

Continuous Delivery

by example .NET

Oktober 2012



Trifork in brief

- Trifork is a software development company
- Founded 1996, listed on NASDAQ OMX
- 270 employees in located in Aarhus,
 Copenhagen, Zürich, London, Amsterdam,
 San Francisco, Stockholm, Krakow, Beijing
- 3 focus areas: Mobile, Agile, Cloud
- Focus sectors:

Finance, Manufacturing, Telecom Health, Government





Anders Hejlsberg Dan North Rick Falkvinge Damian Conway Scott Hanselman Martin Fowler

Jutta Eckstein Michael Nygard Martin Thompson Chris Anderson Adrian Cockcroft Sam Newman Linda Rising Steve Freeman Brian Cantrill Kasper Lund Andy Gross Gabrielle Benefield Simon Brown Steve Vinoski Randy Bias Jez Humble Jonas Boner Dion Almaer Ben Galbraith Brian LeRoux

> Follow us on Twitter @GOTOcon www.gotocon/aarhus-2012

Jasper Arildslund

- Software Pilot in Trifork since 2010
 Scrummaster, teamlead and software developer
- Microsoft development since 2001
- Agile development since 2001

Goal

- Principles important!
- Methods a few.
- Practices/Tools .NET



TRIFORK.

• Focus??

Survey

- Dev, Ops, Manager, etc.?
- Continuous Integration Knowledge?
- Continuous Integration Experience?
- Continuous Delivery Knowledge?
- Continuous Delivery Experience?



Agenda

- My perspective on Continuous Delivery
- Continuous Delivery (~20 slides)
 - DevOps
 - Deployment strategies
 - Deployment pipeline
 - Agile Testing
 - Environment
 - Design for Continuous Delivery
 - Tools & Automation
 - Metrics
 - Version Control System
 - Artifacts
 - Deploying Databases
 - Managing Continuous Delivery
- Continuous Delivery by examples (.NET)

Agile perspective



Continuous Integration Principles

- Maintain a Single Source Repository.
- Automate the Build
- Make Your Build Self-Testing
- Everyone Commits To the Mainline Every Day
- Every Commit Should Build the Mainline on an Integration Machine
- Keep the Build Fast
- Everyone can see what's happening
- Test in a Clone of the Production Environment
- Make it Easy for Anyone to Get the Latest Executable
- Automate Deployment

Martin Fowler, 2006

TRIFORK.

٠

Quote

"Det bedste der er sket det sidste år er, at releases kan ske direkte fra byggeserveren."

Images from FreeDigitalPhotos.net



Credits

Continuous Delivery

The Addison Wesley Signature Series

RELIABLE SOFTWARE RELEASES THROUGH BUILD TEST, AND DEPLOYMENT AUTOMATION

Jez Humble David Farley



+

Foreword by Martin Fowler

Continuous Delivery

reduce the cost, time, and risk

of delivering incremental changes

to users

TRIFORK,

Continuous Delivery Principles

- Create a Repeatable, Reliable Release Process
- Automate Almost Everything
- Keep Everything in Version Control
- Bring the Pain Forward
- Build Quality In
- Done Means Released
- Everybody Is Responsible for the Delivery Process

DevOps – Agile operations

Everybody Is Responsible for the Delivery Process, but..

Incentives for IT-operations?

Where is IT-operations in the organization?

The biggest challenge in Continuous Delivery

The art of the possible

DevOps - example

Ascio (Speednames)

Miljøstyrelsen

DSB

Finansiel Stabilitet



Design/Architecture for CD

- Enabling small incremental releases
 - Decompose Your Systems
 - Model Your Domain
 - Get Interfaces Right
- · Feature toggle and 'branch by abstraction'
 - Dependency Injection

Deployment strategies

- Seperate Deployment From Release
 - Smoketest
 - Rollback
 - Self-service deployments and promoting
 - Monitoring
 - Zero downtime releases
 - Decoupling releases
 - Installers?
 User-Installed software
 - Hotfixes?

Deployment Pipeline

- A repeatable, reliable and predictable release process
- Model your value stream and create a walking skeleton.
- Automate
 - the build and deployment process.
 - unit tests and code analysis.
 - acceptance tests.
 - releases.

Deployment Pipeline - example



Deployment Pipeline - principles

- Only Build Your Binaries Once!
- Deploy the Same Way to Every Environment
- Smoke-Test Your Deployments (Fail early)
- Deploy into a Copy of Production
- Each Change Should Propagate through the Pipeline Instantly
- If Any Part of the Pipeline Fails, Stop the Line (Fail fast)

Agile testing

Agile Testing

The Addison Wesley Signature Series

A PRACTICAL GUIDE FOR TESTERS AND AGILE TEAMS

LISA CRISPIN JANET GREGORY

The state

Forewords by Mike Cohn and Brian Marick

The Agile Testing Quadrants



Buildserver

Task scheduler





Visualizer

Images from FreeDigitalPhotos.net

Environment - Deploy the Same Way to Every Environment

Virtualization

Cloud - Environment as a Service

Infrastructure

Platform

Software

Define the environment for your project

It should be possible to have an environment on a single laptop

Tools

Principles over methods over Tools (practices)

Go for simple tools

... or go for the mainstream tools

Command line rules!



Automation

- If it hurts do it more
- If doing it more automate it
- Don't automate directly in the buildserver!

Managing infrastructure

- Test Driven
- Chef, Puppet, Vagrant etc.



TRIFORK.

And much more....

Metrics

.

•

Capture and visulization

First part of an immune system

- Build process metrics in the buildserver Code quality
 - **Test results**
 - Unit
 - Accept
 - Performance

Runtime metrics

- Infrastructure
 - Application

Version Control System

- All resources in VCS, including
 - Configuration
 - Build and deployment scripts
- Versioning

•

.

Branch alternatives
 Branch by abstraction

Artifacts

There can only be one!

- build Your Binaries Once!

• Put them in an artifact repository

Components and dependencies

Deploying databases

- SQL metadata and data artifacts
- Test data
- Incremental deployment
 - Fælles Medicin Kort

Managing Continuous Delivery

- Risk management
- Maturity Model
- ITIL

Continuous Delivery summary

- Create a Repeatable, Reliable Release Process
- Automate Almost Everything
- Keep Everything in Version Control
- Bring the Pain Forward
- Build Quality In
- Done Means Released
- Everybody Is Responsible for the Delivery Process

Pause

Examples and inspiration

- **Build Server** •
- Artifact repository •
- Tools
- Deplo atabases
 - **Environment**
 - Infrastructure code •
 - **Metrics** •

Buildserver - example



Environment - example

- Simple environment
 - User account
 - Files
 - IIS (7) website/App-pool
 - Windows service
 - MS SQL Server (2005+) Database
- Azure

•

- https://manage.windowsazure.com
- **Command line tools!**
 - Virtual server

Metrics - example



Hybrid Cloud Environments



Tools won't do it alone

Principles over methods over Tools

But tools are nice

Will get you far:

- Mercurial (Git)
- Jenkins (TeamCity)
- MSBuild with extensions
- Nunit
- Log4Net (NLog)
- Zip and command line-install

Will maybe get you further:

- Nuget (http://docs.nuget.org/docs/creating-packages/hosting-your-own-nuget-feeds)
- MSI-install
- Migrator.NET
- Powershell with extensions
- Tools for agile testing (e.g. SpecFlow & Watin)
- OctupusDeploy (Web Deploy, Chef/Puppet, Vagrant)
- Cloud (Azure, Cloudify)
- Loggr, Loggly and New Relic(Splunk/Logstash/Graylog2/Nagios)

Questions

?

mailto:jba@trifork.com