Panoramic Quality

Brendan Connolly

@theBConnolly || http://brendanconnolly.net

Sr. Quality Engineer, Procore Technologies, Santa Barbara CA

1

Panoramic Quality

Brendan Connolly

@theBConnolly || http://brendanconnolly.net

Sr. Quality Engineer, Procore Technologies, Santa Barbara CA

@theBConnolly

2

What’s So Different?

@theBConnolly

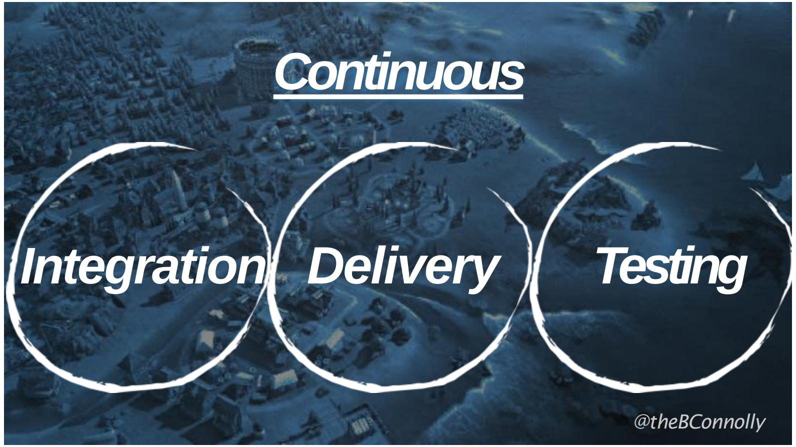
3

SPEED

- Orders of Magnitude
- Risk / Release Inversion

@theBConnolly

4



Continuous

Integration

Delivery

Testing

@theBConnolly

5

OWNERSHIP

Developers Devour All

@theBConnolly

6

Our Landscape Is Changing

@theBConnolly

7

Core Responsibilities

Know

Protect

Verify

@theBConnolly

8

Regions of Responsibility

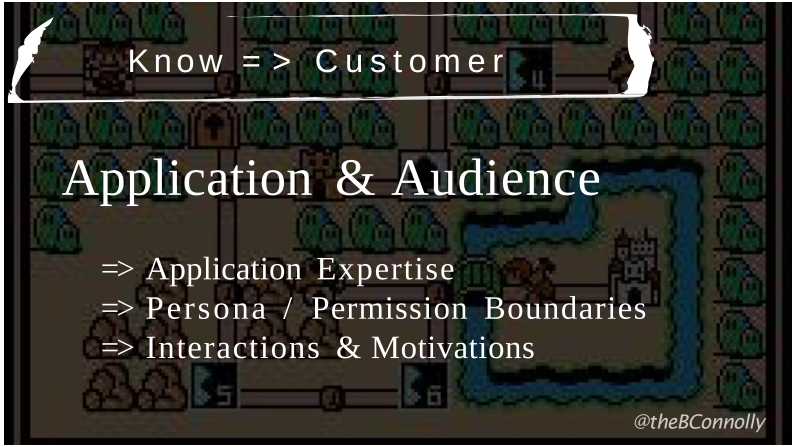
@theBConnolly

9

Customer

@theBConnolly

10



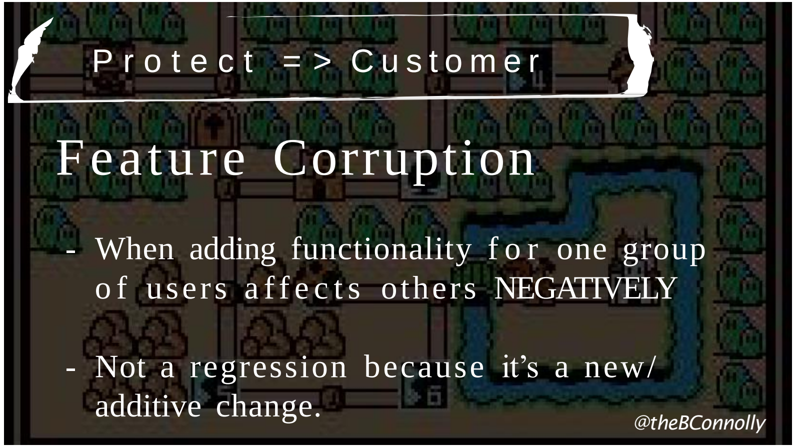
Know => Customer

Application & Audience

=> Application Expertise
=> Persona / Permission Boundaries
=> Interactions & Motivations

@theBConnolly

11



Protect => Customer

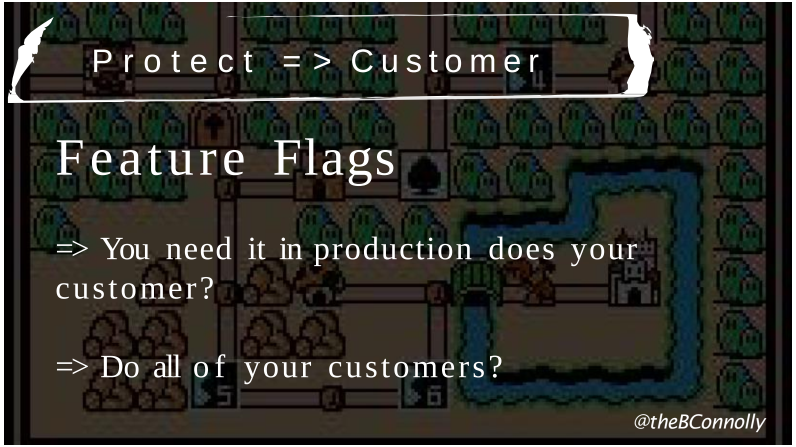
Feature Corruption

- When adding functionality for one group of users affects others NEGATIVELY

- Not a regression because it's a new/ additive change.

@theBConnolly

12



Protect => Customer

Feature Flags

=> You need it in production does your customer?

=> Do all of your customers?

@theBConnolly

13

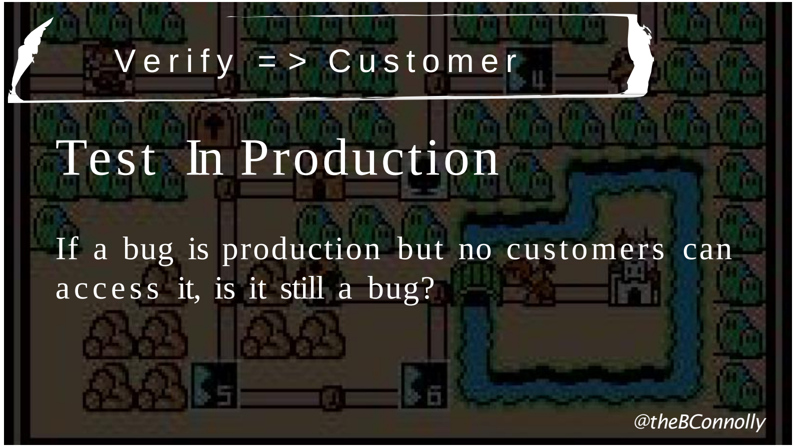
Verify => Customer

Change Tolerance

=> Instability is indistinguishable from rapid change

@theBConnolly

14



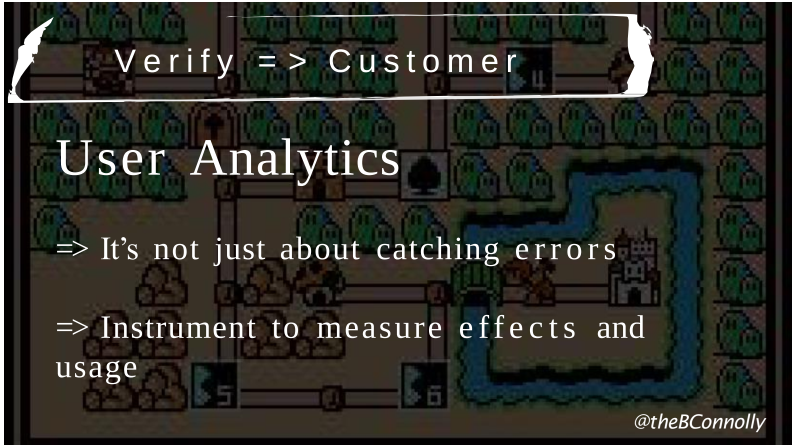
Verify => Customer

Test In Production

If a bug is production but no customers can access it, is it still a bug?

@theBConnolly

15



Verify => Customer

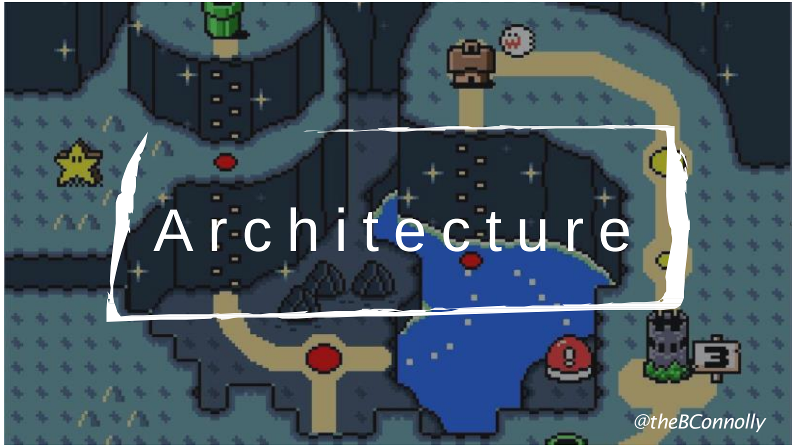
User Analytics

=> It's not just about catching errors

=> Instrument to measure effects and usage

@theBConnolly

16



Architecture

@theBCConnolly

17

Know => Architecture

The Geography of Software

=> How it’s built / deployed
=> Where it’s running
=> How it solves your customers problems

@theBCConnolly

18

Know => Architecture

Environmental Impact

=> Local, Staging, Production
=> What Works Where?
=> What Doesn’t?

@theBConnolly

19

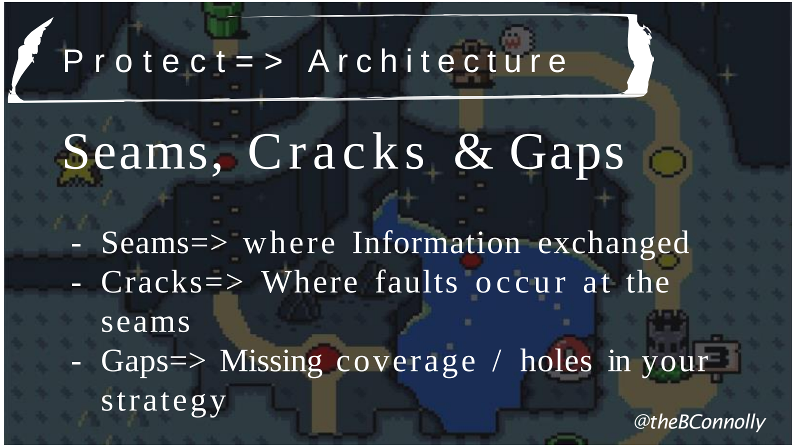
Know => Architecture

Boundaries of Risk

=> Boundaries of components can be borders between developers

@theBConnolly

20



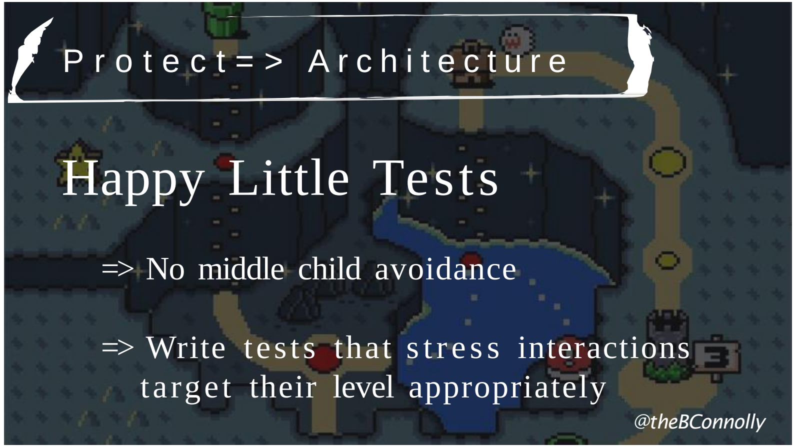
Protect=> Architecture

Seams, Cracks & Gaps

- Seams=> where Information exchanged
- Cracks=> Where faults occur at the seams
- Gaps=> Missing coverage / holes in your strategy

@theBConnolly

21



Protect=> Architecture

Happy Little Tests

=> No middle child avoidance

=> Write tests that stress interactions target their level appropriately

@theBConnolly

22

Verify => Architecture

Minimum Viable Environment

=> Work as close to home as you can
=> Take the Developers local changes
=> Identify tests

@theBConnolly

23

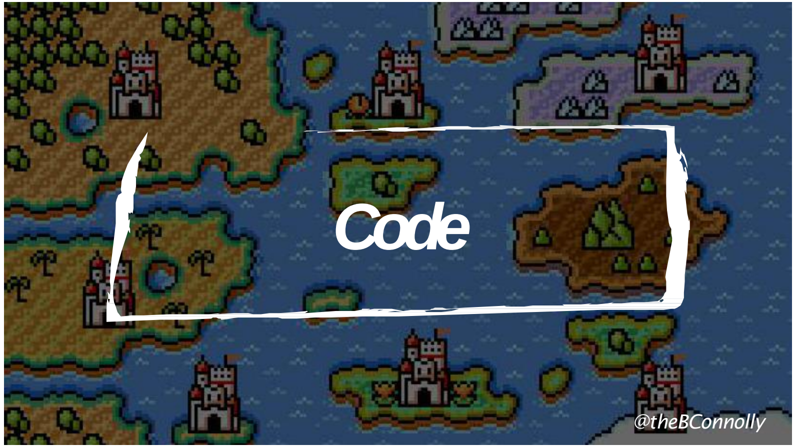
Verify => Architecture

Health & Wellness

=> Monitoring
=> Surfacing information

@theBConnolly

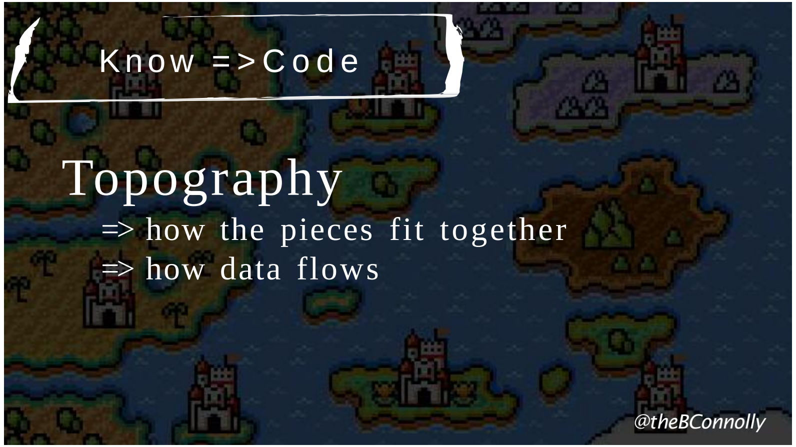
24



Code

@theBConnolly

25



Know =>Code

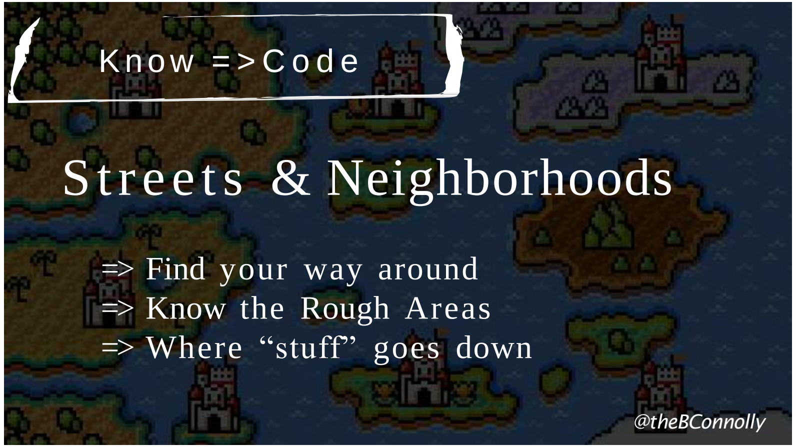
Topography

=> how the pieces fit together

=> how data flows

@theBConnolly

26



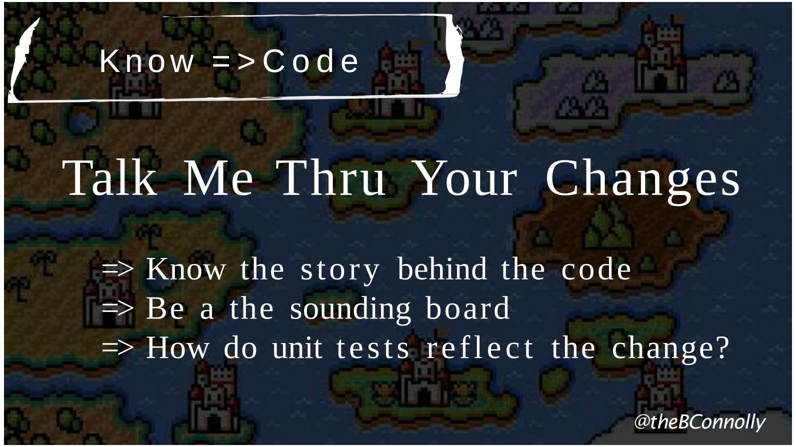
Know =>Code

Streets & Neighborhoods

=> Find your way around
=> Know the Rough Areas
=> Where “stuff” goes down

@theBConnolly

27



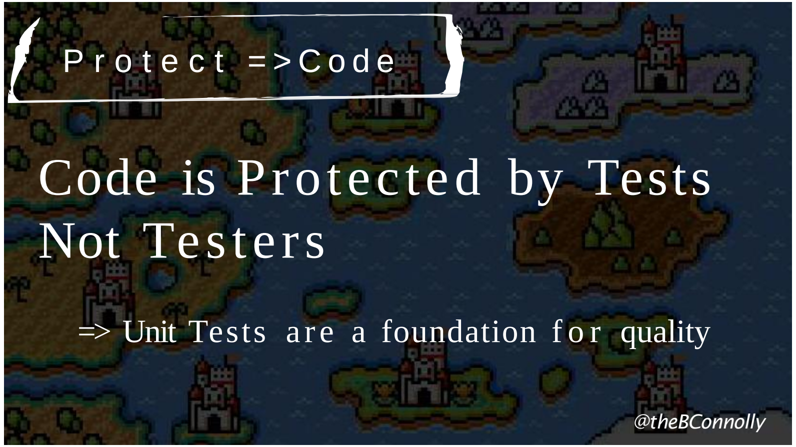
Know =>Code

Talk Me Thru Your Changes

=> Know the story behind the code
=> Be a the sounding board
=> How do unit tests reflect the change?

@theBConnolly

28



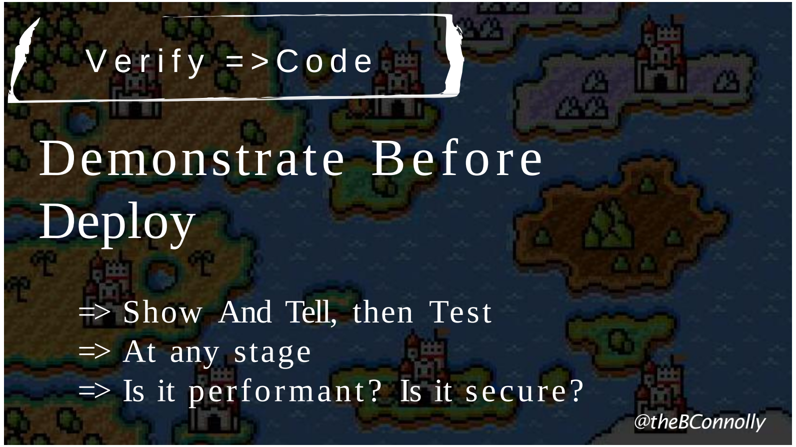
Protect =>Code

Code is Protected by Tests Not Testers

=> Unit Tests are a foundation for quality

@theBConnolly

29



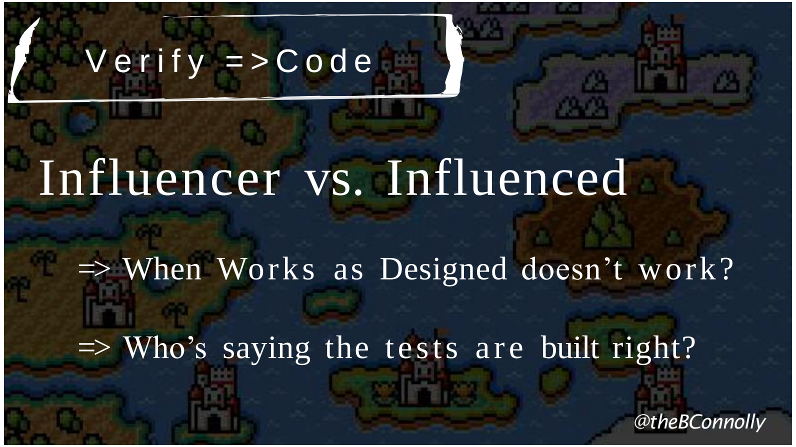
Verify =>Code

Demonstrate Before Deploy

=> Show And Tell, then Test
=> At any stage
=> Is it performant? Is it secure?

@theBConnolly

30



Verify =>Code

Influencer vs. Influenced

=> When Works as Designed doesn't work?

=> Who's saying the tests are built right?

@theBConnolly

31

PAUSE

@theBConnolly

32

Embracing Ownership

@theBConnolly

33

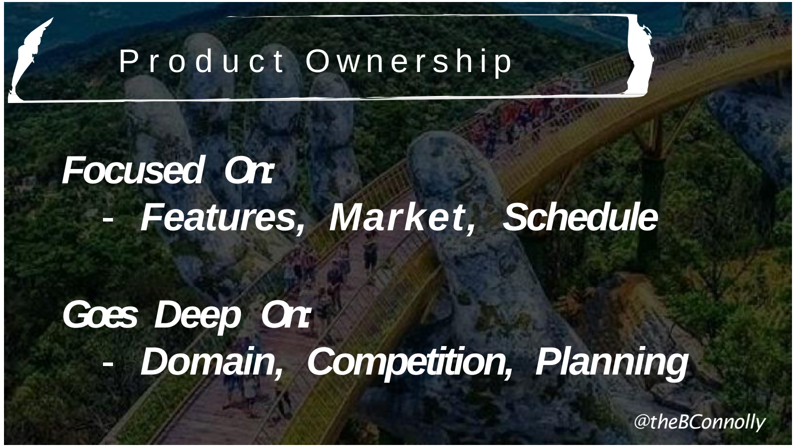
Product Ownership

- Define the product backlog
- Groom and prioritize the backlog
- Ensure completed user stories fulfills the acceptance criteria

https://medium.com/@melissaperri

@theBConnolly

34



Product Ownership

Focused On:

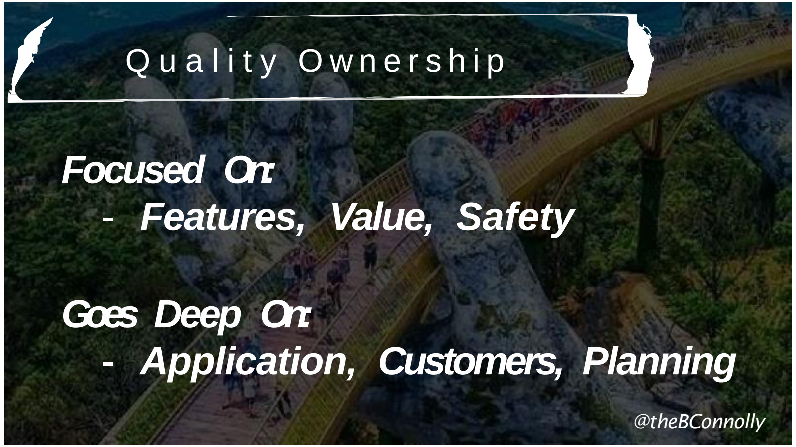
- *Features, Market, Schedule*

Goes Deep On:

- *Domain, Competition, Planning*

@theBConnolly

35



Quality Ownership

Focused On:

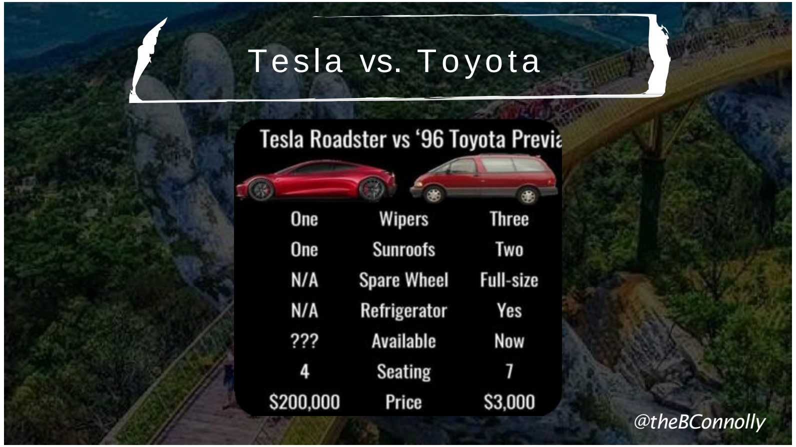
- *Features, Value, Safety*

Goes Deep On:

- *Application, Customers, Planning*

@theBConnolly

36



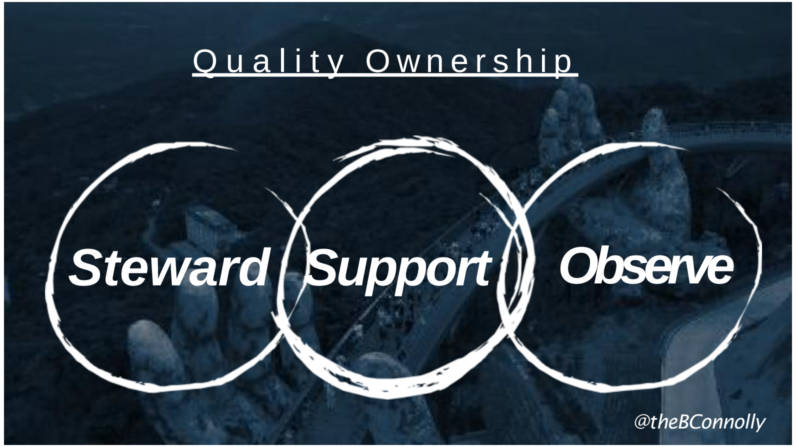
Tesla vs. Toyota

Tesla Roadster vs '96 Toyota Previa

One	Wipers	Three
One	Sunroofs	Two
N/A	Spare Wheel	Full-size
N/A	Refrigerator	Yes
???	Available	Now
4	Seating	7
$200,000	Price	$3,000

@theBConnolly

37

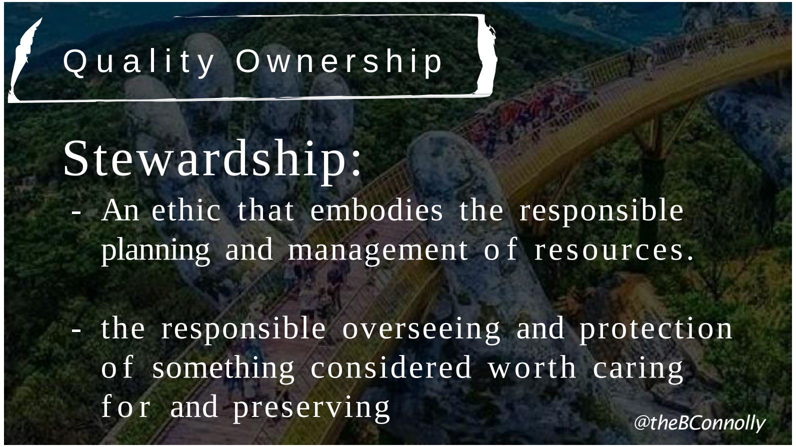


Quality Ownership

Steward Support Observe

@theBConnolly

38



Quality Ownership

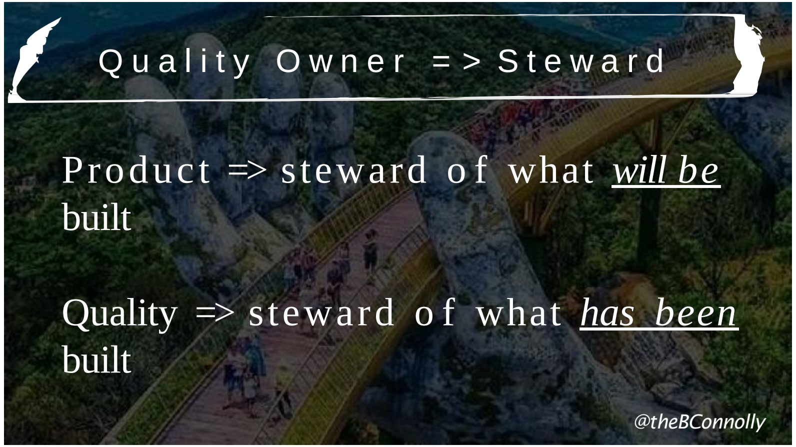
Stewardship:

- An ethic that embodies the responsible planning and management of resources.

- the responsible overseeing and protection of something considered worth caring for and preserving

@theBConnolly

39



Quality Owner => Steward

Product => steward of what *will be* built

Quality => steward of what *has been* built

@theBConnolly

40

Quality Owner => Stewardship

Quality is in Your "Charge"

Responsibility for controlling or caring for something

A formal statement accusing someone of committing a crime

@theBConnolly

41

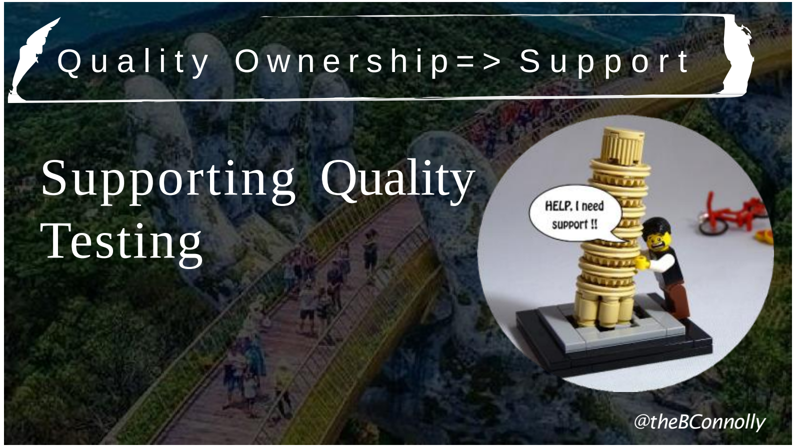
Quality Ownership=> Support

Open Source your Mindset

=> Be an *#ALLY* for testing
=> Enable & Mentor Quality
=> Infuse Quality into automation

@theBConnolly

42



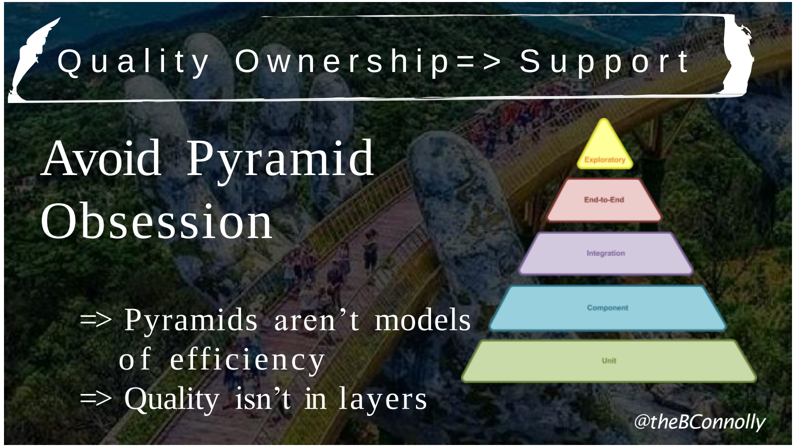
Quality Ownership=> Support

Supporting Quality Testing

HELP, I need support !!

@theBConnolly

43



Quality Ownership=> Support

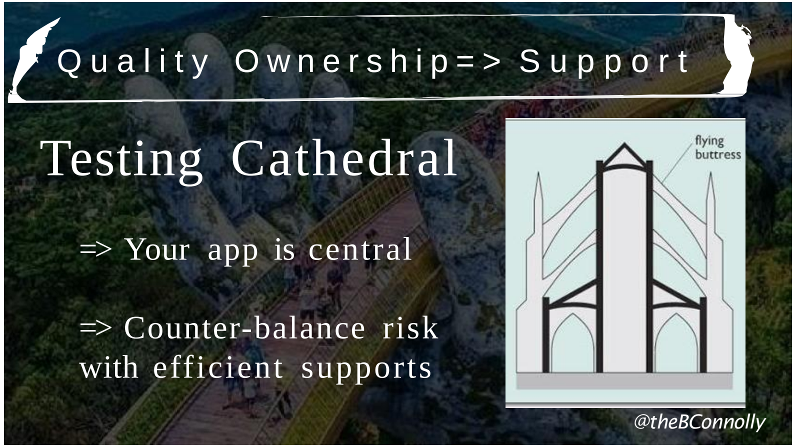
Avoid Pyramid Obsession

=> Pyramids aren’t models of efficiency
=> Quality isn’t in layers

Exploratory
End-to-End
Integration
Component
Unit

@theBConnolly

44



Quality Ownership=> Support

Testing Cathedral

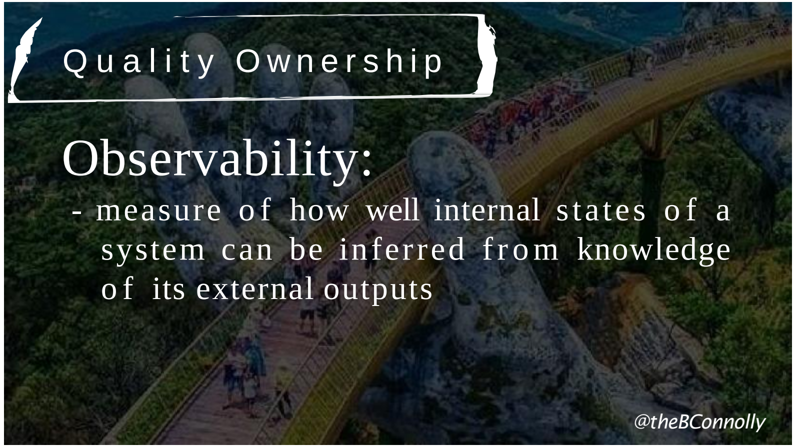
=> Your app is central

=> Counter-balance risk with efficient supports

flying buttress

@theBConnolly

45



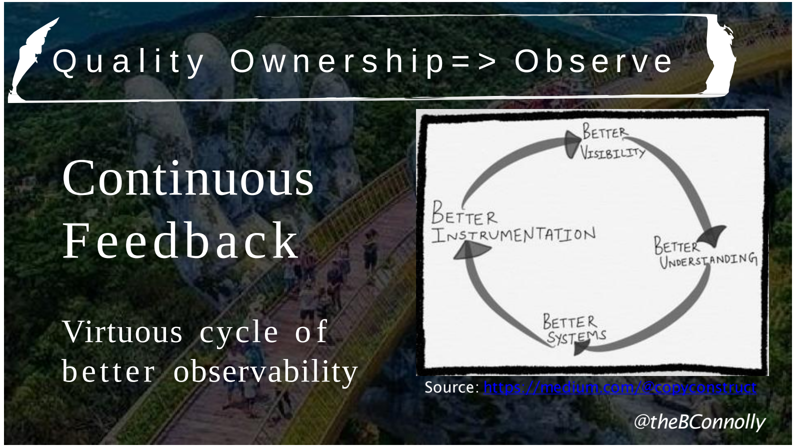
Quality Ownership

Observability:

- measure of how well internal states of a system can be inferred from knowledge of its external outputs

@theBConnolly

46



Quality Ownership=> Observe

Continuous Feedback

Virtuous cycle of better observability

Source: https://medium.com/@copyconstruct

@theBConnolly

47

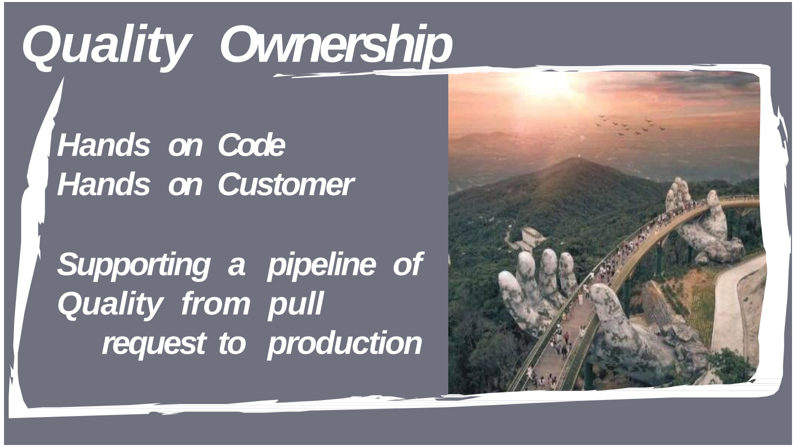
Quality Ownership=> Observe

Continuous Collaboration

=> Build skills to break down barriers to collaboration

@theBConnolly

48



Quality Ownership

Hands on Code
Hands on Customer

Supporting a pipeline of Quality from pull request to production

49

Be Well, Connect, Share

Connect With Me
Twitter / LinkedIn : @theBConnolly
Blog: http://brendanconnolly.net

@theBConnolly

50