

## **GiggleGit**

### **AGILE:**

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

Epic: Onboarding Experience

User Story 1:

As a vanilla Git power-user that has never seen GiggleGit before, I want to understand how memes are used during merges without interrupting my existing workflow on Git, so I can integrate the new product into my work flawlessly without setbacks on my productivity.

User Story 2:

As a team lead onboarding an experienced GiggleGit user, I want to create guidelines for the entire time for how to use memes for merge conflicts so that there can be consistency in which memes are used for what purpose and where they are used in the development process.

User Story 3:

As a developer that works on large projects regularly, I want an automated system to deal with merge conflicts without my interference so that I can focus on my more complex tasks and priorities instead of manually fixing conflicts.

Task: Automatic meme selection for conflicts

Ticket 1: Develop a process for automated meme selection for merge conflicts

Details: Create an algorithm that is able to identify merge conflicts and selects a meme best suited to solve that merge conflict from a pre-existing dictionary. This can be customized to the users preference through a settings menu

Ticket 2: Logs for conflicts when merge conflicts are resolved using the automated system

Details: Maintain a log for the automatic meme selection so that users can go back and update/change if needed. This should include details such as the date, time, type of conflict, and the users involved. Logs should be easily accessible through the UI to allow for review.

As a user I want to be able to authenticate on a new machine:

- This is not a user story because there is no context on who the user is. A user story needs to state who its for and how and why they need a specific function out of the product. This is more akin to a functional requirement

## **FORMAL REQUIREMENT**

Goal: To improve user engagement and productivity through humor using SnickerSync, a diff tool that adds humor to merge processes.

Non-Goal: Have all users use SnickerSync for merges. Users should be able to use whatever diff tools they desire without being forced to be locked into a system

### **Non Functional requirement 1: PM Access for maintaining humor concepts**

- Functional requirement 1: PMs should have a secure way to access the control system that allows them to manage the themes. This should be role based or clearance based so that only authorized personnel can make changes.
- Functional Requirement 2: Should have an interface for PMs to make updates and view, edit, or delete concepts without interrupting workflow of other users

### **Non Functional Requirement 2: Performance and reliability**

- Functional requirement 1: Merges should be made with minimal latency to make sure that it does not interrupt workflow of users
- Functional requirement 2: It should be able to handle a large number of users, merges, and repositories during times with high traffic so that crashes or performance issues do not occur if too many people log on at once and create merges.