

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 that has never seen GiggleGit before, I want to learn about its unique features through a familiar interface that will allow me to incorporate it into my workflow without requiring extensive documentation reading.

- Task: Create a command-line interface that is familiar to Git users.
 - Ticket 1: Plan and structure the interface
 - Outline the structure of the interface, specifying key options and layouts similar to Git's interface, ensuring a logical flow to minimize the learning curve for users.
 - Ticket 2: Implement key commands that are similar to those in Git
 - Develop the core GiggleGit commands so that they closely mirror their Git counterparts, ensuring they function similarly while still incorporating GiggleGit's meme-based features. Include clear help messages for each command, highlighting differences from Git behavior.

User Story 2: As a team lead onboarding an experienced GiggleGit user, I want to provide clear, concise onboarding documentation, so my team can start using GiggleGit efficiently.

- Task: Write an onboarding guide for experienced users.
 - Ticket 1: Develop advanced feature and workflow guide
 - Create a comprehensive guide focusing on GiggleGit's unique features, advanced workflows, and best practices. Include detailed explanations of meme-based merge strategies, team-specific meme customization, and configurations for team sizes and project types.
 - Ticket 2: Create troubleshooting and FAQ section
 - Add a section that covers common issues and questions that the user might encounter, such as troubleshooting merge conflicts or integrating GiggleGit with other tools used by the team.

User Story 3: As a new developer using GiggleGit, I want to view example meme-based resolutions for merge conflicts so I can learn how GiggleGit handles merges in practice.

- Task: Create merge conflict tutorials that showcase GiggleGit's merge resolution system.
 - Ticket 1: Set up a repository with complex and varied merge conflicts
 - Build a repository introducing different types of merge conflicts that simulate real-world conflicts that may occur in practice.
 - Ticket 2: Design meme-based resolution examples for each conflict type
 - Create a series of meme-based merge conflict resolutions that showcase how different types of conflicts are resolved using GiggleGit's meme system, with explanations for how each meme resolution works.