Understanding the Evolution On-Line Peer-Support Communities

By Andrew Gorman and Gerhard Fischer

Table of Contents

List of Tables

List of Figures

Introduction

This research compares efficacy of three *peer-support communities*, such as those that exist in SAP development communities and Open Source Software (OSS) communities. It then more deeply explores usage patterns and changes within SDN over time. The analysis shows that extrinsic motivators, such as a point system that was in introduced in August of 2004, did not cause an increase in contributors, but did improve response time and the percentage of threads that were marked as "answered." It also revealed a shift in distribution user roles over time. There was a significant decrease in percentage of users who replied to threads compared to those who only initiated threads (the ration of users asking questions and users providing potential answers). This shows what could be a core user group coalescing as the community matures, but it also highlights the fact that there is an increased burden on each helper. This resulted in design recommendations for reducing the burden on the helper community by 1) increasing the reuse of exiting solutions in the forums and 2) by encouraging user to shift into a helper role.

Development of a Framework for Understanding On-line Peer-Support Communities

This research compares *peer-support communities*, such as those that exist in SAP development communities and Open Source Software (OSS) communities. Software developers who use SAP and OSS libraries are similar in that they both need to learn and apply complex APIs and need to quickly overcome obstacles in order to perform their job. To achieve their goal both have grown to rely heavily on *peer- support communities*.

This research attempts to create a deeper understanding of the effectiveness of social support provided by peers in software development communities from the following perspectives:

1. How responsive are communities to the needs of its members? This is understood by measuring the response rate at which peers answer questions, and the time it takes to get an initial response.
2. How does the peer help process work? This is understood by looking at the length it takes for a question to be answered, the number of messages needed, and the number of members to be involved.
3. How wide is the participation of users and in what kind of roles do they participate? This is understood by examining how distributed are the two major roles: those who asks questions and those who answer questions.
4. What is the impact of explicit reward (point) systems on community behavior? This can be inferred from and examination of the relationship between the three aforementioned perspectives.

This research proposes three metrics—responsiveness, engagement intensity, and role distribution—to measure the effectiveness of peer support in API usage (such as SAP and OSS library systems. The metrics can provide important efficacy measurements to those who develop socio-technical environments for peer-support.

Data Collection and Analysis

Systems studied.

1. **SAP** (http://www.sap.com) provides a wide range of enterprise software solution that include powerful, but complex, customizable frameworks. The application of these frameworks is a highly specialized skilled endeavor.
2. **Apache Commons** (http://commons.apache.org) is an Apache project that creates and maintains a library of reusable Java components that provides common functionality to other Apache projects as well as other projects.
3. **Lucene Java** (http://lucene.apache.org) is a project that creates and maintains a high-performance and full-text search engine library written in Java. The library has been widely reused by a number of high-profile projects, including the Eclipse IDE and Wikipedia.

For SAP we studied the SDN forums, which are the primary means of peer-support. The data collected contains threads from June 12, 2003 to May 6, 2008. NOTE: Because we had not received any SDN data until 3/17/2008, we began developing our own data collection software on 2/10/2008. We continued to develop and use our own data throughout the analysis.

For Apache Commons and Lucene Java we analyzed dedicated mailing lists, which contain data from the very beginning of their projects, for reusing programmers to sharing experience and knowledge about reusing library components. The analysis is based on the archives of the two user mailing lists from Jan 1, 2003 to Nov. 30, 2007.

Table 1 shows the scope of the data collection in number of messages members, messages threads and average messages per thread for the three groups. The numbers of members only counted members who have posted at least one message, not including lurkers [18] who become members but never post any messages.

Table 1: number of messages members, messages threads and average messages per thread

	No. of members	No. of messages	No. of threads	Avg. messages per thread
SDN	**120709**	**2954520**	**683901**	**4.32**
Commons	3347	18516	5648	3.28
Lucene	2586	27742	6538	4.24

Results

Responsiveness.

Table 2 gives a general sense of *responsiveness* as the percentage of threads that get a response from other members. As shown in Table 2, SDN has an 87% Response Rate, meaning that 87% of the threads have received a reply from other members. Commons and Lucene have respective *response rates* of 67% and 80%. These rates indicate that a very high percentage of askers got help, or at least attempted help, from their peers in the online peer support community. All three communities have *response rates* that are higher than the 61% reported by Lakhani et al in their study of the peer support among users of the Apache Web Server [7].

Table 2: Response Rate

	Total No. of threads	No. of "no response" threads		No. of threads with responses	
SDN	**683901**	**92664**	**14%**	**591237**	**86%**
Commons	5648	1870	33%	3778	67%
Lucene	6538	1302	20%	5206	80%

Response Time.

To understand how quickly members received responses, we looked at the difference between the time the thread was created and the time that a message was posted to the thread by a different member. Only threads that received a response were included (N = 591237 or 86% of all threads shown in Table 2).

Table 3 presents the *response time* for the three groups: the first quartile time (Q1) at which 25% questions got the first response, the second quartile time (Q2) or the median time at which 50% questions got the first response, and the third quartile (Q3) time at which 75% questions got the first response. The median *response time* in (Table 3) indicates that half of all SDN threads get a response within 23 minutes. This is in contrast to Commons and Lucene, which have a median response time of 3 hours, 56 minutes and 1 hour, 27 minutes respectively. Table 4 shows the *discussion duration* for the three groups: the first quartile discussion time (Q1) at which 25% were completed, the second quartile time (Q2) or the median time at which 50% threads were completed, and the third quartile (Q3) time at which 75% threads were completed. As can be seen, 25% of SDN threads require 46 minutes or less to complete; 50% are completed in 5 and a half hours or less. These are less than the *discussion durations* for the other two groups.

From Table 3 and Table 4 one can see that SDN requires less time, in terms of getting an initial response and also overall duration of the thread.

Table 3: Response Time

Response Time	Q1 (25%)	Q2 (Median)	Q3 (75%)
SDN	**6 m**	**23 m**	**3 h 10 m**
Commons	49 m	3 h 56 m	14 h 15 m
Lucene	24 m	1 h 27 m	5 h 51 m

Table 4: Discussion Duration

Discussion Duration	Q1 (25%)	Q2 (Median)	Q3 (75%)
SDN	**46 m**	**5 h 30 m**	**2 d 3 h 38 m**
Commons	3 h 02 m	13 h 37 m	2 d 0 h 49 m
Lucene	2 h 59 m	9 h 28 m	32 h 53 m

Effect of group size on response time.

An initial hypothesis for explaining fast response times in SDN was that a greater number of group members would decrease response time. There is a high negative coloration (-.76) between medium response time (

Table 3), and number of members (Table 1), suggesting that more users decrease the time it takes to get a response. However, an analysis of variance (ANOVA) shows *no*

significant effect of group size on response rate. Additionally, because of the small sample size of N=3 groups, we tested the hypothesis within SDN *between all forums* (N=162). An ANOVA between SDN's forums showed *no significant effect* of the *number of members who posted to a forum* and the *mean response time* for that forum. However, a second look at response time shows a more complicated picture. Looking at group size and response time *for all forums together* over time, group size had a significant effect on decreasing response time. This implies differences between the forums that are not seen in the total population. Furthermore, looking at the data at a finer grain shows a significant effect in the few years (months 1 – 29) and no significant effect in the last few years (months 20 – 59). This shows on complex phenomenon that suggest patterns in the evolution of the community, which are not yet well-understood.

Change in role distribution over time.

To explore the evolution of the community, we looked at the distribution of roles over time. We group users into two groups: those that ask questions (post threads) and those that answer questions (reply to threads). There was a dramatic change role distribution over time, which was not associated with the point system. The percentage of users that reply to threads has significantly decreased over time and this has a significant effect on response time. This is likely to be the result of a core group of helpers coalescing over time. Although a smaller percentage of helpers are required to support community, each helper is relied on more heavily.

Table 5: Change in Role Distribution Over Time

Year	Asker Count	Helper Count	Percent Helpers
June-03 – May-04	2963	3781	62%
June-04 – May-05	12584	13665	53%
June-05 – May-06	21875	18977	52%
June-06 – May-07	99472	73315	43%
June-07 – May-08	133648	75205	33%
Total	**270538**	**184936**	**41%**

Effect of thread helper count on answer rate.

Although an increase in the number of helpers participating in a thread can be accompanied by an increase in the likelihood that the question will be marked as answered, Figure 1 shows that after a point this likelihood actually decreases. This suggests a "Too many cooks in the kitchen" effect. It is possible that the thread becomes "hijacked" into discursive discussion rather than focused on answering the question.

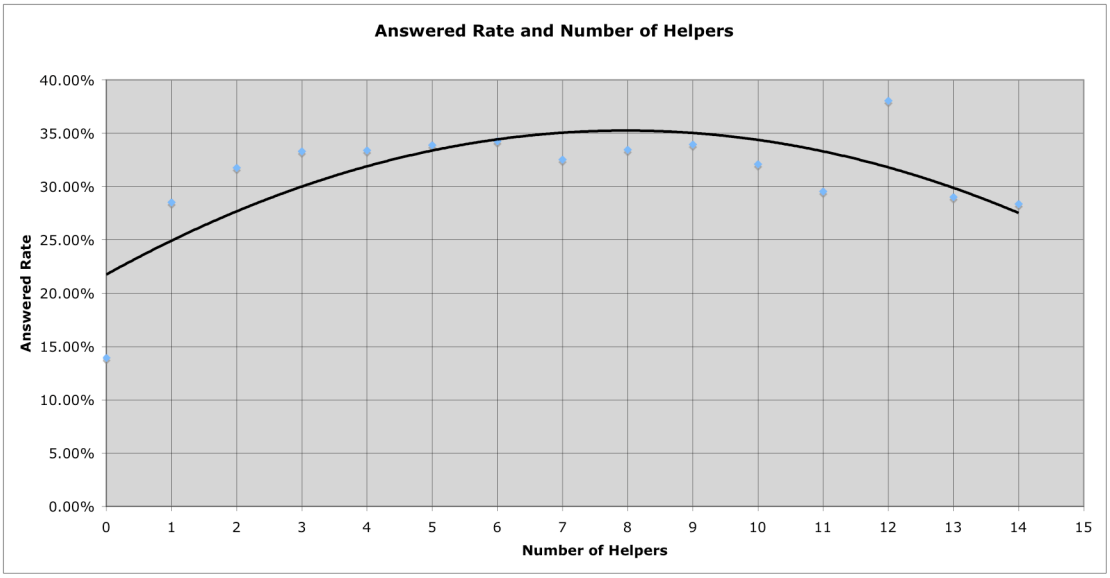

Figure 1: Percentage of Answered Threads by Helper Count

As shown in Table 6, only 0.25% of the threads have more than 9 helpers (about the point where answered rates decline). It is possible that these threads represent either complex questions with no clear answer, which could provide insight into tough problems being experienced by members or discussions for members to express their opinions, which could provide insight into the future needs and desires of the members. A qualitative analysis of a sample of these threads is required to better understand this phenomenon.

Table 6: Helper Counts

Helper Count	Frequency	Relative Frequency	Cumulative Relative Frequency
0	92752	13.56%	13.56%
1	213314	31.19%	44.75%
2	169879	24.84%	69.58%
3	101783	14.88%	84.46%
4	53884	7.88%	92.34%
5	26822	3.92%	96.26%
6	13047	1.91%	98.17%
7	6258	0.91%	99.09%
8	3091	0.45%	99.54%
9	1440	0.21%	99.75%
10	810	0.12%	99.87%
11	457	0.07%	99.93%
12	247	0.04%	99.97%
13	131	0.02%	99.99%
14	74	0.01%	100.00%

Effect of the SDN Point System.

In August of 2004, SAP implemented a point system in which members who post threads can award points to members who help the most. This was to motivate members to contribute. We hypothesize that the point system has a significant impact on the knowledge-exchange behavior: members would be more eager and willing to answer

questions. To test this hypothesis, a t-test was performed on the SDN forums. Pre and post groups were created by selecting all forum threads (that received a response) from the three months before and after the introduction of the point system. A Welch Two Sample t-test showed a significant difference between the two groups. t-tests were also performed for *helper count* (number of users who contributed to a thread), *percent answered* (whether the thread was marked by the as answered by the user who initiated the question), and *discussion duration* (see Table 7). Significant differences were found for *helper count and percent answered*: there were a significantly greater number of helpers and percentage of answered threads after the point system was introduced. At the same time, there was no significant change in *discussion duration*, which is important because it shows that users experienced better results without requiring more time.

Table 7: Welch Two Sample t-test Result Summary

Group	Threads in the three month Prior to Point System N=3766	Threads in the three month After to Point System N=5963
Mean Response Time p-value < 0.01 ***	51 (min.)	34 (min.)
Helper Count p-value < 0.01 ***	1.892378	2.019339
Percent Answered p-value < 0.01 ***	12%	30%
Discussion Duration p-value = 0.1164 (not significantly different)	162 (min.)	149 (min.)

Positive and negative impacts of point system.

By scanning the *content* of SDN threads for topics related to the point system, it is clear that the point system is a very strong motivator. We speculate that some SAP developer consulting firms link point awards with job performance measures. High point scorers, for individual members and grouped by company, are shown for a rolling three-month period. Consulting firms can use this distinction to attract customers in a competitive market. The high stakes in the point system is seen in the following types of examples:

1. helpers remind askers to reward points;
2. helpers complain if no points are awarded;
3. users cheating to gain points; and
4. other users detect and report cheaters.

Faster doesn't always mean better.

While SDN has faster response rates after the point system was introduced, it also requires more members to answer the questions. From a productivity standpoint, this can have drawbacks. There appears to be a "race" condition in a competition to score points. Helpers may be less motivated by the desire to help someone solve a problem. Instead, their motivation is to score points. In this case motivation to help solve a problem and motivation to reply to a question could be two different things.

Design Implications of the SDN Analysis

Turn questions into search queries. A simple approach that could reduce the burden on the core helper community is to add a small step to the question submission process. Often helpers complain that askers have not "done their homework" by searching the forum before asking a question. This is seen in responses that point askers to other threads that have already resolved the question. Coincidently, asking questions that have already been asked is also a method for cheating. Turning questions into search queries could also help automatically detect cheaters—similar to techniques used to detect plagiarism in academic settings.

Provide links between related threads. Another approach, which is a variation of turning questions into queries, is to provide links to related threads. Thread pages can simply provide links to the 5 most closely related threads. This has the advantage of providing two benefits. It benefits the asker, who posts their question and sees existing related threads. It also benefits a searcher, who finds a partial answer to their question by searching the forum, but also finds other threads that are most closely related. This can be seen as a form of refining a query: the user performs a search, selects a search result that seems to fit their question, and automatically sees additional related links that could be different then those showing up in the original search.

Push questions to willing helpers. Table 3 and Table 4 in show that while 75% of threads have a response time of less than 3 hours, 10 minutes, 25% of the threads also have a duration of more than 2 days. This suggests that initial responses do not always adequately answer the question. By sending questions to select users, it might be possible to increase response effectiveness. It can also increase role distribution (the ration of members who ask questions to the number of members who answer questions). One approach could be to send questions to 1) users who have *answered similar questions*, thus increasing effectiveness and 2) users who have *had similar questions answered for them*, thus increasing role distribution by encouraging askers to become helpers.

Conclusion

The analysis shows that the point system that was in introduced in August of 2004 did not cause an increase in contributors, but did improve response time and the percentage of threads that were marked as "answered." It also revealed a changing distribution of user roles as the community evolved. There was a significant decrease in percentage of users who replied to threads compared to those who only initiated. This shows a core user group emerging as the community matures, but it also highlights the fact that there is an increasing burden on each helper. These findings resulted in design recommendations for reducing the burden on the helper community by 1) increasing the reuse of exiting solutions in the forums and 2) by encouraging user to shift into a helper role.