FMG
Advice & Insurance

# Migration off Selenium

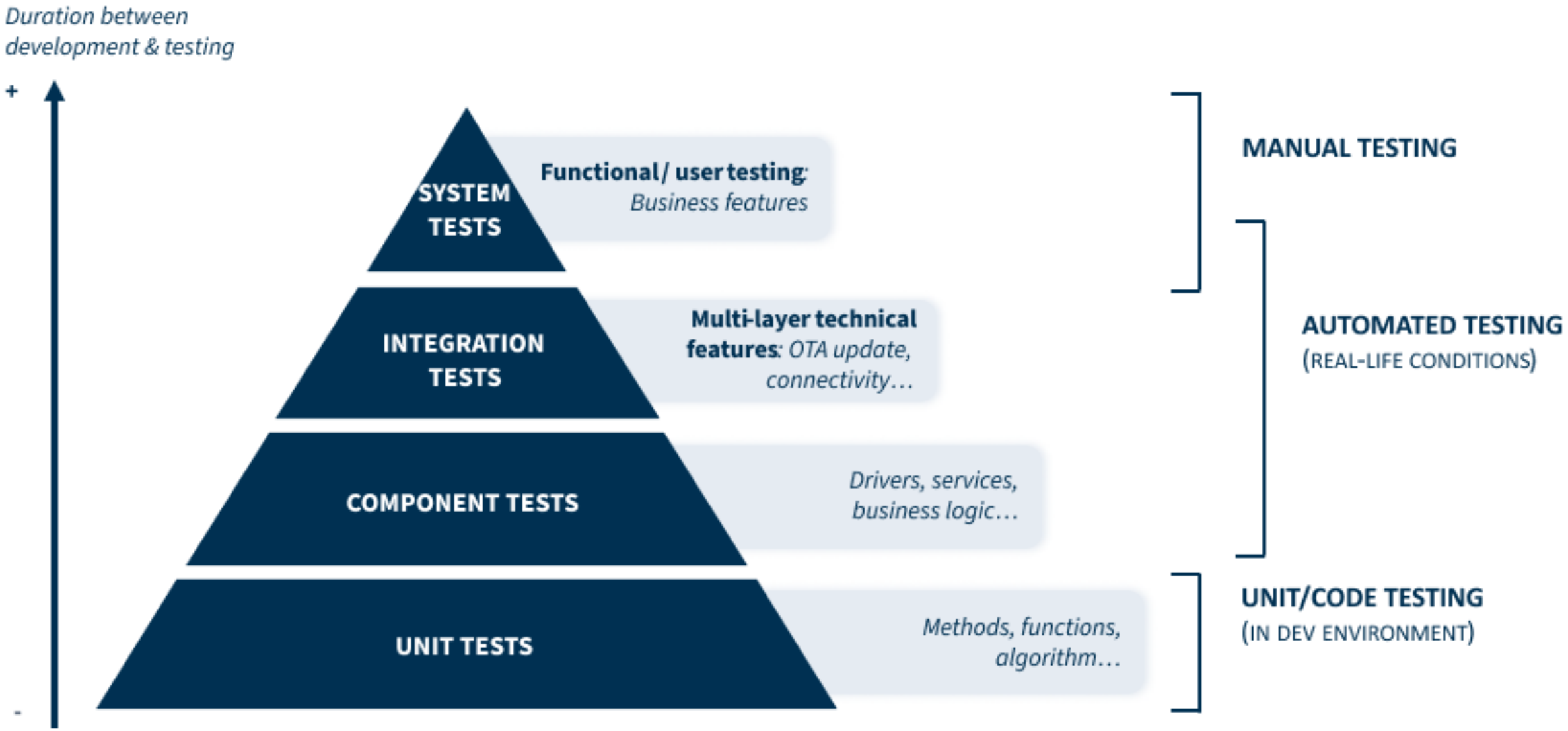
From Selenium to Playwright and the move towards the testing pyramid.

# Starting point

- 300 end to end tests
- 1 mock tests or component tests
- Low coverage of unit tests
- 2 days to do release regression testing
- Not a stable test set, so there was a lot of rerunning tests that were failing.
- Selenium based framework and infrastructure that required a lot of maintenance

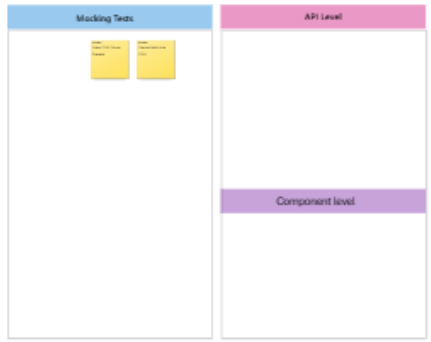
# Goal – dream state

- 10 or less end to end tests
- Majority mock(wiremock) and component(storybook) tests
- 80% coverage in unit tests
- Under an hour to do release testing
- Stable test set, so that any failures are a true reflection of a failing test
- Playwright based framework that is easy to maintain and is adaptable


Duration between development & testing
+
-
SYSTEM TESTS
Functional / user testing: Business features
INTEGRATION TESTS
Multi-layer technical features: OTA update, connectivity…
COMPONENT TESTS
Drivers, services, business logic…
UNIT TESTS
Methods, functions, algorithm…
MANUAL TESTING
AUTOMATED TESTING
(REAL-LIFE CONDITIONS)
UNIT/CODE TESTING
(IN DEV ENVIRONMENT)

# What we did

- We split our application into 6 relevant sections (flows). End to End, Add, Edit, Claims, Remove, and Misc
- We then did a risk assessment on the current tests that we had. We mapped them out to what level we wanted them to be tested at.
- Then we thought of any tests that we would like to have and added them into the table diagram







Framework Estimate: 1 sprint
Github actions Estimate:

E2E Test creation Estimate: 2 sprints

Wire mock upskilling: 1/2 sprint

Rough claims estimate: 3 sprints

Rough edit estimate: 2 sprints

Rough add estimate 2 sprints

Rough remove estimate: 1 sprint

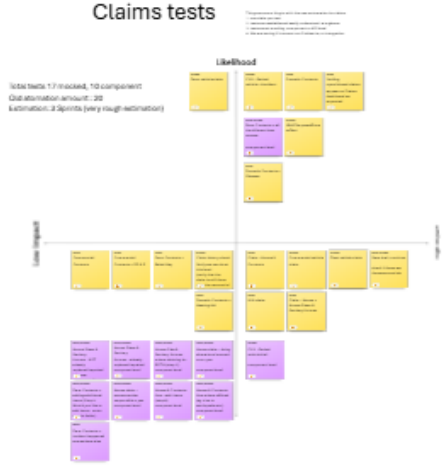

Edit tests

Remove tests


Misc Tests




Claims tests



Add tests



Total unique automation tests 96

# How we did it

- Created the end to end tests first to ensure happy path coverage
- Created the stories for the mock/component tests in Jira and did estimations based on the first story that was completed.
- Built a plan on how to create the mock/component tests.
- Passed along to our test automation engineers to work on it.

# Current State

- 13 end to end tests
- 100 mock and component tests
- 47% Unit test coverage
- Release testing takes 16 minutes to run
- Stable test set
- Playwright based framework that is easy to maintain and is adaptable