Agile

1. **As a vanilla git power-user that has never seen GiggleGit before, I want to…**

    learn how this version control system is different from that of Git so I can decide whether to use it my workflow.

    As a team lead onboarding an experienced GiggleGit user, I want to…

    ensure my team quickly adapts to GiggleGit so that we can maintain high productivity.

2. **As a software engineer using GiggleGit…**

    I want an easy way to track changes made by different team members so that I can quickly understand recent updates and resolve conflicts effectively.

    Task: Implement an intuitive change tracking system.

    Tickets:

    1. **Develop a commit visualization feature**
        - Provide a graphical representation of commit history and branch changes.
        - Allow users to filter changes by contributor or timeframe.
    2. **Enhance conflict resolution tools**
        - Highlight conflicting changes with contextual explanations.
        - Provide an interactive interface to compare and merge code efficiently.

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**

    A user story must include the who, what, and why to ensure clarity however this phrase lacks context on why authentication is necessary.

Formal Requirements

Objective:

Ensure GiggleGit allows users to easily understand and adopt its functionality providing a seamless onboarding process.

Non-Objective:

Support for alternative version control systems other than Git is out of scope.

Non-Functional Requirements

1. Data Security & Access Control

- Ensure that only authorized users can access SnickerSync's data.
- Maintain role-based access control for researchers and administrators.

2. Experimental Integrity

- Ensure that user study participants are randomly assigned and that their assignment remains consistent across sessions.
- Guarantee that experiment parameters are configurable without disrupting ongoing studies.

Functional Requirements

1. Secure Data Access (Supports Data Security & Access Control)

- Researchers must authenticate before accessing study data.
- Implement encrypted storage for sensitive user experiment data.

2. Role-Based Permissions (Supports Data Security & Access Control)

- Differentiate permissions for administrators, researchers, and general users.
- Restrict access to experimental results based on predefined roles.

3. User Assignment & Tracking (Supports Experimental Integrity)

- Users must be randomly assigned to control or experimental groups.
- Assignments must persist across multiple interactions with SnickerSync.

4. Configurable Experiment Settings (Supports Experimental Integrity)

- Allow PMs to modify snickering concepts without disrupting active experiments.

- Provide an interface for researchers to tweak experiment conditions dynamically.