

O'REILLY®

Velocity

WEB OPERATIONS & PERFORMANCE

17-19 NOVEMBER 2014  
BARCELONA, SPAIN

[velocityconf.com](http://velocityconf.com)  
#velocityconf

# Running Cross Functional Service Teams

Philip Reynolds  
Workday

# About me

- Got the internet bug at 14 ... discovered Linux
- Have worn many hats – mainly as a developer & sysadmin
- Joined Workday in 2009
- Manage 2 teams in Workday

# Preface

- I didn't invent this stuff – AmaGooBook
- Shared learning
  - Nothing revolutionary. Not going to change your world.
- My team – they're awesome

# What?

- Mix of developers & operations
- “You build it, you run it” – Amazon
- Reporting structure not that important. Team == “Scrum team”

# Why?

## ■ Process

- Redundancy vs dependency
- Seek to minimize external dependencies
- Scale – Organizational scalability. Easy model to replicate.

# Why? (contd.)

## ■ People

- Drive deeper understanding of the system – grow true domain experts
- Autonomy of teams. Allow teams the freedom to make decisions.
- Cross training
- Team – strength through unity
- Resourcing

# Why? (contd.)

- Technical

- Architecture - Move to Service Orientated architecture
- Promote certain systems to “first class” systems. Having service teams own things like monitoring.

# How? (Ops team perspective)

- Hire a **senior** developer (... a good one!)
  - Lean on your colleagues for help in hiring one or look internally for candidates.
- Treat all code as code.
  - Python – code
  - Application configuration – that's code
  - Switch configuration – that's code too!
- Build dev infrastructure (build & test infrastructure / tooling)



# How? (Ops team perspective #2)

- Create a pipeline
- Rigorous code reviews
- Heavy focus on testing. Invest in your testing.
- Continuous Integration – The Holy Grail

# How? (Ops team perspective #3)

- Train the developer on supporting systems.
  - Litmus test: can they go on-call
- Developers like building stuff. Make sure you recognise that.

# How? (Dev team perspective)

- Hire a **senior** ops person
- Set expectations around code
  - Litmus test: can they triage and fix small bugs?
- Dopamine hit comes from fixing stuff

# How? (Dev team perspective #2)

- Set clear responsibilities
  - Business hours triaging? Config mgmt? Monitoring?
- Operability is a functional requirement
- Everyone on-call

# Hiring

- Culture extremely important – don't compromise
- Soft skills / personality is more important for first CF hire
- You almost certainly have colleagues who can help hire

# Learnings

- Dynamic of team has changed. Develop a rhythm
- You will spend more time talking about work instead of doing work. That's ok.
- Velocity increased – go slower to get faster.
  - Defect rate should go down.
  - Less rework

# Learnings #2

- Collective ownership of the domain is key. Collaboration should have increased (desk time goes up)
- Started using traditional scrum - moved to Kanban.
  - Critical part for ops teams is limiting WIP and breaking down work
  - Critical part for dev teams is facilitating reactive work
- Ops people are often more optimistic in planning(!)

# Learnings #3

- Developers are now second-line on call.
  - Not everyone is ok with that. Be up-front through hiring process



# Summary

- Own your service. All of your service
- Invest in quality
- Team first

# We're hiring!

@philreynolds

We're hiring - [www.workday.com/careers/](http://www.workday.com/careers/)