

Team Agreement

Product Name: Code Performance Analyzer

1. Definition of Done

- **1.1 Definition of Done for User Stories**

- **As a team, when we mark something as done, it should truly be finished. If we say something is done when it is not, it will create our team more problems later. A user story is done when:**

- All acceptance criterias are met.
- The feature works from start to finish and nothing major is missing.
- Everyone on the team understands what the feature does and how it fits into the project.
- The code is clean, readable, and follows our style guide.
- At least one team member reviews the code and approves it.
- There are no leftover debug prints, warnings, or console errors.
- We have tested the feature and confirmed it works correctly.
- The code is committed in a clean and organized way with no unnecessary files or clutter.

- **1.2 Definition of Done for Sprints**

- A sprint is done when our team feels organized and aligned. A sprint is complete when:
 - All user stories are either finished based on the definition above or moved to the next sprint with clear reasoning.

- The scrum board is updated so nothing is left unclear.
- The burnup chart is updated.
- We have completed a sprint review and talked about what went well and what needs to improve.
- Most of the next sprint planning is done before meeting with our TA.
- Everyone understands the project status and the decisions that are being made during the sprint.
- All our work is committed and fully integrated so nothing is left conflicting or unfinished.

2. Style Guide

- **2.1 Backend**
 - Write clear comments and usage instructions in the code and README.
 - Use snake_case for variables and functions.
 - Use capital for constants and directory names.
 - Organize the backend code into:
 - Model Training and Inference
 - Model Handling and Storage (Where models live, push/pull)
 - Data, Prompts, and other text files

- **2.2 Frontend**

- Use camelCase for function names.
- Use clear and descriptive variable names.
- Add comments when the logic is not obvious.
- Keep components simple and easy to read.

3. Project Cleanup (Before Final Presentations)

- Make sure all code is clean and readable.
- Remove unnecessary prints, warnings, and comments.
- Update all README files with clear usage instructions.
- Merged Branches.
- Test the entire project to make sure it works smoothly.