

Quality Assurance Policies

Contents

- [Contents](#)
- [Coding Practices](#)
- [Testing Policy](#)
 - [Rationale for Testing Policy](#)
 - [Note \(19/05/2023\)](#)
- [Git Practices](#)
 - [Note \(19/04/2023\)](#)
 - [Note \(27/04/2023\)](#)
- [Trello Workflow Management](#)
 - [Note \(19/05/2023\)](#)

Coding Practices

The following are some guidelines which should be followed to ensure code is easy to read and follow:

- Use a virtual environment specifically and exclusively for this project, we need to track dependencies so the environment can be reproduced. (Note: running Ubuntu 20.04.5 on VirtualBox with a single local python version qualifies, provided the VM instance is not used for development outside of this project.)
- All classes, functions, and methods should include informative docstrings documenting the expected behaviour of the code.
- PEP 8 guidelines should be followed whenever possible.
- Descriptive names are to be preferred over short and/or meaningless names.
- Type annotations should be used whenever possible.
- Use whitespace, avoid lengthy blocks of uninterrupted code.
- If the code is doing something complex, add a brief comment which states what it is doing.
- Try avoid repetition: if the same or similar lines of code are repeated, they should be factored into a function or method.
- Try avoid long function signatures: if you are passing the same handful of parameters to multiple functions, consider whether it makes sense to create a class to hold these parameters.
- If assigning default arguments of mutable data types, assign None in the signature and use a conditional to assign the desired value inside the body of the function or method; using mutable default arguments inside function or method signatures is likely to cause unexpected behaviours.

Note: When deadlines are approaching, if there is significant time pressure, it may be acceptable to forego certain coding practices (e.g. type annotations, strict adherence to PEP 8, thorough documentation of all methods and functions) in order to deliver certain functionality on time. In such cases, fixing any failures on the submitted work will go into the backlog as a relatively low priority item and be assigned to the developer(s) responsible for delivering the code which fails to meet the practices outlined above.

Testing Policy

Testing policy was formalised on 27 Apr 2022, testing practices prior to this date did not always conform to the policy outlined below.

1. We adopt rostest as the framework to use to write unit tests for our code, not just unittest.

Rationale for Testing Policy

- Policy 1: unittest without rostest expects a different package structure than ROS expects when it is running, leading to import issues; more generally, it is good practice to test ROS software with rostest.

Note (19/05/2023)

Previously the following item was in the testing policy:

- From sprint 3, we adopt a test driven development methodology. This is an iterative process which entails the following:
 - Automated tests are written before new code is developed
 - Minimal functionality to pass the tests is implemented
 - The foregoing two steps are repeated to implement new functionality

During sprint 3 retrospective, it was determined that this item was taking up too much time, and was not well-suited to the stage of development we were at, which was more about integrating all of our code and getting the robot to behave as intended. Unit tests are better for testing code in relative isolation, and not so good for evaluating the complex interactions between many different classes. For this reason, we dropped this part of the policy.

Git Practices

Updated 19/04/2023 and 27/04/2023, please see notes below

1. We are following a GitHub Flow branching strategy. For details about this strategy, please consult the following link: <https://docs.github.com/en/get-started/quickstart/github-flow>
2. Each branch should be associated with a GitHub issue, which is associated with a user story. Work on that branch should only include work towards the user story linked by the issue.

3. Try and work on one small feature at a time to avoid merges resulting large updates to main, and to minimise the odds of having to resolve inconsistencies upon merge.
4. Push your code often.
5. Follow a conventional commit style of the format '<type>[optional scope]: <description>' - for more details, see here: <https://www.conventionalcommits.org/en/v1.0.0/#summary>
6. Make your commit messages informative; someone should be able to understand what you have added by reading the message, they shouldn't have to read your code.

Note (19/04/2023)

Points 3 and 5 were added on 19/04/2023; version control practices prior to this date often do not conform to these.

Note (27/04/2023)

The second bullet point was formally added on 27 Apr 2023 , version control practices prior to this date often did not conform to this practice.

Trello Workflow Management

- Team members will be allocated **one** development card at a time.
- Work on an assigned card should commence shortly after allocation.
- It is the responsibility of the card assignee(s) to move the card from in-progress to ready for test or done.
- Cards of 5 points should, whenever possible, have two assignees - when self-allocating a card of this many points, the assignee should post in Slack asking for someone else to work on it with them.
- Cards of 8 or more points need to be decomposed further, anyone interested in working on a card with this many points should discuss how to decompose it with the scrum master, Joel.

Note (19/05/2023)

Previously the following item was in the Trello Workflow Management policy:

- Once a teammate moves a card from in-progress to ready for test or done, they will promptly assign themselves another card in the backlog.

During sprint 3 retrospective, it was determined that this item sometimes led to cards being held up, because the policy necessitated that everyone was assigned a development card, but sometimes this forced people to sign onto cards when they had competing commitments. Abandoning this element of the policy should help prevent Trello cards being held up by assignee's who are busy with other commitments.