AGILE:

Project: GiggleGit Onboarding Experience

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 to quickly learn GiggleGits differences from Git to get started.

Task: Introduce GiggleGit's core differences

- **Ticket 1:** Create onboarding flow
    - Design a flow highlighting unique GiggleGit features.
- **Ticket 2:** Develop interactive demo
    - Build a demo showing basic Git operations in GiggleGit.

User story 2:
As a team lead, I want to onboard my team to GiggleGit seamlessly.

Task: Facilitate team adoption

- **Ticket 1:** Implement repo compatibility checks
    - Add checks for Git repos to provide migration suggestions for the team's onboarding.
- **Ticket 2:** Write migration guide
    - Create a clear guide for teams transitioning to GiggleGit.

User story 3:
As a older user new to memes, I want a guide to help me understand the meme-based merge visualizations so I can follow the process without confusion.

Task: Provide a guide to understand meme-based merges

- **Ticket 1:** Create an onboarding tutorial for meme-based merges
    - Develop an interactive tutorial that explains what memes are and how it represents the merge process
- **Ticket 2:** Add contextual explanations during merges for the memes used
    - Implement explanations that help the user understand the meme's meaning at each stage of the merge process, making it so there is no confusion during the merge.

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

- This is not a user story because it is a requirement. It has no context for why they want to authenticate on a new machine. It also doesn't really have a goal or value at all.

FORMAL REQUIREMENTS

Goal and Non-Goal:

Goal: Enable seamless integration of SnickerSync with the base GiggleGit packages, allowing users to sync with a snicker during the merging process.

Non-goal: Implement an advanced AI-based meme generation system for SnickerSync.

Non-functional Requirement 1: Access Control

Functional Requirements:

Role-based Access Control:

- Implement role-based access for the SnickerSync settings, so that only PMs or administrators can modify settings.
- Roles should be assigned and access to certain tools and features should be restricted based on these roles

Audit Log:

- Implement an audit log that tracks changes made to snicker concepts
- This will allow administrators to see who changed things
- Ensure that all modifications are logged for transparency and accountability in everything.

Non-functional Requirement 2: User Study Randomization

Functional Requirements:

Random Assignment of Users:

- Develop a system that randomly assigns users to either the control group or variant groups in the study.
- Ensure that its unbiased, for transparent and fair user study/distribution of users in the study

Tracking Membership:

- Store and track each user's membership in their group within the study to make accurate analysis and reporting and to overall help ease of use
- Include identifiers in the data itself to differentiate between user groups (control group or variant groups) for easier data analysis as well