Research and Academic Software Projects at the Institute for Quantitative Social Science

Mercè Crosas, Ph.D.
Chief Data Science and Technology Officer
IQSS, Harvard University
twitter: @mercecrosas web: mercecrosas.com

The Big Picture

Identify a problem or need in research or academia

The Big Picture

Identify a problem or need in research or academia

Build a technology solution, easy- to-use, gives control to researcher

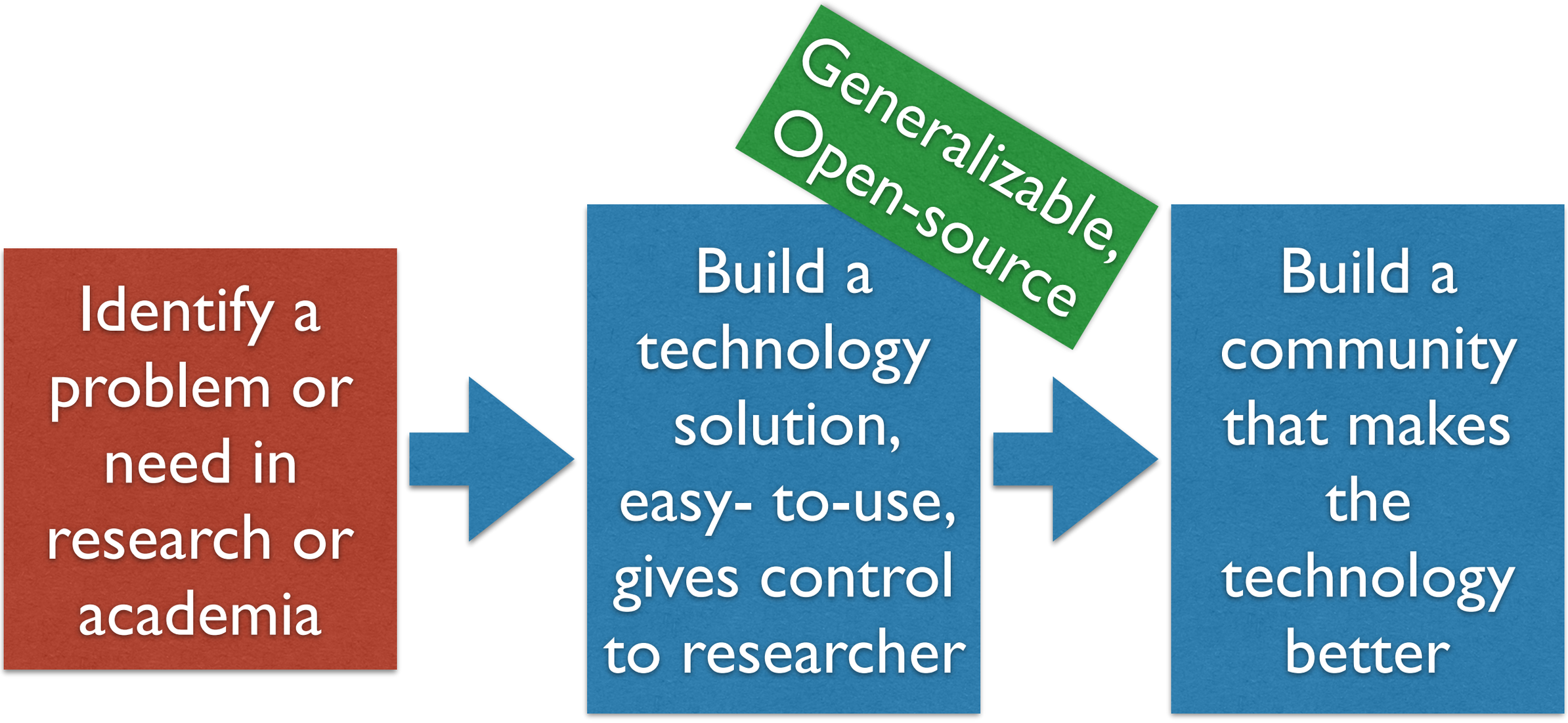
The Big Picture

Identify a problem or need in research or academia

Build a technology solution, easy- to-use, gives control to researcher

Generalizable, Open-source

The Big Picture



Example: Dataverse

Example: Dataverse

- How do we increase data sharing to improve research transparency and replication with incentives to researchers?

Example: Dataverse

- How do we increase data sharing to improve research transparency and replication with incentives to researchers?

- Provide a repository solution, where researchers have control of branding and access of their data, and get credit through data citation.

Example: OpenScholar

Example: OpenScholar

- How do we enable scholars to build their academic web sites in a cost effective way?

Example: OpenScholar

- How do we enable scholars to build their academic web sites in a cost effective way?
- Provide a web site builder with pre-set features for academics, where a single hosting serves thousands of sites.

Example: Zelig

Example: Zelig

- How do we simplify using thousands of R statistical methods built by different authors?

Example: Zelig

- How do we simplify using thousands of R statistical methods built by different authors?

- Provide a statistical package that uses the same three commands for all methods, with consistent documentation.

Example: Consilience

Example: Consilience

- How do we make sense of thousands (or millions!) of texts?

Example: Consilience

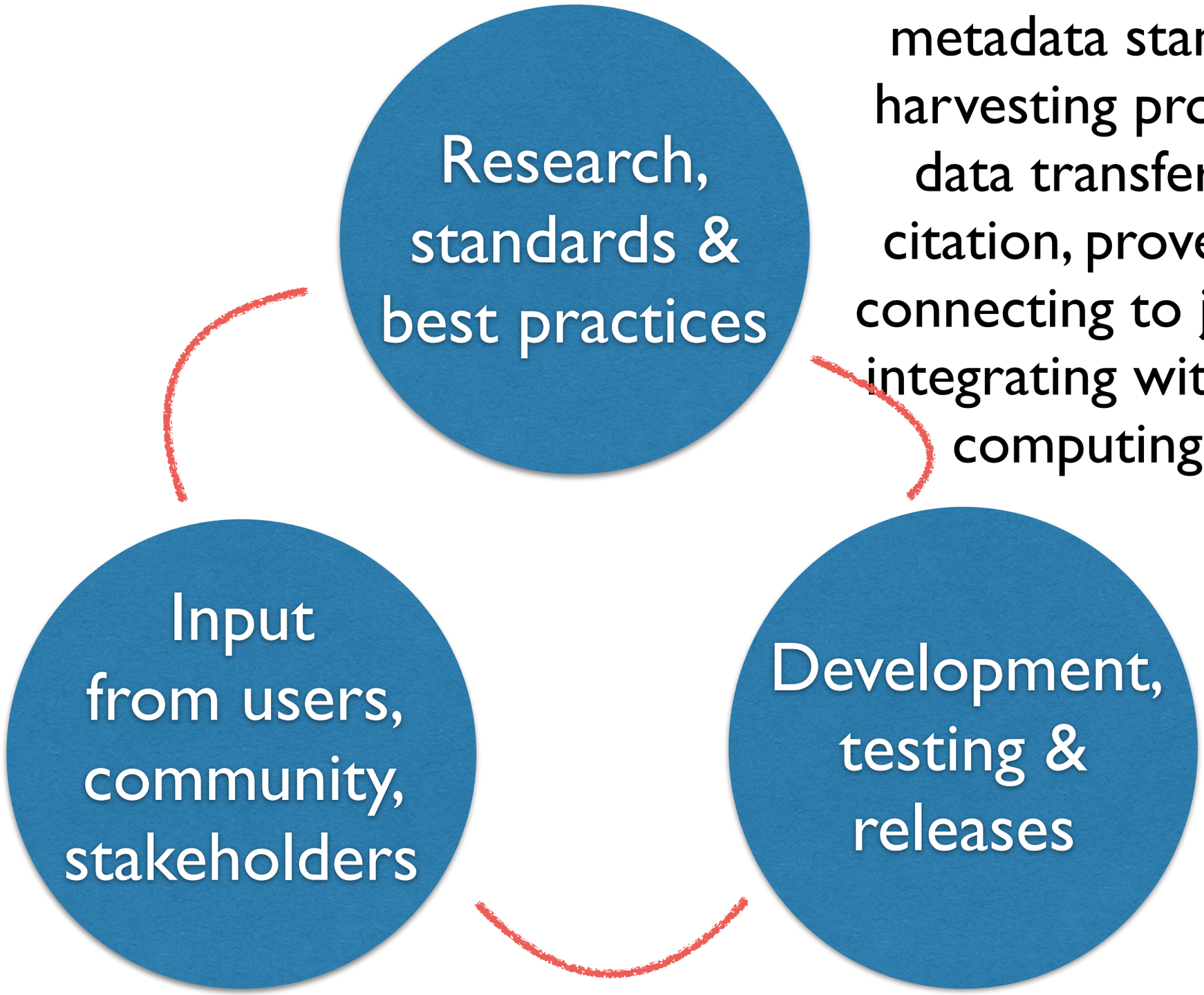
- How do we make sense of thousands (or millions!) of texts?
- Provide an application that helps researchers explore many possible ways of categorizing documents.

Dataverse Case Study

The Process

Dataverse Case Study

The Process



metadata standards, harvesting protocols, data transfer, data citation, provenance, connecting to journals, integrating with cloud computing, ….

Dataverse Case Study

The Process

Research, standards & best practices

metadata standards, harvesting protocols, data transfer, data citation, provenance, connecting to journals, integrating with cloud computing, ….

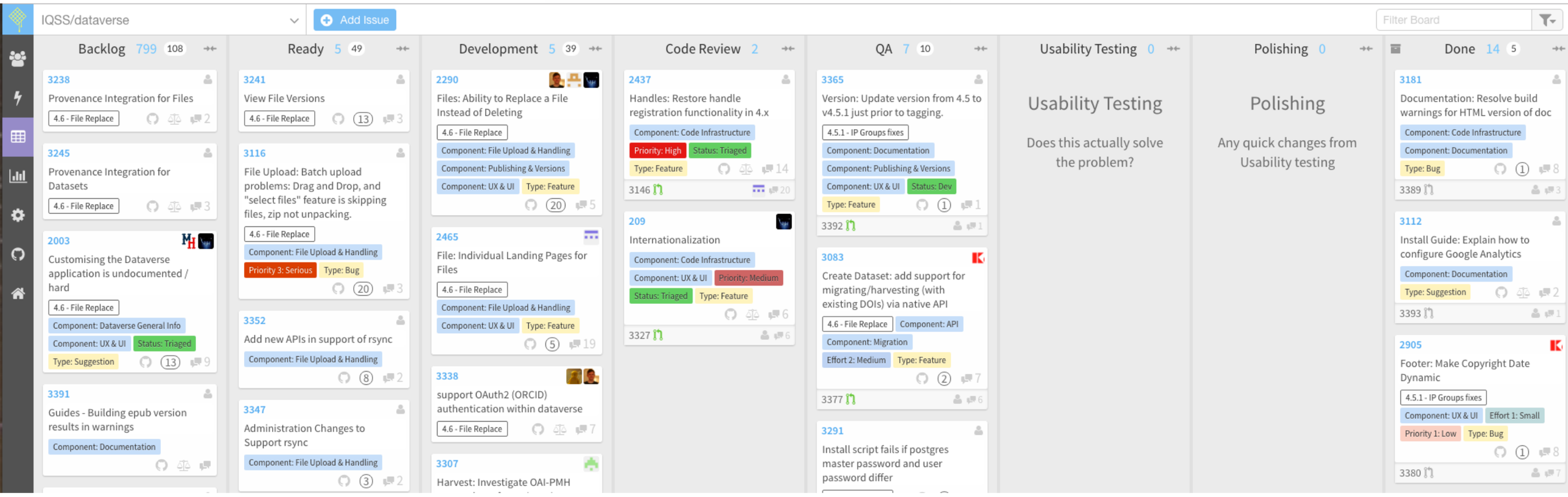
Development, testing & releases

Input from users, community, stakeholders

usability testing, community calls, annual community meeting, pull requests

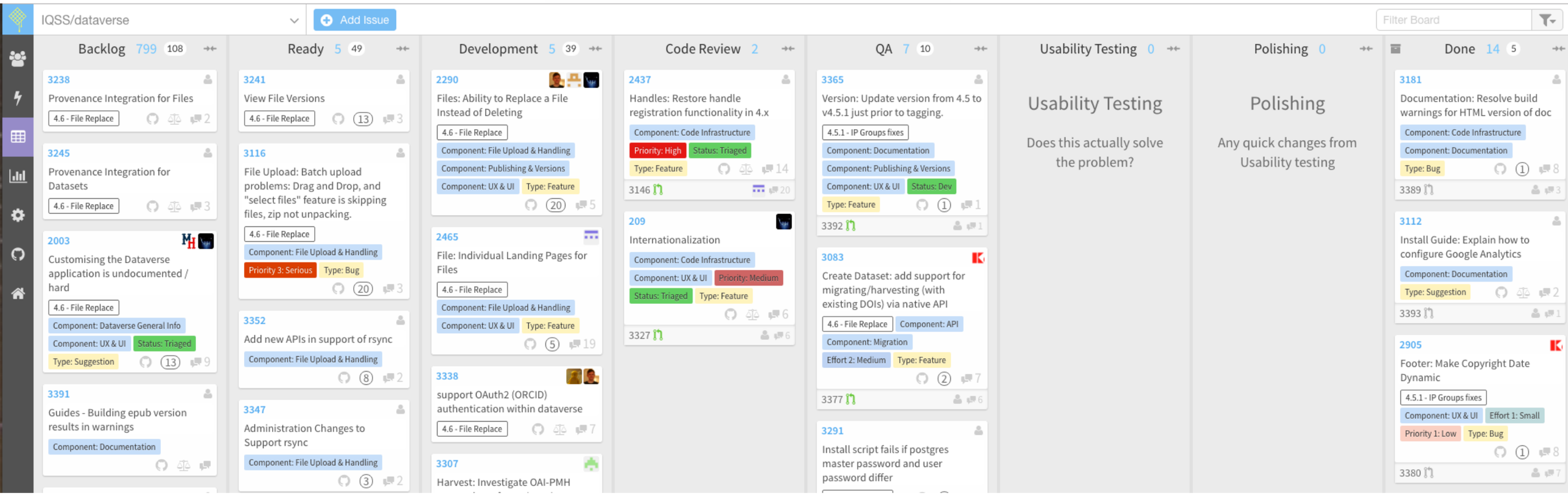
Dataverse Case Study

The Process Details



Dataverse Case Study

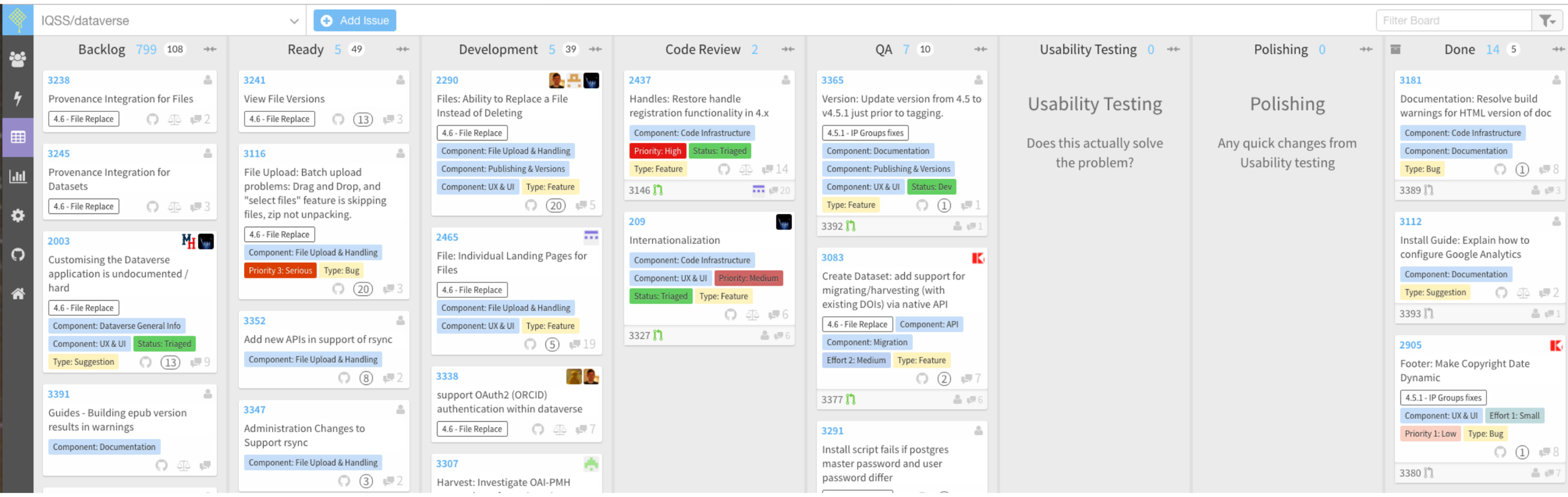
The Process Details



An agile process, integrating Waffle + GitHub + Jenkins, including these steps:

Dataverse Case Study

The Process Details



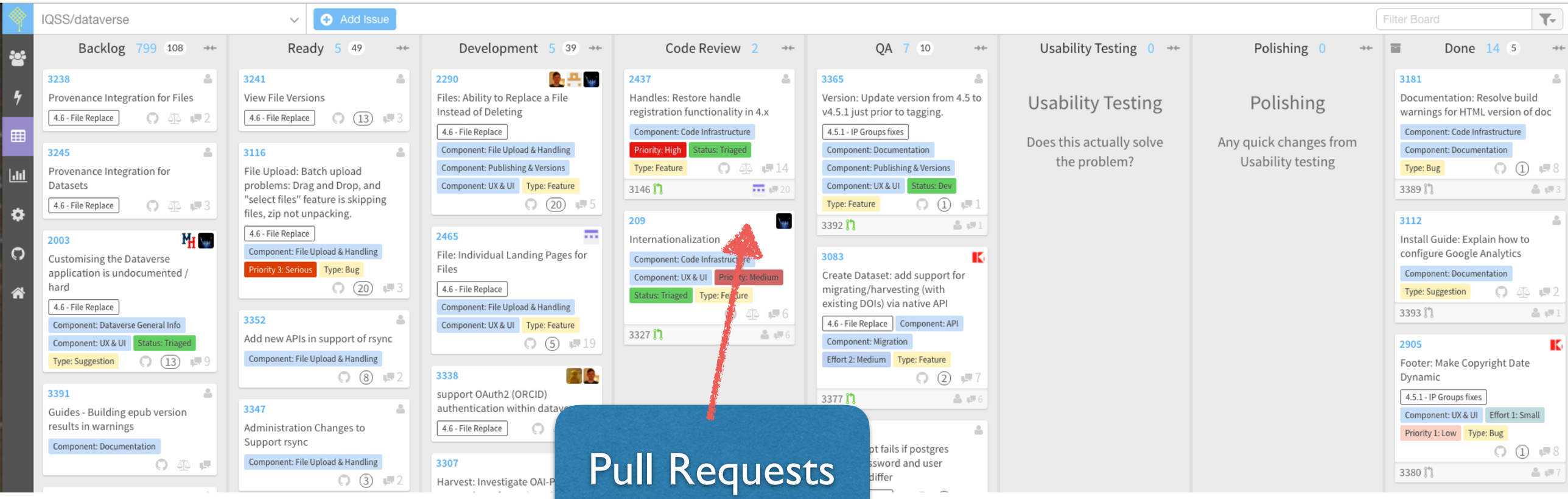
An agile process, integrating Waffle + GitHub + Jenkins, including these steps:

Backlog > Ready > Dev > Code Review > QA > Usability Test > Polishing > Done

Dataverse Case Study

The Process Details

Jenkins



An agile process, integrating Waffle + GitHub + Jenkins, including these steps:

Backlog > Ready > Dev > Code Review > QA > Usability Test > Polishing > Done

Not only Best Practices in Process, but also in Coding

Not only Best Practices in Process, but also in Coding

Best Practices for Scientific Computing

Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, Paul Wilson

Published: January 7, 2014 • http://dx.doi.org/10.1371/journal.pbio.1001745

Not only Best Practices in Process, but also in Coding

Best Practices for Scientific Computing

Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, Paul Wilson

Published: January 7, 2014 • http://dx.doi.org/10.1371/journal.pbio.1001745

1. Write programs for people, not computers.

Not only Best Practices in Process, but also in Coding

Best Practices for Scientific Computing

Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, Paul Wilson

Published: January 7, 2014 • http://dx.doi.org/10.1371/journal.pbio.1001745

1. Write programs for people, not computers.
2. Let the computer do the work.

Not only Best Practices in Process, but also in Coding

Best Practices for Scientific Computing

Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, Paul Wilson

Published: January 7, 2014 • http://dx.doi.org/10.1371/journal.pbio.1001745

1. Write programs for people, not computers.
2. Let the computer do the work.
3. Make incremental changes.

Not only Best Practices in Process, but also in Coding

Best Practices for Scientific Computing

Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, Paul Wilson

Published: January 7, 2014 • http://dx.doi.org/10.1371/journal.pbio.1001745

1. Write programs for people, not computers.
2. Let the computer do the work.
3. Make incremental changes.
4. Don't repeat yourself (or others).

Not only Best Practices in Process, but also in Coding

Best Practices for Scientific Computing

Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, Paul Wilson

Published: January 7, 2014 • http://dx.doi.org/10.1371/journal.pbio.1001745

1. Write programs for people, not computers.
2. Let the computer do the work.
3. Make incremental changes.
4. Don't repeat yourself (or others).
5. Plan for mistakes.

Not only Best Practices in Process, but also in Coding

Best Practices for Scientific Computing

Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, Paul Wilson

Published: January 7, 2014 • http://dx.doi.org/10.1371/journal.pbio.1001745

1. Write programs for people, not computers.
2. Let the computer do the work.
3. Make incremental changes.
4. Don't repeat yourself (or others).
5. Plan for mistakes.
6. Optimize software only after it works correctly.

Not only Best Practices in Process, but also in Coding

Best Practices for Scientific Computing

Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, Paul Wilson

Published: January 7, 2014 • http://dx.doi.org/10.1371/journal.pbio.1001745

1. Write programs for people, not computers.
2. Let the computer do the work.
3. Make incremental changes.
4. Don't repeat yourself (or others).
5. Plan for mistakes.
6. Optimize software only after it works correctly.
7. Document design and purpose, not mechanics.

Not only Best Practices in Process, but also in Coding

Best Practices for Scientific Computing

Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, Paul Wilson

Published: January 7, 2014 • http://dx.doi.org/10.1371/journal.pbio.1001745

1. Write programs for people, not computers.
2. Let the computer do the work.
3. Make incremental changes.
4. Don't repeat yourself (or others).
5. Plan for mistakes.
6. Optimize software only after it works correctly.
7. Document design and purpose, not mechanics.
8. Collaborate.

Impact at Harvard

OpenScholar®

6,833 OpenScholar sites created

13,904 Registered users

75,378 Publications posted

24 Academic departments

Impact at Harvard

The Dataverse Project

243 Dataverses from Harvard affiliates

1,226 Datasets by Harvard affiliates as authors

1,427 Registered Harvard users

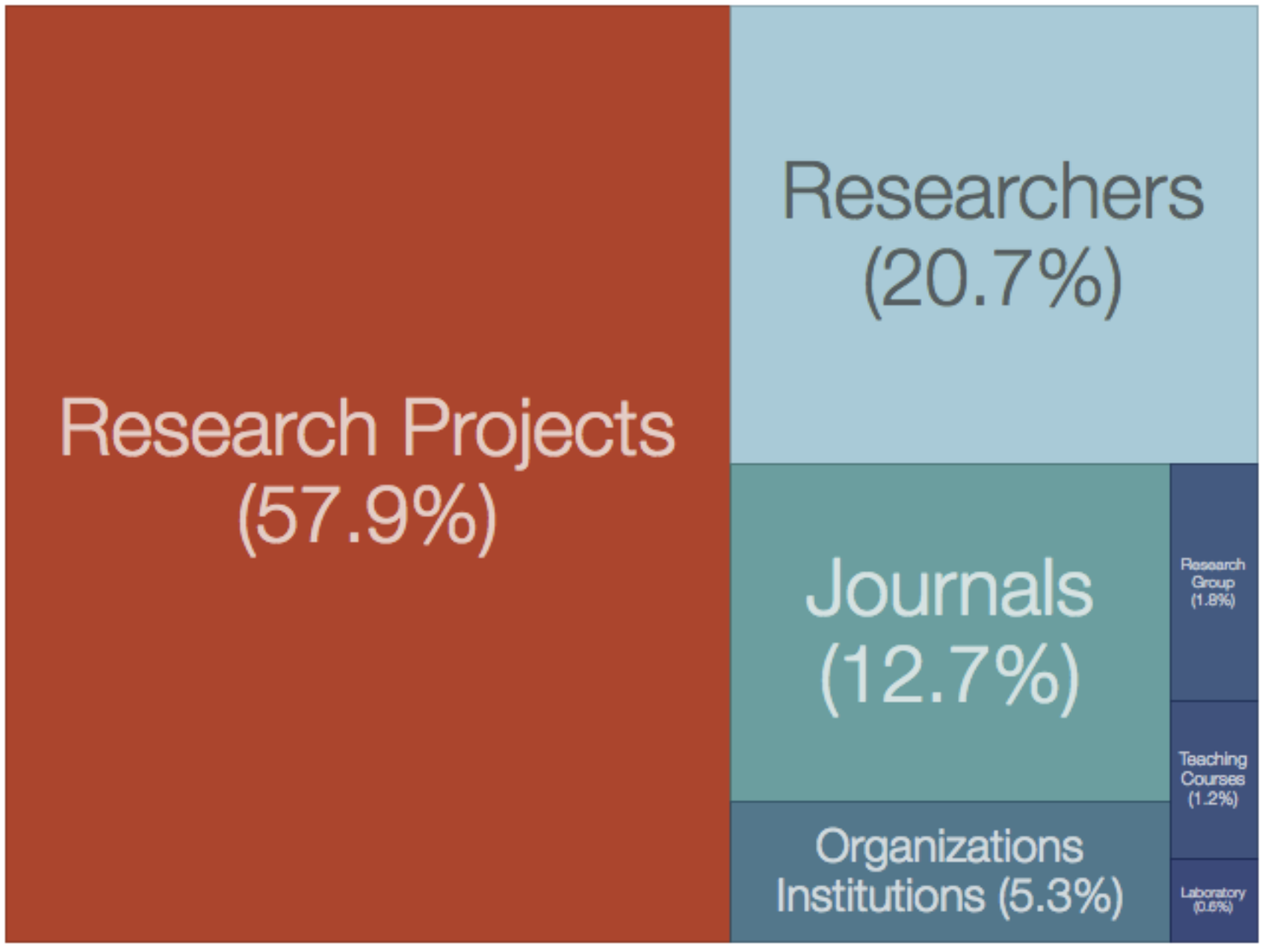
Broader Impact

Dataverse world-wide impact

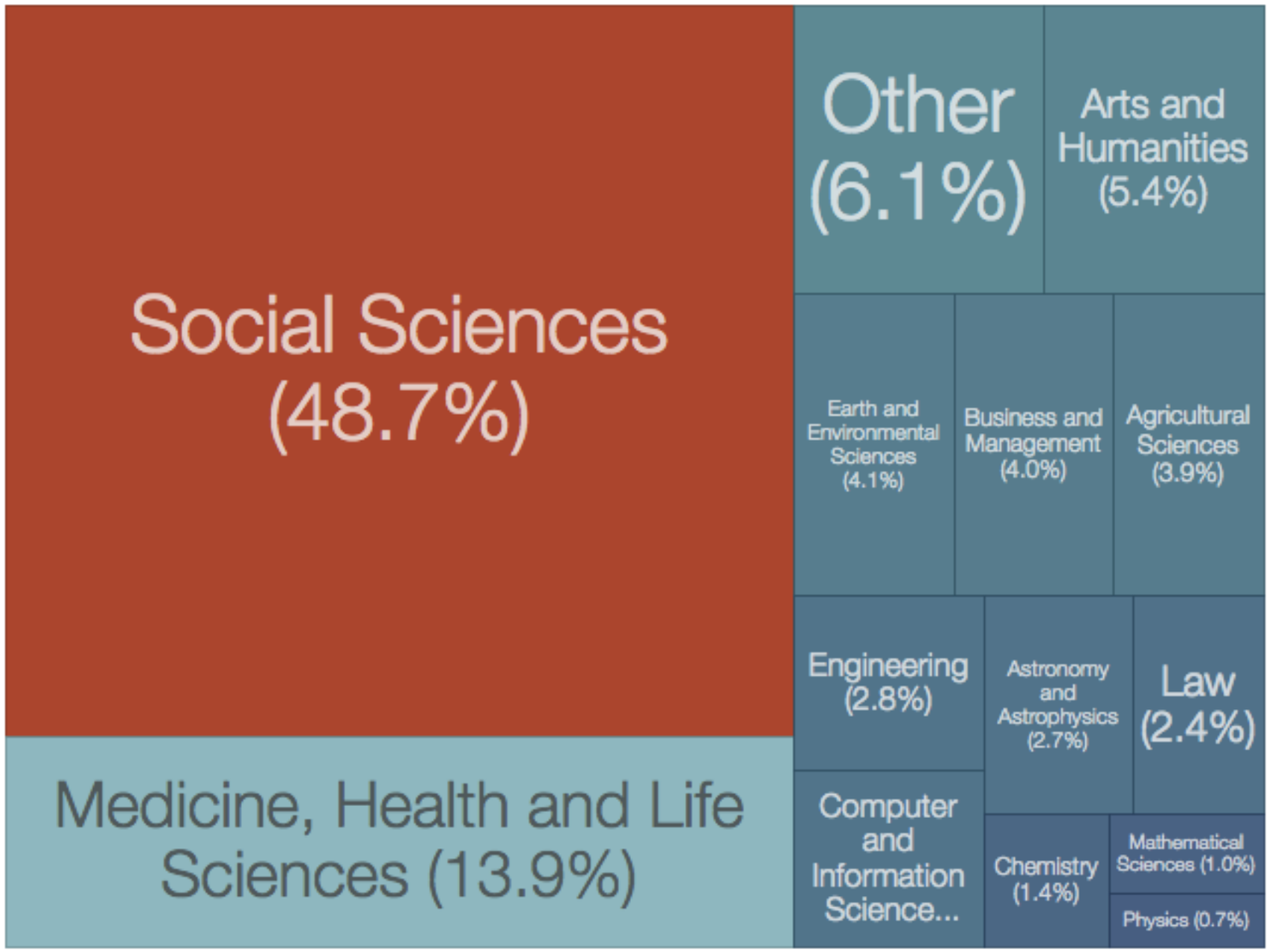
20 Installations | 1,884 Dataverses | 64,149 Datasets | 1,933,476 Downloads

Stats generated: 2nd October 2016 16:13 EDT

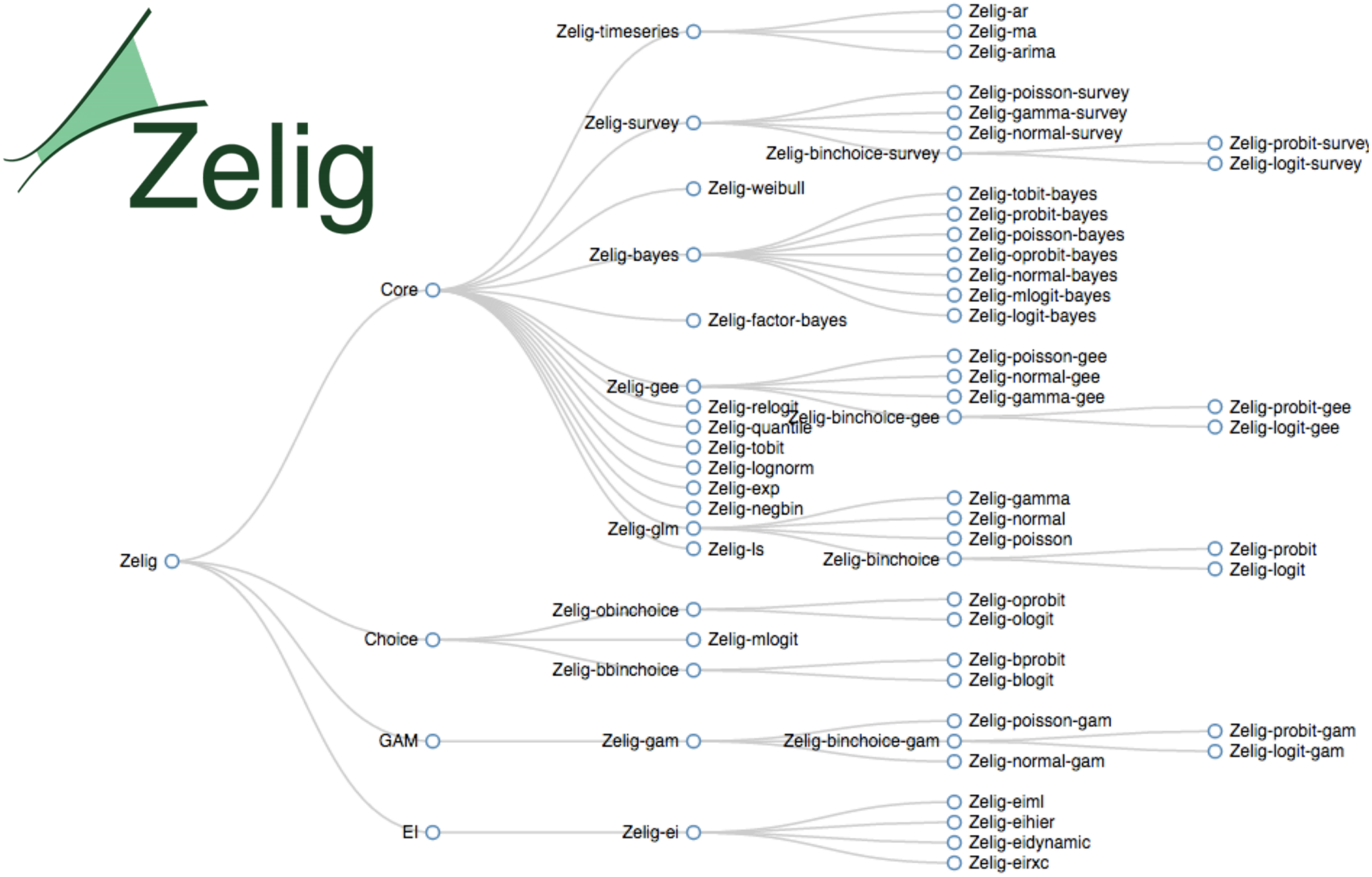
Dataverses by Category



Datasets by Subject



53 Stats Models, easy-to-use



Thank you!

iq.harvard.edu

The Dataverse Project

dataverse.org
dataverse.harvard.edu

Zelig

zeligproject.org

OpenScholar®

openscholar.org
projects.iq.harvard.edu
scholar.harvard.edu

Consilience Beta

Coming soon!

Presented by @mercecrosas mercecrosas.com