Agile

- Theme: Get GiggleGit demo into a stable enough alpha to start onboarding some adventurous clients
- Epic: Onboarding experience

1. As a vanilla git power-user that has never seen GiggleGit before, I want to feel comfortable transitioning to a new version control tool.
2. As a team lead onboarding an experienced GiggleGit user, I want to quickly learn how to use all the tools to transition my whole team to GiggleGit. We want to make our project version control more engaging without losing efficiency.
3. As the project manager who is responsible for tracking the project history, I want to make sure that the version history in GiggleGit is auditable. I'm worried that demonstrating merges with memes can be confusing and sometimes unprofessional.
    a. Task: Make commit history clear
        i. Ticket 1: develop an optional verbal description to meme commit history
        ii. Ticket 2: establish a database of SWE-related memes that are relevant but not unprofessional

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

- This is NOT a user story because a user story has to be nonfunctional req. and also have a clear benefit. This does not outline a benefit and it is more of a functional requirement.

Formal Requirements

CodeChuckle is introducing a new diff tool: SnickerSync—why merge in silence when you can sync with a snicker? The PMs have a solid understanding of what it means to "sync with a snicker" and now they want to run some user studies. Your team has already created a vanilla interface capable of syncing with the base GiggleGit packages.

Goal: Gather qualitative and quantitative data from user studies to evaluate whether SnickerSync is a good tool that syncs with GiggleGit

Non-goal: The study only gathers data from frequent GiggleGit users

Non-functional requirements:

1. Random control group assignments
    a. Functional Requirements:
        i. Randomly assign study users using a coded pseudo-random selection process to determine the control group and all the corresponding variant groups
        ii. Track the results of the study using excel and pre-process the data using Pandas and matplotlib to identify patterns.
2. Evaluate PM happiness
    a. Functional Requirements
        i. Allow the owner of a SnickerSync repository to control the SnickerSync theme by adding an initial theme-selection option
        ii. Add repository settings to allow the PMs to enable who has access to creating merge, branch, or pull requests.