

[www.immobilienscout24.de](http://www.immobilienscout24.de)



# Agile Refactoring @ IS24 gearconf

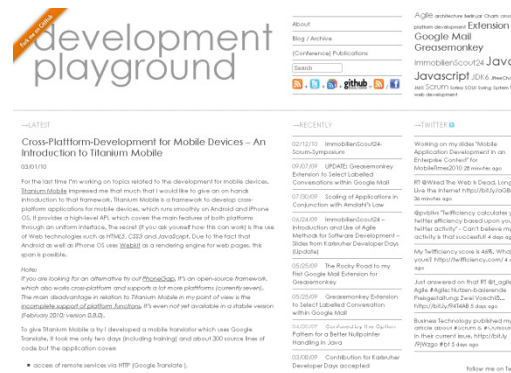
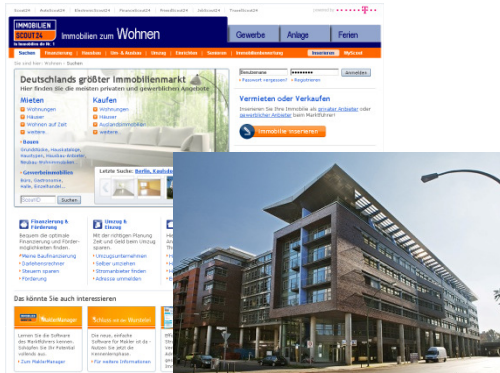
Düsseldorf | 11.10.2010 | André Neubauer & Oliver Schmitz-Hennemann

I'm **André Neubauer**, find me at...

andre.neubauer@...

devpg.com

twitter.com/devpg

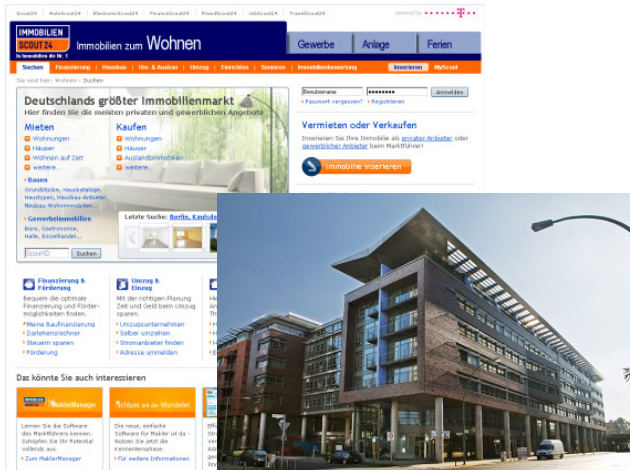


- Team Leader IT Development
  - Researcher, Speaker, Author
- #web #devops #mobileWeb  
#android #ar

I'm **Oliver Schmitz**-Hennemann, find me at...


schmitz@is24.de

twitter.com/OSchmi



- Head of IT Architecture
- OrgDev, agile evangelist, inconvenient

# Warning



Agile  
ideas  
ahead



# Some numbers

500 employees

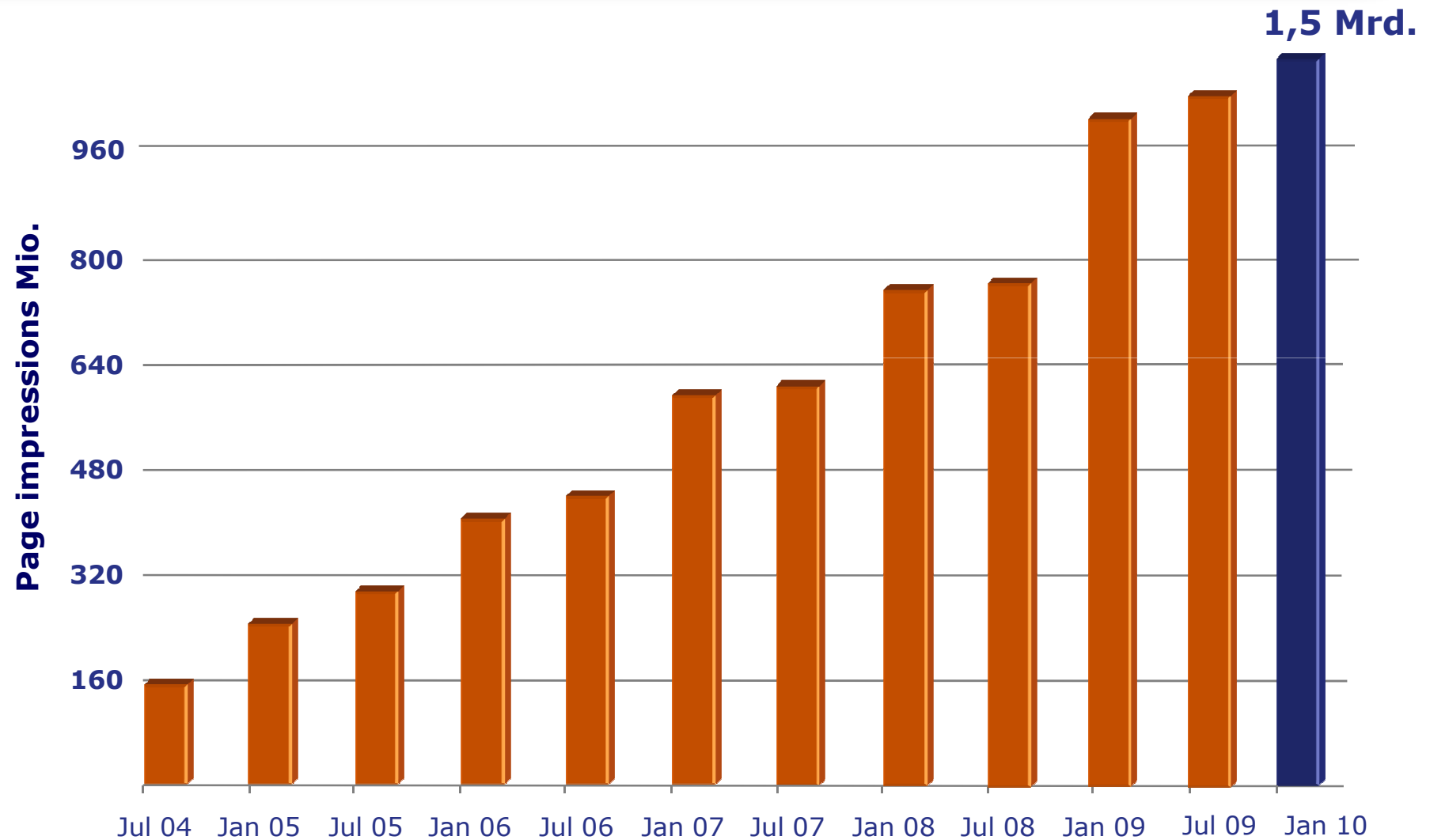
140 IT

67 developer

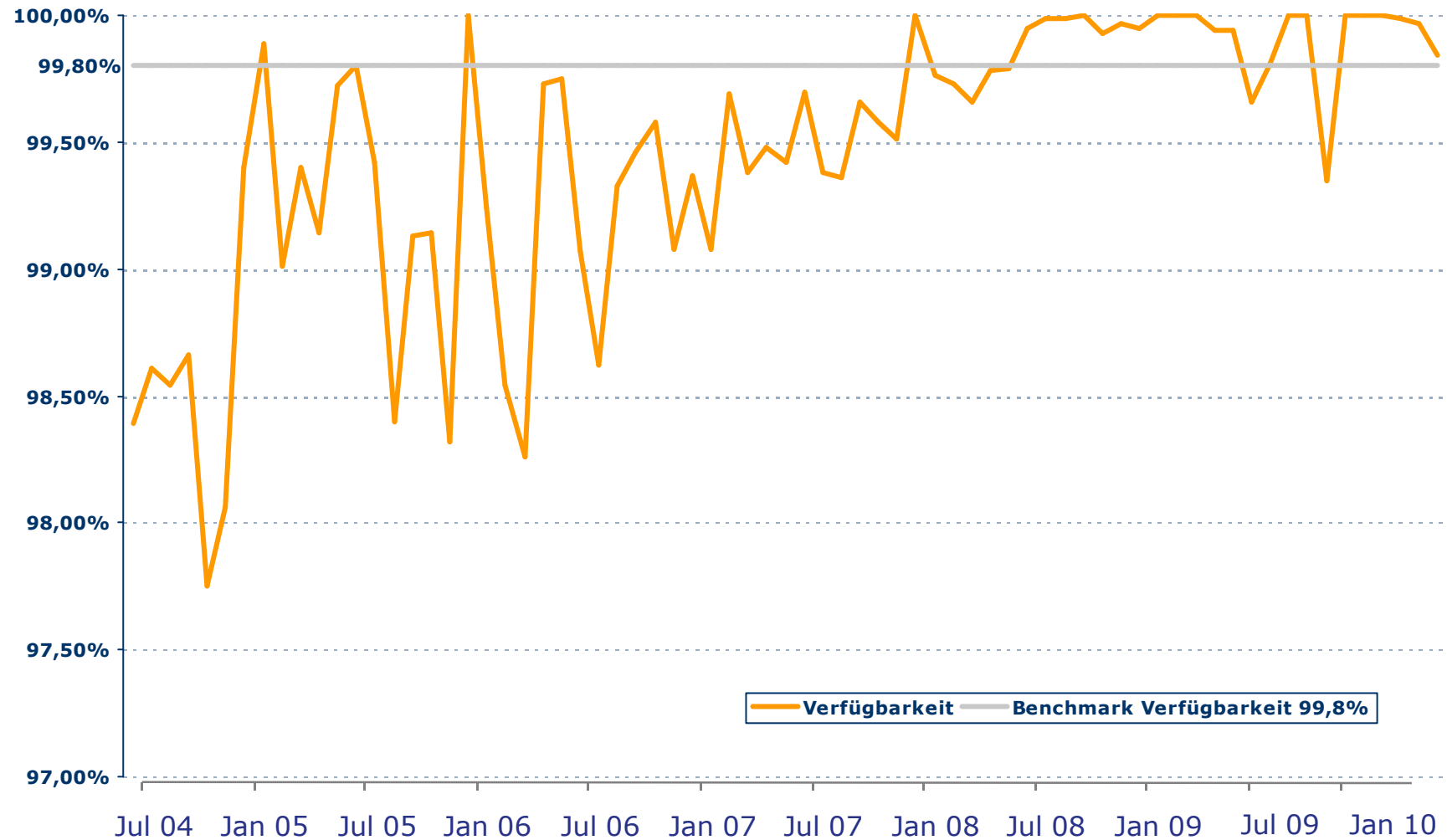
- ➔ > 4,5 Million unique users
- ➔ > 1,5 Billion Page Impressions
- ➔ > 250 Million Expose views
- ➔ ~ 1,2 Million active listings
- ➔ > 120.000 Vendors  
per month



# Page impressions



# Availability





# Agile refactoring

# Epic



Some  
history

# Story

People  
unhappy

# To much legacy



# Story



Legacy

# Story

ARC  
desaster  
2007/2008




# Lesson learned



No super  
goals.  
Focus!



# Lesson learned




If you get  
stuck...  
Get help!

# Lesson learned



Get  
external  
help!

# Lesson learned



> 2 month  
You got  
stuck!

# Lesson learned



# Story



Scrum  
2008

Why

# Delayed projects

# Why

# To many topics per team

Why

Much more  
developers



Teams want it  
CTO wants it


How

# Team by team

How

# External help

# Lesson learned



Prod up!  
Fun up!  
Energy!

# Lesson learned



Identifies  
waste

# A lot of impediments

# Lesson learned



Epic

Agile  
refactoring



# Setting

1/3

2 years

Stop the business

# Setting

1/3

2 years

Don't stop the business

# Solution

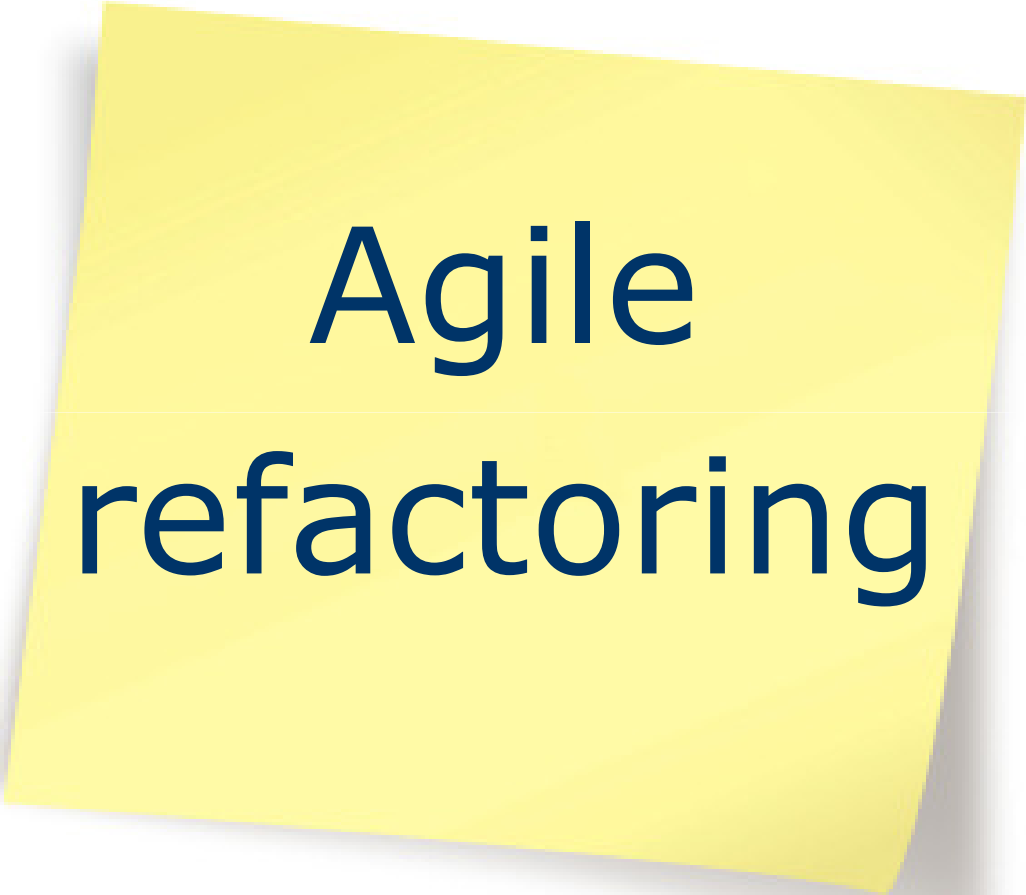
Put 33% of dev  
into ARC renewal team

# Solution

But wait...  
...Scrum Masters will kill  
you


# That's inconvenient

Epic



Agile  
refactoring

# Lesson learned



Do it in  
the teams

# Lesson learned



Spread  
Know how





Decisions (working draft)

Peripheral concerns	Domain Model	Transactions	Persistence Layer	Web
Configuration	Asymmetric	declarative (annotations)	JPA by Hibernate	Flow Logic → Page Flow ⇒ Controller
Order Service is to activate migration and... by i18n	Distinguish domain objects (domain holders) and services (business logic)	Container Managed (no manual code)	DADs efficiently designed	Presentation Logic - Simple (syntactical) - Validation/Formats - mandatory fields
Search Index is separate system (physical)	Services are a layer exposing domain objects to all layers	No nested TXNs No distributed	Use Hibernate's built-in caching	Component Coupling using Java Interfaces
	Only services define business logic	No Long running TXNs	Responsibilities as ODM	See Commit Coupling
	→ One Service Call	DB only	a distinct Team may make sense due to complex technology (close to DB)	SIMBA first for major new framework as second step
	Relations are defined by services or... if it is a composition... by domain objects	TXNs may be only started in services	Performance issues will need to be addressed	Session needs to be externalized
	No Avoid domain object mapping across layers		New architecture produces data events dependencies does not consume	
	Initial Plan: Responsibility: ARC Implementation: DEV			
	How to handle Relations in Domain Model			
	Contextual validation done by business logic (domain specific checks)			
	Modularize DM according to common use cases			

*Handwritten notes on board:*  
 - Separation  
 - CarS Arne  
 - Sebastian K.  
 - MTVB  
 - Barbara  
 - An G.  
 - Uwe  
 - Jule  
 - Anli V.

09.07.09 OC  
 Commitments from  
 Mgmt in *(Handwritten signatures)*  
 09/10/09  
 09/10/09  
 09/10/09  
 \* There will be 3 of our resources for  
 IT-2010 *Dev ✓ Arch ✓*  
 \* Trust our current results ✓ ✓ ✓  
 \* WE DO OUR BEST!  
 IT-SYS IT-PRO ✓  
*(Handwritten signature)*

# Lesson learned



# Story



Standards

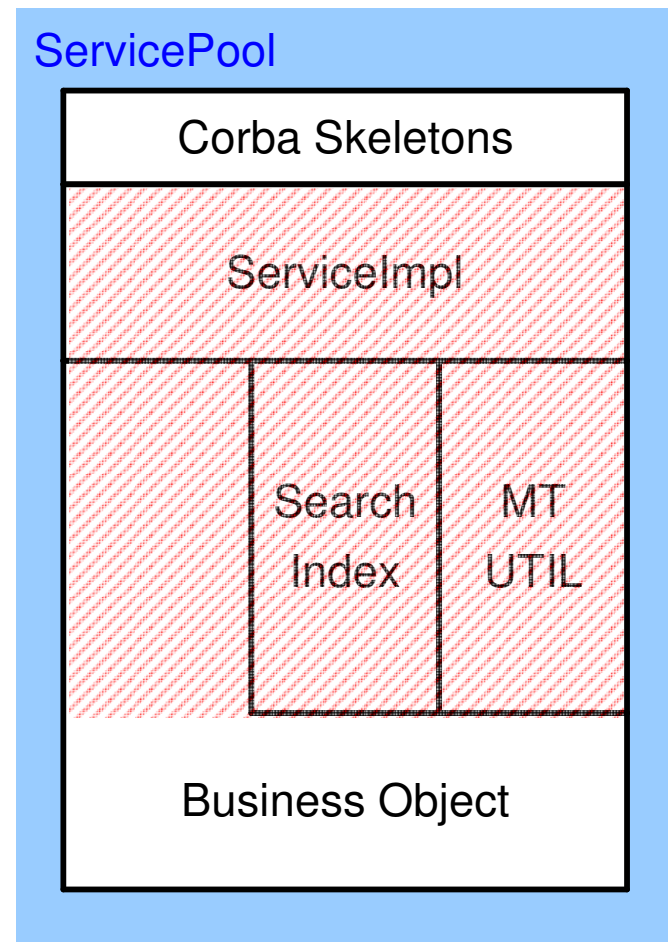
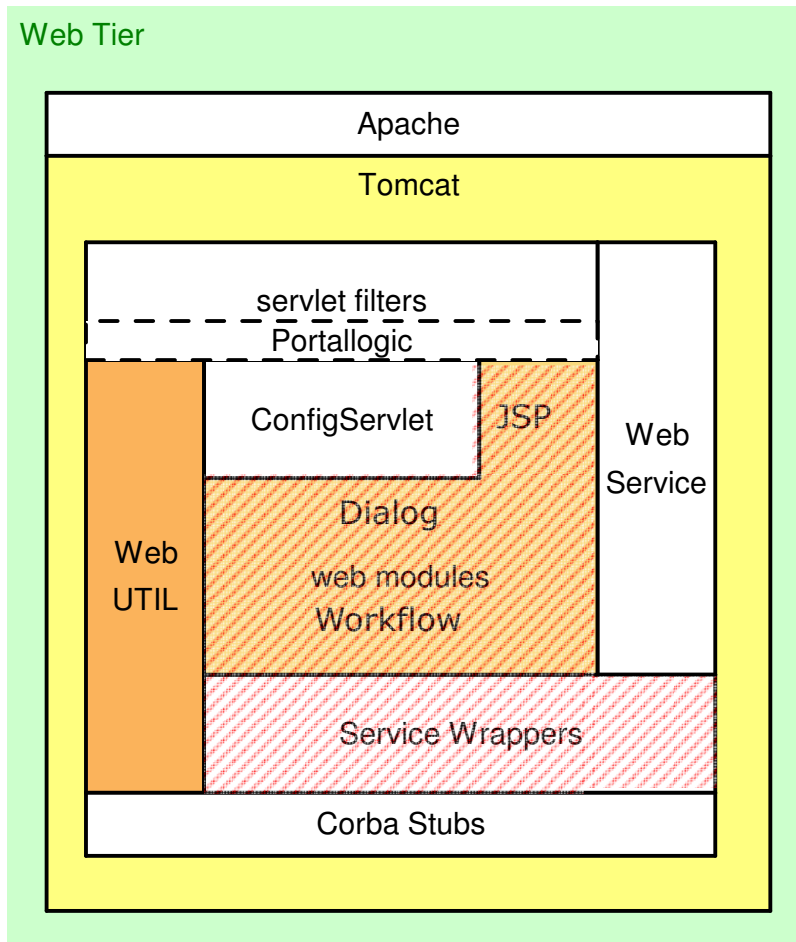
## Some Details

Hibernate/JPA  
Springframework  
SpringMVC/WebFlow

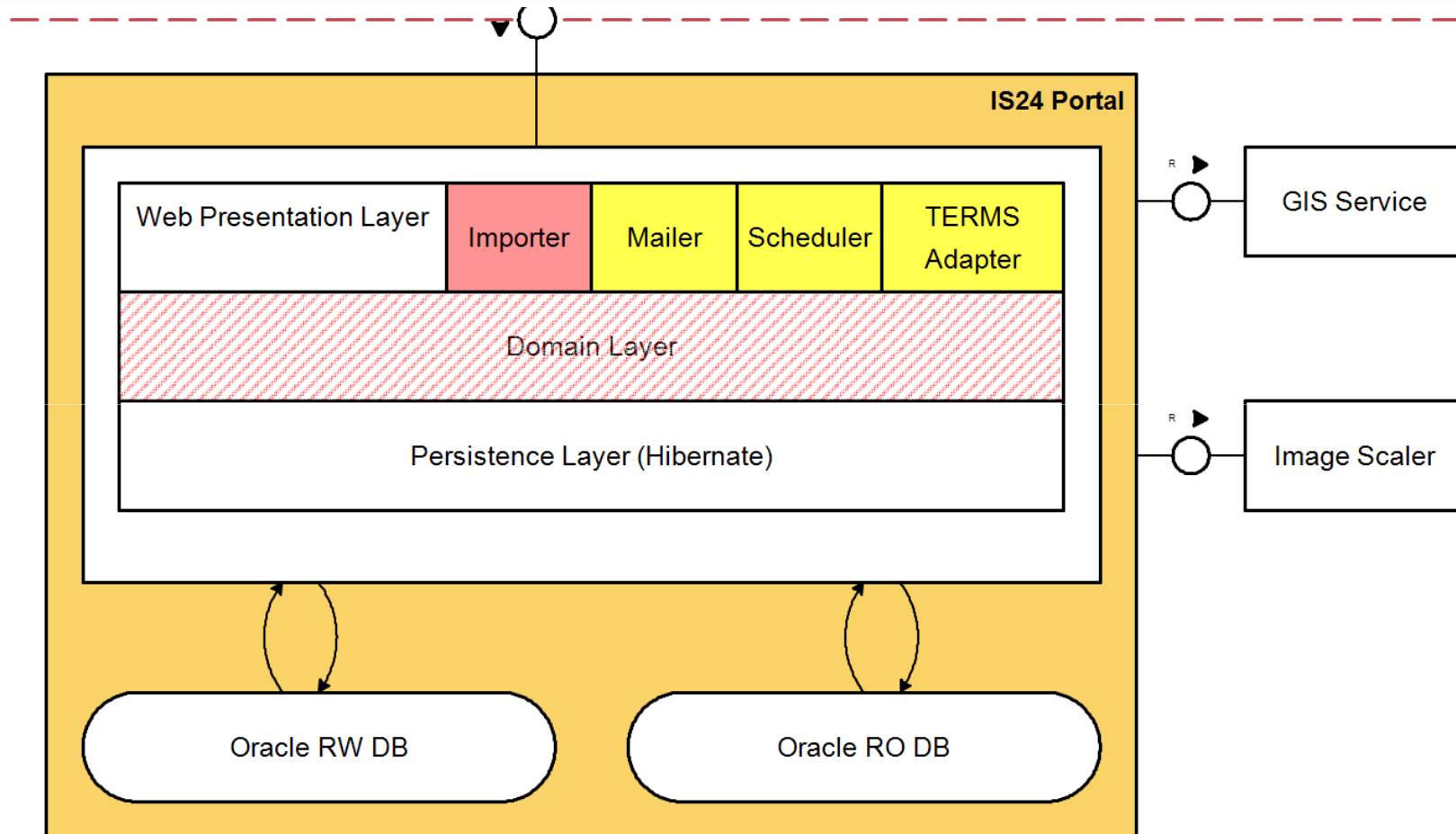
# Task

Start with  
**OODM**

# Old architecture



# Target architecture





# Task

Second step

# Web-Framework migration

# Story

SPoC  
CPoC  
rollout

# Lesson learned




Have  
switches

# Lesson learned



New Features  
after  
SPoC

# Lesson learned



Prod up!  
Fun up!  
Energy!

# Story

ARC

P<sub>roduct</sub> O<sub>wner</sub>

# CTO needs control

# CTO needs someone to ask?

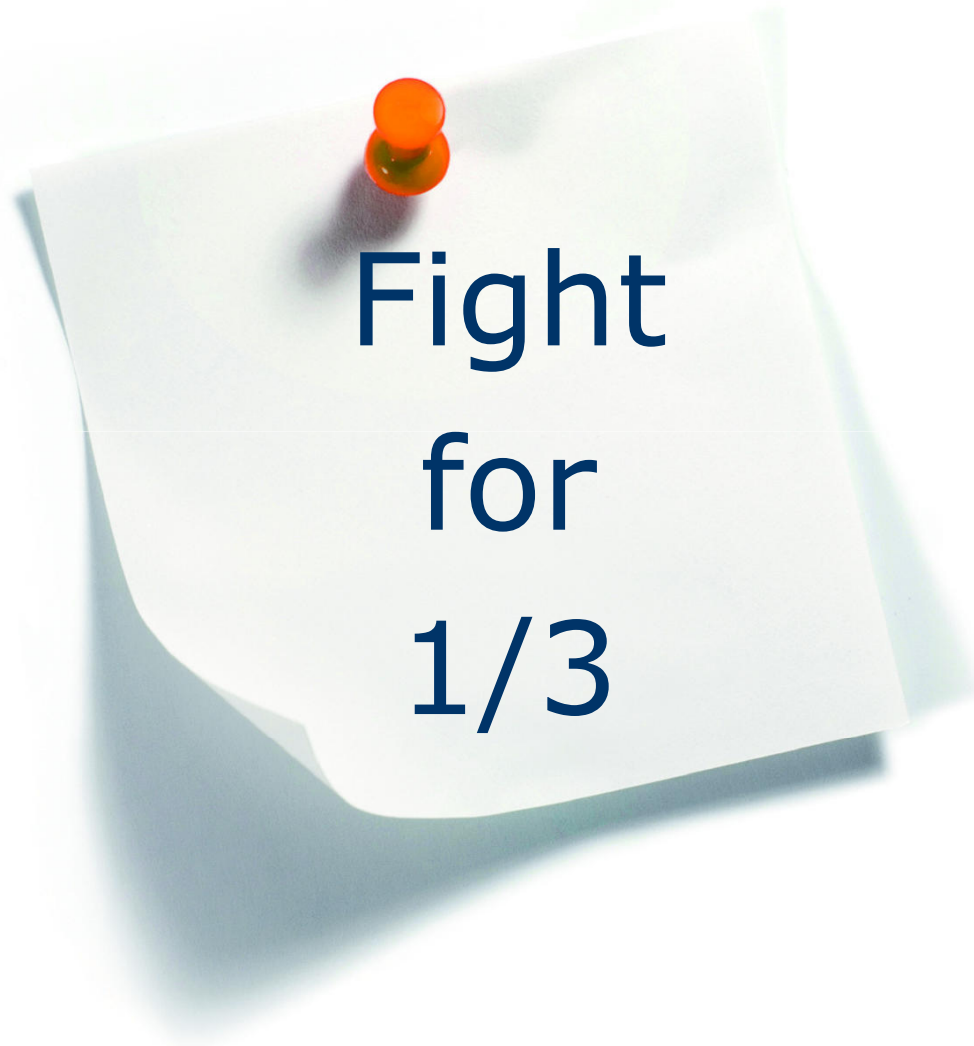


# CTO needs Someone to hang!

# Task

Coordinate  
Teams &  
POs

# Lesson learned




# OODM

## 34 epics

# OODM

## 340 Stories

# Lesson learned




Have  
Mr. Pareto  
in the team

# Lesson learned



Trust in  
stack

# Lesson learned



No more  
big things



# OODM Cockpit "Über den Berg"

gelb: wir brauchen ARC-Stories (Stand 04.06.2010)

Platform	Authentication Account & API	Authorization (Priv. & Roles)	Customer → OC1? Sonst Gerald SPC	ShoppingCart	Multimedia	GIS Geocoding Hierarchy → DEV2 sieht Probleme	Addresses → DEV2/API
Offer	Realtor Impressum Profile	Real estates → SM2.0 ✓ → OC1?!?	Angebots- & Exposé-Manager	Focus Placement Toplisting	Vendor Reporting Umstellung auf online reporting DB ist IT2010	TKM	VÖ / Kanäle
Search	Expose → PMI hat Interesse an Einführung von Expose 3.5 für alle Typen	Search	Shortlist	Saved search	Fulfillment		
BUs	Leads Im alten Stack, wird vom PM migriert/neu geschrieben	Financing Im alten Stack, wird vom PM migriert/neu geschrieben	Valuation	Relocation	Allianz	KSC Guideline machen und Team fragen	
Misc.	Data Events Caching	ESB	Audit Trail	Importer WTH? 	Mailer	Scheduler & Jobs	

## Lesson learned



Being 2<sup>nd</sup>  
PO for 10 Teams  
sucks

# Lesson learned




Not for  
ARC  
but 4 you



# Lesson learned



# Story



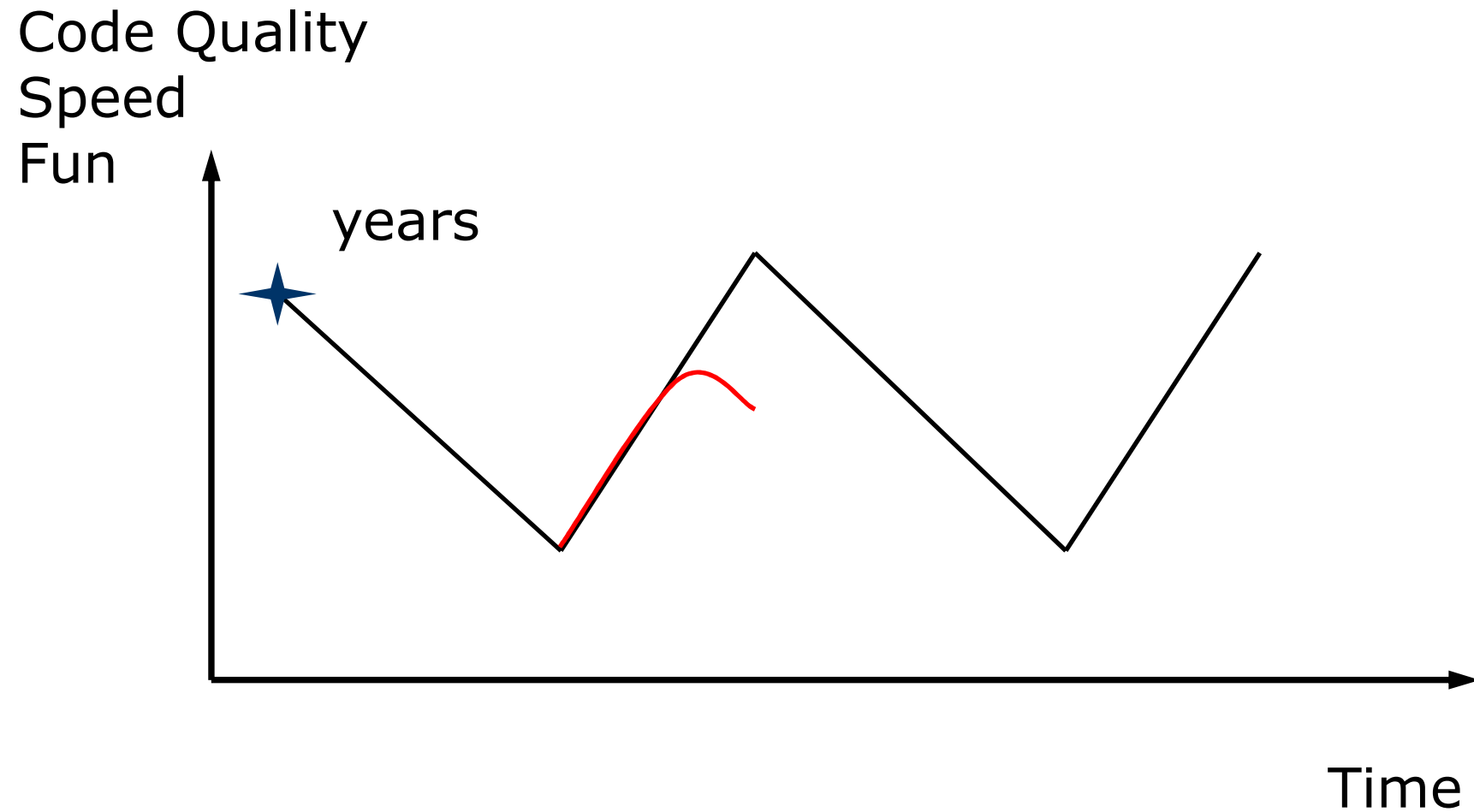
What  
was wrong

# Lesson learned




IT/ARC  
Projects  
are bad

# Project cycle





# Story

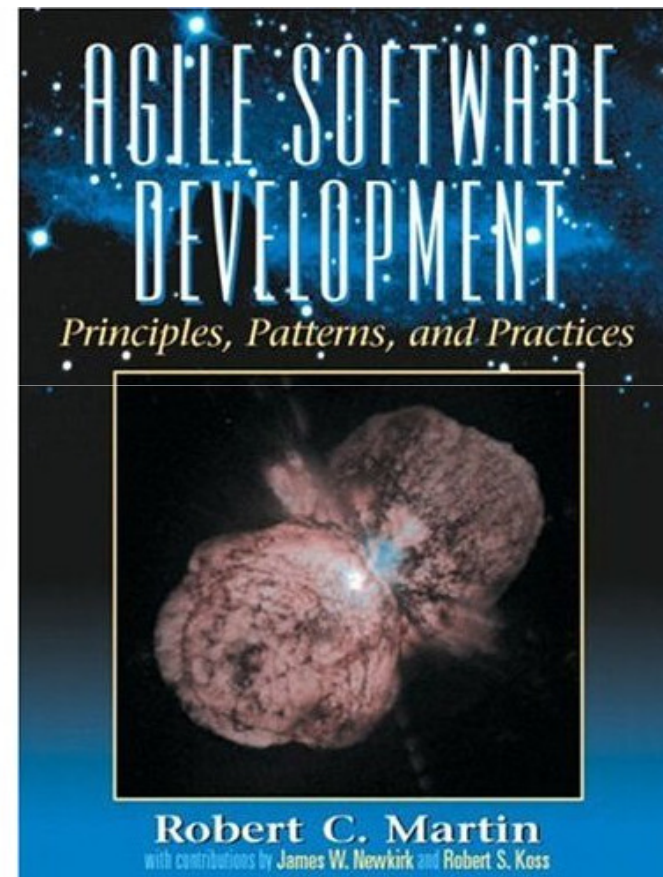
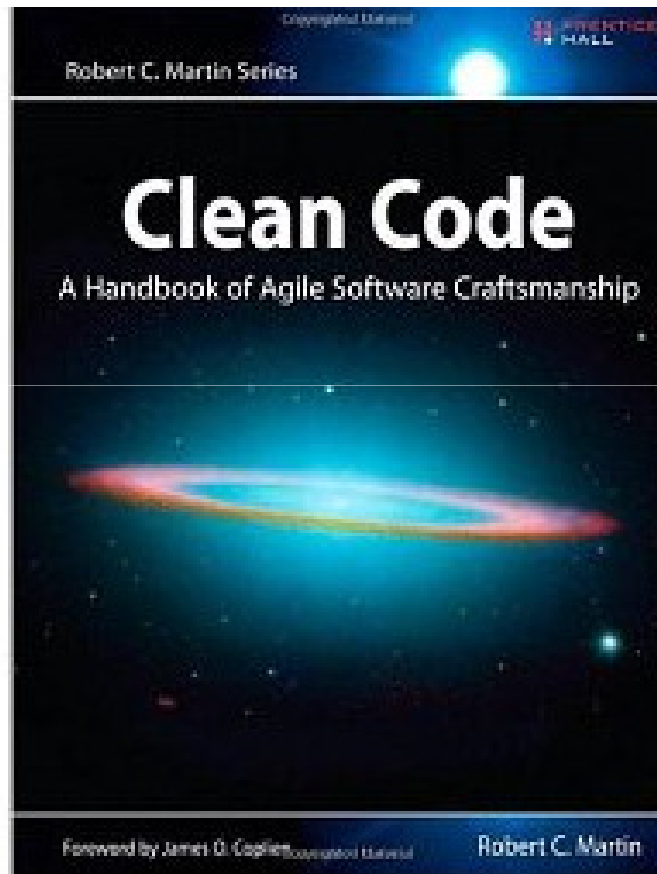


make it  
better

# Task

What would  
be agile

# Read some books



# Lesson learned



Craftsman-  
ship

# Lesson learned



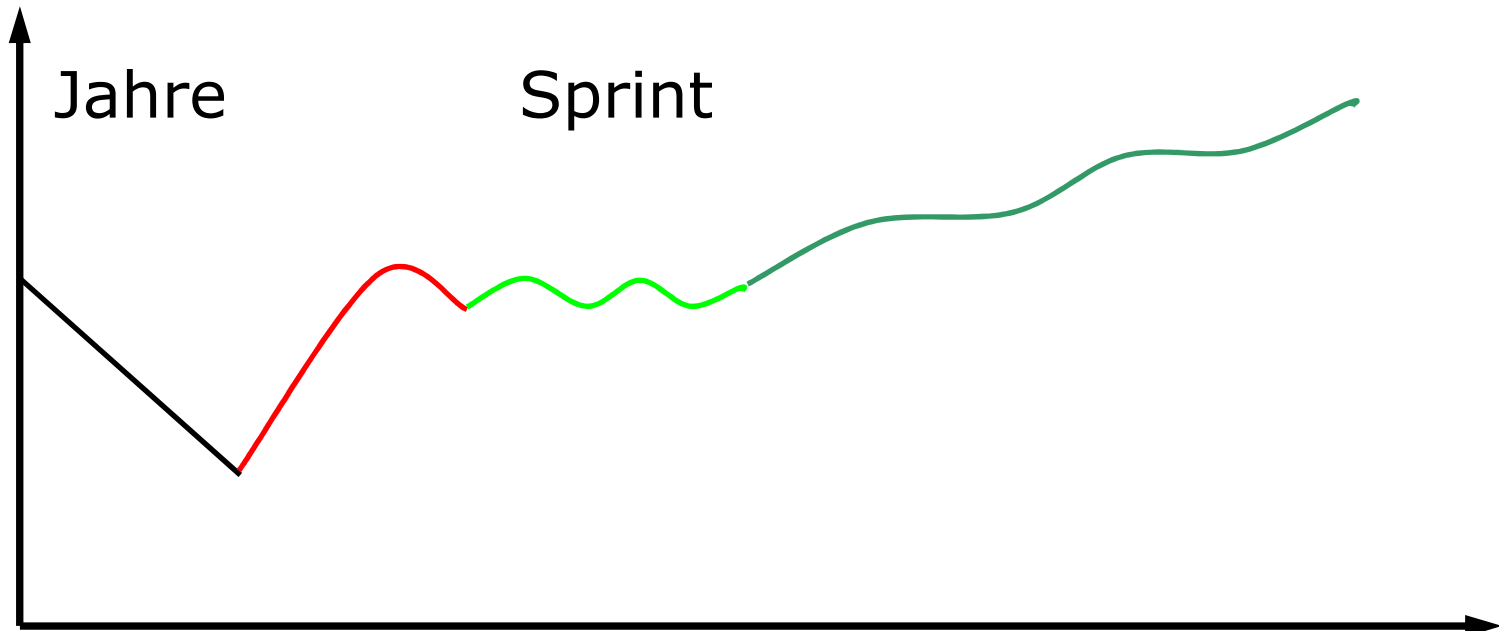
Scout  
principle

# Lesson learned




sustainability

Code Quality  
Speed  
Fun



Time

# Task



How to  
reach it?




# Lesson learned



Fix what  
hurts most

# Lesson learned



Let people  
decide

# Sounds simple

But wait  
a minute

# Task

**S**omebody

**E**lse's

**P**roblem

# Lesson learned



responsibility  
requires  
power

# Build environment

# Lesson learned



Get rid  
of silos




## Lesson learned



With great power  
comes great  
responsibility

# Lesson learned



Give trust  
and help



# Task

Greenfield  
more  
Efficient?

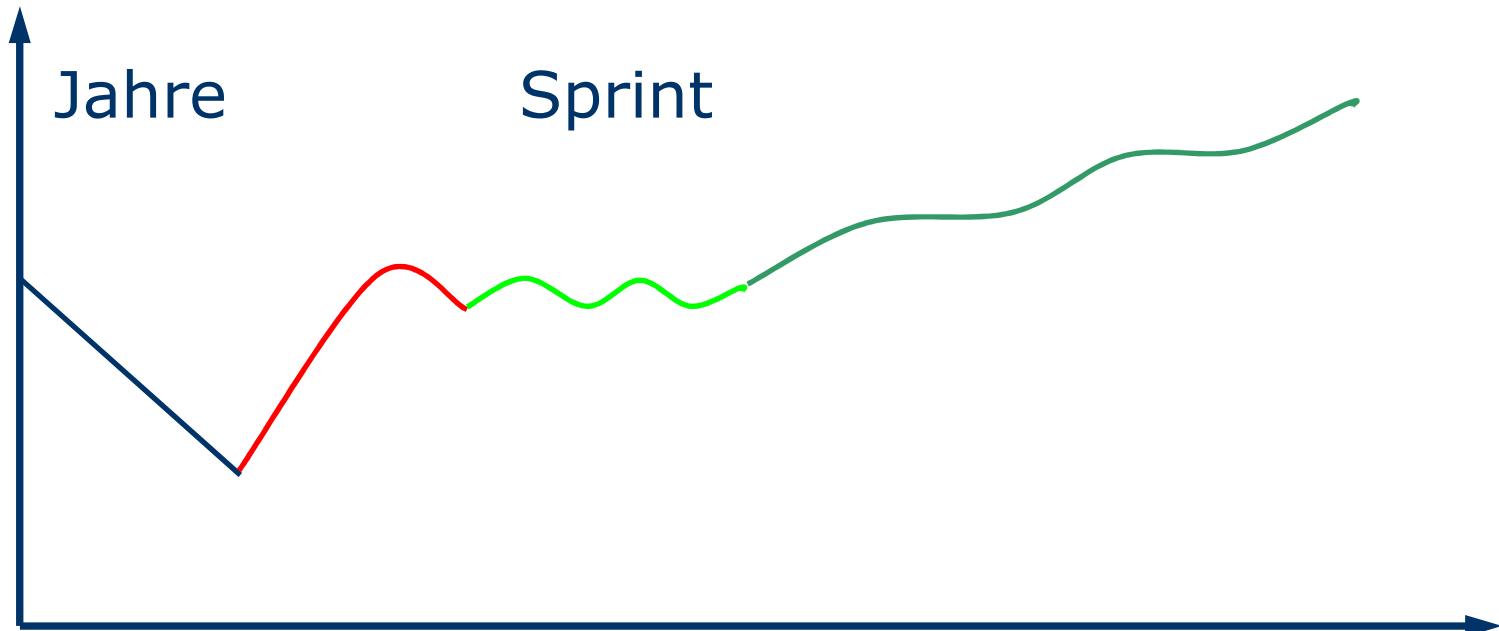
# But...

# Lesson learned



We changed  
our mindset

Code Quality  
Speed  
Fun



Time

# Lesson learned



Enables  
Innovation



The End

People  
Happy!

What's next

# Build 2010

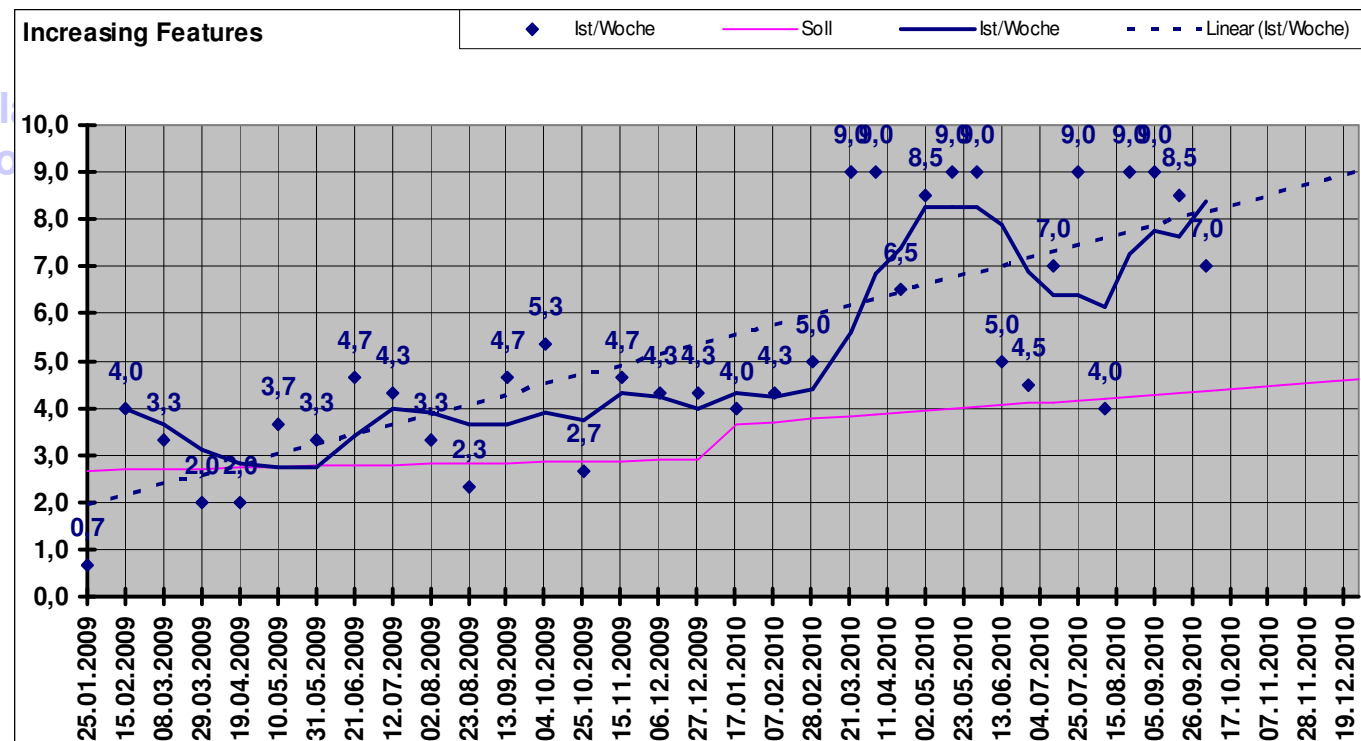
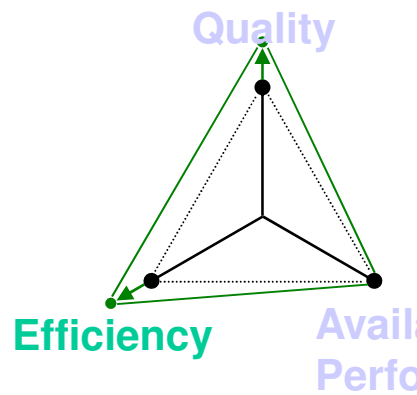
## Continuous deployment

### DevOps?

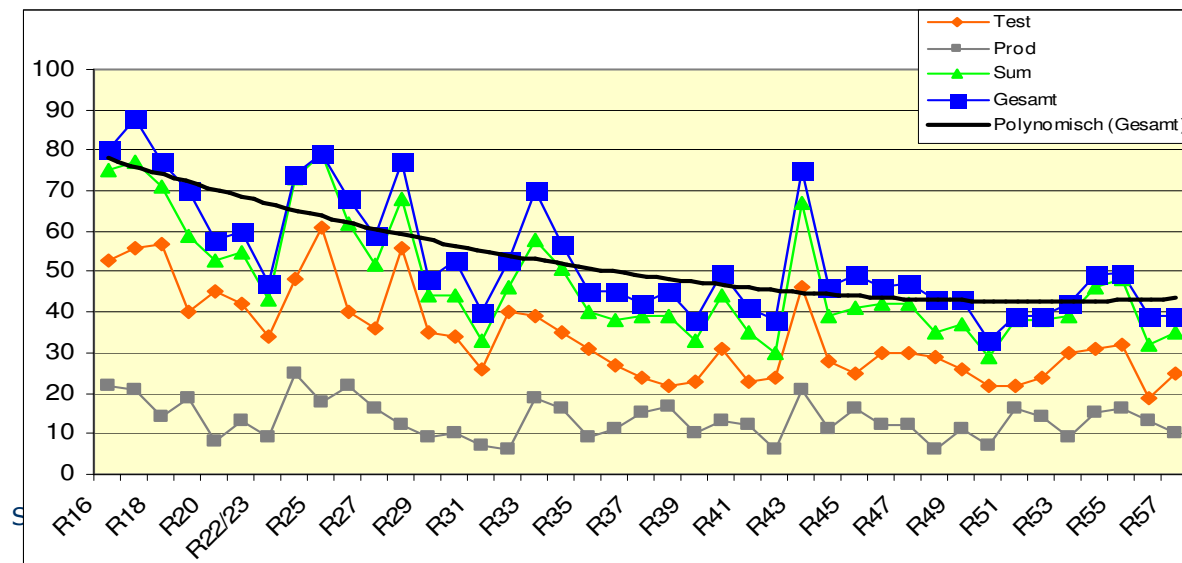
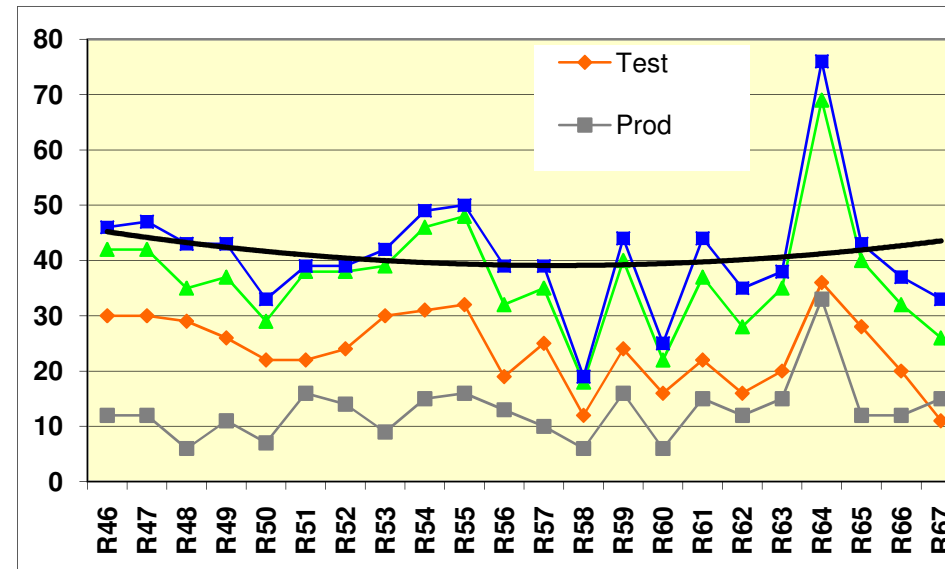
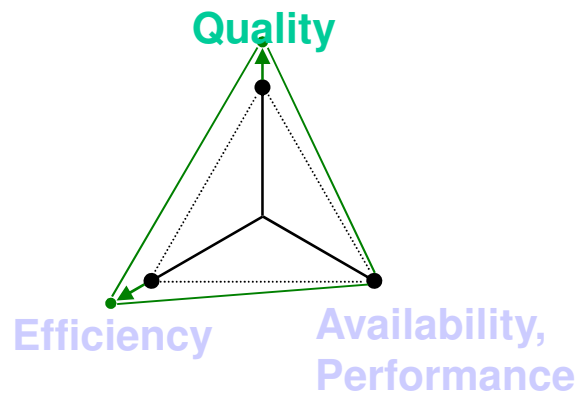
# Thanks



# IT2010 – Increasing Features



# IT2010 – Improved Quality



# Ressources

<http://agilemanifesto.org/>

<http://agilemanifesto.org/principles.html>

<http://www.amazon.de/Clean-Code-Handbook-Software-Craftsmanship/dp/0132350882>

<http://www.amazon.de/Working-Effectively-Legacy-Robert-Martin/dp/0131177052>

<http://www.amazon.de/Software-Development-Principles-Patterns-Practices/dp/0135974445>