Analytics Through The Devops Lifecycle

Andi Mann I Dominic Eger

September 2017 I Washington DC
Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

Splunk, Splunk>, Listen to Your Data, The Engine for Machine Data, Splunk Cloud, Splunk Light and SPL are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2017 Splunk Inc. All rights reserved.
One of the often-cited barriers to DevOps success is visibility across the organization and the plethora of tools. DevOps should break down the department silos. But many tools show their data in a single-tool view, maintaining the silo. Splunk, as a data and analytics platform, provides visibility by collecting and presenting the data from many tools across the DevOps Lifecycle, from planning to dev/test to deploy and monitoring in production. Agile development teams can measure velocity across sprints and releases, finding healthy projects as well as those with hygiene issues, adding to tech debt. With Splunk, you can have end-to-end visibility enabling better collaboration and true data-driven decisions.
DevOps Has A Visibility Problem
The Dev Lifecycle is Complex

No rigid schemas – add in data from any other source.
The Ops Environment is Complex

- Server, Storage, Network
- Server Virtualization
- Operating Systems
- Infrastructure Applications
- Mobile Applications
- Cloud Services
- Custom Applications
- API Services

- SDKs
- UI
- API
- Metrics and Reporting
- Common Data Fabric

- Ticketing/Help Desk
- Other Tools
DevOps is Exponentially More Complex

Plan | Code | Build | Test/QA | Stage | Release | Config | Monitor

Biz | PMO | Dev | Build | QA | Sec | Stage | Ops | Biz

Server, Storage, Network | Server Virtualization | Operating Systems | Infrastructure Applications | Mobile Applications | Cloud Services | Custom Applications | API Services

JIRA | git | Jenkins | AppScan | Amazon Web Services | Octopus Deploy | puppet | New Relic | dynatrace

Visual Studio | Bamboo | Travis CI | sonarqube | Pivotal | Capistrano | Ansible | Chef | dynatrace

CA Technologies | Citrix | Microsoft Hyper-V | Linux | Windows Server Active Directory | iOS | Samsung | Salesforce | .NET

VMware vsphere | Microsoft Windows Server | Citrix XenApp | Windows Server | Android | Amazon Web Services | Amazon Web Services | Java | Python

Emc2 | Amakai | Microsoft | Linux | Windows Server | iOS | Samsung | Salesforce | .NET

Cisco | Citrix | Microsoft | Linux | Windows Server | Android | Amazon Web Services | Amazon Web Services | Java | Python

FedEx | DocuSign | Microsoft | Linux | Windows Server | Android | Amazon Web Services | Amazon Web Services | Java | Python

API Services
“Tell us some of the challenges you have seen customers deal with before they looked to Splunk to solve them”
From every tool, process, or component in Dev.
On-premises, in the cloud, or with 3rd party Ops.
Across diverse teams, activities, and services.
Measuring The DevOps Lifecycle

Metrics that matter for data-driven DevOps decisions
BUT WHAT DATA DRIVES GOOD DEVOPS DECISIONS?

10+ Deploys Per Day: Dev and Ops Cooperation at flickr
John Allspaw & Paul Hammond
Velocity 2009
Sales?
Downloads?
Installs?
Users?

What activities?
What outcomes?
Some DevOps Metrics That *Might* Matter

**Culture**
e.g.
- Retention
- Satisfaction
- Callouts

**Process**
e.g.
- Idea-to-cash
- MTTR
- Deliver time

**Quality**
e.g.
- Tests passed
- Tests failed
- Best/worst

**Systems**
e.g.
- Throughput
- Uptime
- Build times

**Activity**
e.g.
- Commits
- Tests run
- Releases

**Impact**
e.g.
- Signups
- Checkouts
- Revenue
Introducing DevOps Lifecycle Analytics

Using metrics that matter to analyze speed, quality, and business impact
Analytics At Every Phase of The DevOps Lifecycle

- Plan
  - time to deliver
  - idea to cash
  - ROI

- Code
  - code volume
  - commit volume
  - release speed

- Build
  - build speed
  - failure rates
  - manual builds

- Test/QA
  - test volume
  - code coverage
  - exception counts

- Config
  - remediation time
  - code quality
  - access rates

- Stage
  - performance
  - latency
  - scalability

- Release
  - response time
  - uptime/availability
  - resource usage

- Monitor
  - revenue
  - signups
  - cust. sat.
Specific Data For Each Stakeholder

- Biz
- PMO
- Dev
- Build
- QA
- Sec
- Stage
- Ops
- Biz

- process times
- team efficiency
- unplanned work
- build speed
- failure rates
- manual builds
- remediation time
- code quality
- access rates
- response time
- uptime/availability
- resource usage

- time to deliver
- idea to cash
- ROI
- code volume
- commit volume
- release speed
- test volume
- code coverage
- exception counts
- performance
- latency
- scalability
- revenue
- signups
- cust. sat.
Shared Data for Multiple Stakeholders

- process times
- team efficiency
- unplanned work

- build speed
- failure rates
- manual builds

- remediation time
- code quality
- access rates

- response time
- uptime/availability
- resource usage

- time to deliver
- idea to cash
- ROI

- code volume
- commit volume
- release speed

- test volume
- code coverage
- exception counts

- performance
- latency
- scalability

- revenue
- signups
- cust. sat.
Shared Data for Multiple Stakeholders

- process times
- team efficiency
- unplanned work
- build speed
- failure rates
- manual builds
- remediation time
- code quality
- access rates
- response time
- uptime/availability
- resource usage

- time to deliver
- idea to cash
- ROI
- code volume
- commit volume
- release speed
- test volume
- code coverage
- exception counts
- performance
- latency
- scalability
- revenue
- signups
- cust. sat.
Specific Data For Each Stakeholder

- **Biz**: time to deliver, idea to cash, ROI
- **PMO**: code volume, commit volume, release speed
- **Dev**: build speed, failure rates, manual builds
- **Build**: remediation time, code quality, access rates
- **QA**: test volume, code coverage, exception counts
- **Sec**: performance, latency, scalability
- **Stage**: response time, uptime/availability, resource usage
- **Ops**: revenue, signups, cust. sat.
- **Biz**
Specific Data For Each Stakeholder

- **Biz**
  - time to deliver
  - idea to cash
  - ROI

- **PMO**
  - process times
  - team efficiency
  - unplanned work

- **Dev**
  - code volume
  - commit volume
  - release speed

- **Build**
  - build speed
  - failure rates
  - manual builds

- **QA**
  - test volume
  - code coverage
  - exception counts

- **Sec**
  - remediation time
  - code quality
  - access rates

- **Stage**
  - performance
  - latency
  - scalability

- **Ops**
  - response time
  - uptime/availability
  - resource usage

- **Biz**
  - revenue
  - signups
  - cust. sat.
DevOps Lifecycle Analytics: Use Case Examples

Customer use cases and examples of how Splunk addresses key visibility problems
Using Splunk for Resource Analytics

Insight and prediction for effective resource allocation

Key Metrics:
- Work time vs. PTO/sick
- Hours by product/project
- Resource shortages

Data Sources:
- Jira
- WorkDay
Using Splunk for Cost Analytics

Measurement and predictability for cost control

Key Metrics:
- Productive hours
- Labor costs
- Plan vs. actual

Data Sources:
- WorkDay
- PeopleSoft
Using Splunk for DevTeam Analytics

Insight to coder activity for teaming & work/life balance

▶ Key Metrics:
• Commit count
• Commits by author
• Commit days/times

▶ Data Sources:
• GitHub
Using Splunk for Code Analytics

Real-time data on code quality and compliance

Key Metrics:
- Code policy compliance
- Code/file/class complexity
- Code analysis coverage

Data Sources:
- GitHub
- Sonarcube
Using Splunk for Build Analytics

Find and fix build issues to accelerate product lifecycle

▶ Key Metrics:
• Build success/failure
• Build queue status
• Build process times

▶ Data Sources:
• Jenkins
• Sonarcube
Using Splunk for Quality Analytics

Automatically review QA results to improve quality

Key Metrics:
- Defects detected
- Test coverage
- Test executions

Data Sources:
- Selenium
- AppScan
- ServiceNow
Using Splunk for Config Analytics

Monitor provisioning/config to accelerate time to ‘done’

▶ Key Metrics:
  • Provisioning success/failure
  • Provisioning times
  • Config drift by node

▶ Data Sources:
  • Puppet
Using Splunk for Release Analytics

Real-time data for better, faster release decisions

Key Metrics:
- Availability by release
- Tickets by release
- Release readiness

Data Sources:
- ServiceNow
- SonarCube
- HP OpenView
Using Splunk for Impact Analytics

Realtime business insight to drive impactful development

▶ Key Metrics:
- Revenue per min
- Checkout rate
- Cart fulfillment/abandon

▶ Data Sources:
- Web logs
- HTTP events
- SFA/CRM
Using Splunk for Value Stream Analytics

KPIs across the lifecycle for value-stream management

▶ Key Metrics:
  • Cycle time
  • Story completion
  • Deployment rate

▶ Data Sources:
  • Jira
  • Bamboo
  • BitBucket
DOM – Color

“Tell us some of the benefits that your customers have seen from using Splunk to address these use cases”
DOM – Demo (Build Analytics)

“Can you show us a live demo of any of these capabilities?”
Other Use Cases In AppDev and DevOps

- Collaborative IT troubleshooting for Dev and Ops
- Build machine data analytics into applications
- ChatOps and Splunk for war room & post-mortem
- Connect DevOps and security via data (DevSecOps)
- Insight into human systems for cultural metrics
Splunk for DevOps

Enabling collaboration, communication, and integration for DevOps teams making data-driven decisions
Splunk Provides Visibility Across Ops

No rigid schemas – add in data from any other source.
Splunk Provides Visibility Across Dev

No rigid schemas – add in data from any other source.
Splunk Enables Analytics Across The Lifecycle

- Process times
- Team efficiency
- Unplanned work
- Build speed
- Failure rates
- Manual builds
- Remediation time
- Code quality
- Access rates
- Response time
- Uptime/availability
- Resource usage

Plan
- Time to deliver
- Idea to cash
- ROI

Code
- Code volume
- Commit volume
- Release speed

Build
- Test volume
- Code coverage
- Exception counts

Test/QA
- Performance
- Latency
- Scalability

Config
- Revenue
- Signups
- Cust. sat.
DevOps Lifecycle Analytics: Splunk Capability

DevOps Organizational Analytics
- Resource Analytics
- DevTeam Analytics
- Cost Analytics

Application Lifecycle Analytics
- Code Analytics
- Build Analytics
- Config Analytics
- Quality Analytics
- Release Analytics

Value Stream Analytics

Continuous Learning
Continuous Delivery
Organizational Alignment
Agile Team Collaboration
CI-Aligned Task Automation

DevOps Maturity

Business Impact Analytics

TAs, HEC, etc.

Splunk Enterprise
Splunk Cloud

DevOps Lifecycle

JIRA, git, Jenkins, Bamboo, Travis CI, AppScan, Ansible, Puppet, AWS, Pivotal, Octopus Deploy, Electric Cloud, Capistrano, Nagios, etc.
Splunk + Partners Offer Complete Visibility

Accelerate product delivery
Use data from teams, tools, and activities to find bottlenecks in the CI/CD pipeline and eliminate waste

Improve product quality
Understand code quality from QA, pre-prod, staging, and post-release to ensure continuous improvement

Deliver on business goals
Drive continuous innovation by using real-time insight into release impact on business goals to iterate rapidly
Key Benefits of Splunk for DevOps

**Faster Time-to-Market**

Shrink the time it takes to get code through dev/test to market—through faster issue resolution and reduced cycle time.

“Our devs are now able to find and fix issues 5-10 times faster.”

**Agility with Confidence**

Real-time visibility into processes like code check-ins, builds, tests to support continuous integration and continuous delivery.

“We can monitor all the automation and handoffs it takes to deploy 5-10 times a day.”

**Business Insights**

Instrument customer engagement and application usage to capture critical business events, outcomes, and user behavior.

“My code isn’t ready until it’s Splunk-ready.”
Successful Businesses use Splunk for DevOps
Next Steps
How Do You Eat An Elephant?
One bite at a time!
Start with a Build Analytics POC

Find and fix build issues to accelerate product lifecycle

- Start with a pre-defined capability for a proof of concept in a key problem area
- Use your data and systems with a prebuilt, sandboxed Splunk deployment
- Our team onsite to walk through deployment, configuration, and proof of value

E-mail DevOps@splunk.com to get started!
Remember To Check Out These Resources

► Splunk DevOps ecosystem apps: splunkbase.splunk.com
► Splunk blogs: www.splunk.com/blog/tag/devops
► Splunk community: www.splunk.com/community
► DevOps demo available – e-mail DevOps@splunk.com
DevOps
Making machine data accessible, usable and valuable to everyone.
Improve the Impact of Application Delivery

Accelerate Delivery Velocity

Improve Code Quality

Increase Business Impact
Thank You

Don't forget to rate this session in the .conf2017 mobile app