

Customer Requirements Document

Project: Changelogify Automated Release Note Generator

Veridian Dynamics (Client)

November 9, 2025

1. Project Overview

This document outlines the business and technical requirements for "Project Changelogify." This internal tool will automate the tedious process of compiling release notes (changelogs) for our software projects.

The system will work by integrating with our source control (GitHub) and task tracking (Jira) systems. It will scan for completed work within a given release period, parse the information based on defined conventions, and generate a draft of release notes for review and publication.

2. Business Objective

The primary objective of Project Changelogify is to dramatically reduce the manual effort required by developers and release managers to create changelogs. By automating this, we aim to:

- Save significant developer-hours currently spent on manual documentation.
- Ensure all user-facing changes (new features, bug fixes) are consistently documented.
- Improve the accuracy and professionalism of our release communications.
- Enable faster, more frequent release cycles by removing a common bottleneck.

3. Key Features

The platform will be delivered with the following core features:

1. **Secure Authentication:** Users (Release Managers, Developers) must log in via our company's SSO (Single Sign-On) system.
2. **Project Configuration:** An admin interface to "link" a new project. This must store:
 - The GitHub repository API path.
 - The associated Jira Project key.
 - API credentials for both services.
3. **Release Definition:** A user must be able to define a "release" by selecting a "start" and "end" reference (e.g., Git tag 'v1.1.0' and 'v1.2.0', or two commit SHAs).
4. **Automated Data Aggregation:** A backend process that, when triggered:
 - Fetches all Git commit messages between the two release references.
 - Parses commits based on the **Conventional Commits standard** (e.g., 'feat:', 'fix:', 'docs:').
 - Extracts Jira ticket numbers (e.g., 'PROJ-123') from the commit messages.
 - Queries the Jira API to get the official title for each ticket.
5. **Draft Editor UI:** A web interface that presents the generated draft, grouped by type (e.g., "New Features," "Bug Fixes"). Users must be able to edit this text before saving.

4. Core Technology Requirements

To ensure maintainability and compatibility with our existing infrastructure, the solution **must** adhere to the following technology stack. This is the same stack required for Project CodePulse, and all new internal tools should conform to this standard.

- **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 project configurations and saved release notes **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 a Release Manager, I can log in to the Changelogify dashboard using my company SSO credentials."
- **Project Setup:** "As an Administrator, I can create a new 'Product' in the UI and securely provide its GitHub repo URL and Jira project key."
- **Generation Trigger:** "As a Release Manager, I can select my project, choose a 'start' tag (e.g., 'v2.1.0') and an 'end' tag (e.g., 'v2.2.0'), and click 'Generate Draft'."
- **Parsing Logic (Features):** "As a Release Manager, when I generate a draft, I can see a 'New Features' section that lists the titles of all Jira tickets associated with commits prefixed with 'feat:'."
- **Parsing Logic (Fixes):** "As a Release Manager, when I generate a draft, I can see a 'Bug Fixes' section that lists the titles of all Jira tickets associated with commits prefixed with 'fix:'."
- **Editing:** "As a Release Manager, I can click into the generated text in the UI, manually add a new bullet point, and save my changes to the database."