

Agile:

1.
 - a. As a vanilla git power-user that has never seen GiggieGit before, I want to be able to easily switch between GiggieGit and vanilla git in order to use commands and features that I am more familiar with in case of an emergency.
 - b. As a team lead onboarding an experienced GiggieGit user, I want to have the new features highlighted for experienced users in order to allow them to save time that would have used going through unnecessary tutorials.
2. Theme: Get GiggieGit demo into a stable enough alpha to start onboarding some adventurous clients
Epic: Onboarding experience
User story: As a new user that has never used Git I want to have clear description of all features and commands in order to make learning how to use the software faster
Task: Documentation
Ticket 1: Tutorial for users
 - Create a tutorial for new users so they can see what tools are available for them. You can have a little aid showcase the purpose of each button/feature.Ticket 2: Document of commands and shortcuts
 - Have a document with all commands and shortcuts listed that can be accessed by the user. Each command/shortcut will be accompanied by a description of what it does and when to use it.
3. This is not a user story because it does not have a benefit and lacks a specified actor listed, it simply tells us what features they want but not the benefit of it or what kind of user they are. Something like: As a user with several devices, I want to be able to authenticate on a new machine so I can have access across different devices would be a user story. This would be a non-functional requirement.

Traditional:

CodeChuckle is introducing a new diff tool: SnickerSync—why merge in silence when you can sync with a snicker? The PMs have a solid understanding of what it means to "sync with a snicker" and now they want to run some user studies. Your team has already created a vanilla interface capable of syncing with the base GiggleGit packages.

- Goal: Evaluate the user satisfaction using SnickerSync as well as how effectively it allows the user to perform merges.
- Non-Goal: Measure the retention rate as well as the satisfaction rate of the users
- Non-functional requirement 1: Access Management
 - Functional requirements:
 - Give the PMs access to the panel (settings) so that they can properly update and maintain the different memes and snickers
 - Ensure that the users are unable to access beyond what's necessary
- Non-functional requirement 2: Randomized Experiment
 - Functional requirements:
 - Randomly assign each user to either a control group or a testing group
 - Keep a log (track) of each user's interaction with the interface making sure to keep track of the group they are in