Consensus Building in Distributed Technical Discussions

Roshanak Zilouchian Moghaddam, Brian P. Bailey, and Wai-Tat Fu
Department of Computer Science
University of Illinois
Urbana, IL 61801
{rzilouc2, bpbailey, wfu}@illinois.edu

ABSTRACT
The issue management systems in open source software projects contain different categories of discussions, but they all share the goal of reaching consensus on solution proposals. In this paper, we examine the consensus building process in distributed discussions of *technical* issues in one mature open source software community. Our analysis shows that providing a concrete solution in the form of a patch implementation is most predictive of reaching consensus in *technical* discussions. This is in sharp contrast to prior work on consensus building in distributed *UI design* discussions which showed that having participants with more experiences and prior interaction histories are more predictive of reaching consensus. Our results highlight that consensus building depends on the nature of the issue. In *technical* issues that tend to be more driven by objective evaluations providing more patches promotes consensus. However, in *UI design* issues that tend to be more driven by subjective evaluations, having more experience participants helps discussions to reach consensus.

Author Keywords
Consensus; technical discussions; open source software;

INTRODUCTION
In Open Source Software (OSS) projects, resolving software issues (bugs) through distributed discussions is a key activity in the development process. An important but often ignored aspect of issue resolution is reaching consensus on a solution proposal. Reaching consensus on a proposal both supports its integration into the product distribution and enhances participants feeling of valued contribution. Due to its impact, studying consensus building in distributed OSS discussions is of significant importance.

Prior work has studied the consensus building process in distributed *UI design* discussions [12]. However, different categories of discussions exist in the issue management systems of OSS projects that vary in terms of both content and process [8]. For instance, in terms of software component, discussions can be categorized into two main categories: discussions dealing with the software core (*technical* discussions) and discussions dealing with the graphical user interface (*UI design* discussions).

The nature of the task performed in these categories usually differs. For example, it is commonly believed that *UI design* discussions rely more on subjective evaluations of the artifacts (e.g. design ideas), but *technical* discussions rely more on objective evaluations. Studies of group decision making have shown that the nature of the problem has a large effect on group decision behavior and outcomes [6, 7]. Therefore, it's imperative to examine and compare consensus building in different categories of distributed discussions.

In this paper we analyzed a large corpus of interaction data collected from the Drupal OSS community to test how different elements of the *technical* discussion affect consensus. The main result from this analysis shows that *technical* discussions with higher number of software patches are more likely to reach consensus. This outcome differs from prior analysis of *UI design* discussions where it was found that prior interaction history and experience, but not patches, were predictive of consensus [12].

RELATED WORK
Consensus is defined as willingness to commit to a proposal despite the fact that objections may remain [1]. Making a consensus-based decision will increase participants' understanding of the issues involved, allow the participants to explore various solution proposals, build trust between participants, and encourage participants to be more committed [9]. Due to these advantages, consensus building is widely used in collaborative problem solving efforts that deal with complex problems (e.g. distributed collaborative software design and development) [11].

Recognizing the importance of consensus building, researchers have studied consensus building in OSS discussions. For example, the consensus building process in distributed *UI design discussion* was studied in [12]. They analyzed how metrics related to content, process, and user relationships correlate with reaching consensus using a binary (consensus or not) logistic regression. The main result from the analysis showed that discussions having participants with *more experience* and *prior interaction history* are more likely to reach consensus. Our work builds upon and extends this line of work by providing a comprehensive analysis of consensus building in *technical* discussions and shedding more light on the differences of *technical* discussions and *UI design* discussions from the perspective of consensus building. This extension is necessary since prior work has shown that task type is among the factors affecting decision behavior and outcome [6, 7], therefore the nature of consensus building should be

different in discussions around performing technical tasks and designing a UI.

Another related thread of research has employed lab experiments to show it is more difficult and more time consuming to reach consensus when groups use computer mediated technology than face-to-face [6, 10]. Similarly controlled studies examining the effect of group composition factors in FTF condition have shown that it is more difficult to reach consensus as group size increases [5]. They have also suggested that social interaction among group members can promote consensus [3] and groups whose members had considerable experience working together are superior in decision making than ones with a brief history [4]. However, in lab experiments, it is difficult to simulate how consensus unfolds in real world discussions, especially those in mature online social production communities such as OSS.

METHODOLOGY

Through collecting and analyzing a large corpus of interaction data, this research aimed to shed more light on the differences between distributed technical and UI design discussions for consensus building and identify opportunities for enhancing participants' experience as well as the discussion interface.

Characteristic	Type	Mean	SD
Discussion dur. (weeks)	C	44.59	58.89
	NC	88.76	66.82
Number of comments	C	44.28	49.58
	NC	45.05	52.40
Number of participants	C	11.30	9.78
	NC	12.36	9.57

Table 1. Summary statistics for the consensus and non-consensus performance discussions. 'C' indicates consensus and 'NC' indicates non-consensus discussions.

Data Collection

The interaction data was extracted from the discussion threads (discussions) in the issue management system of Drupal. There were 43, 848 discussions in the issue management system at the time of data collection. From the *technical* discussions regarding performance, security, API changes, etc. we extracted the discussions tagged with "Performance" (415 discussions) as a sample of *technical* discussions. These discussions occurred between March 2004 and May 2012.

Similar to [12] we used the status of the discussions to categorize discussions as consensus, non-consensus, or ongoing (unclear if consensus has been reached) and then filtered the data set to only include discussions that have at least seven comments. Table 1 reports summary statistics for the consensus and non-consensus performance discussions after filtering.

From the total of 415 technical discussions we extracted for analysis, 43% did not reach consensus. This outcome is similar to percentage of non-consensus UI design discussion (42%) [12] and indicates a need to promote consensus in technical discussions as well as UI design discussions.

	Metrics	F1	F2	F3
Content	(1) Total # of words *	.94	-.17	.02
	(2) # of "benchmarks"s *	.48	.03	-.06
	(3) # of "summary"s *	.72	-.06	.15
	(4) # of "code review"s *	.81	-.14	-.11
	(5) # of non-Drupal links	.61	-.19	-.06
	(6) # of "IRC"s *	.70	.00	.09
	(7) # of "?"s *	.89	-.21	-.03
Process	(8) Duration of the thread	.34	-.24	-.69
	(9) # of comments *	.95	-.20	-.10
	(10) # of patches	.85	-.05	-.01
	(11) # of contributors *	.81	-.32	-.24
User Relationships	(12) # of triads in the social graph	.40	-.01	.66
	(13) Avg. page rank score *	-.22	.76	.27
	(14) Avg. participation duration	.03	.80	-.24
	(15) Avg. # of prior comments	-.21	.76	.27
	(16) # of alternate replies *	.86	-.22	.10

Table 2. The second column shows the metrics used in factor analysis and regression analysis. The last three columns show the loadings of metrics in three factors. The metrics marked with (*) were later removed from the analysis to avoid problems with collinearity.

Research Framework

To understand quantitatively which factors correlate with consensus is *technical* discussions, we borrowed the framework developed in [12] for understanding consensus building in *UI design* discussions. Based on user interviews and literature review they uncovered 23 metrics that may relate to consensus in three categories of content, process, and user relationships.

For the purpose of our analysis, we only included a subset of these metrics that were meaningful for performance discussions. For example, participants seldom provide screenshots in technical discussions; therefore we didn't include the number of screenshots. Similarly, in performance discussions participants use benchmarks instead of "usability testing". Table 2 lists the metrics we used for our analysis.

To calculate values for these metrics, we incorporated information from the discussion content, metadata of the discussions, and contributor's Drupal profile.

ANALYSIS

In order to examine how these factors predict whether a consensus is reached in performance discussions, we performed a binary logistic regression. To aid interpretation of the results, we also analyzed thirty of the performance discussion threads that reached consensus.

Binary Logistic Regression

In the regression analysis the dependent variable was whether the discussion reached consensus. In order to determine a set of non-collinear independent variables to use in the regression, the data on the 16 metrics from Table 2 were first subjected to a factor analysis. The analysis resulted in a three-factor model. The loadings of the 16 metrics onto the three factors are shown in Table 2.

	B	Df	Sig.	Exp(B)
F1	-.004	1	.00	.996
F2	.020	1	.00	1.02
Constant	2.84	1	.00	17.18

Table 3. Results of the logistic regression on the factor scores. The model is not a good fit for our data set (X^2=17.515, p=0.03).

The first factor concerns the content of the discussion and the process of participation. 11 metrics contribute significantly to this factor (F1). The second factor represents previous contributions along with participants' experience in the community (F2). The third factor concerns participants' prior interactions with each other and the duration of the discussion thread (F3).

	B	Df	Sig.	Exp(B)
# of patches	.033	1	.04	1.03
Duration of the thread	-.016	1	.00	.984
Avg. # of contributors' prior comments	-.003	1	.00	.997
Constant	2.167	1	.00	8.732

Table 4. Results of the binary logistic regression. The Hosemer-Lemoshow test confirmed the validity of our regression model (X^2=12.366, p=0.136).

We used the weighted sum scores method (cut-off value = 0.45) to calculated factor scores [2] and used the factor scores as independent variables in our regression analysis. We performed binary logistic regression as implemented in SPSS and used step-down regression to identify our partial model (Table 3). To assess the goodness of fit of our model, we performed the Hosmer-Lemoshow test (X^2=17.515, p=0.03). In this test, the model is valid if the p-value is greater than 0.05 and the model is not a good fit otherwise. The Hosmer-Lemoshow test showed that the model was a poor fit for the data.

Since factor scores did not create a good model, we performed a regression analysis on individual metrics listed in Table 2. To reduce problems caused by collinearity of metrics in regression, we only kept two metrics in each factor that were least correlated with each other ($r < 0.4$). Three of these six factors showed significance (Table 4).

To aid interpretation of the results, we analyzed thirty of the performance discussion threads that reached consensus. The discussions were sorted based on the three factors that showed significance in our analysis and ten threads from the top of each of these three lists were reviewed.

Number of patches

Our regression analysis showed that submitting more patches can promote consensus in a discussion thread. Reviewing the discussion threads, we found that patch submissions promote consensus in performance threads that are dealing with complex problems, where multiple solution alternatives have been proposed. Because of the impact of these issues on the final product, participants carefully review each patch and ask for modifications. The patch submitter then submits another patch to address the reviews. Therefore, more number of patches means trying to address more participants' concerns, and getting closer to consensus. For example, one of discussions that contained a lot of patches was regarding a performance issue with the Drupal admin overlay. The overlay had different levels of performance deficiency in different browser and system settings, therefore participants had to patch different alternatives to find one that works for everyone.

Duration of the thread

According to our analysis, discussion threads that were resolved quicker were more likely to come to consensus. A review of these discussion threads showed that some of these threads are reaching consensus faster, because they are dealing with small performance issues, minor code refactorings, or a subset of larger performance issues.

We also found that some of these issues resolve quickly, because the proposed solution doesn't show a performance improvement or even hinders performance; therefore participants conclude that the issue won't fix and close the discussion.

Participants' prior contribution

Our analysis showed that having participants with less prior contribution to Drupal increases the likelihood of reaching consensus. Careful investigation of the discussion threads that reached consensus while their participants had a short history of contribution illustrated why this is happening. We learned that the start date of these threads ranges between December 2006 and January 2009, while the start date of all the performance issues that reached consensus in our dataset ranges between August 2006 and January 2012. The majority of the threads having participants with a short prior history of contribution, started early in the development process (i.e. closer to 2006), therefore

participants in these threads did not have time to build a history in comparison to other discussions given the way we measured history.

We also found that some of the discussions are initiated and actively followed by a less experienced member, because they are dealing with smaller performance issues that do not need a lot of experience and prior activity.

IMPLICATIONS

Our work has implications for community participants and community designers. Our analysis showed that providing more patches helps technical discussions reach consensus. Therefore, if community participants provide a patch when they propose an alternative solution or receive feedback on a submitted patch, the discussions are more likely to reach consensus. Providing patches is especially helpful in complex issues where the patch enables participants to compare and decide between different alternatives.

Our results also revealed a major difference in the consensus building process between UI design and technical discussions. While we showed that in *technical* discussions the number of patches correlate with reaching consensus, prior work showed that *UI design* discussions having participants with more experience and prior interaction history are more likely to reach consensus. In technical discussions patches are helping a solution to gain support, because people are able to objectively evaluate the proposal with the patch (e.g. run performance tests). However in UI design discussions there may not be an objective approach for evaluating solutions therefore proposals are gaining support from arguments provided by experienced members [12]. Participants also rely on the opinions of members whom they had prior interaction when building consensus on a proposed design solution.

At a higher level, our result indicates that if we follow the common definition that consensus building is moving in a direction where there are fewer objections to a proposed solution, then the best approach for reducing the objections depends on the discussion category (e.g. UI design and or technical discussions). This means that the nature of consensus building can be different in discussion types where the topic is different.

The fact that about half (43%) of the discussion threads that we analyzed did not reach consensus indicates a need for techniques to enhance consensus building in technical discussion. For community designers, a key challenge is how to design an interface that promotes consensus in different types of discussions where the nature of task differs. One solution might be to include different modes in the interface that accommodate different discussion types and enable users to activate one or multiple modes in their local interface. For example, in *technical* discussions a mode could be activated that employs visual cues to highlight solution alternatives or code reviews that have not been patched yet. While in *UI design* discussions another mode could be activated that aids participants in inviting experienced members as suggested in [12].

CONLUSION

We studied consensus building in technical discussions occurring in one established open source community. The main outcome from our analysis highlights a major difference in consensus building process between *technical* and *UI design* discussions. Our analysis showed reaching consensus in *technical* discussions demands objective evaluations, therefore submitting more patches that facilitate objective evaluations can promote consensus. Whereas prior work found that *UI design* discussions demand more subjective evaluations from experience members to reach consensus.

REFERENCES

1. Briggs, R. O., Kolfschoten, G. L. and Vreede, G.-J. d. Toward a Theoretical Model of Consensus Building. In *Proc. of the ACIS* 2005.

2. DiStefano, C., Zhu, M. and Mîndrilã, D. Understanding and Using Factor Scores: Considerations for the Applied Researcher. *Practical Assessment, Research & Evaluation*, 14 (2009).

3. Fiol, C. M. Consensus, diversity, and learning in organizations. *Organization Science*, 5, 3 (1994), 403-420.

4. Hall, J. and Williams, M. A. Comparison of Decision-Making Performances in Established and Ad Hoc Groups. *Journal of Personality and Social Psychology*, 3 (1966), 214-222.

5. Hare, A. P. A. Study of Interaction and Consensus in Different Sized Groups. *American Sociological Review*, 17, 3 (1952), 261-267.

6. Hiltz, S. R., Johnson, K. and Turoff, M. Experiments in Group Decision Making Communication Process and Outcome in Face-to-Face Versus Computerized Conferences. *Human Communication Research*, 13 (1986), 225-252.

7. Hirokawa, R. Y. and Poole, M. S. *Communication and group decision making*. Sage, Beverly Hills, CA, 1986.

8. Li, Z., Tan, T., Wang, X., Lu, S., Zhou, Y. and Zhai, C. Have things changed now?: an empirical study of bug characteristics in modern open source software. In *Proc. of the ASID* 2006.

9. Sidaway, R. *Consensus Building*. Scottish National Rural Partnership, Edinburgh, 1998.

10. Straus, S. G. and McGrath, J. Does the Medium Matter? The Interaction of Task Type and Technology on Group Performance and Member Reactions. *Journal of Applied Psychology*, 79 (1994), 87-97.

11. Susskind, L., McKearnan, S. and Thomas-Larmer, J. *The Consensus Building Handbook*. Sage Publications, Thousand Oaks, CA, 1999.

12. Zilouchian Moghaddam, R., Bailey, B. P. and Fu, W. T. Consensus Building in Open Source User Interface Design Discussions. In *Proc. of the CHI* 2012.