

How Do Open Source Developers Talk about Users?

Andrew J. Ko and Parmit K. Chilana

Information School

University of Washington

Seattle, WA (USA)

{ajko, pchilana}@u.washington.edu

ABSTRACT

When open source software developers are making design decisions, how do they talk about users? To begin to answer this question, 100 contentious Firefox bug reports were analyzed for distinct uses of the word “user.” The results show that developers use authoritative words (such as allow, educate, and require) to describe what software does for users. Most statements involved confident speculation about what users need, expect and do, whereas a minority of statements demanded evidence for such unsubstantiated claims. The results also show that when describing users, developers describe them in general terms, rather than referring to particular uses cases or user populations. These results suggest that, at least in the broader Firefox developer community, developers rely largely on stereotype and instinct to understanding user needs.

Author Keywords

Open source software, usability, Mozilla, Firefox

ACM Classification Keywords

H.5.2 [Information interfaces and presentation]: User interfaces—*evaluation/methodology*.

INTRODUCTION

Nearly 25 years ago, Gould and Lewis [4] found that many designers believed investigating user needs would be fruitless as users do not know what they need. Some designers also tended to underestimate or overestimate user diversity, arguing that user research was unnecessary because user behavior was either too homogenous to require such investigation, or too diverse to benefit from it. Arguably, some of these beliefs sound naive today. Many companies are competing on the usability of their software, and particularly on the web, user-centered design principles have received broad awareness. User-centered design is taught in universities around the world, and in the past

decade, software companies and even open source communities have begun to form dedicated user research teams in order to improve the user experience of their software.

Given these shifts, have software designers’ beliefs about users changed? For example, the Firefox web browser is thought of as one of the most usable browsers relative to many other browsers, and one of the most usable open source applications available (despite issues with the usability of other open source projects [6]). Is there something particular about the developer community and how they perceive users in projects like Firefox that allows them to better achieve usability?

We investigated this question by analyzing how Firefox developers use the word “user” in design discussions embedded in bug reports. Unlike asking developers about their attitudes about user-centered design, this data allows us to see, as independent observers, how developers’ attitudes influence practice. For example, do developers discuss users in objective, substantive ways? Or do they invoke stereotypes and generalizations about users and design from instinct? What are the different ways in which the word “user” is used to convey arguments and make claims about user behavior? By answering these questions, we can begin to understand how the software designers of today reason about user needs.

To answer these questions, we analyzed a large corpus of statements in 100 public Firefox bug reports containing the word “user.” Our goal in analyzing these reports was to understand developers’ use of the word “user,” but bug reports cover a much broader spectrum of collaboration, including reproduction, code review, and repair [5]. In order to focus our analysis on discussion, we focused on reports of problems that had been reproduced and decided upon, by downloading only those marked as RESOLVED, VERIFIED, or CLOSED and resolved as FIXED, INVALID, or WONTFIX.

KEY FINDINGS

Developers Make Strong Claims about User Impact

About 27% (124 of 452 statements) of the occurrences of “user” described what the current or future version of the software would make possible for users (independent of whether such capabilities would be useful to users). A useful way to analyze these statements is by the verb used. In our sample, there were 11 kinds of verbs used to describe what affect the software would have on users (e.g. *allows, enables, educates, requires, restricts, breaks, helps, protects, confuses, satisfies, annoys*).

Another interesting trend in these statements about user impact was that, at least when talking about what the system would make possible, developers usually spoke universally about user impact. In 90 of the 124 statements, developers referred to “users” or “the user” in general when speculating about what the system would make possible, instead of identifying a particular group of users who would benefit. In the other 34 statements, developers tended to identify groups who performed particular actions or used particular platforms or systems.

Developers Speculate Confidently about User Behavior and Needs

About half of the uses of user involved claims about user behavior or needs (260 of 452 statements). We again found it informative to consider the verbs that developers used in describing users’ behaviors and needs (e.g., *do, understand, expect, want, don’t care, need, confused by, benefit from, annoyed by, know*). The most common claims that developers made were about what users would understand, want or expect in particular design changes and what users do with the existing software design. Developers also speculated about what users care or do not care about, what they would be confused and annoyed by, and what they know and need.

Few Developers Demand Evidence

Although the previous findings paint a picture of developers as highly speculative and overconfident, there were some developers who criticized these practices, or at least reflected on them. The first critique raised concerns about the lack of evidence that developers had about user behavior or user needs and the problems with making changes to the software on such unsubstantiated grounds. The second critique raised concerns about the lack of depth in developers’ understanding of user needs, critiquing the ill-defined nature of the problems that developers were trying to solve. The last critique was for developers patronizing, deriding, and even insulting users. Unfortunately, few of these comments led to substantive discussion about how to substantiate developers’ claims or how to better understand user needs.

DISCUSSION

In general, our findings reveal a number of interesting trends in the use of the word “user.” In particular, our results show that the word helps accomplish two major tasks: (1) *Describing what the software enables users to do,*

implicitly connecting these capabilities to possible user needs. (2) *Speculating about user needs*, but rarely invoking even simple forms of objective evidence. This is similar to the notion of the “elastic user” [3], in which users are described in whatever way would support developers’ opinions. What this reveals is an imbalance of evidence: not surprisingly, developers were confident in their reasoning about system capabilities, but lacking in evidence about to what extent these capabilities matched user needs.

Although our data suggests that Firefox developers lack rigor in their understanding of users, the frequency with which users are mentioned implies that the community has high awareness of the importance of user needs. The lack of user research, with the high awareness of the user perspective and the relative usability of Firefox suggests that simply bringing the user perspective into discussions may promote more iterative development and refinement. For example, perhaps exaggerated, speculative claims about users play a positive role in design discussions, helping groups to identify more conservative ground truths, even in the absence of evidence.

Returning to our original question, have software designers’ beliefs about users changed in the past 25 years? At least with regard to our sample, the answer is yes: attitudes have shifted from one where the developer knows best to one where designers are more aware of their role as a service provider. What has not changed is the degree to which designers seek evidence about actual user needs and the degree to which they use such evidence to inform their design decisions. These findings both affirm the importance of user-centricity in software design, but also call for future work in evidence about user needs more widely available in the resource- and time-constrained context of larger software development practice. For example, Gould and Lewis [4] originally proposed a form of participatory design, in which users are active participants of a design team; these practices are now being seen in open source projects [1,2]. Both researchers and practitioners, in industry and open source communities, should think carefully about tools and methods that make learning about user needs more feasible.

REFERENCES

1. Bach, P.M., DeLine, R., and Carroll, J.M. Designers wanted: participation and the user experience in open source software development. *Proc CHI '09*, (2009), 985-994.
2. Barcellini, F., D tienne, F., and Burkhardt, J. User and developer mediation in an Open Source Software community: Boundary spanning through cross participation in online discussions. *Int. J. Hum.-Comput. Stud.* 66, 7 (2008), 558-570.
3. Cooper, A. and Reimann, R. About Face 2.0: The essentials of interaction design. *Information Visualization* 3, (2004), 223-225.
4. Gould, J. and Lewis, C. Designing for usability: key principles and what designers think. *Comm ACM* 28, 3 (1985), 300-311.
5. Ko, A., Deline, R., and Venolia, G. Information Needs in Collocated Software Development Teams. *Proc ICSE '07* (2007), 344-353.
6. Nichols, D. and Twidale, M. The usability of open source software. *First Monday* 8, 1-6 (2003).