DevQASecPerfOps

Considerations for Building Out a Robust Continuous Delivery Pipeline

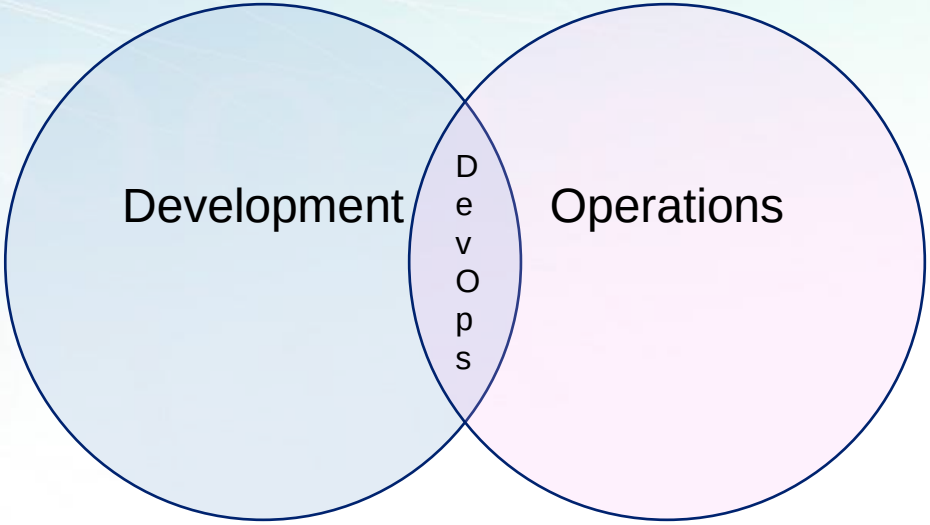
Presented by Tim Lyon
lyonmsu@yahoo.com
https://www.linkedin.com/in/timlyon

Who is Tim Lyon?

- **Advisory Board Member for Astegic who consults upon strategic independent QA offshore solution offerings and market strategy**
- **Former Vice President, Quality Assurance & Shared Delivery Services at nThrive**
- **Over 20 years of Software Development Quality Assurance Experience, mostly in “Agilish” Teams**
- **Managed Quality Assurance, Automation, Business Analyst, Development, Configuration Management, and Release Management Teams**
- **Proactive Driver Towards Continuous Delivery with Continuous Testing**
- **“Accomplished Assured Advancement Advocate”**

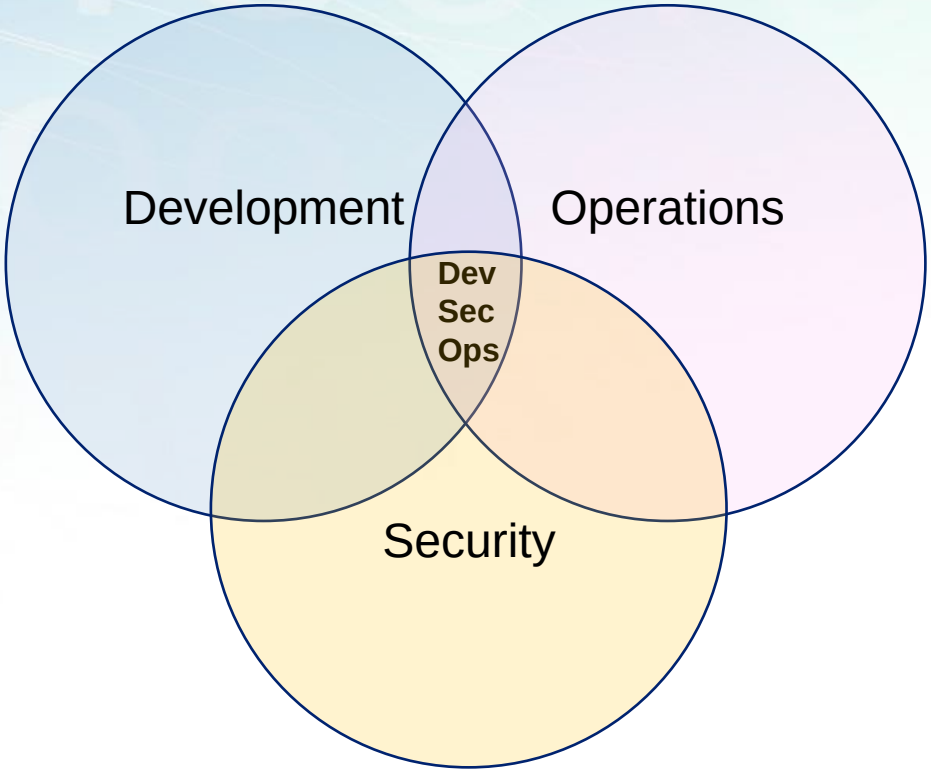
DevOps

- Emphasizes ***organizational change*** with greater collaboration between the various groups involved from development on through release and ongoing operations and support
- Everyone is responsible for quality
- Focuses on lean and agile principles with automating the process



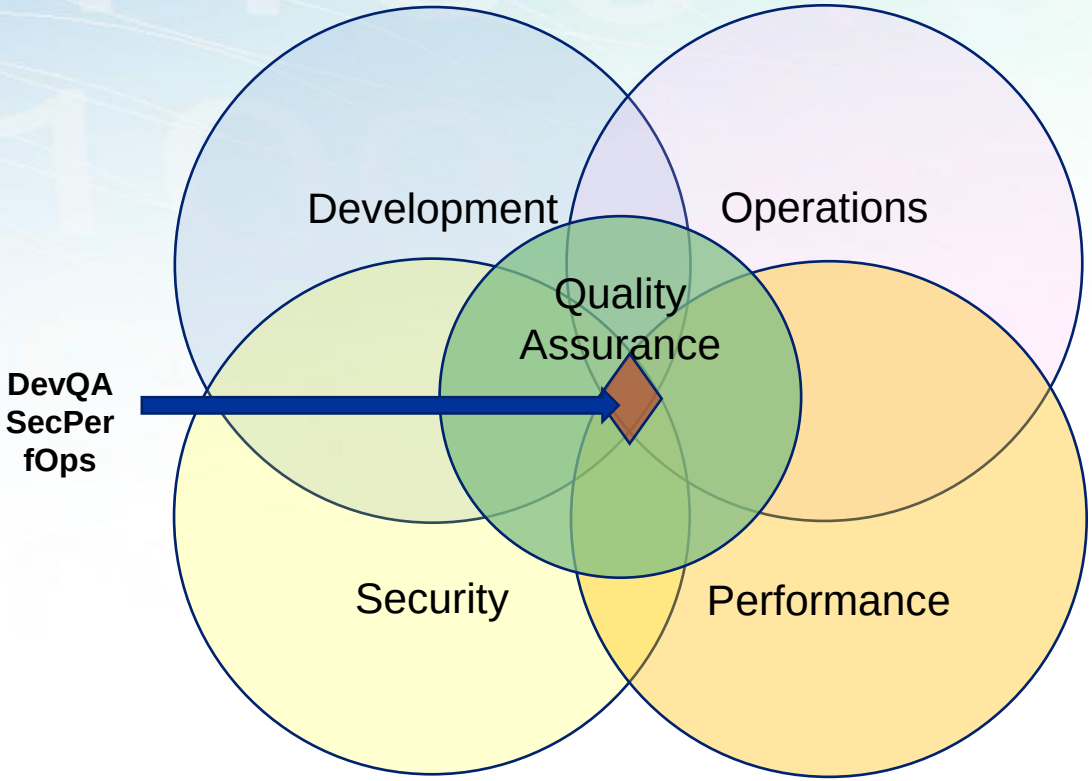
DevSecOps

- Extends DevOps with the inclusion of ongoing proactive security testing, analysis, monitoring, and resolution
- Everyone is responsible for quality AND security
- Objective is to validate security throughout without slowing the delivery lifecycle



DevQASecPerfOps

- Extends DevSecOps with the inclusion of ongoing quality assurance and performance testing to yield better overall reliable and scalable software deliveries
- Everyone is responsible for quality, security, reliability, & scalability
- Objective is to incorporate testing in various capacities throughout while not slowing the delivery lifecycle



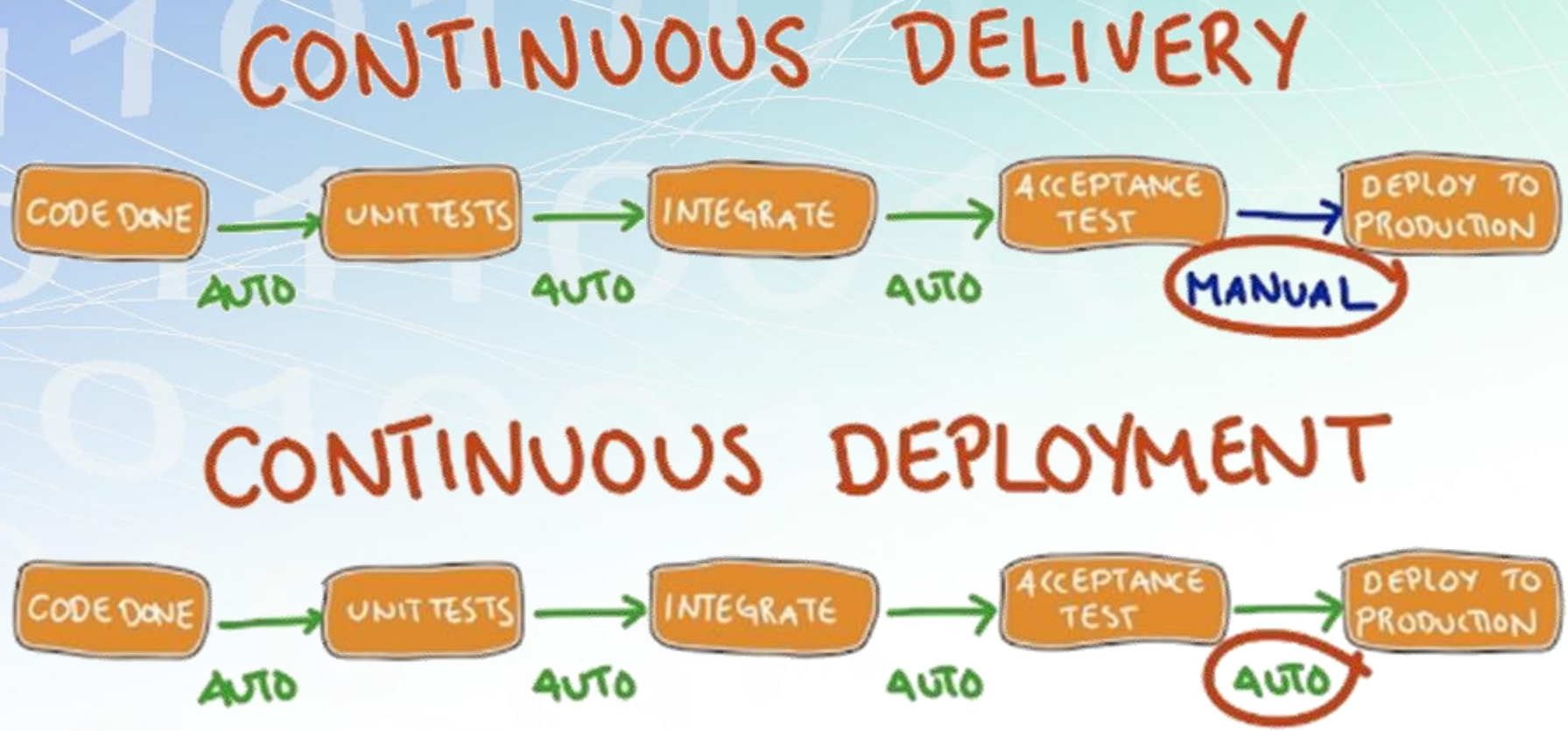
How to Proceed

- **Change Organization** – Incorporate Dev, QA, Security, Ops within one organizational structure
- **Change Culture** – No silos, everyone responsible for quality, security, reliability
- **Automate Delivery Pipeline** – Reduce constraints to delivering quality , secure, reliable, scalable releases to production
- **Monitor, Analyze, Adapt** – Continually monitor KPIs of affected business and systems and adapt to increase performance and reduce constraints

Reality Check?

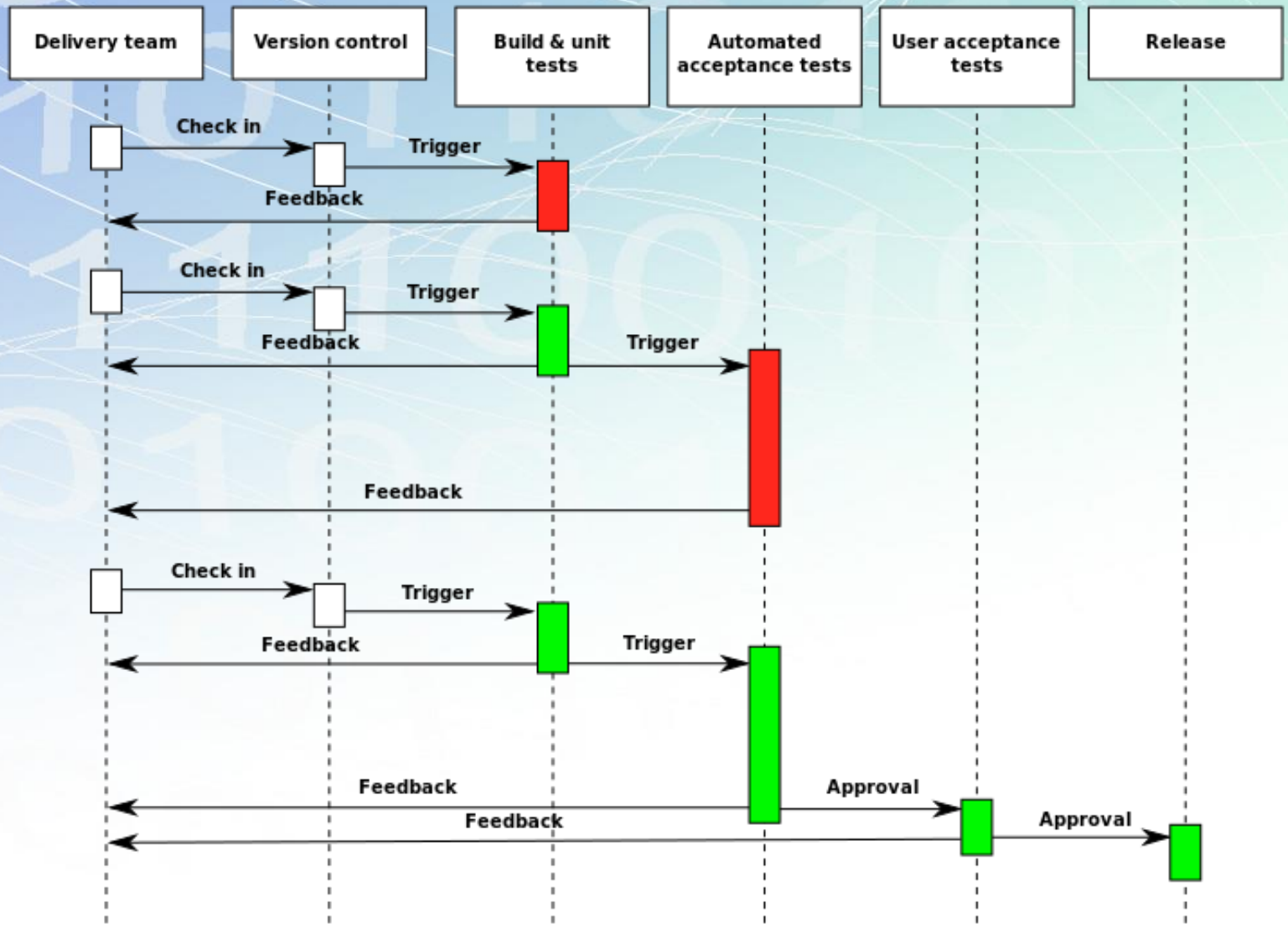
- ~~**Change Organization** – Incorporate Dev, QA, Security, Ops within one organizational structure~~
- **Change Culture** – No silos, everyone responsible for quality, security, reliability
- **Automate Delivery Pipeline** – Reduce constraints to delivering quality , secure, reliable, scalable releases to production
- **Monitor, Analyze, Adapt** – Continually monitor KPIs of affected business and systems and adapt to increase performance and reduce constraints

Delivery Pipelines Types



http://blog.crisp.se/2013/02/05/yassalsundman/continuous-delivery-vs-continuous-deployment

Continuous Delivery Practices



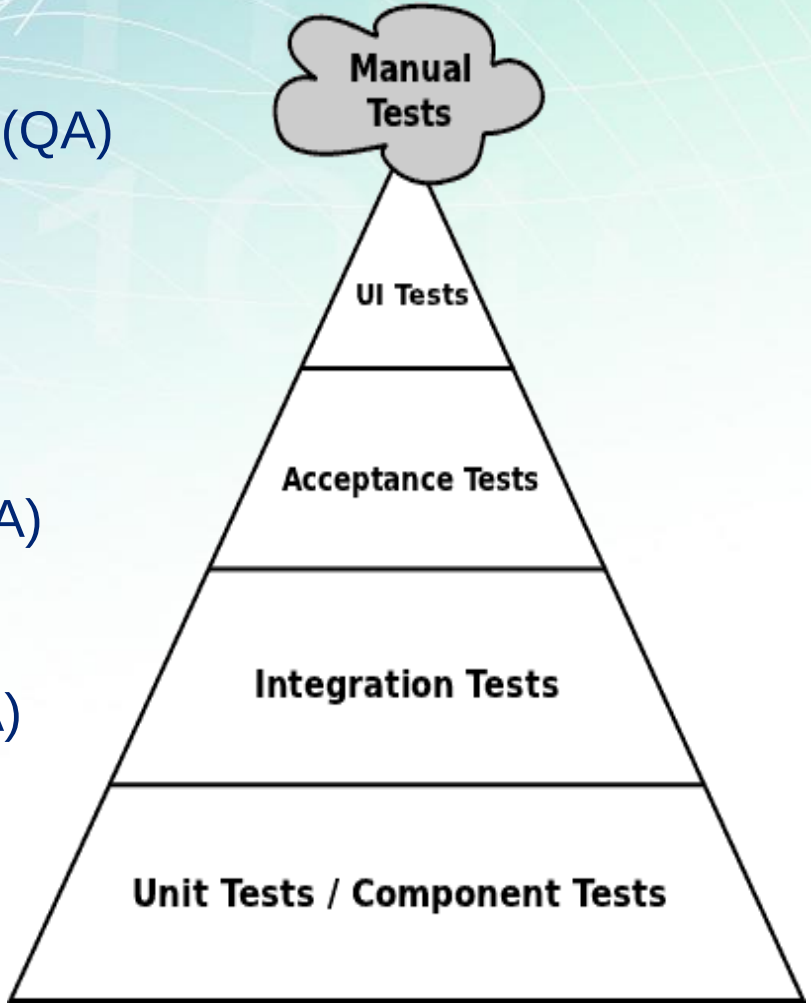
https://en.wikipedia.org/wiki/Continuous_delivery#/media/File:Continuous_Delivery_process_diagram.svg

What About QA?

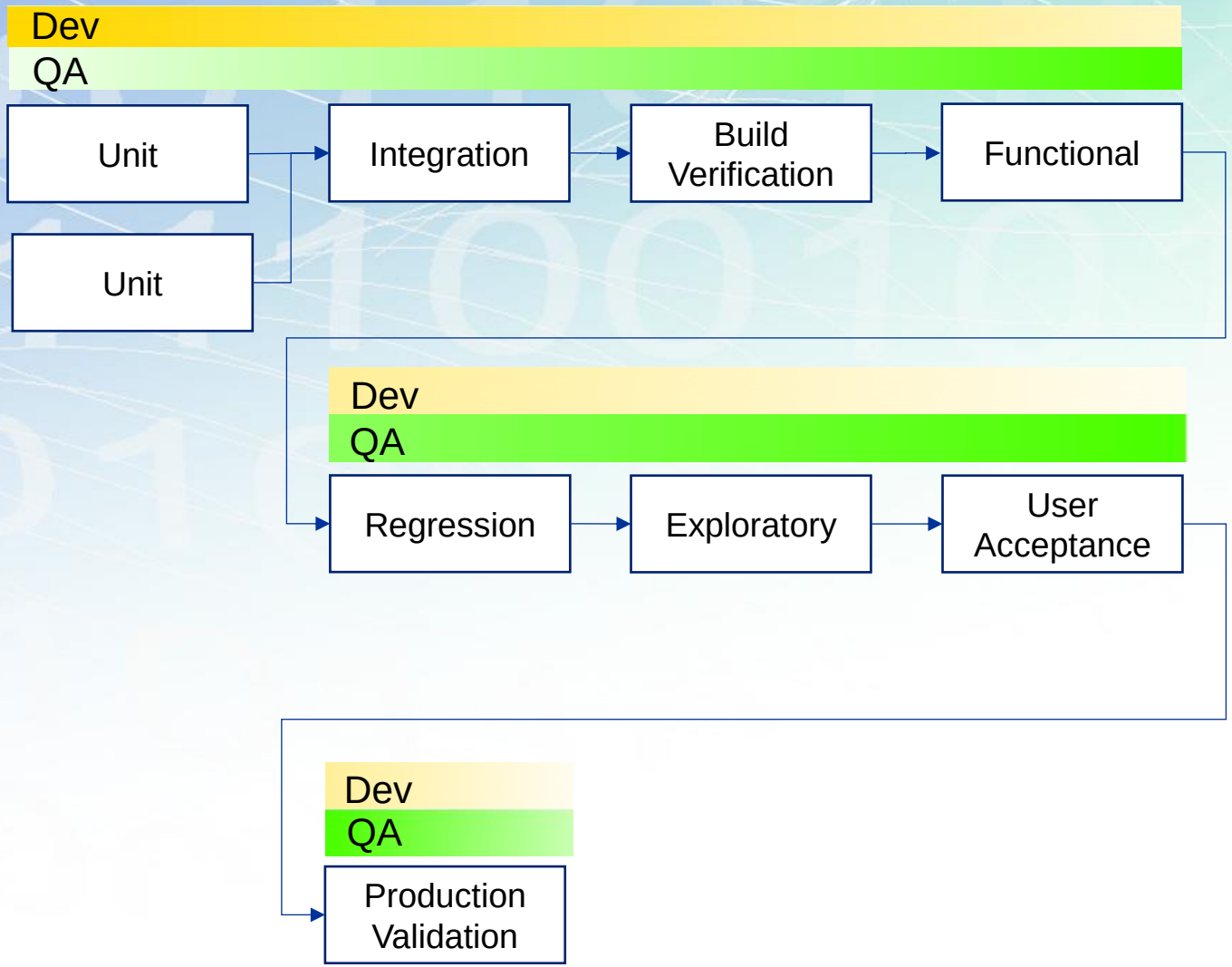
- **Get Involved Early and Often (Shift Left)**
  - **Assist Developing New Acceptance Tests**
  - **Collaborate Upon Naming Conventions and Identifiers**
  - **Identify Negative and Alternate Paths to Acceptance Criteria**
  - **Plan for Changes to Existing Regression Tests**
  - **Look for Opportunities to Optimize Tests and Coverage**
  - **Update Functional, Integration, System, Security, Performance & Regression Tests**
  - **Execute Exploratory/Session Based Tests**

Test Pyramid

- Manual Tests
  - Complex, Alternate, Negative, etc. (QA)
- UI Tests
  - Compatibility & Usability (QA)
- Acceptance Tests
  - Specifications Validation (Dev & QA)
- Integration/API Tests
  - Integrated Functionality (Dev & QA)
- Unit Tests
  - Foundation Building Blocks (Dev)



Incorporating QA Tests



What About Security?

- **Types of Security Tests to Consider**
  - **Vulnerability Scanning** – Scanning for known vulnerability signatures
  - **Security Scanning** – Identifying network and system weaknesses
  - **Penetration Testing** – Simulates an attack from a malicious hacker
  - **Risk Assessment** – Analyzing security risks observed in an organization
  - **Security Auditing** – Internal inspection of apps and OS for security flaws
  - **Ethical Hacking** – Using common hacking techniques against an organization
  - **Posture Assessment** - Combines Security scanning, Ethical Hacking and Risk Assessment

Zed Attack Proxy (ZAP)

- **Mature Open Source Project from OWASP Foundation (Open Web Application Security Project) founded 2001**
- **Combines penetration testing and vulnerability scanning**
- **Supports both manual and automated modes**
- **Written in Java, includes a robust client and a proxy-based API that works with any language**
- **Checks for a wide range of web vulnerabilities**

https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

Vulnerabilities Identified

- Cookie Secure Flag
- Xpath Injection
- Cross Domain Script Inclusion
- XXE
- Header XSS Protection
- Padding Oracle
- Mixed Content
- Charset Mismatch
- Password Autocomplete
- Cookie - Loosely Scoped
- Private Address Disclosure
- Insecure JSF View State
- Session Id in URL Rewrite
- Servlet Parameter Pollution
- X-Content-Type-Options
- Viewstate
- X-Frame-Option
- CSRF Token
- Heartbleed OpenSSL Vulnerability
- Insecure HTTP Method
- LDAP Injection
- Remote Code Execution
- Session Fixation
- Source Code Disclosure (3 Types)
- SQL Injection (4 Types)

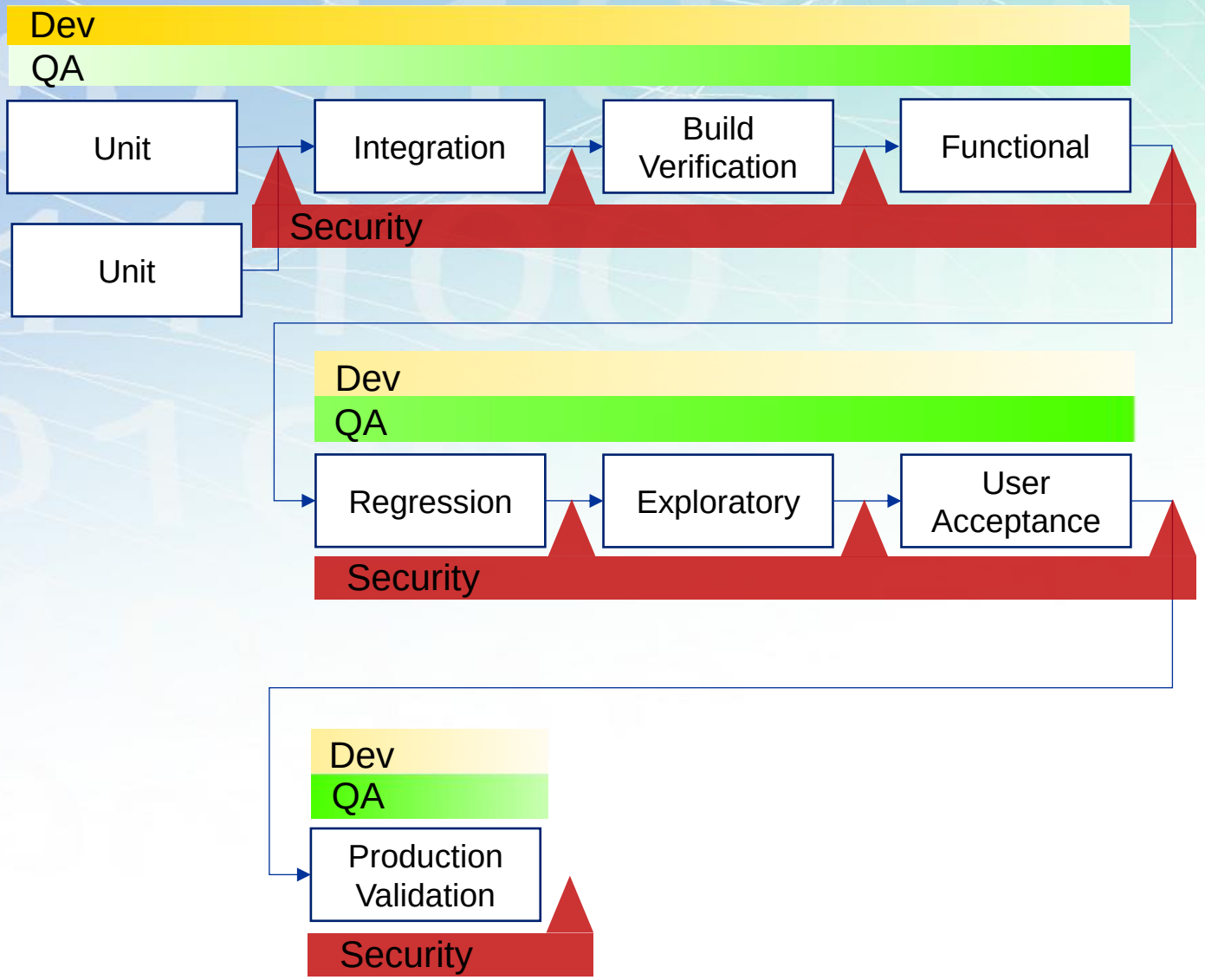
Security Test Methodology

- **Selenium Testing Framework**
- **Log in to AUT using Selenium**
- **Load ZAP and Invoke Scanner via Proxy API, passing URL of application**
- **Capture XML report of identified vulnerabilities (Low, Medium, High)**
- **Analyze and save baseline**
- **On subsequent runs, compare report against baseline**
- **Any new or high risk issues cause test to fail**
- **Results reported back with pass/fail and test log like any other test**
- **Tests run as part of Continuous Integration triggers**
- **New issues either fixed or added to baseline**

Security Test Summary

- **Regular and periodic automated penetration tests uncover potential vulnerabilities before they are released to production**
- **Increases awareness among developers of common mistakes that can expose the web app to malicious attacks**
- **An ounce of prevention is worth much more than a pound of cure where security is concerned**
- **A relatively small development effort can yield big payback if it catches even one potential security risk**

Incorporating Security Tests

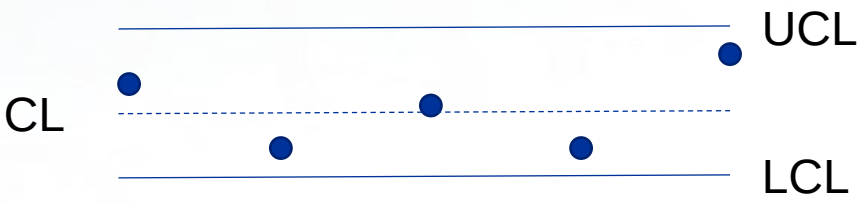


How About Performance?

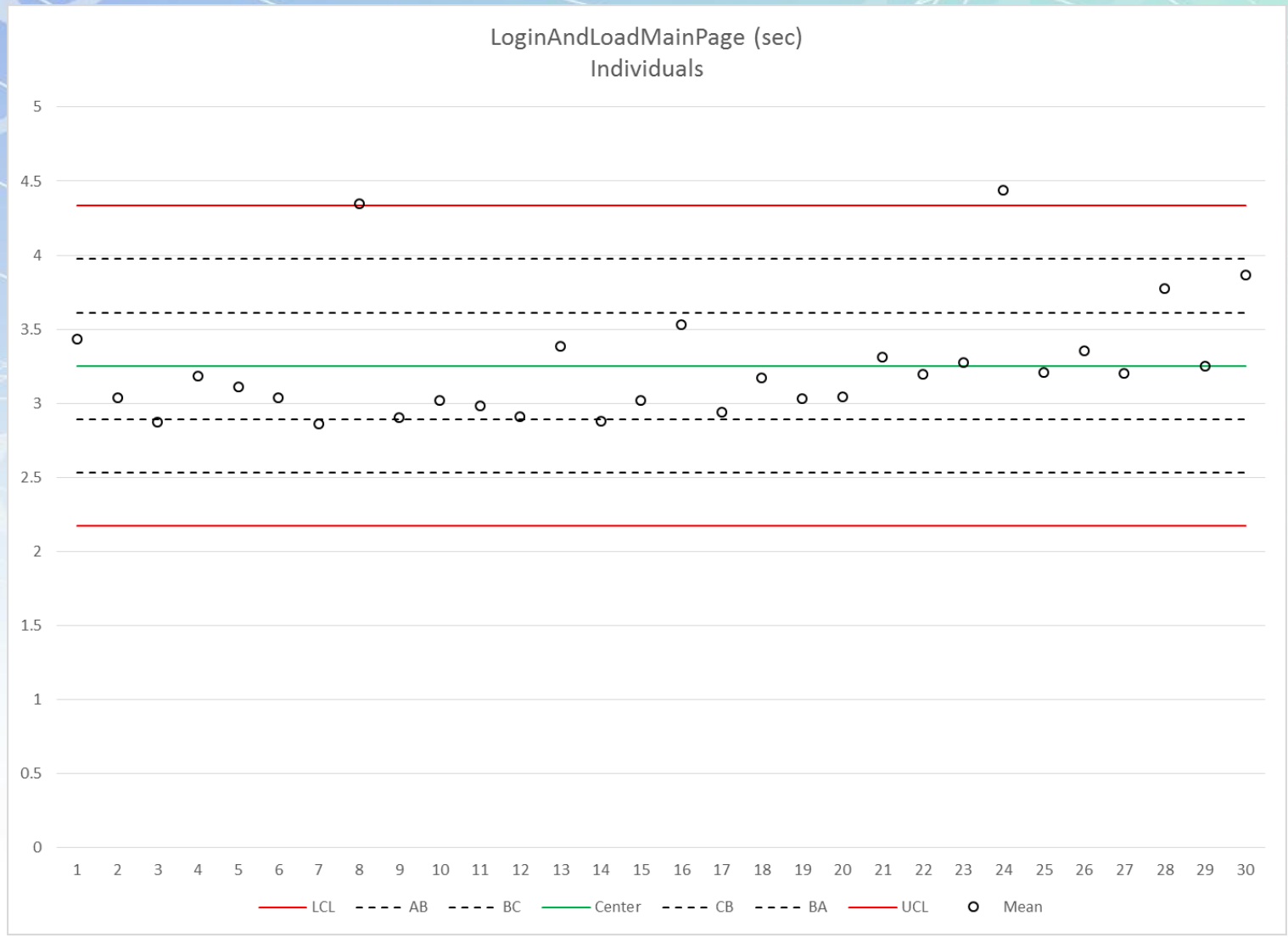
- **Objectives**
  - **Discover performance issues early**
  - **Reduce $, time, risk to fix discovered issues**
- **Requirements**
  - **Detailed knowledge of the system**
  - **Cross-team collaboration**
  - **Well written tests**
- **Challenges**
  - **False positives (Find perf issue when none exists) – Type I errors**
  - **False negatives (Don’t find perf issue when it exists) – Type II errors**

Baseline Approach – Control Charts

- **Control Charts**
  - **Goal of control charts is to determine if performance change is due to common cause (e.g. network latency variation) or special causes (e.g. defects)**
- **Concepts**
  - **Center Limit (CL) = mean (or median) of all readings in the baseline set**
  - **Upper Control Limit (UCL) = upper limit of normal behavior**
  - **Lower Control Limit (LCL) = lower limit of normal behavior**



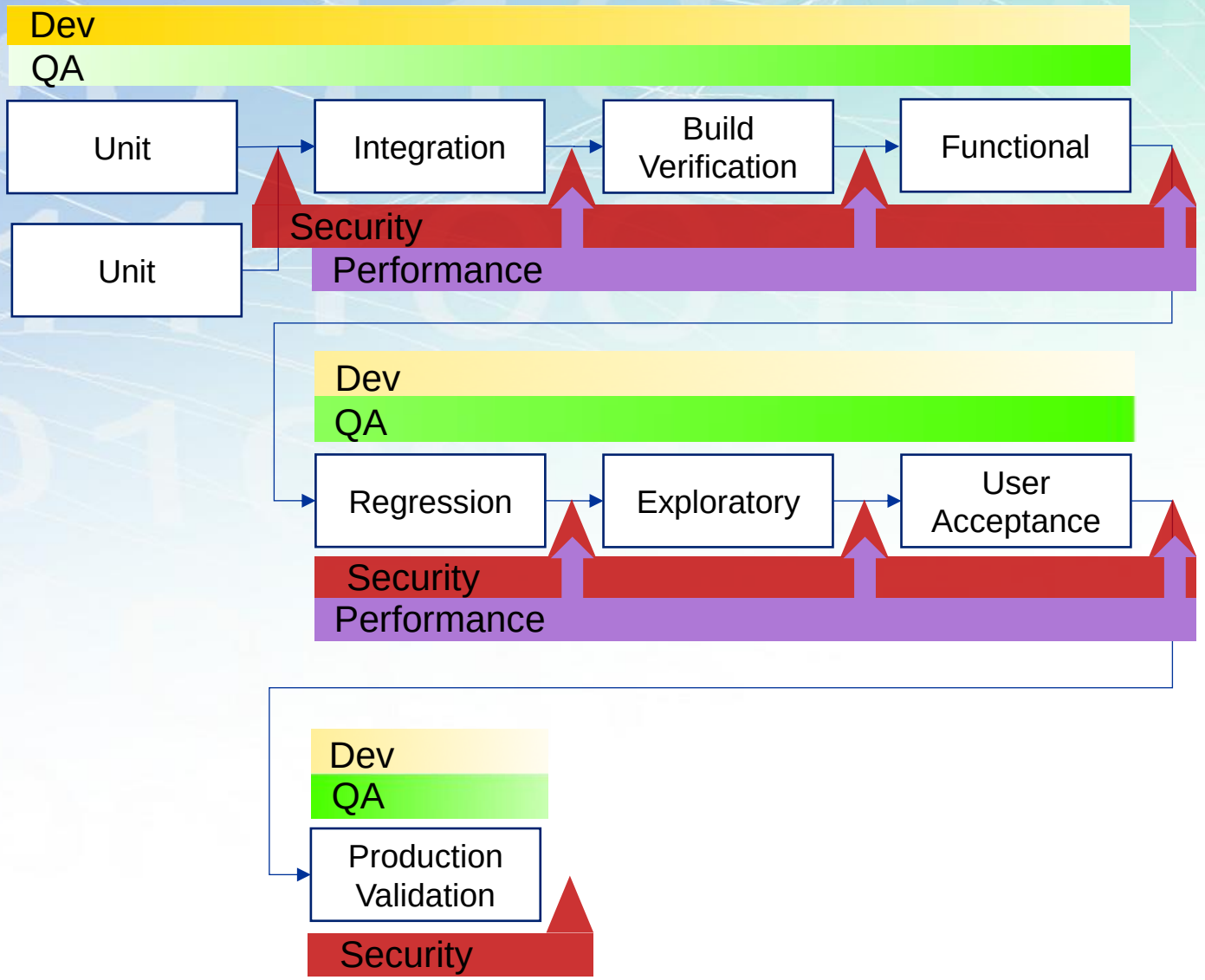
Control Chart Example



Baseline Control Charts Summary

- **Baseline enable any performance test**
  - **Create baseline**
  - **Verify against baseline**
  - **Control charts created for each transaction defined in performance test**
- **Can be integrated with test management tools**
  - **Ad-hoc as well as automated pipeline runs**
  - **Consolidated Pass/Fail Reporting**
- **Foundation of industry standard statistical process control concepts**
- **Visualization of performance**
- **Allows detailed detection of where the performance issue is found**

Incorporating Performance Tests



Overall Recommendations

- **Identify and Prioritize All Your Delivery Constraints**
- **Address One Bite at a Time**
- **Be Solution Oriented**
- **Remove (or at Least Reduce) Constraints**
- **Work with Teammates, Developers, Operations, Security and Any and All Between Solution Deliveries**
- **Keep Time of Delivery as a Prime Consideration to Evaluate and Refactor Processes and Implementations**

Tools & Sites

- **Open Web Application Security Project (OWASP) – Freeware with Great Security Testing Articles**
  - https://www.owasp.org
  - **OWASP Zed Attack Proxy** https://www.owasp.org/index.php/GPC_Project_Details/OWASP_ZAP
- **SonarQube – Open Source Code and Vulnerability Analysis Dashboard**
  - https://www.sonarqube.org/
- **Jmeter - Open Source Performance Testing**
  - http://jmeter.apache.org/
- **Awesome-DevSecops - GitHub Site with Great Articles and Links**
  - https://github.com/devsecops/awesome-devsecops
- **Thanh H. D. Nguyen - Published Control Chart Research Papers**
  - **"Automated Verification of Load Tests Using Control Charts" – 2011 IEEE Asia-Pacific Software Engineering conference**
    - http://barbie.uta.edu/~jxu/Automated%20Verification%20of%20Load%20Tests%20Using%20Control%20Charts.pdf
  - **"Automated Detection of Performance Regressions Using Statistical Process Control Techniques" - 2012 ACM/SPEC International Conference on Performance Engineering**
    - http://sail.cs.queensu.ca/Downloads/ICPE2012_AutomatedDetectionOfPerformanceRegressionsUsingStatisticalProcessControlTechniques.pdf

DevQASecPerfOps

Considerations for Building Out a Robust Continuous Delivery Pipeline

Presented by Tim Lyon
lyonmsu@yahoo.com
https://www.linkedin.com/in/timlyon