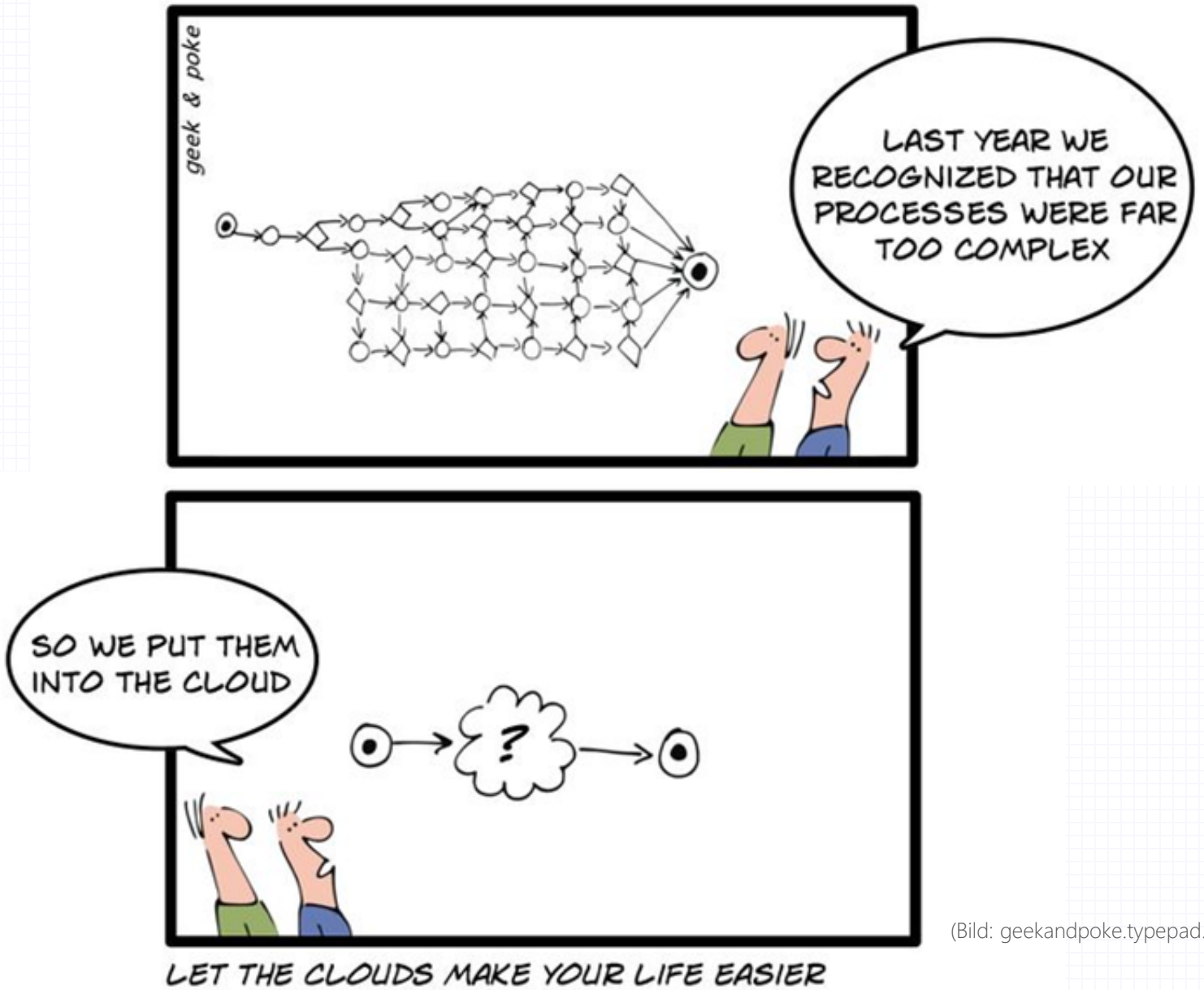
SOFTWARE-ENTWICKLUNGSPROZESSE
Moderne Prozesse zur Softwareentwicklung

Herwig Mayr

Fakultät für Informatik, Kommunikation und Medien
Fachhochschule OÖ, Hagenberg



(Bild: geekandpoke.typepad.com)

Moderne Softwareentwicklung

(Bild: http://www.viatec.at)

Positive Charakteristika:

+ enge Kundeneinbindung
+ iterativ-inkrementelle Entwicklung
+ effiziente Werkzeugnutzung

Negative Charakteristika:

− weniger erfahrene Entwickler (heterogene Teams)
− Aversion gegenüber Management-Information
− kaum Prozessverständnis, geschweige denn -verbesserung

Agile Softwareentwicklung: Anlass

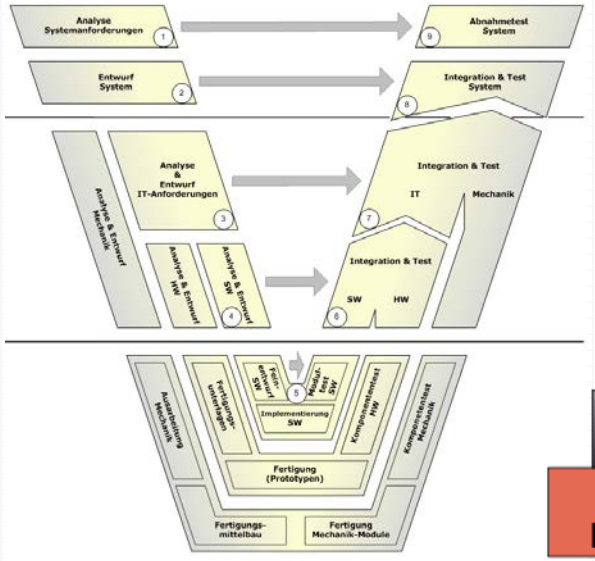
Software-Prozessmodelle wurden in den Neunziger Jahren immer umfangreicher!

(Grafik: http://oeag.test.oeag.at)

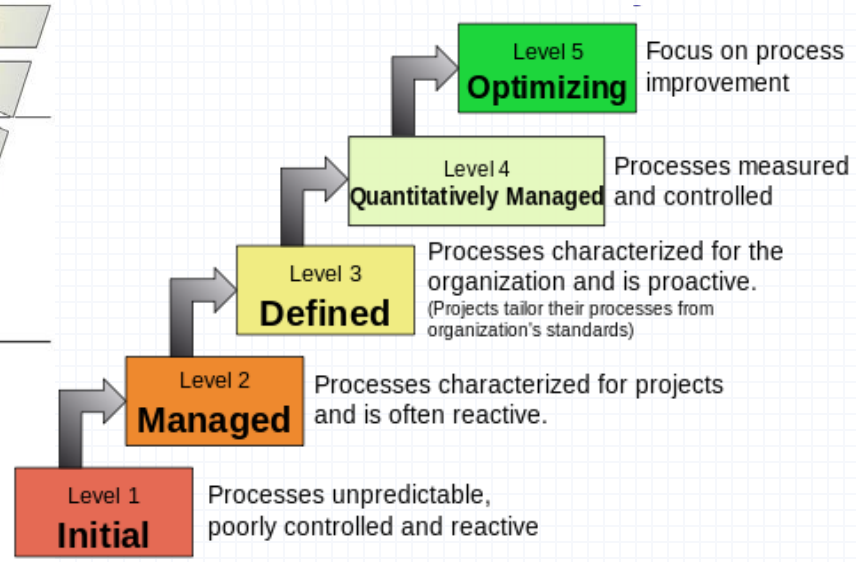
⇒„Dokumentationsmodelle"

z.B.

- in Deutschland „V-Modell"
- in USA „SW-CMM"



(Grafik: TU München)



(Grafik: Wikipedia)

Agile Softwareentwicklung: Auslöser

Agiles Manifest (nach Beck *et al.*, 2001):

Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

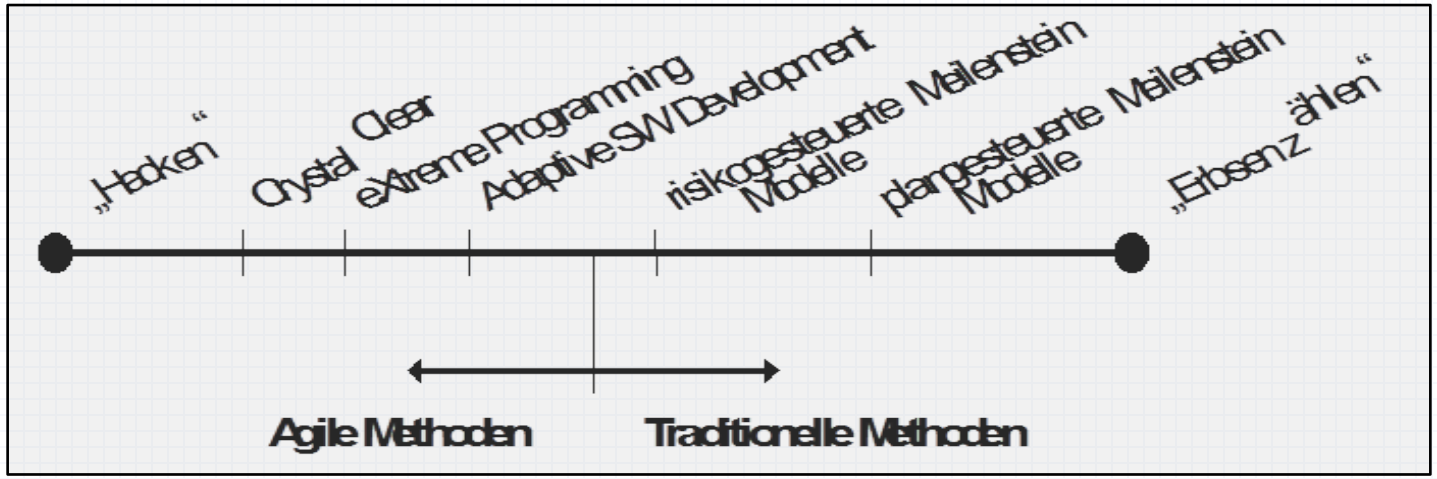
Individuals and interactions over *processes and tools*,
Working software over *comprehensive documentation*,
Customer collaboration over *contract negotiation*,
Responding to change over *following a plan*.

That is, while there is value in the items on
the right, we value the items on the left more.

(Grafik: Snowbird, sandy,utah.gov)

Agile Vorgehensmethoden

Bandbreite des Methodenspektrums:



Paradigmenwechsel durch agile Methoden:

- einheitlicher Entwicklungsprozess nicht definierbar
- Projekte nicht langfristig detailliert planbar
- Projektkontrolle ist ohne Entwicklerkooperation unmöglich

Vorteile agiler Vorgehensmethoden

+ iterativ-inkrementelle Entwicklung
+ kurze Releasezyklen
+ überprüfbare Qualität
+ intensive Kundeneinbindung
+ intensive Kommunikation im Team
+ Anwendung von Programmierstandards
+ einfaches Design
+ laufende (Code-)Reviews
+ testgesteuerte Entwicklung
+ kontinuierliche Integration

(Bild: www.dilbert.com)

Nachteile agiler Vorgehensmethoden

- nur hoch qualifizierte Mitarbeiter brauchbar
- schlechte Skalierbarkeit
- mangelnde Transparenz für das Management
- keine Migrationsmöglichkeit aus bestehenden Prozessen
- fehlender Qualitätsplan
- kontinuierliche Anforderungsänderungen
- keine Design-Reviews
- stark inkrementelles Design, fehlende Dokumentation dazu
- Code-Zentriertheit statt Design-Zentriertheit
- fehlende Quellen für Systemtests

(Bild: www.dilbert.com)

Größtes Problem: fehlendes „Big-Picture"-Design (Architektur!)

Folgen des Agilen Manifests (I)

(Bild: www.marconetz.de)

Prozesskrieg

traditionell	modern
schwergewichtig	leichtgewichtig
rigoros	agil
stabil	fragil

„If you want to start a religious or software war, issue an edict or a manifesto“ (K. Orr, Cutter 2003)

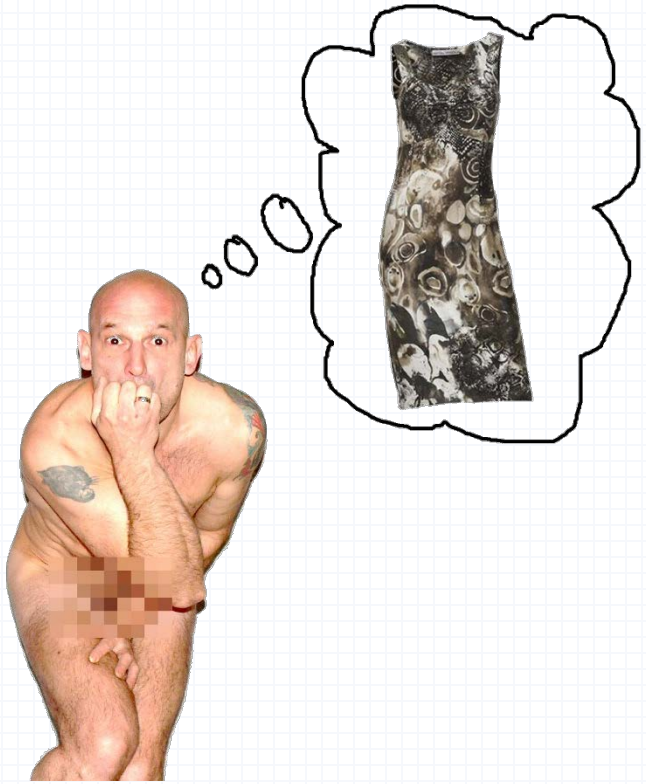
Folgen des Agilen Manifests (II)

„Des Kaisers neue Kleider“

⇒„Dokumentationsmodelle“ werden zu skalierbaren (Dokumentations-)Modellen.

z.B.

- in Deutschland vom „**V-Modell**“ zum „**V-Modell XT**“ (2005)
- in USA vom „**SW-CMM**“ zum „**CMMI**“ (2002)



(Bild: http://freie-zeit.eu/2012/07/24/des-kaisers-neue-kleider/)

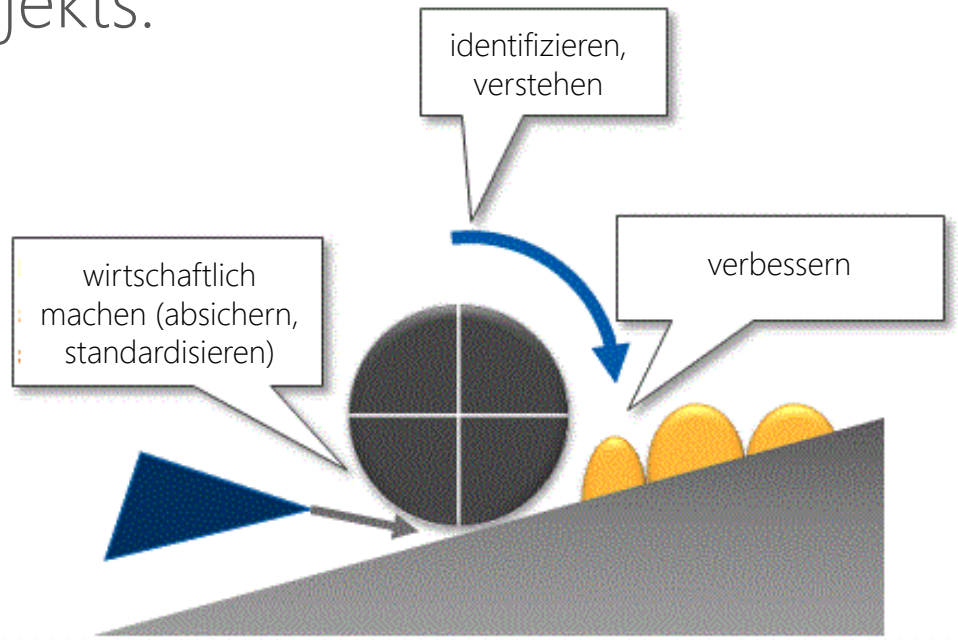
Prozessoptimierung [Wiederholung]

Motivation: Definierte (Teil-)Prozesse müssen **laufend angepasst und verbessert** werden, damit sich der **Aufwand amortisiert**!

Vor allem bei agilen Methoden Änderungen nicht nur von Projekt zu Projekt, sondern auch während eines Projekts.

Vier wichtige Schritte:

1. Prozess identifizieren,
2. Prozess verstehen,
3. Prozess wirtschaftlich machen,
4. Prozess verbessern.



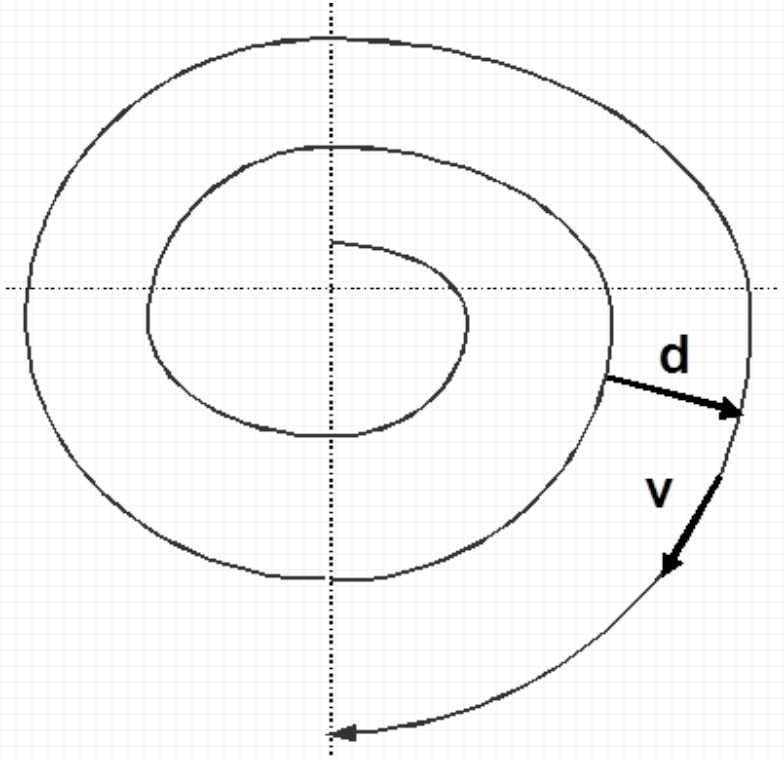
(Bild: www.ingenieurbuero-wilkens.de; adaptiert)

Prozessidentifikation

Bedeutung des Messens

Zwei Aspekte:

- Messen des **Projektfortschritts**
- Messen des Grades der **Zielerreichung**



traditionelle Methoden: **v** klein, **d** groß
agile Methoden: **d** klein, **v** groß

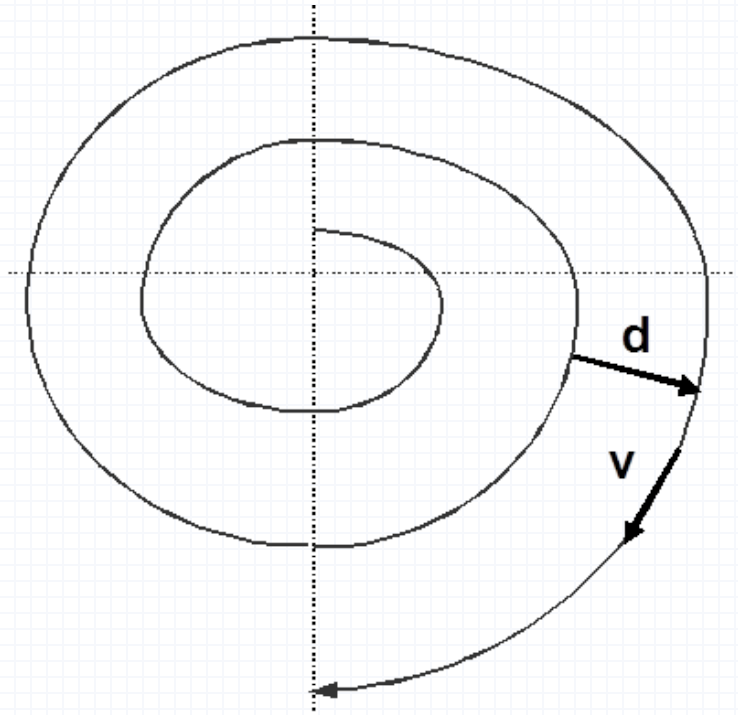
Gerade bei agilem Vorgehen Anvisieren eines beweglichen Ziels – Fortbewegung in raschen, kurzen Schritten nötig!

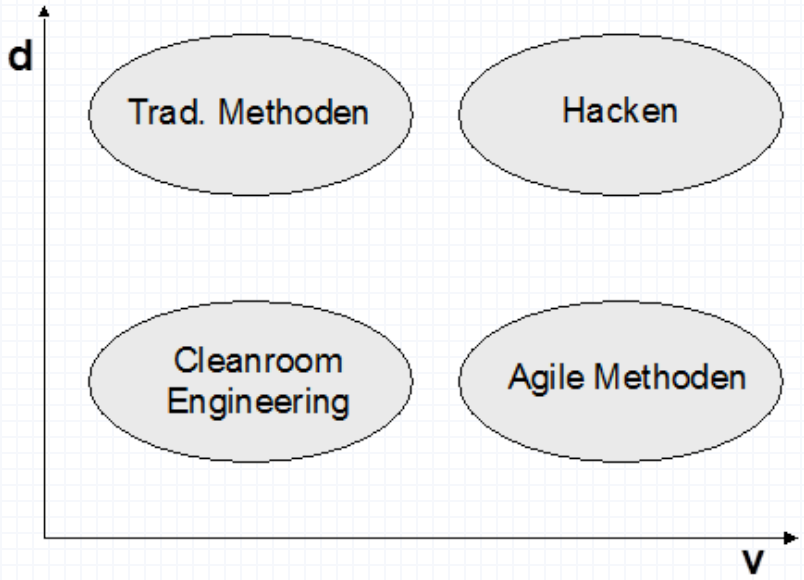
Prozessverständnis

Regelmäßiges Feedback durch iterativ-inkrementelles Vorgehen:

- kurze **Iterationszyklen**
- häufiges **Feedback** vom Auftraggeber
- genauere Ermittlung des **Projektstatus**
- rasche **Adaptierbarkeit**

Durch laufende Validierung auch klare Abgrenzung vom Hackertum!





Prozessökonomie: Traditionelles Vorgehen

Traditionelle Vorgehensmethoden:

- Vorhersehbarkeit
- Wiederholbarkeit
- Optimierbarkeit
- Streben nach vollständigen Anforderungen
- Plan- und Modell-fokussiert

Risiken:

- Anforderungen ändern sich.
- Aktualisierung der Pläne und Modelle ist sehr aufwändig.

(Bild: hdwallpapersinn.com)

Prozessökonomie: Agiles Vorgehen

Agile Vorgehensmethoden:

(AgileAlliance.org, adaptiert durch J. Coldewey)

Agiles Manifest

„*Kollaboration* statt *Formalismus*,
kurze Inkremente statt *jahrelanger Meditation*,
Flexibilität statt *Planungsorgien*,
Einbindung des Kunden statt *Absicherung*."

- Reduktion von Plänen und Modellen auf das Mindestmaß
- (ursprünglich) keine eigene explizite Designphase
- (ursprünglich) keine Prozessverbesserung

Prozessverbesserung

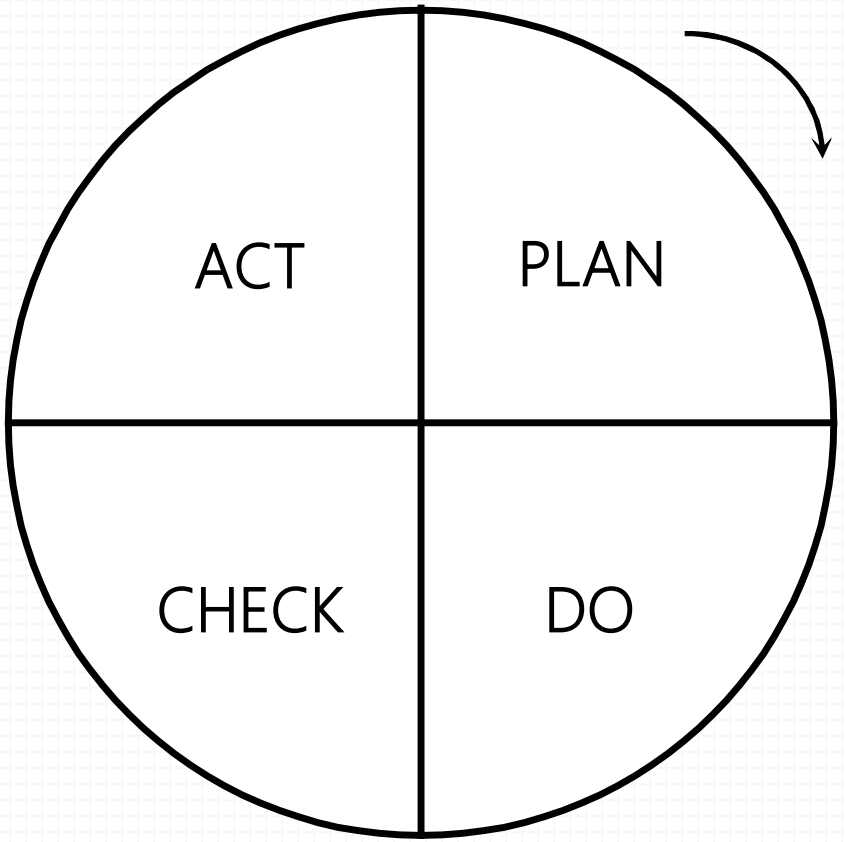
Gemeinsamkeiten aktueller Vorgehensmethoden:

- Projektziel ist **Vision**, die im Auge behalten werden muss.
- **Kunde** ist laufend eingebunden.
- Realisierung wird nach Festlegung minimaler **Mindestanforderungen** begonnen.
- **Testfälle** werden bereits mit der Anforderung erstellt (vor der Implementierung!).
- **Entwicklung** erfolgt **iterativ-inkrementell** mit kurzen Zyklen.
- Laufende **Risikobehandlung** ist wichtig.
- Nur für Auftraggeber oder Auftragnehmer **essenzielle Dokumente** werden erstellt.
- **Anforderungsänderungen** sind alltäglich und eingeplant.
- **Pläne** werden laufend geprüft und **adaptiert**.
- **Design** ist **einfach**, aber leicht erweiterbar.
- **Tests** werden möglichst **automatisiert** durchgeführt.

(Bild: hundeschule-pfotenteam.de)

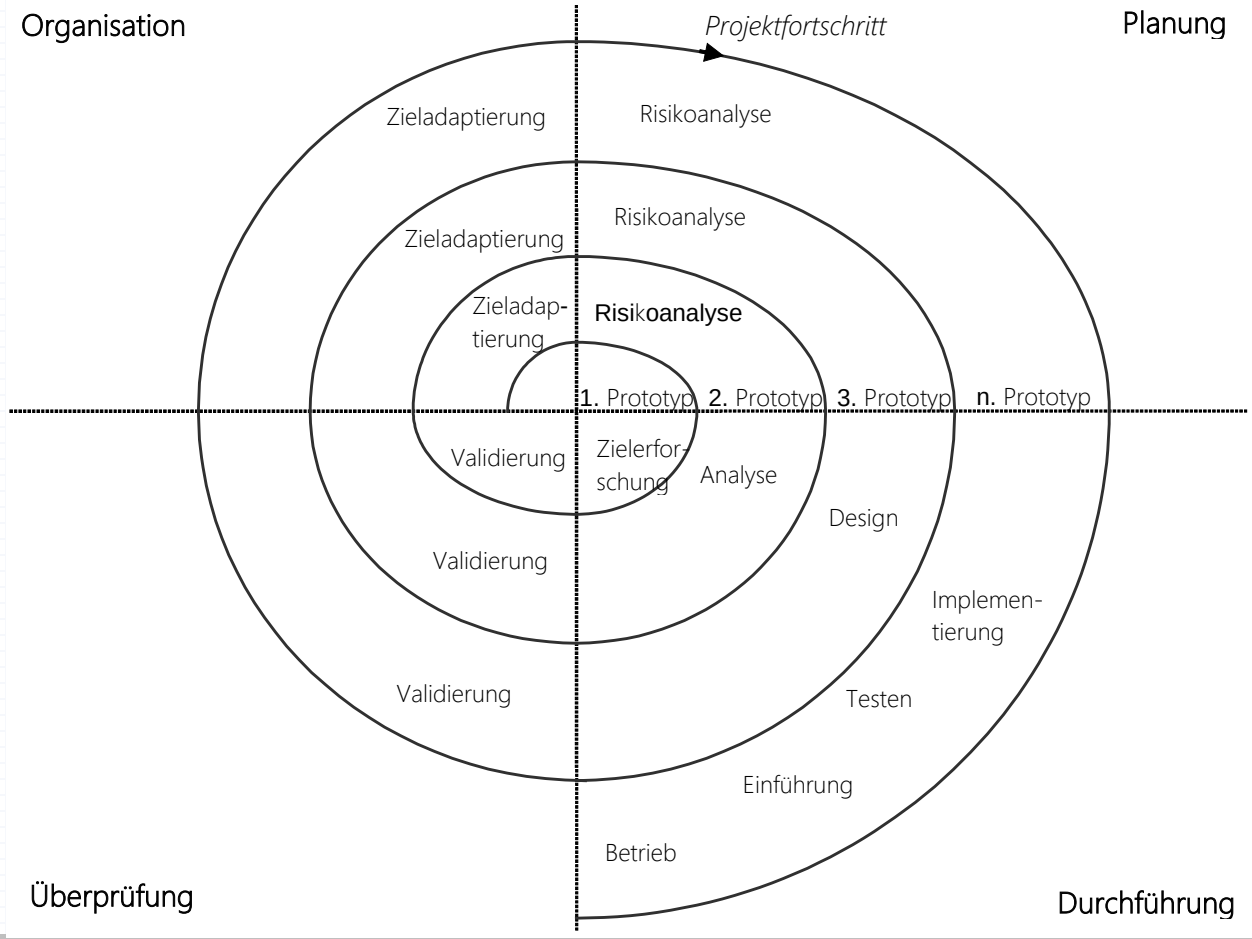
Iteratives Vorgehen: Kaizen

Idee der kontinuierlichen Verbesserung [Deming]



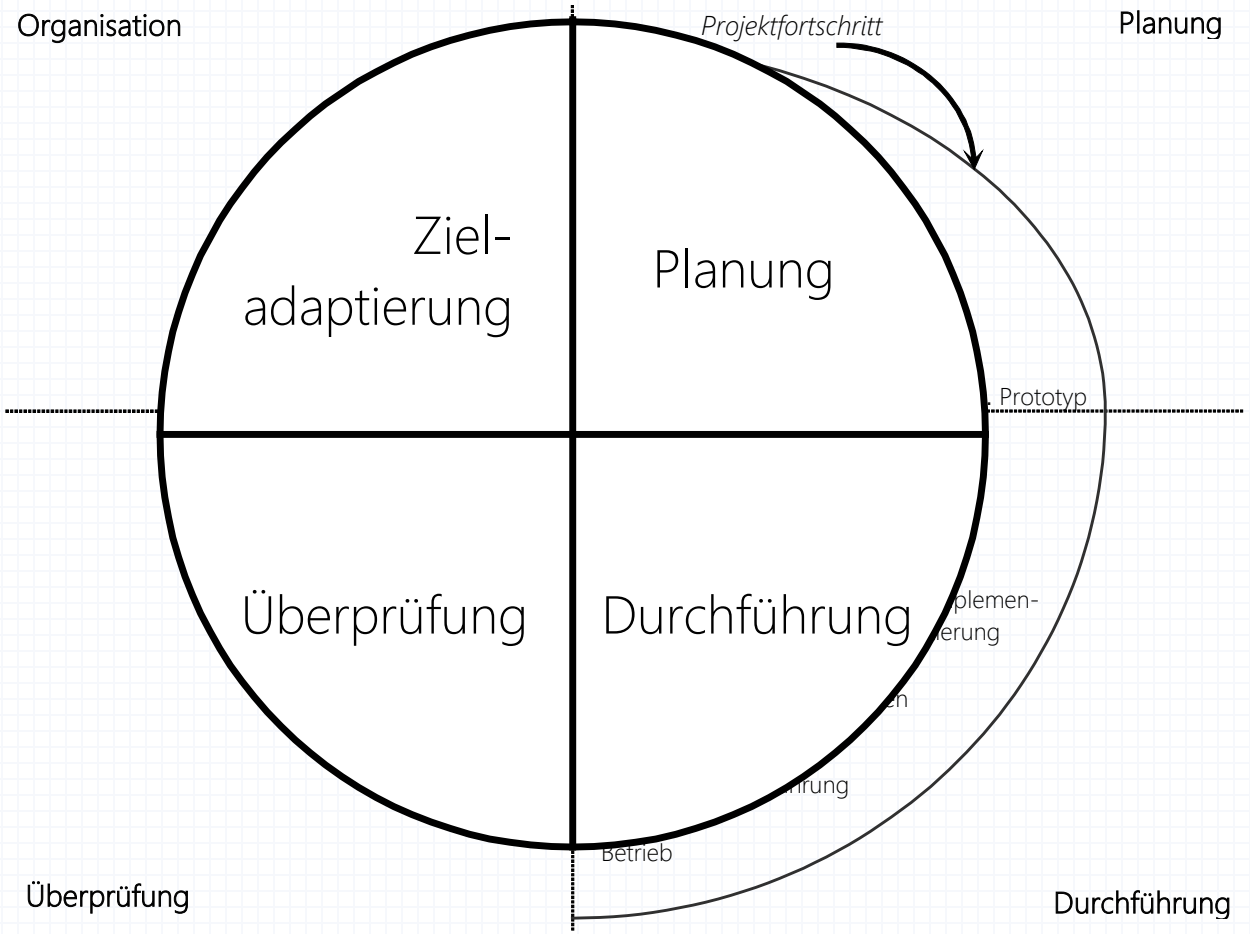
Iteratives Vorgehen: Spiralmodell

Einbringen von Prototyping-Konzepten [Boehm]



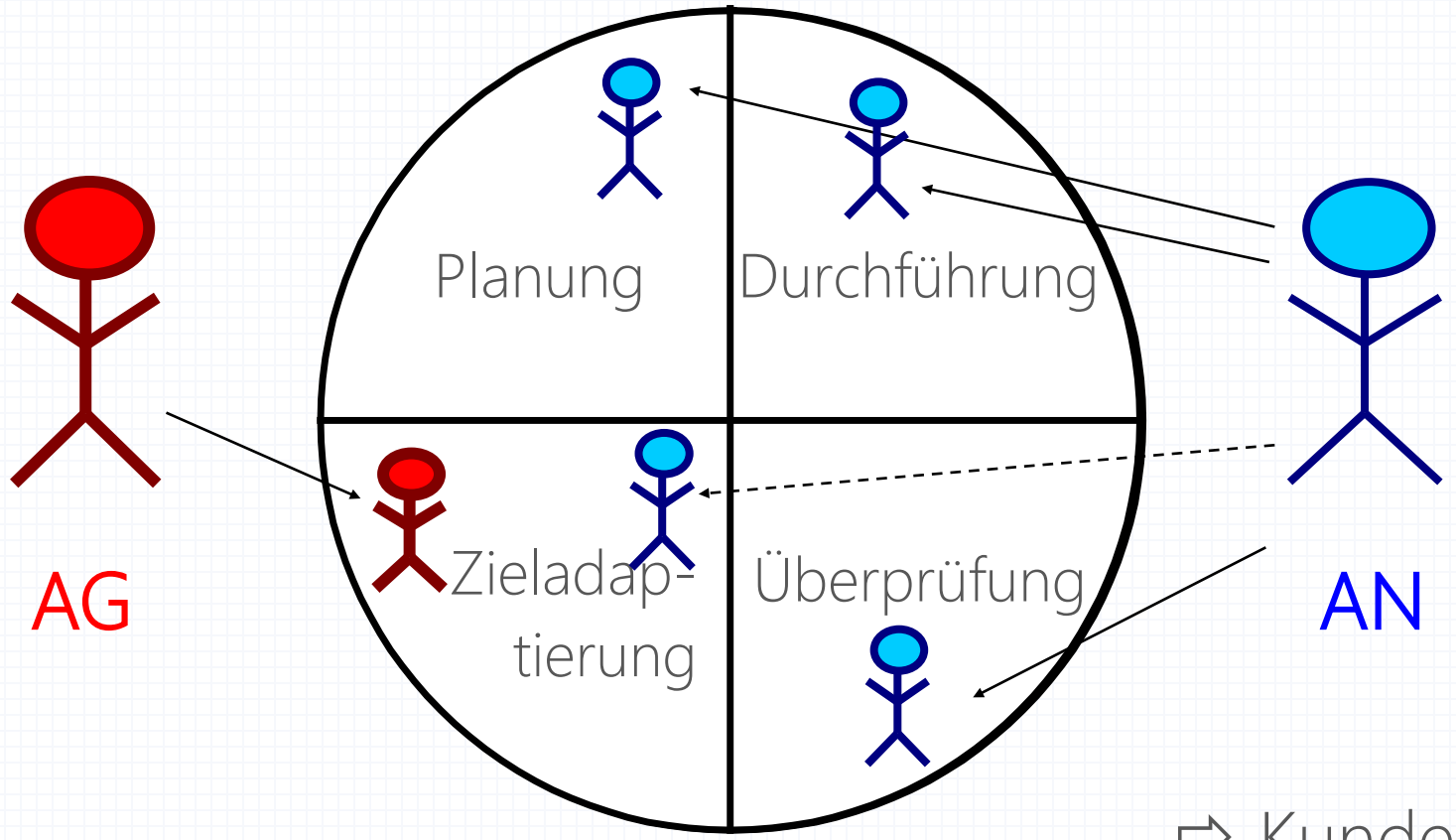
Iteratives Vorgehen: Spiralmodell

Einbringen von Prototyping-Konzepten [Boehm]



Schritt 1: Iterative Entwicklung

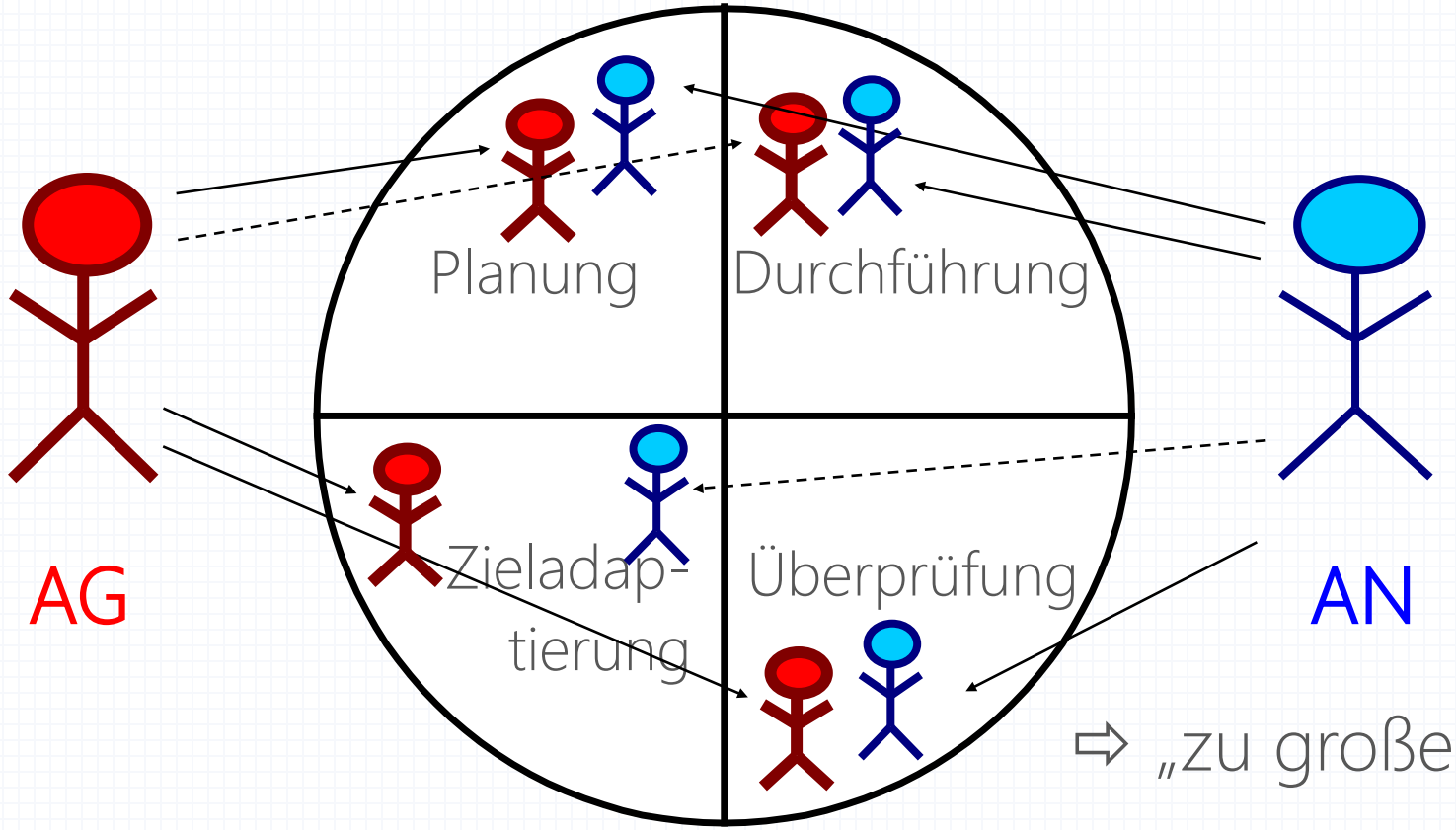
„klassisches“ Spiralmodell



⇨ Kundenferne!

Schritt 2: Kunden-integrierte Entwicklung

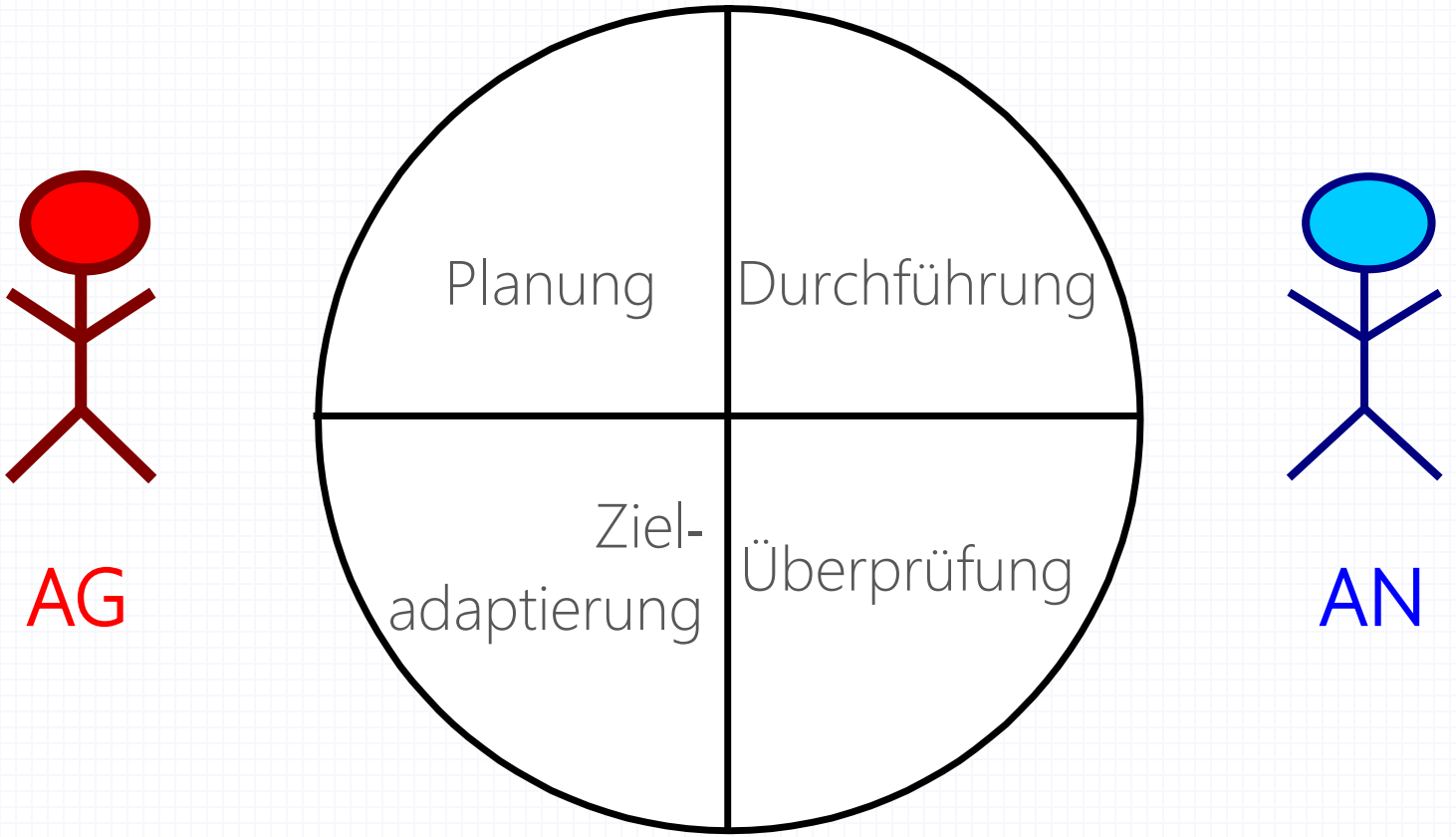
„agil“ (XP-Hype)



⇨ „zu große“ Kundennähe, Einmischung!

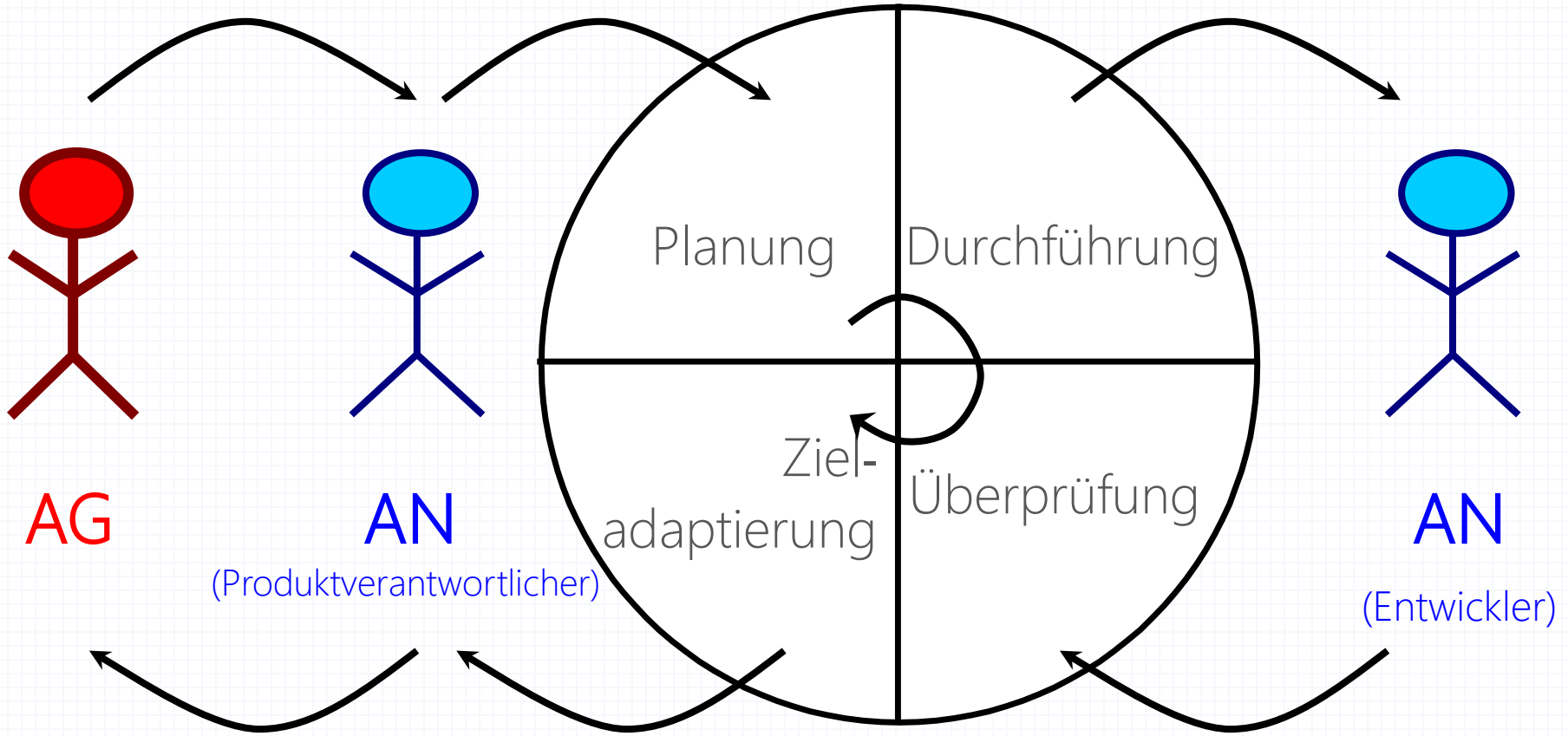
Schritt 3: Iterativ-inkrementelle Entwicklung

„ideale“ Kundennähe



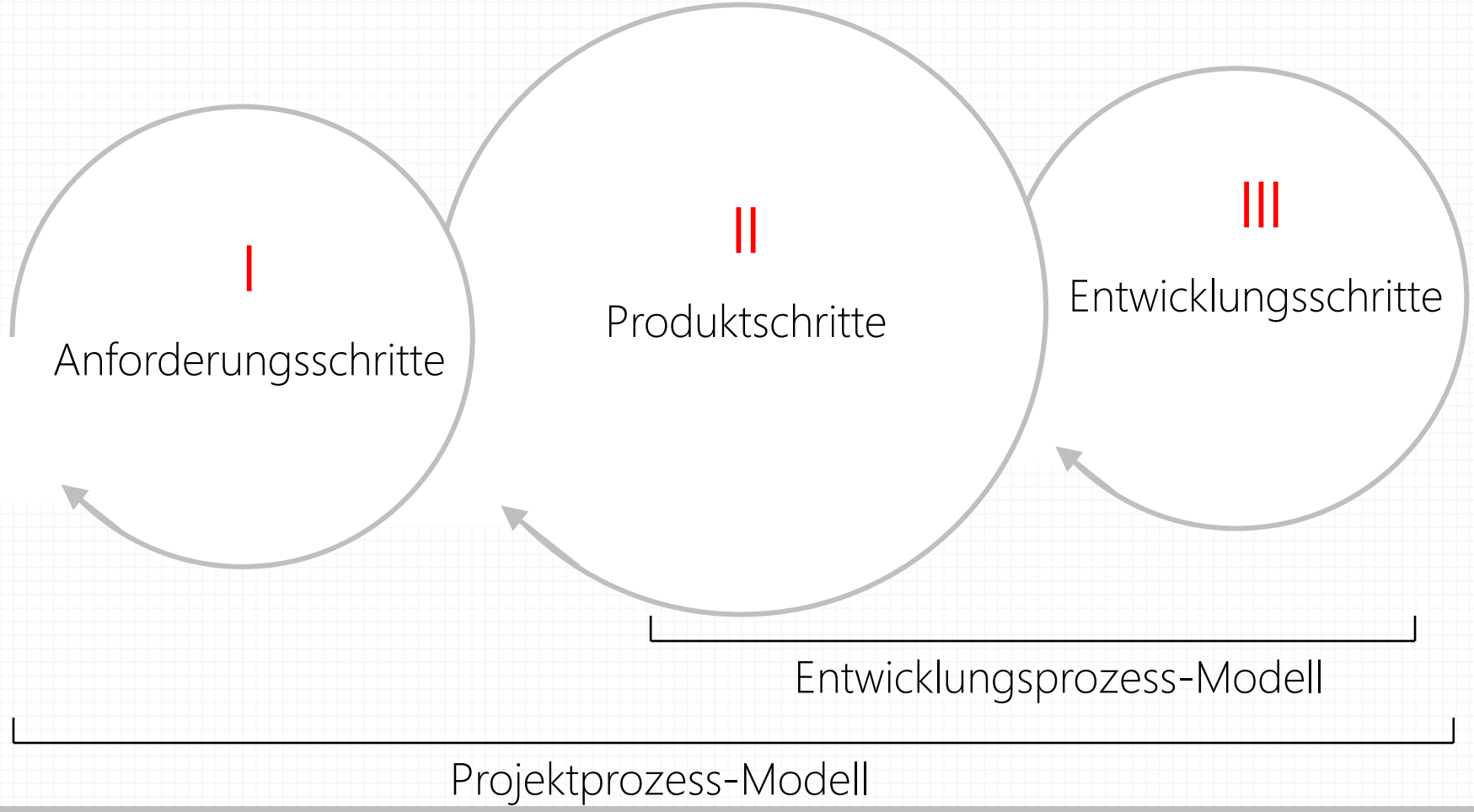
Schritt 3: Iterativ-inkrementelle Entwicklung

„ideale“ Kundennähe



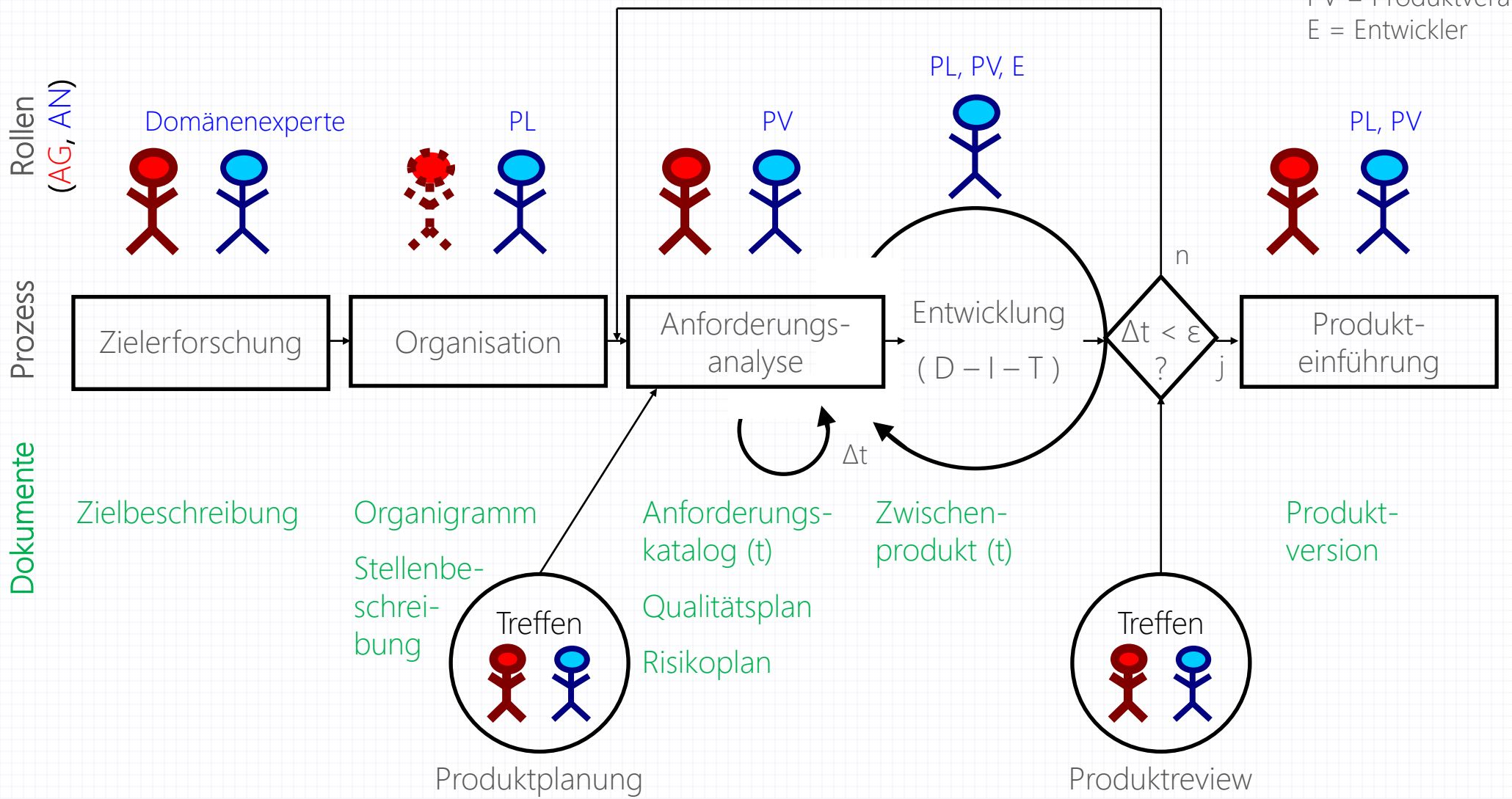
Iterativ-inkrementeller Entwicklungsprozess

dreifach rückgekoppelter Prozess:

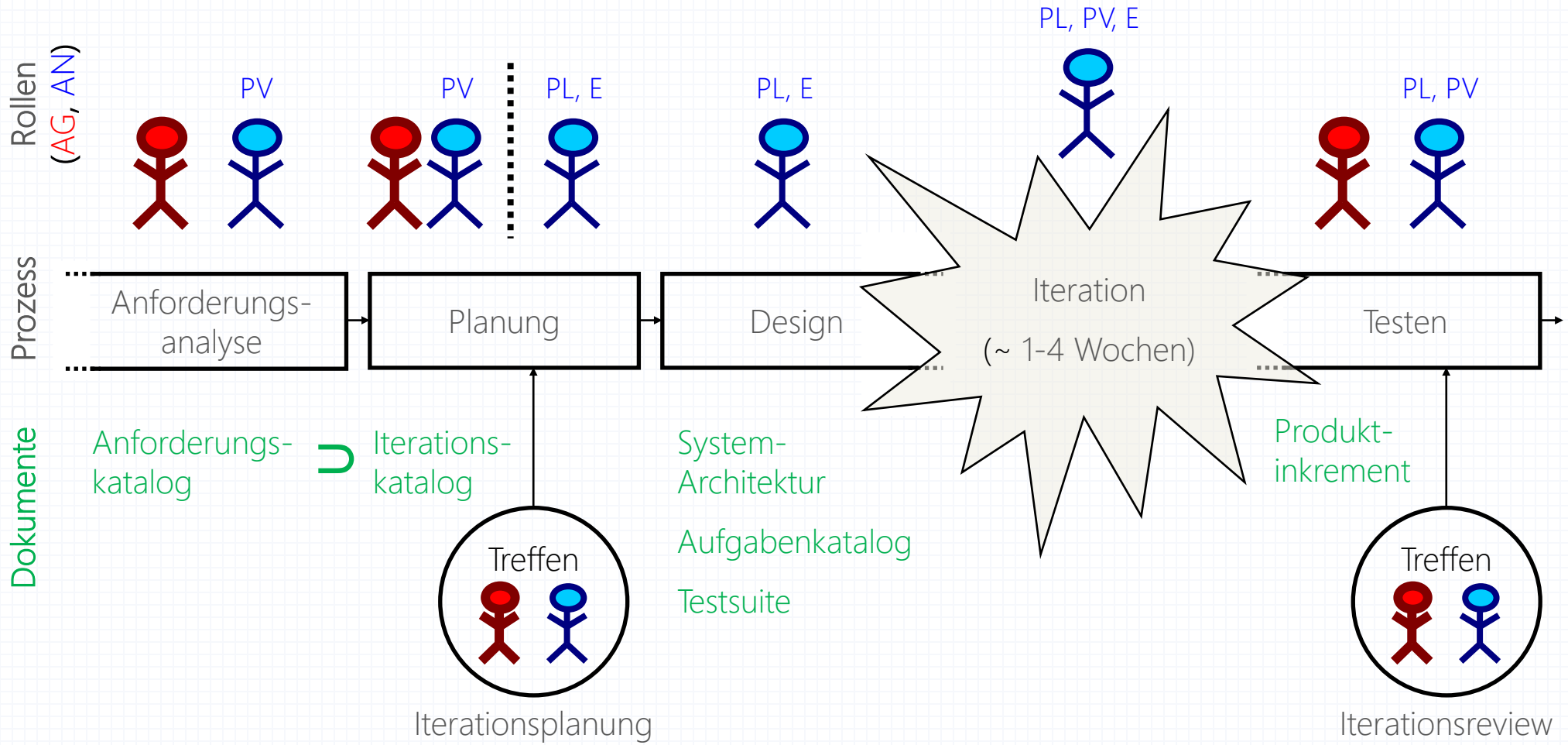


Einbettung in den Entwicklungszyklus (I)

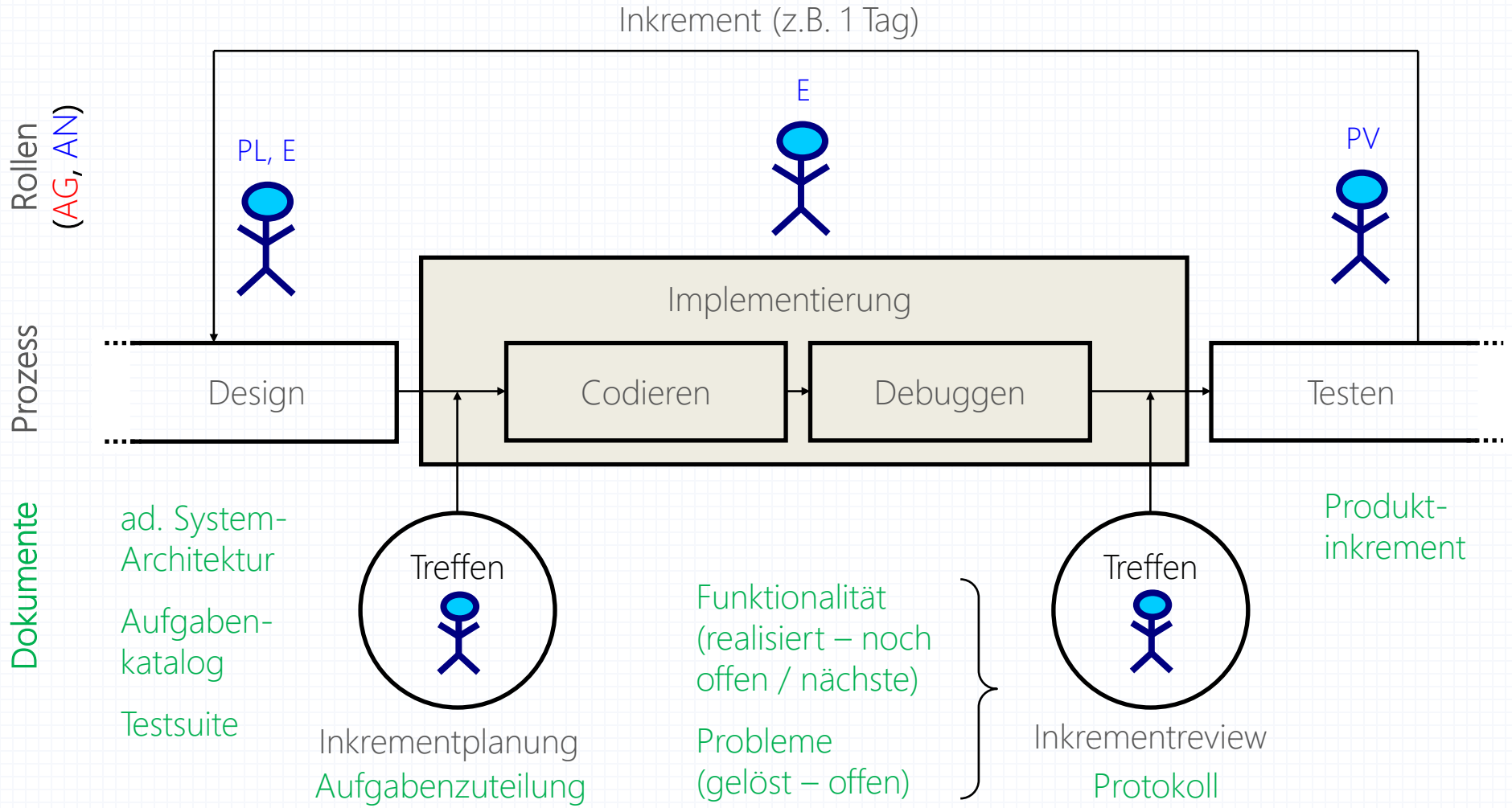
AG = Auftraggeber
AN = Auftragnehmer
PL = Projektleiter
PV = Produktverantwortlicher
E = Entwickler



Einbettung in den Entwicklungszyklus (II)



Einbettung in den Entwicklungszyklus (III)



SOFTWARE-ENTWICKLUNGSPROZESSE
Moderne Prozesse zur Softwareentwicklung

Herwig Mayr

Fakultät für Informatik, Kommunikation und Medien
Fachhochschule OÖ, Hagenberg