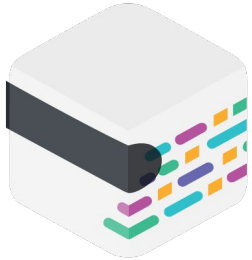
Modeling your test automation strategy

Part 3: The journey

A little about me



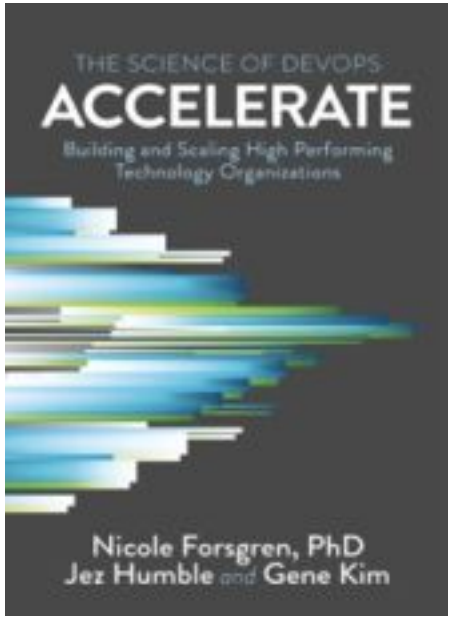
With Janet Gregory

Lisa Crispin
Testing Advocate at mabl
lisa@mabl.com

Today, we'll look at ways to:

- Implement your strategy and reduce manual regression testing
- Succeed by engaging the whole team
  - And dealing with resistance
- Choose the right tools for your situation
- Evolve your strategy based on small experiments
- Incorporate test suites into your pipeline

A reminder of why we automate

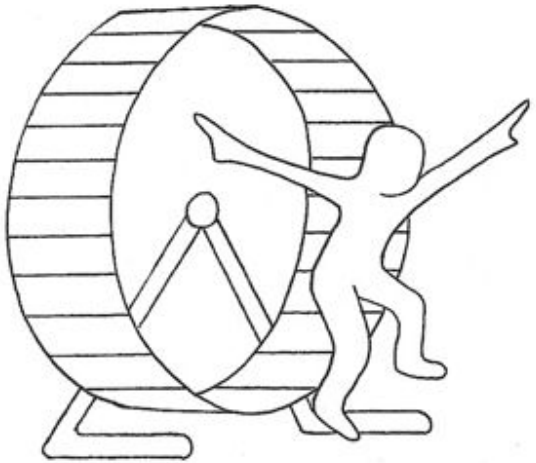


State of DevOps Survey

Predictors of high-performing teams:

- Reliable automated tests run every commit
- Developers create, maintain automated tests
  - With help from testers
- Testers do exploratory, other types of testing

You can get off the hamster wheel



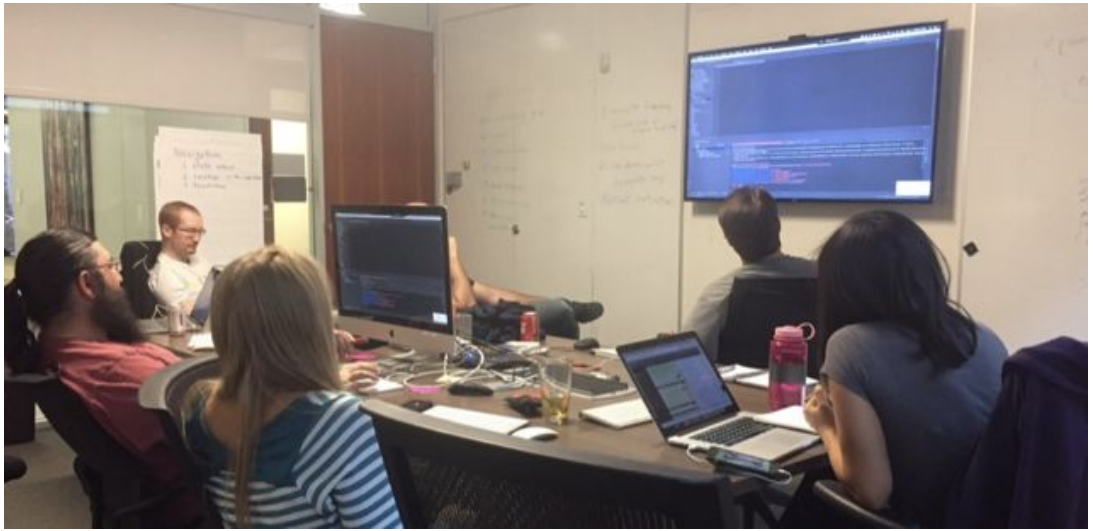
- You've used visual models to have good conversations
- Your strategy engages the whole team
- You've identified high-risk and high-value areas of your product

How to get going with your strategy?

Photo by Jaromír Kavan on Unsplash

Keep the whole-team engagement

- Share the manual regression testing pain
- Set goals and do small experiments
- Frequent retrospectives & adjustments

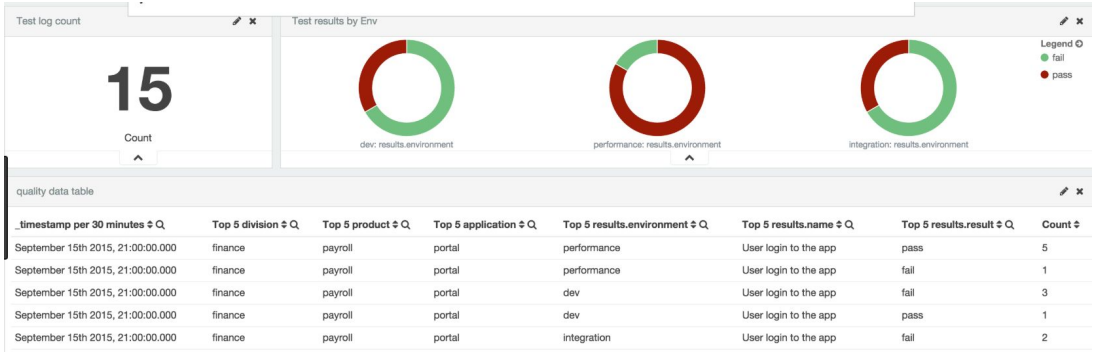


When you face resistance

- Listen
- Can they agree to a short experiment?
- What’s in it for them - offer them something
- Ask them for help

Keep your progress visible

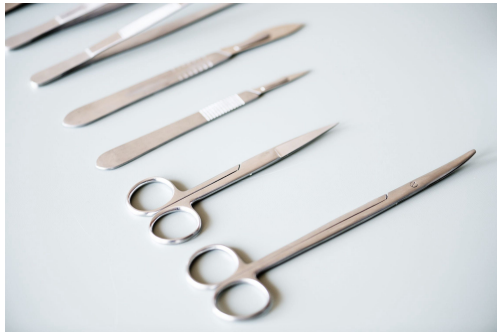
- Big wall monitors with CI results, Slack notifications
  - Fast feedback
- Goals, milestones, progress charts on the wall
- Shared dashboards for distributed teams



_timestamp per 30 minutes	Top 5 division	Top 5 product	Top 5 application	Top 5 results.environment	Top 5 results.name	Top 5 results.result	Count
September 15th 2015, 21:00:00.000	finance	payroll	portal	performance	User login to the app	pass	5
September 15th 2015, 21:00:00.000	finance	payroll	portal	performance	User login to the app	fail	1
September 15th 2015, 21:00:00.000	finance	payroll	portal	dev	User login to the app	fail	3
September 15th 2015, 21:00:00.000	finance	payroll	portal	dev	User login to the app	pass	1
September 15th 2015, 21:00:00.000	finance	payroll	portal	integration	User login to the app	fail	2

Building your tool box - at each level

- How do you want your tests to look?
- Skill sets of the people writing, using the tests
- Budget time to research and try tools
- Gather your “requirements”, constraints



Tests that work in your context

- Tests as living documentation
- Test creators feel comfortable writing

Scenario: Timeslot is not available

Given Cindy has viewed available timeslots for the service option she wants

When she *selects a timeslot that has a conflict*

Then she sees a helpful message

```
describe "user books an appointment" do
  It "returns a helpful error message for unavailable time slot"
    given_available_timeslots_viewed
    when_cindy_chooses_unavailabile_timeslot
    then_she_sees_error_message
  end
```

booking.checkTimeslotFixture (day, timeslot, option, size)						
Day	Timeslot	Available	Option	Size	Can book?	Message
Mon	11:00 – 12:30	Yes	Full Clip	Miniature	yes	Booking is confirmed

Example tool requirements

- Support collaboration of testers, business & devs
- Uses same language as our app under test
- Support product for at least the next 5 years
- Robust result reporting, easy to analyze failures
- Easy to integrate with our existing continuous integration

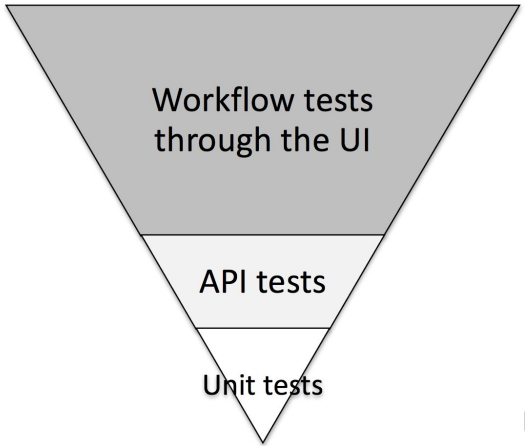
Other considerations

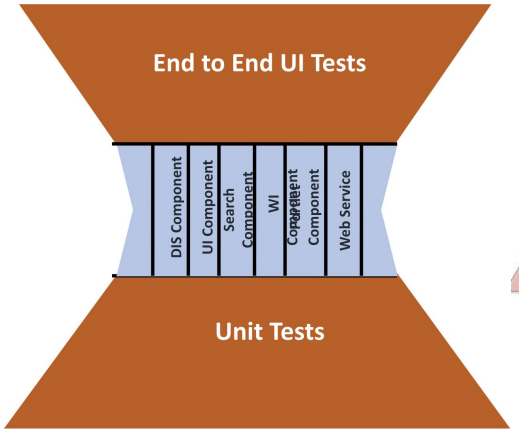
- Microservices vs monolith architecture
- Support exploratory, other testing activities
- Data-driven tests, using tables
- Keyword-driven tests
- Legacy code
- Workflow

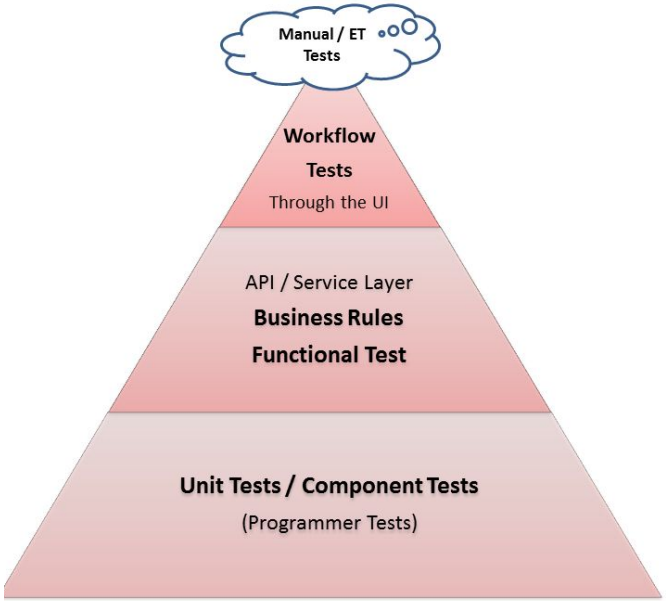
Example of whole-team tool selection

- Sys admin spiked on driver library
- Testers, devs and sys admin did two rounds of “bake-offs” between two frameworks
- Found solution that worked for everyone
- Then got training to get up to speed fast

Transitioning with models







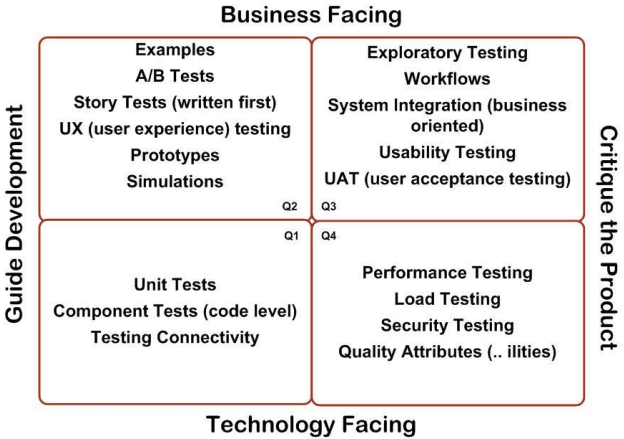
Evolve them over time...

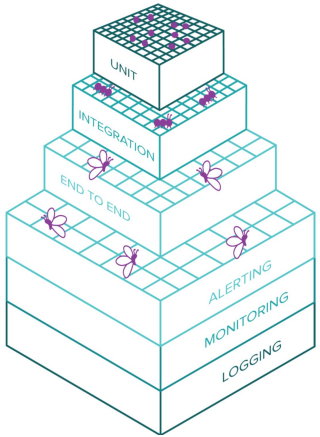
Start simple

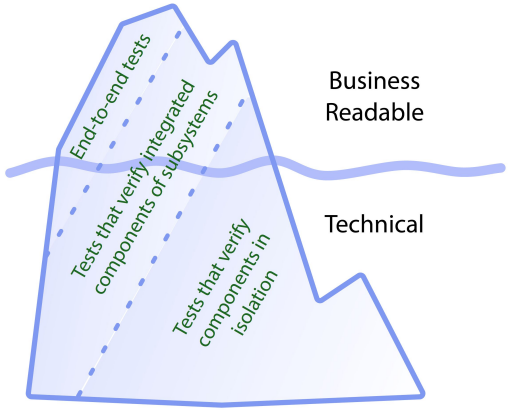
- Automate tests for the highest-risk area
- Or go for easy wins
- “How will we automate tests for it?”
- Small, iterative increments - use feedback

Experiment!

If your initial strategy isn't moving you to your goal, try another model







Check Parts 1 and 2 of this series for more

Collaborate

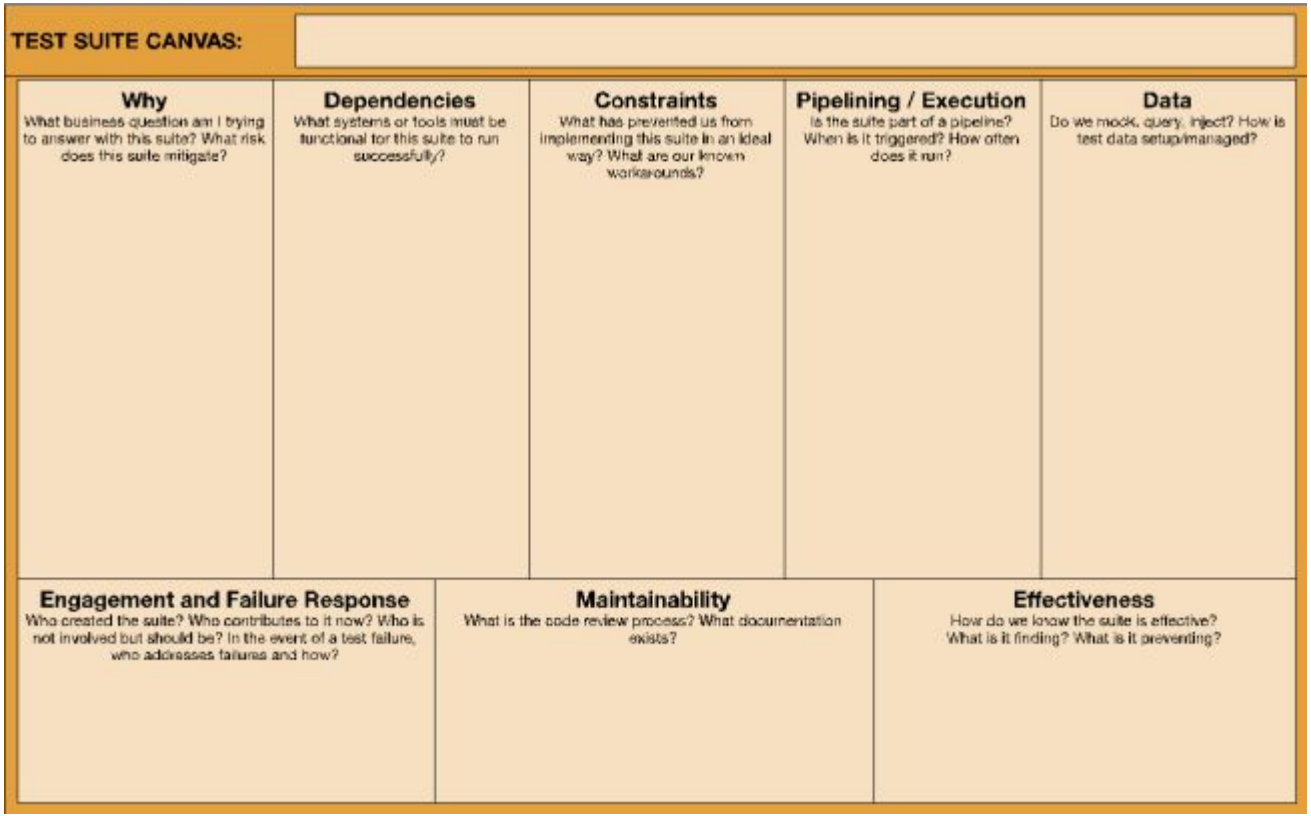
- Pair and mob to learn new skills
- Take time for learning
- Bring in specialists if needed

Treat test code like production code

- Good heuristics practices, patterns, principles
- Minimal time to maintain tests, analyze failures
- Refactor as necessary

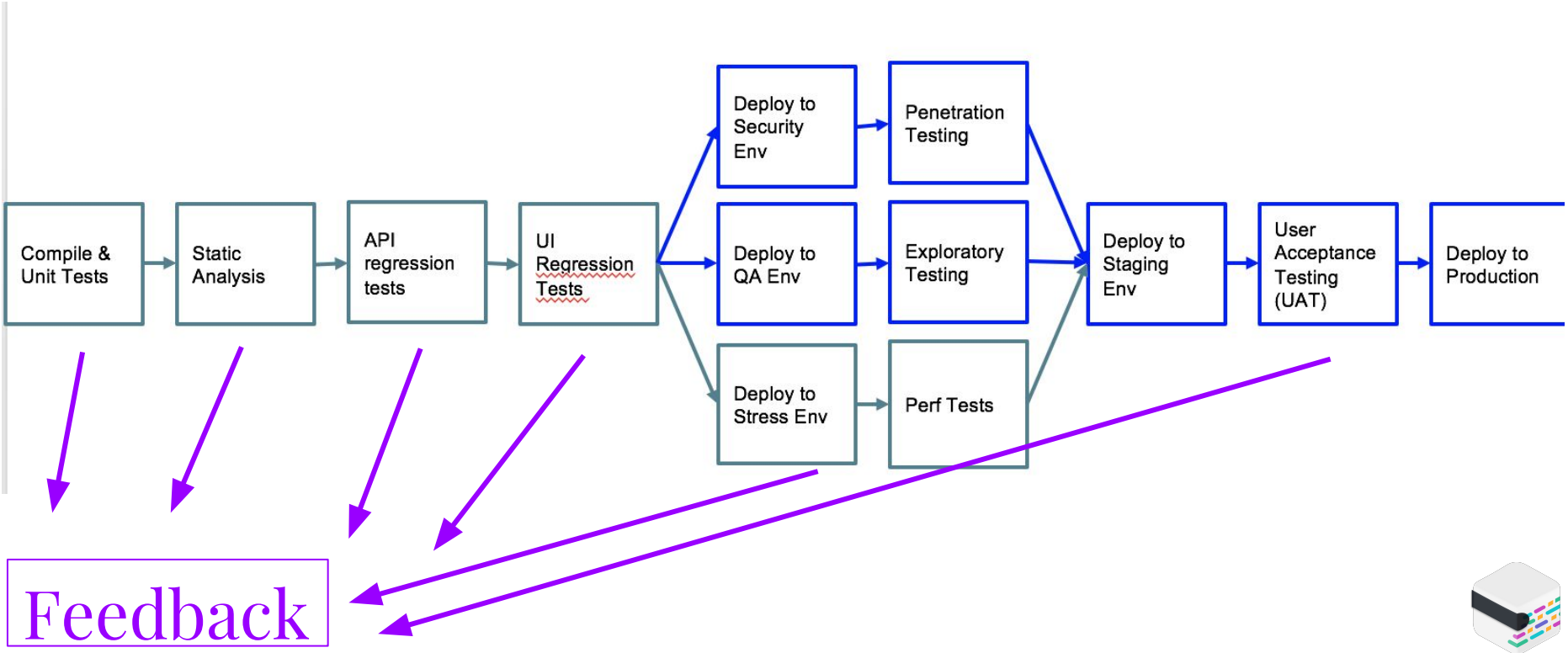
See resources at end

Think through each test suite



See https://github.com/ahunsberger/testSuiteDesign

Add test suites to your CI pipeline



Celebrate every small success

Ring a bell, go for ice cream, play some ping pong

- For successful pipeline, merging PRs, whatever was hard in the past
- Example: Company party for 1000 JUnit tests
  - With explanation of why that is valuable

Visual models

- Enable conversations in your context
- Help you track progress over time and stay on track
- Help focus automation on the most valuable areas

Your test automation strategy helps you:

- Overcome the many obstacles to success
- Build confidence to release small changes frequently
- Automate the boring stuff & free up time for the valuable testing activities

There’s a lot packed into these 3 webinars

- Set a goal ---> biggest pain point
- If models in Parts 1 and 2 don’t look useful for your context, design your own
- Visualize why you are automating, who can help, how can they help, what are your action items
- Do small experiments, track progress, readjust frequently

Questions? Stories to share?

Resources for more learning

- "Modeling your test automation strategy", parts 1 and 2, https://www.mabl.com/blog/modeling-your-test-automation-strategy-webinar, https://www.mabl.com/blog/modeling-your-test-automation-strategy-webinar-2
- "Test automation: Five questions leading to five heuristics", Joep Shuurkes, https://testingcurve.wordpress.com/2015/03/24/test-automation-five-questions-leading-to-five-heuristics/
- "Powerful test automation practices", parts 1 and 2, Lisa Crispin and Steve Vance, https://www.mabl.com/blog/powerful-test-automation-practices-pt-1, https://www.mabl.com/blog/powerful-test-automation-practices-pt-2
- "Test Suite Design", Ashley Hunsberger, https://github.com/ahunsberger/testSuiteDesign
- *Accelerate: The Science of Lean and DevOps*, Nicole Forsgren et al, https://itrevolution.com/book/accelerate/
- *Agile Testing: A Practical Guide for Testers and Agile Teams*, and *More Agile Testing: Learning Journeys for the Whole Team*, Lisa Crispin and Janet Gregory, https://agiletester.ca
- "Analyzing automated test failures", https://www.mabl.com/blog/lisa-webinar-analyzing-automated-ui-test-failures
- *Strong-Style Pair Programming* and *Mob Programming Guidebook* by Maaret Pyhäjärvi, https://leanpub.com/u/maaretp