FETCH BOT DEVELOPMENT STANDARDS

AI based robot with image processing.

https://cshadd.github.io/fetch-bot/

Contents

1. Overview

Clause that defines the overview of the standards of development for Fetch Bot.

1.1. Scope

Development process for Fetch Bot.

1.2. Purpose

The point of this document is for the developers of this project to learn the model of our project management and the tools we use. Furthermore, it is important for our stakeholders to understand our process cycle.

2. Project Management

IEEE Standards IEEE Std 1074-1997 recommends we have a software life cycle model and IEEE Std 1490-1998 recommends we have a project management body. When we use terms related to the software life cycle model and the project management body, we refer to these IEEE Std documents for reference. Most importantly we understand that these are abstract and we may choose to partially implement these ideas.

2.1. Model

Researching for a specific model began during the early stages of the project. We were using a loosely based Scrum model. After a few weeks in, we have transitioned to a hybrid Agile Scrum model as our main provider, GitHub, did not provide the necessary tools for time management within the scope of our project.

We used https://www.zenhub.com/

Zenhub Guide: https://www.zenhub.com/guides?utm_source=Dashboard

2.1.1. Model Elements

- User Stories – Also known as Issues, these are tasks that the developers acknowledge must be completed. There are two types of Stories:
    - Stories – Stories that can be grouped together into an Epic.
    - Standalone Stories – Stories that cannot be grouped together into an Epic and are completed separately from other Stories.
- Epic – Also known as “super” Issues, these are collections of User Stories the developers acknowledge must be completed. Every Story within an Epic must be completed for the Epic to be completed.
- Sprint – Also known as a Milestone, this is a deadline for a set of User Stories that need to be completed.
- Release – Not to be confused with product release, an Agile Scrum “Release” is a deadline for a set of Epics and Standalone Stories that need to be completed.
- Points – Priority labels given to estimate the priority and/or completion of a Story or Epic.

2.1.2. Model Organization

Below is a table for the organization of Model Elements. First Column is the Type, First Row is the Need.

Take note that if something says NEEDS ALL, that refers to all the elements within the deadline of the Type.

	User Story	User Story (Standalone)	Epic	Sprint	Release	Points
User Story						NEEDS
User Story (Standalone)						NEEDS
Epic	NEEDS					NEEDS
Sprint	NEEDS ALL	NEEDS ALL				
Release	NEEDS ALL	NEEDS ALL	NEEDS ALL			
Points						

2.2. Scrum Meetings

We shall have our meetings every Tuesday and Thursday until the project is completed. There may be exceptions given on some days. The point of a Scrum meeting is to assign User Stories and Epics to be completed within a Sprint. Every week will count as one Sprint.

Furthermore, a Scrum Meeting will help us keep in check with our progress, and timelines to completion. Once a Sprint is assigned with its Stories, we must not make any more major changes to the Sprint issues unless certain developmental issues come up. In fact, these meetings should really be the only time we change Stories, Epics, Sprints, and Releases. It is important we ask questions during the meeting and list any possible problems that may arise.

2.3. Perfection

We are only human, if we were perfect we would be able to complete this project in one day. Therefore, we keep these Standards as a reference for development and may or may not fully follow it as time goes on. These standards are subject to change. We are using a hybrid version of Agile Scrum.