Testing Less, Achieving More: How we mastered the Art of Strategic Testing

Daniela Aquilina

HUSTEF
HUNGARIAN SOFTWARE TESTING FORUM

“

Quality is never an accident; it is always the result of intelligent effort.

John Ruskin

Sharing my story...

My First Task - Problem Identification

- Too many tests
- Not enough resources to keep up with test maintenance
- Flaky automated tests
- Too much manual testing due to flaky tests
- Slow release cycle
- Persistent bugs on production despite passing regression

Problem Breakdown

Too many tests

Not enough resources to keep up with test maintenance

+

Flaky tests

Too much manual testing due to flaky tests

=

Slow release cycle

Persistent bugs on production

The Solution

1

Identify a new list of test cases (simplified & relevant)

2

Automate the new test cases in a robust framework

3

Transition to a faster and more reliable release cycle

A Closer Look into the Finer Details ...

Development Teams Structure

Team 1	Team 2	Team 3	Team 4
FE Engineers BE Engineers	FE Engineers BE Engineers	FE Engineers BE Engineers	FE Engineers BE Engineers
1 Test Engineer	1 Test Analyst	1 Test Engineer	1 Test Analyst

Testing Responsibilities

Release Testing
Ticket Testing + Regression

(Test Analysts & Test Engineers)

Manual Testing
of all Sprint Tickets

(Test Analysts & Test Engineers)

Automation
Development &
Maintenance

(Test Engineers)

The Interesting Part!

Implementation Details

Implementation - Phase 1

Come up with a new set of test cases (aka “Critical Paths”)

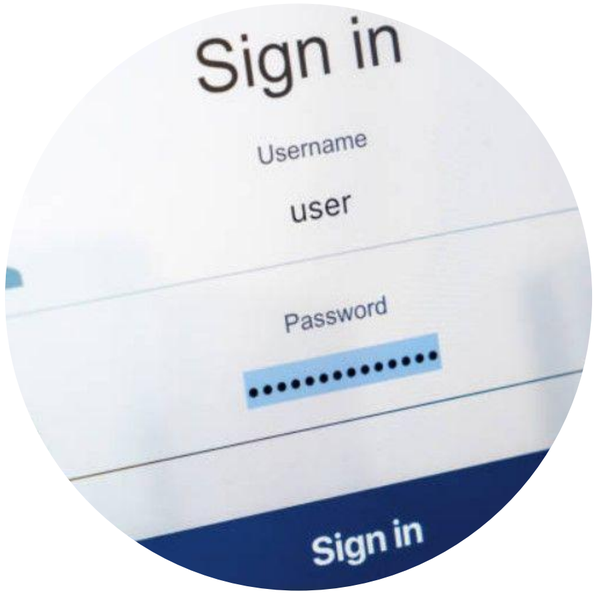
Document the critical paths

Test Analysts executed the tests manually

What makes a test a critical path?

1. If this test fails, it is considered as a blocker
2. A regulatory requirement
3. Functionality that is crucial for your business

Examples of a Critical Path



Login

Cookie Policy

Payments

Types of Critical Paths

Smoke Tests

Regression
+
Suite

Sanity Tests

How to come up with your own list of Critical Paths

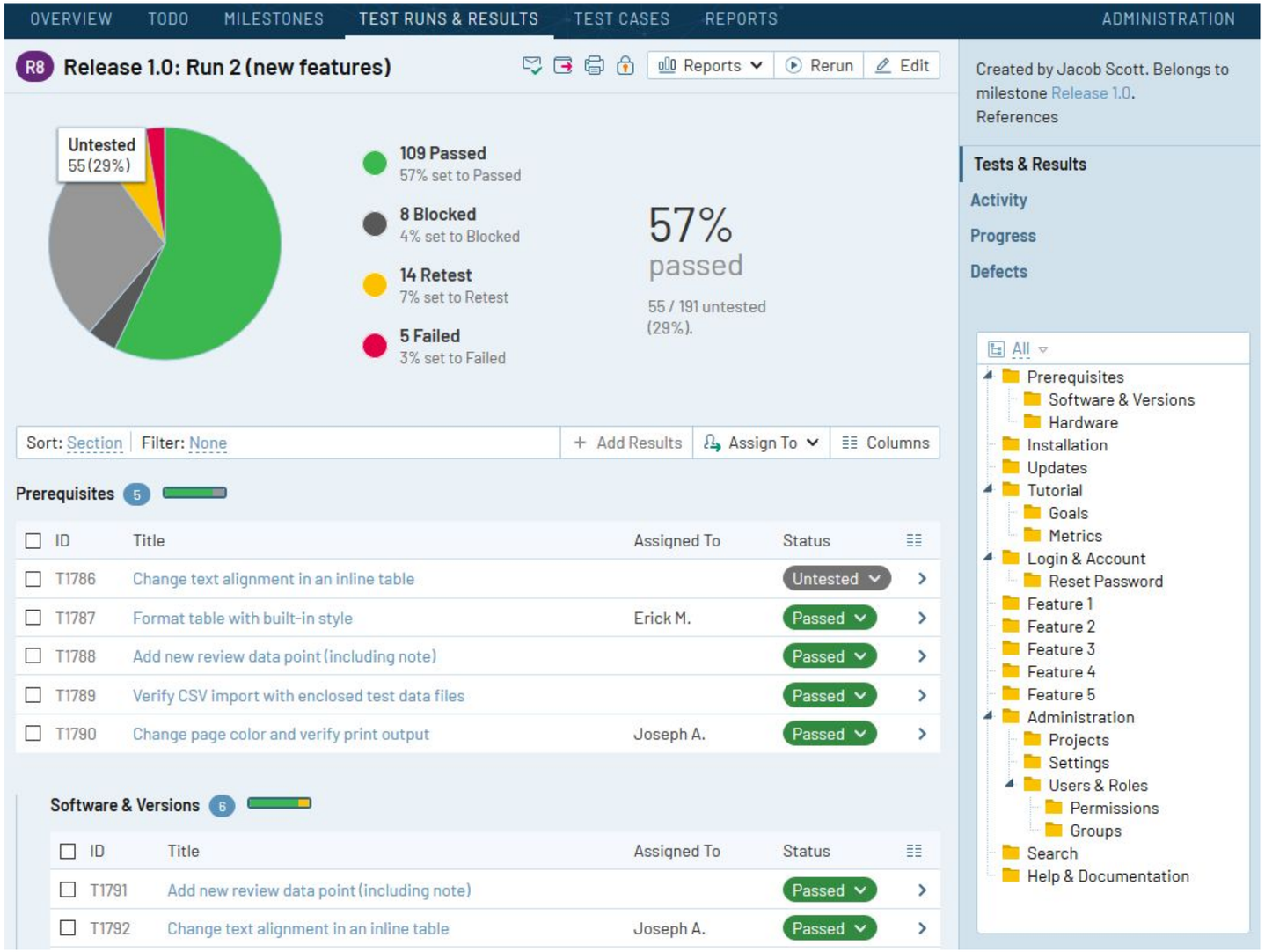
01 Pair up with someone who has deep product knowledge

02 Identify problematic areas which currently are not being covered

03 Identify areas which are crucial for the business - regulatory / fundamental requirements / features which generate most revenue

04 Structure these tests in an E2E fashion to keep them to a minimum

Document the Critical Paths



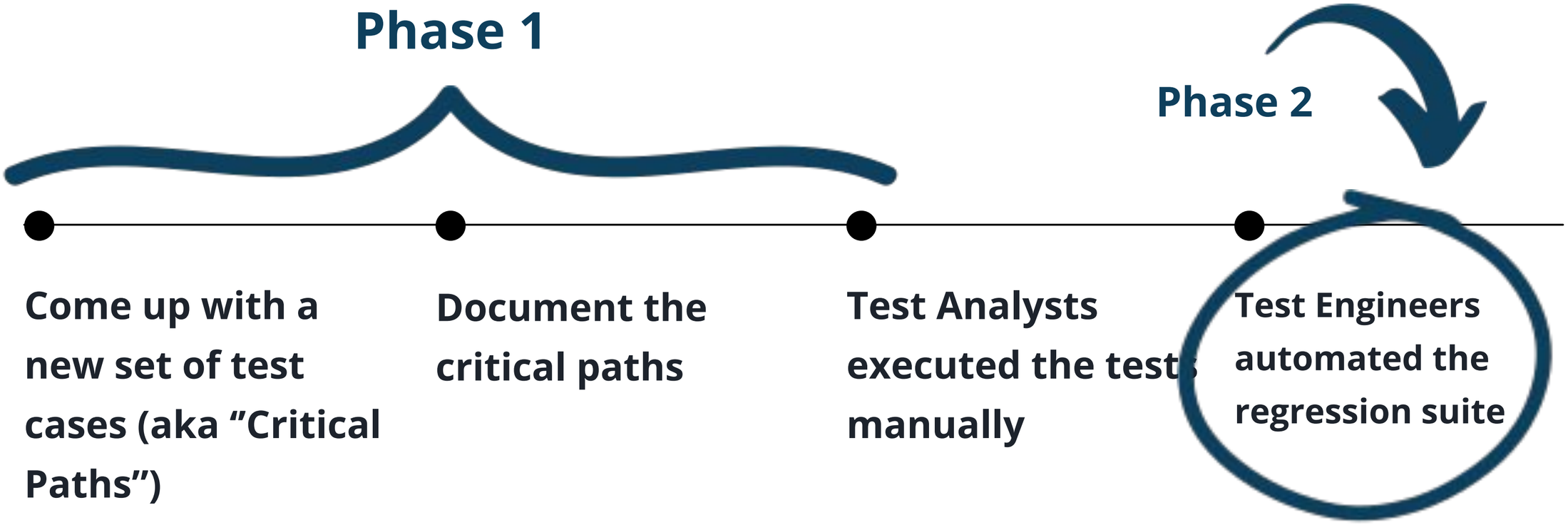
Releases (Phase 1)

Weekly Releases

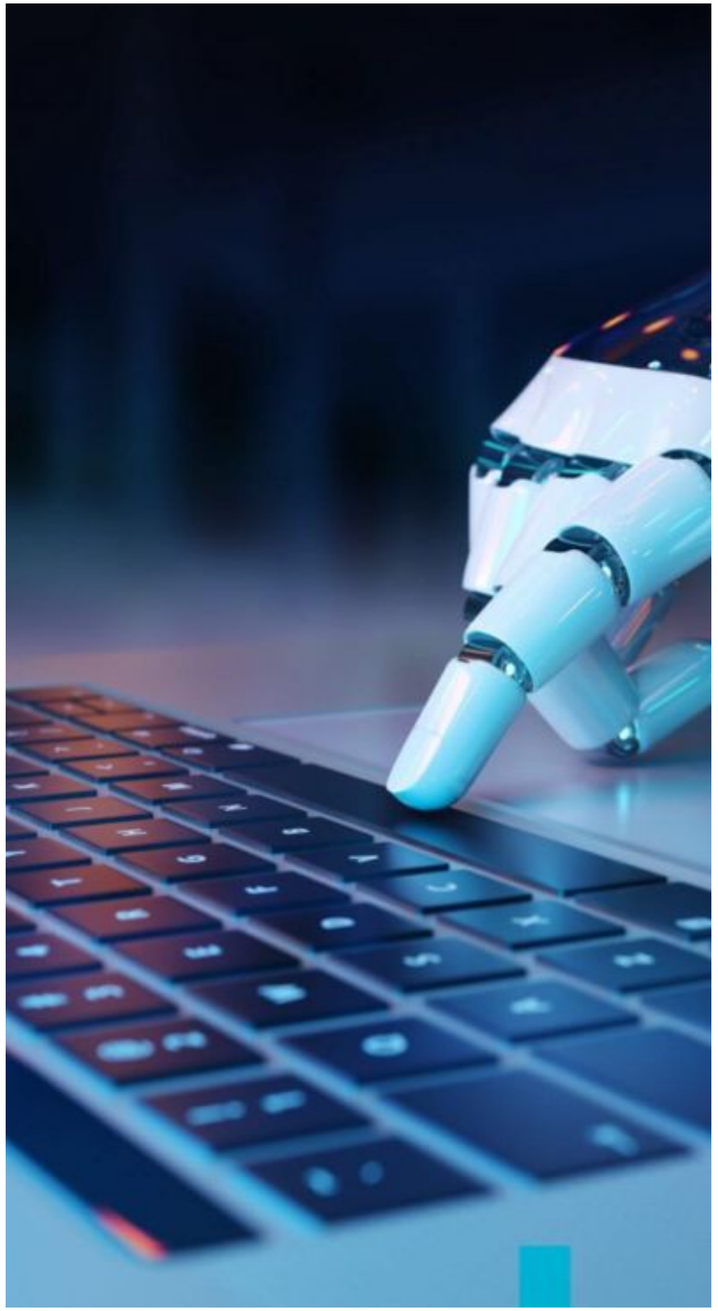
Test Analysts executed the regression tests all manually

Regression time took approximately 1 day

Implementation - Phase 2



Test Automation Implementation



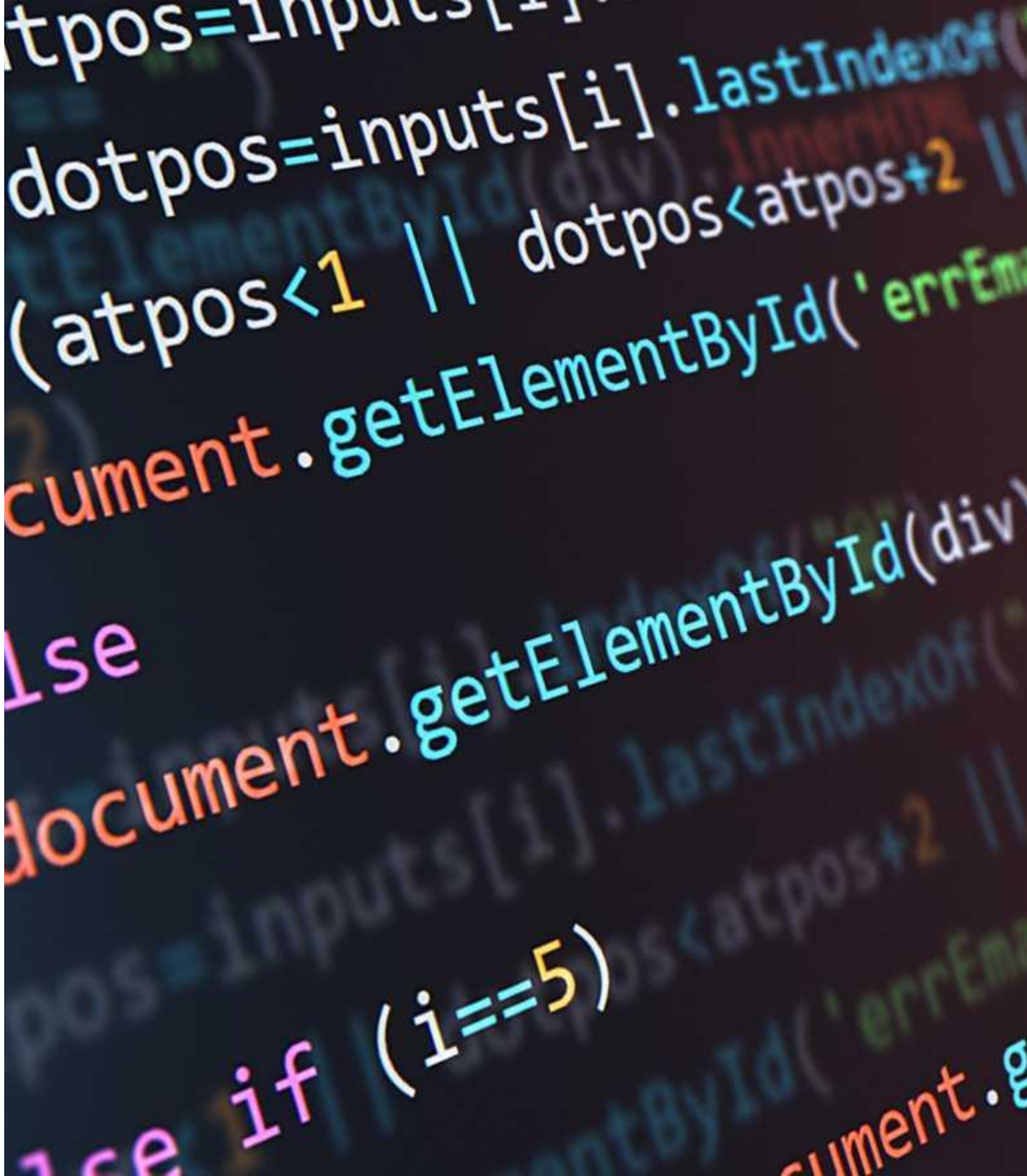
Test Automation Implementation - Challenges

The main challenge was **managing time constraints**. Hence, we implemented the following measures:

- Quality Assistance Model
- Empowered Test Engineers to push back
- 70/30 Rule

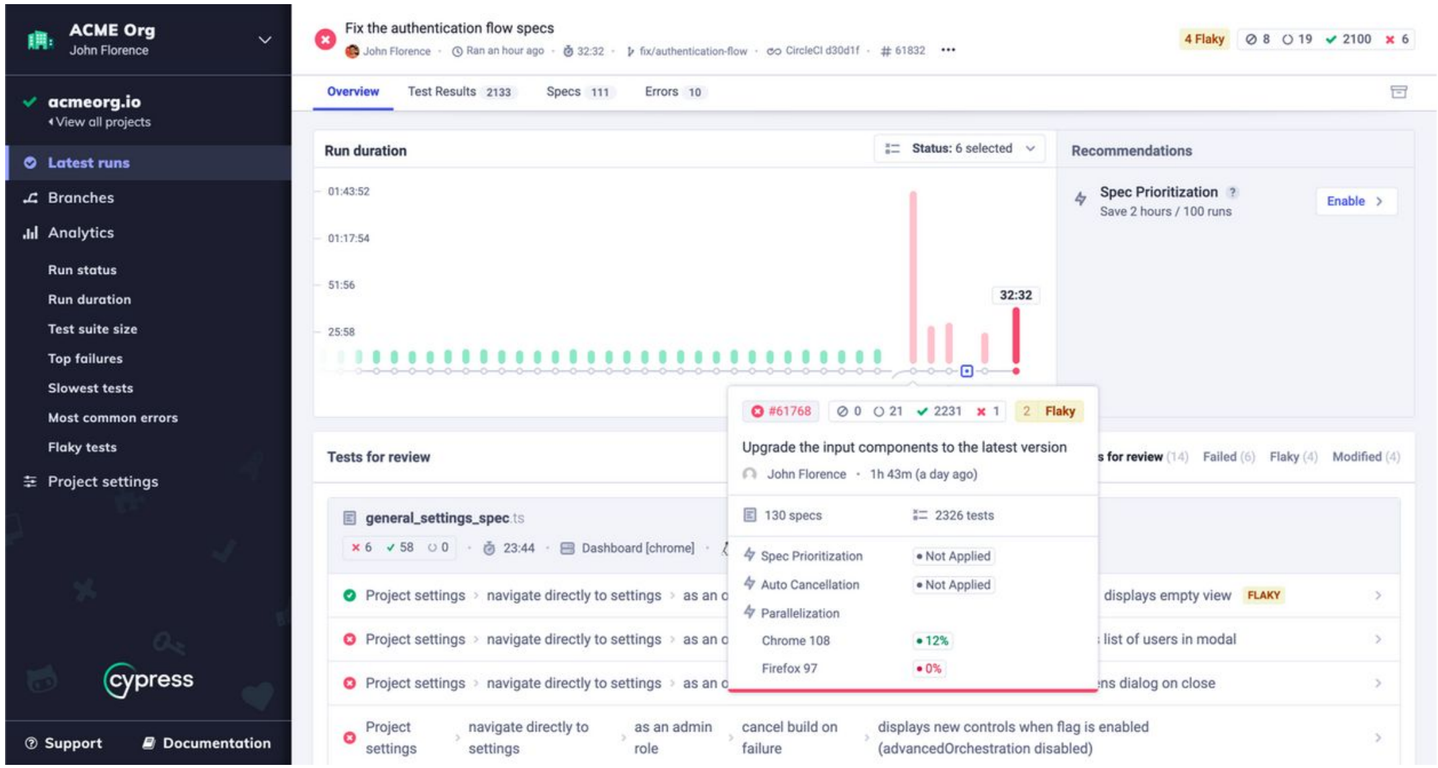
Test Automation Implementation - Tips & Tricks

- Test setup & teardown via API where possible
- Avoid code duplication
- Avoid static waits
- Maintainable locators (IDs are ideal)



Test Automation Implementation - Test Execution

- Automated Nightly Jobs
- Manual trigger before release



Releases (Phase 2)

Daily Releases

Regression was fully automated

Test Analysts executed only the failing tests

Regression time took approximately 1 hour

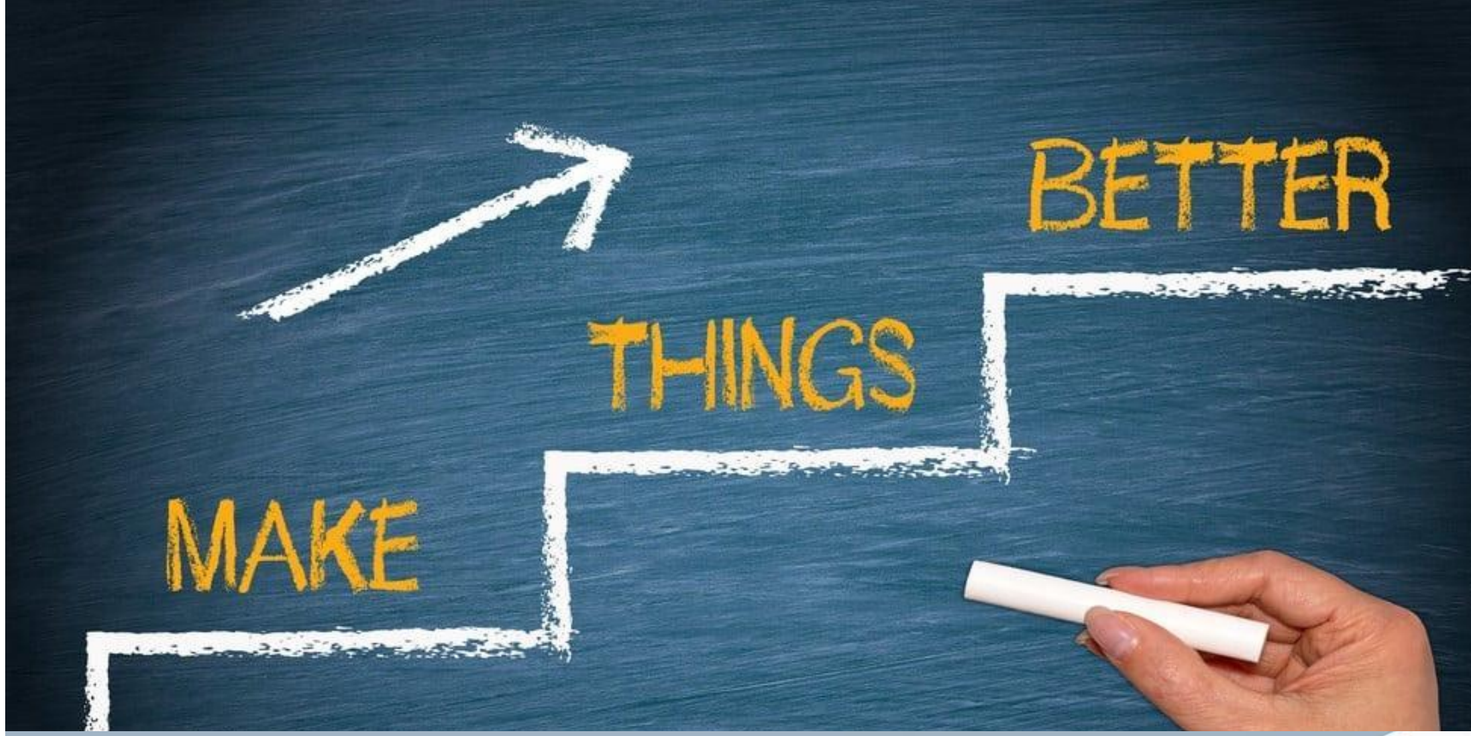
Notice any improvements?

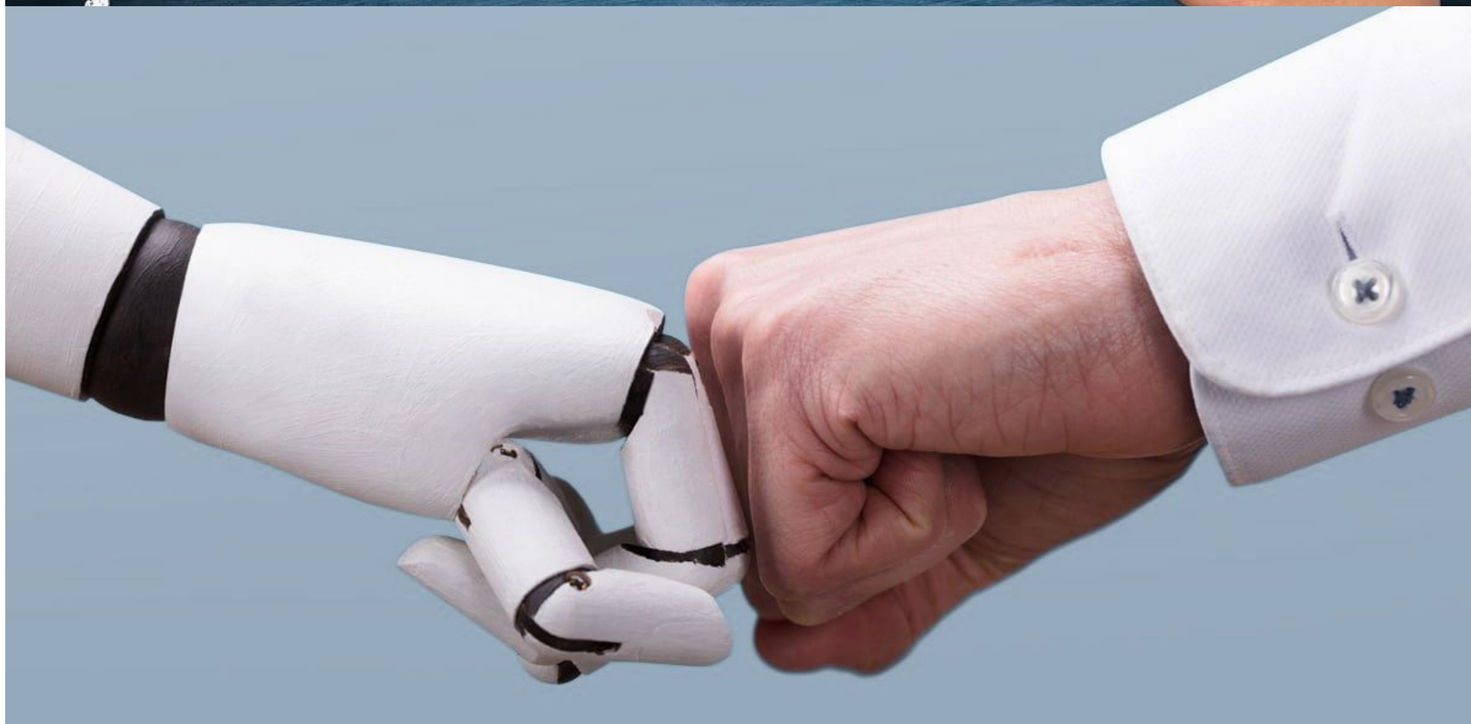
Release Frequency

Weekly to Daily

Regression Time

1 Day to 1 Hour





Testing Responsibilities - Post Improvements

Release Testing
Regression only

(Test Analysts & Test Engineers)

Manual Testing
of only Major Features

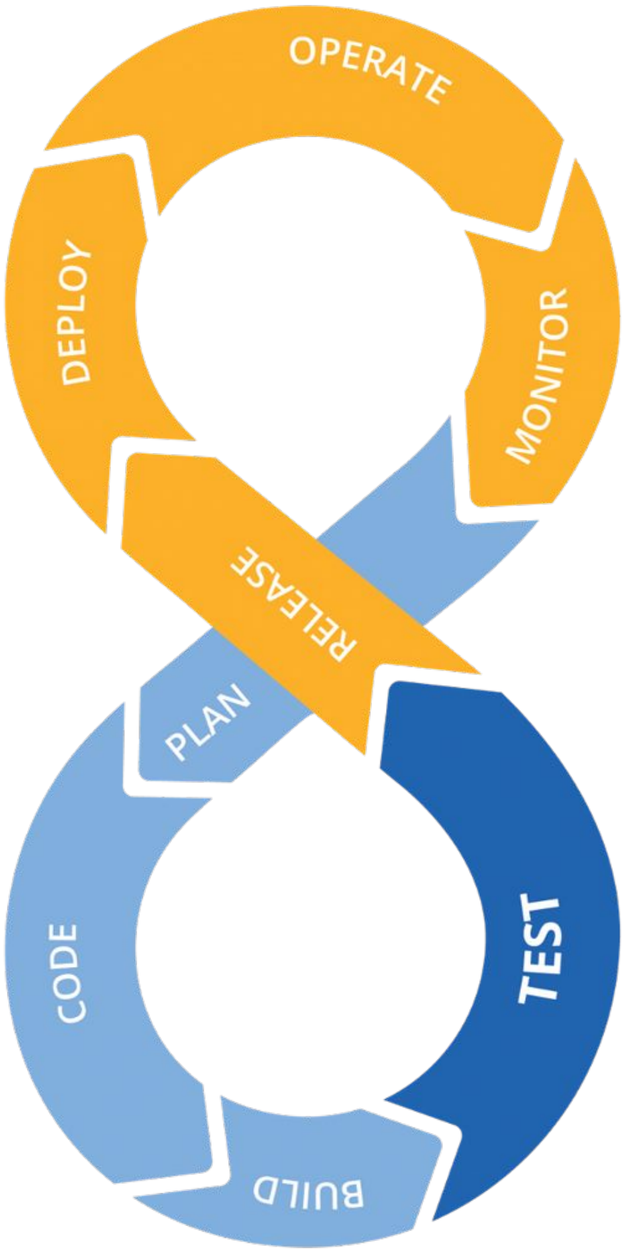
(Test Analysts & Test Engineers)

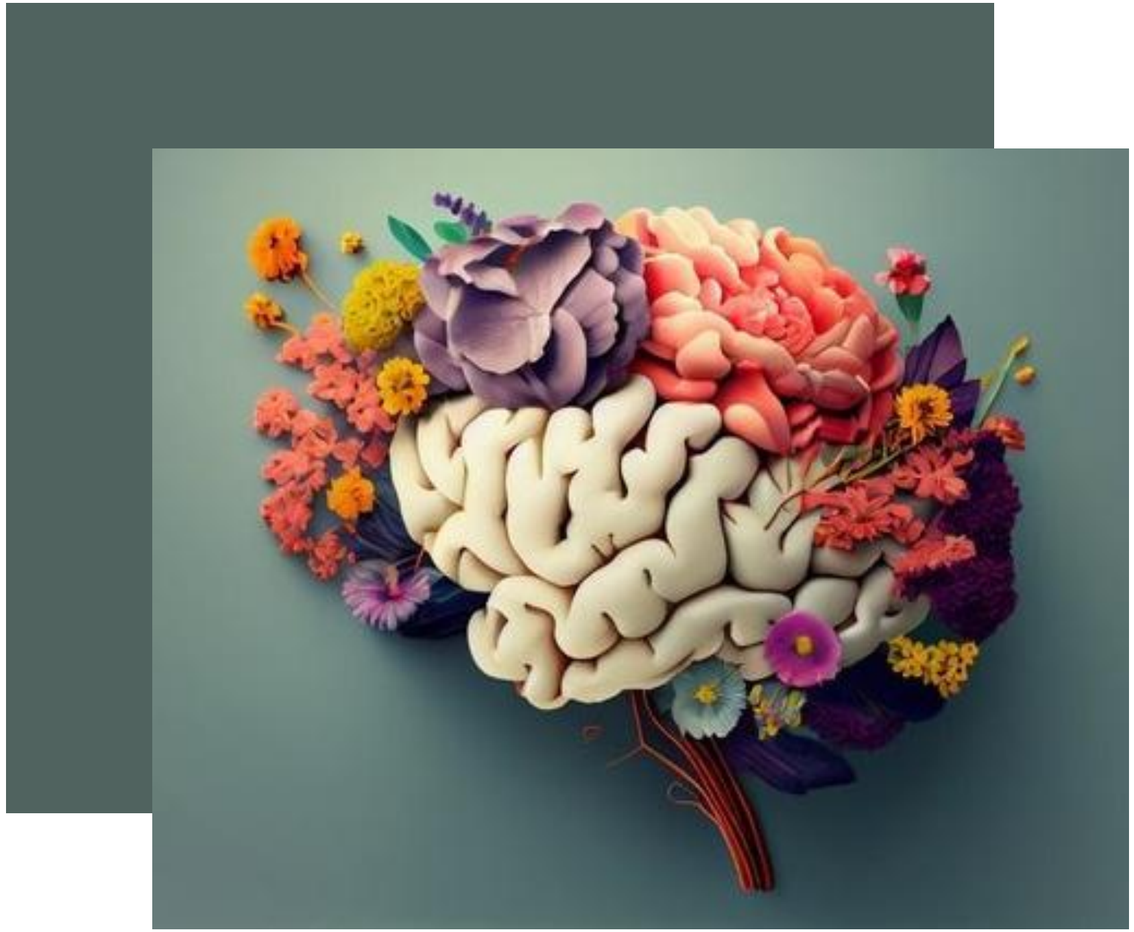
Automation
Development &
Maintenance

(Test Engineers)

A year later...

- Catching issues early
- Maintainable test suite
- Little to no test failures
- Significantly faster regression
- Daily releases
- Transitioning to CI/CD





Other Positive Impacts

- Organized work environment boosts motivation
- Reliable automated tests free up time
- More time for learning and self improvement

Future Plans

Mobile Automation

API Automation

Recap

More tests does not mean more quality

Test Automation does make a difference

An organised work environment boosts employee wellbeing

Thank you!

Questions?