Abstract:

How would you like to increase the speed of your server delivery pipeline by 3 ORDERS OF MAGNITUDE? Optum Technology—the technical arm of UnitedHealth Group, one of the largest companies in the United States—did just that. Disruptive technical innovation can be driven in a company of any industry, size and complexity; it is no longer limited to startups and technical companies.

This is a story of a sea change taking place within a gargantuan company—a grassroots effort to leverage a series of open source tools and a question-everything attitude to create massive efficiency gains, leveraging the DevOps philosophy, replacing traditional delivery pipelines with new methods that are already saving millions.

One example that will be discussed is a new server delivery pipeline. This approach:
• Delivers servers in less than 6 minutes,
• Servers are complaint with applicable Regulatory requirements and have the documentation to prove it
• Solution automatically monitors & maintains compliance moving into the future
• Security and compliance are built directly into the delivery pipeline and, best of all…
• There are no Information Security personnel involved in the delivery process

In a complex regulatory environment such as the U.S. healthcare industry, this is no mean feat. Come learn of our journey and how you can take the next step in yours.

#ShiftLeft #DevOps #Chef #Jenkins #InSpec #GauntLt #Optum #UHG #QuinnShamblin #AaronRinehart

Key Takeaways:

• Actions steps you can take to begin a DevOps Security movement within your organization.
• How to make developers excited to be part of the solution, delivering security on behalf of the organization.
• How to get out of the way of your developers and IT operational delivery people
• The names of a few important tools that can leveraged to improve your delivery and security by orders of magnitude.
How long does it take for your team to deliver a new virtual server?
A day? A week? Longer?
How long would it take to deliver 100? 1000?
What if they all have to be compliant with HIPAA, PCI, NIST, etc.?

- How would you like to deliver one in 6 minutes?
- AND improve security at the same time
- AND be able to prove your compliance
- AND cost less

My name is Quinn Shamblin
My background includes 5 years as the CISO of Boston University, CISO of University of Cincinnati, time with HP, P&G and as an Officer in the U.S. Navy.
You can find me pretty easily on LinkedIn if you search for “Quinn Shamblin”. I usually come up on the first page.

- So why are we here?
- (Slide Read)

...So we aren’t going to tell you what it is. 😊
- It isn’t a specific series of steps you take
- It is a cultural shift to focus human resources on things that humans can bring value too and leverage technology to bring the value it can bring
- It does however have a few core ideas and techniques
- These can be found in two important works, which, if you have not encountered them yet, you should carve out time to absorb
• For those of you that read “The Goal” by Eli Goldrat, which revolutionized manufacturing in the 80s,
• Like “The Goal”, this book is also a novelization, and applies the lessons of “The Goal”, making them accessible to those of us in IT
• To those of us that have been in IT for awhile, the first 6 chapters are a painful reminder of just how bad things can be
• But fight your way through them; there are some really important lessons after the authors set the stage
• I highly recommend the audio book version vs the traditional book. Will help you fight through the painful parts

• The Phoenix Project is a novelization, this is the actual handbook to talk you through how to lead this revolution

TPP provides 12 key takeaways. Here are a few highlights’

• Pillars – Development and operations work better when they are considered two halves of the same team, not competitors
• Bottleneck – WIP is the enemy of efficient progress.
  • Controlling WIP means making work visible so you find the most limiting resource, the constraint
  • The more utilized a particular resource is, the longer tasks will wait in a queue before receiving hands-on attention. These wait times multiply when the test is handed off between workstations multiple times.
  • Improving efficiency anywhere but the constraint is a waste of time
• Gears – The Ultimate goal of making work visible is to understand it, make it replicable, understand how to best alleviate the constraint and, where possible, automate the work

• So... DevOps sounds like an IT initiative, what does it have to do with Security?
This is where “Rugged DevOps” or DevOps Sec comes in to play.

Just as Dev and Ops need to be thought of as being on the same team, so too does Security.

Developers should be responsible for delivering security along with the other functions

And in this new world, it is often possible to remove security personnel from the delivery pipeline altogether

Refocusing their efforts on working with developers to create systems to allow developers to easily deliver security along with capability

A common comment on DevOps is that “surely this is something that is only realistic for startups or tech companies?”

We did not find that to be true...

...as long as you take the lessons from those that have been successful

Developers are understanding the power of Continuous Delivery,

but security techniques of the past (mostly waterfall) cannot keep pace.

Our goal was to try to find tools and techniques that would enable the developers to move at the new speed of business, while not sacrificing security.

But we *are* a ridiculously large company.

Actually, we are a bit like the Borg, “Your market segmentation and technological distinctiveness will be added to our own.”

We are a collective of hundreds of companies, with incredibly diverse technical capabilities and maturity levels

Operating in a highly regulated field

...so how could we begin to tackle this issue in a company so large and complex? How could be possibly get funding for an effort so vast? ...

...We didn’t

We took a lesson from start-ups and worked directly with developers to solve one issue for one small group and grow from there

We showed an open mind and enthusiasm for solving issues that had been plaguing the company

There was skepticism at first, but we put our money where our mouth was so to speak

We had to build trust that we really would work to change policy to support

Reason for Success #1: We didn’t ask executives for money, we asked for a little time from a few passionate people
**OUR PRINCIPLES FOR DevOps SECURITY**

- Shall prefer Open Source solutions
- Shall support automation
- Shall provide self-service
- Shall be flexible to continuous iterative change
- Shall perform baseline and health-check validation
- Shall be able to execute in a lean and rapid manner
- Shall be standardized
- Shall be easy to learn
- Shall be API driven
- Shall be able to monitor and prevent accidental disruptive behavior

**Reason for Success #2:** We didn't need money in order to build value, because we favored open source solutions

As we were beginning our journey, we worked up a set of principles

Principles intended to guide us toward speed and scalability

(There are other principles frameworks as well.)

**Reason for Success #3:** We focused our efforts on pain points that people were passionate about

Where were our pain points?

Some of these efforts succeeded, some failed

- For some of the ones that failed, we will try again

**FAILURES AND SUCCESSES**

- One of our core goals was to change the way people interact with Security and fit what *they* are doing.
- One example: We need to enable compliance at speed and scale

**COMPLIANCE AS CODE WITH CHEF + INSPEC**

- Include deployment, security, and compliance configuration at speed and scale
- Get security (people) out of the pipeline.
- Security is embedded into the delivery pipeline through security-approved compliance cookbooks
- Compliance is immediately confirmed prior to release for use.
- Compliance is an integral part of server build and configuration, instead of a late “add-on”, “shifting it left” on the deployment timeline
- This doesn’t replace vulnerability scans. But Servers will largely be compliant at provisioning, which can later be verified with formal scans
- Reduces the end-of-project surprises and project risks with the old way
- This does require upfront investment to create solid compliance cookbooks. We expect that to be someone’s full-time job.
- An experienced chef developer with knowledge of the IaaS or PaaS service under development might expect between 80-120 hours upfront

**Chef + InSpec: Automated Security Configuration & Validation at Speed**

*Case Study*

State Health Exchange

**Reason for Success #3:** We focused our efforts on pain points that people were passionate about

- Where were our pain points?
- Some of these efforts succeeded, some failed
  
  - For some of the ones that failed, we will try again

**BUILD IT TOGETHER AND THEY WILL COME**

- Target things people are passionate about

  - Begin with six core DevOps Security Problem Sets:
    - Security baseline + configuration validation w/ Chef & Inspec
    - Automated State + Application + Security + Compliance
    - Automated Application Vulnerability Scans in Jenkins via API
    - Automated Static Code Analysis in Jenkins via API
    - DevOps Self-Governance & Operationalization Framework

**FAILURES AND SUCCESSES**

- One of our core goals was to change the way people interact with Security and fit what *they* are doing.
- One example: We need to enable compliance at speed and scale

**Chef + InSpec: Automated Security Configuration & Validation at Speed**

*Case Study*

State Health Exchange

**Reason for Success #3:** We focused our efforts on pain points that people were passionate about

- Where were our pain points?
- Some of these efforts succeeded, some failed

  - For some of the ones that failed, we will try again

**FAILURES AND SUCCESSES**

- One of our core goals was to change the way people interact with Security and fit what *they* are doing.
- One example: We need to enable compliance at speed and scale

**COMPLIANCE AS CODE WITH CHEF + INSPEC**

- Include deployment, security, and compliance configuration at speed and scale
- Get security (people) out of the pipeline.
- Security is embedded into the delivery pipeline through security-approved compliance cookbooks
- Compliance is immediately confirmed prior to release for use.
- Compliance is an integral part of server build and configuration, instead of a late “add-on”, “shifting it left” on the deployment timeline
- This doesn’t replace vulnerability scans. But Servers will largely be compliant at provisioning, which can later be verified with formal scans
- Reduces the end-of-project surprises and project risks with the old way
- This does require upfront investment to create solid compliance cookbooks. We expect that to be someone’s full-time job.
- An experienced chef developer with knowledge of the IaaS or PaaS service under development might expect between 80-120 hours upfront
- Compliance experience:

- One State Health Exchange was facing a migration from their data center to ours.
  - SEs stood up all required servers (300+) over a weekend via service catalog requests.
  - We then waited weeks to get extra build services to customize requirements that were required for that State beyond what the catalog would offer.
  - SecurityBlanket was to be used for compliance, but we were running into license complications which were taking weeks to resolve, so we allowed app team and middleware teams to jump in and configure hosts.
  - After a couple of months total time, we then applied compliance rules via SecurityBlanket which took a couple weeks to get installed and executed on all 300+ hosts across environments.
  - We found issues, so we then went through a series of remediation tickets and rework to get systems compliant.
  - We lost tremendous time in the coordination between teams who each needed to restart systems through their work.
  - This started in early March for host requests and then it was mid July when the production systems were compliant and ready for application deployment and testing.

- Next we wanted a self-service application vulnerability scanning solution that could be integrated into the Jenkins build pipeline

- Gauntlt is an open source application vulnerability scanner aggregator engine that allows for multiple vulnerability security scanning tools (like SSLyze, Arachni, Garmr, Nmap, CURL, more) to be packaged in a way to allow developers to easily create security checks against their applications.

- We currently have ours configured with six open source security scanning tools. We plan to add more

- So we have built a lightweight, yet comprehensive scan,

- Again, right now it does fully replace other techniques, but it enables developers to detect and fix issues early—while still actively developing—so that later formal scans come back clean.

- Which will be used in tandem with other programs to generate metrics based on the security of the application and to help bring applications up to compliance

- What do you notice about both of the above two successes?

- Reason for Success #4: We are building rewards into the new processes (1. Speed. 2. You don’t have to talk to us, you can move at your own pace.)

- Another Reasons for Success: We start small with just the teams that were involved in our effort. Their colleagues are seeing their success with the new technique and they are flocking to the new techniques. We—security—are not driving or funding anything.
• Part of fail fast
• Teasing out the Grand Unification Theory will delay the small, important wins—and associated groundswell—you could be building

• Our process and tool failures are actively harming their productivity
• The WANT to help us be better
• Yes, there are cowboys out there that just want us out of the way, but there are also plenty of them that are more mature and want to do things with quality.

• Each of our efforts that were successful, were successful because of the small number of passionate individuals involved that wanted it solved.

• Don’t automate bad processes
• Simplify and streamline your processes first, then automate
Embrace Failure as a Friend

Plan and expect failure as a positive outcome.
Encourage teams to fail quickly and learn from them.

- This helps you understand peoples priorities
- Rid yourselves of people that are not producing, let them drop back to their regular roles
- Refocus the producers—the passionate ones—on other efforts that are seeing traction

WHAT’S NEXT

KEY TAKEAWAYS

- DevOps is not a fad, it is the future—IT is a culture shift, not just about automation
- Principles
  - Drive out complexity. Complex things don't scale
  - Avoid analysis paralysis
  - Fail small, fail fast
  - Let humans focus on where they add value; automate everything else
- Reasons for success
  - We didn’t ask for money, we asked for passionate people
  - We didn’t need money in order to build value, because we favored open source solutions
  - We focused our efforts on pain points that people were passionate about
  - We built rewards into the new processes

Comment for 5 (i.e. Netflix Security Monkey)

Q&A

Another great lead is:
Aaron Rinehart
aaron.rinehart@optum.com
https://www.linkedin.com/in/aaronsrinehart