

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:

1. Complete these user stories:
 - As a vanilla git power-user that has never seen GiggleGit before, I want to know my projects using GiggleGit are secure.
 - As a team lead onboarding an experienced GiggleGit user, I want to make sure my projects can transfer from other version control systems.
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

New user story:

As a user transferring over to GiggleGit I want to make sure color themes and design are customizable

Task: GiggleGit user design experience

- Ticket 1: Design Tab
 - Make a design tab that allows users to control the colors on their GiggleGit account
 - Ticket 2: ThemeMasterSoftware API
 - Integrate the ThemeMasterSoftware API so that users can use their favorite themes from the design tab
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

This is not a user story because if it starts with “as a user I want to” it is 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: Create designation for some users to use SnickerSync and some to use NormalSync

Non Goal: Have the best diff tool be available for the users

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: User designation

Functional requirements: assign users one tool they use randomly, link that tool to the account so when they login they always only have the one tool available

Non functional requirement 2: Get user feedback

Functional requirements: measure how many times user has used that sync tool, when user has used tool 100 times ask them for feedback about the sync tool and store that for PMs