

# Open Source Technical Support: A Look at Peer Help-Giving

Vandana Singh, Michael B. Twidale, Dinesh Rathi  
*Graduate School of Library and Information Science,  
University of Illinois at Urbana-Champaign, IL, USA*  
*vsingh1, twidale, drathi@uiuc.edu*

## Abstract

*We explore online technical support of open source software by a study of postings to discussion boards. Our results indicate that there are several types of detail that are required by the help-givers to be able to diagnose and remediate help-seekers' difficulties. As a result help interactions may iterate somewhat inefficiently. These findings are compared with studies of telephone technical help lines for commercial software, and library reference interviews. By considering certain rather problematic interactions we can identify ways to improve the process.*

## 1. Introduction

For any technology to be adopted it needs to be usable and useful to the people considering adoption, and have clear advantages over alternatives, including those currently being used. If Open Source Software (OSS) is to widen its appeal beyond very technically adept user-developers, this requires a consideration of aspects of functionality, interface and wider support. If we consider a non-technical computer user, who may be very skilled in other domains but lacks skills and indeed interest in programming, the great appeal of open source software (than one can adapt the code and share one's changes with one's peers, hopefully leading to even further improvements to one's original ideas) is irrelevant. What is more likely to be of concern is whether the software does what the user wants (rather than esoteric features only of interest to a power user), how hard it is to learn and to use, how reliable it is and what kind of help is available when things go wrong. All of these might be informally aggregated into a qualitative total cost of ownership computation, measured in effort, mental anguish or dollars. The fact that the software itself is free does not mean that this total cost of ownership is zero.

In other work [25, 26, 34] we have looked at the usability of OSS, why there might be barriers to usability, how usability issues are discussed within OSS projects, and what can be done to improve OSS interfaces. In this paper we begin to explore technical support as another

critical aspect of the extended interface [22]. One approach is to sell such support as a commercial extended interface wrapper around an OSS product. Commercial Linux distributions do just that with a mixture of better interface design to support the rather complex installation process, and various kinds of technical help in the form of manuals, online help, FAQs, and telephone help lines. The O'Reilly series of books explaining how to use different OSS products is another example using the traditional medium of print. In this paper we look at one kind of support that is tightly linked to the OSS ethos – online peer support provided by community members.

OSS project websites contain several resources such as a place to download the latest versions of the software, and mechanisms for reporting bugs and participating in development. There is also usually a support section. It typically includes discussion forums or mailing lists to which users can post questions and get help from developers or other users. These forums are unrestricted and act as a learning environment for novices and experts alike.

We wanted to understand the nature of the discussions in these forums. In particular we believe that an examination of the more complex and problematic help-giving interactions may guide an exploration of ways in which those interactions might be improved.

## 2. Related Work

The OSS technical support bulletin boards that we studied are asynchronous, remote, mostly text-based, and provided by volunteers rather than paid experts. As such they have similarities and differences from the OSS development process itself and from other kinds of help-giving, each of which has its own research literature.

### 2.1 Open Source Software Development Studies

Research in OSS has examined many issues including the evolution of the developer community [6, 24], developers' motivations [18], adoption in commercial businesses [29, 36, 38], the culture of developers [9], policy issues [32], and problem solving [14]. There have

been only a limited number of multiple-project studies [9, 32], reflecting the early stage of OSS research, the complexity of the phenomenon and the difficulty in obtaining comparable data across projects.

Missing from almost all the OSS literature that we are aware of is much consideration of end users and their problem solving approaches. The main exceptions are papers considering the usability of OSS (see [25] for a review of that literature), but even there end users are mostly notable for the way that they are an absence from processes and discussions that is to be remarked upon, rather than a presence whose activity is studied. As noted by Stewart [33] “The least explored areas of OSS success included in the model are user impact and the role of user factors.....users may vary in their ability to use and benefit from OSS based on factors such as their technical expertise.” Maass [23] sums up the present status of OSS field research as “most empirical studies on OSS communities concentrate on secondary logging information such as that provided by mailing lists, IRC chat logs and code repositories.”

The paper by Lakhani and von Hippel [19] is the only one we are aware of that looks at technical help for OSS, but it is mostly concerned with motivation, trying to answer *why* people bother to help others, and not really examining *how* the help unfolds, which is our main interest. Although widely cited, it seems that all those citing papers are also concerned with the motivation problem rather than the process of help-giving. One reason for this lack of research may be that technical help is considered either trivial, or just not interesting enough to merit attention, other than amazement that anyone who wasn't being paid would bother to do it. Indeed Lakhani and von Hippel call technical help a “mundane but necessary task” [19]. Nevertheless, we do think it merits attention, since it can play a critical role in facilitating adoption, and perhaps can be improved to minimize the burden it imposes on participants. We want to study what happens when users who are not developers or programmers download an OSS product to their desktop and start using it. What are the problems that they face and what are the options of help (if any) available to them? In particular, how do users as help-seekers collaborate with others to collectively solve their problems, and share their findings with others? This last issue is clearly very similar to the OSS development process, even if it does not involve contributing code.

## 2.2 Telephone support for commercial software

Pentland, [27] amongst others has studied telephone technical support. Those calling commercial software telephone help lines will have very similar needs to those of users of OSS. The help-seeker has a problem with the software, and help-giving involves a series of interactions

to determine the details of the underlying need, the symptoms of the problem, what the user may have already tried, diagnosis of the problem and any connected problems, and the determination and execution of candidate solutions. The main differences are the nature of the help-givers and the medium of the interaction.

Commercial software technical support is provided by people who get paid for the support that they provide. Usually this is a full time job, and standard protocols have evolved. These can aid in the efficiency of the process (important for cost minimization in a commercial help center), and their development can be discounted over a large volume of calls, as can training of the staff. By contrast, OSS technical support is provided by volunteers who are themselves users of the software and some of whom may also be involved in the development of the software, providing an invaluable design feedback loop. Even the most enthusiastic help-giver is only going to spend a small fraction of their time on help-giving, and although frequently having substantial technical expertise, may not be skilled in the best ways to give help. (Of course commercial technical help-giving is not always ideal, and can involve interactions with personnel who seem to be very inexperienced.) The different media of the interaction (a live telephone conversation versus an asynchronous series of textual messages within a bulletin board) also change the nature of the interaction somewhat, although the consequences need further work to fully understand. On a basic level, a telephone conversation is typically less effort for the help-seeker, and allows ambiguities to be identified, queried, and rectified much more easily than with textual interactions. On the other hand, a carefully written textual request with appropriate background is particularly easy to read and address for the help-giver. Although considerable work has been done on telephone help lines [5, 10, 27], they mostly focus on aspects of the work other than the help-giving interaction such as information sharing between the help-giving team [13] and re-use of solutions by creating FAQs etc., either for end users or for the professional help-givers [15].

## 2.3 Knowledge Management Systems and Communities of Practice

Knowledge management systems (KMS) have a variety of implementations [11] including document repositories, expertise databases, discussion lists, and context-specific retrieval systems incorporating collaborative filtering technologies. Those KMS where users can pose questions and discover answers are most similar to our data set of OSS discussion lists. In that sub-area, most of the work has involved the use of collaborative filtering methods to propose Usenet news posts to users [17, 31]. There have been a number of

studies of the visualization of discussion forums to represent patterns in conversations [12, 38]. These show who starts discussions, who provides responses to questions, and how active the discussion is. Answer Garden, an organizational memory system, archives answers to frequently asked questions to reduce the overload of experts [2].

Wenger [37] described the concept of Communities of Practice (CoP) as self-organizing systems that evolve over time around a specific issue or a problem, have their own set of rules and regulations (informal or formal) and are managed by the members. The members determine the goals and objectives of the community and hence the CoP is a reflection of the understanding of its members. They evolve around common interests and the quest to know more or share knowledge on some common issue. Membership is not bound by any organization but by participation and hence a community of practice has no geographic boundaries. The members' interaction is what determines the existence of a community of practice. The members share knowledge in a free flowing manner and contribute to each other's problem solutions. The main attributes of a CoP are common problems, practices and languages, common sense of purpose, common objectives of learning, sharing and knowledge creation and an unwritten commitment to the community.

The participants in an OSS help bulletin board may or may not constitute a CoP. They are somewhat like an online CoP in that there is a common theme, purpose and sharing objective. The main difference is that a CoP is a group of people who through a longer time investment in the community develop stronger ties like a common parlance. Much OSS help can be very ephemeral, so the participants may not be really in a CoP as it fits Wenger's definition. We do not have any evidence that people use OSS discussion lists for a long period of time or if it is something that they only visit when they have a problem. Unlike discussion forums, CoPs are not just about questions and answers. They are more about learning and developing community knowledge.

## 2.4 Over The Shoulder Learning

Our interest in OSS technical support was in part inspired by earlier work on informal face to face help-giving between peers in the workplace, an interaction we termed Over The Shoulder Learning [35]. A recurrent question arising from this study of the nature of co-located help-giving was its implications for remote technical help. Among the findings that are of relevance in this new setting were that peer help in the workplace enabled participants to exploit considerable local contextual knowledge. This allowed help-seekers to phrase requests in terms of the underlying work goals, rather than purely in terms of an application's

functionality. As a result, solutions or workarounds could be constructed that involved a completely different application, or a combination of applications, frequently coordinated by simple copying and pasting – a mechanism that seemed to be used effectively even by those who did not consider themselves to be computer experts. The ability to share the same screen helped to determine and maintain a shared context for complex conversations about what was wanted, what had been tried, what might be tried and how to perform a complex sequence of steps. This was found to be a major component of successful interactions, particularly with novices who may lack a sophisticated technical vocabulary even to talk about the nature of their problem. Even with these substantial advantages of co-located synchronous interactions, help giving episodes could still be complex and problematic, since systems are rarely designed to actively support this kind of interaction. Hence we were curious about how help-giving is managed in the much more challenging setting of remote asynchronous interactions between people who do not share the same work context.

## 2.5 The Reference Interview

Another kind of help-giving that has a number of interesting parallels with technical help-giving is the reference interview between a patron and a librarian. This particular interaction has been the subject of analysis for many years [7] and the skills of how to conduct a successful yet efficient reference interview have been systematized and taught to generations of students of schools of library and information science.

The Reference Interview is a means of determining information needs and has four basic steps: outreach, determining needs, filling needs, and follow-up. Outreach concerns techniques to ensure that the librarian is considered to be easily approachable. The determination of the patron's underlying information need is perhaps the most critical step and one where errors are most likely to arise. Essentially this is because the patron may not fully be able to articulate their need and the librarian may misinterpret it. Techniques to address this problem include the use of open rather than closed questions, paraphrasing the request to check for shared understanding, clarificatory sub-questions, and other verification moves. Filling needs is about using available resources within the constraints of the context, and follow-up is a reminder to check that the information needs have really been met.

Asynchronous reference is most similar to an OSS discussion forum. Pomerantz et al. [28] describe digital reference as containing five steps: *Question Acquisition* details the process of obtaining question related information from a user. *Triage* is when the question is

assigned to a reference or subject expert. This could be automated or conducted via humans and also filters out repeated or out-of-scope questions. *Answer Formulation* covers the steps taken by the expert to produce the answer up to sending it to the user. *Tracking* is the quantitative and qualitative monitoring of repeat questions for trends. *Resource Creation* works as a feedback loop where data from tracking informs the librarians to develop their collection according to users need.

As with commercial technical help, a major difference between a reference interview and OSS support discussions is the expertise of the help-giver. The reference librarian is an expert in the help-giving process as well as the content. The nature of the resultant help is of course also different, typically pointers to textual resources (books, papers etc.) rather than how-to instructions for interaction with a piece of software. Conventionally reference interviews have been face to face and so can exploit the richness of voice, gesture, and facial expression, as well as ease of pointing to resources (paper or computer based) to help clarify complex conceptual ideas. However telephone reference has been part of the culture for decades and computer based remote reference (using email and instant messaging) is a focus of substantial research and practice [1, 16, 21].

### 3. Studying Help

The related work summarized above reinforces the challenge of online OSS technical support by peers. All of the other areas benefit from various richer contexts, such as full time professional help-givers, same-time same-place local knowledge, or extended membership of an ongoing community of practice, including the CoP of OSS developers themselves. How can OSS technical support operate under far less context-rich conditions? By considering the help-giving activities that occur, particularly those that look more problematic, can we begin to consider how to improve the helping process? Can those other research literatures setting help inform analysis and redesign?. A preliminary study was undertaken to begin addressing these questions, by trying to understand the nature of OSS help-giving interactions. Seven OSS websites were chosen (Table 1). They include text based as well as Graphical User Interface (GUI) based forums. All were publicly accessible.

#### 3.1 Data Collection Criteria

Given the aim to understand the nature of help-giving, we wanted to only consider interactions on that topic, rather than others such as a new version release, or discussions related to group members and not to the

software. A search with the keywords “help” and “problem” helped narrow down some suitable threads. We were particularly interested in the more complex interactions on the assumption that these are more problematic and so in most need of analysis and approaches to improve them. Consequently only discussions that had a thread of at least five email messages were selected. To ensure we were looking only at peer support, cases where the problem solvers were the moderators or appointed experts were excluded. Finally, we wanted completed discussions, and so only selected threads where the last posting was at least a month old. Given those criteria, finding a problem to include meant rejecting on average four others, so it should be noted that our small sample is itself a sample of only 20% of all postings on the bulletin boards. We chose 10 problems from five sites and 15 problems from two others that happened to have particularly rich sets of data.

Project	Number	Website
Opera	15	www.opera.com
Greenstone	15	www.greenstone.org
Gnome	10	www.gnome.org
Mandrake	10	www.mandrakeuser.org
KDE	10	www.kde.org
BCS	10	www.bcs.org
Mozilla	10	www.mozilla.org
Total	80	

**Table 1. List of websites for data collection**

#### 3.2 Data Analysis

Grounded Theory [8] was used as a basis for data analysis, an approach that has been used in other OSS and online content analyses [20, 30]. In the process of grounded theory development, theory generation and development is done inductively by studying the phenomenon it represents. Concepts are discovered, developed, and provisionally verified through systematic data collection and analysis. One does not begin with a theory, then prove it. Rather, one begins with an area of study and what is relevant to that area is allowed to emerge. All the problem interactions were read and re-read to develop some basic concepts, producing 30 categories. These were then further grouped into five broader categories. Given the small size of the study and its limited focus noted above, our findings are tentative. Also our categories are still somewhat low-level. We expect to be able to develop richer but fewer key categories as our work continues. All the broader and specific categories and one off cases are presented in Table 2. We note the categories, provide an example from the data set and discuss them.

Broader Category	Specific Category	Occurrences Out of 80
------------------	-------------------	-----------------------

Types of Questions		
	How to	34
	What is wrong	12
	Error, stuck	7
	Where can I find a function/file	5
	Extra question – not related to the thread	8
More details needed		
	System details needed	14
	More details of history	7
	More problem details needed	6
	More details needed of what is on screen	2
Responses		
	Exact step by step solution provided	16
	Multiple answers for one problem by one person	13
	Exact codes given to copy and paste	15
	This works for me	13
	Detailed description of what needs to be done	11
	Speculation about what the help-seeker might have done wrong	7
	One line exact answer	5
	Series of questions to know what the help-seeker did	5
Other recurring themes		
	Pointer to alternate resource – tutorial, manual, website, image, email, another thread , article, tool	12
	Alternative media – email and attachments, msn, image attached	12
	Detailed description of what has been done	12
	I had the same problem	9
	Detailed description of problem provided	8
	Troubleshooting – do this then this and tell me	6
	If X happened Y will happen description	5
	Use of text for GUI description	5
One off Cases		
	Explanation give for both command line and GUI	1
	Typo leading to a wrong solution, corrected later	1
	Tips – without a question being asked	1
	Solution to something that was not asked	1

**Table 2. Preliminary results of the study with the number of instances of occurrence of each category**

### 3.3 Types of questions asked

a) **“How to” questions:** Various types of questions are asked on the discussion forums. The most common kind was what we call “How to” questions. These were asked by users when they did not know how to use a particular feature of the software they were using or when they were not able to complete a task that they were trying to accomplish. These questions had a simple format, usually consisting of one or two lines.

#### Example

*How do i put my old MIE bookmarks into Firefox?*

b) **“What is wrong” questions:** The second most common question type; these questions were asked when the help-seeker thought that she was doing everything right and still she cannot get her work done. The user

thought that she knew what she wanted to do and how to do it but she was not getting the desired results. In these questions comparatively more details were provided and they had a format in which the help-seeker said what she was doing, what she did and then what happened.

#### Example

*I use SuSE 9.1 and upgraded to kde 3.3, and when I login the system tray says that it is empty, but when I remove it and put it back it is full of stuff like it is supposed to. I tried renaming ~/.kde to ~/.kde~ and it still didn't work. What is wrong?*

c) **“I am stuck” questions:** These were questions where the user was unable to go ahead with his activities because of a computer problem. He had tried to solve the problem, but was stuck somewhere with no idea of the

next step to take. The help-seeker could not proceed with his troubleshooting skills and was at a loss as to why his system is not doing what he wanted it to do.

#### **Example**

*Hi, I am running KDE 3.2.3-1.0.1 (from kde-redhat) on Fedora Core 1. One user has recently complained of being unable to login. When he enters his username and password into KDM, the screen goes blank briefly and then the login screen reappears. He could log on earlier in the day, and claims to have changed nothing. Other users can still logon. I have checked that he has a proper shell (/bin/bash), and tried deleting ~/.kde, /tmp/orbit~, /tmp/mcop~, /tmp/ksocket~, /tmp/kde~, /tmp/gconfd~ and /var/tmp/kdecache~. What else could be the problem, and how can I fix it?*

The “how to” questions most often did not have answers that were as simple and initiated a longer question response sequence. Since the details provided in such questions were minimal, they took a long time to be completely understood and finally solved. Most of the times the responses to these questions were looking for more details, the details required ranged from the details of the operating system, what you see on screen, what have you already tried, what task you are working on, etc.

### **3.4 More details needed for resolution**

**a) System details needed:** These were posted by help-givers when the information provided by the help-seeker was not complete enough to judge what the problem was. The diagnosis or solution was dependent on the type of the system that the help-seeker was using. System details sought by help-givers included the user’s operating system, version of the software installed and other system setup-related questions.

#### **Example**

*Hello, We can almost certainly help you with the problem, but not with the amount of information you have given us about it. Useful information you could give us would include:*

- (1) the operating system you are using (Windows Me/ Windows 95/ Linux/ MacOS/ etc)*
- (2) some indication of the nature of the data you're trying to import including*
  - (a) the file format (doc/pdf/ps/bib/html/avi/etc)*
  - (b) the language of the text in the documents (English/ German/ Chinese/ etc) and*
  - (c) whether the data is in a few large files or many small files*
- (3) the version of the (software name) you are using*
- (4) the command line you are using*
- (5) the last line that appears on the screen before it hangs*

There were many instances when the answer took multiple iterations and indeed it took a considerable time

### **3.5 Multiple solutions given for one problem**

for the help-giver to know just the basics of the help-seeker’s system. The example above is exceptional in showing very clearly what information is needed. In some instances the question came in a more terse form like “What are the details of your system?” which can lead to incomplete details, yet more iterations and hence longer time in problem solving. The operating systems and software details are needed in most of the cases to understand what is going on.

If there were an easy way to provide this information upfront to the help-givers it would improve the efficiency of the whole process. Possible solutions can be automation or keeping that information in the help-seeker’s website profile so it is accessible from all subsequent posts.

**b) More details of history:** These were questions regarding the activities of the help-seeker before encountering the problem. The help-givers in these cases wanted to know what actions the help-seeker had taken that led her to the problem. Also they wanted to know details about the actions that the help-seeker had already attempted before posting the question, what the results she got, and how she interpreted those results.

#### **Example**

*Some more details would help.*

*When you say "About 25%", what does that mean in real terms? One screenful, perhaps? One paragraph?*

*Next question, how are you pasting it? Did you select Copy in OoWriter and then Paste in Opera, or are you using the Linux middle-click paste feature?*

*Third question - system details. Version of Linux, desktop that you use, etc. I know that KDE can be somewhat strange about pasting from one app to another, usually clicking on the Klipper icon (that little clipboard on KDE's task manager) and selecting the text to paste generally works.”*

#### **c) More details of the task at hand:**

These were requests for the exact details of the problem, what happened before and after any action and exactly what the user was trying to do.

#### **Example**

Same as for 2 (a)

#### **d) More details needed of what is seen on screen:**

These were requests for details such as the exact error message on the user’s screen at the time of the problem.

#### **Example**

*This is strange. I say this because KDM runs as root so if you tell KDM to shut down, it has root privileges and it should do it -- it is its responsibility to control who can shut down. Exactly how are you asked for the root password? Is it the KDESU dialog or is it something else?*

Also last line of 2 (a)

Sometimes the help-giver would provide multiple alternative options for a solution.

#### Example

Hi, I can think of a couple of things to try:

1) href=<http://gsdl/collect/science/import/page.html> this won't work as it is looking for an internet server named "gsdl" and then /collect on that server. You could try href="/gsdl/collect/science/import/page.html" but I don't think it's a good idea to link to the import directory. Greenstone can handle internal links...

2) If you really want to give hard-coded links, edit your collect.cfg file so that for HTMLPlug you include a certain option, like: plugin HTMLPlug -nolinks This means that greenstone won't do any interpretation of the links and they will be displayed exactly as they are in the source documents.

3) You could use the "-file\_is\_url" option to HTMLPlug as above. This is normally used when building a collection from a web mirror, so the file might be called [www.example.com/somedirectory/somefile.html](http://www.example.com/somedirectory/somefile.html) etc. Internal links work for collections I've built when I mirrored some of our university pages... I don't know if it will work in your situation though. Let the list know if it does!

These are the types of responses that can be useful to anyone who has the same problem later. They might be presented on the first page of the support forums as examples of well documented help leading to efficient solutions.

### 3.6 Use of alternate media and resources

**a) Pointer to alternate resource:** tutorial, manual, website link, image, email, another thread, article, and tool. These were instances where the help-seeker would be directed towards places where the solution of his problem already exists, like a website, a tutorial or another thread of discussions. These responses can also be saved for future use by others to save time and effort.

#### Example

Hi, I have like one thousand books as html and txt files and metadata in Dublin core format for each book. I want to make a collection but I don't have .txt files (e.g.. sub.txt, org.txt etc). Can you help me to make these .txt file using Collection Organizer or any other method. The help menu of Collection Organizer is not sufficient to me so can you prescribe any other guide for this.

#### Response

To create a basic collection from html and txt files you don't need files like sub.txt and org.txt - these are only necessary for creating specialized browsing structures. More information about this can be found in Section 2.2 of the Greenstone Developer's Guide (available from <http://www.greenstone.org/english/docs.html>). If you haven't already, I suggest you try out the new

"Greenstone Librarian Interface". This is a graphical tool that leads you through the collection building process: gathering files, assigning metadata, formatting your collection, and finally building it. The Librarian Interface also allows you to create browsing hierarchies (generating files like sub.txt and org.txt), should you need to, and comes with extensive help files. Hope this helps.

**b) Alternative media used:** email and attachments, msn messenger, image attached. These were the special cases in which the text description of the problem or solution was not enough and hence for better understanding alternate media was used. Images, screenshots, attachments were used to clarify the points raised in the posting.

#### Example

I want to know how to change the font of the yellow tips, I have changed the "perference"- "font", but no help.



#### Response

if the font is set in css or in the code, the only way i know of overriding the page rendering is to set up your own user css and tell opera to surf in user mode as opposed to author mode. Tools > Preferences > Page Style. i think you need to create your own css manually though.

### 3.7 I had the same problem

This was a response posted by someone who does not have an answer to the problem but who faced the same problem some time ago.

#### Example

Hi I have been using 7.0 with no problems till today. I dl'ed 7.3, followed instructions uninstalled 7.0 from add/delete programs and installed 7.3. Now when i try to start thunderbird it doesnt completely start. In task manager it shows under open processes thunderbird.exe running using 99 percent of my cpu resources bogging down everything else but the window with the program never opens and i have to eventually hit close task to get rid of it. I really like thbird and would hate to go back to outlook but i cant seem to get it to work now...ive

*uninstalled and reinstalled it a couple of times now with no luck*

#### Response 1

*I've been having a similar problem, I recently re installed thunderbird after having upgraded to SP2 and now the program crashes as soon as I click on the icon, it just brings up the Quality Feedback agent or whatever it's called. I've tried deleting the chrome and extensions folder but it's not helping, any ideas?"*

#### Response 2

*Having the same problem*

These were the problems that multiple people were facing. People who were posting the original problem get a validation that it is not something that they are doing wrong, or something wrong with their system but it more of software problem that needs to be fixed or does not have an obvious solution. These recurrences of the same problem led to suggestions to post it onto a developers forum or to check if this problem exists in a bug database.

### **3.8 Pointers to Bugzilla- existing bug**

These are the instances where the particular problem has already been identified as a bug and is in the process of being evaluated, verified or solved. Help-seekers were pointed to Bugzilla where they could follow the bug if they wanted. It would be beneficial to put these as bugs already encountered or bugs already identified so that people do not stumble upon them as a problem that they need to solve.

#### **Example**

*vnc in 9.1 is whacky! i get icewm whereas my config is kde AND i cant run any kde based programs at all. no guarddog for me from vnc  
i havent tried the http method coz sometimes i get icewm, if i m not lucky i get a coloured background and a wait icon.. at least the mouse works. anyways, does anybody know how to fix this?*

#### Response

*See bugzilla bug 3081 to see what the problem is, why it can't really be solved by an official update at present, then proceed to pclinonline.com, and update your qt packages from Textar's urpmi medium.*

## **4. Discussion**

Although only a small scale pilot study, we believe we have identified certain key points that can inform future larger and more detailed analysis. We found several instances where the problem solving time could have been reduced if information were provided in the initial help request or automatically extracted before a question is posted. Techniques from the reference interview can help in eliciting important background information, including carefully determining the real underlying need, not just

the problem as presented. Vital contextual details are often easily obtainable in co-located support such as in over the shoulder learning. Remote help means that more effort is required to achieve clear efficient communication without easy access to physical resources, shared screens and common knowledge about the work context. Remote asynchronous help means that the rapid interactivity of telephonic technical support is lost and must be compensated for. Telephone support enables the quick detection and repair of breakdowns in understanding what the help-giver needs to know in order to help.

Details such as software version and operating system might be captured automatically and be posted below the question text as standard information [26]. Another potential candidate for automation or at least semi-automation is the history of actions of the user before the problem was encountered. As well as technical interventions, another approach is to both explain and model good practice in asking for and receiving help (e.g. [3, 4]), just as norms and procedures have been developed to improve the efficiency and effectiveness of OSS bug reporting. Alternative media, most notably screenshots were frequently used. It was striking to see how providing an image or an attachment of the errors on the screen could speed up the process, recovering some of the advantages of face to face help. These were cases where we could see clearly the advantages of these discussion forums over telephonic support.

There were many clarificatory questions that were asking for similar details and were posing the same question to the help-seeker. Similarly there were common elements that the help-seekers were posting or that if they had been posted earlier would have made the communication much quicker. It would be beneficial to have a standard format for the questions and the answers to be posted. These can be form-like but without mandatory field filling requirements. Even the existence of certain fields would prompt the users to put in the kind of information that would improve the time delays in the discussion and hence problem solving caused by the need for additional rounds of clarification.

Help discussion elements can be reused. Candidate materials include pointers to resources and well documented answers to frequent problems. This information could be refactored into lists of common questions and their solutions. This is clearly a kind of FAQ, but derived from particular problems and linking back to the discussion of those problems. It is potentially richer than conventional FAQs in also showing how the solution was obtained rather than just a distilled result. Techniques from knowledge management will be needed to organize and make such resources sufficiently accessible that a help-seeker will bother to try looking to see if their problem is already addressed in the knowledge base before posting a request

Unlike commercial telephone technical support which typically involves one person helping a user (except for formal escalations), in OSS support discussion lists, several people can be involved in helping solve a problem, and sometimes a composite solution is found from all the responses. So this kind of interaction is not a one to one process of problem solving, but many to one, and in cases of re-use, many to many.

The discussion forums also echo the ethos of open source software as a community for sharing and learning. People post problems as well as solutions for the benefit of other members of the community. A solution does not just benefit the help-seeker. The public nature of discussion boards means that they are frequented by people who do not post either questions or answers but benefit from the discussions anyway. There were several instances of messages posted a long time after the problem had been solved; where someone else who had the same problem, looked at this discussion and solution and then posted so that other also benefited from it. There were also instances similar to the “I have the same problem” which proved that this process was helpful to not only the people who were asking but to the community at large. In this way online support discussion forums are different from synchronous remote tech support, or face to face tech support.

We found distinct parallels with the reference interview, especially remote reference, most notably the iterations between the patron and librarian to get the correct question and complete information about the question. The provision of answers to the patron is also similar but there are several intermediate steps in digital reference that we did not see in our study. There was nothing similar to the formal triage, tracking or resource creation stages of digital reference in the data we examined, maybe because reference interviews are professionalized and the skills are formally taught. In technical help-giving the software under discussion is common between the two parties and a vital part of the conversation. Both the help-seeker and the help-giver share this common referent that binds their interaction. In digital reference, on the other hand, the only shared entity is the question being asked. In many cases, the use of information must be negotiated along with the question itself. Digital reference must have at its center the creation of this common context [28].

The study reported here is a preliminary one, illustrating the power of relatively lightweight methods to give an initial understanding of a complex issue. Its focus in selecting only certain kinds of problems to study has limitations that need to be addressed in future work. By intentionally only looking at longer, more complex and (it was presumed) more problematic interactions, we were able to see patterns emerging across multiple projects. However it would be helpful to compare those

findings with shorter, presumably more efficient or less problematic interactions from the same data sources. We also want to remove the constraint of only considering peer support and hence excluding moderators and appointed experts. Does this actually make any difference to the nature of the interactions, or is it just a matter of greater expertise allowing a broader range of diagnoses and solutions? A larger scale analysis using more quantitative methods would help validate the findings from this detailed but small qualitative analysis.

## 5. Conclusion

Our pilot study revealed several interesting similarities and differences between open source technical help-giving and both conventional technical help and library reference interviews. We believe that understanding how OSS technical help is given is important, particularly as a way to improving the process. Unfortunately to date it seems to have been rather overlooked.

As with other research topics in OSS, we can choose to study it in its own right as a phenomenon, at times an unusual or intriguing one that looks as if it breaks conventional rules and seemingly should not work at all. As such the very exceptionalism of OSS becomes the focus, to be compared with conventional software development processes, rather as biologists study cyanobacteria or duckbilled platyuses as strange boundary-challenging exceptions. Alternatively, we may study OSS because of its *similarity* to other software development processes, but which has the advantages of particular richness and particular visibility. In that case the biological metaphor might be fruit flies or Galapagos finches. We see much value in this latter view – that studying aspects of OSS (in this case OSS technical support) by virtue of its visibility and accessibility, in addition to improving the usefulness and usability of OSS, can also teach us a lot about how to improve many other kinds of technical support.

## 6. References

- [1] Abels, E.G., “The E-mail Reference Interview.” *RQ*, 35(3), 1996, 345-358.
- [2] Ackerman, M.S., “Augmenting Organizational Memory: A Field Study of Answer Garden,” *ACM Transactions on Information Systems*, 16(3), 1998, 203-224.
- [3] Agre, P. “The art of getting help.” *The Network Observer*, 1(2). 1994.
- [4] Agre, P. “How to help someone use a computer.” *The Network Observer*, 1(5), 1995.
- [5] Adhikari, R., “Help desks encouraging end users to help themselves”, *Software Magazine*, 14, 1994, 49.
- [6] Bettenhausen, K.L, and Murnighan, J.K., “The development of an intragroup norm and the effects of

- interpersonal and structural changes”, *Administrative Quarterly*, 36, 1991, 20-35.
- [7] Bopp, R. & Smith, L., *Reference and Information Services: An Introduction*. 3rd ed. Englewood CO: Libraries Unlimited, 2001.
- [8] Corbin, J., and Strauss, A., “Grounded theory research: Procedures, canons, and evaluative criteria.” *Qualitative Sociology*, 13, 1990, 3-21.
- [9] Crowston K., and Scozzi B., “Coordination Practices for Bug Fixing within FLOSS Development Teams”, *Proceedings of the First International Workshop on Computer Supported Activity Coordination, (CSAC 2004)* Porto (Portugal), April 2004.
- [10] Das, A., “Knowledge and productivity in technical support work.” *Management Science* 49(4), 2003, 416–431.
- [11] Davenport, T.H., De Long, D.W., and Beers, M.C. “Successful Knowledge Management Projects,” *Sloan Management Review*, Winter 1998, 43-57.
- [12] Donath, J., Karahalios, K., and Viegas, F., “Visualizing Conversation,” *Proceedings of the 32nd Hawaii International Conference on System Sciences (HICSS- 32)*, Track 2, 1999, 1-9.
- [13] Ehrlich, K., and Cash, D., “Turning information into knowledge: Information finding as a collaborative activity”, *Proceedings of Digital Libraries '94*, College Station, TX, 1994, 119- 125.
- [14] Gasser, L. and Ripoché, G., “Distributed Collective Practices and F/OSS Problem Management: Perspective and Methods,” presented at the *Conference on Cooperation, Innovation & Technologie, (CITE 2003)* University de Technologie de Troyes, France, 2003.
- [15] Halverson, C.A., Thomas, E. and Ackerman, M.S., “Behind the Help Desk: Evolution of a Knowledge Management System in a Large Organization”, *Proceedings of CSCW'04 Chicago, IL*, 2004, 304-313.
- [16] Howard, E.H., and Jankowski, T.A., “Reference Services via Electronic Mail.” *Bulletin of the Medical Library Association*, 74, 1986, 41-44.
- [17] Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., and Riedl, J.T., “GroupLens: Applying Collaborative Filtering to Usenet News,” *Communications of the ACM*, 40(3), 1997, 77-87.
- [18] Lakhani, K.R. and Wolf, R.G., “Why hackers do what they do: Understanding motivation and effort in Free / Open Source software projects”, Technical report, *MIT Sloan School of Management*, 2003, 1-27.
- [19] Lakhani, K.R., and von Hippel, E. “How Open Source Software Works: Free User to User Assistance.” *Research Policy* 32(6), 2003, 923-943.
- [20] Lamb, R., “Interorganizational Relationships and Online Information Resources”, *Proceedings of the 29th Annual Hawaii International Conference on System Sciences*, Maui, Hawaii, (HICSS-29) January 1996, 82-91.
- [21] Lankes, R.D., “The Digital Reference Research Agenda”, *Journal of the American Society for Information Science and Technology*, 55(4), 2004, 301-311.
- [22] Lewis, C. & Rieman, J. “Task-centered user interface design: A practical introduction.” Shareware book 1994, available at [<http://hcibib.org/tcuid/tcuid.pdf>].
- [23] Maass, W., “Inside an Open Source Software Community: Empirical Analysis on Individual and Group Level,” *Proceedings of the 4th Workshop on Open Source Software Engineering*, , Edinburgh, Scotland, May 2004, 65-71.
- [24] Nakakoji, K., Yamamoto, Y., Nishinaka, Y., Kishida, K. and Ye, Y., “Evolution Patterns of Open-Source Software Systems and Communities”, *Proceedings of 5th International Workshop on Principles of Software Evolution (IWPSE2002)*, Orlando, FL., May, 2002, 76-85.
- [25] Nichols, D.M. & Twidale, M.B. “The Usability of Open Source Software.” *First Monday* 8(1), 2003.
- [26] Nichols, D.M, McKay D, Twidale M.B. “Participatory Usability: supporting proactive users,” *Proceedings of 4th ACM SIGCHI NZ Symposium on Computer-Human Interaction (CHINZ'03)*, New Zealand, 2003, 63-68.
- [27] Pentland, B.T., “Read me what it says on your screen: The interpretative problem in technical service work” *Technology Studies*, 2(1) 1995, 50-79.
- [28] Pomerantz, J., Nicholson, S., Belanger, Y., and Lankes, R. D., “The Current State of Digital Reference: Validation of a General Digital Reference Model through a Survey of Digital Reference Services”, *Information Processing & Management*, 40(2), 2004, pp. 347-363.
- [29] Rossi, C. and Bonaccorsi, A., “Intrinsic vs. extrinsic incentives in profit-oriented firms supplying Open Source products and services” *First Monday*, 10(5), May 2005.
- [30] Sandusky, R.J., Gasser, L., and Ripoché, G., “Bug Report Networks: Varieties, Strategies, and Impacts in a F/OSS Development Community”, *Proceedings of the 1st International Workshop on Mining Software Repositories (MSR 2004)*, Edinburgh, Scotland, UK, 2004, 80-84.
- [31] Sarwar, B.M., Konstan, J.A., Borchers, A., Herlocker, J., Miller, B., and Riedl, J., “Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System”, *Proceedings of CSCW '98*, Seattle, WA, 1998, 345-354.
- [32] Scacchi, W., “Understanding the Requirements for Developing Open Source Software Systems”, *IEE Proceedings — Software*, 148(1), 2002, 24-39.
- [33] Stewart, K.J., “OSS Project Success: From Internal Dynamics to External Impact”, *Proceedings of the 4th Workshop on Open Source Software Engineering*, Edinburgh, Scotland, 2004, 92-97.
- [34] Twidale, M.B. and Nichols, D.M., “Exploring Usability Discussions in Open Source Development”, *Proceedings of the 38th Annual Hawaii International Conference on System Sciences, HICSS-38*, January 2005, 198c.
- [35] Twidale, M.B. and Ruhleder, K. “Where am I and Who am I? Issues in collaborative technical help.” *Proceedings of CSCW'04 Chicago, IL*, 2004. 378-387.
- [36] Wang, H., and Wang, C., “Open source software adoption: a status report”, *IEEE - Software*, 18(2), 2001.
- [37] Wenger, E. *Communities of Practice: Learning, Meaning and Identity*, NY: Cambridge Univ. Press, 1998.
- [38] Xiong, R., and Donath, J., “PeopleGarden: Creating Data Portraits for Users”, *Proceedings of UIST'99*, Asheville, NC, 1999, 37-44.