Tackling Flaky Tests: Strategies for Reliability in Continuous Testing

Reasons for test flakiness

Timing dependencies

Tests that depend on timing conditions, such as delays or waiting for an asynchronous process to complete

Bad test data

Tests using randomly generated or inappropriate data as inputs, which could result in failures

Platform or environment issue

Tests that pass on one machine or configuration but fail on another

Resource constraints

Tests that fail due to resource limitations on the host. CPU, memory, and stuff like that

Dependence on external services

Tests that rely on external APIs, databases, or other services that might be unavailable or slow

Network connectivity

Kinda similar to 'platform or environment issue', but specifically the network connection being used to connect somewhere failing intermittently, resulting in connectivity issues

Order-dependency

Tests that depend on the order of execution, often caused by shared state between tests

Concurrency issues

Tests involving multiple threads or processes that access shared resources

UI Automation timing

Automated UI testing trying to interact with a page element which either does not exist, or is not ready.

The impact of flaky tests

- Flaky tests undermine developer confidence 🤔
- Flaky tests slow down CI/CD pipelines 🐢
- There's a monetary value associated with a test being flaky 💸

Flaky test stats (according to ChatGPT 🤔)

- Google reports that 16% of their tests are flaky
  https://blog.mergify.com/understanding-flaky-test-meaning-developers-guide

- 60% of developers regularly encounter flaky tests
  https://blog.mergify.com/understanding-flaky-test-meaning-developers-guide

- Order dependency is a dominant cause of flakiness, responsible for 59% of flaky tests, followed by test infrastructure problems at 28%
  https://arxiv.org/abs/2101.09077

1: Isolate and Identify Flaky Tests

Isolate and Identify Flaky Tests

Look for signs of flakiness

What does flakiness look like?

- Intermittent test failures without code changes
- Tests passing locally but failing in CI
- Outcome depends on non-deterministic factors like time or environment

Isolate and Identify Flaky Tests

Look for signs of flakiness

Re-run things a few times

Utilize test observability tools

tl;dr

Figure out which test are actually flaky

2: Optimize Test Design and Implementation

Optimize Test Design and Implementation

Make your tests ‘atomic’

What is an ‘atomic’ test?

- **Independent** - doesn’t rely on the execution of other tests
- **Tightly scoped** - tests a single unit of functionality
- **Deterministic** - avoid sources of entropy
- **Performant** - small and fast

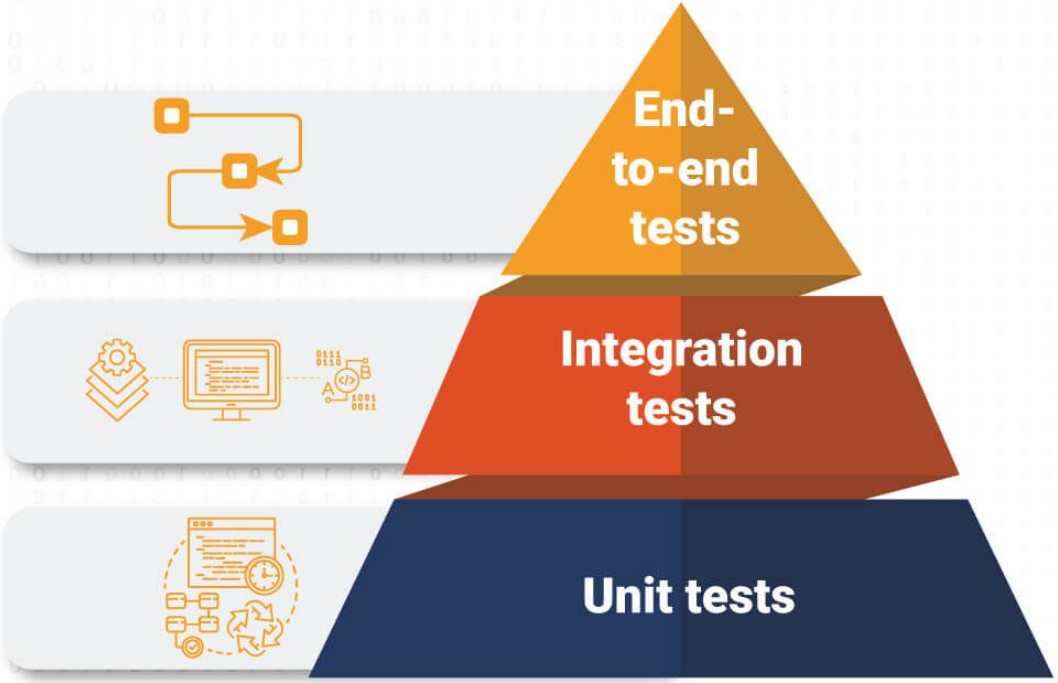
Optimize Test Design and Implementation

Make your tests ‘atomic’

Adopt mocking and stubbing

Structure tests hierarchically

The test pyramid thing



tl;dr

Invest time ensuring your tests are designed optimally

3: Improve Test Environment Stability

Improve Test Environment Stability

Strive to achieve a consistent test setup

How do I keep my build environment clean?

- Hosted/managed ephemeral runners
- Self-hosted runners (EC2, K8s etc) utilizing short-lived compute infra
- Docker 🐳

Improve Test Environment Stability

Strive to achieve a consistent test setup

Try and determine which failures are related to infrastructure

Replace external dependencies with mocks

tl;dr

Your test environment is crucial for reliability

4: Level-up your test management and monitoring tooling

Level-up your test management and monitoring tooling

Integration

RSpec | Minitest | Jest | Mocha | Cypress | Jasmine | Playwright

Swift | Android | pytest | Go | JUnit | .NET | Elixir | Rust

Level-up your test management and monitoring tooling

Integration

Observability

Flaky

Flaky unassigned | Mark flaky as resolved | Enabled

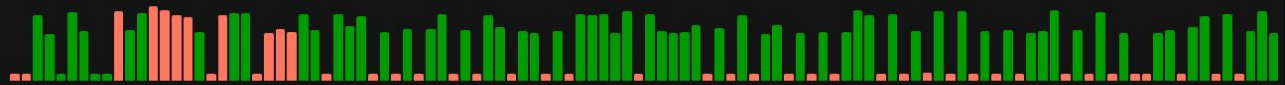
All branches | ./spec/ | github

Last 7 days | Last 14 days | Last 28 days

Test trends

Test reliability	Test execution count	Test execution duration (p50)	Test execution duration (p95)
68.79%	455	10.04s	16.75s

Recent test executions



Flaky instances | Failed test executions | Test Events

d469e1a2ab

View execution · Fri 17th Jan 2025 at 05:15 UTC · Build #127615

View execution · Fri 17th Jan 2025 at 05:15 UTC · Build #127615

```
Failure/Error: expect(bundle_next).to eq(true)
```

add specs

cb56acd2bd

a78b7d6e9f

Buildkite

Level-up your test management and monitoring tooling

Integration

Observability

Flaky test detection

Test assignment

Test quarantining

tl;dr

Observability is key

5: Foster a Reliability-Focused Culture

Foster a Reliability-Focused Culture

Make unblocking everyone else a top priority

Quarantining tests is fine, but then fix things after

When necessary, swarm

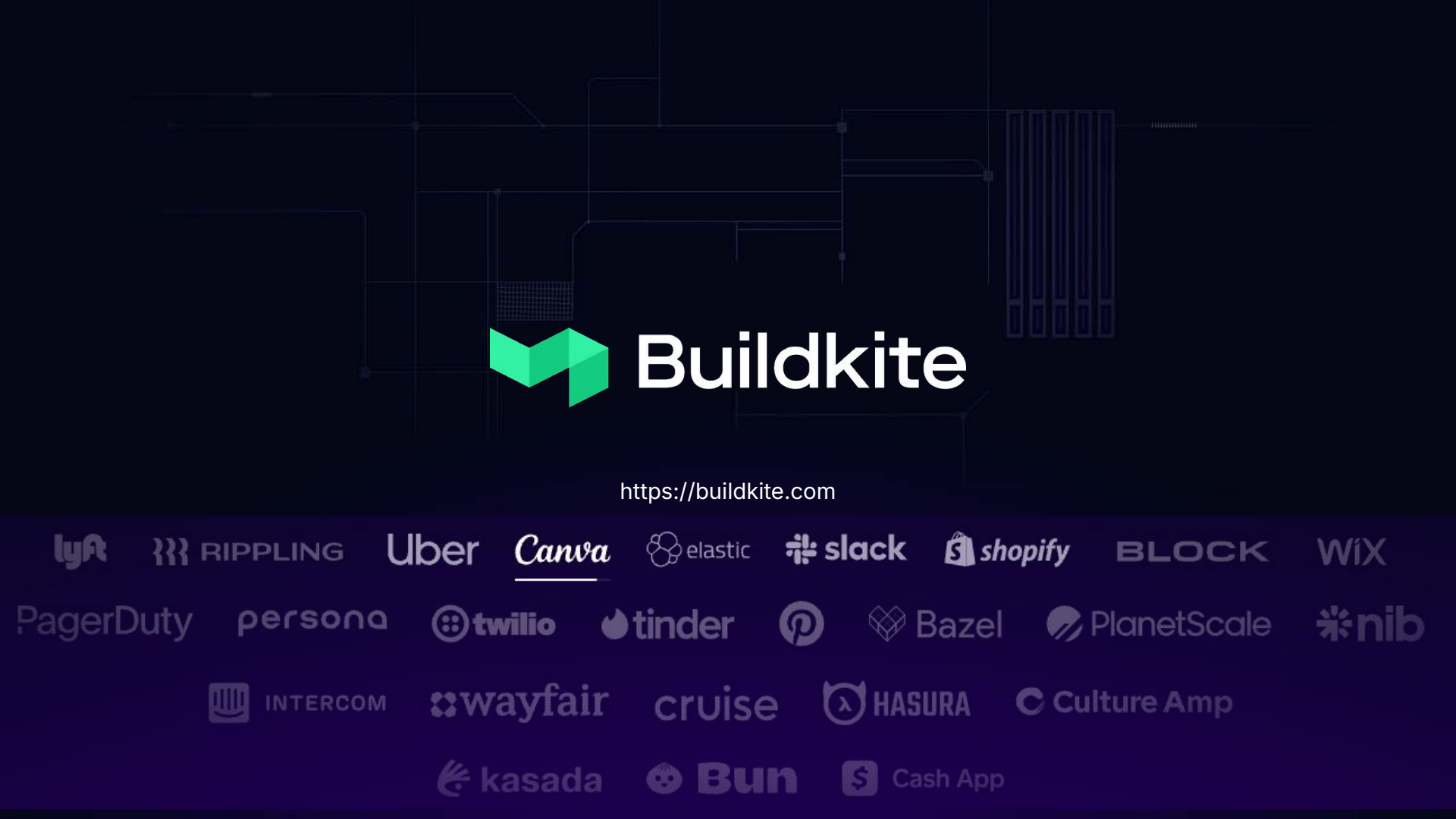
Avoid silos of information

tl;dr

People and processes are as important as tools

Recap

- Figure out which tests are flaky
- Optimize test design
- Work on environment stability
- Use the best tooling for the job
- Foster a reliability culture


Buildkite
https://buildkite.com
lyft
RIPPLING
Uber
Canva
elastic
slack
shopify
BLOCK
WiX
PagerDuty
persona
twilio
tinder
Bazel
PlanetScale
nib
INTERCOM
wayfair
cruise
HASURA
Culture Amp
kasada
Bun
Cash App