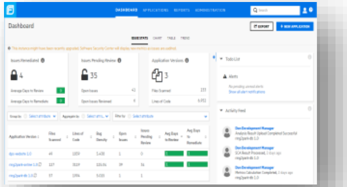
MICRO FOCUS®

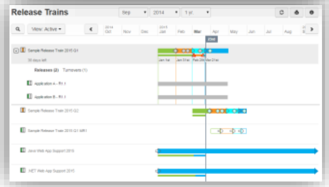
DevSecOps: How to integrate Security & Compliance into your Continuous Delivery pipelines

Kevin A. Lee – kevin.lee@microfocus.com
Senior Solutions Architect

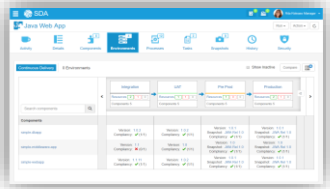
DevOps and Release Management Webinars (UK)



- **Automating DevSecOps**: How to embed security into your continuous delivery pipelines



- **Enterprise DevOps**: Release Management for the multi-modal Enterprise



- **Continuous Delivery Pipelines**: Automating the value stream through continuous release

https://www.microfocus.com/campaign/serena-release-management/

Agenda

01. What is DevSecOps?

02. Implementing DevSecOps

03. Micro Focus Solutions

04. Demonstration

05. Q&A

What is DevSecOps

Security threats are multiplying exponentially…

… this is just in the UK!!!

TalkTalk

157,000 customers data breached in 2015, £400k fine

SPORTS DIRECT.com

employee data compromised in 2016

3

over 133,000 accounts hacked to acquire handsets in 2016

TESCO

theft of £2.5m from 9,000 customers' account in 2016

ticketmaster

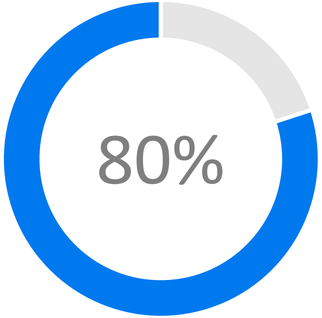
Data breach stealing customer payment details who transacted on site in 2018, potential £17m fine

BRITISH AIRWAYS

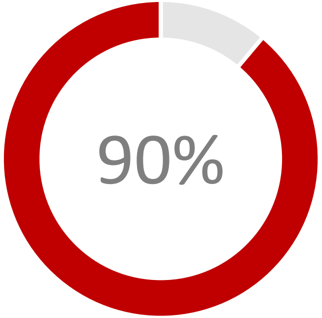
Personal and financial data of 380,000 passengers hacked in 2018, potential £500 fine

Potential fines scaling massively under EU GDPR (May 2018)

The majority of security breaches today are from application vulnerabilities



Percentage of applications containing at least one critical or high vulnerability. [1]

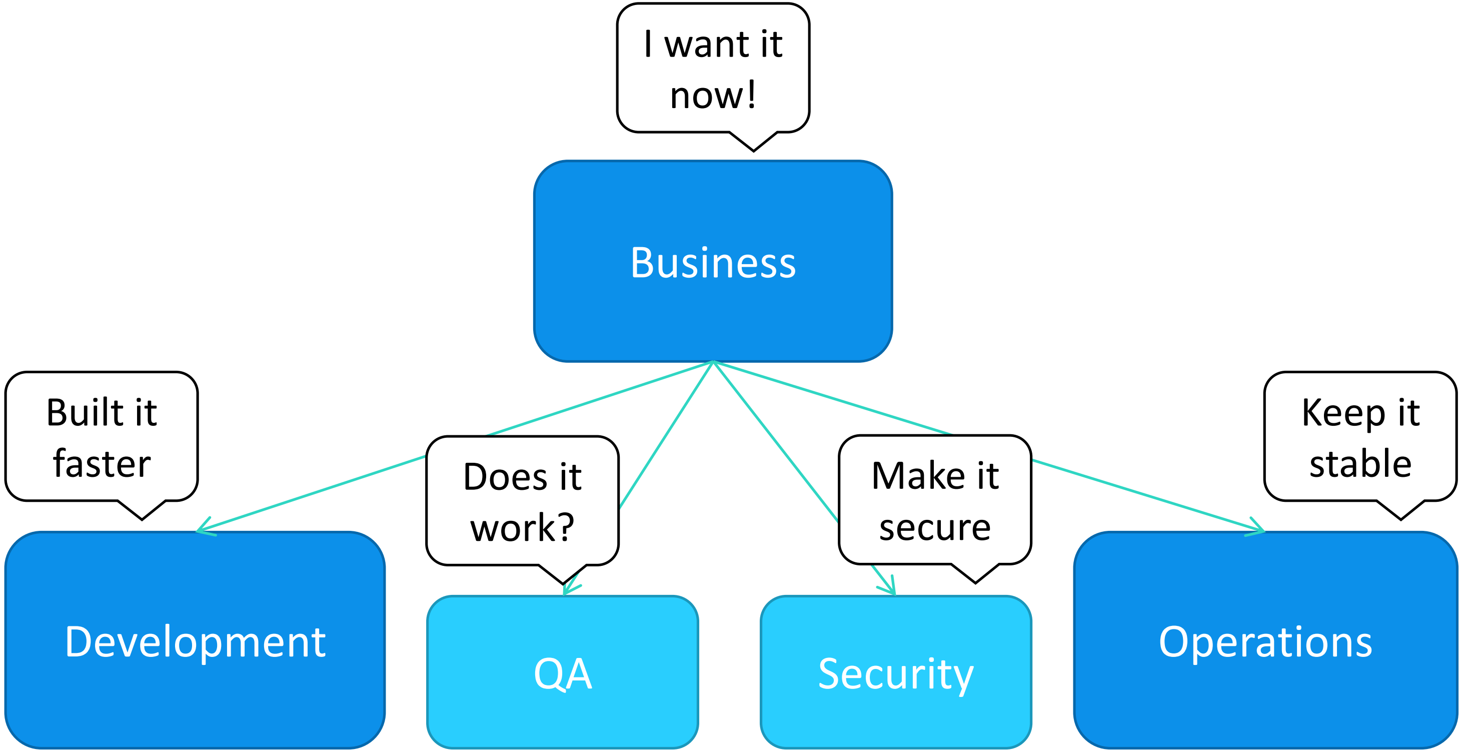


Security incidents from exploits against defects in the design or code of software. [2]

[1] 2017 Application Security Research Update" by the HPE Software Security Research team

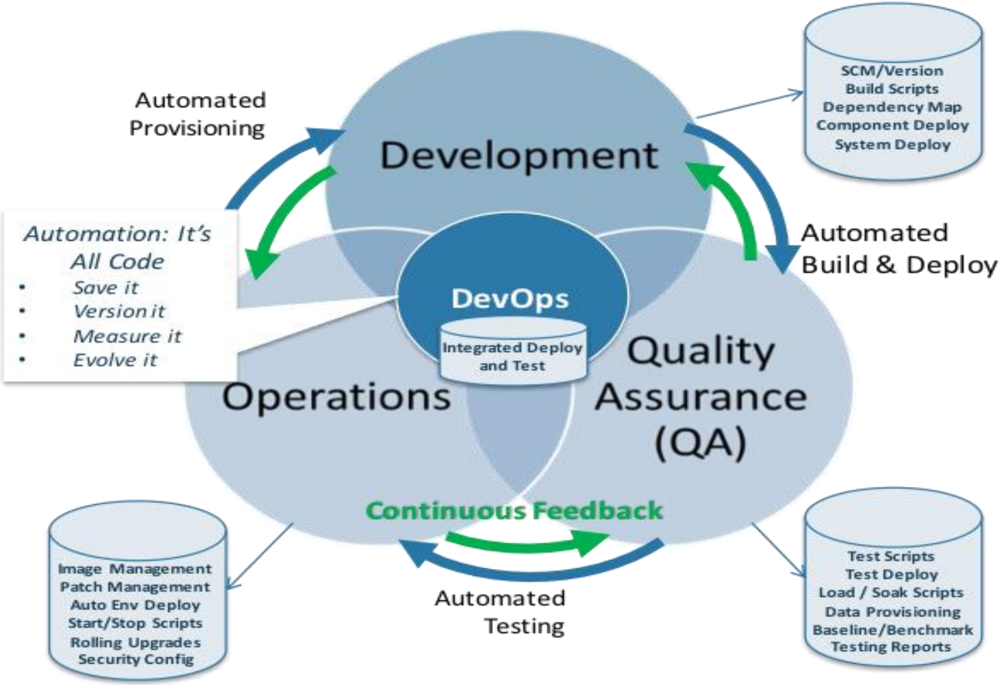
[2] U.S. Department of Homeland Security's U.S. Computer Emergency Response Team (US-CERT)

Enterprise organizations have competing forces…



DevOps

“**DevOps** (development and operations) is an enterprise software development phrase used to mean a type of agile relationship between development and IT operations. The goal of **DevOps** is to change and improve the relationship by advocating better communication and collaboration between these two business units.” *(webopedia.com)*



Security??

Make it secure … the old way

- Security was often tested and verified only after deployment:
  - A different team who knew all about security
  - They were disconnected from the development process
  - And slow to feedback issues and resolve them
- This does not work in a modern Agile / DevOps environment:
  - Security analysis needs to be automated & continuous
  - Security experts need to act as part of the sprint team
  - Security can/should be continually reviewed as part of code peer review

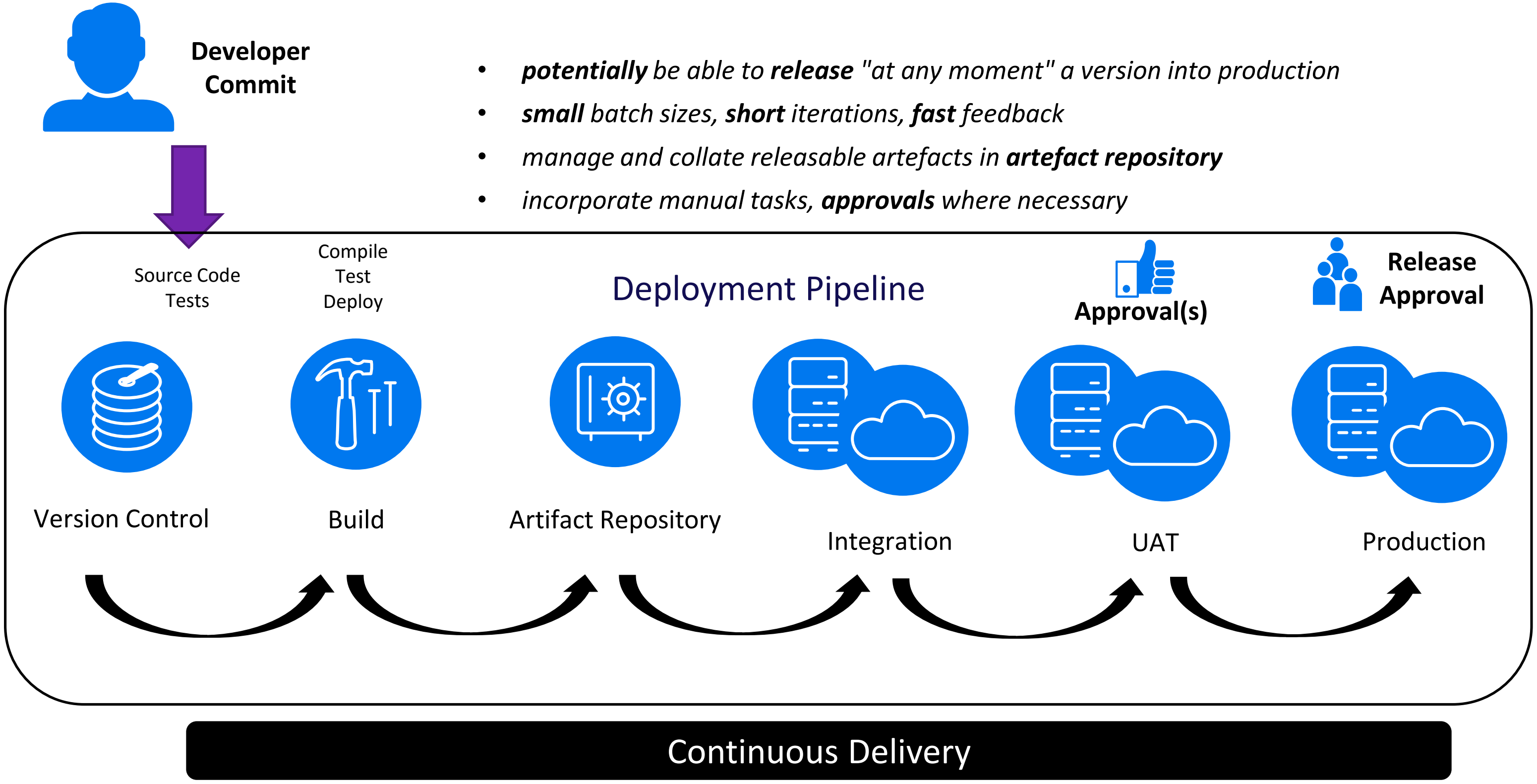
DevSecOps

- DevSecOps is:
  - A team/community effort, not a person
  - Automated and autonomous security
  - Security at scale
- DevSecOps role:
  - Is not there to audit code
  - Is there to implement the control segments to validate and audit code and artefacts as part of Continuous Delivery
  - Should be (mostly) automated…

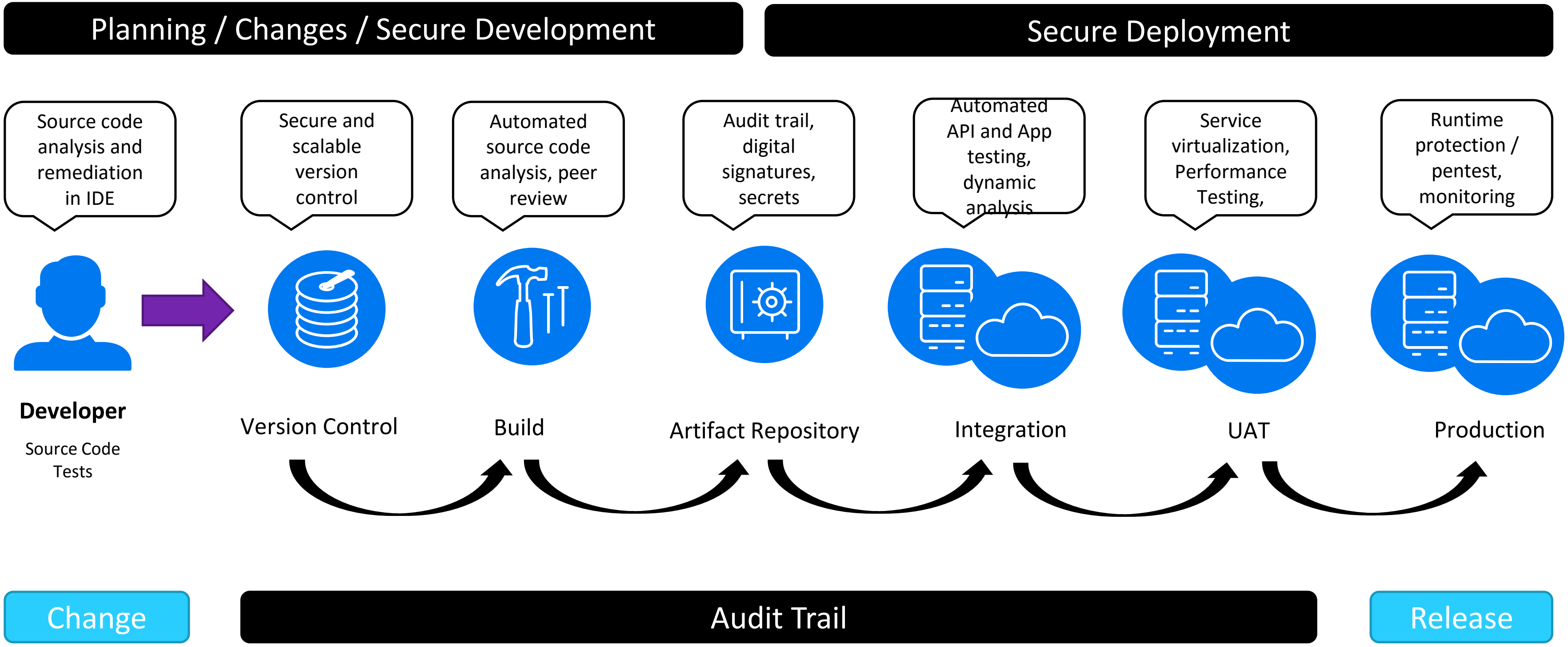
DevSecOps make everyone responsible for security

Continuous Delivery

- ***potentially*** *be able to* ***release*** *"at any moment" a version into production*
- ***small*** *batch sizes,* ***short*** *iterations,* ***fast*** *feedback*
- *manage and collate releasable artefacts in* ***artefact repository***
- *incorporate manual tasks,* ***approvals*** *where necessary*



Security in the Continuous Delivery Pipeline



Planning / Changes / Secure Development

Secure Deployment

Source code analysis and remediation in IDE

Secure and scalable version control

Automated source code analysis, peer review

Audit trail, digital signatures, secrets

Automated API and App testing, dynamic analysis

Service virtualization, Performance Testing,

Runtime protection / pentest, monitoring

Developer

Source Code Tests

Version Control

Build

Artifact Repository

Integration

UAT

Production

Change

Audit Trail

Release

Implementing DevSecOps

DevSecOps Challenges

Culture

- Siloed Dev, Ops & Security teams

Resourcing

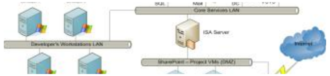
- Limited security personel who can validate or impart knowledge

Domain Knowledge

- Limited knowledge of secure coding in development

Frequent Releases

- Security validation cannot be adhoc - need to continuously validate security through automation

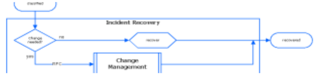


Configuration Drift

- Servers and environments become out of date and inherently insecure

Shadow IT Systems

- Teams choose their own development and deployment tools – makes it difficult to validate security across the enterprise



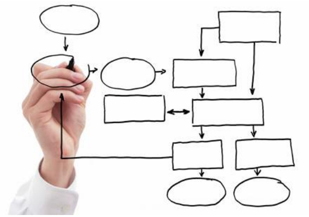
Incident Resolution

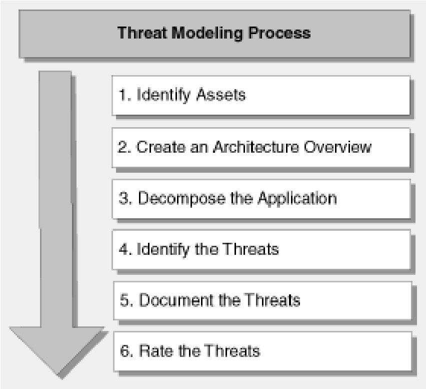
- Slow and manual incident resolutions process does not work with frequent releases

Recommendations

1. Consider security with every change
2. Train developers on the basics of secure coding
3. Embed security into the developer eco-system
4. Isolate code changes on feature branches?
5. Continuously inspect and review vulnerabilities
6. Implement a closed loop security analysis, review and remediation process
7. Decouple release from deployment
8. Harden Continuous Delivery infrastructure
9. Implement immutable infrastructure
10. Adopt a continuous incident response process

Consider security with every change





Threat Modeling Process

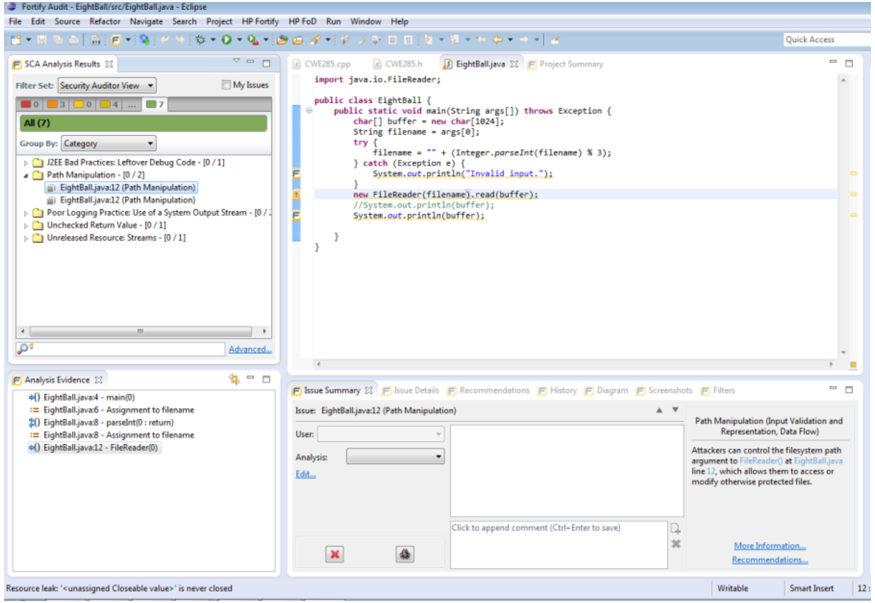
1. Identify Assets
2. Create an Architecture Overview
3. Decompose the Application
4. Identify the Threats
5. Document the Threats
6. Rate the Threats

- Embed security in the planning process:
  - Identify regulatory and compliance requirements
  - Trace throughout the lifecycle
  - Define “Abuse” cases
  - Carry out “Threat Modelling”
- Prioritize security issues
- Ensure a full audit trail of all changes:
  - Plan -> Develop - > Test -> Deploy

Train developers on the basics of secure coding

- How to build and maintain simple “Threat Modelling” scenarios
- Input whitelisting, filtering, sanitization
- SQL injection
- Cross-site scripting
- Cross-site request forgery
- Credential management

Embed security into the developer eco-system



Filter Set: Security Auditor View ☐ My Issues

🟥 0 🟧 3 🟨 0 🟨 4 ... 🟩 7

All (7)

Group By: Category

- J2EE Bad Practices: Leftover Debug Code - [0 / 1]
- Path Manipulation - [0 / 2]
  - EightBall.java:12 (Path Manipulation)
  - EightBall.java:12 (Path Manipulation)
- Poor Logging Practice: Use of a System Output Stream - [0 / 2
- Unchecked Return Value - [0 / 1]
- Unreleased Resource: Streams - [0 / 1]

```
import java.io.FileReader;

public class EightBall {
    public static void main(String args[]) throws Exception {
        char[] buffer = new char[1024];
        String filename = args[0];
        try {
            filename = "" + (Integer.parseInt(filename) % 3);
        } catch (Exception e) {
            System.out.println("Invalid input.");
        }
        new FileReader(filename).read(buffer);
        //System.out.println(buffer);
        System.out.println(buffer);

    }
}
```

- EightBall.java:4 - main(0)
- EightBall.java:6 - Assignment to filename
- EightBall.java:8 - parseInt(0 : return)
- EightBall.java:8 - Assignment to filename
- EightBall.java:12 - FileReader(0)

Issue: EightBall.java:12 (Path Manipulation)

User:

Analysis:

Click to append comment (Ctrl+Enter to save)

Path Manipulation (Input Validation and Representation, Data Flow)

Attackers can control the filesystem path argument to FileReader() at EightBall.java line 12, which allows them to access or modify otherwise protected files.

Resource leak: '<unassigned Closeable value>' is never closed

Writable

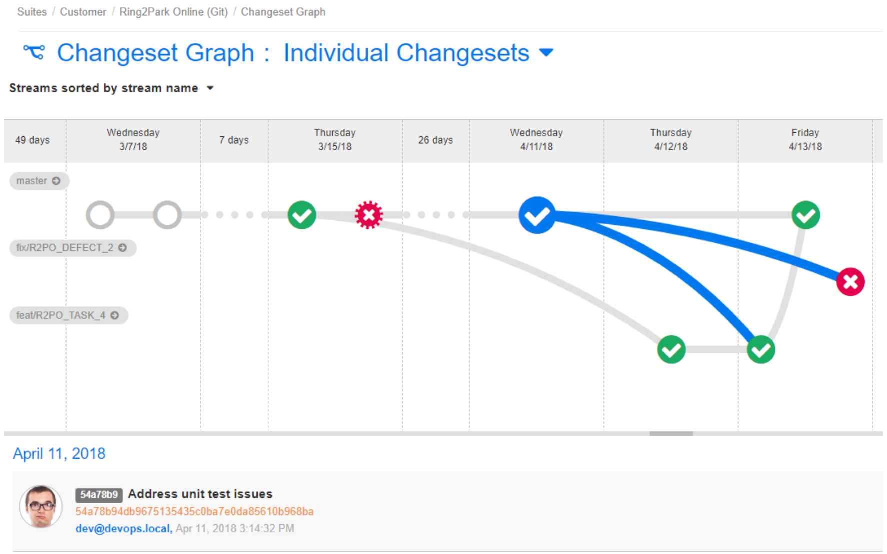
Smart Insert

12 :

- Ensure security tools are embedded in developer IDEs:
  - Invoke analysis tools directly from IDE
  - See security issues assigned and recommendations
  - Remediate issues directly
- Ensure security vulnerabilities are linked to defects
- Invoke security tools automatically as part of CI (build process)

Isolate code changes on feature branches?

- Mature Continuous Delivery works best with trunk based development…
- But branch per feature allows:
  - isolate changed, easier to rollback / remediate security issues
  - only deliver features that have passed security testing
- Make sure feature branches are short-lived (not re-used)!



April 11, 2018

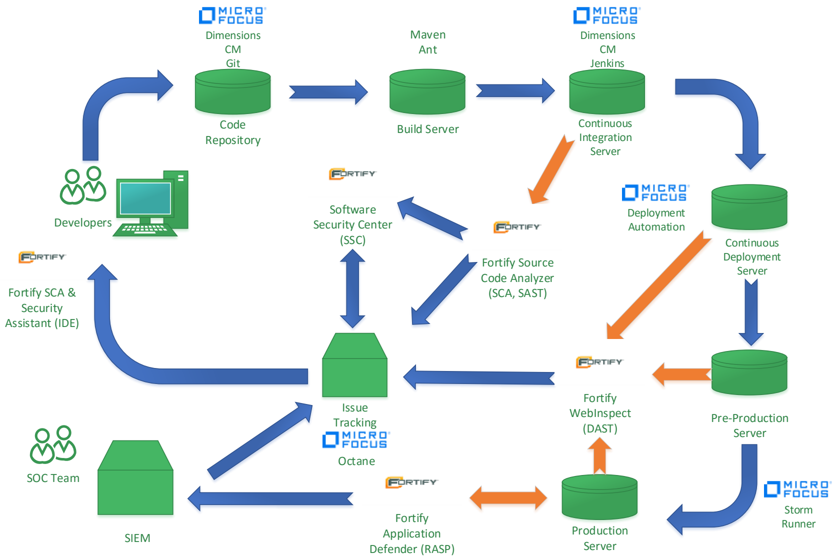
54a78b9 Address unit test issues
54a78b94db9675135435c0ba7e0da85610b968ba
dev@devops.local, Apr 11, 2018 3:14:32 PM

Continuously inspect and review vulnerabilities

- Continuous Inspection
  - Static Analysis
    - Look for security issues in your code
  - 3rd Party Dependencies
    - Identify components used
    - Look for security & quality issues
  - Needs to be automated…

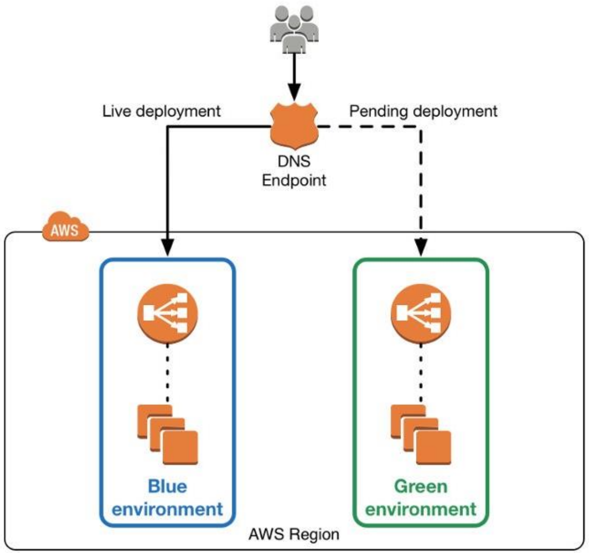
- Vulnerability Review
  - Review every vulnerability
  - Annotate code with security analysis findings
  - Route reviews to the correct experts
  - Full audit trail of the review
  - Link vulnerabilities to Defects

Implement a closed loop security analysis, review and remediation process



- CI tools execute **Static Analysis**
- CD tools execute **Dynamic Analysis**:
- Security findings are centrally reviewed, managed and assigned
- Developers work on security issues in their IDE
- Runtime Application Self-Protection (**RASP**) applied and integrated with SIEM

Decouple Release from Deployment



Rolling Deployments

- New versions deployed onto a limited set of servers to see how they perform
- Typically load balancer points at multiple current versions and one instance of new “canary” release

Blue-Green Deployment

- Running versions of your app in “blue” production environment
- New versions deployed to “green” environment, switched over (via “load balancer” on successful deployment, test ….

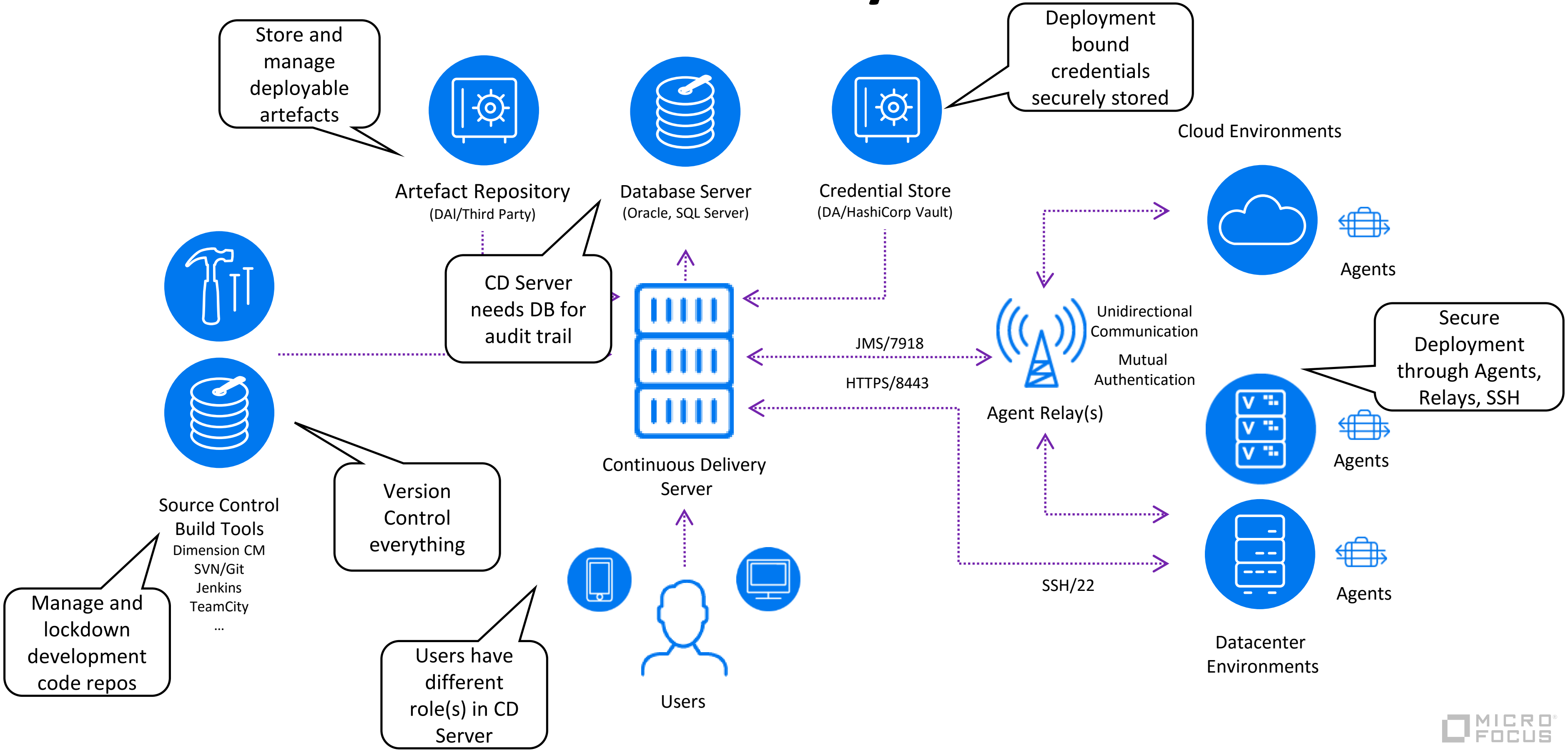
Configuration Updates

- No code deployment, only make changes to configuration files (debugging / feature flags)

Feature Flags

- Turn already deployed features on/off through updates in configuration

Harden Continuous Delivery infrastructure



Store and manage deployable artefacts

Artefact Repository
(DAI/Third Party)

Database Server
(Oracle, SQL Server)

Credential Store
(DA/HashiCorp Vault)

Deployment bound credentials securely stored

Cloud Environments

Agents

CD Server needs DB for audit trail

JMS/7918

HTTPS/8443

Unidirectional Communication

Mutual Authentication

Agent Relay(s)

Secure Deployment through Agents, Relays, SSH

Agents

Continuous Delivery Server

Source Control
Build Tools
Dimension CM
SVN/Git
Jenkins
TeamCity
…

Version Control everything

Manage and lockdown development code repos

SSH/22

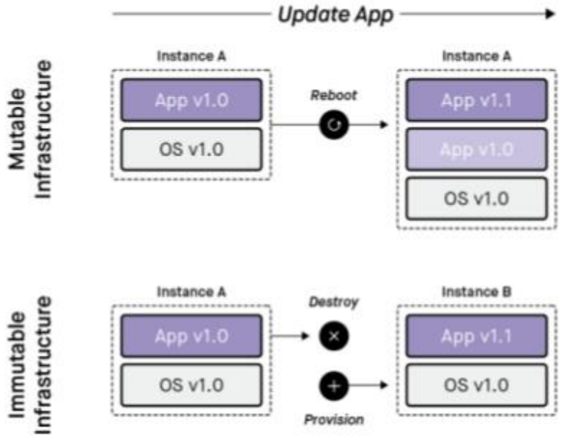
Agents

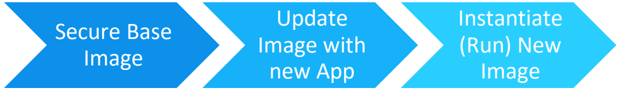
Datacenter Environments

Users have different role(s) in CD Server

Users

Implement immutable infrastructure





Secure Base Image

Update Image with new App

Instantiate (Run) New Image

- Next level on from Blue-Green deployment
  - ~~Deploy onto existing infrastructure!~~
  - Programmatically spin up new servers for each new application deployment
- Guarantees validity / security of infrastructure and remediates configuration drift
- Really needs Infrastructure as a Service platform.
- Can similarly be implemented using containerization (Docker) – but base image needs to be secure.

Adopt a continuous incident response process

Security

Shift your security mindset from “incident response” to “continuous response,” wherein systems are assumed to be compromised and require continuous monitoring and remediation.[2]

- Emergency response process – after the event
- Continuously monitor your Apps and Infrastructure
- Identify vulnerabilities before they happen
- Define an IR plan – update it, test it, run it frequently
- Create actionable alerts – who needs to respond and what action needs to be taken
- Identify vulnerabilities so threats can be dealt with before they become problems
- Automate as much as possible

Micro Focus Solutions

Micro Focus enables DevSecOps at enterprise scale

Plan/Govern

Optimize Value Streams

Develop/Test

Continuous Quality & Security

Deploy/Release

Accelerate Delivery

Operate/Monitor

Increase Service Reliability

Micro Focus Portfolio| End-to-End DevSecOps

PLAN

Project, Portfolio and Requirements

- □ Project & Portfolio Mgmt
- □ Atlas
- □ Caliber
- □ Dimensions RM
- □ Rhythm

Mainframe + COBOL

- □ Enterprise Developer
- □ Visual Cobol

OPERATE

Application and User Monitoring

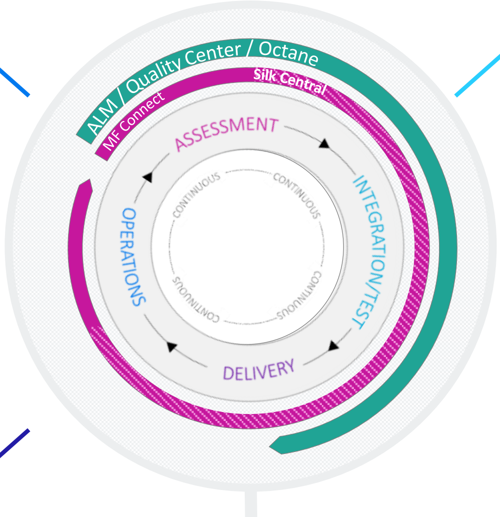
- □ AppPulse
- □ Silk Performance Manager

IT Operations

- □ Hybrid Cloud Automation
- □ Data Center Automation

RELEASE/DEPLOY

- □ Release Control
- □ Deployment Automation



Data Insights

- □ Vertica

BUILD

Software Change & Configuration Mgmt

- □ AccuRev
- □ Dimensions CM
- □ Star Team
- □ PVCS

Mainframe + COBOL

- □ ChangeMan
- □ StarTool
- □ ESync

TEST

Functional Test

- □ UFT
- □ BPT
- □ Sprinter
- □ StormRunner Functional
- □ Silk Test
- □ Silk WebDriver

Security Test

- □ Fortify

Performance Test

- □ LoadRunner
- □ Performance Center
- □ StormRunner Load
- □ Silk Performer

Digital Lab

- □ Mobile Center
- □ Service Virtualization
- □ Network Virtualization

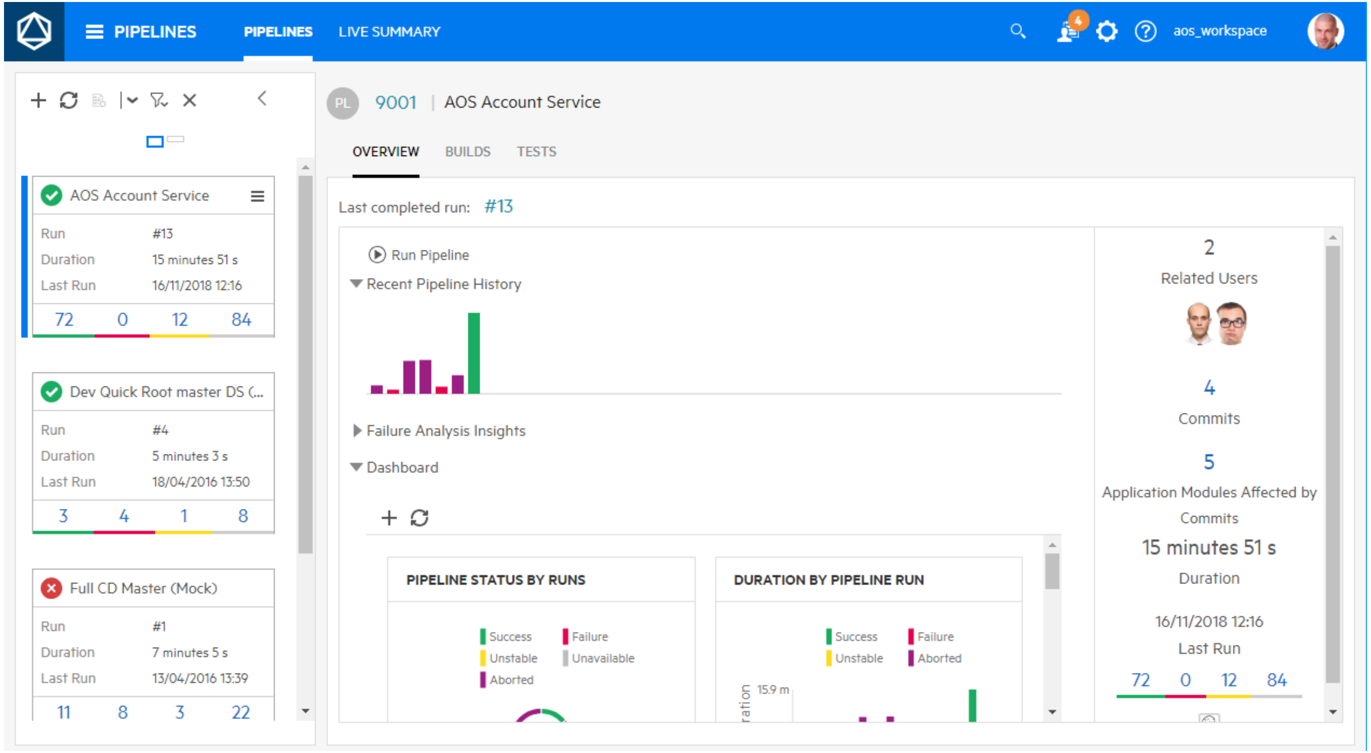
ALM Octane – Development and Test Governance

GOVERNANCE AND TRACEABILITY (COMMERCIAL AND OPEN-SOURCE TOOLSETS)

AGILE AND TRADITIONAL WORK ITEM MANAGEMENT (BACKLOGS ETC)

DEVOPS PIPELINES (INCLUDING COMMITS, CHANGES, TESTS AND SECURITY VULNERABILITIES)

DEEP TEST/QA INTEGRATIONS



AOS Account Service
Run #13
Duration 15 minutes 51 s
Last Run 16/11/2018 12:16
72 0 12 84

Dev Quick Root master DS (...
Run #4
Duration 5 minutes 3 s
Last Run 18/04/2016 13:50
3 4 1 8

Full CD Master (Mock)
Run #1
Duration 7 minutes 5 s
Last Run 13/04/2016 13:39
11 8 3 22

PL 9001 | AOS Account Service

Last completed run: #13

2
Related Users

4
Commits

5
Application Modules Affected by Commits

15 minutes 51 s
Duration

16/11/2018 12:16
Last Run

72 0 12 84

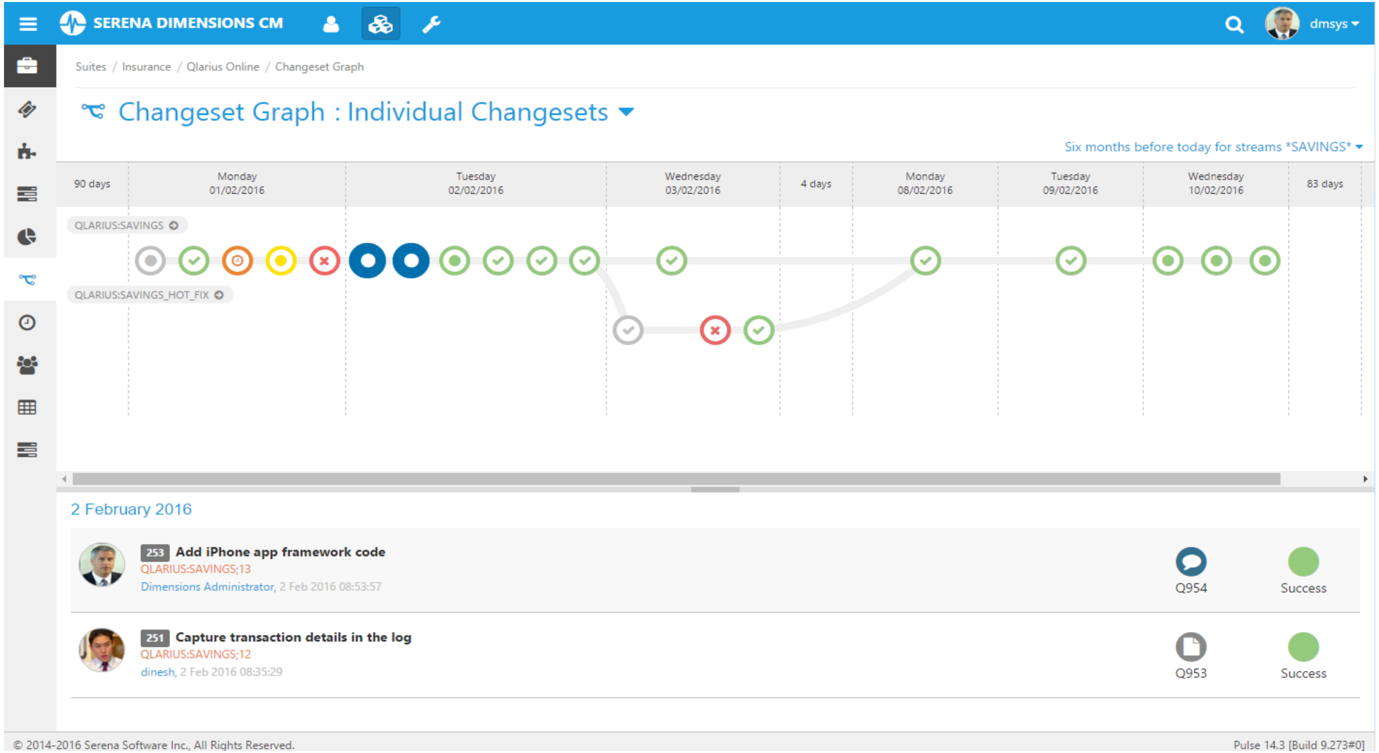
Dimensions CM / AccuRev – Hardened SCM

STREAM BASED APPROACH - CHANGESET VISUALISATION

INTEGRATED CODE REVIEW AND CHANGE MANAGEMENT

ENTERPRISE SCALABILITY AND GRANULAR ROLE BASED SECURITY

NATIVE VERSION CONTROL OR ACT AS GIT REPOSITORY SERVER



Suites / Insurance / Qlarius Online / Changeset Graph

Changeset Graph : Individual Changesets

Six months before today for streams *SAVINGS*

2 February 2016

253 **Add iPhone app framework code**
QLARIUS:SAVINGS;13
Dimensions Administrator, 2 Feb 2016 08:53:57
Q954 Success

251 **Capture transaction details in the log**
QLARIUS:SAVINGS;12
dinesh, 2 Feb 2016 08:35:29
Q953 Success

© 2014-2016 Serena Software Inc., All Rights Reserved.
Pulse 14.3 [Build 9.273#0]

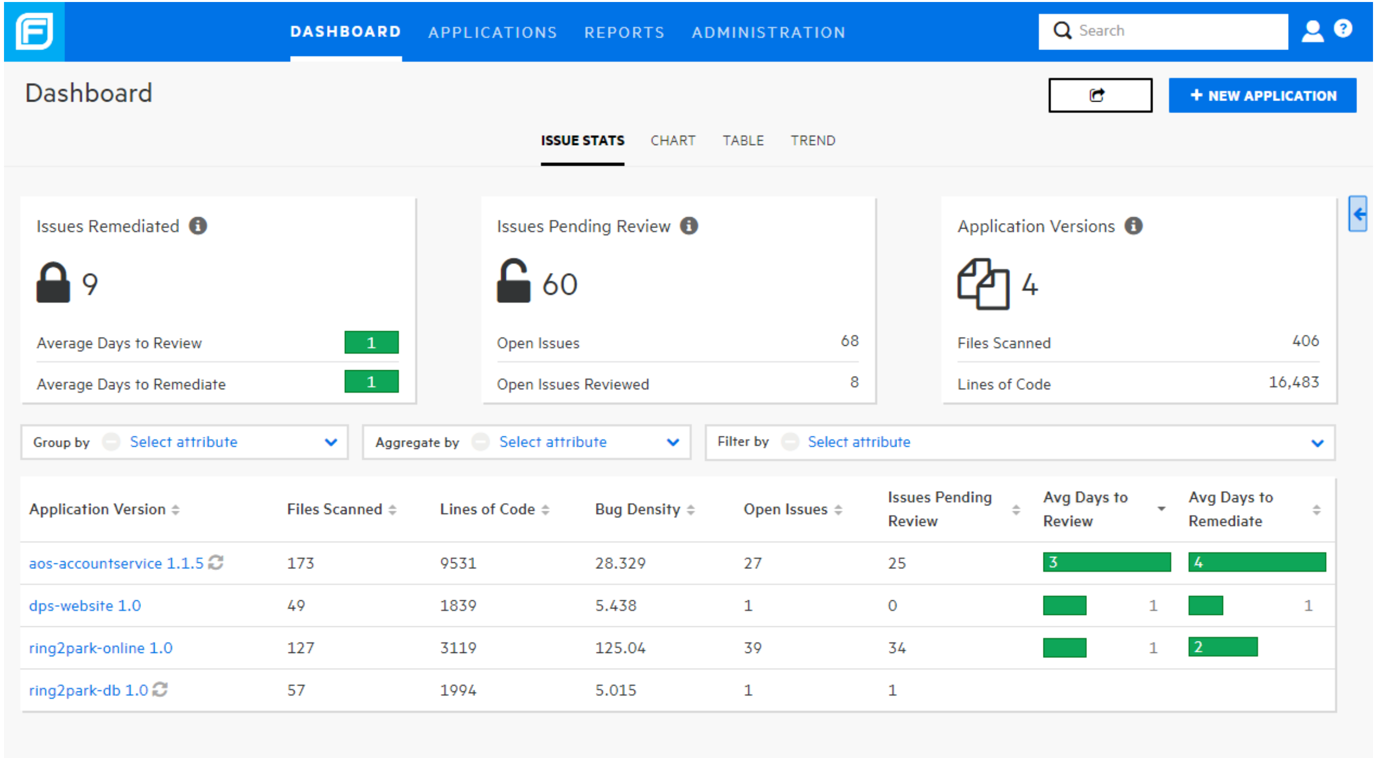
Fortify – Application Security Management

EMBEDS SECURITY INTO DEVELOPMENT ECO-SYSTEM (IDES, BUILD TOOLS ETC)

SUPPORT FOR 25+ PROGRAMMING LANGUAGES

STANDALONE SECURITY CENTER AND/OR EMBEDDED INTO OCTANE

STATIC, DYNAMIC ANALYSIS AND RUNTIME PROTECTION



Dashboard

Issues Remediated

9

Average Days to Review	1
Average Days to Remediate	1

Issues Pending Review

60

Open Issues	68
Open Issues Reviewed	8

Application Versions

4

Files Scanned	406
Lines of Code	16,483

Application Version	Files Scanned	Lines of Code	Bug Density	Open Issues	Issues Pending Review	Avg Days to Review	Avg Days to Remediate
aos-accountservice 1.1.5	173	9531	28.329	27	25	3	4
dps-website 1.0	49	1839	5.438	1	0	1	1
ring2park-online 1.0	127	3119	125.04	39	34	1	2
ring2park-db 1.0	57	1994	5.015	1	1		

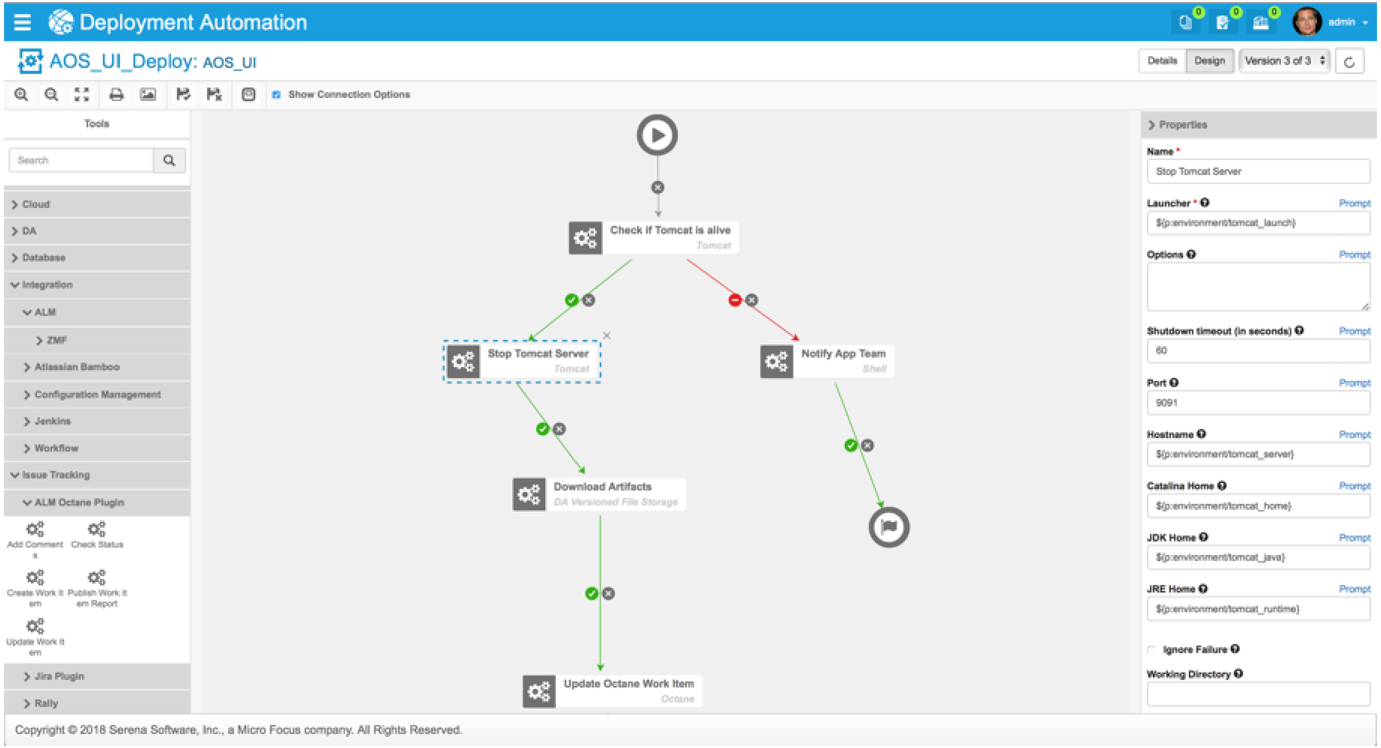
Deployment Automation – Secure Deployment

DRAG-AND-DROP PROCESS DESIGNER

EASILY CREATE, VISUALIZE AND REUSE DEPLOYMENT PROCESSES

SECURE AGENT BASED AND AGENTLESS DEPLOYMENT

ENTERPISE SCALABILITY AND GRANULAR ROLE BASED SECURITY



Name * Stop Tomcat Server

Launcher * Prompt ${p:environment/tomcat_launch}

Options Prompt

Shutdown timeout (in seconds) Prompt 60

Port Prompt 9091

Hostname Prompt ${p:environment/tomcat_server}

Catalina Home Prompt ${p:environment/tomcat_home}

JDK Home Prompt ${p:environment/tomcat_java}

JRE Home Prompt ${p:environment/tomcat_runtime}

□ Ignore Failure

Working Directory

Copyright © 2018 Serena Software, Inc., a Micro Focus company. All Rights Reserved.

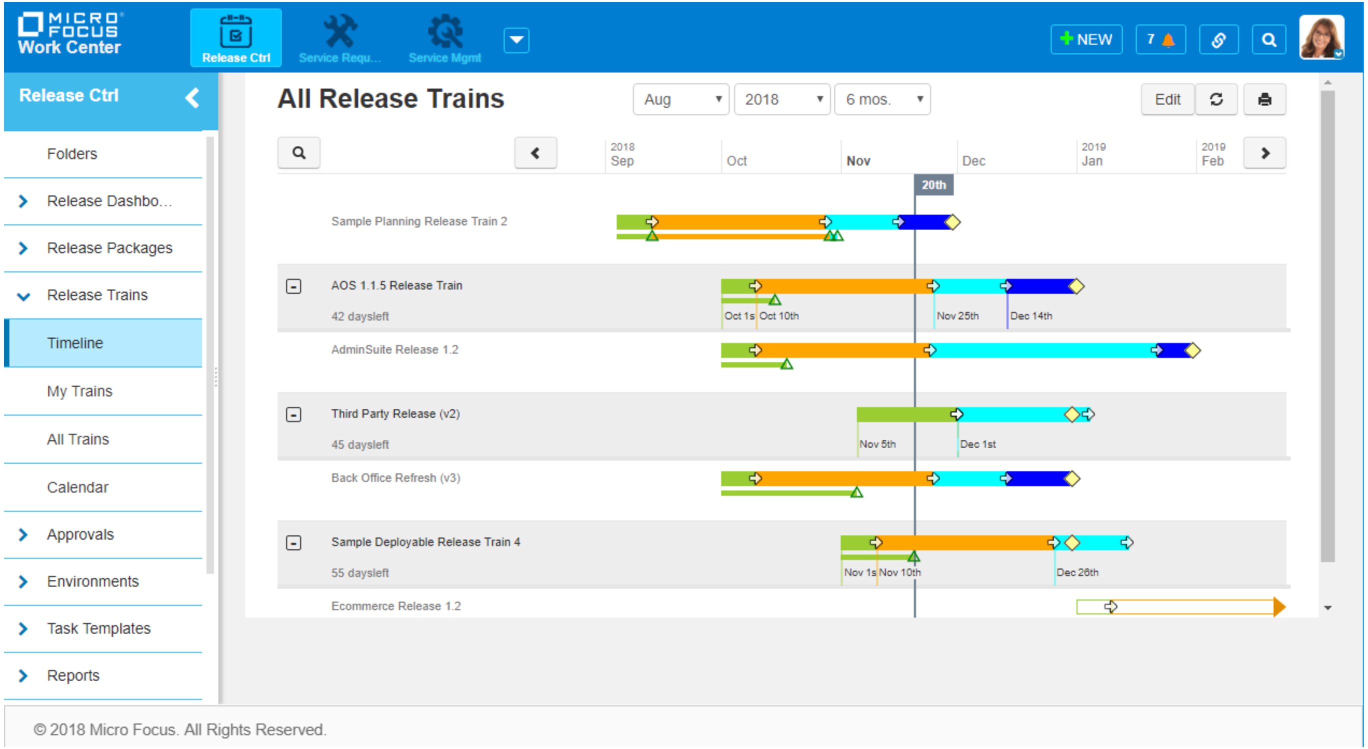
Release Control – Release Governance

VISUALIZE & CUSTOMIZE ALL RELEASE ACTIVITIES SIMPLY AND EASILY

RELEASE PLANNING AND EXECUTION CO-ORDINATION

REDUCES RELEASE RISK BY PROVIDING VISIBILITY OF RELEASE PROCESSES TO ALL AREAS OF YOUR ORGANIZATION

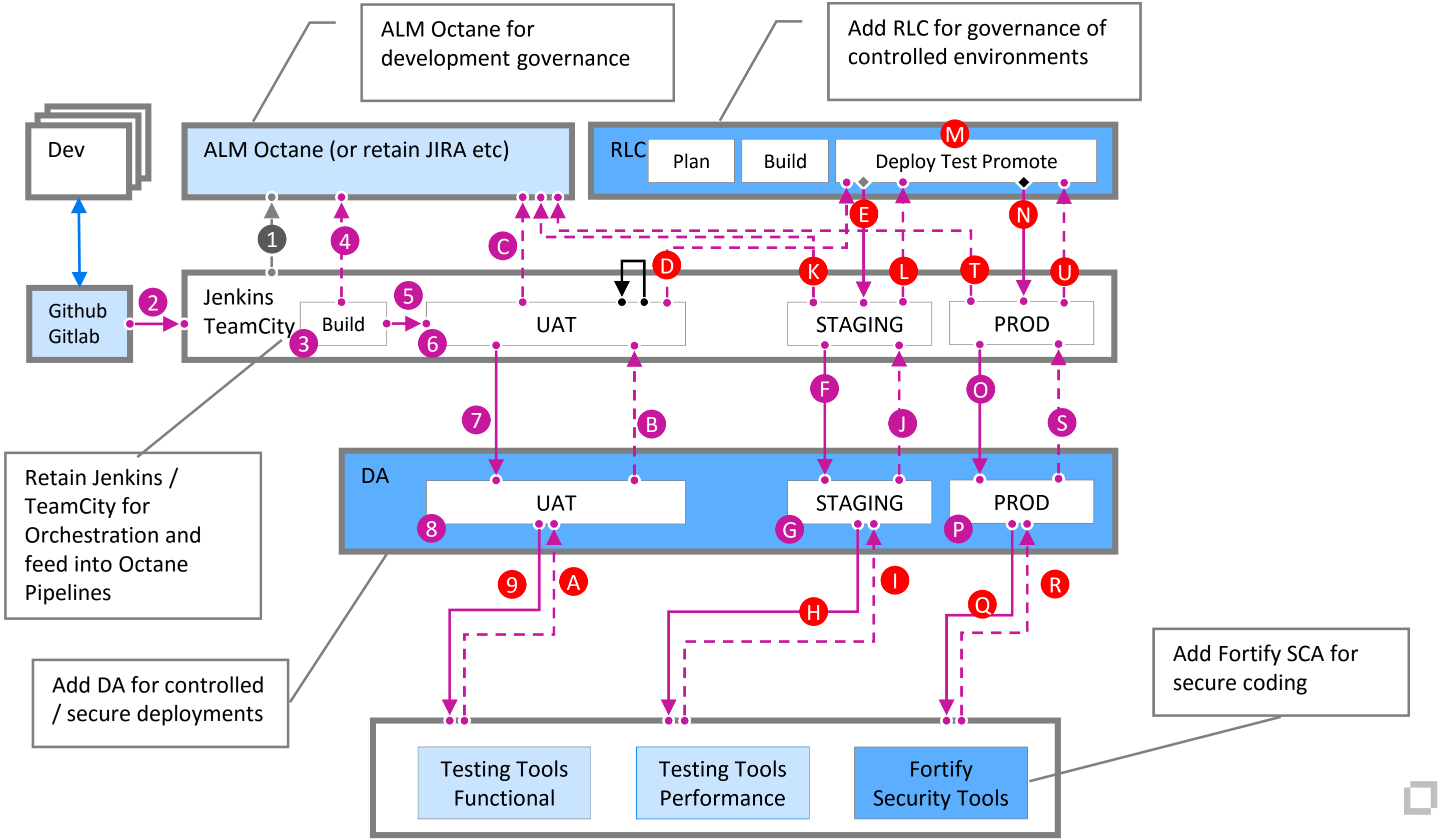
ENSURES AUDIT AND COMPLIANCE BY TRACKING ALL RELEASE ACTIVITIES



Micro Focus Option

DevSecOps Example Integrations

Leverage customers existing portfolio



ALM Octane for development governance
Add RLC for governance of controlled environments
Dev
ALM Octane (or retain JIRA etc)
RLC
Plan
Build
Deploy Test Promote
M
Github Gitlab
Jenkins TeamCity
Build
UAT
STAGING
PROD
1
2
3
4
5
6
C
D
E
K
L
T
N
U
7
B
F
J
O
S
DA
UAT
STAGING
PROD
8
G
P
9
A
H
I
Q
R
Retain Jenkins / TeamCity for Orchestration and feed into Octane Pipelines
Add DA for controlled / secure deployments
Add Fortify SCA for secure coding
Testing Tools Functional
Testing Tools Performance
Fortify Security Tools

Demonstration

Q&A

Kevin A. Lee
kevin.lee@microfocus.com
+44 (0)7799 072507

https://www.linkedin.com/in/kevinalee/
https://akevinlee.github.io/

Thank You!!