From 2½ days to 2½ seconds - the birth of DevOps

Dan North
@tastapod

Once upon a time...

Once upon a time...

back in the mists of 2005

Once upon a time...

back in the mists of 2005

before Continuous Delivery™

Once upon a time...

back in the mists of 2005

before Continuous Delivery™

a team was stuck.

Once upon a time...

back in the mists of 2005

before Continuous Delivery™

a team was stuck.

This is their story.

Life was tough back then

Life was tough back then

2½ days to deploy a build

Life was tough back then

2½ days to deploy a build

Testers were frustrated!

Life was tough back then

2½ days to deploy a build

Testers were frustrated!

Developers were frustrated!

Life was tough back then

2½ days to deploy a build

Testers were frustrated!

Developers were frustrated!

We were bottlenecked at testing :(

Life was tough back then

2½ days to deploy a build

Testers were frustrated!

Developers were frustrated!

We were bottlenecked at testing :(

actually bottlenecked at deployment!

The causes were obvious

The causes were obvious

"snowflake" environments

The causes were obvious

"snowflake" environments

too many moving parts

The causes were obvious

"snowflake" environments

too many moving parts

multiple teams

The causes were obvious

"snowflake" environments

too many moving parts

multiple teams

contention for resources

Where do we start?

Where do we start?

"Elevate the constraint"

Where do we start?

"Elevate the constraint"

Engage Operations as customer

Where do we start?

"Elevate the constraint"

Engage Operations as customer

Understand the manual steps

Where do we start?

"Elevate the constraint"

Engage Operations as customer

Understand the manual steps

Identify where time is being sunk

Where do we start?

"Elevate the constraint"

Engage Operations as customer

Understand the manual steps

Identify where time is being sunk

Focus on the outcome, not the steps

Idea: Treat the container as code

Idea: Treat the container as code

WebLogic is just XML

Idea: Treat the container as code

WebLogic is just XML

so create templates

Idea: Treat the container as code

WebLogic is just XML

so create templates

and version-control them!

Idea: Treat deployment as code

Idea: Treat deployment as code

Instrument the UI tools

Idea: Treat deployment as code

Instrument the UI tools

Learn the command-line tools

Idea: Build the app as a single entity

Idea: Build the app as a single entity

Single Deployable Artefact

Idea: Make all the servers look the same

Idea: Make all the servers look the same

- Same OS

Idea: Make all the servers look the same

Same OS

Same packages

Idea: Make all the servers look the same

Same OS

Same packages

Same versions

Idea: Make all the servers look the same

Same OS

Same packages

Same versions

Same(ish) settings

Idea: Automate the entire deployment

Idea: Automate the entire deployment

Conan the Deployer!

Conan the Deployer

Conan the Deployer

- Create a new container instance

Conan the Deployer

- Create a new container instance
- Generate container config

Conan the Deployer

- Create a new container instance
- Generate container config
- Deploy container config

Conan the Deployer

- Create a new container instance
- Generate container config
- Deploy container config
- Bring up the master node

Conan the Deployer

- Create a new container instance
- Generate container config
- Deploy container config
- Bring up the master node
- Deploy the app into the node

Conan the Deployer

- Create a new container instance
- Generate container config
- Deploy container config
- Bring up the master node
- Deploy the app into the node
- Bring up the app

Conan the Deployer

- Create a new container instance
- Generate container config
- Deploy container config
- Bring up the master node
- Deploy the app into the node
- Bring up the app
- Smoke test the environment

This was the big breakthrough

This was the big breakthrough

Any build into any environment

This was the big breakthrough

Any build into any environment

2½ days down to 25 minutes!

This was the big breakthrough

Any build into any environment

2½ days down to 25 minutes!

Deterministically.

Then optimise...

Then optimise...

Blue-Green Deployments

Then optimise...

Blue-Green Deployments

Create self-serve deployments

Then optimise...

Blue-Green Deployments

Create self-serve deployments

Sub-second cutover, during office hours!

What did we learn?

What did we learn?

Focus on the outcome

What did we learn?

Focus on the outcome

Focus on the current bottleneck

What did we learn?

Focus on the outcome

Focus on the current bottleneck

You don't "have an investment" in hardware

What did we learn?

Focus on the outcome

Focus on the current bottleneck

You don't "have an investment" in hardware

DevOps requires collaboration

Epilogue

Epilogue

Sam Newman develops DbDeploy

Epilogue

Newman develops DbDeplo

Jez Humble and Chris Read (and me :)
describe the Build Production Line

Epilogue

Newman develops DbDeploy

ible and Chris Read (and me :)
the Build Production Line

Chris Read co-creates DevOps Days

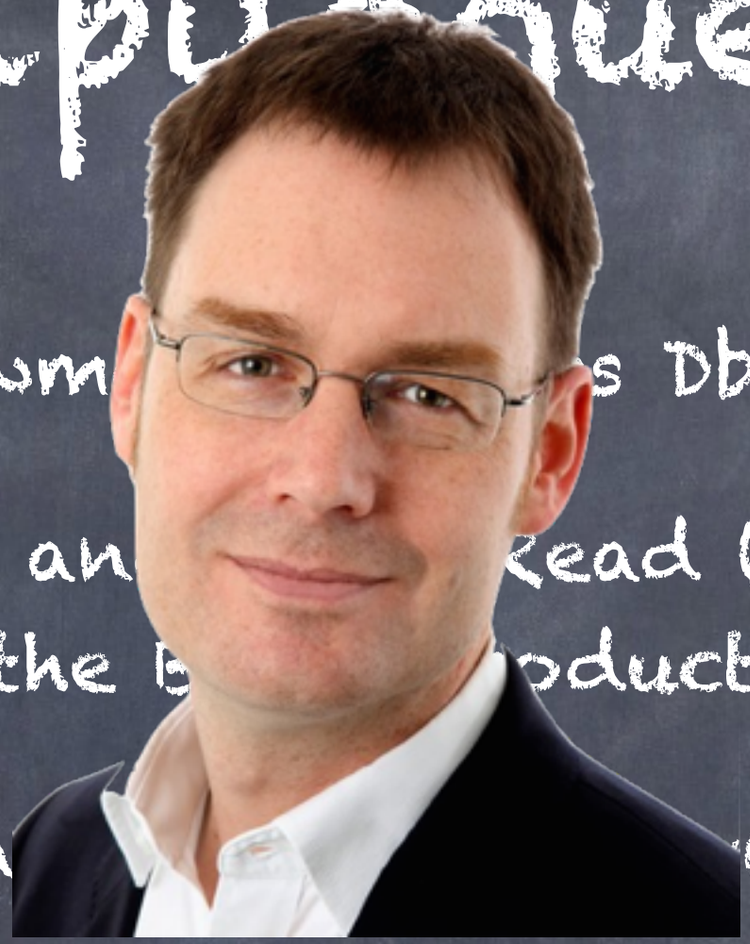
Epilogue

Sam Newm s DbDeploy

Jez Humble an ead (and me :)
describe the P oduction Line

Chris Rea Ops Days

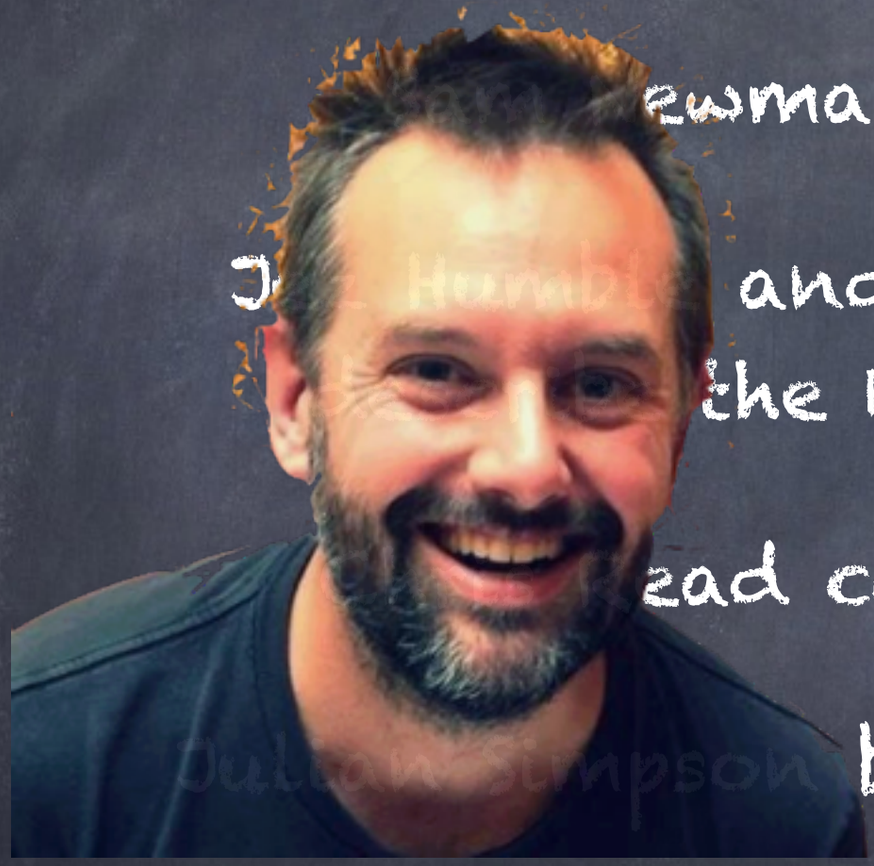
Julian Simpson becomes the Build Doctor



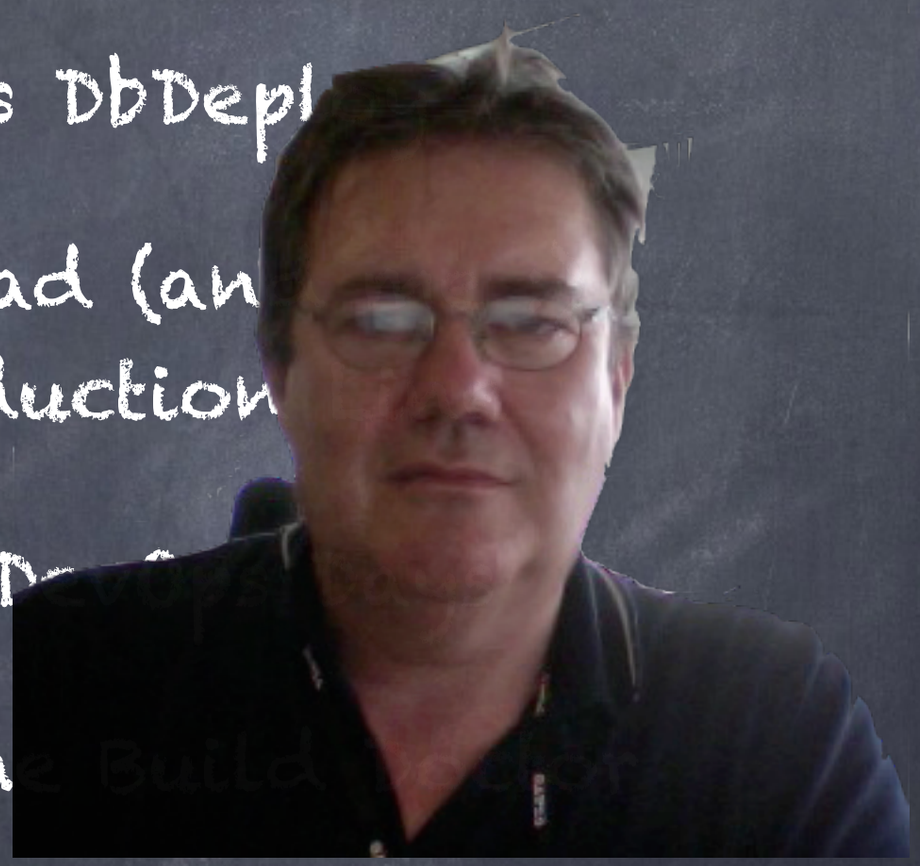
Epilogue

ewman develops DbDepl

J and Chris Read (an
the Build Production



ead co-creates D



Julian Simpson becomes the Build

Jez Humble and Dave Farley write "Continuous Delivery" and win Jolt Award

Whither DevOps?

Whither DevOps?

Commoditise All the Things!

Whither DevOps?

Commoditise All the Things!

Cloud All the Things!

Whither DevOps?

Commoditise All the Things!

Cloud All the Things!

Hug a SysAdmin today

Whither DevOps?

Commoditise All the Things!

Cloud All the Things!

Hug a SysAdmin today

And remember: A "build team" is still an anti-pattern!

Thanks for listening

@tastapod

dan@dannorth.net

http://dannorth.net

Don't forget to vote!