

[Vincent van Gogh - Self-Portrait](#) by Vincent Van Gogh, from Google Art Project. Public domain, via [Wikimedia Commons](#).



Multimedia Quarterly Review

Wikimedia Foundation - Q2 of FY14-15

Agenda

Team intro - 1 minute

What we said - 2 minutes

What we did - 8 minutes

What we learned - 3 minutes

Metrics and callouts - 6 minutes

What's next - 5 minutes

Asks - 5 minutes

Photo Credit:

CC BY-SA 3.0 Anna Lantz and Myleen Hollero / Wikimedia Foundation
CC BY-SA 3.0 Matthew Roth
CC BY-SA 2.5 Gilles Dubuc
CC BY-SA 3.0 Sage Ross
CC BY-SA 3.0 Lane Hartwell / Wikimedia Foundation

Team and staffing numbers



Mark Holmquist



Tisza Gergő



Gilles Dubuc



Fabrice Florin (PM in Q2)



Guillaume Paumier (Metadata cleanup drive)



Pau Giner (UX)



Keegan Peterzell (Community)

FY 2013-2014

FY 2014-2015

FY 2015-2016



Q4



Q1



Q2



Q3



Q4



Q1

What we said

Goals

- Release Media Viewer improvements
- Implement server-side JS error logging
- Improve upload pipeline reliability
- Improve metrics for image performance and the upload funnel
- Continue planning and discussions on Structured Data on Commons project
- Start metadata cleanup drive to improve accuracy of metadata displayed in Media Viewer
- Respond to emergencies on all the components the team supports

Success metrics had not been established for that quarter, since the discussions about this topic happened after the associated quarterly planning meeting.

What we did

Media Viewer

Made final improvements to the UX, many of which came out or were confirmed by the community consultation:

- Large reorganization of the metadata panel
- Options menu with prominent opt-out + counterpart on the file page
- Improved experience for ellipsed content that can be expanded
- Tracking more metrics
- Thumbnail prerendering
- Performance research
- Monobook support
- GCI students tackled several small improvements from our backlog

JS server-side error logging



The goal of this project is to get visibility on the JS errors encountered by our users. We have no tracking at the moment and entirely rely on user bug reports, this project will benefit the entire organization.

We researched available tools and settled for evaluating a popular commercial (but also FOSS!) product: [Sentry](#) which is used in production by other large web properties.

A [new extension](#) dedicated to supporting Sentry in MediaWiki was written.

Upload pipeline

Since the ability to identify the biggest issues is largely dependent on collecting JS error information with Sentry, the focus of this effort has been refactoring, cleaning up and bugfixing in UploadWizard.

The main issue with UploadWizard's codebase is its high tanglement and interdependencies, making it difficult to unit test it. Mark is steadily working his way through making all the parts isolated from each other to greatly increase the maintainability of UploadWizard.

Structured data on Commons



A bootcamp was held in Berlin, hosted by the WMDE Wikidata team.

Approaching this problem from a technical and product angle has allowed us to fully realize the scope of this project.

Given that the Multimedia team no longer has a PM and that the activity on the Wikidata Query Service is keeping WMDE engineering busy, this project has been put on the backburner for now. We have a clear idea of next steps from an engineering perspective, but this is a large undertaking.

Metadata cleanup drive

Premise: Machine-readable file information enables flexibility in presentation (MediaViewer, Mobile) and facilitates consistent attribution & reuse (third-party users).

Goal: Reduce the number of files without machine-readable metadata across Wikimedia sites.

How: By adding machine-readable markers to file templates, and by adding file templates if they're missing, through direct editing and community coordination.

Emergencies responded to

- Regressions on Firefogg support and Flickr import in UploadWizard due to refactoring.
- Thumbnail chaining ran into a couple of issues in production that had to be quickly fixed (PNG support failing and later rendering failing for specific sizes).
- Ultimately thumbnail chaining caused extra sharpening on JPGs that was unwelcome by the Commons community, had to be reverted.
- During the final Media Viewer UI improvements, IE was affected by a bug where the metadata panel was open by default, had to be fixed in a hurry.

What we learned

Structured data on Commons



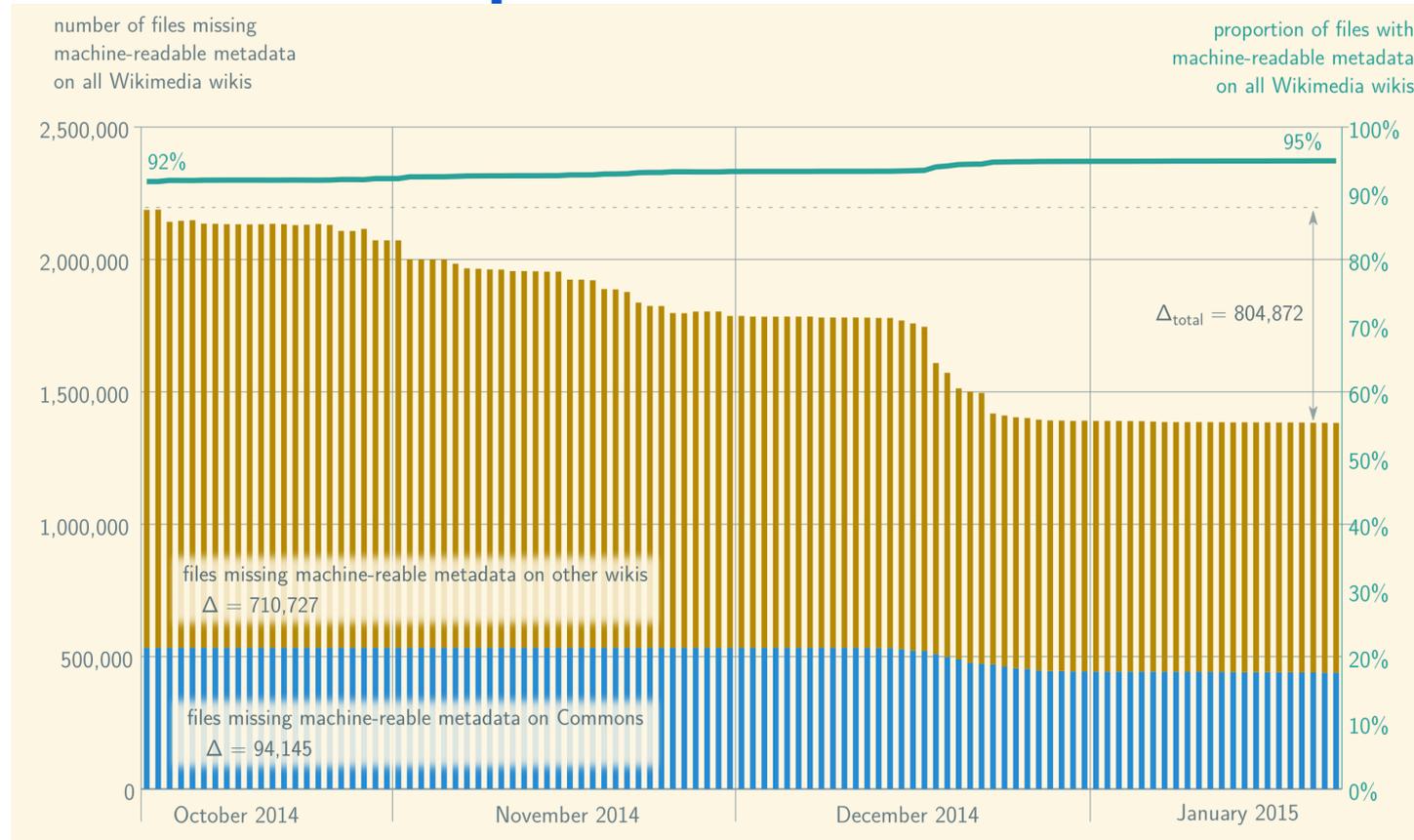
- This project will require extensive participation in the data model design. Our attempts to come up with one during the bootcamp quickly showed that it was difficult to reach consensus. Proof being that Wikidata's own information modelling is a constant work in progress and wasn't set by the Wikidata team, it has been entirely community-driven.
- A larger team is needed to tackle this.

Media Viewer

- User design research was a critical factor in the success of this round of improvements.
- Things have been a lot more quiet on the community front since the improvements have been launched. That being said, it's hard to establish causation between our efforts and increased community satisfaction, because we're not really measuring the latter.
- We've estimated that Media Viewer took around 4 man-years to build.

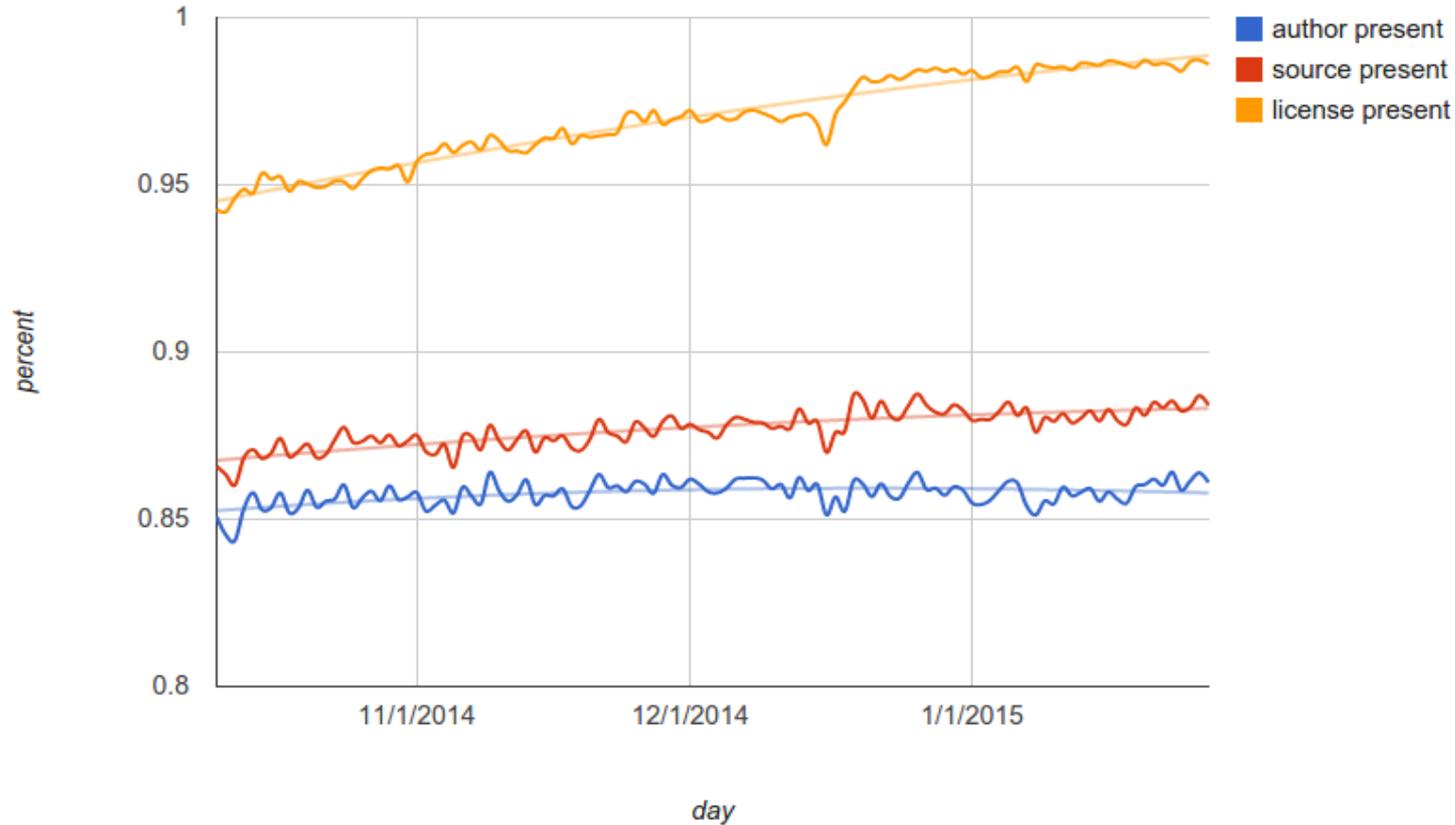
Metrics & other key accomplishments

Metadata cleanup drive



Details: https://commons.wikimedia.org/wiki/File:File_metadata_cleanup_drive_impact_Oct2014-Jan2015.svg

Attributions & Licenses



Percent of images with author, source or license information in Media Viewer, Nov.-Dec. 2014. [Source: Wikimedia Foundation](#)

Media Viewer

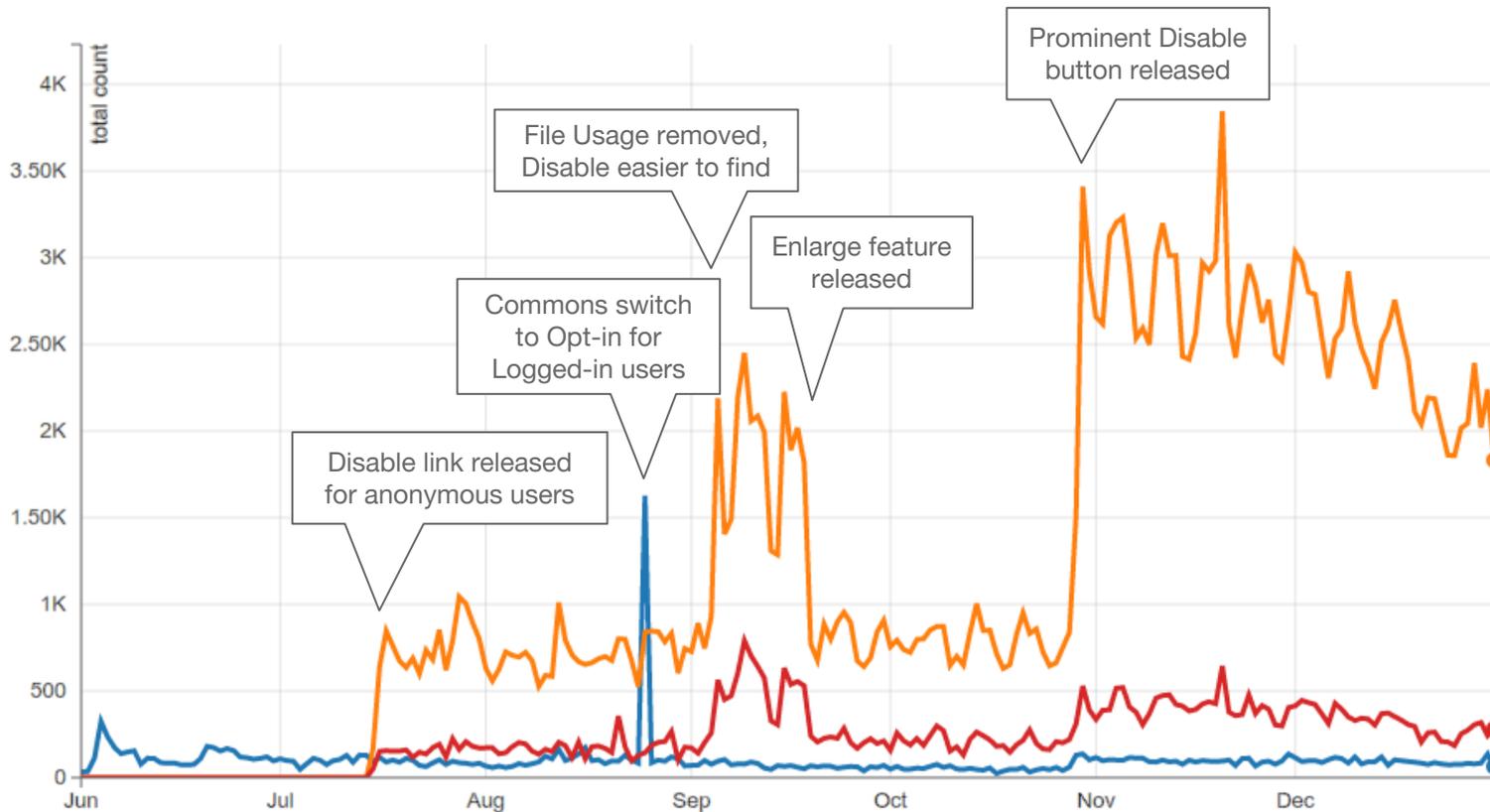
- **17M intentional image views** / day
- **99.5% enabled rate** ⁽¹⁾

(1) Percent of enabled logged-in visitors in past 30 days in Nov.-Dec. 2014.

Opt-ins/outs - Logged-in vs. anonymous

31 Dec 2014

opt-out (anon)	1.83K
opt-in (anon)	281
opt-out (loggedin)	65

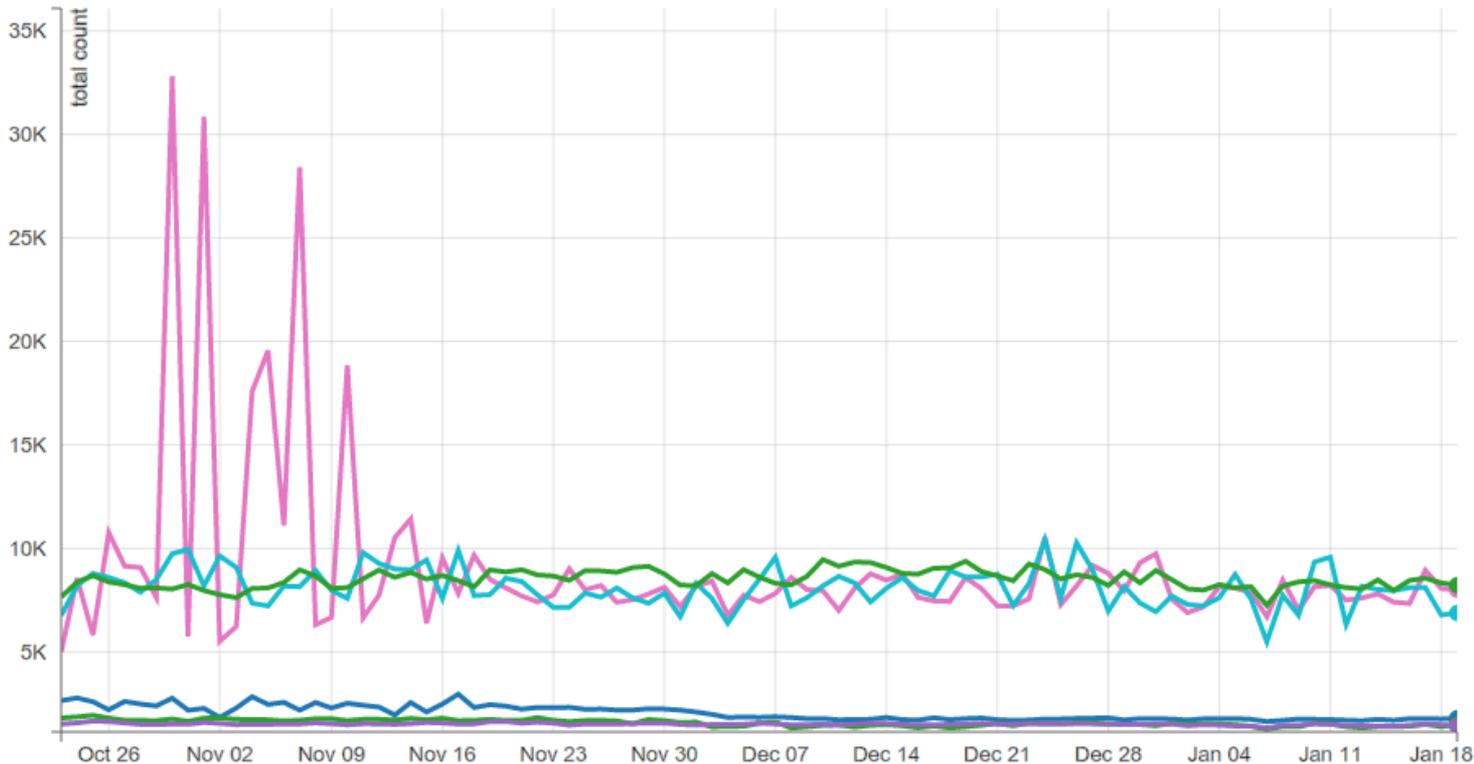


Global opt-outs by anonymous users varied as new features were released, then decreased by end of 2014. [Source: WMF](#)

Media Viewer vs. File Page

19 Jan 2015

file page (logged-in), median	1.80K
file page (logged-in), 95th percentile	8.00K
file page (anonymous), median	1.52K
file page (anonymous), 95th percentile	6.88K
Media Viewer, median	1.44K
Media Viewer, 95th percentile	8.23K



Global image load times for Media Viewer vs. File Pages are on par, for either median or 95th percentile, Dec. 2014. [Source: WMF](#)

Upload Wizard

Improvements in Code Quality

- Far fewer circular references to calling classing from child classes
- Modern development models implemented (inheritance)
- Investigation of possible implementation of a MVC framework

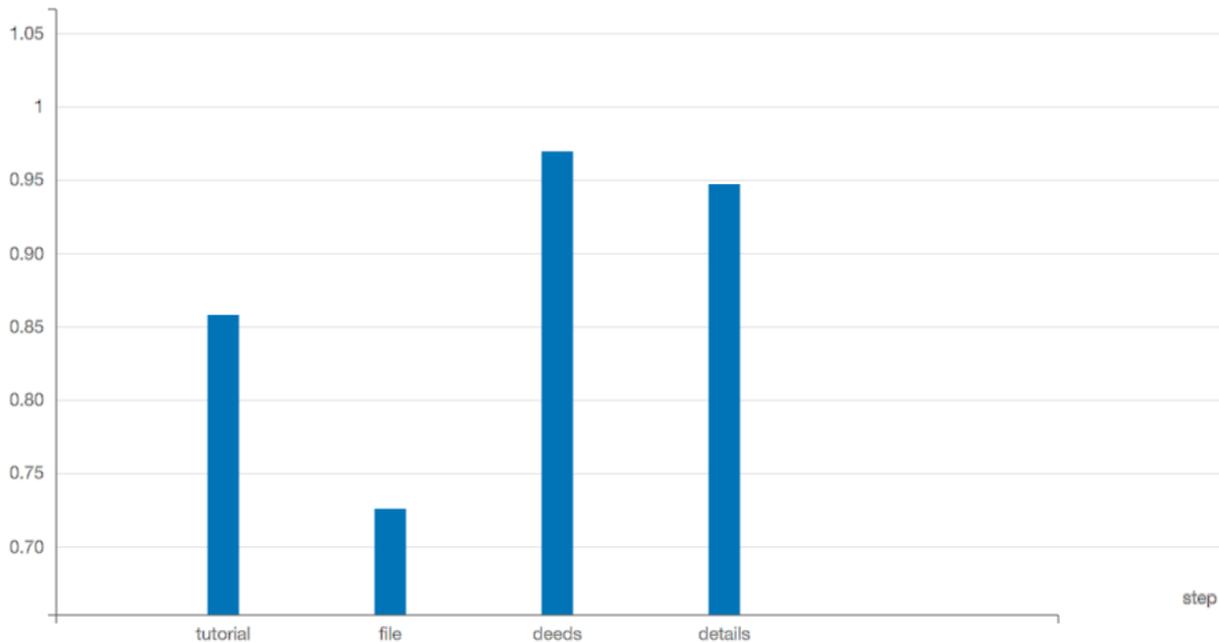
Improvements in Testing

- Unit test coverage for areas of the code that previously had none
- Better browser testing on several fronts

Upload Wizard

Funnel overview (relative, average over last 30 days)

Fraction of users surviving this step

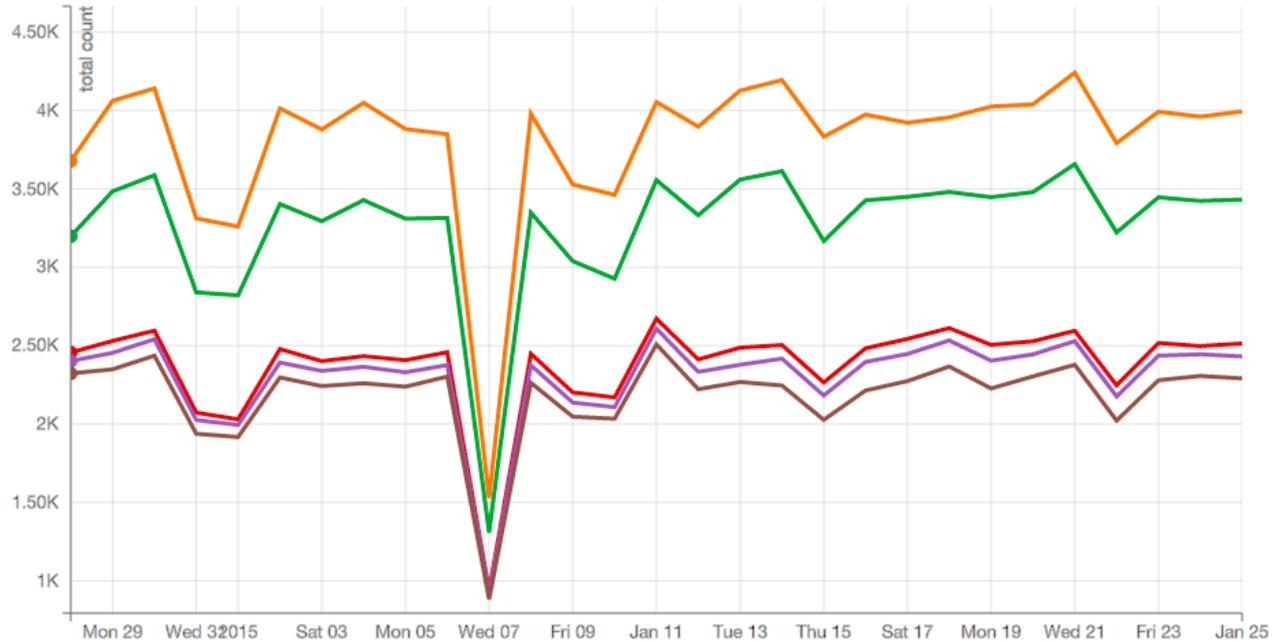


Funnel analysis for Upload Wizard. The data displayed is the average over the last 30 days. The value for each step is the percentage of users surviving this step to the next one.

Upload Wizard

Funnel analysis (absolute over time)

28 Dec 2014	
tutorial	3.68K
file	3.20K
deeds	2.46K
details	2.40K
thanks	2.32K



Funnel analysis for Upload Wizard. Displays the absolute survivors over the last 30 days.

What's next

JS server-side error logging



- Deploying Sentry and its companion extension to Beta
- Devising a plan for production deployment
- Advertise its availability internally

Success metrics

- Go from not collecting any JS errors anywhere to collecting all JS errors on Beta
- Get at least another team to use it (hopefully JS error logging will sell itself)

Platform

- Investigate a better solution for thumbnail storage. This is a long-standing request from the Ops team.

Success metrics

- Identify the most efficient strategy that will let us get rid of the storage waste currently caused by Swift. 32TB is currently used for thumbnail storage that is needlessly redundant (3 copies of each thumbnail stored across the Swift cluster, we should have only 1)

UploadWizard

- More refactoring and unit testing (and possibly stability and UX improvements enabled by that).
- Bugs encountered while undertaking this effort will be fixed, but might not be as high-profile as the ones we'll handle once we have JS error visibility.

Success metrics

- Significantly increase UploadWizard test coverage. We've tried using tools to quantify that properly, but it's quite challenging since out of the box these tools measure core JS in addition to UploadWizard's.
- Completely remove interdependencies between the main parts (step controllers) of UploadWizard

Asks

Analytics

- More streamlined metric setup. We're currently finding ourselves spending a lot of time doing things manually that should be automated. We can actually improve some of that ourselves, so it's not a 100% ask.
- We look forward to the new dashboard tools, as limn has proven to be clunky with the amount of metrics we're tracking.

Product

- More diligent research about which Multimedia projects would have the highest impact. Media Viewer seems like it might not have been worth the amount of man-years put into it compared to the impact it had.
- Relatedly, better scoping. Feature creep was a significant problem with Media Viewer. Besides the features where we caved in and built things out of scope, there's a large backlog of things we were pressured to build next and managed to push back, at the expense of a lot of energy.
- Giving us the necessary time to work on metrics. The pressure to add last-minute features before the launch was too high, and we weren't able to track some metrics that we wanted to. The hunger for metrics shouldn't come from engineers alone, or at least it shouldn't be pitted against features.
- Putting resources into measuring qualitative metrics (eg. user satisfaction)

C-level

- Make a clear decision about Multimedia resources. Currently the team is too small to take on any large project like Structured data on Commons.
- Should our team be disbanded in favor of more project-focused ones?
- Should it be expanded so that we can take on important projects like Structured data on Commons?
- This meeting took a lot of time to prepare, what did you learn out of it that you didn't know already? I think there's room to streamline how C-level management gets up to date on what the teams are doing, quarterly reviews feel like a heavy process to me.
- Is it wise to mix platform and feature responsibilities within the same team? So far it has seemed counter-productive to us from an engineering/internal team resourcing standpoint.