

# Five Keys to Agile Test Automation for Government Programs

Robert Binder and Suzanne Miller

Software Engineering Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213

# Notices

Copyright 2017 Carnegie Mellon University. All Rights Reserved.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM17-0395

# Overview



- 1) How is testing different in Agile software development?**
- 2) What kind of testing should/should not be automated for Agile software development?**
- 3) Who should develop automated tests?**
- 4) What kind of tool chain do I need to support automated testing?**
- 5) What are some ways that DoD acquisition programs can successfully adopt Agile/Automated testing?**



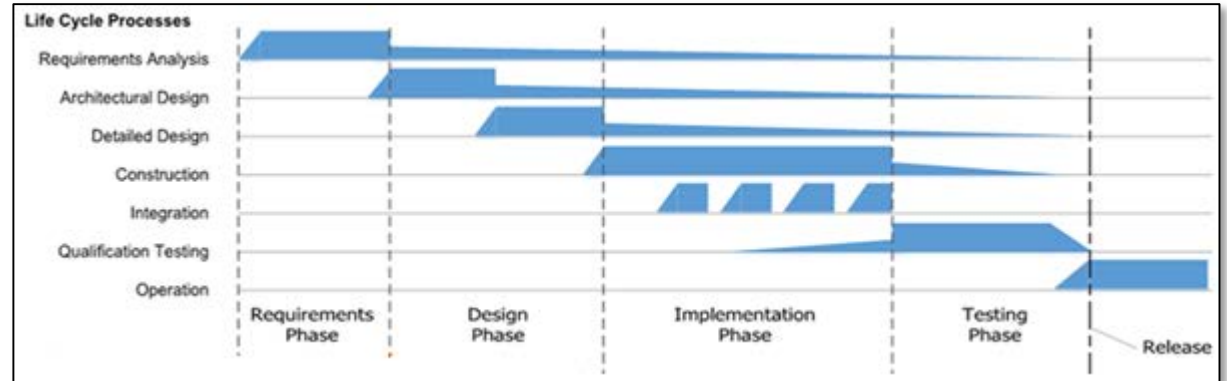
Five keys to effective Agile test automation for Government programs

# How is testing different in Agile software development?

# Testing is development, development is testing

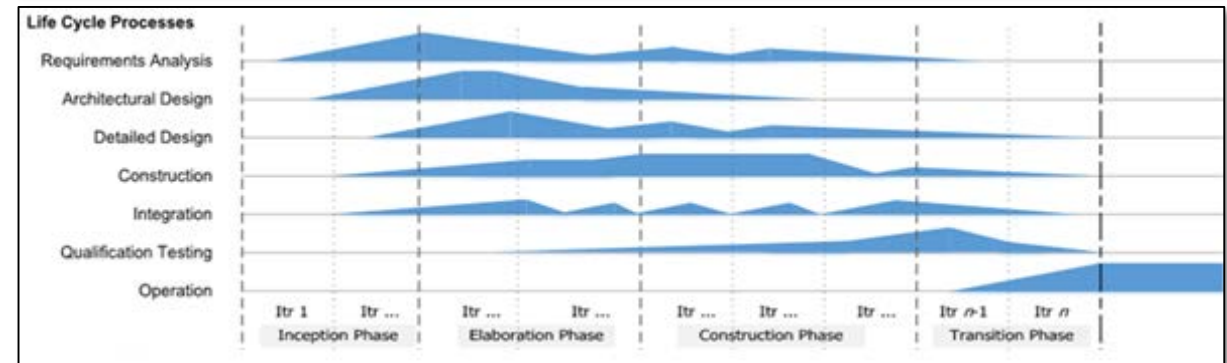
## Phased

- Design, Code, Test, Test, Test
- Release



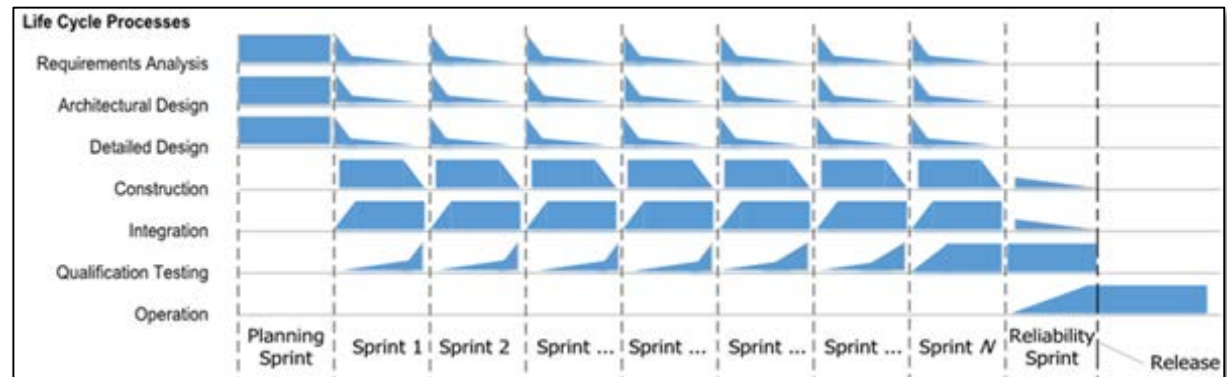
## Incremental

- Design a little, code a little, test a little...
- Test, Test, Test
- Release



## Agile

- Test, fail, code, test, pass ...
- Test, fail, code, test, pass ...
- Test, fail, code, test, pass ...
- Release



Source: IEEE Standard 1633P, *Recommended Practices for Software Reliability Engineering*

# Early and repeatable testing is Agile key to quality

## Agile Quality Practices

- Voice of the customer
- Commitment management
- Definition of Done
- Demos
- Retrospectives
- **Test Driven Development**
- Exploratory Testing
- Living Tests
- Find and fix within sprint

## Test Driven Development

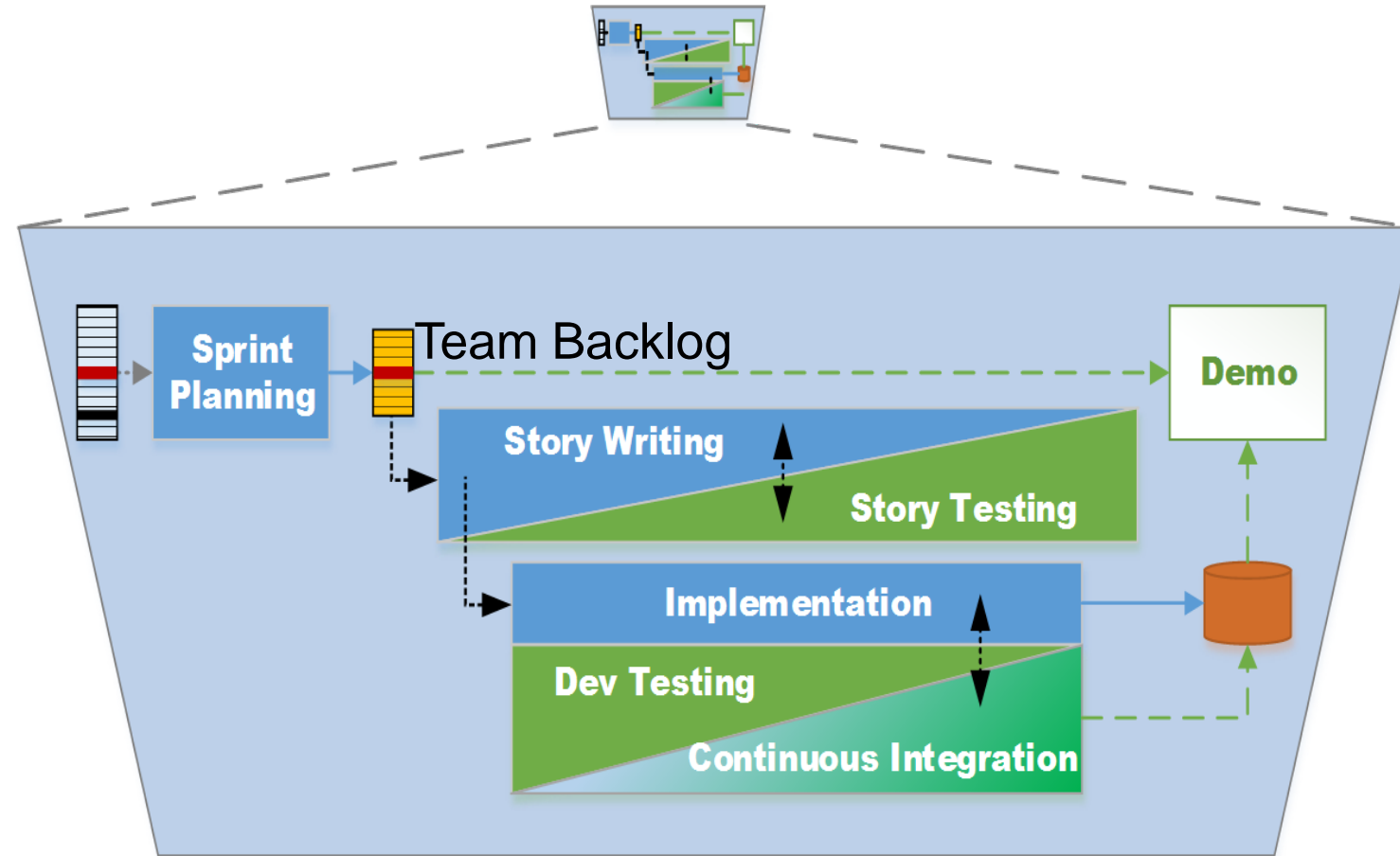
- TDD – Test Driven
- BDD – Behavior Driven
- ATDD – Acceptance Test Driven

## Typical Agile Testing Tool Chain

- Component/API testing: Junit, Nunit...
- BDD/ATDD: Cucumber, SpecFlow...
- GUI testing: Selenium, Ranorex...
- Continuous Integration: Jenkins, Hudson ...
- DevOps: Chef, Docker, Puppet...

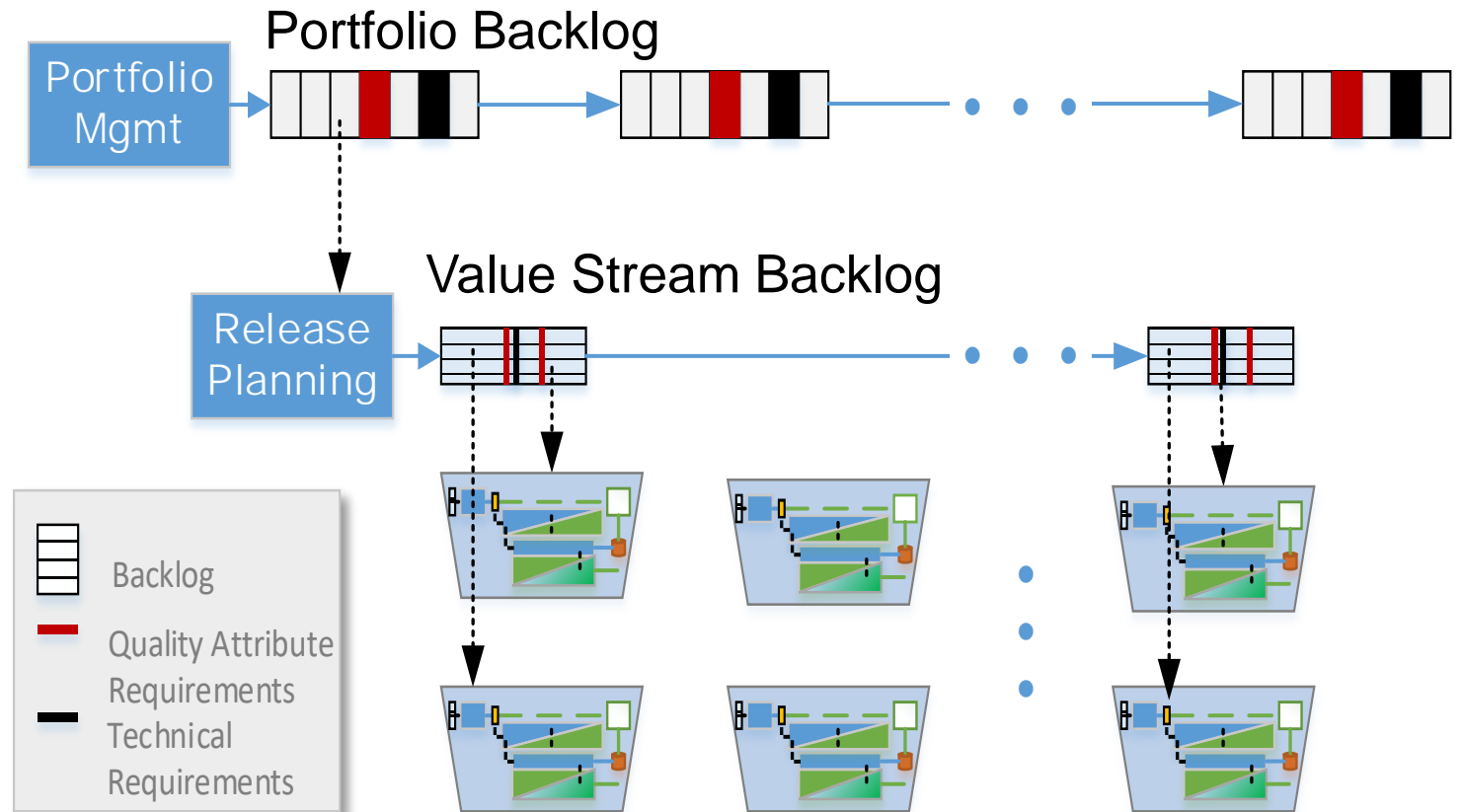
# Agile “Inter-twingles” Development and Test

- Development and test are not separate or standalone
- Tests are added to a repeatable test suite
- Test suite is repeated for each change
- Demos *not* a replacement for testing



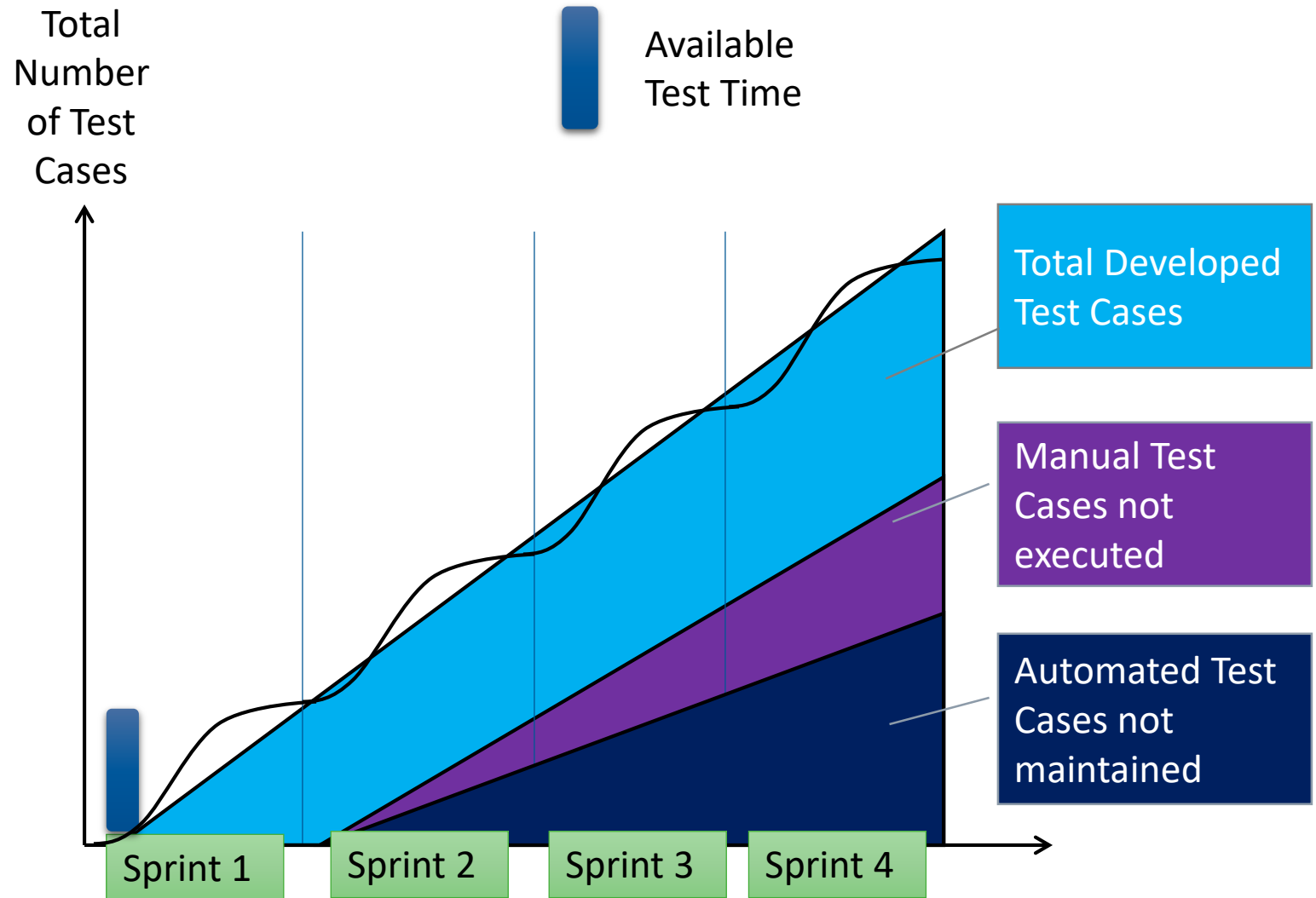
# Agile iteration relies on more testing earlier

- Testing used for rapid refinement of loosely understood requirements, architecture, and design
- Requirements and implementation are finely sliced
- Each slice is tested immediately and repetitively



# No Automation? *The Backblob is going to get you!*

- The extent of manual testing is limited to the capacity of testers
- The extent of automated testing is limited to the capacity of test scripters
- Total number of tests increases as project progresses
- Typically, only the newest features are tested



# DoD Acquisition requirements are unique

DoD requires large systems to plan for and undergo independent Operational Test & Evaluation

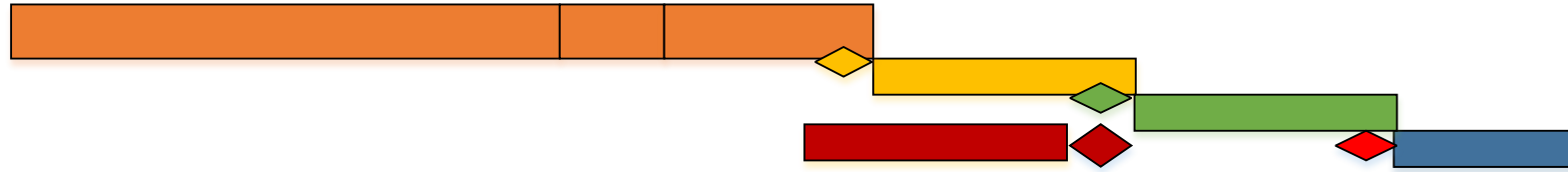
- Planned and executed by a different organization than the Program Office acquiring the software system
- Requires a very early Test and Evaluation Strategy more compatible with a “big bang” delivery than the incremental delivery typical in Agile

USAF guidance(AF99-103) now aligns independent testing with a more incremental approach

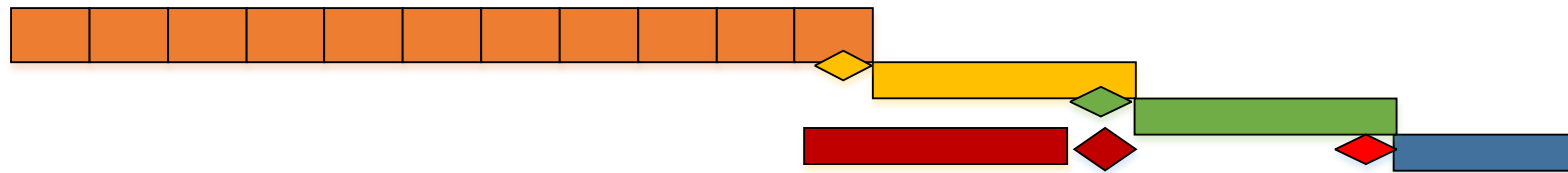
- Integrated testing and integrated test teams are a specific strategy called out
- Incremental testing is specifically discussed and encouraged prior to full operational testing of a deployed capability

# Left-shift with Agile Testing

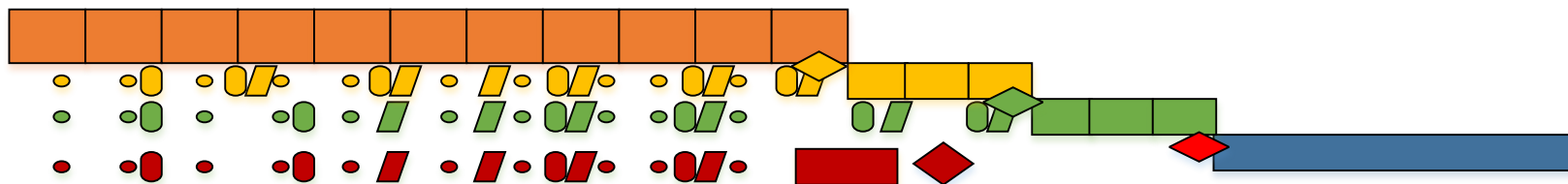
Traditional Vee-process



Agile development with traditional DT and OT (Hybrid)



Agile development with traditional DT and OT, early integration synch points



Legend:  OEM Dev     Developmental Testing (DT)     Cyber     Operational Testing (OT)     Deployment

Moving from phased and siloed testing to Agile testing is the "Big Deal"

Integrating Agile cadence with DT/OT is a key challenge

# Poll 1



*A: What software life cycle do you use?*

- A. Phased, “waterfall”
- B. Incremental
- C. Evolutionary, “Agile”
- D. Hybrid

*B: Is a different life cycle:*

- A. Being evaluated
- B. Being piloted
- C. Active rollout
- D. No change anticipated



Five keys to effective Agile test automation for Government programs

# Who should develop automated tests?

# Some factors to consider

## SDET skill set versus functional tester skill set

- Writing automated tests IS writing software
- Not all functional testers come from a software development background

## Organic or Outsource

- For some types of testing, there are consulting groups that will convert manual to automated tests
- Works best with well understood systems with well-documented tests and sufficient subject matter experts to answer the myriad questions that will come up

# OEM or government engineering or test staff?

OEMs with active Agile practices typically use local automated testing for component testing and local integration testing

- Contracting to obtain those tests as part of software delivery increases the test base for regression testing
  - Be sure to ask for the test environment as well, so the OEM tests can be run by others!

DT&E (Development Test & Engineering) government staff can interact with Agile OEMs via Iteration, System, and Release Demos and the activities that lead up to them

- Leveraging automated tests that come out of early activities can build confidence in the accumulation of evidence related to the system's robustness

# Poll 2



*Who is responsible for automating tests in your project?*

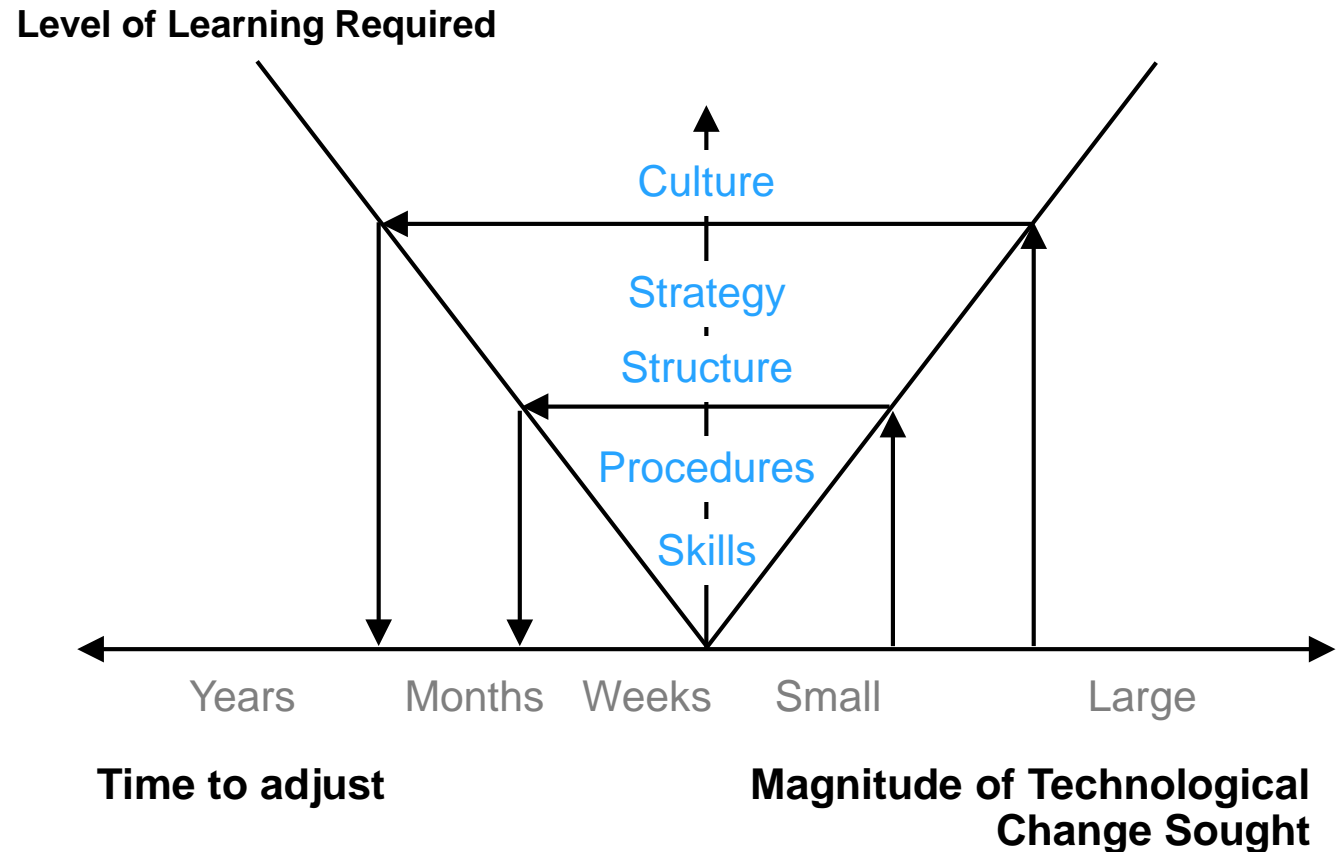
- A. We don't automate
- B. Non-technical testers or QA staff
- C. Application developers
- D. Test engineers
- E. Software development engineer in test (SDET)

Five keys to effective Agile test automation for Government programs

# How can my program successfully adopt Agile/Automated testing?

# Automated Testing = Structure and Strategy Change

- Scope of change is similar to automating a manual business or operational process
- Requires many behavioral changes
- Not plug-and-play
- Learning curve

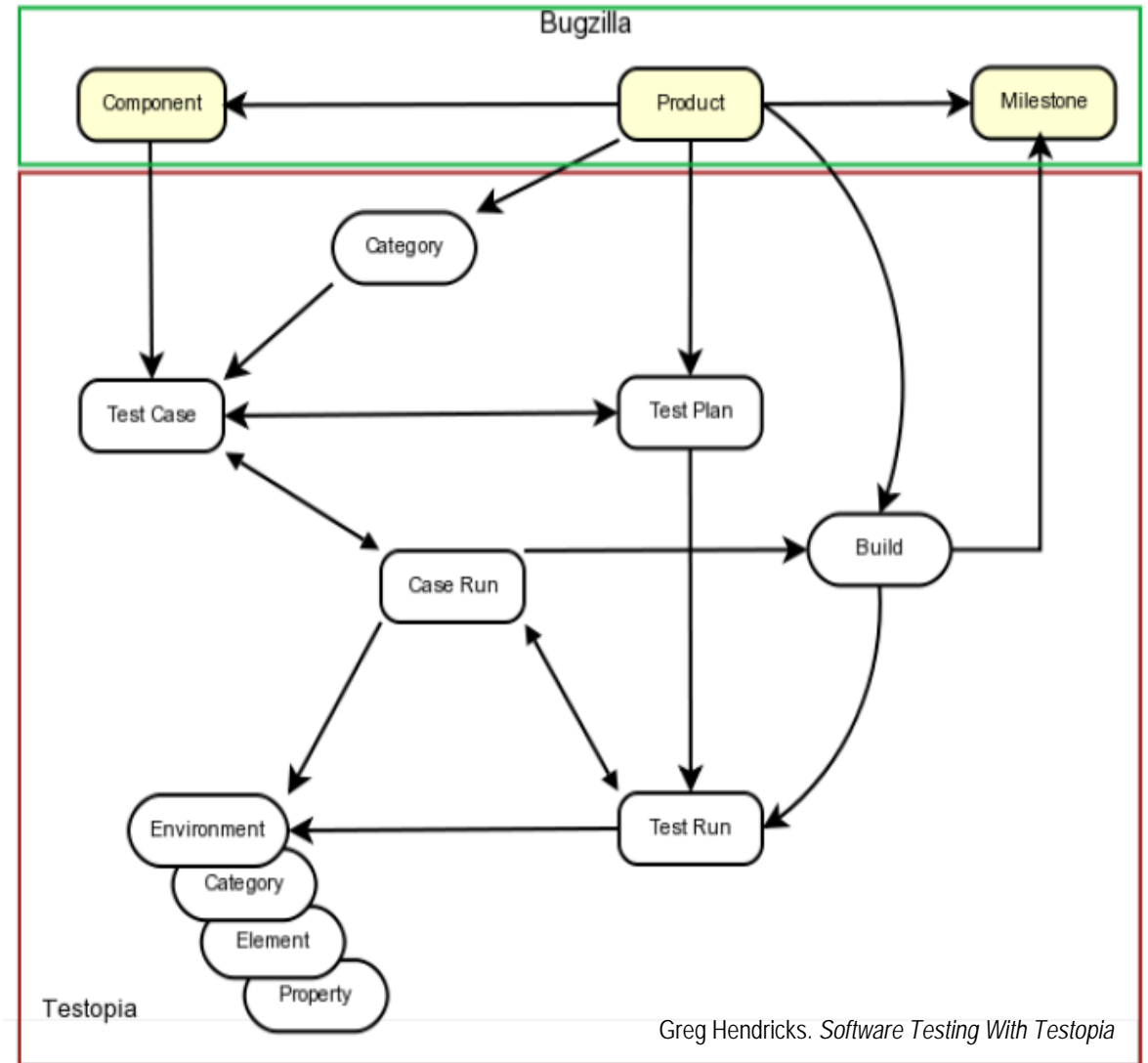


# Ensuring Test Automation Success

- Treat development and operationalization of your test automation system like any other critical operational system
  - Agile development of test infrastructure & assets
  - Rollout support: deconfliction, training, socialization, funding of maintenance
- Grow or hire Software Development Engineers in Test (SDETs)
- The payoff of automation is repeatability and consistency
- Automation doesn't eliminate the need for manual testing (and testers)
- Training and support for tools is critical

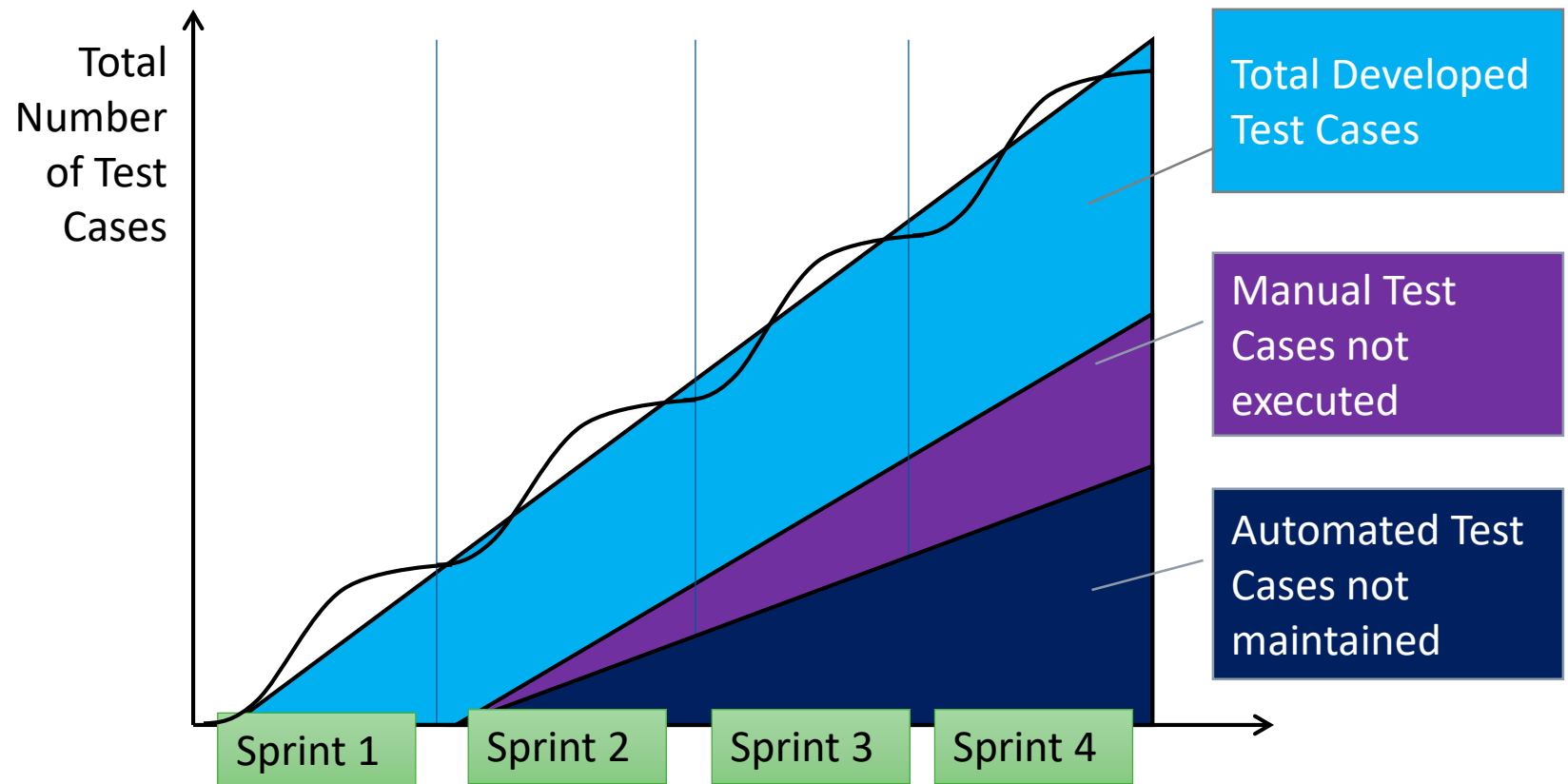
# Program-Wide Test Asset Management System

- Team-wide repository, test meta-data
- Tool chain interfaces
  - Test harness(es)
  - Bug tracking
  - Requirements
  - Version control
  - Continuous integration
- Part of all leading Application Lifecycle Management (ALM) systems
- Several open source systems
- *Can track status of **all** test activity*



# REMEMBER: No Automation? *The Backblob's gonna getcha!*

- The extent of manual testing is limited to the capacity of testers
- The extent of automated testing is limited to the capacity of test scripters
- Total number of tests increases as project progresses
- Typically, only the newest features are tested





Five keys to effective Agile test automation for Government programs

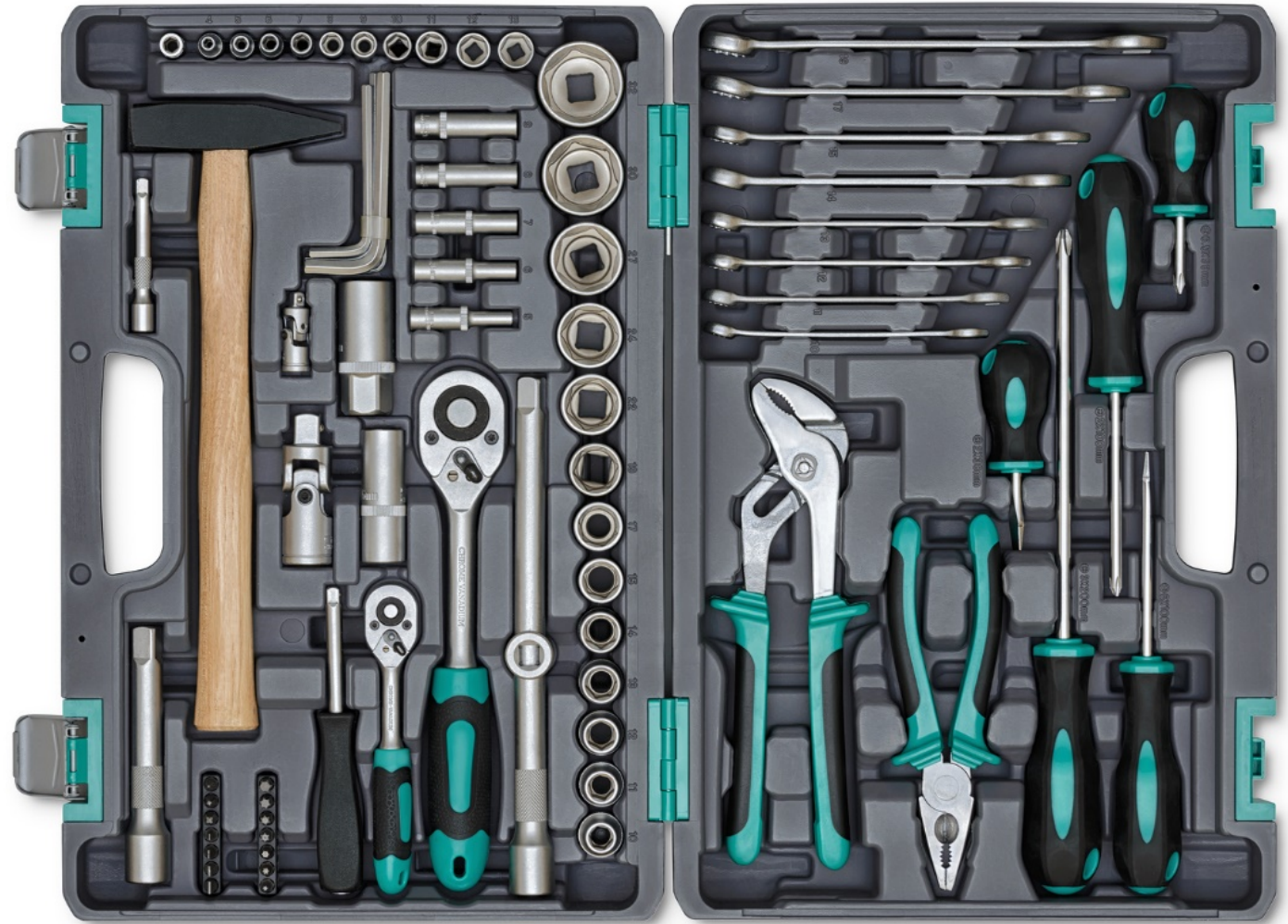
# What kind of tool chain do I need to support automated testing?

# Testing scope, lanes, focus, tooling

	<i>Scope</i>	OEM/ Dev	PO FAT	DTO	OTO	<i>Focus</i>	<i>Example Tool</i>
Unit, CSCI		✓				Functions	Junit SonarCube
Component, Subsystem		✓	✓			Features, User stories	Selenium SOAP UI
System, SoS			✓	✓	✓	Use cases, Performance Mission threads,	Jmeter

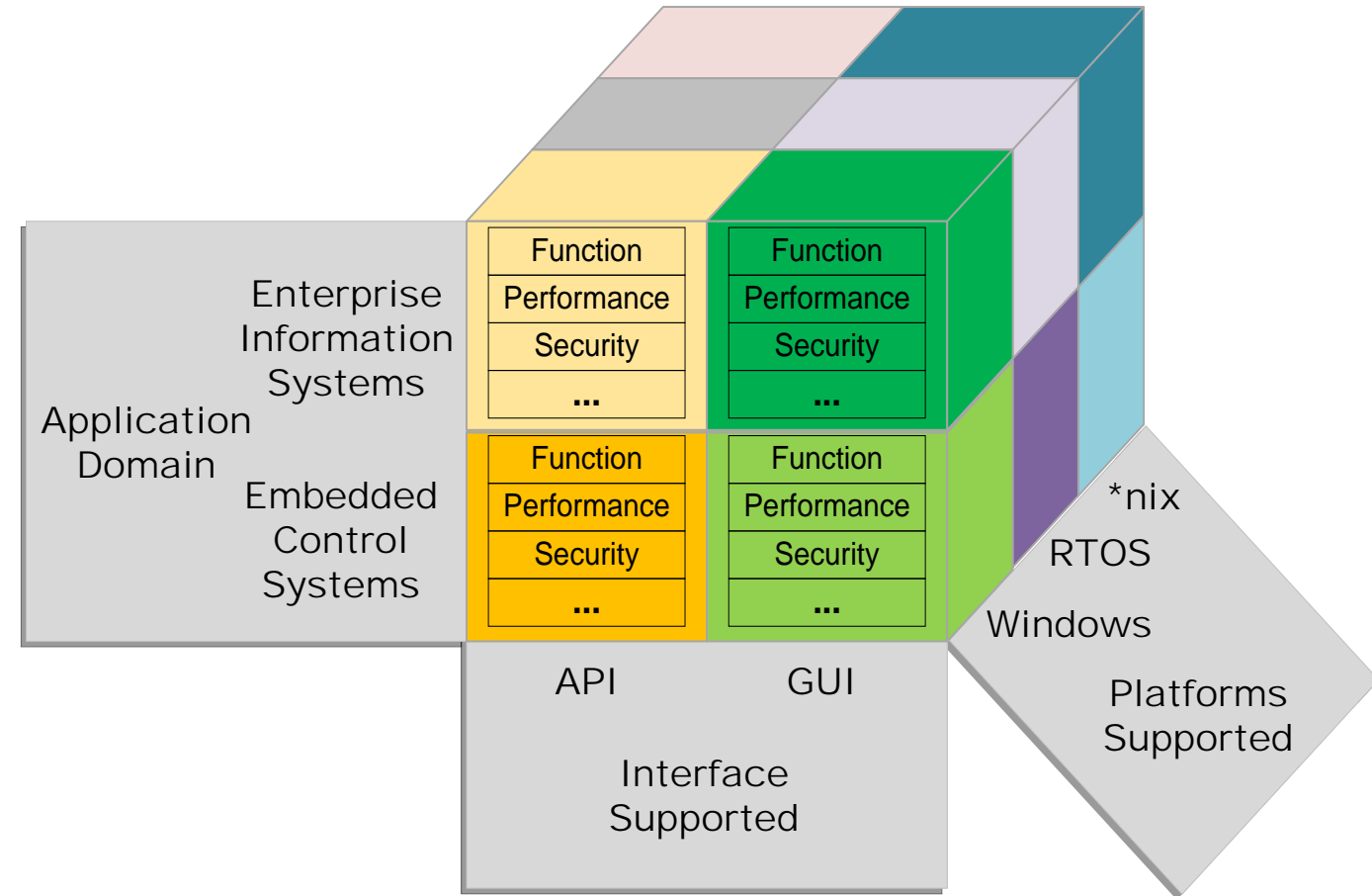
# Tools for Test Automation

- There are *hundreds* of COTS, FOSS, and GOTS software testing tools

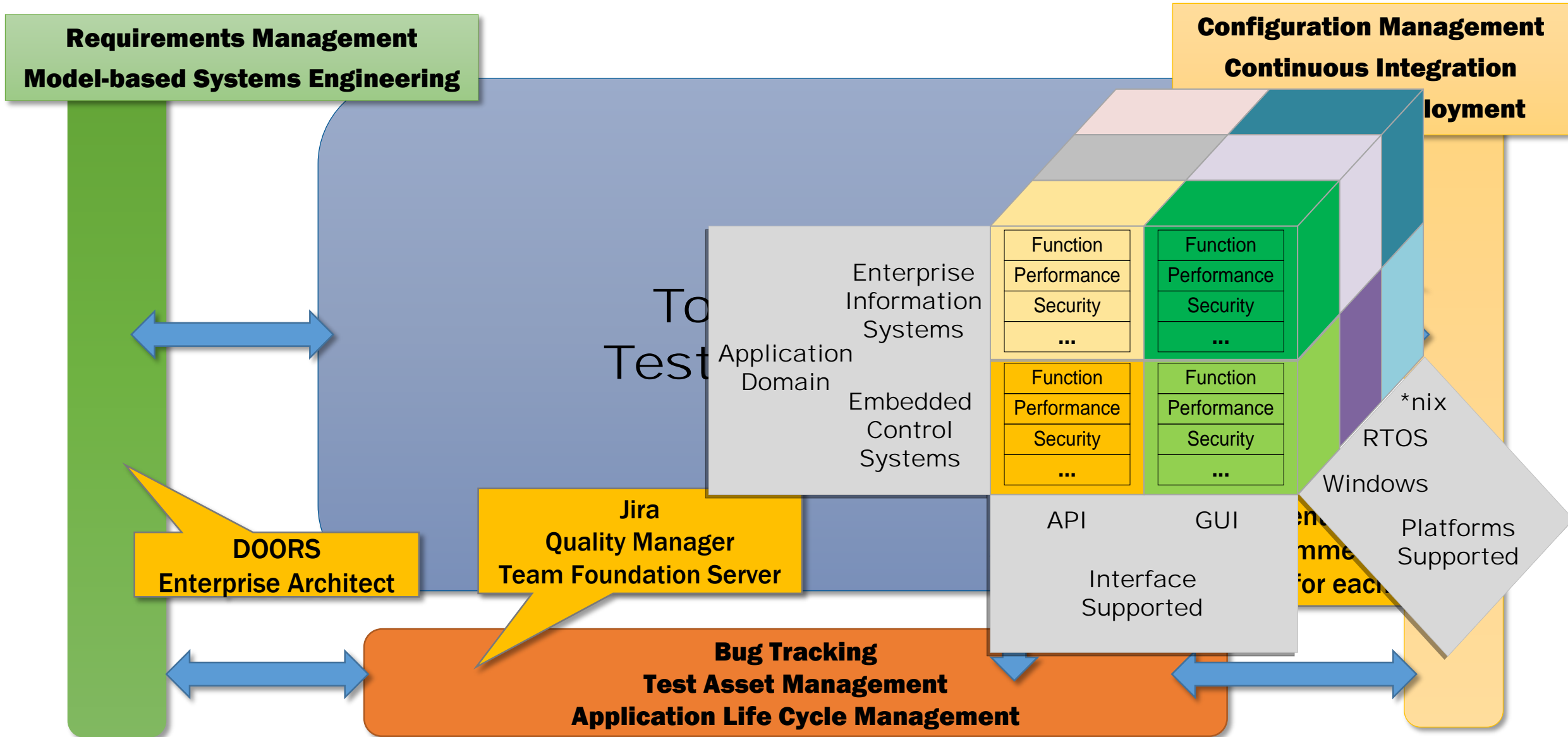


# Tools for Test Automation

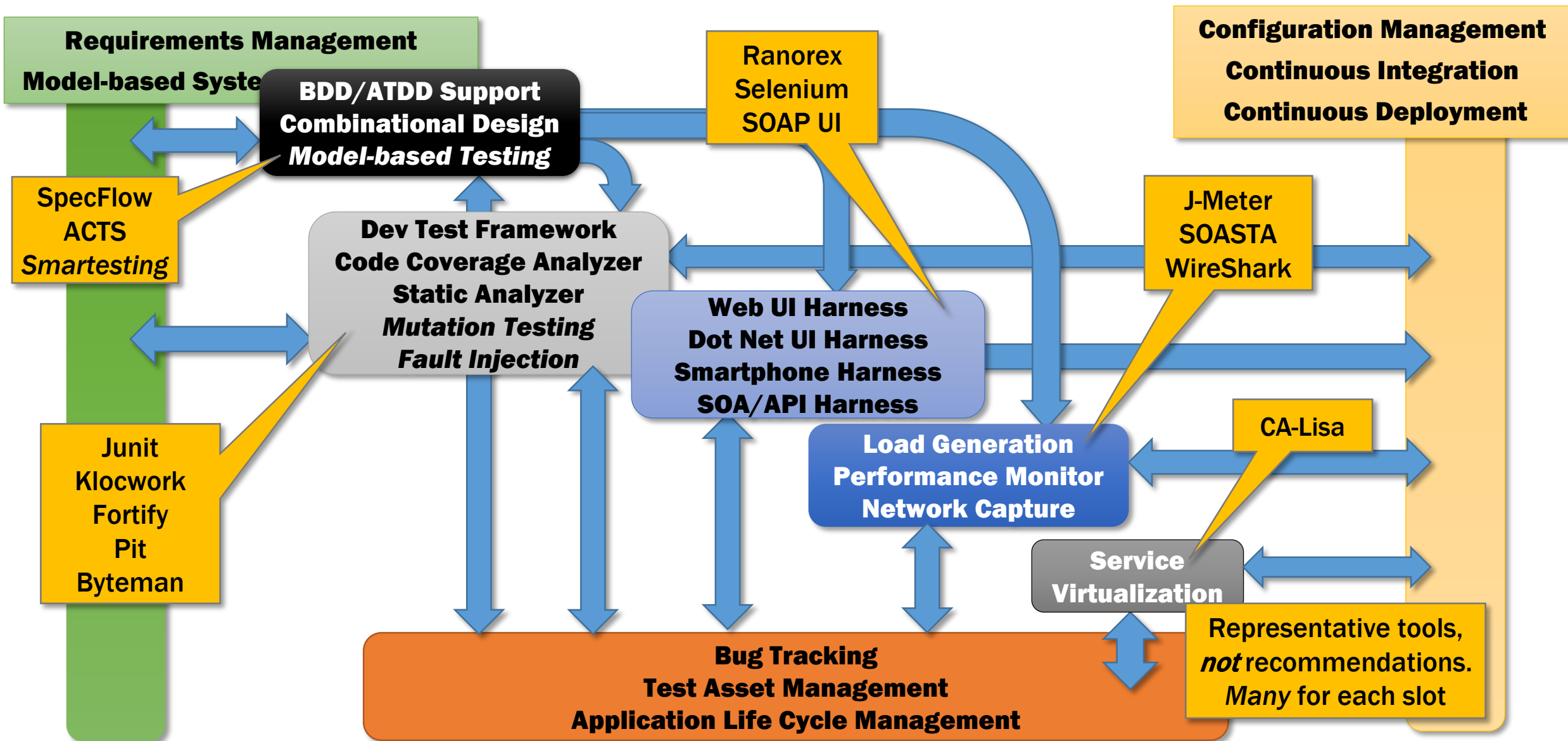
- There are *hundreds* of COTS, FOSS, and GOTS software testing tools
- Each tool is specialized for a certain kind of testing
- Each tool is specialized for a tool stack, target stack, and target interface



# Test Automation Reference Architecture



# Test Automation Reference Architecture



# Poll 3



*What kinds of testing are automated in your project?*

- A. Unit/CSCI testing
- B. User interface functional testing (desktop, mobile, web page)
- C. APIs, web services
- D. Software/Hardware in-the-loop testing
- E. Performance testing
- F. Robustness testing
- G. Other



Five keys to effective Agile test automation for Government programs

# What kind of testing should/should not be automated for Agile software development?

# Testing scope, focus, automation

Notional proportion of automated and manual test cases

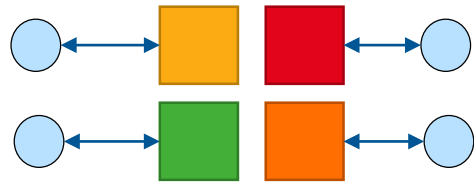
*Scope*

*Focus*

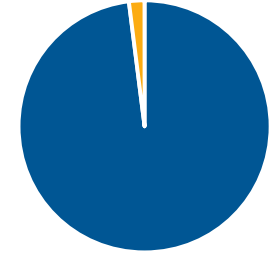
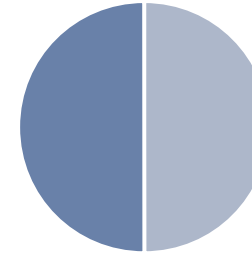
*User Interface*

*SW Interface*

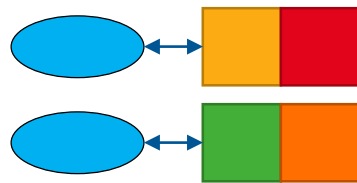
Unit,  
CSCI



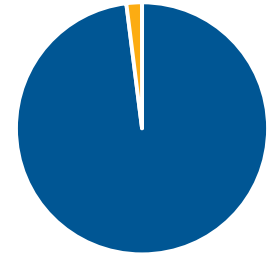
Functions



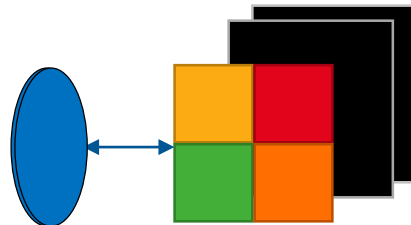
Component,  
Subsystem



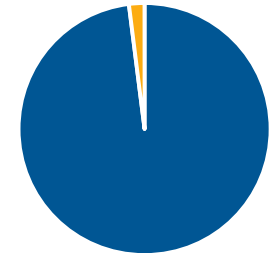
Features,  
User stories



System,  
SoS



Use cases,  
Mission threads,  
Performance



# Avoid test automation gotchas

- Much more work converting manual to automated testing than expected
- Don't automate tests for unstable interfaces
- Do use exploratory testing
- Do automate tests for APIs, stable user interfaces
- Tools don't automate test design and judgement
- Capture/replay usually results in breakage and/or test script re-recording
- Set expectations for test asset maintenance
- Automate performance testing incrementally
- Follow test automation design patterns

## Poll 4



*What kinds of testing are routinely done in your project?*

- A. All manual
- B. Mostly manual, some automated
- C. About same manual and automated
- D. Mostly automated, some manual
- E. All automated



Five keys to effective Agile test automation for Government programs

# Summary

# No Magic!

*When done skillfully, test automation can deliver strong benefits to support organizations in delivering high-quality software at a faster pace.*

*When done badly, it creates more problems than it solves.*

Sobejana and Herschmann, "The Eight Essentials When Moving to Automated Software Testing," Gartner Inc., 2016.

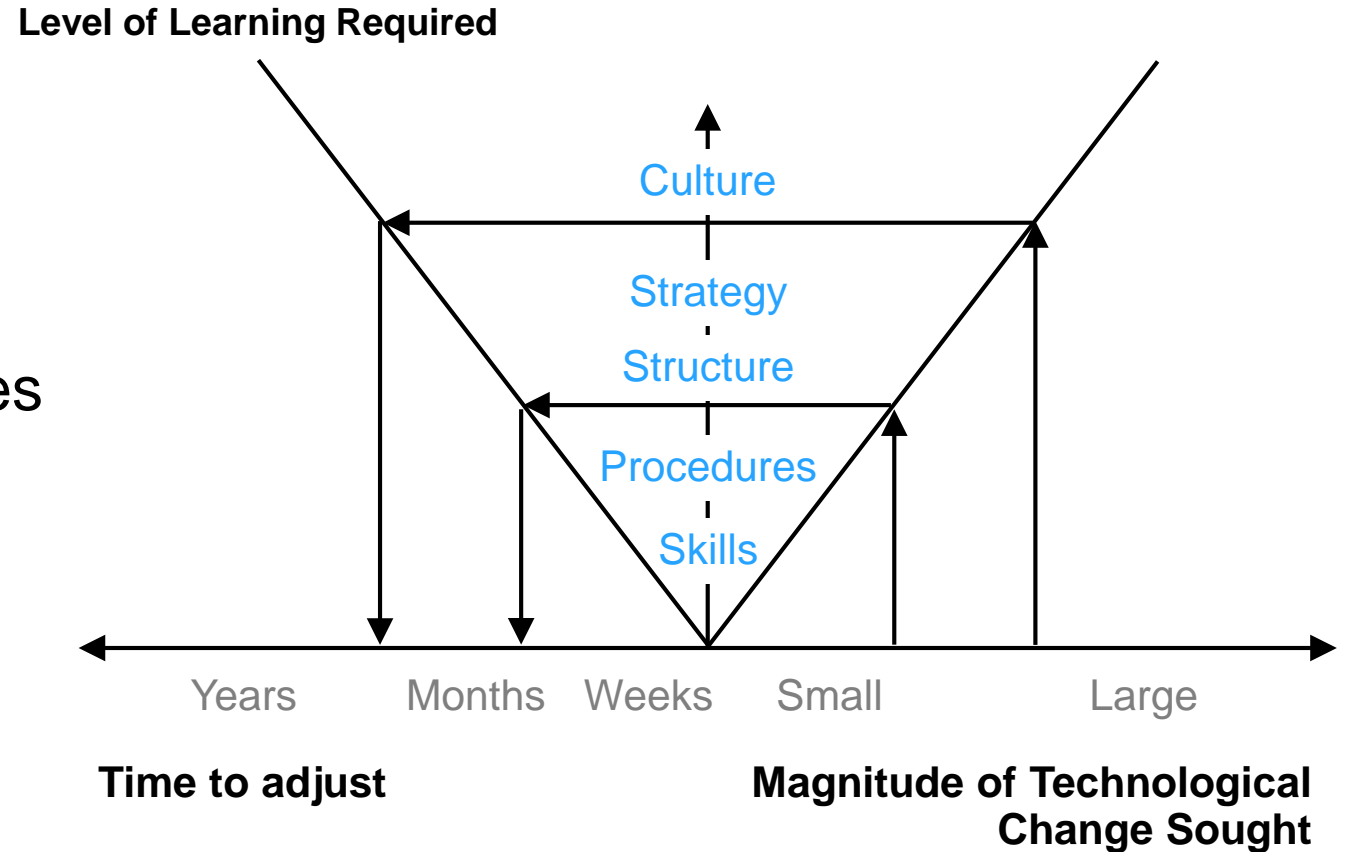


# Reasons Automated Test Investment is Often Delayed in Agile Adoption

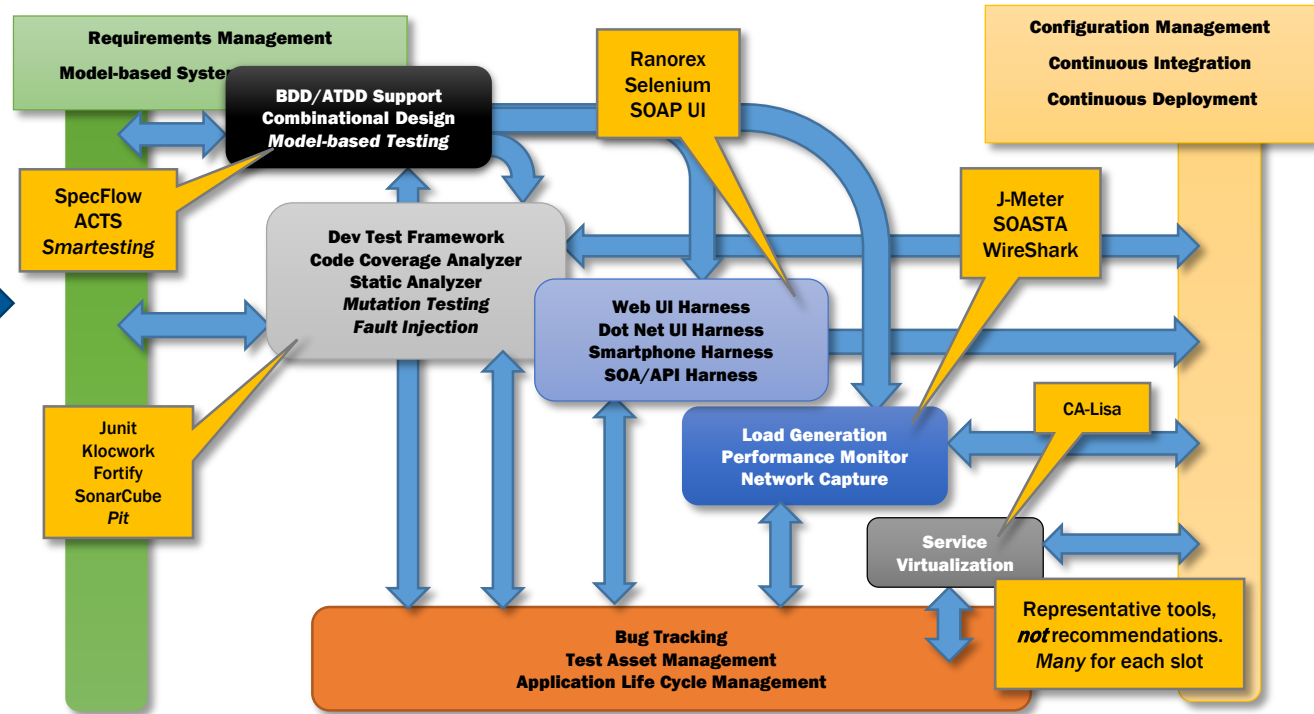
- Testers aren't included in initial adoption strategy
- Difficulties in obtaining required tooling to support
- Some can be bought, but some has to be built from scratch
- Training test personnel in tooling/approach while executing their normal activities
- Hard to pause ongoing work to establish a new capability going vs build it piecemeal
- Organizational challenges related to dev/test structures
- Using automated mechanisms to do things that used to require human governance
- This isn't just "throwing a switch"
- Evolutionary implementation means incremental utility, so monetary benefit won't be seen right away, even though risk reduction might be

# Remember: Automated Testing = Structure and Strategy Change

- Scope of change is similar to automating a manual business or operational process
- Requires many behavioral changes
- Not plug-and-play
- Learning curve



# Start early, keep at It!



Modernizing legacy testing is just like modernizing legacy operations!



## Five keys to effective Agile test automation for Government programs

# Q & A