can tag their feedback to help other users find it and to facilitate its analysis.

In this paper, we report on a case study we conducted to explore, which role collaborative tagging could play to improve user feedback, and especially to overcome some of the challenges developers face while working with user feedback. In particular, our goal was to investigate the impact of collaborative tags on navigation within, understandability, and structure of user feedback systems. To this end, we implemented a simple prototype using data of issue tracking system with already assigned tags. To analyze the effects of collaborative tagging on user feedback, we observed subjects while using the prototype to complete a given series of simple tasks and analyzed their answers to a subsequent questionnaire.

The remainder of this paper is structured as follows. Section 2 summarizes related work about user feedback and tagging in software development. Section 3 explains our

SSE'14, November 17, 2014, Hong Kong, China

http://dx.doi.org/10.1145/2661685.2661692

study setting in terms of research questions and research method. Section 4 describes the results, while Section 5 discusses our findings, their implications, and limitations. Finally, Section 6 concludes the paper.

2. BACKGROUND

We divide the background discussion into two main areas: research on user feedback in software development and research on tagging in software development.

2.1 User Feedback in Software Development

Although user involvement has been researched over the last three decades, user input and feedback mechanisms in software systems lack a common theory and are characterized by huge communication gaps [12]. To address these limitations, Maalej et al. [12] propose a continuous and context-aware model for gathering and understanding user input. Moreover, Seyff et al. [20] and Schneider et al. [19] propose to continuously elicit user requirements with feedback from mobile devices, including information on the application context. However, eliciting requirements from continuous feedback with thousands of stakeholders bears several technical challenges due to its quantity, quality, structure, and content [15–17]. Pagano [15] proposes a framework for the systematic analysis of continuous user input. Finally, Castro-Herrera et al. [4] propose a framework which uses data-mining and recommender system techniques to guide end-users into the right discussion forum to facilitate their collaboration.

Our work builds on these studies and investigates, if and how applying collaborative tagging to user feedback systems can contribute to help developers dealing with the gathered feedback. In particular, we evaluate collaborative tagging as an approach to connect related feedback and to improve its quality, in order to explore if it can help to improve the navigation within, understandability, and structure of the gathered data.

2.2 Tagging in Software Development

The concept of tagging originates from the social computing domain and is one of the defining characteristics of Web 2.0 [25]. Storey et al. [23] define tags as "unconstrained keywords freely associated to a piece of information to describe it and assist in later refinding". Further, collaborative tagging was described by Golder and Huberman [5] as "the process by which many users add metadata in the form of keywords to shared content". It has been shown that collaborative tagging is both better manageable and more scalable than other categorization approaches, due to its flexibility and lightweight, bottom-up nature: users can add their own tags without the need to predefine tags, while an arbitrary number of tags can be assigned per resource [2, 24, 25]. But the real strength of collaborative tagging lies in its inherent use of the wisdom of the crowd [11], a first class aid to ensure a specific level of quality among the tags.

The main use cases of tags are typically to categorize and retrieve specific information [25]. Developers use tags to categorize commits and branching events in software version control systems, and as source code annotations [23, 25]. Moreover, tags have been introduced in popular bug tracking systems, such as Bugzilla and JIRA, to help documenting and structuring reported issues, as well as on question and answer (Q&A) websites, such as Stack Overflow, to categorize each question with user-defined topics. Although different names are used to refer to tags in these systems *(keywords* in Bugzilla, *labels* in JIRA, *tags* on Stack Overflow), the underlying functionality is similar.

The introduction of tags into software development is not new, several studies have investigated the adoption rate and implications of tagging in software development. In the early stages of software development, Ossher et al. [14] propose tagging as a valuable approach to identify and organize concerns during early exploration in business analysis and explore how tags can be hardened as the analysis evolves. Treude and Storey [25–28] examined how tags are used by professional software developers to document work items, the different roles of tags, and how tagging mechanisms were adopted by developers. The authors showed that tags were eagerly adopted by developers to support informal processes such as life-cycle management and the identification of cross-cutting concerns. Their findings indicate that tagging may play an important role in collaboration and improving team-based software development practices. They suggest future work to examine the benefits of integrating tagging and other social computing mechanisms in other areas of software development and how these mechanisms may improve software development practices. A recent tool that uses the concept of social tagging to support communication and coordination among software developers is described by Storey et al. [23]. Their tool TagSEA (Tags for Software Engineering Activities) allows tagging of locations in source code, called waypoints, and artifacts such as files. A preliminary evaluation of TagSEA [22] showed that it is promising in supporting navigation and communication in distributed software development.

Most existing studies explore tagging as a communication and collaboration mechanism among developers in software development teams, where tags are assigned to work items and development tasks. On the contrary, the goal of our study is to explore, which roles tagging can play for the communication between developers and *users.*

3. STUDY SETTING

Our goal is to explore the benefits of applying collaborative tagging to user feedback systems and to investigate, if and how it may contribute to overcome some of the challenges that developers face when working with large amounts of collected user feedback. In particular, we want to explore, how collaborative tagging may enhance the navigation within, understandability, and structure of user feedback systems.

In general, collecting user feedback without any structure is neither helpful for developers, nor for users. Therefore, our plan is to observe effects of tagged user feedback on both stakeholders, with the goal to hypothesize the common benefits. We use the notion of collaborative tagging to refer to tags assigned by both developers and the user community.

In the remainder of this section, we outline our research questions and setting.

3.1 Research Questions

We are interested in collecting users' and developers' opinions about the effectiveness of tags and impressions about *how* they use existing tags, rather than analyzing their tagging behavior, i.e. *what* they tag, or *what type* of tags they insert. We are particularly interested in the effects of collaborative tagging on three dimensions of user feedback: naviga-

tion, understandability, and defining structure. Furthermore, we investigate the tagging permissions, i.e. who should be granted the required rights to add or change tags on user feedback. Specifically, we asked the following questions.

3.1.1 *Navigation:*

- How can tagging presentation techniques, e.g. tag cloud and tag list, improve browsing and navigation through available feedback?
- Between tag cloud, tag list, and search interface: which method would users prefer for the following tasks:
  - Forming an impression of the available user feedback
  - Finding related user feedback
  - Searching for a particular user feedback

3.1.2 *Defining Structure:*

- How can tags help in connecting semantically related user feedback?
- How can tag clouds or ordered tag lists help in identifying the parts of the software receiving the most feedback?

3.1.3 *Understandability:*

- How can tags help in forming a basic idea about the feedback content?
- How can tags improve the understandability of the feedback?

3.1.4 *Tagging Permission:*

- Who should be able to tag user feedback?

We ask these questions with the goal to explore the implications of applying collaborative tagging to user feedback systems for both developers and users.

3.2 Data Sample and Prototype

This case study has two main goals. First, we want to observe *how* users use existing tags. Second, we aim to investigate the effectiveness of tagging user feedback in improving navigation, structure, and understandability of user feedback. To this end, we implemented a simple prototype that displays Spring platform's issues together with the tags that have been assigned to them.

Spring platform[1] is an open source lean, and modular platform for modern applications. We had exported the data we used in our study from the Spring platform JIRA issue tracker on May 11, 2012. It consists of 8,416 issues tagged with 492 tags (min. 0 tags, max. 11 tags, and avg. 0.195 tags per feedback). In JIRA, tagging permission is associated with "edit issue" permission, if the user has no edit permission, she can not tag the corresponding issue. While there is no limit on the number of tags that can be defined, it is not mandatory to tag each issue. The defined tags can be viewed, searched and removed.

[1]http://spring.io/

(a) Alphabetical tag list

(b) Tag list sorted by popularity

(c) Tag cloud

Figure 1: Tags presentation techniques used in the prototype.

In our prototype, a user can view tags in three different tag presentations techniques in addition to a basic search interface. These techniques are alphabetical tag list, tag list ordered by popularity, and tag cloud, as illustrated in Figure 1. In the alphabetical tag list, tags are listed alphabetically whilst in the prioritized tag list, tags are listed in a descending order according to the number of issues tagged by them. In both lists, *tag frequency,* i.e. the number of issues tagged by that tag, is displayed beside each tag. Finally, the tag cloud displays tags, as its name suggests, in a tag cloud view and the bigger the text, the more popular the tag. However, as displaying too much information on the tag cloud degrades its meaning to nothing more than clutter [9], tags that are used only once are not included in the tag cloud. Six discrete font sizes were used for tag cloud rendering. However, the prototype does not include tagging functionality, i.e. inserting tags or editing existing tags, as it exceeds the scope of this study.

When the user clicks on a tag in any of the tags presentations techniques, all the issues tagged by that tag are displayed. In addition to these issues, tags related to the chosen tag are displayed in a descending order according to their similarity with the current tags.. The similarity between tags is measured according to the relative co-occurrence between tags. We used Jaccard coefficient as suggested by Hassan-Montero and Herrero-Solana [7]:

$$RC\,(A,B) = \frac{|A \bigcap B|}{|A \bigcup B|},$$

where RC is the relative co-occurrence and A and B are the set of resources described by two tags. So the relative

co-occurrence is equal to the ratio between the number of resources in which tags co-occur, and the number of resources in which any one of the two tags appears.

3.3 Research Method

Our research method consists of two parts: observing subjects while using the prototype for a given series of simple tasks, followed by a questionnaire. The study was conducted in sessions, with one subject each session. Before starting each session, we briefly explained the concept of tagging and the study purpose. The subjects were asked to complete three tasks using the prototype. While the subjects were using the prototype, they could request help and comment on the prototype, e.g. suggesting features and improvements. When the subject completed the given tasks successfully, she was asked to fill out a questionnaire in the second part of the session. While filling out the questionnaire, participants could ask for illustration if the question was not clear. The time of completing the requested user tasks varied depending on the user skills and experience with tagging systems. Each session lasted between 20 to 30 minutes. During the session, we recorded observations about users, their interactions withe the prototype, and their comments or suggestions for further analysis.

3.3.1 Subjects

18 participants, mainly computer science and computer engineering students at Technische Universität München, took part in the experiment: 7 females and 11 males. All participants had considerable experience in software development, making them good candidates to mimic Spring users suitable for our data sample. In addition, they worked in teams to develop software applications as part of their studies which allows them to mimic developers as well.

12 participants had previous experience with tagging web resources and 4 had used tags in software development projects, in particular, for tagging projects, specific releases, bugs, and issues. However, most tagging activities they had experience with, were done internally by developers.

3.3.2 Tasks

We asked the subjects to complete a series of tasks using our prototype. The tasks were chosen to inform subjects and to ensure a common understanding of collaborative tagging of user feedback before answering the questionnaire. First, each participant was presented with a tag and asked to find its frequency, i.e. issues tagged by this tag, and its related tags. Upon the successful completion of this task, we asked the participant to find a specific issue that has been tagged by a given tag, read the issue details, and give her assessment whether the assigned tag reflected the issue correctly. Finally, the participant was asked to find the group of issues tagged by the tag most related to a given tag.

For each task, the subjects had the freedom to choose between the presentation techniques described above besides the basic search interface. The tasks were identical for all subjects.

3.3.3 Questionnaire Setup

Upon completion of the tasks, the participants were asked to fill out an online questionnaire[2]. A link to the questionnaire was included in the prototype and all responses were collected anonymously. The questionnaire consists of 5 main areas: demographic information, navigation, defining structure, understandability, and tagging permissions. It is completed as soon as the respondent has answered all questions.

[2]Available on: http://bit.ly/1zicnJe

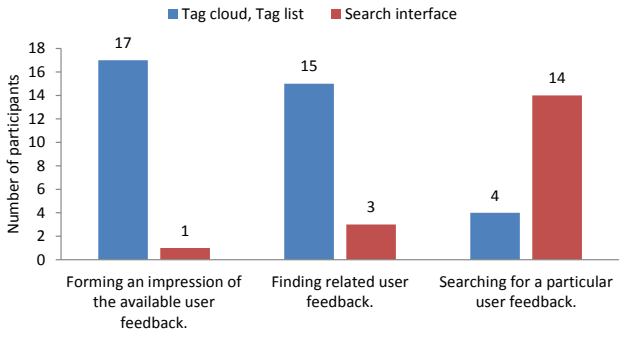

Figure 2: Method selected by participants for the given tasks.

3.3.4 Data Analysis

We mainly analyzed the questionnaire responses to answer our research questions. In addition, although the tasks were designed basically to inform subjects, we analyzed the observations and comments made by subjects while using the prototype to complete these tasks.

4. RESULTS

This section presents our results and findings on the effects of collaborative tagging on the three dimensions of user feedback: navigation, understandability, and defining structure respectively. Additionally, we summarize participants' opinions on tagging permission.

4.1 Navigation

All participants agreed that browsing through available feedback is improved by tagging and tagging presentation techniques, i.e. tag cloud and tag list. In addition, thirteen participants agreed that tag cloud and tag list are better for browsing feedback than a regular search. However, one participant disagreed and preferred the regular search over the tag cloud or tag list, while four participants gave a neutral answer. Figure 2 indicates the participants' preferences, i.e. tagging presentation techniques or search interface, for performing different tasks. The results suggest that tag presentation techniques are typically preferred over regular search interface for free browsing through tags. One of the participants explained: *"I like tag cloud, it is used widely and I would like to use it in the context of software engineering"*. On the other hand, for performing goal oriented tasks, such as finding a specific issue, participants tend to use regular search interface.

When asked to find a specific tag, three participants copied the name of the required tag and used the browser built-in search functionality. For that purpose, it did not make a big difference for them, which tag presentation technique was used (tag cloud or tag list). On the other hand, the majority of participants chose to use the alphabetically ordered tag list to search for a particular tag.

Although it may seem trivial to some users that the font

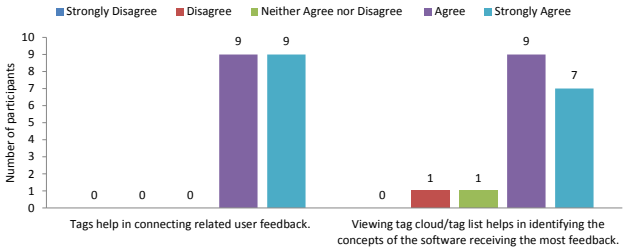

Figure 3: Participants agreements with tags role in defining structure of user feedback.

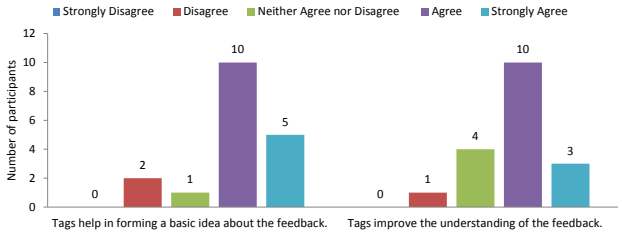

Figure 4: Participants agreements with tags role in improving feedback understandability.

size of each tag represents its relative weight within the group of tags being rendered, some participants did not perceive the distribution of font weight through tag cloud and asked questions like "what do the larger fonts represent?". The following statement was included in the description above the tag cloud: "The bigger the text, the more popular the tag", but obviously some participants only read the first sentence of the description and neglected the rest. Based on this observation, we think it might be wise to summarize the descriptive information as much as possible to avoid loosing users' attention. One of the participants suggested a number of improvements to the tag cloud interface, including ordering the tags in the tag cloud alphabetically and using different shades of the same color for different tags to represent their relative popularity. Improvements to the tag list were also suggested, for example a participant indicated that ordering the list vertically is better and easier for browsing than horizontal ordering (horizontal ordering was used in the prototype).

Two participants stressed on showing the tag frequency whenever the tag appears. Further, a participant suggested to display the frequency of tags in the tag cloud, e.g. displaying the precise frequency in a tooltip when moving the mouse over the tags in the tag cloud, as the font size only indicates the relative popularity of the tag withing the displayed group of tags. Moreover, one participant indicated that the ability to search tags is an important feature in tagging systems.

4.2 Defining Structure

As shown in Figure 3, all participants agreed that tagging helps in connecting semantically related user feedback and thus supports defining the underlying structure. Furthermore, sixteen participants agreed that browsing through tag clouds and tag lists helps them in recognizing the parts of the software receiving most feedback. On the other hand, one participant did not believe that tags can reflect the amount of feedback received for different parts of software and one participant remained neutral.

The majority of participants showed their interest in the related tags and indicated that it was a very important feature in the tagging system. Although the following description was presented above the related tags section on the prototype: "Below are the most related tag to *the current tag*, listed in a descending order according to their similarity", some participants did not understand what related tags represent and requested further explanation. Some other participants commented that it was difficult to find the related tags. One participant described the problem: *"By clicking on the tag, the user is expecting to see the associated issues. Displaying related tags beside the issues makes them difficult to notice"*. As an improvement to this problem, she suggested including a cloud view of the related tags where the font size of the tag indicates its relative similarity to the chosen tag (i.e. the larger the font, the more similar the tag). However, two participants had the opposite opinion and commented that related tags are displayed on the right place. Another suggestion was to add more description about similarity between tags, e.g. describing that similarity between tags is calculated based on their co-occurrence.

A remarkable suggestion made by two participants was having fixed categories, which could be created by administrators or the development team, such as 'debugging', 'development', 'UI', and 'testing'. This suggestion, which relates also to tagging permissions, is interesting because automated clustering techniques can be easily applied to group tags under those categories.

4.3 Understandability

Fifteen participants agreed that viewing the tags assigned to a given feedback helped them in forming a basic idea about that feedback without the need to read its detailed description. However, the number of neutral participants, i.e. neither agree nor disagree, increased in deciding whether tags assigned by users improve the understandability of that feedback, as shown in Figure 4.

Whether tagging can improve the understandability of user feedback is highly dependent on whether the assigned tags reflect the meaning of the feedback correctly. As one of our participants put it: *"The most important aspect of tagging user feedback is tags' ability to reflect the content of the feedback; this is what can make a difference"*. Another participant commented that adding a description to each tag will improve its usability and understandability by other users. One other participant thought of improving understandability as the main purpose of tags, *"especially if the feedback itself was not written in a clear form"*.

Three of the participants believed that using specific, discriminating tags and avoiding general keywords represents an important factor impacting the role of tags. Another participant trusted that a collection of tags associated with a feedback could reflect the meaning of feedback better than a single tag. This result encourages the collaborative tagging of user feedback.

4.4 Tagging Permission

When participants were asked about their preferences on tagging permissions, i.e. who should be able to add new tags, an equal number of participants voted for both the feedback creator and the members of development team, as

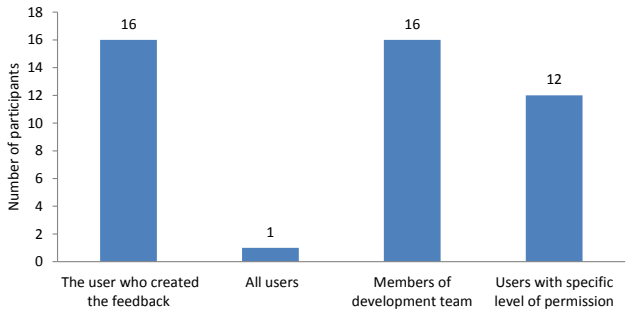

Figure 5: Participants opinions on tagging permissions.

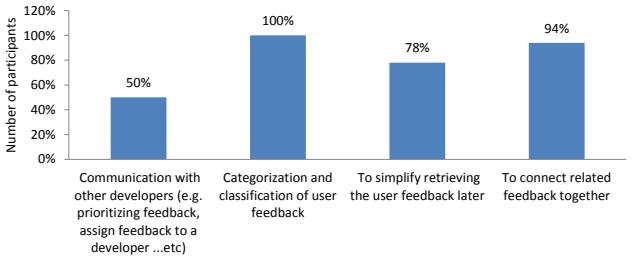

Figure 6: Tagging roles from participants' perspective.

shown in Figure 5. The second popular opinion in the list is to give a different level of permission to different users, e.g. specific users with edit permissions, or members of a specific group. An interesting finding of the study is that participants seemed to be reluctant to give a tagging permission to all users, as only one among eighteen participants voted for this option. One participant was strongly against giving all users the permission to tag: *"users usually assign bad tags and may add conflicting tags to related feedback, thus they should not be given the complete freedom in adding new tags"*. This agrees with a statement made by another participant: *"If all users can tag, too general or wrong tags may be assigned which is not good to describe the issue. If it must to be so, new tags have to be checked and reviewed before accepting them"*.

Three participants suggested to give different weights to tags according to their creator, for example tags created by the development team, expert users, or users with a specific minimum reputation in the user community get higher weights. Another suggestion to give tagging permission to the system or project administrator was proposed by two participants. Additionally, one participant suggested that redundancy should be prohibited, i.e. a specific tag can not be assigned to the same issue more than one time. This suggestion contradicts with a comment made by another participants that the number of times the tag has been applied to the feedback is important as an indicator of which tag represents the feedback best.

Finally, participants were asked about their opinion, which roles tags can play for user feedback. Participants could select all applicable options. Figure 6 summarizes their responses. All participants agreed that tags are important for the categorization and classification of feedback. The second most important role of tags according to participants' responses is connecting semantically related feedback, which was selected by seventeen participants. This means that tags can play an important role in identifying structure among feedback in the eyes of our subjects. Fourteen participants agreed that tags are helpful to simplify retrieving user feedback later, which is another evidence that tagging can play an important role in the navigation through user feedback. Half of the participants voted for using tags to communicate with other developers, e.g. prioritizing feedback or assigning feedback to developers. This result supports the statement "tags are applied to help other developers on the same team with the management of their work and to make them aware of particular incoming work items" which was framed by Treude and Storey [25] in their study on how tagging has been adopted and adapted by a large project with 175 developers. In the same light, one participant suggested to use tags to categorize different releases of the software product. Interestingly, one of the respondents to this question also commented that once tags are used for one of these goals, this implies that they are also used for the other goals in consequence. For example, using tags to categorize and classify user feedback simplifies retrieving this feedback later, which was also explained by one of the respondents: *"I use bug tracking system everyday at work and thing can be really difficult to find, e.g. what release or programming language. I think tagging bugs can help a lot"*.

5. DISCUSSION

We discuss the implications of our findings for developers and users, and summarize the limitations of our study.

5.1 Implications

5.1.1 Implications for Developers

To benefit from user feedback, developers need to understand, what the user wanted to say, if the feedback is relevant, and with which level of importance and impact it should be treated. But because developers typically have to manually analyze the gathered messages to get this information, the effort to make is large and unfortunately grows with the volume of feedback [15, 16].

Our results suggest that collaboratively tagging user feedback might contribute to decrease the emerging challenges in three ways. First, tags can help to group semantically related feedback and thus inherently add a layer of structure to otherwise unstructured messages. This might help developers to access specific feedback more easily, for instance when they need more information about a given failure or more opinions about a missing feature. It also has the potential to mitigate duplicates, as finding related feedback could be easier when similar messages are grouped via tags. Second, the assigned tags might help to improve the understandability of user feedback, as they potentially provide an immediate glance at the most important aspects of the feedback from the author's perspective. Third, tags might contribute to understanding more quickly, which aspects of the software, such as a specific feature or component, receives most feedback. This would help developers to get a more fine-grained impression of the software quality as perceived by their user community.

However, the positive effects of collaborative tagging depend to a large extent on the quality of the assigned tags

and whether they reflect the feedback correctly. For instance, Barua et. al [3] recently found that collaborative tagging on Stack Overflow has led to a "tag explosion", because erroneous and inconsistent tags have been supplied over time. Since such an effect naturally reduces the potential benefits, we think it would be best not to give the permission to create new tags immediately to all users. In our data sample, for instance, only a small number of tags –compared to the number of issues– had been assigned to the issues(492 tags for 8,416 issues). One possible explanation could be the limited tagging permissions granted to novice users in the feedback system from which we drew our sample. In the same light, it could be helpful to recommend existing tags to novice users, which – when done right – could increase the probability of obtaining more accurate tags, while maintaining the lightweight nature of collaborative tagging.

5.1.2 Implications for Users

Users provide feedback, because they want to let developers know about their experience with the software – either to provoke a change, e.g. reporting a bug or requesting a feature, or to communicate praise or displeasement. Their goals typically do not include helping developers to organize their work items. However, users do care about their opinion being heard once they have taken the time to formulate it. Therefore, any means which makes their feedback more valuable without too much additional effort, should be in their interest.

Our findings suggest that collaborative tags could offer these characteristics. When browsing existing feedback, e.g. to check if their issue has already been reported, tag clouds or tag lists might provide an easier access than searching. The structure which emerges from a tag landscape, might then even help to guide and educate users on how they should provide feedback for a specific software. On the individual level, the tags applied to an existing feedback could help to clarify it and make it more understandable for other users, which might reduce the amount of duplicate feedback.

Tags can be presented in various techniques, e.g. in our prototype we included ordered tag lists and tag clouds. Feasibility and uses of the different techniques as well as the comparison between them have been an active research area in the past few years [6, 8, 10, 18, 21]. Based on our findings, we think it could be beneficial to use multiple presentation forms for existing tags, as users tend to make use of different presentations for different tasks. For instance, our study subjects preferred an ordered tag list when searching for a particular tag while they switched to a tag cloud for general browsing.

The fact that there already are feedback systems which allow users to tag their messages, illustrates that such an approach is feasible. For instance, the feedback in our data set had been collected using JIRA[3], an established issue tracker. But unfortunately not all user feedback systems provide tagging facilities. Indeed, issue trackers like JIRA are typically used to gather feedback in more developer-oriented environments or open source communities, where the boundaries between users and developers are blurred. In fact, the typical Spring user is most probably a developer herself. More consumer-like end-users might not have the same time and will to tag their feedback. It should therefore be made as easy as possible for users to tag feedback.

[3]www.atlassian.com/software/jira

5.2 Limitations

As with any chosen methodology, there are limitations to our research methods. The main threat to the external validity of our findings lies in the small number of participants, which does not allow for generalization. However, this case study aims at exploring the opinions of users and developers and observing them while using our prototype, rather than collecting quantitative data, which should be done by further studies with larger populations. Nevertheless, using an existing, large database of already tagged issues allows us to observe, how users could cope with a feedback system once a substantial number of tags have been assigned.

The second limitation lies in the population of our subjects. Our participants were students, who did not develop for Spring and thus were not familiar with the data sample used in the prototype. But as computer science and computer engineering students, they have considerable experience with software development, which makes them good candidates to mimic Spring users. In addition, the fact that they all had worked in teams to develop software applications as part of their studies, should allow them to mimic developers as well.

Finally, large number of JIRA issues in the data sample have no tags at all which may limit the implications of collaborative tagging. In any case, our findings should be substantiated by further studies, including controlled experiments with different types of user feedback systems for both users and developers. We have provided the link to our online questionnaire to allow future research to replicate our work.

6. CONCLUSION AND FUTURE WORK

With the emergence of social media, software users can use various tools to communicate their needs, report bugs, and request features from developers, which results in high volume of gathered user feedback, complicating its analysis by developers. In this paper, we report on a case study to explore if and how collaborative tagging may contribute to decrease the challenges developers face when analyzing user feedback. In particular, we were interested in the impact of freely assigned tags by developers and users on navigation within, understandability, and structure of user feedback systems. Our findings suggest that collaborative tagging may help in connecting semantically related feedback and improving the understandability of the feedback. In addition, the emerging tag space provides easier access to user feedback. However, the ability of tags to improve the user feedback is highly dependent on the quality of assigned tags. The results of our study call for further investigation of the impact of collaborative tagging on user feedback system, its advantages and shortcomings. Future work should analyze how and how fast tags are being adopted in feedback systems, which types of tags are being used, and explore the tagging behavior of both users and developers.

7. REFERENCES

[1] N. Ahmadi, M. Jazayeri, F. Lelli, and S. Nesic. A survey of social software engineering. In *2008 23rd IEEE/ACM International Conference on Automated Software Engineering - Workshops*, pages 1–12. IEEE, Sept. 2008.

[2] E. Bagheri and F. Ensan. Semantic tagging and linking of software engineering social content. *Automated Software Engineering*, Mar. 2014.

[3] A. Barua, S. W. Thomas, and A. E. Hassan. What are developers talking about? An analysis of topics and trends in Stack Overflow. *Empirical Software Engineering*, 19(3):619–654, Nov. 2014.

[4] C. Castro-Herrera, C. Duan, J. Cleland-Huang, and B. Mobasher. A Recommender System for Requirements Elicitation in Large-Scale Software Projects. In *Proceedings of the 2009 ACM Symposium on Applied Computing*, pages 1419–1426. ACM New York, NY, 2009.

[5] S. A. Golder and B. A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, Apr. 2006.

[6] M. Halvey and M. T. Keane. An assessment of tag presentation techniques. In *Proceedings of the 16th international conference on World Wide Web*, pages pp. 1313—-1314, New York, New York, USA, 2007. ACM Press.

[7] Y. Hassan-Montero and V. Herrero-Solana. Improving tag-clouds as visual information retrieval interfaces. In *International Conference on Multidisciplinary Information Sciences and Technologies, InSciT2006*, 2006.

[8] M. A. Hearst and D. Rosner. Tag Clouds: Data Analysis Tool or Social Signaller? *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, page 160, Jan. 2008.

[9] K. Hoffman. In Search Of . . . The Perfect Tag Cloud, 2006.

[10] O. Kaser and D. Lemire. Tag-cloud drawing: Algorithms for cloud visualization. In *WWW'07 Workshop on Taggings and Metadata for Social Information Organization*, 2007.

[11] A. Kittur, E. Chi, B. A. Pendleton, B. Suh, and T. Mytkowicz. Power of the Few vs . Wisdom of the Crowd : Wikipedia and the Rise of the Bourgeoisie. *World Wide Web*, 1(2):1–9, 2007.

[12] W. Maalej, H. J. Happel, and A. Rashid. When users become collaborators: towards continuous and context-aware user input. In *Proceeding of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*, pages 981–990. ACM, 2009.

[13] W. Maalej and D. Pagano. On the Socialness of Software. In *In Proceedings of the International Conference on Social Computing and its Applications*. IEEE, 2011.

[14] H. Ossher, D. Amid, A. Anaby-Tavor, R. Bellamy, M. Callery, M. Desmond, J. De Vries, A. Fisher, S. Krasikov, I. Simmonds, and C. Swart. Using tagging to identify and organize concerns during pre-requirements analysis. *2009 ICSE Workshop on Aspect-Oriented Requirements Engineering and Architecture Design*, pages 25–30, May 2009.

[15] D. Pagano. Towards systematic analysis of continuous user input. In *Proceedings of the 4th international workshop on Social software engineering*, pages 6–10. ACM, 2011.

[16] D. Pagano and B. Bruegge. User involvement in software evolution practice: A case study. *2013 35th International Conference on Software Engineering (ICSE)*, pages 953–962, May 2013.

[17] D. Pagano and W. Maalej. User feedback in the appstore: An empirical study. In *In Proceedings of the 21st International Conference on Requirements Engineering*, pages 125–134. IEEE, July 2013.

[18] A. W. Rivadeneira, D. M. Gruen, and M. J. Muller. Getting our head in the clouds: toward evaluation studies of tagclouds. In *Proceedings of CHI 2007*, pages 995–998, 2007.

[19] K. Schneider, S. Meyer, M. Peters, F. Schliephacke, J. Mörschbach, and L. Aguirre. Feedback in context: Supporting the evolution of IT-ecosystems. In M. Ali Babar, M. Vierimaa, and M. Oivo, editors, *Product-Focused Software Process Improvement*, volume 6156, pages 191–205. Springer Berlin Heidelberg, 2010.

[20] N. Seyff, F. Graf, and N. Maiden. Using Mobile RE Tools to Give End-Users Their Own Voice. *2010 18th IEEE International Requirements Engineering Conference*, pages 37–46, Sept. 2010.

[21] J. Sinclair and M. Cardew-Hall. The folksonomy tag cloud: when is it useful? *Journal of Information Science*, 34(1):15–29, May 2007.

[22] M.-A. Storey, L.-T. Cheng, I. Bull, and P. Rigby. Shared waypoints and social tagging to support collaboration in software development. *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work - CSCW '06*, pages 195–198, 2006.

[23] M.-A. Storey, J. Ryall, J. Singer, D. Myers, L.-T. Cheng, and M. Muller. How Software Developers Use Tagging to Support Reminding and Refinding. *IEEE Transactions on Software Engineering*, 35(4):470–483, July 2009.

[24] M.-A. Storey, C. Treude, A. van Deursen, and L.-T. Cheng. The impact of social media on software engineering practices and tools. In *Proceedings of the FSE/SDP workshop on Future of software engineering research - FoSER '10*, pages 359–364, New York, New York, USA, 2010. ACM Press.

[25] C. Treude and M.-A. Storey. How tagging helps bridge the gap between social and technical aspects in software development. In *ICSE 09: Proceedings of the 2009 IEEE 31st international Conference on Software Engineering*, pages 12–22, 2009.

[26] C. Treude and M.-A. Storey. Bridging lightweight and heavyweight task organization: the role of tags in adopting new task categories. *In ICSE 10: Proceedings of the 32nd ACM/IEEE international Conference on Software Engineering*, 2:231–234, 2010.

[27] C. Treude and M.-A. Storey. The implications of how we tag software artifacts: exploring different schemata and metadata for tags. In *Proceedings of the 1st Workshop on Web 2.0 for Software Engineering*, pages 12–13, 2010.

[28] C. Treude and M.-A. Storey. Work Item Tagging: Communicating Concerns in Collaborative Software Development. *IEEE Transactions on Software Engineering*, 38(1):19–34, Jan. 2012.