Agile

User Story 1: As a vanilla git power-user that has never seen GiggleGit before, I want to quickly understand how GiggleGit's works so I can use it efficiently for my version control management.

User Story 2: As a team lead onboarding an experienced GiggleGit user, I want to be able to guide my team to transition out of Github to GiggleGit with minimal confusion and using my already established knowledge of Github.

User Story 3: As a rising developer with no experience with Github and starting off with GiggleGit, I want to be able to learn about version control and how it works on this platform through guided tutorials. I also want to feel confident that I am using GiggleGit right by completing the guided tutorial with a test case.

Task: Create onboarding tutorials for rising developers new to version control.

Ticket 1: Implement a walkthrough to teach the basics of GiggleGit

Develop a series of short video tutorials. When a snippet of the video tutorial is completed have arrows pointing to each step the users have to click on to test the contents of the tutorial on their own. Once the user completes one step the arrow will move so the user can follow the next steps.

Ticket 2: Develop a Virtual Assistant for Tutorial Purposes

Create a virtual assistant that can answer any additional questions the user may have about using GiggleGit. The virtual assistant can only be used during the tutorial.

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

This is not a user story. User stories should not only contain technical wording such as authenticate.

Solution: As a team I want to be able to authenticate on a new machine so I can efficiently access my team's shared repository.

Formal Requirements

Goal: Provide users in the user study with an interface to test the SnickerSync tool and obtain their feedback.

Non-goal: Developing performance metrics to identify areas where the SnickerSync tool may not work effectively and where improvement may be needed.

Non-functional requirement 1: The interface should be intuitive and the snicker should keep the users engaged and entertained.

Functional Requirement 1: The snickering sound effect should be distinct and should be paired with a fun visual pop-up when merging.

Functional Requirement 2: The interface must be easy to navigate. There must be interactive steps to guide users through the syncing process.

Non-functional requirement 2: The interface should have a level of personification. The users should be able to adjust the snicker sound effect and the colors and theme of the interface.

Functional Requirement 1: There must be a customization button that the user can click on to toggle between different snicker sound effects and volume.

Functional Requirement 2: There must be an option to adjust the color and theme of the interface during the syncing process. Users should be able to preview the changes they make before finalizing it.