

Agile:

1. Complete these user stories:

- As an amateur git power-user that has never seen GiggieGit before, I want to...

As an amateur git power-user that has never seen GiggieGit before, I want to be able to intuitively use its core version control features without needing to read a great deal of documentation.

- As a team lead onboarding an experienced GiggieGit user, I want to...

As a team lead onboarding an experienced GiggieGit user, I want to be able to track my team's merges and understand their context through the meme history.

2. Create a third user story, one task for this user story, and two associated tickets.

User Story:

As a first-time user, I want to be welcomed with a fun, interactive tutorial on how GiggieGit works so I can quickly understand its unique features.

Task: Create an interactive onboarding tutorial.

Ticket 1: "Design a guided onboarding experience"

- Design a playful, interactive tutorial that walks new users through a sample git commit, merge, and meme-based conflict resolution.

Ticket 2: "Test and refine onboarding flow"

- Perform usability tests with first-time users and refine the onboarding flow based on feedback to ensure ease of understanding.

3. This is not a user story. Why not? What is it?

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

This is not a user story because it focuses only on a technical feature rather than a user's need or outcome, therefore, it's more of a functional requirement.

Formal Requirements:

1. List one goal and one non-goal

Goal: Make sure that users can effectively sync changes using SnickerSync with a clear understanding of the "sync with a snicker" concept.

Non-goals: Don't introduce any new git commands or syntax for basic version control operations outside of syncing with SnickerSync.

2. Create two non-functional requirements. Here are suggestions of things to think about:
 - Who has access to what
 - PMs need to be able to maintain the different snickering concepts
 - A user study needs to have random assignments of users between control groups and variants
 1. Only PMs should have the ability to modify and manage different snickering concepts within the system.
 2. The user study system must randomly assign participants to control and variant groups to make sure for unbiased testing.
3. For each non-functional requirement, create two functional requirements (for a grand total of four functional requirements).
 1.
 - PMs must have a separate interface where they can manage snickering concepts without impacting users' regular sync operations.
 - Make sure that any changes made to snickering concepts by PMs are logged for future auditing.
 2.
 - Write an algorithm that randomly assigns users to either the control group or the variant group during the study session.
 - Track and store group assignments securely in the database to make sure no duplication or bias in future assignments.