

# The History of DevOps And what you need to do about it

#### **Damon Edwards**

Managing Partner DTO Solutions, Inc.





#### **Damon Edwards**



# Control Con

(http://twitter.com/damonedwards) Link to slides is here



## I get to see a lot...





# Dev ←-→















Why isn't there conferences were Dev and Ops gets together to talk about their problems?



#### October 30 - 31, 2009: Ghent, Belgium



















































cc 🌣





#### The Rise of a New IT Operations Support Model

By 2015, DevOps will evolve from a niche strategy employed by large cloud providers into a mainstream strategy employed by 20% of Global 2000 organizations.

#### Why DevOps will not emerge:

- Cultural changes are the hardest to implement, and DevOps requires a significant rethinking of IT operations conventional wisdom.
- There is a large body of work with respect to ITIL and other best practices frameworks that is already accepted within the industry.
- Open source (OSS) management tools, which are more aligned with this approach, have not seen significant enterprise market share traction.

#### Why DevOps will emerge:

- DevOps is not usually driven from the top down and, thus, may be more easily accepted by IT operations teams.
- ITIL and other best practices frameworks are acknowledged to have not delivered on their goals, enabling IT organizations to look for new models.
- The growing interest in tools such as Chef, Puppet, etc., will help stimulate demand for OSS-based management



#### March 18, 2011



Statistics show that software projects are not living up to the challenge. Up to 50% of apps released to production are later rolled back.\* 51% of projects are over budget and often lack critical features. \*\* 60 – 80% of the cost of software development is rework. \*\*\*

Request a quote

Or call us at: 1-800-728-1212 Priority code: 109HG03W

Find out how PMC or

down to two.

administrators to roll out a single new application - they

between development, QA, and operations, ensuring

Improve auditing and governance by detecting and

remediating changes and demonstrating effective

worry-free deployments

# Why is the history important?

- It reminds us that DevOps is...
  - from practitioners, by practitioners
  - not a product, specification, job title
  - an experience-based movement
  - decentralized and open to all



**DevOps Days** Ghent 2009 Sydney 2010 Mountain View 2010 Hamburg 2010 São Paulo 2010 Boston 2011 Mountain View 2011 Melbourne 2011 Bangalore 2011 Gothenburg 2011 Manilla 2011 Austin 2012 Tokyo 2012 Mountain View 2012 +62 more (and 19 in 2016)



Patrick has handed off organizing leadership to group of community members

# Culture Automation **Neasurement** Sharing

A cultural and professional movement, focused on how we build and operate high velocity organizations, born from the experiences of its practitioners.



# What have we learned?

### **Recent IT Performance Data is Compelling**

High performers compared to their peers...

**30x** more frequent deployments

**60x** the change success rate

**2x** more likely to exceed profitability, market share & productivity goals **200x** faster lead times

**168x** faster mean time to recover (MTTR)

**50%** higher market capitalization growth over 3 years\*



puppet labs



#### **Recent IT Performance Data is Compelling**

High performers compared to their peers...

**30x** more frequent deployments

**60x** the change success rate

**2x** more likely to exceed profitability, market share & productivity goals **200x** faster lead times

**168x** faster mean time to recover (MTTR)

ly to higher market capitalization growth over 3 years\*

4



Faster

Higher

Quality





SOLUTIONS

Data from 2014/2015 State of DevOps Report - https://puppetlabs.com/2015-devops-report

#### **Conventional Wisdom**





#### Faster, Better, and Cheaper?



#### Faster, Better, and Cheaper?



## Faster when labor is our largest cost



Assumptions: Both groups are 30 people at \$100/hr; backlogs are identical



#### Faster, Better, and Cheaper?



#### Faster, Better, and Cheaper?



"In the Lean community, there is a deeply held belief that lead time is one of the best predictors of quality, customer satisfaction and employee happiness. And what we found in our benchmarking work of over 20,000 IT professionals is that it is absolutely true for the technology value stream, as well."





"The data shows that throughput and stability metrics move in tandem — effectively not supporting ITIL claims that tradeoffs should be made in throughput in order to get stability. The pattern of needing to trade throughput for stability simply doesn't appear in the data."



# How about some anecdotal proof?



## But let's not talk unicorns...





#### Let's talk horses





# Nordstrom











#### Why does this work?

#### **Fast Feedback**



#### **Benefits of focusing on Fast Feedback**

• Errors are caught sooner when they are cheaper to fix (less rework)



#### **Benefits of focusing on Fast Feedback**

 Errors are caught sooner when they are cheaper to fix (less rework)





#### **Benefits of focusing on Fast Feedback**

- Errors are caught sooner when they are cheaper to fix (less rework)
- Encourages working in smaller batches (less simultaneous moving parts.. quick diagnosis and fix)
- Encourages getting really good at getting through the lifecycle (lower the "transaction" cost)
- Organization learns faster at all levels ("good at getting better")

Faster feedback leads to... higher quality, less rework, shorter cycle times, and lower costs.



# But how do we actually do that?



# Simple System vs

- Can be understood by reducing it to a set of simpler parts
- Can have complete information about it
- Can perfectly predict behavior and consequences of changes

# **Complex System**

- Can't be understood by reducing it to a set of simpler understandable parts
- Can't have complete / prefect information about it
- Can't perfectly predict behavior and consequences of changes



#### Your organization is a complex system<sup>2</sup>



#### Complex System

interacting with a

Complex System



#### Silos are the #1 enemy of throughput and quality



**Operational Knowledge** 



**Business Intent** 

#### Your organization is a complex system<sup>2</sup>



#### Complex System

interacting with a

Complex System



#### **The "Big Bang" Transformation Dream**





#### The "Big Bang" Transformation Reality





#### Your organization is a complex system<sup>2</sup>



#### Complex System

interacting with a

Complex System



#### Replace the "Big J" with "Little J's"





#### Your organization is a complex system<sup>2</sup>

#### Complex System

#### The only way to fix a sufficiently complex system is to create the conditions for the system to fix itself.



Complex System



# You need an Improvement System

- Not for your technology
- Not to build individual skills
- ... but for how your organization works

SOLUTION



#### Make it Repeatable and Sustainable

Improvement System is going to have to...

- Scale quickly
- •Span multiple organizational boundaries
- Keep improvement efforts aligned
- •Work with substantial numbers of legacy technologies
- Develop existing staff in mass
- Be self-funding after initial seed investment



#### Be self-funding after initial seed investment



#### Example: "DevOps Kaizen"





#### Provides a mechanism to introduce new ideas



#### **Improvement System**



#### **Examples of DevOps Patterns**



**Continuous Delivery** 



#### **Testing in Production**



**Ops as a Service** 



**Infrastructure as Code** 



**Everything in SDLC** 





**Blameless Post Mortems** 



**Prod-like Dev Environments** 

#### **Recommended Reading**









#### Want to talk Improvement Systems or DevOps?



# damon@dtosolutions.com @damonedwards Link to slides is here

