Requirements Acquisition in Open Source Development: Firefox 2.0

John Noll

Computer Engineering Department
Santa Clara University
500 El Camino Real, Santa Clara, CA 95053 USA
`jhnoll@gmail.com`

Abstract. Open Source Software Development appears to depart radically from conventional notions of software engineering. In particular, requirements for Open Source projects seem to be asserted rather than elicited.

This paper examines features of the latest major release of the Firefox web browser in attempt to understand how prevalent this phenomenon is. Using archives of mailing lists and issue tracking databases, these features were traced from first mention to release, to determine the process by which requirements are proposed, adopted, and implemented in Firefox. The results confirm the importance of user participation as developers of open source products.

1 Introduction

Open source products have garnered significant interest in the popular as well as research literature. Banner projects like Linux, Mozilla, and Apache seem to represent a radical departure from conventional ideas of software engineering, harnessing the labor of numerous volunteers who are distributed throughout the world, contribute according to their desires and abilities, and are motivated apparently only by the joy of creating software, to produce high quality products that dominate their respective markets.

In his essay "The Cathedral and the Bazaar," Eric Raymond captured this popular conception of the differences between open source and conventional software development, arguing that open source project organization emerges from a marketplace of developers, who gradually take on increasing leadership responsibilities as their contributions to the project are recognised by their peers (the "Bazaar"). This is in contrast to the traditional top-down approach where managers set schedules and assign tasks (like priests in a "Cathedral") [43].

Research has shown that the perceived differences between open source and conventional software development projects are only partially true, at least for the highly visible projects. For example, the most successful open source software projects em-

ploy substantial numbers of paid programmers [25]. The Mozilla project has regular face-to-face meetings (augmented by teleconferencing) to review project status [35].

One area where open source software development does appear to differ is in requirements acquisition. Requirements for open source products are often asserted by developers, rather than elicited from users, in contrast to the conventional approach in which user needs are discovered through focus groups, interviews, and other means. This phenomenon has been observed by several researchers studying open source software development [13, 45].

How frequently are requirements asserted in an open source project? This paper presents results of a study of one open source product: the Firefox web browser produced by the Mozilla foundation [30]. The study attempted to determine how many features in the third major release (2.0) of Firefox were actually asserted by developers, rather than derived from user input or other sources.

Of fifteen new features in the Firefox 2 release, nine were determined to be asserted by developers, three were derived from user input, and two features had origins in "extensions" implemented by third parties using Firefox's extension mechanism.

The rest of this paper is organized as follows: the next section describes the method used to study the product's features; following that, the study results are presented. The paper concludes with a survey of related work, and conclusions.

2 Method

The Firefox project is interesting for a number of reasons. It has a large user community and is considered to be among the best web browsers available, with superior functionality and security [19]. The Firefox development organization is large and mature, so the processes used for the current release are well established and therefore represent a "typical" way of doing things for a project of this size.

As an open source project, Firefox is unusual in that it incorporates some aspects of more conventional software development organizations: regular (weekly) in-person status meetings are held (although remote participants can contribute via conference calling) [35]; a formal requirements document for Firefox 2 was written [33] and updated periodically [34]; project schedules are created and posted for review [32]. As such, one would expect to see elements of conventional software engineering practices in Firefox development processes, including requirements engineering.

The method employed for this research involved three steps:

1. Identify the set of features comprising the current (2.0) Firefox release.
2. Examine Internet resources related to Firefox, such as archives of discussion groups, web logs, issue databases, and other online forums, to discover when the feature was first proposed, and what role the person proposing the feature played (such as user or developer).
3. Determine the initial implementation of the feature (prototype by a developer, Firefox extension, or enhancement to the core codebase).
4. Categorize the requirement as asserted by a developer, either from his or her personal experience or knowledge of user needs; derived from user contribution, for

example by filing a bug report or "Request for Enhancement" in the Bugzilla issue database; or derived from competition.

The release notes for Firefox 2 (code named "Bon Echo") list fifteen new features, or enhancements to existing features, for this release:

1. "Visual Refresh" (enhancement).
2. Phishing protection (new).
3. "Enhanced search capabilities" (new).
4. "Improved" tabbed-browsing (enhancement).
5. Resumption of previous browsing session (new).
6. Web feed (RSS) preview and subscription (enhancement).
7. Spell checking (new).
8. "Live Titles" (new).
9. "Improved Add-ons manager" (enhancement).
10. Update to JavaScript version 1.7 (enhancement).
11. Search engine "plugins" in Sherlock or OpenSearch format (new).
12. Updated extension mechanism (enhancement).
13. Support for SVG text (bug fix).
14. New installer for Microsoft Windows©systems, based on the Nullsoft Scriptable Install System (enhancement) [31].

Some of these (phishing protection, session resumption, RSS subscription, spell checking, and "live titles") represent new features that were not present in previous releases. For these, the first expression of the need for the new feature was used to establish the source of the feature. Other features, such as tabbed-browsing improvements and updated extension mechanism, are modifications to existing features. In these cases, the first expression of need for enhancement (or fix) was used to establish the source of the requirement.

Some "features" listed in the release notes are really collections of enhancements to some category of Firefox behavior or appearance. For example, the "visual refresh" involves changes to icons, colors, and the shapes of buttons and panels. Likewise, the updated extension mechanism involves several changes to the way Firefox incorporates extensions developed by third parties (for example, MultiZilla [42] and Zotero [56]). In such cases, an attempt was made to establish whether the initial need was expressed as a collection, or whether separate needs were later collected under a single category.

3 Results

The results of the analysis are summarized in Table 3. The second column identifies the artifact where the first expression of the requirement was found. In two cases (Enhanced Search and JavaScript 1.7), no mention of the requirement appears before the Firefox Product Requirements Document was published; the source document for these is identified as 'PRD.'

Table 1: Source and classification of Firefox 2 Requirements

Feature	Requirement Source	Initial Implementation	Classification
Visual Refresh	discussion forum[8, 10]	core	asserted, from knowledge of user needs
Phishing Protection	Bugzilla[48], Internet Explorer[49]	extension (Google Safe Browsing[17])	asserted, from knowledge of user needs and competition
Enhanced Search	PRD[33]	extension (Google search toolbar[18])	derived from extension
Improved Tabbed Browsing	Mozilla Wiki[3, 16], Bugzilla[14]	prototype	asserted and formally validated
Session Resume	extensions (SessionSaver[4], Tabbedbrowser Extensions[21]), Opera	extensions	derived from extension; competition[12]
Web Feed Preview	Bugzilla[2, 28]	extensions (RSS Reader Panel[22], Sage[42])	asserted, from personal experience
Spell Checking	Bugzilla[5, 9]	extensions (SpellBound[51], Torisugari[53])	user contributed
Live Titles	discussion forum[26], developer web log[27]	prototype[26]	asserted, from personal experience
Improved Add-ons Manager	Mozilla wiki[15]	core	asserted, from personal experience
Javascript 1.7	PRD[33]	core	asserted, from personal experience[6]
Search Plugins	Safari[24], Mycroft[36], Sherlock[50], Opera[40]	core	competition
Updated Extension Mechanism	Bugzilla[39]	core	user contributed
SVG Text Support	Bugzilla[41]	core	user contributed (bug)
Windows Installer	Bugzilla[1, 38, 52]	prototype[7]	asserted, from knowledge of user needs

The third column identifies where the first implementation of the behavior appeared in Firefox; 'core' refers to the Firefox core codebase; 'extension(s)' to a package implemented using Firefox's extension mechanism that allows third parties to enhance Firefox's features with substantial functionality; and 'prototype' to a developer's prototype distributed for evaluation by the community.

The last column categorizes how the requirement was proposed; there are five categories:

1. asserted by a developer, from his or her own personal experience;
2. asserted by a developer, based on his or her personal knowledge of what users need;
3. contributed by a user who is not a developer, by posting a bug report or request for enhancement to the Bugzilla issue database;
4. derived from the success of an extension;
5. motivated by appearance of the feature in a competing product.

The majority of requirements (seven total) were asserted by developers, based on either their personal experience or knowledge of user needs.

Three features were influenced, at least in part, by competition from other browsers: Search Plugins are supported by Opera [40] and Safari [24], and are defined by the open-source search plugin efforts Mycroft [36] and Sherlock [50]; Phishing Protection was rumored to be part of an upcoming release of Internet Explorer [49]; and, Session Resume was implemented by Opera [12].

Two features (Phishing Protection and Enhanced Search) were derived directly from successful extension implementations (Google Safe Browsing [17] and Google Search toolbar [18], respectively).

Three features were derived from user contributed bug reports or requests for enhancement: Spell Checking [5, 9], Updated Extension Mechanism [39], and SVG Text Support [41].

The distinction between requirements asserted by developers, and requirements elicited from users in the textbook manner, is not as extreme as it might appear. In the case of Firefox, developers are users as well, so the difference is more one of how requirements are validated than from where they originate: in the Firefox project, requirements asserted by developers are rarely validated by a formal process; in only one case – Improved Tabbed Browsing – was any formal validation process undertaken to confirm that users did indeed need the feature [16].

Rather, requirements are typically posted to discussion forums for informal validation through feedback from other developers, and any dedicated users that participate in the forums. As a consequence, it appears that an open source project like Firefox must have a certain number of developers who are also users, in order to provide accurate requirements: since developers are the source of requirements, enough developers must be in touch with the larger user community to keep the feature set relevant to the bulk of users.

In a few cases, features were initially implemented as extensions, rather than as part of the Firefox core codebase; these extensions proved popular enough, by virtue of the number of people who installed and used them, to be considered for incorporation

into the core. Similarly, some features are influenced by the same functionality being present in a competing product. Both cases, in effect, represent requirements that are implicitly validated by the market.

However, most other requirements, including those originating with user contributions, are informally validated by consensus of the core developers.

Since web browsers are a ubiquitous tool, one would expect that the majority of Firefox developers are also Firefox users with needs that are the same as the larger user community, and so this process of informal validation could be expected to work well. This may not be the case for other open source products with more focused user communities, such as ERP or health care applications.

4 Related Work

Studies of open source software development projects address a wide range of topics from economics [55] to maintainability [47].

A number of case studies have examined open source development processes, including those employed by Apache and Mozilla [29, 44, 46]. In particular, Reis and de Mattos Fortes, in their study of Mozilla development processes, report that high level requirements are specified by the Mozilla Foundation management, but all development on the Mozilla code base originates with a "bug" report, which might be submitted by another developer, tester, or end user [44]. These reports may document some product failure, or a request for enhancement.

Feller and Fitzgerald note that users are a "critical feature" [11, 10] of open source projects, as the source of new requirements. Scacchi has made several studies of requirements acquisition in open source software development; he observes that requirements "emerge" from on-line discussions which are usually open forums, rather than through traditional requirements elicitation processes, but that this emergent process, though less formal, is also effective [45, 46]. He also notes that requirements are "asserted" after the fact; other researchers have echoed this observation. In particular, German reports a similar situation in the Gnome project [13]. This seems to contradict conventional understanding that cites failure to understand requirements as a major source of software project failure.

But Trudelle observes, in his discussion of lessons learned from experience working on Mozilla, that this approach led to rework of some of the Mozilla implementation in response to user-submitted bug reports; his view is that this rework could have been avoided with traditional up-front requirements analysis and design activities [54]. Henderson echoes this view, claiming that open source projects do not employ "requirements elicitation," but that this could (and should) be easily added to open source processes [20]. Further, Nichols and Twidale observe that usability requirements are not captured well by open source projects, due to the mismatch between developers and users; their view is that the open source approach of "coding as early as possible" violates "good interface design [37]."

These observations run counter to the prevailing open source view that de-emphasizes formal design and requirements gathering, yet also hint at the possible evolution of

mature projects like Firefox. One the one hand, Trudelle's view – that open source software projects need an overarching UI design and design function – seems to contradict the current success of Firefox, which is widely recognized as among the most innovative web browsers. In particular, Nichols and Twidale's assertion that "commercial software establishes the state of the art" [37] seems to be contradicted by Opera and Firefox, both of which included UI features (tabbed browsing, for example) well before Internet Explorer. Yet, the results of this study of Firefox 2 show that some formal user experience design is being undertaken; and, at least three features did follow the lead of competing commercial products.

5 Conclusions

Most Firefox requirements are asserted by developers. The Firefox project does, however, occasionally resort to more rigorous requirements validation, as in the case of the "Visual Refresh" where usability studies were conducted at Google on behalf of the Firefox effort. This confirms the views of Trudelle [54] and Henderson [20], that open source software projects could benefit from more formal requirements analysis.

Firefox is not a "pure" open source project in the spirit of Raymond's community of volunteers scratching an "itch [43]." The Firefox project incorporates some aspects of "traditional" software engineering such as requirements documents and usability studies; it also inherits face-to-face meetings, paid developers, and a formal corporate infrastructure from the Mozilla Foundation. At the same time, the Firefox project has the key aspects of an open source project that distinguish this kind of software development from conventional commercial development: transparency through publicly accessible documents and discussion forums, and direct participation by enthusiastic users.

This philosophy seems to be summed up nicely by the following comment to a usability issue in the Mozilla issue database:

> As an average user I would think: how many clicks does it need to get there? But there are many different average users :)
> What people really find convenient you can only know when you make a test browser with the new feature, let a thousand users use it for some weeks and then ask the right questions [23].

References

[1] alanjstr: Bug 231062 provide Firefox MSI package. `https://bugzilla.mozilla.org/show_bug.cgi?id=231062` (2004). Issue posted to Firefox issue database, accessed December 12, 2007.
[2] aldo-public: Comment on bug 270541. `https://bugzilla.mozilla.org/show_bug.cgi?id=270541c3` (2005). Comment on request for enhancement posted to Mozilla issue database, accessed October 29, 2007.

[3] Beltzner, M.: Firefox/feature brainstorming. `http://wiki.mozilla.org/index.php?title=Firefox/Feature_Brainstorming&oldid=16495` (2006). Entry in Mozilla Wiki, accessed December 4, 2007.

[4] Burnett, G.: Firefox 1.5 session saver extension. `http://www.yista.com/gburnett/firefox-15-session-saver-extension/` (2005). Entry posted to the *Yah, I Saw That Already* web log, accessed December 15, 2007.

[5] Cassin, R.: Bug 56301 (spellchecker) connect a spellchecker engine for Mozilla. `https://bugzilla.mozilla.org/show_bug.cgi?id=56301` (2000). Issue posted to the Mozilla issue database describing a need for spell checking in Mozilla Composer, accessed December 14, 2007.

[6] Champeon, S.: JavaScript: How did we get here? `http://www.oreillynet.com/pub/a/javascript/2001/04/06/js_history.html` (2001). Web page, accessed December 18, 2007.

[7] Delahaye, S.: Firebird installer. `http://seb.mozdev.org/firebird/` (2003). Web page, accessed December 12, 2007.

[8] Dotzler, A.: Reply to 'complete support of XP visual style?'. `http://forums.mozillazine.org/viewtopic.php?p=2668` (2002). Reply to comment posted to Mozilla Themes Development forum, accessed October 3, 2007.

[9] Epperson, B.: Bug 16409 invoke spell check in browser window (multiple form fields). `https://bugzilla.mozilla.org/show_bug.cgi?id=16409` (1999). Issue posted to the Mozilla issue database describing a need for spell checking in the Mozilla web browser, accessed December 14, 2007.

[10] ExNihilo: Complete support of XP visual style? `http://forums.mozillazine.org/viewtopic.php?t=452` (2002). Comment posted to Mozilla Themes Development forum, accessed October 3, 2007.

[11] Feller, J., Fitzgerald, B.: A framework analysis of the open source software development paradigm. In: Proceedings of the International Conference on Information Systems, pp. 58–69. Association for Information Systems (2000)

[12] Ganesh, V.: Top 10 usability features I would like in a browser. `http://geekswithblogs.net/vganesh/archive/2005/09/22/54669.aspx` (2005). Web log entry, accessed December 18, 2007.

[13] German, D.M.: GNOME, a case of open source global software development. In: Proceedings of the 6th International Workshop on Global Software Development. Portland, OR USA (2003)

[14] Goodger, B.: Bug 308396 - UE fixes for tabbed browsing. `https://bugzilla.mozilla.org/show_bug.cgi?id=308396` (2005). Issue posted to Firefox issue database, accessed December 5, 2007.

[15] Goodger, B.: Firefox:extension manager UI. `http://wiki.mozilla.org/Firefox:Extension_Manager_UI` (2005). Entry posted to Mozilla Wiki describing problems and possible solutions with the Firefox extension manager user interface, accessed November 20, 2007.

[16] Goodger, B.: Link targeting. `http://wiki.mozilla.org/Link_Targeting` (2006). Entry in Mozilla Wiki, accessed December 4, 2007.

[17] Goodger, B.: Phishing protection. `http://wiki.mozilla.org/Phishing_Protection` (2006). Mozilla wiki entry describing history and design of Firefox Phishing Protection feature.

[18] Google, Inc.: Google toolbar for Firefox. `http://www.google.com/tools/firefox/toolbar/FT3/intl/en/` (2007). Web page for Google Toolbar, accessed December 12, 2007.

[19] Hamm, S.: A Firefox in IE's henhouse. Business Week (2004)

[20] Henderson, L.G.R.: Requirements elicitation in open-source programs. CrossTalk - The Journal of Defense Software Engineering **13**(7), 28–30 (2000). `http://www.stsc.hill.af.mil/crosstalk/2000/07/henderson.html`

[21] Hiroshi, H.: Tabbrowser extensions. `http://piro.sakura.ne.jp/xul/tabextensions/old_index.html.en` (2007). Web home page for (now obsolete) Tabbedbrowser Extensions, accessed December 15, 2007.

[22] jacob667: Using the RSS reader panel [tip 28]. `http://jacob667.livejournal.com/9948.html` (2004). Entry posted to "Jacob's Mozilla Tips" web log, accessed October 25, 2007.

[23] Klassen, R.: Comment on bug 335781. `https://bugzilla.mozilla.org/show_bug.cgi?id=335781#c2` (2006). Comment to issue posted to Mozilla issue database, accessed November 19, 2007.

[24] Kottke, J.: Why are Safari and Sherlock two different applications? `http://www.kottke.org/03/01/safari-sherlock-different` (2003). Entry posted to *kottke.org* web log, accessed November 12, 2007.

[25] Lakhani, K.R., Wolf, R.G.: Perspectives on Free and Open Source Software, chap. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. MIT Press (2005)

[26] Melez, M.: Microsummaries in Firefox 2.0. `http://groups.google.com/group/mozilla.dev.apps.firefox/tree/browse_frm/thread/e6b47d5f7af7d77d/670a06e452b63b9d?rnum=1&_done=%2Fgroup%2Fmozilla.dev.apps.firefox%2Fbrowse_frm%2Fthread%2Fe6b47d5f7af7d77d%2F670a06e452b63b9d%3F#doc_670a06e452b63b9d` (2006). Post to `mozilla.dev.apps.firefox` discussion group, accessed December 10, 2007.

[27] Melez, M.: son of live bookmarks. `http://www.melez.com/mykzilla/2006/01/son-of-live-bookmarks.html` (2006). Web log entry, accessed December 10, 2007.

[28] Milford, D.: Bug 270541 - support drag-and-drop of RSS feed (icon) from Firefox. `https://bugzilla.mozilla.org/show_bug.cgi?id=270541` (2004). Request for enhancement posted to Mozilla issue database, accessed October 29, 2007.

[29] Mockus, A., Fielding, R.T., Herbsleb, J.: A case study of open source software development: The Apache server. In: Proceedings of the 22nd International Conference on Software Engineering, pp. 263–272. Limerick, Ireland (2000)

[30] Mozilla Corporation: Firefox 2. `http://www.mozilla.com/en-US/firefox/` (2007). Web page about the Firefox web browser, accessed December 12, 2007.

[31] Mozilla Development Team: Firefox 2 release notes. `http://en.www.mozilla.com/en/firefox/2.0/releasenotes/` (2007). Web page, accessed October 26, 2007.

[32] Mozilla Foundation: Firefox2/HistoricalSchedule. `http://wiki.mozilla.org/Firefox2/HistoricalSchedule` (2006). Web page describing the Firefox 2 release schedule, accessed December 12, 2007.

[33] Mozilla Foundation: Firefox2/PRD. `http://wiki.mozilla.org/Firefox2/PRD` (2006). Web page containing Mozilla Firefox 2 Product Requirements Document, accessed August 24, 2007

[34] Mozilla Foundation: Firefox2/Requirements. `http://wiki.mozilla.org/Firefox2/Requirements` (2006). Entry in Mozilla Wiki describing Firefox 2 requirements.

[35] Mozilla Foundation: Firefox2/StatusMeetings. `http://wiki.mozilla.org/Firefox2/StatusMeetings` (2006). Web page indexing Mozilla Firefox 2 status meeting notes, accessed October 2, 2007.

[36] Mycroft Project: Mycroft project: Sherlock & OpenSearch search engine plugins. `http://mycroft.mozdev.org/index.html` (2007). Web page describing the Mycroft search engine plugin collection project, accessed November 4, 2007. The "Mycroft" name comes from the name of Sherlock Holmes's brother, Mycroft, from the novels of Arthur Conan Doyle.

[37] Nichols, D.M., Twidale, M.B.: The usability of open source software. First Monday **8**(1) (2003)

[38] Nicholson, G.: Bug 238212 Firefox should have a net installer, like Seamonkey. `https://bugzilla.mozilla.org/show_bug.cgi?id=238212` (2004). Issue posted to Firefox issue database, accessed December 11, 2007.

[39] Nicolas: Bug 285848 - extension manager should be able to manage the language of the extensions. `https://bugzilla.mozilla.org/show_bug.cgi?id=285848` (2005). Request for enhancement posted to Firefox issue database, accessed November 21, 2007.

[40] Opera Software ASA: Changelog for Opera 6.1 beta 1 for Linux. `http://www.opera.com/docs/changelogs/linux/610b1/index.dml` (2001). Web page, accessed November 5, 2007.

[41] Ortyl, P.: Bug 282579 - implement <svg:textPath>. `https://bugzilla.mozilla.org/show_bug.cgi?id=282579` (2005). Issue database entry describing Firefox deficiency handling <svg:textPath> tag.

[42] van Rantwijk, H.J.: MultiZilla's home page. `http://multizilla.mozdev.org` (2006). Home page for the MultiZilla project, cited September 6, 2006.

[43] Raymond, E.S.: The cathedral and the bazaar. In: The Cathedral and the Bazaar. O'Reilly and Associates (1999)

[44] Reis, C.R., de Mattos Fortes, R.P.: An overview of the software engineering process in the Mozilla project. In: Proceedings of the Open Source Software Development Workshop. Newcastle upon Tyne, UK (2002)

[45] Scacchi, W.: Understanding the requirements for developing open source software systems. IEE Proceedings – Software **149**(1), 24–39 (2002)

[46] Scacchi, W.: Free and open source development practices in the game community. IEEE Software pp. 59–66 (2004)

[47] Schach, S.R., Jin, B., Wright, D.R., Heller, G.Z., Offut, A.J.: Maintainability of the Linux kernel. IEE Proceedings – Software **149**(1) (2002)

[48] Schneider, F.: Bug 329292 - add SafeBrowsing anti-phishing extension to trunk for evaluation. `https://bugzilla.mozilla.org/show_bug.cgi?id=329292` (2006). Request for enhancement posted to Mozilla issue database, asserting need to integrate Safe Browsing into Firefox core.

[49] Schultz, K.: Comment on bug 329292. `https://bugzilla.mozilla.org/show_bug.cgi?id=329292#c9` (2006). Comment #9 posted to Mozilla issue database discussion of Bug 329292.

[50] Sellers, D.: Better searching with Sherlock 2. `http://www.computeruser.com/articles/1906,5,18,1,0601,00.html` (2000). Article posted to the Computer User web site, access November 12, 2007.

[51] Strong, R.: SpellBound - release notes. `http://spellbound.sourceforge.net/relnotes` (2005). Web page containing release history for the SpellBound extension to Firefox, accessed December 14, 2007.

[52] Strong, R.: Bug 326580 - Firefox 2.0 Windows installer. `https://bugzilla.mozilla.org/show_bug.cgi?id=326580` (2006). Issue posted to Firefox issue database, accessed December 11, 2007.

[53] Torisugari: Spell checker for Firebird. `http://forums.mozillazine.org/viewtopic.php?t=34799&postdays=0&postorder=asc&postsperpage=15&start=0` (2003). Entry in MozzillaZine "Extension Development" discussion forum describing initial port of Thunderbird spell checking mechanism to Firebird v. 0.7, accessed December 14, 2007.

[54] Trudelle, P.: Shall we dance? Ten lessons learned from Netscape's flirtation with open source UI development. Tech. rep., Mozilla.org (2002). URL `\url{http://www.iol.ie/\~calum/chi2002/peter_trudelle.txt}`. Presented at the Open Source Meets Usability Workshop, Conference on Human Factors in Computer Systems (CHI 2002), Minneapolis, MN. Accessed December 28, 2006.

[55] Wheeler, D.A.: Why open source software / free software (OSS/FS, FLOSS, or FOSS)? Look at the numbers! Technical report, dwheeler.com (2005). URL `http://www.dwheeler.com/oss_fs_why.html`

[56] Zotero Project: Zotero - the next generation research tool. `http://www.zotero.org/` (2007). Home page of the Zotero Project, accessed December 15, 2007.