

## 1. What - tested areas

- General user experience
- Start guide
- Node specification: `nerocli --addnode`
- Experiment config files
- Experiment specification: `nerocli --addexp`
- Experiment submission: `nerocli --submit`
- Experiment result fetching: `nerocli --fetch`
- Overall status report: `nerocli --status`
- Experiment status report: `nerocli --status EXP_ID`

## 2. Why - goal and focus

The first purpose of our testing is to receive feedback from testers that are completely unfamiliar with the Neronet system and its goals but might have lots of ideas about how to improve its front end.

Another goal is to see how quickly they can make sense of the current start guide without needing any other guidance. We want to use the peer team testing to find out how the start guide and the software could be made clearer and more understandable.

A third potential benefit of the testing is to discover potential errors in important functions such as cluster specification, experiment specification and experiment submission.

In addition to testing Neronet using the CLI, we will also test the GUI to confirm that it can also be used to perform the most important functions intuitively and without confusion.

## 3. How - approach

For the testing session we will meet at Maarintalo, in the class 240/241. To test Neronet, we will use a UNIX system which can be a personal laptop or simply a public access desktop at Maari. If need be we will set up a `virtualenv` so that the installation doesn't conflict with other Python packages on the computer. After this we instruct the testers to try and use Neronet as a normal user might.

Testing data to be used is simply the example experiments we have prepared.

## 4. Exploration log

To be filled!

- SESSION START TIME: 13:10
- DURATION (hours): 4h
- TESTER: Kaarlo, Olli, and Joni
- BUILD: 0.3.1
- ENVIRONMENT: Teemu's, Tuomo's and Samuel's Linux laptops

### 4.1 Data files

- test/experiments/sleep
- test/experiments/sleep001
- test/experiments/theanotest

### 4.2 Test notes

We executed our plans almost completely as planned. We used the laptops of our team members as the testing environment to simplify and speed up the process. One of our developers was always guiding the tester through the test focus areas while interviewing him/her regarding his/her experience. We acknowledged that it was pretty hard for outsiders to get to grips with what Neronet actually is and how it is supposed to be used. Most test subjects were unaccustomed with using the commandline and unfamiliar with SSH keys and related tools. Due to these facts the testers required more guidance.

We omitted most technical steps taken with the test subjects from this section because the start guide essentially depicts what we went through.

### 4.3 Bugs

- iss011 Color coding bug
- iss033 Making a cluster with a same name
- iss022 Submitting an experiment multiple times causes stdlog to behave erroneously
- iss023 GUI change highlight color
- iss024 GUI accepts empty fields
- iss025 GUI move errors to GUI

## 4.4 Issues

The following is a list of all the issues or improvement notes we took during the testing:

- iss012 Think about what is interesting enough to take space in the GUI
- iss013 Fetch should happen automatically when changes are made
- iss014 Submit and other core buttons/stuff should be highlighted
- iss015 More feedback from the CLI and GUI (when deleting is successful etc, experiment successfully defined)
- iss016 Submit could also define the experiment given a experiment folder
- iss017 Sending multiple experiments via CLI
- iss018 Nerocli should print help commands given no other parameters
- iss019 Cluster help is inadequate
- iss020 Status label to GUI
- iss021 Handling status reports after the size of the table increases (so that the status report isnt 100+ lines)
- iss026 Hide experiments by rightclicking label
- iss027 Hide params by right clicking labels
- iss028 Highlight all if name is clicked, else cell
- iss029 Experiment file must be on the experiment folder
- iss030 Better distinction between fixed parameters and user defined parameters
- iss031 Experiment command is confusing maybe “add experiment instead” also for clusters
- iss032 Load explanation
- iss034 Better explanation what experiment id is (same for clusters) perhaps exp name instead
- iss035 Where to see that data after fetching