Log Analysis 1 - Project Documentation

1. Project Overview

Project Name:

Log Analysis 1

Team Members and Roles:

Nurlan Yagublu - Scrum Master
Siyu Chen - Product Owner
Nazrin Ibadli - Developer
Fei Wang - Developer
Zebai Tian - Tester

Description:

The Log Analysis 1 project involves parsing log files to extract structured data. The main goal is to analyze log entries with keywords like PORTEVENT and TIMEROP to understand system communication and behavior.

Key Objectives:

- Extract data from log files focusing on lines containing PORTEVENT and TIMEROP.
- Comprehensively analyze system communication and behavior.

Technologies Used:

Programming Language: Rust
IDE: RustRover
Platform: macOS 13.0
Version Control: Git, GitHub
CI/CD: GitHub Actions

2. Sprint Summaries

Sprint 1 (Demo1)

Sprint Duration: March 10 - March 25
Goals: Initial log parsing and data extraction focusing on PORTEVENT and TIMEROP.
Completed Tasks:

- Set up development environment.
- Initial implementation of log parser.
- Basic extraction of data from log entries.

In-Progress Tasks:

- Refinement of extraction algorithm.

Sprint 2 (Demo2)

Sprint Duration: April 2 - April 23
Goals: Optimization of the analysis and structured log extraction.
Completed Tasks:
- Optimization of extraction algorithm.
- Performance improvements in log parsing.
- Structuring extracted data.

In-Progress Tasks:
- Further performance optimization.
- Handling edge cases in log data.

3. Technical Implementation

Log Parsing Methodology:

The log parsing process involves reading log files and extracting relevant information based on specific keywords like PORTEVENT and TIMEROP. The data is then structured into a format that allows for easy analysis and understanding of system behavior.

Data Extraction Techniques:

- Using spaces as delimiters to split strings into multiple parts.
- Building a data frame from the extracted information.

Algorithm Optimizations:

Asynchronous multi-threading model to process multiple files concurrently.
Performance benchmarks: parsing 200,000 lines of log data in approximately 300-400 ms.

4. CI/CD Pipeline

Setup and Configuration:

CI/CD Tool: GitHub Actions
Pipeline Configuration Files:
- Build Pipeline: https://github.com/bytemaker-io/log-parser/blob/main/.github/workflows/rust-build.yml
- Release Pipeline: https://github.com/bytemaker-io/log-parser/blob/main/.github/workflows/release.yml

Continuous Integration Process:

Automated builds triggered on new commits.
Running tests to ensure code quality.

Continuous Deployment Process:

Automated release process including compilation, testing, packaging, compression, and uploading.