Agile:

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

1. Complete these user stories:
    - As a vanilla git power-user that has never seen GiggleGit before, I want to quickly understand how to use SnickerSync to sync my repositories.
        - i. Task: Create an onboarding tutorial for SnickerSync
        - ii. Ticket 1: Develop SnickerSync onboarding guide
            1. Create a simple step-by-step guide for experienced Git users that shows how to navigate GiggleGit's SnickerSync. Key features that differentiates SnickerSync from standard Git should be highlighted.
        - iii. Ticket 2: Implement in-app tutorial pop-ups for SnickerSync
            1. Develop tutorial pop-ups that walk users through SnickerSync's features as they use the interface for the first time.
    - As a team lead onboarding an experienced GiggleGit user, I want to ensure that my team members have the proper access and permissions to explore SnickerSync independently without my direct supervision.
        - i. Task: Develop user role management for team leads.
        - ii. Ticket 1: Implement role-based access control for team leads
            1. Allow team leads to set and modify user permissions within their repositories, including access to SnickerSync features.
        - iii. Ticket 2: Create a dashboard for team leads to monitor team members activity
            1. Build a dashboard for team leads where they can monitor team members' usage of SnickerSync. This dashboard would include the last time they synced, errors encountered, and how many merges were completed. This will help team leads keep track of their team's progress without needing to supervise them directly.
2. Create a third user story, one task for this user story, and two associated tickets.
    - Tasks should be a single phrase. (As should themes and epics. See those provided.)
    - User stories should be one to three sentences.
    - Tickets should have a title consisting of a single phrase and details that are long enough to sufficiently describe what needs to be done. You do not need to assign points to the tickets

As a PM conducting a user study on SnickerSync, I want to track how different users interact with the syncing process, so I can analyze feedback and improve the product.

Task: Implement user interaction logging for SnickerSync.

Ticket 1: Design user activity logging for SnickerSync

- Create a logging system that records user interaction during the syncing process including start and completion times, errors encountered, and meme interaction.

Ticket 2: Integrate user logging with data analysis tools

- Build interface to export user interaction logs and integrate this with the data analysis tools, enabling PMs to analyze user behavior within SnickerSync.

3. This is not a user story. Why not? What is it?
    - As a user I want to be able to authenticate on a new machine

This lacks an explanation of why the user wants to authenticate on a new machine and what benefit it would provide to them. A better user story would focus on the user's goals and needs, for example: "As a user I want to authenticate on a new machine so I can continue using GiggleGit without needing to repeat setup steps".

Formal Requirements:

- Goal: Create a smooth onboarding process for new users to easily interact with SnickerSync and understand its core functionalities focusing on sync version control with humorous feedback (i.e., syncing with a snicker).
- Non-Goal: Implementing meme generation for merges.

- Non-functional requirement 1: Security
    - Functional requirement 1:

Secure user authentication: Use OAuth for user authentication to ensure users securely access GiggleGit and their repositories.

- Functional requirement 2:

Data Encryption: Implement end-to-end encryption for all repository-related data during sync and merge operations to prevent unauthorized access.

- Non-functional requirement 2: User Study Management
    - Functional requirement 1:

Randomized User Assignment: Randomly assign users into control or experimental groups during onboarding to participate in SnickerSync user studies.

- Functional requirement 2:

Versioning and Concept Management for PMs: Allow PMs to define and maintain different SnickerSync interaction concepts (snicker types) and relate them to specific user study groups.