

Being a square keeps you from going around in circles.

10 November 2016 · automation, investigation, questioning, regression testing

After a weary few hours sorting through, re-running and manually double checking the “automated test” results, the team decide they need to “run the tests again!”, that’s a problem to the team. Why? because they are too slow. The ‘test’ runs take too long and they won’t have the results until tomorrow.

How does our team intend to fix the problem? ... make the tests run faster. Maybe use a new framework, get better hardware or some other cool trick.

The team get busy, update the test tools and soon find them selves in a similar position. Now of course they need to rewrite them in language X or using a new [A-Z]+DD methodology. I can’t believe you are still using technology Z , Luddites!

Updating your tooling, and using a methodology appropriate to your context makes sense and should be factored into your workflow and estimates. But the above approach to solving the problem, starts with the wrong problem. As such, its not likely to find the right answers

The team are spending hours unpicking the test results. The results can’t be trusted and need to be rerun or manually reviewed. They are the problems. Until you address the reliability, accuracy and precision of the automated checks they will always be a major source of failure demand

That dream of freeing up the team to move quicker or let the testers do more exploratory or security focused testing will remain a dream - while the team spend excessive time picking through the bones of your test results.

Your “automated tests” are a measuring tool. They help you measure the quality of your app. Imagine if your ruler reported a different length every 3rd time you used it! You’d blame the ruler and build or buy a better ruler. Rather than bemoan the time it takes to get an accurate measurement - while re-measuring objects to get “best of three!”.

Try fixing or just disabling the flaky tests. Test your automated tests. Don’t “create a failing test then see it pass” - investigate whether it was failing for the right reasons and then passing for the right reasons. Speak to your team mates e.g.: “How can I create Problem X realistically to check that my tests pick it up reliably?”

Do you hear these sort of conversations in your team? If so, then your team might need some coaching.