Development principles

NJRB2 | MVS9 | FA296

# Methodology

We have decided to use a customised variant of SCRUM Agile development, which draws in elements of Extreme Programming and adjusts the standard SCRUM timescales to better suit part-time development.

We will be using pair programming to help keep us up-to-date on what one person is doing. To keep some flexibility, solo programming can be done if one team member is feeling inspired or there is time pressure, but they should ensure their comments and commits are very well-explained, and message the group saying what the changes were. One other group member could then look at the code to check for problems. We have allocated pair programming timeslots in an attempt to ensure this style of working happens.

We will have a standup meeting three times a week, and use two week sprints. Each sprint will begin with a sprint planning session and end with a retrospective where we can report what we think is going well and what we think needs to be improved. The choice of a two week sprint was to allow us to work on other pieces of coursework without feeling too pressurised by the project work. We will use Trello to track our work items, and assign points to each work item to estimate how long they will take to complete. We will figure out how much work to assign for each sprint depending on how much other coursework each group member has during that sprint. We should be very transparent with each other about whether we’re stuck on something, or if we’ll miss a session.

Due to the time constraints of the project, and the fact that we are not making a safety-critical system (aside from the emergency drone shutdown) we will not be practicing Test-Driven Development. However, we will write tests afterwards for assurance in the event that we need to change the code.

# Design and planning

In advance designs could be drawn up once we are more familiar with ROS – this could be done in a UML style (though without as much detail) based on what we know so far, and these plans should be changed if needed, in order to show progression throughout the development. Based on advice from our supervisor in our first meeting, we should attempt to make as many ‘blobs’ of isolated code as we can. Taking this to the extreme could be interesting and useful.

We might need to consider particular restrictions when we are thinking of solutions and designing our code. Because we are working with a camera feed we will need all of our code to execute before the next frame arrives, although we will be able to reduce the framerate if needed. Depending on the computer we are running the code on, we may need to be mindful of CPU resources. Security is unlikely to be important for this project.

We will define very clear requirements (quite easy given the clear nature of our project outcomes), which will help inform our designs and workloads. These should not only be code requirements, but also involve documentation and tests needed.