Continous Delivery ist keine Technology

Marco Kisperth, Jörg Müller

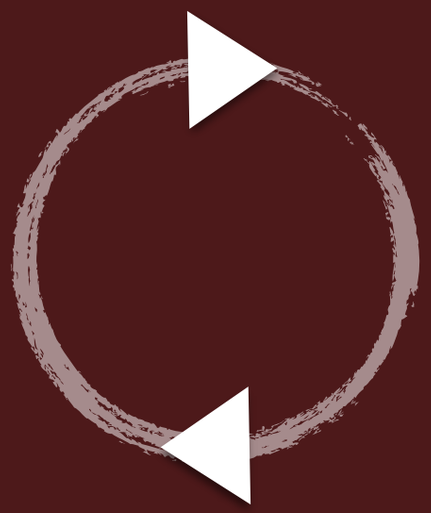
Berlin Expert Days 2014

Marco Kisperth
der Product Owner
@kisperth

Jörg Müller
der Techniker
@joergm

Unser Product Owner hatte eine Vision!

Update der Anwendung mindestens
einmal pro Stunde möglich



Features werden während der Umsetzung
laufend ausgerollt

Ein System für Produktion, Akzeptanz,
Preview und Schulung

Wir hatten erstmal gewaltige Zweifel!

„**Jeder Commit** automatisch auf Produktion!“

Wer bezahlt den Aufwand für die automatisierte **Testabdeckung**?

Wie soll so ein automatisches
Deployment funktionieren?

Jetzt sollen **If-Bedingungen** für neue Features in den Code?

Warum wollte der Product Owner so etwas?

Nur ein **genutztes Feature**
ist ein gutes Feature

Keine **goldenen Wasserhähne**

Features **in kleinen Schritten** zur Verfügung stellen

Weniger Komplexität

Mehr **Zufriedenheit**

Was mussten wir tun, damit es funktioniert?

Kein extra **QA**

Operations im Team

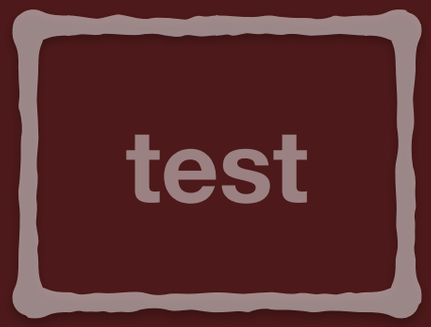
Support durch Business Analysten
und Developer

Alle **Verantwortung** und **Fähigkeiten**
mussten in ein Team

Technologisch brauchten wir eine

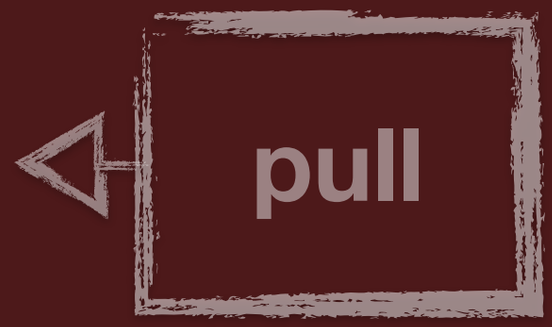
Deployment Pipeline

Automatisierte Test-Stages aber nur
ein Stage für Menschen



Der **Test-Modus** ersetzt das Preview-System

Feature Switches sind nötig, aber
seltener, als man denkt



Kanban statt Scrum

Commanders Intent

Acceptance Tests

Kommunikation

keine detaillierte Spezifikation

Community mit Anwendern und Entscheidern aufbauen

Anwendungs-**Controlling**

Die **Anwendung informiert** selber
über Änderungen

Wie weit sind wir damit gekommen?

500.000 Zeilen Code
(Java, Javascript, Groovy, XML, HTML)

21.000 Unit Tests & **400** Selenium Test

Fehler **schnell beheben** ist wichtiger als vermeiden

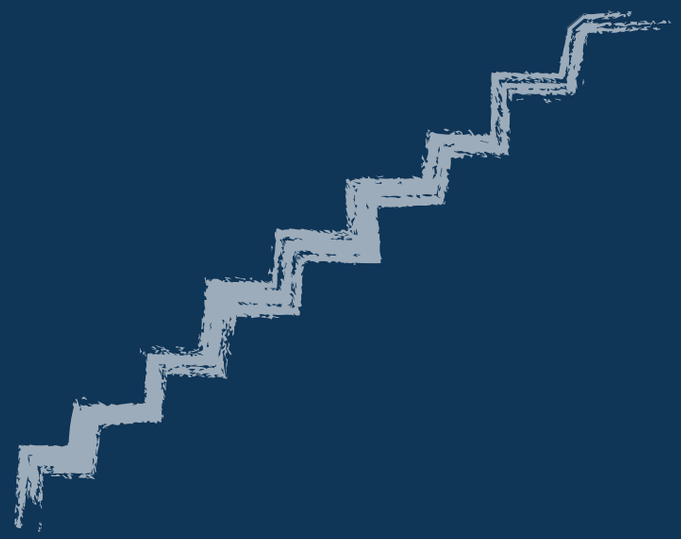
17 „Entwickler“, **3** BA, **1** UX, **1** PO

Klassische Rollenbilder **gemischt**

Was hat es in uns verändert?

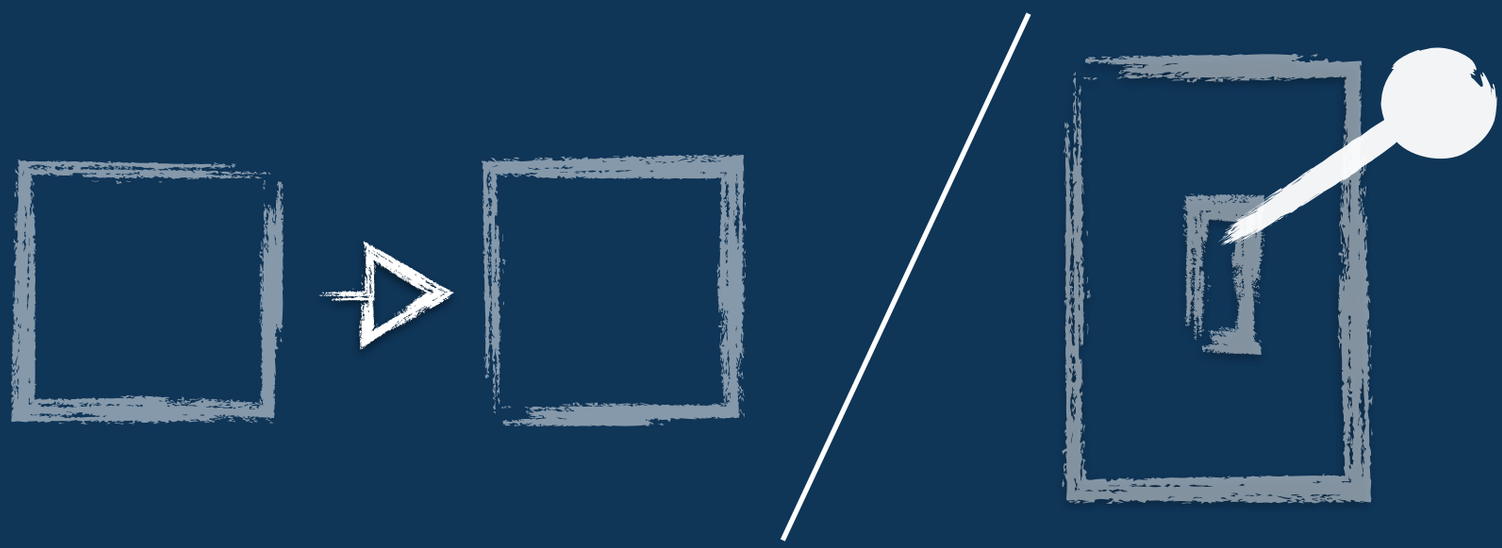
„Das ist **mein Produkt**“

Test Driven Development



Refactoring in **small steps**

Agile Methoden und Continous Delivery
greifen **perfekt** ineinander!



Keine Abstimmung zum **Rollout**
nur **Freischalten** ist Entscheidung des
Product Owners

Feature oder **Bugfix**
Wo ist der Unterschied?

Neue Prioritäten für Bugfixes/Findings

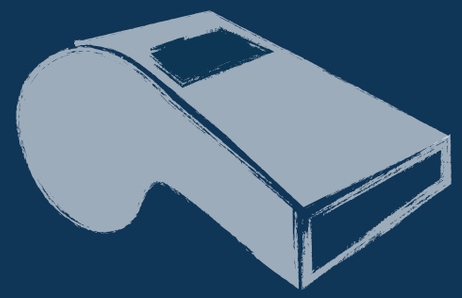
Quickwins first

Hard-Coded ist die neue
Konfigurierbarkeit

Was sind die Voraussetzungen?

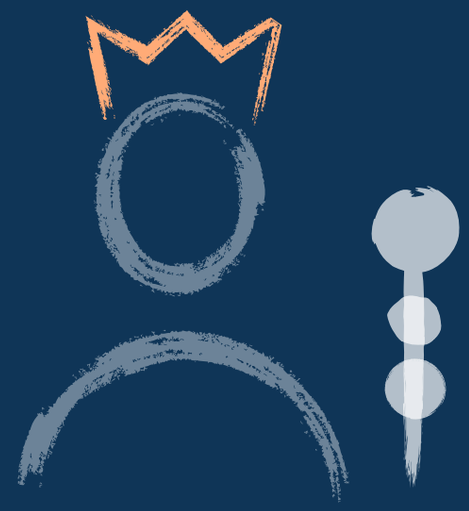
„Einfach“ **beginnen**
Es gibt kein **„Fertig“**

„Seniore“ Teammitglieder



Ohne **Disziplin** geht es nicht

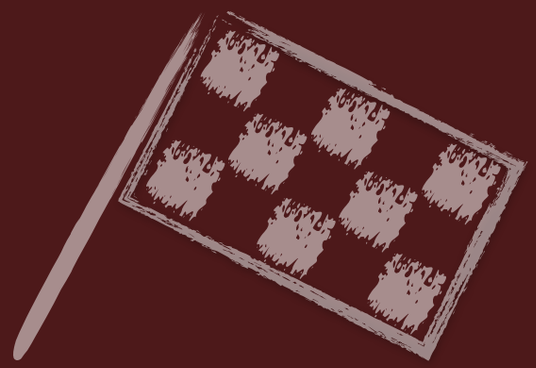
Nein sagen!
Der **Sog** zum klassischen
Vorgehen wird stark sein



Eigenverantwortung muss gewollt
und möglich sein

Haben sich die Erwartungen erfüllt?

JA!



Themen können technisch und fachlich
fertiggestellt werden!

Continuous Delivery ist ein **Mindshift**

Marco Kisperth
marco@kisperth.de
@kisperth

Jörg Müller
joerg.mueller@hypoport.de
@joergm
blog-it.hypoport.de

Vielen Dank an Verena Würfel für die Erstellung der Grafiken