Introduction

This document outlines the requirements for the 8Knot Metrics Display Page, a software product designed to provide real-time insights into open-source project health. It details the system's function and non-functional requirements, design constraints, and the intended user experience.

System Requirements

Use Case 1: View Project Metrics
Actor: General User
Main flow:

1. User Navigates to the metrics display page on 8Knot.
2. User looks up an open source project to analyze.
3. System retrieves and displays the various metrics.

Postconditions: User gains insight into the projects health.
Exception paths: If the project data is unavailable, an error message is displayed.

Use Case 2: Change Graph Filters
Actor: General User
Preconditions: 8Knot metrics page is open.
Main flow:

1. User navigates to a graph.
2. User edits filters (e.g. date interval)
3. Graph data is refactored to match desired output.
4. Correct graph is displayed.

Postconditions: User is able to analyze project health based on new filter selection.
Exception paths: Project data unavailable, error message is displayed.

Two Metric Models:

1. **Starter Project Health**
    a. Bus Factor
    b. Change Request Closure Ratio
    c. Release Frequency
    d. Time to First Response
2. **Community Activity**
    a. Contributors Count
    b. Issues Closed
    c. Commit Frequency
    d. Recent Releases Count

Starter Project Health Metric Model Requirements:

- **Pie Chart: Bus Factor**
    - Pie chart displays workers on a given project and their level of significance to the project
    - The data needs to be displayed as parts of a whole project to analyze the bus factor
- **Histogram: Time to First Response**
    - Histogram displays each of the contributors on the project and the derived first-response-time for each of their work
- **Bar Graph: Release Frequency**
    - The bar graph will represent the release frequency.
    - Each bar will represent a period of time
    - The height of the bar will represent the number of releases in that period
- **Stacked Bar Graph: Change Request Closure Ratio**
    - This bar graph will effectively represent the ratio between change request
    - Similarly to the release frequency, each bar will represent a time period, however they're will be two bars for each time period
    - The height of the bars will represent the number of releases in that period

Community Activity Metrics Model Requirements:

- **Bar Graph: Contributors Count**
    - Each bar will represent a period of time and the height will be the number of contributors in the period
- **Bar Graph: Issues Closed**
    - Each bar will represent a period of time and the height will be the number of issues closed in the period
- **Line Graph: Commit Frequency**
    - This graph will be able to show the trends in commit frequency
- **Pie Chart: Recent Releases Count**
    - This would allow the user to see what year has the most recent releases effectively

System Functional Requirements:

- The system shall provide a real-time overview of key performance indicators for open-source projects.
- The system shall allow users to select a project from a searchable list.
- The system shall refresh metrics data periodically and upon user request.
- The system shall refractor displayed data when filters are changed.

System Non-Functional Requirements:

- Performance**:** Pages should load within 3 seconds under typical usage conditions.

- Reliability: System uptime should be at least 99.9%.
- Usability: The interface should be intuitive, requiring no training for new corporate users.
- Security: All data should be transmitted over HTTPS, and sensitive data must be encrypted at rest.
- Maintainability: The code should be well-documented to facilitate maintenance and further development.
- Portability: The display page should be responsive and work across the latest versions of major browsers (Chrome, Firefox, Safari, Edge).

Design Constraints

1. Interface Layout: The metrics must be displayed in a dashboard format, with graphs and charts that provide at-a-glance insight without overwhelming the user. The design must be clean and minimalistic to avoid clutter.
2. Color Scheme: The page must adhere to the corporate branding guidelines, using the company's color palette for consistency across the platform.
3. Responsive Design: The page must be fully responsive and capable of adjusting to various screen sizes, including desktops, tablets, and smartphones, without loss of functionality or aesthetic appeal.
4. Interactivity: Graphs and charts should offer interactivity, such as tooltips and zooming, but must not include animations that detract from the data's readability or performance.
5. Technology Stack: The design must be implementable with the current technology stack used by 8knot, without requiring additional frameworks or libraries that could introduce complexity or compatibility issues.
6. Update and Maintenance: The design should facilitate easy updates to metrics and features, with a design modular enough to allow for maintenance without significant downtime.

System Use

Overview: The 8knot Metrics Display Page is a dashboard designed for users to view and analyze metrics of open source projects.

User Interactions:

- Users will interact with the system primarily through a web-based interface that provides access to various metrics.
- The interface includes interactive charts and tables that users can sort, filter, and export.

Frequency of Use:

- It is anticipated that corporate users will access the system daily for up-to-date information on their tracked open source projects.

Sample Actor Survey Questions:

1. How often do you analyze community activity metrics for open source projects?
2. What specific metrics are most valuable to you when assessing the health of an open source project?
3. What features do you find most useful in a metrics dashboard?
4. How important is the ability to customize the dashboard layout and displayed metrics to you?
5. What are your biggest challenges when using existing metrics dashboards or tools?
6. How do you prefer to receive updates or alerts about significant changes in metrics?
7. Is there any other functionality you would like to see that would aid in your project analysis?
8. Please rate the importance of real-time data updates in your analysis.

Interfaces

1. **User Interface:** A web-based interface that prioritizes ease of use and an intuitive user experience. This interface will be designed to ensure simplicity in design while also allowing easy navigation. The interface will enable users to utilize the tool to its full potential. The goal is to create a user-friendly environment where the complexity of the data does not confuse the user. The design will be clean, effective, and meaningful. The interface will display graphs of the data that the user wants to see.