Scrum

UPscaled

ohne murcS

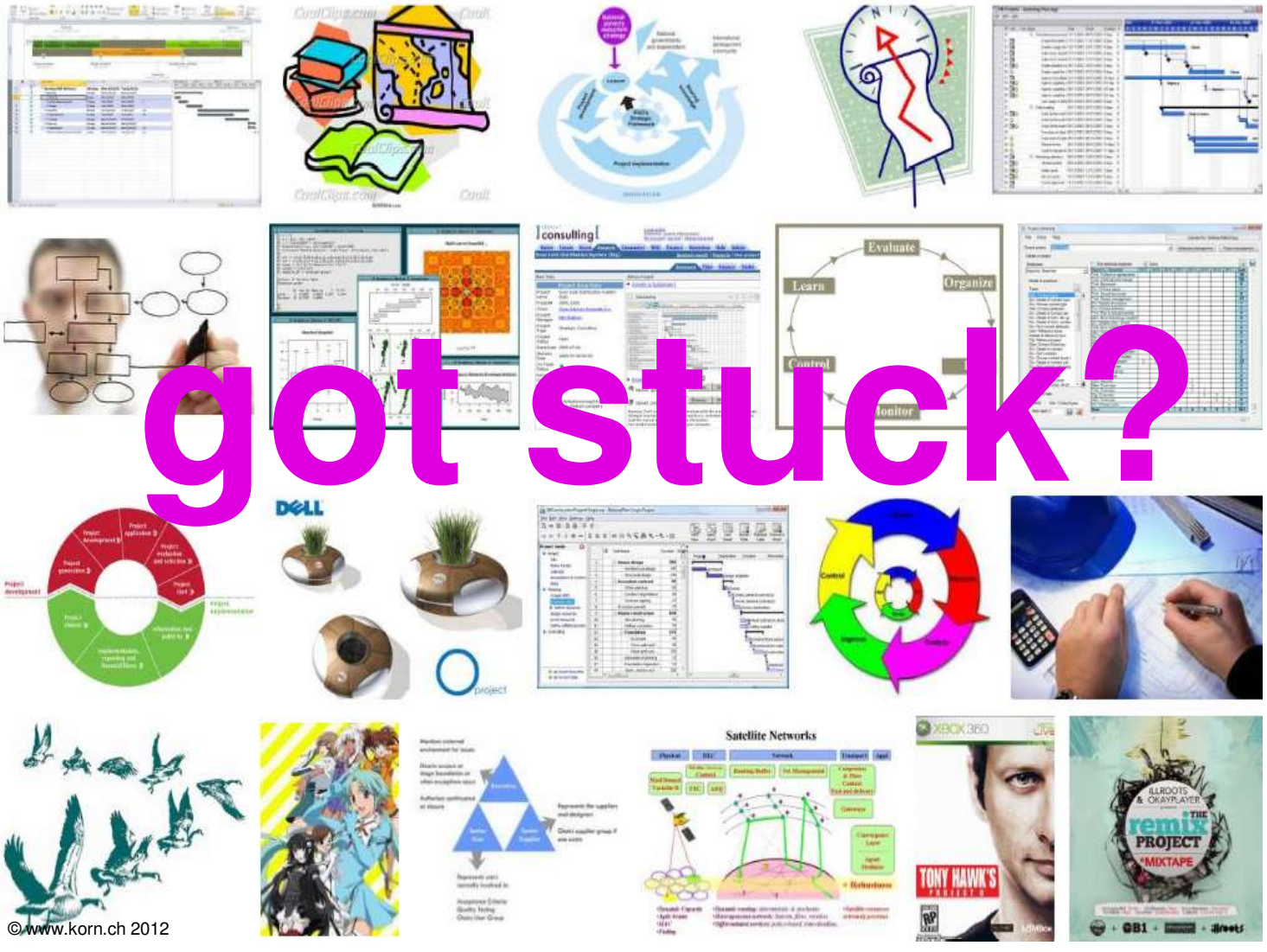
Dr. Hans-Peter Korn

KORN AG

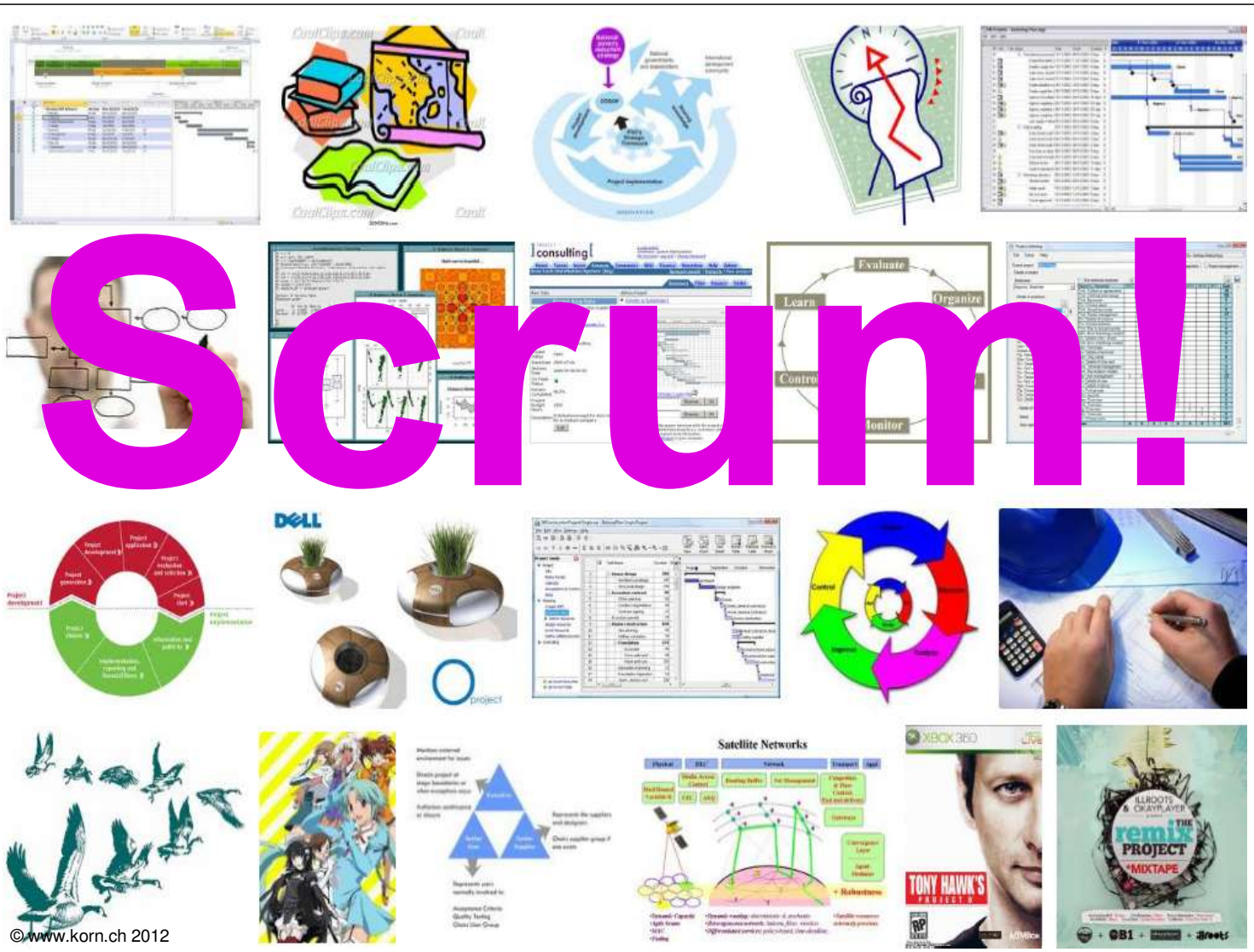
© www.korn.ch 2012

Das ist ein Teaser für
unternehmensspezifische
inhouse-Workshops
als Kickoff zum
(Re)Design der Agility.

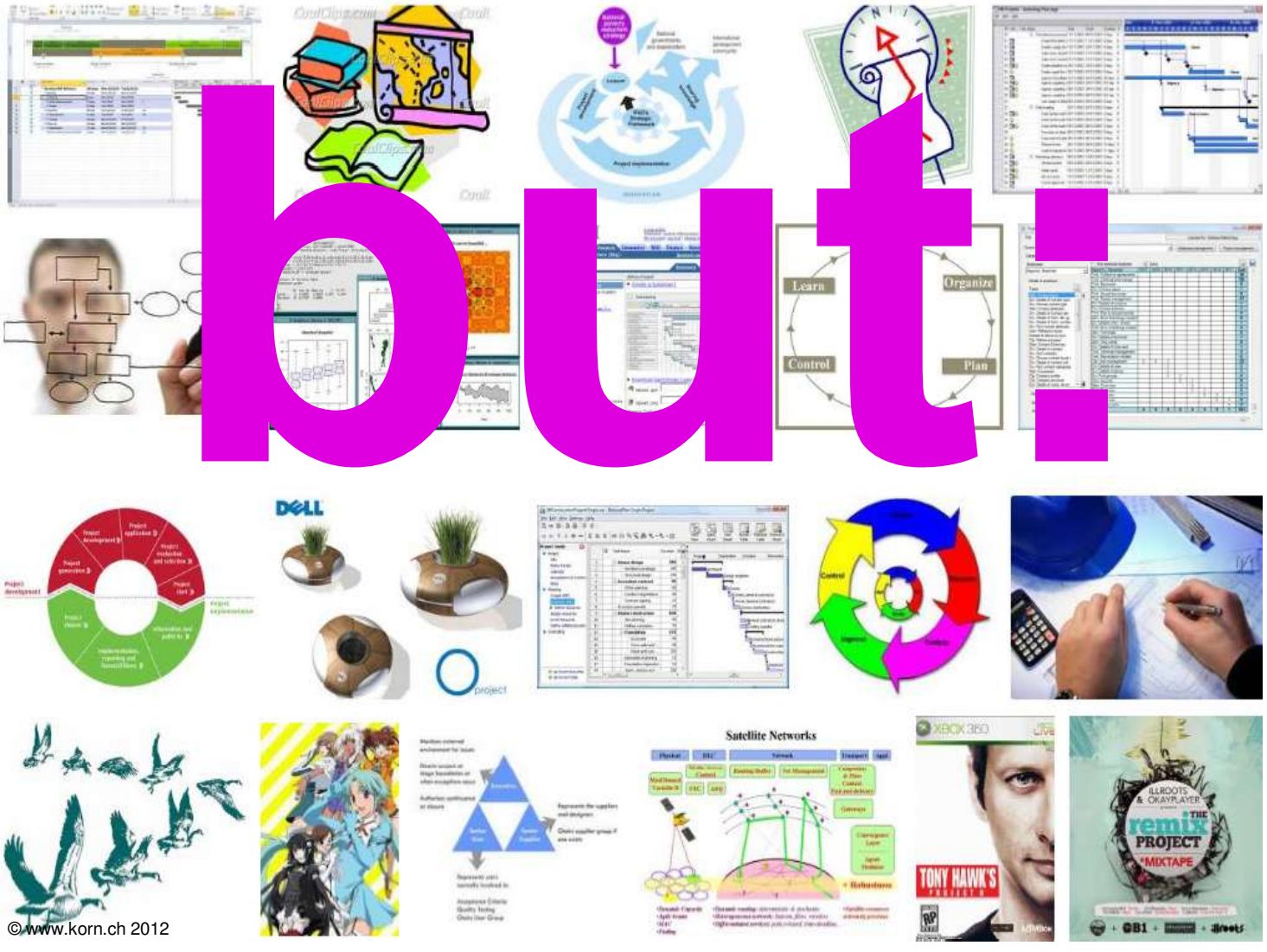
© www.korn.ch 2012



© www.korn.ch 2012



© www.korn.ch 2012



© www.korn.ch 2012

© www.korn.ch 2012

hilft Scrum wirklich?

© www.korn.ch 2012

Story(1):

(1) Ähnlichkeiten mit real existierenden Unternehmen sind möglich und beabsichtigt

© www.korn.ch 2012

2 bis 3 Jahre vor heute:

Begeisterte Führungskraft in der IT

Ein bis zwei erfolgreiche Pilotprojekte

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

HURRA!

1,5 bis 2 Jahre vor heute:

- 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 jetzt arbeiten alle 7 Teams / 60 Entwickler nach Scrum“

Zu Beginn (1 – 3 Monate) Begleitung durch externe Coaches

1,5 Jahre bis heute:

„Wie können wir möglichst viel vom bisher Angenehmen in den „Scrum-Container“ packen und Unangenehmes „dank“ Scrum aufgeben?“

➔ Scrumade (Scrum als Fassade)

➔ nach ein paar Scrumaden-Sprints: Ein neu „eingekaufter“ SM will richtiges Scrum machen. Team: „Warum müssen wir ….. ändern? Wir machen doch Scrum! Und es funktionert ja …“

© www.korn.ch 2012

Scrum Guide

Der gültige Leitfaden für Scrum:
Die Spielregeln

Scrum ist kostenlos und wird in diesem Leitfaden angeboten. Die Rollen, Artefakte und Ereignisse von Scrum sind unantastbar. Obwohl es möglich ist, nur Teile von Scrum zu implementieren, ist das Ergebnis nicht Scrum. Scrum existiert nur in seiner Gesamtheit - und funktioniert gut als Container für andere Techniken, Methodologien und Praktiken.

Juli 2011

© www.korn.ch 2012

heute:

- Alle Teams machen „Daily“ (tw. nur 2/Woche) und erzählen dabei irgend etwas reihum
- Alle nutzen Jira, vor allem aber als „komplette Stundenkontrolle“
- Nur 1/4 der Teams pflegen auch ein physisches Taskboard
- SM (jeweils einer der Entwickler) ist primär „Taskboardadmin“
- Sprintplannings auf Basis Person / Tasks / Stunden
- PO (vm. Teamleiter) plant pro Person (u.a. wegen Auslastung aller Spezialisten)
- Viele „Pseudostories“, alle Stories isoliert und erstellt ohne RE-Mittel
- Keine Backlog-Groomings
- Items im Gesamt-Backlog nur mit grober Priorität, aber weder geschätzt noch gereiht
- Planung des PO nur für nächsten, tw. übernächsten, Sprint
- Review = Statusmeeting für PO, ohne Stakeholder (weil jedes Team nur - nicht demonstrierbare - Teile von „end to end“-Funktionen realisiert)
- Eigentliche Inkremente pro Release (alle 6 Monate), dann erst Nutzerfeedbacks
- Keine Retrospektiven (weil immer dasselbe und ohne Wirkung)
- Im Sprint Einzelarbeit an eigenen Tasks mit viel unerwarteten Zusatzanalysen
- Viele „Stories“ gleichzeitig in Arbeit damit alle etwas tun können
- Bei Sprintende stets nur 50 - 60 % der Stories „fertig“ (= bereit für Integrationstest)
- Viele - überwiegend nicht als Tasks sichtbare - ungeplante Kleinarbeiten („Bugs“, Support, Korrekturen aus Integration früherer Sprintergebnisse) und unterschätzte Umfänge (z.B. wegen Zusatzanalysen und Fehlannahmen)
- „Scrum of Scrum“ vor etlichen Monaten wegen Ineffektivität aufgegeben
- Seit etwa 1,5 Jahren keine „Konzepte“ mehr und kein aktualisiertes Applikations- und Systemmodell ➔ Jeder Change Request erfordert „Systemarchäologie“
- Bei allen Teams und Stakeholdern Gefühl der „Überadministration“ und eines hektischen, konzeptlosen, „Sprint-zu-Sprint-Gehetzes“
- **Aber: Deutlich bessere Kommunikation im Team, insbesondere dank „Daily“**

© www.korn.ch 2012

Höchste Zeit für redesigned Agility

© www.korn.ch 2012

Zentral:
Abbau der nach wie vor ungetilgten „Vorgehens-Schulden“

- Welche der strategisch relevanten Probleme löst Scrum tatsächlich?
- Was erwartet das Management effektiv von Agilität und Scrum?
- Wie passt „SW-Agilität“ zu den weiterhin sequentiellen Phasen und Gates der übergeordneten Produktentwicklung und zum weiterhin "traditionellen" Portfolio- und Programm-Management?
- Wie passen nutzerbezogene Releases (ca. 3 - 6 Monate) zu entwicklungsseitigen Sprints (ca 2 - 3 Wochen)? Was ist deren Nutzen?
- Was sind die "Produkte" z.B. eines unternehmensinternen IT-Bereichs und der einzelnen Teams?
- Was genau bedeutet "potential shipable increment" bei einem komponenten-spezifischen Team?
- Wie funktioniert die teamübergreifende Koordination?
- Wie ermöglichen wir teamübergreifende Continuous Integration / Delivery und Testautomation in heterogenen Systemlandschaften?
- Arbeitet die IT jetzt nur noch sprintweise "auf Zuruf" ohne übergeordnete Fach- und DV-Konzeption und ohne Architekturrahmen?
- Wer genau übernimmt die Rolle des Product Owners so, dass die damit verbundenen Aufgaben tatsächlich erfüllbar sind?
- …

© www.korn.ch 2012

Scrum:

Illusion

Realität

© www.korn.ch 2012

Endlich: Schlanke Entwicklungs-methode

Scrum = „Container“ als *Vorgehens-Framework*; kein Methodenersatz

Endlich: Rasante Effizienz-steigerung

Adaptivität, Qualität, *Effektivität;* insgesamt nicht schneller / billiger

Endlich: Kein „big bang“ sondern rasch nutzbare Teillieferungen

Nicht nur inkrementell und adaptiv sondern vor allem auch *„agil“*

Endlich: Legalisierte Improvisation

Strikte *Professionalität*

Endlich: Die IT arbeitet auf Zuruf

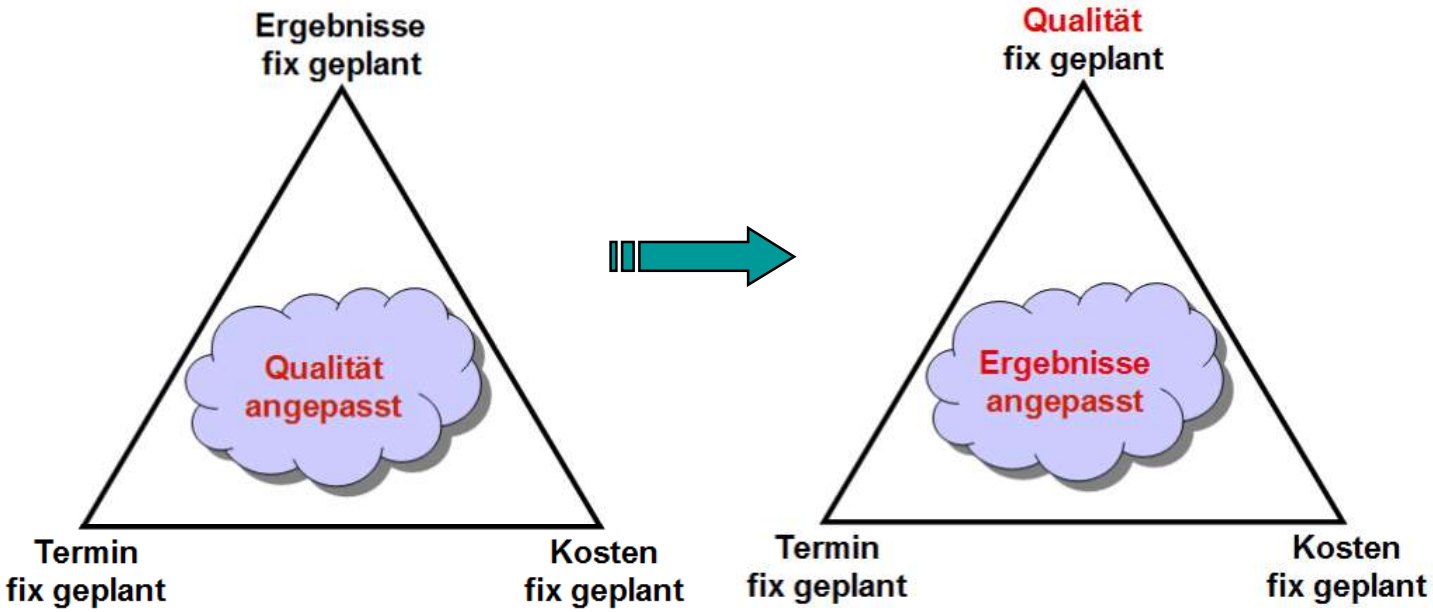
Agiles *Anforderungs-management*

Endlich: Selbstorgani-sation statt Vorschriften

Teams als Ganzes selbstverantwortlich innerhalb klarer *Spielregeln*

© www.korn.ch 2012

Mindchange nötig:

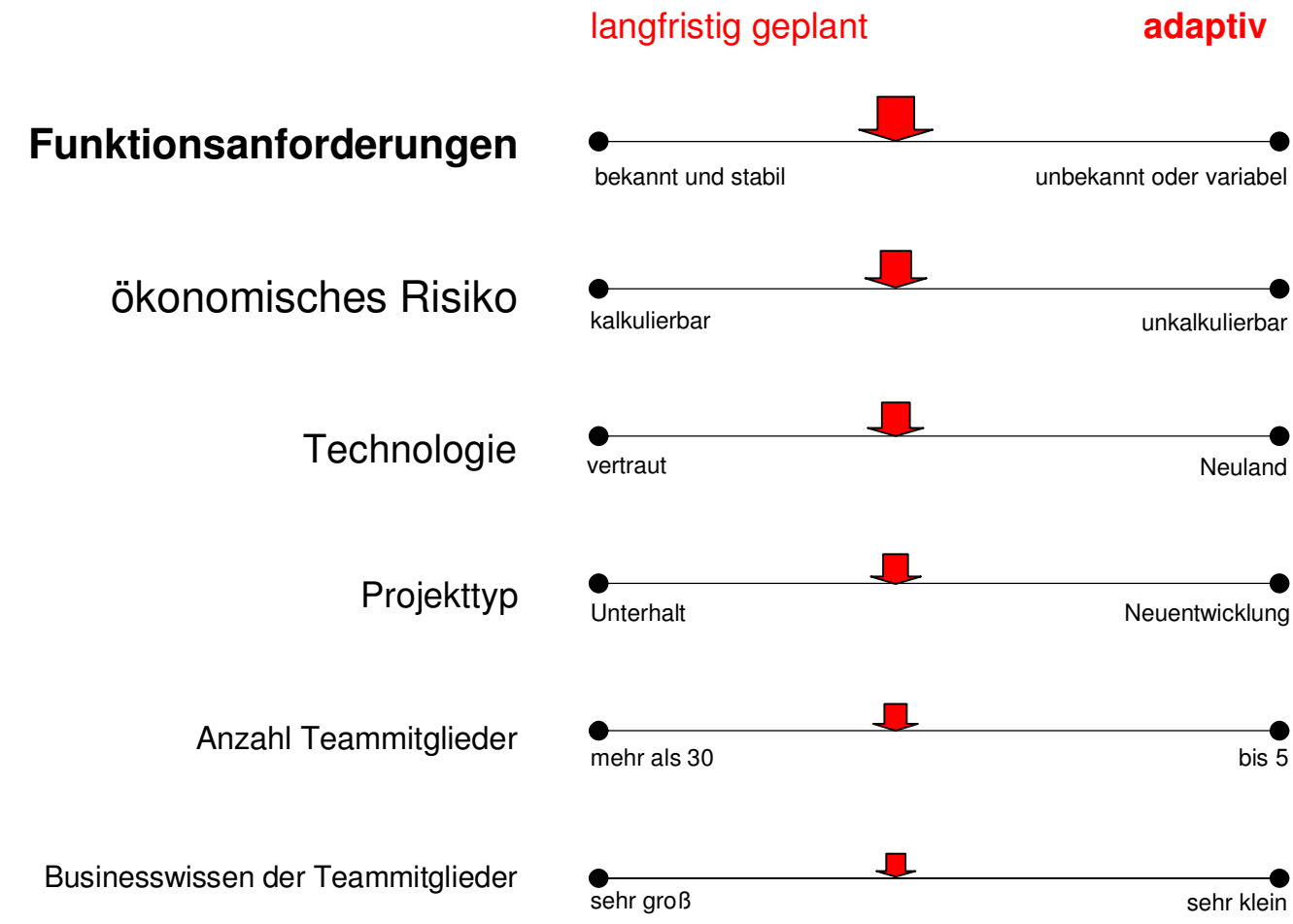


© www.korn.ch 2012

Wieviel Adaptionsfähigkeit braucht Ihr Unternehmen?

➔ Volatilitätstest

© www.korn.ch 2012

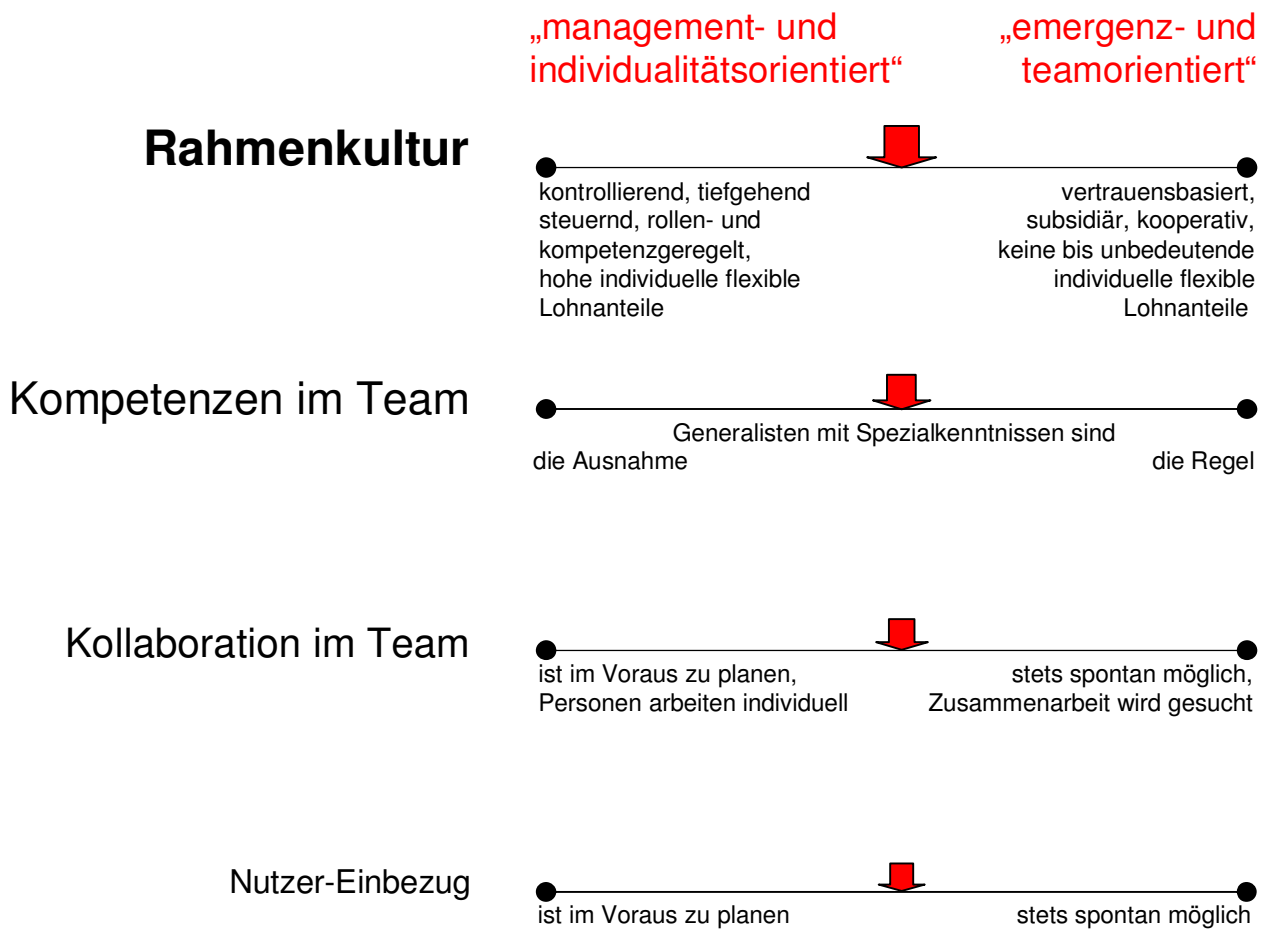


© www.korn.ch 2012

Wie agilophil ist Ihr Unternehmen?

➔ Kulturtest

© www.korn.ch 2012



© www.korn.ch 2012

Wenn „agilophil“ & hohe Adaptivität nötig:

Wie Agilität im Unternehmen neu einführen oder „redesignen“?

© www.korn.ch 2012

© www.korn.ch 2012

Agilität als adaptives und inkrementelles Vorgehen schrittweise einführen!

© www.korn.ch 2012

Inkrementelle Einführung / Überarbeitung quer über alle (geeigneten) PRODUKT-Teams

Schritt 1

- „**Produkte“ bzw. „Services“** identifizieren und Teams (als Produkt- bzw. Service-„Supplier“) danach strukturieren.
- Je ein **Product Owner** bzw. **Service Owner** (als Repräsentant des „Customers“) pro Produkt- (bzw. Service-)Team.
- Verantwortung für **„WAS“ und „WIE“ trennen:**
  - PO erstellen **Product Backlogs** als einzige Basis für das „WAS“ der Arbeit der Teams.
  - Teamleiter als „WIE-Verantwortliche“ etablieren. Sie sind Vorgesetzte der Entwickler, nicht aber der POs.
- Pro Produkt oder quer über alle Produkte **Releases** mindestens alle **3 Monate**
- **Abbau der „Vorgehens-Schulden“**, z.B:
  - Welche strategisch relevanten Probleme sind zu lösen?
  - Was bedeutet für uns "Agilität", was soll sie - strategisch relevant - konkret bewirken?
  - Abstimmung mit der IT-übergordneten Produktentwicklung und dem Portfolio-, Programm- und Release-Management
  - DoR & DoD der "potential shipable increments" pro Release und (später möglichen) Sprints und pro Team; (Re-)Professionalisierung des Requirement Engineerings
  - Teamübergreifende Koordination pro Produkt und Release
  - (Re-)Professionalisierung der SW-Entwicklung; TDD & ATDD; Integration von Entwicklung & Test
  - Continuous Integration, Continuous Delivery und Testautomation
  - Reduktion der Systemheterogenität; Applikations- und Systemarchitektur modellieren & bereinigen

© www.korn.ch 2012

Schritt 2 für Teams, die mehr in Richtung Agilität 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 als Detail-Koordinatoren jetzt Rahmensetzer und Unterstützer (**„dienende Führungskraft“**)
- .**Teams** erhalten möglichst viel **Eigenverantwortung** und Raum zur Selbstorganisation.

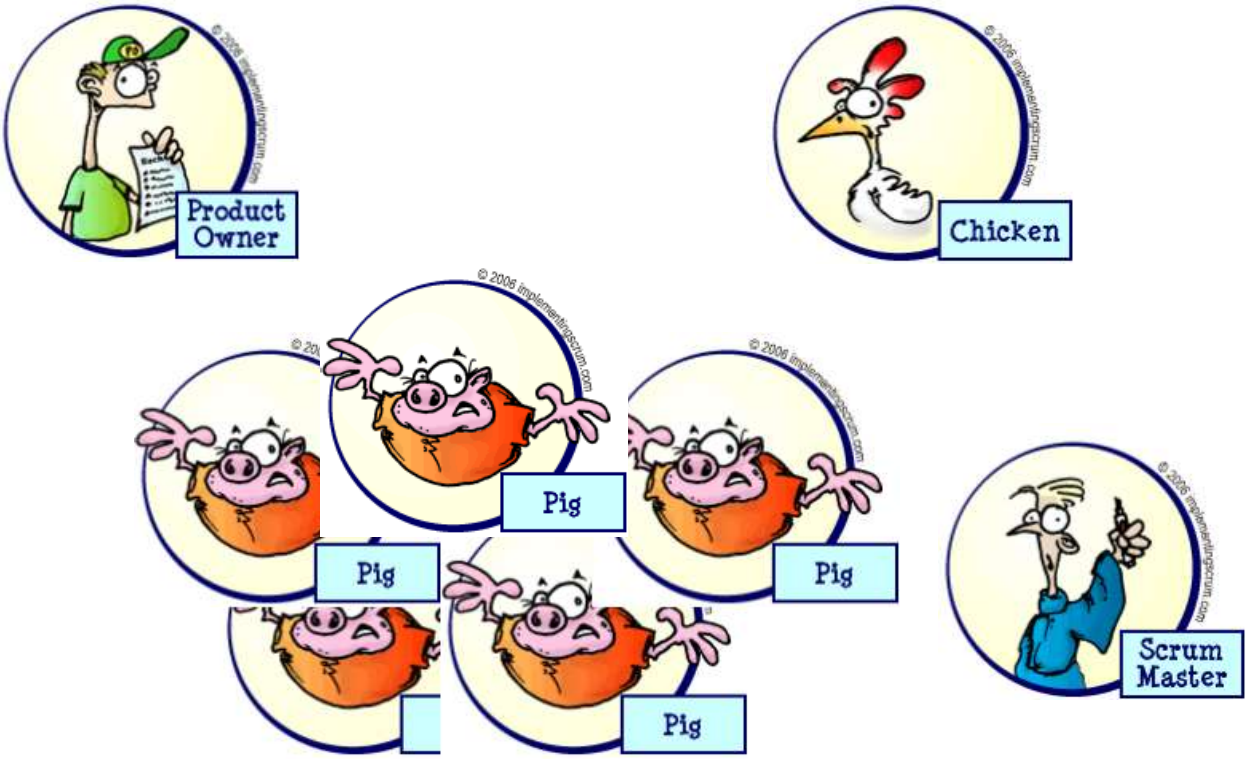
Schritt 3 für Teams, die voll auf Basis von Scrum arbeiten möchten:

- Bei längerfristig (> 18 Monaten) stabilen Produkt- oder Service-Teams:
  Teamleiter übernehmen auch die Rolle der Scrum Master
- Bei kurzlebigen Produkt- oder Service-Teams:
  Entwickler gehören zu "Entwicklerpools" (je rund 25 Personen) mit je einem personellen Vorgesetzten. Teams haben je einen Scrum Master, aber keinen Teamleiter

© www.korn.ch 2012

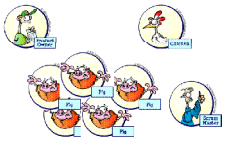
UPscaled

© www.korn.ch 2012



© www.korn.ch 2012

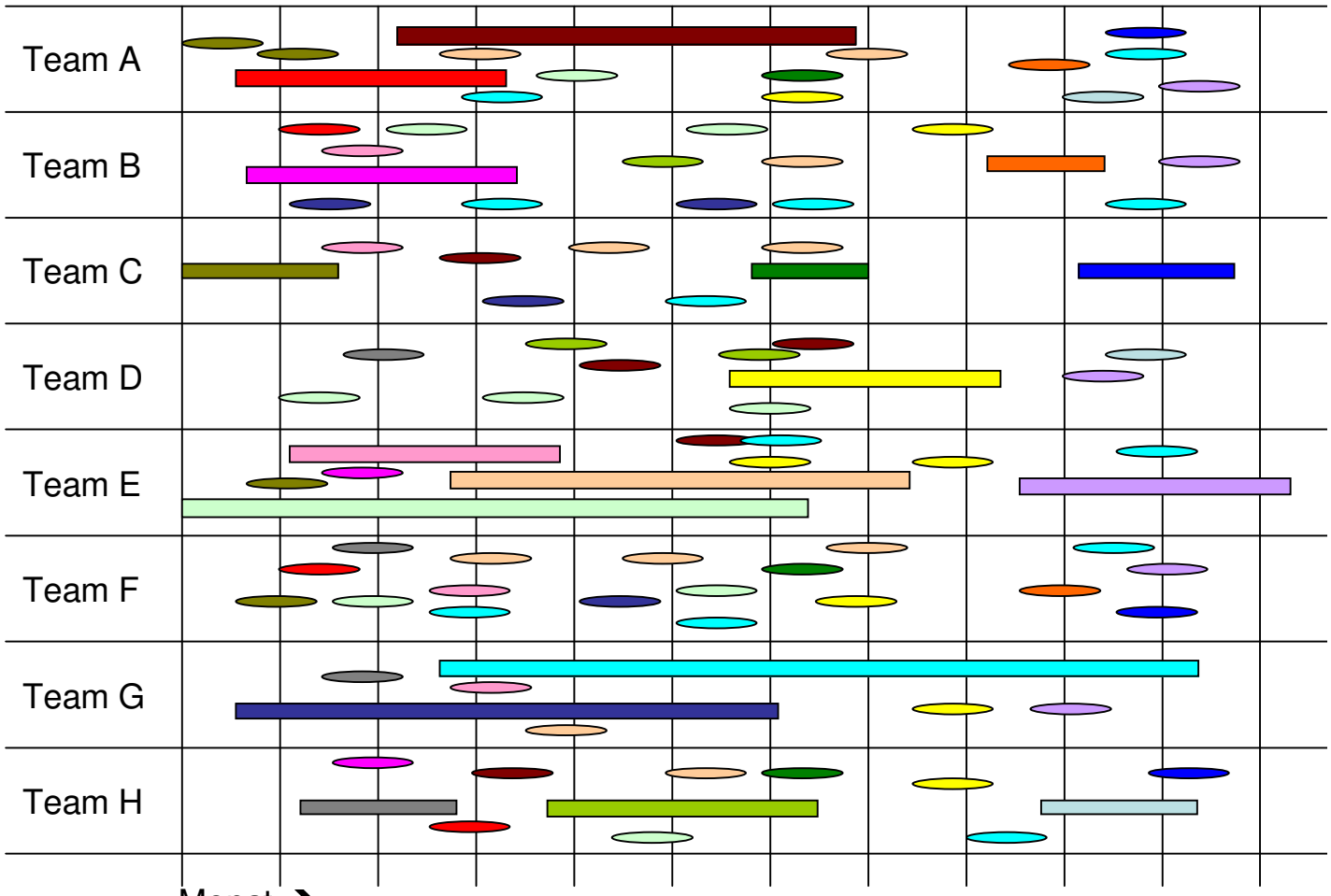
easy!



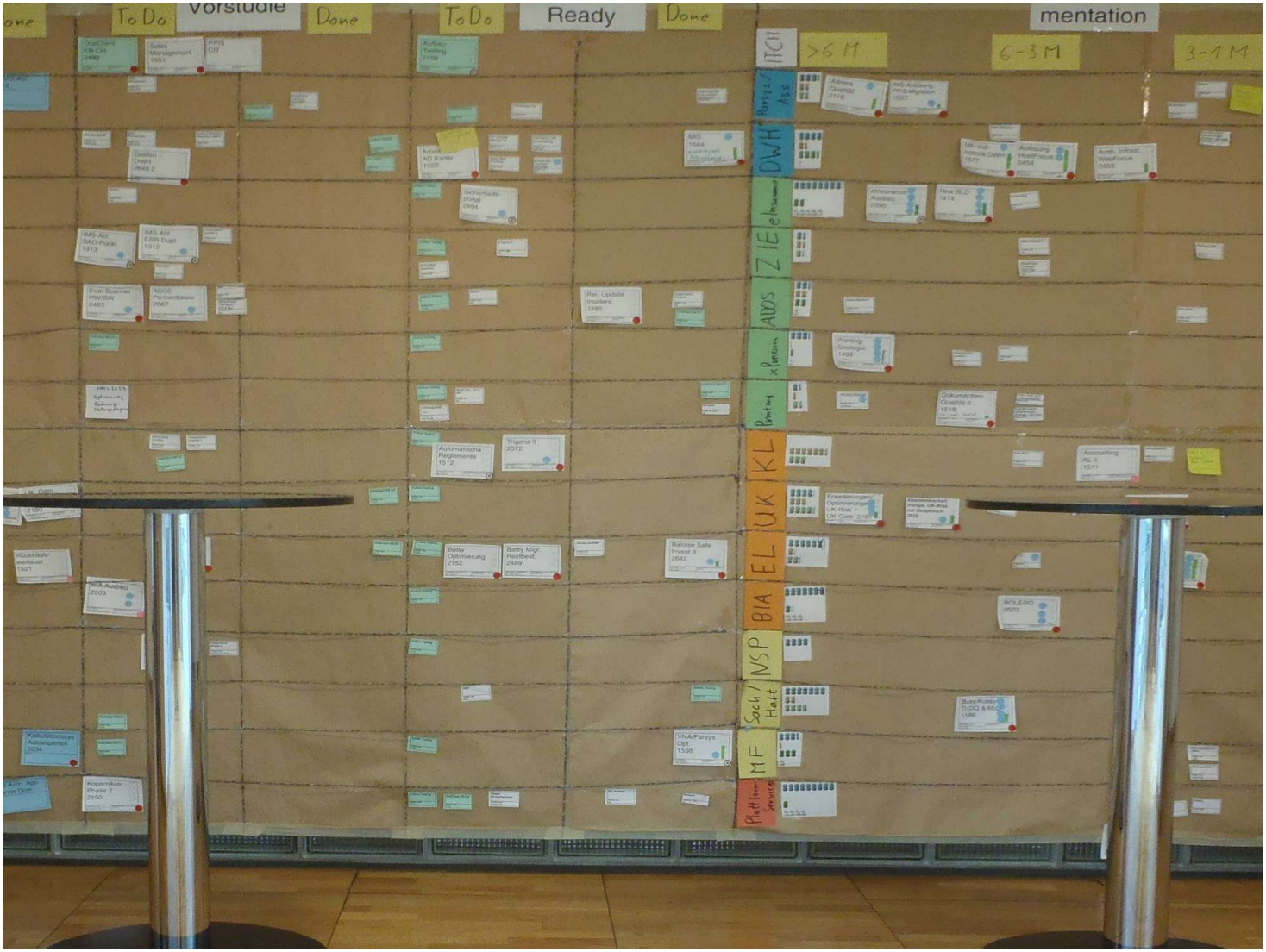
© www.korn.ch 2012

© www.korn.ch 2012

Multiprodukt / Multiteam - Management



© www.korn.ch 2012



Beispiel: Business Process Framework

(TM-Forum tmforum . org)

Customer & Market-Products:
End-to-End Business Processes

Strategy, Infrastructure & Product

Strategy & Commit	Infrastructure Lifecycle Management	Product Lifecycle Management
Marketing & Offer Management		
Service Development & Management		
Resource Development & Management (Application, Computing and Network)		
Supply Chain Development & Management		

Operations

Operations Support & Readiness	Fulfillment	Assurance	Billing & Revenue Management
Customer Relationship Management			
Service Management & Operations			
Resource Management & Operations (Application, Computing and Network)			
Supplier/Partner Relationship Management			

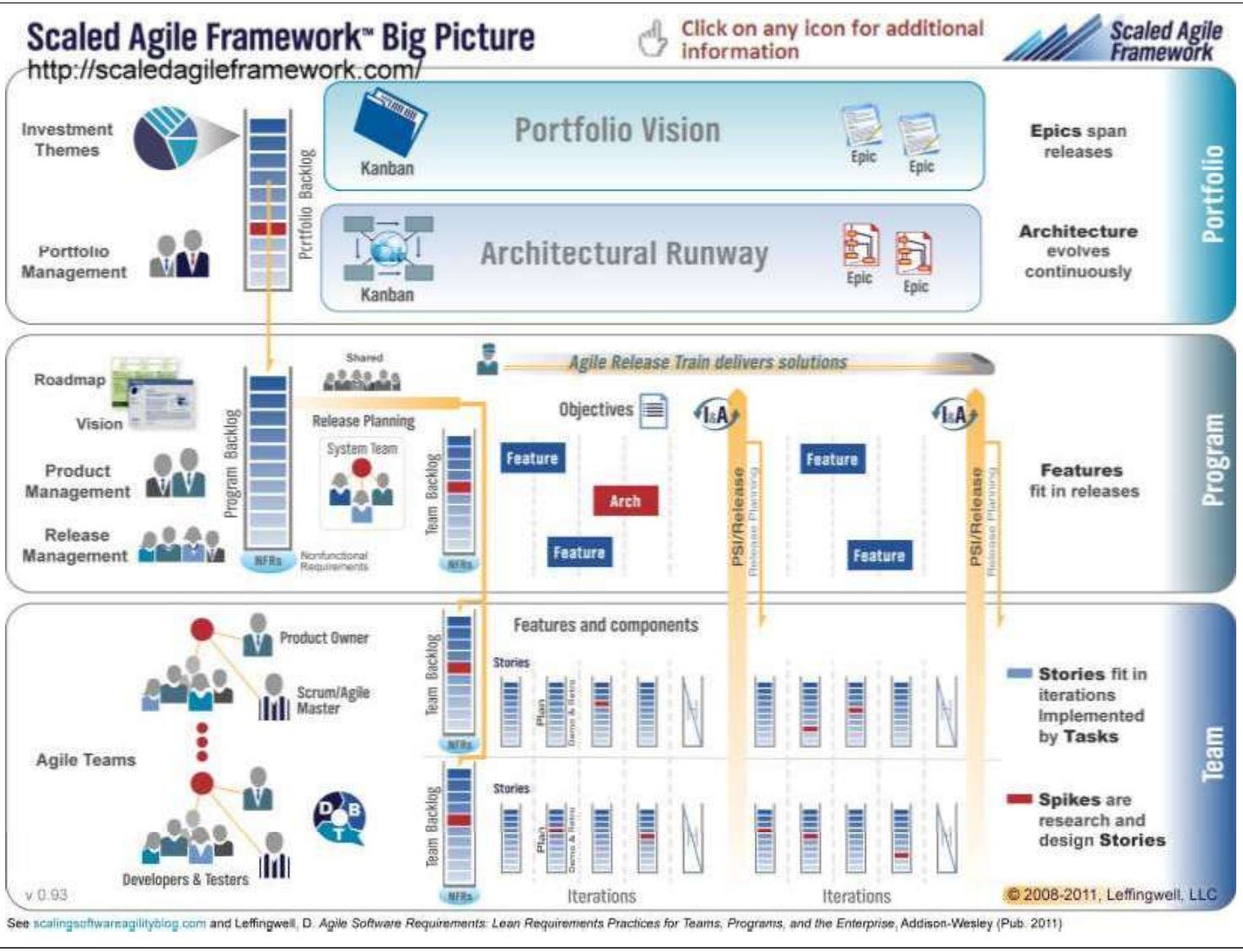
Functional Process Groups / IT-Systems

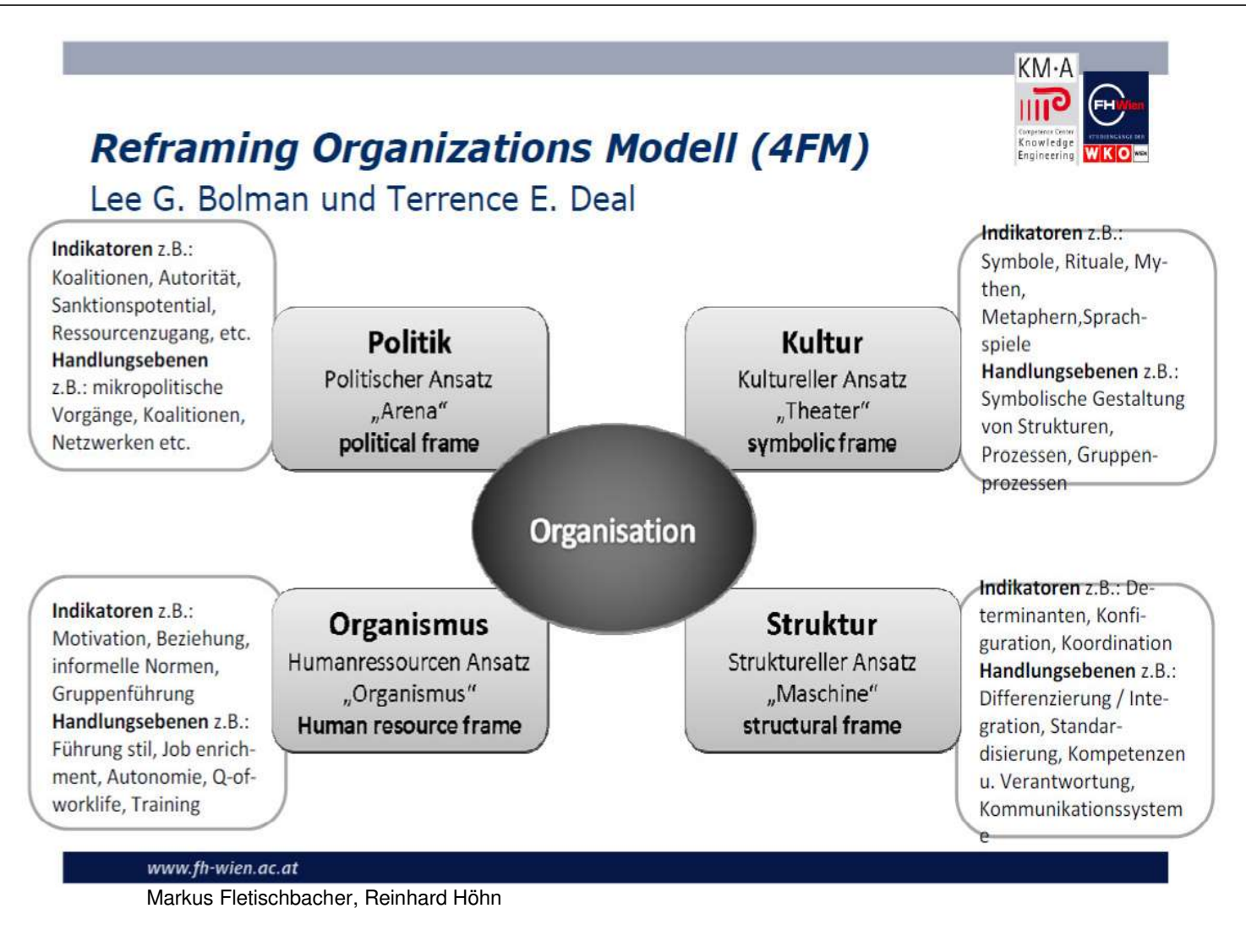
Functional Process Groups / IT-Systems

Functional Process Groups / IT-Systems

Enterprise Management

- Strategic & Enterprise Planning
- Enterprise Risk Management
- Enterprise Effectiveness Management
- Knowledge & Research Management
- Financial & Asset Management
- Stakeholder & External Relations Management
- Human Resources Management



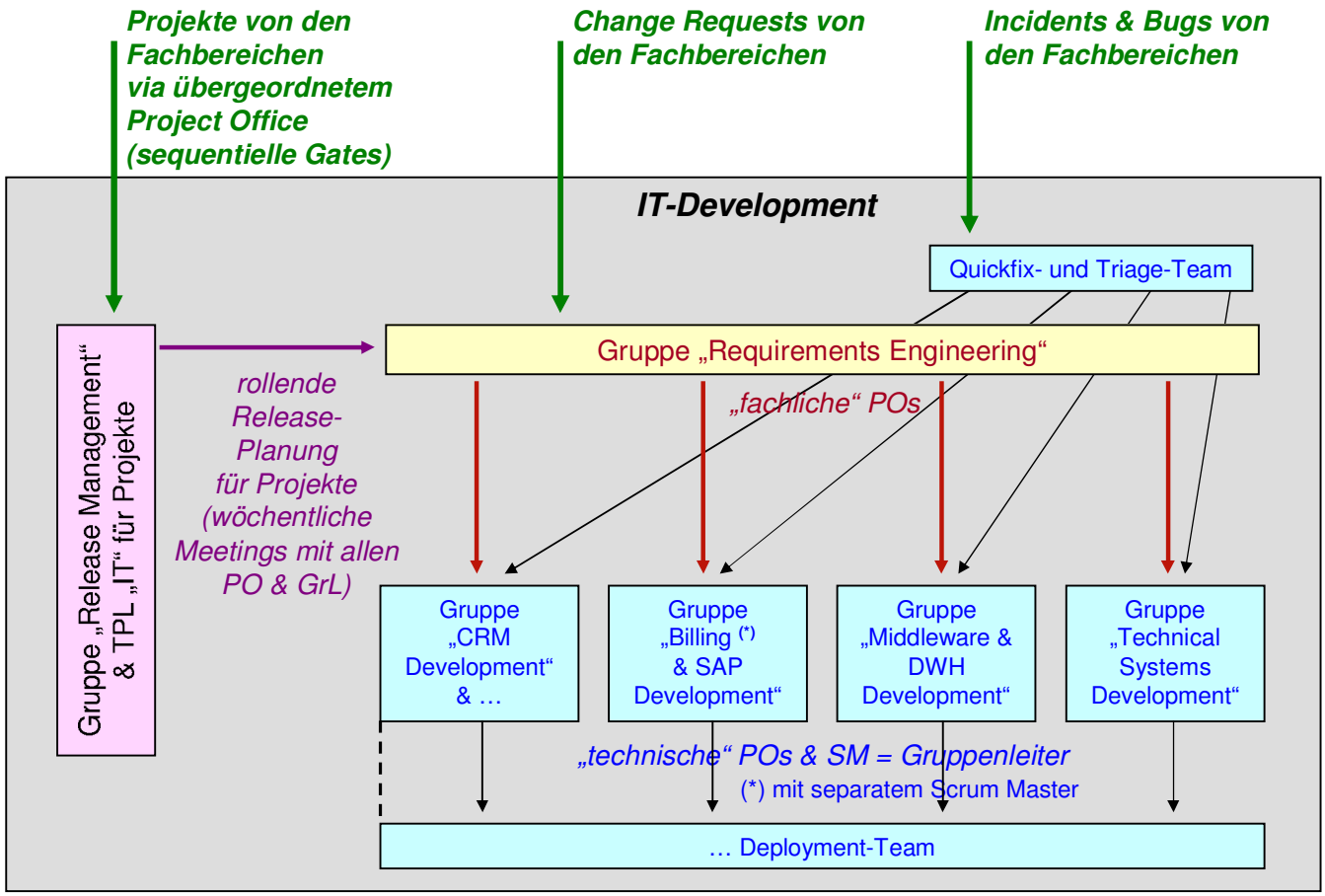


Markus Fletischbacher, Reinhard Höhn

Redesign der Agilität:

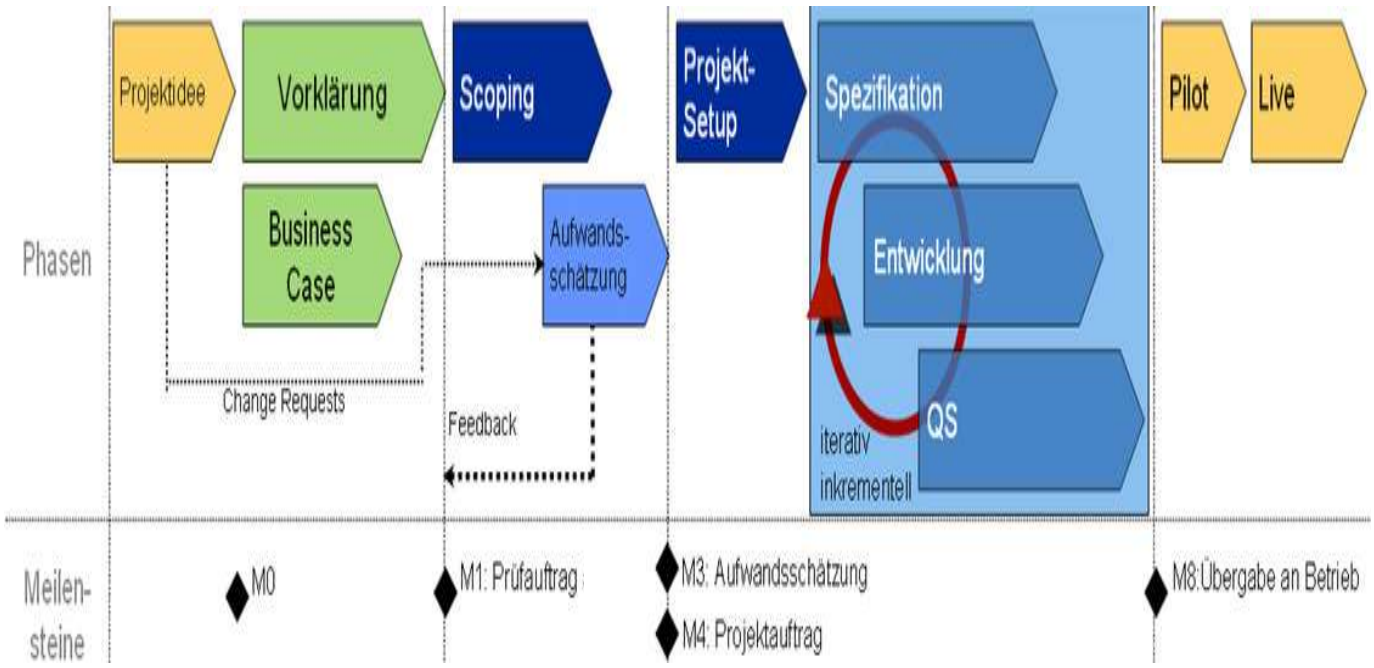
Ein realer Fall

© www.korn.ch 2012

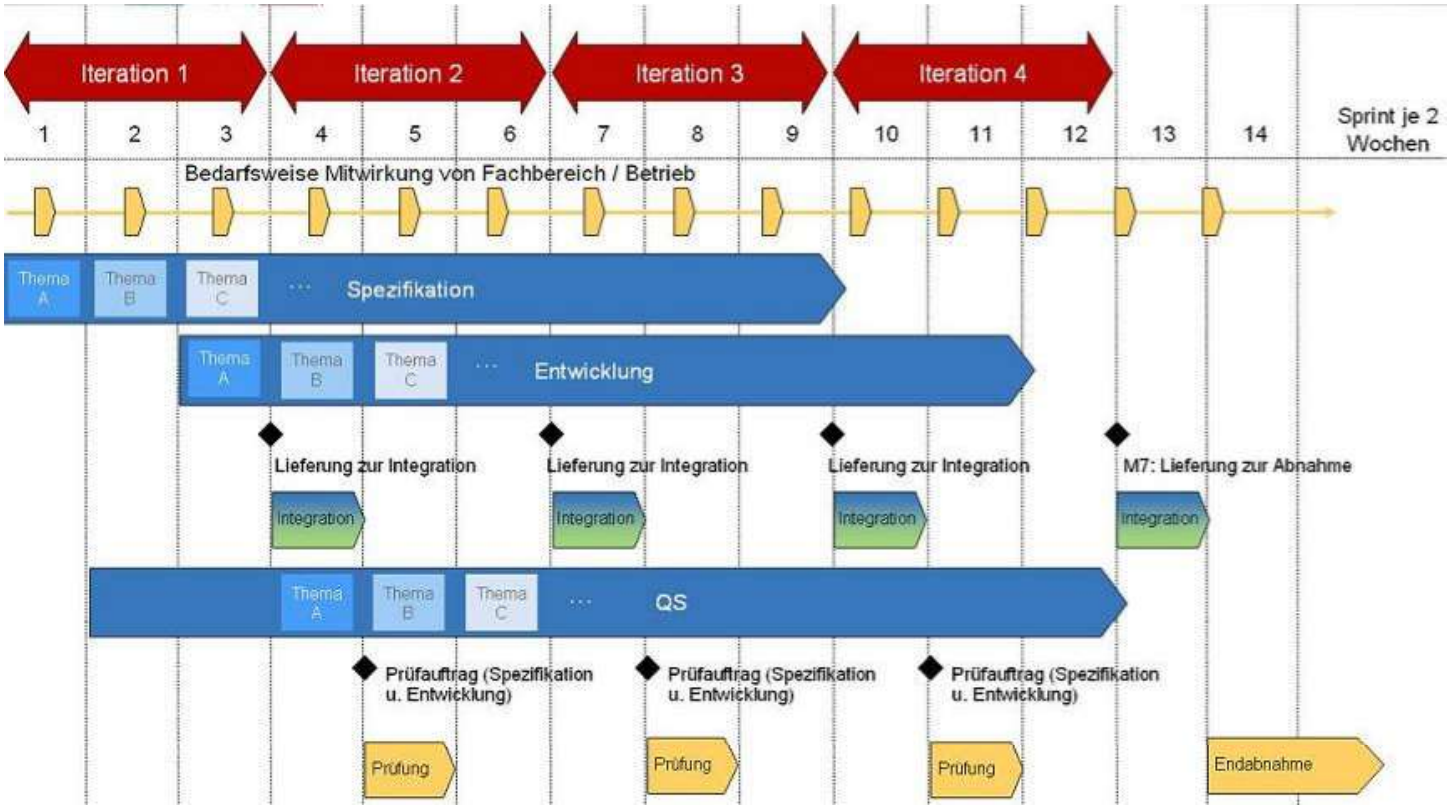


„Scrum“ vor rund 2,5 Jahren eingeführt

© www.korn.ch 2012



© www.korn.ch 2012



© www.korn.ch 2012

Probleme aus Sicht der Gruppenleiter:

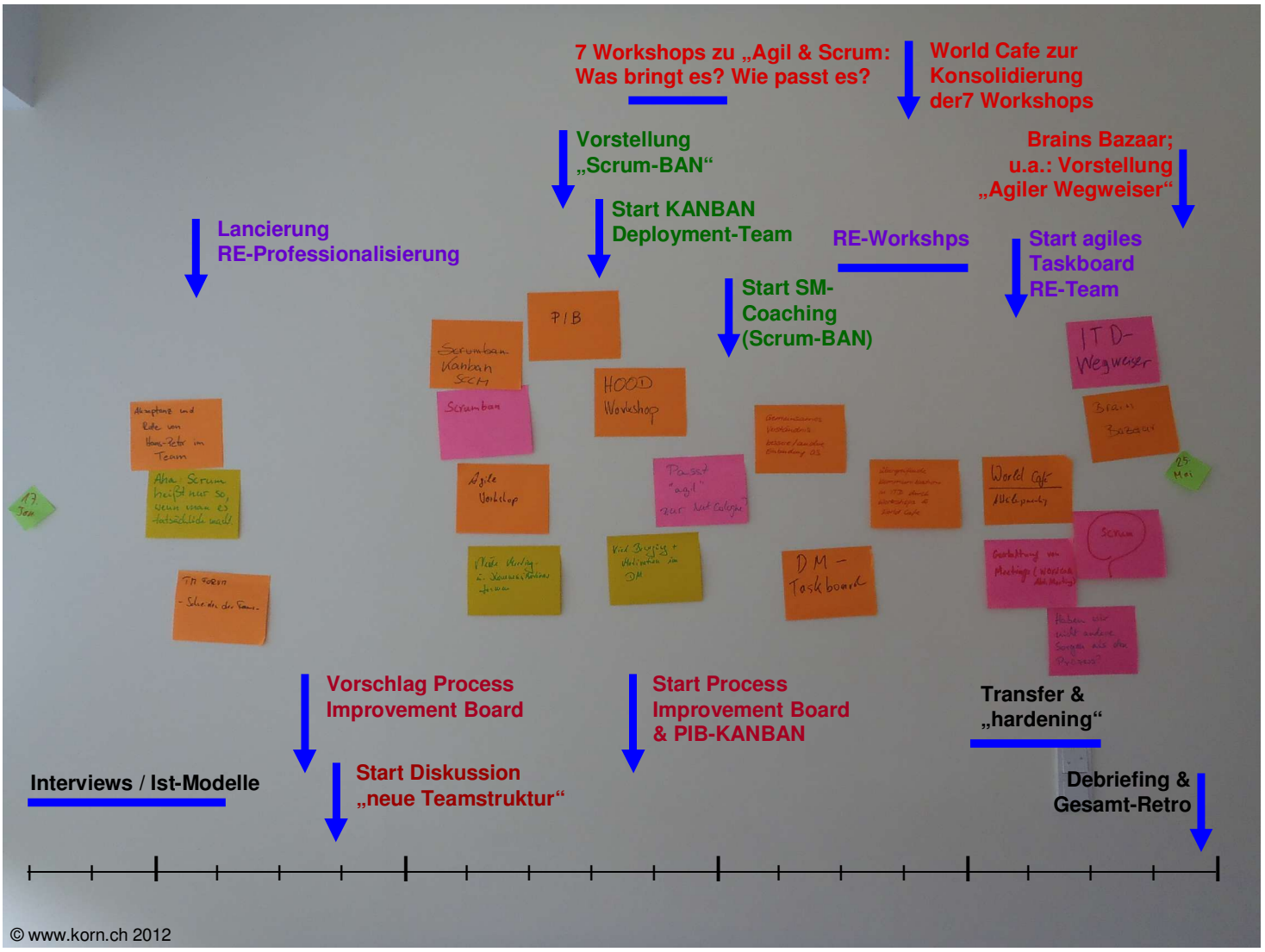
- Zu viele "Dinge" parallel
- Unbefriedigende SW-Qualität
- Hoher Wartungsanteil
- Know how (auch in den Fachbereichen) ungenügend
- Schlechte Qualität der 3rd-level-Tickets
- Zu enges fachliches / technisches Expertentum
- Zu grosse "Top-Projekte"
- Explodierender Umsetzungsaufwand
- Eskalationswege unbefriedigend
- Aufwand systematisch unterschätzt
- Unnötige Dokumentation
- Fehlende Dokumentation
- Beschreibung der fachlichen Anforderungen ungenügend zur Ableitung von Akzeptanzkriterien

© www.korn.ch 2012

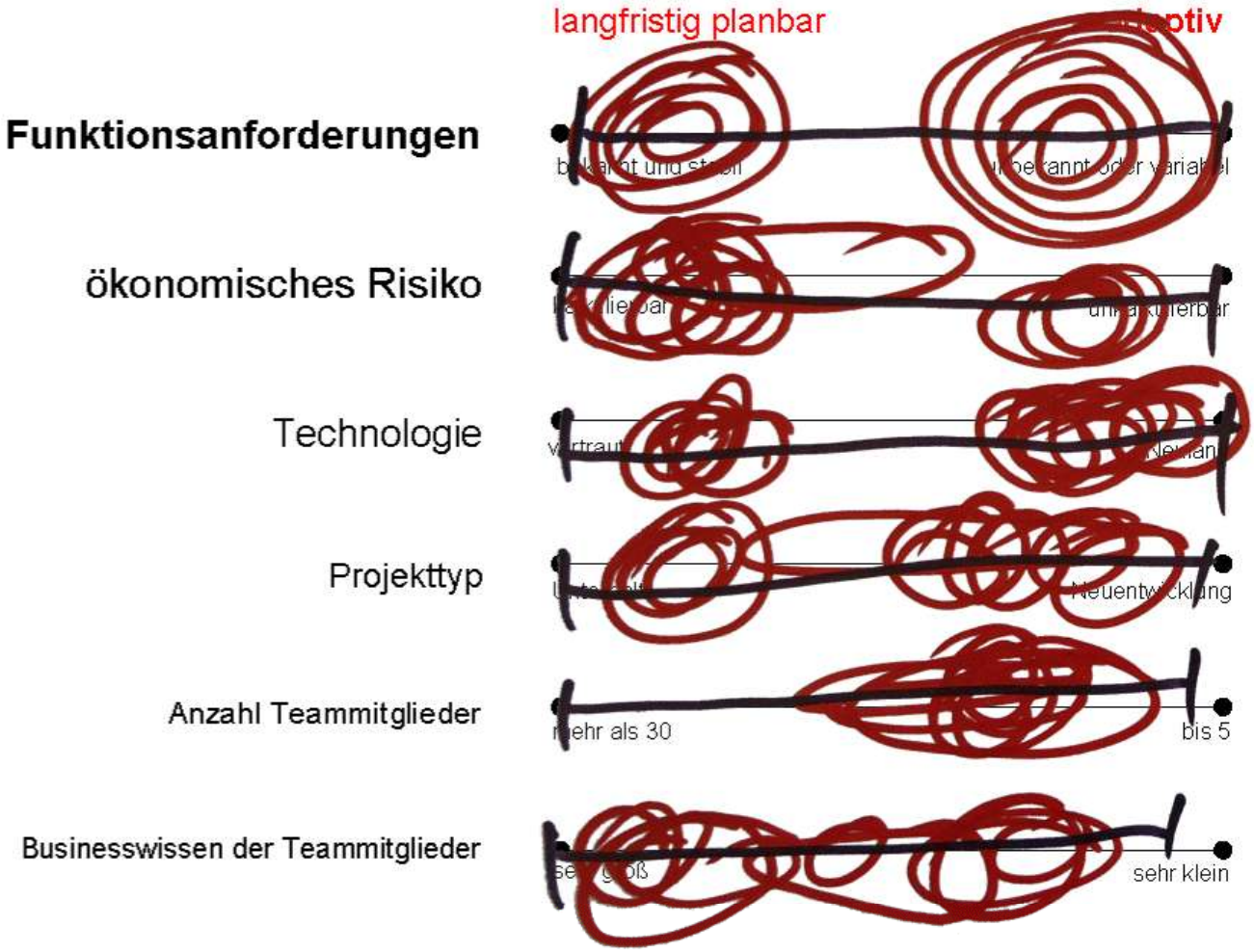
Sicht aller IT-Dev. Mitarbeitenden vor Redesign:

Was alles hat sich aus eurer Sicht in den letzten zwei Jahren verändert – und mit welcher Wirkung?							**Was davon ist bereits „gut genug“?**					
Scrum - aber nur in einzelnen Teams und autonom und unterschiedlich praktiziert, Verantwortlichkeiten unklarer	x	x	x	x	x	5						
weniger Doku, AWK nur optional, kein DVK, keine Modelle, AWK-Revies und andere bewährte Methoden	x	x	x	x	x	5						
mehr Kommunikation & Kooperation im Team	x	x	x	x		4	x	x	x	x	4	
Entwickluingsprozess mehrfach geändert -> wenig Nachhaltigkeit, Methode um der Methode willen. "Trial", ohne aus "errors" zu lernen	x	x	x	x		4						
mehr Austausch zwischen PO und Team, Team identifiziert sich besser mit Aufträgen des PO, Interaktion Anford.Manag. & Team besser	x	x	x			3	x	x			2	
Häufiger Leiterwechsel, Organisationsänderung, mehr Umzüge	x	x	x			3						
Mehr Transparenz (z.B. der Arbeitslast) & Planungssicherheit (... leider aber ohne Konsequenzen)	x	x				2	x				1	
besser motivierte Teams, Teams wurden zu "echten" Teams	x	x				2	x				1	
von Wasserfall (AWK>DVK) zu Planung, Lieferung, Review pro kurzen "stabilen" Sprints / Timeboxes / Inkrementen (AWK>Umsetzung)	x	x				2						
Sehr (Zu) viel Administration und Planungsaufwand & Planungsänderung wegen kurzer Sprints	x	x				2						
mehr externe DL; diese bestimmen zu sehr unser Vorgehen	x	x				2						

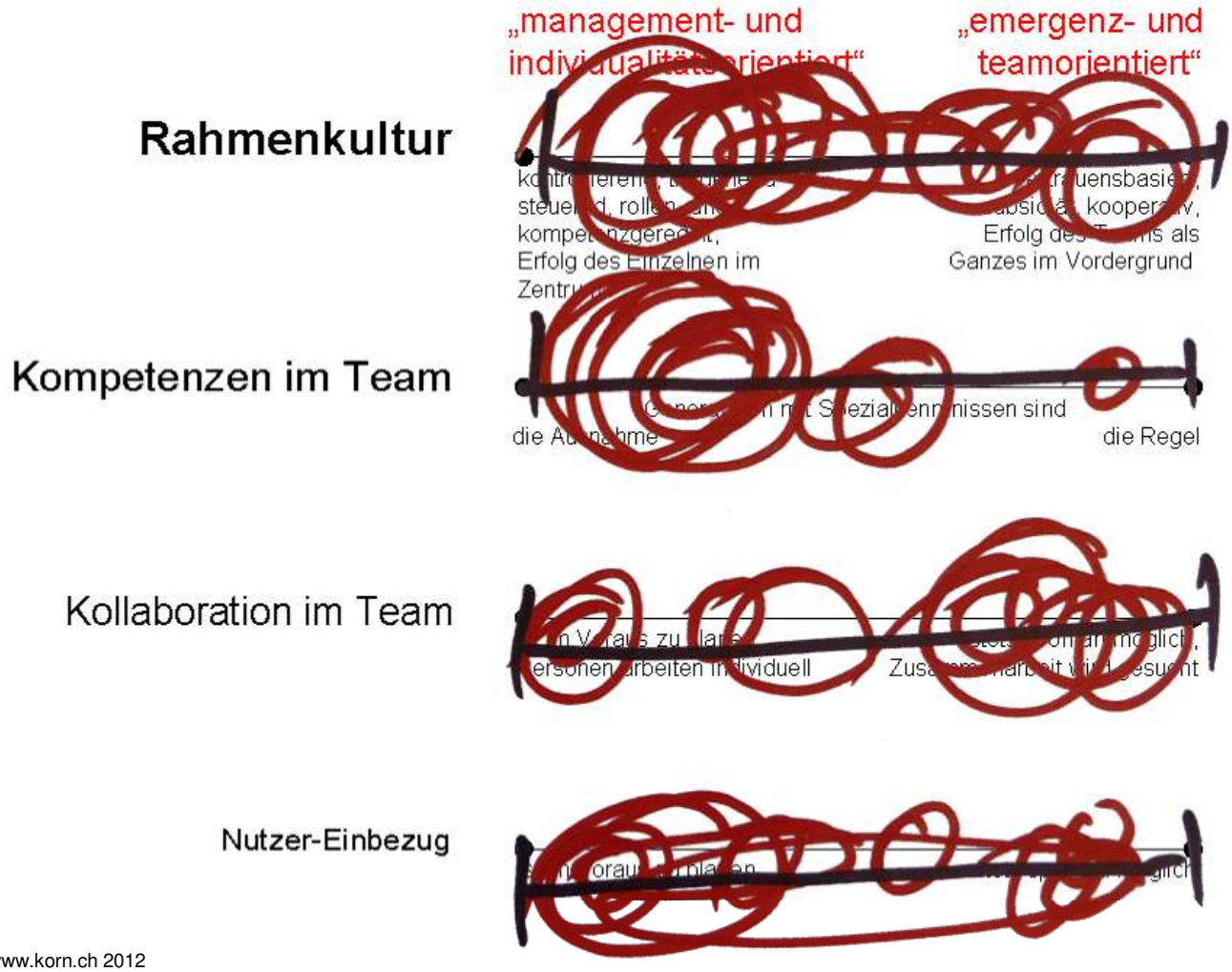
© www.korn.ch 2012



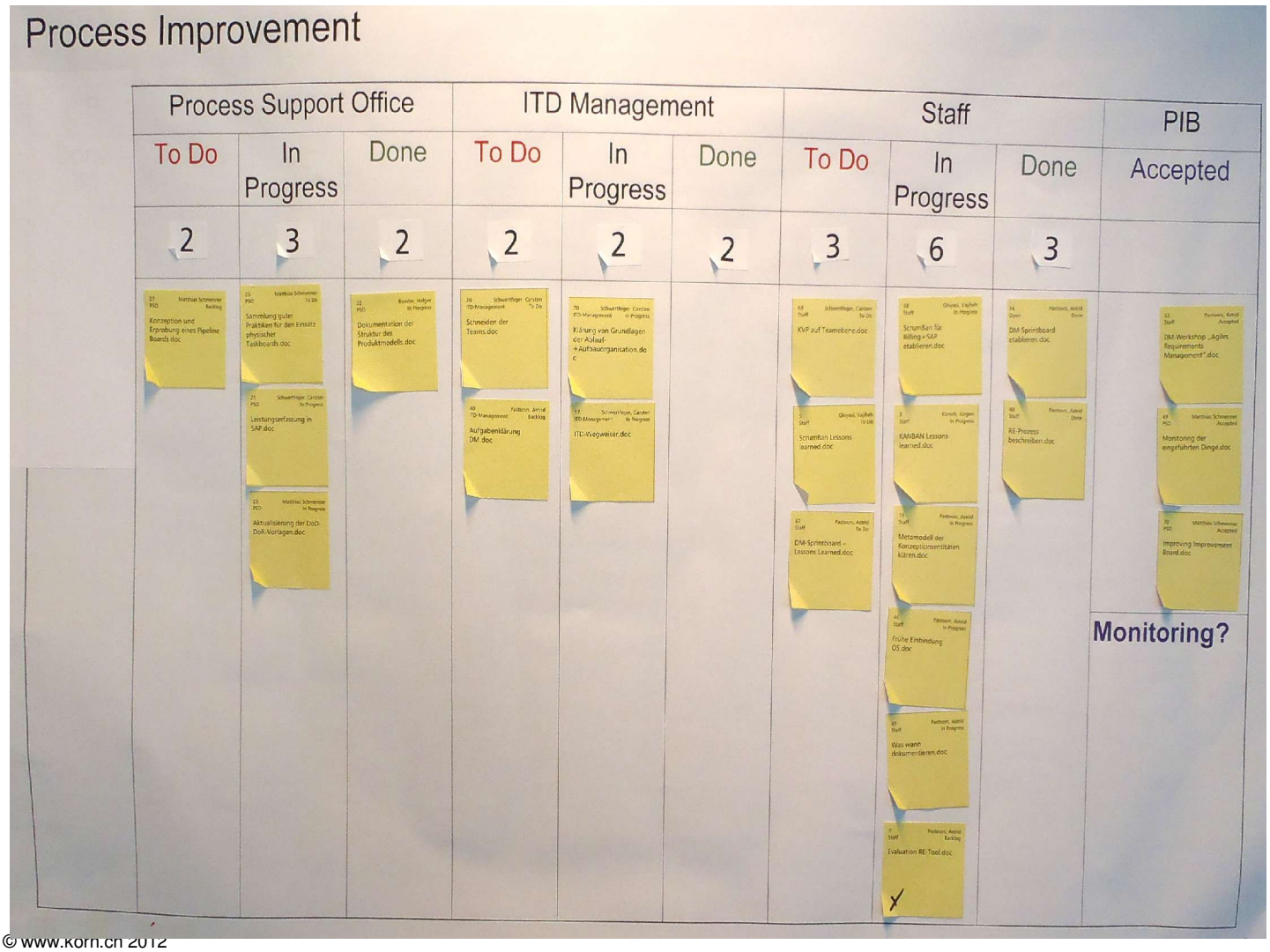
© www.korn.ch 2012



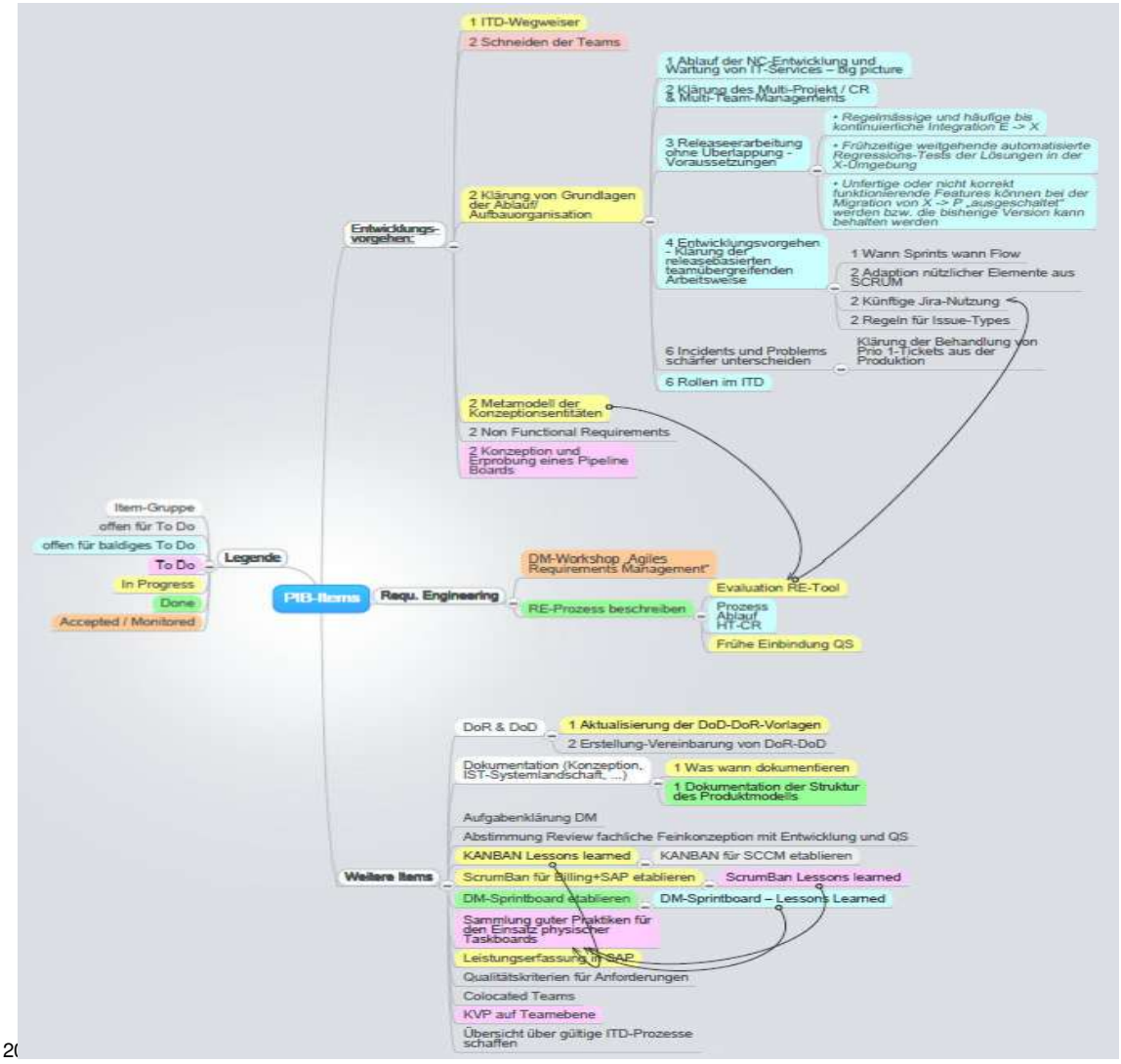
© www.korn.ch 2012



© www.korn.ch 2012



© www.korn.ch 2012



© www.korn.ch 2(

Unser Wegweiser:

Wir (IT Dev) entwickeln Lösungen, damit die Kunden unseres Unternehmens den bestmöglichen Service erhalten und unser Unternehmen den Mitbewerbern vorziehen. Mit diesem Kundenfokus engagieren wir uns dafür, dass wir den Fachbereichen die zur Serviceerbringung optimalen Lösungen unter Berücksichtigung der Kosten-Nutzen-Betrachtung zur Verfügung stellen.
Dazu nutzen wir situationsspezifisch angepasste Vorgehensweisen auf Basis dieser Grundlagen:

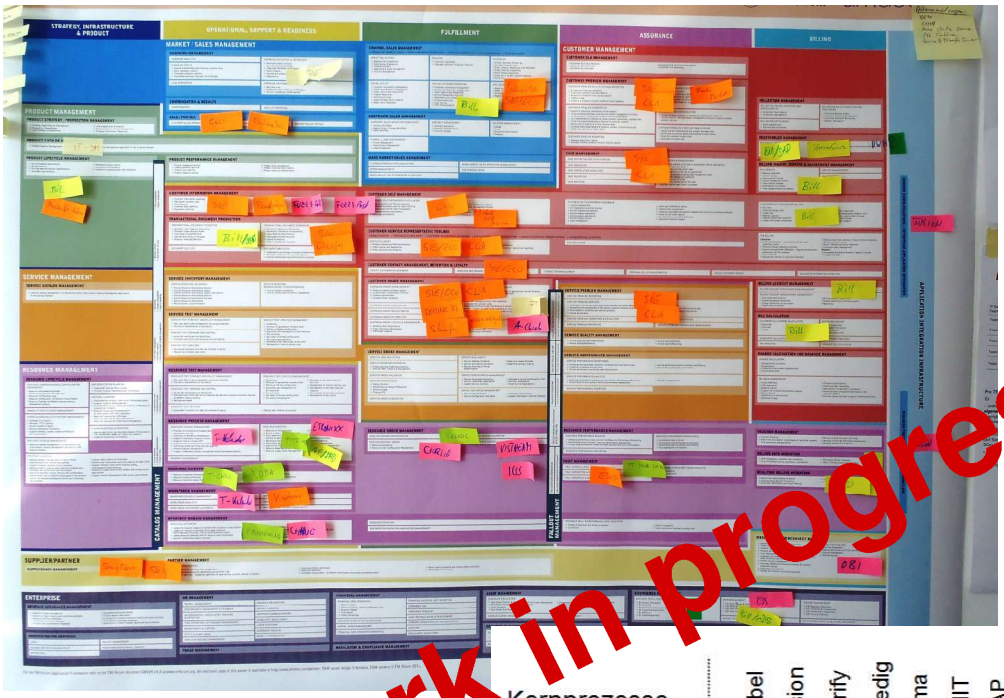
Die Kooperation und persönliche, spontane und informelle Kommunikation professionell arbeitender Menschen steht im Vordergrund. Dokumentationen, Tools und Prozesse sind Hilfsmittel dazu, nicht Ersatz. Diese Hilfsmittel nutzen wir in stets situationsgerecht angepasster Form.

Die bedarfsgerechte Lieferung fehlerfrei funktionierender Software steht im Vordergrund. Die dazu nötige vorangehende Konzeption und Planung beschränken wir auf jenen Umfang, der für die persönliche und spontane Kommunikation und Kooperation der Beteiligten nötig ist.

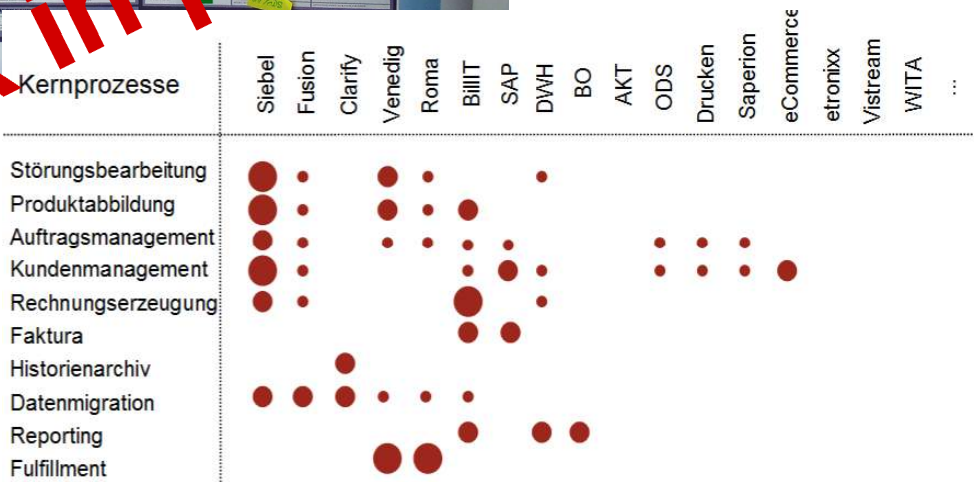
Die laufende persönliche Kooperation mit den Produktmanagern in den Fachbereichen als Treuhänder der Kundeninteressen steht im Vordergrund. Formelle Verträge und die Freigabe von Konzeptions- und Planungsdokumenten in situationsgerecht angepasster Form sind Hilfsmittel, kein Ersatz.

Wir sind Profis im konstruktiven Umgang mit Planabweichungen. Wir bevorzugen eine rollende Planung und planen Details erst dann, wenn sie sinnvoll planbar sind. **Unser Vorgehen unterstützt und fördert das fortlaufende Erkennen und koordinierte Umsetzen von Verbesserungen im Sinne des Kundennutzen oder der Kostenoptimierung.**

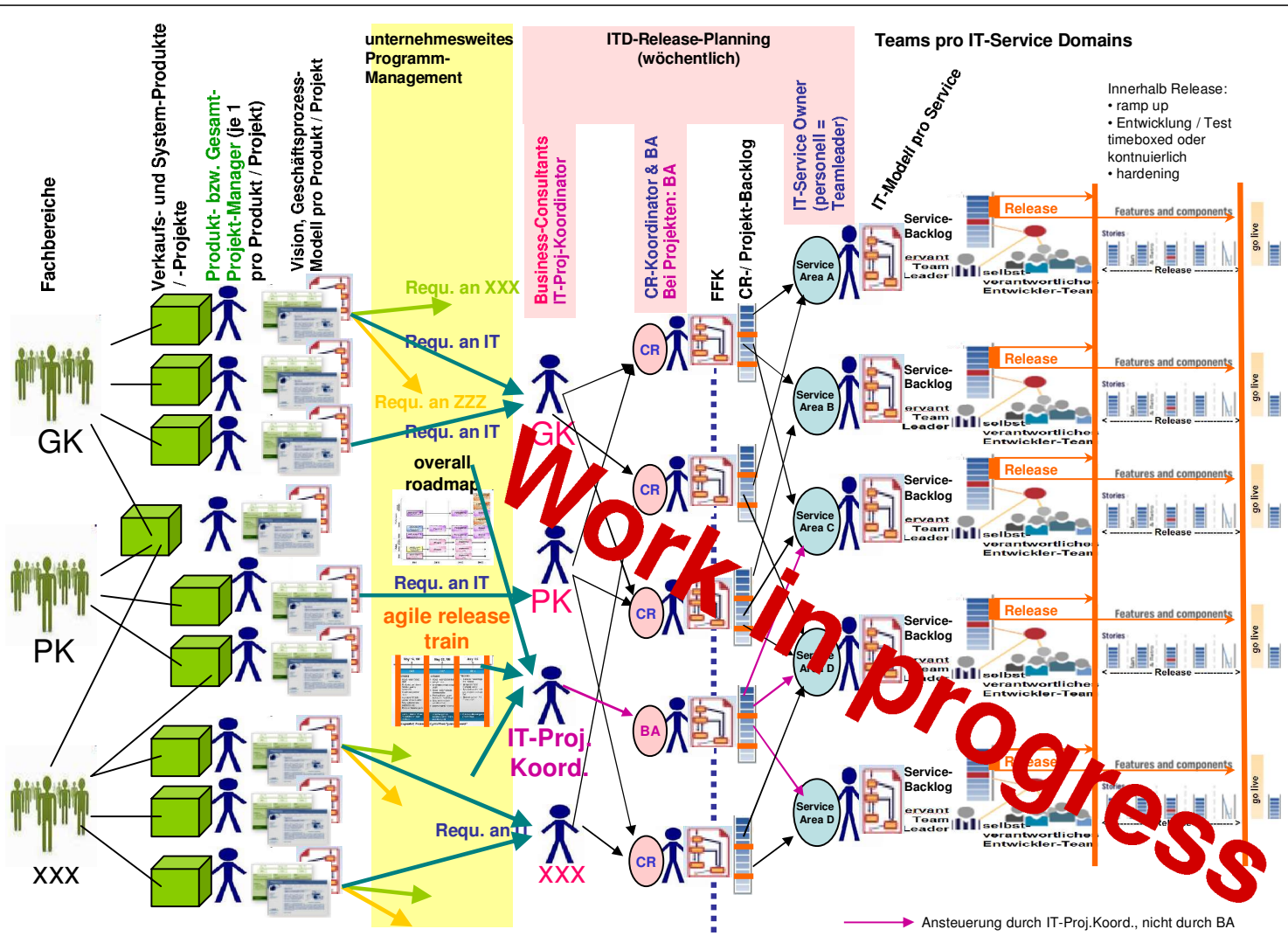
© www.korn.ch 2012



Work in progress

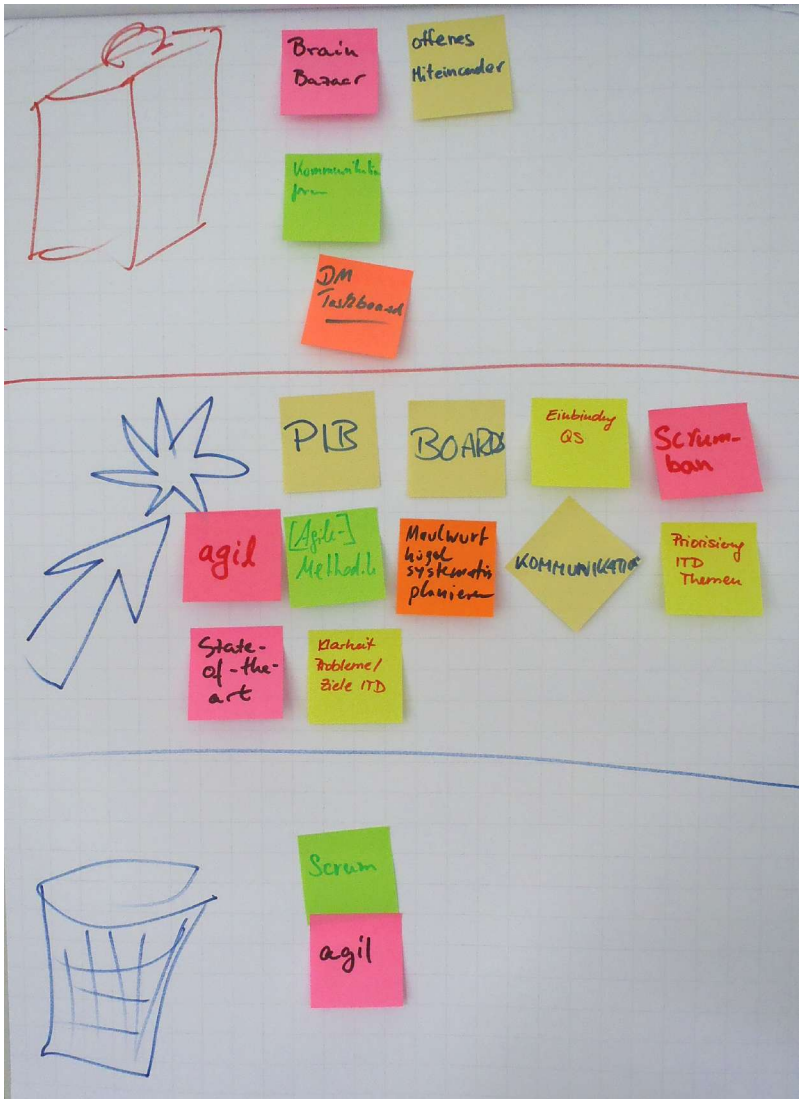


© www.korn.ch 2012

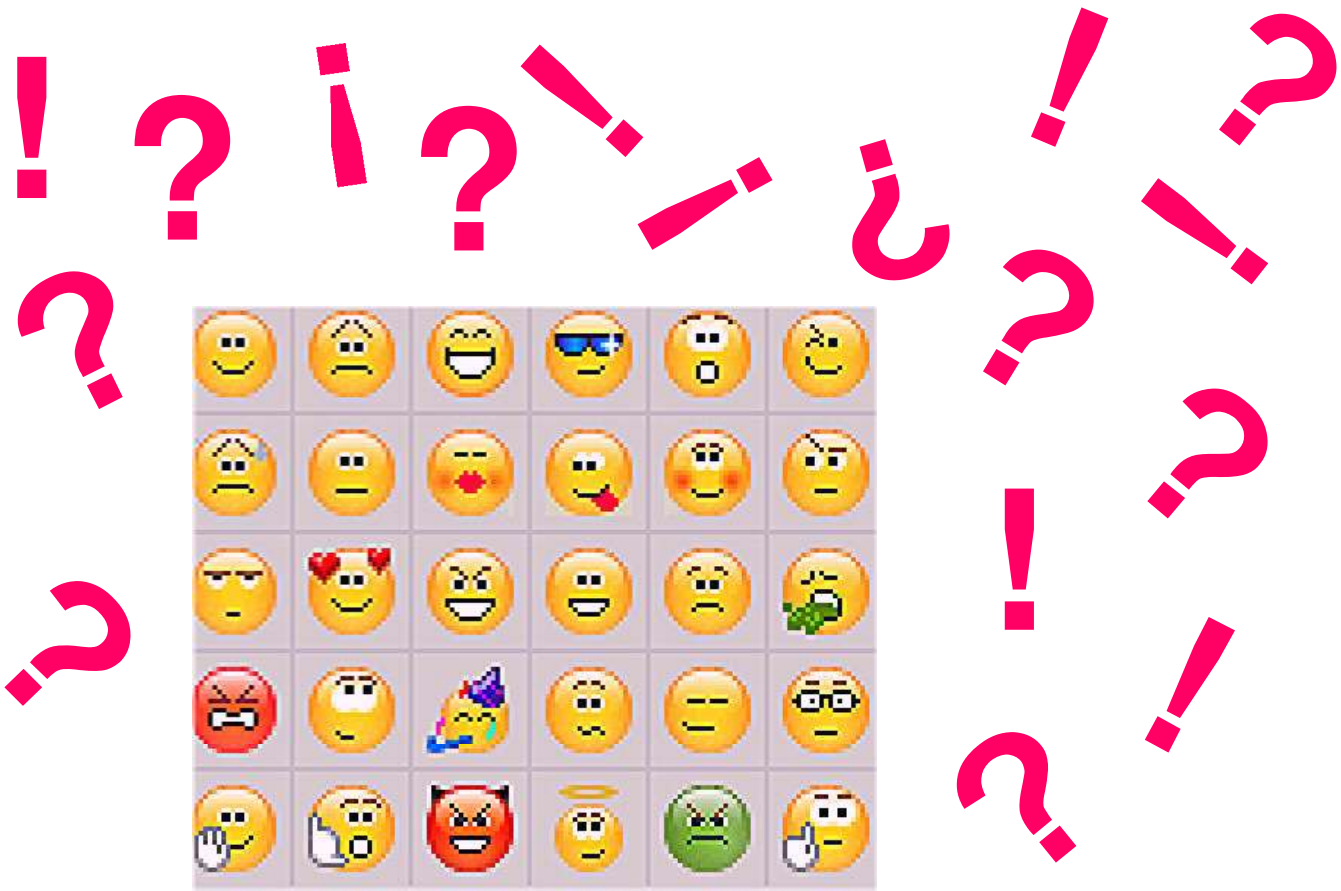


Work in progress

© www.korn.ch 2012



© www.korn.ch 2012



© www.korn.ch 2012