ConFoo
DEVELOPER CONFERENCE

Implementing Shift-Left with Cypress Component Testing

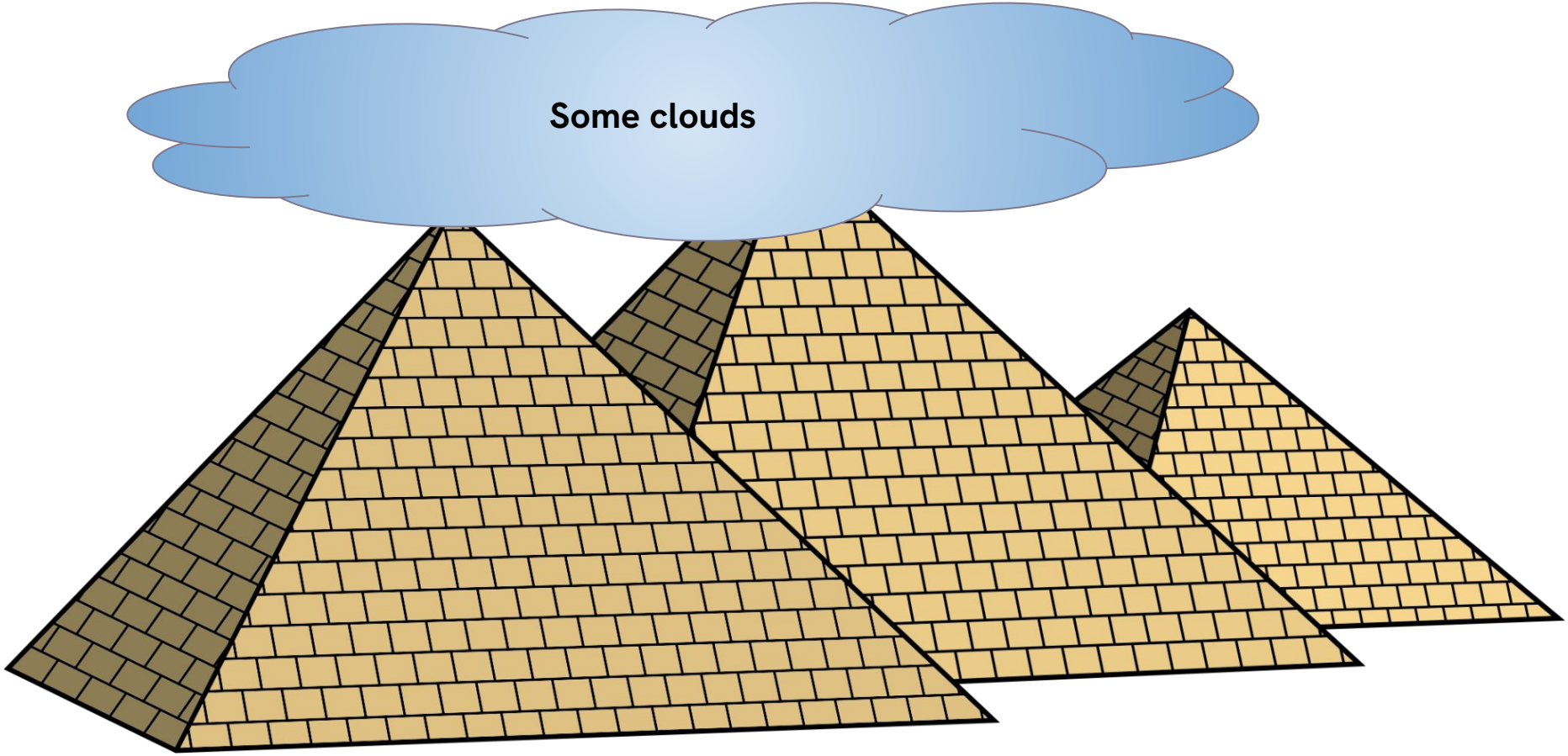
February 28, 2025

Vitaly Skadorva

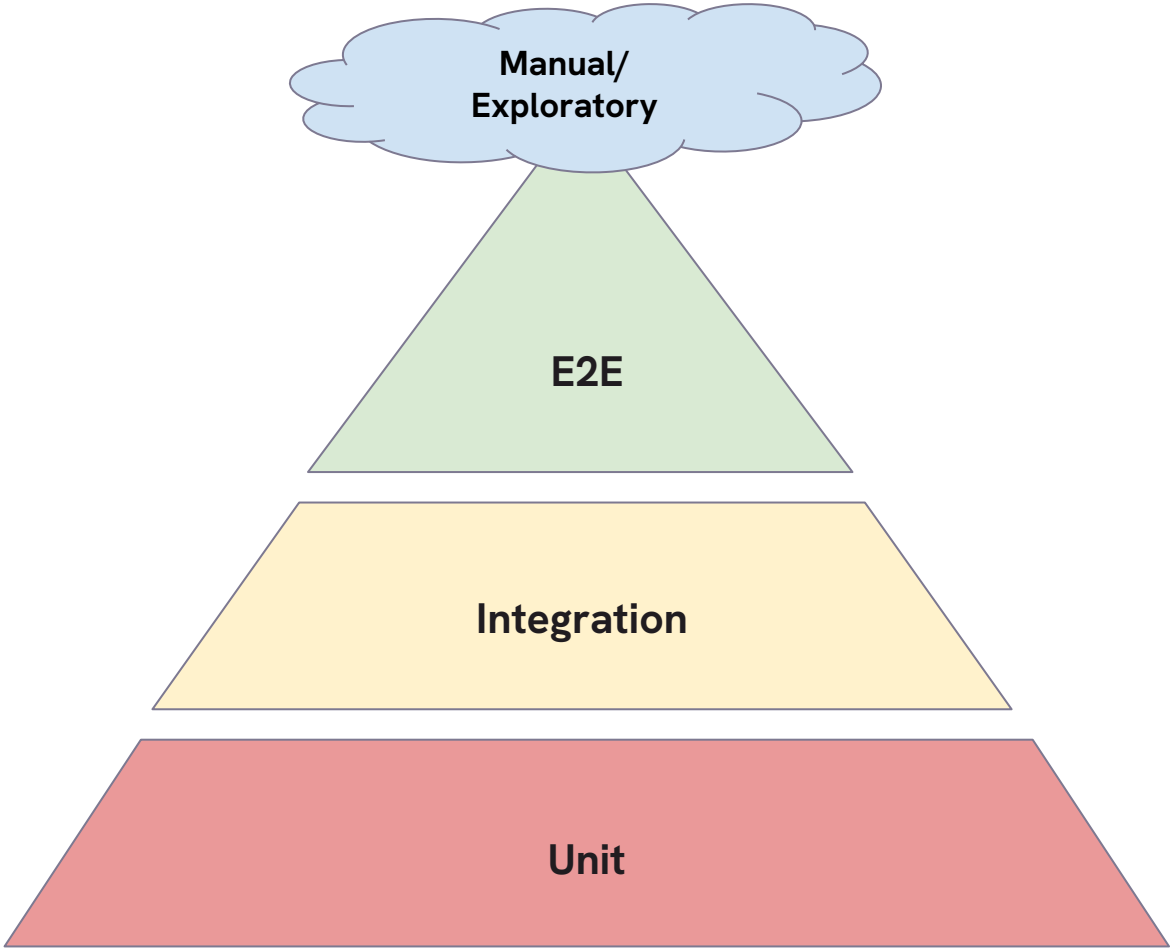
- **Speaker:** Vitaly Skadorva
- **Role:** QA Automation Specialist at Intact Financial Corporation
- **Linkedin**: https://www.linkedin.com/in/vitalyskadorva/
- **My Book**: https://a.co/d/gGls5fn

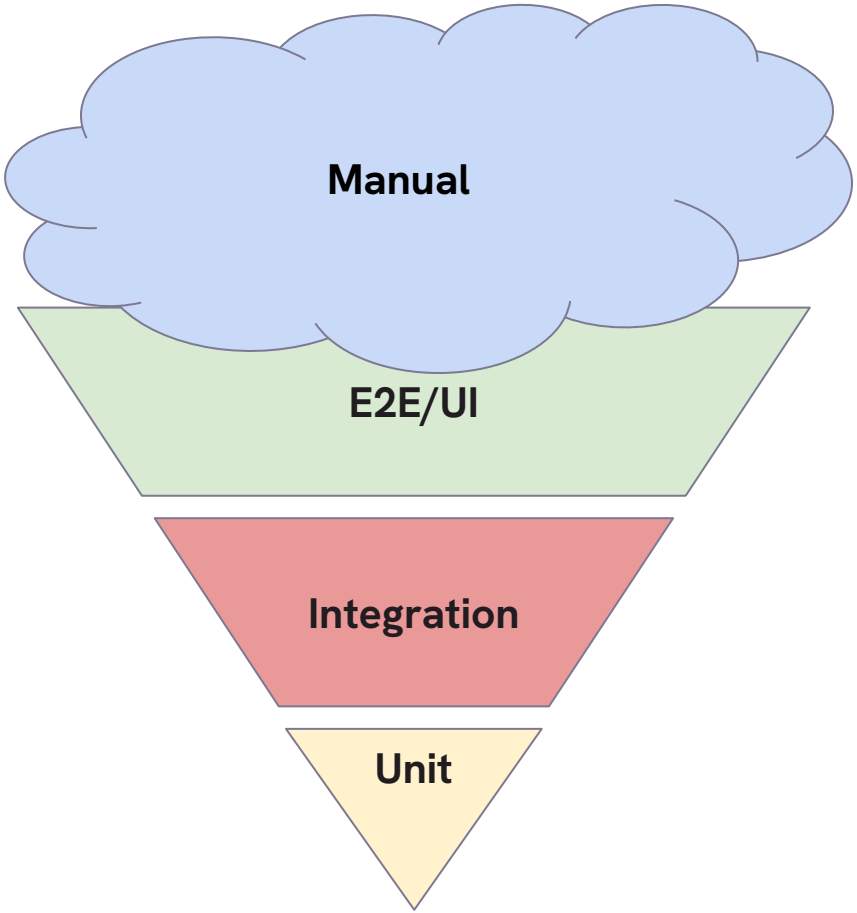
My Story at Intact

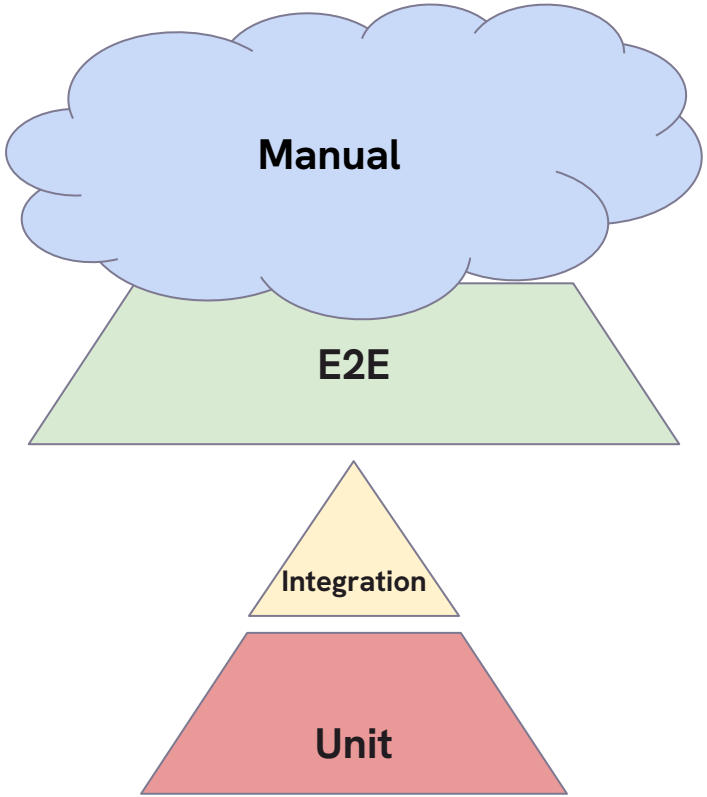
Pyramid


Some clouds

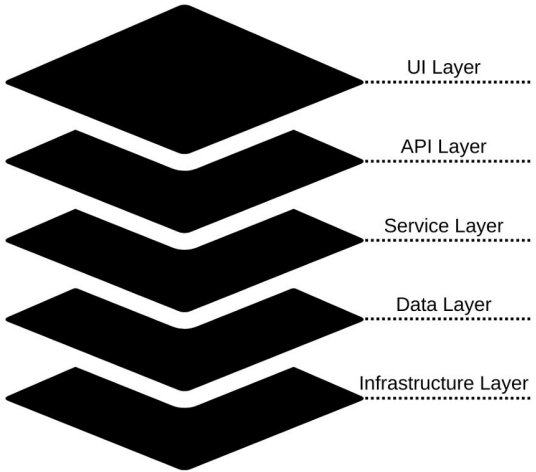
Testing Pyramid


Manual/
Exploratory
E2E
Integration
Unit

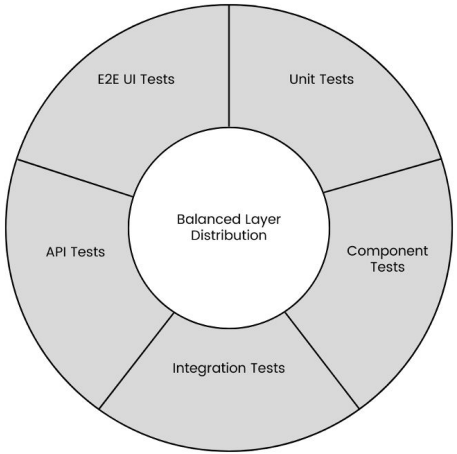

Manual
E2E/UI
Integration
Unit


Manual
E2E
Integration
Unit

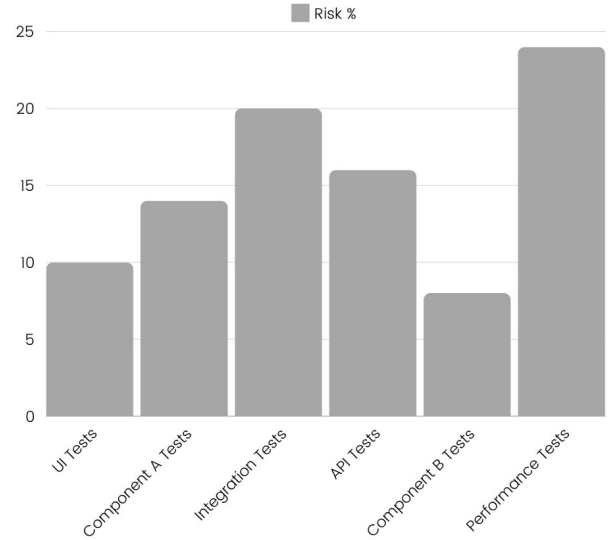
Other Testing Architectures



Layered Testing Architecture

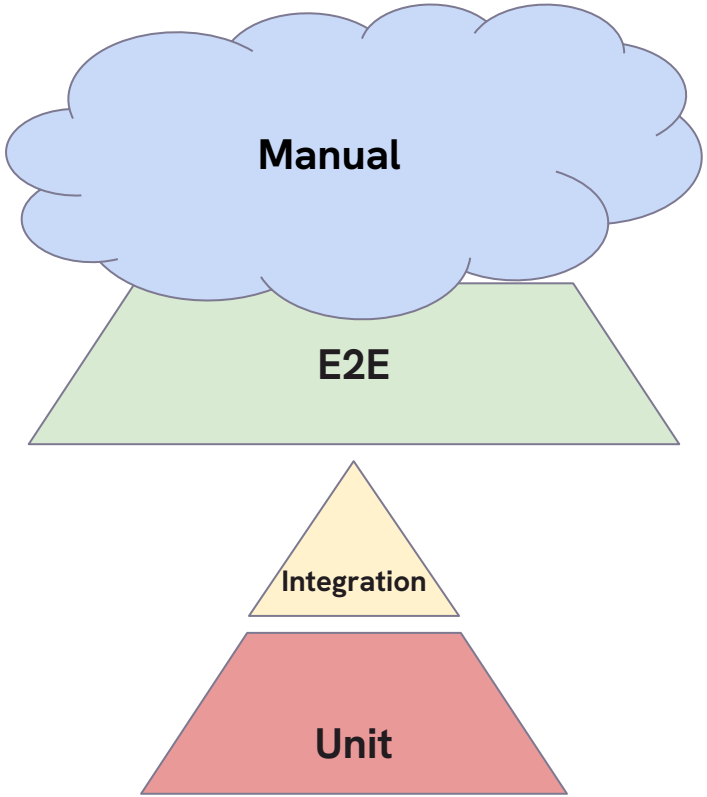


Balanced Layer Distribution



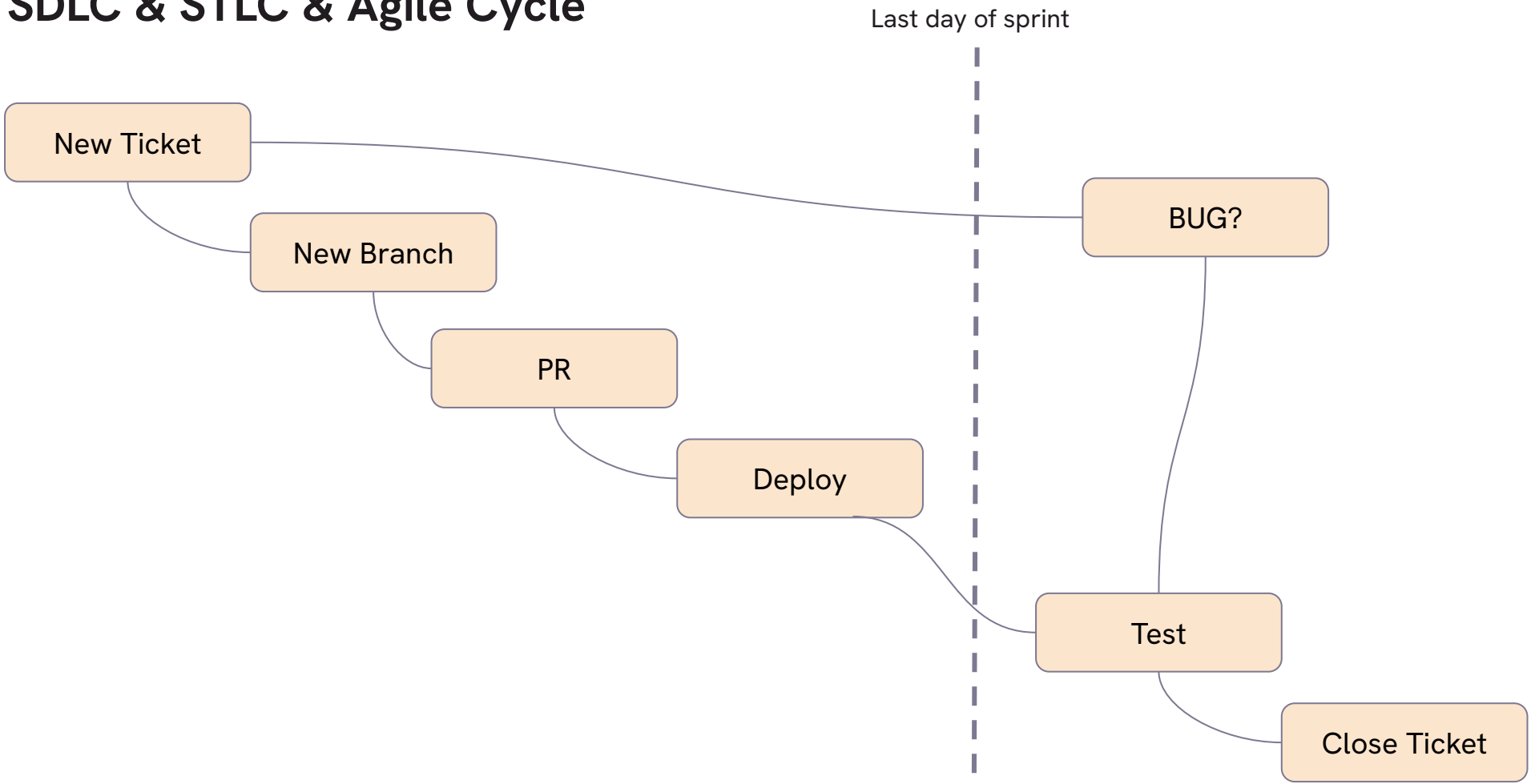
Risk-Based Layer Coverage

Have you seen any of the perfectly implemented?


Manual
E2E
Integration
Unit

Anything else?

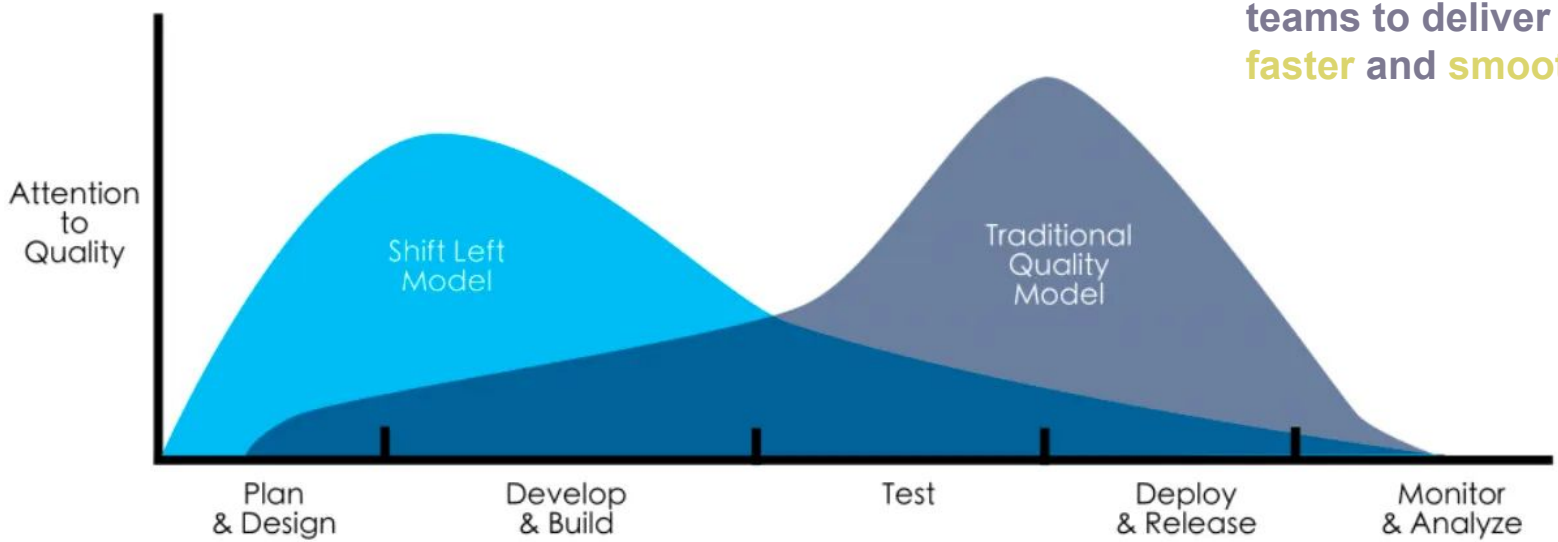
SDLC & STLC & Agile Cycle



“Shifting Left”

What is Shift-Left Testing?

Enables the development teams to deliver better, faster and smoother.



SHIFT LEFT =

EARLY Detection

COST Effectiveness

TIME Saving

RELEASE Smooth

CUSTOMER Satisfaction

Where we were?

Left (Quality Engineering)

- Universal Testing Engagement
- Company-Wide Quality Focus
- Rapid Feedback Emphasis
- Automated Testing as Gate
- Trigger Testing with PRs
- Short Defect Cycle
- Fail Forward Learning
- Shift Left and Right
- **Sustainable QA Staffing**
- **Technical QA for Analysis**

Middle (Quality Assistance)

- **Test with Development**
- Manual to Exploratory Testing
- **Automated Testing Scheduling**
- Faster Defect Resolution
- **Agile Teams Gate Releases**
- **Whole Team Quality Focus**
- Risk-Based Testing Priority

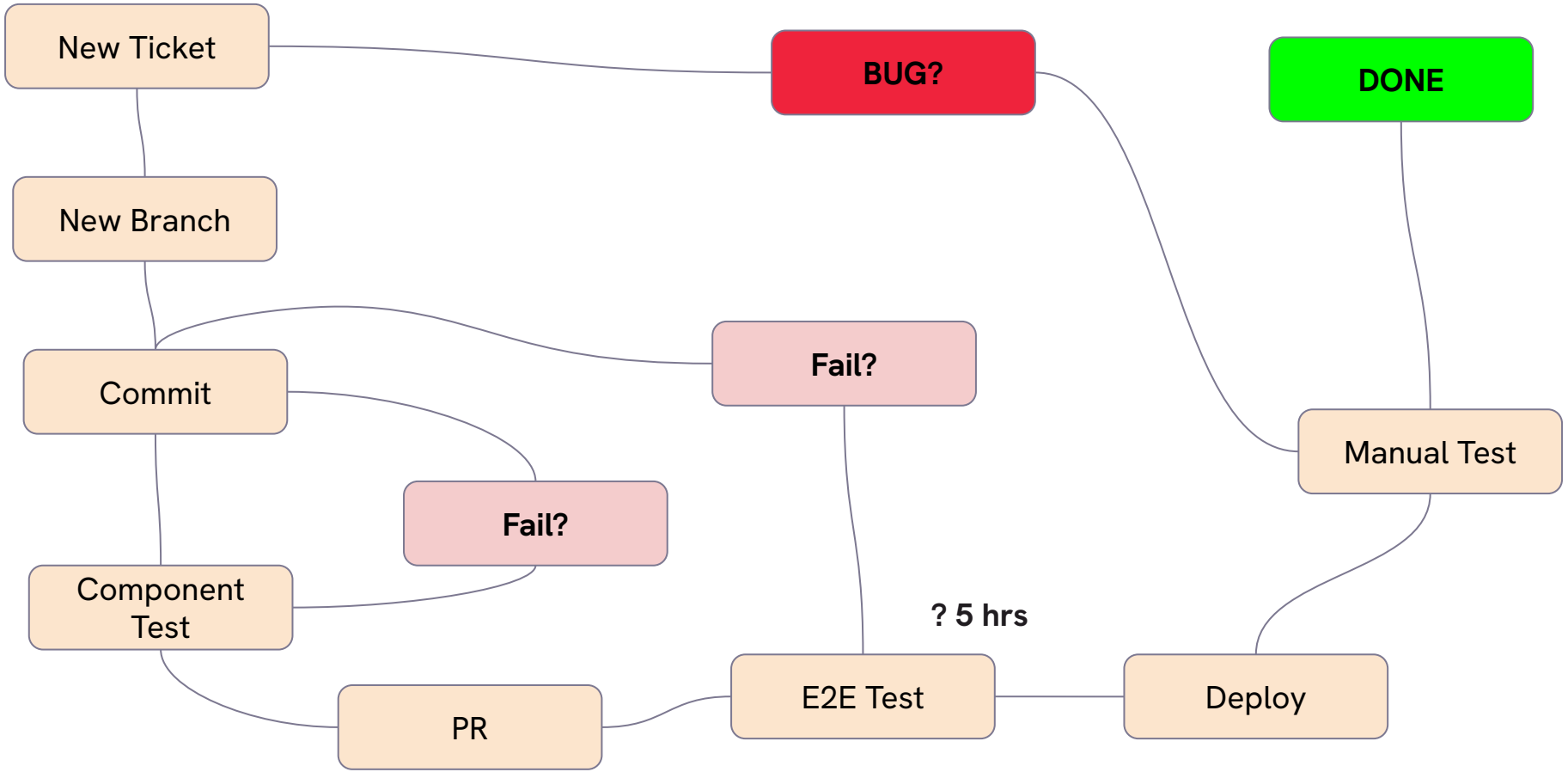
Right (Quality Control)

- **Post-Dev Testing**
- High Manual Costs
- **Slow Testing**
- **Release Bottleneck**
- **Long Fix Cycles**
- Limited Collaboration
- Focus on Detection
- **Triggered Automation**
- High Staffing Needs
- **Validation Over Exploration**

Where we were?

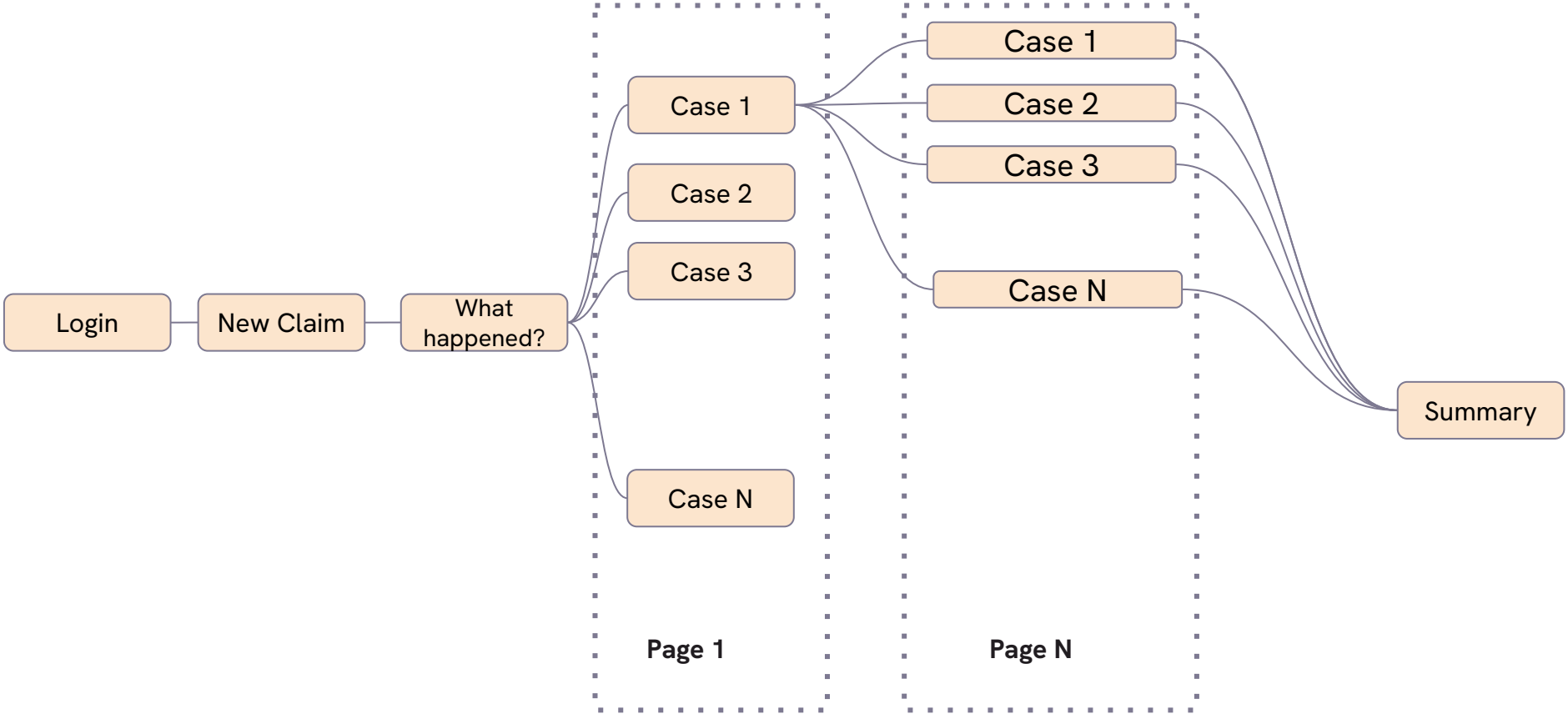
- >500 - E2E tests
- 3-5 hrs - Execution time
- 20% - Flakiness rate

Let's try to shift left

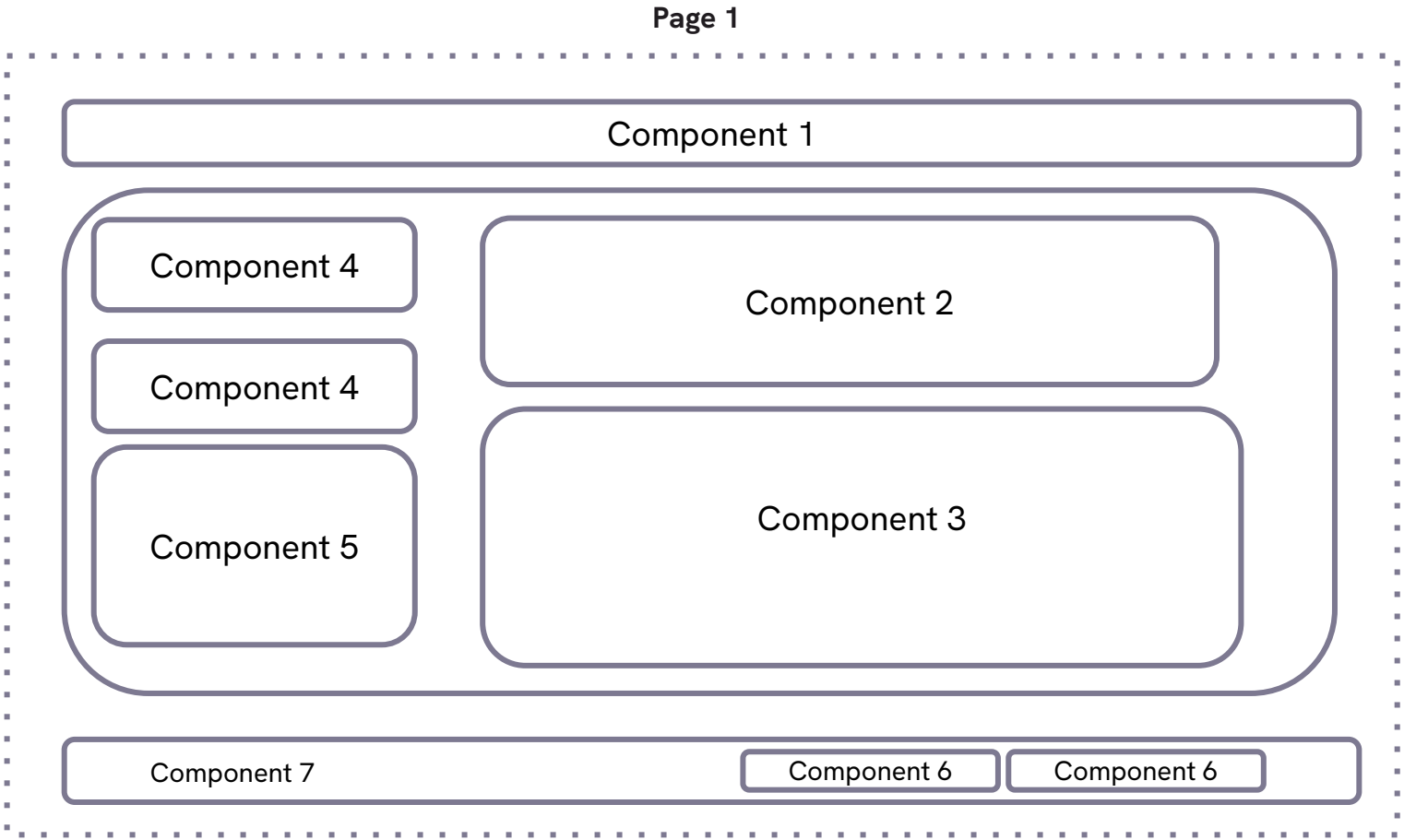


We need Decomposition

Let’s analyze our app



Let’s analyze our app



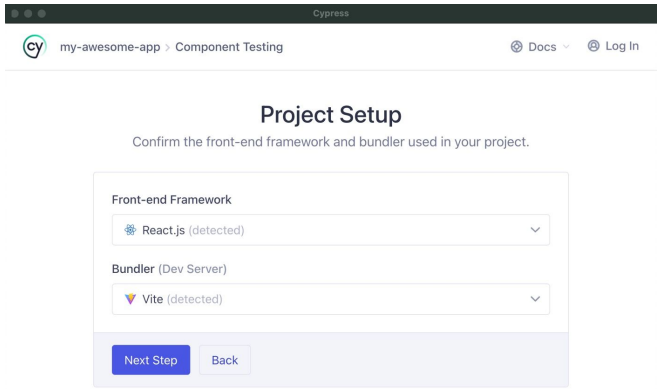
Can you imagine?

1 component reused on 32 pages

1 component X 32 pages X 5 brands X 2 languages = 320 tests

Use Component tests to reduce E2E tests

Installation & Setup for Cypress Component Testing



Install Cypress

Use npm to install Cypress:

`*npm install --save-dev cypress*`.

Open Cypress

Simply open Cypress with the command `*npx cypress open*`.

Follow Launchpad Instructions

Open Cypress and follow the instructions from the Cypress Launchpad to proceed.

Start Testing

Start testing with:

```
`import Button from
'path/to/your/component'

it('renders', () => {

cy.mount(<Button>);
cy.get('button').should('be.visible');

})`.
```

Component Example

```
Button.propTypes = {
 children: PropTypes.node.isRequired,
 onClick: PropTypes.func,
 variant: PropTypes.oneOf(['primary', 'secondary']),
 size: PropTypes.oneOf(['small', 'medium', 'large']),
 disabled: PropTypes.bool,
 loading: PropTypes.bool,
 icon: PropTypes.node,
 loadingIcon: PropTypes.node,
 className: PropTypes.string,
 type: PropTypes.oneOf(['button', 'submit', 'reset']),
 'data-cy': PropTypes.string
};
```

Let’s test it

```
it('renders primary button by default', () => {

   cy.mount(<Button>Click me</Button>);

   cy.get('[data-cy="button"]')
     .should('have.class', 'button--primary')
     .and('contain', 'Click me');
 });
```

Let’s test it

```
it('handles click events', () => {
   const onClick = cy.spy().as('clickHandler');

   cy.mount(<Button onClick={onClick}>Click me</Button>);

  cy.get('[data-cy="button"]').click();
  cy.get('@clickHandler').should('have.been.calledOnce');
 });
```

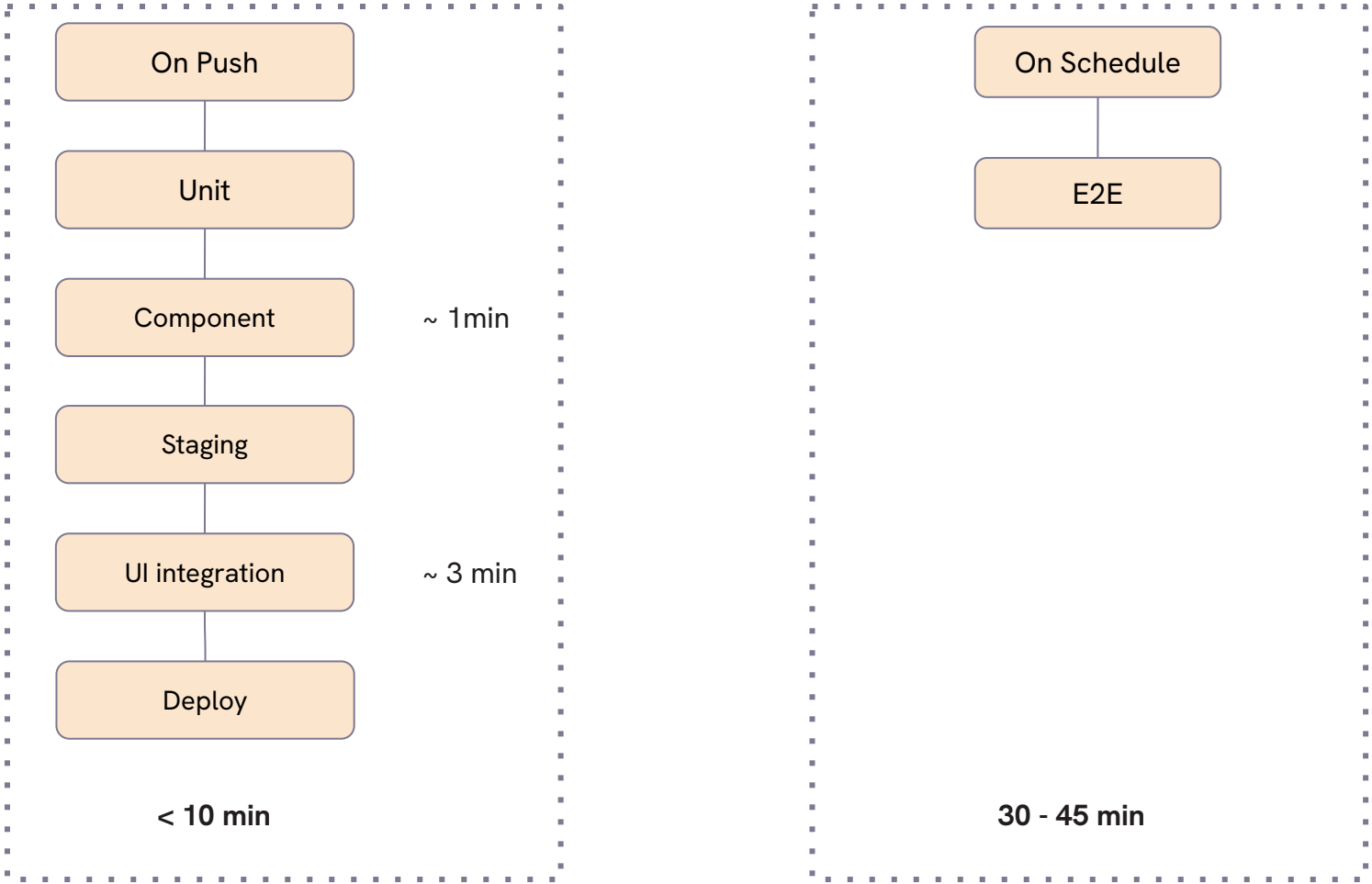
DEMO

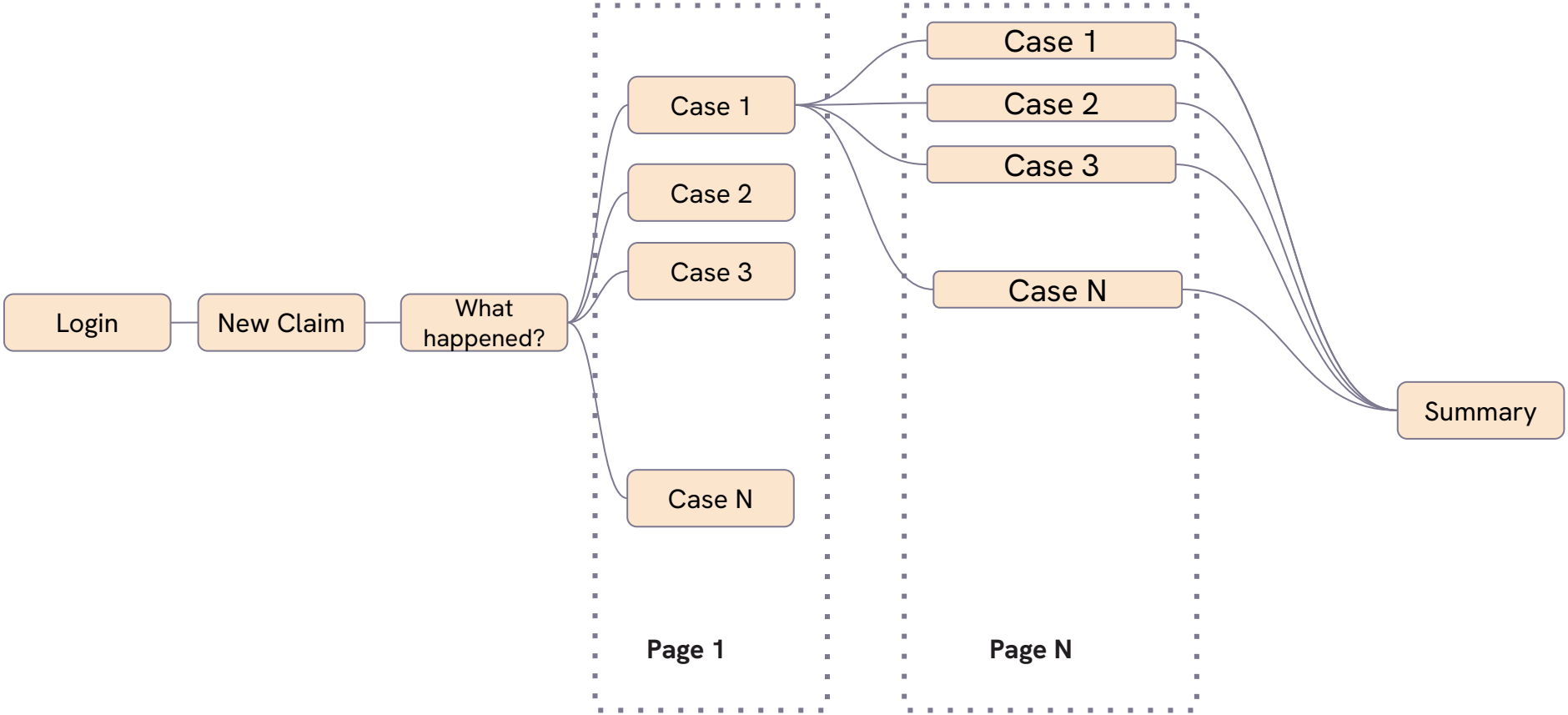
Make your tests running

Put your tests in CI/CD

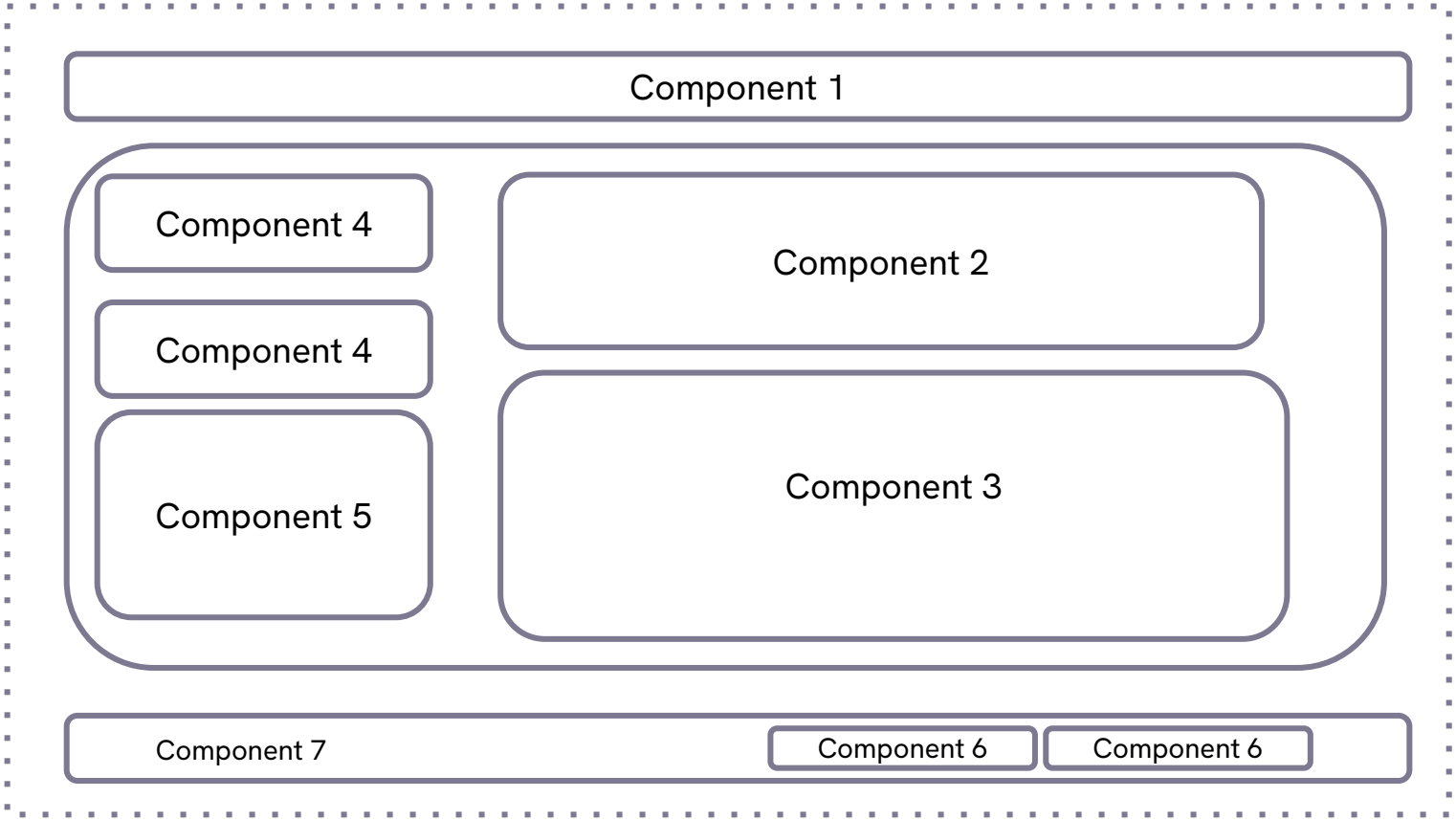
```yaml
name: Cypress Component Tests
on: [push]
jobs:
  cypress-run:
    runs-on: ubuntu-22.04
    env:
      TZ: America/New_York
    steps:
      - name: Checkout
        uses: actions/checkout@v4
      - name: Cypress run
        uses: cypress-io/github-action@v6
        with:
          working-directory: apps/your-app
          browser: chrome
          headed: true
          command: npm run cy:component
      - uses: actions/upload-artifact@v4
        if: failure()
        with:
          name: cypress-screenshots
          path: apps/your-app/cypress/screenshots
```

Where we are?


On Push
Unit
Component
~ 1min
Staging
UI integration
~ 3 min
Deploy
< 10 min
On Schedule
E2E
30 - 45 min


Login
New Claim
What happened?
Case 1
Case 2
Case 3
Case N
Page 1
Case 1
Case 2
Case 3
Case N
Page N
Summary

Page 1


Component 1
Component 4
Component 4
Component 5
Component 2
Component 3
Component 7
Component 6
Component 6

Q&A