Process

This document tries to summarize the process that the team follows to develop Feed Amalgamator.

Project Management Framework:

Bugs Bunny adheres to the Agile methodology in the development of Feed Amalgamator, employing the Kanban Framework as our chosen approach for the development process. This decision stems from our recognition that our team lacks extensive experience in the project domain, which could result in unforeseen tasks or stories arising. Given that adding stories during a sprint is generally discouraged (Stack Exchange - Adding Stories to Sprint Backlog, Scrum.org Forum - Adding New Story in Current Sprint), we decided to check for an agile framework without the concept of sprints and this is when we found out about Kanban.

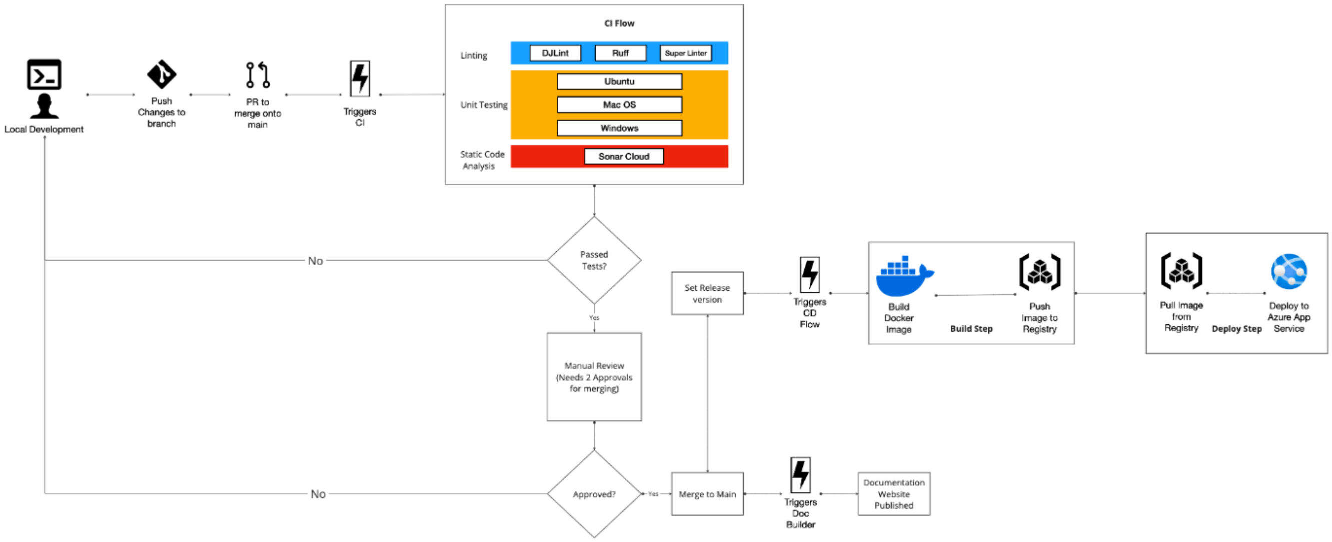
Opting for the Kanban approach resonated with us as it encourages a more adaptive "go with the flow" mindset. This methodology has proven effective in constraining the number of tasks concurrently undertaken. Despite embracing Kanban, we have integrated certain principles from Scrum into our workflow, including regular interval meetings, periodic retrospective sessions, and the continual addition of tasks and stories to the backlog.

Meetings:

Typically, we convene for team meetings every Tuesday and Thursday, during which we review our project's advancements and address any challenges we've encountered. These sessions are well-structured, with clearly defined agendas and time constraints to ensure efficiency. Embracing a hybrid approach, team members have the flexibility to participate virtually if attending in person is not feasible.

Additionally, we make it a point to gather before impending deadlines or presentations to synchronize everyone's understanding and align our efforts. This proactive approach ensures that our team remains cohesive and well-prepared for upcoming project milestones.

Way of Working:



Issue Assignment and Branching

In our software development workflow, we have utilized GitHub's Issue Assignment feature for the purpose of task distribution among team members. This functionality has facilitated a straightforward and organized approach to managing issues, bugs, or new features. By assigning specific tasks to individual team members through GitHub's interface, we have streamlined the process of task allocation, fostering a clear sense of responsibility for each issue.

In conjunction with the Issue Assignment feature, we have employed a branching strategy to address and resolve each identified issue. This involves creating separate branches within our version control system for each specific task or problem. Branching allows our development team to work on distinct issues simultaneously without interfering with the main codebase. This approach not only ensures a focused and isolated development environment for each task but also facilitates efficient collaboration, as team members can work independently on their assigned branches. As a result, the combination of GitHub's Issue Assignment and branch-based development has proven to be an effective methodology for managing and resolving issues within our software projects.

Pull Requests and CI - CD Flow

We employ pull requests as our method for merging changes into the main branch, and our current protocol mandates two approvals along with the successful completion of CI checks before any merges are executed.

Our CI pipeline encompasses three key processes: Linting, Unit Testing, and Static Code Analysis. Linting involves Ruff linting for Python, Djlint for HTML templates, and Super Linter, which encompasses multiple linters. Unit Testing is conducted on three major platforms - Windows, macOS, and Ubuntu - ensuring the application's behavior aligns with our expectations to a certain extent. Additionally, we conduct static code analysis using SonarCloud.

Following the successful merging of changes into the main branch, an automated process generates documentation and renders a static documentation website. Subsequently, we have the option to select a release version, triggering the automatic deployment of the website on Azure App Services.