

Ryan Rodriguez

Professor Golbus

CS 411

2/13/25

## **HW1 Create Requirements CS411**

### **Agile Requirements: User Stories**

#### **User Story 1**

- As a vanilla git power-user that has never seen GiggleGit before, I want to seamlessly transition my existing git workflows into GiggleGit while learning the meme-based merge system.

#### **User Story 2**

- As a team lead onboarding an experienced GiggleGit user, I want to efficiently delegate merge permissions and meme templates to maintain consistent merge standards across the team.

#### **User Story 3**

- As a developer working in a regulated industry, I want to ensure my meme-based merges comply with corporate communication policies while maintaining the fun aspect of GiggleGit.

### **Task for User Story 3:**

- Implement corporate meme filtering system

### **Tickets:**

- (1) Title: Create meme content filter API
  - Details:
    - Develop an API that can scan memes against a configurable set of corporate guidelines
    - Image recognition capability
    - Text extraction and analysis
    - Configurable filtering rules
    - Rejection/approval workflow
  - Title: Design meme template approval workflow
    - Details:
      - Create an administrative interface for managing pre-approved meme templates

- Allows bulk template upload
- Provides preview functionality
- Supports template categorization
- Includes audit logging of template approvals

**Note on Authentication Request:**

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

This is not a user story because it lacks the context of who the user is and why they need this functionality. A proper user story follows the format "As a "type of user", I want "goal" so that "benefit". Currently, it is just a feature request without context or purpose.

**Formal Requirements:**

**Goal:**

- Create an engaging and measurable user experience for meme-based code merging that increases team collaboration and satisfaction.

**Non-Goal:**

- Create a comprehensive meme creation tool or replace existing version control functionality.

**Non-functional Requirement 1: Experimental Validity:**

- Functional Requirement 1.1:
  - System must randomly assign users to control or experimental groups with equal probability
- Functional Requirement 1.2:
  - System must track and record all merge actions and timing data without revealing group assignment to users

**Non-functional Requirement 2: Accessibility:**

- Functional Requirement 2.1:
  - All meme-based interactions must have text-based alternatives for screen readers
- Functional Requirement 2.2:
  - System must support keyboard-only navigation for all merge operations