goto;

GOTO
AMSTERDAM 2023

#GOTOams

goto;

GOTO
Guide

LET US HELP YOU

Ask questions through **the app**

★★★★★

also remember to rate session

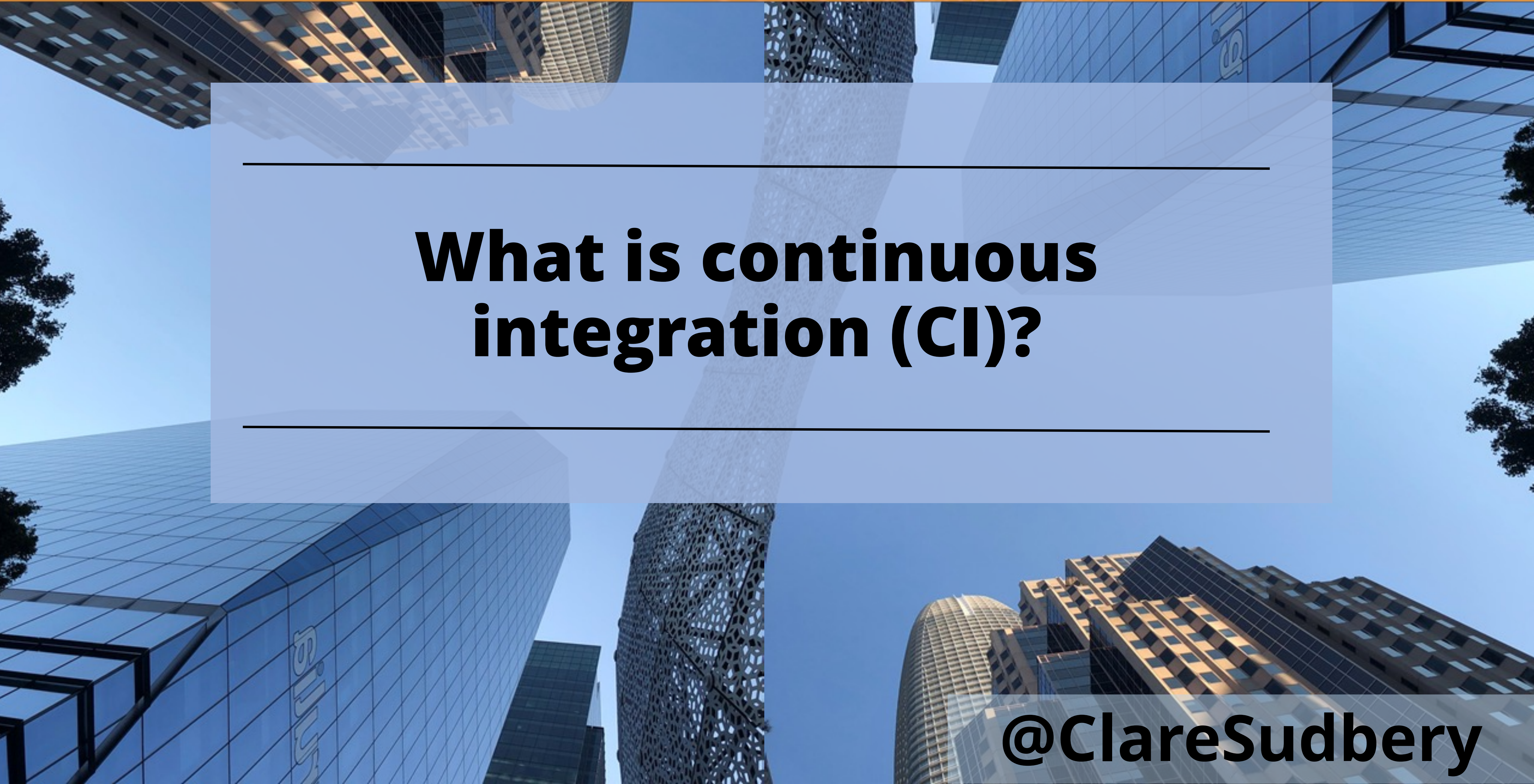
THANK YOU!

#GOTOams

Continuous integration – that's not what they meant

@ClareSudbery

@ClareSudbery



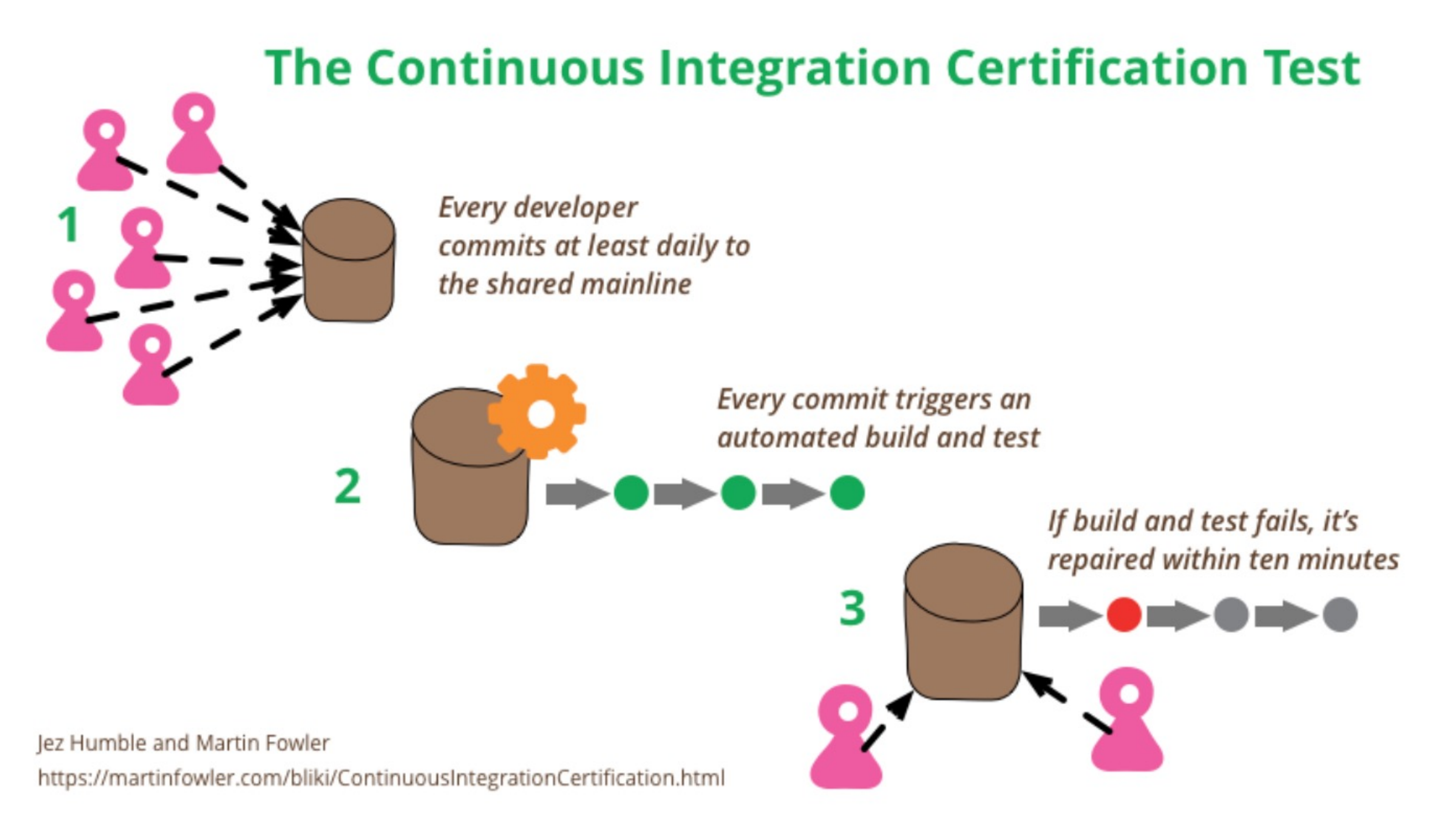
What is continuous integration (CI)?

@ClareSudbery

@ClareSudbery

Jez Humble's Continuous Integration Test

@ClareSudbery



The Continuous Integration Certification Test

1 *Every developer commits at least daily to the shared mainline*

2 *Every commit triggers an automated build and test*

3 *If build and test fails, it's repaired within ten minutes*

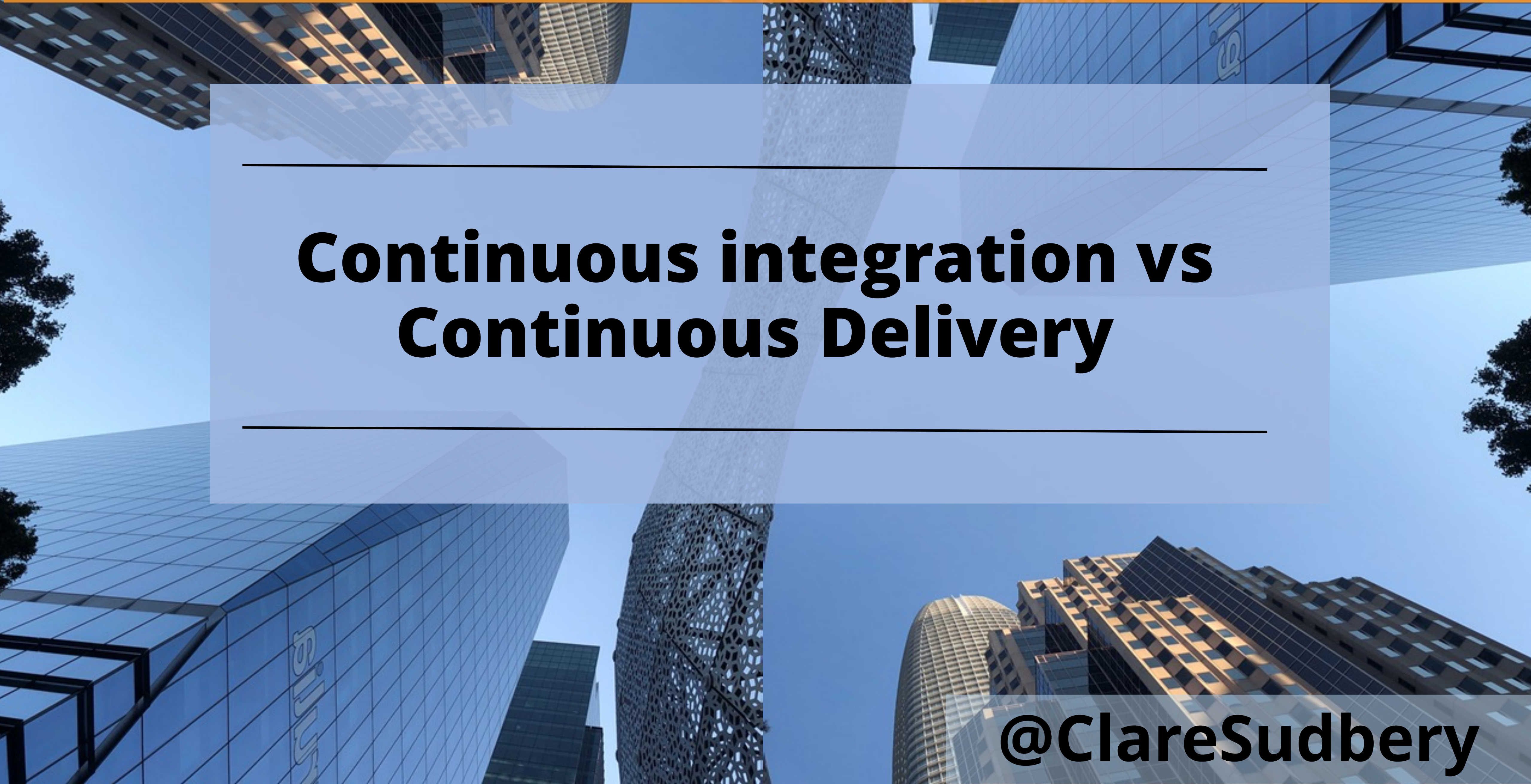
Jez Humble and Martin Fowler
https://martinfowler.com/bliki/ContinuousIntegrationCertification.html

@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



Continuous integration vs Continuous Delivery

@ClareSudbery

@ClareSudbery

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

@ClareSudbery

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

@ClareSudbery

@ClareSudbery

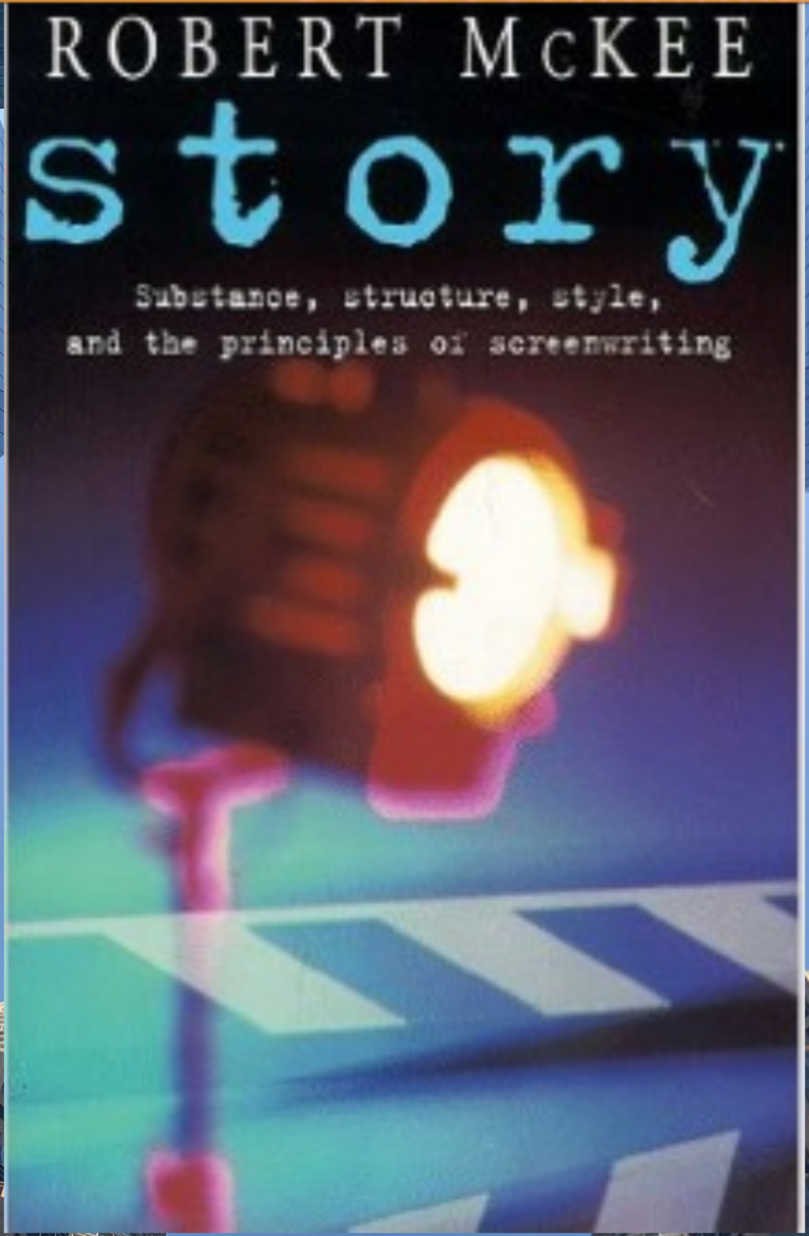
I’m not the boss of you

@ClareSudbery

I’ll tell you a story…

@ClareSudbery

goto;



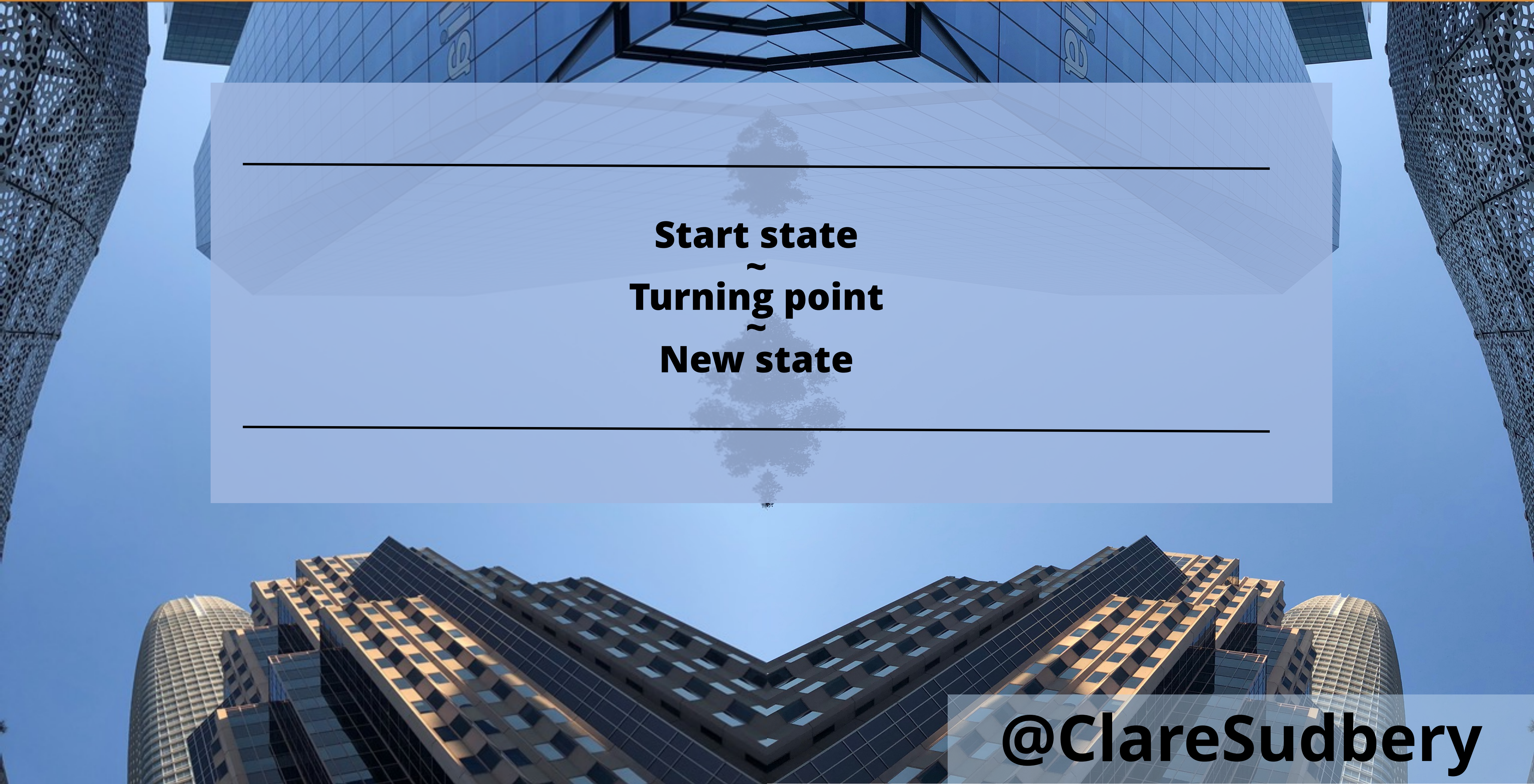
ROBERT McKEE

story

Substance, structure, style,
and the principles of screenwriting

@ClareSudbery

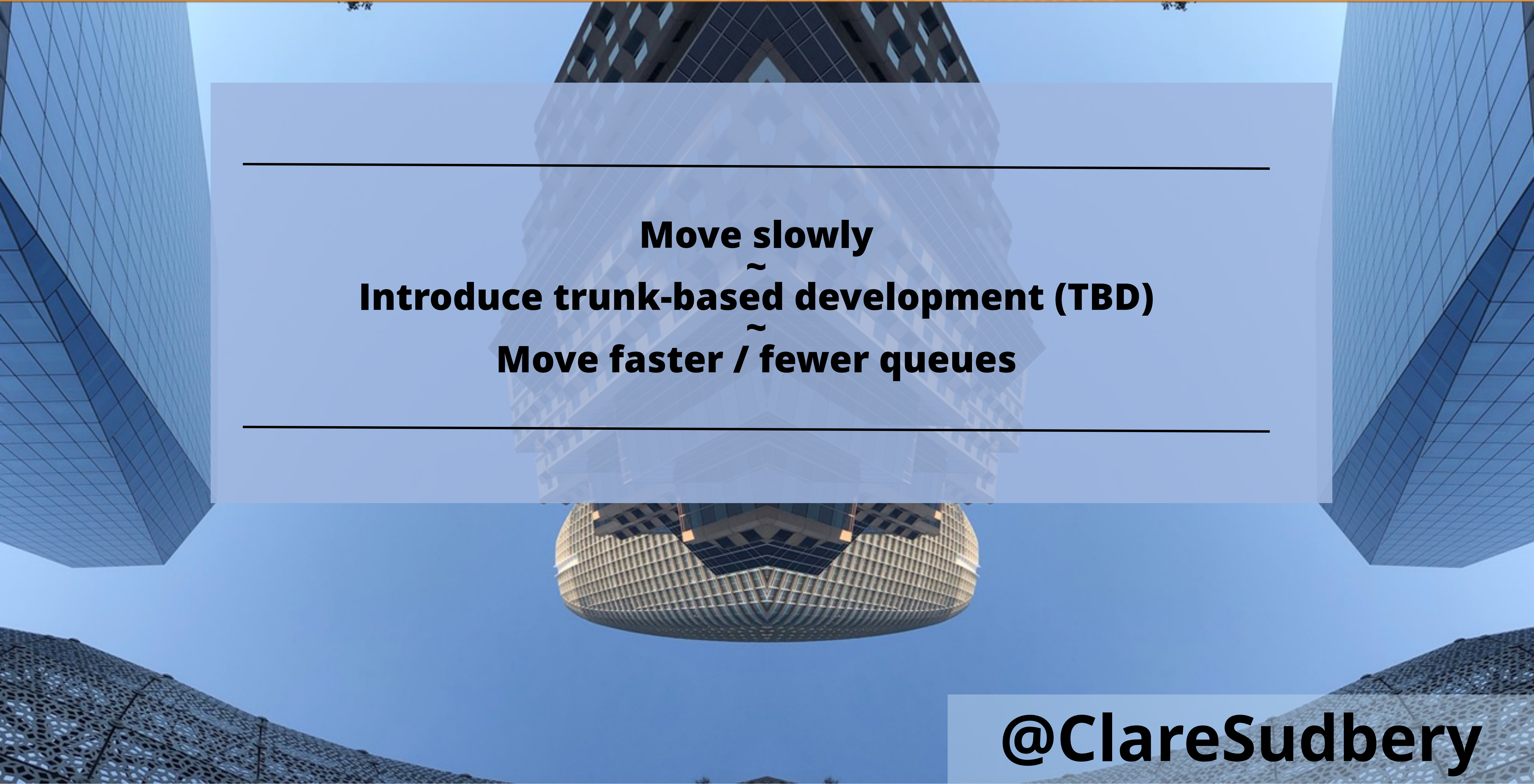
@ClareSudbery



Start state
~
Turning point
~
New state

@ClareSudbery

@ClareSudbery



Move slowly

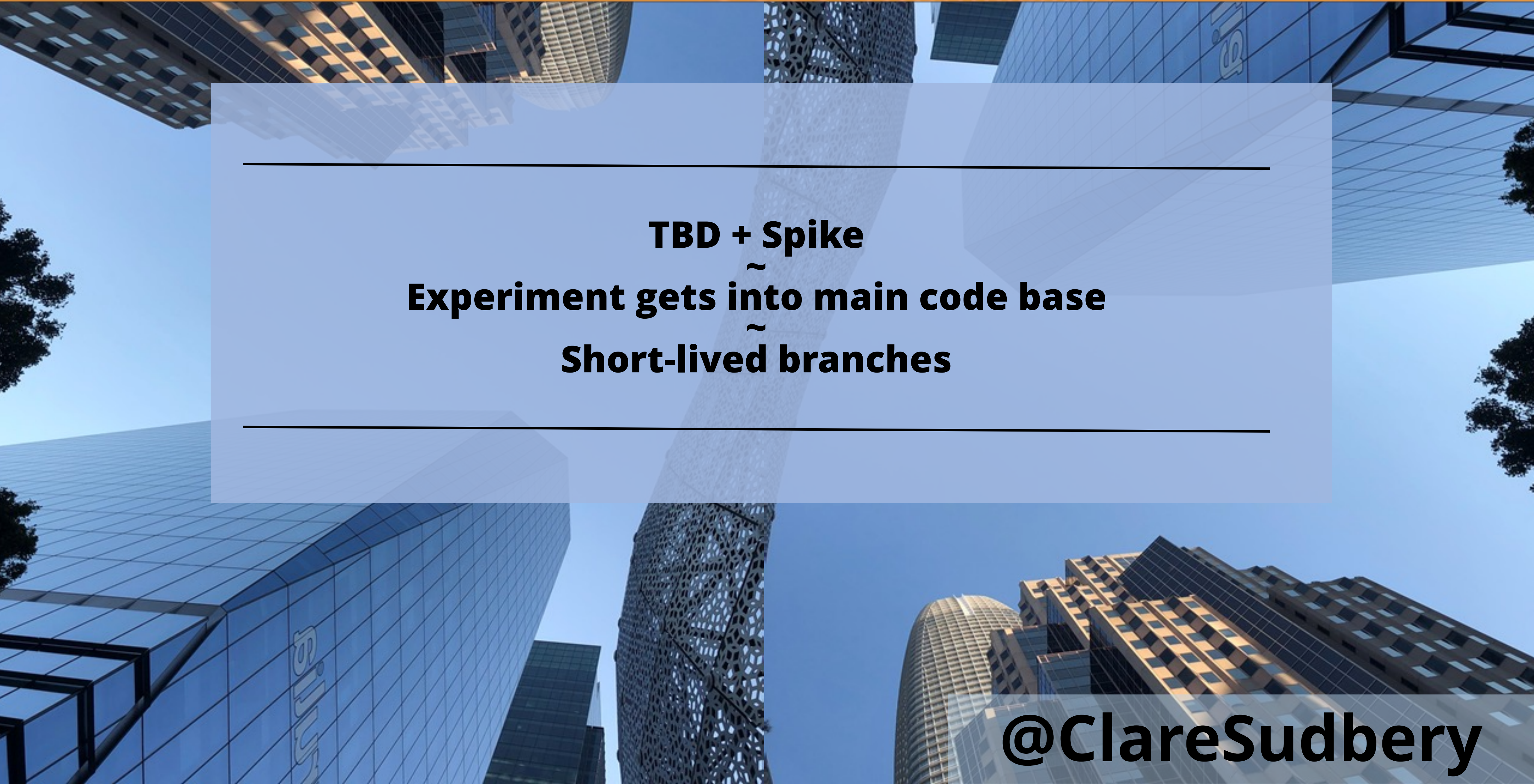
~

Introduce trunk-based development (TBD)

~

Move faster / fewer queues

@ClareSudbery



TBD + Spike

~

Experiment gets into main code base

~

Short-lived branches

@ClareSudbery

Short-lived branches

~

Merge hell

~

Go back to trunk-based development (TBD)

@ClareSudbery

@ClareSudbery

Trunk-based development (TBD)

~

Regulatory non-compliance

~

Branches, PRs and code reviews

@ClareSudbery

Branches, PRs and code reviews

~

Lack of trust

~

More Pull Requests (PRs)

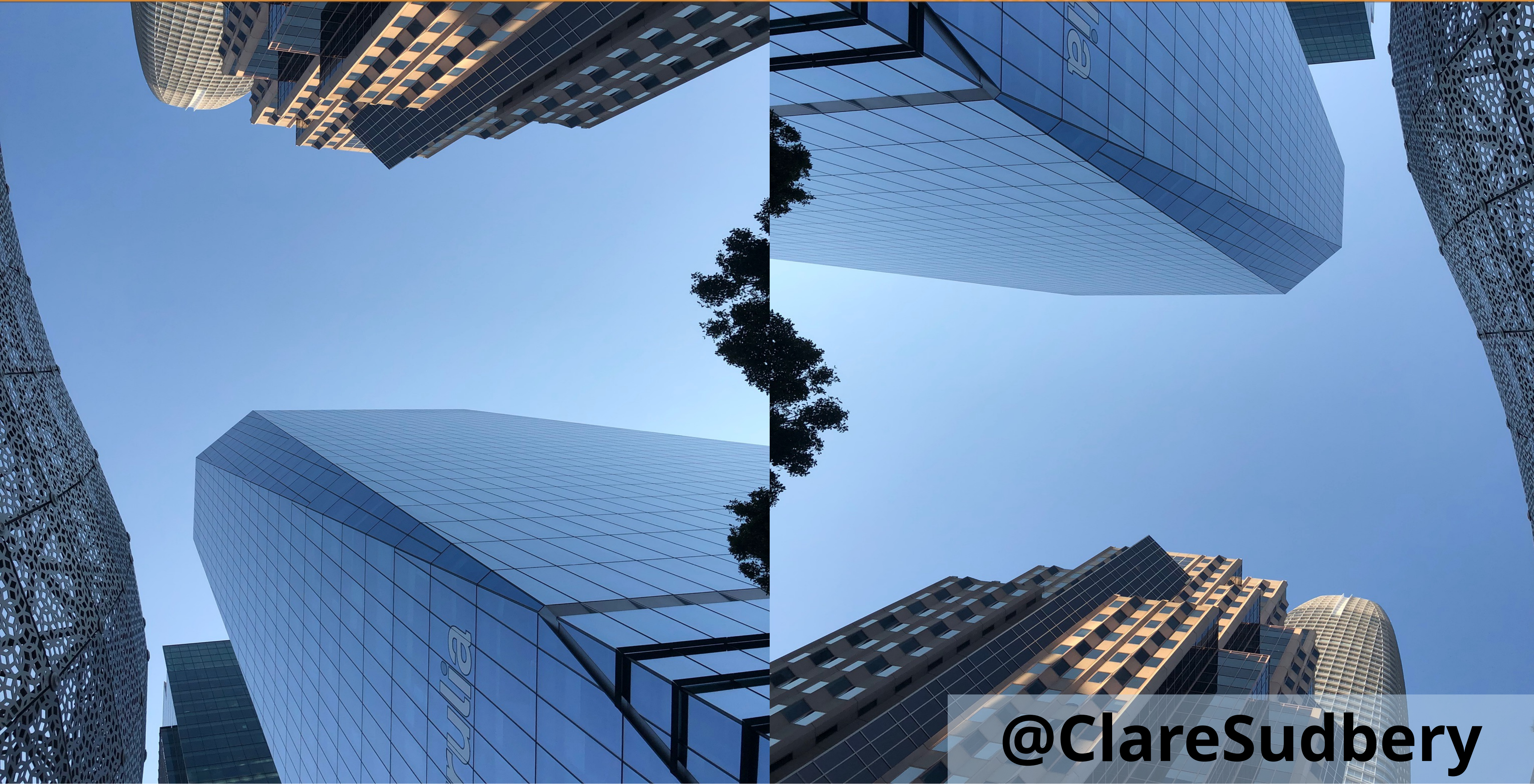
@ClareSudbery

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

@ClareSudbery

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

Dan Abel

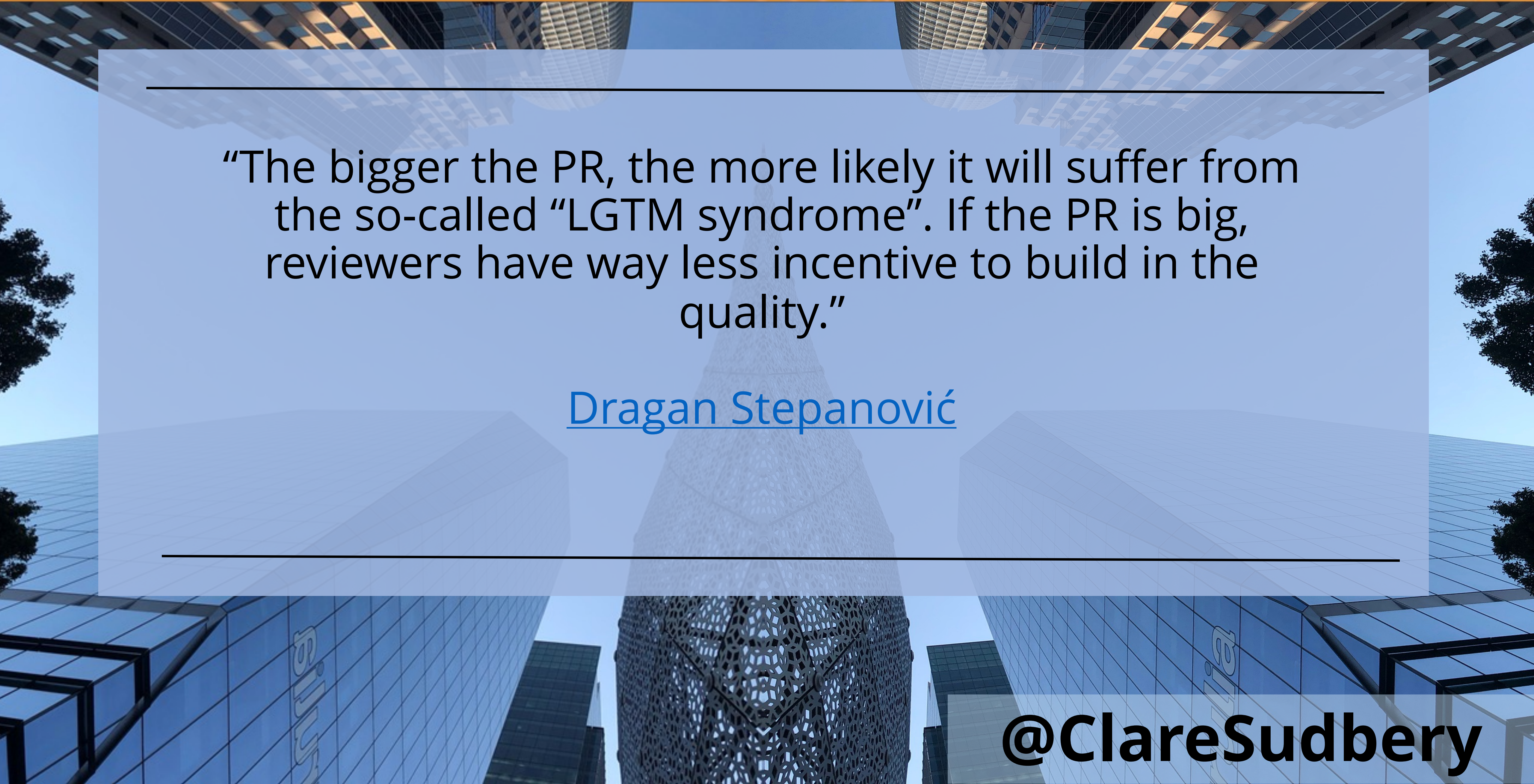


@ClareSudbery

> ” 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

> “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

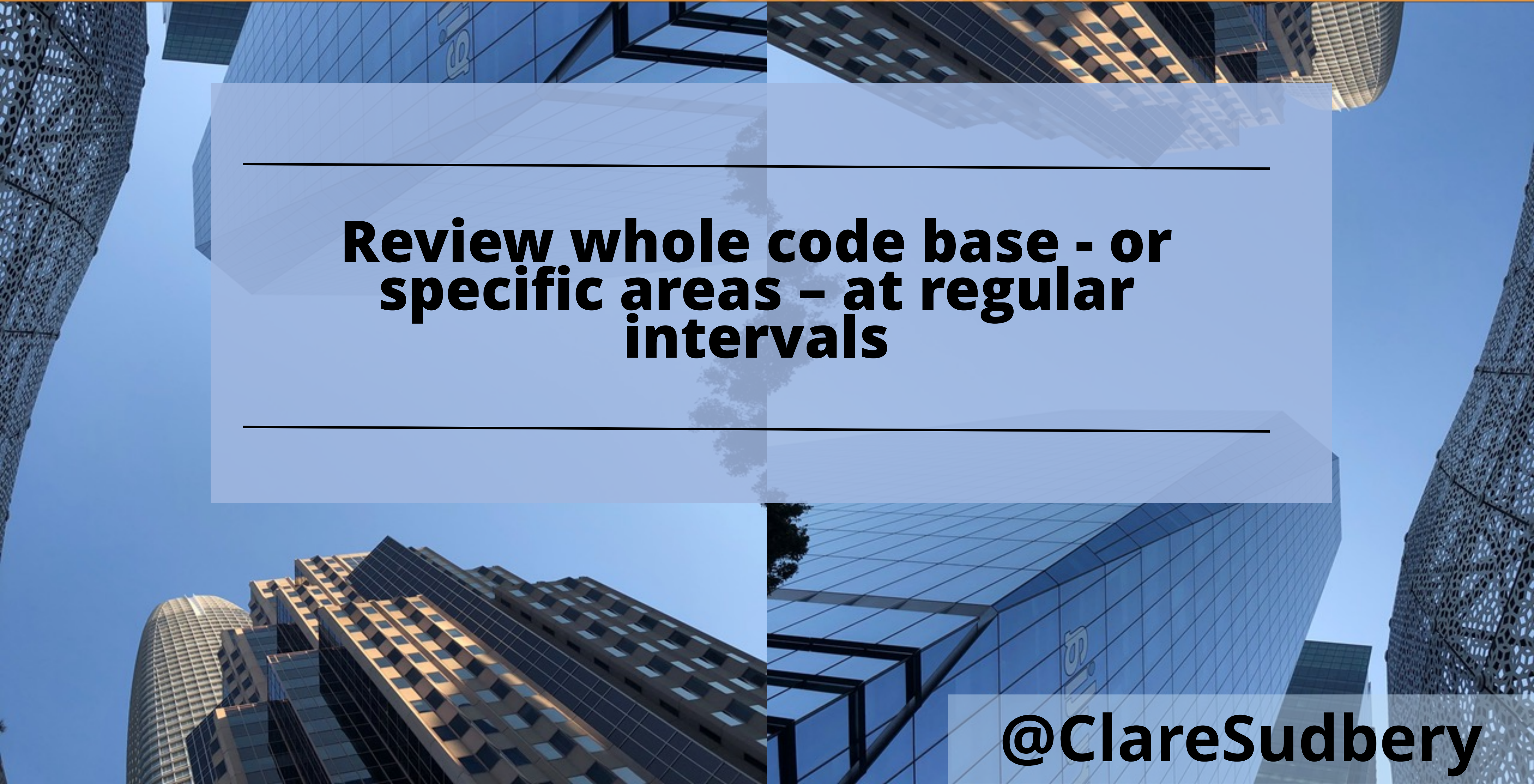
@ClareSudbery

”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

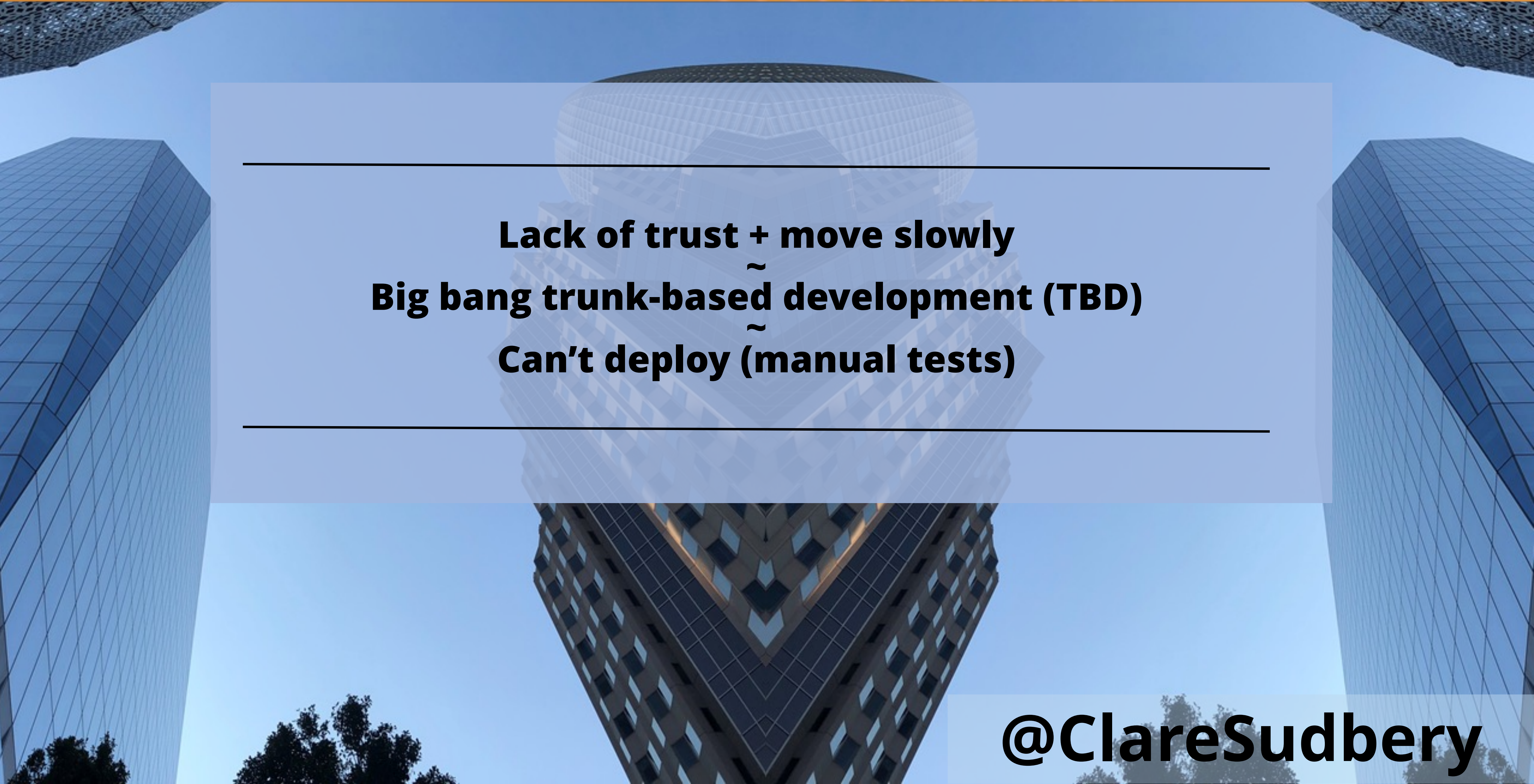
@ClareSudbery



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

Big bang TBD + manual tests

~

Deploy without testing + system failure

~

BLAME

@ClareSudbery

BLAME
~
Guilt and shame
~
GET HELP

@ClareSudbery

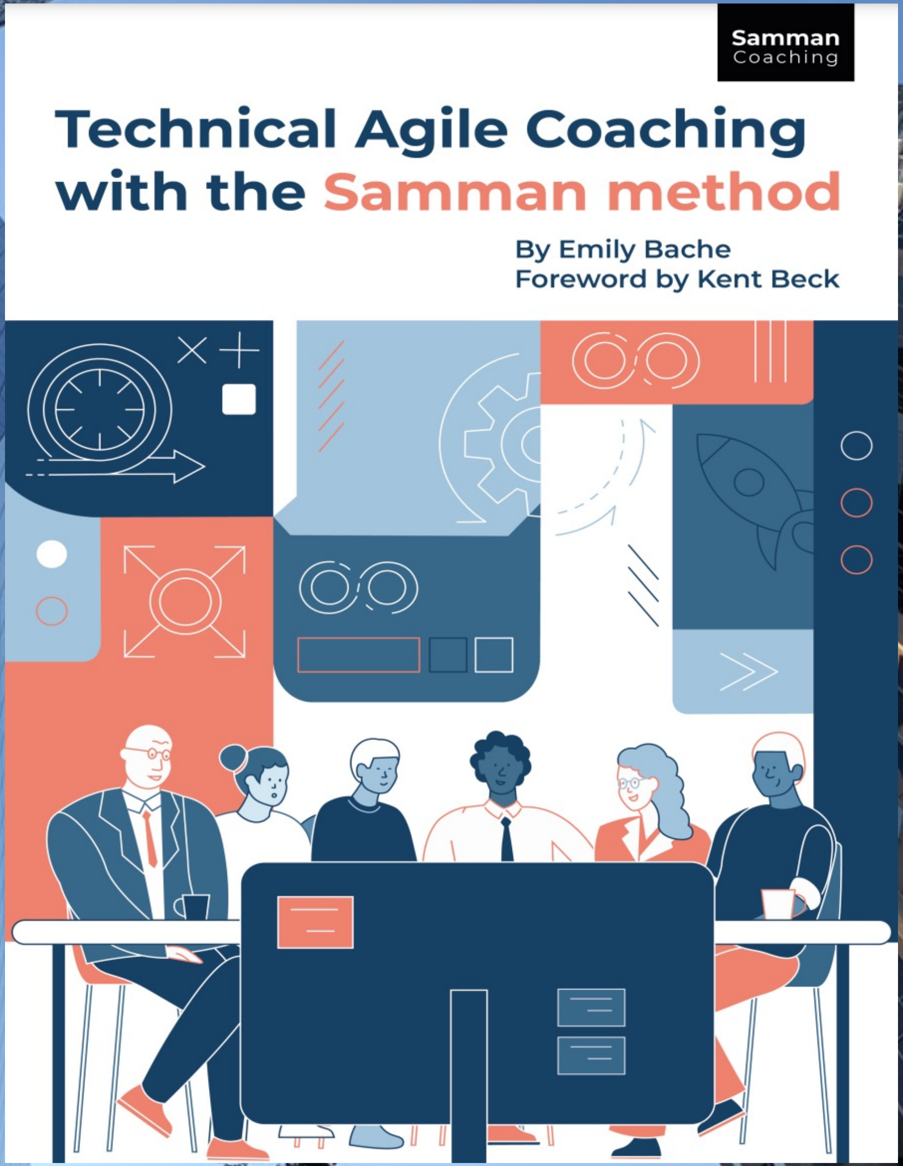
"In order to be successful in modern development organizations, software developers need new skills.

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]



Samman Coaching

Technical Agile Coaching with the Samman method

By Emily Bache
Foreword by Kent Beck

@ClareSudbery

GET HELP

~

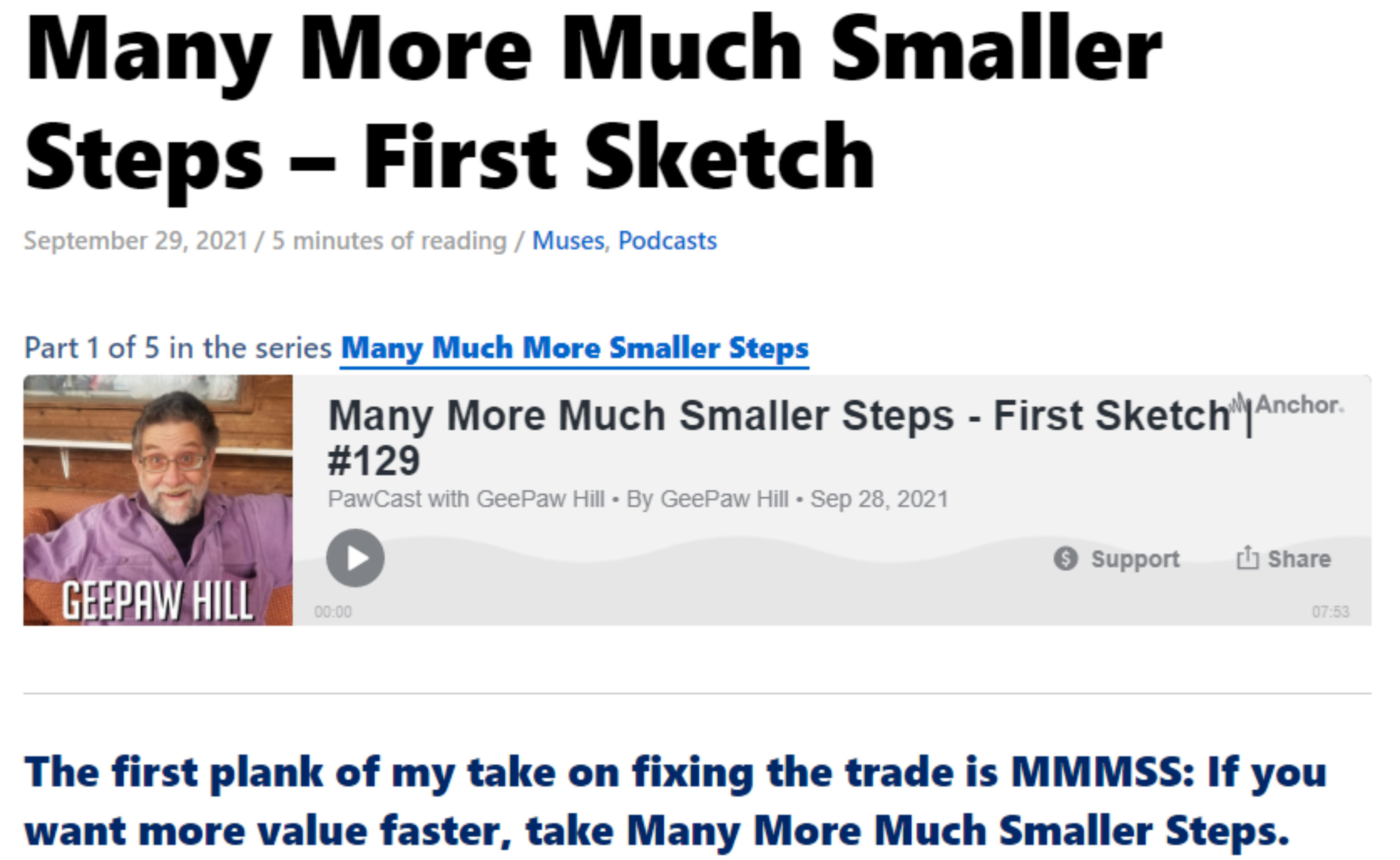
Many more much smaller steps (MMMSS) in journey

~

Trust, collaboration and patience

@ClareSudbery

https://www.geepawhill.org/2021/09/29/many-more-much-smaller-steps-first-sketch/



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 | #129

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

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

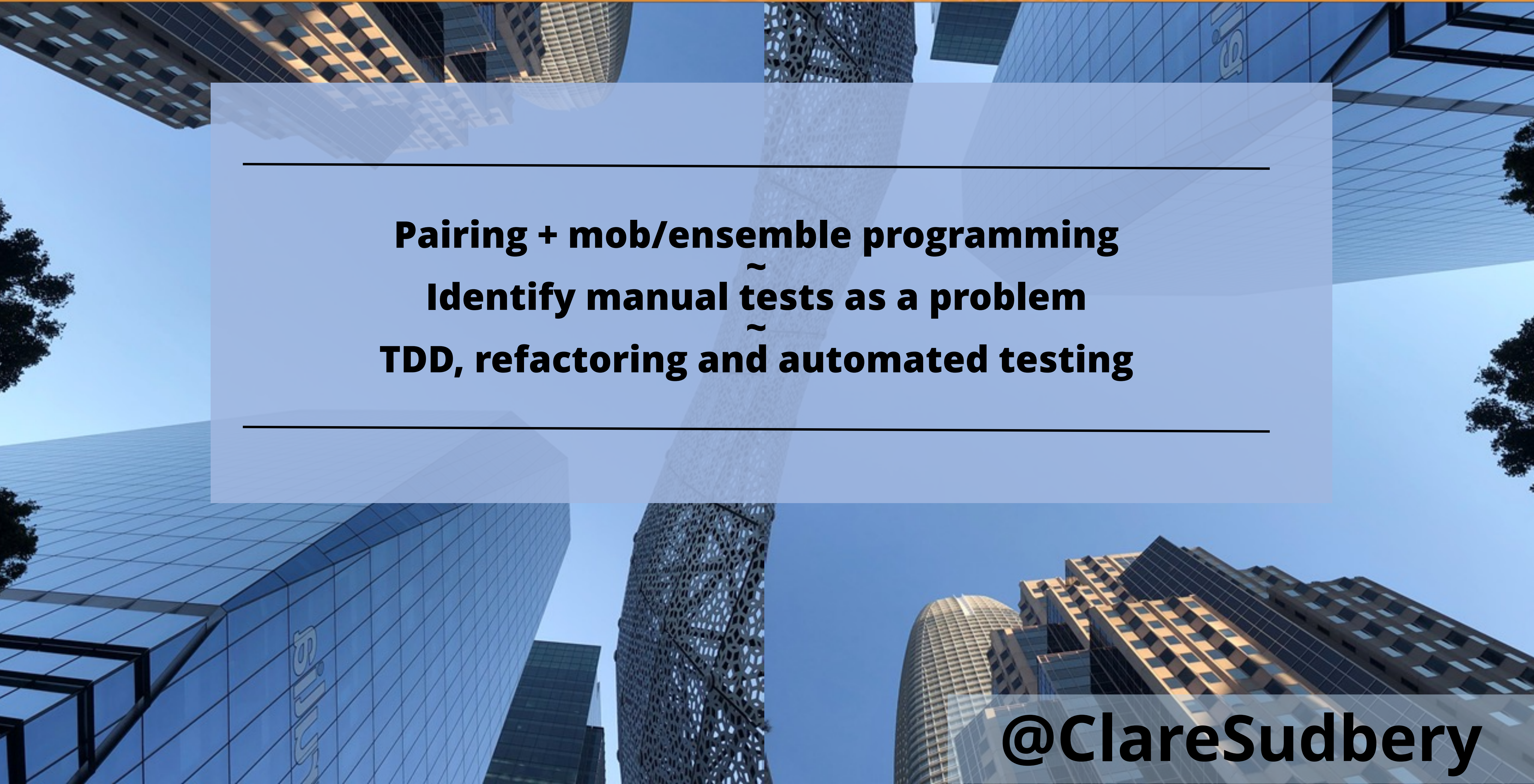
@ClareSudbery

@ClareSudbery

Pair and mob/ensemble programming

@ClareSudbery

@ClareSudbery



Pairing + mob/ensemble programming

~

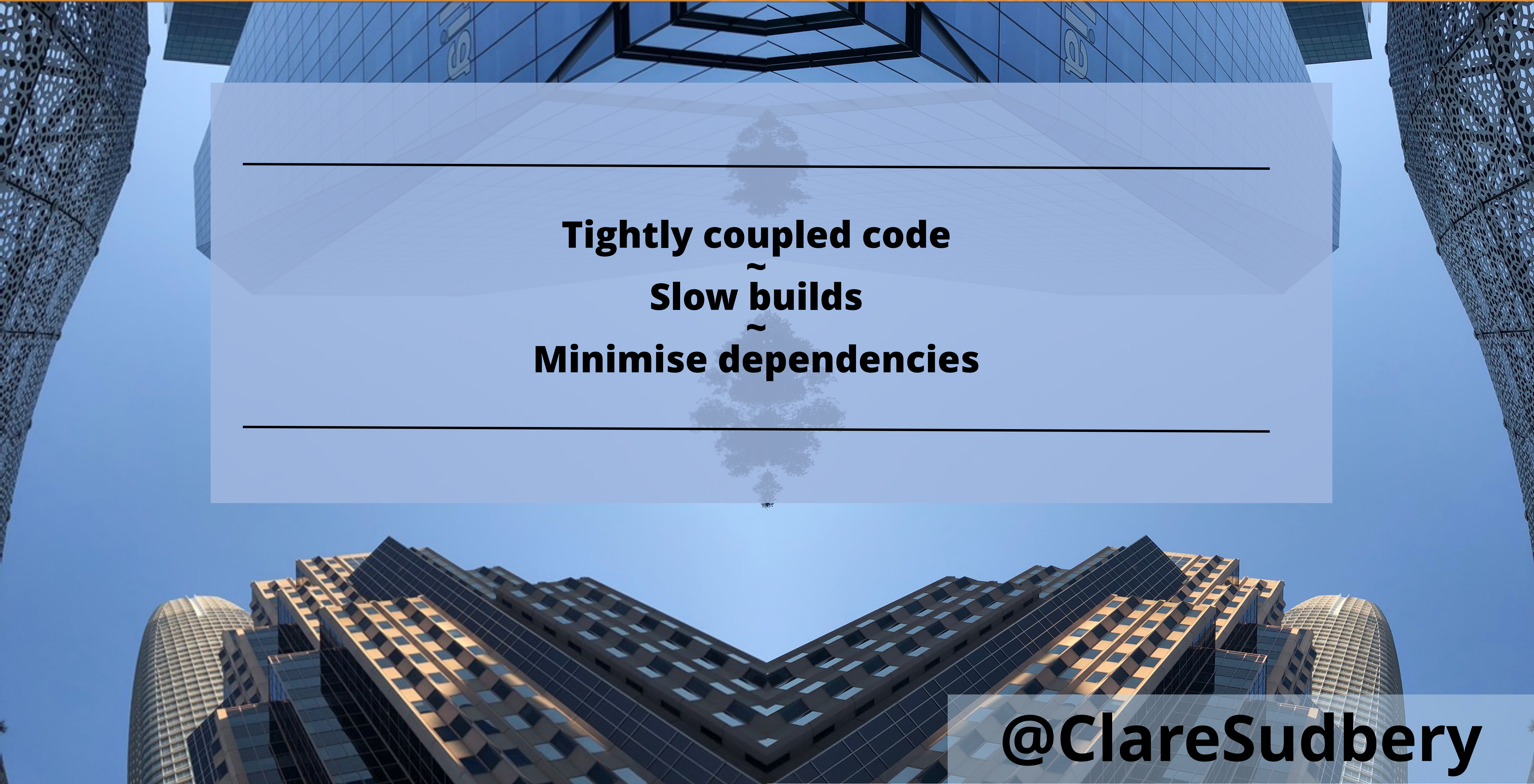
Identify manual tests as a problem

~

TDD, refactoring and automated testing

@ClareSudbery

@ClareSudbery



Tightly coupled code

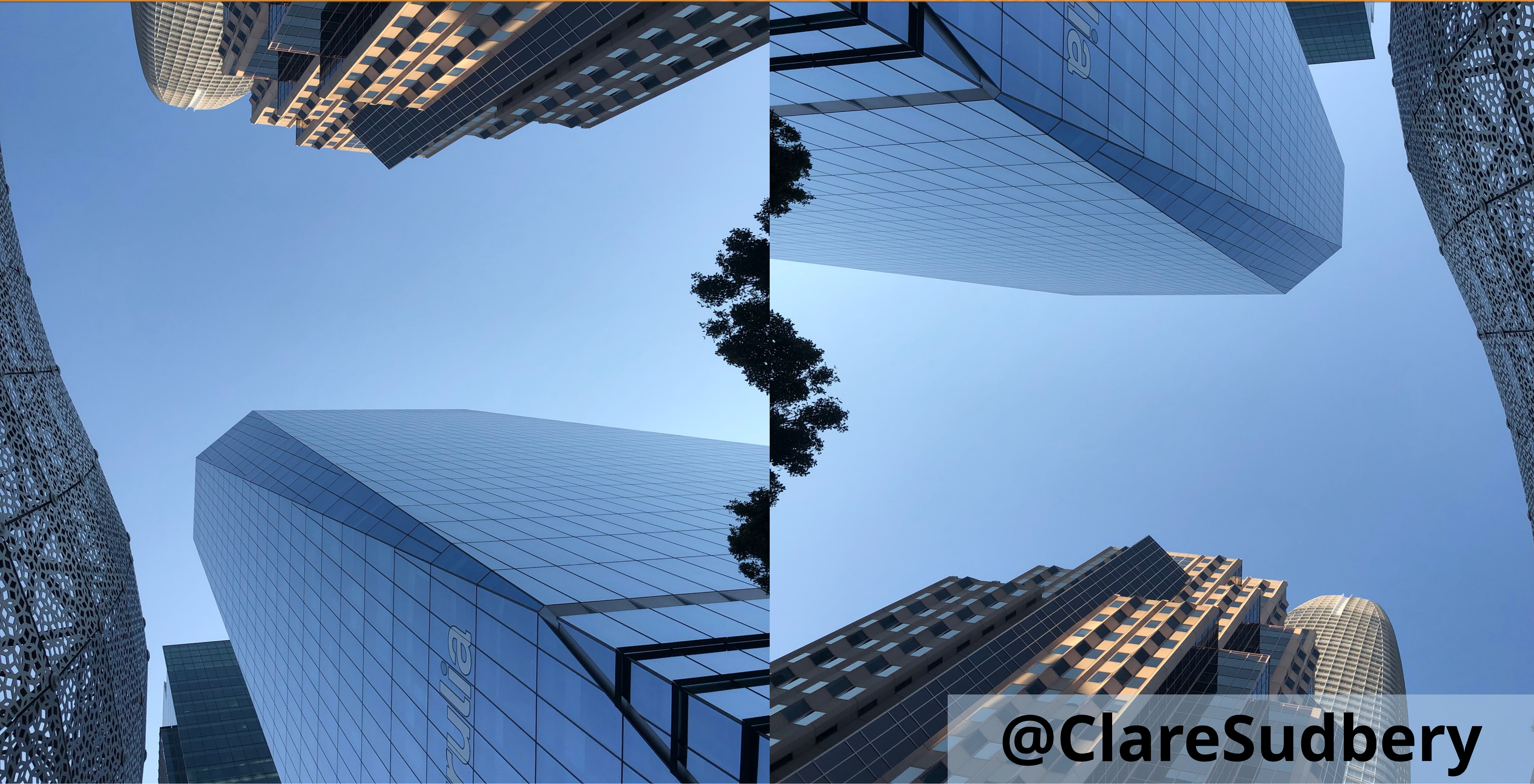
~

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.”

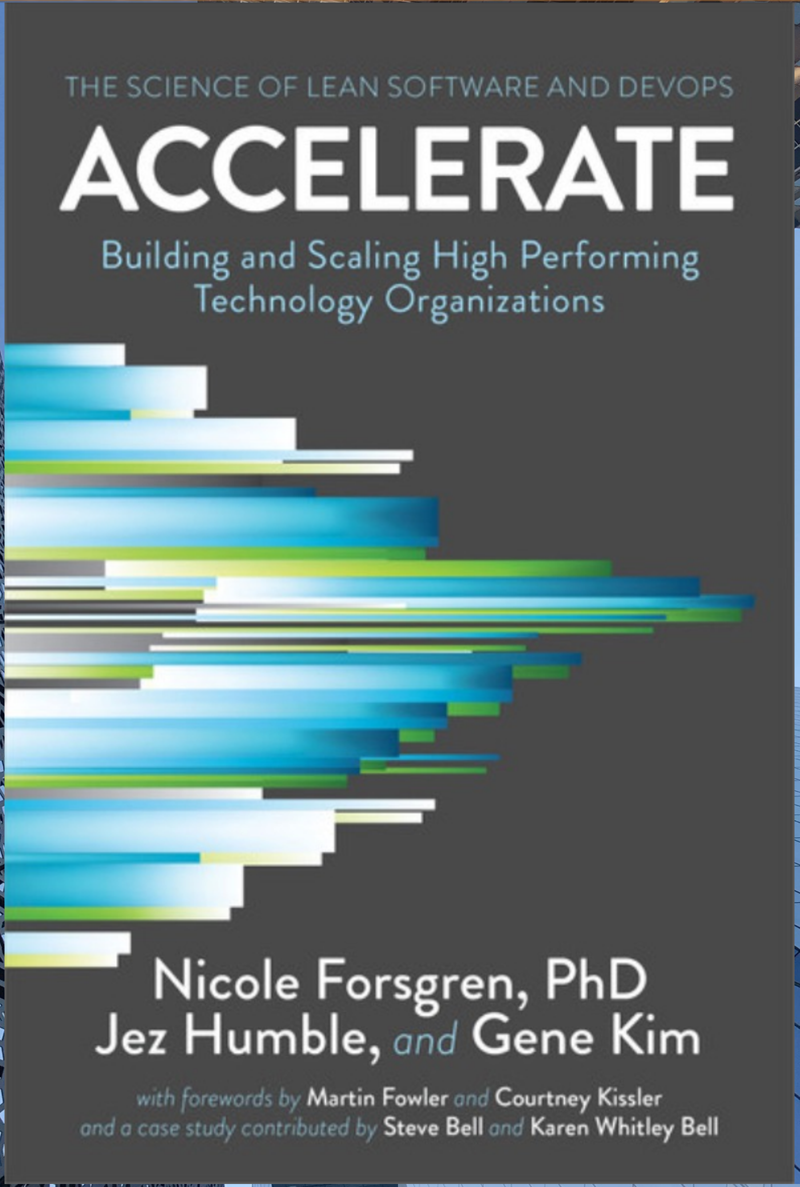
Accelerate – Forsgren et al

@ClareSudbery

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

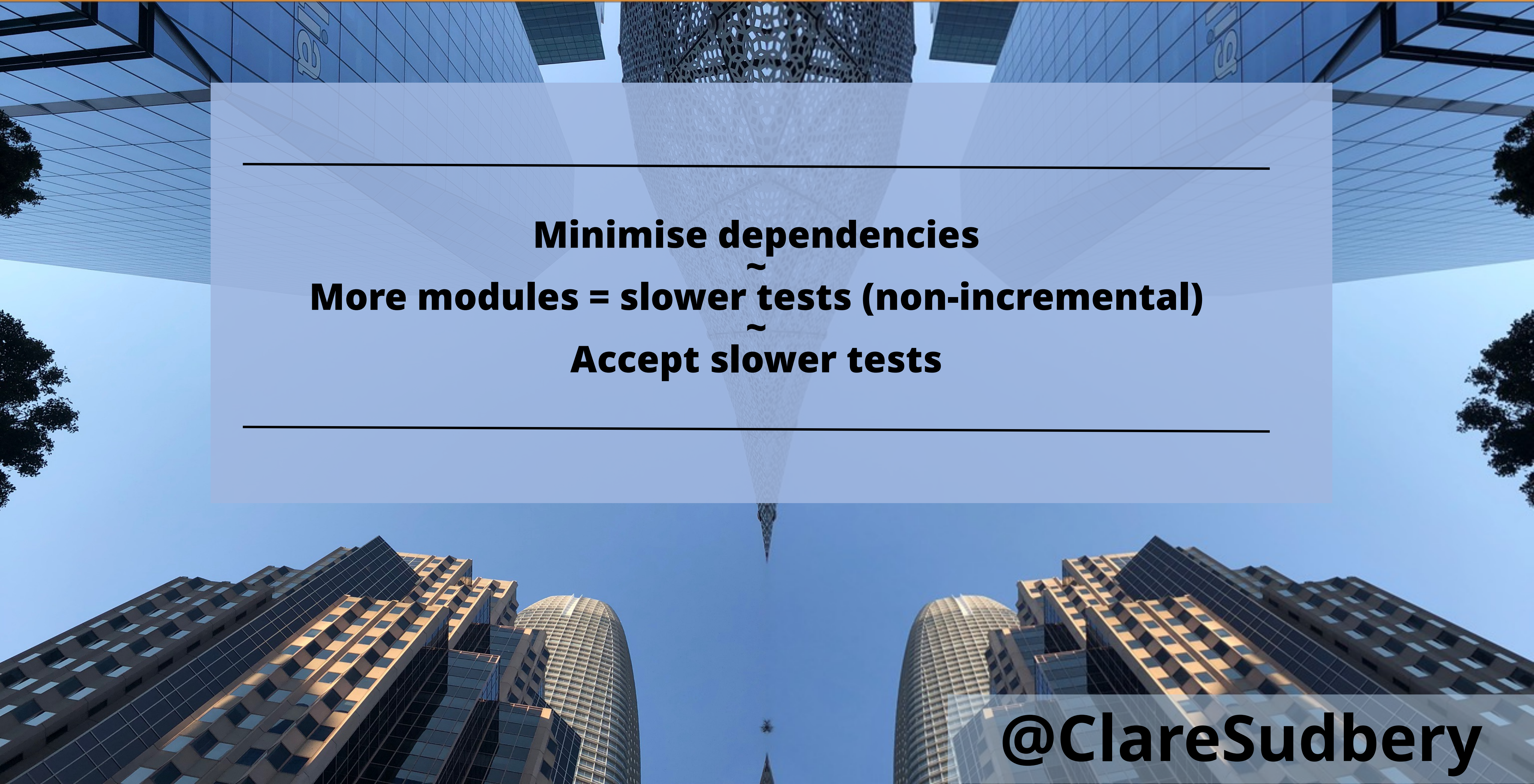
Building and Scaling High Performing Technology Organizations

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



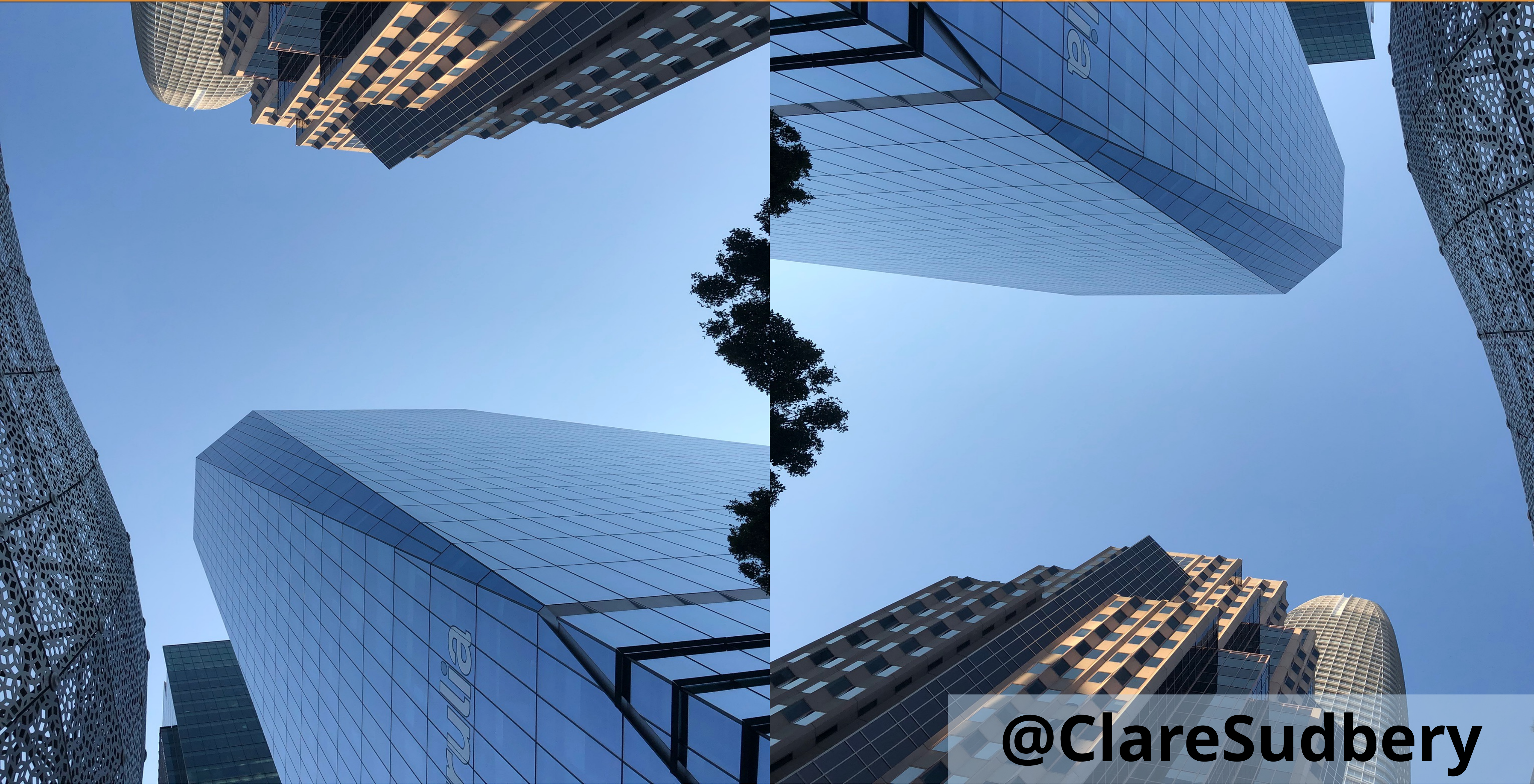
Minimise dependencies

~

More modules = slower tests (non-incremental)

~

Accept slower tests



@ClareSudbery

Slow tests

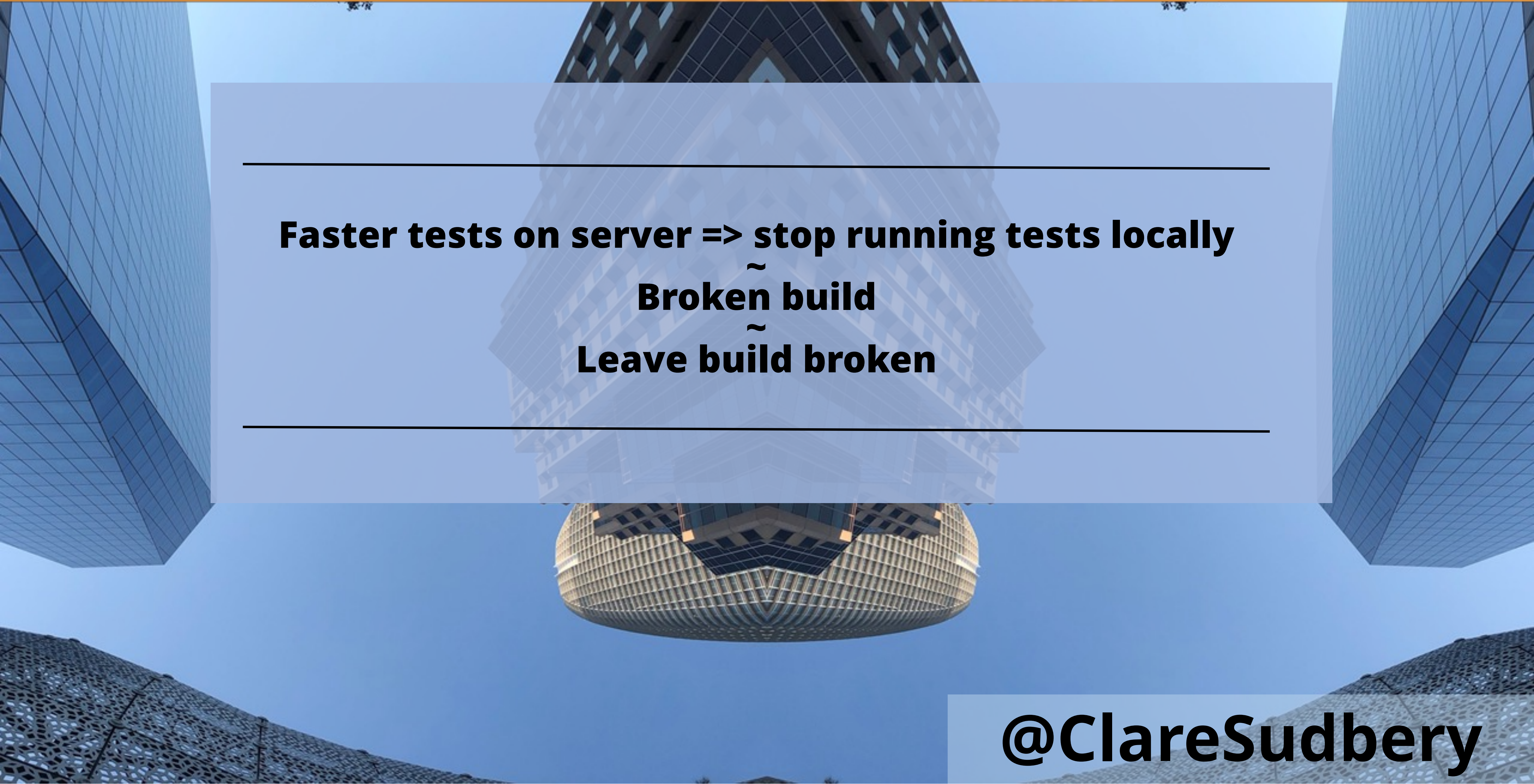
~

People ignore failing tests / don’t run tests

~

Introduce faster, incremental tests

@ClareSudbery



Faster tests on server => stop running tests locally

~

Broken build

~

Leave build broken

@ClareSudbery

Leave build broken

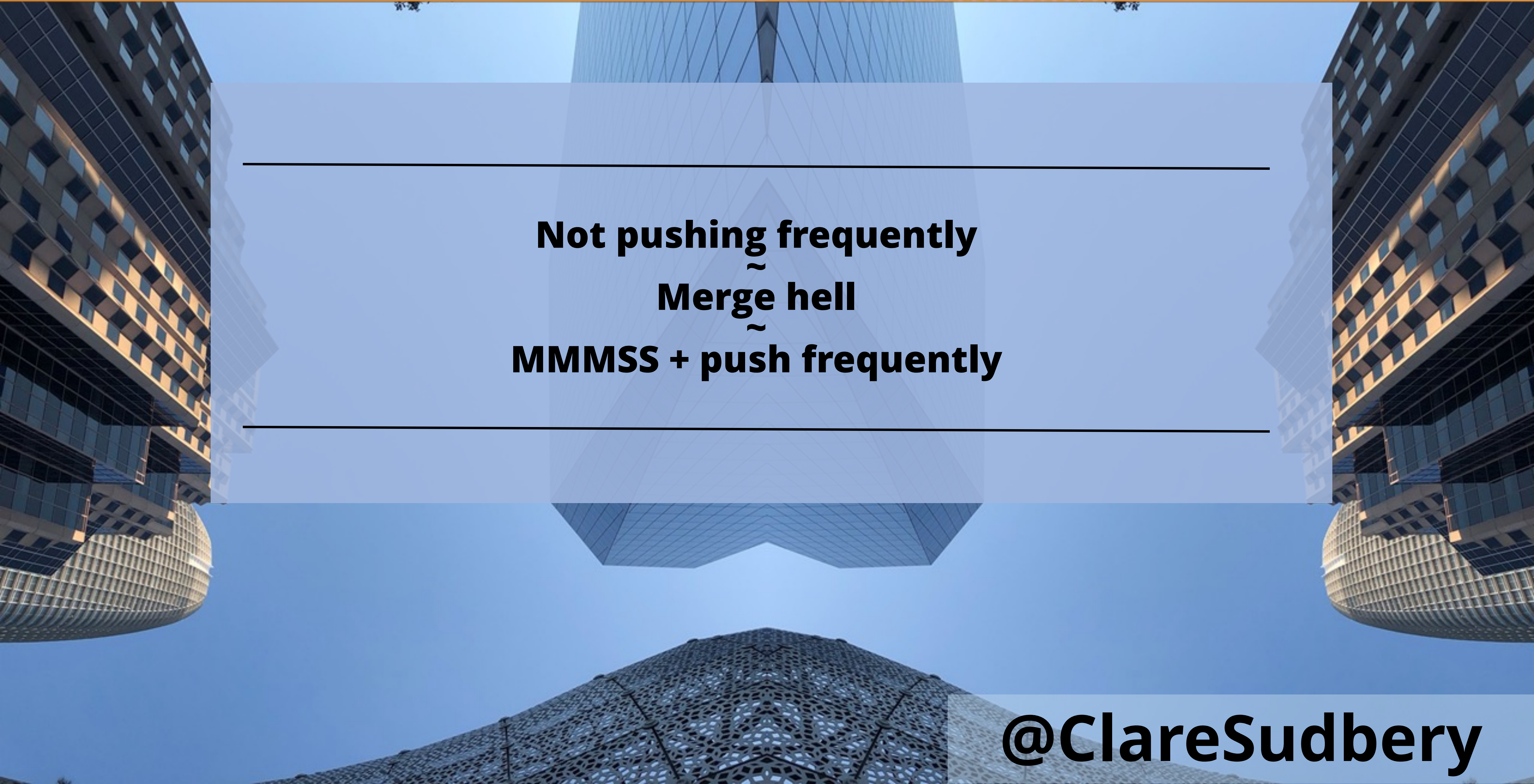
~

Can’t deploy critical patch

~

Run tests locally + keep build green

@ClareSudbery



Not pushing frequently

~

Merge hell

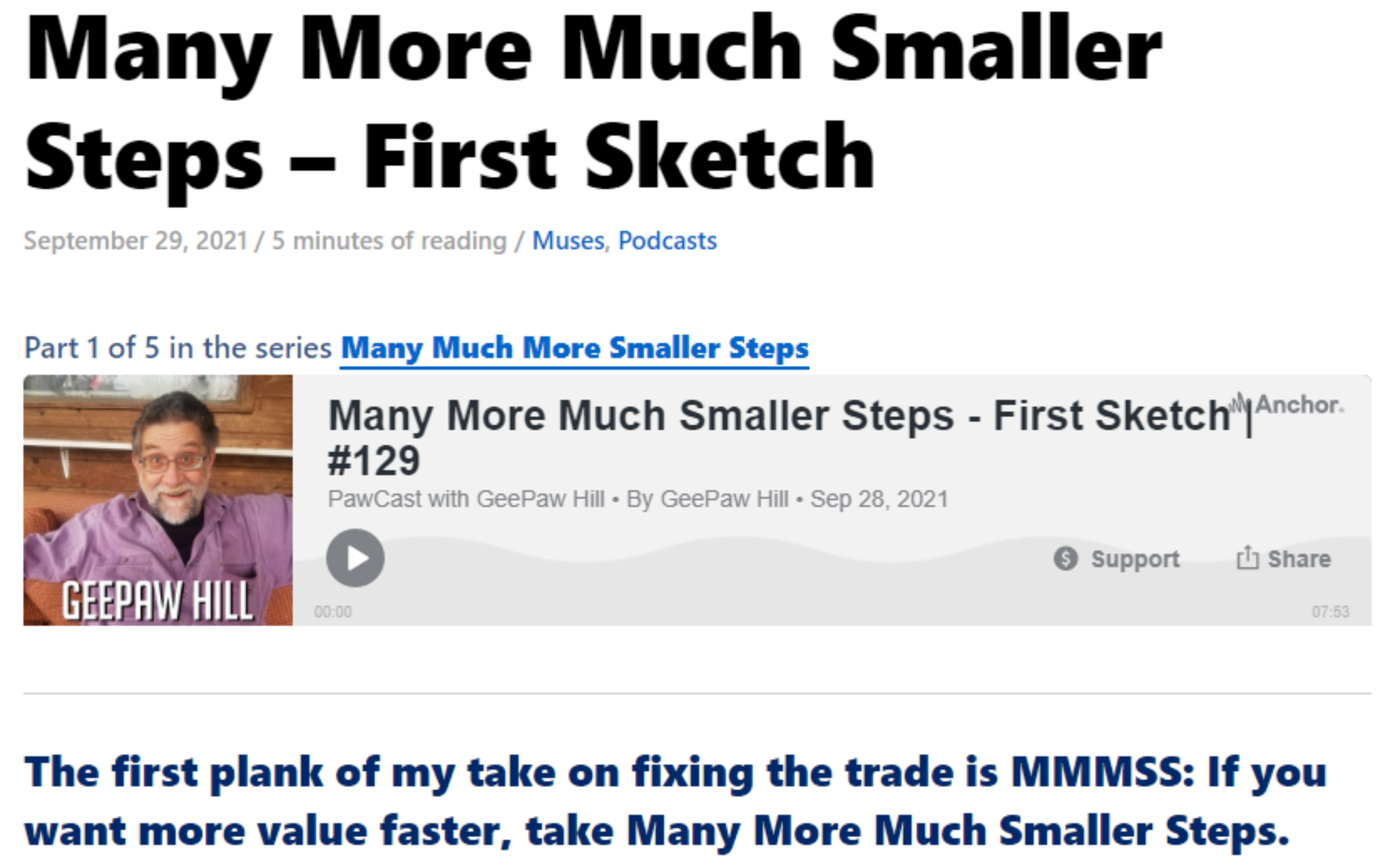
~

MMMSS + push frequently

@ClareSudbery

@ClareSudbery

https://www.geepawhill.org/2021/09/29/many-more-much-smaller-steps-first-sketch/



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 | #129

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

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

@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

CTBD (Continuous trunk-based development)

~

Minor merge issues

~

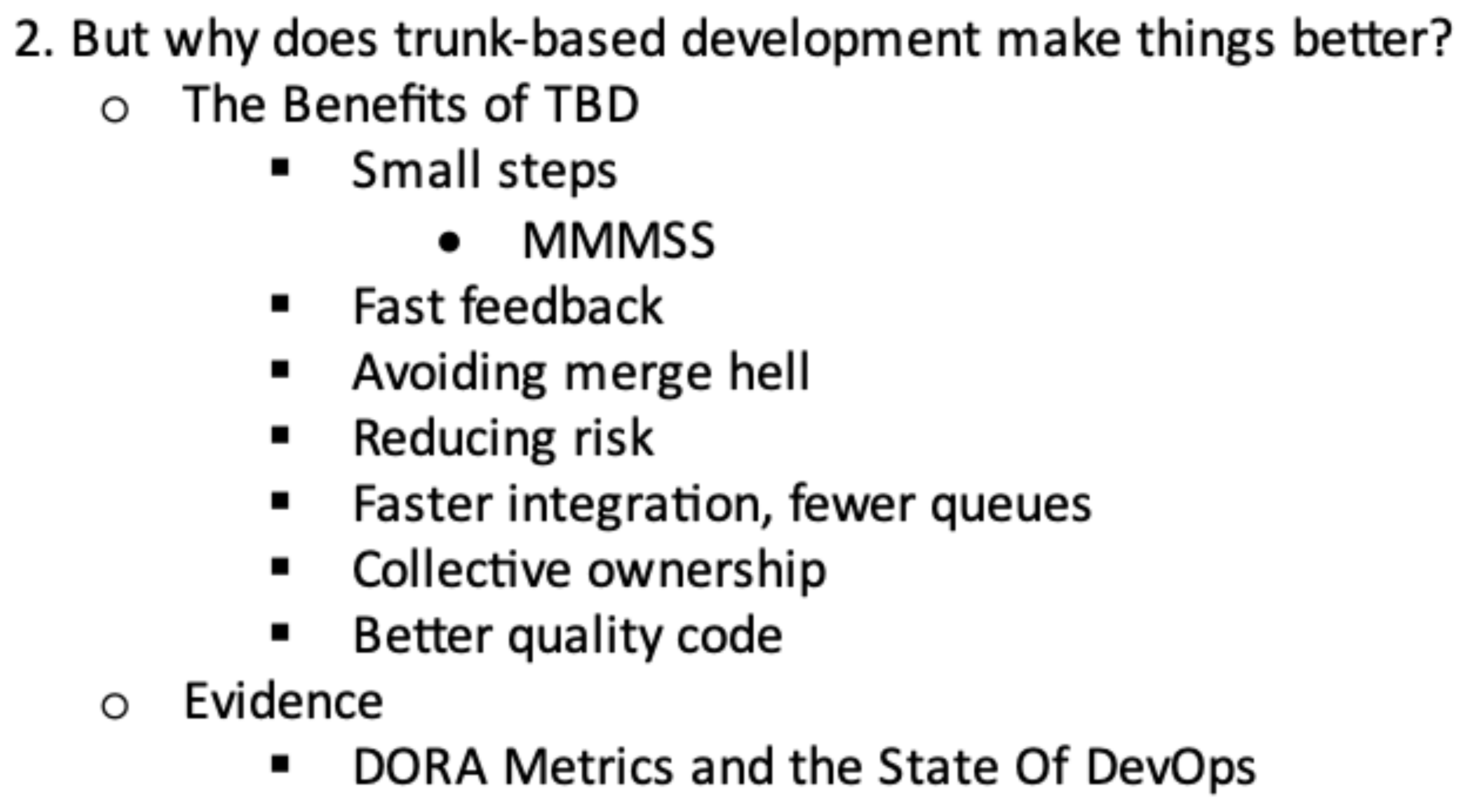
Throw things away

@ClareSudbery

@ClareSudbery

CONCLUSIONS

@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

@ClareSudbery

https://dora.dev/

DORA

DEVOPS RESEARCH & ASSESSMENT

State of DevOps Reports

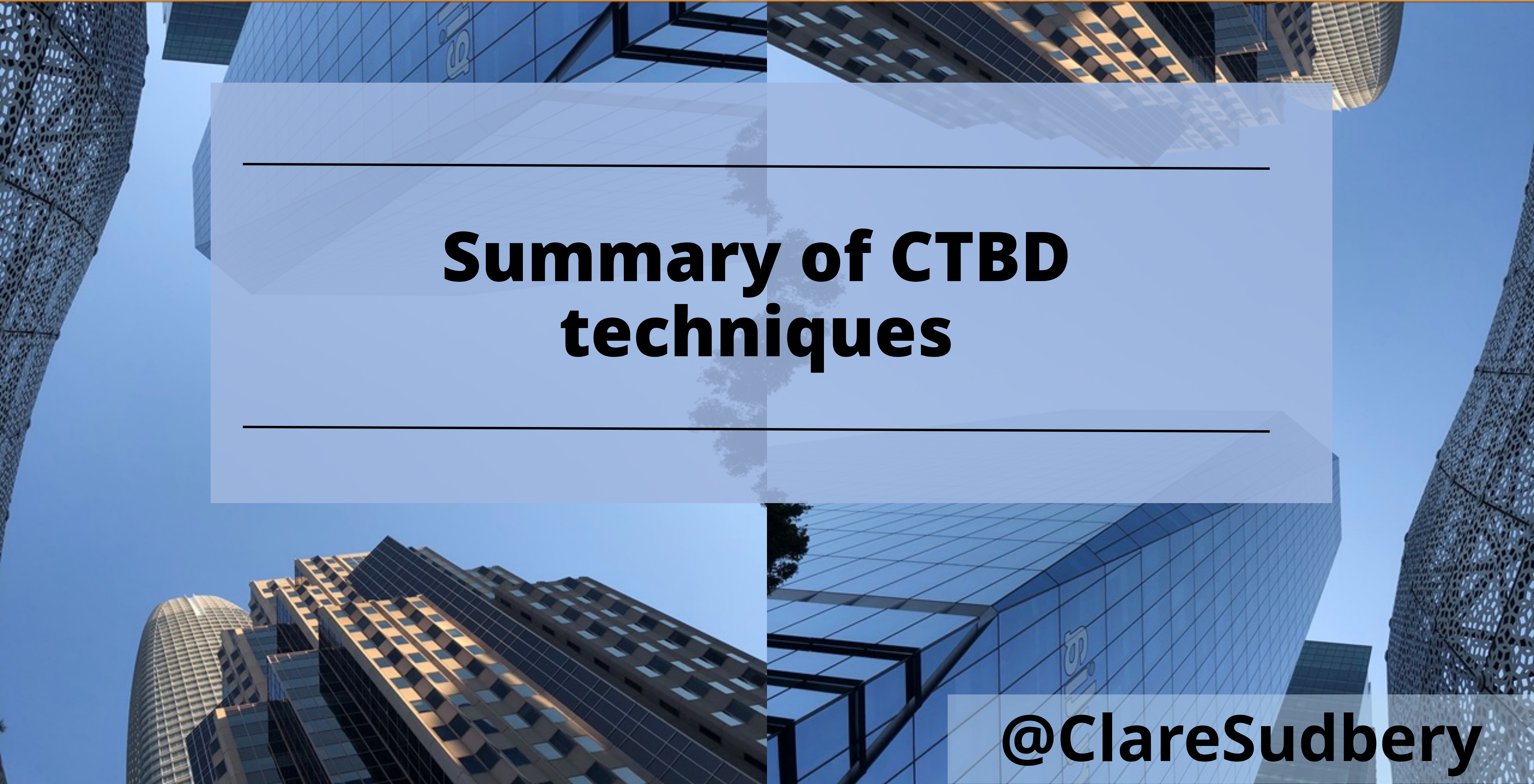
Capability catalog

@ClareSudbery

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

DORA report, 2022

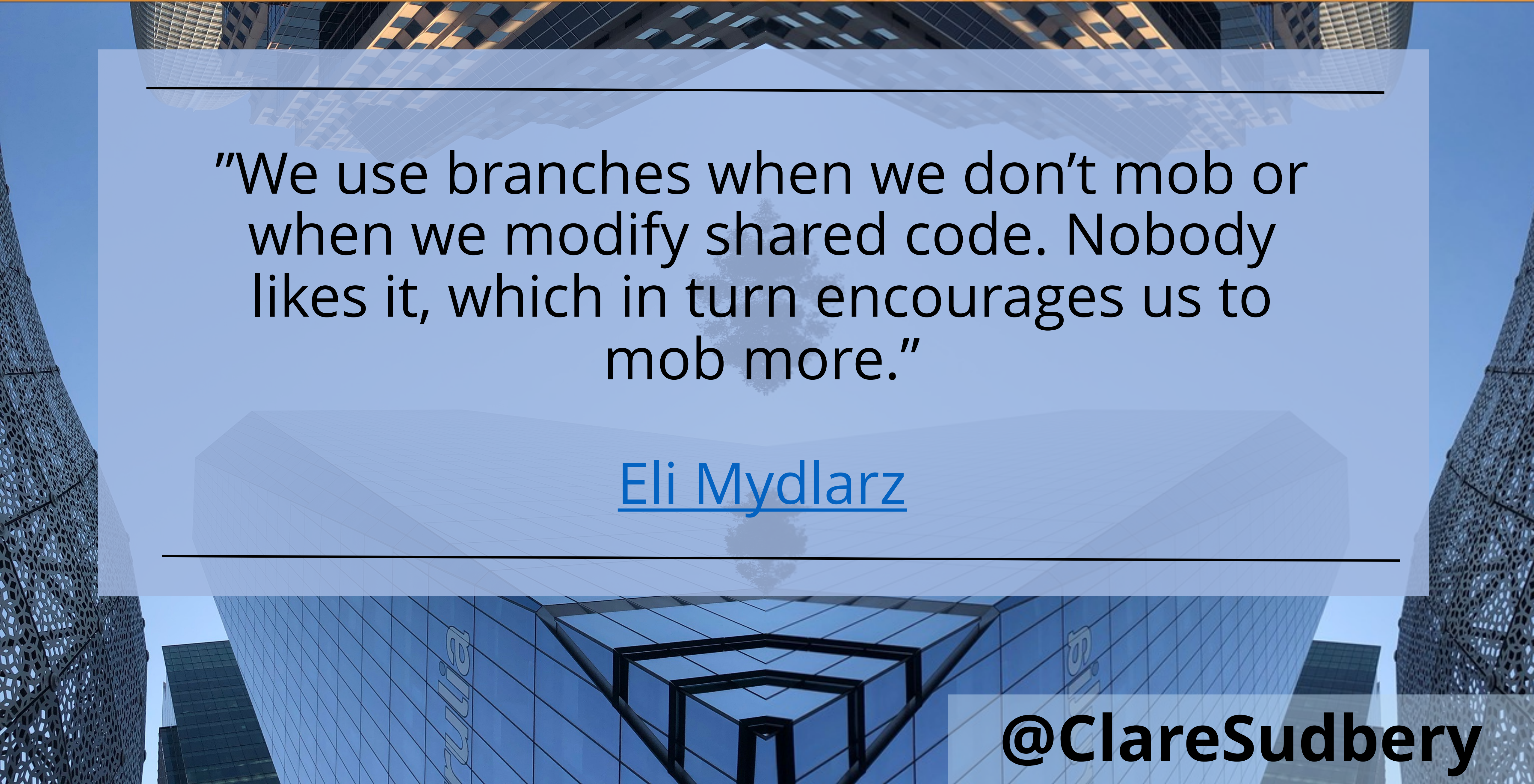
@ClareSudbery



Summary of CTBD techniques

@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

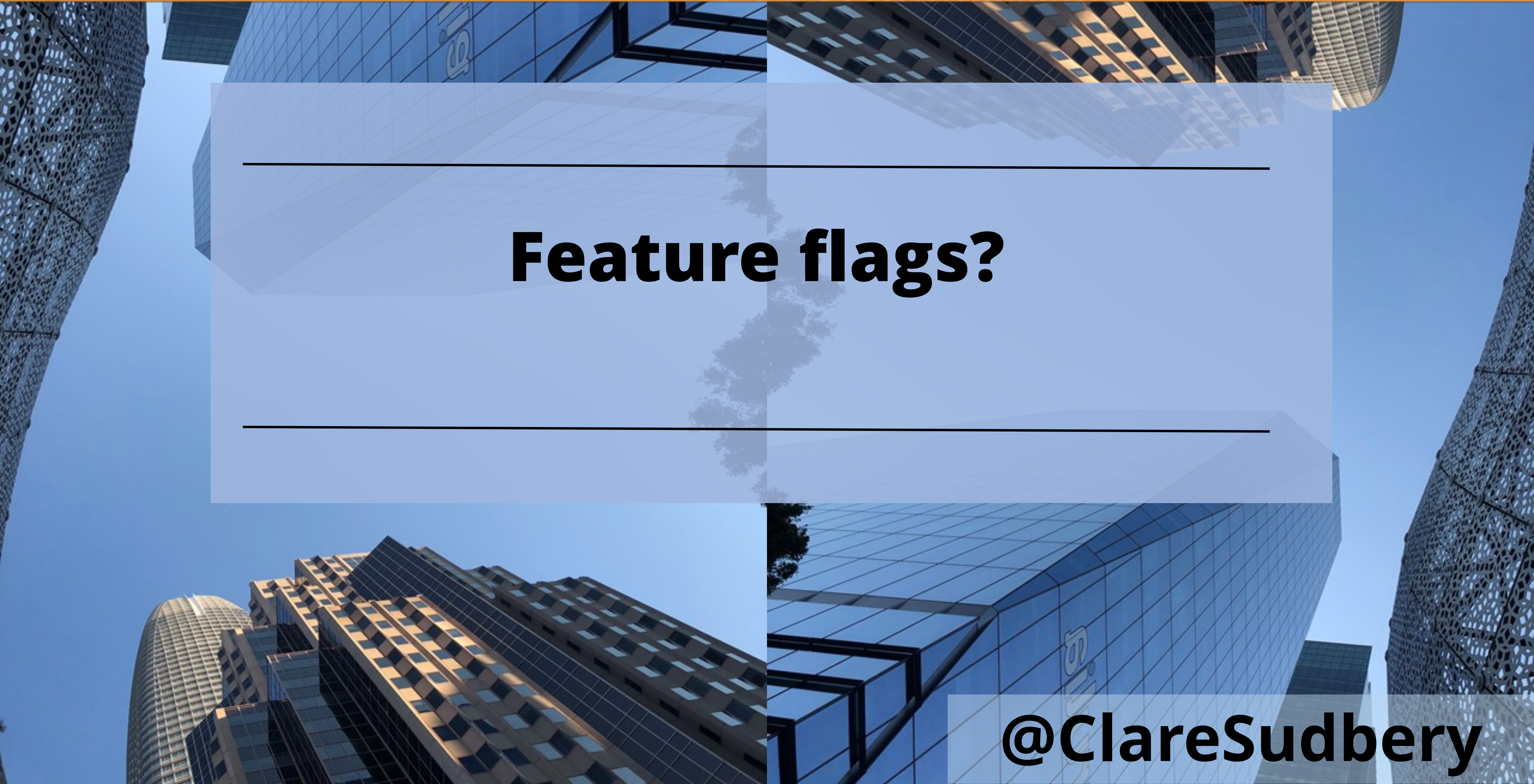
”Any form of branching runs counter, in principle, to the ideas of Continuous Integration.

…

Don’t branch! Don’t branch! Don’t branch!”

Dave Farley

@ClareSudbery



Feature flags?

@ClareSudbery

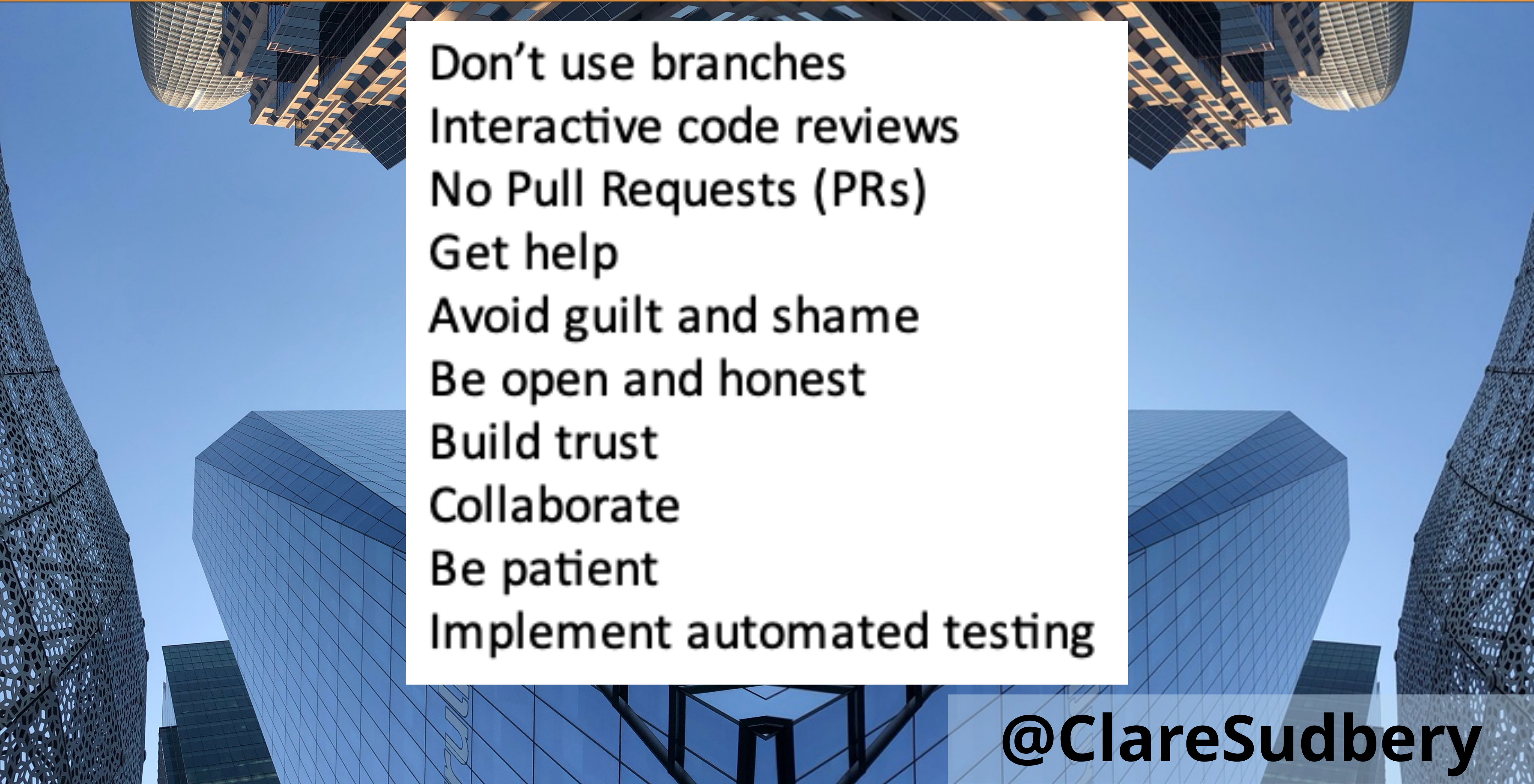
@ClareSudbery

”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



Don’t use branches

Interactive code reviews

No Pull Requests (PRs)

Get help

Avoid guilt and shame

Be open and honest

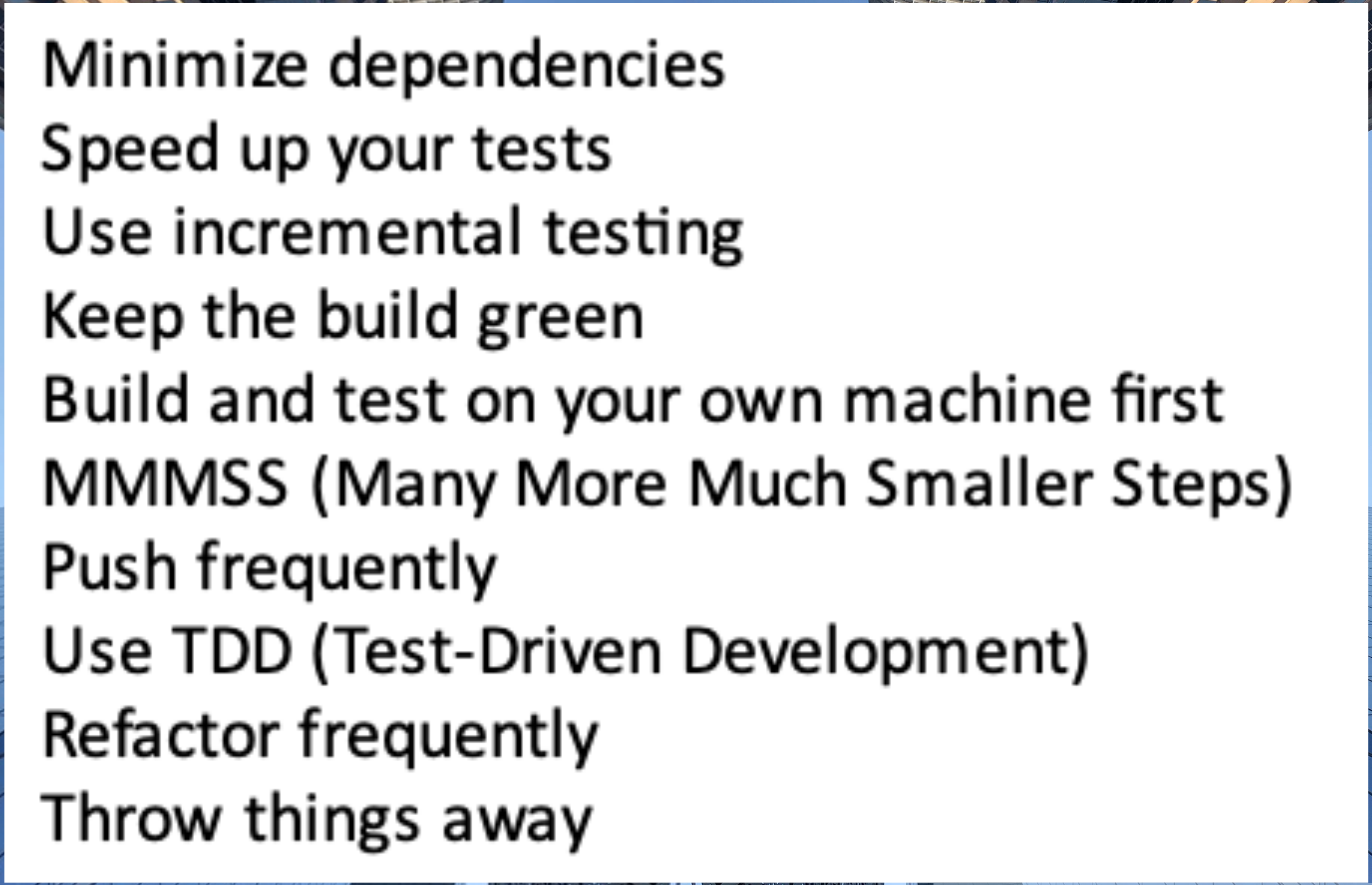
Build trust

Collaborate

Be patient

Implement automated testing

@ClareSudbery



Minimize dependencies

Speed up your tests

Use incremental testing

Keep the build green

Build and test on your own machine first

MMMSS (Many More Much Smaller Steps)

Push frequently

Use TDD (Test-Driven Development)

Refactor frequently

Throw things away

@ClareSudbery

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

@ClareSudbery

@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

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]

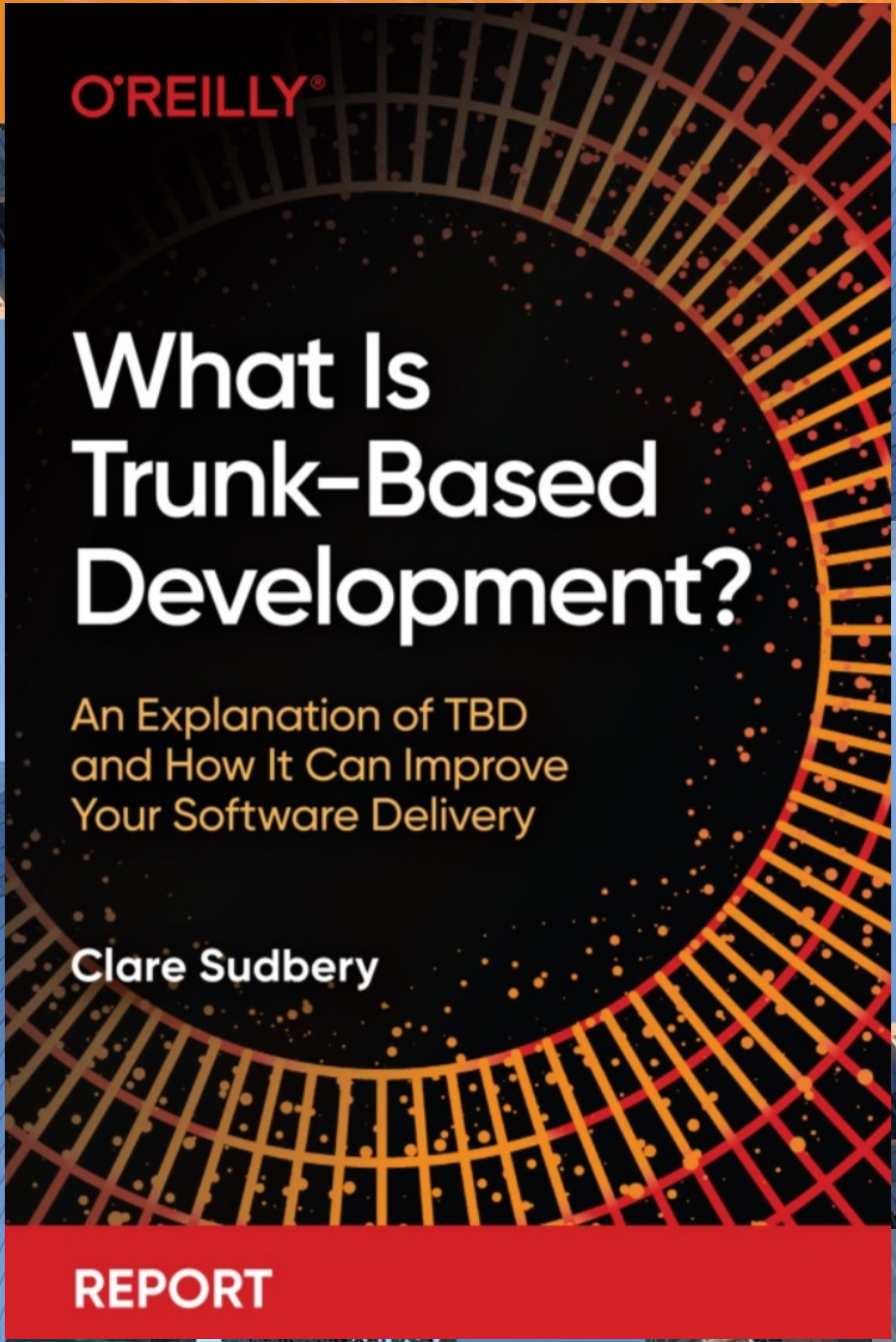
[Dragan Stepanović on Async Code Reviews]

@ClareSudbery

@ClareSudbery

I hosted a podcast…

tinyurl.com/tech-better-pod



O'REILLY

What Is Trunk-Based Development?

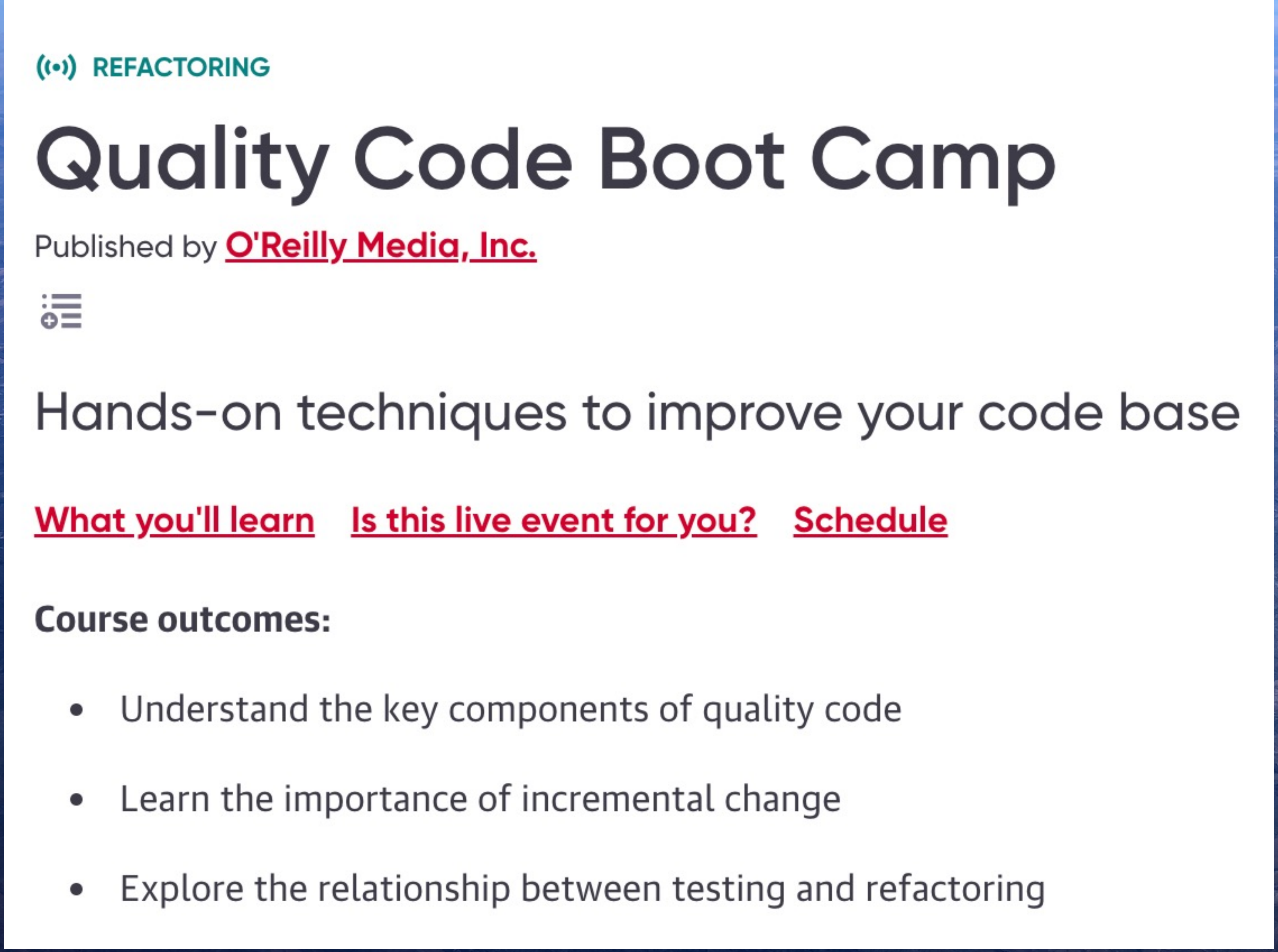
An Explanation of TBD and How It Can Improve Your Software Delivery

Clare Sudbery

REPORT

tinyurl.com/TBD-clare

@ClareSudbery



((•)) REFACTORING

Quality Code Boot Camp

Published by O'Reilly Media, Inc.

Hands-on techniques to improve your code base

What you'll learn   Is this live event for you?   Schedule

Course outcomes:

- Understand the key components of quality code
- Learn the importance of incremental change
- Explore the relationship between testing and refactoring

Every Wednesday in July – Starts Wed July 5th 2023

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

Twitter: @ClareSudbery

LinkedIn: tinyurl.com/lin-csudbery

Podcast: tinyurl.com/tech-better-pod

@ClareSudbery

Questions / discussion

@ClareSudbery

goto;

GOTO **Guide**

★★★★★

Remember to
rate this session

THANK YOU!

#GOTOams