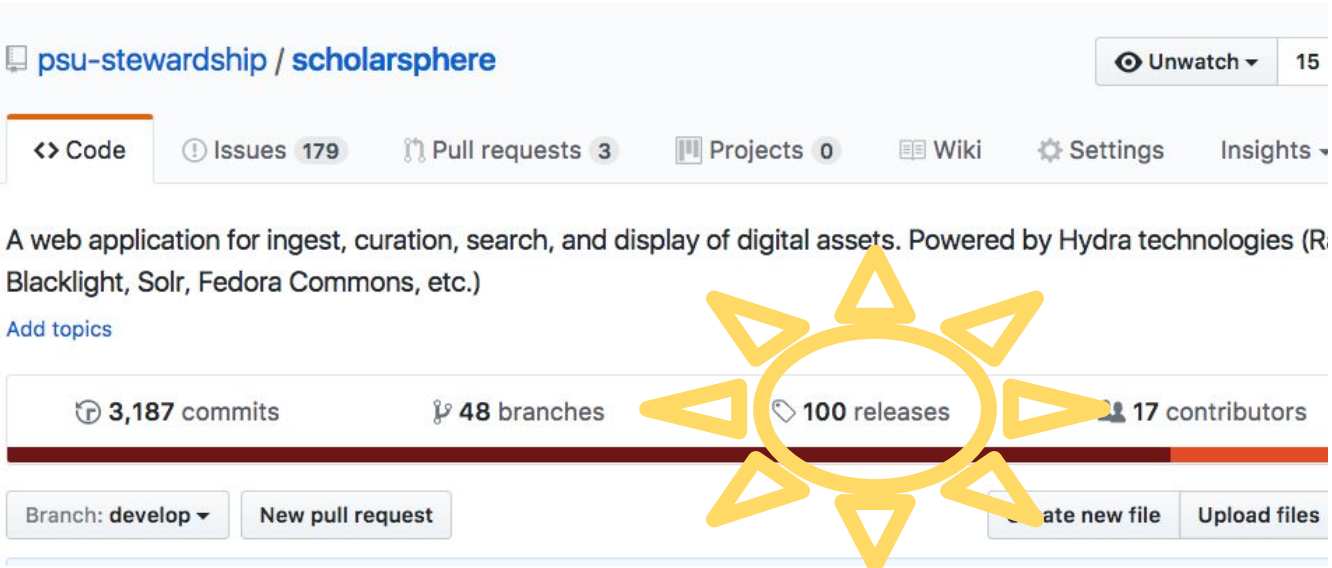
Lessons Learned From 100 Releases

http://bit.ly/2yte3aB

100 Releases Wow!



Outline

1. Review Release history
2. Lessons Learned

Releases

What Does 100 Releases really mean?

http://bit.ly/2lmdAlp

Major ScholarSphere Releases

1.0 - Initial Release - September 11, 2012

2.0 - UI Redesign - September 10, 2014

3.0 - Multi File Works - September 26, 2017

http://bit.ly/2gp4zmj

Major Sufia Releases

1.0 - Initial Release - April 17, 2013

2.0 - Dropbox and cleanup - June 19, 2013

3.0 - Rails 4 - July 22, 2013

4.0 - UI Updates - August 21, 2014

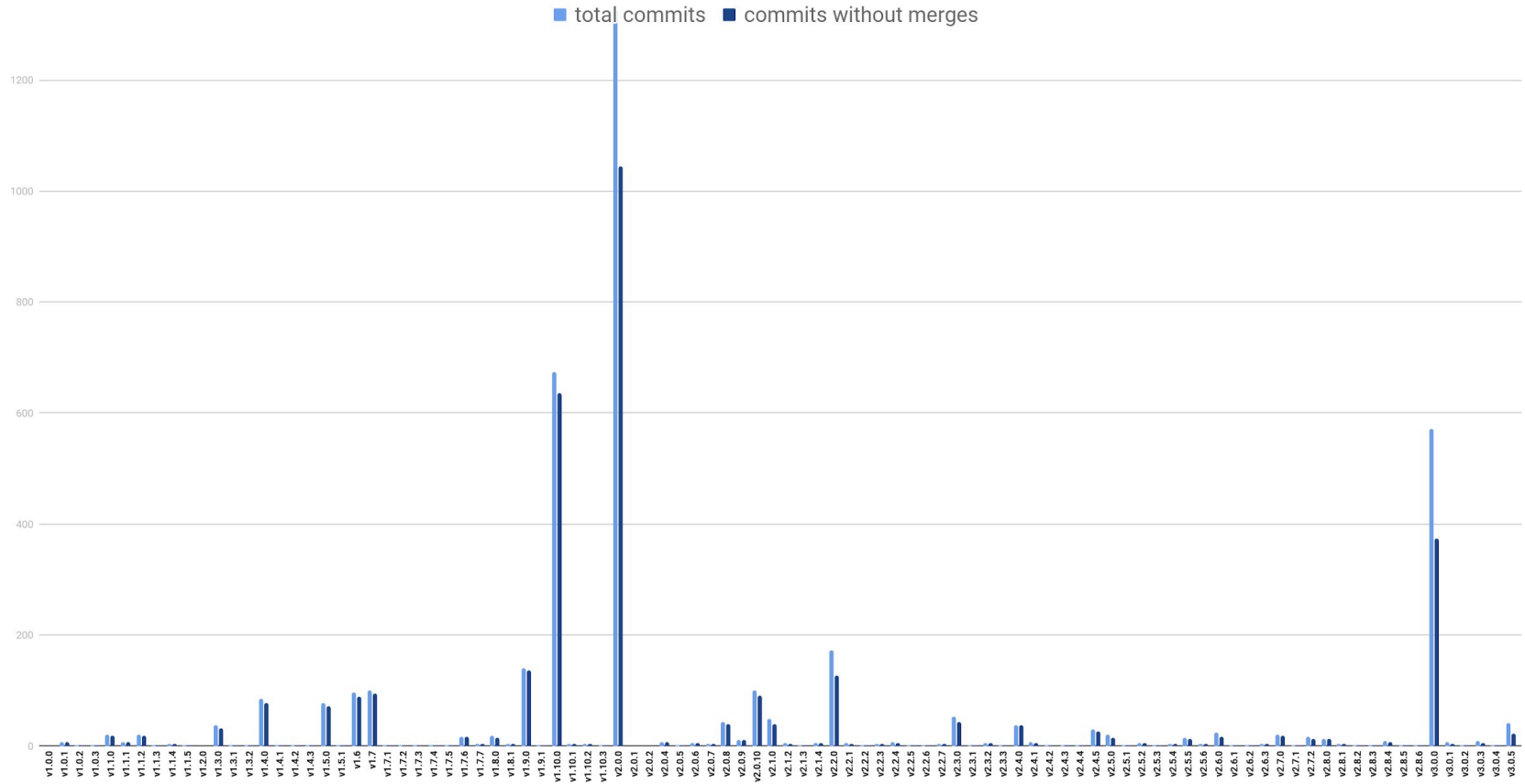
5.0 - Edit Form Rework - January 20, 2015

6.0 - Fedora 4 - March 27, 2015

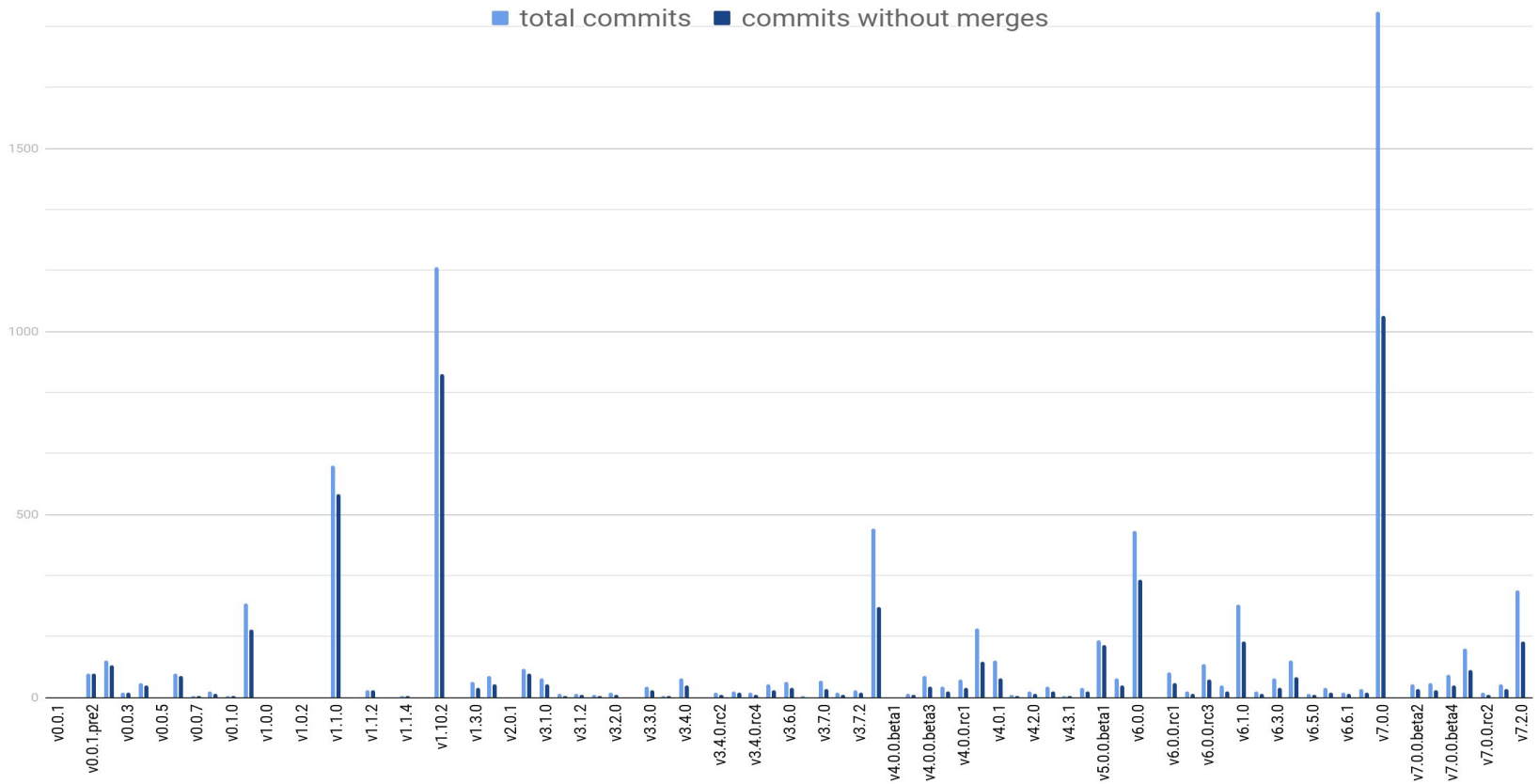
7.0 - PCDM - August 1, 2016

http://bit.ly/2gtvrFd

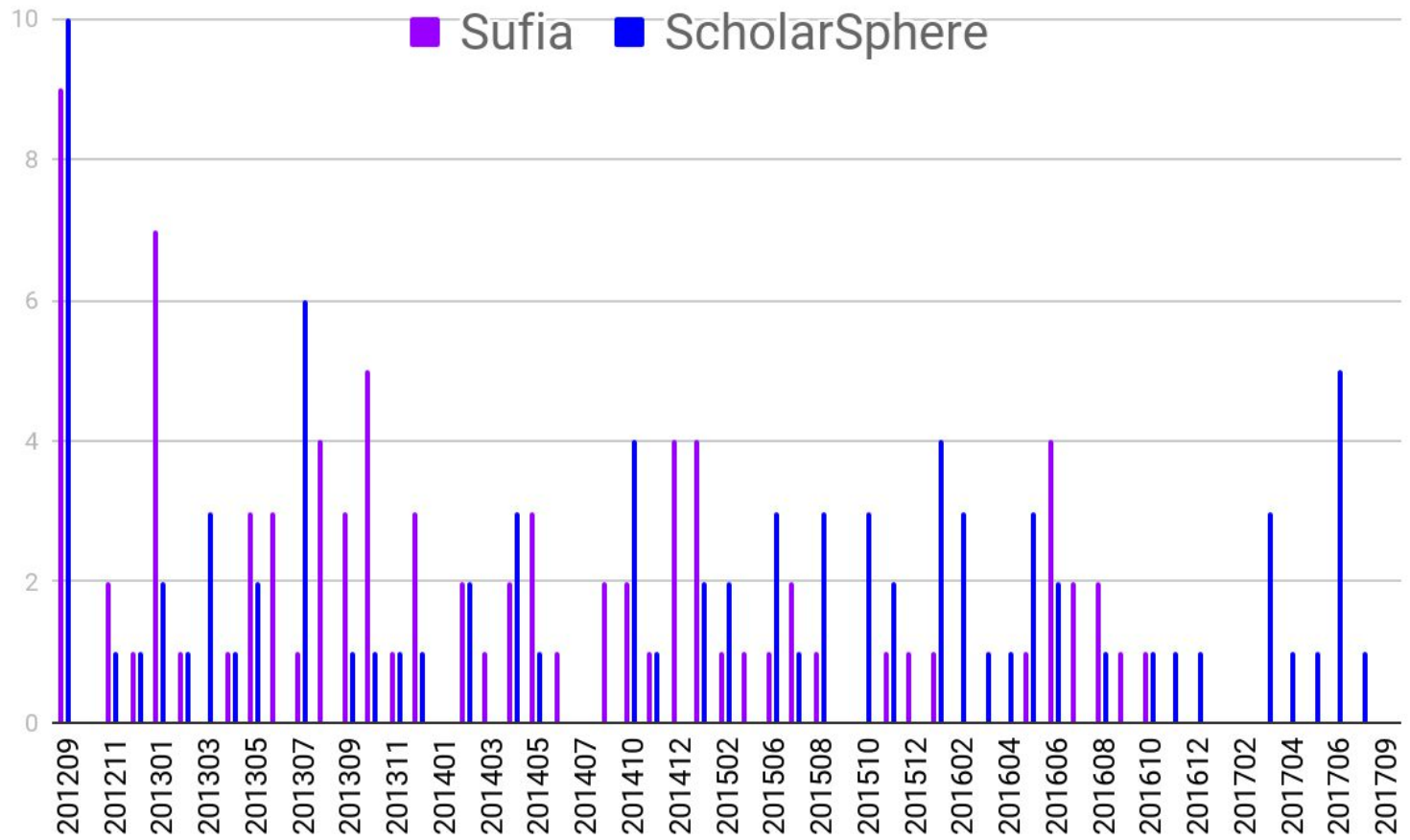
ScholarSphere Commits Per Release



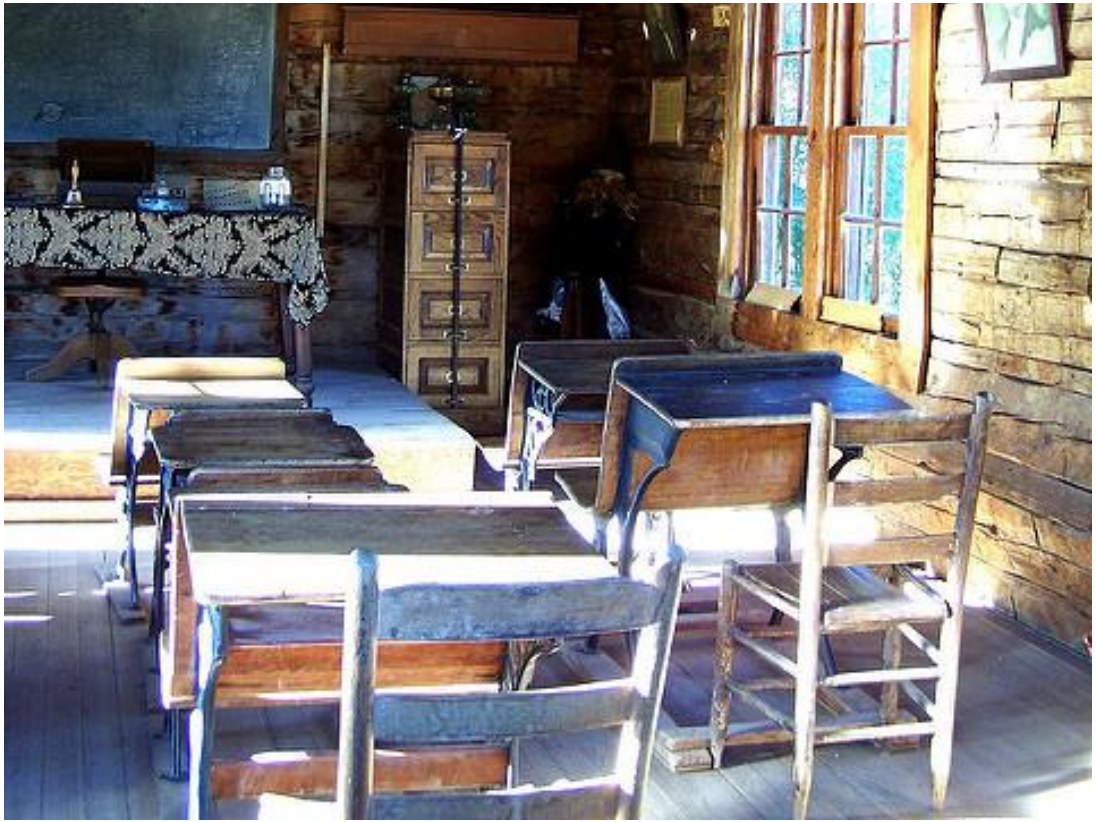
Sufia Commits Per Release



Releases Per Month



Lessons Learned



http://bit.ly/2y9vIF5

Small Changes Are Easier

80% of our Releases were less than 40 commits

http://bit.ly/2yamsjK

Details are important

But are really hard to see when you just commit directly.

Automate your deploy

20ish Commands on 2 machines quickly brings you down

http://bit.ly/2zKUEPl

Test Your Deployment

No one wants to be fixing system issues or installing dependencies during a code release.

Release When It Is Good Enough

If you release and continuously improve you will create a product that is more useful to your customers

http://bit.ly/2iEm0zL

Test! And Test Again

Thorough automated tests are only one part. You need:

- Automated testing
- Human testing
- End User testing
- Accessibility testing
- Performance testing

User Test

And then ignore 50% of what they tell you.

http://bit.ly/2iqXkKS

You Read Your Code

"Indeed, the ratio of time spent reading versus writing is well over 10 to 1. We are constantly reading old code as part of the effort to write new code. ...[Therefore,] making it easy to read makes it easier to write."

— **Robert C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship**

Don’t Forget what is behind the code

- Hardware
- MariaDB
- Solr
- Fedora
- Redis
- Fits
- Imagemagick

http://bit.ly/2imDcxi

Testing Servers are Essential

Especially when rails production mode is different than development mode you may find errors in production that do not occur in development.

Divide and Conquer

Isolation ~ Stability

http://bit.ly/2yaDjTO

Collaboration Is good

If you have the similar goals...

Goals include not just the feature, but also how that feature will be used!

Document Issues

Document the issue and the solution. You may see it again.

http://bit.ly/2iUh3qu

The Problem May be Behind the Keyboard

“Do not fear mistakes. You will know failure. Continue to reach out.”

- Benjamin Franklin

We will Clean up later...

Is a trap! Later never comes and you end up with a real mess.

Communicate!

The single biggest problem in communication is the illusion that it has taken place.

- George Bernard Shaw

Conclusion

“Be faithful in small things because it is in them that your strength lies.”

Mother Teresa

Thank You!

Carolyn Cole

Penn State University

cam156@psu.edu

Questions?

http://bit.ly/2yAoryc