

Agile

User Story 1: As a vanilla Git power-user that has never seen GiggleGit before, I want to be able to easily pick up basic Git actions like clone, commit, push, and pull on GiggleGit in order to transition to GiggleGit as my primary version control system.

User Story 2: As a team lead onboarding an experienced GiggleGit user, I want to be able to instruct new team members to transition from Github to GiggleGit so everyone can use a universal version control system.

User Story 3: As a new client, I want to be able to see how Git operations work so I can understand the advantages of using GiggleGit.

Task: Demonstrate basic Git workflows for new clients

Ticket 1: Make a tutorial video demonstrating Git workflows on GiggleGit

Ticket 2: Make a final short assessment so new clients can apply what they learned

The reason “As a user I want to be able to authenticate on a new machine” is not a user story is because it doesn’t follow to the structure of “I want to...in order to...”, it just mentions what it wants to achieve by using a very technical term authenticate, but doesn’t follow up on why.

Project Requirements

Goal: Build on the vanilla interface to allow users to merge with a snicker with the “sync with a snicker” tool.

Non-goal: Visually and audibly add a snicker to every merge

Non-functional Requirement 2: Security

Functional requirements:

- Implement permissions to control access based on assigned roles
- Use OAuth and multi-factor authentication to ensure that only authorized users can access the platform with their specific permissions.

Non-functional Requirement 1: Reliability

Functional requirements:

- Implement retry mechanisms in the case that there are network interruptions or API failures during syncing
- Track and log all sync activities and errors for troubleshooting