Womit beginnt die Anforderungsanalyse?
Systemziele und Systemkontext sind die Basis für alle Anforderungen und damit auch für das gesamte Projekt. Wichtig ist ein wertungsfreier Umgang mit Systemzielen und Systemkontext. Ebenso wichtig ist es, zu Beginn alle am Projekt Beteiligten Personen mit ihren Rollen und Interessen zu kennen.

Wie werden Anforderungen ermittelt?
Kreavitätstechniken (Brainstorming, Wechsel der Perspektive, ...)
Befragungstechniken (Fragebogen, Interview, ...)
artefaktbasierte Techniken (Eingabe- , Ausgabedokumente, → Wiederverwendung
Audio-, Videoaufzeichnungen
Anwendungsfallmodellierung, Essenzbildung

Welche Perspektiven hat die Anforderungsmodellierung?
funktionale P., datenorientierte P. → Struktur (temporär, persistent),
datenflussorientierte P., objektorientierte P., zustandsorientierte P.

Perspektive von außen, Perspektive auf das Innere

Perspektive auf die Struktur gerichtet, Perspektive auf das Verhalten gerichtet

Wie werden die Anforderungen beschrieben?
- natürlichsprachlich
- durch Modelle → grafische Darstellungen
- natürlich sprachlich, aber nach Schatzschablonen

Welche Diagrammarten bietet die UML zur Unterstützung der Anforderungsanalyse?

- Anwendungsfalldiagramm
- Aktivitätsdiagramm
- Klassendiagramm
- (Sequenzdiagramm)
- Zustandsdiagramm

Praktikum

UML: Unified Modeling Language

Welchen Sinn hat Konfigurationsverwaltung?

(siehe Ludewig, Lichter „SW-Engineering“, 3. Auflage, S. 558)
Die Konfigurationsverwaltung ist eine Rolle oder Organisationseinheit im Kontext des SW-Entwicklungsprozesses, die die SW-Einheiten

- identifiziert,
- bei Bedarf bereitstellt
- ihre Änderungen überwacht und dokumentiert.

Dazu gehört auch die Rekonstruktion älterer SW-Einheiten und Konfigurationen.

D.h. ein Konfigurationsverwaltungssystem ermöglicht das effiziente Bereitstellen definierter Konfigurationen eines SW-Systems abhängig von Varianten und Versionen.

Das Konfigurationsmanagement ist demzufolge eine Methode für die Organisation der täglichen Arbeit. (→ organisatorische Qualitätssicherung)

Welche Beziehung besteht zwischen Konfigurations-, Varianten- und Versionsverwaltung?
Versions- und Variantenverwaltung sind Bestandteile der Konfigurationsverwaltung.

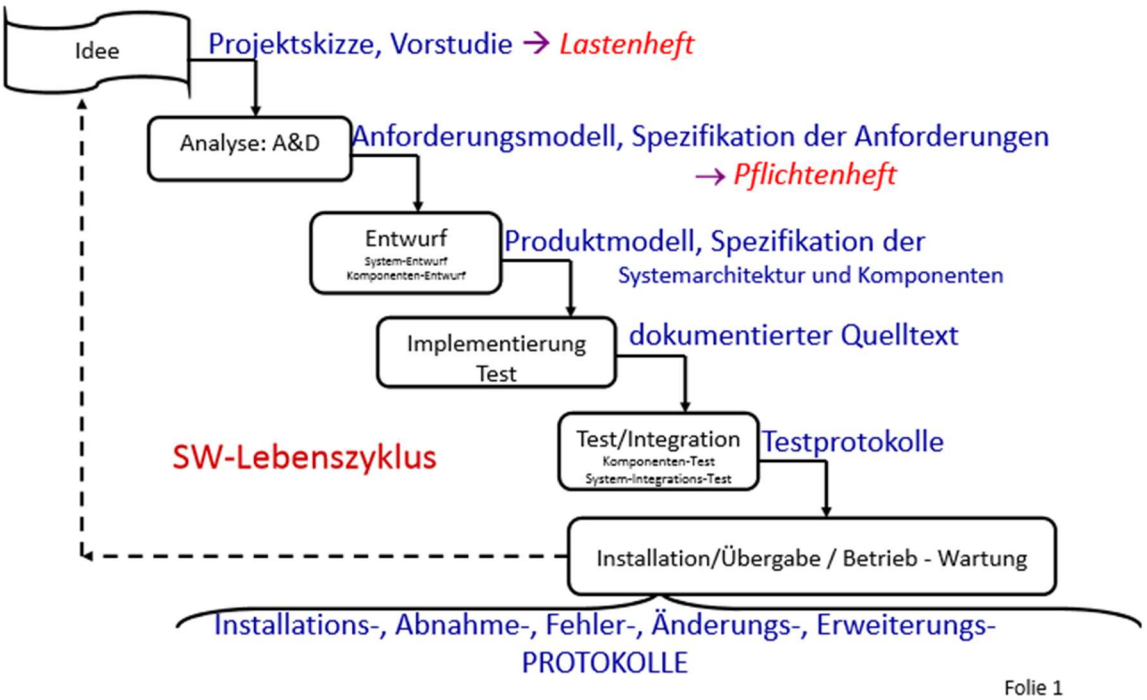
Begriffe „Lastenheft“ und „Pflichtenheft“:

Klaus Pohl, Requirements Engineering, 978-3-89864-342-9 , S. 232 bzw. 234

Das Lastenheft (siehe DIN 69905 1997)
„Das Lastenheft enthält
eine Definition der Systemvision
eine Beschreibung der wesentlichen Systemziele (Funktionen und Qualitäten)
und benennt wichtige Kontextaspekte (z.B. Rahmenbedingungen)
der vier Kontextfacetten sowie ihre Beziehung zur Vision
und zu den definierten Systemzielen.“

Das Pflichtenheft (siehe DIN 69905 1997)
„Das Pflichtenheft detailliert die im Lastenheft beschriebene Vision und die Systemziele (abstrakte Funktionen und Qualitäten) sowie ggf. im Lastenheft
definierte Rahmenbedingungen im Hinblick auf die
angestrebte technische Umsetzung (Realisierung) des Systems.“

Klassisches Vorgehen bei der Software-Entwicklung - SW-Lebenszyklus



Folie 1

Was unterscheidet Lastenheft und Pflichtenheft?
Das Lastenheft beschreibt die System-VISION des Auftraggebers.
Das Pflichtenheft detailliert diese Vision aus der Sicht des Auftragnehmers.
Es ist eine mögliche Realisierungsvariante.

An wen ist ein Pflichtenheft adressiert?
- an den Auftraggeber
- an den Auftragnehmer
- ggf. an Dritte

Welchen Charakter hat ein Pflichtenheft?
Vertragscharakter

Was muss ein Pflichtenheft enthalten?

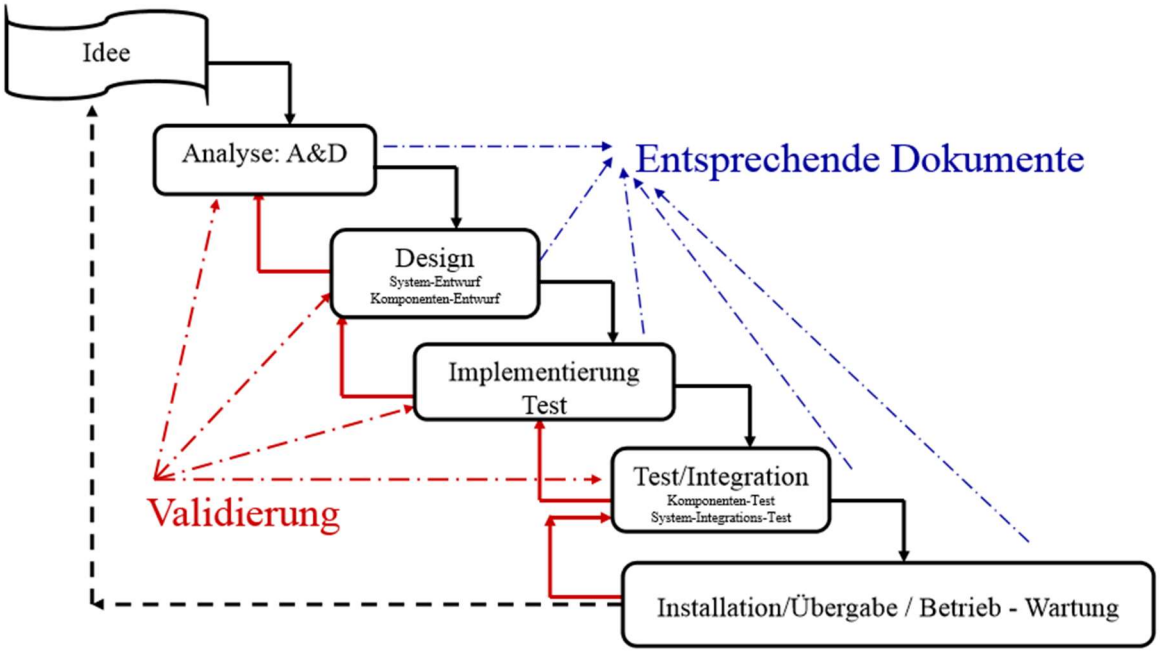
Alle Anforderungen an das zu entwickelnde SW-(Teil-)System,
die (zu dieser Zeit) erkennbar sind.
D.h. funktionale und Qualitätsanforderungen ebenso wie einschränkende Rahmenbedingungen (organisatorische, rechtliche und technische/technologische,).

Da es in der Regel zwei Adressaten (AG/AN) und quasi Vertragscharakter hat, muss es analog zur Anforderungsanalyse folgende Eigenschaften haben:
- verständlich sein für den Auftraggeber und hinreichend präzise für den Auftragnehmer,
- korrekt, vollständig, eindeutig, widerspruchsfrei, minimal, erweiterbar sein,
- realisierbar, nachvollziehbar und hinsichtlich der Erfüllung überprüfbar sein.

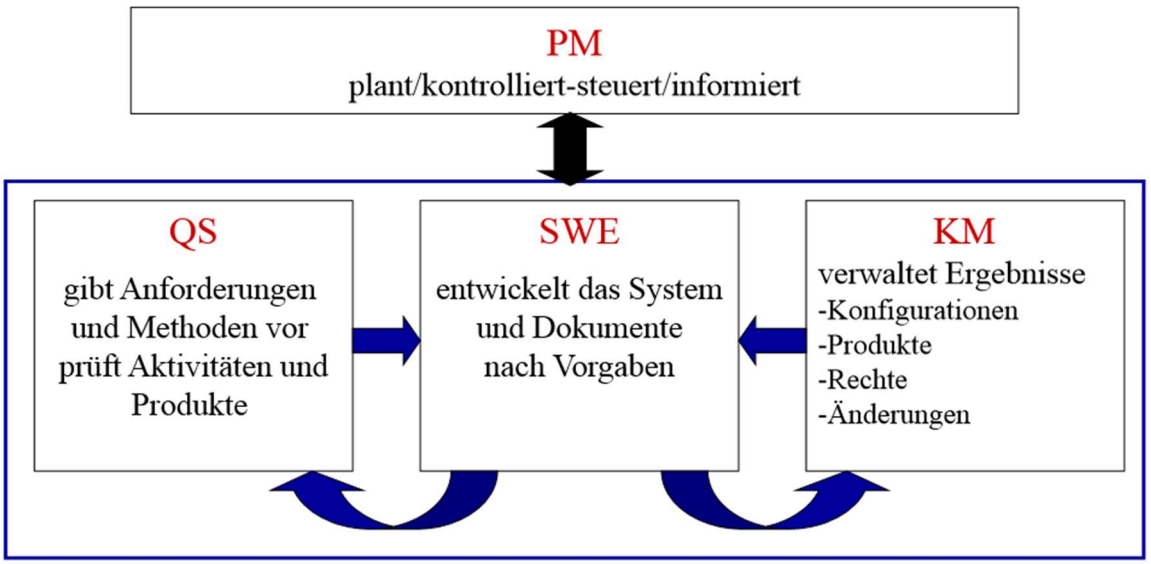
Wie kann das Pflichtenheft aufgebaut sein ?

(1) Ausgangssituation und Zielsetzung
(2) Systemeinsatz und Umgebung
(3) Benutzerschnittstellen
(4) Funktionale Anforderungen
(5) Qualitätsanforderungen
(6) Rahmenbedingungen (techn./technologisch, org., rechtlich)
(7) Fehlertoleranzmaßnahmen
(8) Anforderung an die Dokumentation
(9) Abnahmekriterien
Glossar (Begriffslexikon)
Index
Anhang

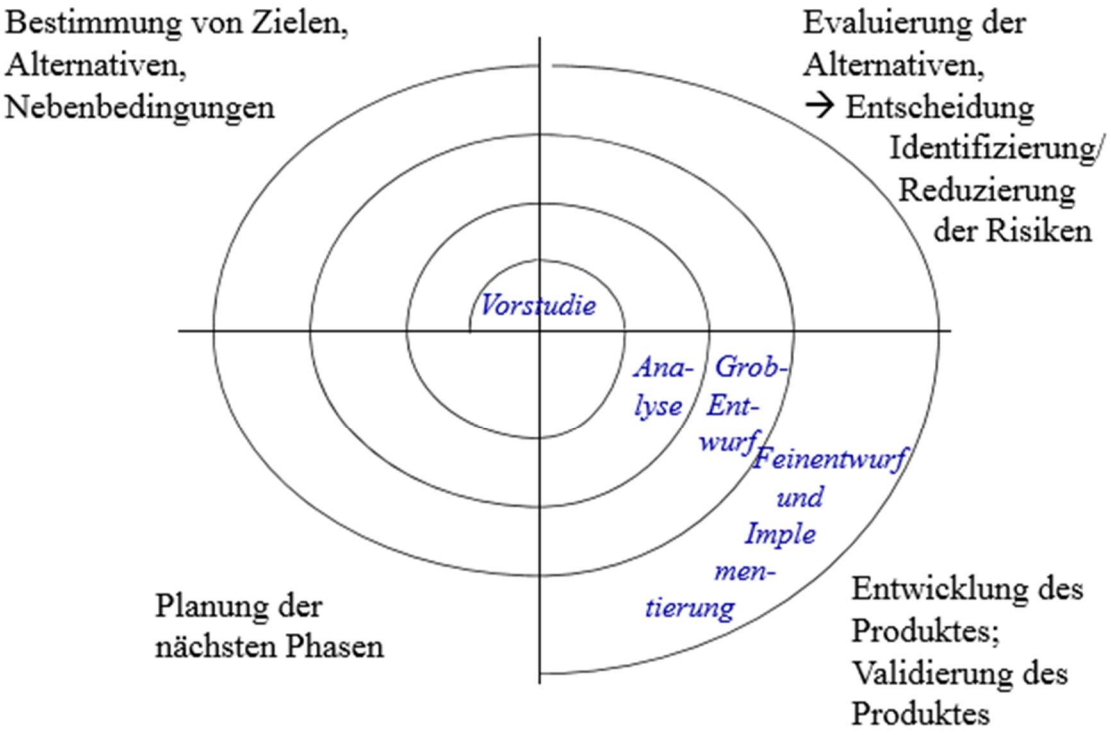
SW-Lebenszyklus mit Rückkopplungen



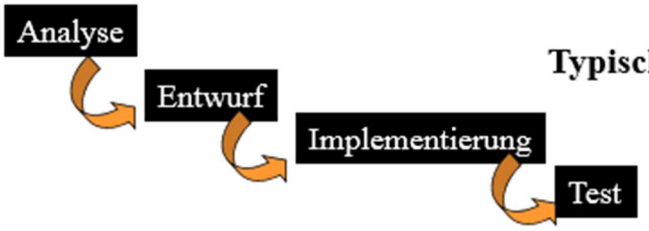
V-Modell - Interaktion der Submodelle – 3 -



Spiralmodell (nach Boehm)

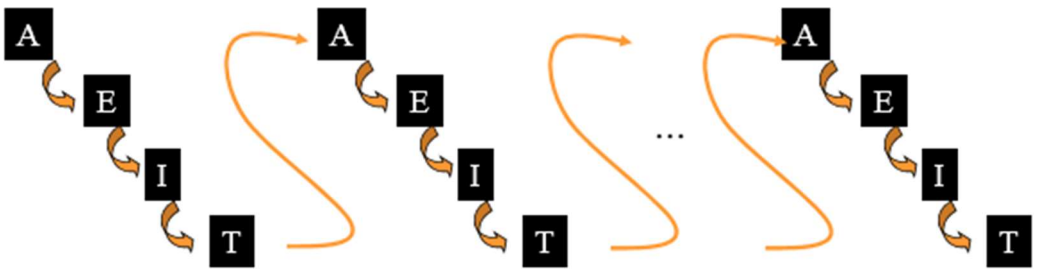


Unified Process → **R**ational **U**nified **P**rocess → RUP
(Booch, Jakobsen, Rumbaugh)

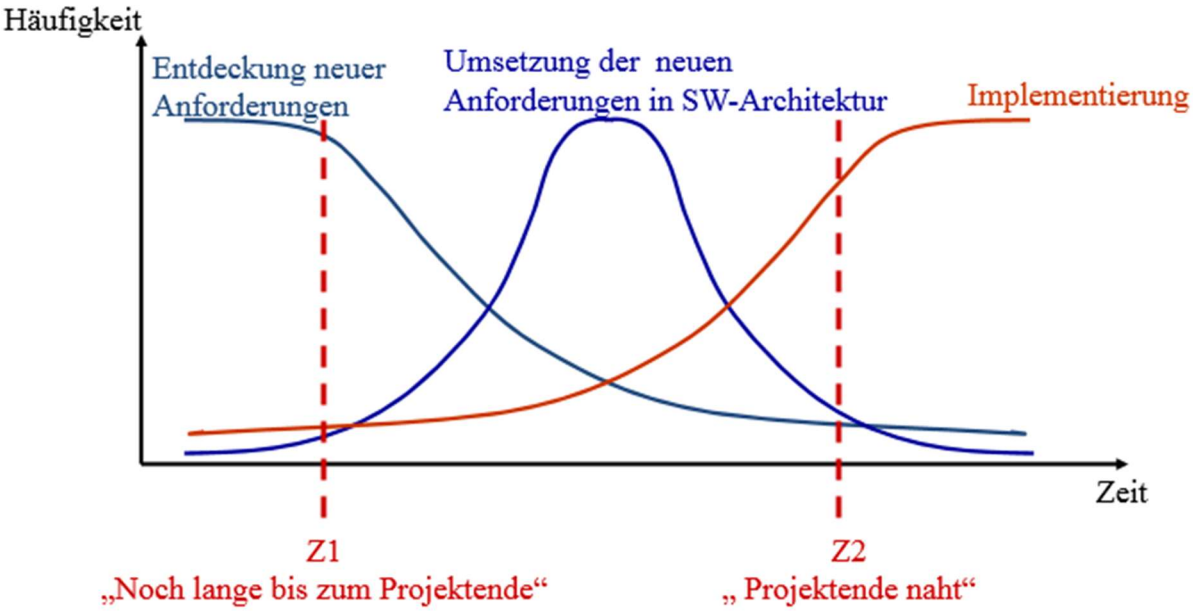


Typisches Wasserfallmodell

Unified Software Development **Process:**
iterativ, inkrementell – **„kleine Wasserfälle“**



Iterativ-inkrementelles Vorgehen



Welche Vorgehensmodelle gibt es in der SW-Entwicklung?

SW-Lebenszyklus

SW-Lebenszyklus mit Rückkopplungen

V-Modell

Spiralmodell nach Boehm

iterativ-inkrementelles Vorgehen (Unified Process)

agile SW-Entwicklung

agil - aber wodurch?

„Kleine Schritte führen zum Ziel."

- Stück für Stück implementieren
- immer wieder überprüfen
- immer wieder miteinander reden
- einfache Lösungen suchen und implementieren
- wenn nötig korrigieren
- ...

d.h. agile SW-Entwicklung bedeutet:

Akzeptanz von Wandel u n d Forderung von Wandel,
wenn es sinnvoll erscheint

SW-Entwicklungsteam muss gewandt sein

nötig ist:
- umfangreiches theoretisches Wissen
- praktische Erfahrung
- Bereitschaft zur Überarbeitung und Änderung

Cockburn, Alistair
Agile SW-Entwicklung
Bonn mitp Verlag
2003

extremes Programmieren (XP)

Kent Beck, Ward Cunningham, Ron Jeffries

⇨ für kleinere/mittlere Projekte
mit sich immer ändernden Anforderungen

⇨ Kunden **und** Software-Entwickler
bewusst auf das gleiche Ziel ausgerichtet

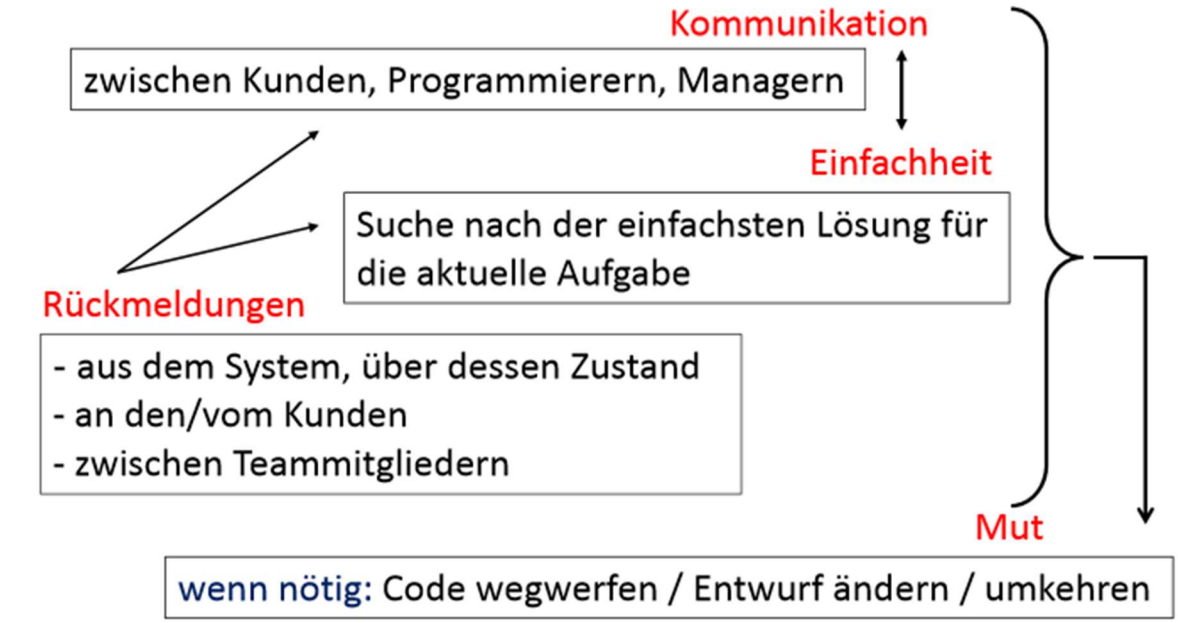
⇨ code-zentriert (nicht architekturzentriert wie bisher)

⇨ vier Grundwerte:
Kommunikation, Einfachheit, Rückmeldung, Mut

⇨ bewährte Praktiken:
konsequent und im extremen Maß angewandt

hochspezialisierte, weitblickende, disziplinierte Entwickler

extremes Programmieren (XP) 4 Grundwerte



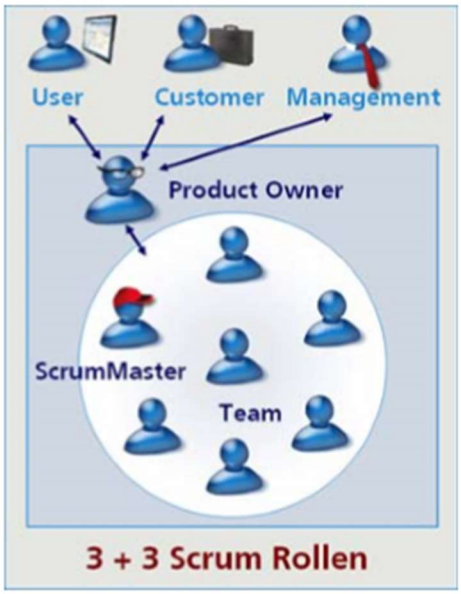
SCRUM

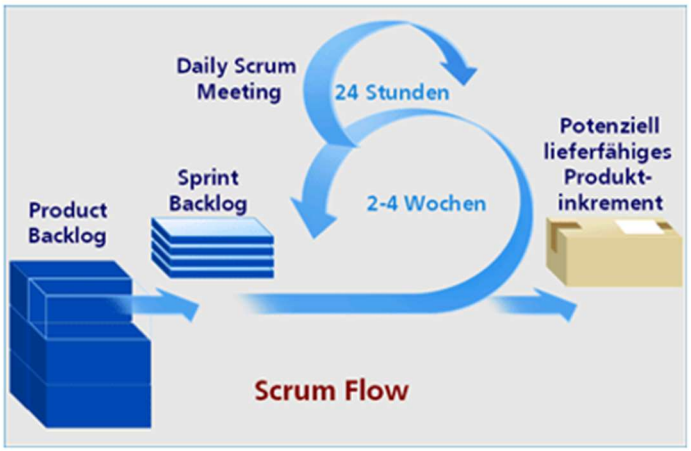
SCRUMGUIDE
Ken Schwaber, Mai, 2009

grundlegende Forderung:

- Transparenz
- Überprüfung
- Anpassung

agile SW-Entwicklung → Beispiel: Scrum
→http://www.microtool.de/instep/de/prod_scrum_edition.asp





Praxisbericht:
Dipl.-Inf. Stephan Janosch, mpi cbg
Max-Planck-Institut für Zellbiologie und Genetik, DD
24.1.2017, Vorlesungszeit: 13:20 Uhr