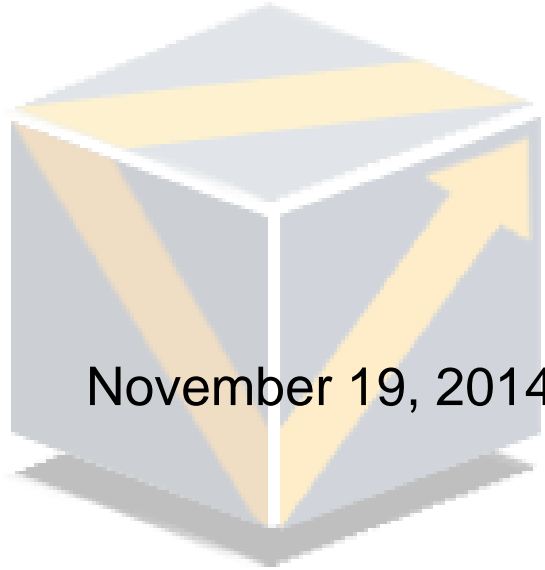


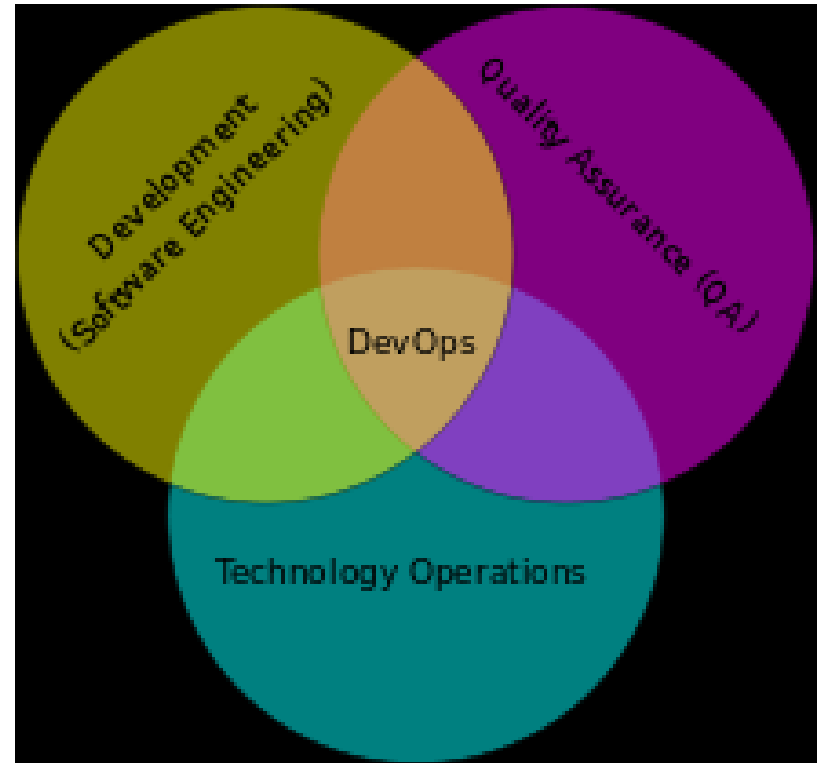


Test Ops Overview



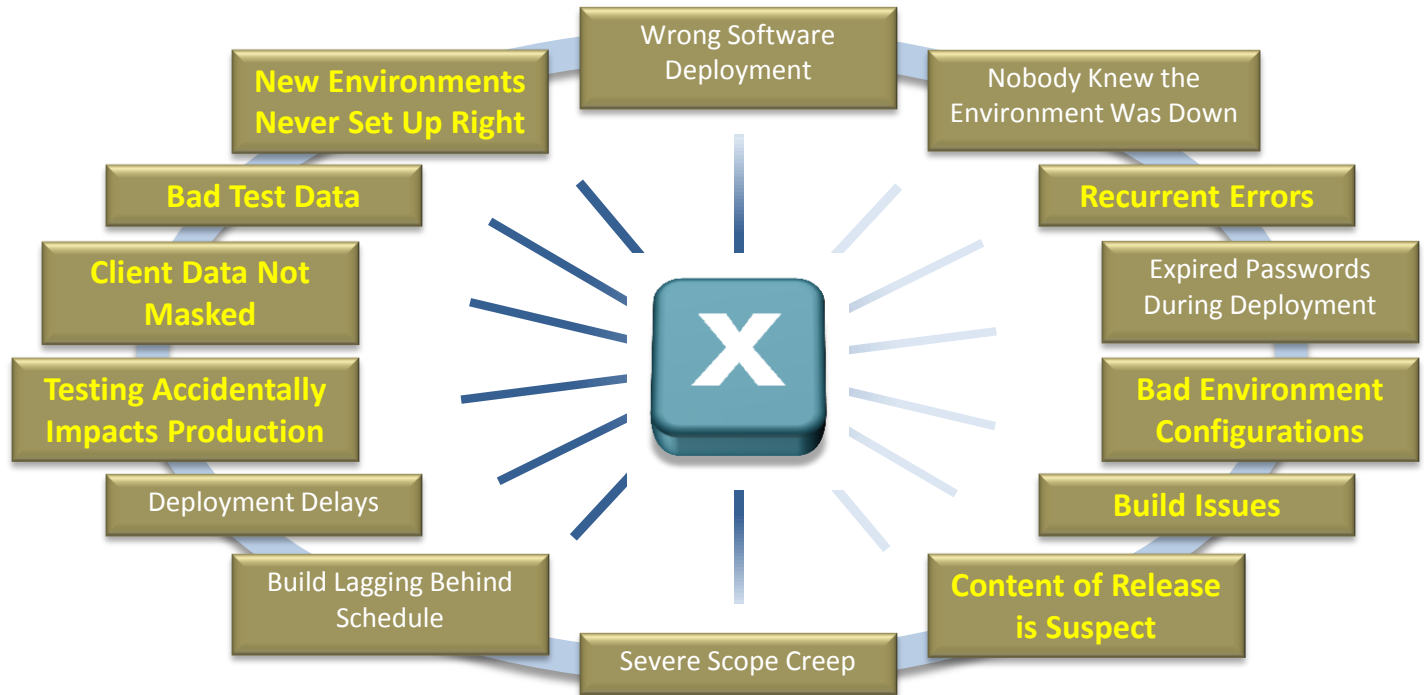
DevOps (a portmanteau of “development” and “operations”) is a software development method that stresses communication, collaboration and integration between software developers and Information Technology (IT) professionals. DevOps is a response to the interdependence of software development and IT operations. It aims to help an organization rapidly produce software products and services.

<http://en.wikipedia.org/wiki/DevOps>



Do any of these problems look familiar?

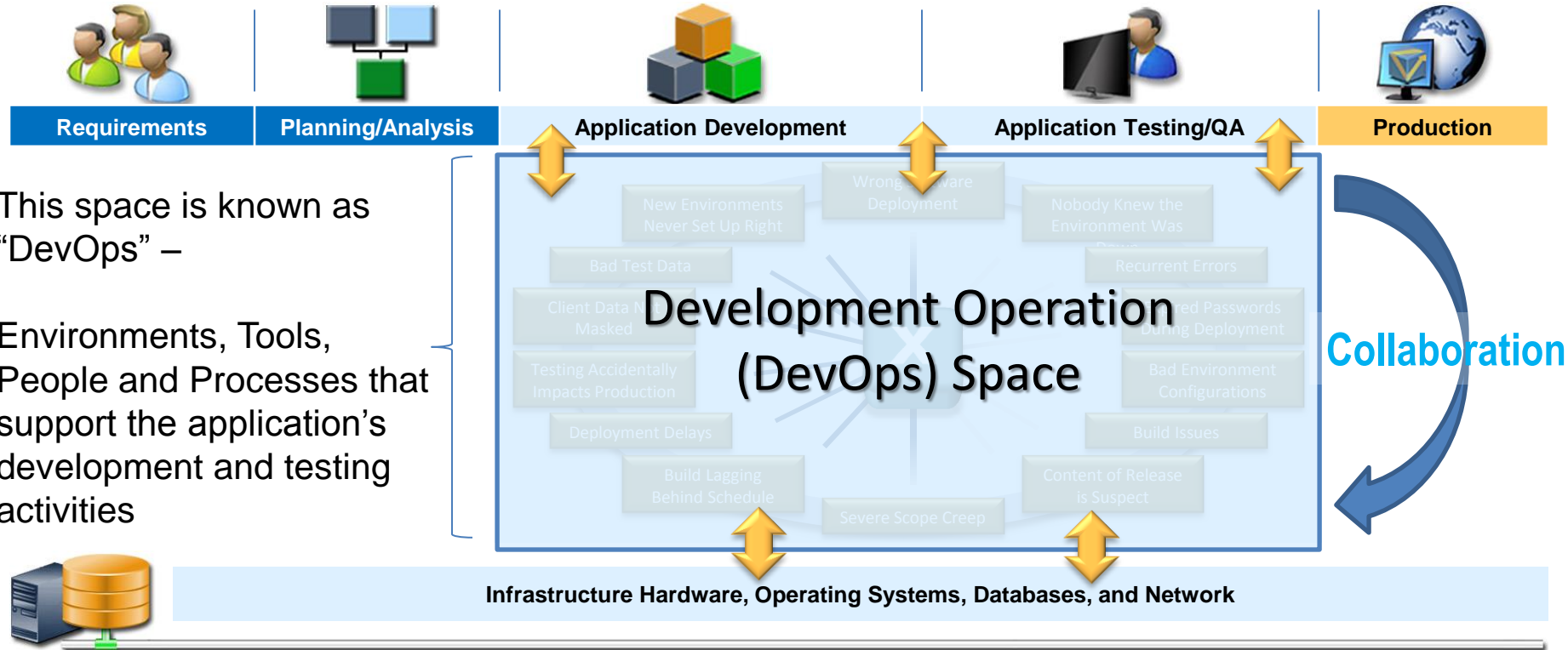
- How many are QA related?
- How many impact testing activities?





What is DevOps, really?

Application Life Cycle



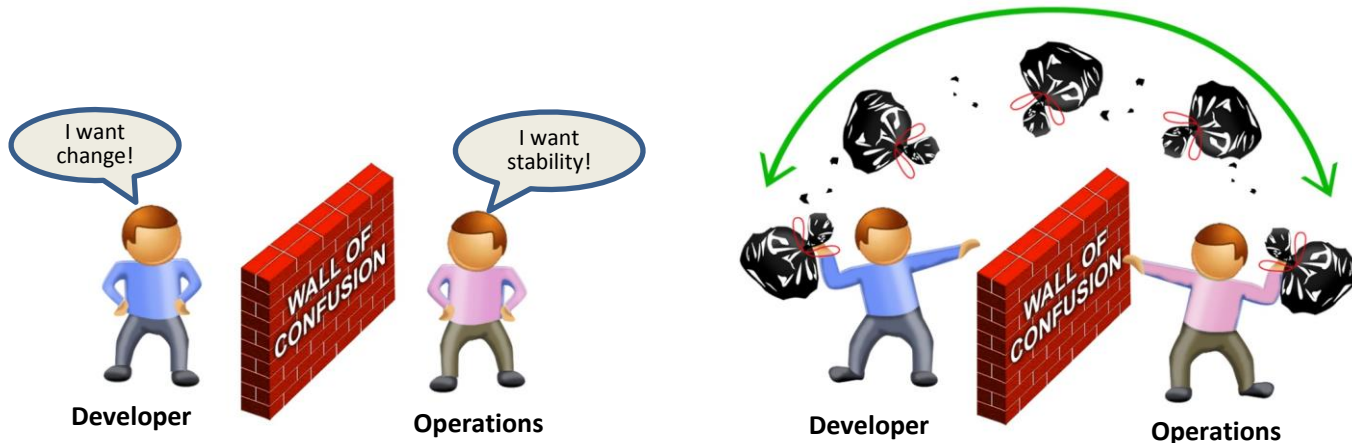
This space is known as “DevOps” –

Environments, Tools, People and Processes that support the application’s development and testing activities

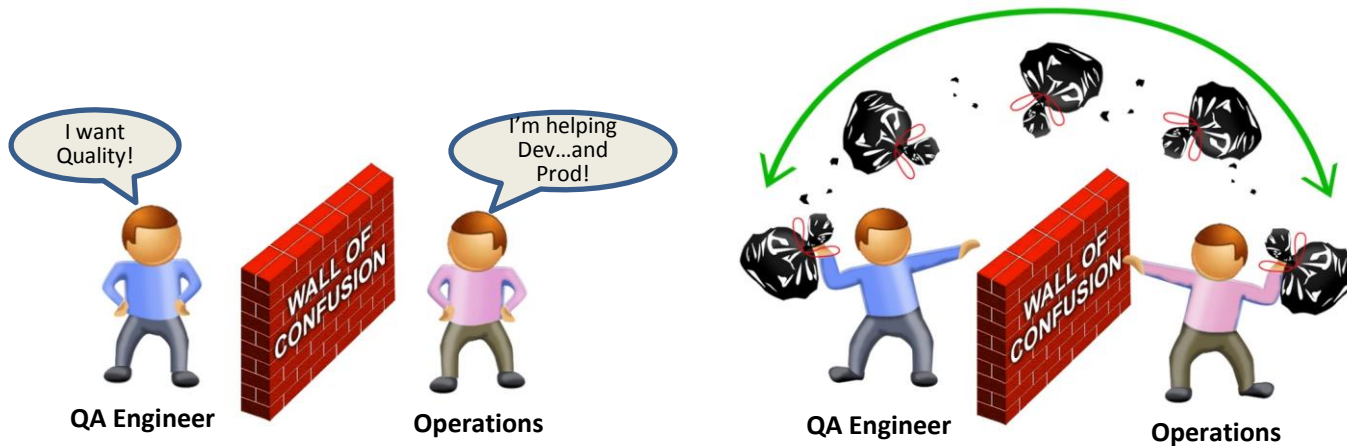


Where did the DevOps need come from?

Was it always there? Is it due to system complexity? An increase in demand from Agile? Does Cloud have anything to do with it?



What is the relationship between QA and Operations? Can we call it TestOps?



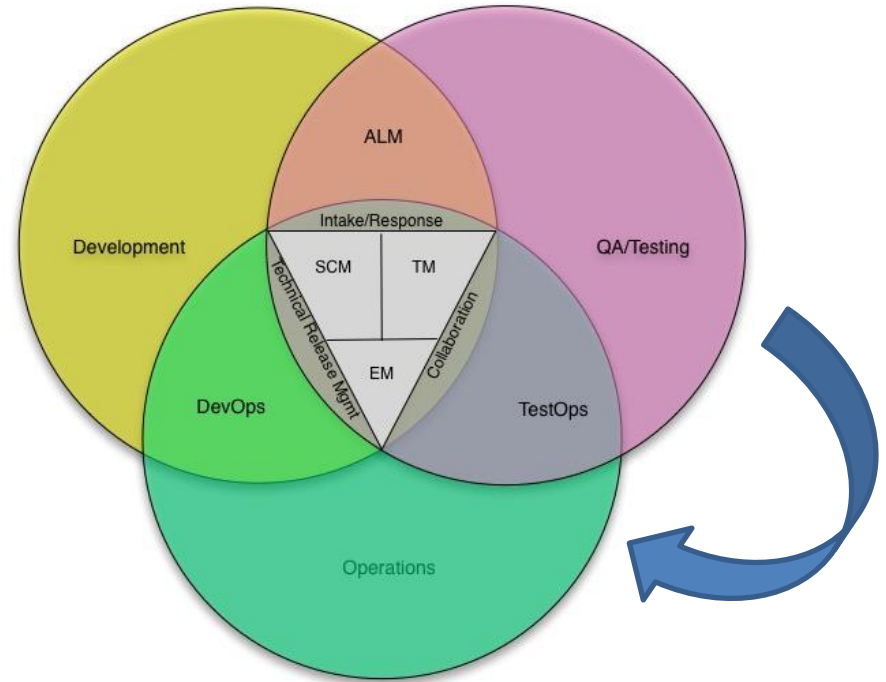
In addition to Quality:

- *Needs Environments and Test Data*
- *Automation Script Development*
- *Automation and Testing Tools Administration*
- *Traceability between Defects and Builds*

This diagram shows the areas of Development, QA/Testing, and Operations.

It also shows the overlaps between the areas and some of the functions that reside in these overlaps.

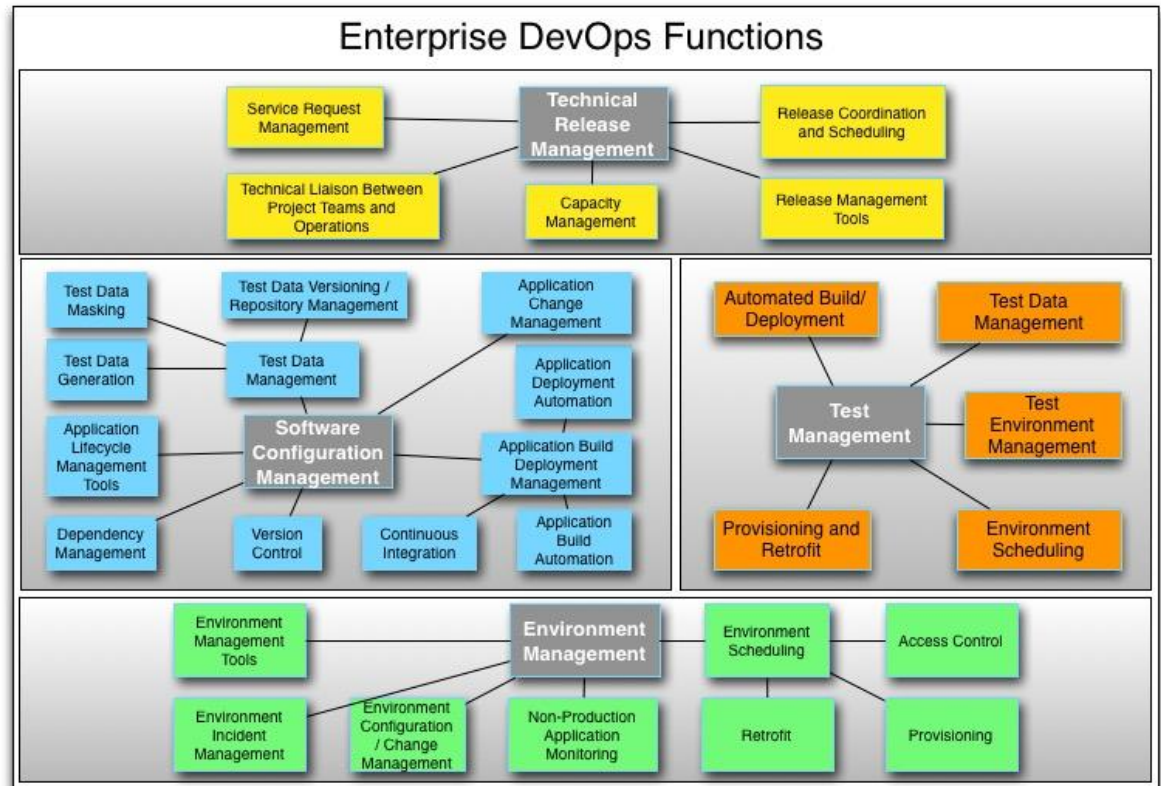
It's easy to understand how changes in one functional area will impact the other areas.



There are four Enterprise DevOps and TestOps Capability Areas –

- Test Management
- Software Configuration Management
- Environment Management
- Technical Release Management

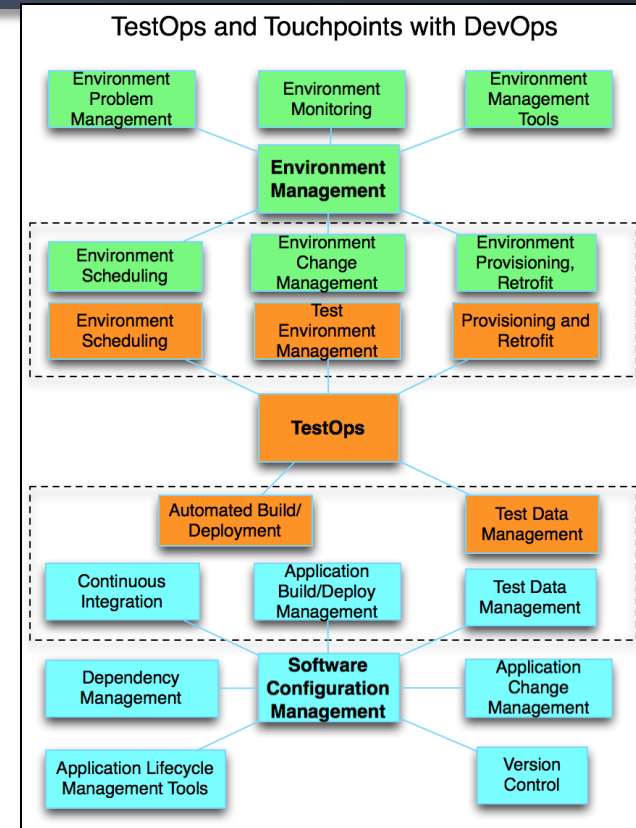
Test Management functions focus on the needs of the QA community... which are Collaboration and Automation opportunities between QA and Operations



As you can see from the diagram at right, TestOps is closely coupled with DevOps.

- On the Environment Management side, there are environment-related functions that are very similar between TestOps and DevOps.
- On the Software Configuration Management side, there are build-, deploy-, and data-related functions that are very similar between DevOps and TestOps.

As organizations become more mature, they will develop stronger capability in these areas and eventually implement automation across these functions.





Considerations

Below are Considerations to as you progress in your TestOps Journey.

Note that the considerations require resources with Environment and Automation skillsets that will engage across Dev, Test and Ops Team lines.

Note that, to be effective, the resources should be connected with each other through community, federation, or shared services.

- Automate the Test Cycle. Get all tests automated.
- Work with Dev Team to set up an automated “Smoke” or “Gate” Test as criteria to promote from Dev to Test.
- Work with Dev Team to set up automated build and deploy processes.
- Define and automate the hand-offs between the Dev Team and the QA Team
- Create test environment profiles in preparation for automatic environment provisioning.
- Use Cloud or other virtual technology to set up testing environments on demand.
- Integrate the automated testing capability with the Dev Team’s continuous integration capability to establish continuous testing.



Akaizen – another definition

Akyzen provides IT Strategy, Transformation, and Staffing Services. Founded by former Accenture DevOps experts, the company works with Fortune 100 companies to achieve their goals in the DevOps and TestOps Spaces.



“So, how do you implement DevOps in a large enterprise? At the risk of throwing in an old cliché, you do it the same way you eat an elephant – one bite at a time.”

“We began calling this approach “Application Kaizen”, or “A-kaizen”. Once we develop and fully understand the DevOps vision and strategy for an application or an enterprise, we define finite, foundational improvements we can execute in the DevOps Capability Areas. All the improvements are designed to improve the quality and speed of software development and deployment.”

“Some argue that DevOps is just better communication and collaboration between the Development Team and the Operations Team. Some argue that a “super” or “end-to-end” developer is the answer - someone to follow through on all the details. And some argue that you need tight process and the right tools to pull it off. There are some that argue that DevOps is an integral part of Agile (which is true) and that DevOps is exclusively used with Agile (which is not true). You could make a case for each of these solutions and guess what? You’d be right.”

Akyzen (a new technical term) is a webification of “Application Kaizen”, or “A-Kaizen”. In Japanese, “kaizen” means “good change”. Over the years, it has become synonymous with stepwise refinement and continuous improvement. “Application Kaizen” is the principle we live by here at Akyzen.

Walgreens

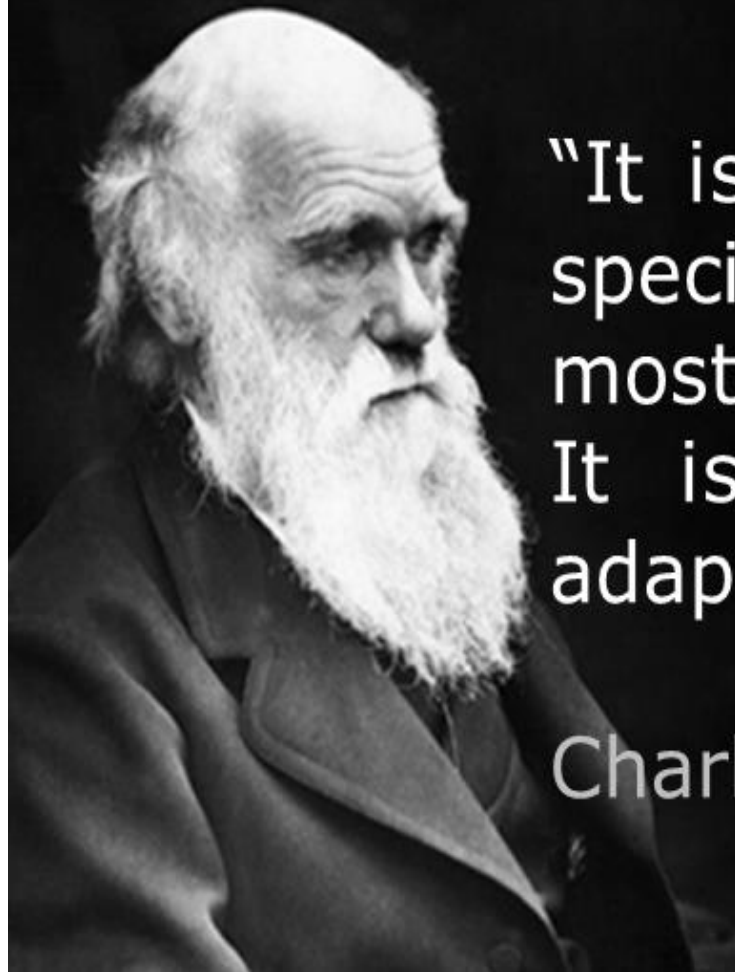
AT THE CORNER OF HAPPY & HEALTHY™

Continuous Testing in DevOps

Gowri Selka

Quality Assurance & Engineering

November 19, 2014



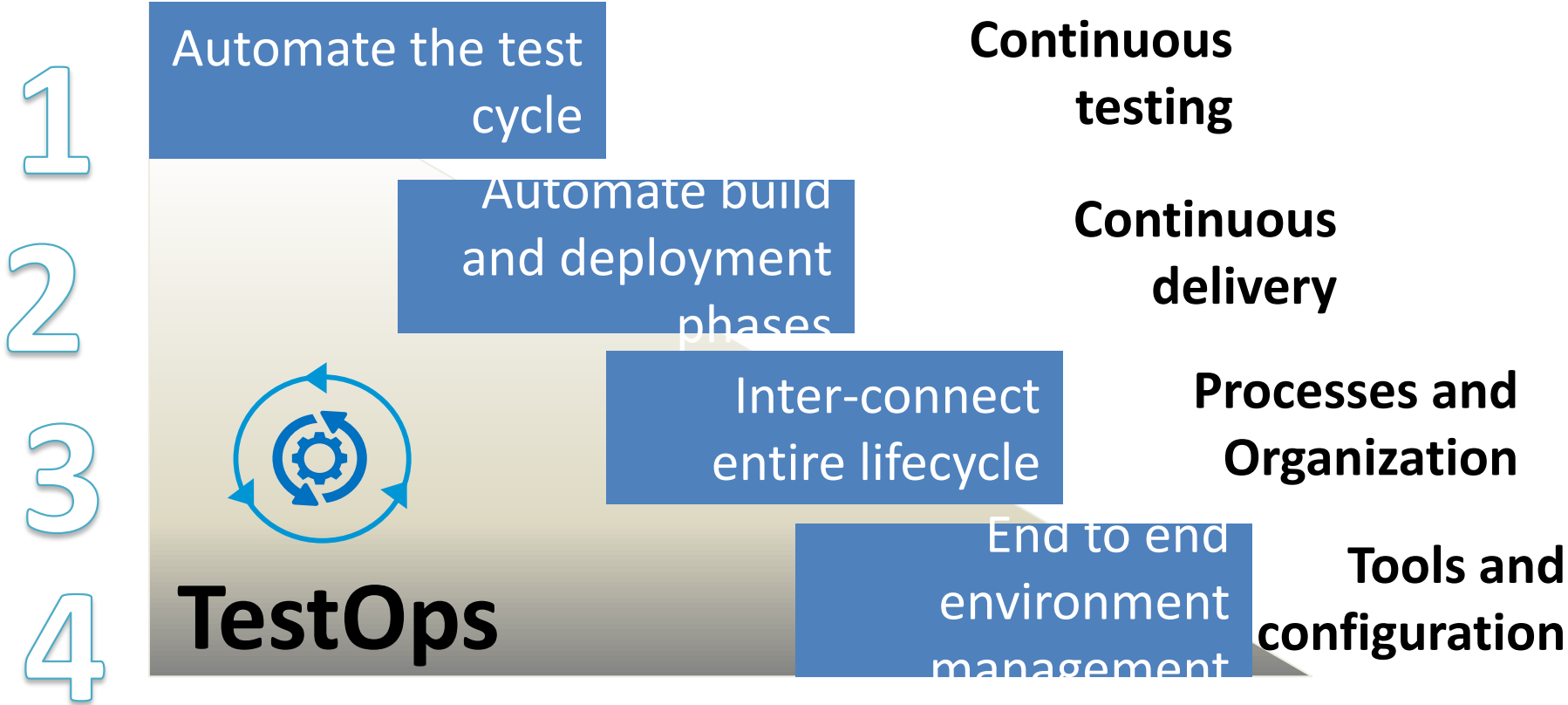
"It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is most adaptable to change."

Charles Darwin (1809 – 1882)

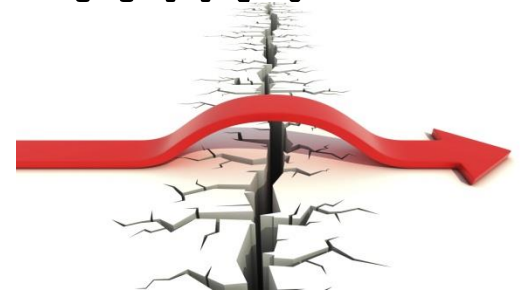
World Quality Report 2014-15 Key Recommendations

- 1) Adapt your QA and Testing models to new technologies required by digital transformation.
- 2) Find your own formula for a successful TCOE.
- 3) Focus on all-channel experience testing for validating the end-to-end customer experience.
- 4) Create faster, yet more structured, testing solutions for agile and DevOps driven projects.
- 5) Continue improving testing efficiency while controlling cost levels.
- 6) Get a head start on automation with test-driven development.
- 7) Increase focus on non-functional testing – specifically security and performance testing.
- 8) Consider increasing the use of cloud and virtualisation solutions for your test environments.

Four Focus areas for Continuous Integration



Continuous Integration – Why?



- Increase transparency
- Repeatable & consistent build process
- Verification of code quality and functionality at every build

- Repeated manual low-value tasks are eliminated
- Test Automation enabling early identification of defects

- Eliminates human errors in the build process
- Immediate notification and visibility to team on potential issues

Continuous Testing– Implementation

Assess

- Analyze application landscape
- Understand the technology and tools landscape for Source code management and Build
- Identify tools for Continuous integration setup
- Identify all required plugins
- Identify development team roles and maturity
- Gather required metrics to develop business case

- 📄 Environment heat map
- 📄 Business case
- 📄 Testing maturity and capability

- 📄 Tools strategy

Define

- Define environment strategy
- Utilize virtualization tools
- Define process for developer's
 - code branching and merging strategy
- Define interaction model between Dev and QA for Collaboration and automation
- Define criteria for build, deploy and script execution
- Define end to end test strategy
- Define automation framework

- 📄 Code branch-merge strategy
- 📄 Interaction model & guidelines
- 📄 CIT workflow
- 📄 Automation framework

- 📄 End to end test strategy

Setup

- Install tools
- Configuration of CI tool with version management tool, build and test management tools
- Build business processes
- Link all test cases to be run in CI tool
- Perform POC
- Validate business case
- Reinforce processes for sustainability

- 📄 Inter connected tools between Dev, Build, QA
- 📄 Automated test scripts
- 📄 Virtualize environments

- 📄 Continuous testing

Execute & Maintain

- Define test strategy for each project
- Maintain business process flows and link to test components
- Automate deployment and testing for every build
- Establish feedback loop based on success criteria
- Maintain environment and configurations

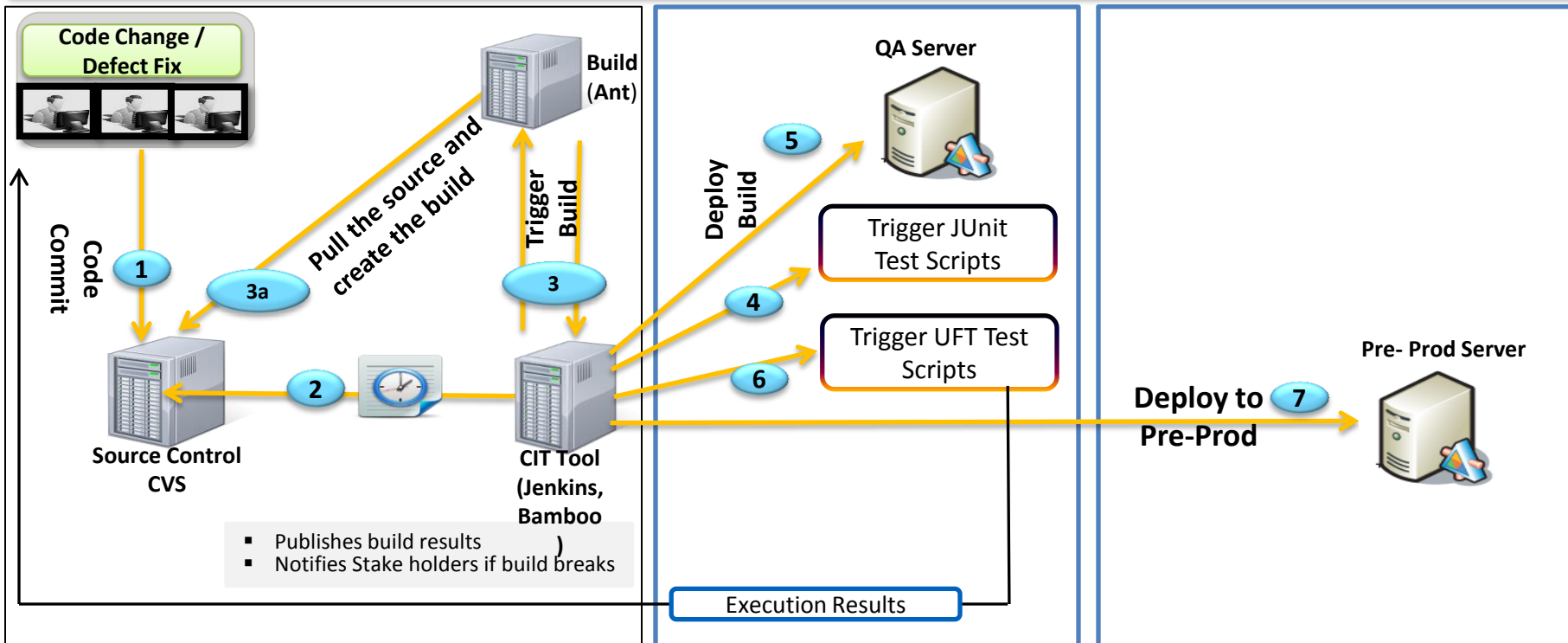
- 📄 Build and deployment report
- 📄 Unit test results
- 📄 Test execution report
- 📄 Metrics to measure

improvements and benefits

Deliverables

Continuous Testing Solution Approach

Process Flow: End to End Continuous Testing



Continuous Build

Continuous Validation

Continuous Delivery