

Agile**Questions:**

1. Complete these user stories:
 - As a vanilla git power-user that has never seen GiggleGit before, I want to...
 - As a team lead onboarding an experienced GiggleGit user, I want to...
2. Create a third user story, one task for this user story, and two associated tickets.
 - Tasks should be a single phrase. (As should themes and epics. See those provided.)
 - User stories should be one to three sentences.
 - Tickets should have a title consisting of a single phrase and details that are long enough to sufficiently describe what needs to be done. You do not need to assign points to the tickets
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

Answers:

1.
 - a. As a vanilla git power-user that has never seen GiggleGit before, I want to be able to easily access the features of the program and be provided resources explaining how to utilize this control system.
 - b. As a team lead onboarding an experienced GiggleGit user, I want to be able to utilize more advanced features of the program and be able to further improve my team's ability to make use of GiggleGit.
2.
 - a. User Story: As somebody looking to use GiggleGit for my team, I want to know how this product will impact the ability for people to collaborate and work on a project with each other
 - b. Ticket 1: Comparison of Using and Not Using GiggleGit
 - i. Details: As GiggleGit is supposed to aid in merges through the use of memes, it must be noted how this system actually affects team members' ability to work on a project together. Productivity should be compared when using this system and when it is not in use.
 - c. Ticker 2: Features of GiggleGit
 - i. Details: The features of GiggleGit should be examined to see how they are actually aiding team members working together on a project. Looking at features specific to GiggleGit through the lens of collaboration is important in determining the effectiveness of the program.
3. This is not a user story because of the lack of specification. A user story usually includes who the user is, what their need is, and why they have that need. In this instance, there is no relationship between who they are and what this program is. It would be closer to a requirement/what a user wants as they do not state why this would be beneficial to them.

Formal Requirements:

Questions:

1. List one goal and one non-goal
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
3. For each non-functional requirement, create two functional requirements (for a grand total of four functional requirements).

Answers:

1. One goal for the user studies would be to see how this new system of adding the “snicker” feature to merges benefits teams in any way, whether it be performance-wise or user morale. One non-goal here could be deciding if there should be different “snickers” for different situations as a possible added feature.
2. Two Non-Functional Requirements
 - a. Reliability
 - b. Performance (Response Time)
3. Functional Requirements
 - i. Reliability
 1. The system should be able to effectively and accurately play a “snicker” when a merge/sync occurs.
 2. The system should be able to play the correct “snicker” concept depending on the decisions of the PM.
 - ii. Performance (Response Time)
 1. The system should be able play the “snicker” feature within a set amount of time of the sync occurring.
 2. The system should report an error if the “snicker” does not play within the allotted time frame.