

**Continuous
integration - that's
not what they meant**

@ClareSudbery



@ClareSudbery



What is continuous integration (CI)?

@ClareSudbery



@ClareSudbery



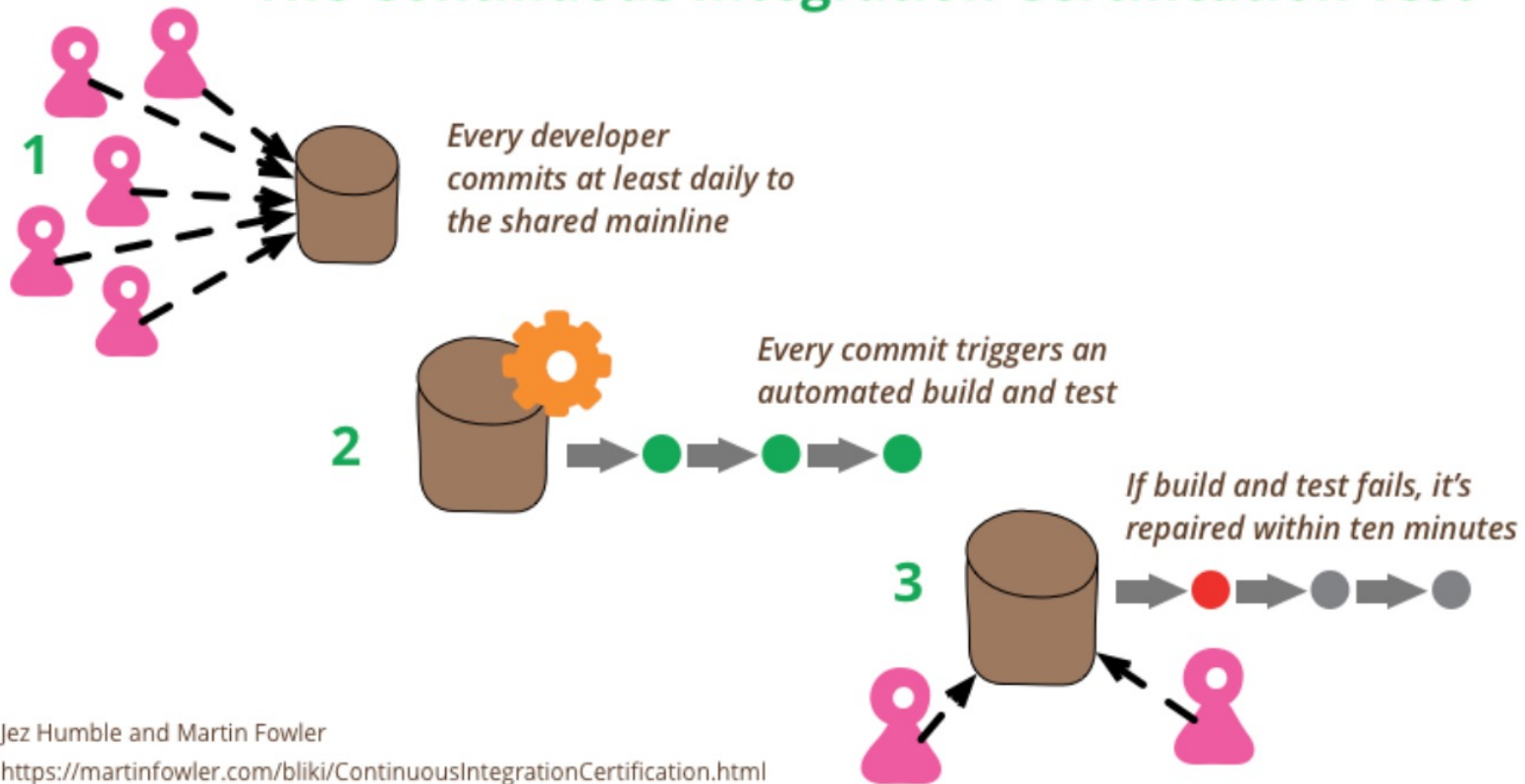
Jez Humble's Continuous Integration Test

@ClareSudbery



@ClareSudbery

The Continuous Integration Certification Test



Jez Humble and Martin Fowler

<https://martinfowler.com/bliki/ContinuousIntegrationCertification.html>



@ClareSudbery

"No code sits unintegrated for more than a couple of hours. At the end of every development episode, the code is integrated with the latest release."

Kent Beck
Extreme Programming Explained, 1999

@ClareSudbery



@ClareSudbery



Continuous integration vs Continuous Delivery

@ClareSudbery



@ClareSudbery



Trunk-Based Development (TBD)
=
Continuous Integration (CI)

@ClareSudbery



@ClareSudbery



**Trunk-based development (TBD).
Better. Faster. Happier.**

@ClareSudbery



@ClareSudbery



I'm not the boss of you

@ClareSudbery



@ClareSudbery



**Branches can slow you
down and increase risk**

@ClareSudbery



@ClareSudbery

I'll tell you a story...

@ClareSudbery



@ClareSudbery

ROBERT McKEE
story

Substance, structure, style,
and the principles of screenwriting



@ClareSudbery



@ClareSudbery



Start state
~
Turning point
~
New state

@ClareSudbery



Palladium

Palladium

@ClareSudbery



Branches
~
Merge hell
~
Trunk-based development (TBD)

@ClareSudbery



@ClareSudbery




Trunk-based development (TBD)
~
Regulatory non-compliance
~
Branches, PRs and code reviews

@ClareSudbery



@ClareSudbery



Branches, PRs and code reviews
~
Lack of trust
~
More Pull Requests (PRs)

@ClareSudbery



@ClareSudbery

More Pull Requests (PRs)
~
Even less trust
~
Move slowly

@ClareSudbery

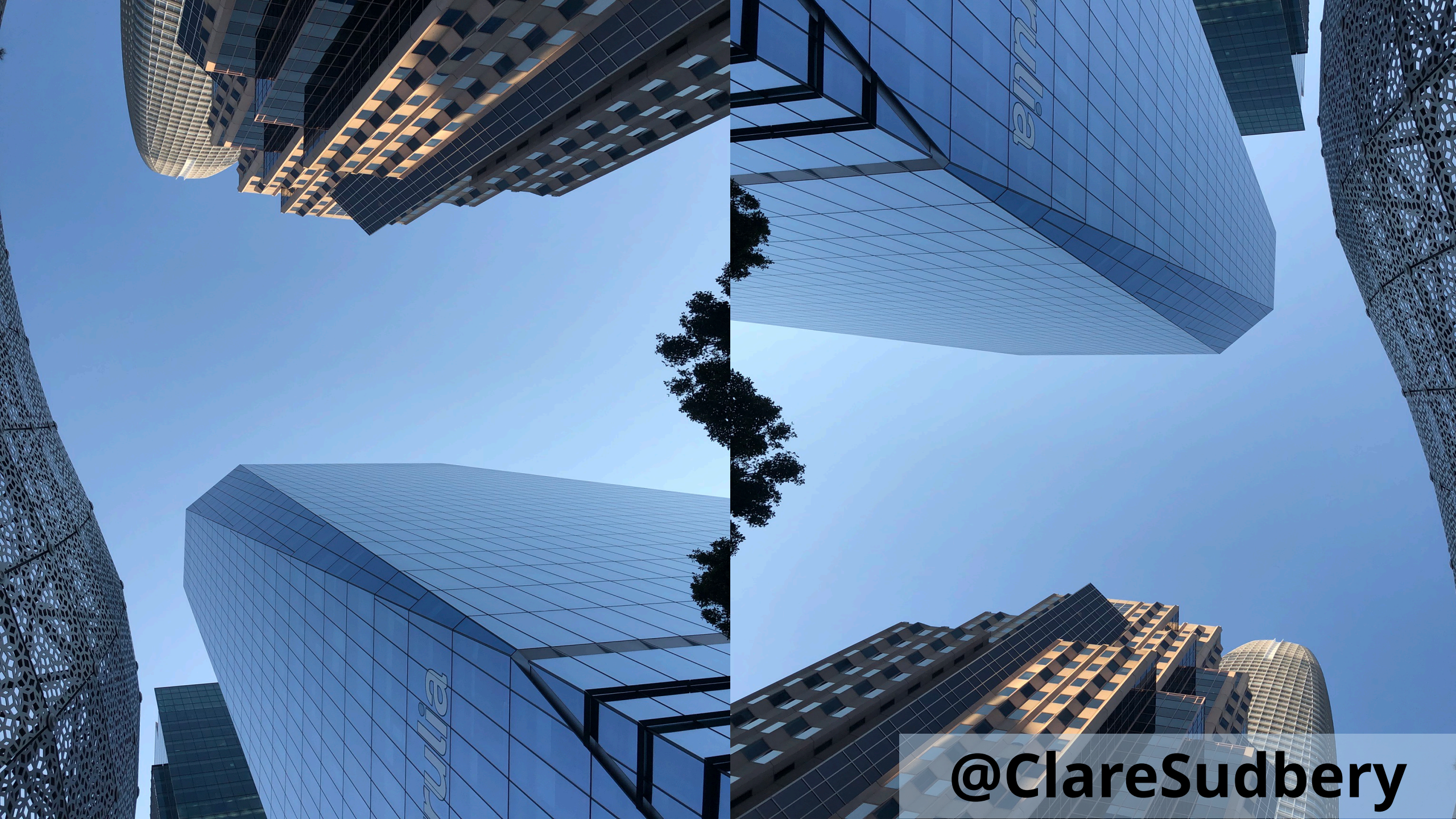


@ClareSudbery

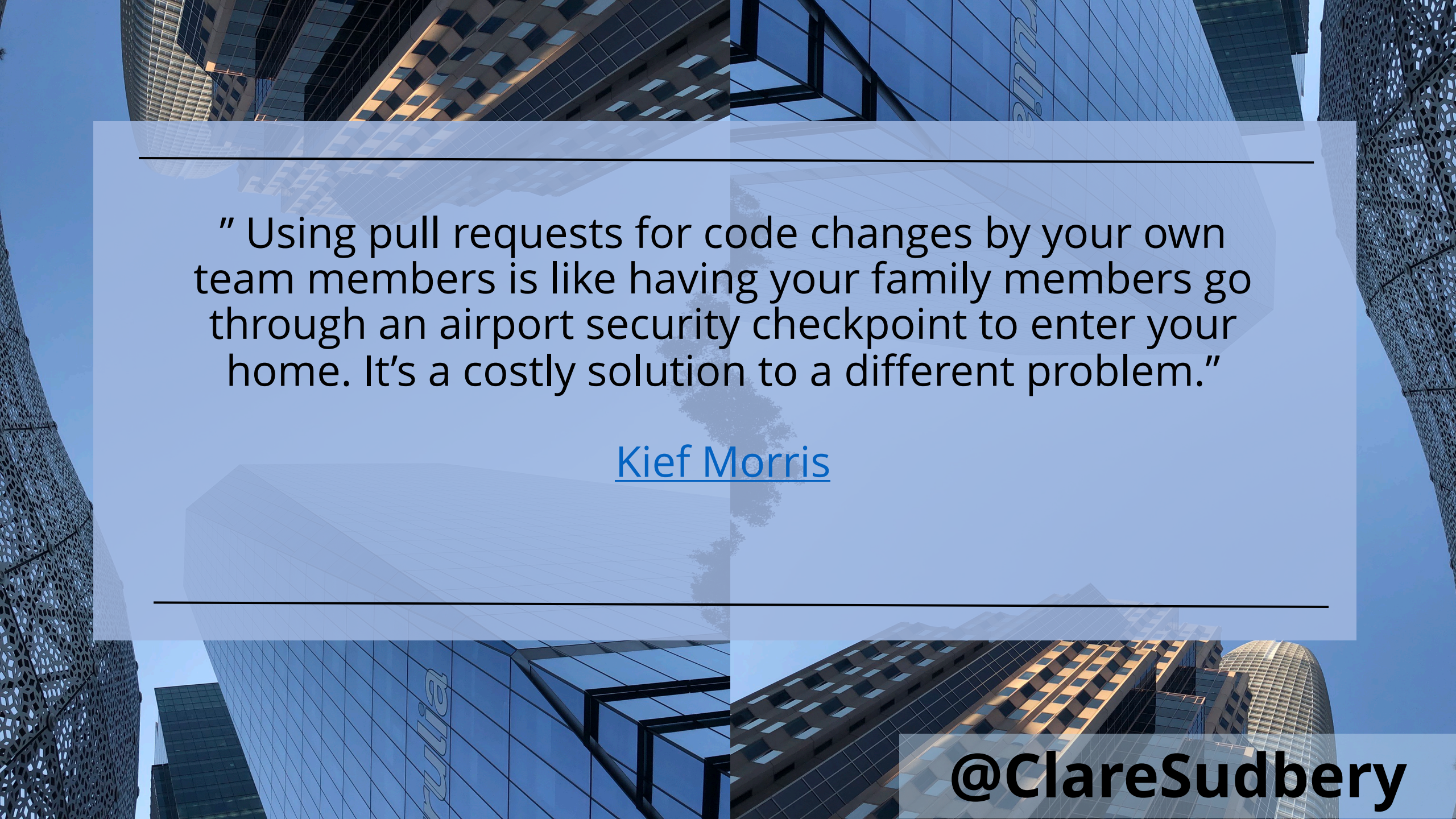
“PR feedback is the last point in the dev cycle where feedback is useful, and often the most wasteful.”

[Dan Abel](#)

@ClareSudbery



@ClareSudbery




“ Using pull requests for code changes by your own team members is like having your family members go through an airport security checkpoint to enter your home. It’s a costly solution to a different problem.”

[Kief Morris](#)

@ClareSudbery



@ClareSudbery



“The bigger the PR, the more likely it will suffer from the so-called “LGTM syndrome”. If the PR is big, reviewers have way less incentive to build in the quality.”

[Dragan Stepanović](#)

@ClareSudbery



@ClareSudbery


“My experience, across the board, has been that regulators prefer these approaches [TBD and CD], once they come to understand them, because they provide a better quality experience all around.”

Dave Farley

@ClareSudbery



@ClareSudbery



"The more complex the compliance mechanisms around change, the lower the quality of the software. ... I believe that it is not possible to implement a genuinely compliant, regulated system in the absence of CD!"

[Dave Farley](#)

@ClareSudbery



allure

OP

@ClareSudbery




**Review whole code base - or
specific areas - at regular
intervals**

@ClareSudbery



@ClareSudbery




Lack of trust + move slowly
~
Big bang trunk-based development (TBD)
~
Can't deploy (manual tests)

@ClareSudbery

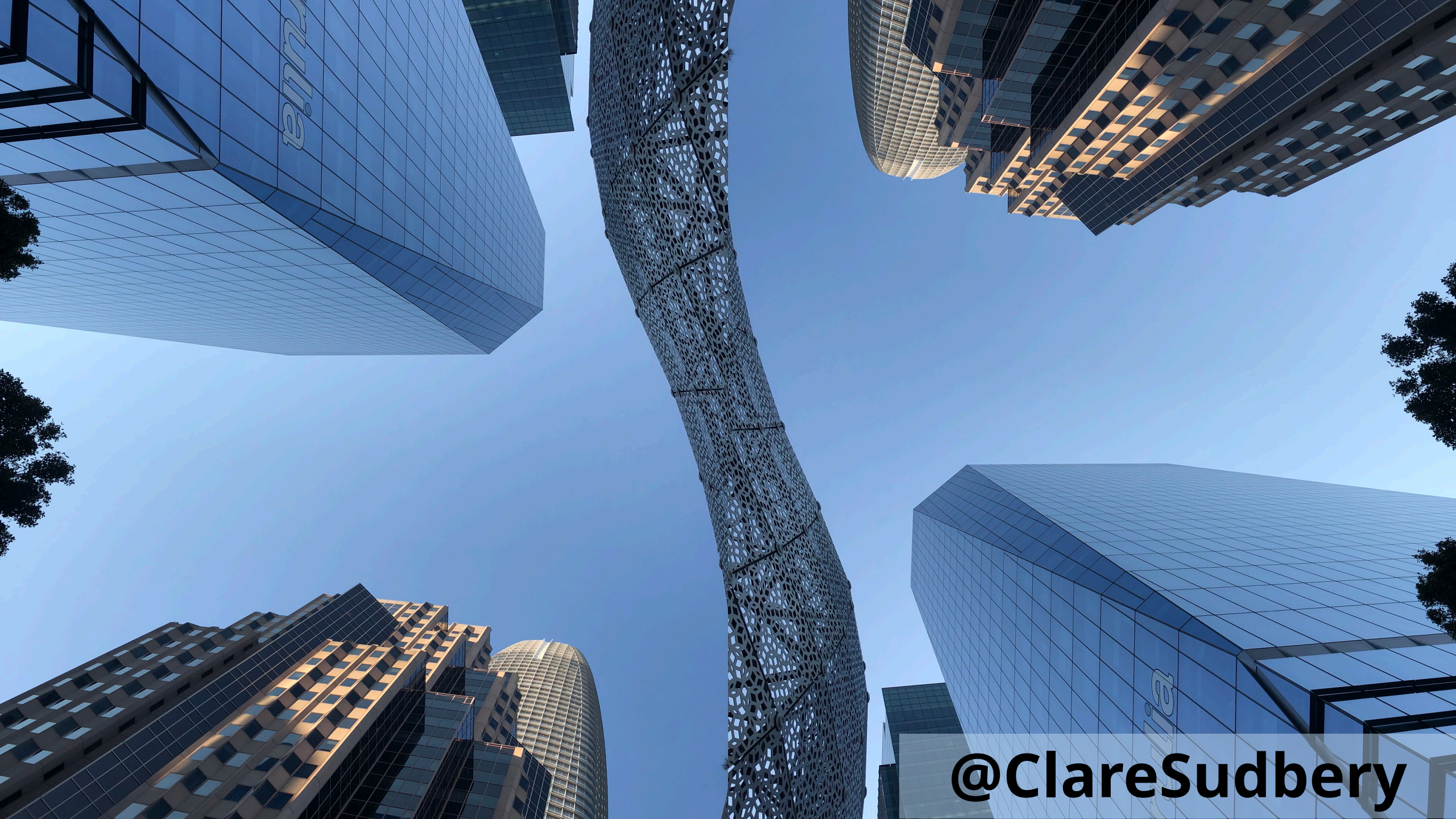


@ClareSudbery



Big bang TBD + manual tests
~
Deploy without testing + system failure
~
BLAME

@ClareSudbery



@ClareSudbery



BLAME
~
Guilt and shame
~
GET HELP

@ClareSudbery



@ClareSudbery

Technical Agile Coaching with the **Samman method**

By Emily Bache
Foreword by Kent Beck



“In order to be successful in modern development organizations, software developers need new skills.

These skills are not easily learnt on a short training course or at a university.

Practices like Continuous Integration and Test-Driven Development require developers to change their minute-by-minute habits and ways of working.”

[Emily Bache](#)

[Samman Technical Coaching Society](#)



@ClareSudbery

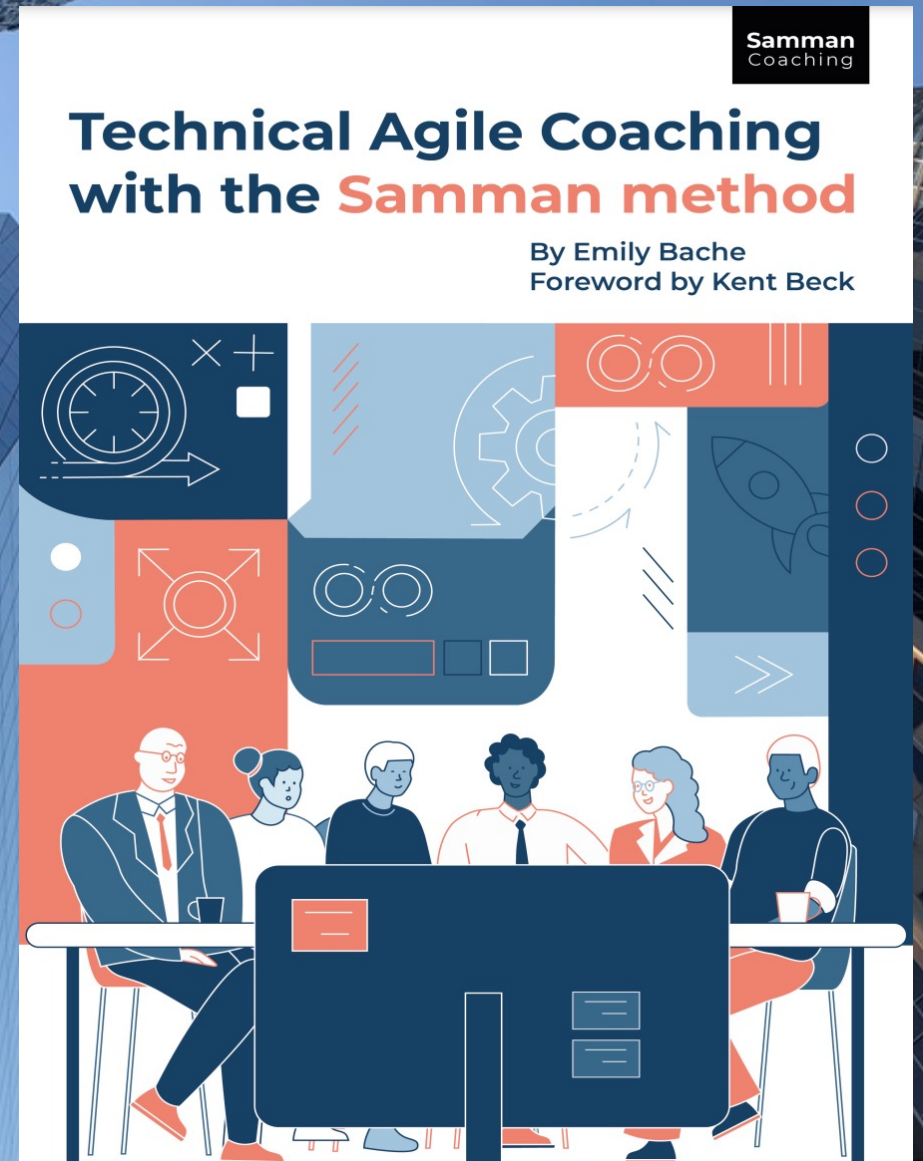
“Software development is best seen as a learning process where running programs are the proof of learning, rather than a factory or production process.

Walking a new path together along with an experienced guide reduces time spent not learning because of fear.”

Kent Beck

**Foreword, Technical Agile Coaching
with the Samman Method**

@ClareSudbery





@ClareSudbery


A low-angle, upward-looking photograph of several modern skyscrapers. The buildings feature glass facades and intricate architectural details, including a prominent tower with a perforated facade. The sky is a clear, bright blue. A semi-transparent white rectangular box is centered over the image, containing text and horizontal lines.

GET HELP
~
Pairing + mob/ensemble programming
~
TDD, refactoring and automated testing

@ClareSudbery

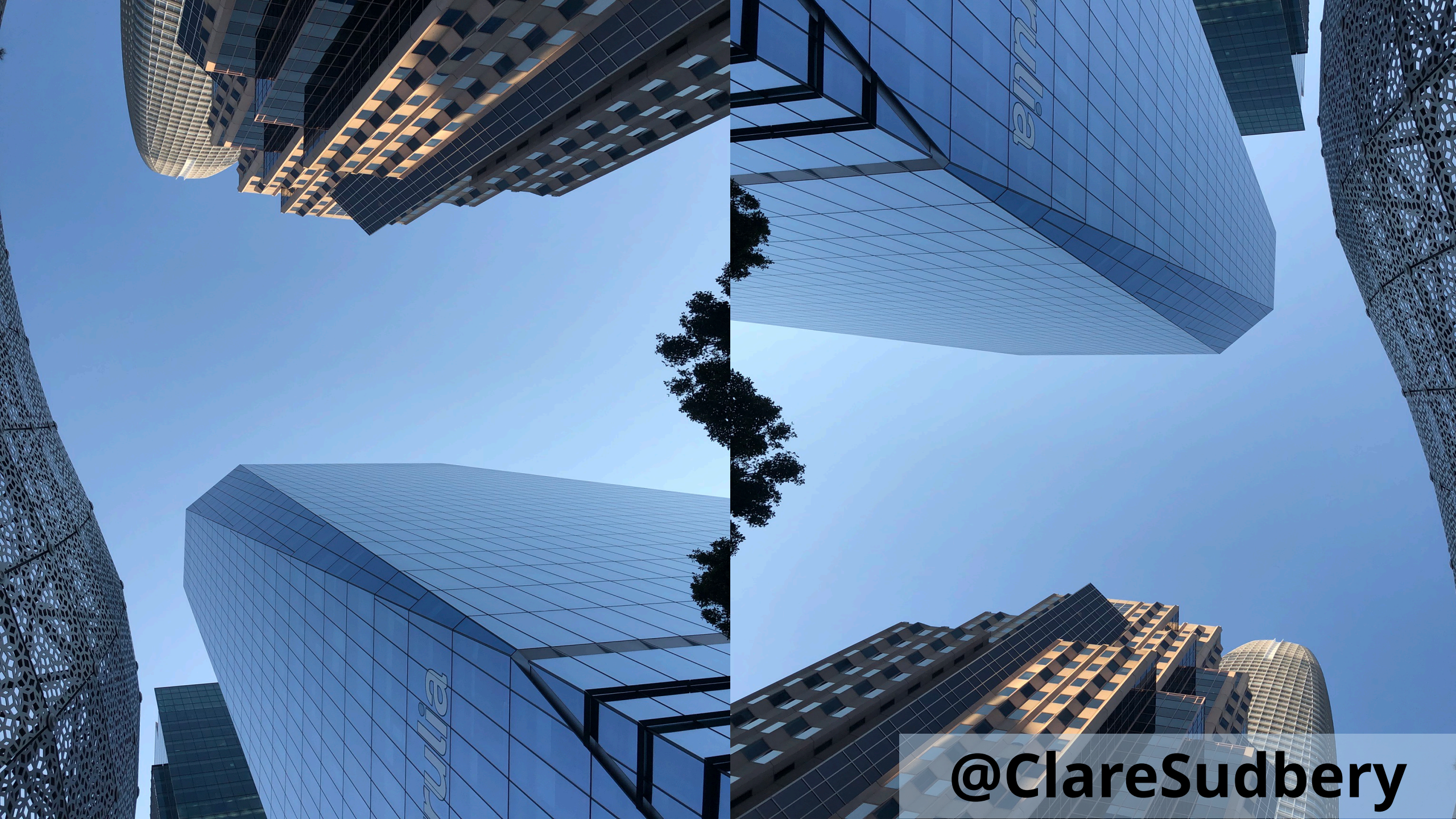


@ClareSudbery




Tightly coupled code
~
Slow builds
~
Minimise dependencies

@ClareSudbery



@ClareSudbery



Continuous Integration requires “relentless work to simplify systems architecture on an ongoing basis to ensure that this automation isn’t prohibitively expensive to create and maintain.”

Accelerate - Forsgren et al

@ClareSudbery



@ClareSudbery

Nicole Forsgren, Jez Humble, and Gene Kim

Accelerate: Building and Scaling High Performing Technology Organizations

(IT Revolution Press, 2018)

THE SCIENCE OF LEAN SOFTWARE AND DEVOPS

ACCELERATE

Building and Scaling High Performing
Technology Organizations


Nicole Forsgren, PhD
Jez Humble, and Gene Kim

*with forewords by Martin Fowler and Courtney Kissler
and a case study contributed by Steve Bell and Karen Whitley Bell*

@ClareSudbery



@ClareSudbery



Minimise dependencies
~
More modules + slower tests
~
Introduce faster, incremental tests

@ClareSudbery



@ClareSudbery



Stop running tests locally
~
Broken build
~
Run tests locally + keep build green

@ClareSudbery



@ClareSudbery



Not pushing frequently
~
Merge hell
~
MMMSS + push frequently

@ClareSudbery



@ClareSudbery

Many More Much Smaller Steps – First Sketch

September 29, 2021 / 5 minutes of reading / [Muses](#), [Podcasts](#)

Part 1 of 5 in the series [Many Much More Smaller Steps](#)



Many More Much Smaller Steps - First Sketch | Anchor. #129

PawCast with GeePaw Hill • By GeePaw Hill • Sep 28, 2021



00:00

Support

Share

07:53

The first plank of my take on fixing the trade is MMMSS: If you want more value faster, take Many More Much Smaller Steps.



@ClareSudbery

“Integrating small changes regularly into your code is usually less painful than a big merge at the end of a longer period of time.”

- [Trisha Gee](#)

@ClareSudbery



@ClareSudbery



CTBD (Continuous trunk-based development)

~
Minor merge issues

~
Throw things away

@ClareSudbery



allure

OP

@ClareSudbery



What about feature flags?

@ClareSudbery



@ClareSudbery

"To get maximum bang for your TBD buck, all developers all the time should be running with 100% of the features turned on.

A great deal of the benefit of CI and TBD both is in the extensive 'pre-testing' workout the code gets, as it's used by the rest of your team."

- Geepaw Hill

@ClareSudbery



@ClareSudbery

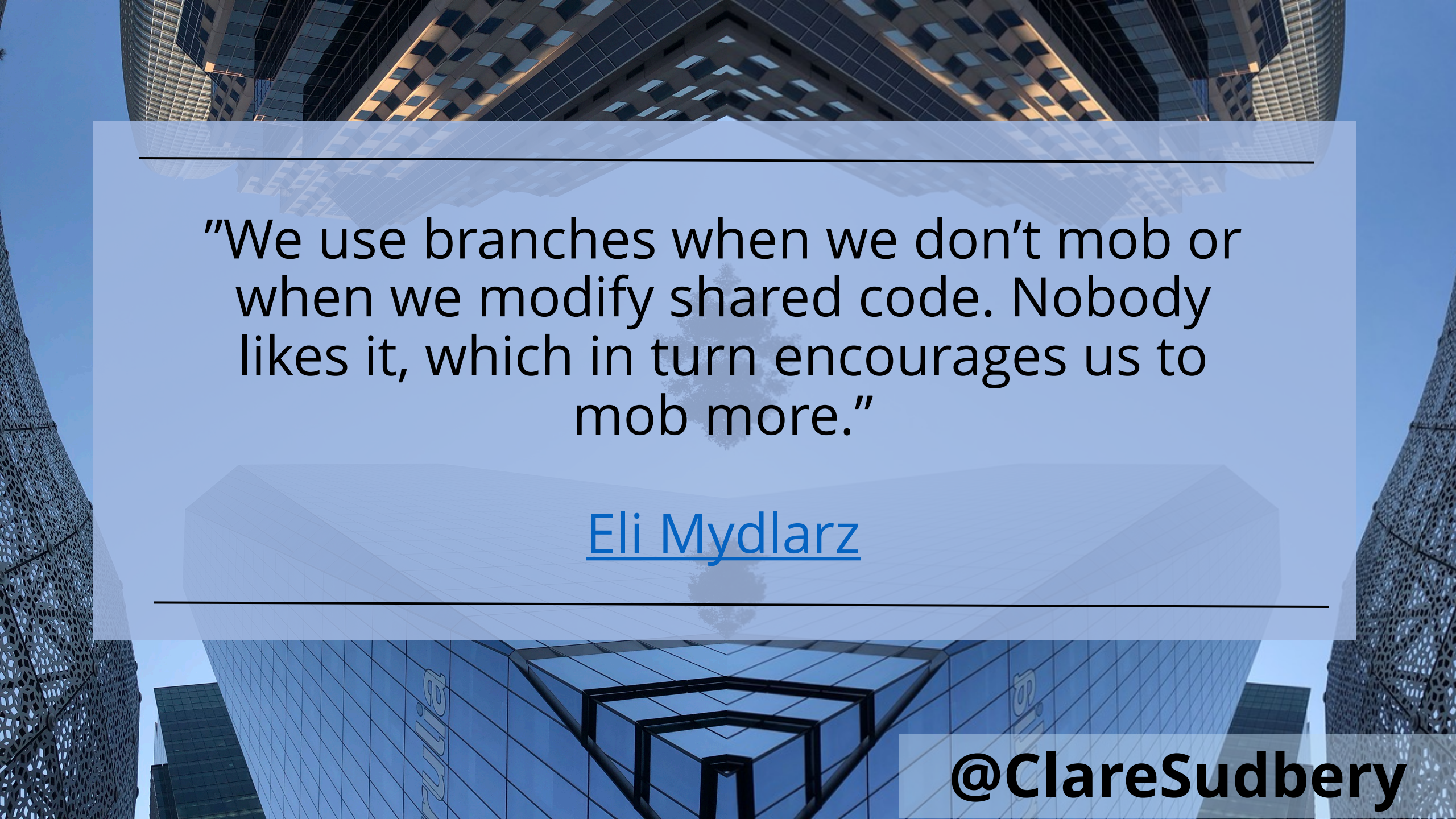


CONCLUSIONS

@ClareSudbery



@ClareSudbery



"We use branches when we don't mob or when we modify shared code. Nobody likes it, which in turn encourages us to mob more."

[Eli Mydlarz](#)

@ClareSudbery



allure

OP

@ClareSudbery



***I'm not the boss of you
...but Dave says...***

@ClareSudbery



@ClareSudbery



"Any form of branching runs counter, in principle, to the ideas of Continuous Integration.

Don't branch! Don't branch! Don't branch!"

[Dave Farley](#)

@ClareSudbery



@ClareSudbery



(just sayin')

@ClareSudbery



@ClareSudbery

2. But why does trunk-based development make things better?

- The Benefits of TBD
 - Small steps
 - MMMSS
 - Fast feedback
 - Avoiding merge hell
 - Reducing risk
 - Faster integration, fewer queues
 - Collective ownership
 - Better quality code
- Evidence
 - DORA Metrics and the State Of DevOps



@ClareSudbery



“ Trunk-based development ... has been shown for years to accelerate software delivery velocity.”

DORA report, 2022

@ClareSudbery



allure

OP

@ClareSudbery



Summary of CTBD techniques

@ClareSudbery



allure

OP

@ClareSudbery



**You can do some of these things
without doing all of them**

@ClareSudbery



@ClareSudbery

- Implement automated incremental testing
- Minimize dependencies
- Speed up your tests
- Keep the build green
- Build and test on your own machine first
- Push frequently
- Use TDD (Test-Driven Development)
- Refactor frequently
- Throw things away

@ClareSudbery



@ClareSudbery

- MMMSS (Many More Much Smaller Steps)
- Get help
- Build trust
- Collaborate
- Be patient
- Interactive code reviews
- And if you have all that, then maybe...
 - Avoid branches
 - No Pull Requests (PRs)



@ClareSudbery

Links

[DORA annual reports](#)

[Geepaw Hill, MMMSS \(Many More Much Smaller Steps\)](#)

[Martin Fowler, Continuous Integration](#)

[Jez Humble's Continuous Integration test](#)

Books

[Accelerate, Nicole Forsgren et al](#)

[DevOps Handbook](#)

[Extreme Programming Explained, Kent Beck](#)

@ClareSudbery



@ClareSudbery

More links

[My O'Reilly report on trunk-based development](#)

[Trisha Gee, "Why I prefer trunk-based development"](#)

[Eli Mydlarz, "How to stop branching"](#)

[Kief Morris on pull requests \("Infrastructure as Code"\)](#)

[Dave Farley on compliance and continuous delivery](#)

[Dave Farley on trunk-based development](#)

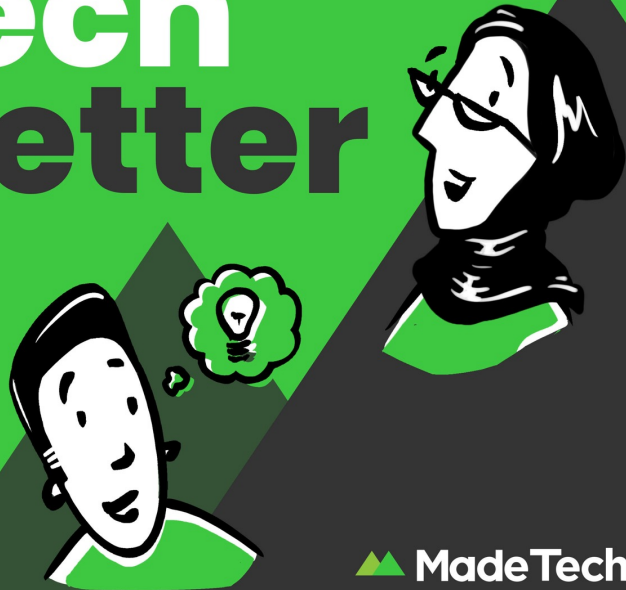
[Dragan Stepanović on Async Code Reviews](#)

@ClareSudbery



@ClareSudbery

making tech better



*I hosted a
podcast..*

tinyurl.com/tech-better-pod

@ClareSudbery

O'REILLY®

What Is Trunk-Based Development?

An Explanation of TBD and How It Can Improve Your Software Delivery

Clare Sudbery

REPORT

tinyurl.com/TBD-clare

@ClareSudbery

Sudbery Software Engineering Ltd

Does your team want help to improve their software delivery?



Coaching, keynotes & workshops



Events and info: tinyurl.com/csudbery

Mastodon: mastodon.social/@ClareSudbery

LinkedIn: tinyurl.com/lin-csudbery

Podcast: tinyurl.com/tech-better-pod





@ClareSudbery

A low-angle, upward-looking photograph of several modern skyscrapers. The buildings feature various architectural styles, including glass facades and perforated metal screens. The sky is a clear, bright blue. A semi-transparent white rectangular box is centered over the image, containing the text 'Questions / discussion' and two horizontal lines. In the bottom right corner, there is a semi-transparent dark grey box containing the text '@ClareSudbery'.

Questions / discussion

@ClareSudbery