Using the Magic of GitHub to Increase Developer Collaboration

DevOps Milwaukee
09/20/2016

Agenda

- Introductions
- Collaboration data
  - Intent
  - Examples
  - Tools
  - Visualization Techniques
- Q&A

Introduction

- Christian Weber
- Solutions Engineer
- Member since 2011
- Joined GitHub Q3 2015
- christian.weber@github.com
- github.com/webdog
- github.com/webdog/mke-meetup

The latest numbers

The latest numbers

PROJECTS	GLOBAL RANK	REGISTERED USERS
38M	#54	15M +

UNIVERSITY #'S	MONTHLY VISITORS	CORPORATE CUSTOMERS
100,000+	34M	65k

Our Products

Our Products

	.COM	ENTERPRISE ON-PREMISE	ENTERPRISE SELF-HOSTED/CLOUD
Platforms	Shared multi-tenant	vSphere, HyperV, Xen, OpenStack	AWS, Azure, IBM Bluemix
Common Use Case	• Public/Private Repos • Desire to engage others in the GitHub community • Committed to cloud	• Regulatory compliance • Internal constraints: audit, managing backups • Integrate existing on-premise tools into the workflow • Security: LDAP integration for provisioning / de-provisioning • Administer multiple organizations	

Recent Updates

- GitHub Enterprise 2.7
    - GPG Signed commits
    - Multiple assignees Issues/PRs
    - Reactions API
- GitHub.com
    - Projects now live
    - Code Review
    - GraphQL
    - Updated user profiles

Using the Magic of GitHub to Increase Developer Collaboration

Agenda

- Why?
- Overview of our API
- Using/Accessing the API
- Working with API Data
- Defining Collaboration Data
- Visualize our examples
- How to use this data

Why?

- Why?
  - Are we working the best way together?
  - Examining data surrounding our code helps us
  - We can identify blockers
  - Let's philosophize and ask questions

WORKING TOGETHER AS A TEAM?
IT'S A BOLD STRATEGY COTTON, LET'S SEE IF IT PAYS OFF
quickmeme.com

Collaboration on GitHub

- Opening/Closing Issues
- Opening, Merging Pull Requests
- Committing to the repository
- Conversations

API Overview

- API v3
- All requests done through HTTPS
- Can authenticate with username and password
  - Recommend Personal Access Tokens (OAuth)
- All data sent and received is JSON
- Timestamps are ISO-8601 (YYYY-MM-DDTHH:MM:SSZ)
- Summary vs Detailed Representations
- Requests for multiple items paginate to 30 items per segment
- Request capped at 5000 (Auth'd)/60 (Unauth'd) per hour

Accessing the API/Tools

- Python 3
  - Anaconda Distribution (3.5)
    - Lots of great scientific and mathematical modules included
  - Third party modules:
    - GitHub3 (Preferred module, supports both GitHub.com and GitHub Enterprise)
      - To access GitHub via the API
    - textblob (A lightweight front-end for NLTK)
      - Helps us clean up and tokenize data with minimal extra effort
      - Using default corpus for sentiment analysis

Askin' the hard stuff

- Questions we'll ask
  - What are my developers contributing?
  - What makes up a good (Merged) Pull Request?
    - What's our ratio of merged/unmerged PRs?
    - How often are we submitting net-negative PRs?
    - Codebase Contributor vs Repository Contributor?
  - How are we talking to each other?

Nuts and Bolts

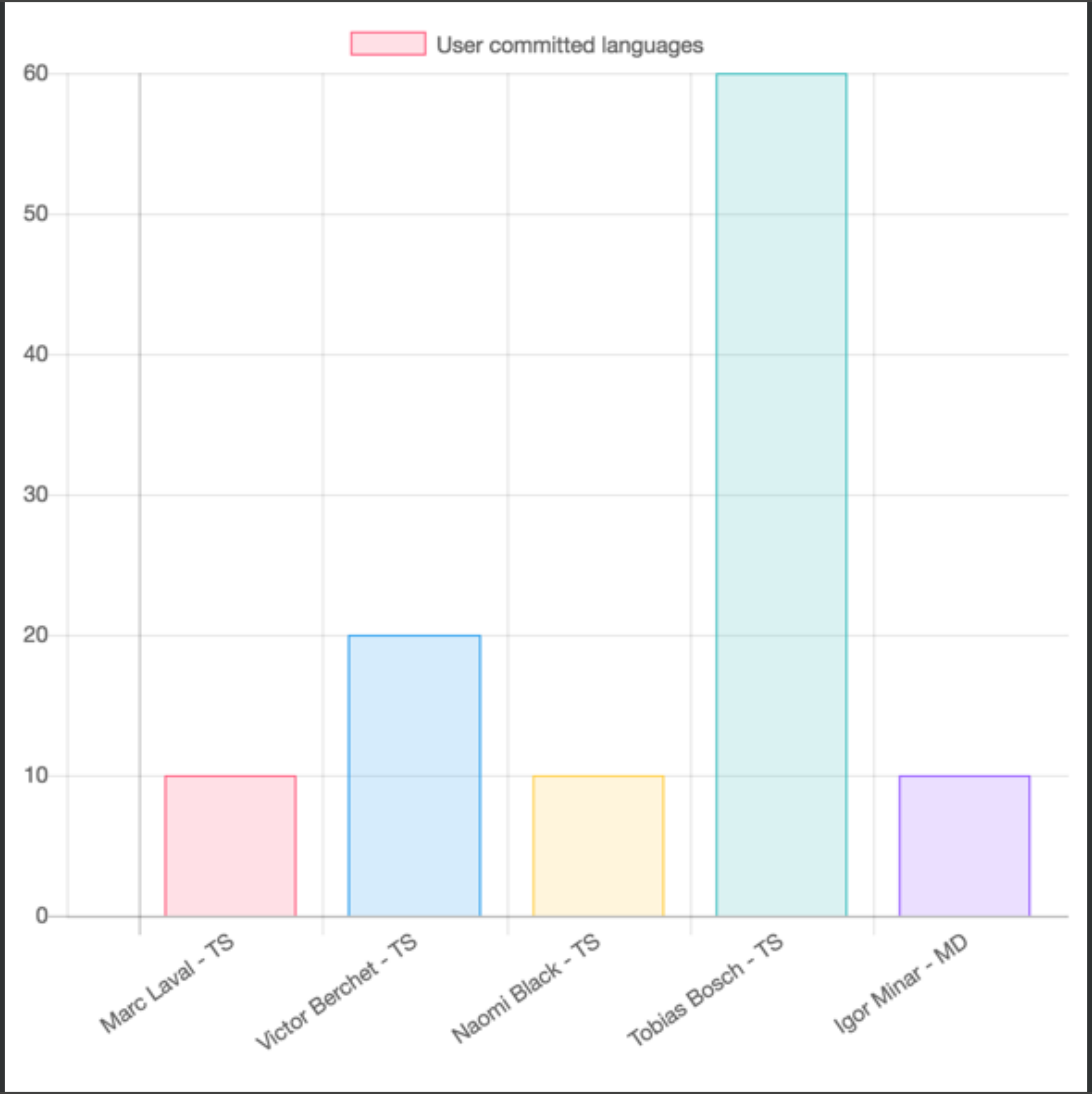
- The API calls for everything:
    - GET /repos/:owner/:repo/issues
    - GET /repos/:owner/:repo/issues/:number
    - GET /repos/:owner/:repo/pulls
    - GET /repos/:owner/:repo/pulls/:number
    - GET /repos/:owner/:repo/commits - Get the SHA ID
    - GET /repos/:owner/:repo/commits/:sha

What are my developers contributing?

- What are we going to look at?
    - Language Contributions by User in a Repository
    - Total number of commits by user, by file extension
- Subset of commits from the public Angular Project

What are my developers contributing?

- What can we solve?
- Identify large spreads
- Are my developers contributing to their strength?
- Are the contributions too large/ small?

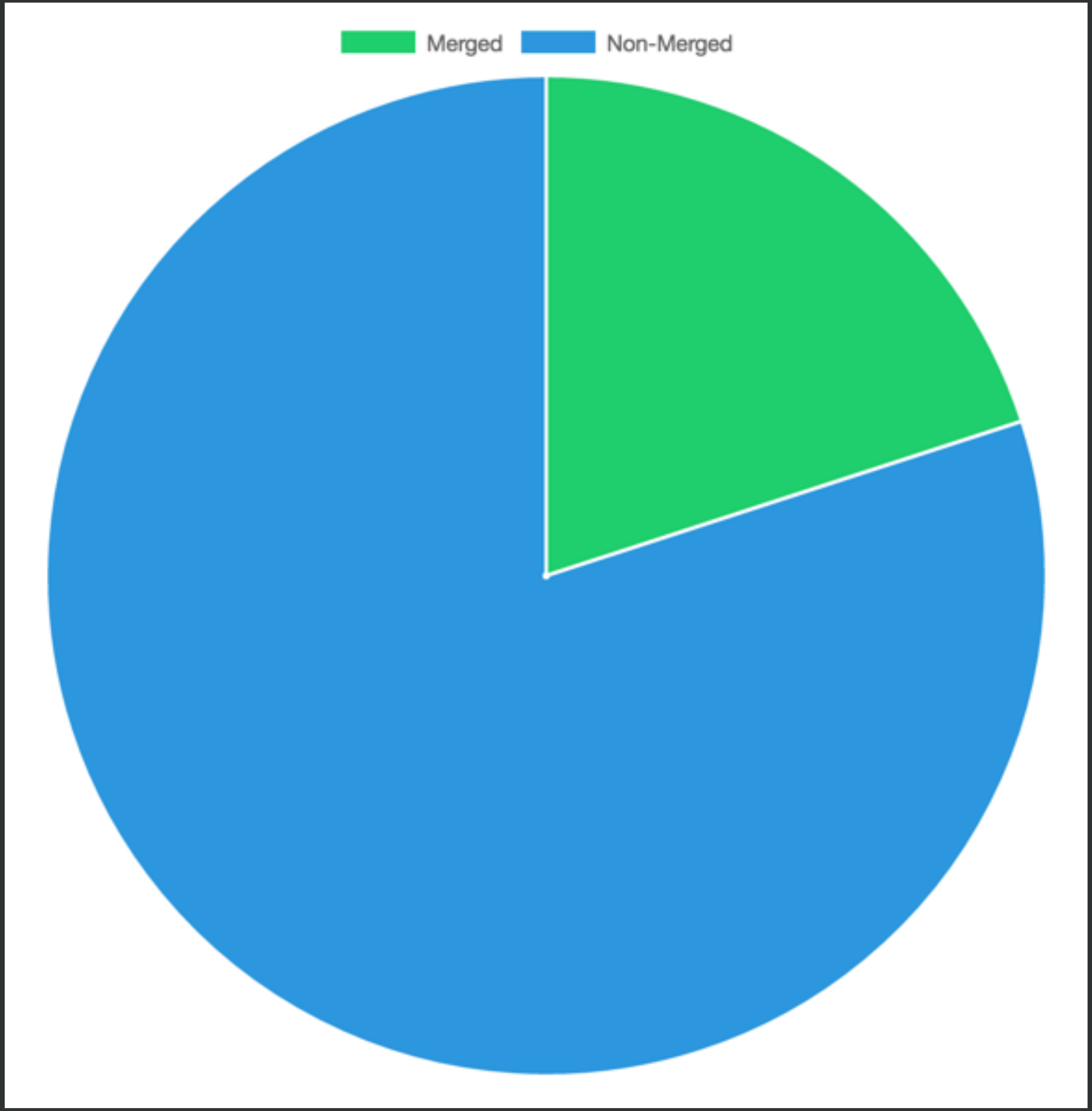


Merged vs Unmerged PRs

- What are we going to look at?
  - Merged vs Unmerged Pull Requests
  - Is a closed Pull Request necessarily merged?
  - Looking at a subset of Pull Requests from the Angular Project

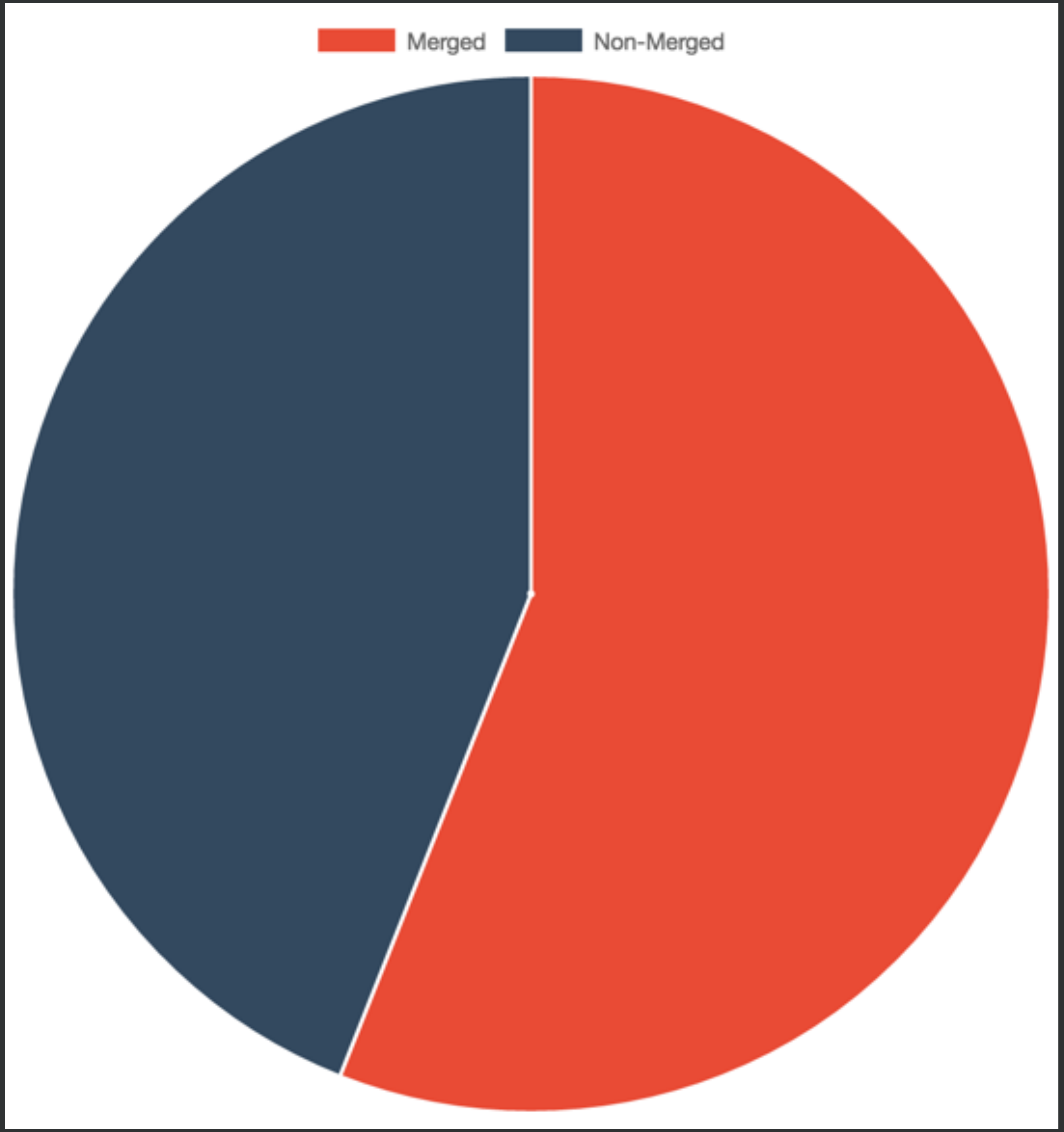
Merged vs Unmerged PRs, State=All

- Are we giving each PR it's due diligence?
- Are we writing duplicate code?
- Are we communicating requirements?



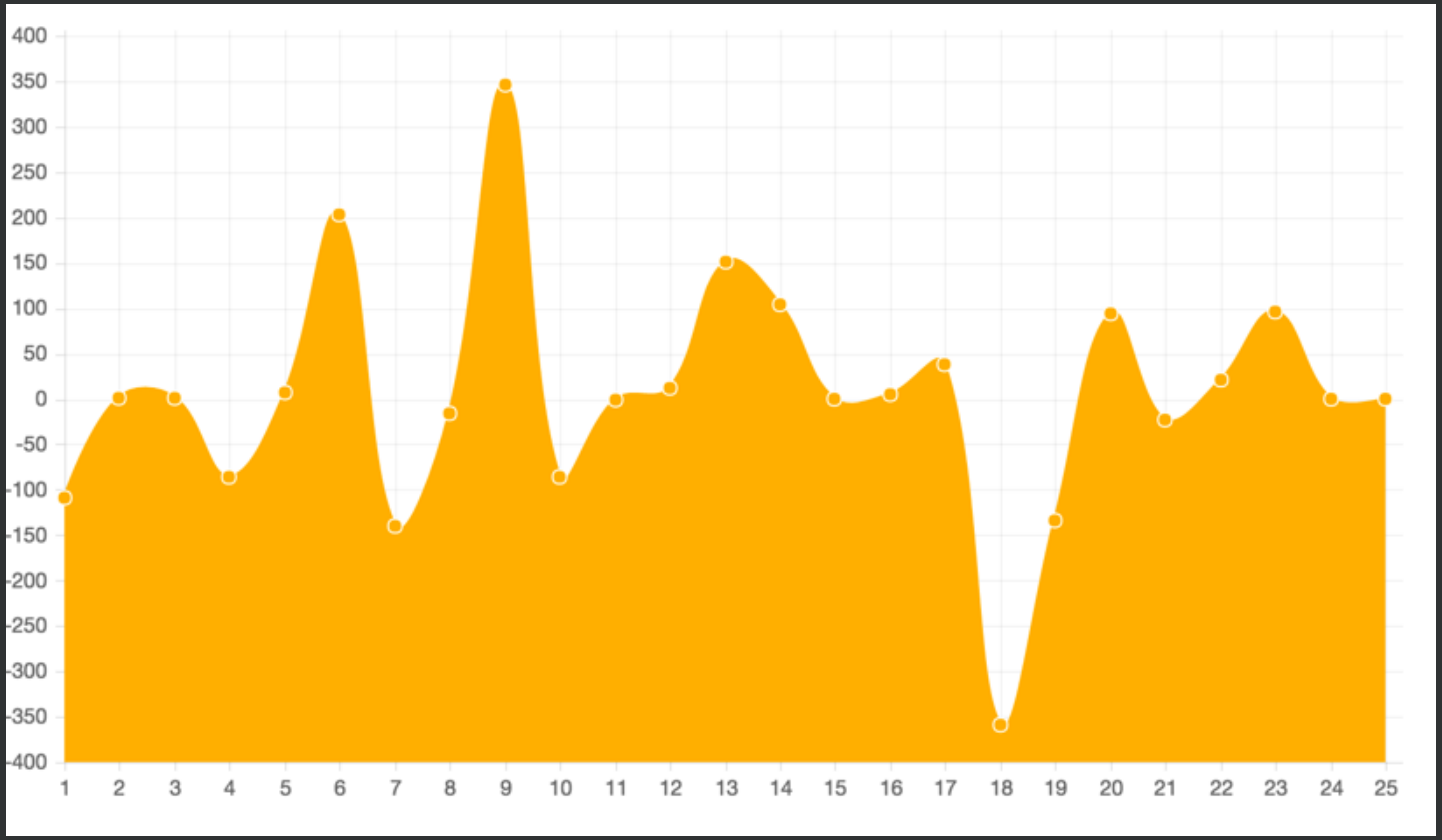
Merged vs Unmerged PRs, State=Closed

- What are we closing 1/2 of our PRs for?
- Do the contributions from non-merged-closed PRs live on elsewhere?



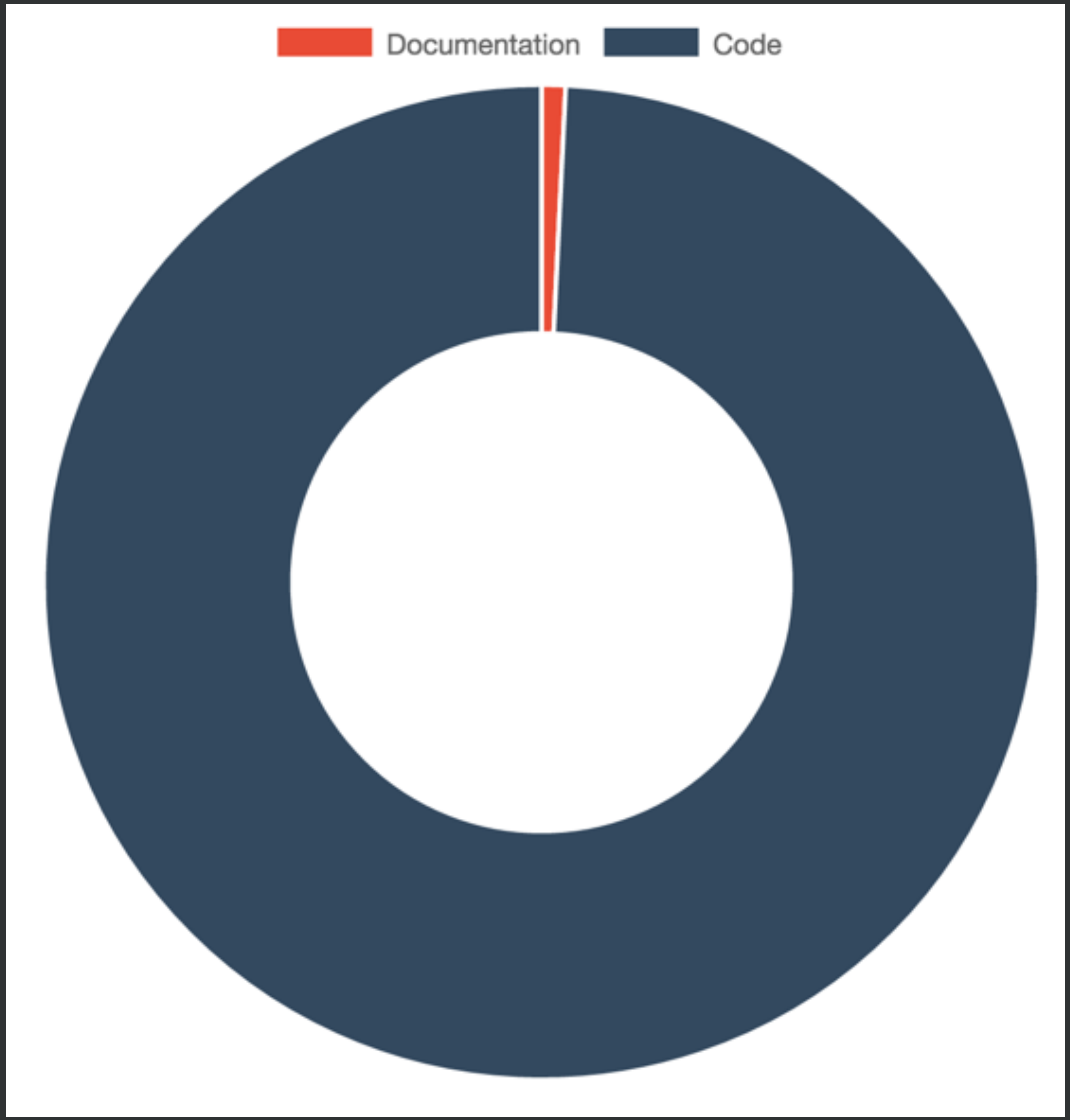
How often are we cleaning up code?

- Are there chances to refactor?
- Can we identify non-used code?
- Can we maintain a 1:1 neg/pos ratio?



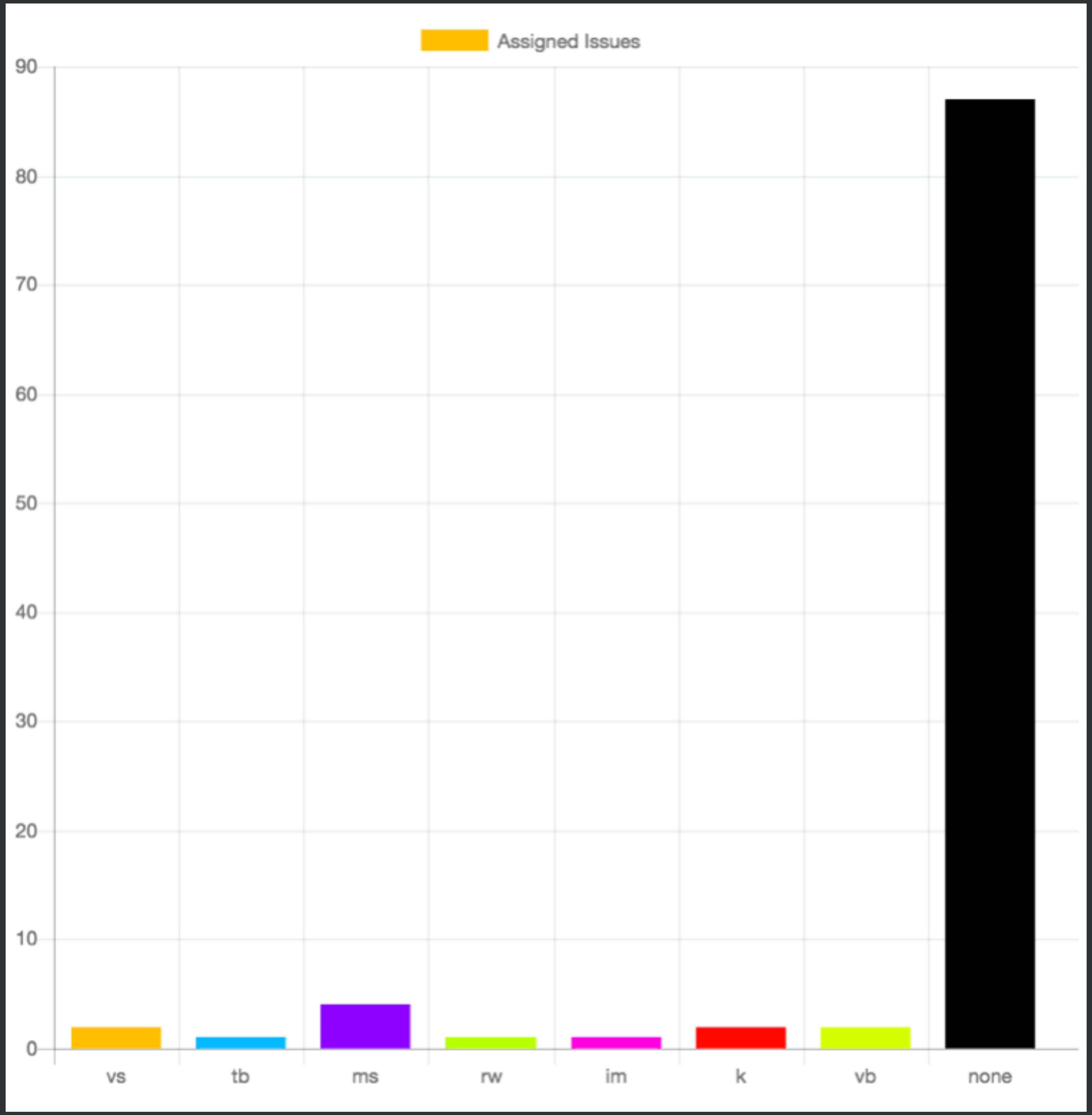
Repository Maintenance and Upkeep

- Are we keeping our docs in the repo?
- Are we updating docs with each PR?
- Can we have a developer in charge of docs?



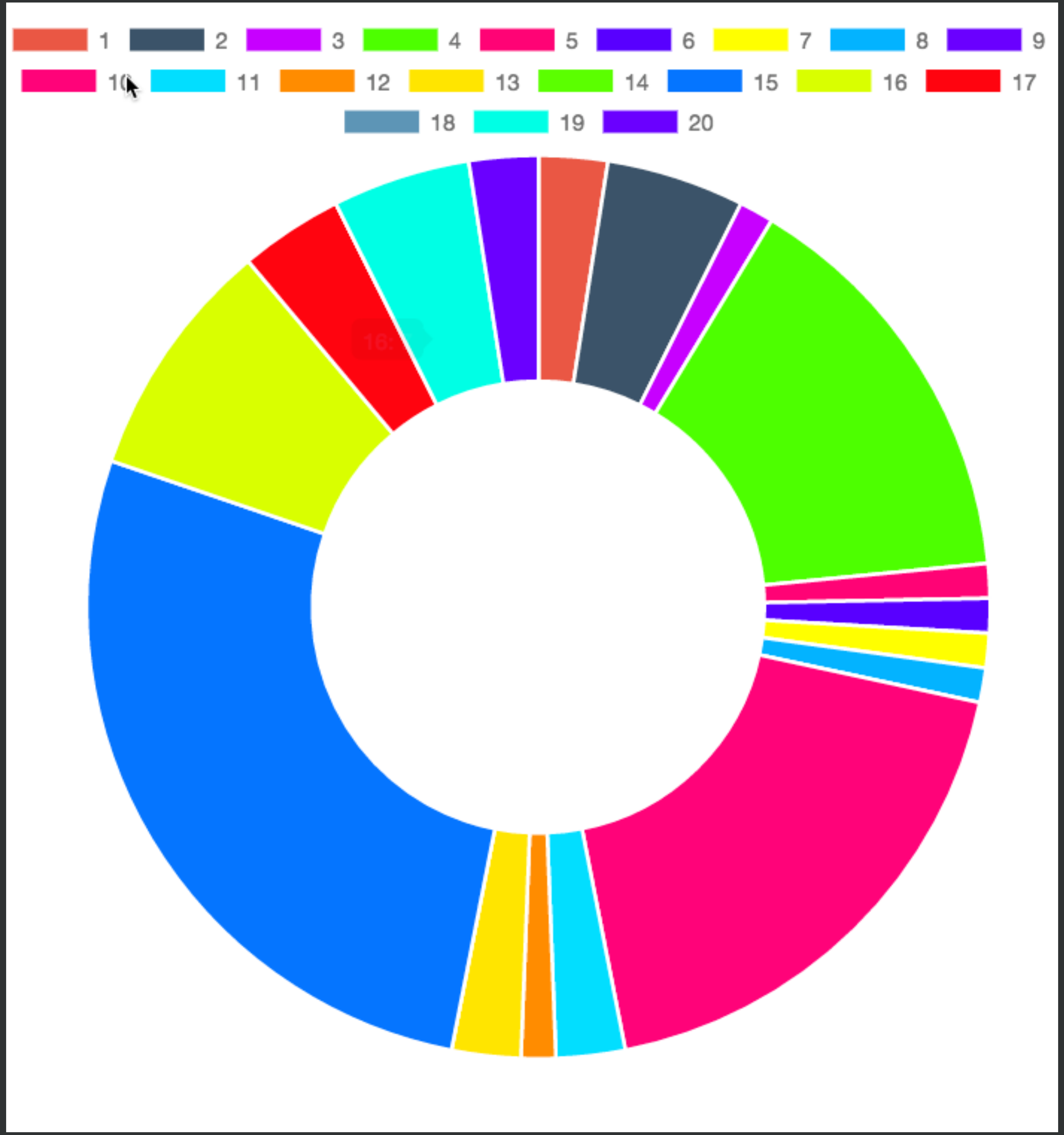
Assigned Issues - Open

- Reason for not assigning?
- Can we find blockers?
- Do developers feel empowered to take on issues?



Comments on Closed Issues

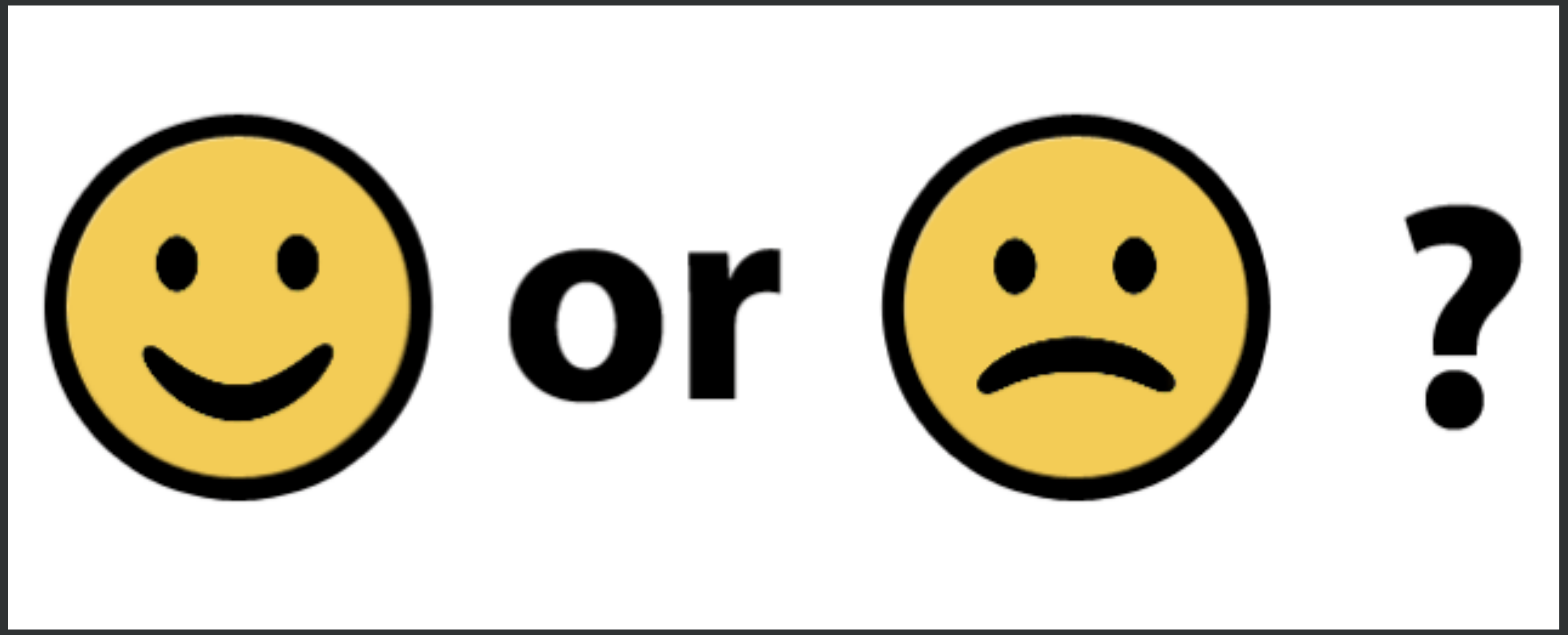
- Are we commenting enough on issues?
- Are we closing duplicates over and over?
- Am I communicating with my community
- Maybe Issue Templates can help spur conversation?



How are we talking to each other?

- What is it?
  - Opinion Mining/Sentiment Analysis
  - Helps determines attitude in context

- How can we use it?
  - Are we talking to each other in our best way?
  - Build discussion in our teams

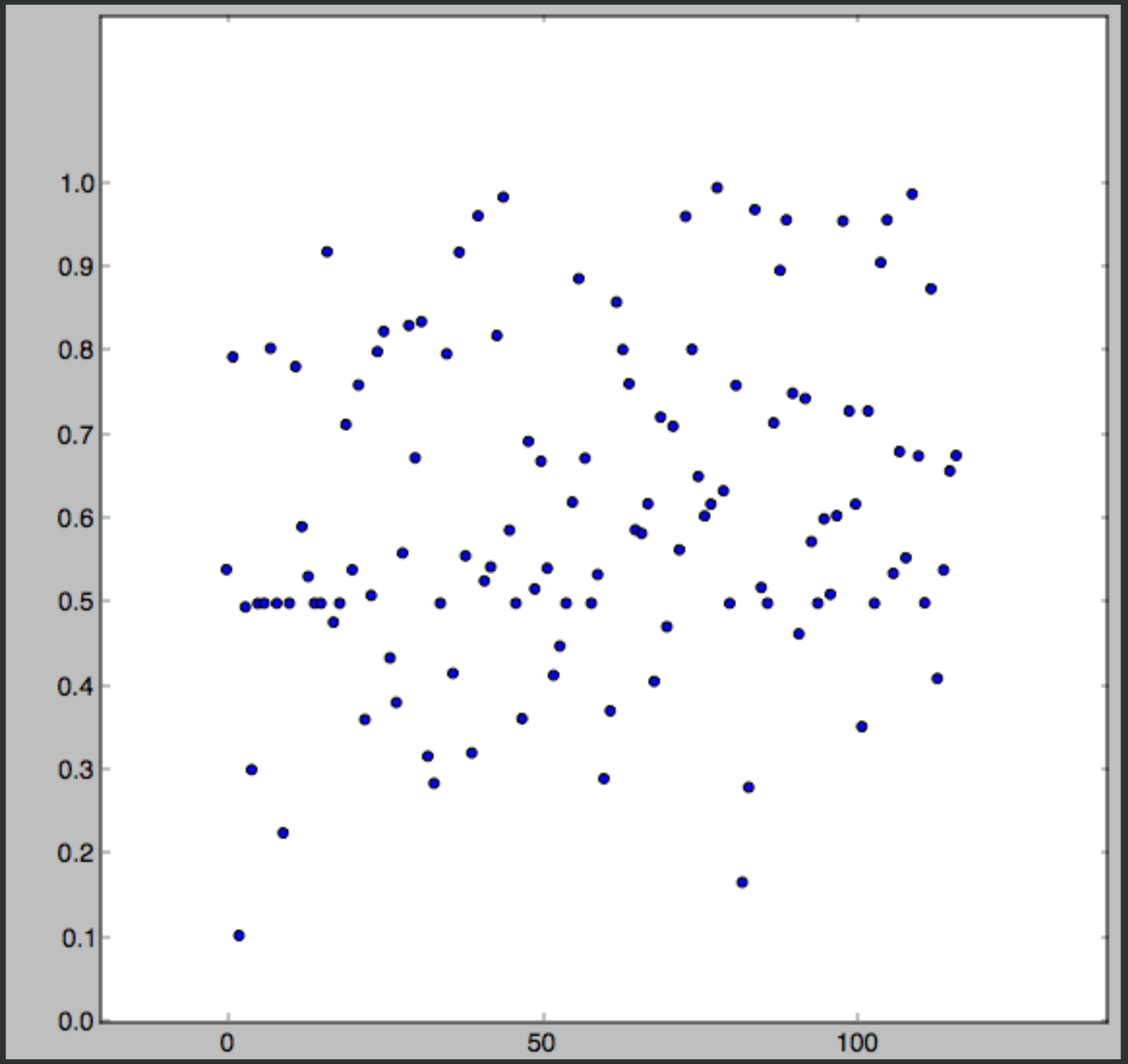


Sentiment Analysis Cont'd

- How to achieve in GitHub?
  - API calls (relative to api.github.com/v3):
    - By Org:
      - /user/orgs/:org/repos/:repo/pulls/:number/comments
    - By User
      - /users/:username/repos/:repo/pulls/:number/comments
  - textblob allows us to run a built in sentiment analyzer
    - Returns an index of positive, negative, and neutral indices:

```
Analyzing this sentence:
  I don't get this sentence and "anyways" is not a word.
Sentiment(classification='neg', p_pos=0.10376313589896774, p_neg=0.896236864101033)
```

    - Use a scatter plot with matplotlib to draw this dataset
  - Script available offline



Source: Anonymous open-sourced project on GitHub