Wikimedia Foundation metrics meeting
3 September 2015

Agenda

Welcome

Community update

Metrics

Research

Feature

Q&A

Photo by Carlos Perez Couto, CC by SA 3.0

Welcome!

Requisition hires:

- Joshua Minor - Engineering - SF
- Jake Orlowitz - Community Engage - SF (conv)
- James Holder - Talent & Culture - SF
- Eliza Barrios - F&A - SF
- Peter Hedenskog - Engineering - Sweden
- Brendan Campbell-Craven - F&A - SF
- Sandra George - F&A - SF (conv)

Contractors, interns & volunteers:

- Samantha Becker - Engineering - SF
- Charles Roslof - Legal - SF
- Shirley Nguyen - F&A - SF
- Leighanna Mixter - Legal - SF
- Eileen McNaughton - Engineering - NZ

WIKIMEDIA
TALENT&CULTURE

Anniversaries

Mark Bergsma (9 yrs)
Erik Zachte (7 yrs)
Michelle Paulson (7 yrs)
Santhosh Thottingal (4 yrs)
Chris Johnson (4 yrs)
Dan Andreescu (3 yrs)
Kaity Hammerstein (2 yrs)
Dan Garry (2 yrs)
Jorge Vargas (2 yrs)

Joel Krauska (2 yrs)
Ellery Wulczyn (1 yr)
Bahodir Mansurov (1 yr)
Jeff Hobson (1 yr)
Bartosz Dziewonski (1 yr)
Marti Johnson (1 yr)
Marcel Ruiz Forns (1 yr)
Sati Houston (1 yr)
Jake Orlowitz (1 yr)

WIKIMEDIA
TALENT&CULTURE

Community update

WIKIMEDIA
FOUNDATION

Summer highlights

- **Documentation Directory**
  - Helping find outreach-related documentation (link)
- **Visual editor**
  - “Visual Editor made it sooooo much easier for me to edit. I discovered that I love editing Wikipedia…” - MegaLibraryGirl, 600 edits July-August
- **Outreachy**
  - 10 projects complete — congratulations to all who participated!

Two conversations

- **Technical spaces Code of Conduct**
  - “As contributors and maintainers of Wikimedia technical projects, and in the interest of fostering an open and welcoming community, we pledge...”
  - Draft on mediawiki.org
- **Reimagining grants**
  - Goal: “to better support people and ideas in the Wikimedia movement”
  - Consultation on meta
  - There Is A Deadline: **Sept. 7**

Metrics

WIKIMEDIA
FOUNDATION

Discovery

WIKIMEDIA
FOUNDATION

What are we working on?

- **Search**
  - Make our content searching systems better across all wikis
- **Wikidata query service**
  - Allow users to run arbitrary queries on the data in Wikidata
- **Maps tile service**
  - Generate maps tiles that can be used to back map-based features
- **Analysis**
  - Build understanding of how people use search and what they need

What are our goals this quarter?

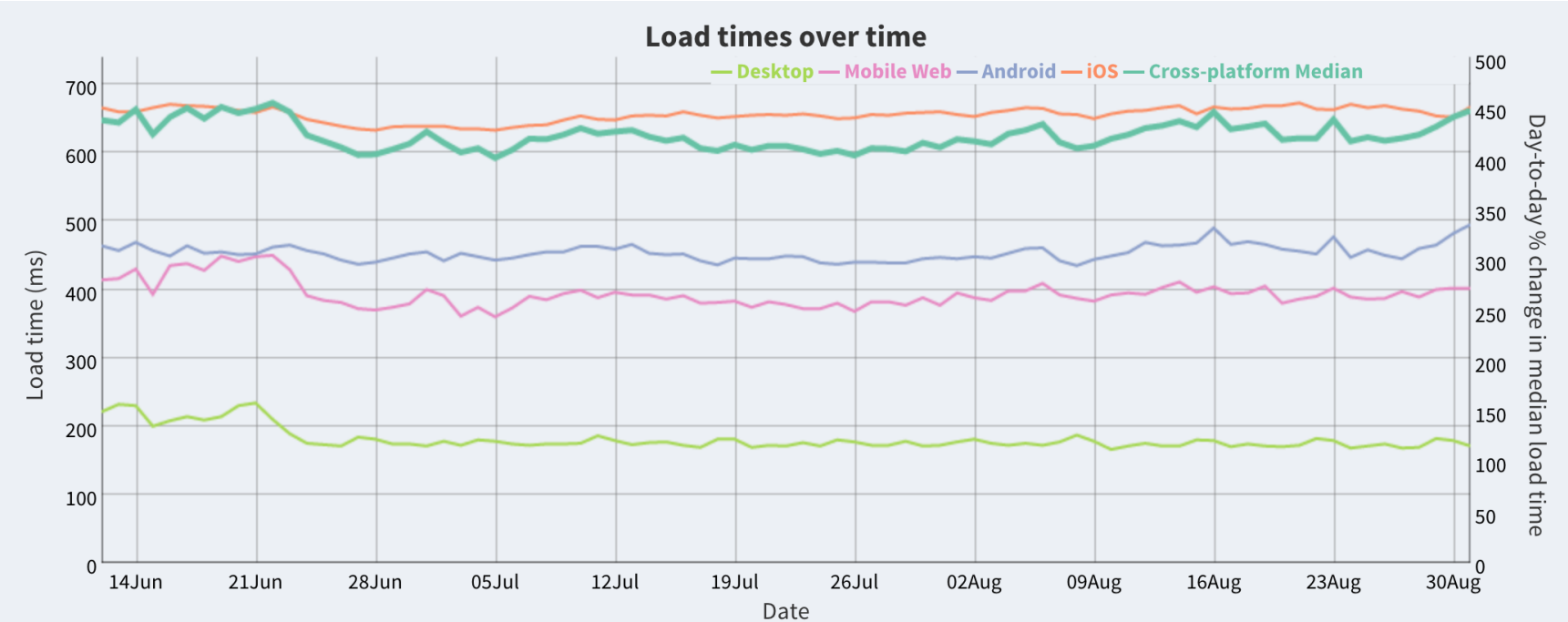
- **Search**
  - Cut the zero results rate in half
- **Wikidata query service**
  - Deploy beta service, monitor usage, collect user feedback
- **Maps tile service**
  - Deploy beta service, monitor usage, collect user feedback
- **Analysis**
  - Understand how relevant the results we serve to our users are

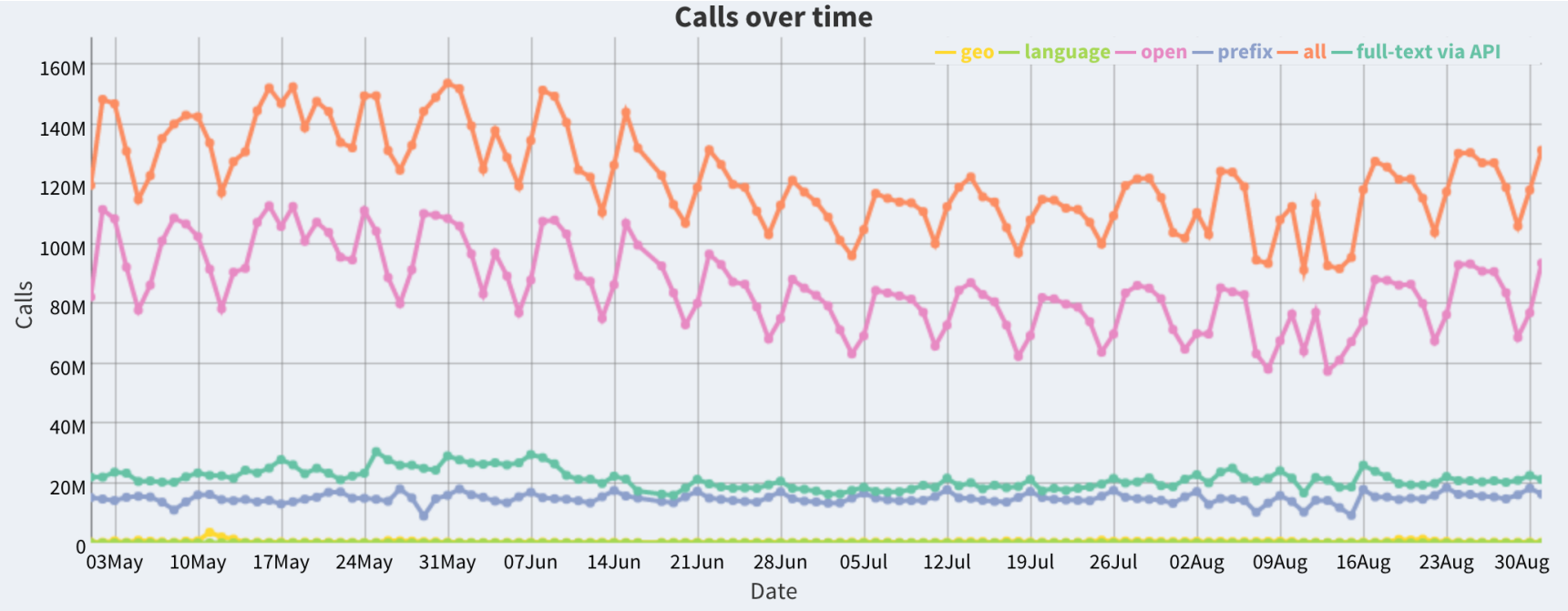
http://bit.ly/jul15goals

What are our key performance indicators?

- **User satisfaction**
  - Users should get relevant search results and be satisfied with them
- **User-perceived load time**
  - Searching should be fast and snappy
- **Zero results rate**
  - If we give users no results, they've not found what they wanted
- **API usage**
  - Third-parties should be able to build experiences based on our search

http://searchdata.wmflabs.org/metrics/


Load times over time
Desktop
Mobile Web
Android
iOS
Cross-platform Median
Load time (ms)
0
100
200
300
400
500
600
700
Day-to-day % change in median load time
0
50
100
150
200
250
300
350
400
450
500
14Jun
21Jun
28Jun
05Jul
12Jul
19Jul
26Jul
02Aug
09Aug
16Aug
23Aug
30Aug
Date


Calls over time
geo
language
open
prefix
all
full-text via API
160M
140M
120M
100M
80M
60M
40M
20M
0
Calls
03May
10May
17May
24May
31May
07Jun
14Jun
21Jun
28Jun
05Jul
12Jul
19Jul
26Jul
02Aug
09Aug
16Aug
23Aug
30Aug
Date

Why do we care about the zero results rate?

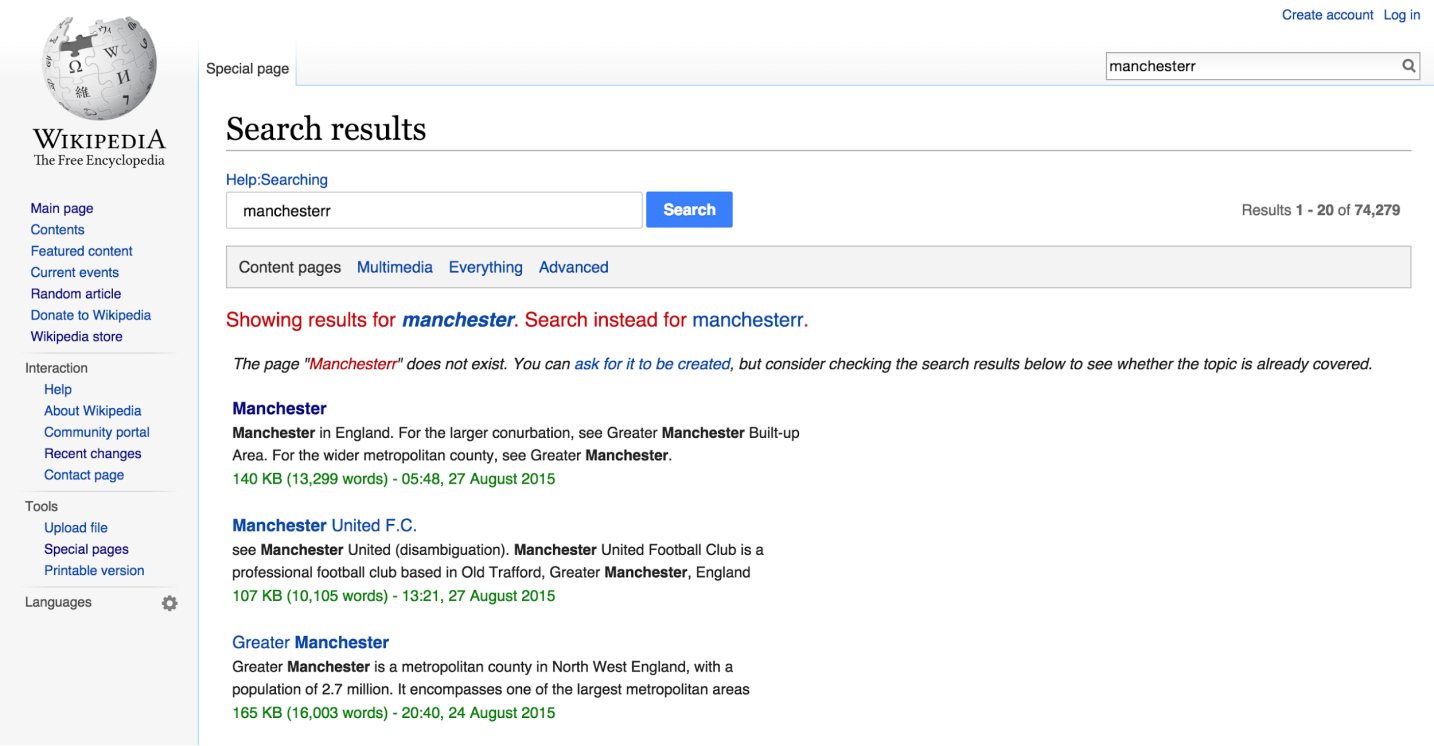
We want users to get relevant results for their search queries.

If we give them nothing, we've not given them anything relevant…

Or have we?

What have we done?

If the searcher gets zero results, and also a suggestion, just run the suggestion.



What have we done?

Run A/B tests to figure out if there are better search parameters to use.

http://bit.ly/zeroABtest

```
15252 'wmgCirrusSearchUserTesting' => array(
15253         'default' => array(
15254                 'suggest-confidence' => array(
15255                         'sampleRate' => 10,
15256                         'buckets' => array(
15257                                 // control bucket, retain defaults
15258                                 'a' => array(),
15259                                 // test bucket, alternative suggestions
15260                                 'b' => array(
15261                                         'wgCirrusSearchPhraseSuggestSettings' => array(
15262                                         'mode' => 'always',
15263                                         'confidence' => 1.0,
15264                                         'max_errors' => 2,
15265                                         'real_word_error_likelihood' => 0.95,
15266                                         'max_term_freq' => 0.5,
15267                                         'min_doc_freq' => 0.0,
15268                                         'collate' => false,
15269                                         'collate_minimum_should_match' => '3<66%',
15270                                         'smoothing_model' => array(
15271                                             'laplace' => array(
15272                                                 'alpha' => 0.3,
15273                                             ),
15274                                         ),
15275                                         ),
```

What have we done?

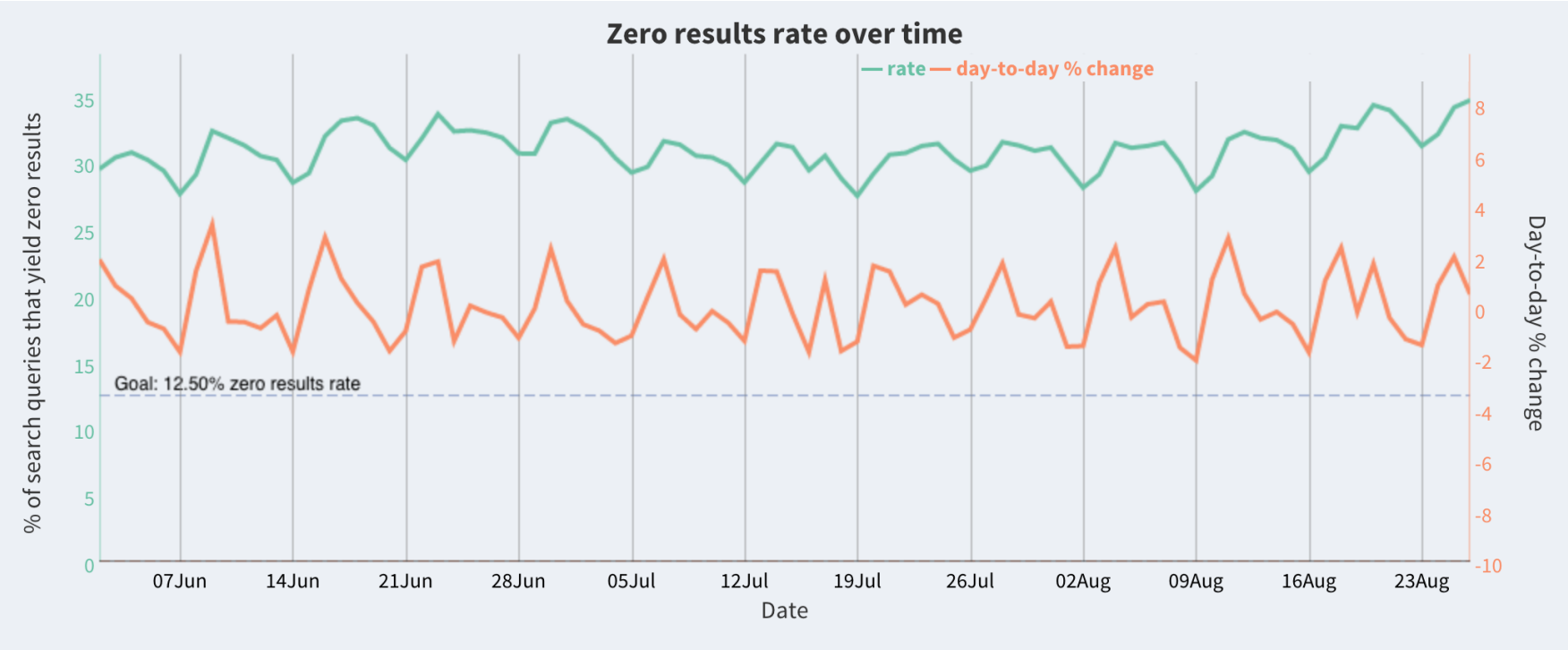
Figure out who's getting zero results, why it's happening, and fix it.

http://bit.ly/zeroresults

Summary Table [edit | edit source]

Query type	Sample 7/10	Sample 7/17	Sample 7/24	% of zero rate (min / max) of 500K samples		Most affected wikis
DOI	15393	96998	50181	3.08%	19.40%	en, nl, ja, zh, war, vi, uk, sv, pt, pl, no, ko, it, id, hu, fr, fi, fa, de, cs, ceb, ca, ar, es, ru
Unix timestamps	42650	26351	28089	5.27%	8.53%	en, it, ru, ja, fa, tr, nl, he, ar, id, cs, hi, vi, ro, hu, uk, etc.
"Article_title" AND "title of link taken from article"	10524	8174	16657	1.63%	3.33%	en
TV Episodes / Movies—"..." film	7989	7878	8794	1.58%	1.76%	en, nl, de, fr, ja
quot	7768	5888	6297	1.18%	1.55%	en
term+term+term country	6725	3437	5645	0.69%	1.35%	es, en
paint	3554	1917	1094	0.22%	0.71%	en
Highly repeated searches	892	1186	3019	0.18%	0.60%	?
term+term+term	2247	1382	2536	0.28%	0.51%	es, en
{searchTerms}	2314	1909	1997	0.38%	0.46%	ru
## <countrycode> tel fax	572	33	1293	0.01%	0.26%	de

How is the zero results rate looking?



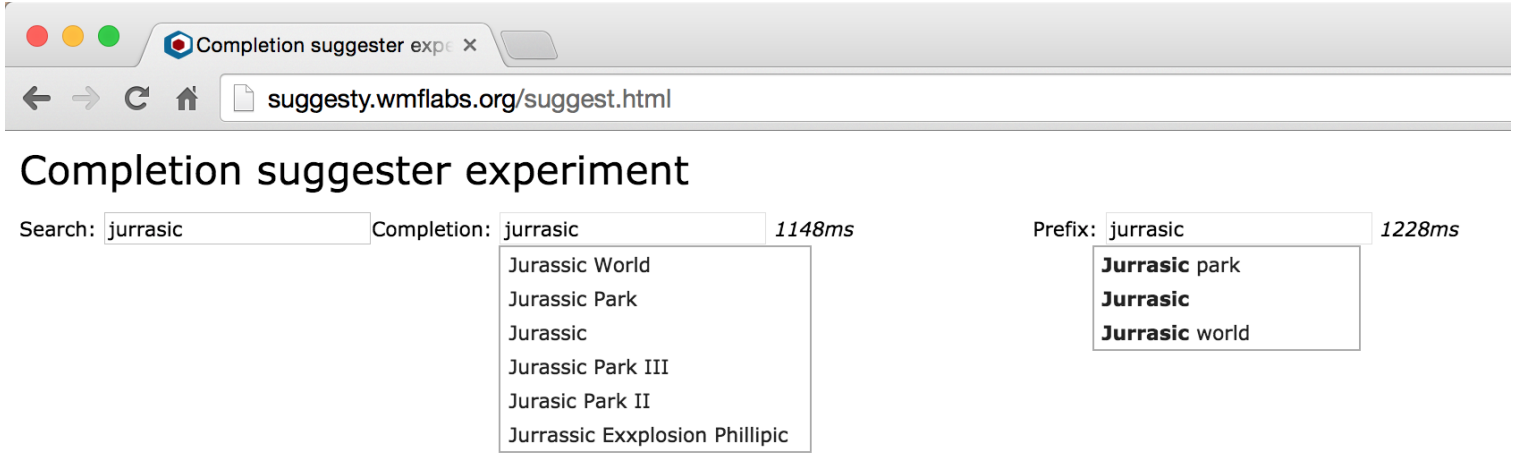
http://searchdata.wmflabs.org/metrics/#kpi_zero_results

What’s next?

We need to try something more radical to achieve our goals.

Why don’t we generate our search results a completely different way?

Elasticsearch Completion Suggester



https://suggesty.wmflabs.org/suggest.html

http://bit.ly/elasticsuggester

Elasticsearch Completion Suggester

Is the completion suggester a
viable alternative to prefixsearch?

Initial tests are promising, showing the completion
suggester cutting zero results rate by nearly 40%.

But what's next?

https://phabricator.wikimedia.org/T109729

Elasticsearch Completion Suggester

We've deployed the completion suggester API to production.

This doesn't change search, it just lets us run tests on the suggester.

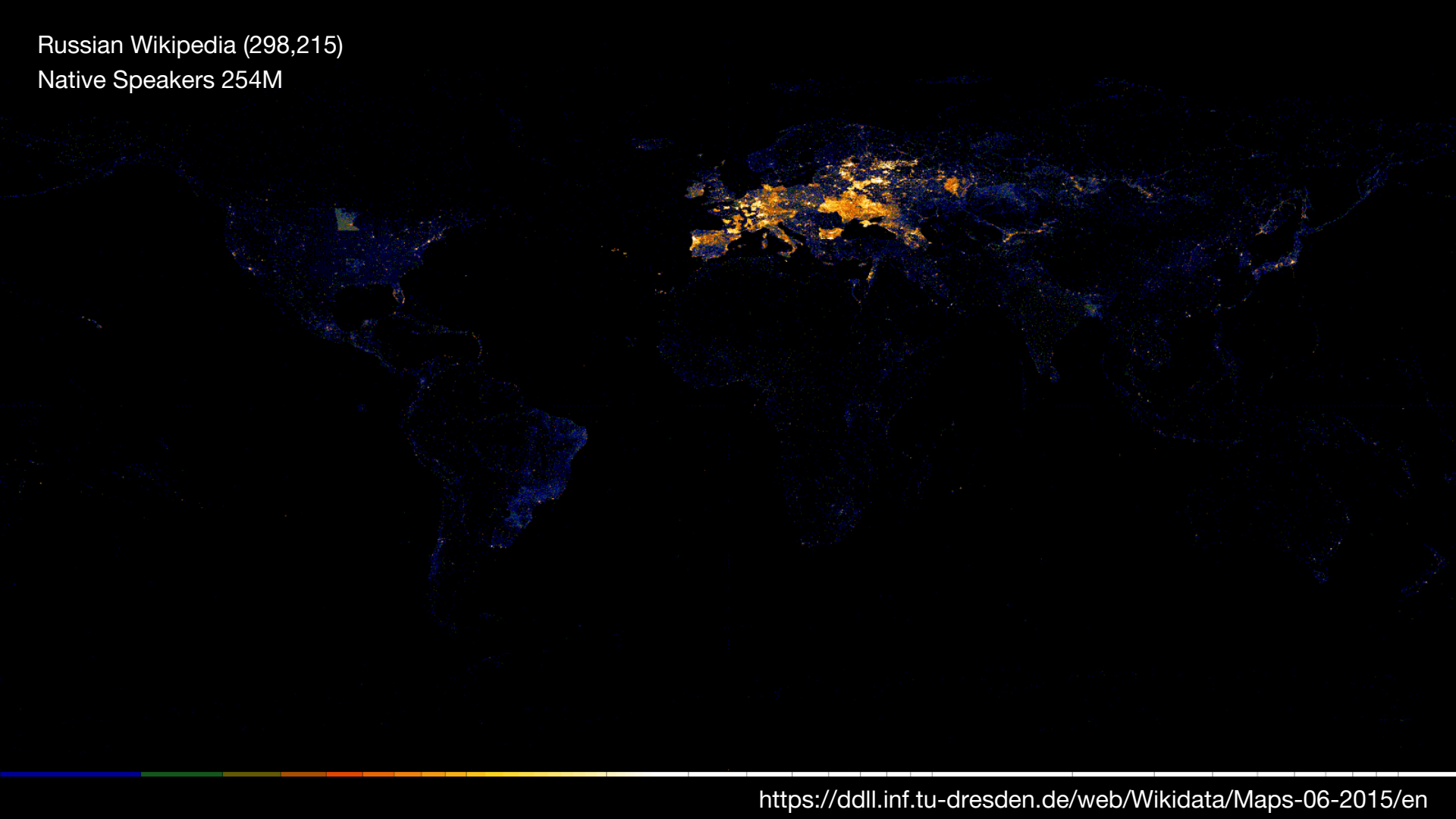
Now we need to A/B test the suggester to see if it's better.

Research

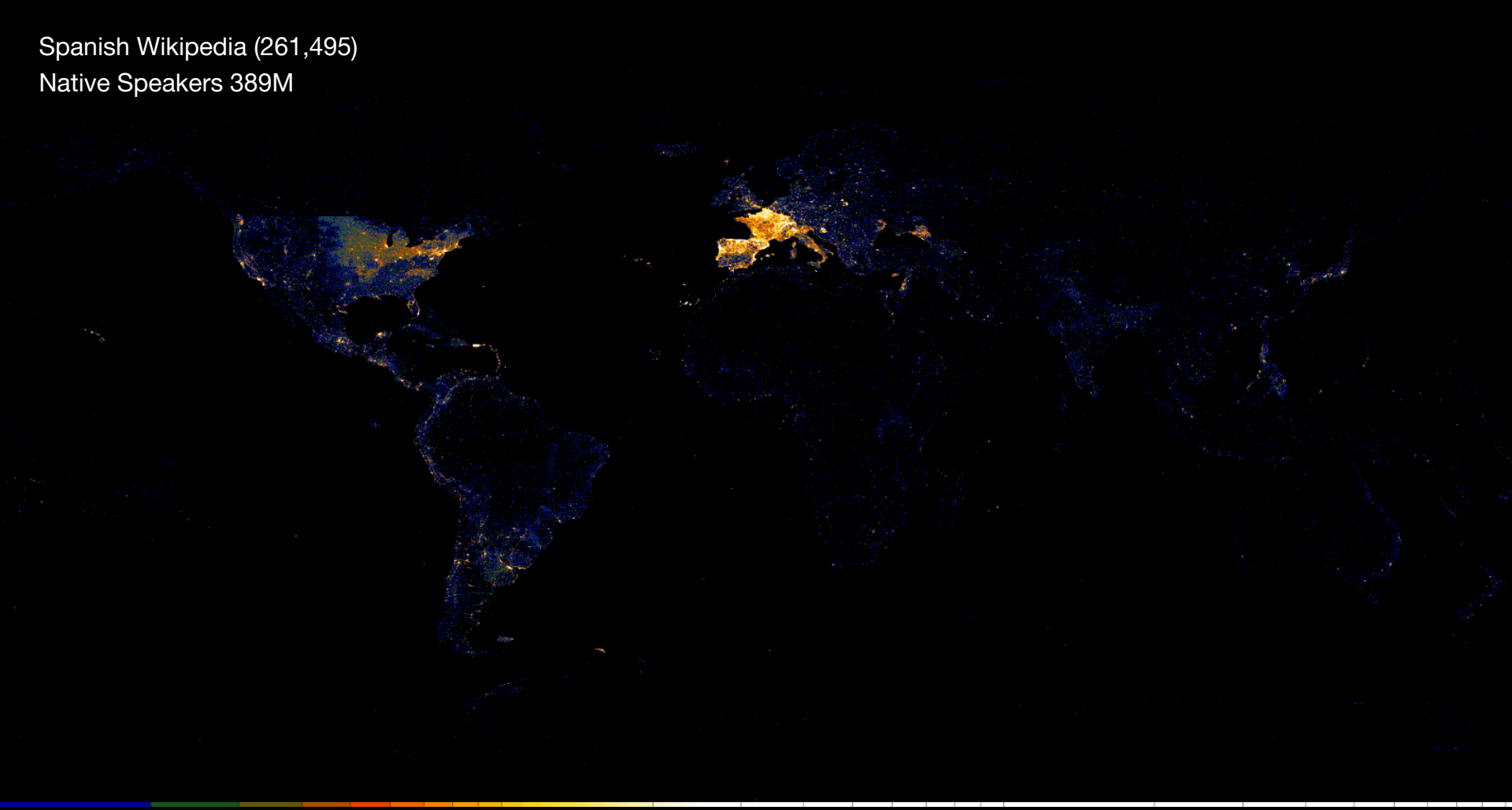
WIKIMEDIA
FOUNDATION

Increasing Article Coverage
Article Recommendation Experiment

https://ddll.inf.tu-dresden.de/web/Wikidata/Maps-06-2015/en

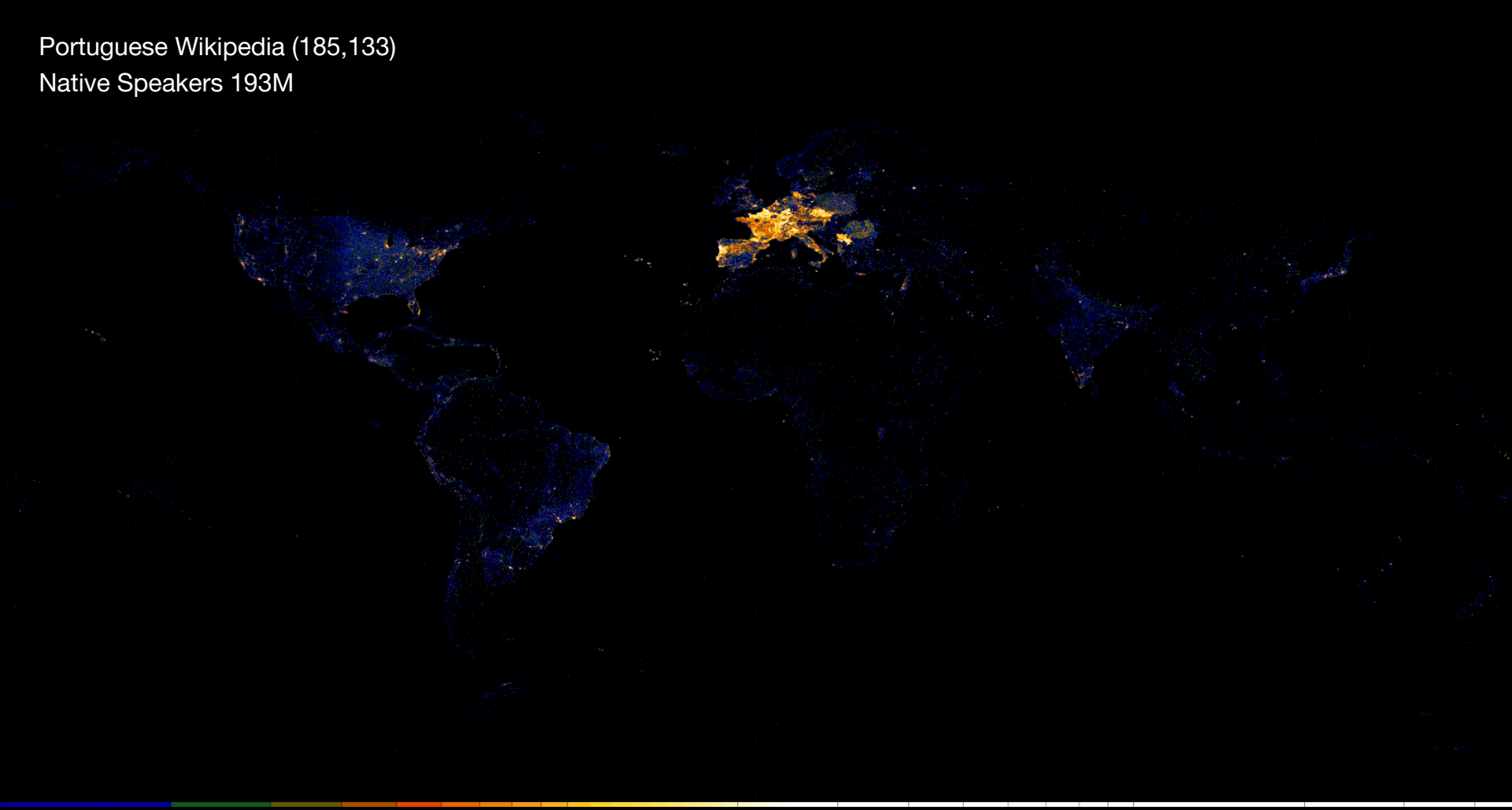

Russian Wikipedia (298,215)
Native Speakers 254M
https://ddll.inf.tu-dresden.de/web/Wikidata/Maps-06-2015/en

Spanish Wikipedia (261,495)
Native Speakers 389M



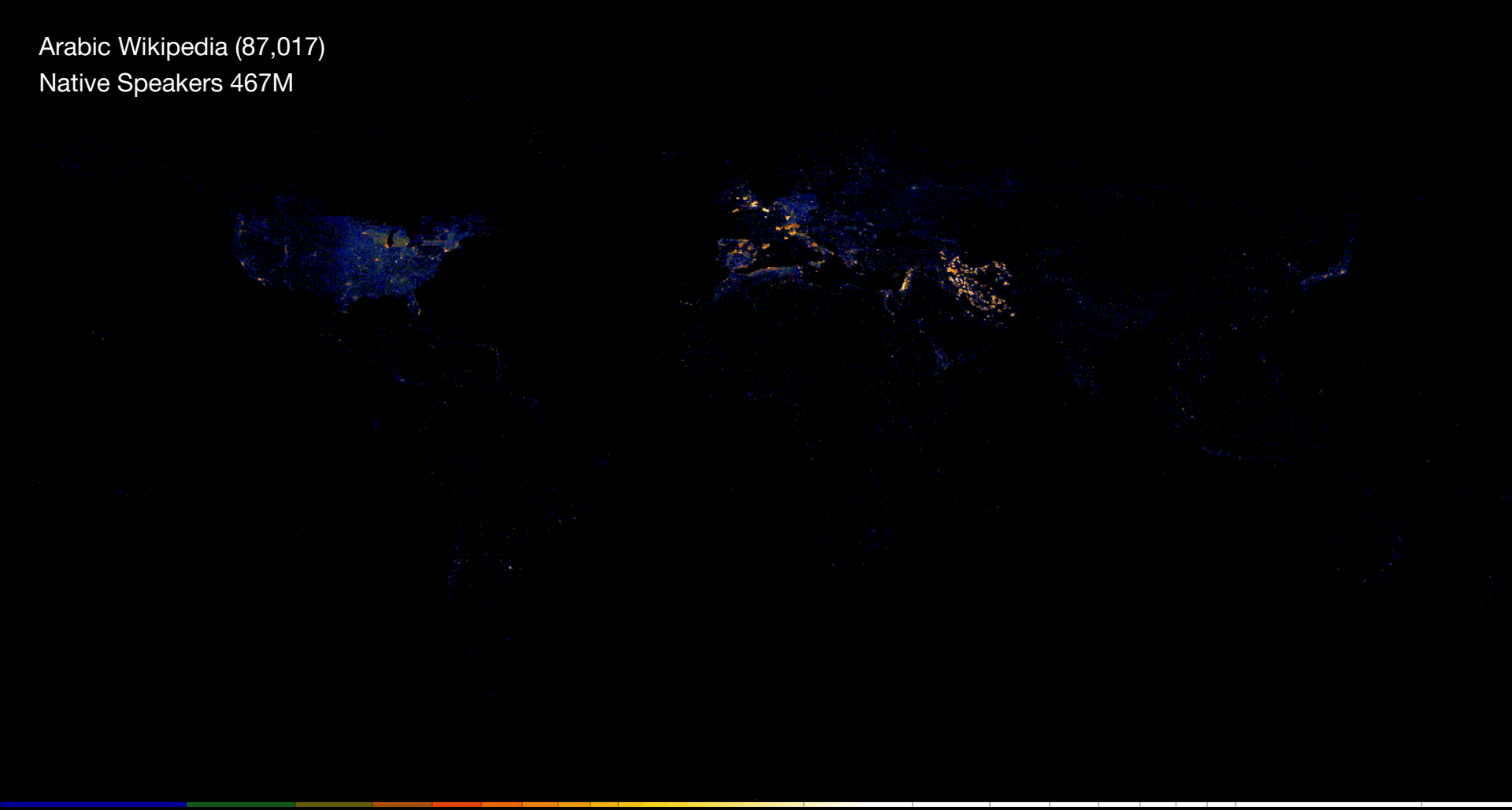
https://ddll.inf.tu-dresden.de/web/Wikidata/Maps-06-2015/en

Portuguese Wikipedia (185,133)
Native Speakers 193M



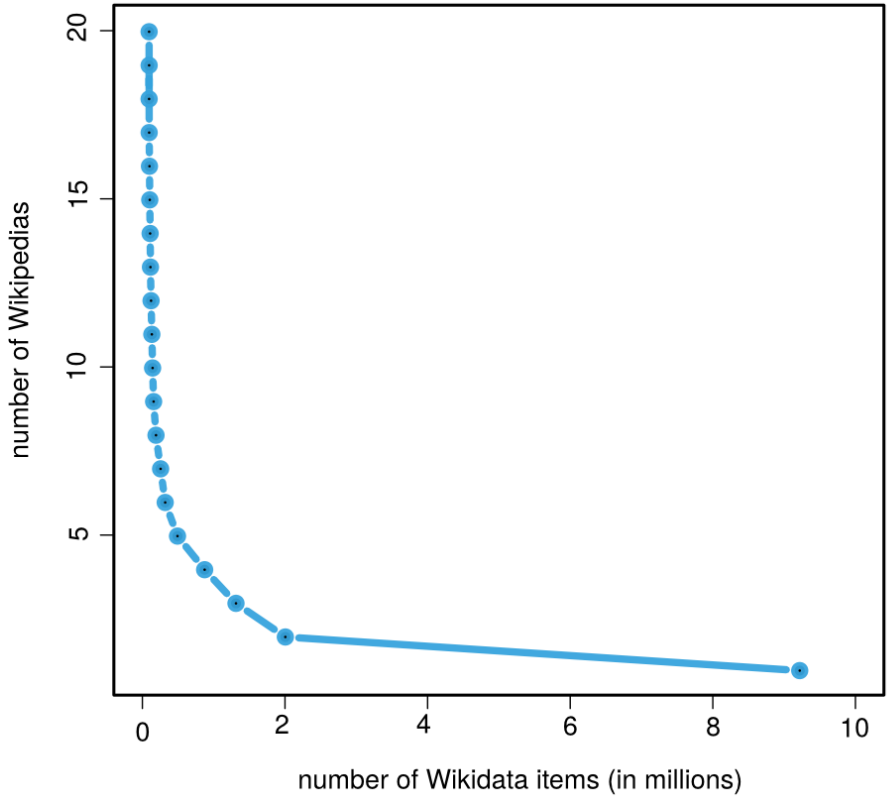
https://ddll.inf.tu-dresden.de/web/Wikidata/Maps-06-2015/en

Arabic Wikipedia (87,017)
Native Speakers 467M



https://ddll.inf.tu-dresden.de/web/Wikidata/Maps-06-2015/en

Article coverage and languages (Cont'd)



Supply and demand

- Demand
  - 2471 languages
  - More than 50% of the world’s population is monolingual
  - The next billion users will come online in 5 years

- Supply
  - Articles are created at a rate of 6500 per day
  - 70K active editors contribute to WP every month and this number has not changed significantly
  - 14K new accounts are created every month

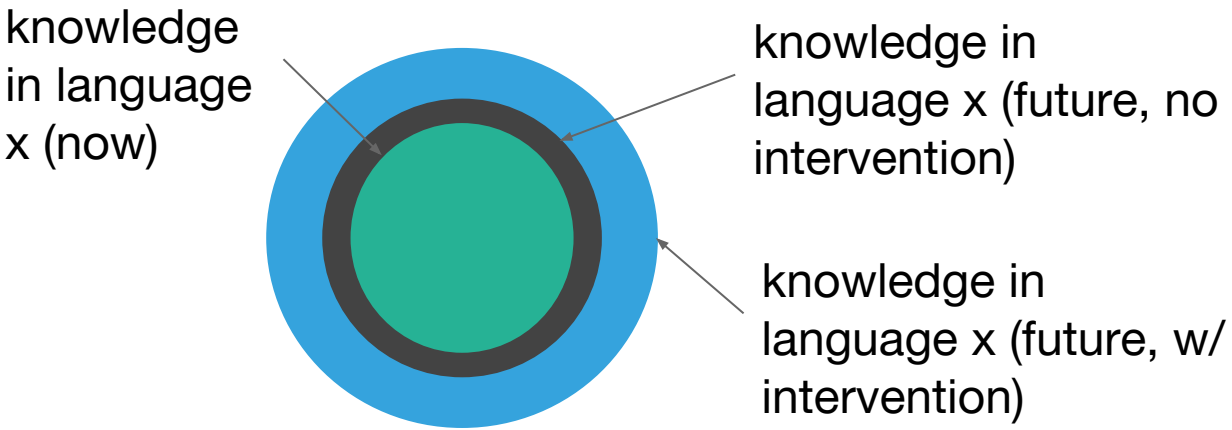
Supply and demand (Cont'd)

- For having at least 40K articles in every language edition, we need at least 6.7M articles, or 3 years.

- For doubling the size of Wikipedia, we will need at least 12 years.

Goal

Increase article coverage in terms of the contents of the articles within a language and in terms of the number of articles in different languages by identifying and prioritizing missing content and routing attention where it's needed.



Methodology

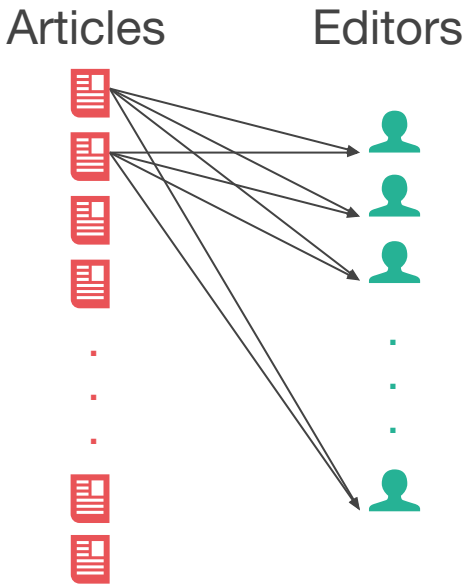
identify *important missing* content in language x

identify potential participants

recommend missing content to participants

1. choose a language pair
2. find missing articles using Wikidata
3. rank missing articles in terms of their importance

1. use Bebal templates
2. use users' edit history



Article-Editor matching (example)

User *E*'s last 15 edits in English Wikipedia are about

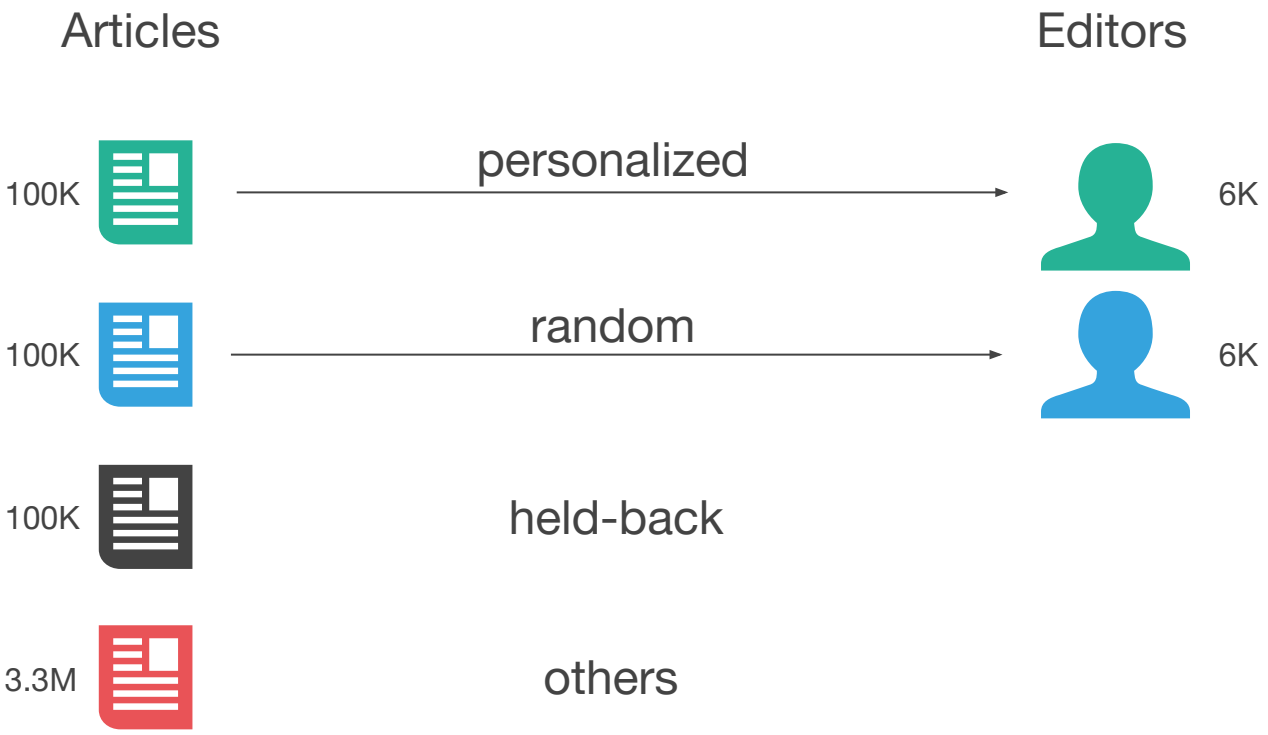
earthquakes, wildfires, robots, Piazza della Loggia bombing, plants, political figures, and heritage campaigns

Here are the list of articles we recommend user *E* considers working on in French:

Tsunami_warning_system, Earthquake_warning_system, Indian_Ocean_Tsunami_Warning_System, Smith_Dharmasaroja, California_Earthquake_Prediction_Evaluation_Council, National Tsunami_Warning_Center, California_landslides, CIA_transnational_health_and_economic_activities

(English, French) Experiment

Design of the experiment



Descriptive statistics

Metric	**Personalized**	**Random**
Number (percentage) of users participated	238 (3.5%)	132 (2%)
Number of articles started	290	158
Number of articles published	123	52
Number of published articles that were deleted	8	6
Ratio of female to male participants	8:106	1:56

Does personalization matter? Yes!

Hypothesis

Users are more likely to translate important articles that were recommended to them based on their interest model as opposed to important but random articles.

personalized

random

On average, personalized recommendations boost the probability of activation by 82%.

Can we increase article creation rate? Yes!

Hypothesis

Article recommendation increases the rate at which articles are created.

On average, random recommendations boost the article creation rate by 78%.

Can we increase article creation rate? Yes!

Hypothesis

Article recommendation increases the rate at which articles are created.

On average, personalized recommendations boost the article creation rate by 220%.

Other important findings

- *Articles that are predicted to be more widely read are more likely to be created.*
- *Articles that are created as a result of recommendations are more viewed.*
- *Editors who were more active prior to the experiment were more likely to respond.*
- *Editors who had made at least one medium size edit (150-900 bytes) in both languages were most likely to respond.*

Summary

- We need to increase article creation rate in the areas that content is needed.

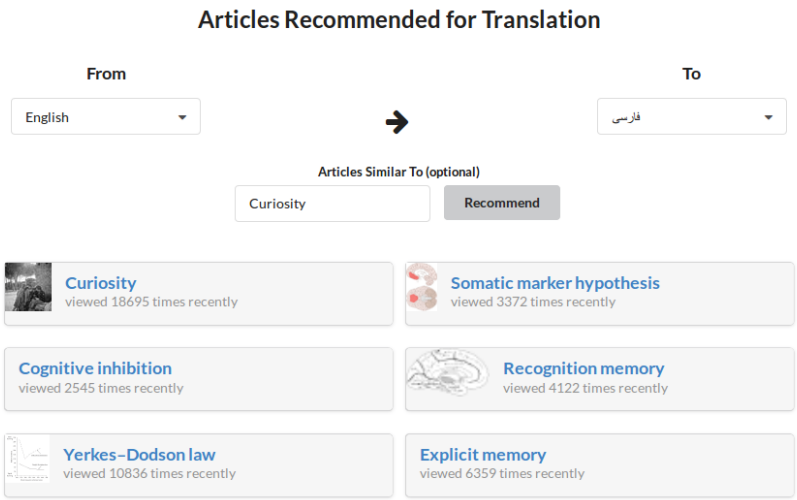
- We proposed article recommendation as one approach to increase content creation rate.

- We showed that personalized recommendations increased editor activation rate for translation on average by 82%.

- We showed that with the current editor population, article recommendation can be used to increase the content creation rate on average by 220%.

Next steps and open questions

- Increasing the potential participant pool by offering more language pairs, offering the tool for editathons, etc.
- Improving the algorithm by building a user feedback loop
- Improving the instance on Labs in terms of the user experience and providing personalized recommendations.
- Offering the recommendation as part of the CX tool
- Rethinking content creation.



Thank you!

Join us in Phabricator: Increasing content coverage project

Documentations

https://meta.wikimedia.org/wiki/Research:Increasing_article_coverage

https://meta.wikimedia.org/wiki/Research:Increasing_article_coverage/Tool

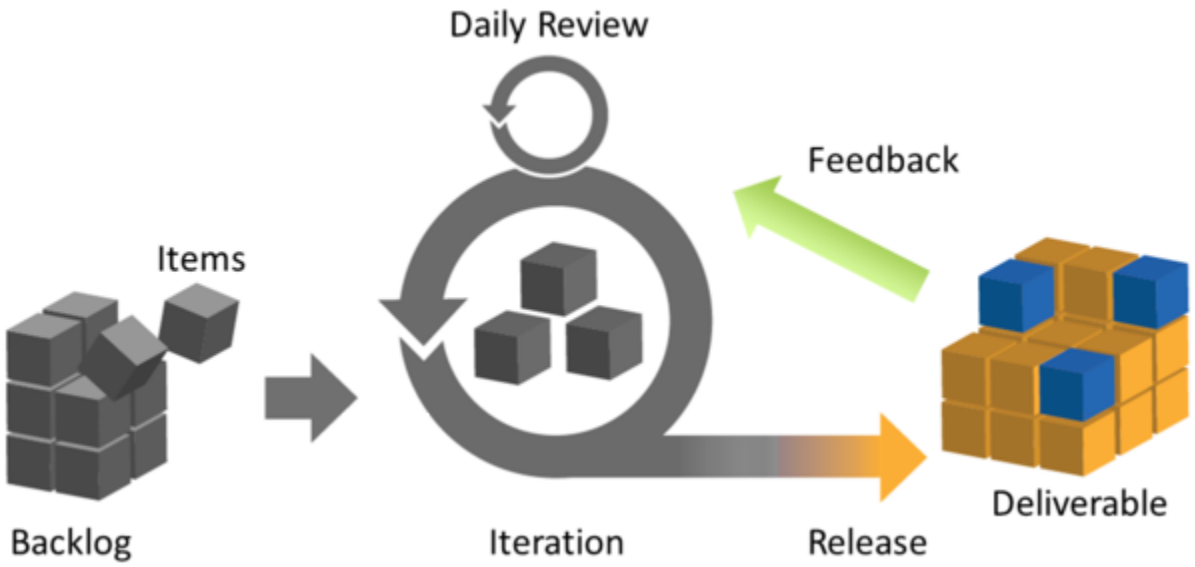
Feature

WIKIMEDIA
FOUNDATION

Voting browser tests

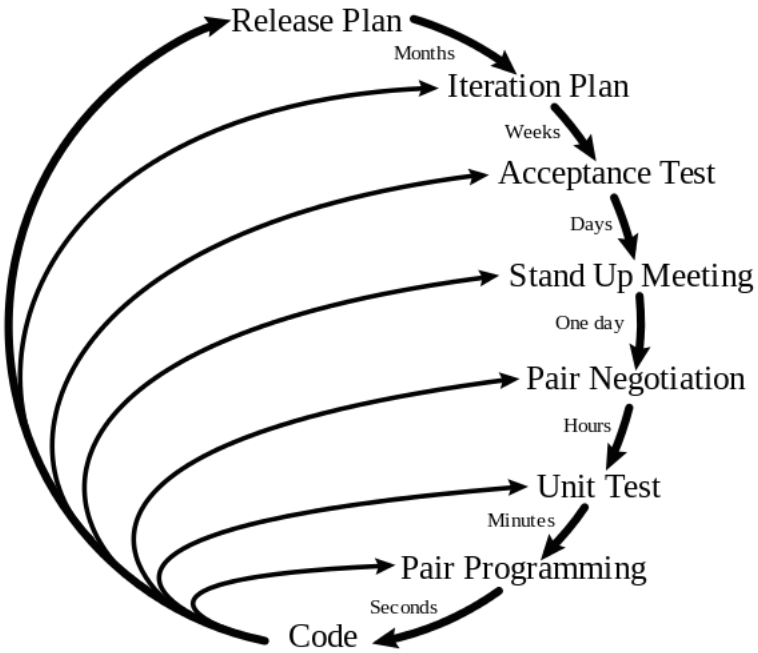
Release Engineering

What we really do

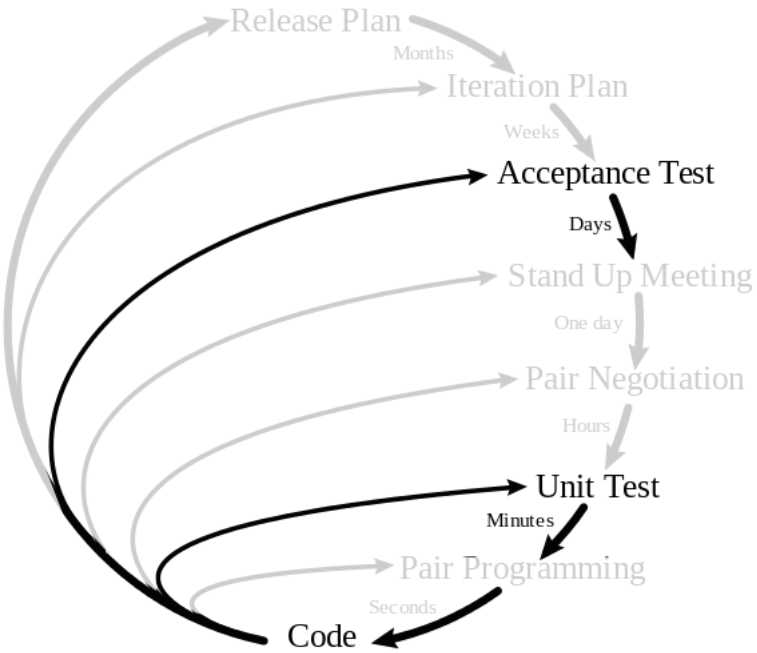


Delivering deliverables delivery since our delivery. —Evil Greg

Feedback


Release Plan
Months
Iteration Plan
Weeks
Acceptance Test
Days
Stand Up Meeting
One day
Pair Negotiation
Hours
Unit Test
Minutes
Pair Programming
Seconds
Code

(Automated) feedback

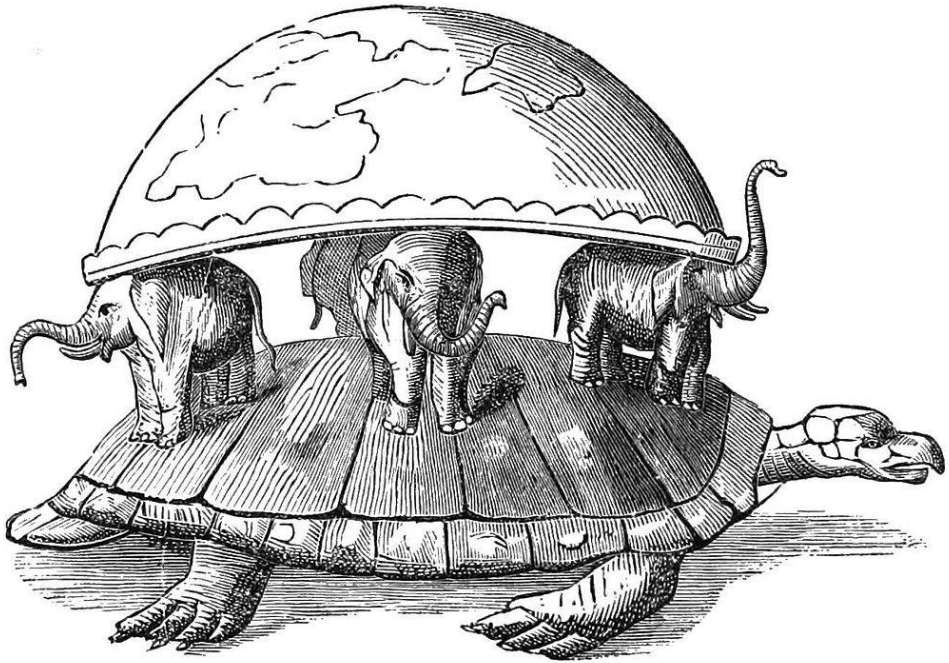


End-to-end (browser) tests

```
Feature: Login form
    As a user …
Scenario: Providing good credentials logs me in
    Given I am on the login page
    When I provide good credentials
    And click the “Log in” button
    Then I have been logged in
```

End-to-end tests expose bugs[citation needed]

Everywhere in the stack.



Daily runs are problematic

Too much code can change over the course of a day.

Build #220 (Aug 7, 2015 2:28:00 PM)

https://phabricator.wikimedia.org/T108356

edit description

Build Artifacts

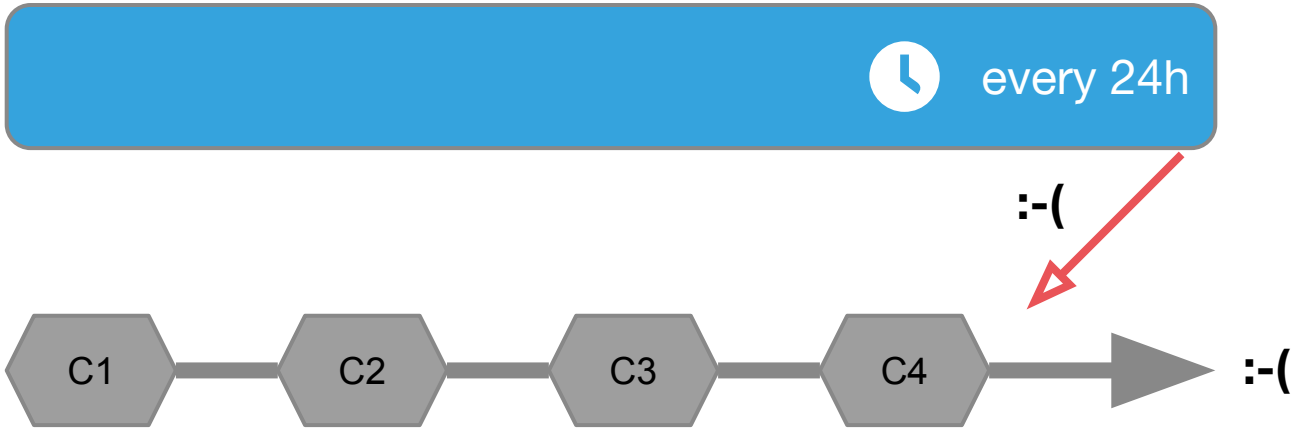
Expand all Collapse all

View

Changes

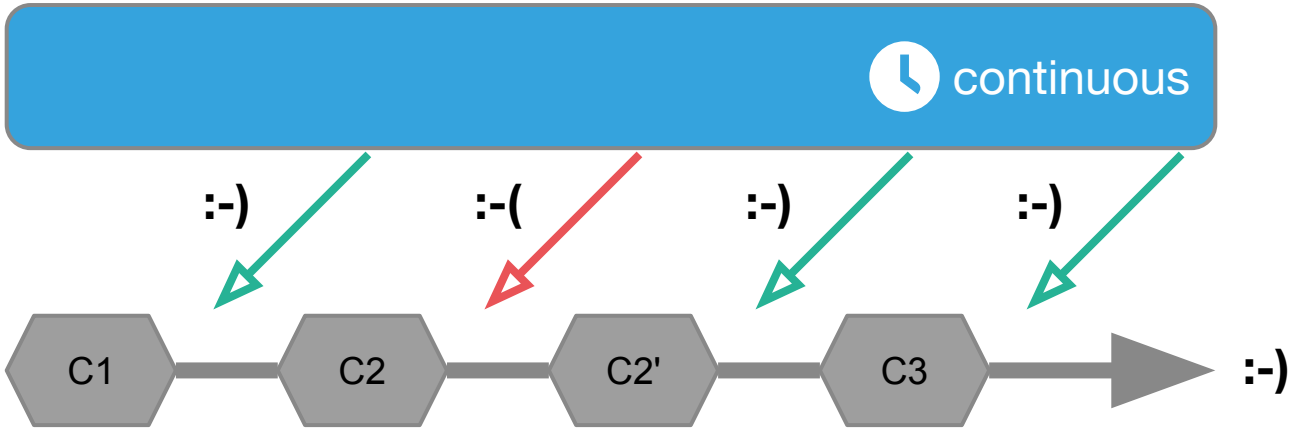
1. Remove CodeMirror support (detail)
2. Delete save process code in favour of VE's own save dialog (detail)
3. Declare correct dependencies for pagelist (detail)
4. Skip tests that have side effects (detail)
5. Revert "Don't register unloadable test modules" (detail)
6. Remove unused toolbar config code (detail)
7. Move title into VE toolbar (detail)
8. Hygiene: Restore some skipped tests (detail)
9. Localisation updates from https://translatewiki.net. (detail)
10. Use default order of footer elements (detail)
11. Remove title styling from heading. (detail)

Feedback is sparse

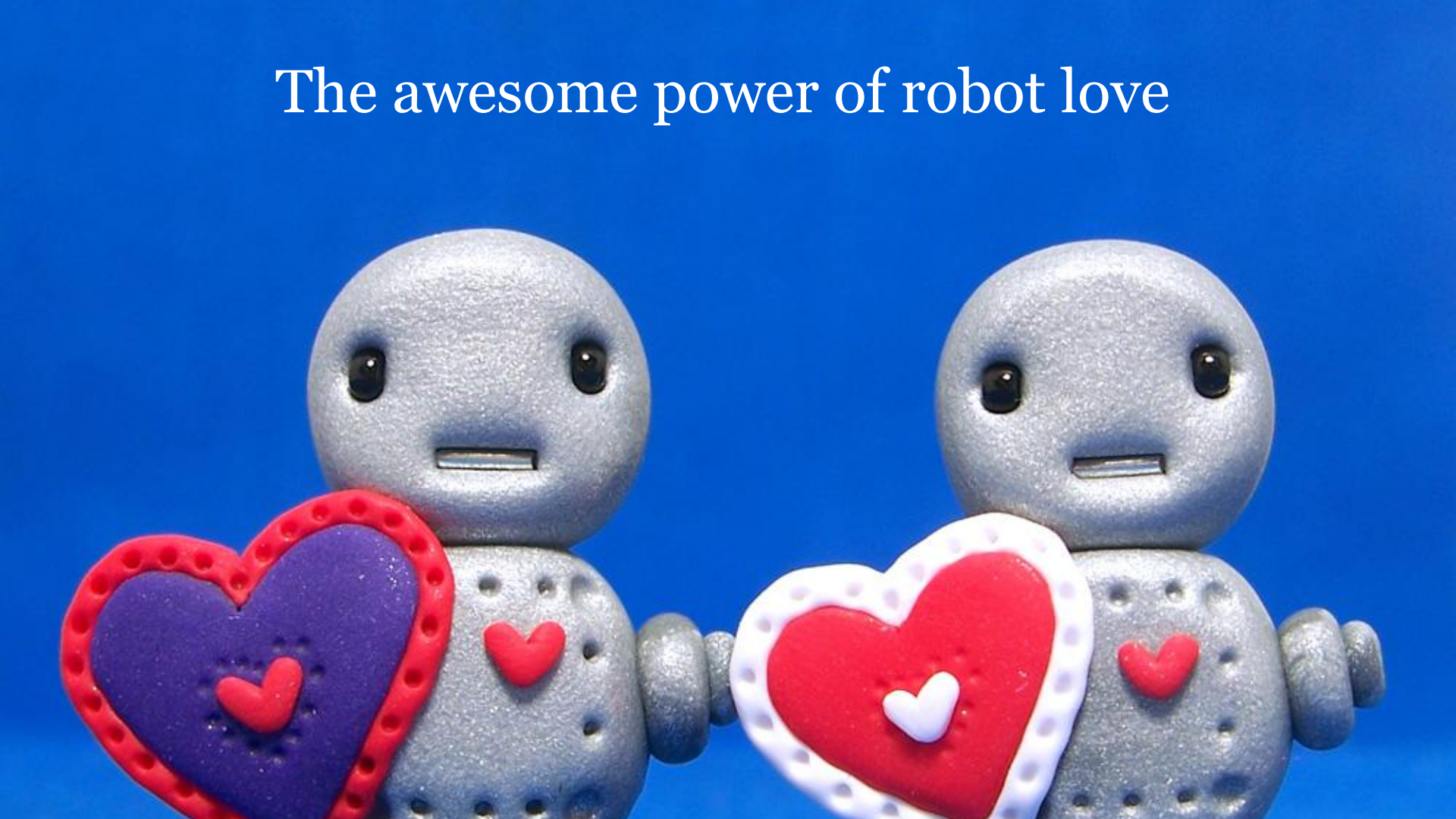


How to reliably tighten the feedback loop

And get feedback for every code commit



The awesome power of robot love



Great proof of concept

Reading’s experiments with Barry taught us some things.

- Value of a well groomed end-to-end test suite
- Viability of running end-to-end suites upon every commit

But ...

- Difficult to setup
- Out of band
- Easy to ignore


We want robots for everyone!

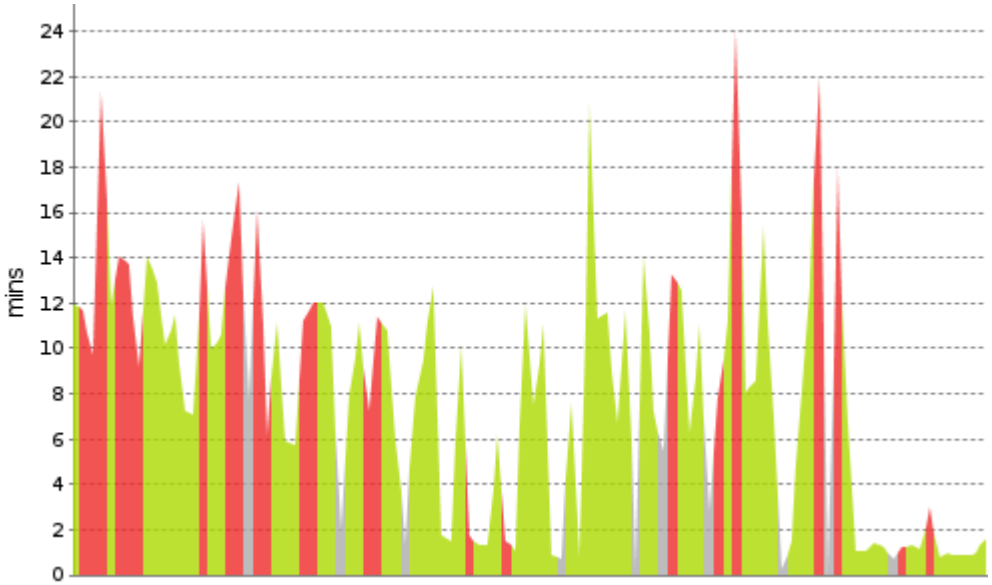
The mwext-mw-selenium job

Runs each time a new change is pushed to Gerrit:

- Checks out master branches of MediaWiki core and dependencies
- Installs a local wiki
- Executes end-to-end test suite using MediaWiki-Selenium
- Runs real browsers (Chromium/Firefox) headlessly
- Records sessions and saves video of failures

https://www.mediawiki.org/wiki/Continuous_integration/Browser_tests

It’s slow but not terribly so



~ 30-60s per scenario

And it gives precious feedback

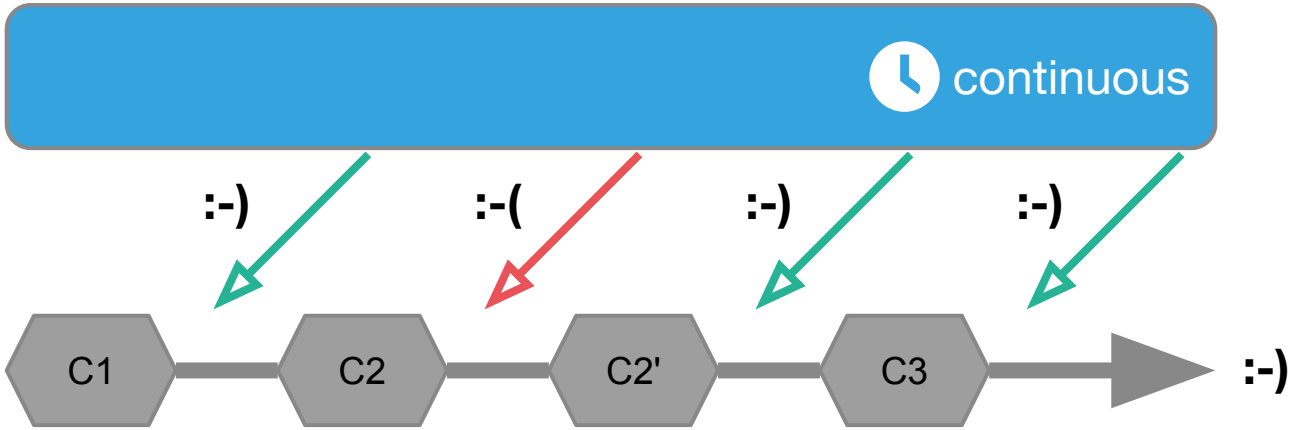
```
Failing Scenarios:
cucumber features/anonymous.feature:28 # Scenario: Anons can see my public collection

21 scenarios (1 failed, 20 passed)
154 steps (1 failed, 153 passed)
21m7.366s
```

Build Artifacts

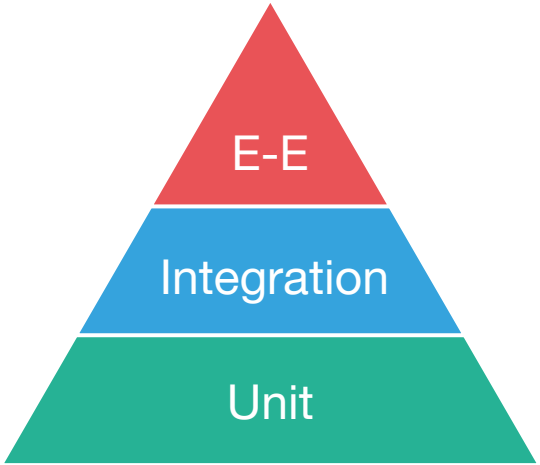
Anonymous users: Anons can see my public collection.mp4

For every code commit



A word to the wise

End-to-end tests give **broad** coverage, but they're **fragile**. Write more unit tests!



Google test engineering recommends starting with a 70/20/10 split.[1]

Thanks!

Q&A