Scaling Devops
Breaking Down the Barriers between Development and IT Operations

October 10, 2011, GOTO Århus

Jez Humble, ThoughtWorks Studios
@jezhumble #continuousdelivery
http://continuousdelivery.com/

http://thoughtworks-studios.com/
“The main obstacles to improved business responsiveness are slow decision-making, conflicting departmental goals and priorities, risk-averse cultures and silo-based information.”

Economist Intelligence Unit: “Organisational agility: How business can survive and thrive in turbulent times”
"the enterprise"

Business

Engineering

Operations

Value stream

Project C

Project A

Project B

DBAs

Infrastructure team

Service desk
Let's create a new product
Let's create a new product.

Value stream:

- Project A
- Project B
- Project C

Enterprise projects:

Business

Engineering

Operations

- DBAs
- Service desk
- Infrastructure team
Let's create a new product

Business

Project C
Project B

Engineering

Project A

Value stream

Operations

DBAs
Service desk
Infrastructure team
Let's create a new product

Business

Engineering

Operations

Project C

Project A

Project B

DBAs

Service desk

Infrastructure team

Value stream
Let's create a new product.
Let's create a new product.

Value stream
We’re going agile!
We’re going agile!

Oh crap!

Business

Project D

Project A

Project B

Engineering

Value stream

Operations

DBAs

Service desk

Infrastructure team

Oh crap!
Project A

Project B

Value stream

Business

Engineering

Operations

Project D

DBAs

Service desk

Infrastructure team
Our test-driven code follows SOLID principles.

Value stream
Our test-driven code follows SOLID principles.

Shame it doesn’t work.

Value stream
Our test-driven code follows SOLID principles.

Shame it doesn't work.

Change management.

Value stream.
culture

- ops involved in inceptions, showcases, retrospectives
- devs go to weekly ops stand-ups
- devs rotate through ops
- devs carry pagers
- organize your incentives (Hawthorne effect)
automation

• build, deploy, test release (deployment pipeline)
• provisioning & management of infrastructure and environments (infrastructure-as-code)
• database migrations and deployments
measurement

- business metrics - revenue, # orders, # users
- ops metrics - changes, incidents, TTD, TTRS, TBF
- technical metrics - TPS, response time, hits
- root cause analysis - which changes break stuff?
sharing

• celebrate success together (one team)
• knowledge
• tools and techniques (bdd, refactoring, continuous integration)
continuous delivery

developers write production-ready code
continuous delivery

developers write production-ready code

everyone collaborates throughout lifecycle
continuous delivery

developers write production-ready code

everyone collaborates throughout lifecycle

faster feedback loops
continuous delivery

developers write production-ready code

everyone collaborates throughout lifecycle

faster feedback loops

lower-risk, more reliable releases
continuous delivery

developers write production-ready code

everyone collaborates throughout lifecycle

faster feedback loops

lower-risk, more reliable releases

done means released
Far more than 50% of functionality in software is rarely or never used. These aren’t just marginally valued features; many are no-value features.

The Standish Group, reported in the IEEE conference 2002
“if we do not know who the customer is, we do not know what quality is”
build quality in

“Cease dependence on mass inspection to achieve quality. Improve the process and build quality into the product in the first place”

W. Edwards Deming
but

ops still has all that legacy crap
but

ops still has all that legacy crap

no way to rationally decide what to retire
but

ops still has all that legacy crap

no way to rationally decide what to retire

deciding what to do is command-and-control
but

ops still has all that legacy crap

no way to rationally decide what to retire

deciding what to do is command-and-control

people who can make the changes aren’t the ones feeling the pain
You build it,
You run it
Treat these services as if they were products
think product

Products / Services

Operations

Value stream

- Products / Services
- Operations
- Value stream

- Ping!
- Ops management
- Service desk
- IaaS
therefore...

teams can measure cost and value delivered per product
teams can measure cost and value delivered per product

teams can rationally determine what to prioritize in their product backlog
teams can measure cost and value delivered per product

teams can rationally determine what to prioritize in their product backlog

teams can self-manage using the lean startup methodology

dependently...
teams can self-manage using the lean startup methodology.

Therefore...

teams can measure cost and value delivered per product.
teams can rationally determine what to prioritize in their product backlog.
teams can self-manage using the lean startup methodology.

architecture group responsible for regulating and monitoring system-level attributes.
teams can self-manage using the lean startup methodology.

Therefore...

teams can measure cost and value delivered per product.
teams can rationally determine what to prioritize in their product backlog.
teams can self-manage using the lean startup methodology.
architecture group responsible for regulating and monitoring system-level attributes.
PMO regulates and measures to prevent “market failure” and manage externalities.
organizational change

architecture group
organizational change

architecture group

PMO
organizational change

architecture group

PMO

operations
organizational change

architecture group

PMO

operations

business
enterprise governance

risk management

SOX, ITIL, COBIT
enterprise governance

risk management

SOX, ITIL, COBIT

segregation of duties
enterprise governance

risk management

SOX, ITIL, COBIT

segregation of duties

change management
enterprise governance

risk management

SOX, ITIL, COBIT

segregation of duties

change management

auditing and compliance
questions

jez@thoughtworks.com
http://continuousdelivery.com/

@jezhumble #continuousdelivery