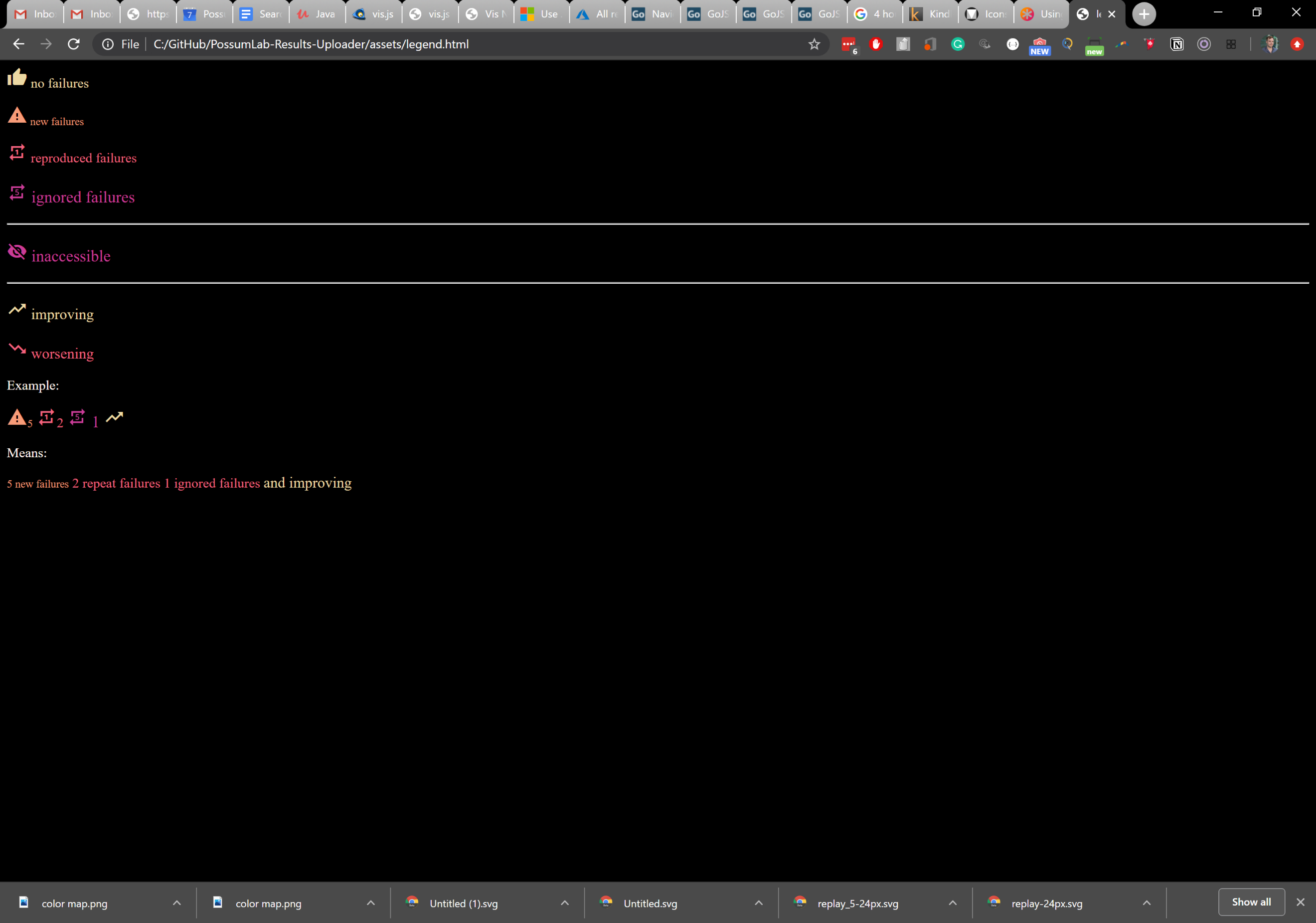
Going beyond the pie chart for test automation

What is better, no automated tests or some failing automated tests? Today, probably no test. The problem of living with failing tests is often that they lead to test results becoming ignored; and that affects not just the flaky ones. What if we could filter out the noise?

# Failure reporting

We want to know a number of things when it comes to any segment of tests we are looking at; first of are there no failures. If so great. Second, do we have new failures, these could be noise after all and tend to get ignored in flaky tests. This means it is useful to separate the failures into 3 buckets, new (happened once), reproduced (happened 2+ times in a row) and ignored (happened 5+ times in a row). Separating this information out makes it harder to ignore repeated failures just because there are some noisy tests.

The second thing we want to communicate is whether the tests were reachable, if we can’t satisfy the setup of a test we want to limit the reporting of these tests so people don’t get overwhelmed by noise due to a lower level failure.

And lastly, we want to show if this is better or worse than yesterday. This again will help communicate in times where there are failures. We won’t remember yesterday’s numbers but we do care if things get worse.

This works for small numbers but once they get big it becomes noisy again. After all if one section of the app has issues we want to make it easy to maintain vigilance in the rest of the application.

# Contextual reporting

When looking at a test failure there are a number of characteristics we have. Where the tests is, where the test failed, where and we last interacted with the application. These are all Path structures, whether it is a path on disk, or the path of a URL, we can aggregate this information in a path like structure.

This allows us to use the build in segmentation of the tests to break up a large number of failures into clusters, and reduce that ability of a large cluster to drown out all the other ones.

This is neat but not every test is equal, some tests verify that a package was signed for, others make sure the package got there. One failure is unfortunate, the other means you no longer conduct business.

# Transition mapping