

Test Automation in DevOps

MOVING TOWARDS CONTINUOUS TESTING

About Me...

- CTO of Utopia Solutions – QA/testing focused service provider
- Involved in software quality and testing for over 25 years
- Most of that time focused on test automation
- Passionate about learning and helping organizations benefit from doing automation the right way

Email: lee.barnes@utopiasolutions.com

Twitter: @USI_LeeBarnes

LinkedIn: [linkedin.com/in/leebarnes](https://www.linkedin.com/in/leebarnes)

Blog: utopiasolutions.com/blog1

Agenda

- Intro
 - What is DevOps
 - Testing in DevOps
 - Continuous Testing
 - Test Automation in DevOps
- Effective Test Automation
 - Scope and Approach
 - Test Environment and Data Management
- Testing in Production

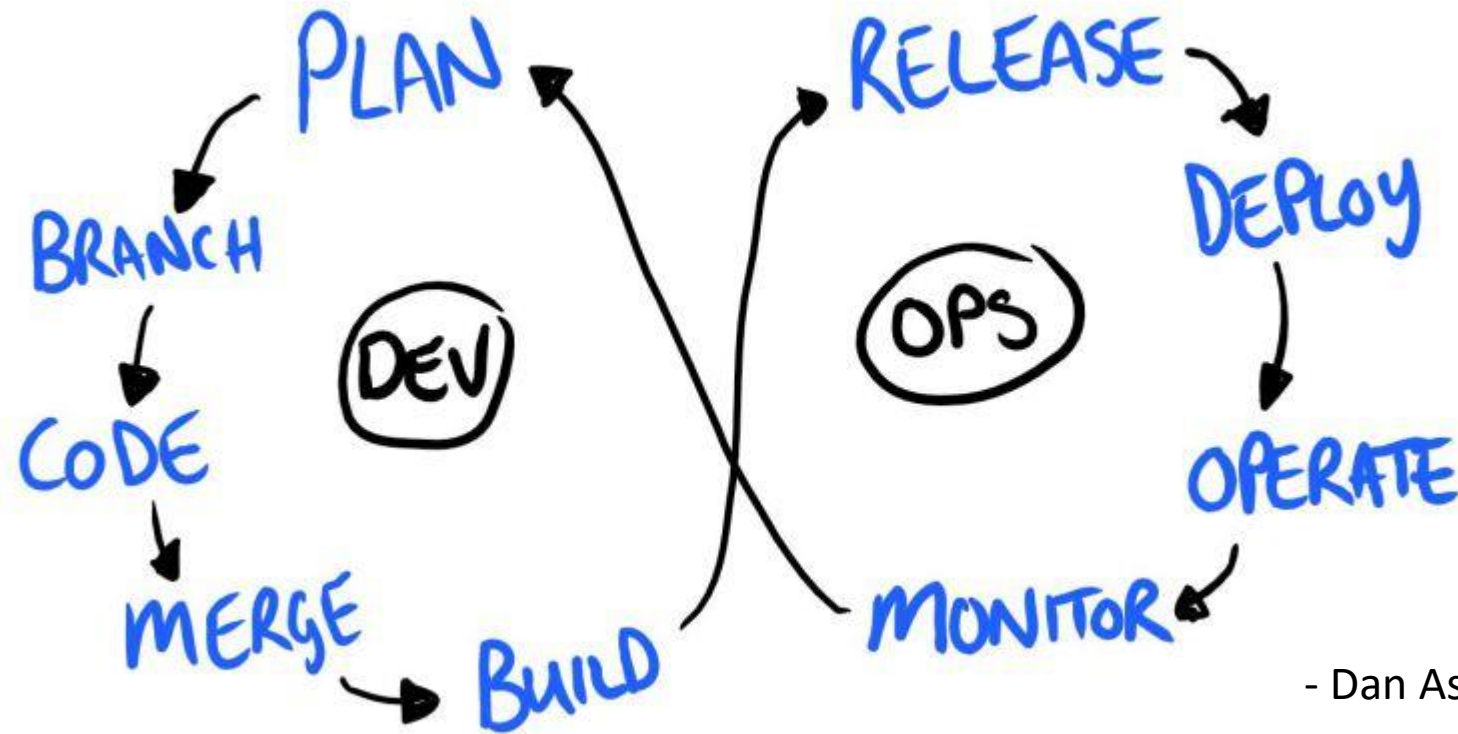
What is DevOps?

Software engineering culture and practice aimed at unifying software development (Dev) and software operation (Ops)



Shorter development cycles
Increased deployment frequency
Higher quality releases
Close alignment with business objectives

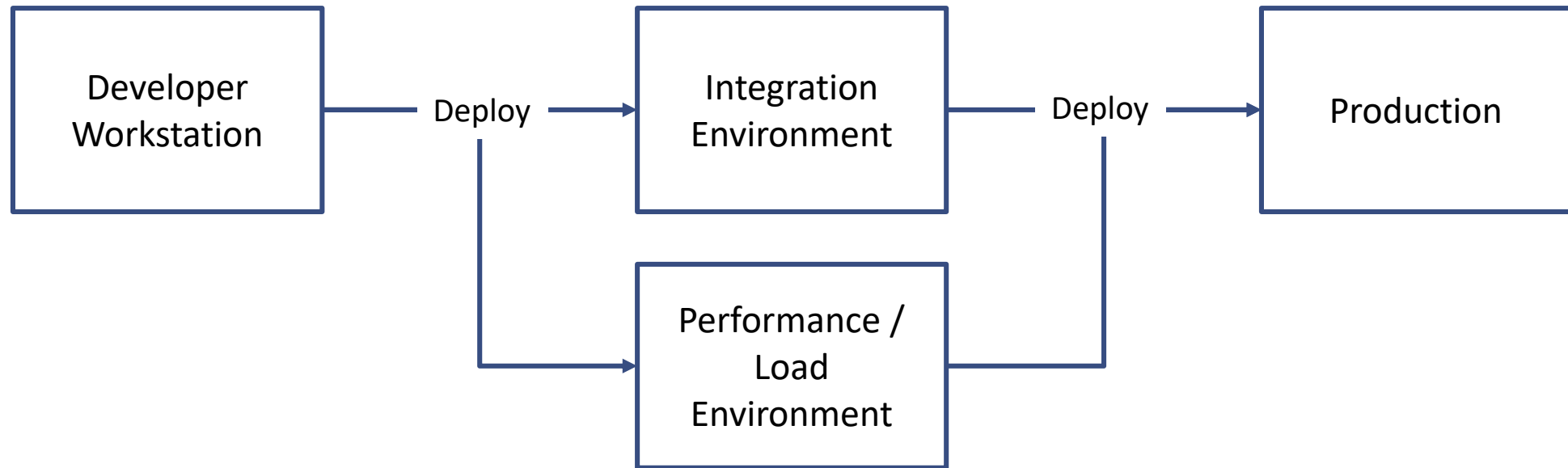
The DevOps Cycle



- Dan Ashby

<https://danashby.co.uk/2016/10/19/continuous-testing-in-devops/>

A Delivery Pipeline



Continuous Integration	Frequent integration of developers' code into a shared repository
Continuous Delivery	Develop, build, test and release software in short cycles
Continuous Deployment	Automated deployment of qualified application features to production
Continuous Testing	<i>Stay tuned!</i>

Continuous Processes ≠ DevOps

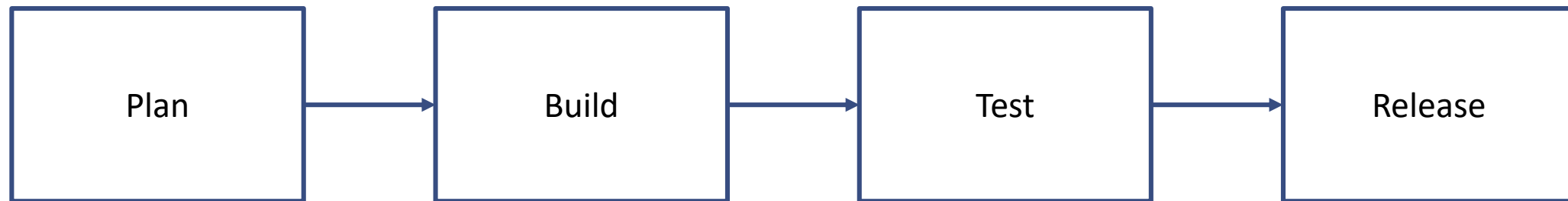
State of DevOps

Aspect of Delivery Performance	Elite	High	Medium	Low
Deployment Frequency	On-demand (multiple deploys per day)	Once per hour to once per day	Once per week to once per month	Once per week to once per month
Lead Time for Changes (commit to production)	Less than one hour	One day to one week	One week to one month	One month to six months
Time to Restore Service (outage or impairment)	Less than one hour	Less than one day	Less than one day	One week to one month
Change Failure Rate (% of changes requiring rollback, hot fix, patch, etc.)	0 to 15%	0 to 15%	0 to 15%	46 to 60%

2018 Accelerate State of DevOps Report

Testing in DevOps

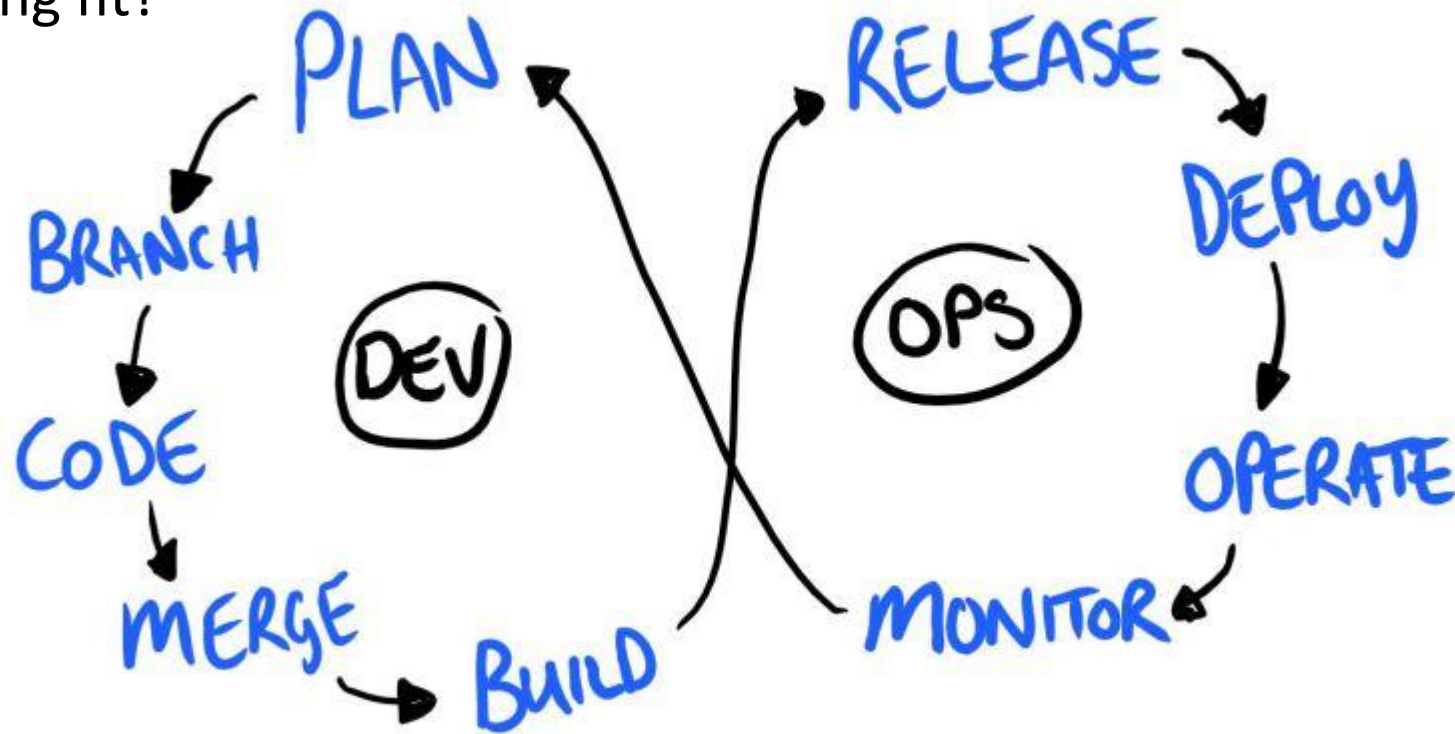
Traditional Testing in DevOps?



Testing is PART of the dev cycle

Testing in DevOps

Where does testing fit?

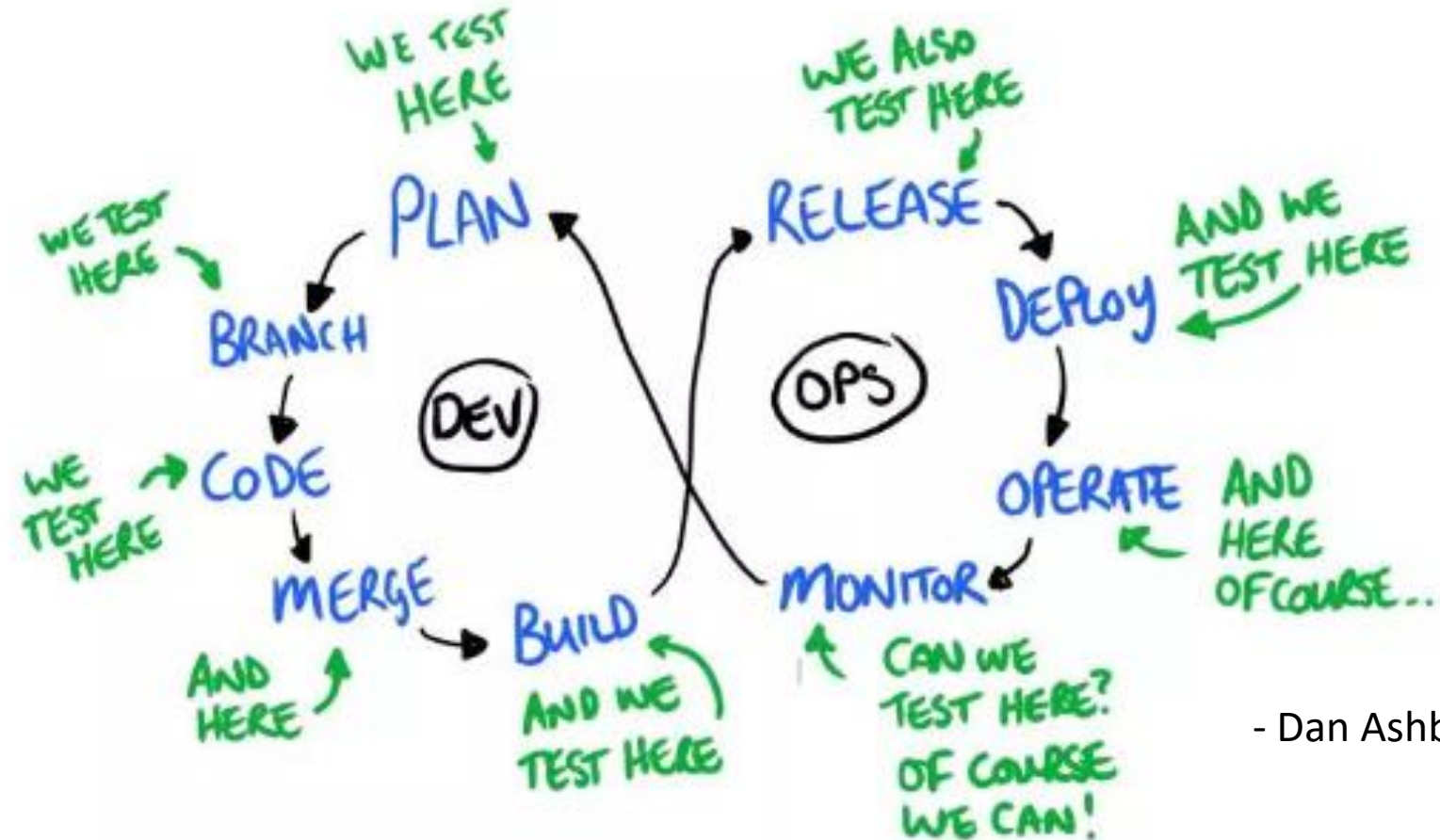


Testing in DevOps

Testing spreads
THROUGHOUT the
development cycle



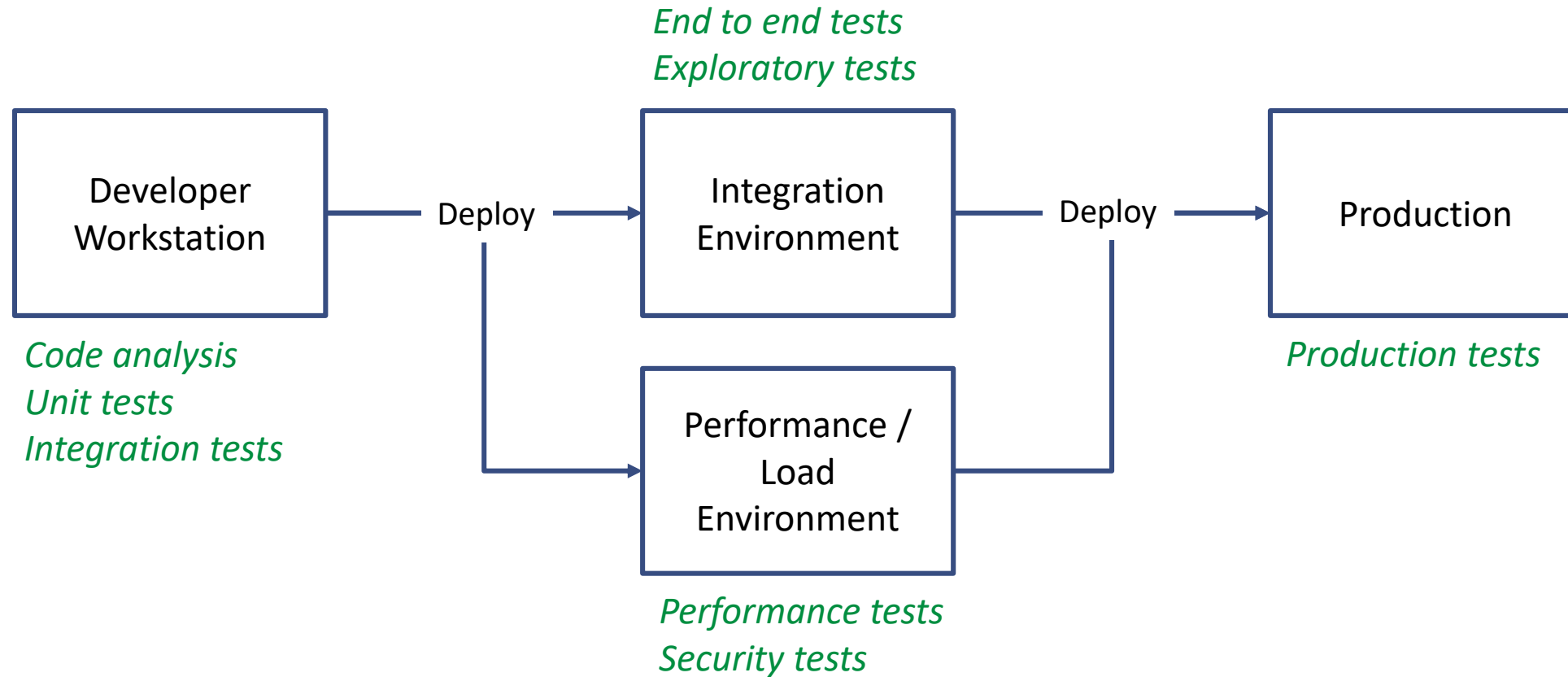
... and becomes
everyone's
responsibility



- Dan Ashby

<https://danashby.co.uk/2016/10/19/continuous-testing-in-devops/>

Testing Across the Delivery Pipeline



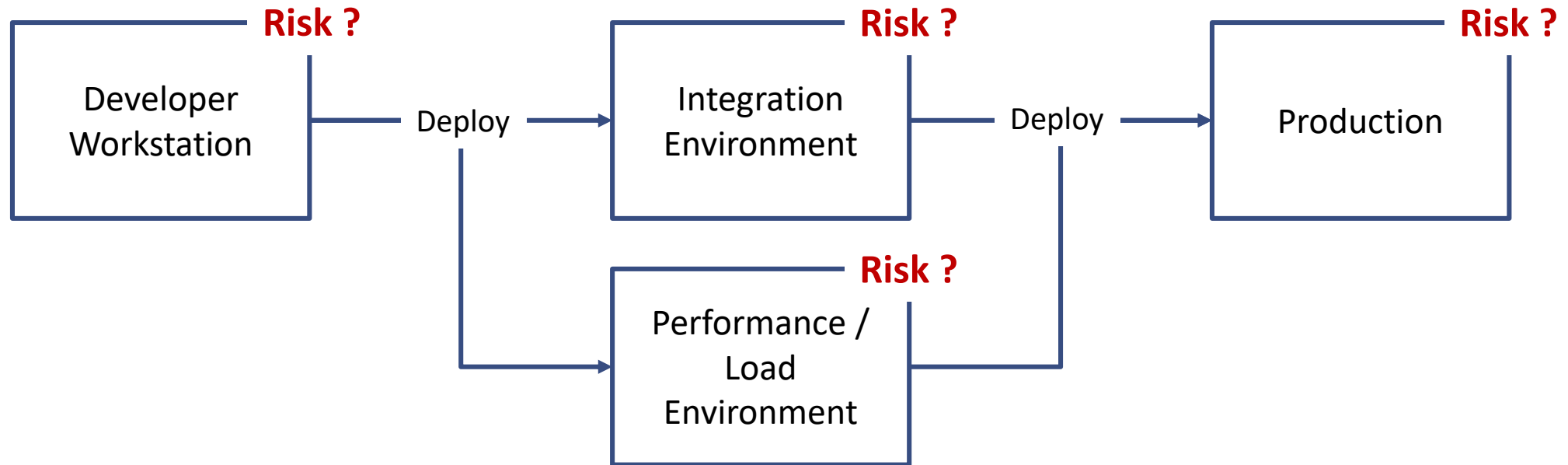
DevOps testing activities must provide

- ✓ Fast
- ✓ Accurate feedback
- ✓ Actionable

Continuous Testing

Continuous Testing

Running the right tests at the right time to assess the risk of promoting a build through the delivery pipeline



Successful Continuous Testing

- Provides fast, accurate and actionable feedback
- Avoids creating a bottleneck in the delivery pipeline
- Is embedded throughout the development process (shift Left AND shift Right)
- Includes effective test automation practices
- Promotes effective manual testing activities
- Is executed in stable test environments with valid test data
- Continually reviews and refines test suites to ensure relevance and efficiency

Continuous Testing vs. Test Automation

Test Automation

Pass/fail results associated with user stories and business requirements



Continuous Testing

Business risk associated with releasing the software

Primary Differences

Risk

Tests built with business risk in mind

Breadth

Tests that detect issues that impact user experience

Time

Tests that provide immediate feedback and are executed as close as possible to app element they are testing

- Adapted from Continuous Testing vs Test Automation by Wayne Ariola

Test Automation in DevOps

Importance of Effective Automation

Continuous Integration requires three capabilities:

- A **comprehensive and reliable set of automated tests** that validate we are in a deployable state
- A culture that “**stops the entire production line**” when our validation tests fail
- Developers working in small batches on trunk rather than long-lived feature branches

The DevOps Handbook (Kim, Humble, Debois & Willis)

Automation Impact on DevOps Performance

Type of Manual Activity	Elite	High	Medium	Low
Configuration Management	5%	10%	30%	30%
Testing	10%	20%	50%	30%
Deployments	5%	10%	30%	30%
Change Approvals	10%	30%	75%	40%

- 2018 Accelerate State of DevOps Report

Current State of Test Automation

Some alarming statistics

The level of test automation is below 20%

2017-2018 World
Quality Report

The value generated by test automation is largely
unchanged

2017-2018 World
Quality Report

Almost 80% of test automation implementations
fail

Unknown

Obstacles to Test Automation Success



Common Automation Traps



- Too much / not enough automation
- Automating the wrong tests
- Focusing on UI tests
- Unstable test environments and data
- Not treating automation as code

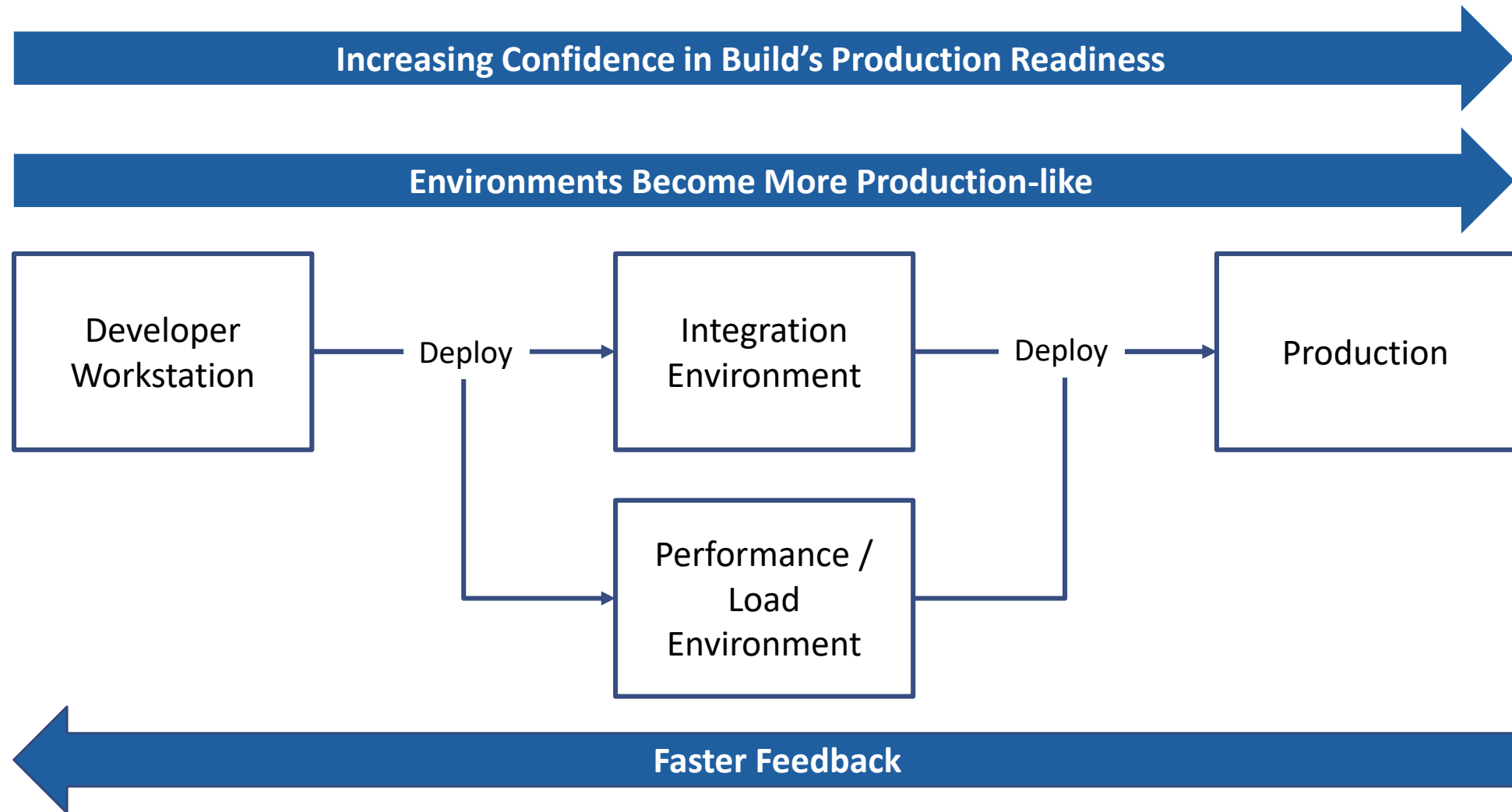
Effective Automation

Pillars of Effective Automation



SCOPE AND APPROACH

Pipeline Tradeoffs



- Continuous Delivery by Jez Humble and David Farley

Which Test Suites Do We Need?

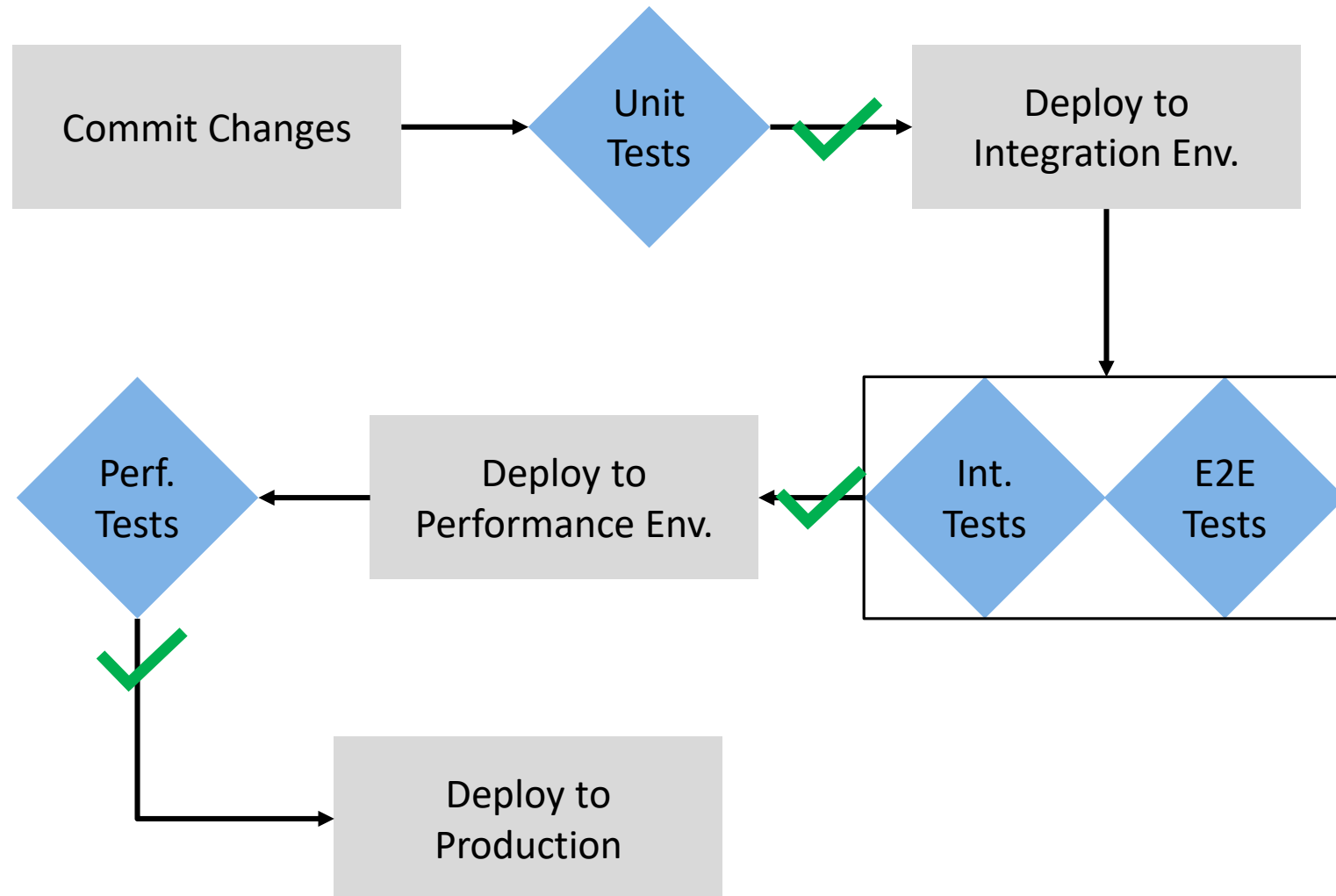
Test Suite Canvas for evaluating test suites

Purpose	Dependencies	Constraints	Pipeline / Execution	Environment / Data
What business risk does this test suite mitigate?	What systems or tools must be in place for this test suite to run successfully?	What can prevent us from implementing this test suite in an ideal way?	Will this test suite be part of a pipeline? When will it be triggered? How often will it run?	What environment will the test suite run in? How will test data be managed?
Ownership and Response		Maintainability		Effectiveness
Who will create the test suite? Who should own it? Who will address test failures and how?	What will the process be to review code? What documentation needs to exist?		How will we know the test suite is effective?	

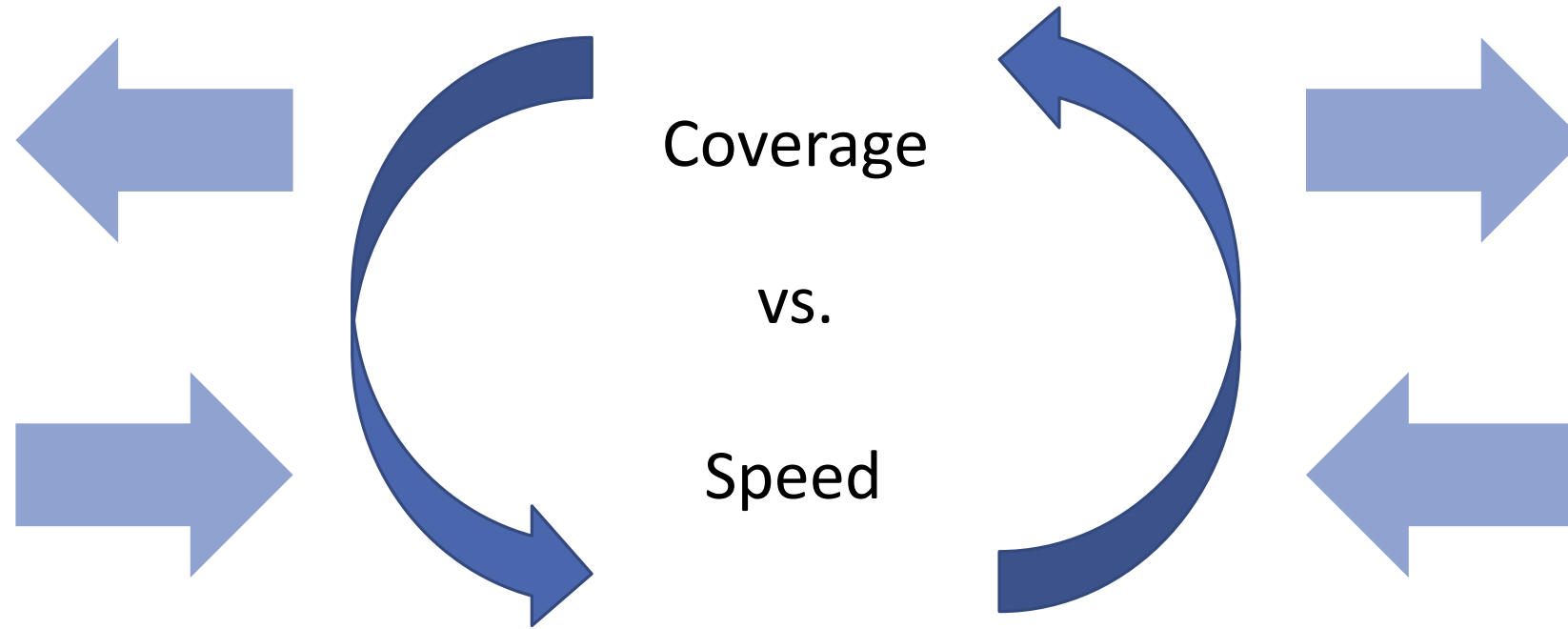
- Ashley Hunsberger (adapted from original concept by Katrina Clokie)

Test Suites Mapped to the Pipeline

Suite	Purpose
Unit Tests	Is the change being pushed ready for a code review?
Integration Tests	Are the integrated system components ready for further testing?
End-to-End Tests	Is the system functionally ready for deployment?
Performance	Does the system meet performance SLAs?
...	...



Test Scope: Risk vs. Speed



Feedback Loop

Which Tests Do We Need?

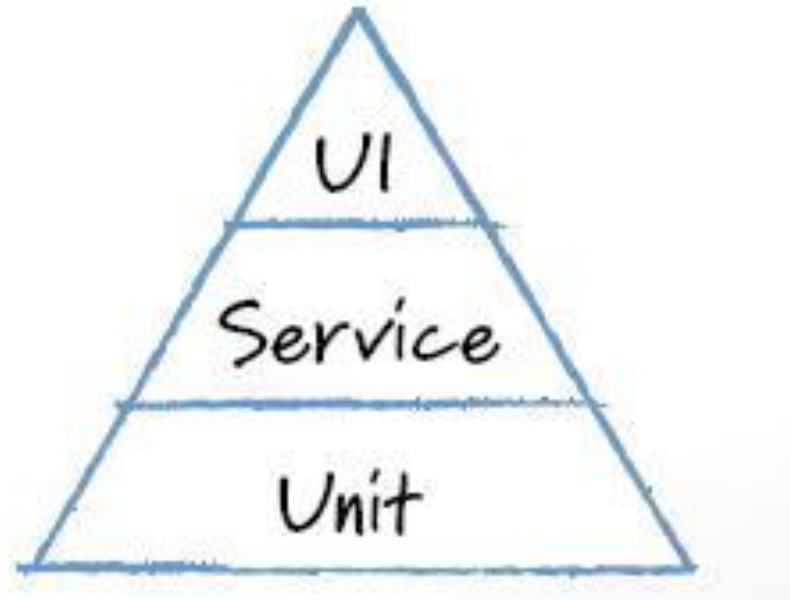
Using **FITR** to assess your automated tests for inclusion

- Focused** Test is tied as close to the functionality being tested as possible ► UI, API, unit, etc.
- Informative** Test clearly and concisely communicates its intent and result
- Trustworthy** Test executes reliably and doesn't provide false negatives / positives
- Repeatable** Test can be executed on demand ► environment and data dependencies

- Bas Dijkstra

<https://www.stickyminds.com/sites/default/files/volume-issue/pdf/V2013.pdf>

Test Pyramid

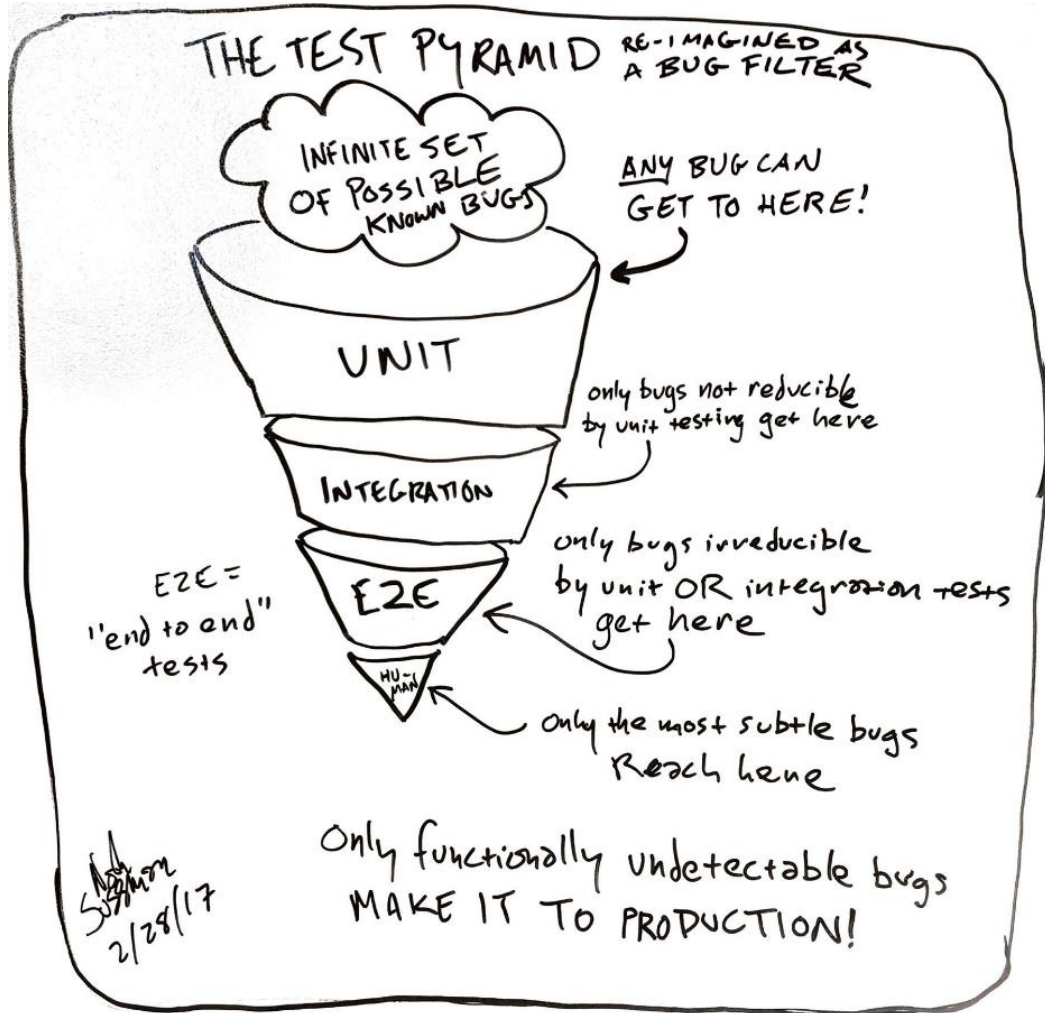


- Mike Cohn, Mountain Goat Software

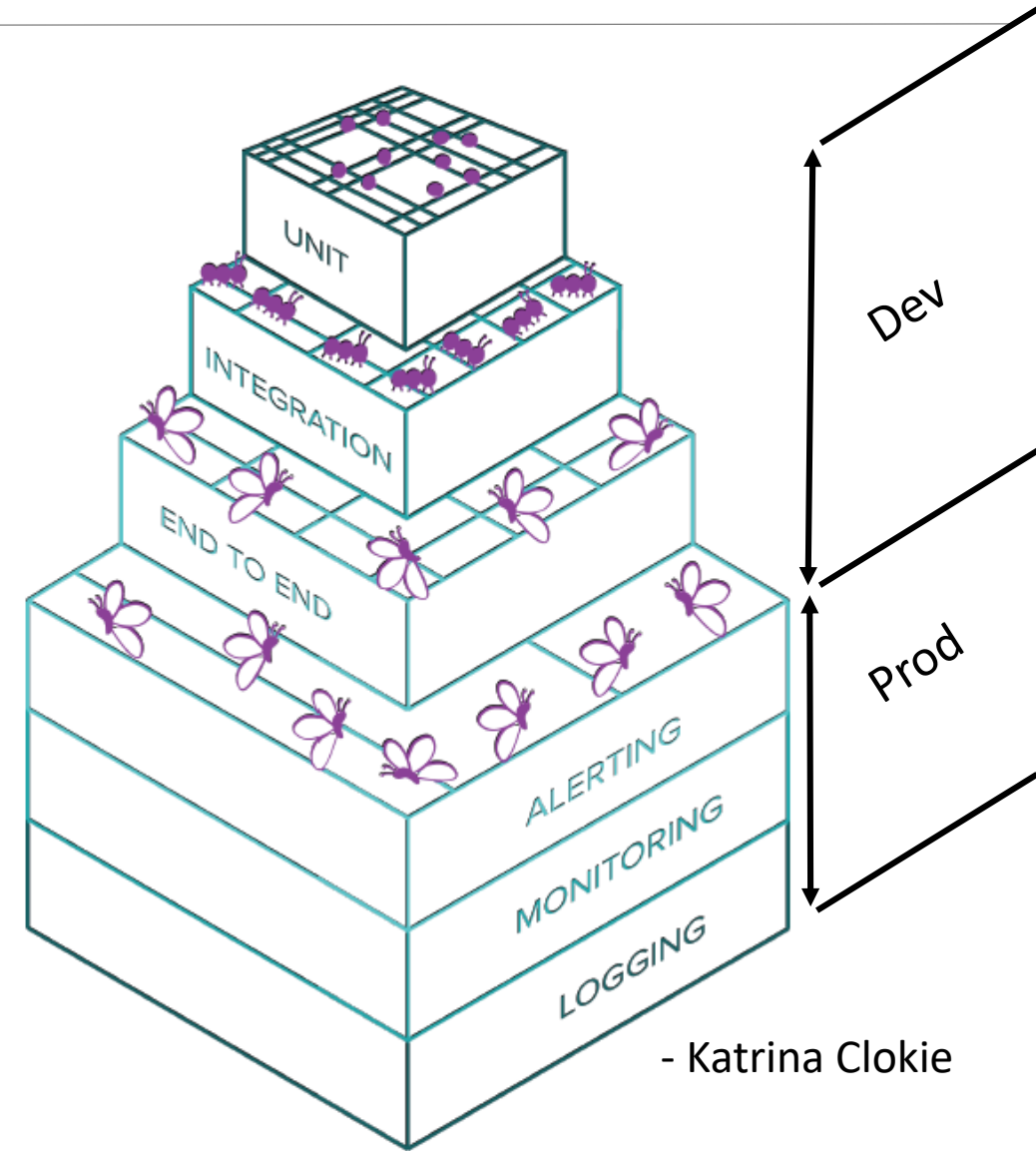
“All models are wrong, but some are useful”

George Box, Statistician

Test Pyramid... Re-imagined

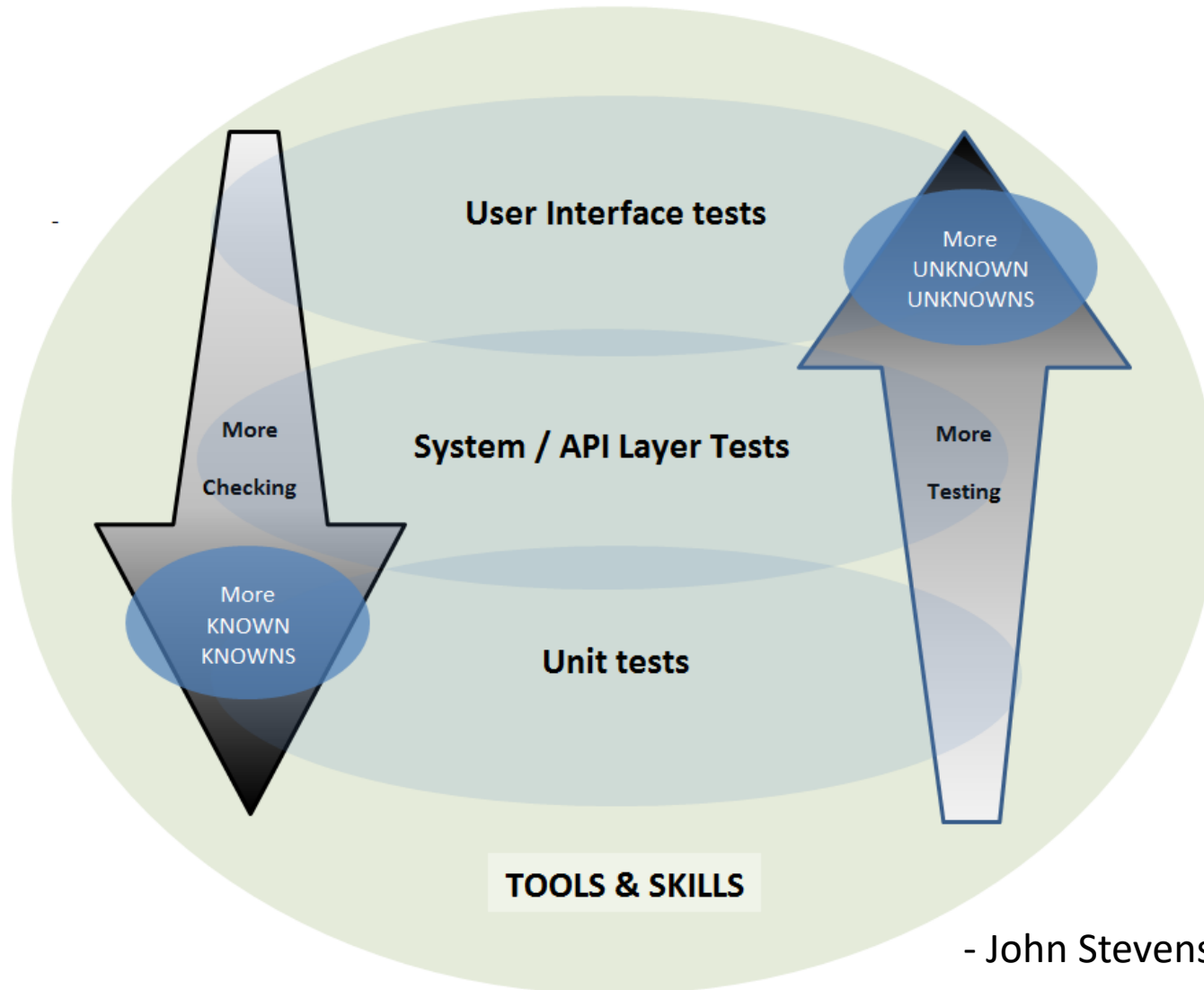


- Noah Sussman



- Katrina Clokie

Or Not a Pyramid at All...



- John Stevenson

- Gives equal importance to each layer
- Removes the focus on solely automation
- Highlights that tools and skills must be used appropriately throughout

Continuous Testing Improvement

- If an automated acceptance test fails -> investigate writing a unit test to catch the problem
- If a defect is discovered in exploratory testing -> write an automated unit or acceptance test as appropriate
- If an automated test generates a false positive error -> fix the test or remove it



If tests are unreliable, they will lose their privilege of being in the build pipeline and eventually decay into uselessness

ENVIRONMENT AND DATA

Biggest Challenge to Automation?

Test environment and data availability continue to be top challenges to achieving desired levels of automation

- 2018-19 World Quality Report

Well... it ~~works on my machine!~~

Test Environment and Data Spectrum

Static
Shared
Stale

Environment / Data Issues

- Data staleness and instability
- Scheduling conflicts
- Not production-like

Automation Averse

- Unreliable execution
- Long/complex setup
- False positives
- Prolonged analysis

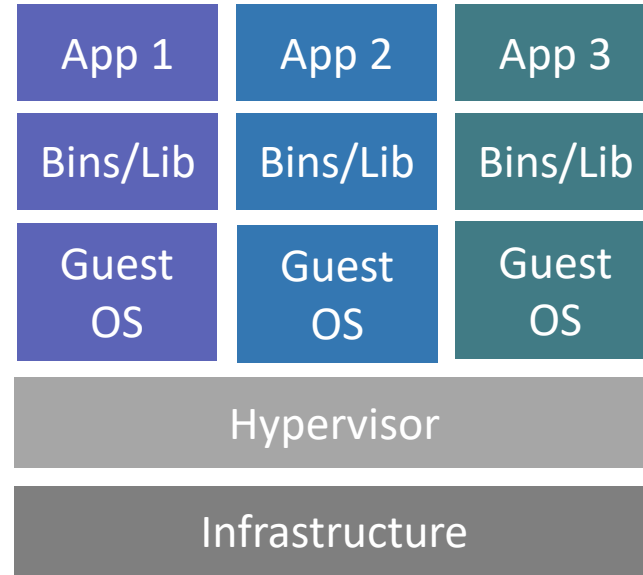
Dynamic
Dedicated
Fresh

Automation Friendly

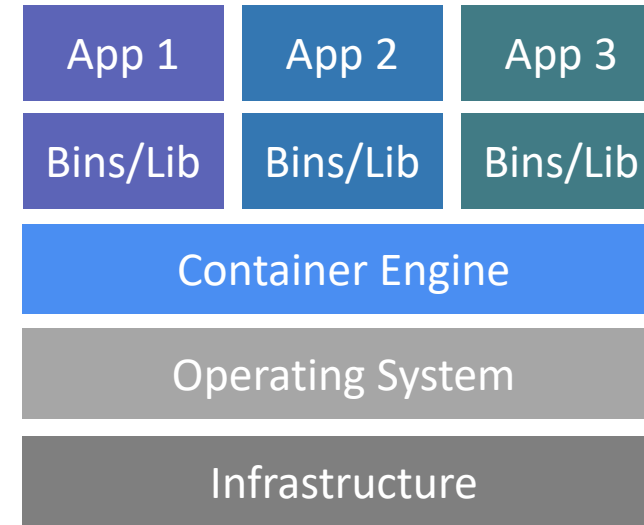
- Create clean environments on demand
- Dedicate environments to single purpose
- Production-like (if needed)
- Load with clean/fresh data

DevOps Environment Technologies

Evolved Platforms



Virtual Machines



Containers



Cloud

Infrastructure as Code



Cloud / Virtual Machines / Containers + Infrastructure as code



- Environment configurations can be version controlled
- Environment state can be included with test results
- Manual process becomes automated ► repeatable, scalable and reliable
- Ability to develop and test in more production-like environments
- Environment creation can become part of the delivery pipeline if desired
- Infrastructure management processes/code can be tested via automation
- Destructive testing

Test Data Challenges

Three of the top test data challenges according to the World Quality Report 2017-18

- Keeping test data in sync with tests
- Lack of test data for complex integration testing
- Maintaining consistent test data across systems under test



Inability to reliably execute automated tests on demand

Test Data Management Strategy

Key Strategy Input Questions

What data do your tests require?

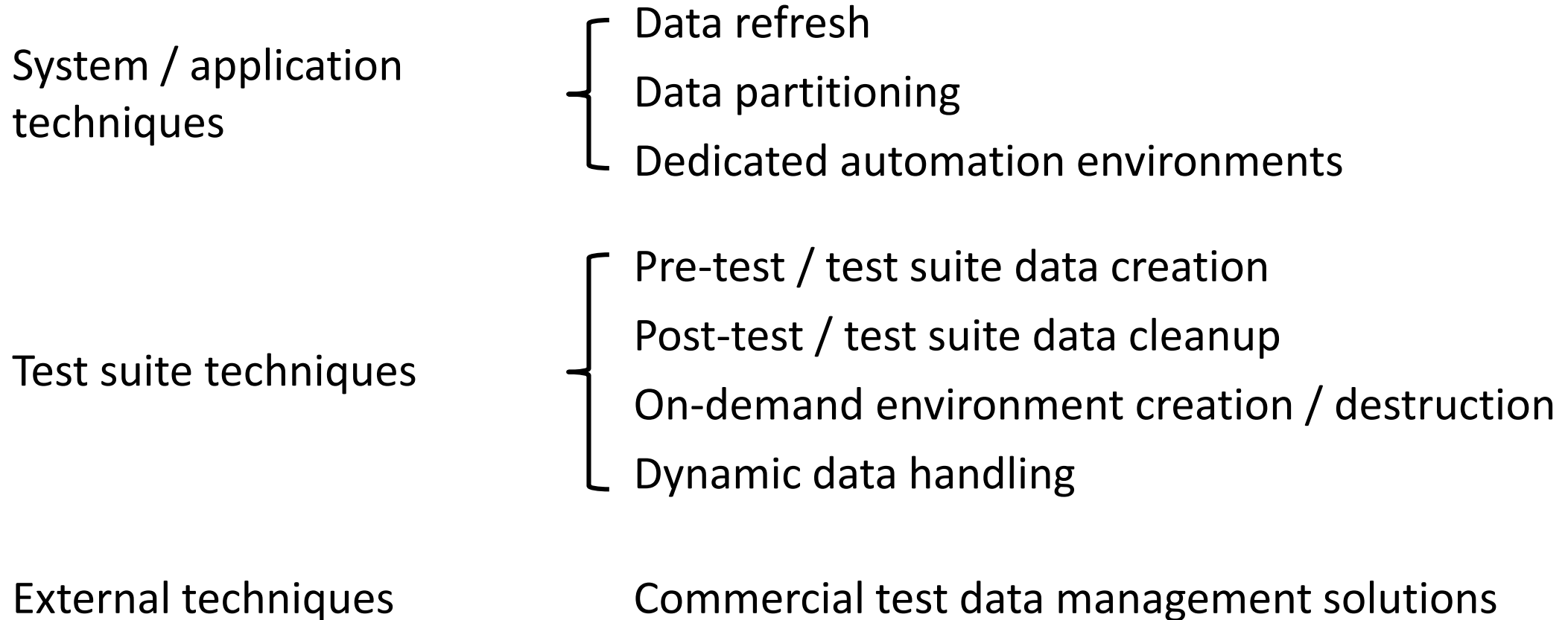
What are the dimensions of the required data?

What are the data sources?

}	Type	is data structured or unstructured?
	Value	what specific values or class of data are required?
	Time	does data change over time?
	Reuse	Can data be used more than once?
	Volume	how much do data you need?

Test Data Management Strategy

Common Data Management Techniques



Dynamic Data Handling Techniques

Handle application/system data that changes periodically



Static metadata tags used in test cases

Login with <ADMIN_CREDENTIALS>



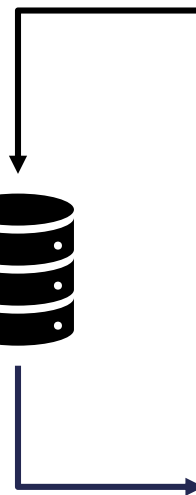
Metadata tags mapped to current data values

ADMIN_CREDENTIALS = Administrator / \$Yyfht%\$as



Test suite replaces tags with current data at runtime

Centralized test data repository

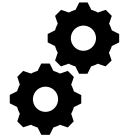


Dynamic Data Handling Techniques

Handle test input data that must be dynamic or unique

Parameter	Value
FirstName	<AUTO_FirstName>
LastName	<AUTO_LastName>
DateOfBirth	<AUTO_DOB 35_45>
Email	<AUTO_Email mailinator.com>

Framework Data
Generation Utilities



Test cases contain data generation tags



Test suite replaces parameter tags at runtime

Parameter	Value
FirstName	John
LastName	Smith
DateOfBirth	06-29-1977
Email	john.smith@mailinator.com

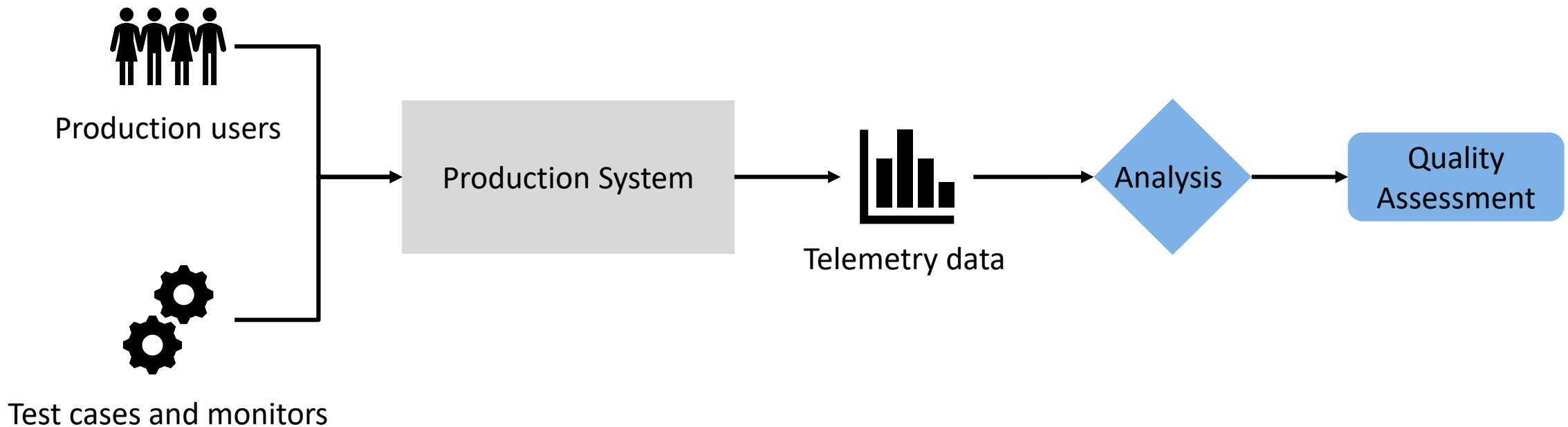
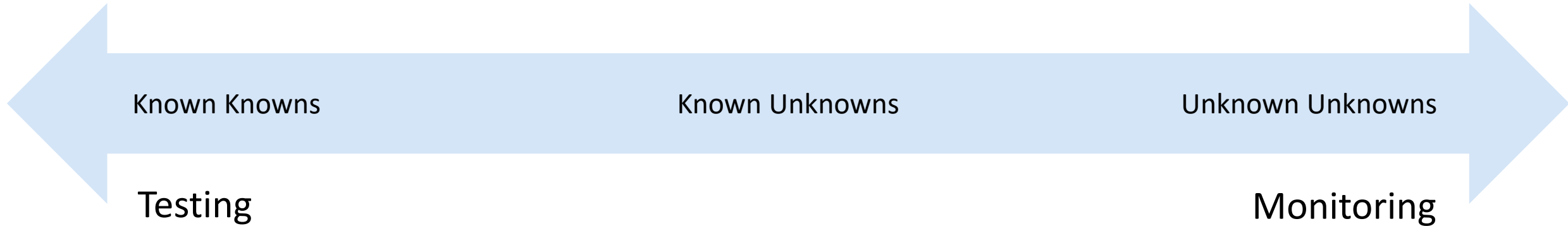
Service Virtualization

Supports stable execution for tests that require access to application dependencies including:

- Dependencies that have not yet been developed
- Dependencies that don't have stable/appropriate data
- 3rd party dependencies that don't allow testing
- Any other situation where the dependency can't be interacted with in a stable and predictable manner

Testing in Production

Testing vs. Monitoring in Production



Production Testing

- Not a replacement for upstream testing
- Tests should evolve from heartbeats to user impacting scenarios
- Systems should “know” they are being tested

Production Monitoring

- Development needs to be aware of monitoring info
- Monitoring and alerting should be tested



- Production testing and monitoring strategy requires broad IT input
- Feedback (analytics/logging) should be clear and concise
- Testability of the system should be continuously reviewed

Key Takeaways

- DevOps is a culture, not just processes and tools
- Effective test automation is required to achieve Continuous Testing in DevOps
- Focus on the three pillars to ensure fast, reliable and accurate feedback
 - Scope
 - Approach
 - Environment and Data Management

Questions...



.... and Answers!

Direct future questions to:

Lee Barnes

CTO

Utopia Solutions, Inc.

Email: lee.barnes@utopiasolutions.com

Twitter: @USI_LeeBarnes

LinkedIn: [linkedin.com/in/leebarne](https://www.linkedin.com/in/leebarne)

Blog: utopiasolutions.com/blog1

References

- Ariola, W. (2018). *Continuous Testing vs Test Automation: What Testers Need to Know*
https://tricentis-com-tricentis.netdna-ssl.com/wp-content/uploads/2018/06/Continuous_Testing_vs_Test_Automation0618.pdf
- Ashby, D. (2016). *Continuous Testing in DevOps*
<https://danashby.co.uk/2016/10/19/continuous-testing-in-devops/>
- Clokie, K. (2017). *Test Automation Canvas*
<http://katrinatester.blogspot.com/2017/07/test-automation-canvas.html>
- Clokie, K. (2017). *A Practical Guide to Testing in DevOps*
- Dijkstra, B. (2018). *Support Continuous Testing with Automation*. Better Software, Summer 2018
<https://www.stickyminds.com/sites/default/files/volume-issue/pdf/V2013.pdf>
- Eliot, S. *The Future of Software Testing Part Two*
<https://www.ministryoftesting.com/dojo/lessons/the-future-of-software-testing-part-two>
- Humble, J., Farley, D. (2010) *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*
- Hunsberger, A. (2016). *Transform Culture Using DevOps Principles*. SauceCon Conference 2018.
<https://www.youtube.com/watch?v=RBrAj9jKgX0>
- Kim, G., Humble, J., Debois, P., & Willis, J. (2016) *The DevOps Handbook*
- Stevenson, J. (2015). Blog Post based on MEWT4 Session titled *Sigh, It's That Pyramid Again* by Richard Bradshaw
<http://steveo1967.blogspot.com/2015/10/mewt4-post-1-sigh-its-that-pyramid.html>
- Wikipedia – DevOps
<https://en.wikipedia.org/wiki/DevOps>
- World Quality Report 2017-18
- World Quality Report 2018-19