

Giggle Git:

Agile

- 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, I want to understand how its meme based merge process differs from standard git and get a quick hands on experience
 - Task: Develop an onboarding guide/tutorial for vanilla git users
 - Ticket 1: Create a written overview of the version control system
 - Highlight GiggleGit's features and a quick overview of how to use them
 - Ticket 2: Create a starting guide
 - Create a starting guide that includes Giggle Git's key commands and workflow that differs from standard Git to get user up to speed with the transition
- User story 2: As a team lead onboarding an experienced GiggleGit user, I want to be able to keep track of the merge activities across team members to ensure collaboration is flowing smoothly
 - Task: Create a dashboard to monitor any existing merge conflict
 - Ticket 1: Design the dashboard
 - Design a dashboard that allows team leads to monitor all ongoing merges in real time and include a team progress overview
 - Ticket 2: Implement conflict alert system
 - Implement an alert system that notifies the team lead if any merge conflict occurs in real time so that the team lead can step in and help to resolve the issues as soon as possible
- User story 3: As a product manager that oversees the development of a GiggleGit project, I want to be able to get a status overview on the ongoing projects and keep track of the project milestones to ensure it's align with the delivery timeline
 - Task: Create a progress tracking branch
 - Ticket 1: Create a tracking dashboard
 - Create a tracking dashboard that gives the project manager an overview of all the progresses and project status
 - Ticket 2: Implement a progress alerting system
 - Implement a progress alerting system so that the product manager will get notified whenever a milestone is made or when the progress is on the edge of falling behind the timeline

SnickerSync:

Project Requirements

- Goal: Create a difference tool that provides the users with a new syncing experience with snickers in their version control system Gigggle Git
- Non-Goal: Add advanced Git features to SnickerSync to enhance the syncing with snicker experience
- Non-functional requirements:
 - Non-functional requirement 1: Access
 - Only project managers are able to modify the snickers. Users and developers should only be able to use the snickers during syncing
 - The interface should adjust accordingly based on the user's role, displaying different options for the users
 - Non-functional requirement 2: Randomization
 - Users in the study should be randomly assigned to either the control group or the experimental group. The control group uses the standard merge tool whereas the experimental group uses SnickerSync
 - The system will track the each participant based on their syncing performance and user feedback for the analysis stage