## Agile

1. **Complete these user stories:**

As a vanilla git power-user that has never seen GiggleGit before, I want GiggleGit to as much functionality as vanilla git while making certain operations simplified to justify switching over from vanilla git

As a team lead onboarding an experienced GiggleGit user, I want to access to clear and understandable documentation to understand how experienced users use GiggleGit to better onboard the new member

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

Story: As a project manager using GiggleGit for the first time, I want to be able to access a dashboard that provides me with key metrics like onboarding completion rate so I can monitor how my team is adapting to GiggleGit

Task: Develop a team performance dashboard that displays key metrics for onboarding

Ticket 1: Design dashboard frontend

* Design and create an appropriate dashboard interface that shows data on onboarding completion rate.

Ticket 2: Implement dashboard backend

* Implement backend data collection and API integration for onboarding metrics. Ensure the data is updated in real-time and create secure APIs to feed the onboarding metrics into the dashboard interface

1. **This is not a user story: As a user I want to be able to authenticate on a new machine. Why not? What is it?**

It’s not a user story because it lacks a clear benefit for the user in question. It is more similar to a task because it is describing a function that needs to be implemented.

## Formal Requirements

1. **List one goal and one non-goal**

Goal: add an feedback rating modal that pops up after a sync that asks the user to rate their satisfaction on every sync on a scale of 1 (terrible) to 5 (great).

Non-goal: the study will not focus on how SnickerSync performs at scale with high user counts, heavy loads, large repos etc.

1. **Create two non-functional requirements**

Non-functional requirement 1: only authorized personnel, such as PMs and developers, should be able to access and modify SnickerSync’s snickering concepts.

Non-functional requirement 2: PMs must be able to easily update the snickering concepts in SnickerSync without needing assistance from developers or IT support.

1. **For each non-functional requirement, create two functional requirements**

For non-functional requirement 1:

1. The system must require users to authenticate using secure login credentials and verify their roles before granting access to the snickering concept management interface.
2. The system must allow administrators to assign roles (such as PM or developer) to users, and ensure that only users with these specific roles can perform actions like creating, editing, or deleting snickering concepts.

For non-functional requirement 2:

1. The system should have an intuitive interface for PMs to directly push updates for snicker concepts in real time
2. The interface must include built-in help features, such as tooltips and step-by-step guides, specifically tailored for PMs, enabling them to troubleshoot common issues and make updates confidently without relying on developers or IT support.

***dependencies on next page***

## Dependencies





