Agile

User stories:

1. As a vanilla git power-user that has never seen GiggleGit before, I want to understand how to commit and merge using GiggleGit, so I can use it for version control in my projects.

As a team lead onboarding an experienced GiggleGit user, I want to have documentation that captures best practices for using GiggleGit to ensure team members are effectively using it for collaboration.

2. As a developer using GiggleGit, I want to view merge conflict resolutions as memes, so I can resolve conflicts in an engaging way.

Task: Create meme selection algorithm for conflict types.

Ticket 1: Implement Conflict Type Classification

Details: Write a function that classifies merge conflicts and tags them with the appropriate category for selection.

Ticket 2: Build Meme Recommendation Engine

Details: Develop an algorithm that selects a meme based on the conflict type, ensuring that the meme fits the context and humor style of the conflict.

3. This statement is not a user story because it does not include the purpose of the feature. It's more of a system requirement since it tells us what feature needs to be implemented but not the motivation behind it.

Formal Requirements

Goal: Create an enjoyable intuitive experience for users merging when using Snicker Sync

Non-goal: Implement additional merging features that are not currently offered in GiggleGit.

1. Non-Functional Requirements: Data Handling

Functional Requirement:

a. Automatically backup user data to prevent data loss in case of a bug.

b. Integrate exception handling to prevent invalid inputs from being entered into the system and what to return if the case.

2. Non-Functional Requirements: Access Control

Functional Requirements:

a. Implement role-based access controls to ensure only PMs can modify users.

b. stories permission data in a secure database to prevent unauthorized access.