

**Der optimale Weg  
von der Idee bis zur Rendite**

28. Mai 2008  
Peter Stevens, Sierra-Charlie Consulting  
[scrum-breakfast.blogspot.com](http://scrum-breakfast.blogspot.com)

Idee

Rendite

## About Peter Stevens...

- 1982-1985: Software Engineer at Microsoft
- 1986: Emigrate to Switzerland
- ... much interesting work
- 2005: First Contact with XP and Scrum
- May 2007 Certified Scrum Master (with Mike Cohn)
- Today Independent Scrum Coach

Training

Coaching

[Crisis] Project Management

Initiator: Scrum Breakfast in Zürich

Author: <http://scrum-breakfast.blogspot.com>

# Themen

- Der Weg von der Idee bis zur Rendite
- Das Toyota Produktions- bzw. Produkt-Entwicklungskonzept
- „Lean“ bzw. schlanke Konzepte und Werkzeuge
  - Der Zusammenhang mit Scrum
- Erste Schritte

# Der Weg von der Idee bis zur Rendite

Rendite

- Welche Ziele und Prioritäten? Das Produkt...
  - nach einem normierten Prozess kreieren?
  - komplett dokumentieren?
  - rasch auf den Markt bringen?
  - innerhalb einer definierten Frist produzieren?
  - kreieren, was der Kunde will?
  - mit möglichst wenig Aufwand produzieren?
  - gelingen beim ersten Wurf?

Idee

# Der Weg von der Idee bis zur Rendite

Rendite

- Welche Ziele und Prioritäten? Das Produkt...
  - nach einem normierten Prozess kreieren?
  - komplett dokumentieren?

2. ➤ rasch auf den Markt bringen?

- innerhalb einer definierten Frist produzieren?

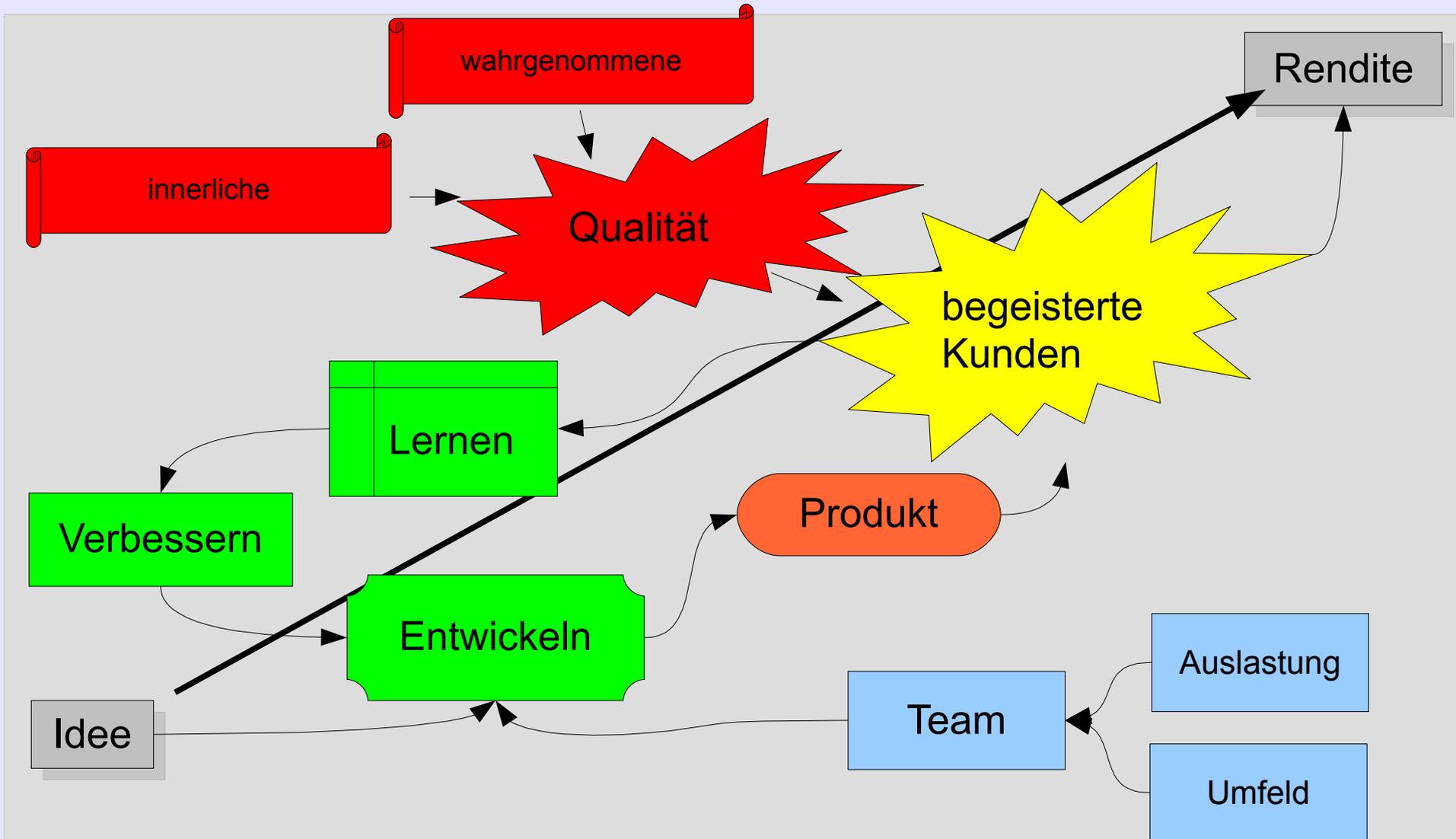
1. ➤ kreieren, was der Kunde will?

falsch mit möglichst wenig Aufwand produzieren?

falsch gelingen beim ersten Wurf?

Idee

# Der Weg von der Idee bis zur Rendite



# Toyota Production System

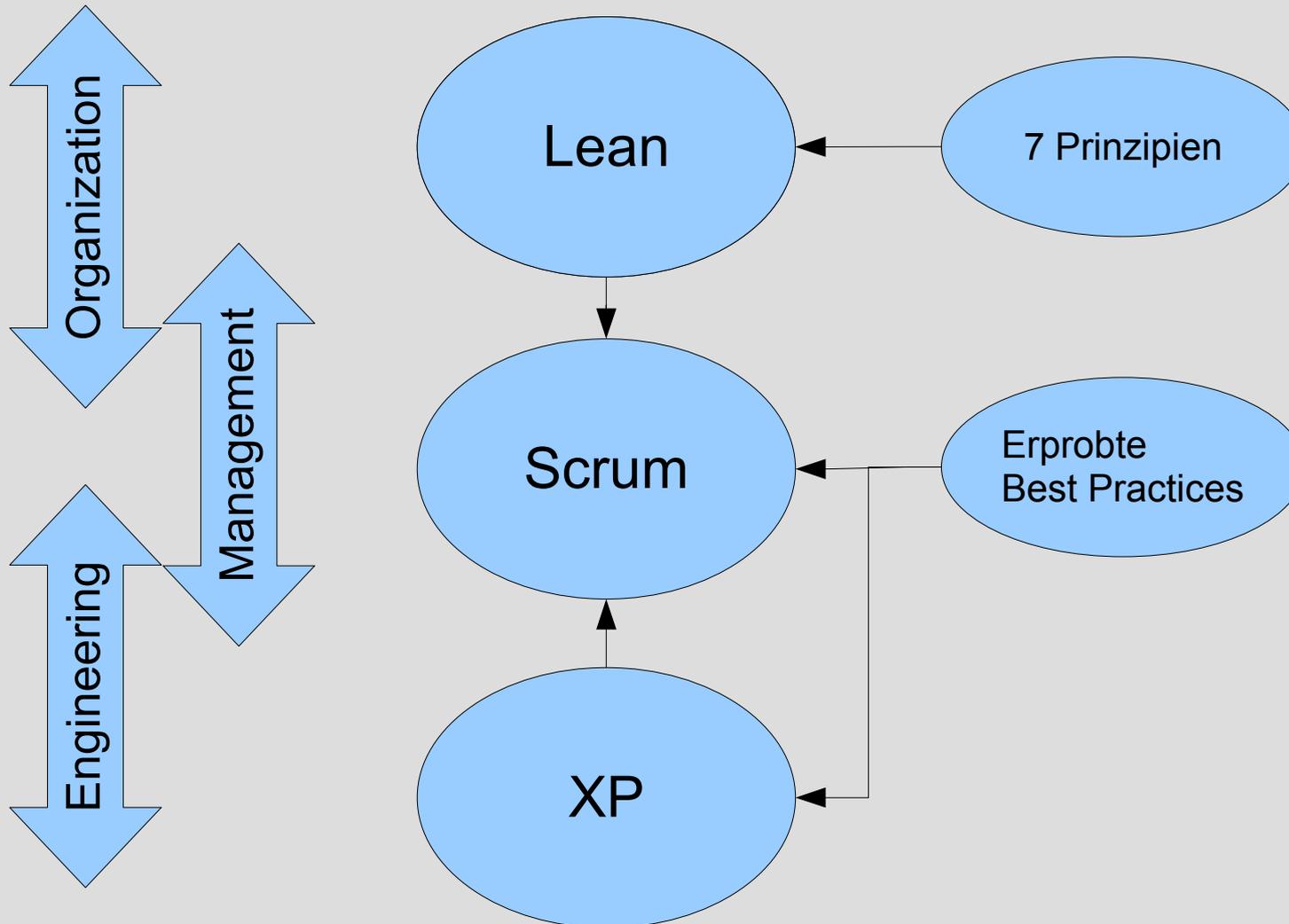
- Heute heisst TPS  
“Thinking Person's System”
- Nach dem Krieg: “Toyoda”  
besuchte Ford. Was sie sahen:
  - Massen-Produktion
  - Kapitalintensiv
  - Platzintensiv
  - Nicht anwendbar
- Dafür “Piggly Wiggly”:
  - Nachschub erst bei Bedarf:  
Just In Time Lieferung
- Daraus ist TPS entstanden und  
wiederum Toyota Product  
Development System
- 1957: Toyota Crown: erster  
Versuch in den USA:  
Katastrophe
- 1989: Lexus: Eintritt in den USA:  
überholte Mercedes & BMW im  
ersten Jahr

## Best Principles vs Best Practices

- Principles (Prinzipien) können überall sinngemäss angewendet werden
- Practices (Methodiken) werden für den Einsatz in einem Gebiet ausgearbeitet

Welche Prinzipien können für die SW-Entwicklung übernommen werden?  
Wie?

# Lean & Scrum in Perspektive



## “Lean” bzw schlank zusammengefasst

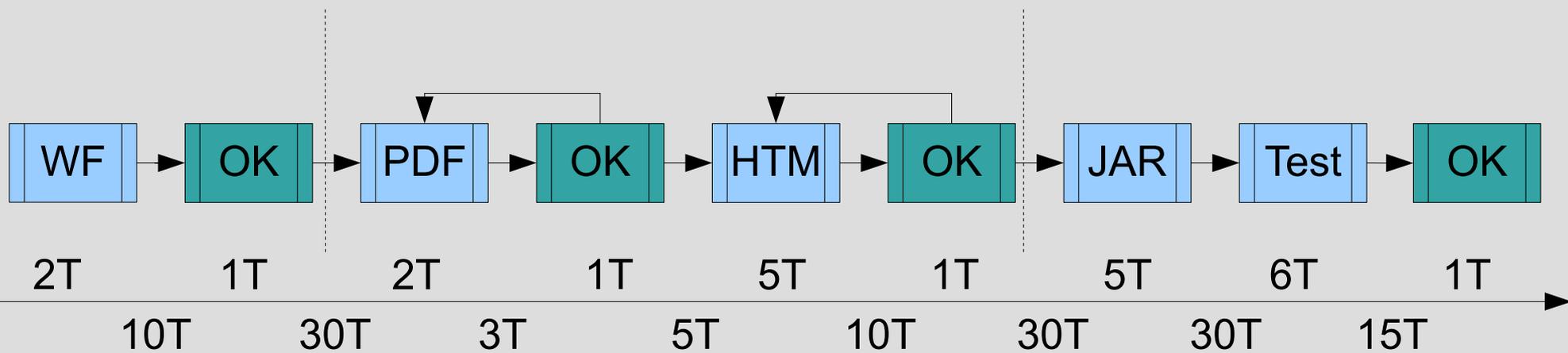
- Verschwendung eliminieren
- Lernen verstärken
- Entscheidungen treffen, so spät wie möglich
- Liefern, so schnell wie möglich
- Menschen respektieren
- Integrität einbauen
- Das Ganze verbessern

# 1. Prinzip: Verschwendung eliminieren



- Überproduktion
- Bewegung
- Transport
- Inventar
- Fehler
- Zusätzliche Schritte
- Warten

# Wertfluss-Diagramm



Arbeiten: 27 Tage

Warten: 133Tage

Gesamt: 160 Tage = 26 Wochen

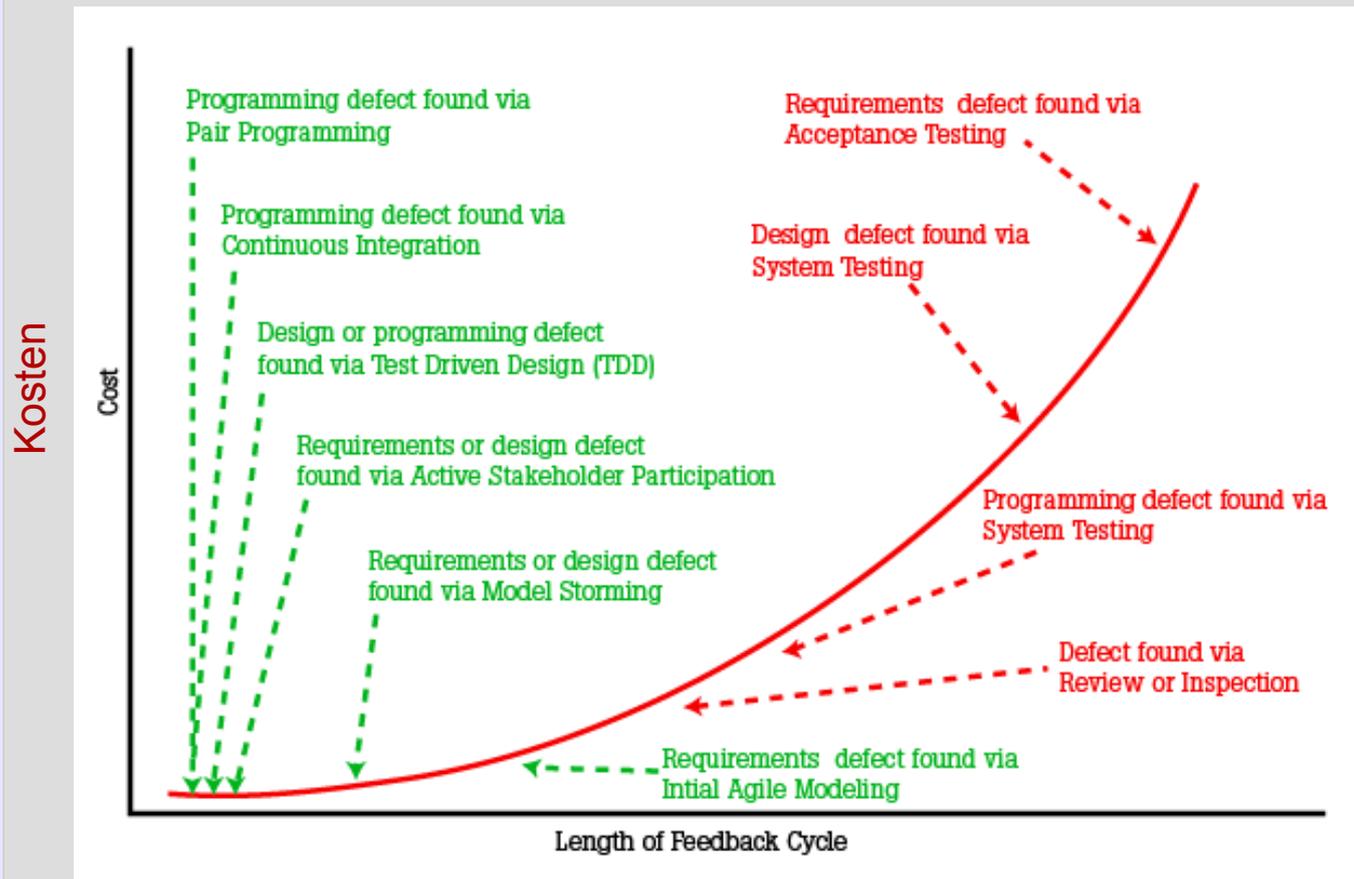
- WF Wireframe erstellen
- PDF Design erstellen
- HTM HTML & Javascript erstellen
- JAR Java Anwendung erstellen
- Test Abnahme-Tests
- OK Kunden-Abnahme

Was passiert während dem Warten?

## 2. Lernen verstärken



## 2. Prinzip: Lernen verstärken



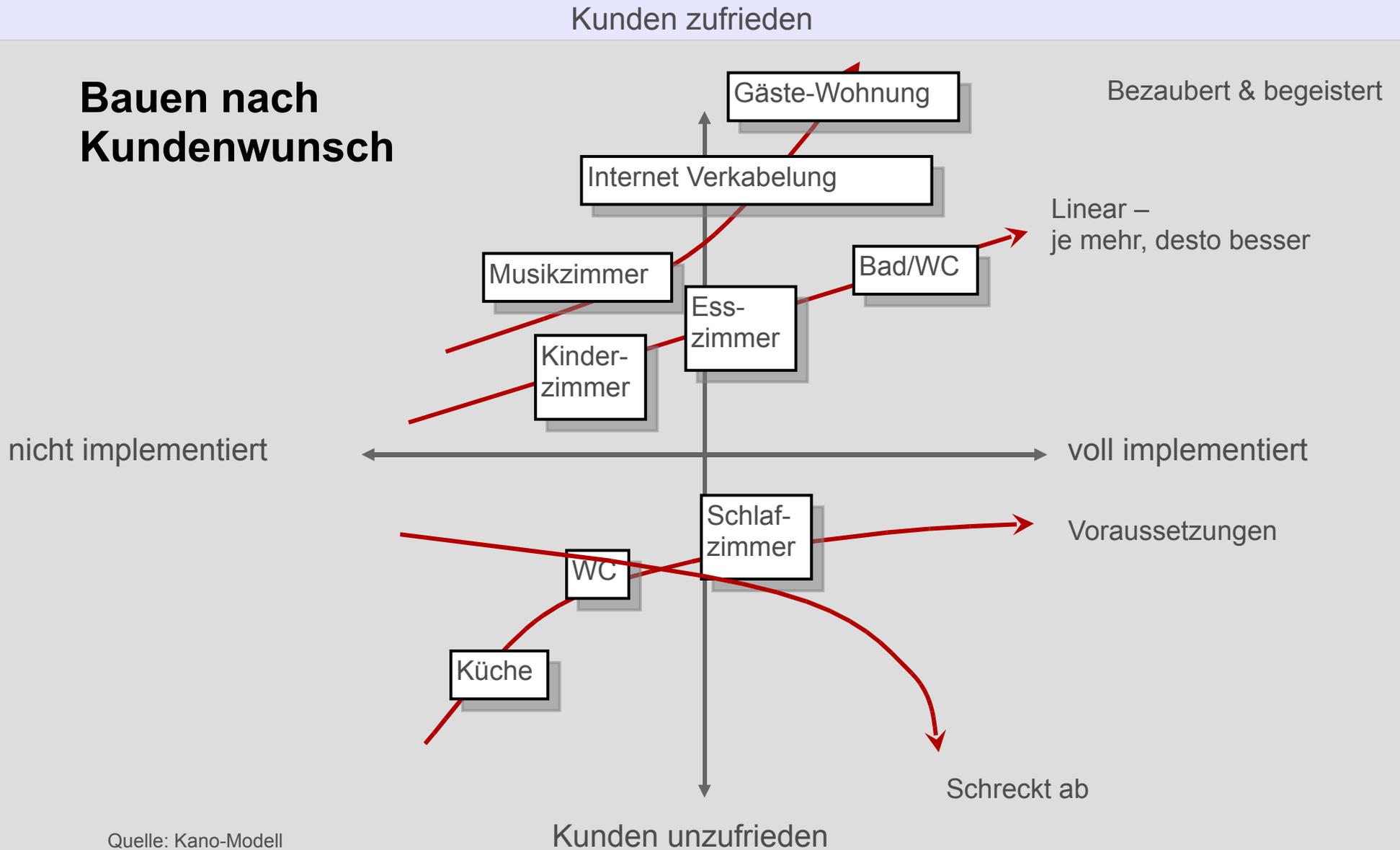
Classical Methods  
Agile Methods

Verzögerung bis zum Feedback

Source: "Pitching Agile to Senior Management"  
<http://www.ddj.com/architect/199300107>

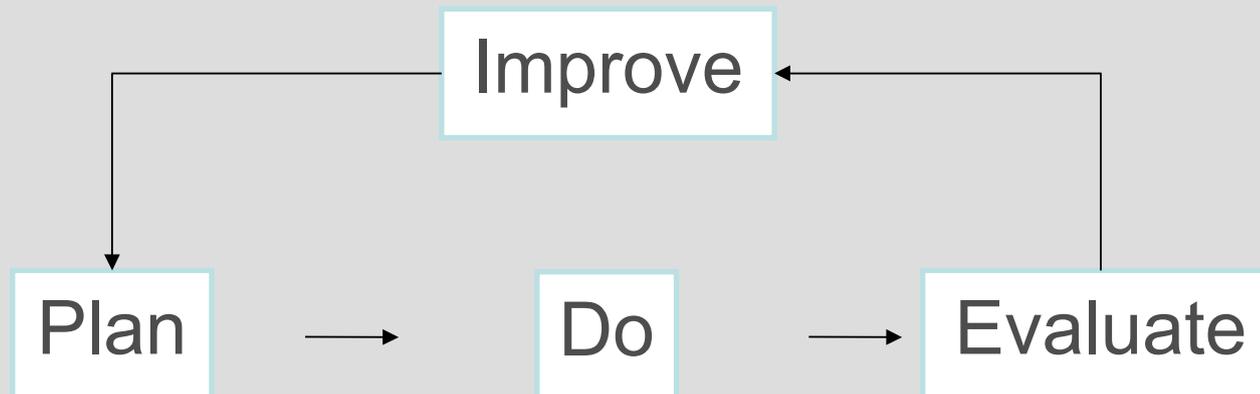
# Lernen Verstärken

## Bauen nach Kundenwunsch



Quelle: Kano-Modell

# Lernen Verstärken



- Scrum Werkzeuge:
  - > Benutzergeschichten (Kano-Modell)
  - > Sprint Planungs- und Vorführungsm meetings
  - > Daily Scrum
  - > Retrospektiven

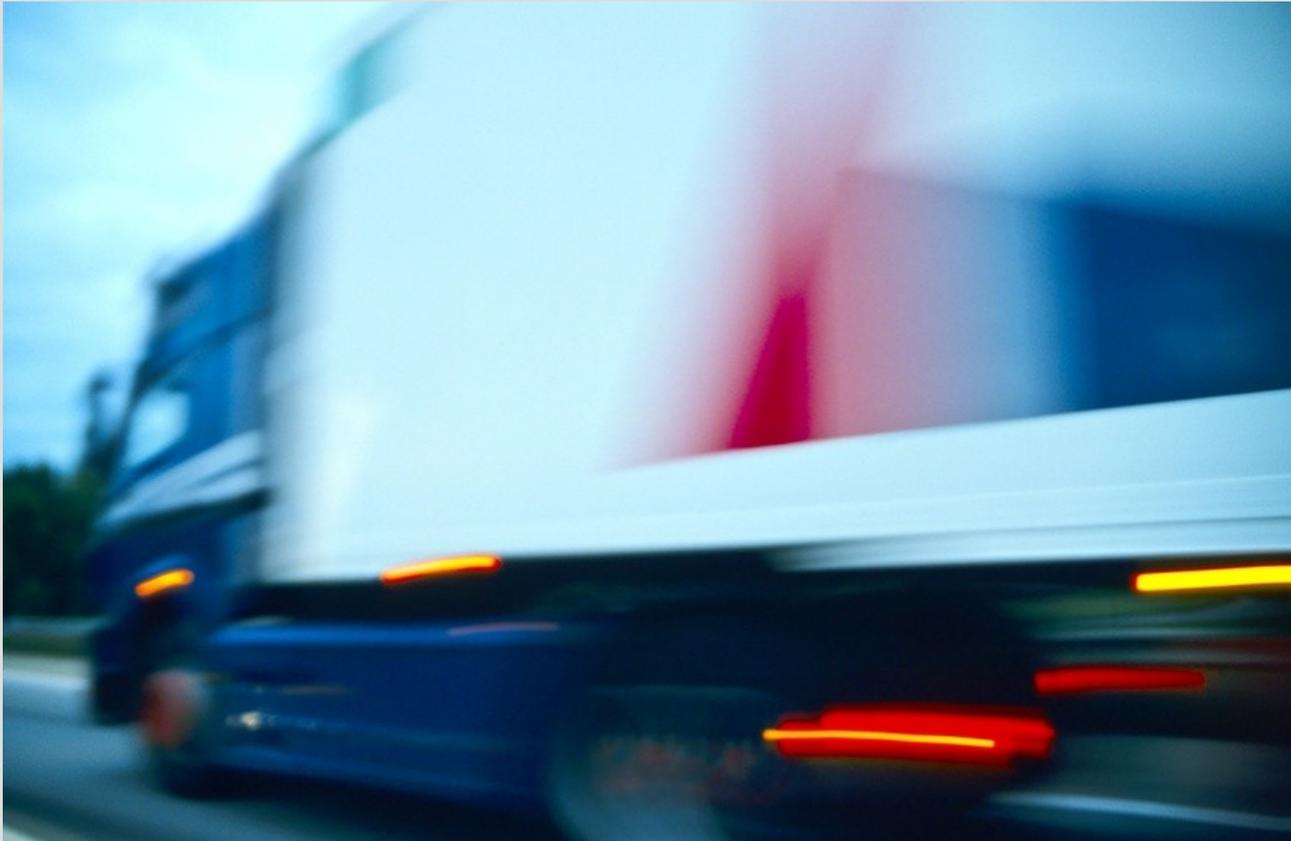
### 3. Prinzip: Entscheidungen treffen, so spät wie möglich



### 3. Prinzip: Entscheidungen treffen, so spät wie möglich

- Was sind die grossen Risiken oder Unbekannten im Projekt?
- Was würde sich am ehesten verändern?
- Wie können die Auswirkungen von Änderungen minimiert werden?
  
- Aber:
  - Entscheidungen nicht über den letzten vernünftigen Zeitpunkt hinaus verschieben
  
- Scrum:
  - Sprint-Rhythmus --> Entscheidungsfindung
  - Was klar ist, kann implementiert werden
  - Was unklar ist, soll noch abgeklärt oder verhandelt werden

## 4. Prinzip: Liefern, so schnell wie möglich



## 4. Prinzip: Liefern, so schnell wie möglich

- Warum?
  - Verderbung von Know-How
  - Cash Flow & Return on Investment
  - Wettbewerbsvorteil
- Werkzeuge
  - Soll-Bilanz in der Projekt-Planung
    - Was passiert, wenn das Projekt in 6, 12 oder 24 Monaten auf den Markt kommt?
    - Auswirkungen auf Marktanteil, Preispolitik?
- Wie?
  - Iterationen („Sprints“)
  - Parallele Entwicklung
  - Warteschlange-Theorie

# Parallele Entwicklung



# Parallele Entwicklung



# Warteschlange-Theorie



## Schlangen-Praxis

Quelle: <http://reptiles.net>

# Warteschlange-Theorie

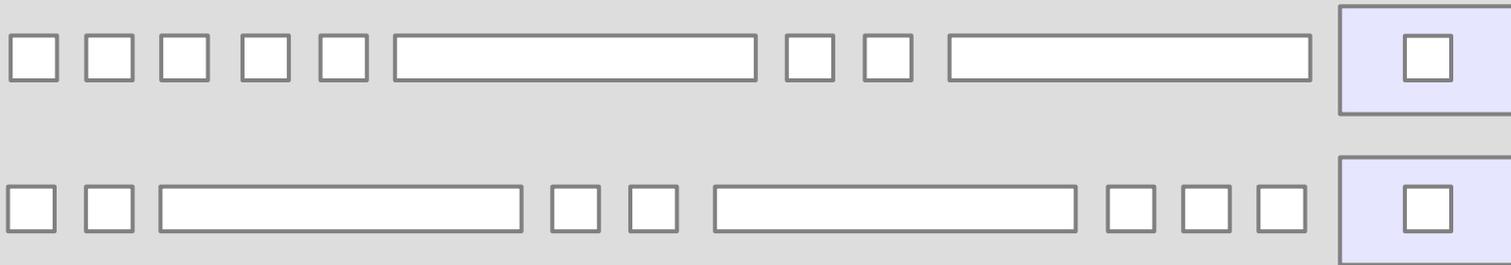


# Warteschlange-Theorie



# Warteschlange-Theorie

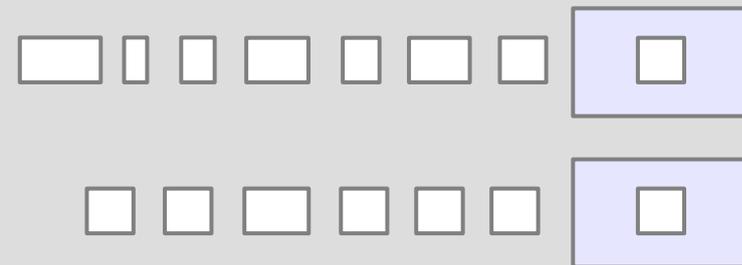
Anstehende Aufgaben



# Warteschlange-Theorie

- Module & Projekt sollen in „vernünftiger“ Zeit realisierbar sein
- Warteschlange kurz halten => Reaktionsfähigkeit

Anstehende Aufgaben



## Scrum

- > Benutzergeschichten innerhalb eines Sprints abschliessbar
- > Projekte werden erst 3 Monate im voraus genau spezifiziert
- > Neu Priorisierung jederzeit möglich
- > Nie mehr als 30 Tage von einer potentiellen Lieferung entfernt

## 5. Prinzip: Menschen respektieren



## 5. Prinzip: Menschen respektieren

- Motivation
- Selbstbestimmung
- Führung
  
- Scrum
  - Selbst organisierend
  - Team geht Commitments ein
  - Scrummaster: Schutz und Coach, nicht „Chef“

## 6. Prinzip: Integrität einbauen



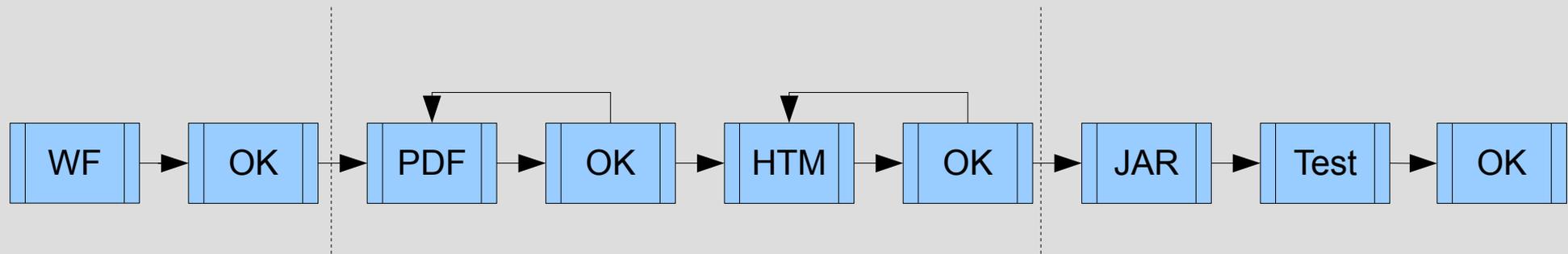
## 6. Prinzip: Integrität einbauen

- Wahrgenommene Qualität:
  - Das Produkt macht das, was „ich“ will, wie ich es machen will
  - Abnahme-Tests, Usability Tests, Fokus-Gruppen usw.
- Innere Qualität
  - Das Produkt macht das, was die Entwickler definiert haben
  - Das Produkt lässt sich leicht modifizieren, um neue- Anforderungen gerecht zu werden
- Werkzeuge
  - Benutzer-Testfälle, wenn möglich bevor die Implementierung beginnt (sog. „Test Driven Development“)
  - automatisierte Test Suite  
== Dokumentation wie gebaut („as built“)
  - Refactoring  
== Code verbessern, Architektur pflegen

## 7. Prinzip: Das Ganze verbessern



## 7. Prinzip: Das Ganze verbessern



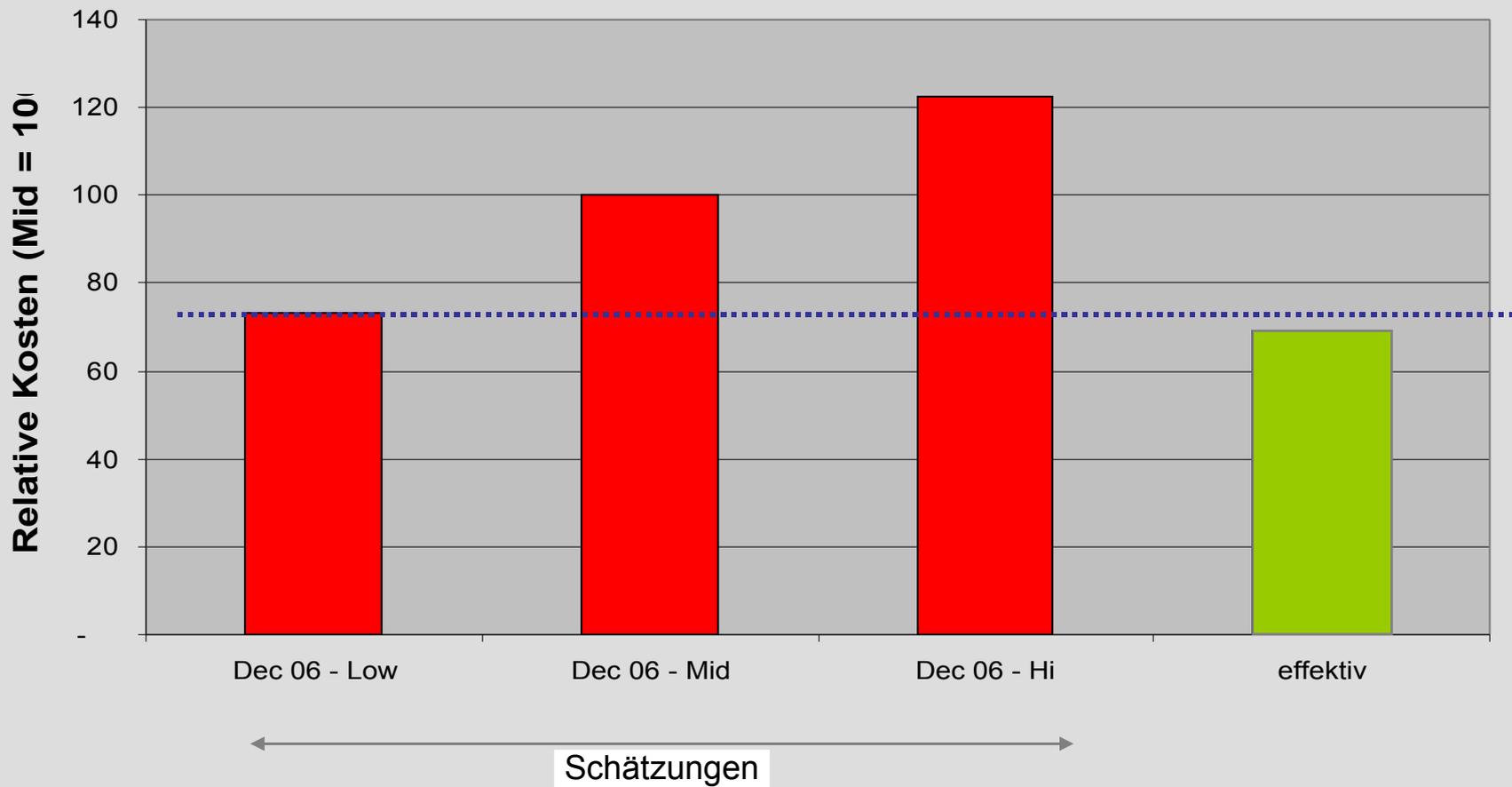
- Messen
  - Gesamt- vs. Etappenziele (Gefahr Suboptimierung)
  - Kosten, Termin oder Eignung zum Zweck (Fitness for Use)?

# Zusammenfassung

- Verschwendung eliminieren
- Lernen verstärken
- Entscheidungen treffen, so spät wie möglich
- Liefern, so schnell wie möglich
- Menschen respektieren
- Integrität einbauen
- Das Ganze verbessern

# Was bringt Lean?

## Budgetierte und effektive Kosten



# Die erste Schritte

- Starten Sie mit Scrum auf 2 Ebenen:
  - Entwicklungsprojekt
  - Veränderungsprozess

## Mehr Informationen

- [scrum-breakfast.blogspot.com](http://scrum-breakfast.blogspot.com)
  - Case Study: Guidewire mit Prof. Stuart Read, IMD (in englischer Sprache)
  - 4. Juni 2008
- [inside-scrum.blogspot.com](http://inside-scrum.blogspot.com)
- Lean Books
  - Implementing Lean Software Development – Mary & Tom Poppendieck
  - Lean Software Development – Mary & Tom Poppendieck
  - Toyota Production System: Beyond Large-Scale Production by Taiichi Ohno
- Lean as Chinese Proverbs
  - <http://ca.rroll.net/2008/04/20/lean-software-development-as-a-bunch-of-chinese-proverbs/>

**Besten Dank für Ihre Aufmerksamkeit.**



[peter@sierra-charlie.com](mailto:peter@sierra-charlie.com)