

Labor

Agile Software Entwicklung

Organisatorisches

Stephan Kiesling, Jil Klünder

{stephan.kiesling, jil.kluender}@inf.uni-hannover.de

26.11.2015

Organisatorisches

- Klausuren oder Termine während der Blockwoche
 - Bitte Abwesenheiten in die Liste eintragen
- Nächste Startzeiten der Veranstaltung
 - 03.12.: 09:00 Uhr
 - 10.12.: 09:00 Uhr
 - Werden auf der Webseite ergänzt:
http://www.se.uni-hannover.de/pages/de:lehre_2015winter_xplab

Einführung in die Agilen Methoden

- Teil 2 -

Stephan Kiesling, Jil Klünder

{stephan.kiesling, jil.kluender}@inf.uni-hannover.de

26.11.2015

Letztes Mal...

- ... haben wir die Werte und Praktiken der agilen Entwicklung kennen gelernt.
- ... haben wir das eXtreme Programming als agile Methode eingeführt.

Heute...

- wollen wir weitere agile Methoden kennen lernen.
- lernen wir, wie wir diese einführen und beibehalten können.

SCRUM

26.11.2015

Andreas Wintersteiger: „SCRUM ist ein Framework und kein Prozess“

(Aus dem Artikel „Das Wesen von SCRUM“ im eclipse Magazin 4.12)

Übergeordnete Werte:

Transparenz	Inspektion	Adaption
-------------	------------	----------

Konkrete Elemente:

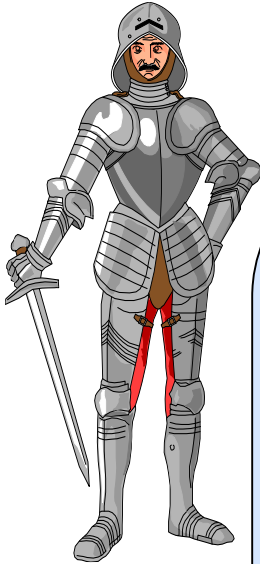
Rollen	Meetings	Artefakte
--------	----------	-----------

Auf einen Blick

Andere



SCRUM Master



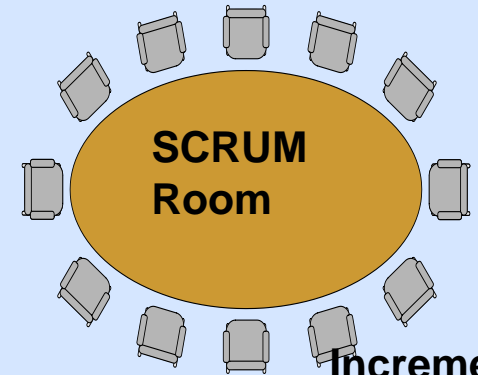
Product Owner



Product Backlog (priorisiert)

SPRINT Backlog

SCRUM team 7+/-2



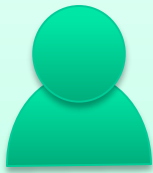
Increment

SPRINT: 30 days

Daily SCRUM 15 min.

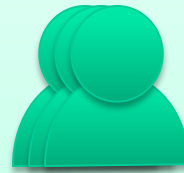


Scrum Master



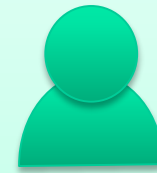
- Effektive Kommunikation
- Einhaltung agiler Werte
- Verantwortung der Gruppe stärken
- Verbesserung des Prozesses

Umsetzungsteam



- Technische Umsetzung
- Aufwandsschätzung
- Technische Risikobewertung
- Verbesserung des Prozesses

Product Owner



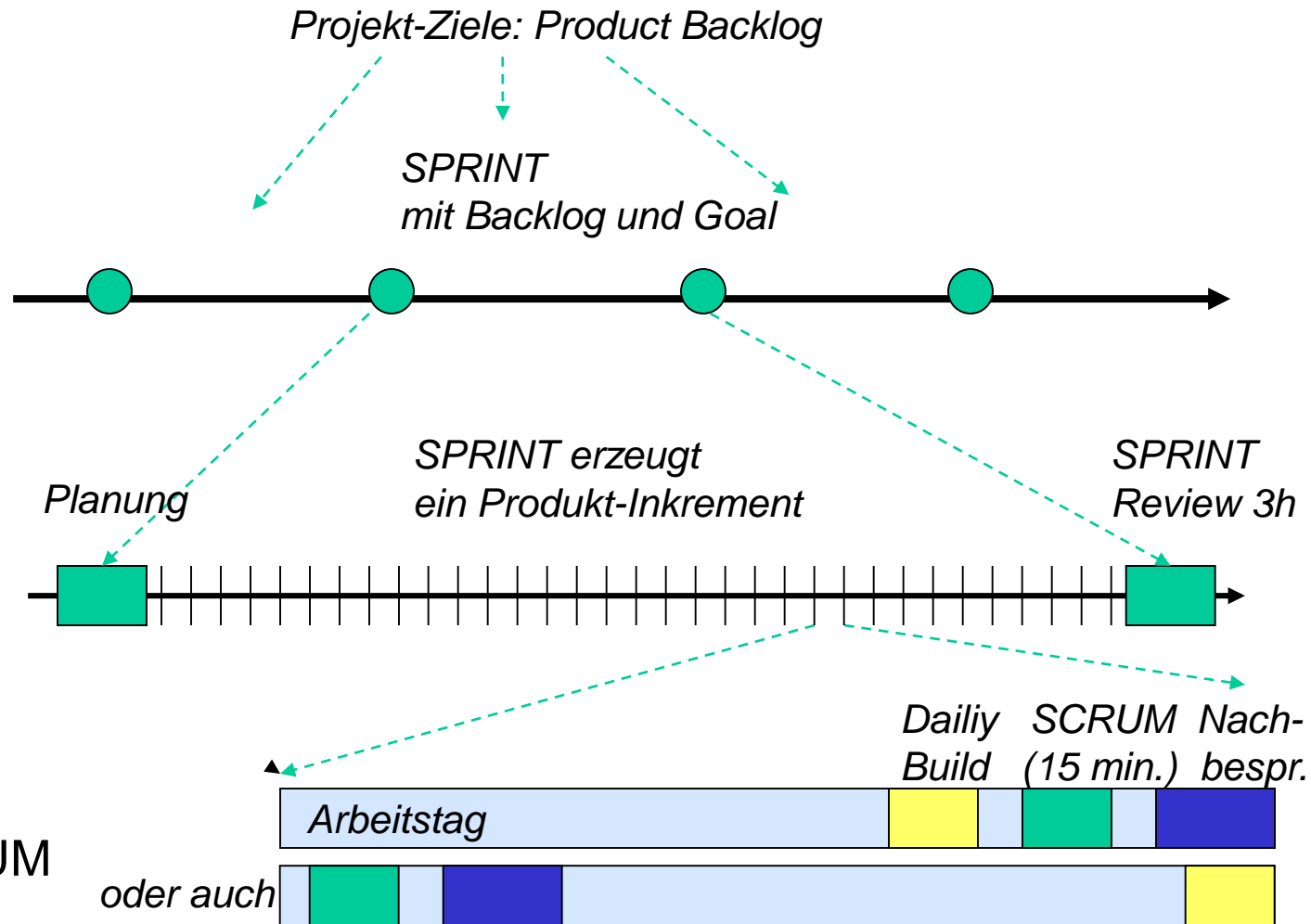
- Kontakt mit Kunden
- Am besten der Kunde selbst
- Fachliche Verantwortung
- Gesamterfolg des Vorhabens

Detailierungsebenen in SCRUM

Projekt aus
SPRINTS

SPRINT im
30 SCRUM-Takt

Ein Arbeitstag:
SCRUM zu SCRUM



Sprint Planning

Jede Iteration

Detaillierte Planung der nächsten
Story Cards

Daily Scrum

Kurze Statusmitteilungen

Sprint Review

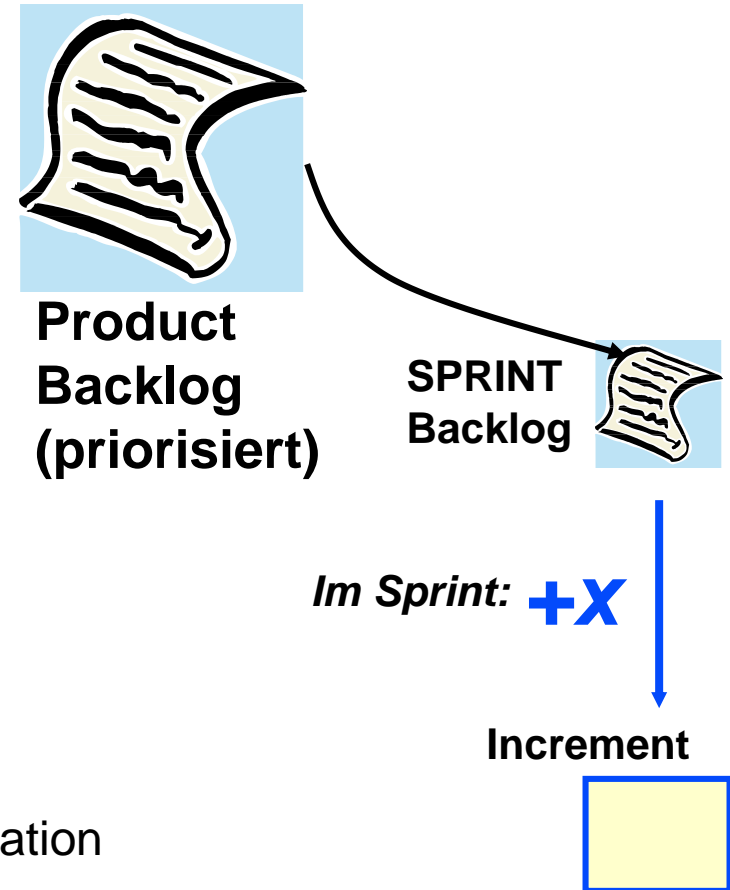
Feedback vom Kunden zum
Produktinkrement

Sprint Retrospektive

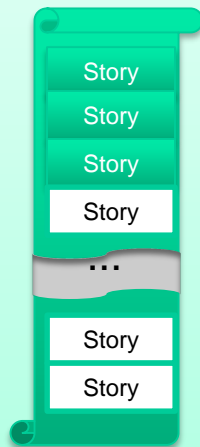
Reflexion über vergangene Iteration
Erarbeitung von Action Items für
nächste Iteration

Dokumentation in SCRUM

- Wichtige Dokumente:
 - Product Backlog
 - Sprint Backlog
 - Increment (Resultat, SW)
- Allenfalls temporäre Dokumente:
 - Aufwandsschätzungen
 - Auswertungen/Kurven
- Nicht vergessen:
 - SCRUM ist nur die Mgmt.-Hülle
 - Darin können viele weitere Entwicklungsdokumente entstehen (+x)
- Fazit
 - SCRUM fordert (praktisch) keine Dokumentation



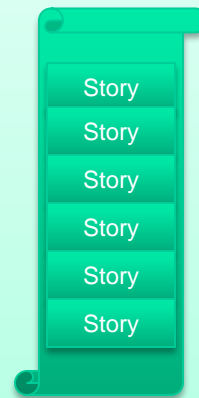
Product Backlog



Alle User Stories, die im Projekt umgesetzt werden sollen

Späte Stories weniger detailliert

Sprint Backlog



Alle *User Stories*, die in dieser Iteration umgesetzt werden

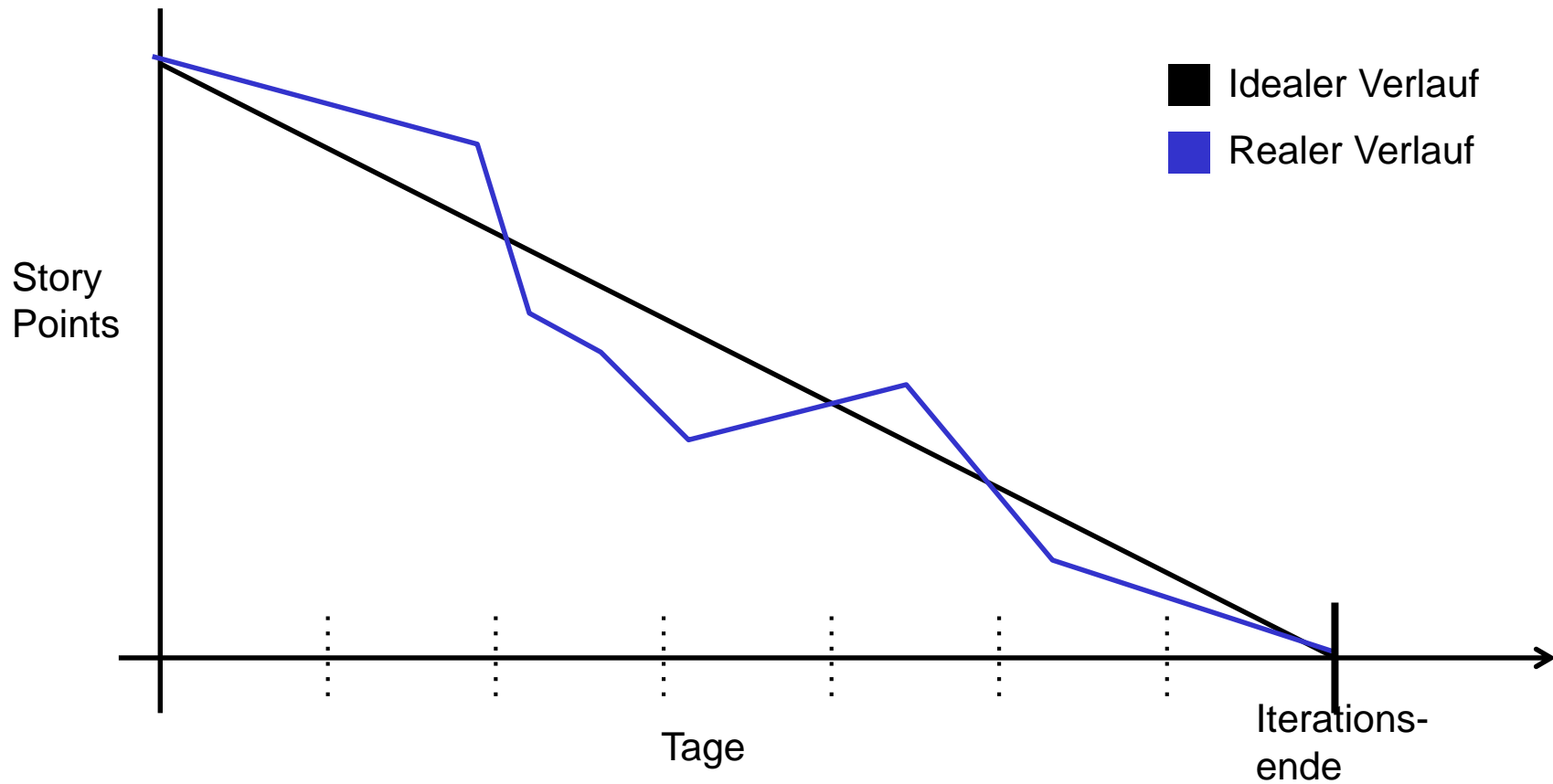
Produktinkrement



Lauffähige, fehlerfreie, qualitativ hochwertige Software

Jederzeit vorhanden

Fortschrittsmessung durch Burndown Charts



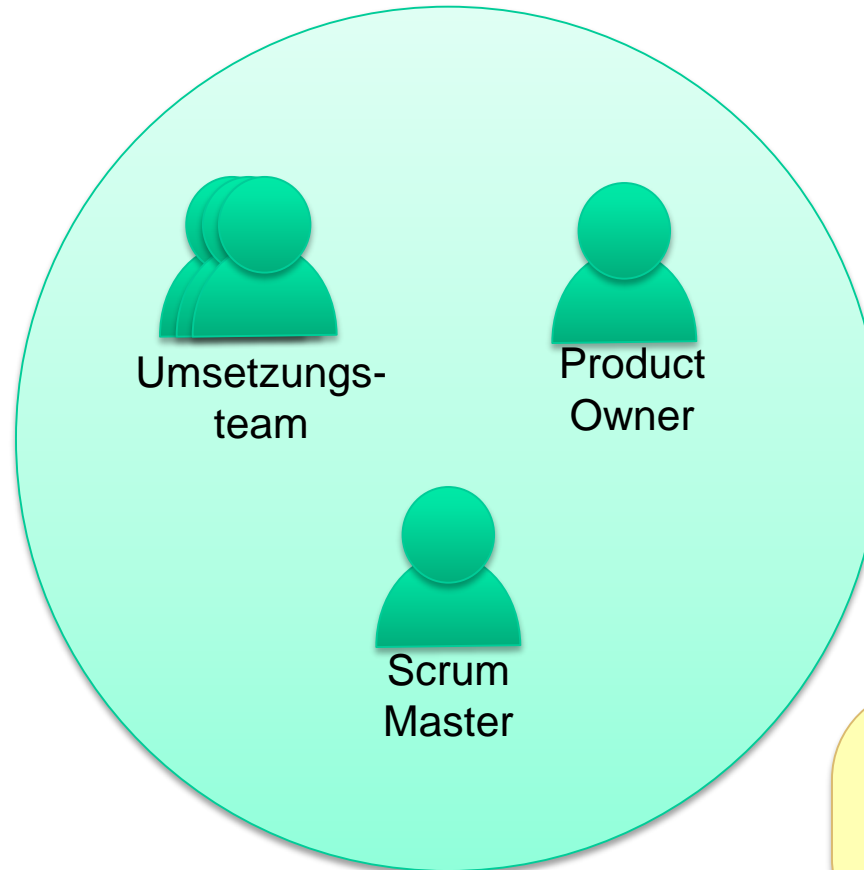
Agile Methoden einführen

26.11.2015

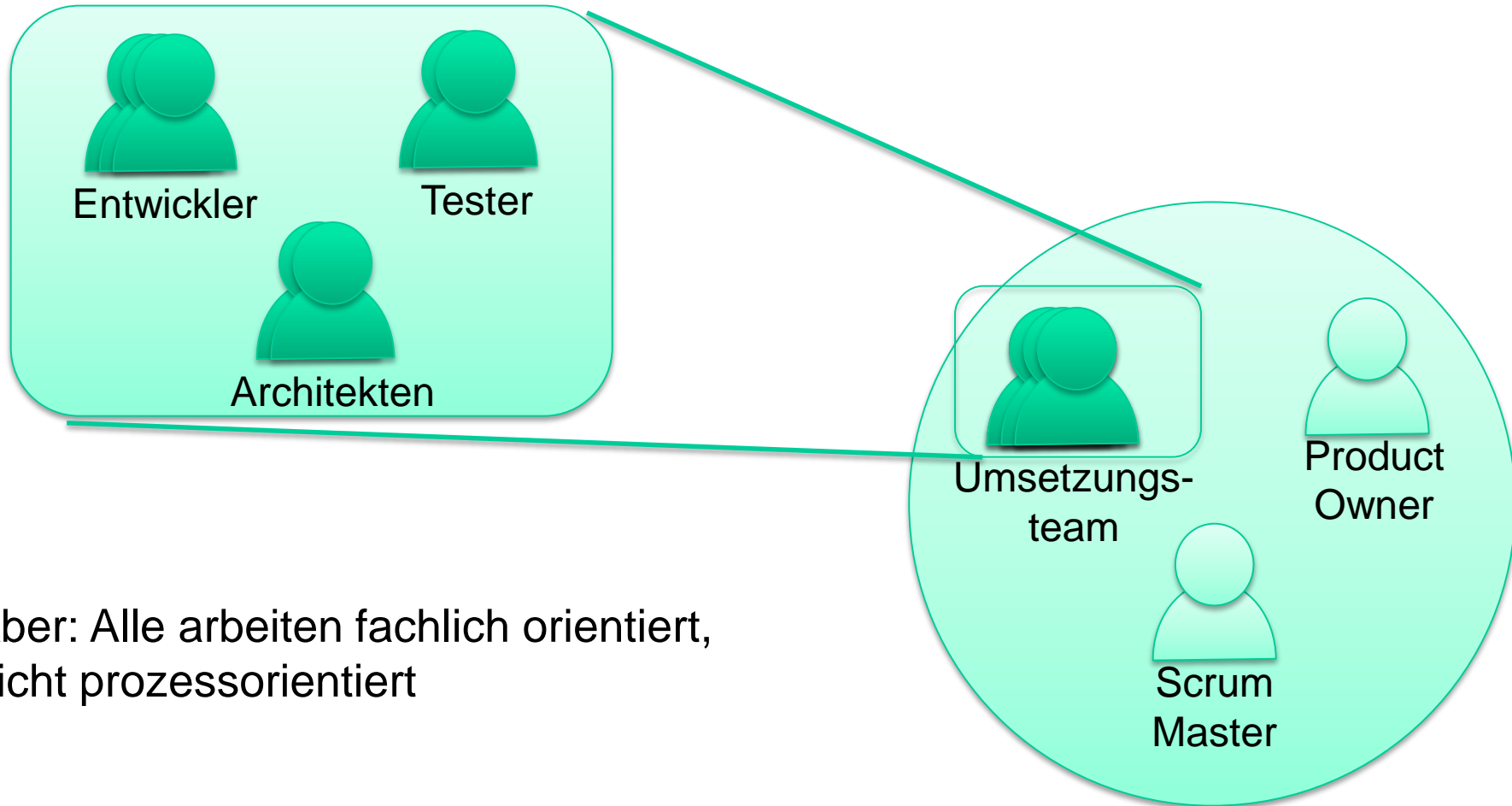
Der Start

- Iterationen
 - Feedback
- Agile Planung (Story Cards, Schätzung, Priorisierung)
 - Kommunikation
 - Einfachheit
- Retrospektiven zur schrittweisen Verbesserung des Projekts
- Erfahrung von Coaches: Je „radikaler“, desto mehr Effektivität kann man erreichen.

Scrum-Sicht der Rollen

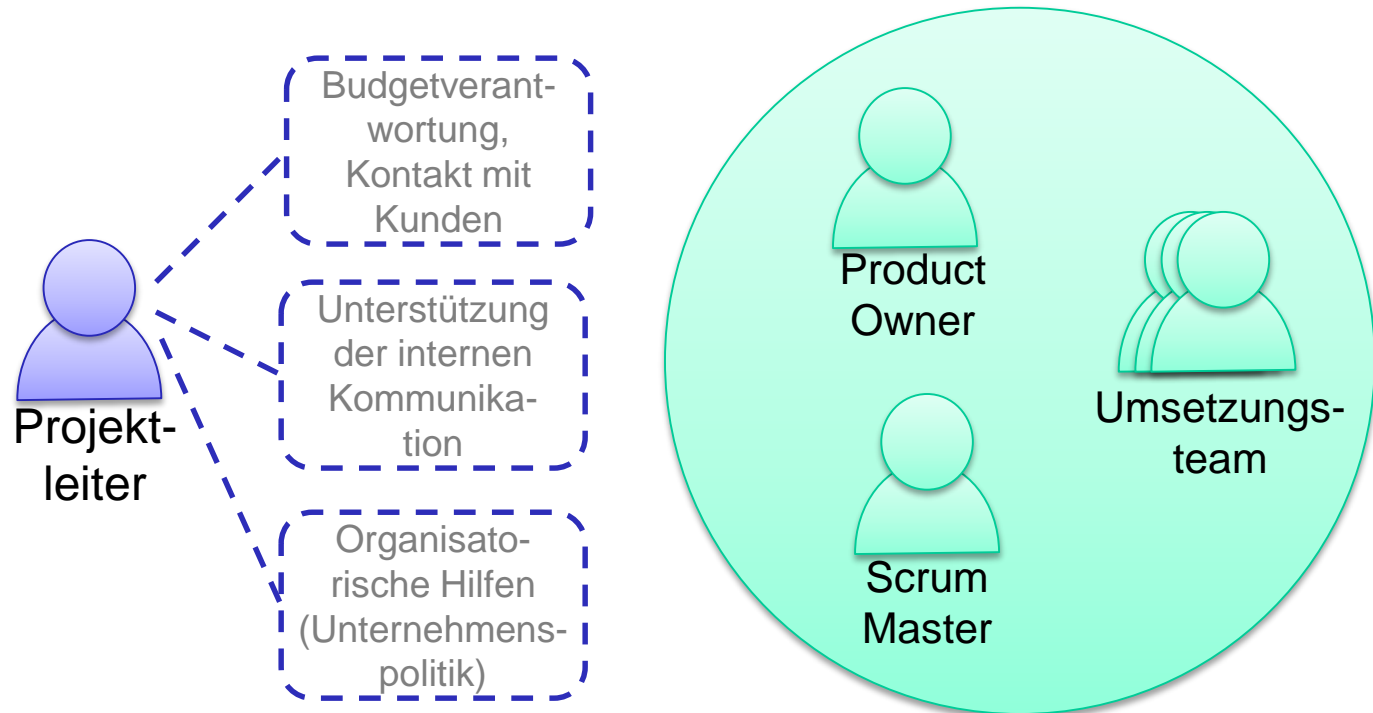


Wie passen die Rollen eines normalen Projektteams hierzu?

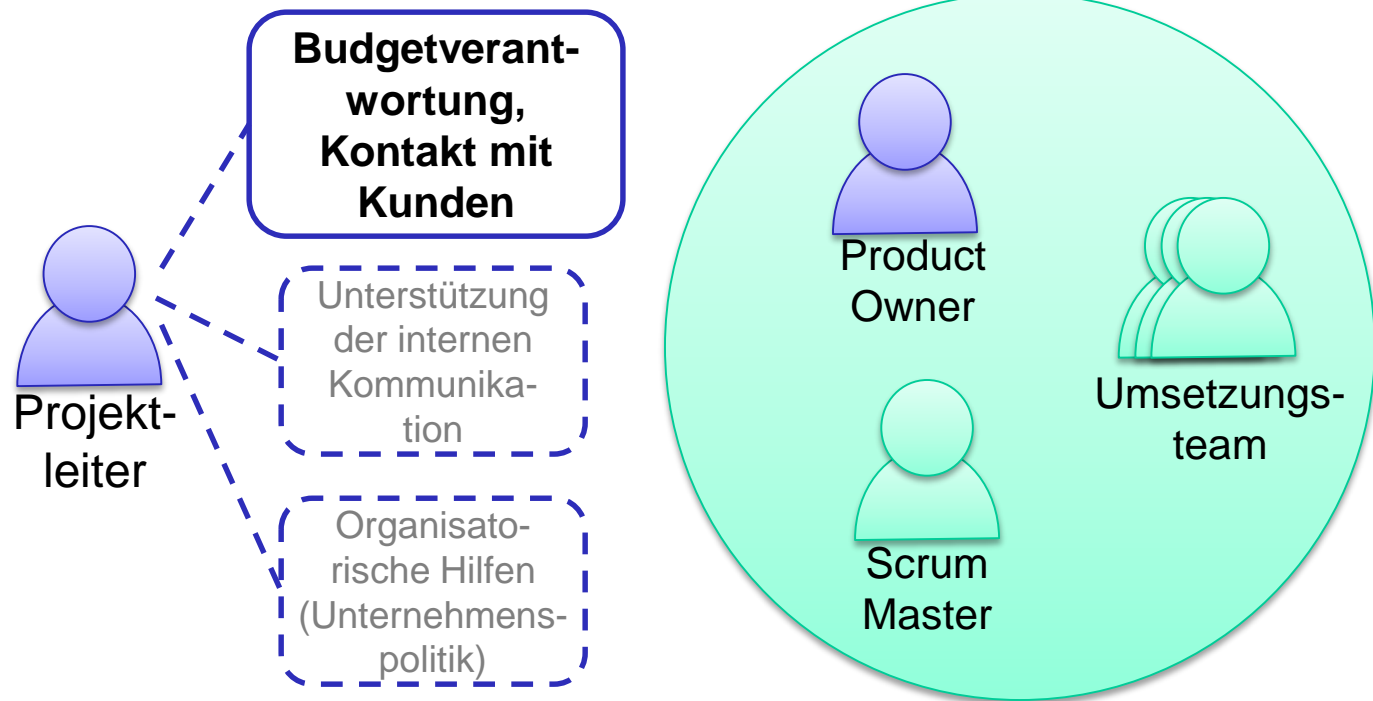


Aber: Alle arbeiten fachlich orientiert,
nicht prozessorientiert

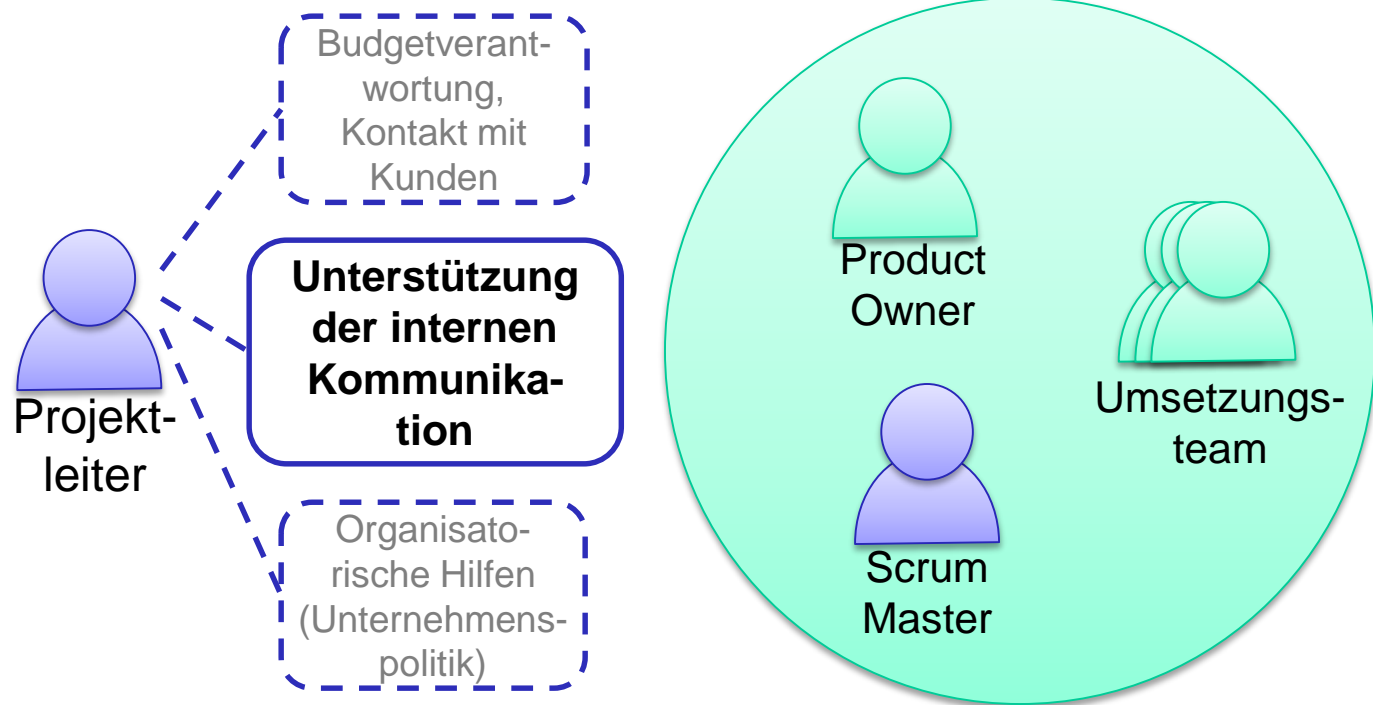
Welche Art von Unterstützung braucht das Team?



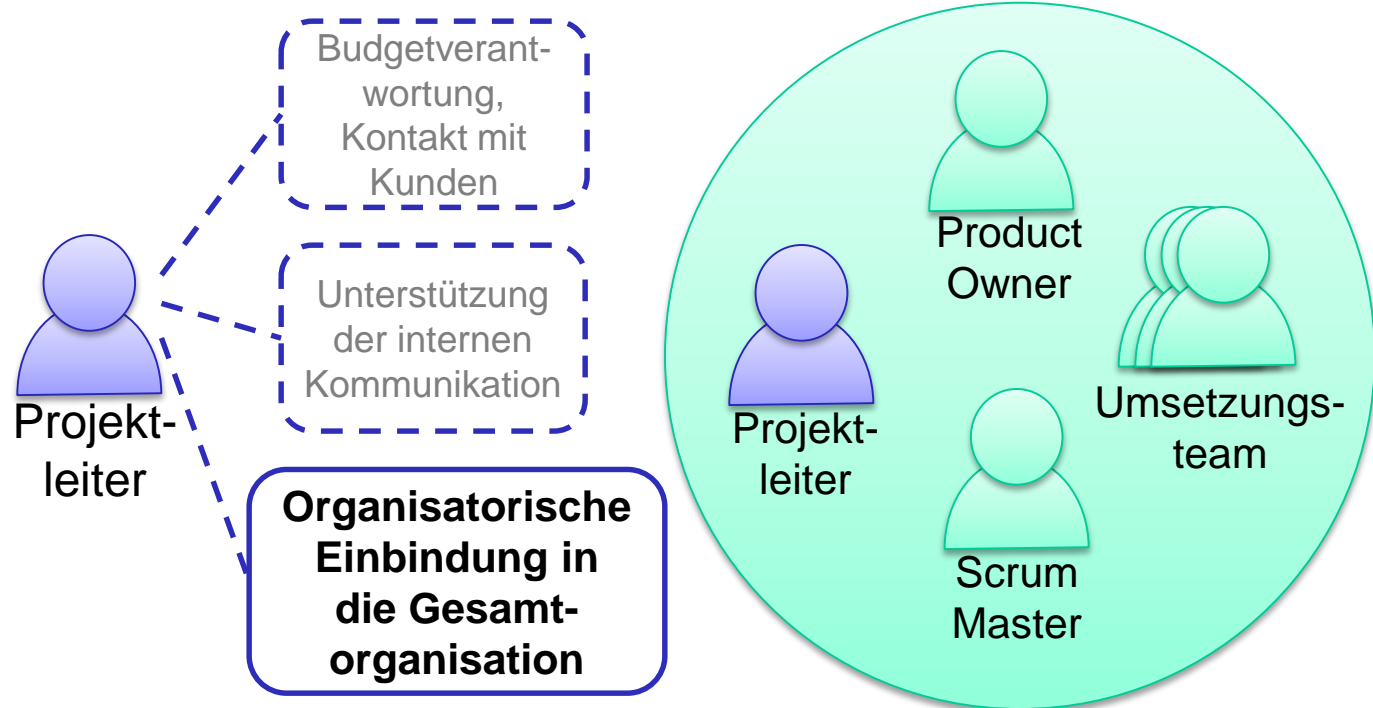
Option 1



Option 2



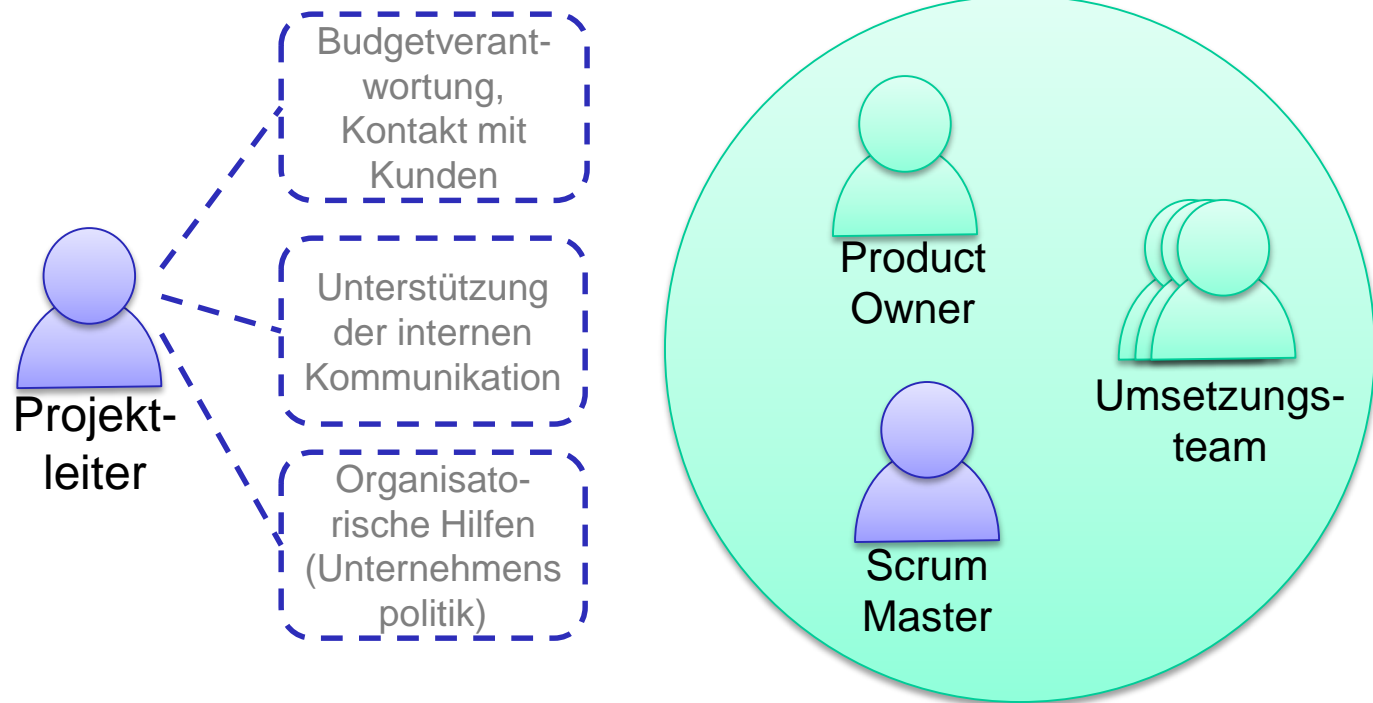
Option 3



Projektleiter wird in „Scrum Master“ umbenannt, behält aber alle bisherigen AKVs.

Geht auch, ist aber weniger effektiv.

Option 4



Agile Methoden skalieren

26.11.2015

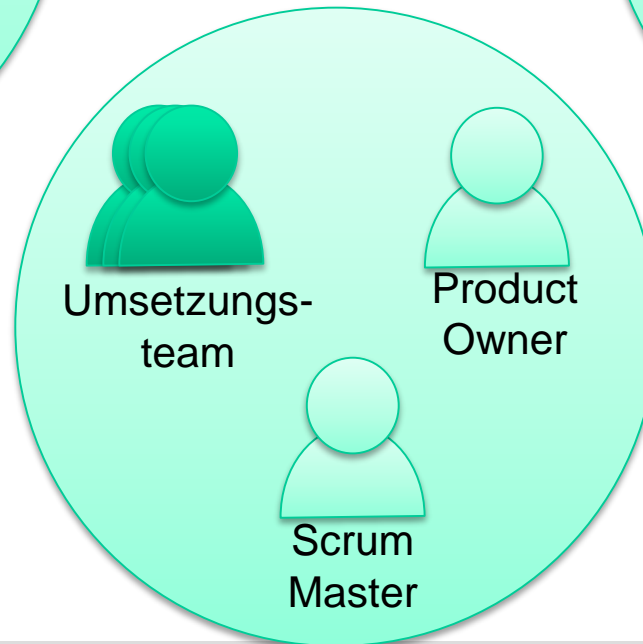
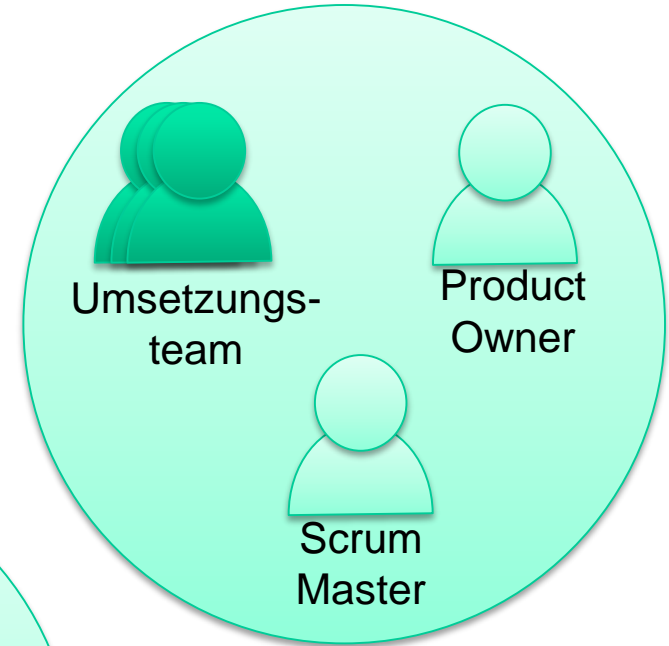
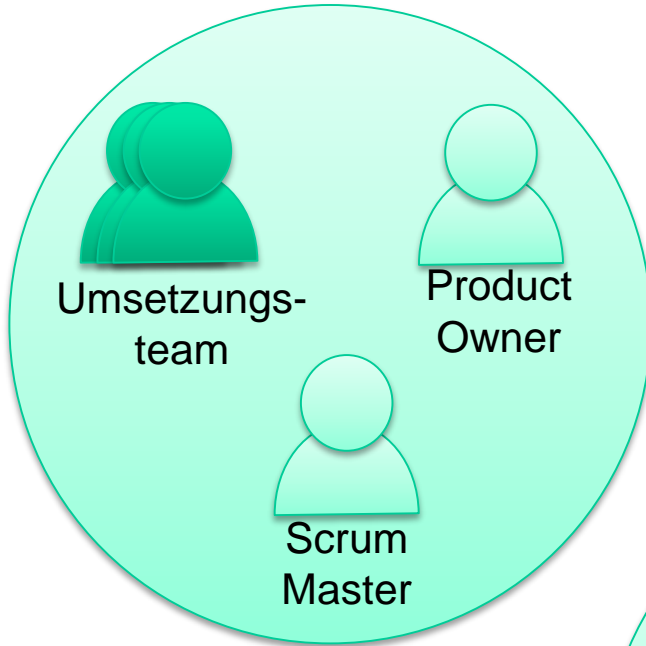
Können wir nun 100 Leute in einen Raum stecken, um wöchentlich ihre nächsten Schritte detailliert zu planen?


Vielleicht nicht,
gemeinsam
Kommunikation



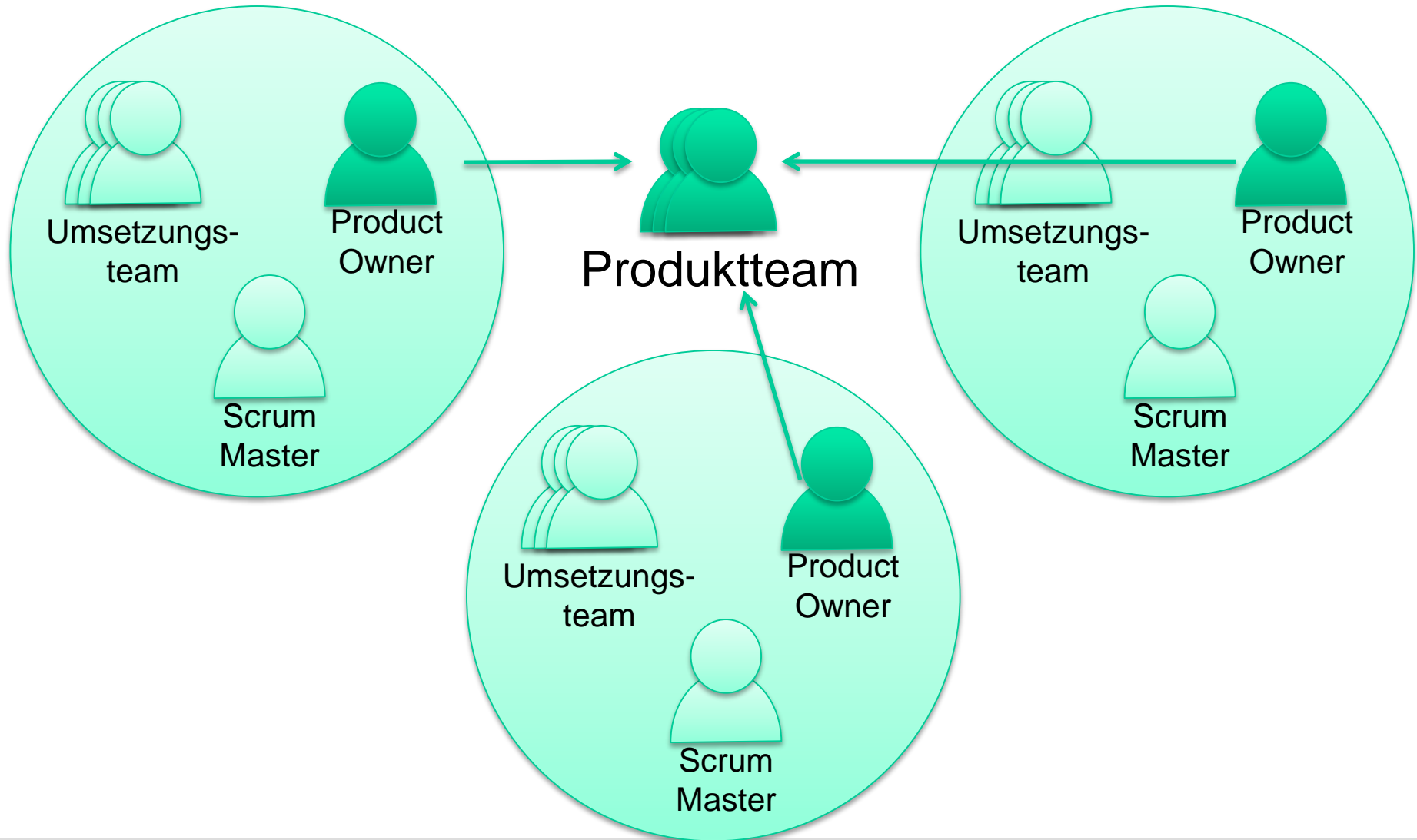
Bewegen,
durch
s bezeichnet.

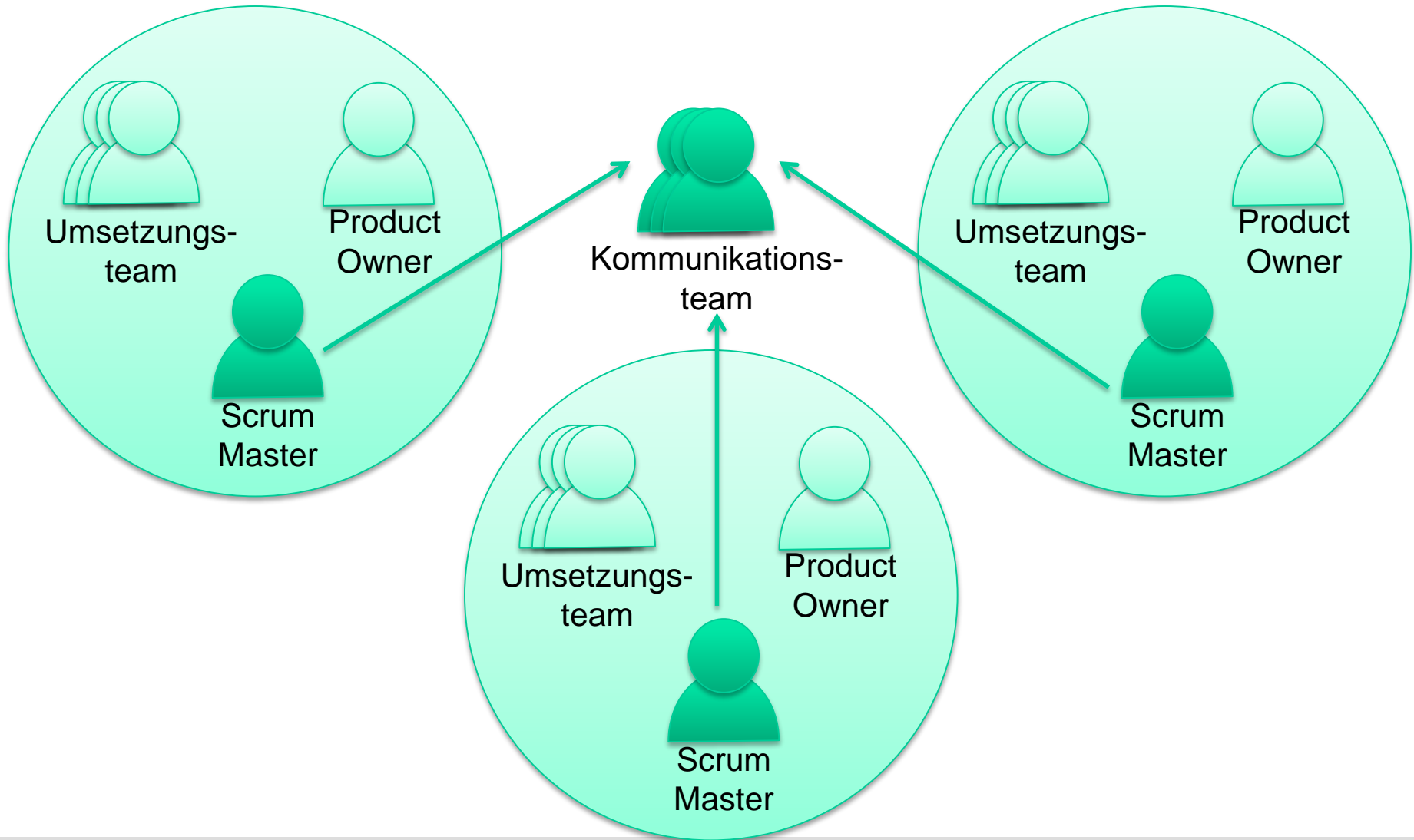
Featureteams



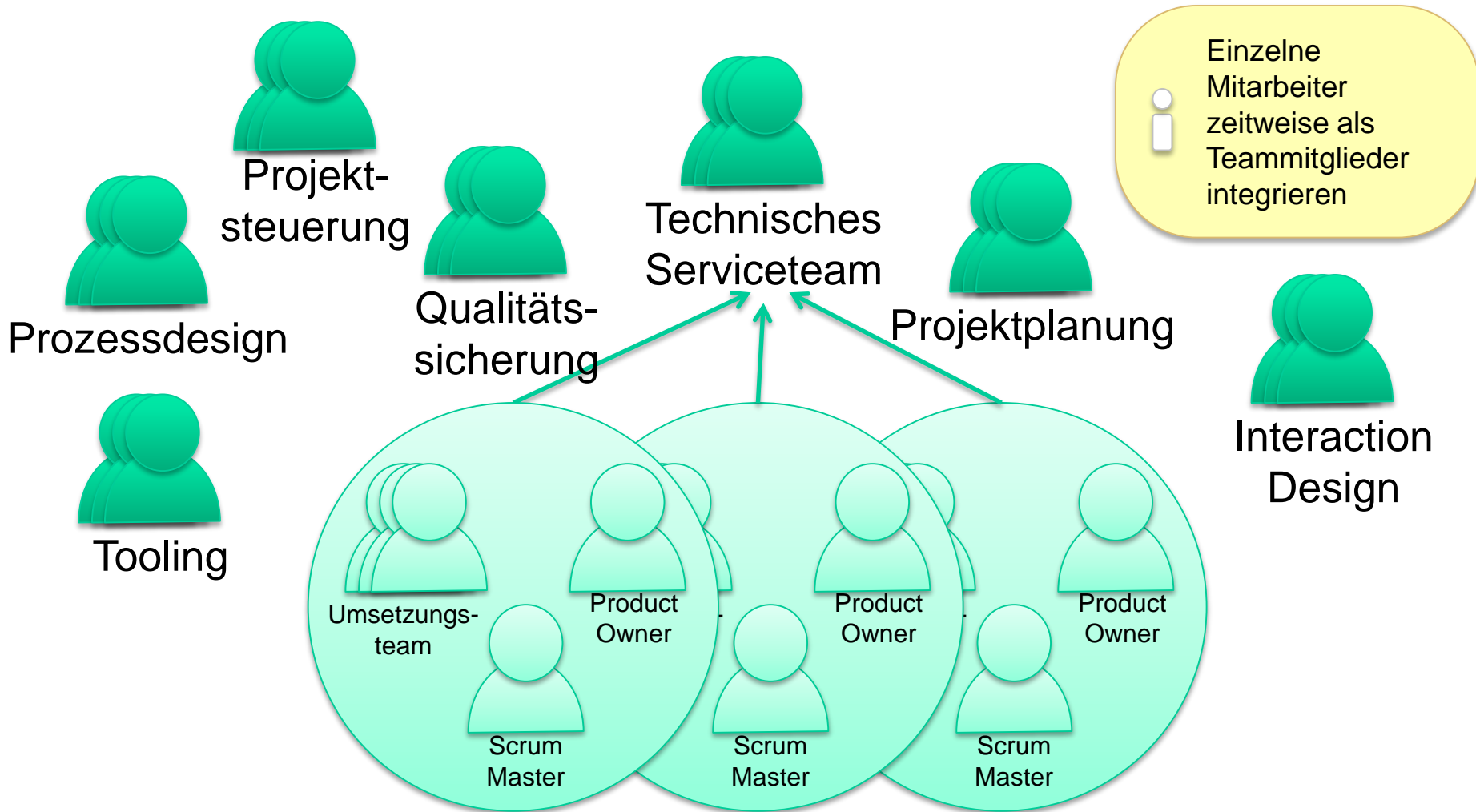
 Aufteilung nach
Divide and
Conquer

Koordination der Features





Unterstützung durch Service-Teams



Auf den ersten Blick...

Schneider, K. (2003):
SQM congress 2003, Köln, SQS

- „paarweises Arbeiten verschwendet Zeit“
- „unsere großen Systeme vertragen keine schnellen Releases“
- „bei 35 Stunden pro Woche ist „sustainable pace“ kein Thema“

Ernster zu nehmen

- „unsere Leute werden paarweises Arbeiten und collective ownership hassen!“
- „bei der Art von Systemen, die wir haben, ist refactoring/cont. integration/test first nicht möglich“

Generelle Einwände

- „das ist nichts Neues - 20 Jahre alt“
- „gerade bei uns geht das nicht“
- „wir machen das schon - wild Hacken, meine ich, hahaha!“

Wirkliche Knackpunkte

Wie meinen Sie das?

- Sie implementieren, ohne genau zu wissen, was wir wollen?
- Sie verlangen einen Kundenvertreter ständig bei Ihnen vor Ort?
- Sie versuchen, Dokumentation „zu vermeiden“?
- Sie wollen sich die Arbeit „so einfach wie möglich machen“? Und QM?

Mal ehrlich!

Du fühlst Dich unwohl, weil sich so viel auf einmal ändert. Kannst Du den Überblick bewahren? Oder würdest Du lieber auf die bewährte Weise scheitern, als riskanten Erfolg zu haben?



Was bekommen wir?



- Wir schreiben ausführliche Spezifikationen - um zu wissen, was wir kriegen. Und jetzt?
- Woher wissen wir, was wir im fünften Inkrement tun?
- Wir kriegen nicht nach jedem Inkrement neues Geld!
- Wir arbeiten mit Festpreisverträgen. Geht das hier auch?
- Wieso sollten wir diesem agilen Auftragnehmer vertrauen?

- Jetzt nicht mehr
 - Spezifikationen sind zu aufwändig - und sie veralten, während man sie schreibt. Ja, Unsicherheit bleibt.
- Gar nicht, wir vermuten es nur
 - Masterplan machen – und ändern
- Stimmt; wir beantragen gleich das gesamte Geld.
- Ja:
 - Basieren auf Explorationsphase
 - Um gute Domain-Beschreibung herum
 - Einfache Änderungen ohne Vertragsänderungen
- Gute Erfahrung in der Explorationsphase



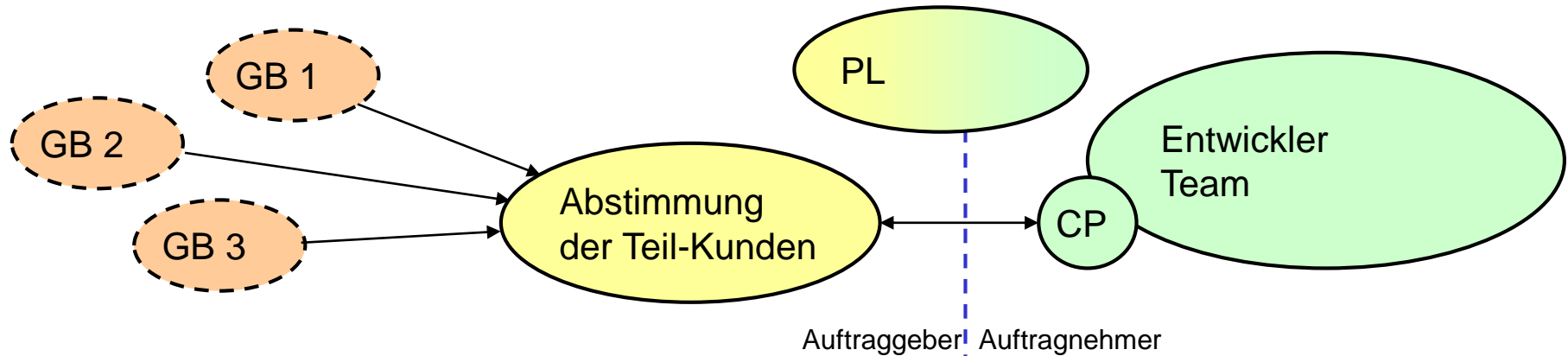
- Auf diese Person können wir nicht verzichten!
- Mehrere GBs sind beteiligt - es gibt nicht den Kunden!
- Eine Person kann nicht *alle* GBs im Detail kennen
- Um Fehler zu vermeiden, ist viel Abstimmung nötig
- Verliert der On-site Customer nicht die GB-Bodenhaftung?
- Jeder GB-Vertreter wird *seinen* Bereich bevorzugen

- Vollzeit ist selten
 - Teilweise reicht Telefon-Verfügbarkeit
 - Customer Proxy erwägen! (unten)
- Sie müssen aber einen benennen!
 - (kann später wechseln)
- Muss er/sie auch nicht!
 - Änderungen sind möglich - Fehler erlaubt
 - Wichtig ist Fähigkeit zu entscheiden - ohne Abstimmung
- Kann passieren!
 - In sehr langen Projekten: auswechseln
- Customer proxy Konzept
 - Produktmanager „spielt“ den Kunden

Aus: Schneider, K. (2003): SQM congress 2003, Köln, SQS

„Customer Proxy“ Konzept

- Entwickler „spielt Kunde“
 - gute Kenntnis des Anwendungsbereichs
 - intensiver Kontakt mit Entwicklern
 - nimmt an Abstimmungen der Teil-Kunden teil
- Wichtig
 - schnelle Entscheidungen durch eine Person
 - falsche Entscheidungen später revidierbar
 - Das haben wir in der Praxis gesehen und selbst gemacht: Es funktioniert!



Aus: Schneider, K. (2003): SQM congress 2003, Köln, SQS



- Software ohne Doku ist nutzlos
- Entwicklerteams verändern sich: ohne techn. Doku sind wir verloren
- QM-Handbücher etc. verlangen viele Dokumente, Reviews, Aktivitäten
 - sollen wir das alles wieder aufgeben?
 - oder brauchen wir 2-Klassen-System?
- Können agile Techniken sich an diese Regeln halten und doch agil bleiben?

- Stimmt!
 - Wenn Sie sie brauchen, bestellen Sie sie (und zahlen dafür)
- Wirklich?
 - Trotz generierter/eingebetteter Doku?
 - Falls wirklich: verlangen+Agilität verlieren!
- Vier-stufiger Ansatz
 - 1 Was man schon tut auf die Regeln abbilden
 - 2 QM-Abteilung beteiligen
 - 3 Verhandeln über das, was noch „fehlt“
 - 4 Mit QM eine agile Variante für QM bauen
- Nicht, wenn man alles übernimmt
 - Das wäre Zauberei.
 - Travel light: wenig Wirksames aufgeben
 - Alle agilen Qualitätsaspekte bewahren