

Sprint 3 Plan

Product Name: Code Performance Analyzer

Sprint 3 Goal

Sprint 3 will focus on building a backend test-generation framework, integrating selected code into test results, and adding file-saving functionality. On the frontend, we will set up a local server, connect the model results to the UI, and add a fallback mechanism for when the server is inaccessible.

User Stories - Backend

User Story 3.1: As a user, I want to be able to generate a test script to profile for real performance metrics like execution time and memory usage.

- **3.1.1:** Create test generation framework and endpoint – *8 hours*
- **3.1.2:** Integrate selected code and complexity for test results – *4 hours*
- **3.1.3:** Add file saving functionality - *6 hours*

User Stories - Frontend

User Story 3.2: As a user, I want to submit code through the extension and have the locally hosted model instantly return real complexity results, so the extension becomes functional.

- **3.2.1:** Create local server to host model endpoints– *8 hours*
- **3.2.2:** Integrate new model results into frontend– *4 hours*
- **3.2.3:** Create fallback for inaccessible server - *4 hours*

Team Roles

Team Member	Role
Philip Pesic	Frontend Developer
Adwaith Madadi	Backend Developer
Rohit Mandal	Backend Developer
Michael Pimentel	Frontend Developer
Juan Alvarez Sanchez	Frontend Developer

Task Assignment

Philip Pesic: Frontend Tasks 3.2.1, 3.2.2

Adwaith Madadi: Backend Tasks 3.1.2

Rohit Mandal: Backend Tasks 3.1.1, 3.1.2, 3.1.3

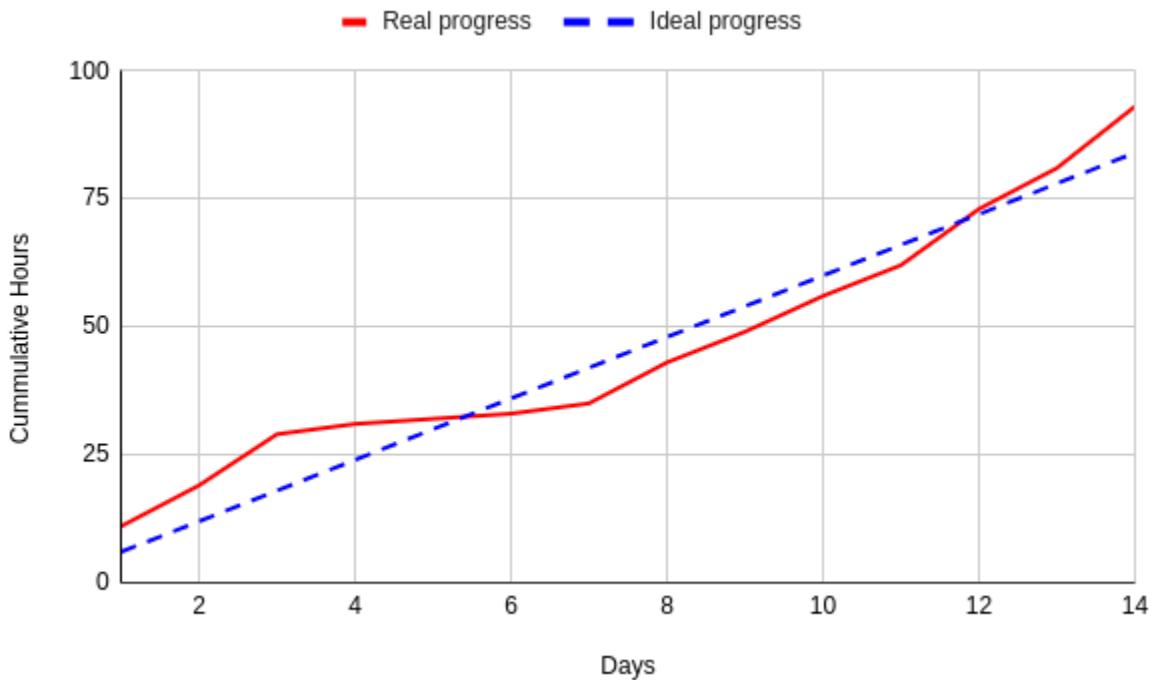
Michael Pimentel: Frontend Tasks 3.2.1, 3.2.2

Juan Alvarez Sanchez: Frontend Tasks 3.2.1, 3.2.3

Scrum Board

https://docs.google.com/document/d/15PlzT7TNhLe6Ni6R2BIPt3uAZVzQrlsptkJnA_Or7GM/edit?tab=t.0

Sprint 3 Burnup Chart



Scrum Meetings:

Day	Location
Saturday	Zoom
Tuesday	Zoom/In Person
Thursday	Zoom and TA Meeting

Sprint 3 Recap

In Sprint 3, the backend team created the test-generation framework, integrated code and complexity into test outputs, and added file-saving support. The frontend team built the local server, connected the model results to the extension interface, and implemented a fallback system to handle cases where the server could not be reached.