

Helpful and Useful - The Open Source Software Metrics Holy Grail

Sean P. Goggins

September 16, 2018

1 Introduction

My colleague Matt Germonprez recently hit me and around 50 other people at CHAOSSCON North America (2018) with this observation:

A lot of times we get really great answers to the wrong questions.

Matt explained this phenomena as "type III error", an allusion to the more well known statistical phenomena of type I and type II errors. If you are trying to solve a problem or improve a situation, sometimes great answers to the wrong questions can still be **useful** because in all likelihood *somebody* is looking for the answer to that question! Or *maybe* it answers another curiosity you were not even thinking about. I think we should call this `_metric` encountering Erdelez (1997). There's an old adage:

Even a blind squirrel finds a nut every once in a while.

For open source professionals a "Blind Squirrel" is little more than the potential name for a Jazz trio, and probably not the right imagery for explaining to your boss that you're "working on open source metrics". Yet these blind squirrels will encounter nuts a LOT more often if we make more nuts! "Metrics are nuts!". Not a good slogan, but that's my metaphor. Making more metrics is easy for us because we have lots of data, we write software, and it stands to reason that more `_metrics` encountering is going to generate more *useful* metrics. If you are the blind squirrel, its useful to find metrics.

Can you imagine all the useful things blind squirrels would find if we let them loose in an Ikea? "I came for the Swedish meatballs, I left with 2 closet organizing systems and a new kitchen"! A lot of things are useful, but *in order for something to be helpful it needs to help **you*** meet an important goal. To summarize:

- **Useful:** Of all the different things I find in the Ikea, many of them are useful. Or, there are 75 metrics on this dashboard, and 3 of them are useful!

- **Helpful:** You go into the endeavor with a goal, and leave with 3 metrics that help you achieve that goal. Or, you're a blind squirrel that just ordered nuts online from Ikea.

2 Open Source Software Health Metrics: Lets go Crazy! Lets Get Nuts!

Great answers to the wrong questions are more commonplace than we prefer because source software work is evolving quickly and we do not yet have a list of *the right questions* for many specific project situations. Lets refer to **questions as "metrics"** now. Questions and metrics are nuts! Still a terrible slogan. Sometimes we do not know the question-metric-nut and foraging through a forest of metrics is, if not helpful, a way to reduce the rising anxiety we feel when we are not sure what data helps to support our explanation of what is *happening* in a project ecosystem. So, if like me and dozens of others working in and around the CHAOSS project, you are trying to achieve a goal for your project there are two orthogonal, strategic starting points our colleague in CHAOSS, Jesus M. Gonzalez-Barahona, suggests:

1. **Goals:** What are metrics going to help you accomplish?
2. **Use Cases:** When you go to use metrics, what are the use cases you have? A case can be simple, ill formed and even 'unpretty':
 - (a) "My manager wants to know if anyone else is working on this project?"
 - (b) "It seems like my community is leveling off? Is it? Or is it just so large now I cannot tell?"

2.1 Taking Action by Sharing Goals and Use Cases

Having a yard full of nuts to sort through can help you work toward the nuts you want. OK. The nut metaphor has gone too far. We are looking to use software, provided as a prototype and an example to help talk through the details of use cases you name. With you. The use cases of open source developers, foundations, community managers and others use to evaluate open source software health and sustainability metrics are probably a manageable number.

We can give you some metrics to work with quickly using the CHAOSS sponsored metrics prototyping tool Augur.

What are we trying to accomplish with metrics? With *Augur*? One of our goals is to make it easier for open source stakeholders to "get their bearings" on a project and understand "how things are going". We think that's most easily accomplished when comparisons to your own project over time, and other projects you are familiar with are readily available. Augur makes comparisons central.

2.2 Building Helpful Metrics

If you have already shared a list of repositories you are interested in with us, here's what you have;

1. an Augur site with those repos
2. The opportunity to look at that site and help the whole CHAOSS community know:
 - (a) What use cases which particular metrics help you address
 - (b) What goals you have that could be met by something like Augur, but you cannot meet yet
 - (c) Something to hate. If you've ever been to an NHL game, you know that hating the other team is how we show our team we love them. Its also a good brainstorming device.

So, OK. What do you want? We want the opportunity speak with you about your goals, use cases, and the failings of tools currently at your disposal for "getting there". If you're feeling adventurous, I would like to be able to reference our conversations (anonymously) in research papers, because research papers are kind of the "code of the academic world". That's less important.

2.3 An Augur Experiment

AUGUR If you do not have a list of repositories you have already shared with us, there are a few examples here: <http://www.augurlabs.io/live-examples/>.

Design Goals The version of Augur that's currently deployed has several design goals that seek to provide useful information through comparison within a project (over time) and across projects. The most fundamental metrics people are interested in include

- What individuals committed the most lines of code in a time period?
- From what companies or other organizations are the individuals who committed the most lines of code in a time period?
- Derivative of the first two: Is this changing? Did I lose anyone? Who can this project NOT afford to lose?

Projects You Care About Figure 1 is an example from Twitter, which shows an instance of Augur configured for all of the repositories in the Twitter ecosystem. When you go to <http://twitter.augurlabs.io> you get the list of repositories that you see in figure 1.

Figure 1: When you follow the URL above, or your own URL, you will see a list of repositories that we have cloned, and using the technology behind "Facade", a tool written by Brian Warner, calculated all the salient, basic, individual repository information about. Here's a list of those repositories.

Downloaded Git Repos by Project ▾

Twitter

URL	Status
github.com/twitter/twemoji	Complete
github.com/twitter/hadoop-lzo.git	Complete
github.com/twitter/thrift_client.git	Complete
github.com/twitter/twurl.git	Complete
github.com/twitter/elephant-bird.git	Complete
github.com/twitter/snowflake.git	Complete
github.com/twitter/spitball.git	Complete
github.com/twitter/ostrich.git	Complete
github.com/twitter/scala_school.git	Complete
github.com/twitter/finagle.git	Complete
github.com/twitter/icauth.git	Complete

Looking at my projects When I look at the most basic data for one of my repositories, I have enough information to answer the most basic questions about it (See above). Figure 2 and Figure 3 illustrate the Augur pages you will see at the next level of "drill down". *Try clicking the months for even more information!* Keep in mind this is ONLY the information for the repositories you shared with us, or the repositories part of one of our other live examples.

Figure 2: You can see the lines of code from the top two authors, as well as the space inefficient Augur tool bar. Please contact me if you have tips and tricks for getting developers to be more comfortable with putting aesthetics behind utility in web page design. I will buy you a case of beer.

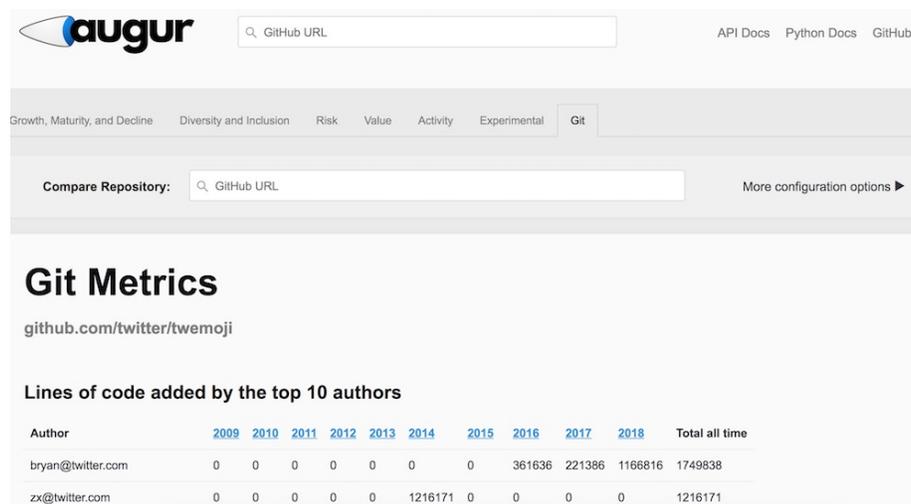


Figure 3 is a second image of the same page, but scrolled down just far enough to see that you can look at the top ten contributors as well as the top organizational contributors. We used a list of over 500 top level domains, as well as tech companies we were able to "guess" to start to resolve even these prototypes to specific companies. **We did this because Amye asked us to, and we're really gunning to make Gluster have more lustre. As if that's possible.**

Figure 3: A more detailed look at some of the information available on a repository by repository basis in Augur. We also show you the organizational affiliation information.

Lines of code added by the top 10 authors												
Author	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	Total all time	
bryan@twitter.com	0	0	0	0	0	0	0	361636	221386	1166816	1749838	
zx@twitter.com	0	0	0	0	0	1216171	0	0	0	0	1216171	
tom@twitter.com	0	0	0	0	0	0	0	869173	0	0	869173	
debian@debian.(none)	0	0	0	0	0	46273	0	0	0	0	46273	
joen@automattic.com	0	0	0	0	0	0	15850	0	0	0	15850	
andrea.giammarchi@gmail.com	0	0	0	0	0	0	1257	11582	135	0	12974	
ndowns@twitter.com	0	0	0	0	0	0	0	0	0	5815	5815	
agiammarchi@twitter.com	0	0	0	0	0	5583	0	0	0	0	5583	
prdolimos@gmail.com	0	0	0	0	0	0	925	0	0	0	925	
derek.mooney@blackboard.com	0	0	0	0	0	0	432	0	0	0	432	

Lines of code added by the top 5 organizations												
Author	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	Total all time	
(Unknown)	0	0	0	0	0	1268072	18478	1242462	221622	1172657	3923291	

3 Explore the Rest of Augur

The focused repositories give that information which many open source folks tell us is their first line of interest when looking at their own projects. Keeping this conversation going is essential for the CHAOSS project, and for Augur's utility for helping us identify which metrics map to which use cases and goals. There's a lot here, and it might give you ideas. Also, as you go through the front end, keep in mind that all of the statistics you see represented as metrics are also available via our Restful API. You can use our data to explore building your own metrics. Or get an app developer to do that for you. Figure 4 provides a high level overview of the metrics representations on Augur that are built off the GitHub API, GHTorrent and Facade's technology.

Figure 4: There's a lot here. At the top of the screen you can enter an owner and a repository name to get information about a particular repository. Each of the CHAOSS Metric working groups are represented in tabs at the top of the screen (number 1). The repository you just searched for is listed below the metric category (number 2). The metric name is listed in the title (number 3), and that title corresponds with a CHAOSS metric that is linked below the graphic. These are line graphs, though other visualization styles are readily available, and the line over time is shown by (number 4). The gray area around (number 4) is the standard deviation. (Number 5) is a slider like you see on Google Finance, so you can zoom in on one period of time more closely. Finally, (number 6) has a LOT of different configuration and filtering options you can explore.

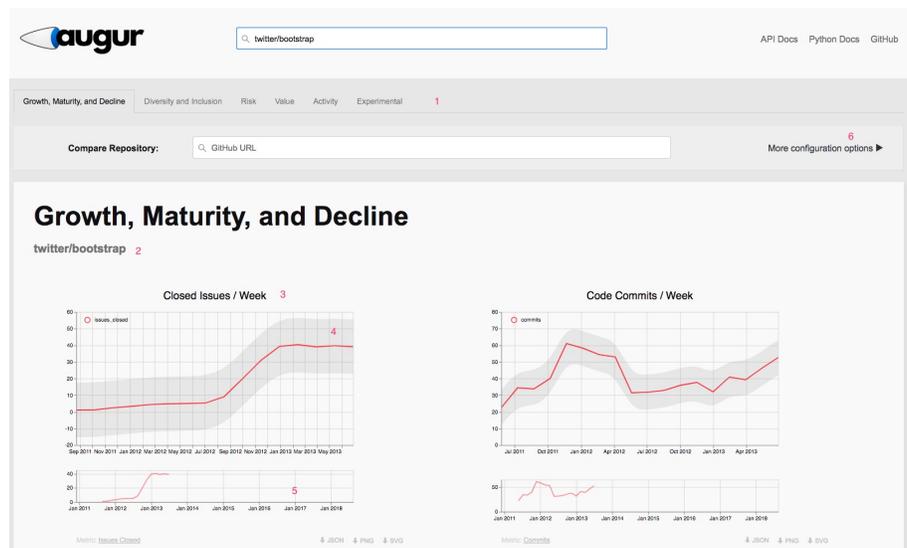
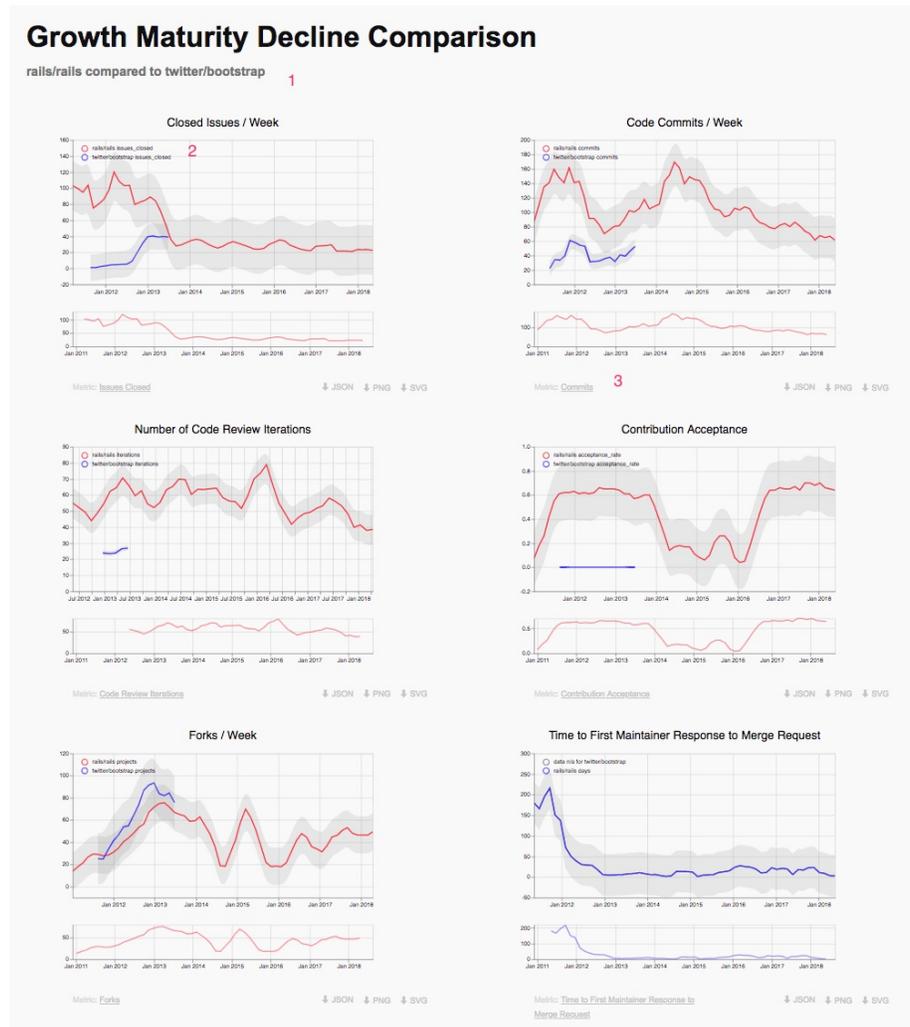


Figure 5: Here is a WAY zoomed out overview of the Growth, Maturity and Decline metrics you might see on the Augur page. (Number 1) is where you might enter another "owner/repo" combination to compare your repository to. (Number 2) illustrates that sometimes there is no data available from the source we use for a particular metric.



Figure 6: This shows you two repositories compared with each other in Augur. Does this fit any of your use cases or goals? How would you make it different? (Number 1) shows what two repositories are being compared. (Number 2) shows the key for knowing which project is which. (Number 3) points out, again, that you can see the CHAOSS definition for the metric any time you like. To the right, you can also see how .json, .csv and .svg representations of the data can be downloaded for you to make whatever use you would like to make of it.)



4 Our Ask: Goals and Use Cases

Metrics use cases What are the questions you have about your project? What metrics will help you to make clearer sense of the answer to that question in a productive way?

Give us your use cases Walk through trying to solve the use case? Where do you get stuck? How might the use case become generalized? If you are expert in openstack you can contribute you can just describe the use case. Draw out the use cases that you see. We can ask back, why not use metric x and y? And the conversation will really get going!

References

Erdelez, Sanda. 1997. *Information Encountering: A Conceptual Framework for Accidental Information Discovery*. Tampere, Finland: Taylor Graham Publishing.