A Tale of Two Browsers

Olga Baysal Ian Davis Michael W. Godfrey
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada
{obaysal, ijdavis, migod}@uwaterloo.ca

ABSTRACT

We explore the space of open source systems and their user communities by examining the development artifact histories of two popular web browsers — Firefox and Chrome — as well as usage data. By examining the data and addressing a number of research questions, two very different profiles emerge: Firefox, as the older and established system, with long product version cycles but short bug fix cycles, and a user base that is slow to adopt newer versions; and Chrome, as the new and fast evolving system, with short version cycles, longer bug fix cycles, and a user base that very quickly adopts new versions as they become available (due largely to Chrome's mandatory automatic updates).

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics—*product metrics*

General Terms

Performance, reliability

Keywords

Software repositories, mining repositories, release history

1. INTRODUCTION

In this paper, we study the release histories and usage patterns of two popular open source web browsers, Firefox and Chrome. We present a comparative analysis of the systems by mining their bug repositories and web traffic data to determine the trends in user acceptance and adoption of a browser that could help to explain factors behind the popularity of an open source browser. To do this, we address a number of research questions that concern release frequency, defect rates, time to fix bugs, user update likelihood, and market share.

In Section 2, we examine the browsers themselves by considering their release histories and comparing their defect rates and time to fix bugs. In Section 3, we enlarge our view to include usage data, and we address questions relating to user adoption and updating. In Section 4, we consider market share and popularity relative to defect rates. Finally, in Section 5, we summarize our findings.

MSR '11, May 21-22, Waikiki, Honolulu, HI, USA

2. BROWSER RELEASE HISTORY, BUGS, AND BUG FIXING

Chrome is the newer one, having been released in 2008 by Google, aiming to be a fast, lean, and simple alternative to existing browsers such as Internet Explorer and Firefox. Chrome uses the Webkit layout engine also used by Apple's Safari browser, which originally comes from the Linux KDE Konqueror browser. Strictly speaking, Chrome is not open source but its core base — a project called Chromium – is.

Officially, Firefox was first released in 2004, but its codebase has a long and rich history going back to Netscape, the second historically important web browser after Mosaic. In 1998, the Netscape codebase was branched into the open source Mozilla project suite, which included the browser, an email client, and an HTML editor. In 2004, the browser and email clients were decoupled into separate projects (Firefox and Thunderbird, respectively), and the Firefox browser was officially born. The design goals of Firefox include being a standards-compliant, feature-rich, modular, extensible, and stable open source web browser. Its rendering engine, Gecko, has been reused by several other projects.

In order to compare two browsers, we explore a number of research questions using a mix of data sources, including release histories, bug reporting and fixing data, and versioned usage data taken from the web servers at the School of Computer Science at the University of Waterloo.

We start by exploring the release histories. Figure 1 shows the lifespan of the major releases of each browser, i.e., the number of days before the next major release. The release history of Firefox consists of 8 major releases[1] [3] and 10 major releases for Chrome starting with version 0.2 [2]. On average, a new release of Firefox browser is launched every 10 months, while a new version of Chrome browser is released every 2.5 months. We note this as a major difference between the two projects: Chrome, the newer project, is much quicker to release new major versions.

[1]Despite their version numbers not ending in a zero, Firefox versions 3.5 and 3.6 are considered to be major releases by the development team.

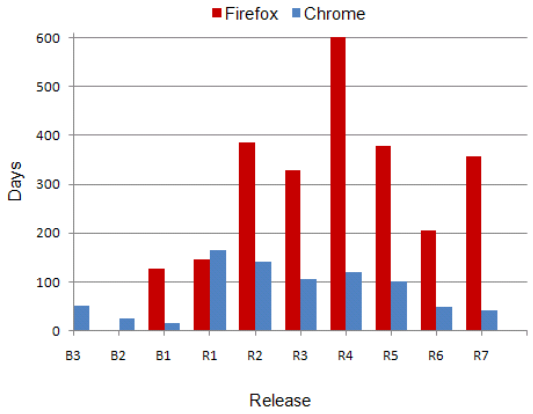

Figure 1: Lifespan of major releases for Firefox and Chrome.

Q1: Which browser is more defect prone?

First, it should be mentioned that Firefox and Chrome use different scales and terminology to denote the perceived importance of a bug: Firefox uses the term "severity" while Chrome uses the term "priority". We will use the term "importance" as a general catch-all to refer to Firefox's high severity (blocker, critical, major) bugs and Chrome's high priority (priority 0 and 1) bugs.

By analyzing the bug history of the browsers, we can plot the growth rate of defects over time (see Figure 2). For simplicity, we tagged Firefox release 0.8 as B1 and releases 0.9, 1.0, 1.5, 2.0, 3.0, 3.5, 3.6 as R1–R7 respectively. Chrome 0.2, 0.3 and 0.4 releases are labeled as B3, B2 and B1; R1–R7 are the remaining major releases for Chrome, i.e., 1.0–7.0. Firefox had a big defect spike at R4 (i.e., 2.0), where almost a quarter (i.e., 24%) of the entire project's bugs were introduced. However, restricting ourselves to only "important" bugs (the dashed lines in Figure 2), the spike is much less severe leading us to hypothesize that there was a big jump in features, but that the core remained relatively stable.

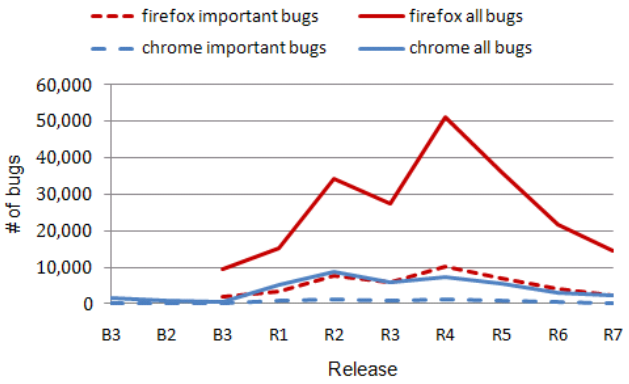

Figure 2: Bug growth over time.

Chrome had two defect spikes at R2 and R4 respectively with 20% and 17% of total bugs. Since Chrome R4 was launched in January 2010, we observe the gradual decay in the number of defects for Chrome. Also, we note that the spikes in the important bugs are also less severe.

The total number of bugs to be fixed during a Firefox release lifespan contains on average about 13% high severity (blocker, critical and major) bugs, while on average Chrome release has about 8% high priority (priority 0 and 1) bugs. Since the two projects use different scales and terminology to describe the seriousness/priority of bugs, it is hard to compare the rates directly. However, we note that the ratio of the number of important bugs to the total number of bugs per release is nearly stable for both Firefox and Chrome.

Q2: How quickly are bugs fixed?

Tables 1 and 2 display the breakdown of bugs by importance, and gives the median and average times in days taken to fix them. In our analysis, we ignored 152,333 Firefox bugs and 24,743 Chrome bugs (contributing to 28% and 35% of total reported bugs respectively) having status set to either new, available, assigned, started, expired, invalid, incomplete, unconfirmed, unknown, untriaged, or re-opened.

We calculated median and average times of a bug fix by comparing open and resolution dates from the bug report. Since some bug reports have multiple resolution dates when being re-opened or re-assigned, we considered the latest resolution date as an indicator of the time the bug was fixed.

Table 1: Breakdown of bugs by severity for Firefox browser.

severity	#_bugs	%	mdn_fix(days)	avg_fix(days)
blocker	7,752	2%	2	39
critical	39,367	10%	12	130
major	44,385	11%	9	138
normal	245,313	61%	13	157
minor	24,231	6%	17	210
trivial	10,725	3%	14	186
enhancmnt	29,614	8%	21	299
total	401,387	100%	mdn/total: 12	avg/total: 165

Table 2: Breakdown of bugs by priority for Chrome browser.

priority	#_bugs	%	mdn_fix(days)	avg_fix(days)
0	538	1%	20	26
1	5,955	13%	27	47
2	32,868	72%	28	72
3	1,904	4%	79	141
none	2,043	5%	22	37
total	45,349	100%	mdn/total: 28	avg/total: 69

Table 3 summarizes the results, showing time-to-fix, number of bugs, and the cumulative percentage of bugs fixed. Startlingly, Firefox bugs are fixed in a median time of only 12 days, with 60% of all bugs being fixed within a month, which is a very rapid resolution of bugs. The median time of a bug fix for Chrome is 28 days, with 54% of all bugs is being fixed within a month. We note that in Firefox, it is common to introduce "tracker" bugs that mark the progress of several related bugs over time. These tend to stay open for very long periods, and should be viewed as an artifact of the bug-tracking philosophy of the Firefox project, rather than a measure of the bug-fixing effectiveness. Manual inspection suggests that these tracker bugs account for many of the outliers that skew the average to be so far from the median for Firefox.

We believe that time to fix higher importance bugs is probably a better measure of responsiveness than lower importance bugs, since developer resources are limited. The

median resolution times of important bugs are 8 days and 27 days for Firefox and Chrome respectively. However, the difference between the two projects was not statistically significant (p-*value*=0.1). Thus, both Firefox and Chrome developers are fast in fixing high priority bugs.

Table 3: Resolution time for browsers.

fix_time:	bugs_firefox	%	bugs_chrome	%
same day	100,050	25%	682	2%
3 days	155,875	39%	2,298	5%
a week	183,603	46%	4,864	11%
a month	239,846	60%	24,615	54%
> month	161,541	40%	20,734	46%

3. USER ADOPTION

Q3: How do adoption trends differ?

In the remaining questions, we added data from web traffic logs for the `cs.uwaterloo.ca` subdomain between February 2007 and November 2010. This data contained the browser and version information for all uses of this domain in the indicated period. Our web usage data has some gaps due to the specifics of the university's backup routine (see Figure 3).

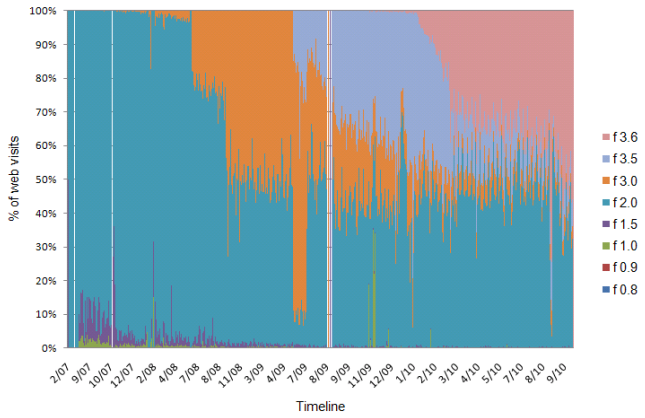

Figure 3: Adoption trends of Firefox browser.

Figure 3 shows the adoption trends of the 8 major releases of Firefox for users of our subdomain in the indicated period. Since our web traffic data goes back only as far as February 2007, the information on the popularity of older releases such as Firefox versions 0.8, 0.9, 1.0 and 1.5 is missing or limited. Version 2.0 dominates the picture, while versions 3.0, 3.5, and 3.6 have similar time of being a browser of user's choice.

Figure 4 shows the adoption trends of Chrome browser of ten releases over 2.2 years. At first, this graph seems startling, as users appear to jump en masse to newer versions immediately as they become available. However, this is because Chrome uses an automatic update mechanism that forces updates to occur quietly in the background as soon as they are officially released. The only flexibility a Chrome user has is whether to follow the current stable or development ("dev") release; if they choose the latter — and our data suggests that an astonishing 30% of Chrome users do — they may start using new versions early, which accounts

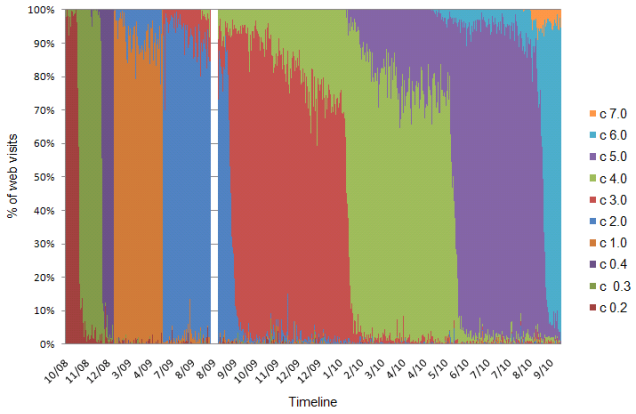

Figure 4: Adoption trends of Chrome browser.

for the leading tail for each version before the spike of the official release.

Q4: How stale is your browser?

Since Chrome major releases occur more often than Firefox, it is not surprising that Firefox accesses should often involve "older" versions than those of Chrome. Consequently, we decided to compute the "staleness" of accesses, which we define as the number of days after a new version has been released that an older version is used. Staleness is in part a measure of the success of the particular browser version: how reliable and how widely deployed it was. But it is also a measure of the individual user: How likely is she/he to be using an up-to-date version?

Table 4: Staleness of a release in days.

firefox	0.8	0.9	1.0	1.5	2.0	3.0	3.5	
mdn_stl	1,512	1,241	1,362	1,036	529	232	126	
stl_%	100%	100%	100%	100%	75%	56%	39%	
chrome	0.2	0.3	0.4	1.0	2.0	3.0	4.0	5.0
mdn_stl	350	305	378	217	153	120	63	13
stl_%	37%	18%	4%	9%	10%	6%	3%	4%

The results are summarized in Table 4, and are quite surprising. Firefox, with its long history as a stable tool that has been widely deployed, is commonly used as stale versions, sometimes years after the initial release. This may be explained in part by the fact that Firefox is commonly used in commercial operating systems installations, which are by nature slow to perform system-wide upgrades due to the cost and risk involved. Chrome, on the other hand, has a remarkable low staleness rate; for example, only 3% of all accesses of Chrome 4.0 are stale.

We must note that the Firefox staleness statistics are somewhat misleading; our records only go back as far as August 2007, so all of our recorded uses of the early versions are marked as stale. What is notable, therefore, is not the percentage but the sheer volume of users who have stale versions of Firefox. Chrome, on the other hand, performs automatic upgrades as soon as new versions are available; stale usage of Chrome is likely due to users who have downloaded new versions but not yet restarted their browser.

4. POPULARITY AND RELIABILITY

Q5: Does browser relative popularity change over time?

To get a sense on the relative popularity of the web browsers over the past several years, we extracted the number of hits on the `cs.uwaterloo.ca` subdomain for each of the major web browsers (Figure 5). They can be compared to the results during the same period as tracked by `w3schools.com` [1]. At first glance, the difference in popularity of the Internet Explorer browser between the two data sets seems curious. However, further study shows that usage percentages also vary significantly in the well known usage tracking sites [4]. Additionally, the heavy reliance on Unix at the University of Waterloo, and the pre-installation of Firefox on these Unix systems, might account for the relatively high Firefox traffic observed.

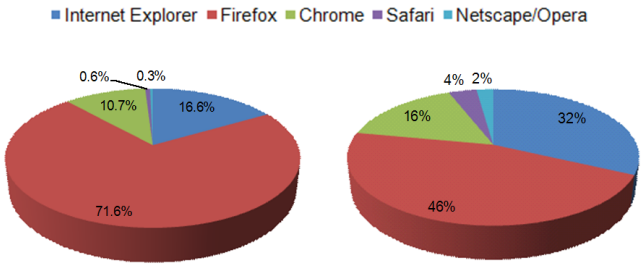

Figure 5: Usage share of web browsers for 2010, UW (left) and w3schools.com [1] (right).

From the web usage data, we observed that Firefox's relative popularity was increasing rapidly for several years, but has recently marginally declined, while Chrome has been steadily (if slowly) increasing.

Q6: Does the volume of defects affect the popularity of a browser?

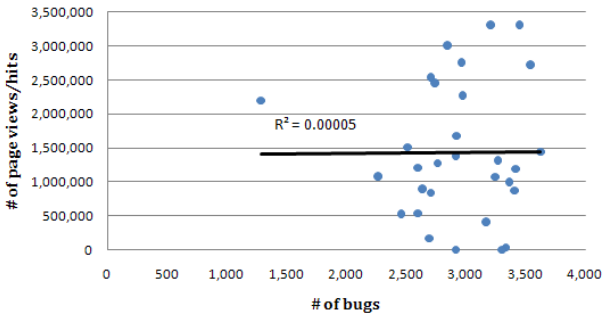

Figure 6: Correlation of the popularity of Firefox browser to the number of defects.

Finally, we decided to examine whether the number of software defects affects the popularity of a browser. By performing simple linear correlation analysis between number of bugs and number of page visits, we found that there is no meaningful relationship between the popularity of Firefox browser and number of defects, where correlation coefficient r=0.007105159, p-*value*=0.0000000101 (Figure 6). And perhaps surprisingly, we found that for Chrome there

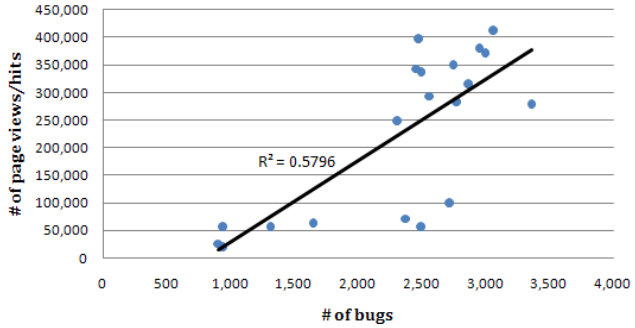

Figure 7: Correlation of the popularity of Chrome browser to the number of defects.

is a moderate *positive* correlation between the two variables with r=0.7613 (p-*value*=0.0000016698) (Figure 7), i.e., a system with *more* bugs appeals to more users. We hypothesize that the more people adopt a software system, the more bugs are likely to be reported. That is, since Chrome is in a period of strong growth in its user base, this may simple be a "sideways" measure of its popularity.

5. CONCLUSIONS

We set out to study the Firefox and Chrome web browsers by examining the release histories, bug reporting and fixing data, and usage statistics to see if we could detect significant difference in the two systems and their user base. What appears to have emerged are two distinct profiles: Firefox, as the older and established system, with long product version cycles but short bug fix cycles, and a user base that is slow to adopt newer versions; and Chrome, as the new and fast evolving system, with short version cycles, longer bug fix cycles, and a user base that very quickly adopts new versions as they become available (due largely to Chrome's mandatory automatic updates).

6. ACKNOWLEDGMENTS

We thank the Computer Science Computing Facility (CSCF) of the University of Waterloo, with special thanks to Isaac Morland and Guoxiang Shen, for providing web traffic log files for this study.

7. REFERENCES

[1] w3schools.com. Web Statistics and Trends. `http://www.w3schools.com/browsers/browsers_stats.asp`.

[2] Wikipedia. Google Chrome — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/wiki/Google_Chrome`. [Online; accessed 28-November-2010].

[3] Wikipedia. Mozilla Firefox — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/wiki/Mozilla_Firefox`. [Online; accessed 28-November-2010].

[4] Wikipedia. Usage share of web browsers — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/wiki/Usage_share_of_web_browsers`. [Online; accessed 26-January-2011].