Metrics Model: Community Activity - Requirements

Metrics (pick 5):

- Contributors - This visualization is a pie chart of top 10 contributors by commit (default), the chart will show the data. The user can change the action type and top numbers of contributions. And the user can range the data in a date they choose.
    - contributor_count : Determine how many active commit authors, review participants, issue authors, and issue comments participants there are in the past 90 days.
    - maintainer_count : Determine the average number of maintainers per repository.
- Code Change Commits - This visualization will be a bar graph of the number of commits per week for the past year or more.
    - commit_frequency : Determine the average number of commits per week in the past 90 days.
- Activity Dates and Times - This visualization will be a bar graph of number of different types of activity over time.
    - updated_since : Determine the average time per repository since the repository was last updated (in months).
    - created_since : Determine the average time per repository since a repository was created (in months).
    - comment_frequency : Determine the average number of comments per issue in the last 90 days.
    - updated_issues_count : Determine the number of issues updated in the last 90 days.
    - recent_releases_count : Determine the number of releases in the last year
    - meeting_count : Determine the number of meetings held in the last 90 days
    - meeting_attendee_count: Determine the average number of attendees per meeting
- Contribution Attribution - This visualization will be a bar graph of contribution types over time and the numbers of contributors.
    - org_count : Determine the number of distinct organizations that contributors belong to
- Change Request Reviews - This visualization will be a bar graph of change requests over time.
    - code_review_count : Determine the average number of review comments per pull request in the last 90 days

- Issues Closed - This visualization will be a bar graph of issues created and closed and a line graph of numbers of issues opened over time.
    - closed_issues_count : Determine the number of issues closed in the last 90 day

Software Overview

In this model, we want to show the user how active the project is often used, the increase and decrease of the change in community activity, simple ways to quickly identify how the project is managed, and how to collaborate with different communities. There are 6 different metrics, these visualizations will show the users different types of data of the software use.

System Requirements

- Software
    - python
    - 8Knot clone and install
    - docker Install
    - docker-compose install
    - git install
    - env.list at the top-level of the 8Knot directory that you clone
- Hardware
    - Not a specific requirement but must have enough memory space

Use Case (at least 1 in 2 metric model)

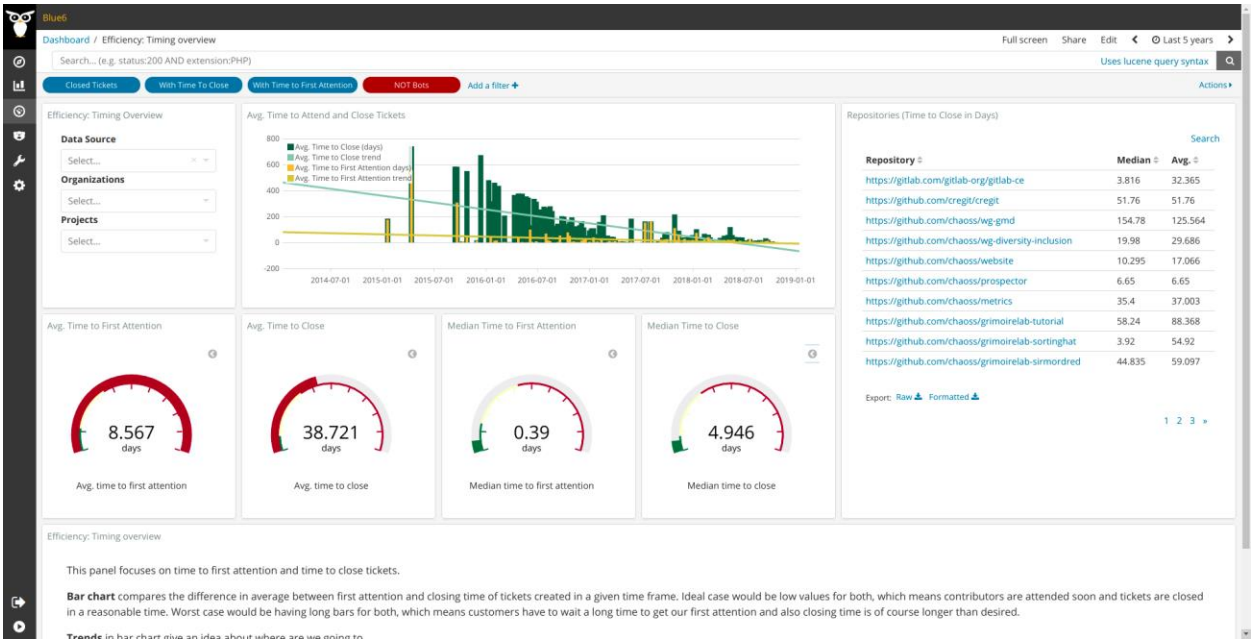
Component	Priority	Requirement Name	Description
Data source	1	Data type	Type of data
Data source	1	Database	The actual data
Data source	1	Data source	The source of data
Data retrieval	2	Software type	Software type, like Augur, Grall, Perceval
Data storage	2	Software type	Software type to store data
Program type	2	python	Type of programming

			language
Virtualization	2	docker	Deliver software in packages
git	2	Github	Place to store code/program
credential	2	env.list	Validation credential
User portal	3	Select database	User select where and which the database they actually want
Raw data	3	Raw data	Get actual data from database
Analyze data	3	Analyze data	Select ways to analyze data
Change data	3	Change data	Change the database
Display visualization	3	Display visualization	Display the visualization

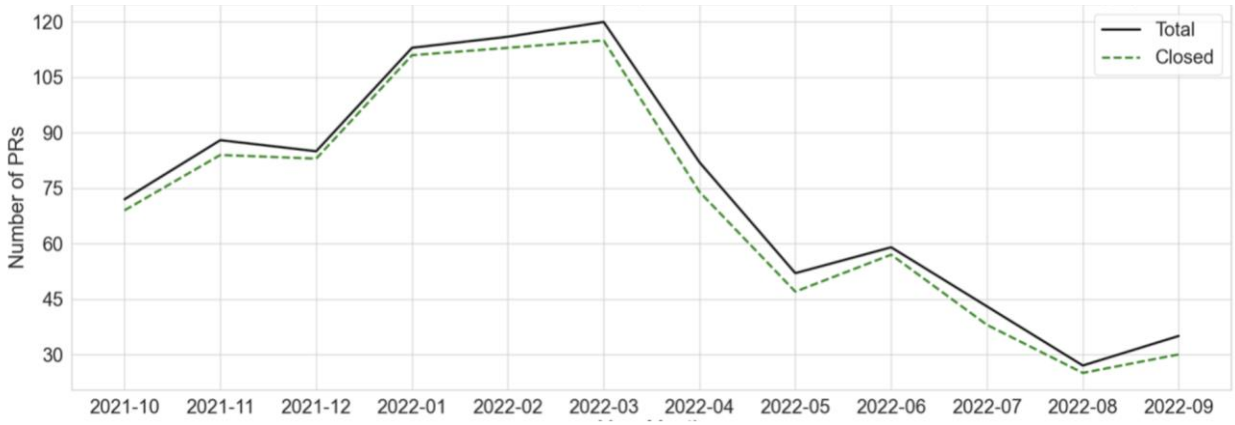
Metrics Model: Starter Project Health - Requirements

Metrics:

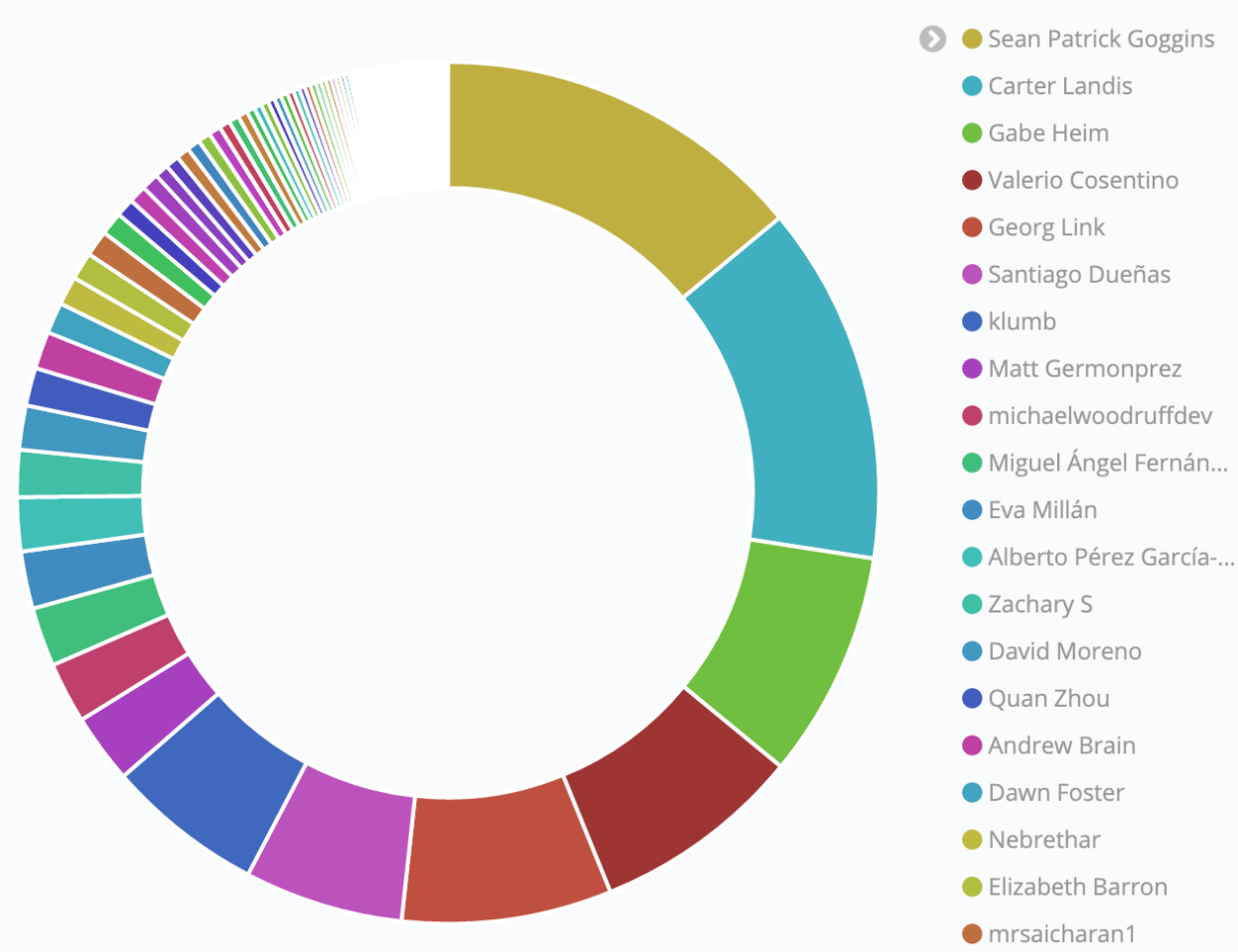
- Time To First Response
  https://chaoss.community/kb/metric-time-to-first-response/
    - Determine the amount of time between when an activity was opened (e.g. Issue or Change Request) and when it received the first response from a human.
        - The visualization will be a **bar graph**, x - axis will be the time goes by, the y axis will be the average time (days) to first response and close.
        - Description of the visualization
        - Display the average time to respond and close.



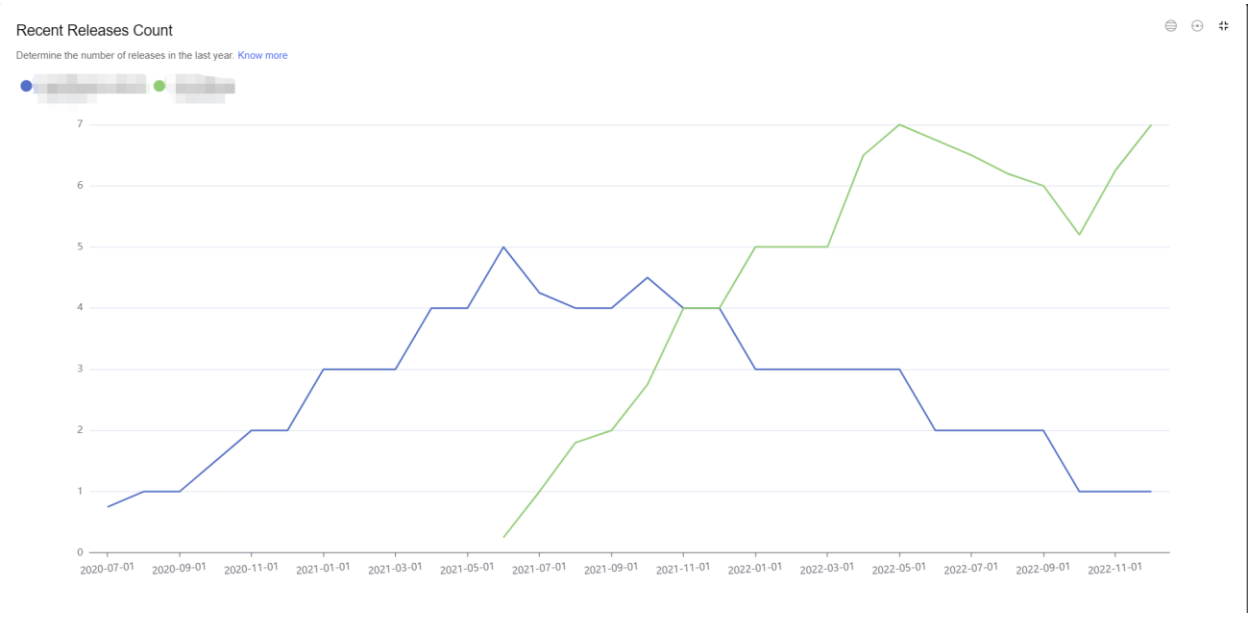
- Change Request Closure Ratio
  https://chaoss.community/kb/metric-change-request-closure-ratio/
    - Measure the ratio between the total number of open change requests during a time period versus the total number of change requests closed in that same period.
        - The visualization will be a **line graph**, x - axis will be the time goes by, the y axis will be changed (create, open, close, merged) over time.
        - Description of the visualization



- Bus Factor
  https://chaoss.community/kb/metric-bus-factor/
    - Determine the smallest number of people that make 50% of contributions
        - The visualization will be a **pie chart**, including the numbers of people's contribution and how much are they in percentage.
        - Description of the visualization



- Release Frequency
  https://chaoss.community/kb/metric-release-frequency/
    - Determine the frequency of project releases (including point releases with bug fixes)
        - The visualization will be a **line graph**, x - axis will be the time goes by, the y axis will be the numbers of project release, and the line graph will clearly show the frequency of the release over time.
        - Description of the visualization



Software Overview
In this model, we want to show the user how to help people get started with four key project health metrics that they can expand on and customize to meet their unique needs later.

System Requirements

- Software
    - python
    - 8Knot clone and install
    - docker Install
    - docker-compose install
    - git install
    - env.list at the top-level of the 8Knot directory that you clone
- Hardware
    - Not a specific requirement but must have enough memory space