Agile

User Story 1:

- As a vanilla git power user who has never seen GiggleGit before, I want to understand the key differences and advantages of GiggleGit compared to the vanilla base git.

User Story 2:

- As a team lead onboarding an experienced GiggleGit user, I want to ensure new team members can easily integrate into the workflow to minimize friction during collaboration.

User Story 3:

- As a developer unfamiliar with both git and GiggleGit, I want to learn how to download and use GiggleGit to track my code versions.

  Task: Implement documentation

    Ticket 1: Create a “quick start” page guide

    - A web app should be created (using something like React) with a download page that allows users to download GiggleGit and it should also have instructions for new users to set up GiggleGit and connect it with other software like GitHub

    Ticket 2: Build a documentation interface

    - Create a UI in the web app that outlines all of the commands, and the syntax, and describes the functionality and parameters of the commands
    - Basically create a manual

The statement "As a user I want to be able to authenticate on a new machine" is more of a functional requirement rather than a user story. A user story should focus on the user's needs and it should provide context for why the feature is important, and especially focus on the exact type of user (frequent user, new user, etc.).

Formal Requirements

Goal:
Conduct user studies with the base vanilla UI and other variants of improved diff UI using different snickering concepts and get user feedback to determine the best UI design.

Non-goal:
Add more compatibility and functionality with other GiggleGit packages

Non-Functional Requirement 1: User Study Group Assignment

- Ensure the ability to assign users randomly to study groups comparing the base vanilla UI and its variants.
- Functional Requirements:
    - The system must randomly assign users to either the vanilla UI group or one of the variant groups to curb biases.
    - The system must log and report the assignments, showing how users were split across the study groups.

Non-Functional Requirement 2: PM Access for Snickering Concepts

- Provide access control so that only PMs can manage, modify, or update the different snickering concepts used in the study.
- Functional requirements:
    - PMs must have an admin interface to create, modify, and delete snickering concepts for the different UI variants.
    - Non-PMs can view, but not modify, the snickering concepts and UI variants, ensuring control over test setups.