

Customer Requirements Document

Project: CodePulse

Engineering Metrics Dashboard

Veridian Dynamics (Client)

November 9, 2025

1. Project Overview

This document outlines the business and technical requirements for "Project CodePulse." This will be an internal web application designed to provide Engineering Managers and Leadership with actionable, real-time insights into the software development lifecycle (SDLC).

The system will function by integrating with our existing development tools (source control, task management) via their APIs. It will aggregate this data and present it in a clear, consolidated, and web-based dashboard.

2. Business Objective

The primary objective of Project CodePulse is to replace manual data collection and subjective reporting with an automated, data-driven platform. By visualizing key performance indicators (KPIs), we aim to:

- Identify and address bottlenecks in the development process.
- Improve development velocity and deployment frequency.
- Provide Engineering Managers with objective data to support their teams.
- Track performance against engineering goals and standards.

3. Key Features

The platform will be delivered with the following core features:

1. **Secure Authentication:** Users (Engineering Managers) must log in via our company's SSO (Single Sign-On) system.
2. **API Integration Module:** A secure, backend-only module for connecting to third-party services.
 - Must support integration with our **GitHub Enterprise** instance.
 - Must support integration with our **Jira Cloud** instance.
3. **Metrics Dashboard:** The primary user interface. This dashboard must display, at minimum, the following widgets for a selected team or repository:
 - **Pull Request (PR) Cycle Time:** Average time from PR creation to merge.
 - **Code Churn:** Amount of code that is rewritten or deleted shortly after being committed.
 - **Deployment Frequency:** Number of deployments to production over time.
 - **Task Throughput:** Number of Jira tickets (Stories, Bugs) moved to "Done" per week.
4. **Team Management:** A simple interface for administrators to define "Teams" by mapping repository/repositories and Jira projects to a single group.

4. Core Technology Requirements

To ensure maintainability and compatibility with our existing infrastructure, the solution **must** adhere to the following technology stack. Non-compliance is not an option without a formal review and exemption.

- **Frontend:** The user interface **must** be a single-page application (SPA) built with **React.js** (v18 or later).
- **Backend:** All backend services **must** be written in **Python** (v3.10 or later) using the **FastAPI** framework.
- **API:** All communication between the frontend and backend **must** be via a secure, stateless **RESTful API** using JSON.
- **Database:** The primary data store for user data, team configurations, and aggregated metrics **must** be **PostgreSQL** (v15 or later).

5. Acceptance Criteria (User Stories)

The initial delivery will be considered complete when the following user stories are demonstrable:

- **Login:** "As an Engineering Manager, I can log in to the CodePulse dashboard using my company SSO credentials."
- **API Connection:** "As an Administrator, I can securely add our GitHub Enterprise API token and Jira API token to the system's backend configuration."
- **Team Setup:** "As an Administrator, I can create a new 'Team' and associate it with one or more GitHub repositories and a specific Jira project."
- **PR Metric:** "As an Engineering Manager, I can view a dashboard widget that shows the average PR Cycle Time for my team over the last 30 days."
- **Task Metric:** "As an Engineering Manager, I can view a line chart showing the total number of story points completed by my team, grouped by week."
- **Deployment Metric:** "As an Engineering Manager, I can view a bar chart showing the total number of deployments to production for my team's primary service, grouped by week."