

Complete these user stories:

User Story 1: As a vanilla git power-user that has never seen GiggleGit before, I want to easily understand how the meme-based merge system relates to traditional git merges.

User Story 2: As a team lead onboarding an experienced GiggleGit user, I want to be able to create an example of what a GiggleGit merge is and how it differs from traditional Git so that anybody who I am onboarding can see through the example of how to use this software.

User Story 3: As a collaborative developer, I want to be able to be able to create rick-roll easter eggs in merge conflicts so I can surprise and amuse my teammates.

- Task: Add rick-roll easter egg feature into merge conflicts
- Tickets:
 - RICK ROLL TRIGGER: Find and create a consistent triggering mechanism depending on the specific conditions such as a specific naming convention/file type, etc.
 - RICK ROLL IMPLEMENTATION: Implement the actual rick-roll video of Rick Astley that replaces the meme based merge conflict that GiggleGit has when the trigger system occurs.

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

- This isn't a proper user story because there is no context, and it's not actually communicating what goal the user is trying to achieve, nor does it contain the value they are proposing with this story. It's primarily a technical need instead of a user need.

Goal: One goal of this project will be to install sounds into merges, providing auditory signals denoting the merge has been completed.

Non-Goal: To replace or eliminate the current meme merges visualization system.

NFR-1: The project manager should be able to update the sounds that are going into the merges without the need of developers.

- FR-1.1: When the Project Manager logs in, they should see a place too add sounds and change sounds
- FR-1.2: When the Developer Logs in, they should only be able to only see what sounds the Project Manager has updated, no area to update nor change.

NFR-2: Implement a randomized user assignment function so that individuals can randomly be tested with control vs non control.

- FR-2.1: Implement a randomizer, that based on all the users within SnickerSync, will select X amount of random users given how many individuals would like to be selected out of the general user base.
- FR-2.2: Implement a randomizer selector that will choose control group out of the selected testers within 2.1