Effective Test Automation in DevOps

MOVING TOWARDS CONTINUOUS TESTING

@USI_LeeBarnes

Agenda

- Test Automation in DevOps

- Common Automation Obstacles

- Pillars of Effective Test Automation
  - Scope
  - Approach
  - Test Environment and Data Management

- Wrap Up & Questions

@USI_LeeBarnes

Testing in DevOps

DevOps testing activities must provide
- ✓ Fast
- ✓ Accurate   feedback
- ✓ Actionable

@USI_LeeBarnes

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)

@USI_LeeBarnes

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

@USI_LeeBarnes

What's the Big Deal?

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

@USI_LeeBarnes

What is "Test Automation"

Software Testing   Using our knowledge, experience, and critical thinking to design experiments with the purpose of exposing issues in software systems

Automation   Operating or controlling a process by highly automatic means, reducing human intervention to a minimum.

We are NOT automating the TESTING!

Obstacles to Test Automation Success

Common Automation Traps

- Too much / not enough automation
- Automating the wrong tests
- Focusing on UI interaction
- Unstable test environments and data
- Not treating automation as code
- Forgetting to improve
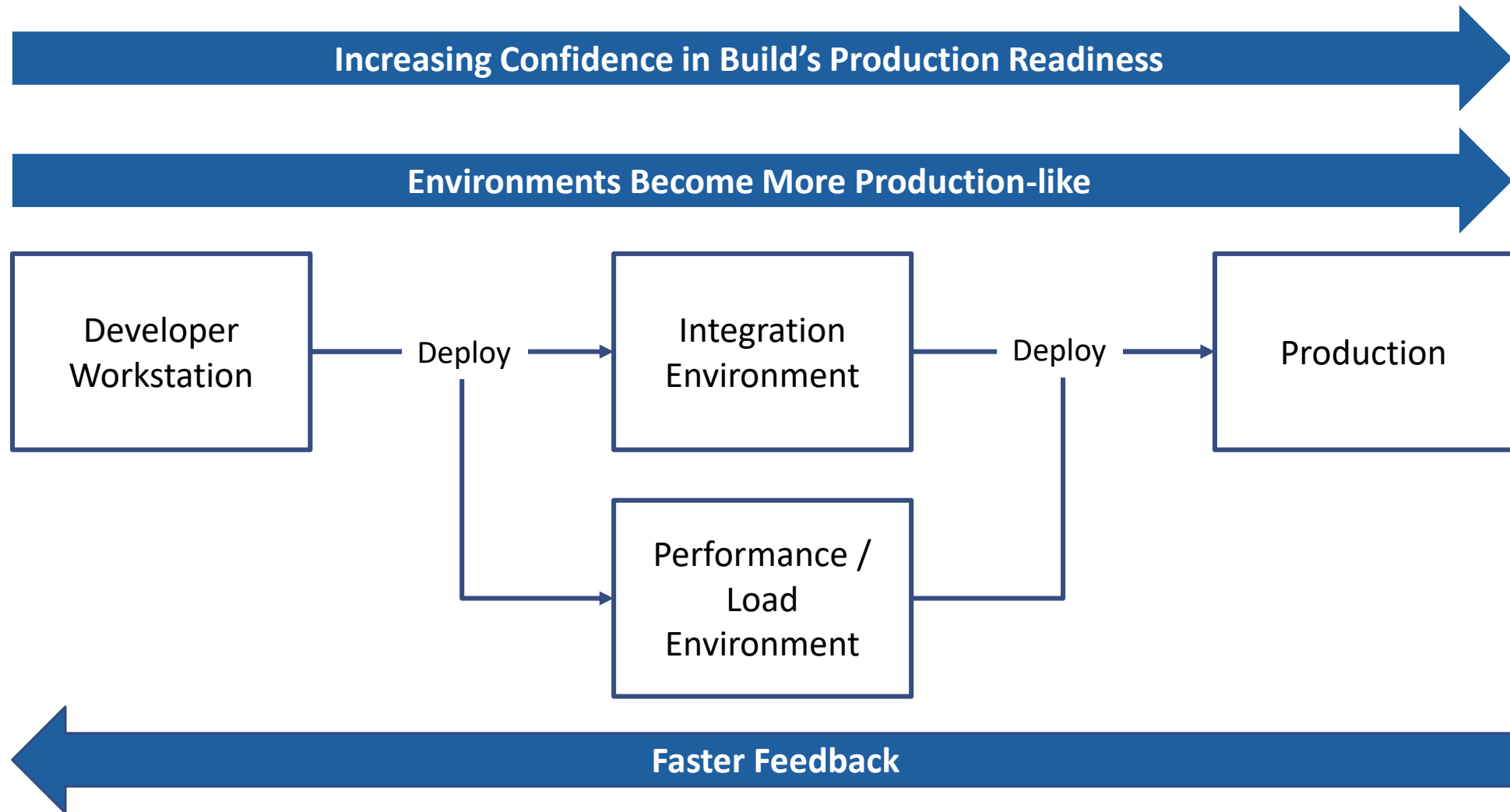
@USI_LeeBarnes

Effective Test Automation

@USI_LeeBarnes

Pillars of Effective Automation

Scope

Approach / Method

Environment & Data Management

@USI_LeeBarnes

Scope and Approach

@USI_LeeBarnes

Pipeline Tradeoffs

UTOPIA
SOLUTIONS

Increasing Confidence in Build's Production Readiness →

Environments Become More Production-like →

```
┌──────────────┐           ┌──────────────┐           ┌──────────────┐
│  Developer   │  Deploy → │ Integration  │  Deploy → │  Production  │
│ Workstation  │           │ Environment  │           │              │
└──────────────┘           └──────────────┘           └──────────────┘
                 ┌──────────────┐
                 │ Performance/ │
                 │     Load     │
                 │ Environment  │
                 └──────────────┘
```

← **Faster Feedback**

@USI_LeeBarnes

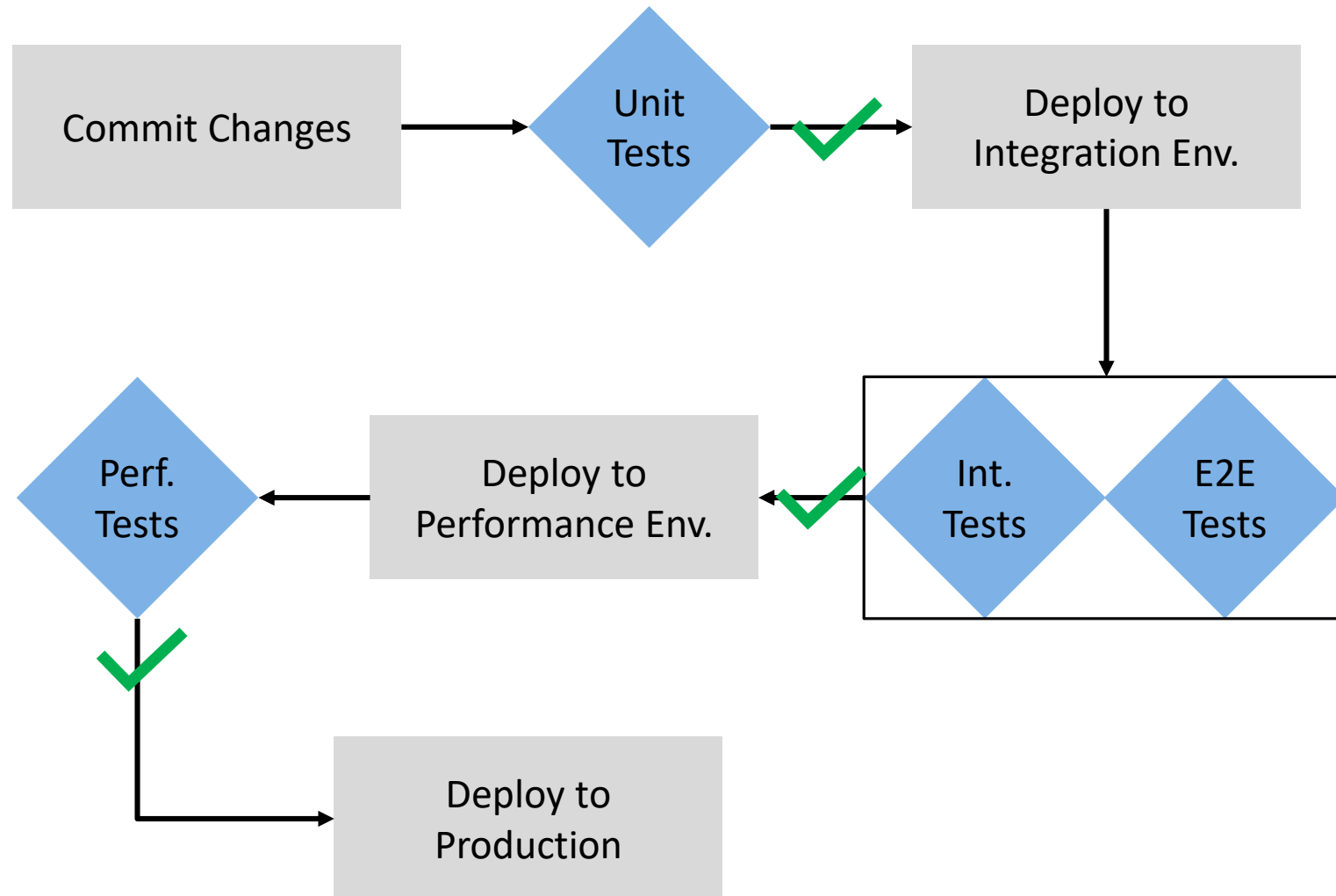- Continuous Delivery by Jez Humble and David Farley

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?

@USI_LeeBarnes

- 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?
…	…

Commit Changes → Unit Tests ✓ → Deploy to Integration Env.

Int. Tests E2E Tests

Perf. Tests ← Deploy to Performance Env. ← ✓ Int. Tests / E2E Tests

✓ → Deploy to Production

@USI_LeeBarnes

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/V20I3.pdf

@USI_LeeBarnes

Environment and Data
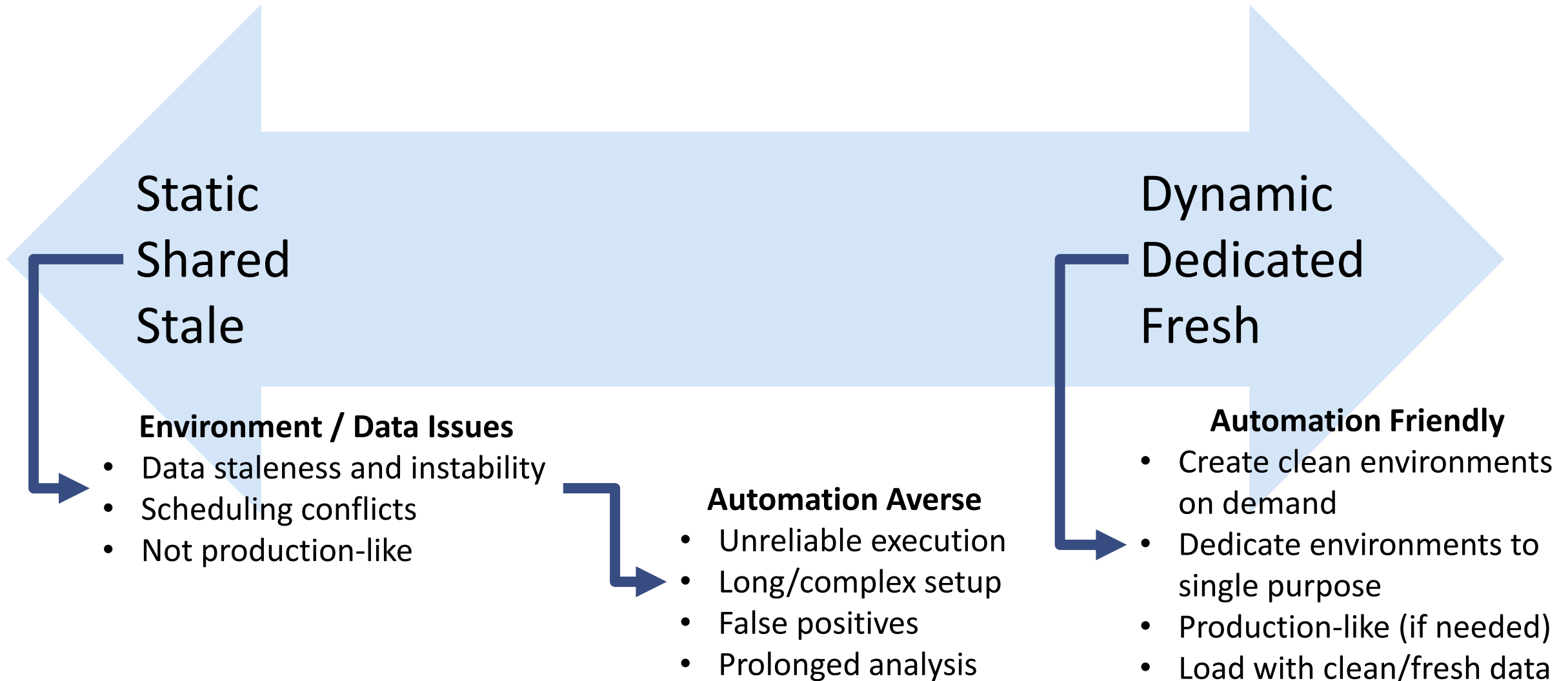
@USI_LeeBarnes

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 worked on my machine!

@USI_LeeBarnes

Test Environment and Data Spectrum

Static
Shared
Stale

Dynamic
Dedicated
Fresh

Environment / Data Issues
- Data staleness and instability
- Scheduling conflicts
- Not production-like

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

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

@USI_LeeBarnes

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

@USI_LeeBarnes

Test Data Management Strategy

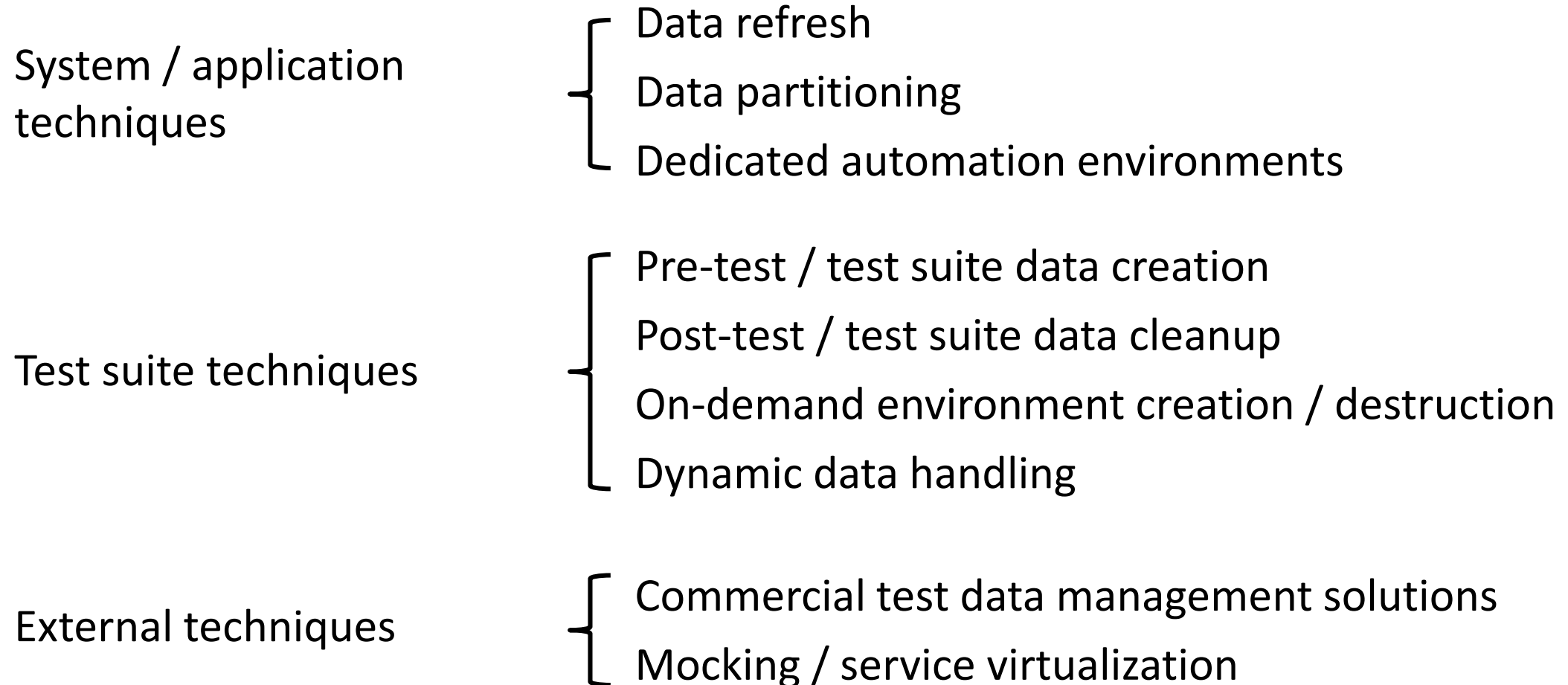**Key Strategy Input Questions**

What data do your tests require?

What are the dimensions of the required data?

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?	

What are the data sources?

@USI_LeeBarnes

Test Data Management Strategy

Common Data Management Techniques

System / application techniques
- Data refresh
- Data partitioning
- Dedicated automation environments

Test suite techniques
- Pre-test / test suite data creation
- Post-test / test suite data cleanup
- On-demand environment creation / destruction
- Dynamic data handling

External techniques
- Commercial test data management solutions
- Mocking / service virtualization

@USI_LeeBarnes

Key Takeaways

- Effective test automation is required to achieve success in DevOps

- Create the right test suites that contain the right tests

- Ensure you understand your test environment & data requirements AND have created a strategy to manage them

- Continuously improve

- Notice that we didn't talk about specific tools...

@USI_LeeBarnes

Continuing the Conversation

Lee Barnes

CTO

Utopia Solutions, Inc.

Email: lee.barnes@utopiasolutions.com

Twitter: @USI_LeeBarnes

LinkedIn: linkedin.com/in/leebarnes

Blog: utopiasolutions.com/blog

@USI_LeeBarnes

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 thinking about automation the right way

Email: lee.barnes@utopiasolutions.com
Twitter: @USI_LeeBarnes
LinkedIn: linkedin.com/in/leebarnes
Blog: utopiasolutions.com/blog

@USI_LeeBarnes