Embedded Testing

Document metadata: LO5 5.3 EVI 1

Contents:

General Approach and Awareness of Limitations

Generally, it makes sense to embed different kinds of testing in the pipeline of the system, such as unit tests, integration tests, and functional tests. Unit tests check individual units of code, integration tests check how different units of code work together, and functional tests check that the overall system meets the required specifications. This is what I would do for the system pipeline.

Furthermore, I would include formatting, linting, static analysis, and code coverage metrics to keep the codebase maintainable and automate processes which would otherwise unnecessarily consume developer time.

Specific Embedding Plans

Below, we showcase how we would evaluate for existing test suites whether to embed them in our CI pipeline or not. We take tests implemented and constructed for LO3.

Test ID	Embed?	Justification
T1	No	UI testing is expensive and complicated to automate in a CI pipeline.
T4	No	The API stress testing is an expensive task and strenuous for the system. This should not be happen automated on every code push.
T6	No	Manual human UI accessibility testing is simply not suited for embedded CI testing
T10	Yes	Compilation ability, depending on the CI pipeline, might need to be tested for specifically. For GitLab and GitHub, a build

		is automated with every push, so that would be the equivalent.
T12	No	See T1.
T15	Yes	It is important that the scheduling unit logic remains coherent throughout any code changes to ensure sound integration.
T18	Yes	It is crucial for the system requirements to constantly maintain a performant scheduling unit.