DevOps' Missing Link: Data

Straight Talk on why DevOps is still failing.

Trac Bannon, Senior Principal
July 2021

MITRE | SOLVING PROBLEMS FOR A SAFER WORLD™

Who am I?

Tracy L. Bannon

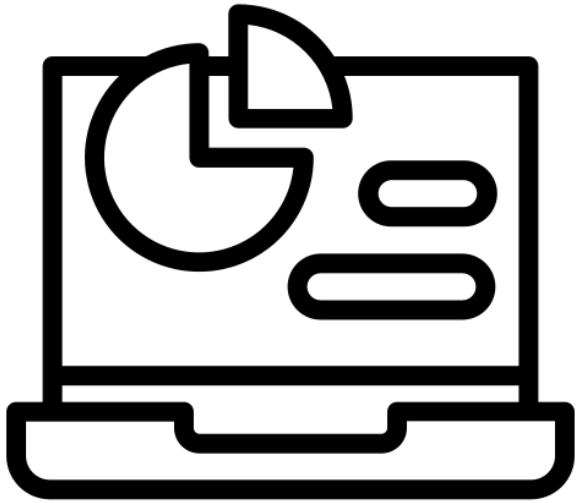
- ✓ Senior Principal with the MITRE Corporation
- ✓ Software Architect and Engineer
- ✓ Focused on problem solving using software

/trās/

What are my tags?

Are we there yet?

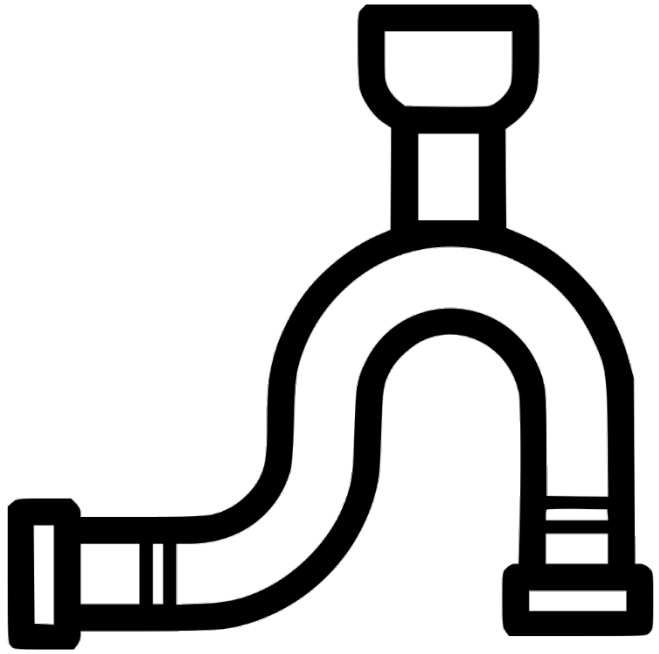
Growing body of knowledge on how to address common DevOps adoption challenges.



- Misunderstanding of what it is
- Resistance to change
- Overcoming the Dev versus Ops mentality
- Too much focus on tools[1]

We are addressing these challenges and there are still problems…

Is Quality Going Down the Drain with the Pipeline?[22]



We are automating testing though quality escapes (aka defects) are making it to production

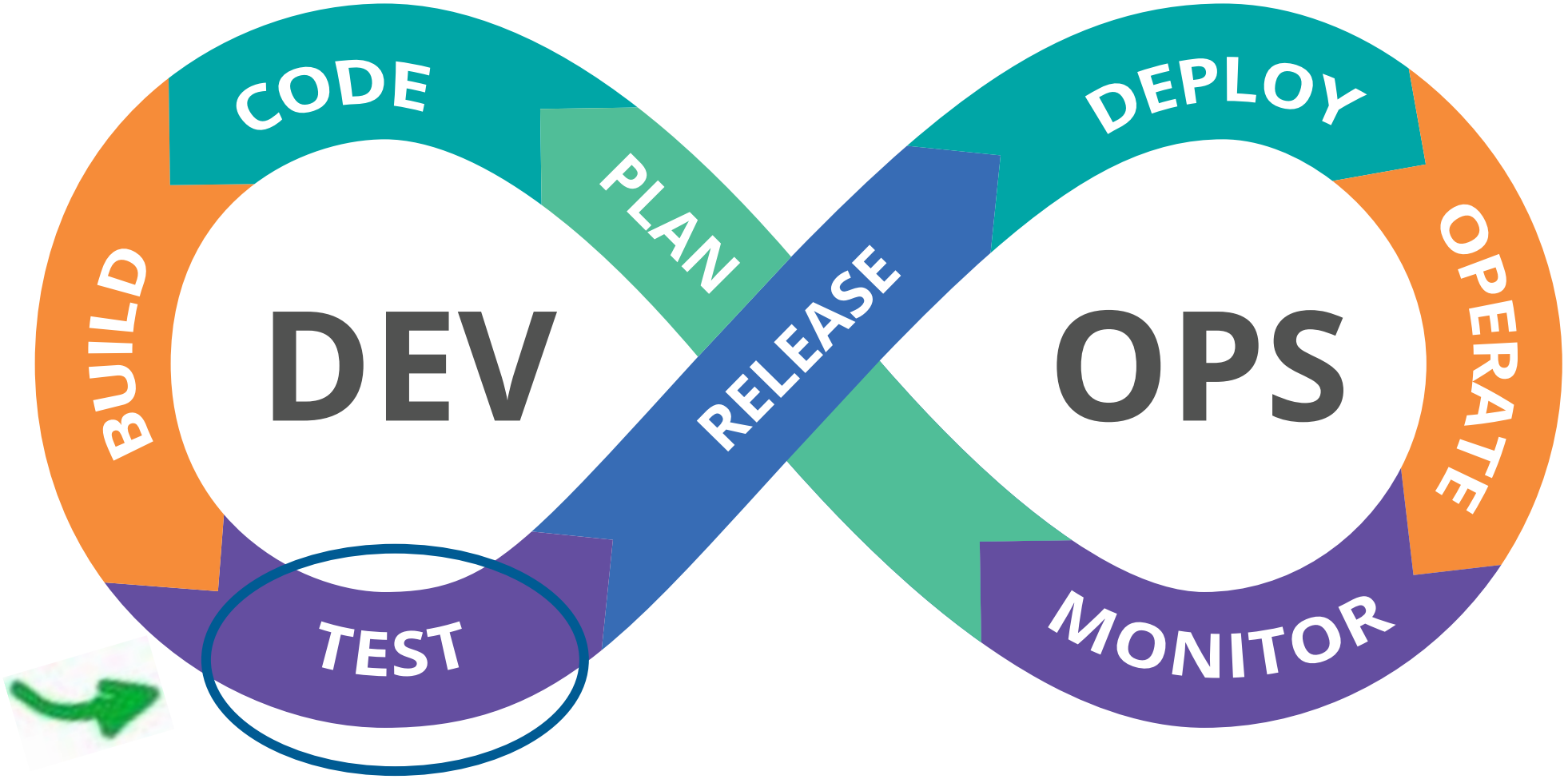
Advice and examples abound for

- Automated testing tools
- Metrics and measurements

With a focus on automated testing, how can quality be an issue?

Image Courtesy of www.pinclipart.com[38]

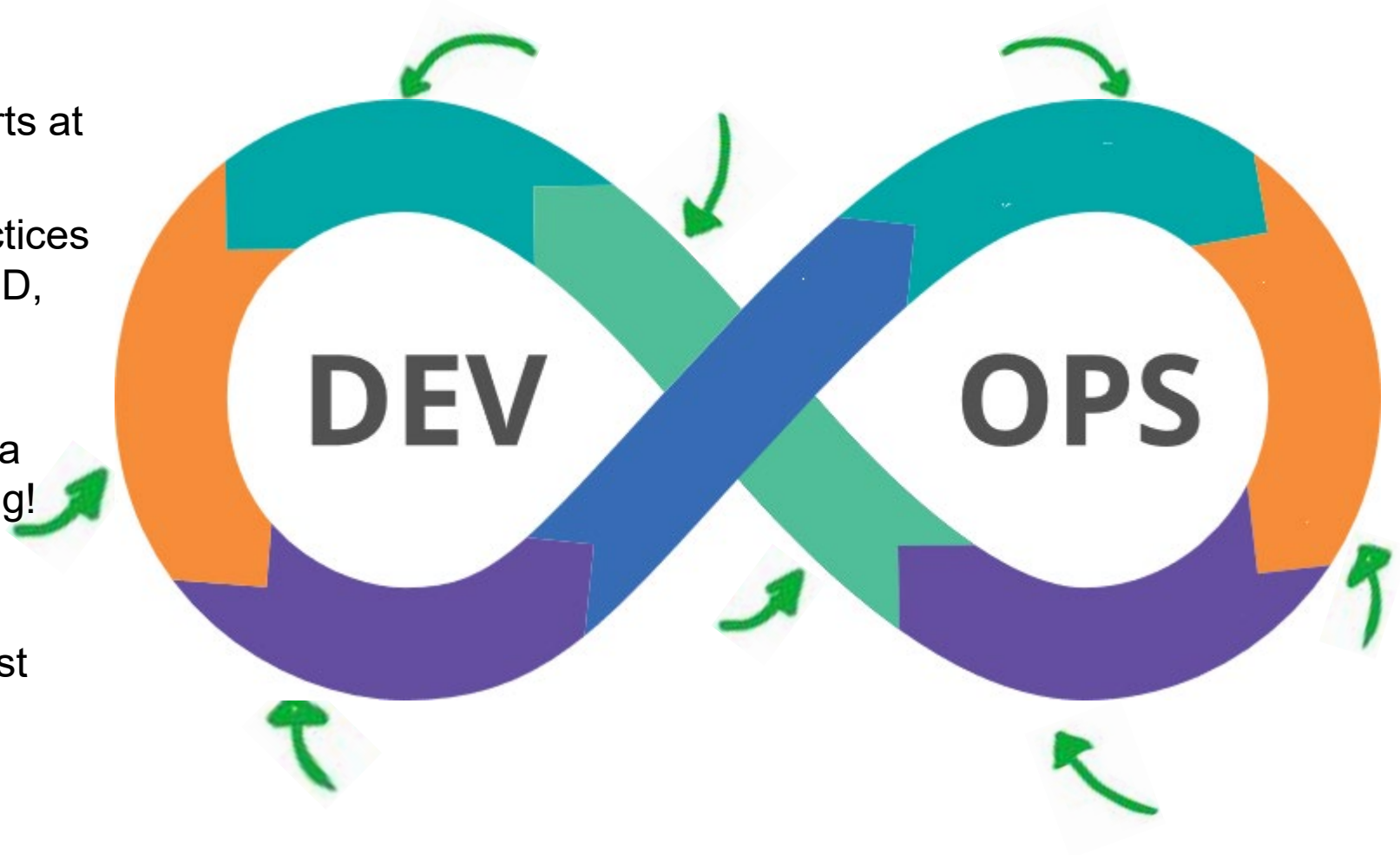
Can you spot the problem?



Base Image Courtesy of the United States Air Force[27]

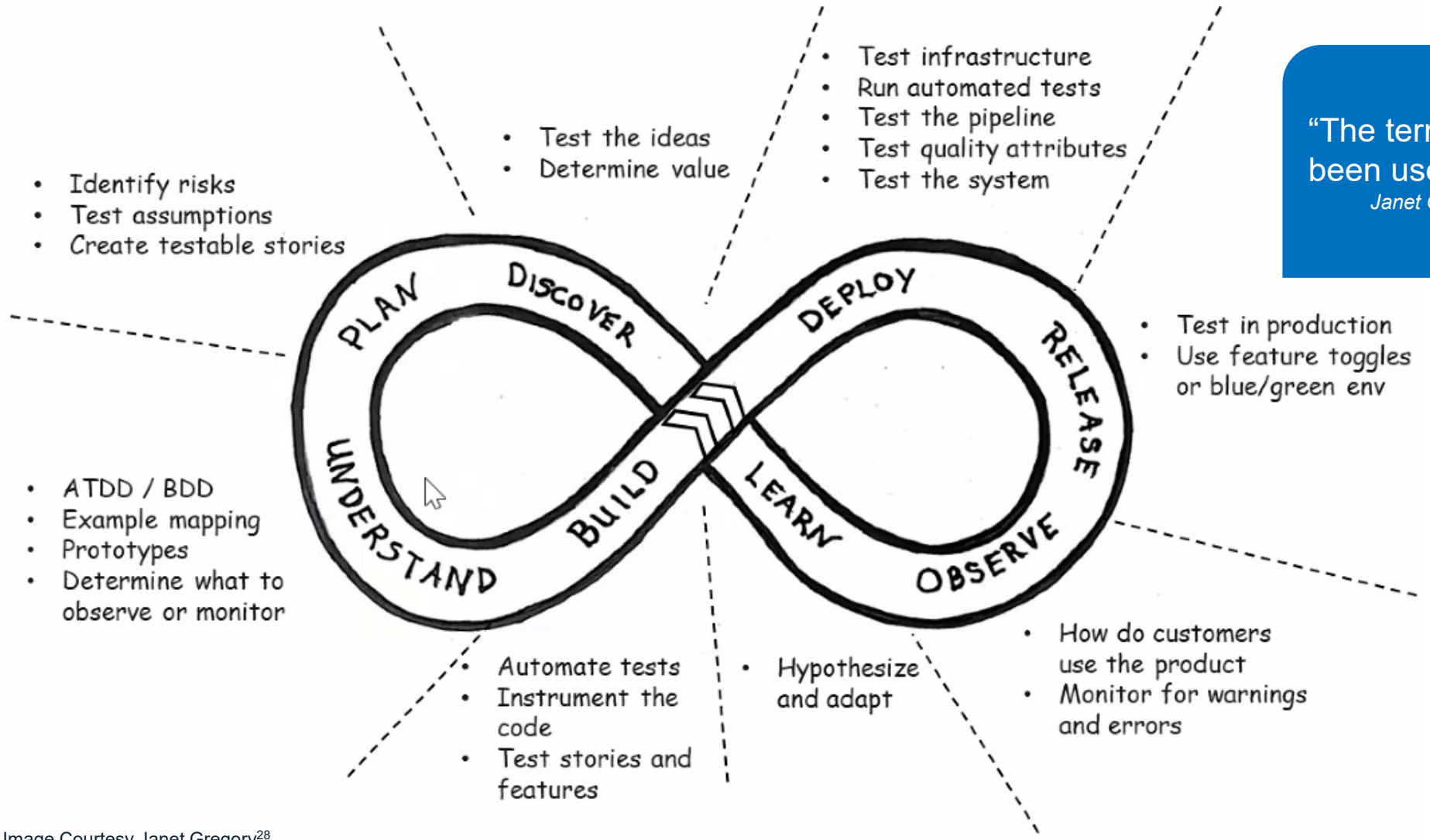
Not Just Left or Right. Be Holistic!

- Planning starts at design[7]
- DevOps practices like TDD, BDD, and CI
- Static code analysis[11] is a form of testing!
- Finding and remove redundant test cases



- Once you deploy, monitor and observe
- Test in production using feature toggles
- Learn from data about production usage and behavior[9]
- Emergence of TestOps[19]

Holistic Testing is More Than Automated Testing



"The term **continuous testing** has been used and abused."

Janet Gregory – Agile Testing Fellowship Founder

Image Courtesy Janet Gregory[28]

What Other Gaps Need to be Filled?

The more obvious and talked about reasons:

- Lack of test planning
- Skills and training gaps
- Fragility and tightly coupled implementation
- Lack of design[2]
- Overly fragmented test cases makes maintenance a nightmare

Failing tests cost the enterprise software market $61 billion

620 million developer hours a year wasted on debugging software failures[3]

Largely, we are unable to reproduce the defects…

Image Courtesy www.learnclick.com [29]

Why is Reproducing Defects Hard?

Do we have what we need?

- Decades of test engineering science on white box testing
- Continuous developer upskilling improving
- Direct access to stakeholders
- Tools, tools, tools

MITRE

Experience Story - 1

Storyline: Enterprise with multiple lines of business supported by a centralized shared service organization that included software quality testing.

Issues

- Continuous testing <> automated testing
- Exploratory testing was sequential
- Phantom defects
- Inconsistent testing results consistent
- Lack of test architecture knowledge

What we learned

- Our software needed to better address state management!
- Tests are too stateful
- Exercising code is not testing
- Test must be repeatable and the data must be reset
- Exploratory (Manual tests) should be async outside the pipeline

Finding and Reproducing Defects Requires Data

The research shows…

"Software failures in QA & test still detrimentally affect delivery speed."[3]

Challenges "integrating into the pipeline, including deliberate testing environments, and test data skills.[30]

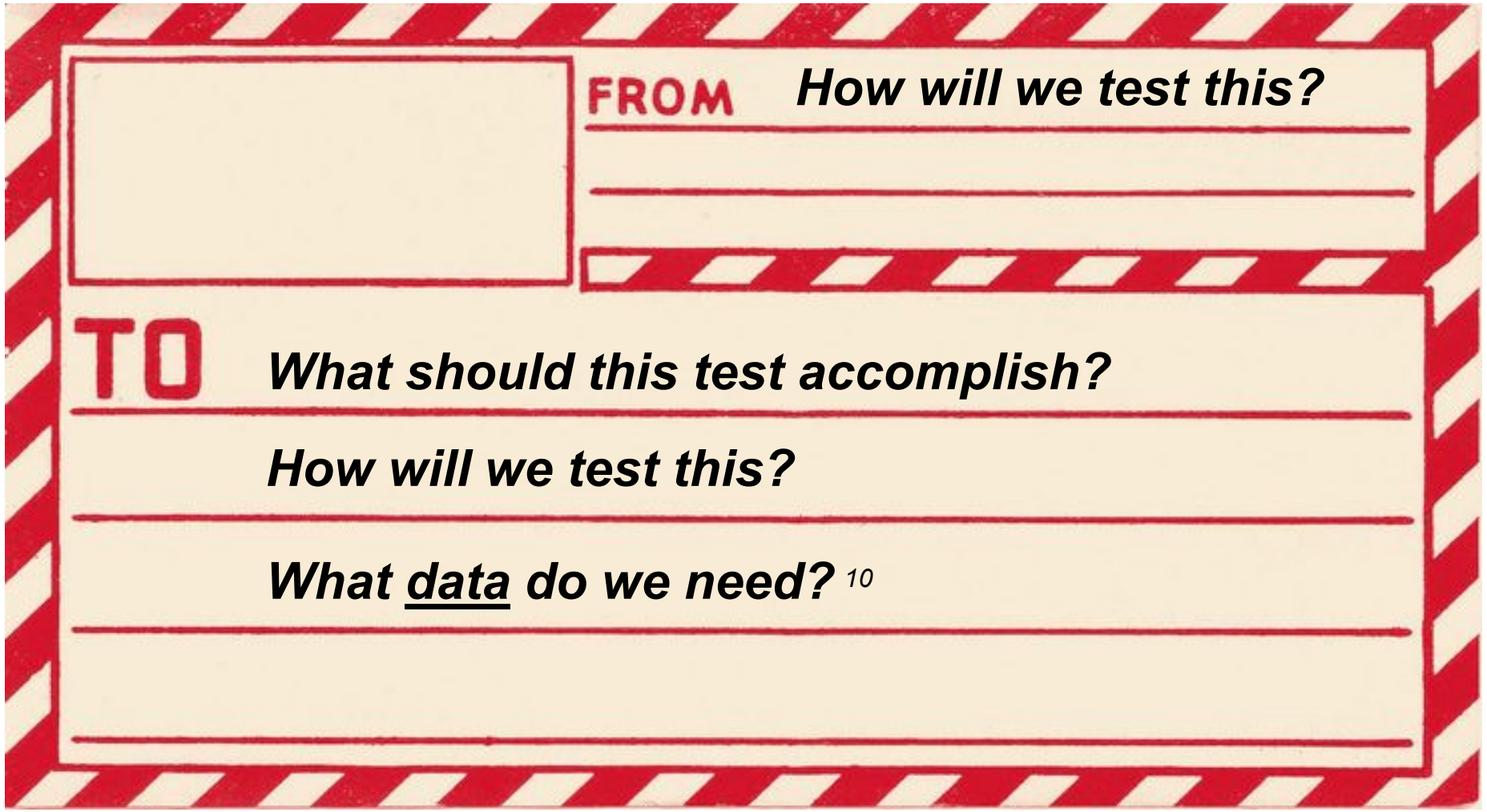
Straight talk:

When we shift, we are leaving test data and test data management behind

- To do any testing, early or late, test data is needed!
- Test data, however, is often an after-thought
- Delays are eating into any productivity gains with DevSecOps or agile investments[10,12]

MITRE

We Must Change the Narrative



Wanted: Data4DevOps

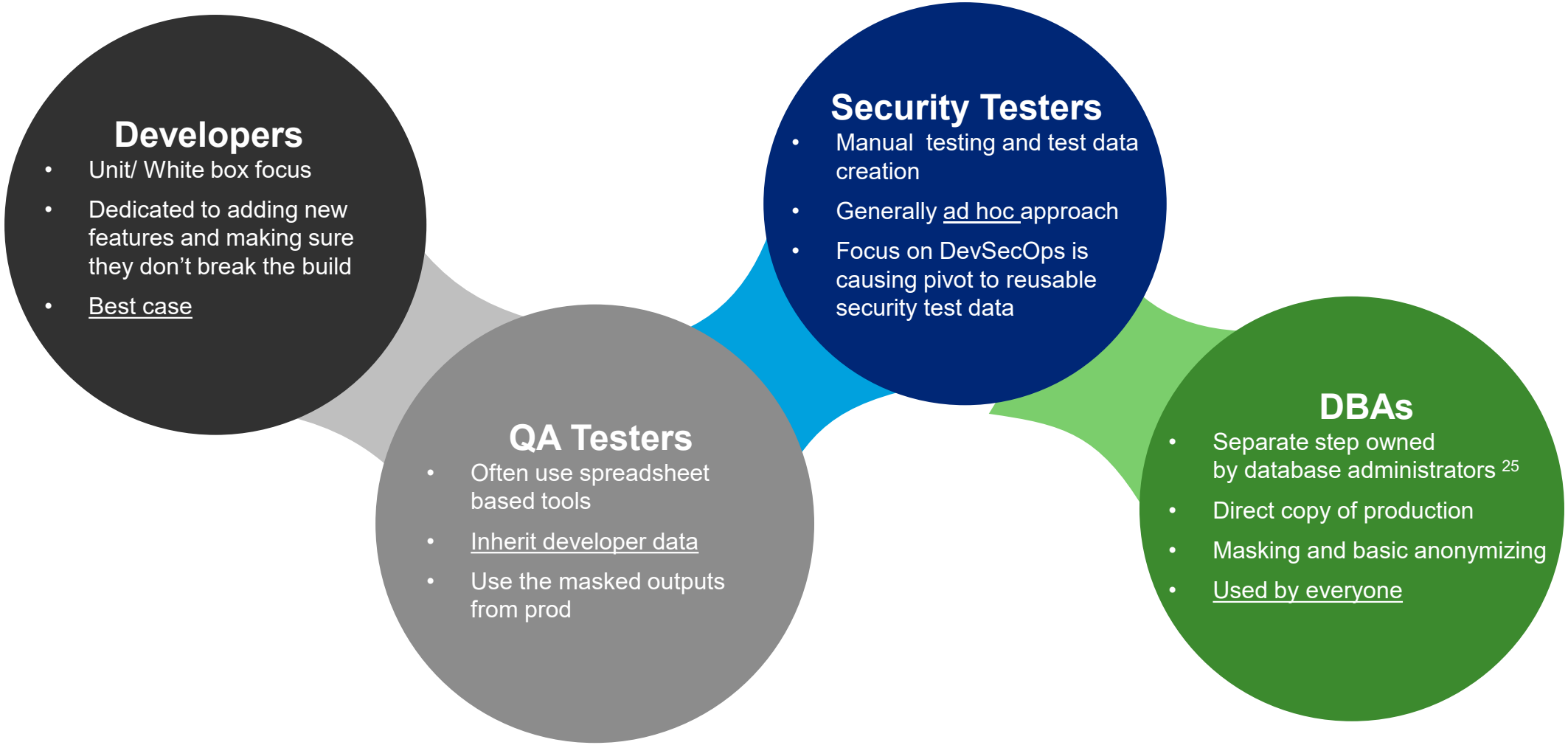
What Data Do We Need?

Other mandatory considerations

- Unique domain requirements
  - ✓ Aviation
  - ✓ Banking
  - ✓ Insurance
  - ✓ Healthcare
- Regulatory constraints and classifications
- Architecture[32]
  - ✓ design to requirements
  - ✓ focus on quality attributes
- Upstream and downstream integrations and external coordination
- Need for APIs and service virtualization

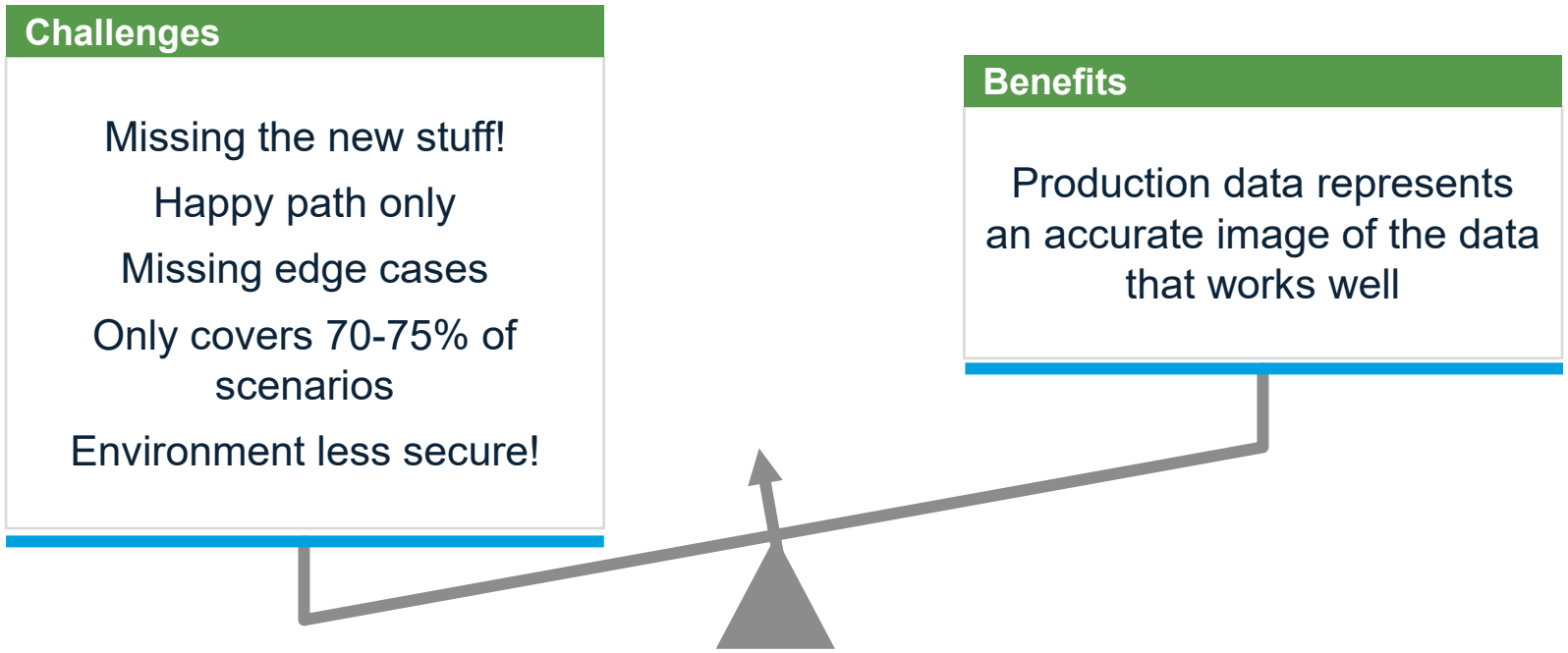
Who's Making the Test Data?

Generally, test data creation is still an afterthought with responsibility loosely defined



Pitfalls with Production Data

High quality testing demands high quality test data



65% of companies copy their production data to be used in testing
36% of companies apply masking techniques to protect test data[21]

MITRE

Options for Test Data

Using different approaches to generate test data provides a richer set

Build

- Manually
- Interact with the front end and explore advanced paths
- Synthetic generation

Internally Borrow

- Import from production
- Profile and mask production
- Duplicate from the prior system and transform

Go External or Buy

- Purchase from other corporations
- Web scrape from public sites and sources [21]
- Source from publicly available sources

Principles for Test Data Acquisition

- Explore the test data[14]
- De-identify
- Validate
- Build for re-usability
- Automate test data management tasks
- Generate data directly into a data store

Two of the Biggest Test Data Stumbling Blocks

Regardless as to approach, acquiring and **preserving** test data is the most common challenge

Loading/reloading the data into test environments is the second most common challenge

If you don’t have a repeatable initial state, it isn’t a valid test

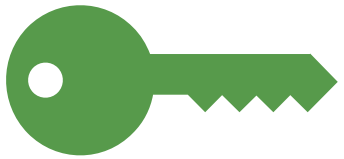
Image courtesy of https://frank-gieth.com/

MITRE

Test Data Challenges for DevOps

Like many DevOps challenges facing government and industry, the technology is not the solution

- No Test Data Strategy
- Manufacturing **all** the data
- Not using data subsets
- Writing your own masking tool
- Unprotected copies of production [36]
- Time consuming approaches
- Budgetary constraints
- Complex extraction
- Data sprawl and storage woes
- Lack of access to logs



The key to improving quality with DevOps is applying strong test data management

MITRE

Experience Story -2

Storyline: Large government agency with financial and banking-like policy that required decades of test data aligned to software versions and the related legal policy and laws

Issues

- Not having an agreed on approach
- Big bang approach
- Heavy lift in managing requests for data sets
- Attempting to automate **all** tests

What we learned

- Make risk based decisions on what tests to automate
- Profiling production to inform synthetic generation
- Validate test data by mapping back to requirements
- Make use of subsets
- Provide self-service including dataset reservations

Managing Test Data

"**Test data management** [TDM] is the creation of non-production **data** sets that reliably mimic an organization's actual **data**"

- Deliberate and controlled data access
- Maintaining test data accuracy
- Appending new data functionality
- Securing the data in non-prod
- Dynamic filtering to create data sets
- Storing filter criteria, data sets, and tagging to user story, test case, or code branch
- Data immutability[15]
- Dataset versioning
- Data refresh on demand
- Storing final state of data/database as reference
- Deleting all test data including databases, logs, and other files

MITRE

TDM Platform Concepts

TDM Platform selection is like any other architectural tradeoff decision

Here are a few criteria to consider:

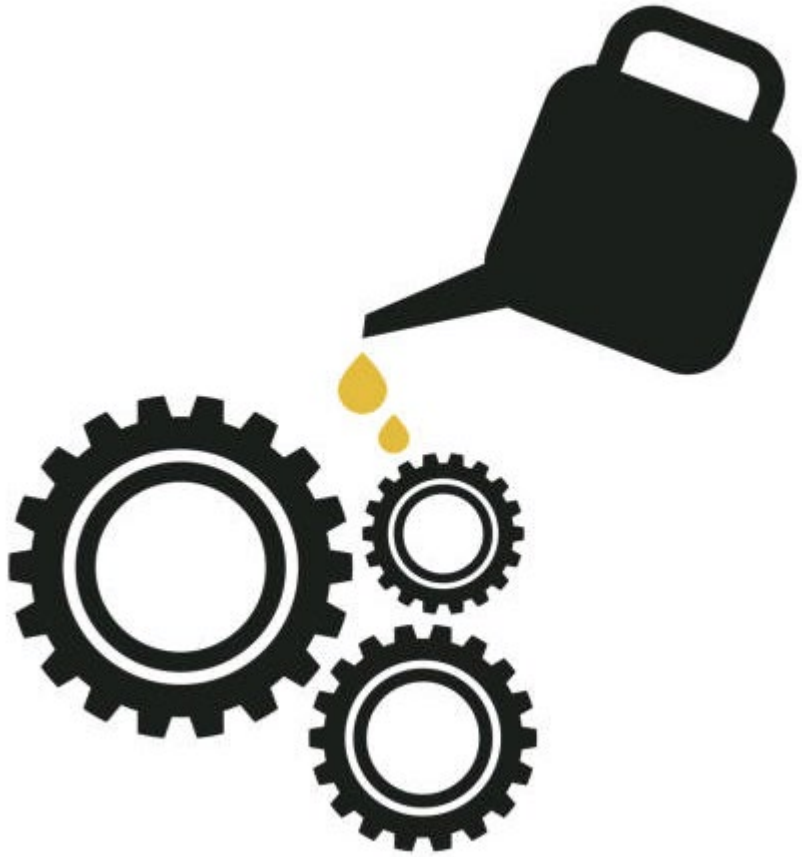
- Data Source Diversity
- Data Cloning and Versioning
- Sensative Data Masking and Encryption
- Security and Access Control
- Data Generation
- Test Data Warehouse
- Open APIs for integration
- Self Service Features

TDM Platform Concepts – The Eyechart

Integrating TDM with Pipelines

Overcoming data friction supports the quality imperative.

- Version datasets alongside code and configurations
- Reduce unnecessary copies but enable self service
- Integrate through APIs
- Leverage virtualization
- Allow saving of data after tests complete
- Ensure clean-up happens
- Remember logs are data too
- Consider adding performance & load testing in the pipeline



Provision @ the Speed of Need

Image Credit: Getty Images; Creator: Ratsanai[40]

What Should Not be Automated?

- Documentation
- Anti-automation (like CAPTCHA)
- Infrequent, low risk tests
- Exploratory / Ad-hoc testing

Example Test Data Management Tools/Platforms

Many platforms and mature offerings to consider:

DELPHIX

Informatica

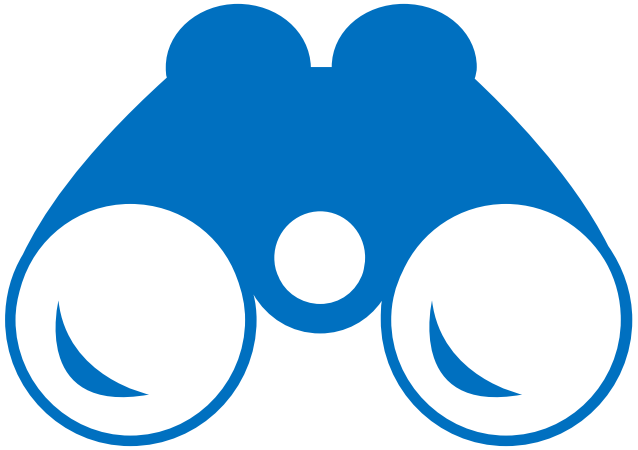
BROADCOM

- https://www.broadcom.com/products/software/continuous-testing/test-data-manager
- https://www.informatica.com/products/data-security/test-data-management.html
- https://www.delphix.com/delphix-data-platform
- https://www.k2view.com/products/test-data-management/

These are industry tools provided as samples only. Mention of a product or service should not be construed as an endorsement.

Looking Ahead

People, process, and technology on the horizon:



- Test Environment Management (TEM)
- Digital twin/digital engineering
- Improved testing with API mocking and Service Virtualization
- Data Virtualization
- AI Driven Testing using AI-driven test engines[25]
- Leveraging Big Data
- Testing AI/ML systems including ModelOps

Tracy L. Bannon
TBannon@MITRE.org
TracyBannon@gmail.com
https://www.linkedin.com/in/tracylbannon
@TracyBannon

#StraightTalk #RealTechnologists #WomenInCloud #DevSecOps #CloudNative

Disclaimer: The views, opinions and/or findings contained in this report are those of The MITRE Corporation and should not be construed as an official government position, policy, or decision, unless designated by other documentation.

MITRE | SOLVING PROBLEMS FOR A SAFER WORLD™

References - 1

1 https://www.contino.io/insights/5-challenges-to-devops-adoption-and-how-to-overcome-them

[2] https://dzone.com/articles/automated-testing-fail

[3] The Business Value of Optimizing CI Pipelines

[4] Chhillar, Dheeraj, and Kalpana Sharma. "Proposed T-Model to Cover 4S Quality Metrics Based on Empirical Study of Root Cause of Software Failures." International Journal of Electrical and Computer Engineering 9.2 (2019): 1122-30. ProQuest. 8 July 2021 .

[5]Crispin, Lisa, and Janet Gregory. *Agile Testing*. Addison-Wesley, 2009.

12 https://lisacrispin.com/2011/11/08/using-the-agile-testing-quadrants/

[6] https://www.guru99.com/data-driven-testing.html

[7] Testim. “Why Shift-Left Testing? Pros and Cons.” *AI-Driven E2E Automation with Code-like Flexibility for Your Most Resilient Tests*, 13 Feb. 2020, www.testim.io/blog/shift-left-testing.

[8] https://danashby.co.uk/2016/10/19/continuous-testing-in-devops/

9 https://www.mabl.com/blog/shift-left-shift-right-shifting-and-why

10 https://dzone.com/articles/shift-test-data-to-the-left-too

11 https://dzone.com/articles/what-is-shift-left-testing

12 https://undo.io/the-cost-of-software-failures/

13 Naik, Kshirasagar, and Priyadarshi Tripathy. *Software Testing and Quality Assurance: Theory and Practice*. 1st ed., Hoboken, New Jersey, John Wiley & Sons, 2008.

14 https://blog.vsoftconsulting.com/blog/significance-of-test-data-management-in-improving-products-test-quality

15 https://www.delphix.com/delphix-data-platform

MITRE

References - 2

16 https://www.informatica.com/products/data-security/test-data-management.html

17 https://www.informatica.com/content/dam/informatica-com/en/collateral/data-sheet/informatica-test-data-management_data-sheet_3234en.pdf

18 https://www.broadcom.com/products/software/continuous-testing/test-data-manager

19 https://devops.com/shift-right-testing-the-emergence-of-testops/

20 https://searchsoftwarequality.techtarget.com/definition/shift-right-testing

34. https://www.testim.io/blog/test-data-is-critical-how-to-best-generate-manage-and-use-it/

16 https://fusionalliance.com/testing-in-the-devops-pipeline/

17 https://fusionalliance.com/software-testing-trends/

18 https://info.eggplantsoftware.com/hubfs/Product%20Marketing%20PDFs/DevOps%20Playbook.pdf

19 https://www.delphix.com/blog/overcoming-data-friction-to-achieve-devops-success

20 https://www.k2view.com/hubfs/Whitepaper%20-The%20DevOps%20Test%20Data%20Problem.pdf

21 https://software.af.mil/training/devops/

22 https://janetgregory.ca/testing-from-a-holistic-point-of-view/

23 https://www.learnclick.com/

24 “Growing Expectations from Quality Assurance. How Can You Meet Them?” Capgemini Worldwide, 5 Nov. 2020, www.capgemini.com/research/world-quality-report-wqr-20-21.

25 Clements, P. , 2011: Improving Testing Outcomes Through Software Architecture . Carnegie Mellon University's Software Engineering Institute Blog,. http://insights.sei.cmu.edu/blog/improving-testing-outcomes-through-software-architecture/ (Accessed July 13, 2021)

26 https://www.linkedin.com/pulse/8-reasons-why-software-architecture-important-ahad-khan-csm/

27 https://blog.vsoftconsulting.com/blog/significance-of-test-data-management-in-improving-products-test-quality

MITRE

References - 3

34. https://www.linkedin.com/pulse/8-reasons-why-software-architecture-important-ahad-khan-csm/
35. https://www.capgemini.com/resources/world-quality-report-20-21-north-america/
36. https://dzone.com/articles/the-5-most-common-test-data-pitfalls-for-devops
37. https://www.informatica.com/services-and-training/glossary-of-terms/test-data-management-definition.html
38. https://www.softwebsolutions.com/resources/devops-automation-strategy.html
39. https://www.pinclipart.com/
40. *Image Credit: Getty Images; Creator: Ratsanai*

MITRE