Group 8

Design Document

Starter Health

Bus Factor - Andrew

Description:

The bus factor refers to the level of risk a project faces based on the knowledge concentration within its team. It measures how reliant a project is on individual team members. A high bus factor is desirable as it indicates that the essential knowledge for a project is distributed among many different team members. A low bus factor means that the essential project knowledge lies with few people. If these people were to be sick or disappear, then the project would be in jeopardy. In our context, the bus factor will be measured by the number of people who do 50% of the work.

Data Needed:

The data needed to calculate the bus factor of a product would consist of contributor IDs, the number of commits, comments, messages, etc associated with that ID, and the overall number of interactions within the project.

Toggleable Features:

The main toggleable feature of this chart would be filtering by type of contribution. You could choose between commits, comments, messages, or any other type of contribution. This is an important feature as commits are not the only way that users can contribute to a project.

Chart:

The bus factor would be best visualized using a pie chart. The bus factor would be displayed as a number above the pie chart while the chart itself is split into that number of slices displaying each user’s contribution percentage

Release Frequency - Joe

Description:

Release Frequency is a metric that determines the timeliness of a project's releases. Generally, more releases could mean often breaking code, or constant updates of software. This metric is great in determining the health of a project over the course of its lifetime. Delays in releases be detrimental to a project and its users, leading to security issues and insatiable versions.

Data Needed:

Release frequency is a function of time so the data needed would simply be a timestamp of a release. Additionally, for the toggleable features, other data needed could be a type of release.

Toggleable Features:

The main toggleable feature for this type of metric would be to filter by the type of release. Examples could include security patches, bug fixes, or feature releases. These features would drastically change the chart represented as software could go long periods of time without feature releases but release bug fixes every other week.
The user should also have the ability to change the view window of the graph.

Chart:

The chart associated with this metric would be a simple line chart, possibly with different points depending on the type of release highlighted. Averages and other stats like quarterly release frequency could be derived from the data.

Change Request Closure Ratio - Chris

Description: Change Request Closure Ratio measures 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. Having a swift decision about a request leads to benefits such as whether the project has enough maintainers, encouraging the closure of change requests that will not be merged, and monitoring long closure times.

Data Needed: This metric would need a time period and the number of closed change requests divided by the number of open pull requests during the same time period.

Toggleable Features: Some features include looking at different time periods and the types of change request/close. Options for the change requests could be whether they are bug fixes or new releases. For close, we can look at if the request was accepted or rejected.

Chart: A line graph would be the visualization plotted on time and number of pull requests. There would be two lines for the number of total pull requests and the number of closed requests.

Time to First Response - Kai

Description:

Time to first response is the time that passes after an activity is first created by somebody (e.g. a pull request, an issue report) before someone else gives feedback. This is important to measuring project health, as projects with lengthy times to first response may make developers feel alienated and ignored. On the other hand, a quick average time to first response is a sign of a healthy project, since development is active and changes can be pushed up and resolved quickly with less delays. For large projects that may take a long time to implement, this metric may have large variations over time as more contributors join or leave the project.

Data Needed:

Not much data is needed to calculate a project's average time to first response, since the metric is relatively simple. An individual activity's time to first response is calculated by the time the activity is first submitted minus the time it takes for a response to be made. With enough data points, we can average these values and get an accurate value of the average time a project's developer would wait before getting a response.

Toggleable Features:

One possible toggleable feature would include a feature that filters out automated responses from bots, as those may inaccurately drive the time to first response down. Another feature would be to sort the time to first response by the type of activity: for example, a pull request may be expected to have a larger time to first response than an issue report, since careful code review may be required for a pull request, while an acknowledgement would suffice for an issue report.

Chart:

Time to first response can be visualized via a simple bar chart that has each bar representing the average time to first response for a certain block time (e.g. a week, a month). Blocking by larger periods of time would give a more accurate impression as it would average out inconsistencies such as weekends and holidays that will have less people working. For this visualization, the x axis would represent time and the y axis would represent the amount of days required for an activity to be responded to.

Funding

Contribution Attribution - Kai

Description:
Contribution attribution describes how a project is developed behind the scenes. It analyzes who has worked on the project and their affiliations, determining whether their contributed work is volunteer work, sponsored work, or a mix of both. This will help viewers determine the dynamics and the diversity of the project community and the level of external support or funding.

Data Needed:
Contribution attribution relies on data that is volunteered from contributors, such as their gender, nationality, organization affiliation, what part of the project they worked on, and other defining traits. A simplified version of this metric would track less data, such as the organization affiliation and gender of the contributor.

Toggleable Features:
A chart showing contribution attribution should be able to toggle between different contributor identifiers. For example, a viewer should be able to switch between viewing the gender ratio of a project to viewing the organizational affiliation of a project.

Chart:

Contribution attribution would be most effectively represented via a pie chart, since this metric mainly focuses on ratios rather than quantities. Another possible implementation would be a bar graph, which would show specific numbers that would help gauge the scale of contributors.

Organizational Influence - Kai

Description:
Organizational influence, like its name suggests, measures the amount of influence an organization has on a project. This is important because an organization that has a large influence on a project may have a larger say on the direction of the project. Additionally, this allows the user to make a lot of assumptions regarding the project, such as how many hoops one has to jump through to contribute to a project, anticipating biases, and assigning accountability.

Data Needed:
The data required to measure organizational influence includes both the total number of contributors and their respective organizational affiliation. More in-depth methods of measuring

organizational influence could include organizational affiliation of members of the project's governing board and the type of contribution by an organization.

Toggleable Features:
A possible toggleable feature would be to change the display of the graph from showing the organizational affiliation for contributors to showing the affiliation for the project's governing board. This would allow users to view disparities between the affiliations of the upper management and the contributors.

Chart:

A pie chart would be an effective method of showcasing organizational influence, as the main focus of organizational influence is the amount of leverage an organization has in the project in comparison to other organizations invested in the project. A pie chart shows data in relation to each other by visualizing ratios, making it a good graph choice.

Types of contributions - Andrew

Description:
Types of contributions include commits, comments, managing the community, supporting users, tracking bugs, etc. Having a steady amount of each contribution is important to ensure an open-source project is healthy. This data provides insight into project governance and how new contributors are interacting.

Data Needed:
This chart would be complete if every single interaction that a user has with the repo could be tracked and queried. We'll only be using the most important pieces of data which include commits, pull requests, comments, messages, and merge accepts

Toggleable Features:
The user will be able to toggle (on/off) each piece of data described above. They can also choose a specific time frame to pull data from

Chart:

The chart will be a pie chart that compares each contribution type and what percentage each contribution type is of the project as a whole

Labor Investment - Chris

Description:
The Labor Investment metric measures the cost incurred by an organization for its employees to contribute to open source projects. It provides insights into the monetary investment made by organizations in terms of labor costs, considering contributions such as commits, issues, and

pull requests. The metric aims to enhance transparency in understanding labor costs, enabling Open Source Program Office (OSPO) managers to compare labor investments across a portfolio of projects.

Data Needed:

1. **Number of Contributions:** The total number of contributions made by employees.
2. **Contributor Types:** Classification of contributors as internal or external.
3. **Contribution Types:** Categorization of contributions (e.g., commits, issues, pull requests).
4. **Hourly Labor Rate:** The average hourly cost of labor.
5. **Average Labor Hours:** The average hours spent to create a contribution, categorized by contribution type.

Toggleable Features:

1. **Internal vs External Contributors:** Toggle to compare labor investment between internal and external contributors.
2. **Issue Tags:** Toggle to filter labor investment based on issue tags.
3. **Project Sources:** Toggle to differentiate labor investment in internal projects, open-source repositories, or competitor open-source repositories.

Chart:
A stacked bar chart or grouped bar chart can visually represent the Labor Investment metric. The x-axis can represent different contribution types, and the y-axis can represent the total labor investment cost. Different colors or patterns in the bars can signify internal vs external contributors or different project sources. Additionally, a line chart can depict changes in labor investment over time, providing a temporal perspective on the metric.

Organizational Diversity - Joe

Description:
Organizational diversity expresses how many different organizations are involved in a project and how involved different organizations are compared to one another. The more diversity within a project the more minds and coding styles are present. If you have one programmer from Amazon and one programmer from Exxon Mobile then you have two vastly different coding styles going into one project, for better or for worse.

Data Needed:
The main data needed to complete this visualization is a contributor id and a company name for a specific repo. The way that you get this company can vary however, this could become a toggleable feature.

Toggleable Features:

A feature that could be toggled is how to company is found example, through a GitHub bio or their email domains submitted with the contributor id. You should also be able to limit the data shown for contributors that have committed a certain amount of times.

Chart:

The chart needed for this visualization would just be a simple pie chart.