Automation – How We Approach it

A Quote to Ground Us

“Test automation makes humans more efficient, not less essential”

Where did we Go Wrong #1?

Monolithic Suites of Tests

We run big, multi-hour suites

Off-hours/overnight

The results are vetted out by a small group

And then action is taken by the same and in a silo

How do we Make it Right #1?

“Multiple Runs for Multiple Dones”

Smaller, centralized and more efficient suites of tests

That provide more concentrated and quicker feedback to the whole team

Running in the pipeline

Results are available and actioned on by the team

Where did we Go Wrong #2?

Automate Everything

A certain percentage/number of test cases automated somehow equates to high quality

Focused on automating everything

A Numbers Game

How do we Make it Right #2?

Know your Landscape

Consider Unit/Integration tests and take action

Start with Smoke tests (CRUD)

Tie it to Definition of Done/Shippable

Measure by A of A (Automated of Automatable)

Where did we Go Wrong #3?

Automation deliverables are owned, executed and maintained by only the Test folks

The Test/Automation Engineers use a tool that Dev is likely not using, familiar with or consulted on

Automation deliverables become the responsibility of one practice rather than the team

When the time comes to address failures, the Devs can’t or won’t contribute because the perception is that it will take too much time

How Do we Make it Right #3?

Use the “Leave No Trace” approach

- Leave things in a state as if you weren’t, or more importantly, can’t be there
- Onboard: assess, confirm, build, and deliver
- Offboard: backlog, maintain, steady state, and leave

Follow Prime Time guidelines

- Standards and coding practices are aligned with Dev’s
- Agreed to by your committee of technical QEs

Now What?

Where do some of these “wrongs” apply to you?

And more importantly, where can you make them right?

- Start small – focus on a Smoke test
- Tie it to Definition of Done/Shippable
- Think like a team member – even if you’re not embedded
- Your end goal should be to provide self-service tests for anyone on the team to use- that is ultimately maintained by the team

So you can focus on innovation driven by change and right the wrongs!

Let’s Talk!

LinkedIn: Melissa Tondi

Twitter: @melissatondi

Email: melissa.tondi@gmail.com