

4 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, power system, or software.

The testing plan should connect the requirements and the design to the adopting test strategy and instruments. In this overarching introduction, given an overview of the testing strategy. Emphasize any unique challenges to testing for your system/design.

4.1 Unit Testing

What units are being tested? How? Tools?

For Towers, the range can be tested to see if it is numerically correct. For the devices, we could test to make sure it has the right signal strength given a set of conditions. We could make sure specific weather conditions affect the specific connection strength variables per device.

4.2 Interface Testing

What are the interfaces in your design? Discuss how the composition of two or more units (interfaces) are being tested. Tools?

The biggest point of interface in our simulator would be the control panel. The control panel will be an in-game tablet where you can check on your devices, the signal strength, and see things that could affect it (ex. weather). We could make sure each device shows up on the list, they have the correct signal strength found through the unit testing, and make sure the current and upcoming weather conditions shown are correct.

4.3 Integration Testing

What are the critical integration paths in your design? Justification for criticality may come from your requirements. How will they be tested? Tools?

We need to make sure that are mod/simulator can be properly integrated into Minetest. For this, we will need Minetest, as well as some sort of mod loader or plugin manager. Since we want to add functionality to a preexisting piece of software, are simulator is much like a plugin, where the original source code stays intact, however there is no code added to it.

4.4 System Testing

Describe system level testing strategy. What set of unit tests, interface tests, and integration tests suffice for system level testing? This should be closely tied to the requirements. Tools?

This will be testing our new objects and additions that work well within Minetest. This can be just ensuring that everything loads in and has correct functionality.

4.5 Regression Testing

How are you ensuring that any new additions do not break the old functionality? What implemented critical features do you need to ensure they do not break? Is it driven by requirements? Tools?

In order to make sure one addition doesn't break a bunch of things, we will have to work on adding one object at a time. There will be certain priorities, with the control panel and the cell tower being the first two, as everything else will have to have some degree of functionality with those two things.

4.6 Acceptance Testing

How will you demonstrate that the design requirements, both functional and non-functional are being met? How would you involve your client in the acceptance testing?

We'll show that our simulator is working as intended through the control panel, as well as some other tests. The control panel should have a lot of useful information pertaining to the connection statuses, as well as the final coverage simulation button.

4.7 Security Testing (if applicable)

Making sure we don't modify any of the preexisting code, and that our new files don't cause any drastic software or OS violations.

4.8 Results

What are the results of your testing? How do they ensure compliance with the requirements? Include figures and tables to explain your testing process better. A summary narrative concluding that your design is as intended is useful.

We can get numerical data, and examples in order to understand how our software operates in order to proceed with development. If everything works as intended, onto the next thing, if not, back to Minetest to understand why things aren't working as intended.