Bringing Continuous Delivery to the DoD and DHS

Gene Gotimer
Coveros, Inc.

All Day DevOps

October 17, 2018

Tale of Two Projects

coveros

U.S. Department of Defense

Exemplar that Agile would work within the DoD

U.S. Department of Homeland Security

Build a Continuous Delivery pipeline for Agile teams

The DoD Project

- 4½ Years, September 2009 – March 2014
- High-risk releases every 6 months or so
  - Freeze 2-4 weeks in advance
  - Deploy Friday evening to Sunday afternoon
  - Repair broken functionality Monday and Tuesday (and on)
- Pre-production environments radically different than Production
  - Different compliance rules
- Development risks
  - No unit tests
  - No continuous integration
  - No automated testing

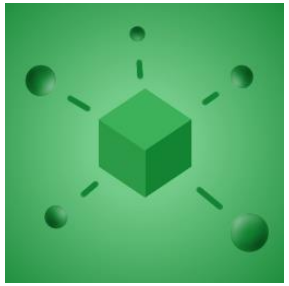
The Approach

- Started with things that were in our control
  - Dev and Test environments
  - Development process
- Made changes behind the scenes
  - Free/open source tools
  - Very few software restrictions
  - Easy to integrate into our CI system
  - Small changes
- Disclose the changes when there was a win
  - “It’s easier to ask forgiveness than it is to get permission” - Admiral Grace Hopper
  - Use as justification for higher environments

Continuous Integration

- Trouble explaining “integration”
  - between two or more developers
  - not between systems
- Set up SecureCI one afternoon

Sonatype

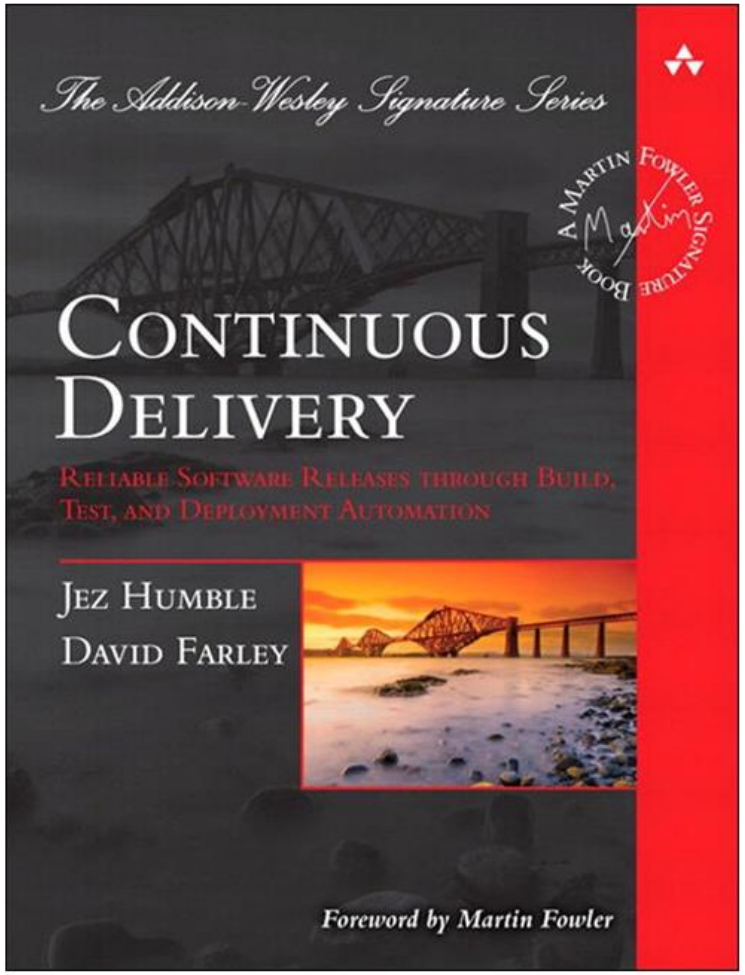


sonarqube

Functional Testing

- Functional testing was done manually
- We waited a year before staging a coup
  - we didn't want to encroach on their domain
- Demo of Selenium
  - demonstrated record-and-playback through the Selenium IDE
  - we recorded the first set of tests
  - then turned it back over to the test team
- They argued later that automated testing was ineffective
  - the automated script (singular) only worked one time
  - needed to be re-recorded when any changes got made to the app

The Book



- Project Manager came across the book in a book store
- Everything made so much sense
- Logical extension of what we were trying to do
- Addressed a lot of the issues we were running into
- No money or time for an effort, so we adopted it as our long-term goal

Automated Deploys

- Started with automating a greenfield web app install
- Then automated the manual COTS install
- Then started reverse engineering the broken COTS installer

Security Testing

- Started running weekly scans of dev and test using OpenVAS
  - no immediate issues, but started seeing package security updates before they became IAVMs

- Implemented OpenSCAP in Jenkins for STIG
  - immediately found issues
  - started adding Puppet manifests for remediation

Culture Clash

- Continuous Delivery was being openly discussed
  - PMO had just started thinking of it as a clear plan
  - Kept asking when “continuous delivery” would be delivered, and how it would be packaged
- Test and Integration started complaining
  - We were pushing them too hard
  - Moving too fast
- People on test and integration team started leaving
  - including “Burt”

The Aftermath

- Releases every 2 weeks
  - Soft freeze Thursday for Friday release
  - Deploy Friday evening
  - 20-minute deploy, practiced dozens of times each sprint
  - 100% working functionality Friday evening
  - Non-event
- Dev and Test environments very closely matched Production
- Development
  - Unit tests
  - Continuous integration
  - Automated testing

The DHS Project

- On-going, March 2018 - present
- Agile development teams
- No automated deploys
- Limited continuous integration
- Working in a government data center and cloud

Continuous Integration

- Teams use Jenkins
  - Primary build is still the button in Visual Studio
- Moving to Jenkinsfile
- Migrating to new infrastructure
  - Exposing many broken, but still passing, builds

Non-Cloud Data Center

- No dynamic provisioning
- Provisioning takes time
  - Create a Visio diagram in a particular format
  - Submit for approval
  - Once approved, provisioning begins
  - Provisioning doesn't necessarily match request
  - Correcting take time, approvals, resubmissions
- From request to ready for use takes days to weeks

Access, but not permission?

- In our integrated test environment, the team
  - Submitted forms to request access
  - Planned to install software using Chef
  - Software was approved and installed in development
  - Coordinated with other teams to provision box
  - Announced on daily status call that we were going to install
  - Installed by us, with other contract watching
- Then
  - We got caught
  - All work stopped
  - More forms submitted, including build instructions
  - Software needs to be installed by other contract, with us watching

The Agile Island

There isn't much a single project can do to change organizational culture, other than show them the right way.

The Agile Island

There isn't much a single project can do to change organizational culture, other than show them the right way.

#Coveros5

Do not set out to do DevOps or CD.
Remove road blocks and bottlenecks.
Fix quality issues.

Adopt change incrementally,
in processes and technology.

Agile teams are better than non-agile teams,
but are no substitute for an agile organization.

Culture is the hardest, and most important, trait to change for DevOps transformation.

There isn't much a single project can do to change organizational culture, other than show them the right way.

Thank You All Day DevOps Sponsors

Platinum Sponsors

Sonatype

cloudbees

Gold Sponsors

puppet

GitLab

XebiaLabs
Enterprise DevOps

GENERAL DYNAMICS
Information Technology

cisco DEVNET
developer.cisco.com

Carnegie Mellon University
Software Engineering Institute

CHEF

SCALED AGILE

Media Sponsors

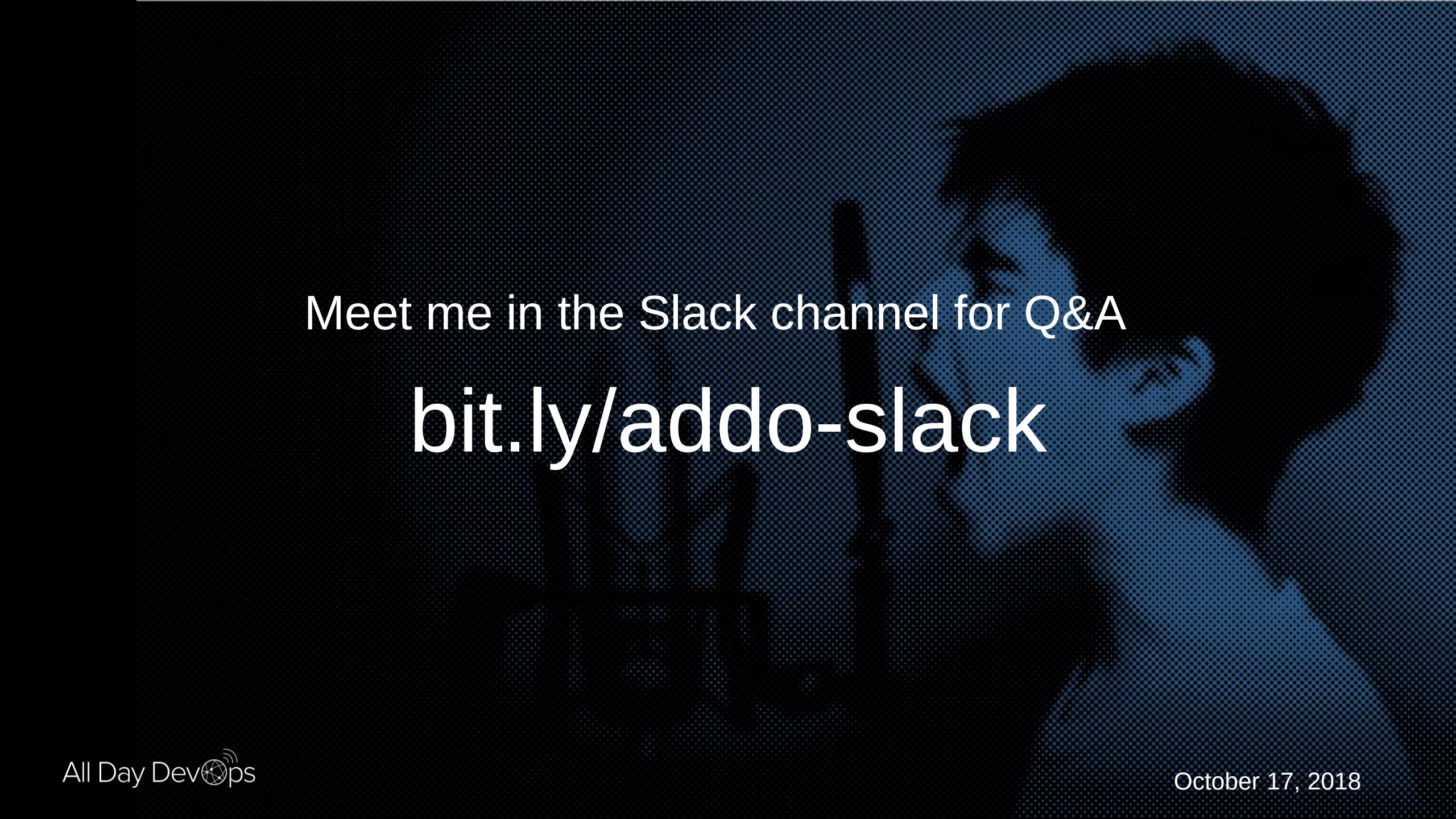
TechBeacon

DevOps.com

DZone

Solutions Review

Thank You All Day DevOps Supporters


Meet me in the Slack channel for Q&A
bit.ly/addo-slack
All Day DevOps
October 17, 2018