Agile

CodeChuckle is a startup whose product is GiggleGit, a version control system “where merges are managed by memes." (It saddens me to say that this was a joke written by ChatGPT for 131)

You have just been hired as employee number n for some small number n. They have the dev chops to make a demo, but you are their first serious developer.

Here is a theme and an epic:

- Theme: Get GiggleGit demo into a stable enough alpha to start onboarding some adventurous clients
- Epic: Onboarding experience

Complete the following tasks:

- Complete these user stories:
    - Story 1: As a vanilla git power-user that has never seen GiggleGit before how can I ensure that uploading my projects is safe and reliable
    - Story 2: As a team lead onboarding an experienced GiggleGit user, I want to make sure that the process for users is as easy as possible
- Create a third user story, one task for this user story, and two associated tickets.
    - User story 3: as a new user to giggle git, how can I make sure that my account is safe from scammer/hackers
        - I want to ensure the safety and security of my project.
            - Task: add functionality for proper safety measures
                - Ticket 1: Implement 2 factor authentication
                    - Implement the correct 2FA for users to use
                - Ticket 2: Make sure that people know about 2FA
                    - We need to make sure that people know about the 2FA system so that we can ensure that they don't get their account stolen
- This is not a user story. Why not? What is it?
    - This is not a user story because it says “as a user I want to…” this makes it a non-functional requirement.

Formal Requirements

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.

Complete the following tasks:

1. List one goal and one non-goal

Goal: Allow users to use SnickerSync and some to use normal sync and collect feedback

Non-Goal: Allow users to have the best diff tool

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).

    Non-functional requirement 1: Enjoyability

    - Functional requirements:
        i. Give users survey after using it and collect feedback
        ii. Measure data from each tool to see which is used more

    Non-functional requirement 2: ease of use

    iii. Functional requirements:
        1. Set a variable to see how long users take to finish using tool
        2. Visualize this data for PM’s