

# Agile

- Theme: Get Gigglegit demo into a stable enough alpha to start onboarding some adventurous clients
- Epic: Onboarding experience
- User Story 1: As a vanilla git power-user that has never seen Gigglegit before, I want to be able to understand the interface without feeling annoyed.
  - Task: Write out a guide on interacting with Gigglegit
    - Ticket 1: Provide what features are different from Gigglegit and Git
    - Ticket 2: Implement "gigglegit help" to provide commands that includes meme responses
- User Story 2: As a team lead onboarding an experience Gigglegit user, I want to add them so that they provide support with no problem.
  - Task: Allow imports of Git repositories to Gigglegit.
    - Ticket 1: Implement a system where a user provides a URL of their git repository and imports it to Gigglegit. Such as using the command `"gigglegit import <repository_url>"`
    - Ticket 2: Create a script for users where it stores the commit and branches when importing to provide full context of changes
- User Story 3: As a new user of Gigglegit, I want to set up my personal profile to keep track to my commit history
  - Task: Allow users to configure name, email, and avatar in Gigglegit
    - Ticket 1: In a terminal, implement a command to add existing username by doing: `"gigglegit config user.name "Your Name"`
    - Ticket 2: For profile avatars, we can create a meme generator is if the user does not upload any images. Allow re-use as well if the user wants a different meme.

"As a user I want to be able to authenticate on a new machine"

It is too weak for a user story because it lacks clear purpose of what the user wants to do.

Authenticating on a new machine is more of a requirement/task than a user story. To fix it, it should have at least one reasoning to add more motivation before approaching the issue.

# Formal Requirements

- Goal: Create a user-friendly interface to allow PMs to keep track of the snickering concepts when syncing with the GigggleGit package
- Non-Goal: Generate new snickering concepts without user input
  - Non-functional requirement 1: Accessibility Controls
    - Functional requirements:
      - Only Project Managers should modify or add snickering concepts. They are able to maintain snickering concepts
      - Users can only see the concepts and would have to ask for permission to modify
  - Non-functional requirement 2: Randomizing User Studies
    - Functional requirements:
      - Randomly assign users to either control groups or variants in user studies
      - System will ensure that the user or participants cannot switch once assigned