

Assignment 3

This assignment focuses on Quality, Continuous Integration and Configuration Management. You may use AI tools for research, but your answers must be written in your own words. Submit your answers in PDF to the ECS submission system <https://apps.ecs.vuw.ac.nz/submit/ENGR301>. You are welcome to use the provided Markdown template to write your answers, saving to PDF for submission.

Core and Completion Questions

Complete the following questions. You will find the GitLab documentation <https://docs.gitlab.com> helpful in answering some of them. Questions worth 20% require an answer 2-3 sentences long.

1. [10%] With reference to the *Guide to the Project Management Body of Knowledge Fifth Edition* (PMBOK5), available via the University Library, explain the difference between Quality Assurance (QA) and Quality Control (QC)?
2. [20%] Does pre-commit's integration in the repository predominantly fulfil a QA or a QC role? You must pick *one* of QA or QC and justify your answer.
3. [20%] pre-commit is designed to run locally on the workstation where changes to the repository are made. In terms of Continuous Integration practice, why is pre-commit <https://pre-commit.com> important and how does it provide benefit?
4. [20%] The `.gitlab-ci.yml` file in the repository contains two examples of `rules` entries with a particular path syntax (example below). What is the purpose of these specific entries and how do they function? Your answer should be 2-3 sentences long.

```
rules:  
  - changes:  
    - /software/device/<subdirectory>/src/**/*
```

Challenge Questions

The configuration and build management of project electronics/hardware are relatively poorly managed at present. Mentioned in several lectures were the Bill of Materials (BOM), schematic and PCB layout PDFs, and Gerber files required for PCB manufacture. These are configuration and artefact files analogous to the "vendored" MicroPython library files in `lib`, the Node `package.json` file, compiled MicroPython code, the static web app files [for those interested, more detail can be found at [ECS BOM and PCB Design](#)

[Requirements](#) (PDF)]. Answer the following questions in around 2-3 sentences each, justifying your answer in each case.

1. [10%] How are the configuration data specifying the the physical pin connections to/from the ESP32 and the addresses of I2C peripherals (rain gauge, battery fuel gauge, etc.) presently represented in the system? What would be a good configuration management strategy regarding hardware configuration?

You may wish to refer to the [Configuration Management](#) (PDF) lecture slides for configuration management principles.

2. [10%] How are BOMs, schematic and PCB layout PDFs and Gerber files generated at present? Specifically, and with some detail, how should the KiBot <https://github.com/INTI-CMNB/KiBot> program be integrated with the project in order to address the deficiencies, if any, with the generation of these files?
 3. [10%] A suite of MicroPython unit tests is part of the embedded code. What are the limitations to these unit tests (hint: consider where these tests must be run) and suggest, with some detail, one way in which testing of the data recorder system could be improved?
-