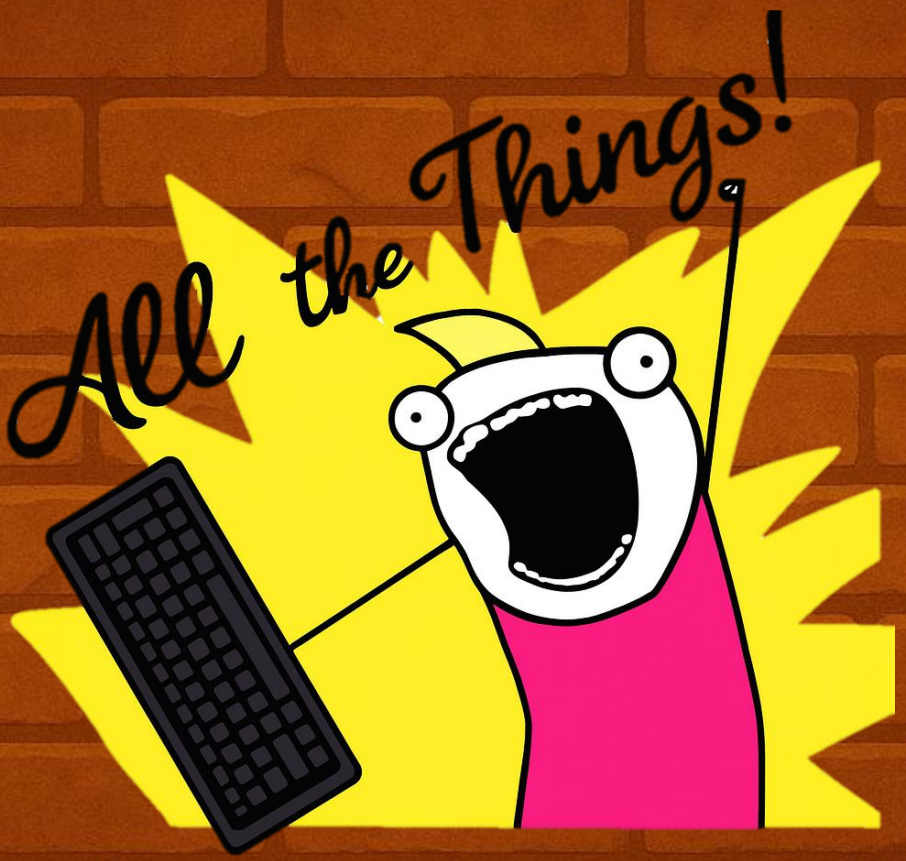
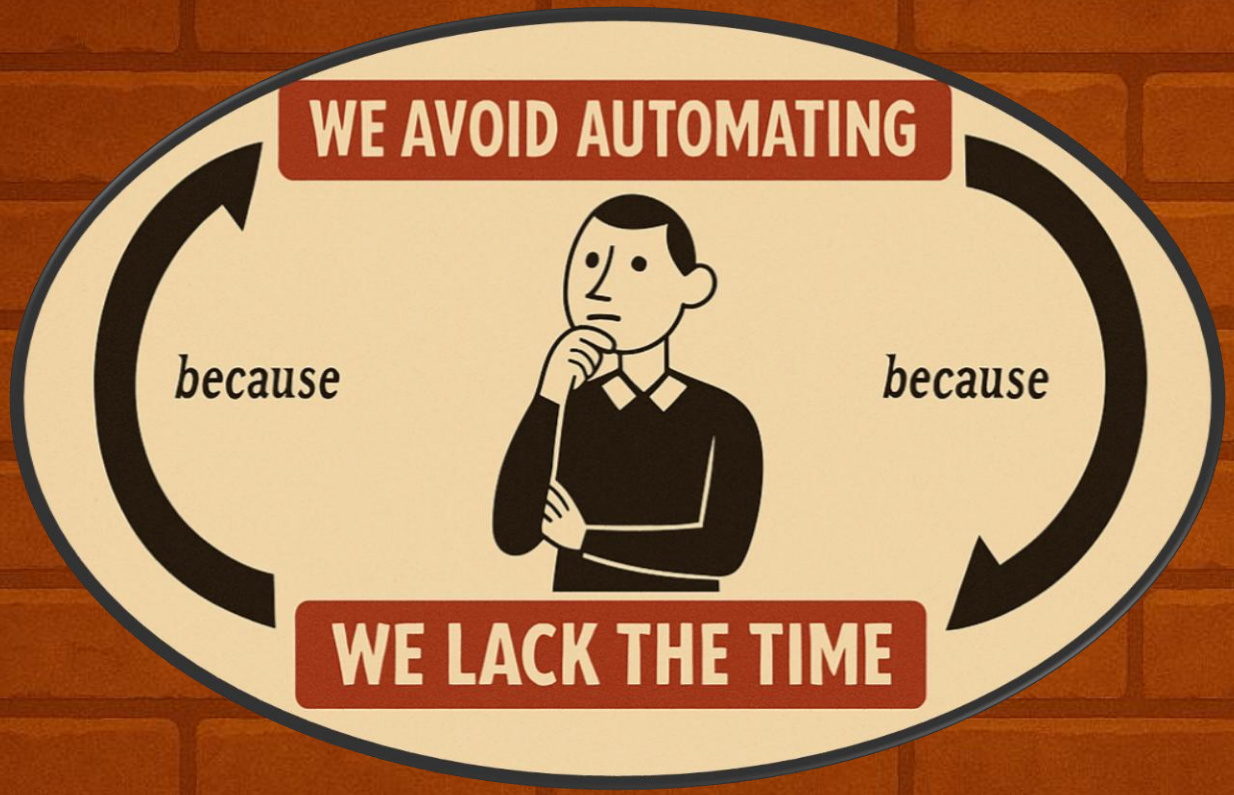


Testing All The things

By Danny Hearit

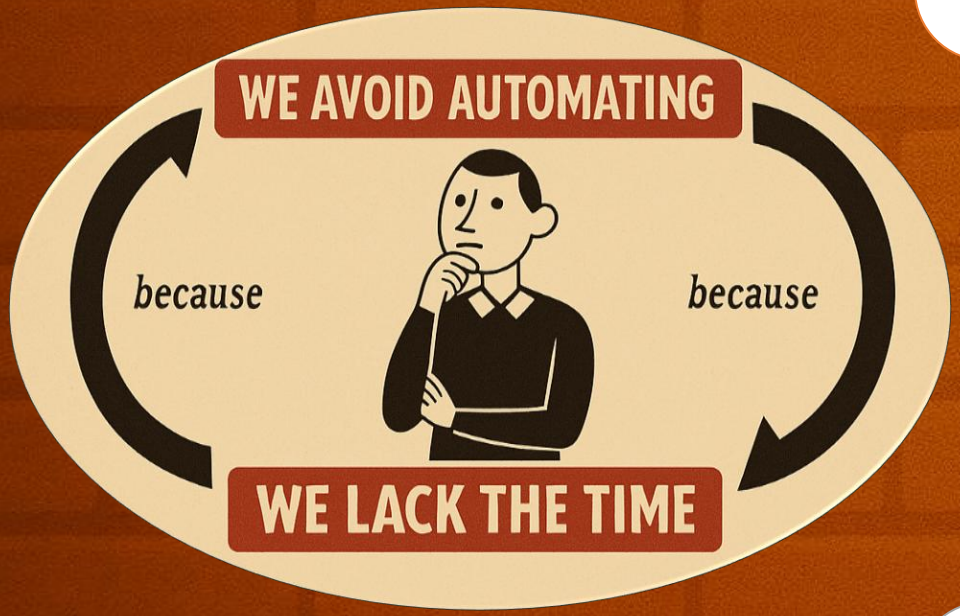
Agenda

- Goals Of Automated Testing
- Testing Types
- Legacy Vs Modern Testing Pyramid
- Increasing Confidence
- E2E On Local - Demo
- E2E On CI - Demo
- Getting Started On Your Team
- Tips And Best Practices
- Q&A



Goals of Automated Testing

Goals of Automated Testing



- Supplement and reduce manual testing efforts
- Less bugs that the make it up to testers and clients
- Faster feedback loops to devs to something is broken
- Tests that scale and are reliable
- Resilient automated testing that does not break with changes

Testing Types

Unit Tests

What is it?

- Unit == Method

What to test?

- Input/output – Not implementation
- Boundaries/good/bad combinations of arguments
- Expected errors/exceptions (and their messages)
- New features or issues

What techniques or tools commonly used?

- Frameworks – Junit, xUnit, Jasmine
- Mocking/stubbing – If needed

Integration Tests

What is it?

- Integration == Interactions between modules and outside of the system

What to test?

- Different parts working together
- External resources
  - Databases, webservers, vendors

What techniques or tools commonly used?

- Scripting
- Functional unit testing
- Test accounts for vendors/external APIs

Component Tests

What is it?

- Component == Service or web components

What to test?

- Aggregation of other calls
- Higher-level operation/abstraction
- Configuration (Angular or not)

What techniques or tools commonly used?

- Angular component testing (Cypress, Angular Unit Tests)
- API/Contract testing

E2E Tests

What is it?

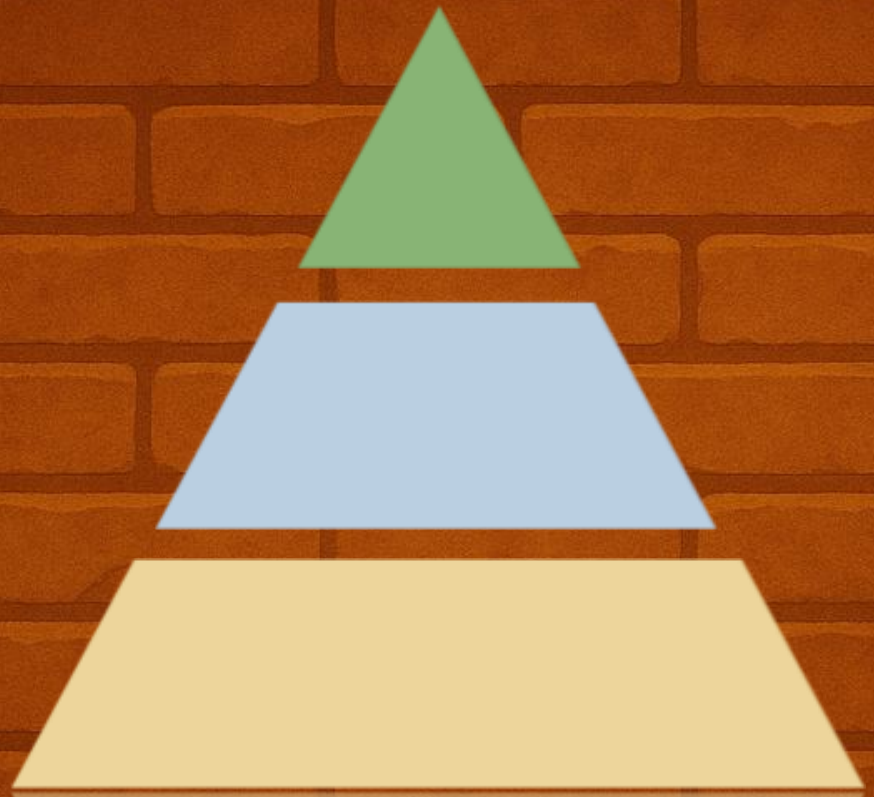
- E2E == User experiences from start to finish

What to test?

- Workflow
- Security
- Happy path
- API responsiveness

What techniques or tools commonly used?

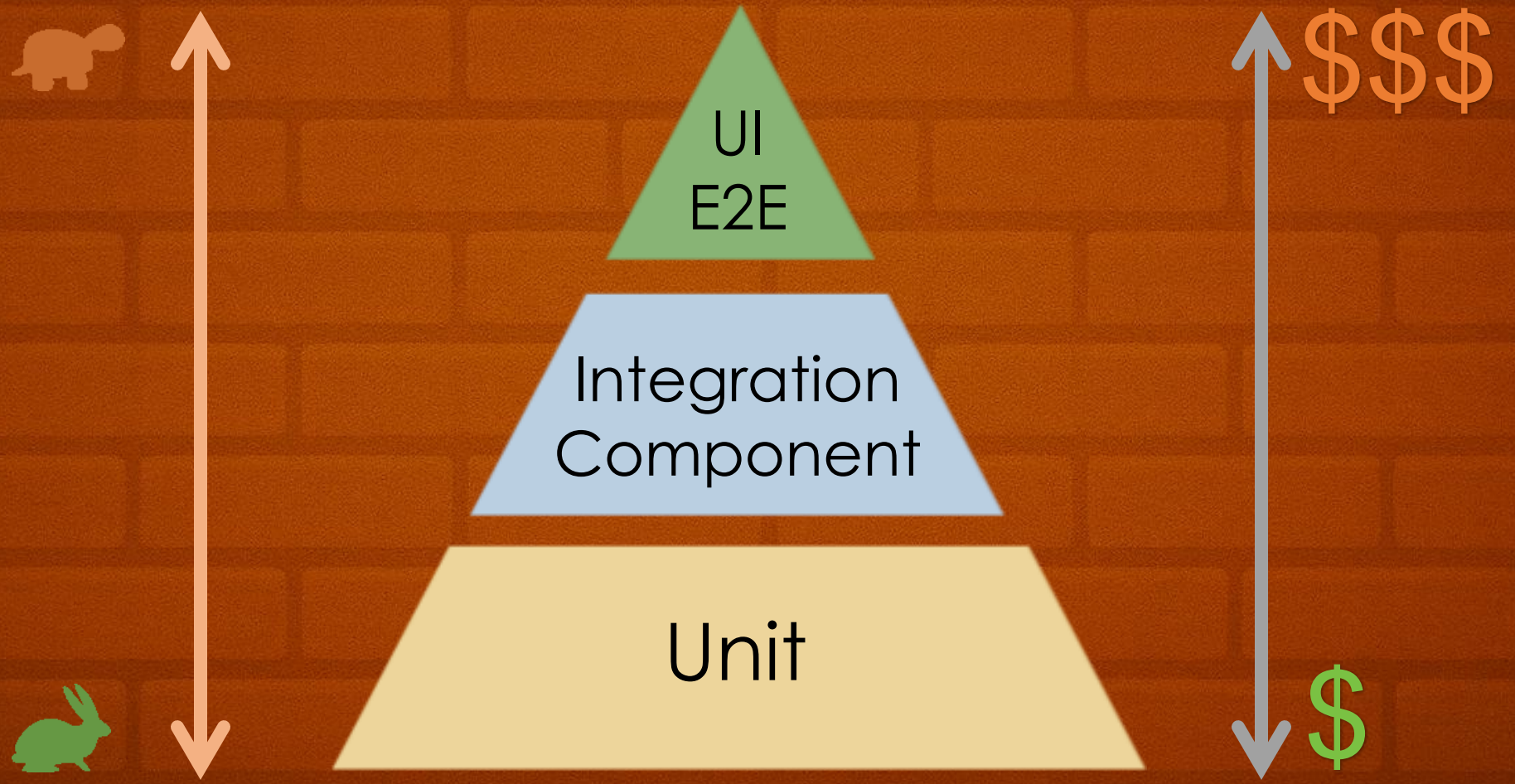
- Cypress, Playwright
- Supplementing manual regression testing



Testing Pyramid

Legacy

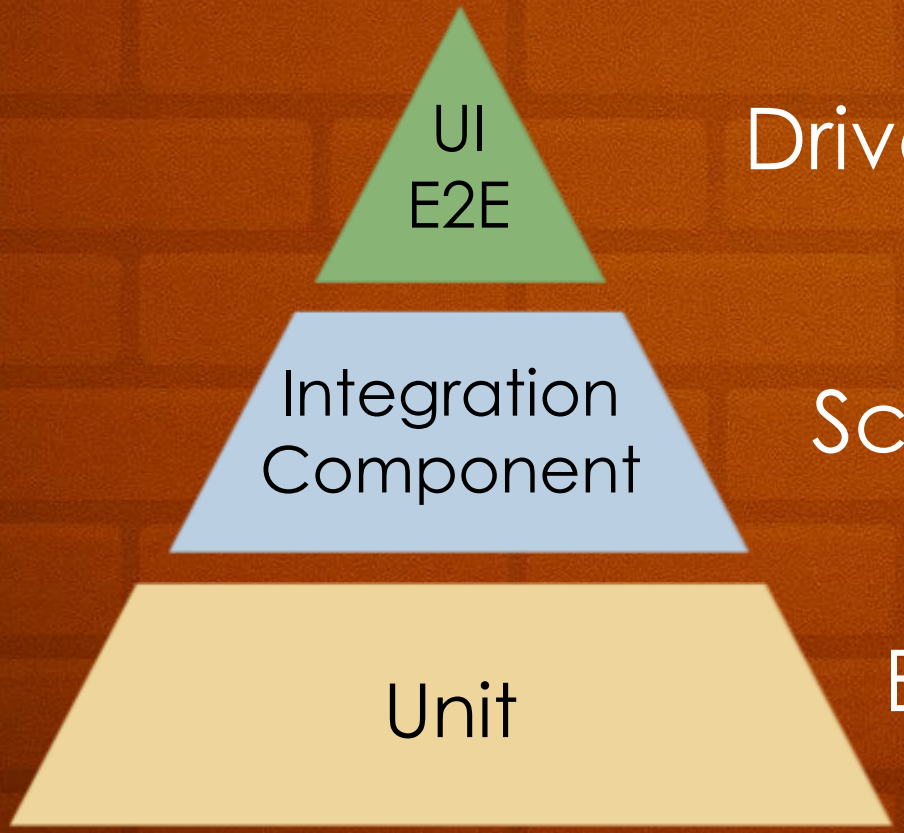
Testing Pyramid



Legacy

Unhealthy Testing

Legacy Pyramid

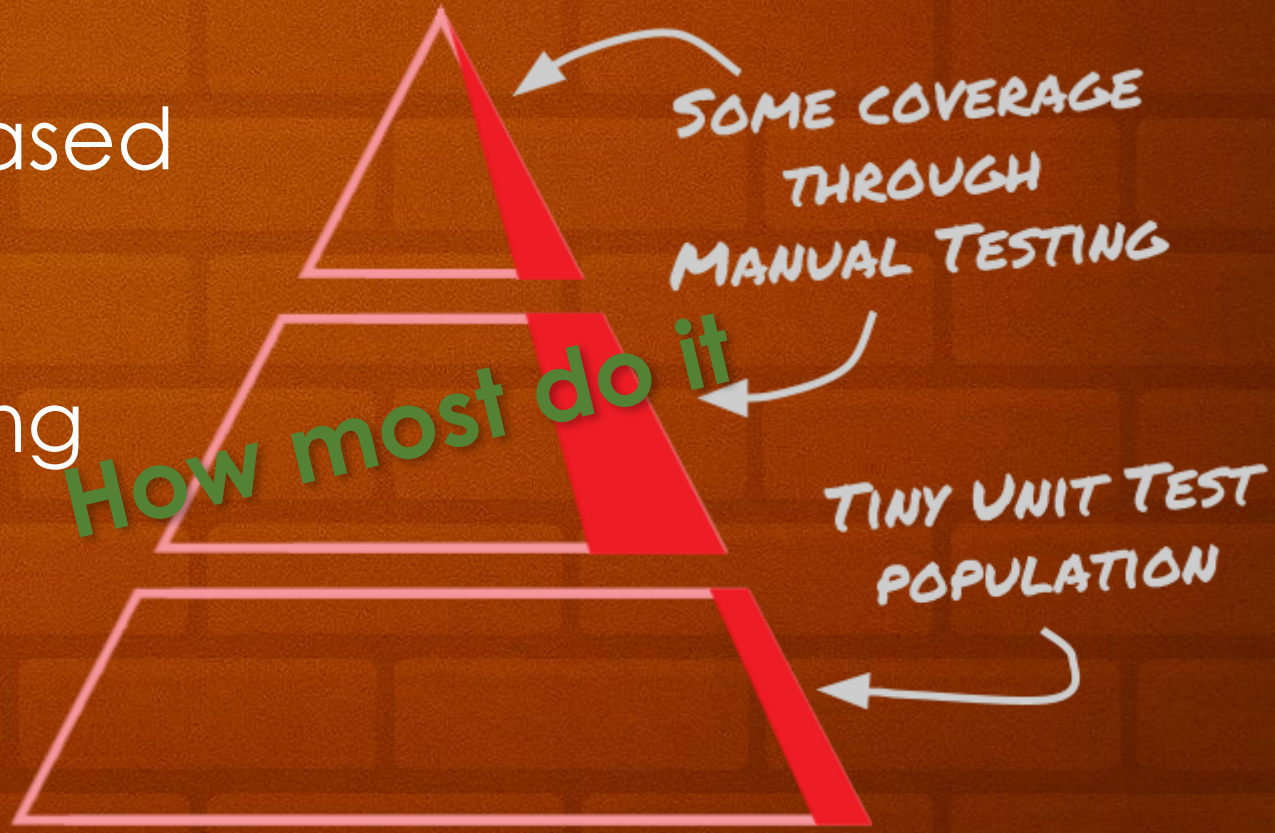


Driver Based

Scripting

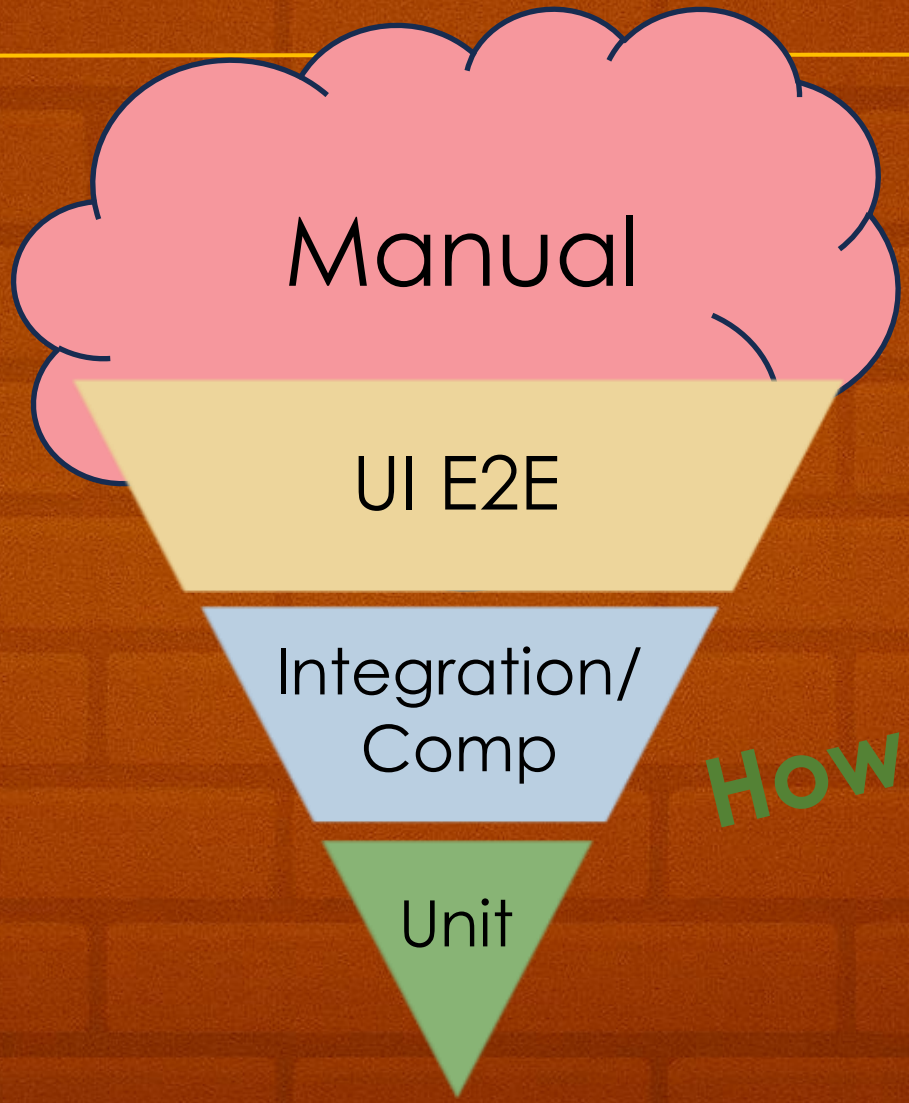
Build

Unhealthy Testing



Legacy

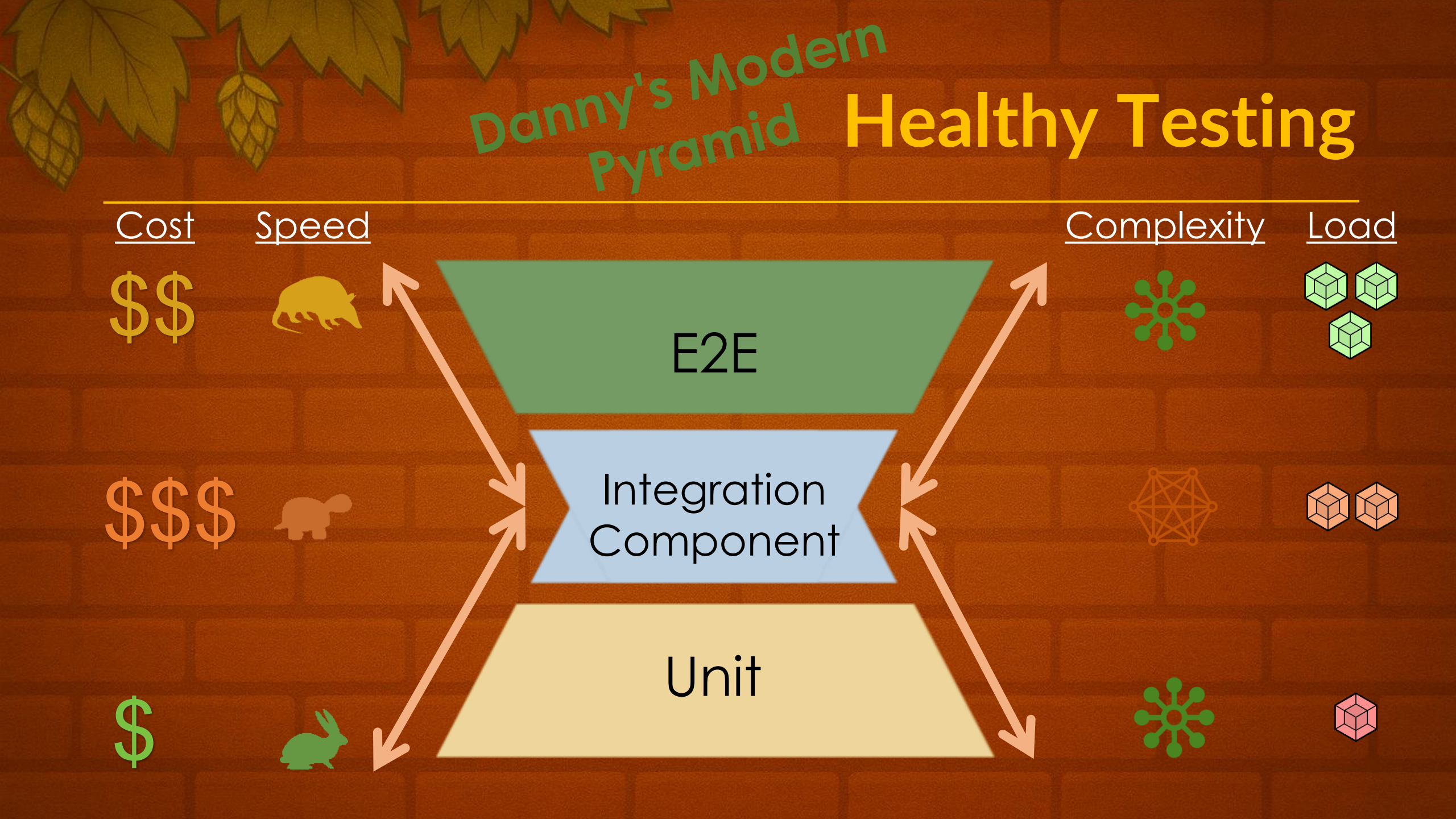
Unhealthy Testing II



How most do it

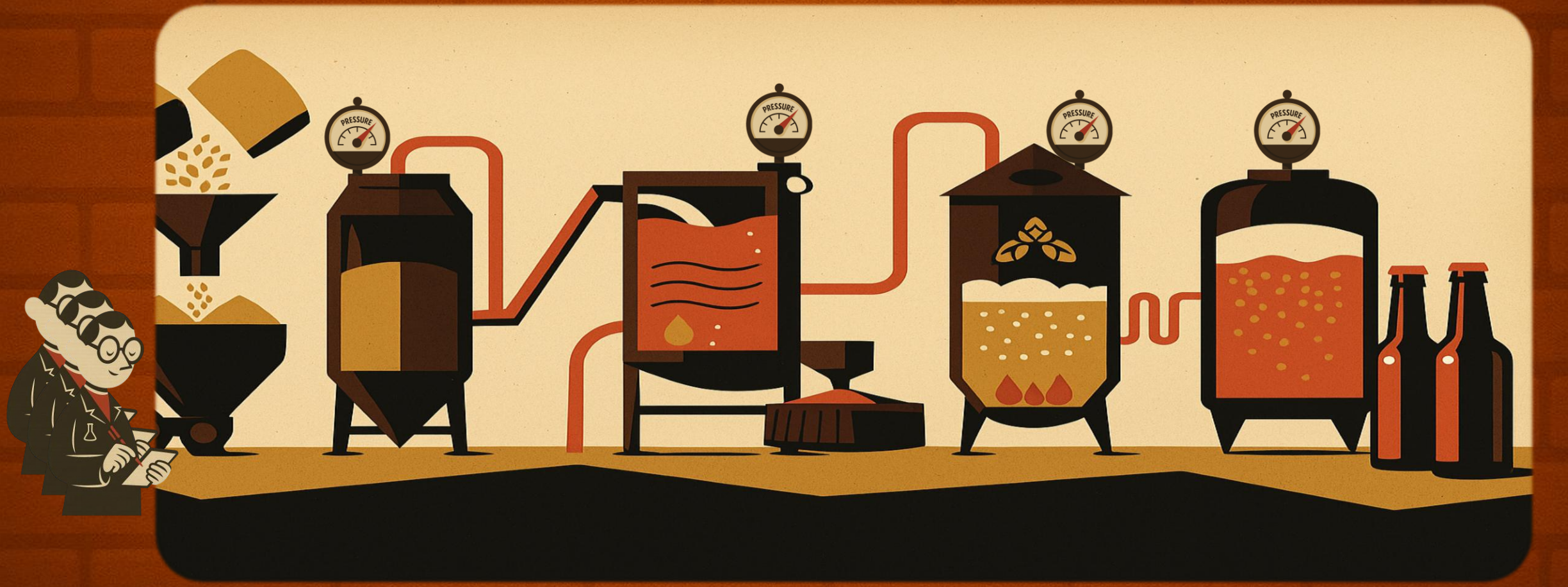
Danny's Modern Pyramid

Healthy Testing

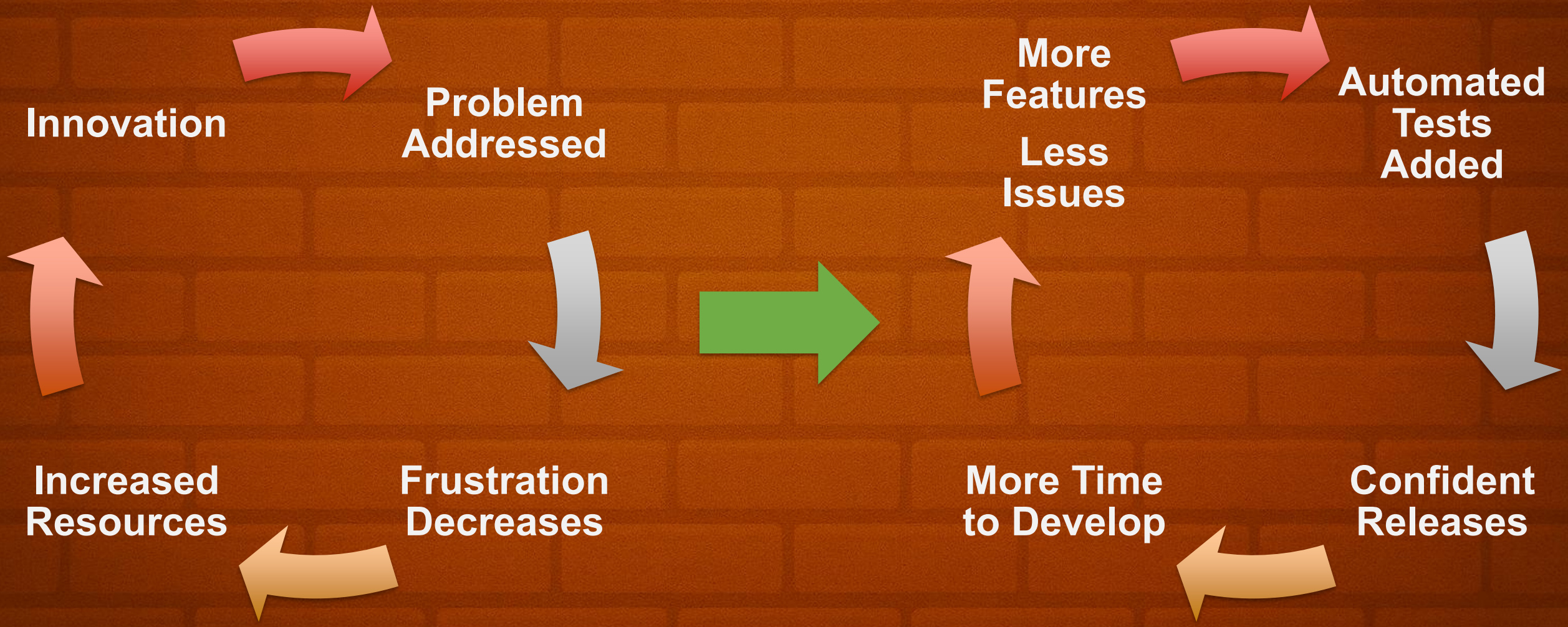


Increasing Confidence

Confident Releases



Confidence Builds On Itself



Case Study

By the numbers

6	Months
3	Devs
80%	Code coverage
0	UAT bugs found
5	Weeks before first error
<.1%	Error rate
>.5%	Baseline error rate
>5x	Better than baseline average

Demo-ish

E2E on Local

WicProduct | task/wicp-2599-setup-jenkins-ad-hoc-e2e | can login with persona cards

Solution

- .nx · no index
- apps
  - participant-portal
  - vendor-portal
  - wic-admin
    - e2e
      - .auth
      - support
      - auth.setup.ts
      - dashboard.spec.ts
      - login.spec.ts
    - src
    - .browserslistrc
    - .eslintrc.json
    - build.js
    - jest.config.ts
    - ngsw-config.json
    - playwright.config.ts
    - project.json
    - tailwind.config.js
    - tsconfig.json
    - tsconfig.app.json
    - tsconfig.editor.json
    - tsconfig.spec.json
- dist · no index
- libs
  - wic-clinic
  - wic-clinic-base
  - wic-clinic-prototype
  - wic-component
  - wic-design
  - wic-e2e
  - wic-food
  - wic-platform
    - authentication
    - authorization
    - blueprint-forge
    - chart

package.json | login.spec.ts

```
36  test("can login with selector", async ({ page } : PlaywrightTestArgs & PlaywrightTestOption... ) : Promise<void> => {
43      await page.getByLabel("Username").click();
44      await page.getByRole("option", { name: "admin@wicvp" }).locator("span").click();
45      await page.getByRole("button", { name: "Sign in" }).click();
46
47      // Wait until the login as succeeded.
48      const fakeLoginInterceptResponse : Response = await fakeLoginIntercept;
49      expect(fakeLoginInterceptResponse.status()).toBe(200);
50  });
51
    Danny Hearit
52  test("can login with persona cards", async ({ page, isMobile } : PlaywrightTestArgs & PlaywrightTestOption... ) : Promise<void> => { - 3 sec, 964 ms
53      // Skip tests conditionally
54      test.skip(isMobile, "This feature is not available on mobile browsers");
55
56      await page.goto("/wic-admin"); - 2 sec, 583 ms
57
58      // Sets up an interceptor to wait for a response from the URL before the request is sent. Do before button click that would send response.
59      // No await yet.
60      const fakeLoginIntercept : Promise<Response> = page.waitForResponse("../api/authentication/FakeLogin"); - 660 ms
61
62      await page.locator("wic-persona").filter({ hasText: "admin@wicvp" }).click(); - 485 ms
63
64
65      // Wait until the login as succeeded.
66      const fakeLoginInterceptResponse : Response = await fakeLoginIntercept;
67      expect(fakeLoginInterceptResponse.status()).toBe(200); - 1 ms
68  });
69
70
```

Stop Recording

wic-product › angular › apps › wic-admin › e2e › login.spec.ts

51:1 | CRLF | UTF-8 | 4 spaces

20°F Cloudy

3:14 PM 2/14/2025

WicProduct | task/wicp-2599-setup-jenkins-ad-hoc-e2e | login.spec.ts

Solution

- .nx · no index
- apps
  - participant-portal
  - vendor-portal
  - wic-admin
    - e2e
      - .auth
      - support
      - auth.setup.ts
      - dashboard.spec.ts
      - login.spec.ts
    - src
    - .browserslistrc
    - .eslintrc.json
    - build.js
    - jest.config.ts
    - ngsw-config.json
    - playwright.config.ts
    - project.json
    - tailwind.config.js
    - tsconfig.json
    - tsconfig.app.json
    - tsconfig.editor.json
    - tsconfig.spec.json
- dist · no index
- libs

package.json | login.spec.ts

```
5   "scripts": {
20      "build:participant-portal:azure-dev": "nx build --configuration=azure-dev --project=participant-portal",
21      "test": "nx affected:test --target=test --base=develop --head=HEAD",
22      "test-all": "nx run-many --all --target=test",
23      "lint": "nx run-many --all --target=lint",
24      "lint-ci": "nx run-many --all --target=lint --configuration=ci",
25      "lint-fix": "nx run-many --all --target=lint --fix",
26      "pre-review": "npm run build:wic-admin:prod && npm run build:vendor-portal:prod && npm run build:participant-portal:prod && npm run lint && npm r
27      "check-circles": "npx depcruise libs --validate --exclude \"^node_modules\" --output-type err-long",
28      "e2e": "nx run-many --all --target=e2e",
29      "e2e:participant-portal": "nx e2e participant-portal --skip-nx-cache",
30      "e2e:participant-portal-ui": "nx e2e participant-portal --ui --skip-nx-cache",
31      "e2e:vendor-portal": "nx e2e vendor-portal --skip-nx-cache",
32      "e2e:vendor-portal-ui": "nx e2e vendor-portal --ui --skip-nx-cache",
33      "e2e:wic-admin": "nx e2e wic-admin --skip-nx-cache",
34      "e2e:wic-admin-ui": "nx e2e wic-admin --ui --skip-nx-cache"
35  },
36  "private": true,
37  "dependencies": {
38      "@angular/animations": "18.2.0",
39      "@angular/cdk": "18.2.0",
40      "@angular/common": "18.2.0",
41      "@angular/compiler": "18.2.0",
```

Run | login.spec.ts

- iPad Safari 20 sec 763 ms
  - login.spec.ts 20 sec 763 ms
    - has title 4 sec 685 ms
    - has login for state work 3 sec 867 ms
    - has sso login for state 3 sec 618 ms
    - has persona login card 3 sec 447 ms
    - can login with persona cards 954 ms
    - can login with selector 4 sec 192 ms

Tests passed: 39 of 39 tests – 2 min 37 sec

```
playwright test
```

Stop Recording

wic-product > angular > package.json

30:44 CRLF UTF-8 4 spaces* JSON: package

20°F Cloudy

3:16 PM 2/14/2025

Demo-ish

E2E on Jenkins

EGLE - MiEHDWIS - Activity His | All [Wic Product] - Jenkins

jenkins.kunzleigh-hosted.com/job/Wic%20Product/

KL&A Housekeeping | Water | WIC Product | KL&A | Water | WIC | Automated Testing | Utils

Jenkins

Danny Hearit | log out

Dashboard > Wic Product >

- Status
- Configure
- New Item
- Build History
- Project Relationship
- Check File Fingerprint
- Convert Folders Job To Pipeline
- Move
- Favorite
- Open Blue Ocean
- Rename
- Config Files
- Credentials

Wic Product

Add description

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration	F	# Issues
✓	☀	Create_Azure_Seed	3 mo 3 days #16	N/A	8 min 49 sec	▷	☆	-
✓	☀	Demo-Deploy	8 days 0 hr #35	5 mo 6 days #23	26 min	▷	☆	30
✓	☁	Dev Deploy	1 day 4 hr #311	2 days 0 hr #310	28 min	▷	☆	31
✗	🌧	E2E	1 mo 2 days #14	28 days #15	8 min 25 sec	▷	☆	-
✓	🌧	LinkDeploy	3 mo 10 days #8	3 mo 10 days #7	26 min	▷	☆	32
✓	☀	Local Azure DB Refresh	7 days 3 hr #55	N/A	7 min 21 sec	▷	☆	-
✗	⛅	Nightly-Cache-Build	1 day 13 hr #46	13 hr #47	23 min	▷	☆	-
✓	☀	NpmCacheRefresh	13 hr #293	N/A	6 min 11 sec	▷	☆	-
✓	☁	Publish WicProduct	5 mo 25 days #46	5 mo 25 days #45	5 min 3 sec	▷	☆	-
✓	☁	Publish WicProduct DEV			5 min 41 sec	▷	☆	-
✓	⛅	Pull-Request-Builder	54 min #839	16 hr #836	25 min	▷	☆	88

Stop Recording

Build Queue

No builds in the queue.

Build Executor Status

- Devops-Windows-Agent 0/1
- azure-devops-agent-large42c700 0/1
- azure-lps-ci-agent- (offline)

20°F Cloudy

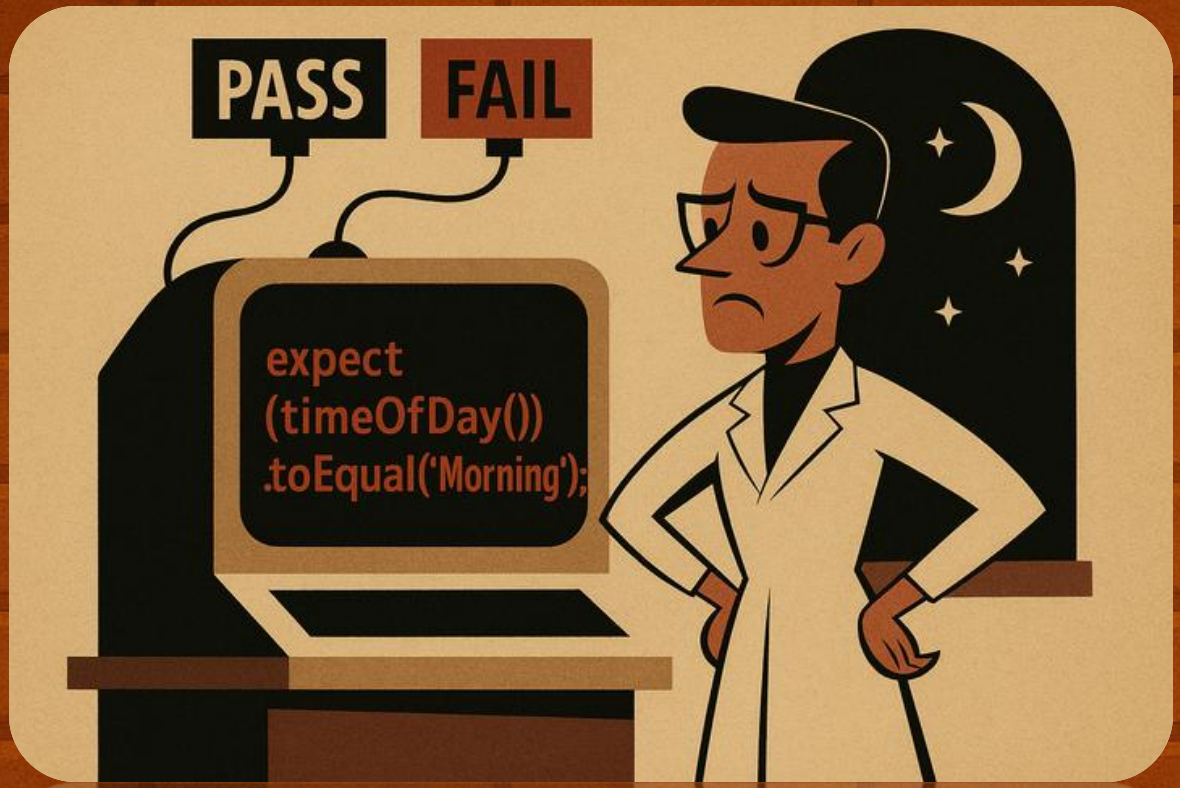
3:19 PM 2/14/2025

Getting Started on Your Team

Follow the Epic Journey

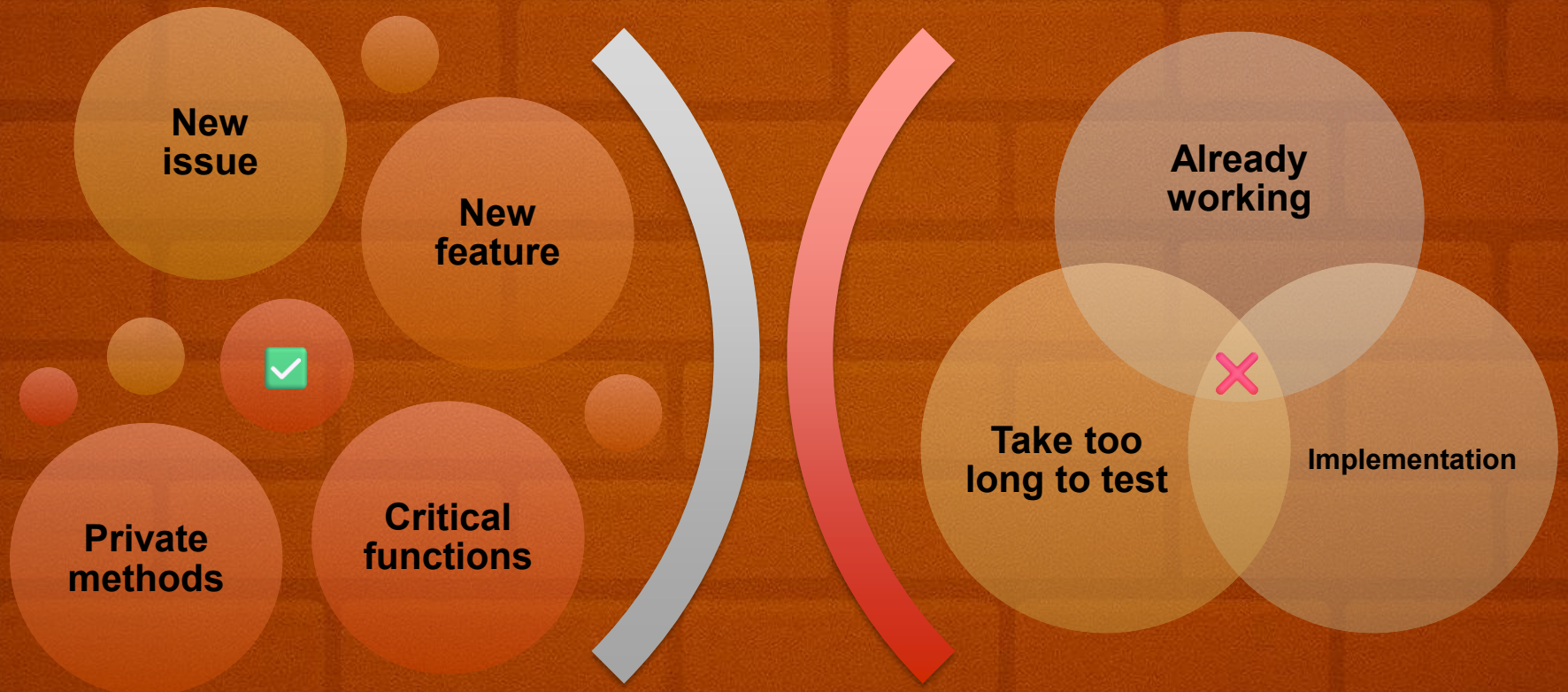
Setup E2E Automated Testing

- [ ] Setup Playwright framework
- [ ] Create basic tests from login page
- [ ] Create CI E2E jobs
- [ ] Develop shared patterns



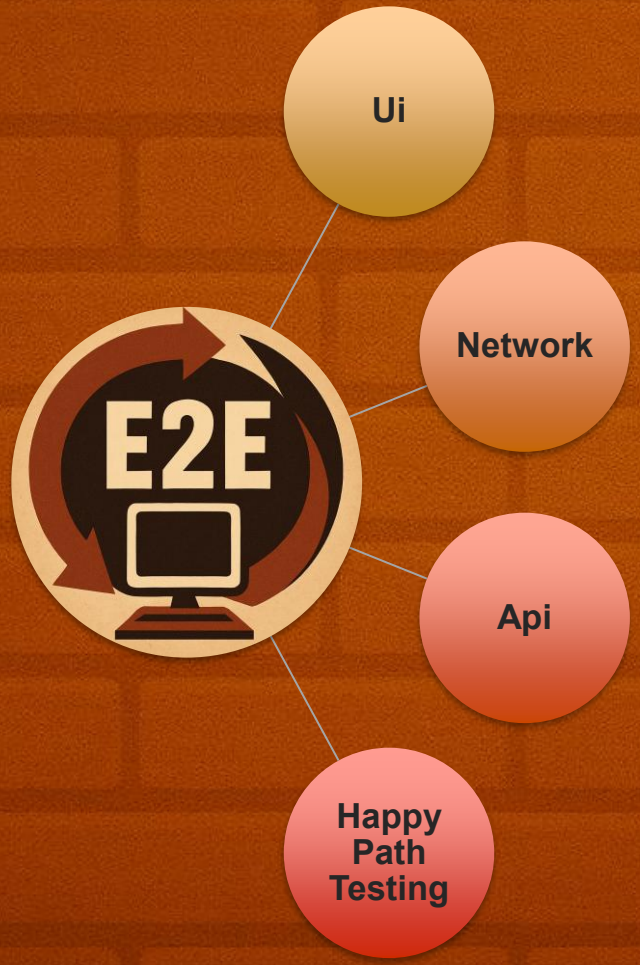
Testing Best Practices

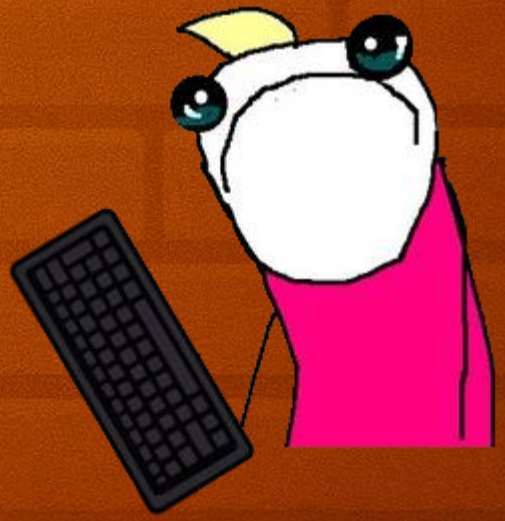
Automated Testing Focus



Test ✅ **Don't Test** ❌

E2E Testing Focus





E2E Testing Focus

Tests Should Be Quick

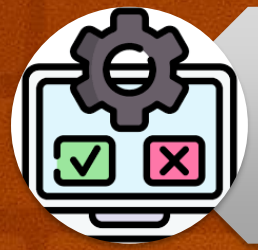
- 5-15 minutes per test to write
  - More granular test or less complex
  - Use google
- 30 seconds or less per test to run
  - Investigate slow system or an alternative testing methods
  - `test.slow()`

Tests Should Be Effective

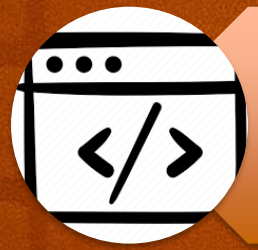
- No testing debt
  - If they are no longer useful, get rid of them!
- Test user behavior
  - Not implementation details
- Fail fast
  - Fail at the non-okay response rather than waiting for a timeout

TDD

Test Driven Development



Testable Code



Design Before Implementation

No Skipping Tests

E2E Testing With Playwright

E2E Tests Written Like Unit Tests

- Typescript
- Assertions
- Async and synchronous checks
- No waiting by time

Content Interactions

- CSS Selectors - Role/Accessibility
- Shared session authentication
- Mature features and content

Danny Hearit *

```
test.describe( title: "as admin - food groups", () : void => {
    test.use({ storageState: adminUserFile });

    test( title: "should have food classification title", async({ page, isMobile } : PlaywrightTestArgs & Playwrig
        await navigateToFoodClassification(page, isMobile);
        await expect(page.getByRole( role: "heading", { name: "Food Classification" })).toBeVisible();
    });
});
```

Playwright Tools

Browser Pugin

Playwright CRX

IDE Plugins

Trace Viewer

UI Runner

Playwright MCP

Playwright Mobile Testing

Important And Easy

- So easy to miss mobile bugs
- Be the hero 🎉

Responsive Design Helps Accessible Design

- ADA & WCAG compliance teams will thank you

Recommended Mobile Safari

- iPad & iPhone

Recommended Mobile Chromium

- Pixel

```
projects: [
    // Device setup
    { name: "setup", testMatch: /.*\.setup\.ts/ },
    {
        name: "Mobile Chromium",
        use: { ...devices["Pixel 5"] },
        dependencies: ["setup"],
    },
    {
        name: "Mobile Safari",
        use: { ...devices["iPhone 12"] },
        dependencies: ["setup"],
        timeout: 60000,
    },
    {
        name: "iPad Safari",
        use: { ...devices["iPad (gen 7)"] },
        dependencies: ["setup"],
        timeout: 60000,
    },
],
```

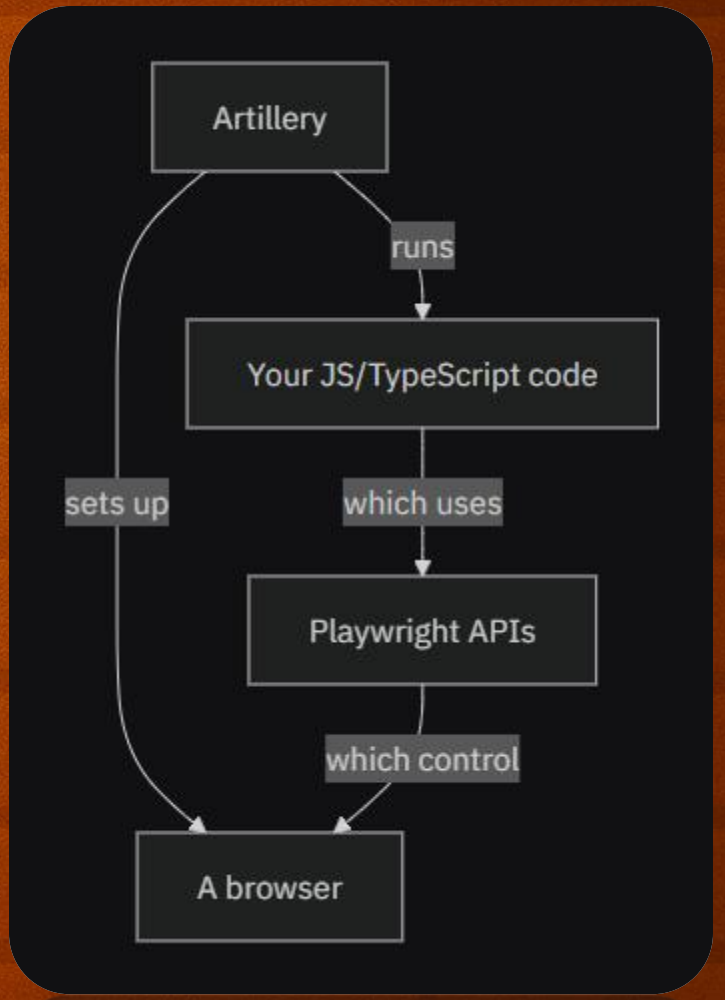
Playwright Load Testing

Artillery.io

- https://www.artillery.io/docs/reference/engines/playwright
- https://www.artillery.io/docs/playwright

With Artillery's Playwright Engine You Can

- Create load tests for complex web apps using Playwright's API
- Reuse existing Playwright code for load testing
- Track Core Web Vitals metrics automatically and measure how they change under load
- Create no-code tests with playwright codegen
- Scale to tens of thousands of headless Chrome instances with AWS Fargate or Azure ACI



E2E Suggested Habits

Suggested Dev Habits

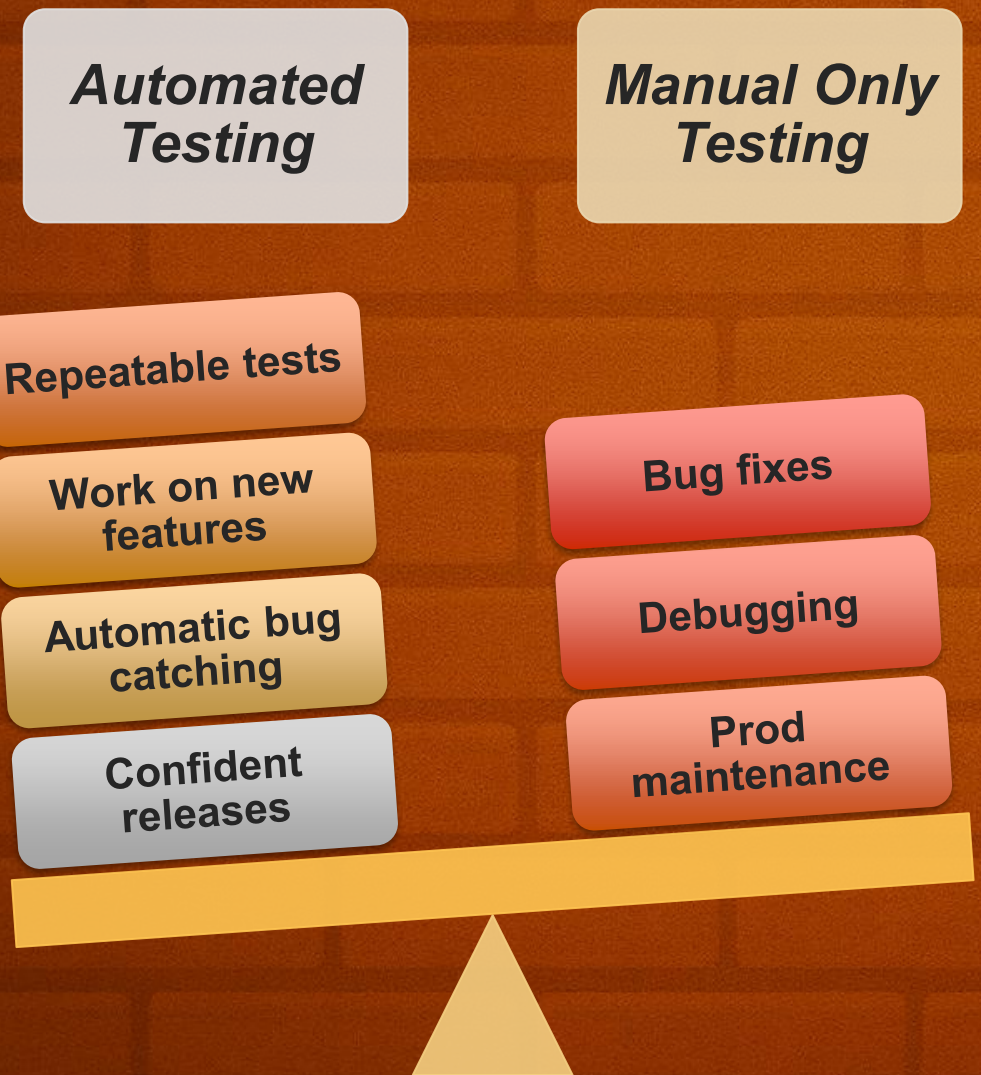
- Check nightly CI runs
- Keep E2E watching while coding

Suggested BA Habits

- Suggest automated tests be written for repeatedly troubled areas

Not a replacement for manual testing

Call to Action - Make your choice



The difference between automated tests and not, is what you will be doing the following months of release.

Making new features?

Or prod maintenance, debugging and bug fixes?

Which would you rather do?

Questions and Resources

Getting Started On Your Team

Setup Playwright

Writing Tests

Help From The Web

Playwright Solutions

Best Practices

Playwright Best Practices

Link to Presentation Resources