User Story (Agile):

1. **User Story 1:** As a vanilla git power-user who has never seen GiggleGit before, I want to understand how GiggleGit integrates with traditional git workflows so that I can quickly understand without having to learn everything.
2. **User Story 2:** As a team lead onboarding an experienced GiggleGit user, I want to have a smooth transition with minimal setup steps and access to team collaboration tools so that my team can be productive quickly.
3. **User Story 3:** As a developer managing multiple machines, I want my GiggleGit settings to be easily transitional across devices so that I can work from any environment without needing to reconfigure everything.
    a. **Task**: Implement settings synchronization across devices
    b. **Ticket 1:** Implement a sync feature for user settings. Develop a mechanism to store user-specific settings (e.g., themes, preferences) in a secure cloud location, allowing automatic syncing between devices.
    c. **Ticket 2:** Ensure secure authentication on multiple devices. Implement the ability for users to securely authenticate on multiple machines without needing to log in repeatedly.
4. **Non-User story:** As a user I want to be able to authenticate on a new machine
    a. This is not a user story because it is phrased as a feature request or functionality description. It is lacking a clear user context or motivation.

Formal Requirements:

1. **Goal:** Ensure that the SnickerSync tool provides users with an entertaining and efficient experience when resolving merge conflicts, integrating into existing GiggleGit workflows.
2. **Non-Goal:** Avoid adding advanced merge conflict resolution algorithms at this stage. Focus only on improving the user interface experience and interaction model.
3. **Non-Functional Requirement 1:** Access Control — Only users with administrative privileges, such as PMs, should have access to maintain or update the different snickering concepts associated with SnickerSync.
    a. **Functional Requirement 1:** Create a specific role for PMs that allows access to the SnickerSync configuration dashboard where they can update or maintain snickering concepts.
    b. **Functional Requirement 2:** Implement a role-based access control mechanism that restricts users without the PM role from accessing SnickerSync maintenance features.
4. **Non-Functional Requirement 2:** User Study Randomization — The user study should support random assignment of participants between a control group using standard merge tools and a variant group using SnickerSync.

a. **Functional Requirement 1:** Develop a system that automatically assigns users to either the control group or the SnickerSync variant group upon initiation of the user study.
b. **Functional Requirement 2:** Implement functionality to log and track which group a user is assigned to, ensuring data from the control and variant groups can be separated for analysis.