

# Agile

1.

## User story 1:

As a vanilla git power-user that has never seen GiggleGit before, I want to understand how GiggleGit can be integrated into my existing workflow, so that I can adopt it without losing a lot of productivity.

## User story 2:

As a team lead onboarding an experienced GiggleGit user, I want to make sure that it can be set up easily, so we can quickly collaborate without losing a lot of time on collaboration.

1.

## User story 3:

*As a new client testing out Gigglegit, I want to have a clear and easy onboarding process so that I can start using it without confusion.*

## Task:

*Simplify the onboarding process for new clients*

## Tickets:

*Ticket 1: Create a step by step onboarding guide.*

Develop an easy, step by step guide for new clients, covering everything essential that you need before you start using it. Ensure that the guide includes screenshots, and video walkthroughs with enough information for it to not be confusing.

*Ticket 2: Create a wizard in the app that helps with onboarding*

Design and implement an interactive onboarding wizard that actively guides new clients through the setup process and first few major actions. Ensure that they are comfortable with atleast the basic features of GiggleGit.

2. This sentence also describes what the user wants, not why and why it matters to them. There is also no mention of who the user is, and what kind of target audience they are. To improve on this, we can add context as to who the user is, what they want and why they want it.

# Formal Requirements

## 1. Goal :

Ensure that SnickerSync offers a dynamic and fun experience for users while maintaining functional version control capabilities.

## Non-goal:

Do not focus on developing SnickerSync for large scale enterprise use; focus on individual user experience.

## 2.

### Non functional requirement 1:

SnickerSync should be optimized for small to medium sized repositories, but not for handling repositories over a certain limit.

### Functional requirement 1:

Display a warning message for users when they attempt to sync repositories over a certain size, giving a warning explaining this.

### Functional requirement 2:

Increase performance monitoring for repositories under 1GB to ensure fast and smooth diffing and syncing.

### Non functional requirement 2:

SnickerSync should maintain a minimal memory footprint and CPU usages, making sure that it can run efficiently on low resource systems.

### Functional requirement 1:

Optimize memory management so that only the necessary data from the current sync or diff is loaded into memory, while releasing unused data.

**Functional requirement 2:**

Implement resource tracking, ensuring that the CPU and memory consumption does not exceed a predefined limit, triggering alerts and changing if this happens.