COLLEGE *of* CHARLESTON

DEPARTMENT OF COMPUTER SCIENCE

Contributing to Mozilla Firefox's Free Open-Source Code

Thomas Setzler, Sydney Jackson, Jason Adler and Levi Hagan

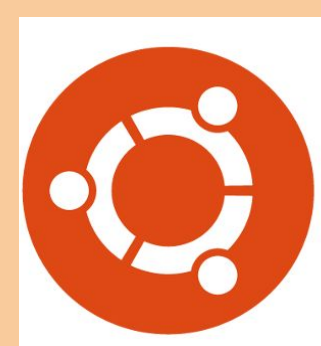### College of Charleston, Department of Computer Science

Spring 2020

Goals

Our goal for this semester was to familiarize ourselves with CO-FOSS projects and the methods of contributing. We applied what we have learned during our time at the College of Charleston to work as a team of people with different strengths to act in the varying roles of a typical software development team.

About Firefox

Mozilla's Firefox browser is the only browser made by a non-profit organization. Following a pattern of universal design, Firefox has been translated into about 90 different languages by volunteer contributors and the Access Firefox team ensures that Firefox is accessible to all users, including those with visual and motor impairments.

Firefox is also dedicated to protecting the online privacy of it's users by proving Private Browsing, Guest Browsing, and Do Not Track features.
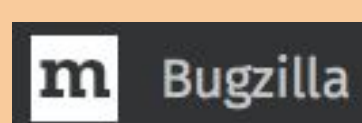
Tools

Linux Virtual Machine: All group members worked on an Ubuntu 18.04 VM for standardization across work

Phabricator: The Mercurial-based tool by which Mozilla accepts and tests all bug fixes to the code base

Bugzilla: The online system Mozilla uses for reporting, tracking, and triaging bugs across all products

Firefox Developer Kit: The developer version of Mozilla Firefox

GitHub: Version control system used to share details of project with team and class

Bug Fixes

1. **1476639 *Remove scrollToIndex*:** Remove a single use method (scrollToIndex) from the test files
2. **1609821 *Make about:about an HTML file:*** Convert an XHTML file into an HTML file and update dependencies
3. **1609822 Make about:checkboard an HTML file:** Convert an XHTML file into an HTML file and update dependencies
4. **1609823 *Make about:crashes an HTML file:*** Convert an XHTML file into an HTML file and update dependencies
5. **1587496 *Update nslBrowserHandle*r:** Replace nslBrowserHandler with a more efficient method, and update function calls
6. **1622743 *BrowserTestUtils.waitForCondition*:** Change BrowserTestUtils.waitForCondition to use a more efficient method
7. **1624236 *Do Not Use Else:*** Update the function logic to be more effective by utilizing break statements
8. **1624232 *Do Not Use Else:*** Update the function logic to be more effective via return statements

Submitting a Patch

1. Run `$ hg update` to update to the current working version of the directory
2. Create or update file with desired changes
3. Run `$ ./mach build` to build browser with desired changes
4. Run `$ hg commit --message "Bug <ID> - <ID>: <change description> r=<reviewer>"` to commit the changes with information about the fix and alerts the designated reviewer
5. Create a Mozilla Phabricator account using an existing Bugzilla account
6. Run `$ moz-phab install-certificate`, then paste in the generated API key
7. Run `$ moz-phab submit` to submit the patch to phabricator
8. Wait for the reviewer to respond

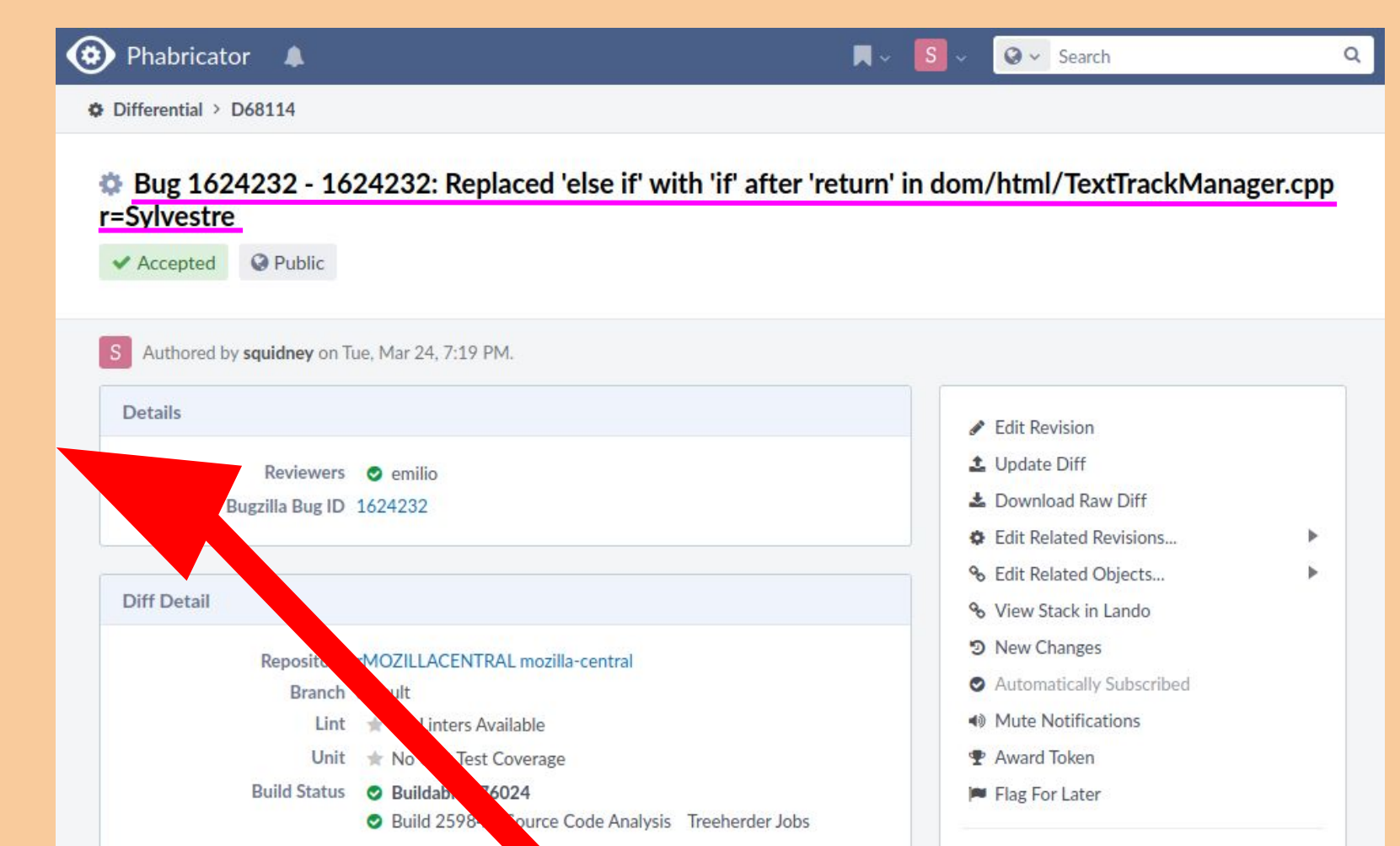Figure 1: Running `moz-phab submit` in the command line and the URL to view the patch on Phabricator (Bug #1624232)

Figure 2: Navigating to the given URL to view the patch on Phabricator

Acknowledgements

COLLEGE *of* CHARLESTON
SCHOOL OF SCIENCES AND MATHEMATICS
1770

moz://a