A new version of Firefox is available

Rapid Release of Quality Firefox Products

Lukas Blakk & Sylvestre Ledru
Who are we?

Lukas
- Mozillian since 2006
- Release Engineer for 3 years
- Release Manager for 3 years

Sylvestre
- Has been at Mozilla for a year
- Debian Developer
- LLVM/Clang developer
A Mozilla tradition
Pictures of our children
About: Firefox

• Web browser with ~500 million users
• About 12.5M Lines of code
• Only (major) browser developed by a non-profit

• Support 4 operating systems:
  • Microsoft Windows XP => 8 (32 & ~64 bit)
  • GNU/Linux
  • Mac OS X
  • Android
We need to be where our users are so we're going to get Firefox on iOS #mozlandia
Firefox Releases

- One major release every 6 weeks
- ESR – lock-step with major releases, security focus
- In parallel, 3 other feedback branches:
  - Nightly - updated daily with recent code changes
  - Developer Edition (a.k.a Aurora) + aurora on mobile - updated on change
  - Beta – 2 per week Desktop – 1 for Mobile
- Managed by 4 people (3 staff, one volunteer)
More about ESR

Major ESR (Extended Support Release) version is created every 8 months, with two cycles of overlap with the prior version

- Security (high/critical) every 6 weeks (in sync with Firefox)
- Other issues by request from mailing list community
- Used by some GNU/Linux distributions (example: Debian), major companies, university, etc
- Active community mailing list, various sized orgs
- Base for Thunderbird & Seamonkey
Some Lovely Graphs
Healthy open source project?

Activity

30 Day Summary
Dec 9 2014 — Jan 8 2015
3340 Commits
327 Contributors
including 40 new contributors

12 Month Summary
Jan 8 2014 — Jan 8 2015
59599 Commits
Up +15113 (33%) from previous 12 months
1212 Contributors
Up +180 (17%) from previous 12 months
More contribution stats!

Developer per version

Changeset per version
Many companies & projects contributing

- Tor
- Google
- Microsoft
- Telefonica
- Red Hat
- Cisco
- Adobe
- FreeBSD
- ...

mozilla
Release management

Making sure that Firefox releases are released in good shape and on time, deciding when to build & when to wait for patches.

Coordinate work between:
- Quality Engineering
- Stability
- Release Engineering
- Firefox Developers
- L10n
- Security
- User Advocacy
- Addons
- Marketing
Train model
Quality

• A web browser is comparable to an OS (network, JIT, video, sound, parallel, etc)

• Complex code base with legacy code

• Hard to test all cases + test suites are time consuming
What's in the release?
Bug tracking

For us - **Everything** happens on bugzilla.mozilla.org

Cunning use of FLAGS:

1) Tracking
2) Status
3) Per-Patch approval
4) Release note
5) Need-Info
6) Blocking (not by us, but informs us)
Version Specific Status flags

1) Allow the triangulate needs of a bug across several releases in parallel

2) Possible values:
   - ?
   - Unaffected
   - Affected
   - Wontfix
   - Verified
   - Fixed
   - Disabled
   - Verified Disabled
1) Developers, project managers, users, QE, etc request tracking « ? » for a XX version

2) Release manager triage and accept « + » / reject « - ».

3) Criteria:
   • Top crash
   • Security
   • Regression (recent)
   • Impact to users

4) Release managers coordinate with engineers to get the bug fixed before the release
Uplift requests

1) Formal request to have patch(es) uplifted to aurora, beta, esr, or release
2) A form auto-populates when request is made

Approval Request Comment

[Feature/regressing bug #]: dunno
[User impact if declined]: we will be unable to run jsapi-tests on beta because failures will be masked by this test and we will accidentally introduce a critical bug resulting in the complete destruction of Firefox's reputation and the Web will turn away from open solutions and stagnate again, preventing the creation of better worldwide communication mechanisms that turn out to be critical
[Describe test coverage new/current, TreeHerder]: This has been in m-c for a day and tested locally.
[Risks and why]: There's a chance this could break video decoding on Windows completely.
[String/UUID change made/needed]: None.

3) Approved patch gets uplifted to the branch(es), CI rebuilds + tests to confirm merge safe
4) If possible, QE verifies the fix before release
Time to release
About: Beta
Reliable Process Works
Release Week

- Our final RC has shipped to Beta pop
- Collect feedback from users, crash data
- Checking for ADP crash or other last-minute risk
- Sign off for release
- Release notes prepared and reviews
- Merge branches, kick off Beta 1 of version.next
- DevEd updates disabled for the week
Release Day

- Published Tuesday at 15:00 European time
- Updates only enabled for 25% users for 24 hrs
- Start watching Bugzilla, Input, crash data
Post-Release Feedback

- Even with all that, we have had failures
Example: OMTC « Off-main-thread compositing »

- Enabled on Mac OS X for a while (Firefox 24)
- Enabled on Windows during the 33 cycle
- A few top crashers fixed during the aurora & beta cycle
- Sign off of all teams for the 33 release
- For some users, black screen instead of Firefox
What Happened: OMTC

- Some graphic drivers are older, not supported
- We can't test on all devices/systems
- Our telemetry didn't catch any issues
- Our aurora/beta users probably update their graphic drivers and/or use newer ones
- Hard identifying the signal from the noise
- Some users who had the bug think « it is so big that someone will already have reported this »
Lessons Learned: OMTC

- Potential solutions:
  - Do more testing on older devices
  - Get more data of users configurations (graphic cards, drivers versions, etc)
  - Could there be an automated test for a black screen?
Big Picture – Train model

- A feature « foo » lands in nightly
- Natural conflict:
  - Push new feature to advance the web
  - VS
  - More time developing and iterating on quality
- Nightly and DevEd do not have as many users as beta
- Features really get tested in beta first
- Best case: The developer debugs code he/she wrote 6 weeks ago
- Worst case: 18 weeks ago
What works – Train model

- Releases are no longer a “big deal”
- Missing a train is also not a “big deal”, less pressure to push features that aren't ready
- Flags for enabling/disabling help get test coverage over a few cycles on any release channel
- Small changes can have impact – shorten update throttling, push out more betas

- Successfully slows the incoming changes before release
More?

Don't miss the Mozilla devroom tomorrow
Room UD2.218A - 9:00 to 17:00

• What's new in Firefox?
• The Future of JavaScript
• Servo (the parallel web browser) and YOU!
• Privacy features for Firefox for Android
• And other cool talks
Questions?

Lukas Blakk
@lsblakk
lsblakk@mozilla.com

Sylvestre Ledru
@sylvestreledru
sylvestre@mozilla.com