



I For One Welcome Our Robot Overlords

How To Build Trust In Test Automation

Maciek Konkolowicz, Senior Software Engineer...In Test

@mkonkolowicz

April 8, 2022

THANKS TO ALL OUR SPONSORS!



Mutually human





Professionally

- Senior Software Developer With A Specialty in Test Automation
- Previous Enterprise Quality Architect
- Focused on quality for about 13 years now
- Manual → Automation → Quality Leader → Architect → Senior Dev



More important things!

Say Hi to Tori and Evy!

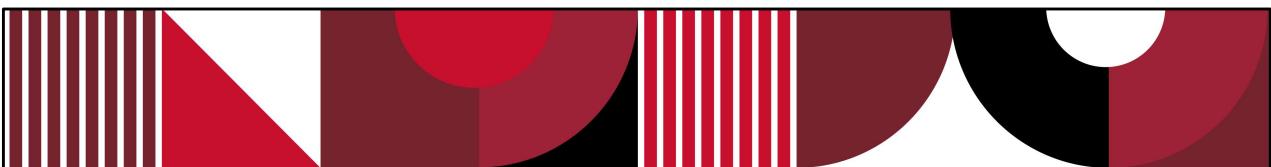


What are we going to be talking about and why

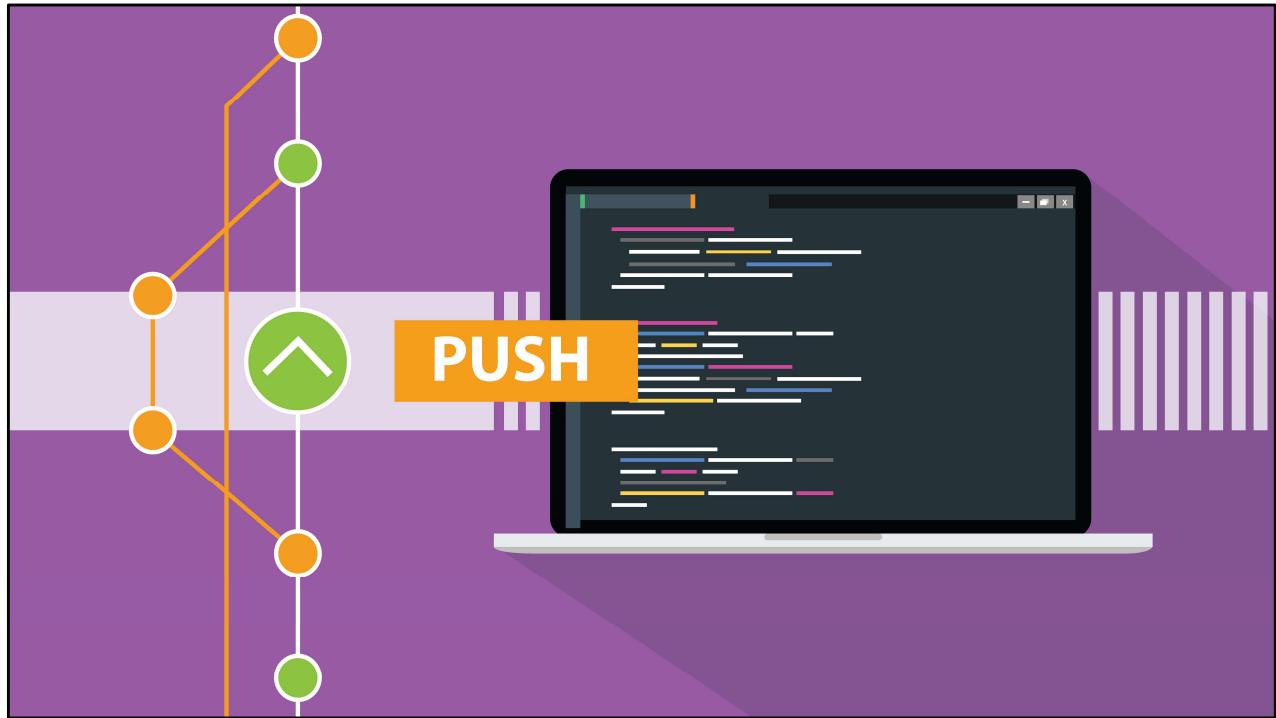
- Lessons in increasing trust in testing robots
- Story of how we built trust in our automated tests

How to Build Trust in Test Automation

- A few main aspects, that have helped our team
- Why each of the aspects is important to building trust in test automation
- What were some difficulties we encountered with improving the aspects
- How we solved our difficulties



What Used To Be.



First night of deployment: bad

- It's time for the big deployment. Whether that means a two week cycle, or a one day cycle, or a coordination of 15 teams, a deployment to production is a big deal to all teams.
- People are invested, and they want the deployment to go well, most importantly they want to know they did a good job!

Script

- Everyone gathers
- Murali, who is the engineer responsible for pushing the code, initiates the pipeline to push the code to the production environment.
- The multi-stage push is finished. Whew. The front end and middleware code has been pushed. And thank goodness there were no DB changes today...because we know how unruly those schema updates get right?
- Now it's time to validate!
- Janice takes the conn on validation. She begins to read through a document, and realizes that she doesn't have access to the system to kick off the tests, because "she doesn't do this every day".
- The team panics a bit.
- Out of left field, Barb, another senior BA speaks up...she was on a push last week and was

able to get to the system for kicking off tests. She'll give it a try.

-Success! Barb has been able to kick off the tests!

-Now. How long do they take? OH, about 35 mins.

-The team waits...Murali asks how the weather in North Carolina is? Barb talk about how there's no snow like there is in Michigan. Dontae cracks a joke, mentioning how cold it is for his Floridian taste, and that he one time had to buy a jacket when visiting Detroit.

-Everyone keeps waiting, it's been 15 minutes.

-OH! The test automation is done! But wait, there's a red circle.

-Everyone gets a bit nervous.

-What went wrong?

-Neither Barb, Janice, Murali nor Dontae really know what the log error "chromedriver only supports chrome version 85" really means...but it seems important?

-What do we do?

-A wild Brandi appears. She's the one that wrote the script.

-Brandi starts digging in.

-Janice and Barb start their own investigation

-Brandi re-runs the test on her machine...it passes

-The team is now waiting for validation for 45 minutes. Murali is thinking about clearing snow from his driveway, and hoping it's the last time this year, while Dontae is playing Breath of the Wild on his Switch.

-Janice and Barb do some manual validation...it also passes

-Finally, Janice asks if this could a glitch in the tests. Brandi, confirms it's a timing error, because she can't reproduce the error. Everyone exhales, calls the push successful , and moves on with their lives.

Second Night Of Deployment

-It's time for the big deployment. Whether that means a two week cycle, or a one day cycle, or a coordination of 15 teams, a deployment to production is a big deal to all teams.

-People are invested, and they want the deployment to go well, most importantly they want to know they did a good job!

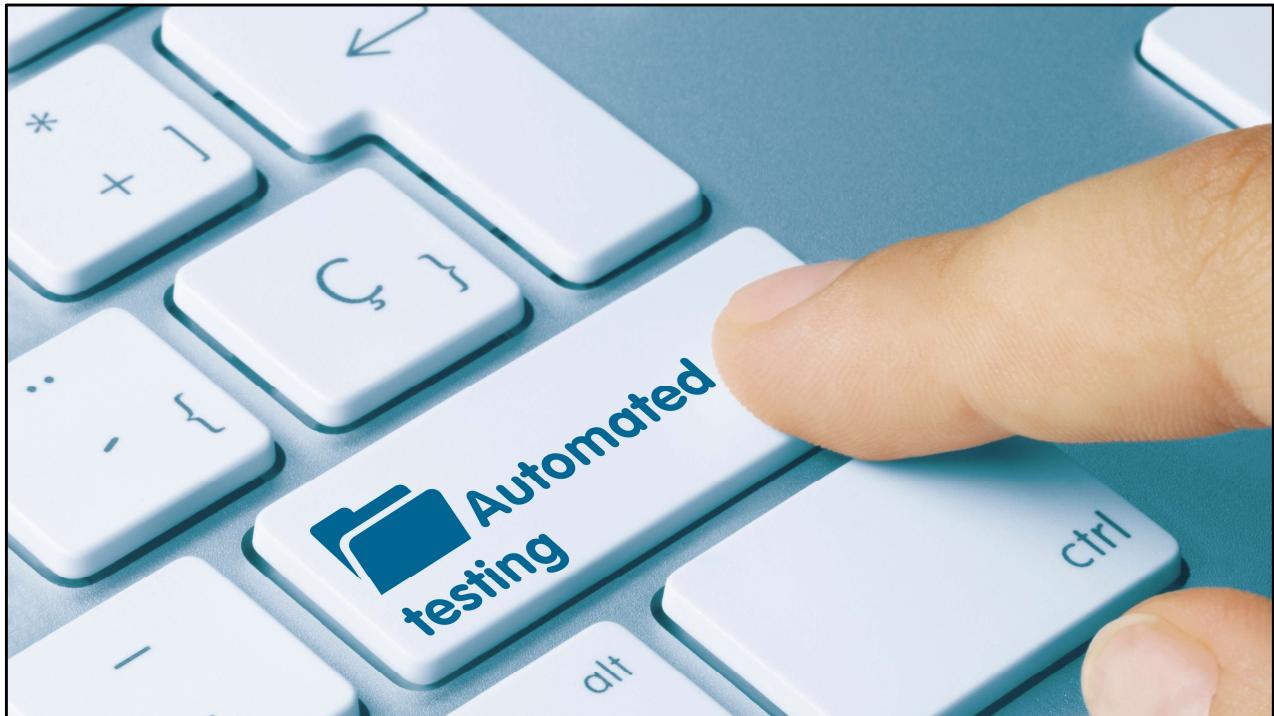
-Everyone gathers

-Murali, who is the engineer responsible for pushing the code, initiates the pipeline to push the code to the production environment.

-The multi-stage push is finished. Whew. The front end and middleware code has been pushed. And thank goodness there were no DB changes today...because we know how unruly those schema updates get right?

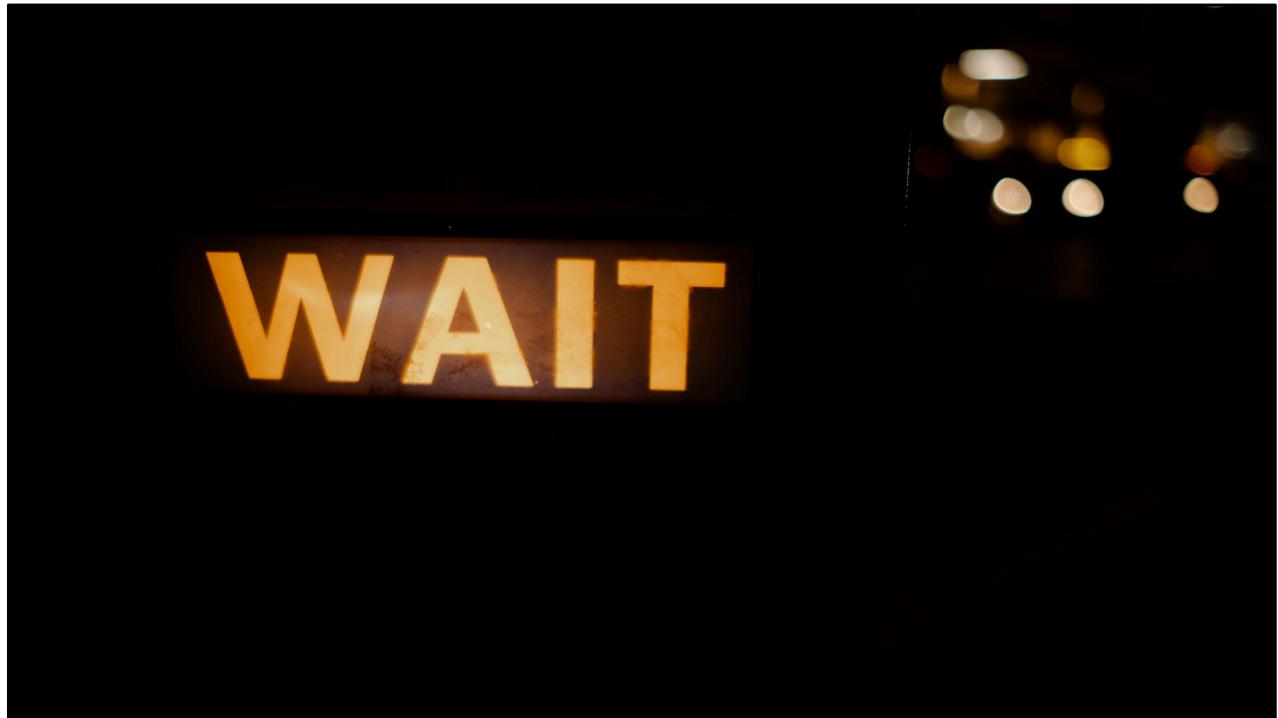
-Now it's time to validate!

-Janice takes the conn on validation. She begins to read through a document, and realizes that she remembers she still doesn't have access to the system to kick off the tests, because "she still doesn't do this every day".



Script

- Now it's time to validate!
- Janice takes the conn on validation. She begins to read through a document, and realizes that she doesn't have access to the system to kick off the tests, because "she doesn't do this every day".
- The team panics a bit.
- Out of left field, Barb, another senior BA speaks up...she was on a push last week and was able to get to the system for kicking off tests. She'll give it a try.
- Success! Barb has been able to kick off the tests!



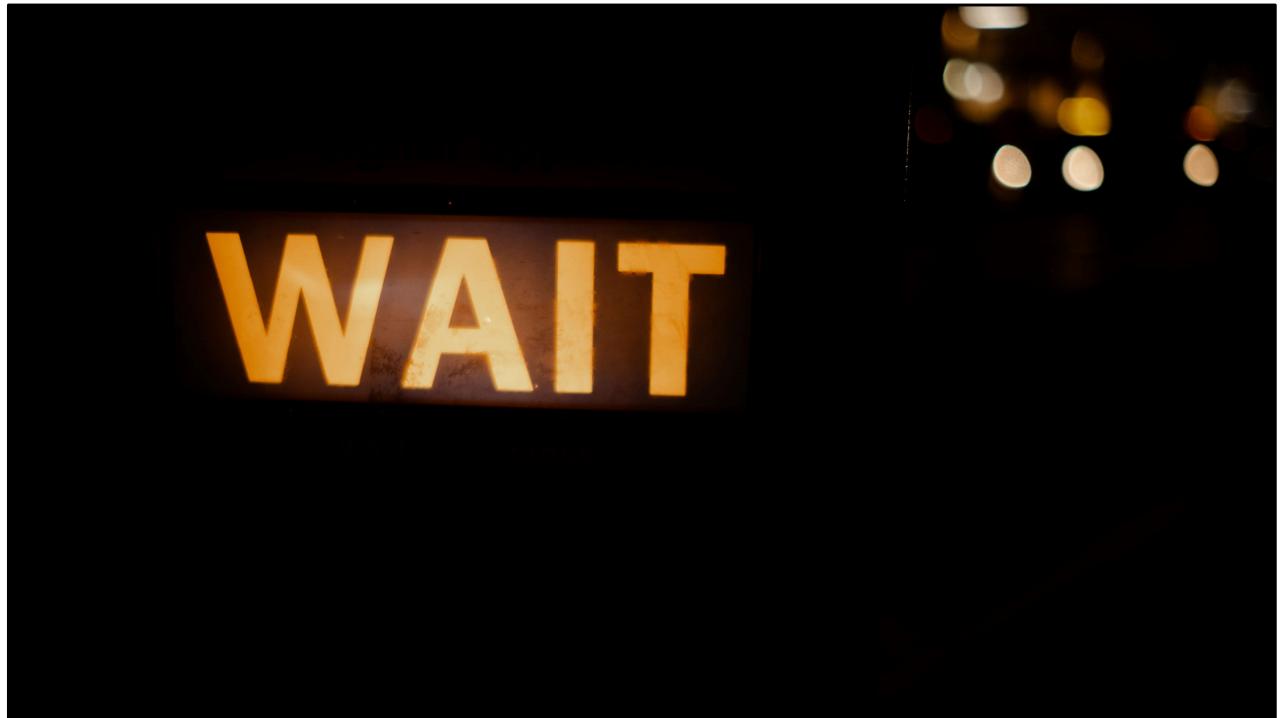
Script

-Now. How long do they take? OH, about 35 mins.

-The team waits...Murali asks how the weather in North Carolina is? Barb talk about how there's no snow like there is in Michigan.



Dontae cracks a joke, mentioning how cold it is for his Floridian taste, and that he one time had to buy a jacket when visiting Detroit.



Everyone keeps waiting, it's been 25 minutes.



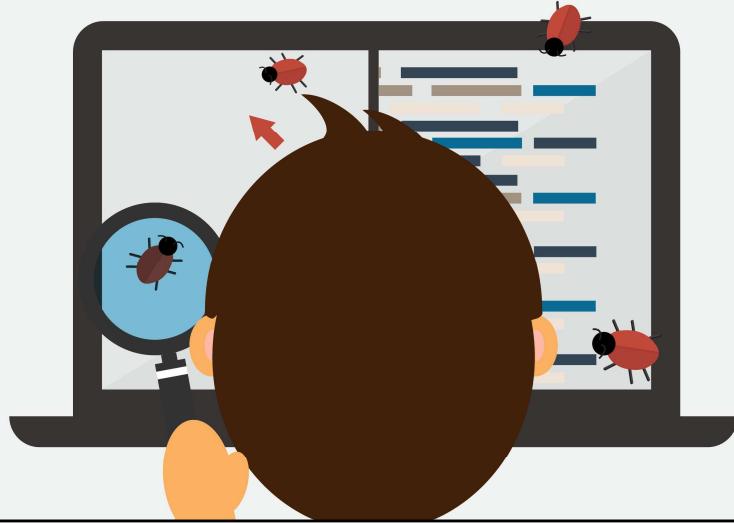
-OH! The test automation is done! But wait, there's a red circle.

-Everyone gets a bit nervous.

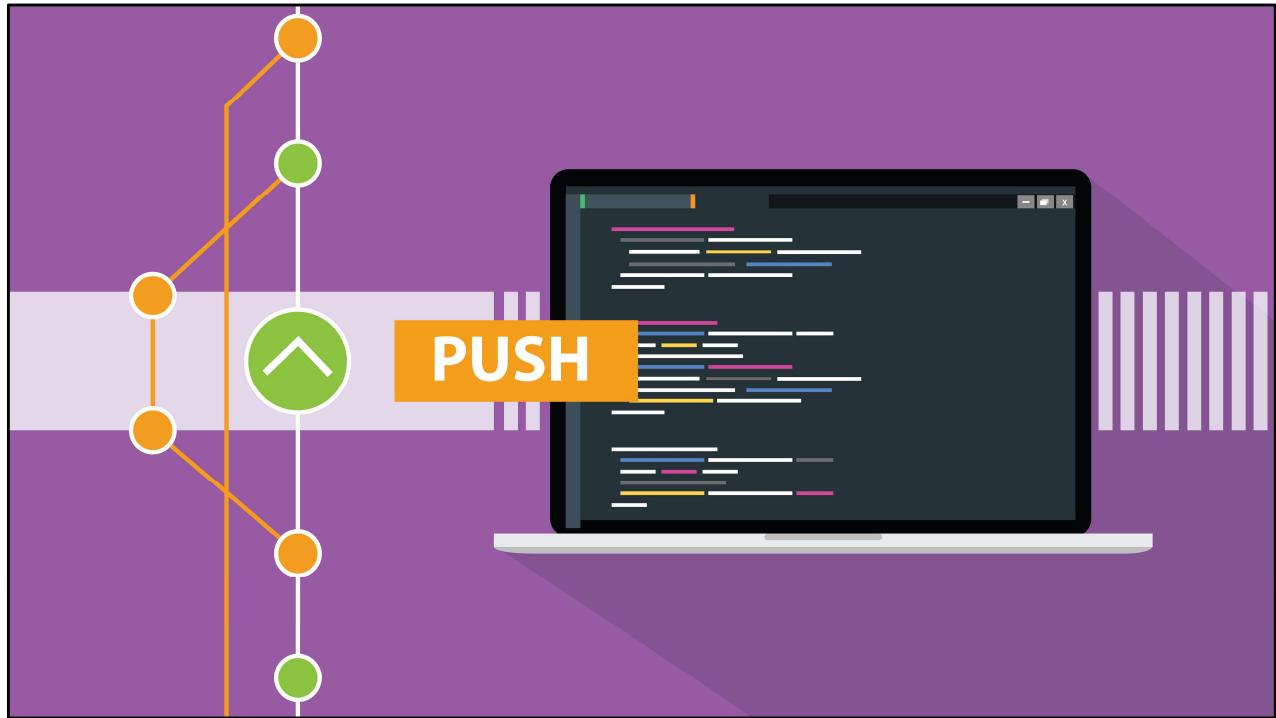
-What went wrong?

-Neither Barb, Janice, Murali nor Dontae really know what the log error "chromedriver only supports chrome version 85" really means...but it seems important?

-What do we do?



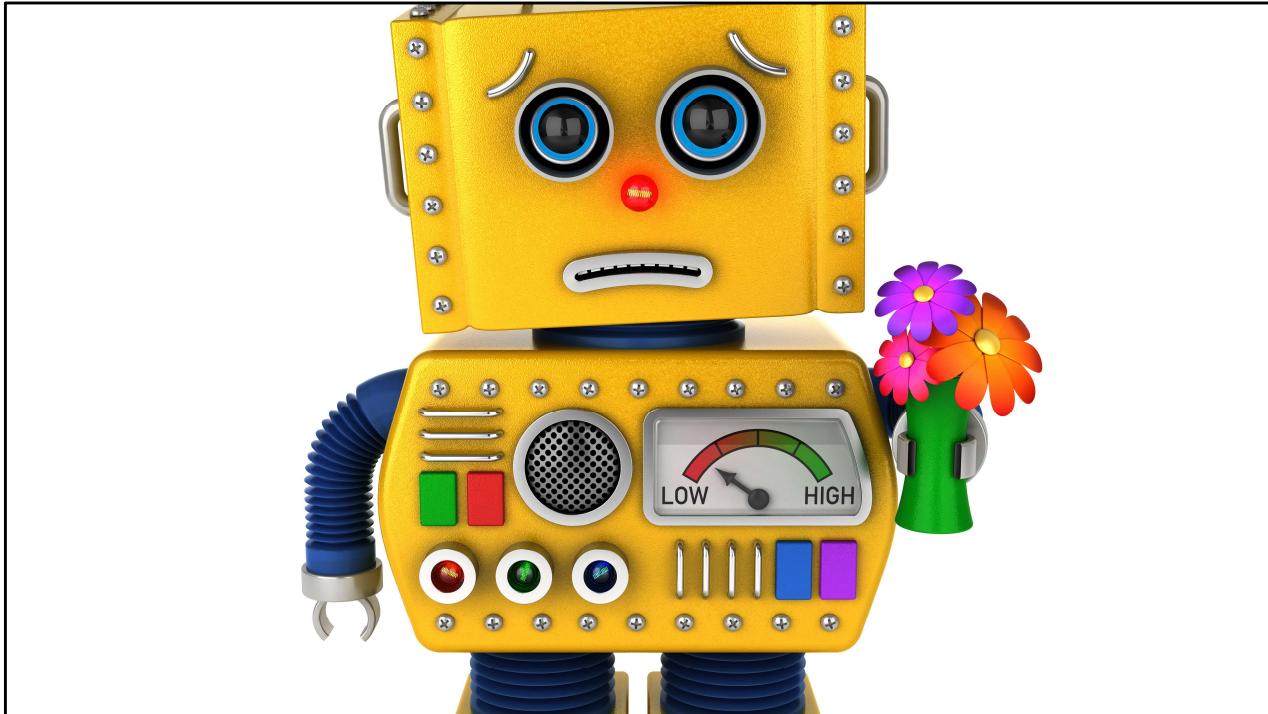
- A wild Brandi appears. Oh thank god. She's the one that wrote the script.
- Brandi starts digging in.
- Janice and Barb start their own investigation
- Brandi re-runs the test on her machine...it passes



Second Night Of Deployment

-It's time for the big deployment. Whether that means a two week cycle, or a one day cycle, or a coordination of 15 teams, a deployment to production is a big deal to all teams.
-People are invested, and they want the deployment to go well, most importantly they want to know they did a good job!

- Everyone gathers
- Murali, who is the engineer responsible for pushing the code, initiates the pipeline to push the code to the production environment.
- The multi-stage push is finished. Whew. The front end and middleware code has been pushed. And thank goodness there were no DB changes today...because we know how unruly those schema updates get right?
- Now it's time to validate!
- Janice takes the conn on validation. She begins to read through a document, and realizes that she remembers she still doesn't have access to the system to kick off the tests, because "she still doesn't do this every day".

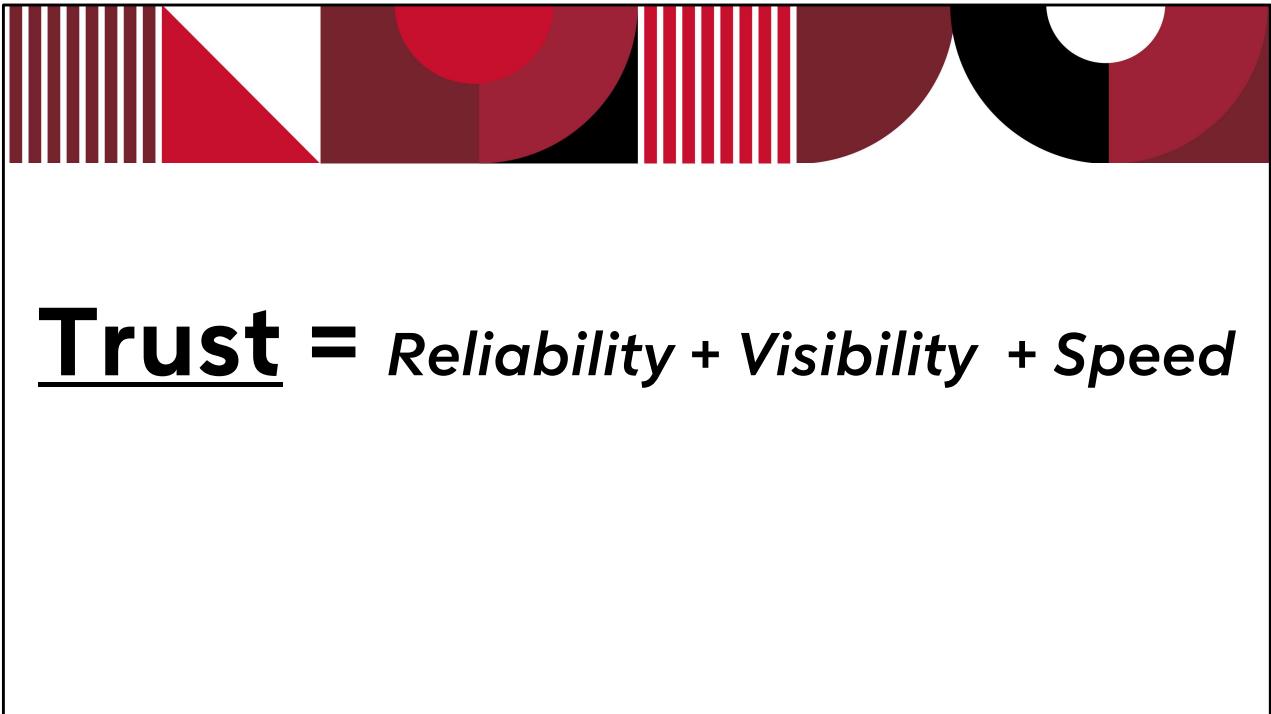


-Janice decides to validate manually. And skips the test automation, because who really wants to wait around for an hour to figure out the reason for a failure was a mismatch between a chrome driver executable and chrome version on the executing machine.

-The team moves on, the risk from not validating is accepted, **because the test automation is not trusted**



- By not having trusted the test automation, the team is really rolling the dice with respect to the product
- The team incurs risk, because manual tests simply do not provide the same consistent outcome as solid test automation does
- How do we build trust in test automation**, after such experiences as the late-night frustrating push, which happened to us regularly?



Trust = *Reliability + Visibility + Speed*

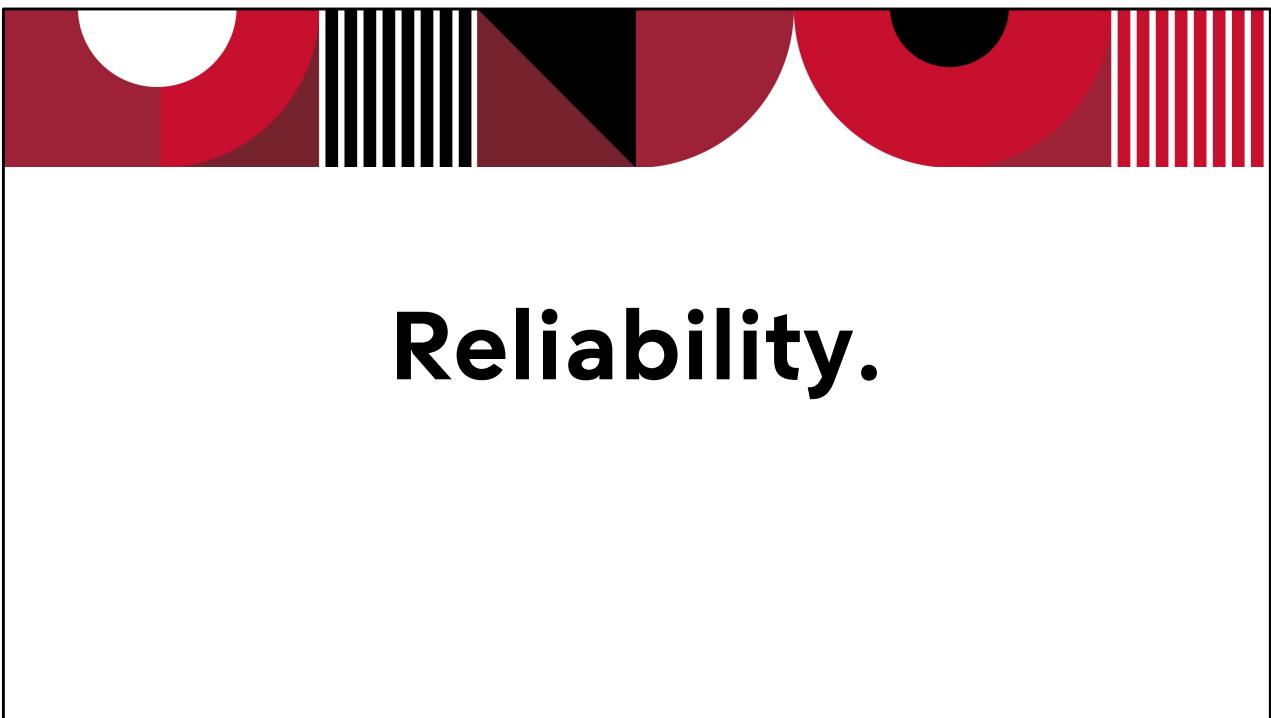
Formula

Achieving Trust in Test Automation = Reliability + Visibility + Speed

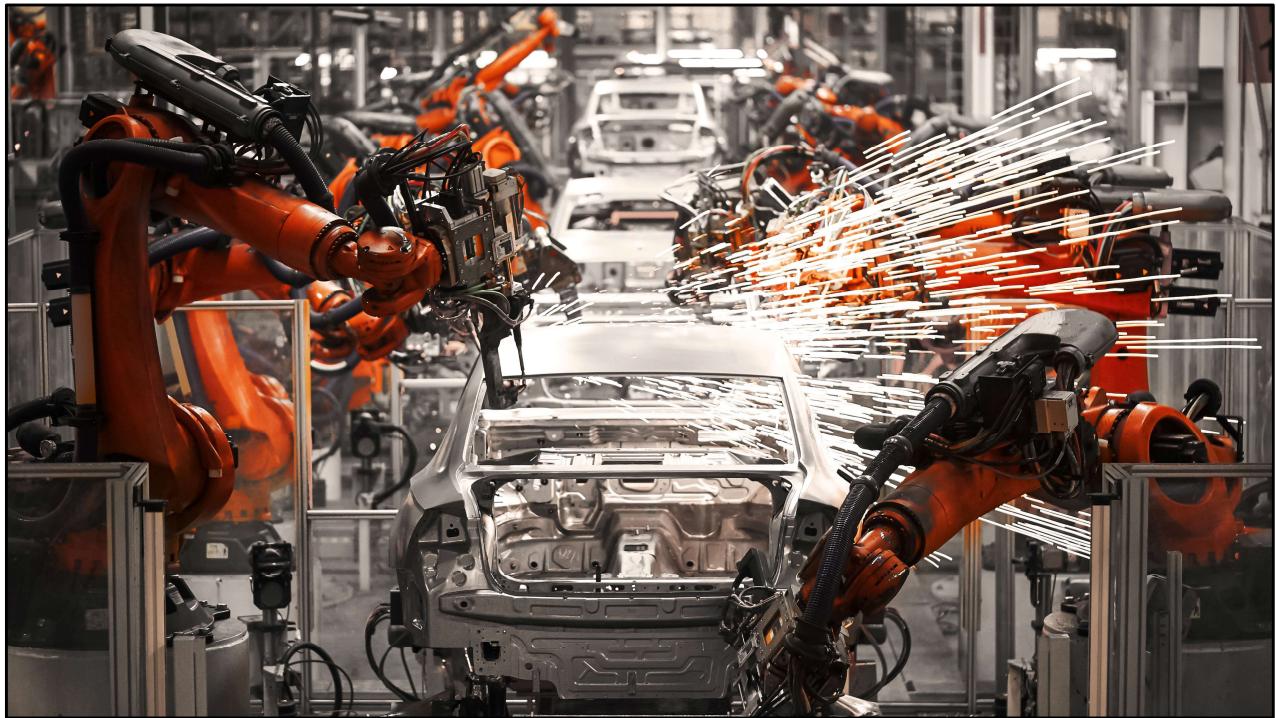
-This formula is what we believe brought us from a low level of trust in test automation, leading to a high level of manual effort and a high level of risk acceptance, to a high level of trust in test automation leading to a low level of manual effort and a low level of risk acceptance.

-By evolving these aspects of test automation, we have been able to make the lives of our team members easier, while ensuring our understanding of functionality remains uniform.
-All because we cemented their trust in our automated systems for testing

-For the rest of this presentation, we will dive into each of these aspects, and unpack why each is important, what problems we faced with each aspect, and how we fixed them.

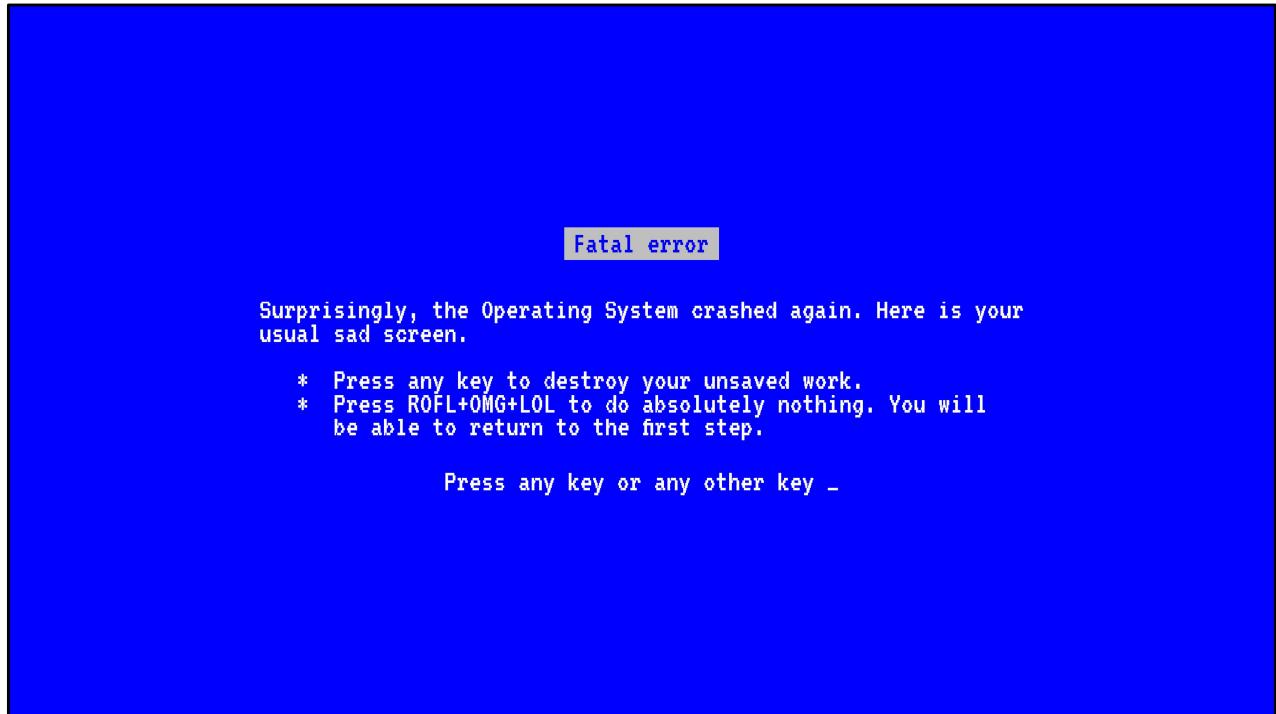


Reliability.



Factory Analogy

- Let's treat our test automation as a production line for validations
- Our tests are the cars being produced our system is the line



How Does Reliability Build Trust?

Factory Analogy

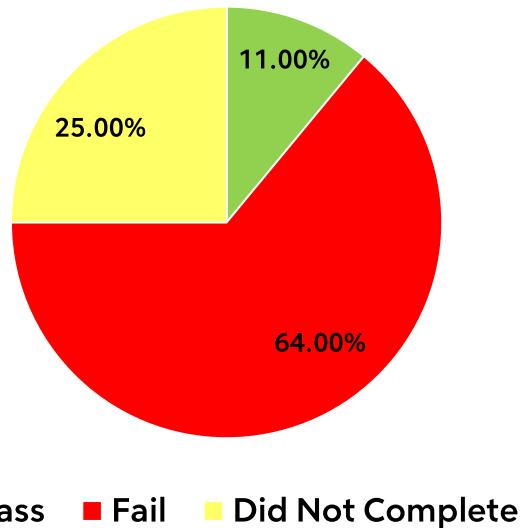
-If we couldn't be sure how often the line was running, would we trust that it would produce our validation as often as we needed them?

-We want to know that the line will consistently produce the same product, when we need it!

Why Is Reliability Important To Trusting Test Automation?

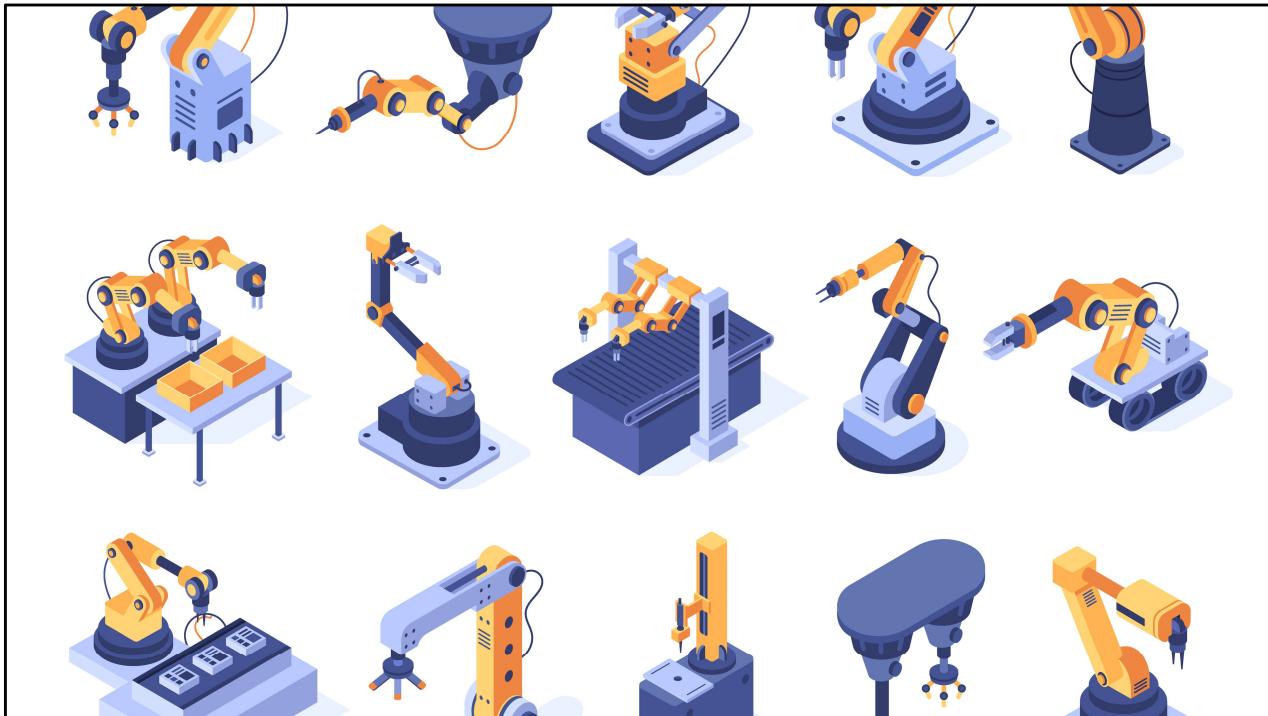
-Without an accurate consistent outcome, test automation is hard to trust!

Legacy Test Automation Execution Sample (100)



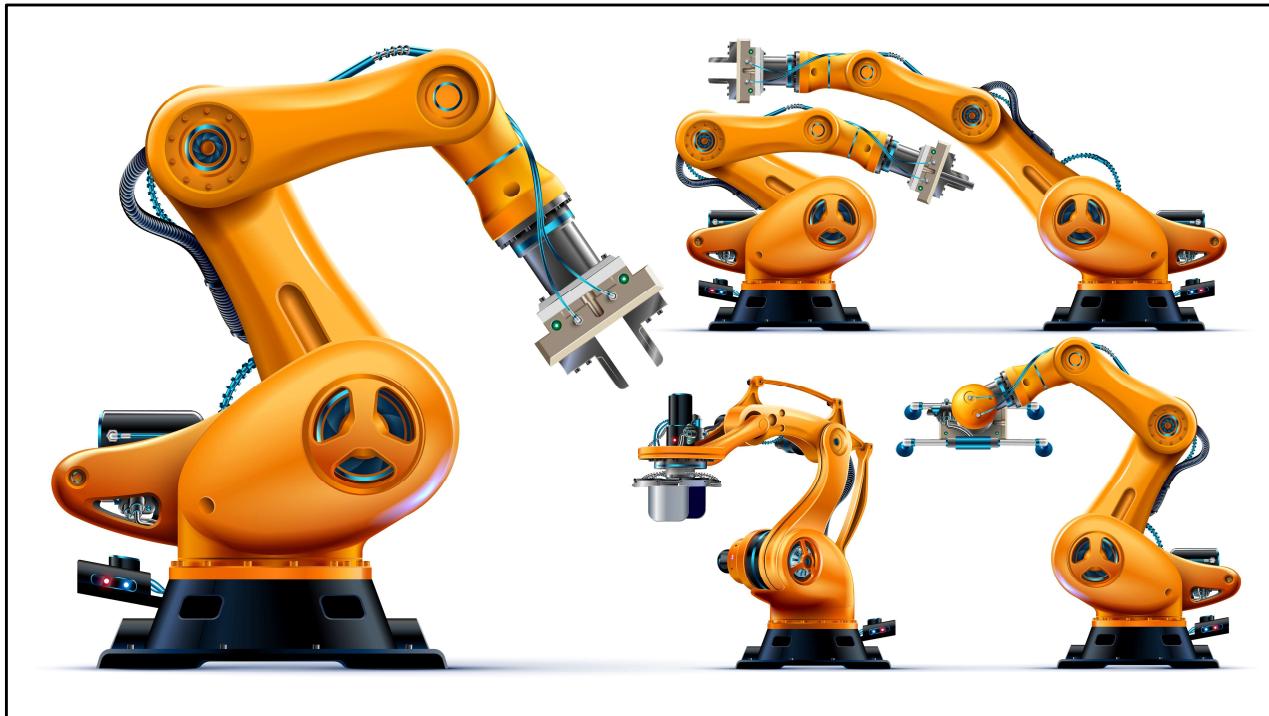
How Reliable Was Our Automation?

- Selenium Builds 420-520: 25 not finished, 11 passed, 64 failed builds
- Our test executions were sometimes passing, sometimes failing, then pass on re-run, and sometimes it wouldn't.
- Our team had to guess whether the outcome of a test run was going to produce a valid result. We had to spend a lot of time proving whether the problems experienced in the test run were related to the product or tests.
- It was a lot of work, and it led to a point where the test automation was not relied on



How Did We Improve The Reliability Of Our Test Automation To Build Trust?

- Our test automation efforts were ongoing for years, and had a lot engineers adding a lot of code, in a lot of different ways
- We had a Selenium C# suite, we had Protractor TypeScript suites, we even had some Katalon suites!
- But the result of a lot of different ideas, was that much knowledge was silo'd, and focused on individuals or small groups.
- Our Selenium suite was written by a centralized QA group which eventually was dispersed to cross functional teams, the Protractor TypeScript suites were a result of the influence of a few senior devs, and the Katalon suite was maintained by a single team.
- So when it came to maintenance, none of those suites were well understood.
- Over time, our application changed, and our automation needed to be updated, but due to the variation of approaches, motivation, and tooling, our test automation did not change enough to stay reliable.



How Did We Improve The Reliability Of Our Test Automation To Build Trust?

Reduced Friction: Single Test Framework

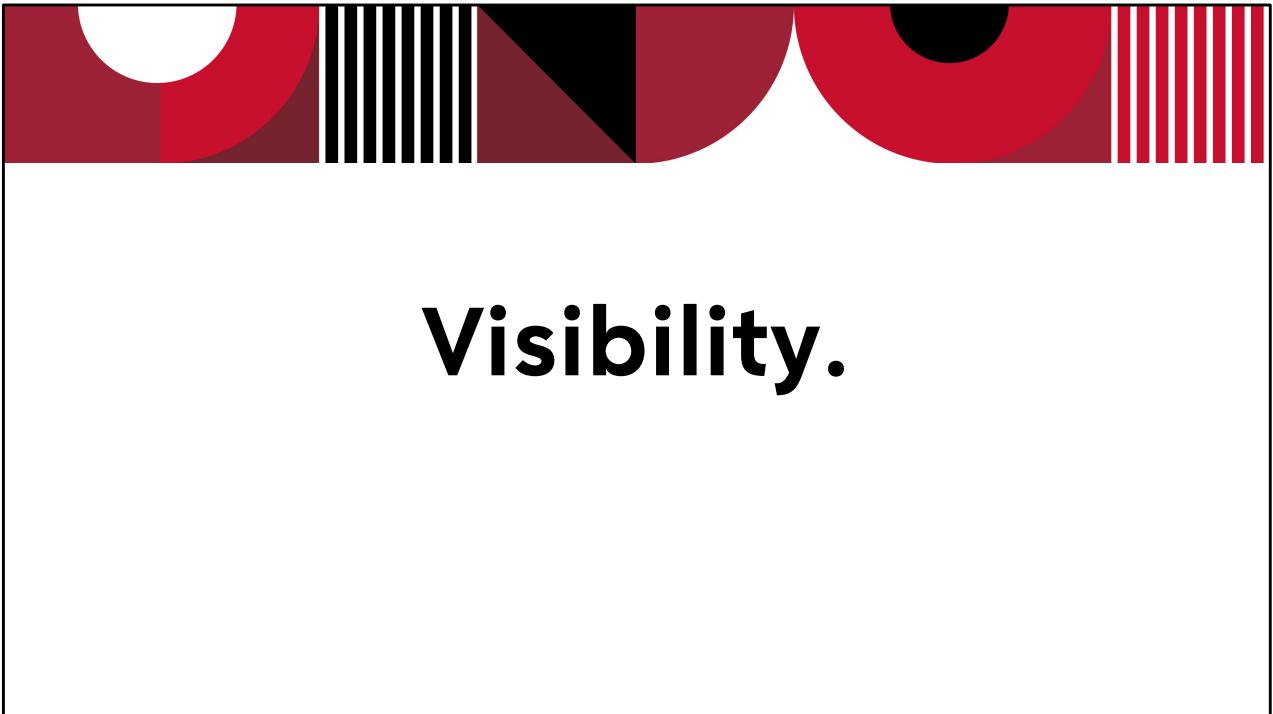
- We found the common denominator between test automation and development
- If you use a testing toolset that aligns with your development pattern, you will set up your initiative for success
- Most of our team writes front end code in TypeScript and we use Angular as the main front end framework
- Upon researching available front end test automation tooling in the Angular space, we chose Cypress
- It has a large footprint in the Angular community
- We had the opportunity to write in the same language as our front end code
- And we had evidence that our senior devs were excited about it.

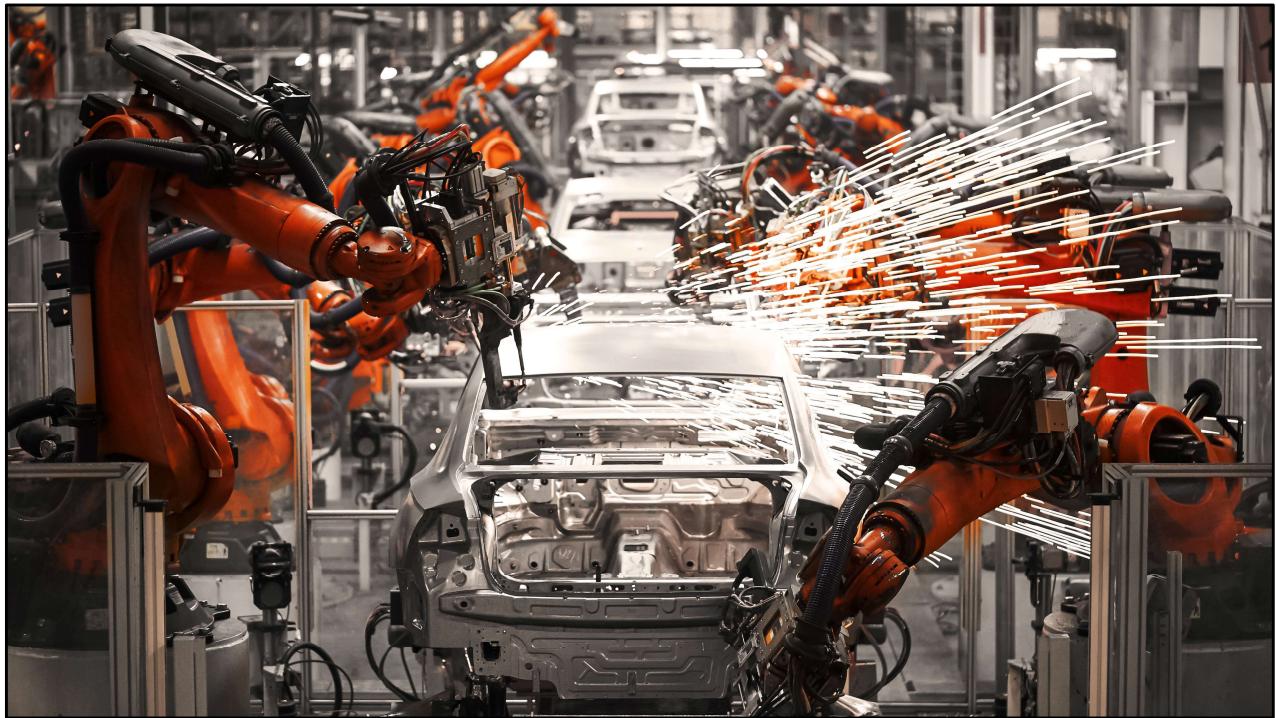


How Did We Improve The Reliability Of Our Test Automation To Build Trust?

Same Factory: Single Execution Pattern

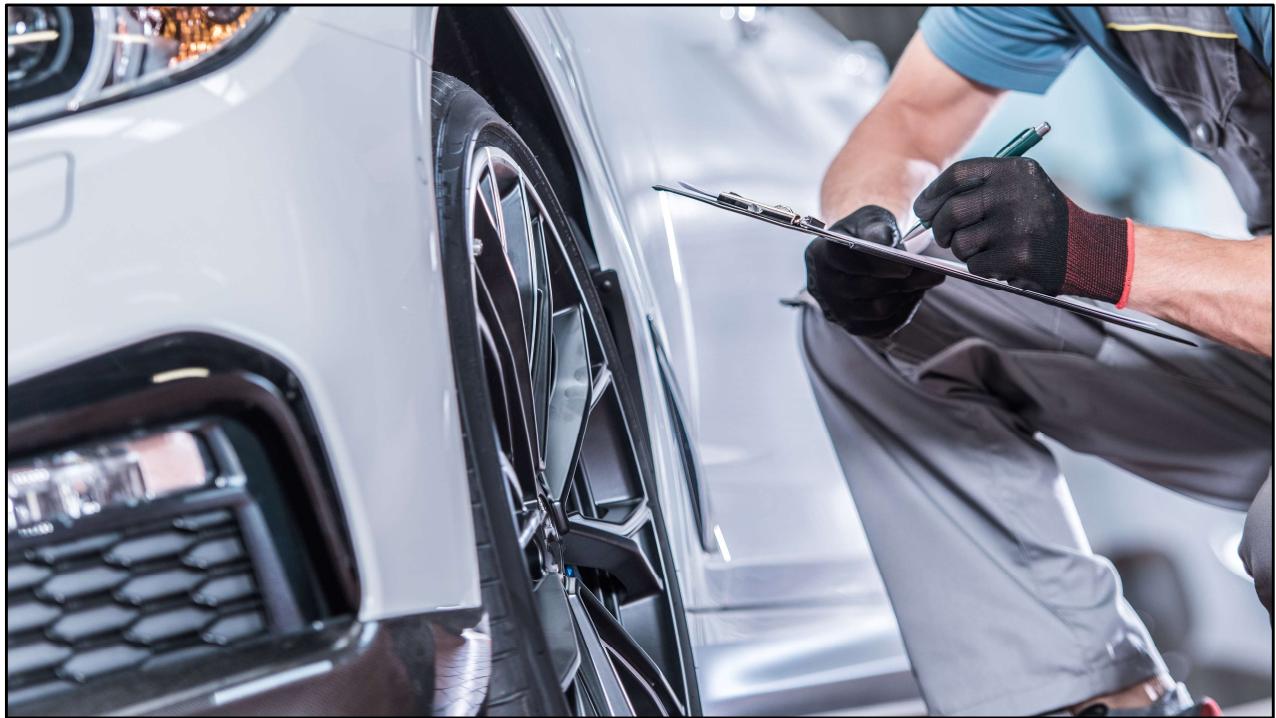
- We also hypothesized that while maintaining uniformity between the languages for production code and test code, it was also important to maintain uniformity with respect to how the tests are executed.
- Maintaining uniformity in execution of our tests and our push pipelines, would reduce friction with respect to maintenance therefore increase reliability
- We chose to align our test execution with the way we pushed our code, namely by utilizing CircleCI as our test kick off platform.





Factory Analogy

- Let's treat our test automation as a production line for validations
- Our tests are the cars being produced our system is the line



How Does Visibility Build Trust?

Factory Analogy

-If our final product that what was coming off the line, wasn't inspected, would we trust it was good?

-We always want to be able to easily inspect our expected product.

Why Is Visibility Important To Trusting Test Automation?

-Without a visible result, test automation is hard to trust!

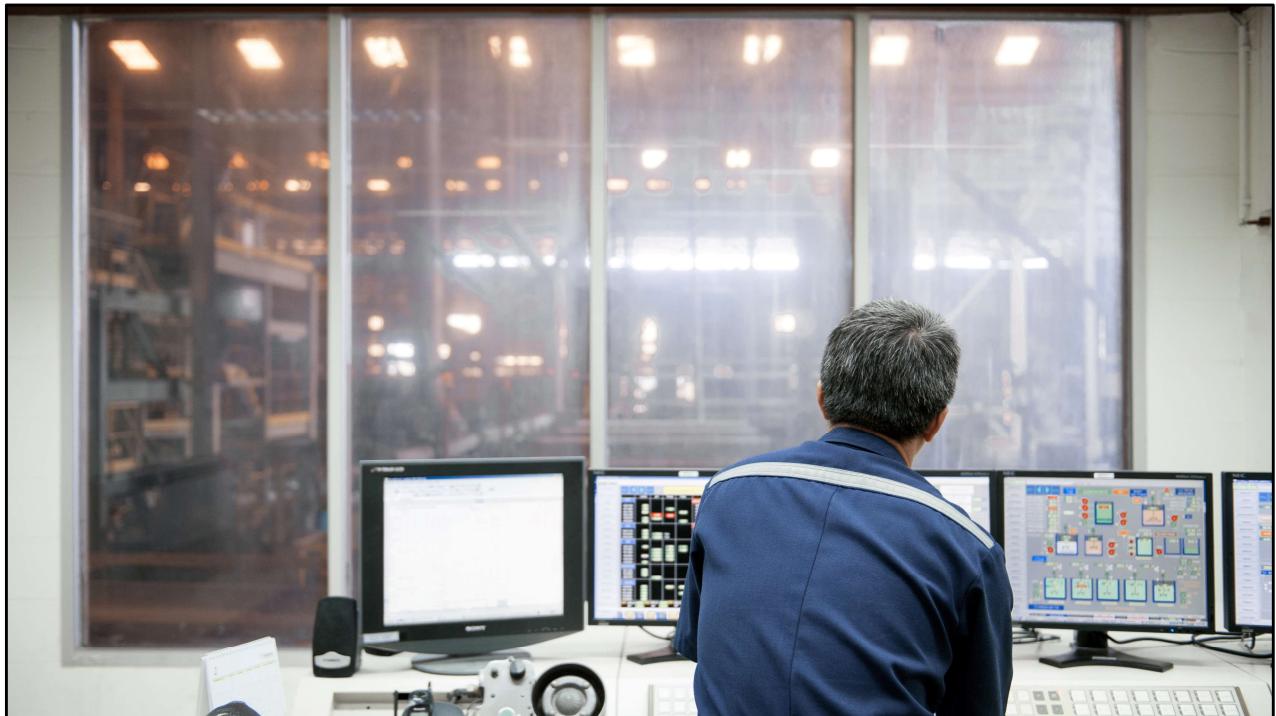


How Visible Were Our Test Runs?

-No one really knew what the tests were doing, and if they ran into problems, what the problem was.

-Our team had to read code to figure out what the test did. When the test failed, someone had to look through logs, from a CI/CD system, or try and repro the scenario manually to understand what the failure was.

-It was a lot of work, and it led to a point where the test automation was not executed, because no one wanted to admit that it was too hard to investigate a problem, if one came up.



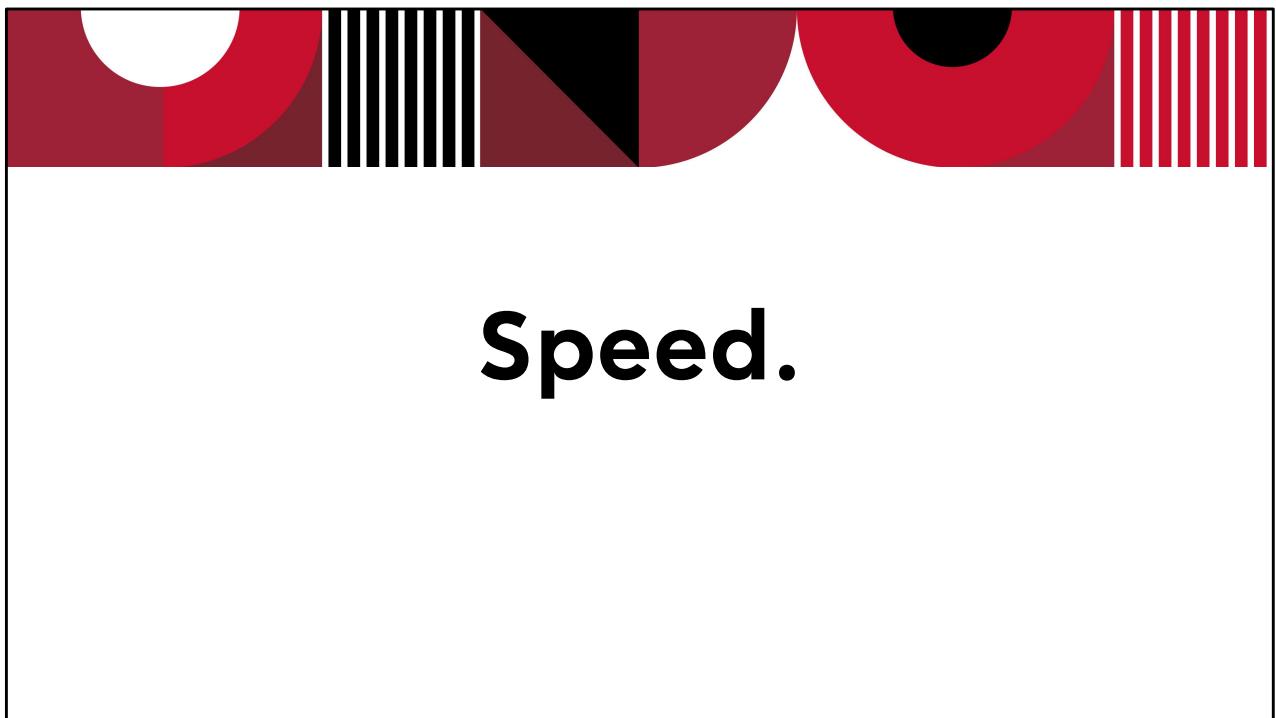
How Did We Improve The Visibility Of Our Test Automation To Build Trust?

-In order to expose what was going on in our tests, we made a conscious choice to pick a test automation framework which produces easy to interpret visualizations, out of the box.

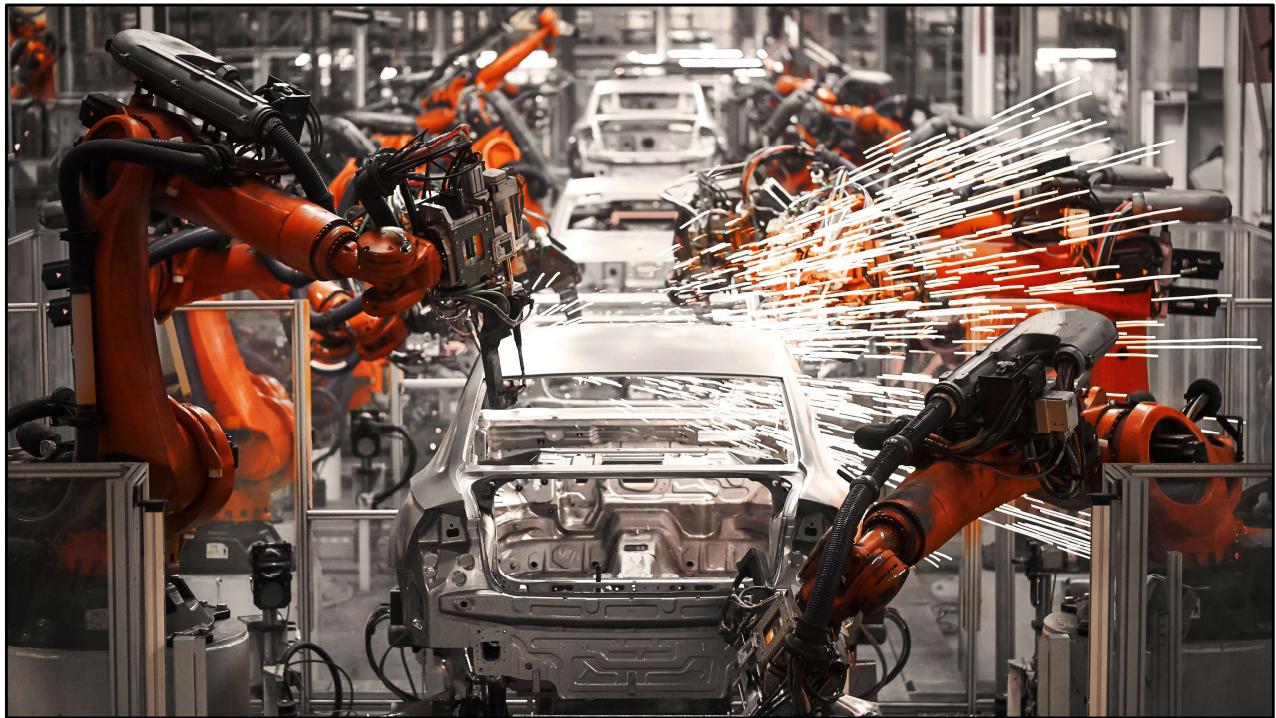
-We coupled that with an open-source test execution framework (sorry-cypress) which makes visualization of the results and problems extremely frictionless and uniform for all team members.

-It's like we built a control room, and were able to inspect our factory's work from above the floor

-Reducing the friction in understanding a test failure, drastically decreased the time between a test failure, and the ability of a team member to determine whether it was a product or test failure, which provided a drastic increase in trust of the test execution suites, since an issue was no longer a mystery with respect to the root cause.

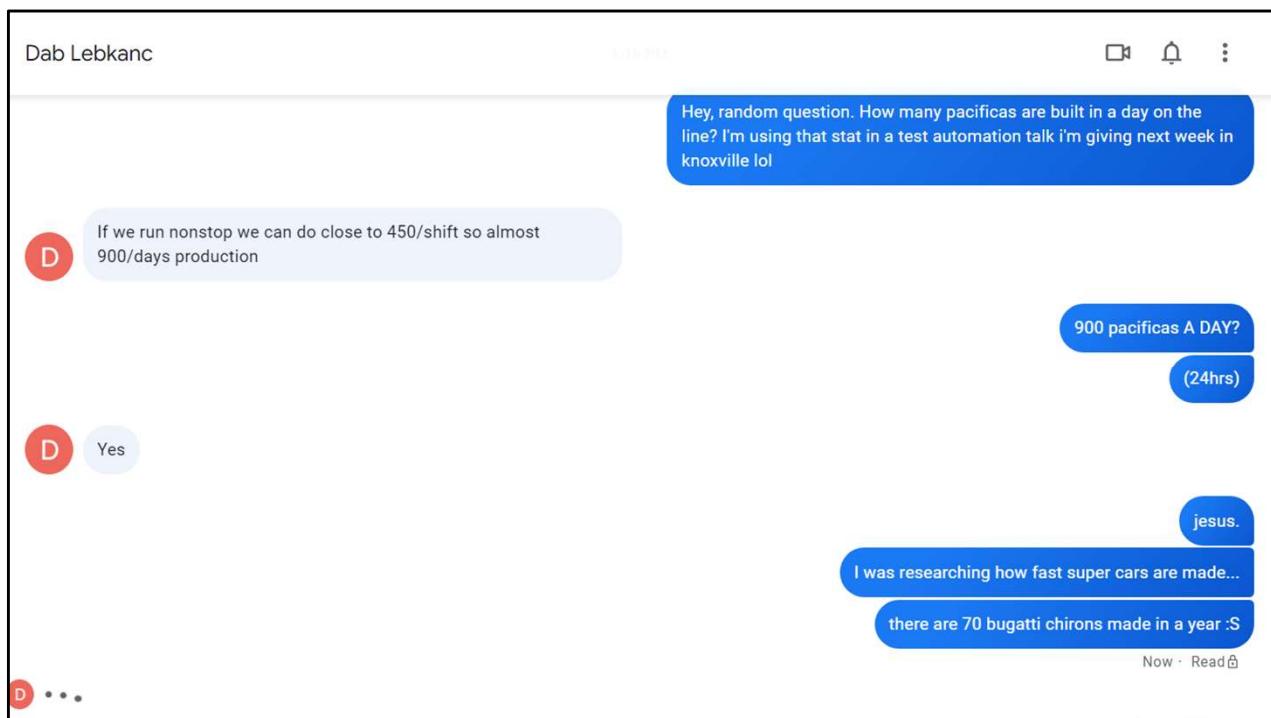


Speed.



Factory Analogy

- Let's treat our test automation as a production line for validations
- Our tests are the cars being produced our system is the line



-One of the core value propositions of test automation is speed!

-We trust test automation, because we anticipate it can do more in less time than we can!

-Our line was moving at the speed of a manual test!

-We could not trust the test automation to complete before the manual testing was done!

-So our folks took to manual testing instead, because they could not trust the test automation from finishing before they did their manual testing.

-They needed fast results

PROD-Smoke Tests

✓ Succeeded

Summary Commits Work Items [View logs](#)

Now at [Release-517](#)
[View all deployments](#)

 Deployment succeeded
on 5/4/2021, 9:45 PM • ⏱ Ran for 29m

29m

Agent phase - **Succeeded**
7/7 task(s) succeeded
11 errors, 1 warning ^
✗ [1620177421.802][WARNING]: Timed out connecting to Chrome, retrying...
✗ [1620177436.150][WARNING]: Timed out connecting to Chrome, retrying...
✗ [1620177537.336][WARNING]: Timed out connecting to Chrome, retrying...
✗ [1620177550.935][WARNING]: Timed out connecting to Chrome, retrying...
✗ [1620177610.200][WARNING]: Timed out connecting to Chrome, retrying...

PROD-Smoke Tests

 Failed

[Summary](#) [Commits](#) [Work Items](#) | [View logs](#)



Now at [Release-517](#)

[View all deployments](#)

45m

Deployment failed

on 5/17/2021, 9:59 PM •  Ran for 45m

Agent phase - **Failed**

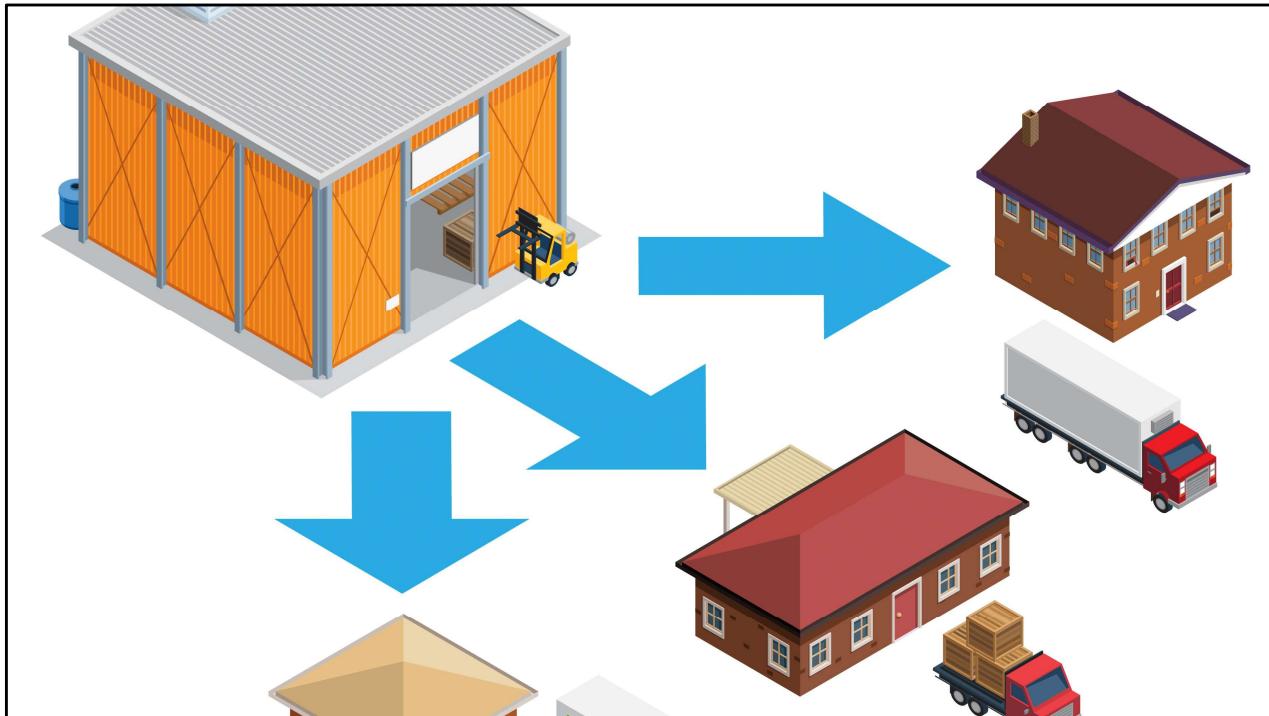
Execute QLMS Smoke Tests failed

9 errors, 1 warning ^

 [1621300476.565][WARNING]: Timed out connecting to Chrome, retrying...

 [1621300492.575][WARNING]: Timed out connecting to Chrome, retrying...

 [1621300590.508][WARNING]: Timed out connecting to Chrome, retrying...



How Did We Improve The Speed Of Our Test Automation To Build Trust?

Modern Testing Framework

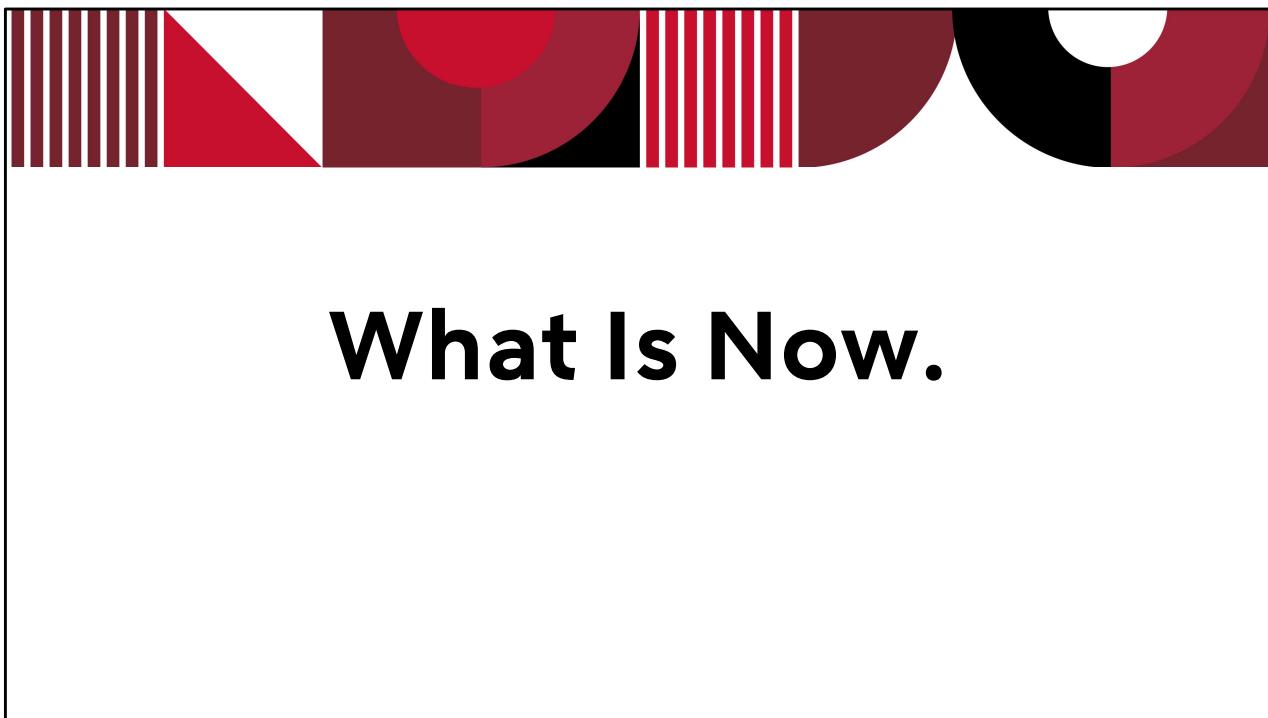
- We took the direct from factory approach for each test
- We decided to re-write our tests using a modern testing framework
- The difference between a traditional and modern testing framework is the lack of a proxy
- A modern testing framework lives in the same process as the browser, via a nodejs server communicating with the browser
- This enables modern testing frameworks (ex. Cypress, Playwright) to have access to most browser functionality, see all traffic within the application, and most importantly implement the required actions quickly (30% faster)



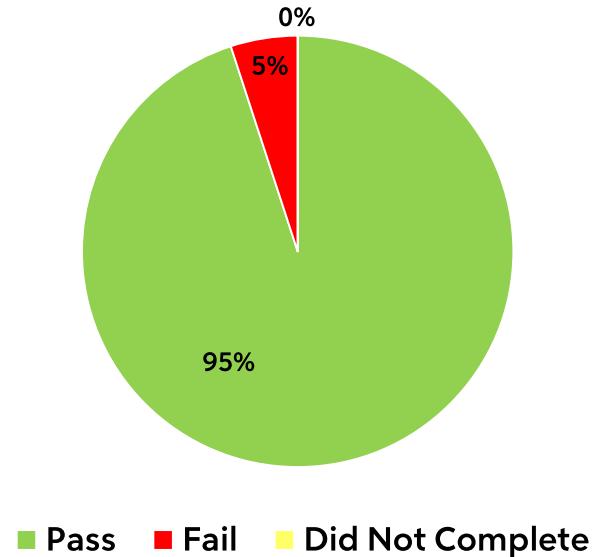
How Did We Improve The Speed Of Our Test Automation To Build Trust?

Maintaining Velocity While Scaling

- While switching to a modern testing framework gave us a 30% speed increase per test, the biggest difference WRT execution speed, was the decision to change the test execution framework, which allowed us to execute our tests in parallel as opposed to in sequence and provided flexibility with respect to the number of tests running at once.
- We switched to a test execution framework which takes advantage of container scaling via Kubernetes.
- The basic principle we stick to is 1 container per test. If our suite has 6 tests, we will run 6 containers at a time. If it has 20, we can run 20 at a time.
- While there are some limits, the speed at which we will run our test suite is dependent on how much money we want to spend on scaling.
- Currently, we have a low enough number of tests, that we can maintain the principle of our test suite run time being as long as our longest running individual test



Today's Test Automation Execution Sample (100)



How Reliable Was Our Automation?

Quality Reports

Portal Test Suite	Next Gen Portal Test Suite	Salesforce Test Suite	Pricing Calc Test Suite	Account Settings Test Suite	New Leads Test Suite	Conditions Component Test Suite
Choose an Environment <input type="radio"/> Beta <input type="radio"/> Prod <input type="radio"/> Prod-experimental <input type="radio"/> Beta-	Choose an Environment <input type="radio"/> Prod-experimental <input type="radio"/> Beta-experimental	Choose an Environment <input type="radio"/> Test <input type="radio"/> Beta ! You've got to pick one.	Choose an Environment <input type="radio"/> Test <input type="radio"/> Beta <input type="radio"/> Prod	Choose an Environment <input type="radio"/> Test <input type="radio"/> Beta <input type="radio"/> Prod	Choose an Environment <input type="radio"/> Test <input type="radio"/> Beta <input type="radio"/> Prod	Choose an Environment <input type="radio"/> Test <input type="radio"/> Beta <input type="radio"/> Prod
! You've got to pick one. Run Tests	! pick one. Run Tests	! You've got to pick one. Run Tests	! to pick one. Run Tests	! to pick one. Run Tests	! to pick one. Run Tests	! to pick one. Run Tests

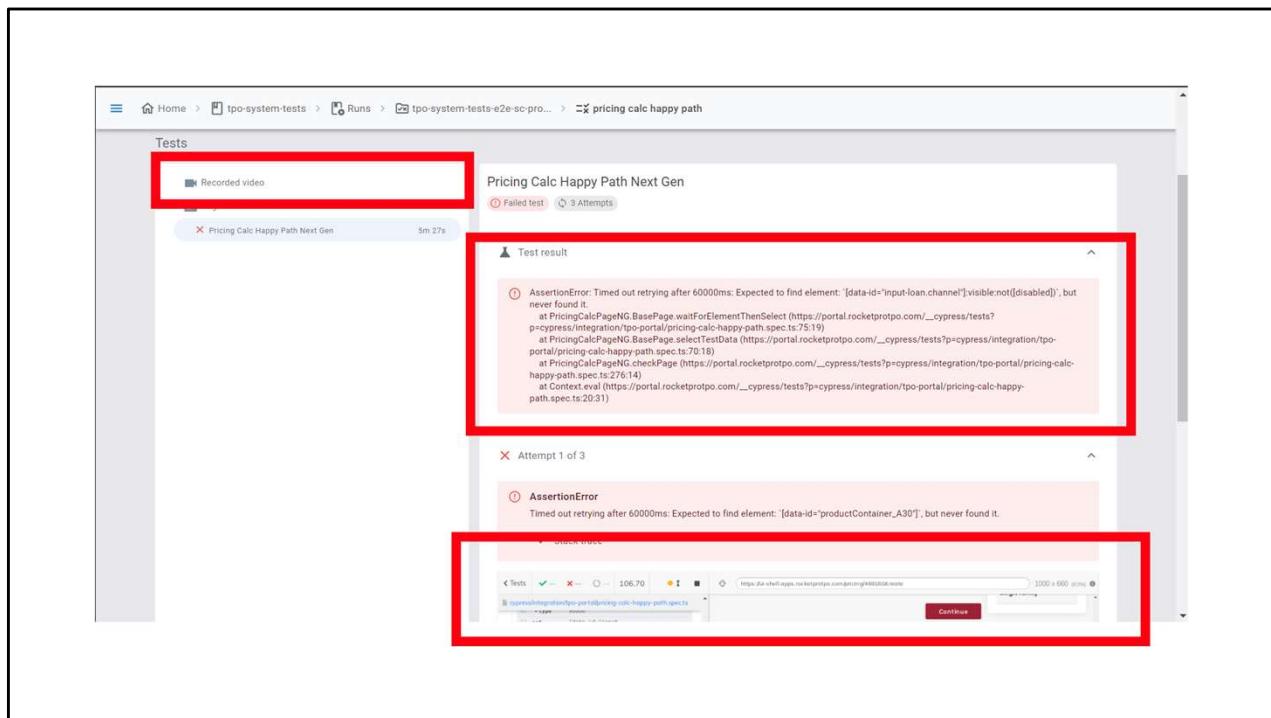
- We no longer have to dig through documentation
- Single Sign On for all team members to start tests

The screenshot shows a web-based interface for managing test runs. At the top, there's a navigation bar with links to Home, tpo-system-tests, Runs, and the current page, which is tpo-system-tests-e2e-sc-prod-3010. Below the navigation is a header for the specific run, showing the date (Mar 28 2022, 01:04:05), duration (8m 13s), and the total number of tests (6 / 6). There are also buttons for Hide Successful Specs and Auto Refresh.

The main content area is titled "Spec Files" and contains a table. The table has columns for Status, Machine #, Group, Spec Name, Duration, Tests stats, and Actions. The data rows are as follows:

Status	Machine #	Group	Spec Name	Duration	Tests stats	Actions
Passed	2439	tpo-system-te	alicepeasy saml purchase correspondent	2m 56s	6 green, 0 yellow, 0 red	🔗
Passed	3560	tpo-system-te	alicepeasy saml purchase va	3m 54s	6 green, 0 yellow, 0 red	🔗
Passed	1707	tpo-system-te	alicepeasy saml purchase	3m 54s	6 green, 0 yellow, 0 red	🔗
Passed	6670	tpo-system-te	johndmaryhomeowner saml refi correspondent only	2m 58s	6 green, 0 yellow, 0 red	🔗
Passed	2439	tpo-system-te	johndmaryhomeowner saml refi ushuffle only	3m 7s	6 green, 0 yellow, 0 red	🔗
Failed	9845	tpo-system-te	pricing calc happy path	4m 24s	0 green, 1 yellow, 1 red	🔗

-Single Sign On to see results in summary



- Single Sign On to see result artifacts
- NETWORK TRAFFIC HINT

Home > tpo-system-tests > Runs > tpo-system-tests-e2e-sc-prod-3010

tpo-system-tests-e2e-sc-prod-3010

(Mar 28 2022, 01:04:05) (8m 13s) (6 / 6)

Spec Files

Status	Machine #	Group	Spec Name	Duration	Tests stats	Actions
Passed	2439	tpo-system-te:	alicepeasy sani purchase correspondent	2m 56s	1 0 0 0 0 0 0	
Passed	3560	tpo-system-te:	alicepeasy sani purchase va	3m 54s	1 0 0 0 0 0	
Passed	1707	tpo-system-te:	alicepeasy sani purchase	3m 54s	1 0 0 0 0 0	
Passed	6670	tpo-system-te:	johnandmaryhomeowner sani refi correspondent only	2m 58s	1 0 0 0 0 0	
Passed	2439	tpo-system-te:	johnandmaryhomeowner sani refi wholesale only	3m 7s	1 0 0 0 0 0	
Failed	9845	tpo-system-te:	pricing calc happy path	4m 24s	0 1 1 0 0 0	

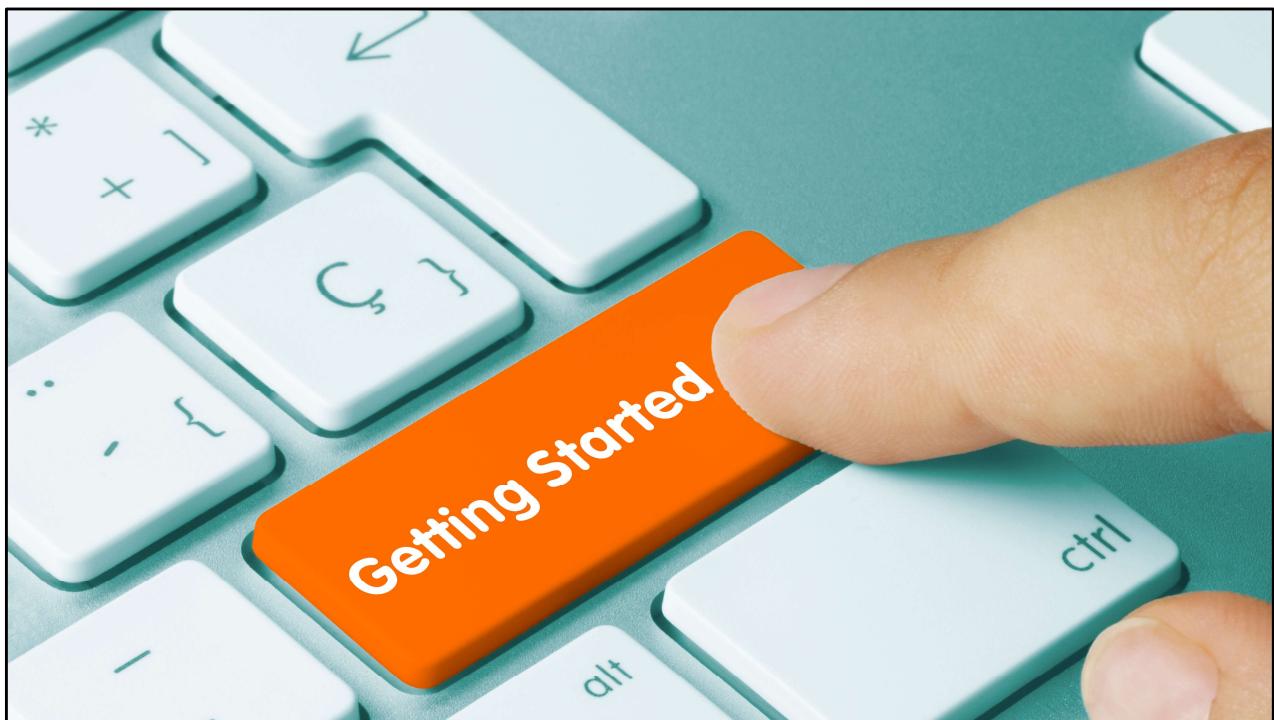
8m 13s

8m vs 45



-We did it!

-Everyone is happy, confidence in the test automation grows, and the entire push takes 15 mins with proper validation.



- 1. Find Test Automation Champions**
- 2. <http://www.cypress.io>**
- 3. <http://www.sorry-cypress.dev>**
- 4. Run In Parallel!**
- 5. Write Some Tests And Show Them Off!**

5 Steps To Instantly Increase Trust In Your Test Automation

1. Find Your Test Automation Champions
2. Identify An Exciting Modern Testing Framework To Use
3. Identify A Test Execution Framework Which Provides A Dashboard
4. Identify A Test Execution Framework That Runs Tests In Parallel
5. Write Some Tests And Show Them Off To Leadership



Trust = *Reliability + Visibility + Speed*

Recap: What Did We Talk About

Reliability

-**Empowered our teams** to participate in the test automation build and maintenance process by aligning with tools most engineers could learn and enjoy using, based on what they already knew!

Visibility

-**Focused on improving our team's** ability to easily understand results both positive and negative

Speed

-**Reduced the amount of time our team** was waiting for results, by switching to modern test automation and execution frameworks

ROCKET Mortgage

@mkonkolowicz

<https://www.mkonk.com>

@mkonkolowicz

April 8, 2022

Questions, Resources, Contacts

Notes:

- Transition to chiron slide
 - We don't want to make chirons
 - We want golfs
- Numbers on the pie chart for after