Software Engineering HW 2

Agile —---

1. Complete the user stories.
    a. As a vanilla git power-user that has never seen GiggleGit before, I want to be able to utilize the basic vanilla git commands so I don't have to learn something new and can start using GiggleGit quickly.
    b. As a team lead onboarding an experienced GiggleGit user, I want to ensure the new team member's proficiency in advanced GiggleGit and be able to teach this new member without having to go through basic training.
2. Own user story.
    a. As a team lead with multiple, large, ongoing projects, I want to be able to track the progress of each member in these large groups, as well as the progress of their projects as a whole in order to track the individual contributions in a timely fashion.
        i. Create/implement a report that tracks user contribution
        ii. Ticket 1
            1. Create a user activity report generator.
            2. Implement a feature in GiggleGit that tracks and can compile the progress of each team member. This will involve a history of commits, merges, and errors for the team member. Ensure that each report is stored and constantly being updated for real-time results.
        iii. Ticket 2
            1. Implement a group-wide report.
            2. Develop a feature that allows team leads to compile all individual reports of team members in the same group as well as overarching group statistics. This could involve total commits, merges, and any errors and how they were resolved.
3. *"As a user I want to be able to authenticate on a new machine."*
    a. No, user stories are supposed to include why the user will benefit from this feature. It is very vague, so there isn't much the designer can build off of in terms of what this feature should prioritize.
    b. This is more of a requirement/feature request whereas a user request would include the 'why'.

Formal Requirements —----------------------------------

1. Goals
    a. Create different versions of SnickerSync in order to grasp overall feedback on the tool and test different features simultaneously.
    b. Focus on gathering initial feedback rather than having a scientifically correct study, since it is the first round of testing.
2. Non-functional requirements
    a. Allow for support of multiple SnickerSync variants within the study for the administrator to deploy.
        i. Develop an interface that randomly assigns participants to the chosen set of variants/control group.
        ii. Ensure that multiple versions of SnickerSync can be run simultaneously so that the study of groups can occur at the same time.
    b. Have a model for structured feedback that can be used on all variants to allow for easy tracking of data.
        i. Implement a sort of feedback survey that asks participants concrete questions about their experience. This should be automatically deployed after the completion of the study.
        ii. Develop a database that tracks participant responses as well as which group they were in. This should also include basic analytics that can be used by the administrators to assess the variants.