

# Scrum

## UPscaled

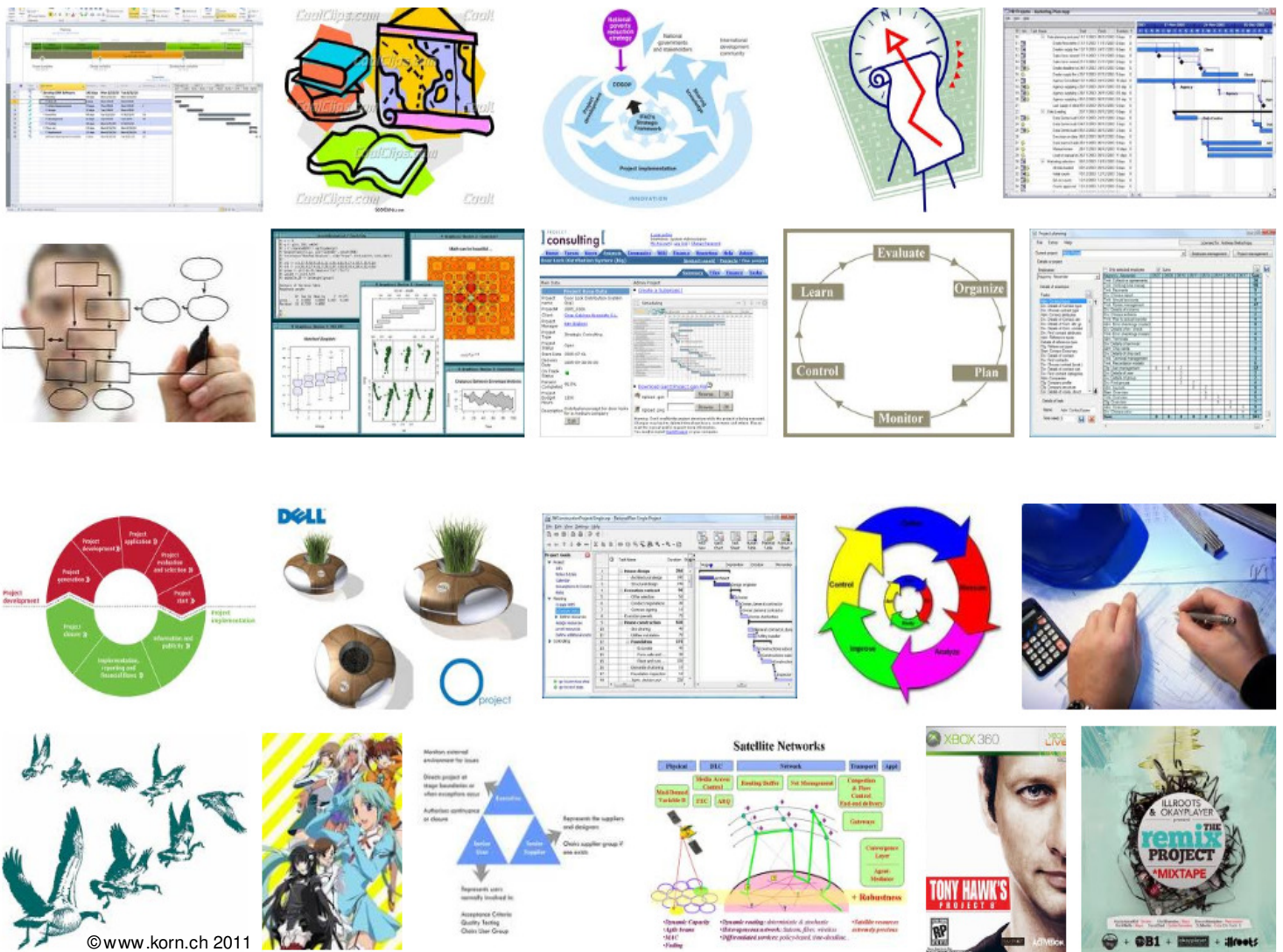
## ohne mürCS



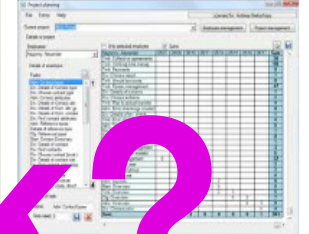
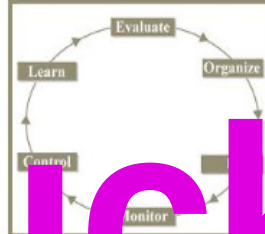
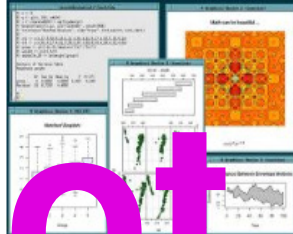
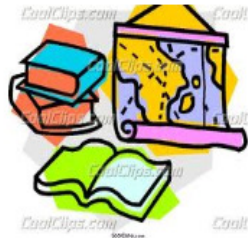
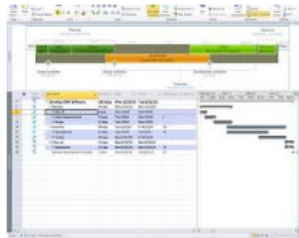
Dr. Hans-Peter Korn



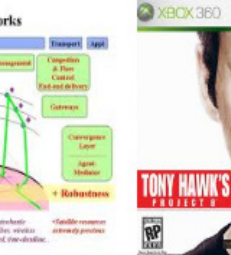
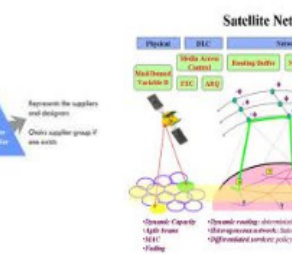
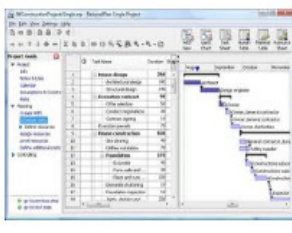
©www.korn.ch 2011



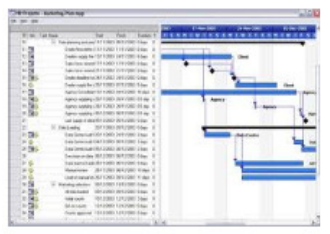
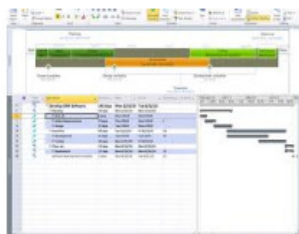
©www.korn.ch 2011



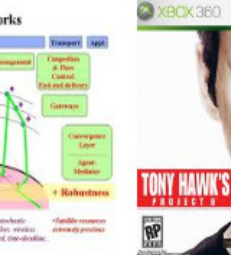
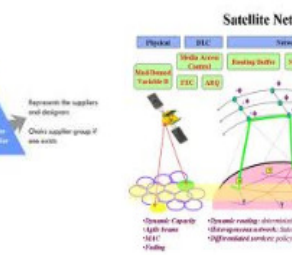
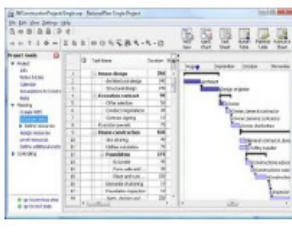
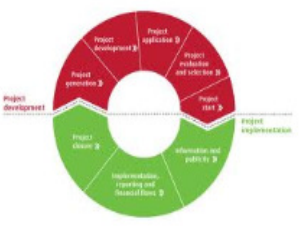
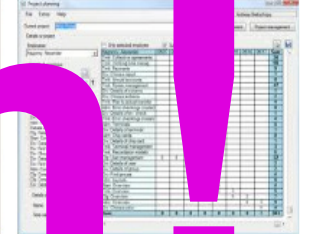
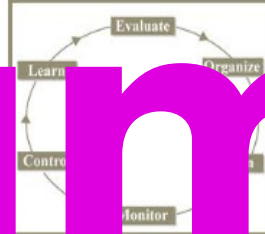
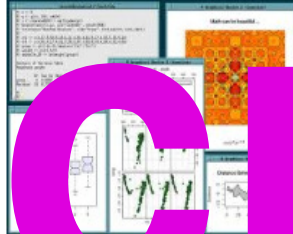
# got stuck?



©www.korn.ch 2011

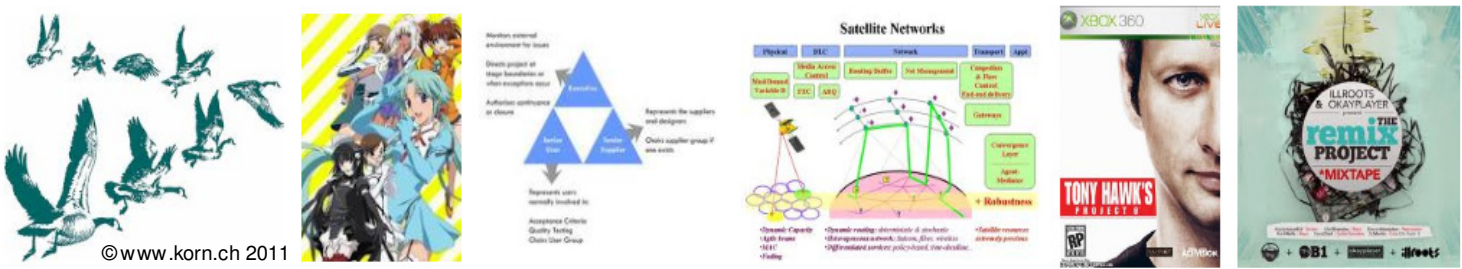
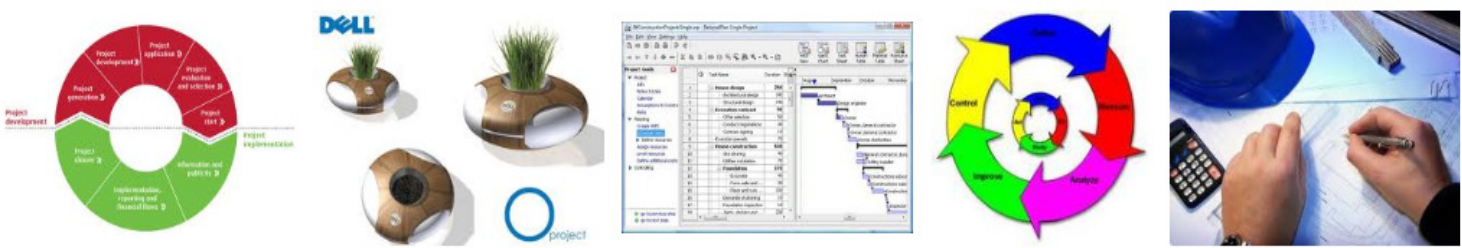
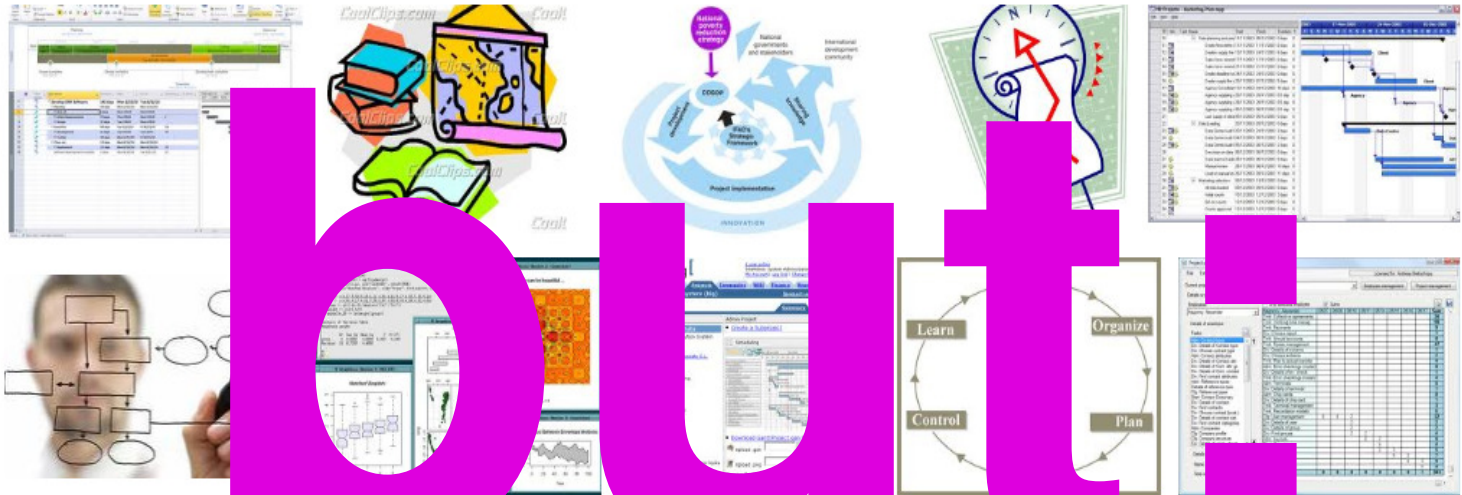


# Scrum!



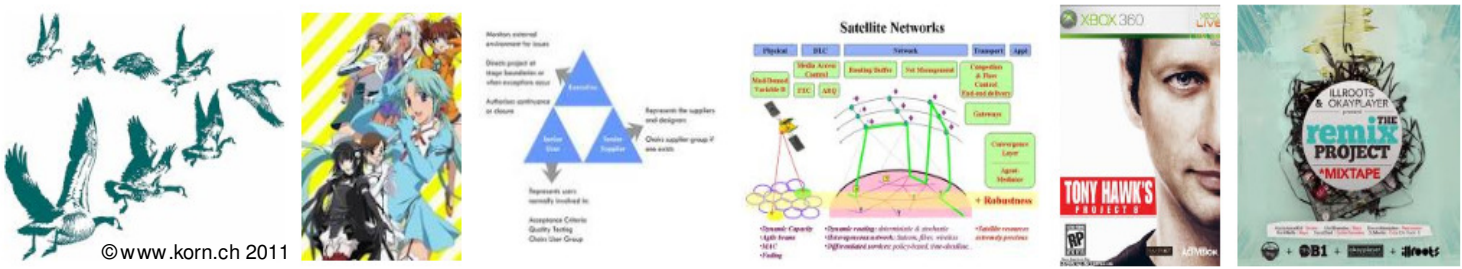
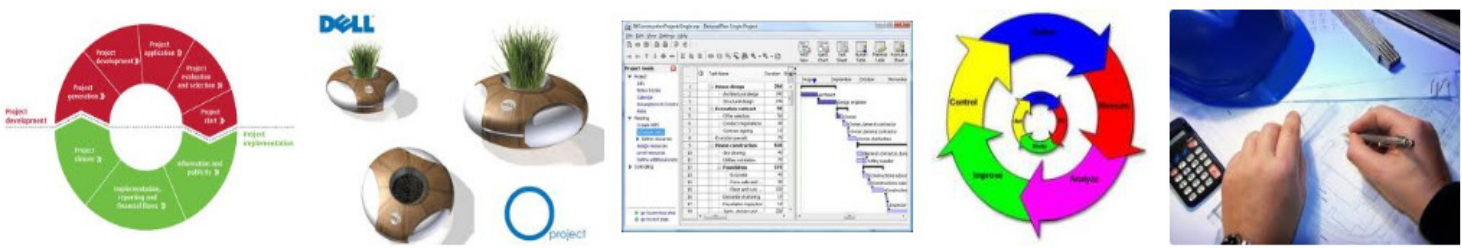
©www.korn.ch 2011

# about

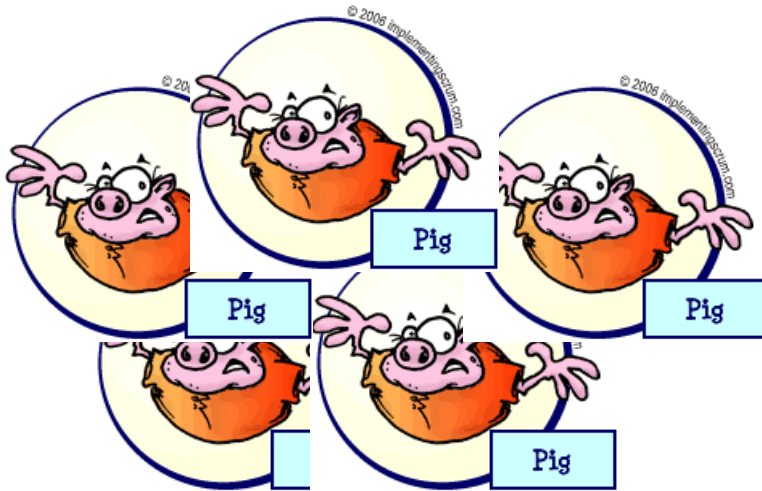
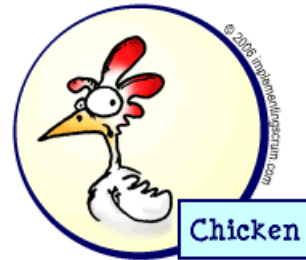


©www.korn.ch 2011

# does it fit?

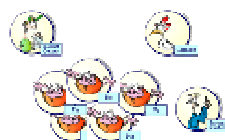


©www.korn.ch 2011



©www.korn.ch 2011

# easy!

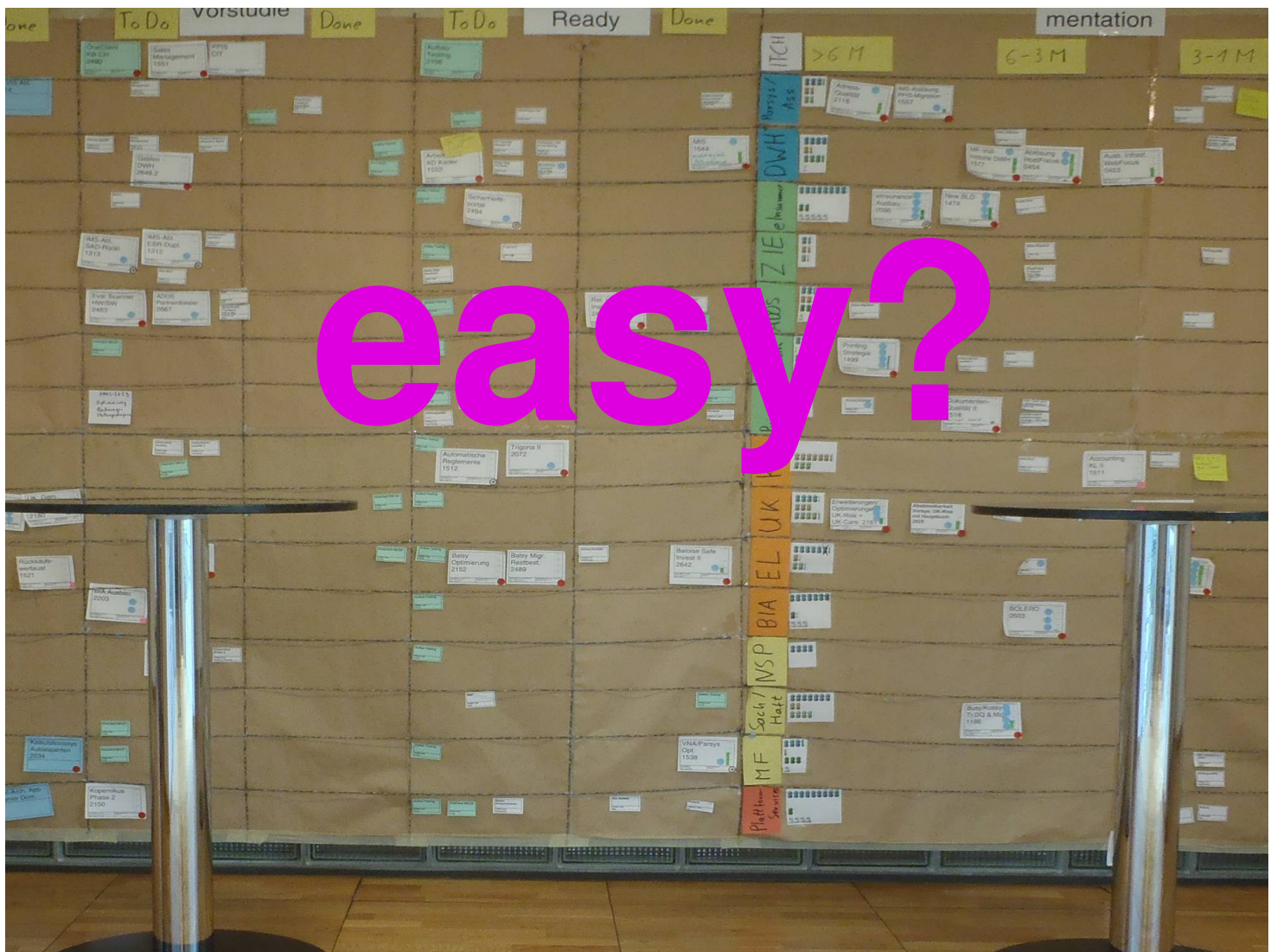


©www.korn.ch 2011

# easy?



©www.korn.ch 2011





hilft  
Scrum?

Scrum:

Illusion  
Realität

# Endlich: Schlanke Entwicklungs- methode

©www.korn.ch 2011

## „Container“ – reduced to the max

Auf das Minimum reduziertes Framework für das agile Vorgehen bei der Produktentwicklung.

Kann NICHT ANSTELLE aller bisher praktizierten Methoden und Techniken der SW-Entwicklung genutzt werden.

Problematisch ist eine Scrum-Einführung, wenn bisher keine Methoden und Techniken der SW-Entwicklung professionell genug praktiziert wurden und von Scrum erwartet wird, dass es damit künftig besser wird.

©www.korn.ch 2011

# Endlich: Rasante Effizienz- steigerung

© www.korn.ch 2011

## Schnelligkeit, Qualität, Effektivität

Nicht möglichst viel sondern das betrieblich relevanteste mit höchster Qualität zuerst.  
(„Cost of Delay“)

Rasches Feedback von den Nutzern →  
rasche Adaption der jeweils relevantesten  
Funktionen.

Scrum erhöht primär die Effektivität und  
unterstützt dabei die auf der Nutzerseite  
betrieblich nötige Flexibilität und Dynamik.

© www.korn.ch 2011

# Endlich: Legalisierte Improvisation

© www.korn.ch 2011

## Strikte Professionalität

Keine Legitimation, schnell und improvisierend zu liefern.

Sondern: Innerhalb strikter und kurzer Zeitrahmen hoch professionell Planen, Konzipieren, Realisieren, Testen.

Scrum macht schlampige Arbeit unmittelbar nach jedem Sprint für alle Stakeholder gnadenlos transparent.

© www.korn.ch 2011

# Endlich: Die IT arbeitet auf Zuruf

© www.korn.ch 2011

## Agiles Anforderungs- management

Kein Freipass für die Anwender, Anforderungen ohne gründliche Vorüberlegungen als „Wunschkonzert“ zu formulieren und alle paar Wochen beliebig zu ändern.

Anforderungen werden vom Product Owner fortlaufend entgegengenommen, soweit als nötig analysiert, abgestimmt und priorisiert. Erfüllung nicht nach „FIFO“ sondern pro Sprint nach Priorität. Neue Anforderungen und Prioritäten vor – nicht während – jedes Sprints möglich.

© www.korn.ch 2011

# Endlich: Selbstorgani- sation statt Vorschriften

© www.korn.ch 2011

## Teams als Ganzes selbst- verantwortlich innerhalb klarer Spielregeln

Den Rahmen für die Selbstorganisation legen nicht die sich selbst Organisierenden fest.

Keine „Basisdemokratie“ sondern klare Verantwortlichkeiten und „servant leadership“

Den Rahmen für die selbstverantwortliche Teamarbeit setzen der Product Owner und der Scrum Master auf Basis der Scrum-Vorgaben für „Artefakte“ und „Timeboxes“.

© www.korn.ch 2011

**Passt etwas  
nicht?**

**Dann:  
Hände weg von  
Scrum!**

©www.korn.ch 2011

**Wieviel Agilität  
braucht Ihr  
Unternehmen?**

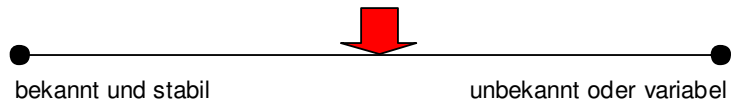
**→ Agilitätstest**

©www.korn.ch 2011

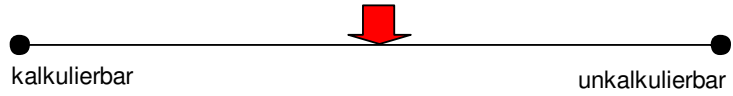
„traditionell“

„agil“

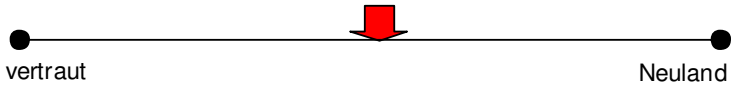
Funktionsanforderungen



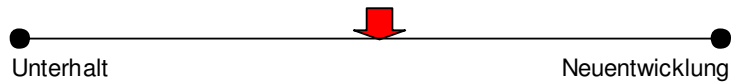
ökonomisches Risiko



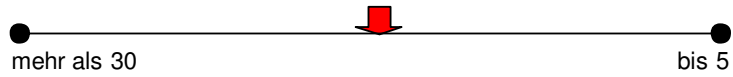
Technologie



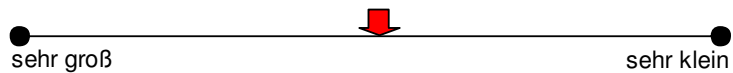
Projekttyp



Anzahl Teammitglieder



Businesswissen der Teammitglieder



# Wie agilophil ist Ihr Unternehmen?

## → Kulturtest

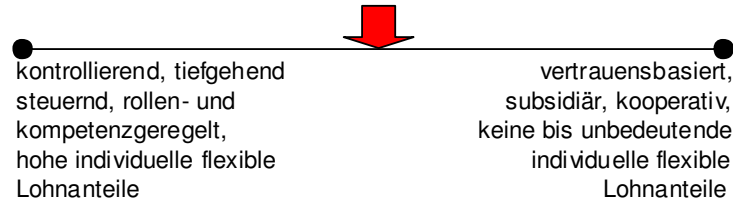
agilophob

agilophil

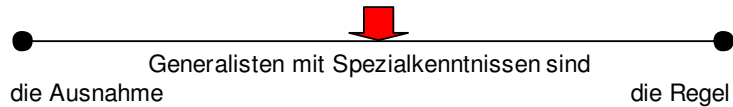
„management- und individualitätsorientiert“

„emergenz- und teamorientiert“

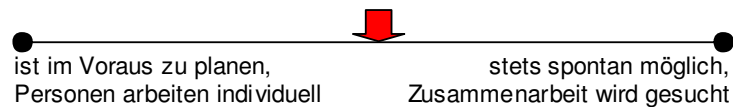
## Rahmenkultur



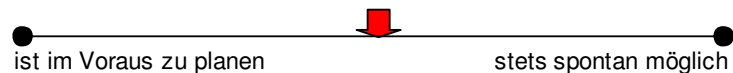
## Kompetenzen im Team



## Kollaboration im Team



## Nutzer-Einbezug



# Wenn das passt: Wie Scrum im Unternehmen einführen?



©www.korn.ch 2011

## **Begeisterte Führungskraft ein bis zwei erfolgreiche Pilotprojekte**

wenig externe Abhängigkeiten  
keine Wartung von Altlasten  
motiviertes Team (weil Pilot)

### **HURRA!**

Bisherige Teamleiter und interessierte Entwickler werden motiviert, SM bzw. PO zu werden

Scrum-Trainings (ca. 2 Tage) für alle SM / PO

Kurze Scrum-Einführungen für alle Entwickler

**„Ab 1.xxx.201n arbeiten alle nn Teams nach Scrum“**

**Zu Beginn Begleitung durch externe Coaches**

→ „Wie können wir möglichst viel vom Bisherigen in den „Scrum-Container“ packen?“

→ Scrumade (Scrum als Fassade)

→ nach ein paar Scrumaden-Sprints: „Warum müssen wir ..... ändern? Wir machen doch Scrum!“

©www.korn.ch 2011

# Scrum, ein framework für inkrementelles Vorgehen, inkrementell einführen!



©www.korn.ch 2011

## Alternative 1: Inkrementelle Einführung quer über alle (geeigneten) Entwicklungsprojekte

### Inkrement 1

- ☞ Je ein PO pro Projekt-Team. Teamleiter bleiben.
- ☞ PO erstellen Product Backlogs als einzige Basis für die Arbeit der Teams.
- ☞ Jedes Projekt plant Releases mindestens alle 3 Monate.

### Inkrement 2 für Projekte, die mehr in Richtung Scrum gehen möchten:

- ☞ Entwicklungsarbeit innerhalb der Releases in Sprints (2 bis 4 Wochen) unterteilen.
- ☞ Planning, Review und Retrospektive und Sprint Backlog einführen.
- ☞ Teamleiter statt Erteiler von Detailaufträgen und Detail-Koordinatoren jetzt Rahmensetzer und Unterstützer („dienende Führungskraft“)
- ☞ Teams erhalten möglichst viel Eigenverantwortung und Raum zur Selbstorganisation.

### Inkrement 3 für Projekte, die voll auf Basis von Scrum arbeiten möchten:

- ☞ Teamleiter werden Scrum Master in einem anderen als ihrem bisherigen Team.
- ☞ Entwickler gehören neu zu "Entwicklerpools" (je rund 25 Personen) mit je einem personellen Vorgesetzten.

*Alles begleitet von scrumerfahrenen Projektmanagement-Coaches (z.B. PMI-ACP)*

©www.korn.ch 2011

# Alternative 2: Inkrementelle Einführung für jeweils nur einige (geeignete) Entwicklungsprojekte

## Inkrement 1

Die agilsten (siehe „Schieberegler“) 2 bis 3 Teams stellen vollständig auf Scrum um:

- ➔ Nominierung der PO, Erarbeiten der ersten Entwürfe des Product Backlogs (unterstützt von Coaches)
- ➔ Nominierung der SM
- ➔ Einrichten der Teamräume (je ein grosser Raum pro Team)
- ➔ Entwickler der Teams gehören neu zu einem "Entwicklerpool" mit rund 25 Personen und einem personellen Vorgesetzten. Das ist weder einer der PO noch der SM.
- ➔ Gemeinsame Scrum-Schulung der Teams, PO und SM unter Benutzung der Product Backlog - Entwürfe
- ➔ Pro Team: erstes Release Planning Meeting, Sprint Planning Meeting, Start der Sprints

## Inkrement 2 ... n

Bestimmen der nächsten 2 bis 3 interessierten Teams und wie oben vorgehen.

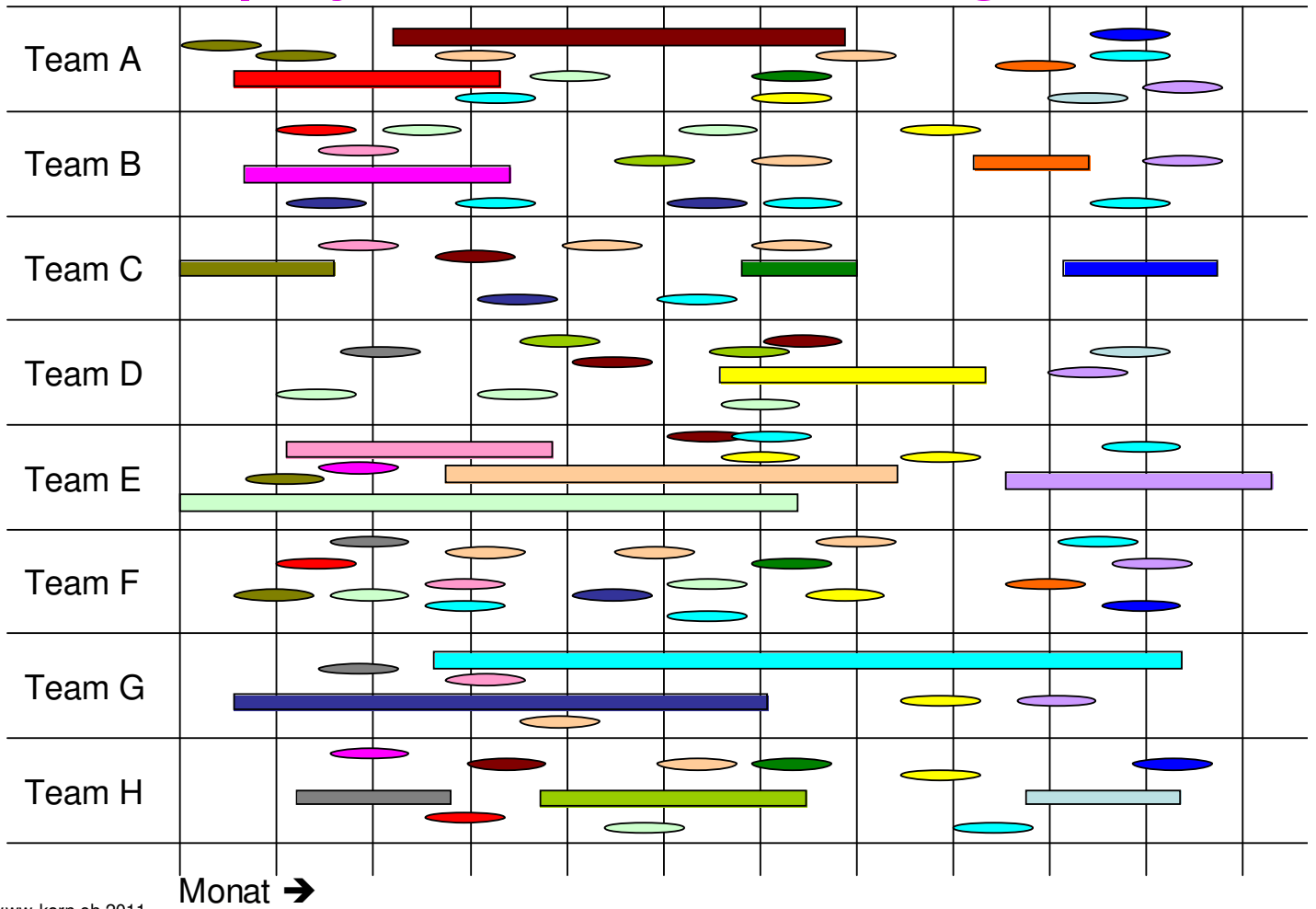
*Alles begleitet von scrumerfahrenen Projektmanagement-Coaches (z.B. PMI-ACP)*

© www.korn.ch 2011



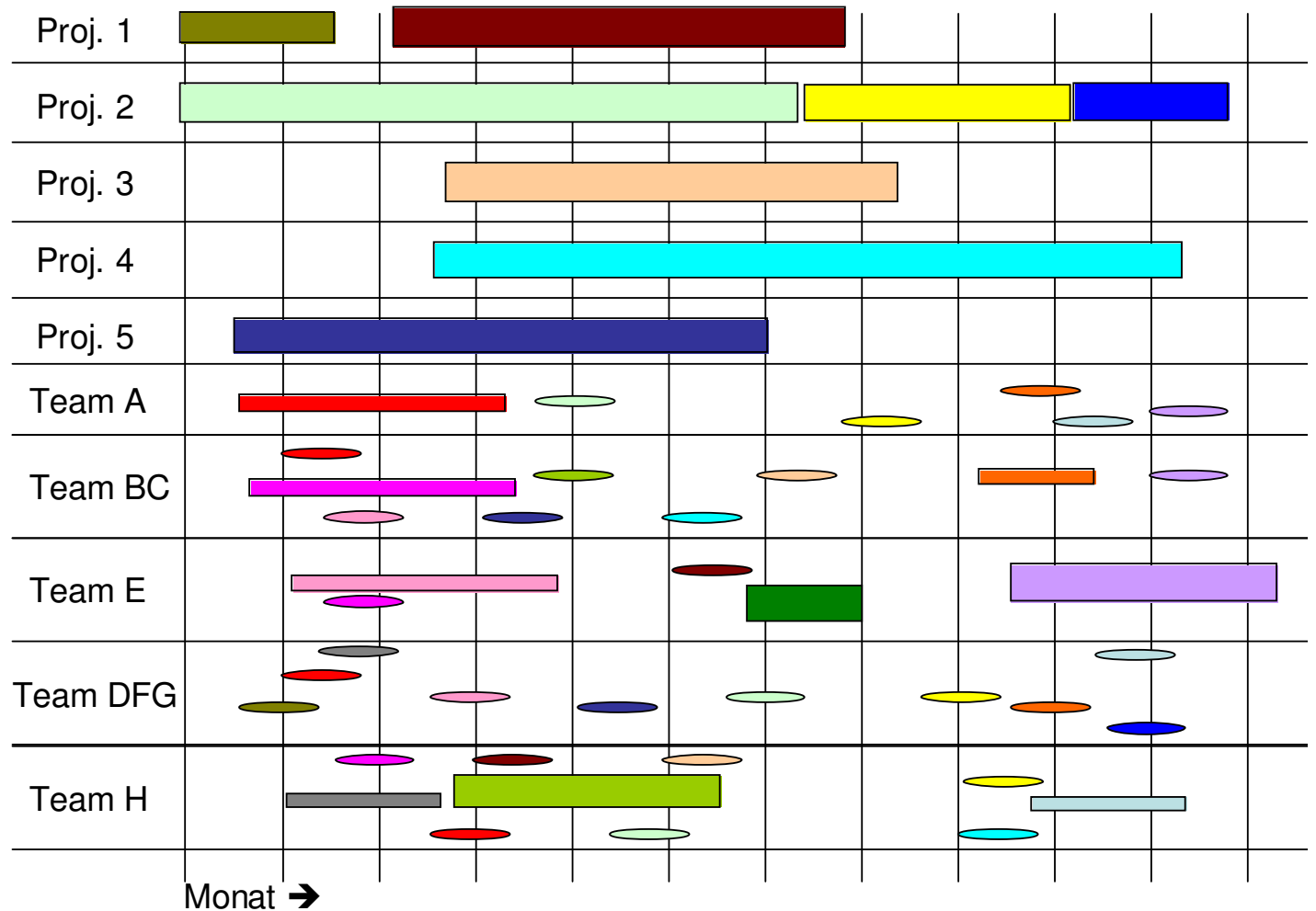
© www.korn.ch 2011

# Multiprojekt / Multiteam - Management



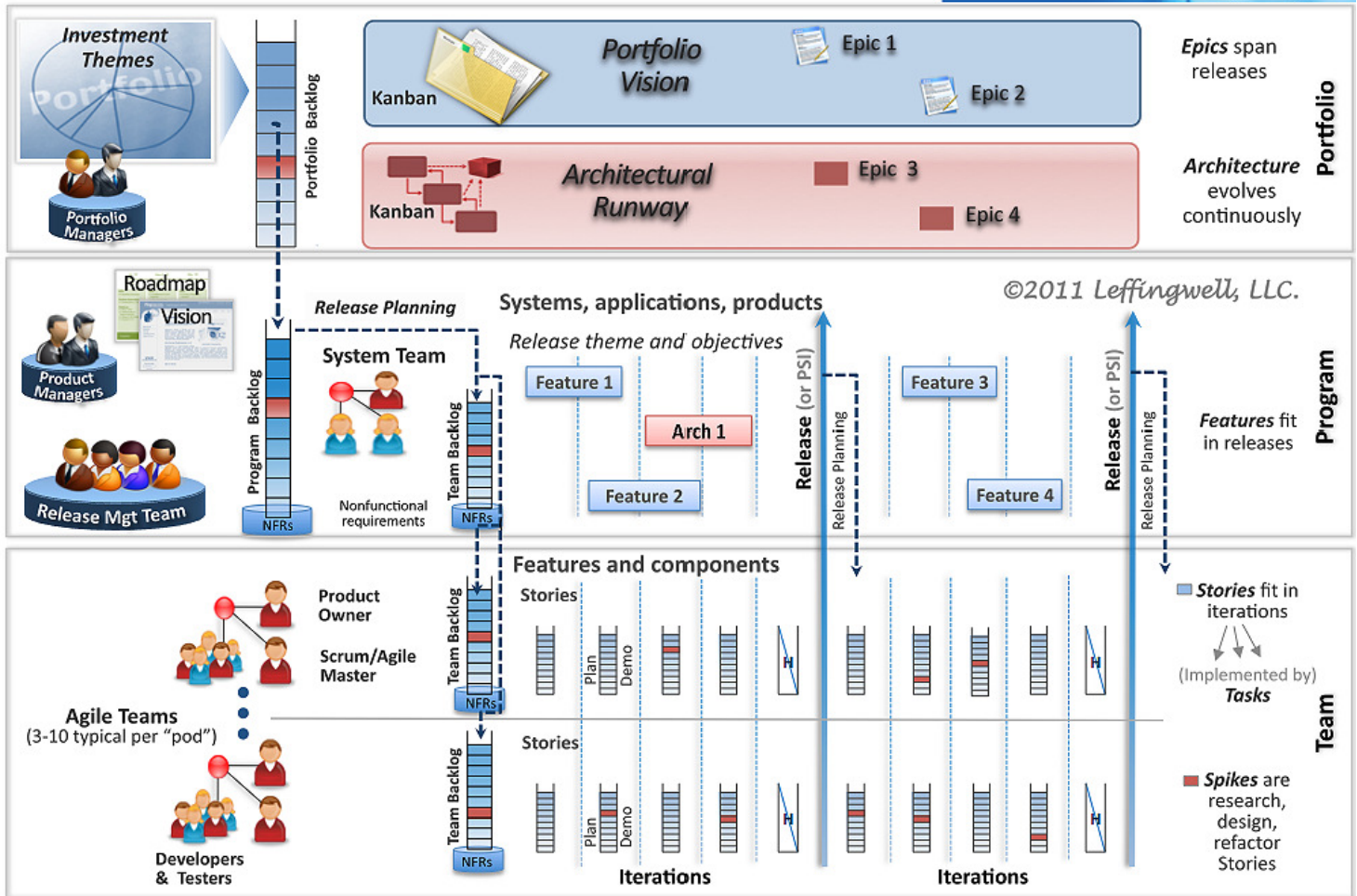
©www.korn.ch 2011

## Bereinigung: Projekt- und Produkt-Teams



# Scaled Agility Framework (Big Picture)

Leffingwell, LLC.



See *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise* and [www.scalingsoftwareagility.wordpress.com](http://www.scalingsoftwareagility.wordpress.com)

Die Langfassung dieser Präsentation erscheint unter dem Titel „Scrum ohne murcS: Agile Projektmanagementpraxis“ im Buch <http://www.it-projektmanagement-buch.de/>