Test Selection by Coverage

My company team has a three-tier testing flow as described below:

* Smoke Testing
  + Executed upon every code check-in
  + A subset of the test cases are run to cover basic functionality
  + Runtime is limited to 4 minutes
  + Code check-in is gated until Smoke passes
* Progress Testing
  + Executed every 4 hours
  + A larger subset of test cases are run to cover most functionality
  + Runtime limited to 15 minutes
  + The complete regression suite will run on the latest codebase that passes Progress. The main idea is to prevent running full regression on broken code – fewer failures to triage.
* Nightly Regress Testing
  + Executed nightly
  + Full test suite is run to cover full functionality and corner cases
  + Unlimited runtime (within reason)

My team uses an agile methodology in which requirements are continually being created and modified. This results in the continual creation and modification of test cases. Newly created test cases are automatically placed in the Regress suite, but are not added to Smoke or Progress until a test suite review is held and then the test cases are re-organized into Smoke / Progress / Regress.

This process is not ideal, and comes with several pain points. The test suite reviews generally involve getting the majority of test developers into a room to decide which subset of tests maximizes coverage while still fitting into the time frame of Smoke and Progress. The discussions center around what portion of code each developer *believes* each test is covering. This generally works, but leaves the possibility of large portions of uncovered code. The man-hours involved to get several developers into a room for the review make this a costly process. Since these redistribution reviews are not held immediately after new development & new test creation, there is a lag where new features are not tested in Smoke & Progress, and are likely to have a larger number of failures in the nightly. This is largely because new test cases generally correlate with new (untested) development code that is more unstable. Debugging 20 failures in the nightly regression takes more time and effort than to debug a single failure in Smoke.

Our project proposal is to automate the selection of tests for Smoke and Progress based on a ranking of statement coverage. Individual test case coverage and runtime will be recorded during each Regress run. The top tier 1 ranked tests that produce the most statement coverage will automatically be applied to the Smoke test case list. The top tier 2 ranked test cases that produce the most statement coverage will be used for the Progress test case list.

We will be testing this project on an existing to be determined open source codebase.