



<https://twitter.com/RamiRantala>

<https://www.linkedin.com/in/ramirantala>

How to build successful Test Automation

Everyone has seen failed test automation effort

“We have tests, but they are unstable”

“We have tests, but they are not maintained”

“We have tests, but they don't test right things”

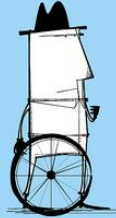
“We have tests, but they are difficult to maintain”

“We will do tests after this release”



Small story

ERRR...



CAN'T STOP.
TOO BUSY!!



First we were too busy to create tests.

We started from UI tests

We used Ruby and of course developers used Swift

Tests were flaky - for several reasons

**We hired consultant to make tests
because we were too “busy”**

Consultant left

So we had UI tests, that did not work, with a language that developers did not know, neither wanted to use

Developers wrote tests as they wanted

Tests were unstable and broken many times

We decided to create PR checks

Developers were really frustrated since it was difficult to merge PRs since UI tests were Flaky

We started to remove flaky tests from test runs

**At the same time we made decision that we will focus
on unit tests.**

Unit tests were really stable.

and we had future again

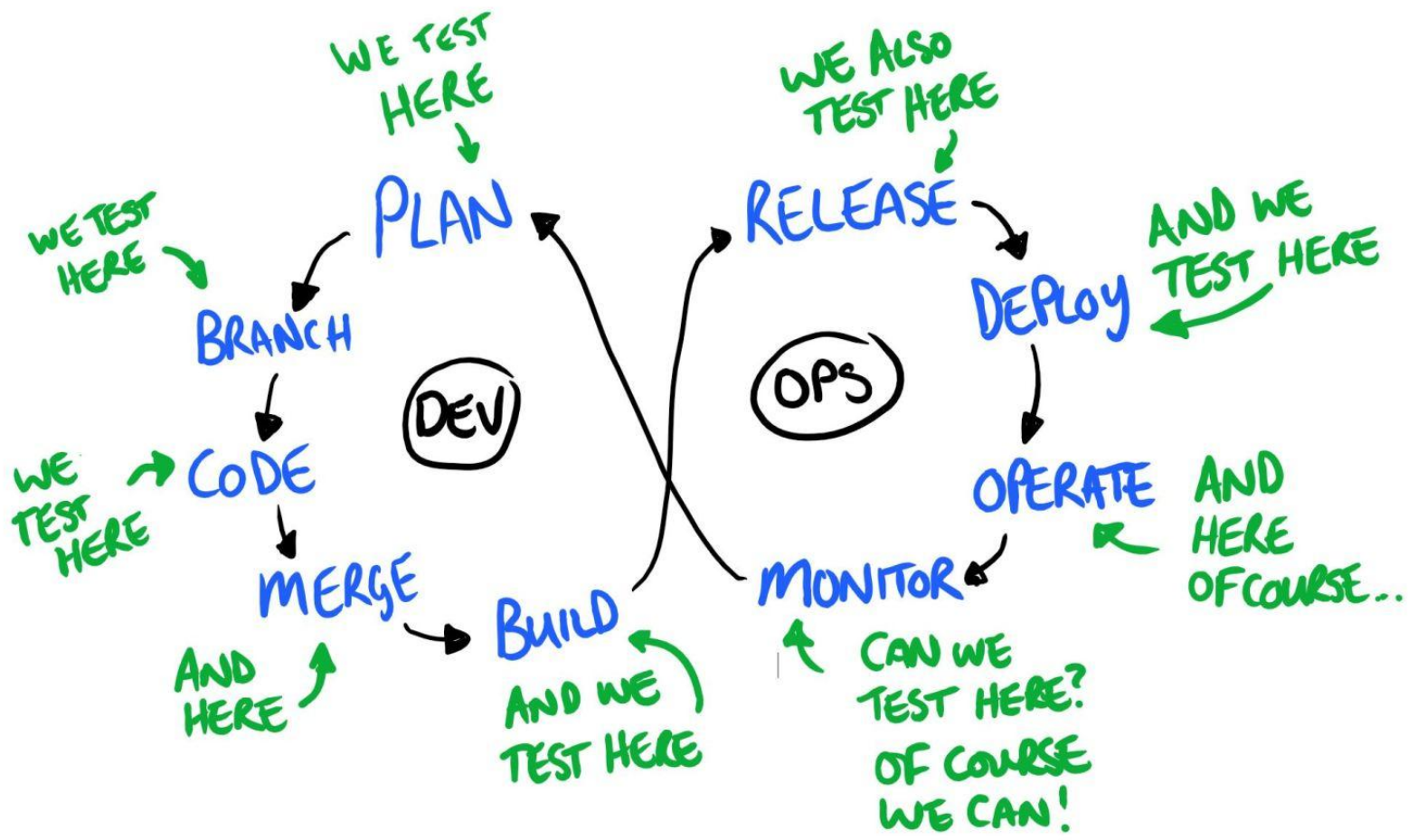
So..

The fundamental

Test Automation is not separate thing

CI/CD/DEVOPS

<https://danashby.co.uk/2016/10/19/continuous-testing-in-devops/>



Test automation is part of development

Test automation is as fundamental thing than code.

If you write code, you also write tests.

Plan and budget time for creating and fixing tests

**How much you should
automate?**

I've always said ..

“automate everything. Aim for 100%”

I might be wrong..

Forget Complete Test Automation

It is a lot of work to maintain tests.

**If you try to automate everything, it might be
that you won't get your investment back.**

Find the most critical paths that gain most value.

Any quick wins?

Test automation is investment

You won't get fast return.

You won't get benefits on current release.

**You'll get your investment back, when running
the tests on several releases**

Do small investment first

**Learn
Try things**

Maintenance

Maintenance is the killer of everything

It is better to have less tests that are easy to maintain and work than a lot of tests that are unstable

What tools you should use?

Use tools that..

are opensource

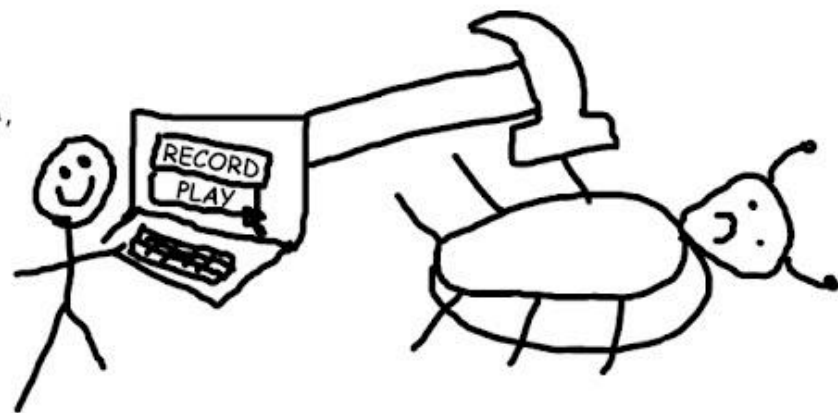
Developers are willing to use and learn

Support the language your developers use ..

**Who should do the
automation?**

AUTOMATION DON'T KILL BUGS, PEOPLE[★] DO

HA, HA,
HA!



★ DEVELOPERS MAINLY - SINCE THEY WERE THE
ONES WHO CREATED THE BUGS IN THE FIRST PLACE

AG


**The best automation results are lead by developers
who have put energy into it.**

Don't outsource test automation effort

consultants


other teams

qa department



I LIKE TO CON
PEOPLE. AND I
LIKE TO INSULT
PEOPLE.

www.dilbert.com scottadams@aol.com



IF YOU COMBINE
CON AND INSULT,
YOU GET "CONSULT."

8/24/98 © 1998 United Feature Syndicate, Inc.



I'M
HERE
TO
CONSULT
YOU.

IT SOUNDS
EXPENSIVE
AND
DEMEANING.
... OKAY.

Implement gatekeepers

“If any of the tests have failed, it will send a “Failure” status to the GitHub API which stops that specific pull request from being merged into our master branch. The second status is a master branch status check which prevents all pull requests from being merged when the master branch itself is broken.”











<http://code.hootsuite.com/continuous-integration-and-delivery-for-mobile/>



Some checks haven't completed yet

[Hide all checks](#)

1 pending and 7 successful checks

-   **zappr** — This PR needs 2 more approvals (0/2 given). Required
-   **Jenkins** — Unit tests are successful 808 tests run, 0 skipped, 0 failed. Required [Details](#)
-   **Jenkins - Build PR** — Building of PR is successful Required [Details](#)
-   **Jenkins - Smoke Tests** — Smoke tests are successful 2 tests run, 0 skippe... [Details](#)
-   **Swiftlint** — Done linting! Found 0 violations, 0 serious in 771 files. Required [Details](#)



Required statuses must pass before merging

All required [status checks](#) on this pull request must run successfully to enable automatic merging.

Merge pull request



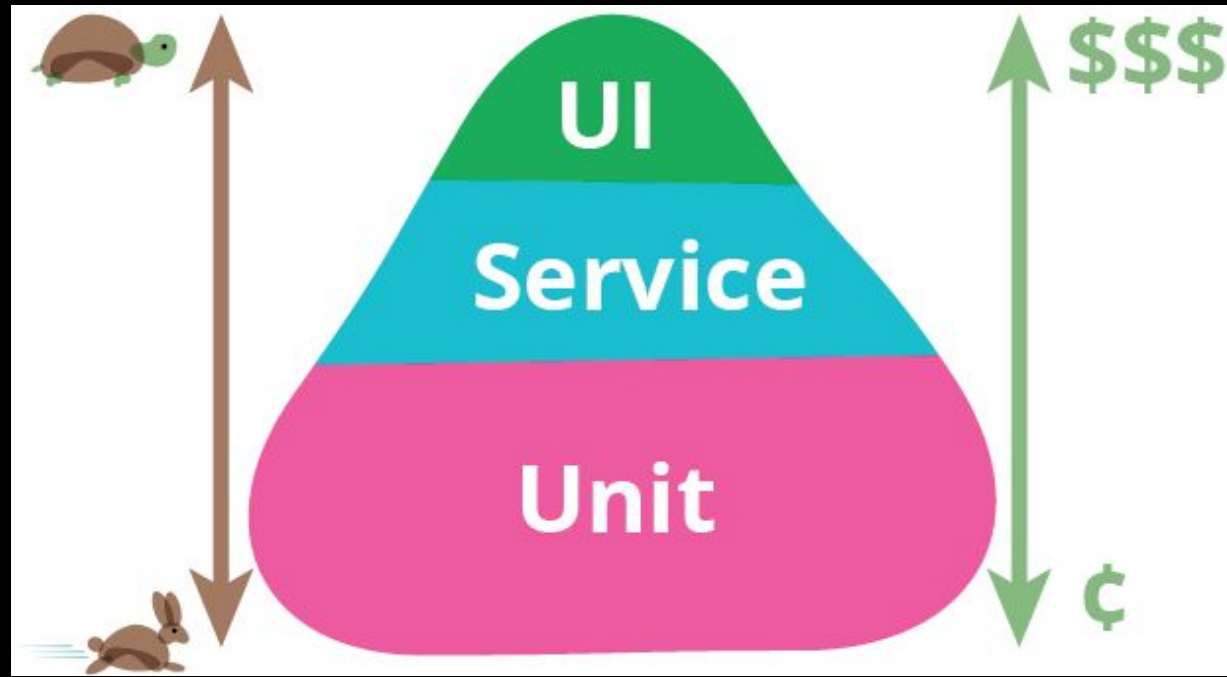
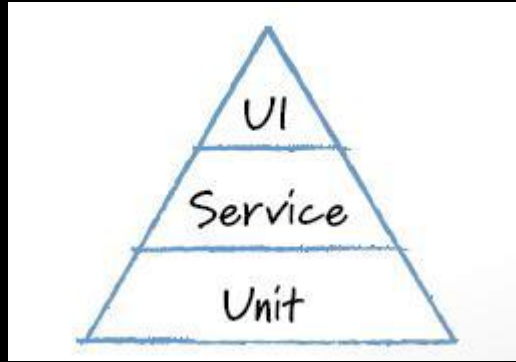
You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

UI vs. Unit?

It is tempting to start automation from UI

Good thing is that UI tests usually go end to end

Bad thing is that UI changes constantly.



<https://martinfowler.com/bliki/TestPyramid.html>

“If engineers spend 20% of their time writing new code and the rest of their time refactoring the fragile UI tests that break every time code is changed, we aren't achieving our goal.”

<https://engineering.linkedin.com/blog/2016/02/3x3--speeding-up-mobile-releases>

“We prefer an approach where product owners decide which production-critical paths require UI tests, and resolve some of the edge/negative cases with unit tests, which are much faster and easier to maintain.”

<https://engineering.linkedin.com/blog/2016/02/3x3--speeding-up-mobile-releases>

**Unit tests can be executed early and they are
precise**

Unit tests are fast to execute

Test Stability

Dev 1: “My build failed, but I didn’t change anything“

Dev 2: “Oh just re-run it. That test has been failing for everyone. It’s not your fault”

“It works on my machine”

Say NO to Flaky tests.

It is better NOT to run tests that have random success

Quarantine **flaky tests**

= **remove them from test runs and report bugs due to flakiness**

<https://testing.googleblog.com/2016/05/flaky-tests-at-google-and-how-we.html>

<https://www.thoughtworks.com/insights/blog/no-more-flaky-tests-go-team>

Investigate why tests are not stable

fix and mock - use mocks if it improves stability

Isolate test environments.

Docker is good tool to setup mini environments
that can be created even for test runs only

Test data is difficult.

**I feel that many times effort put into having perfect
test data does not pay off**

**Having one automated test case that is 100% reliable
is better than 10 which are unreliable**

Thank you

<https://twitter.com/RamiRantala>

<https://www.linkedin.com/in/ramirantala>