Agile

- User story 1: As a vanilla git power-user that has never seen GiggleGit before, I want to quickly understand the interface so that I can incorporate the service into my workflow.
- User story 2: As a team lead onboarding an experienced GiggleGit user, I want to be able to grant access permissions so our new team member can work begin to work with us.

User story 3: As a new team member in a group project, I want to view my group's work history to better understand the architecture and functionality of my group’s project.

- Task: display versions of codebase
  - Ticket 1: implement a clickable list of all project versions
    - Need to be able to see structure and contents of the project for each version upon click
  - Ticket 2: store version history info in MongoDB database
    - Design and create database to store user project info such as code contents, commit history, version numbers, etc

This [As a user I want to be able to authenticate on a new machine] is not a user story because it describes a feature rather than a user benefit. A user story would describe why a user would want this feature. This is a system requirement, not a user story.

Formal Requirements

- Goal: enhance the user interface so that users can test syncing with a snicker in order to collect feedback during user studies
- Non-goal: implement a questionnaire after users are finished testing to get direct feedback
- Non-functional requirement 1: access control
  - Functional requirements:
    - Have auth set up to verify users before allowing them to access SnickerSync interface
    - Implement different access controls so that PMs and devs have different access levels
- Non-functional requirement 2: user study
  - Functional requirements:
    - Assign users randomly to control or experimental groups when running user studies
    - Build a dashboard that PMs can monitor and track metrics in real time