CI & Testing

Based on our CI setup, we will encode assignability into our pipeline. This is done through the fact that we are splitting down each task into various subparts. Doing this allows for more assignable requirements, and makes testing more streamlined. By narrowing the tasks, it also brings in specificity, allows us to examine if the tasks are realistic, and allows us to create a schedule based on when the tasks should be completed.

As far as the other requirements, we will ensure that the assignability is extended to then as well. I believe that all of our requirements follow the SMART methodology so it is quite simple to make a pipeline for them. Due to the nature of our project however, we will add communication with the developers into our testing pipeline. We will do our best to document the tests that we perform on the code for them so they can review it all in a concise manner.

The unit test we will add is a test on if the hex dump widget is able to have a flag added to it. This allows us to get a very basic test in our development cycle which can expand onto edge cases. Unit testing is a vital part of development, so having a simple yet strong first unit test ensures one gets off on the right foot for development. After doing edge cases we will do more general testing. Perhaps one of the members is assigned to some sort of ‘lead tester's role to manage and keep track of all tests done in the code itself.

We will look into Cutter itself to see what automated software quality assurance tool they use, if they use one. We will for sure follow the code guidelines that cutter itself has. If there isn't a standard one that Cutter uses, we will plan to use Code Coverage for quality assurance, as well as confluence and discord to update each other for manual peer review. Ensuring the code is up to the standard of the cutter team is paramount to the project, so we’ll plan to use any tool available to us and have all of us check over the code.