Qualitätssoftware durch Projektmanagement – Motivation am Beispiel MPG

Herwig Mayr

Fakultät für Informatik/Kommunikation/Medien

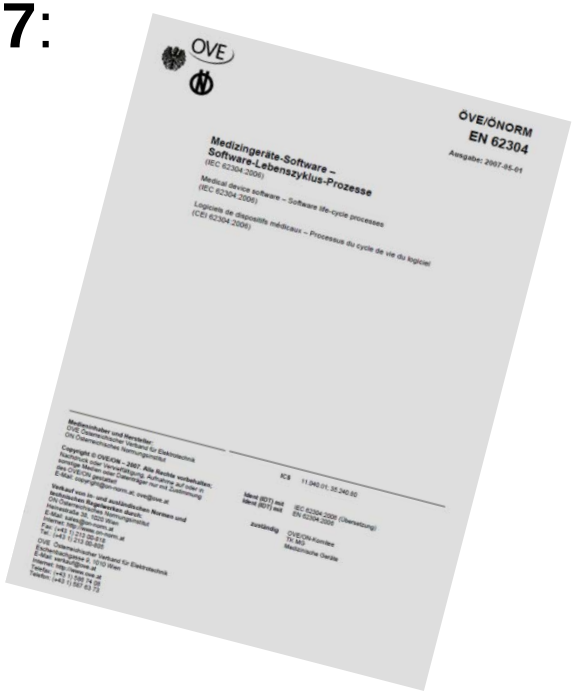
FH OÖ Campus Hagenberg

Ziel

Zuverlässige Software agil entwickeln

Medizinproduktegesetz (MPG) I

IEC 62304:2006, ÖVE/ÖNORM EN 62304:2007:



Medizingeräte-Software – Software-Lebenszyklus-Prozesse

ausgegeben (deutsch) am 1.5.2007,
in Kraft getreten am 21.3.2010

abgestimmt mit anderen Normen
(z.B. ISO 90003, ISO 12207, ISO 14971)

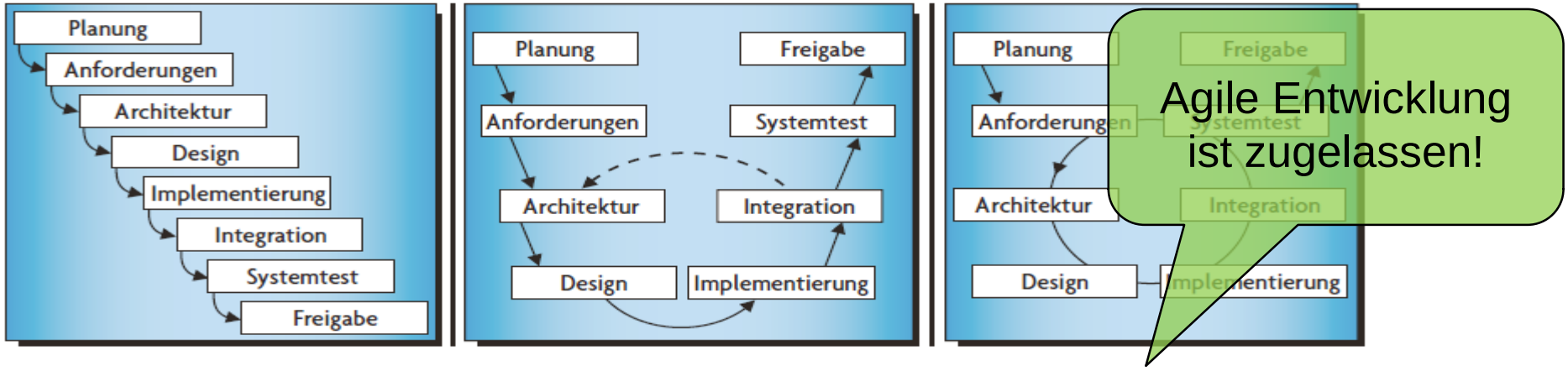
„…eine Zusammenstellung von **Prozessen**, **Aktivitäten** und **Aufgaben**, die … einen allgemeinen Rahmen für **Lebenszyklus-Prozesse** von **Medizinprodukte-Software** fest[legen].“ [IEC 62304, 1.1 Zweck]

Medizinproduktegesetz (MPG) II

geforderte Prozesse:

- Softwareentwicklungs-Prozess („Problemlösungsprozess “)
- Softwarewartungs-Prozess
- Qualitätsmanagement-Prozess
- Risikomanagement-Prozess
- Konfigurationsmanagement-Prozess

erlaubt **verschiedene Entwicklungsmodell-Strategien:**



Wasserfall | inkrementell | evolutionär

Motivation für QM, RM und KM

Agile Entwicklung zuverlässiger Software am Beispiel Medizinproduktegesetz

erfordert verstärktes
Qualitätsmanagement,
Risikomanagement,
Konfigurationsmanagement

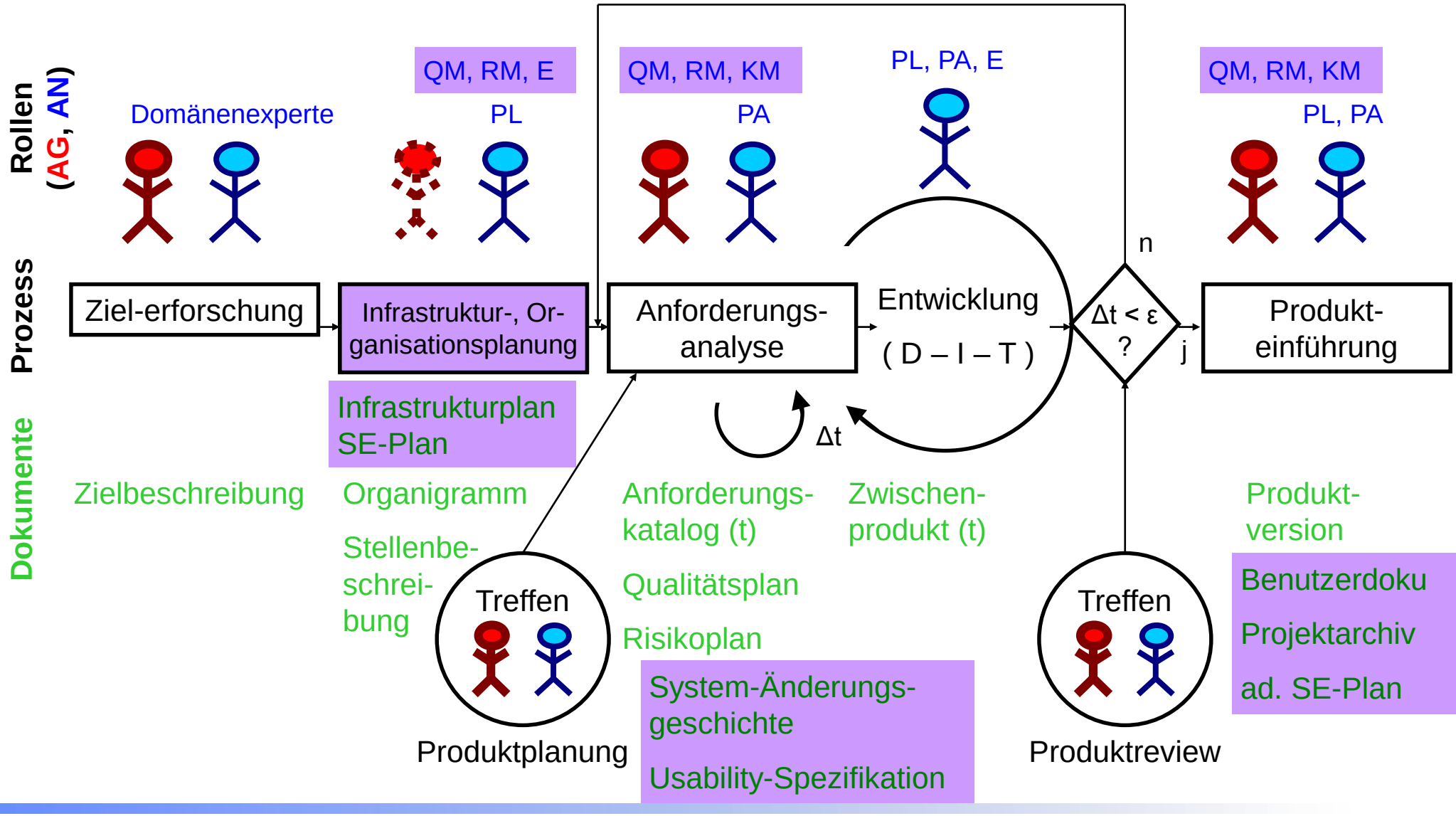
Gap-Analyse Agiles Vorgehen – MPG

Folgende Aktivitäten sind im Agilen Vorgehen nicht (explizit) vorgesehen:

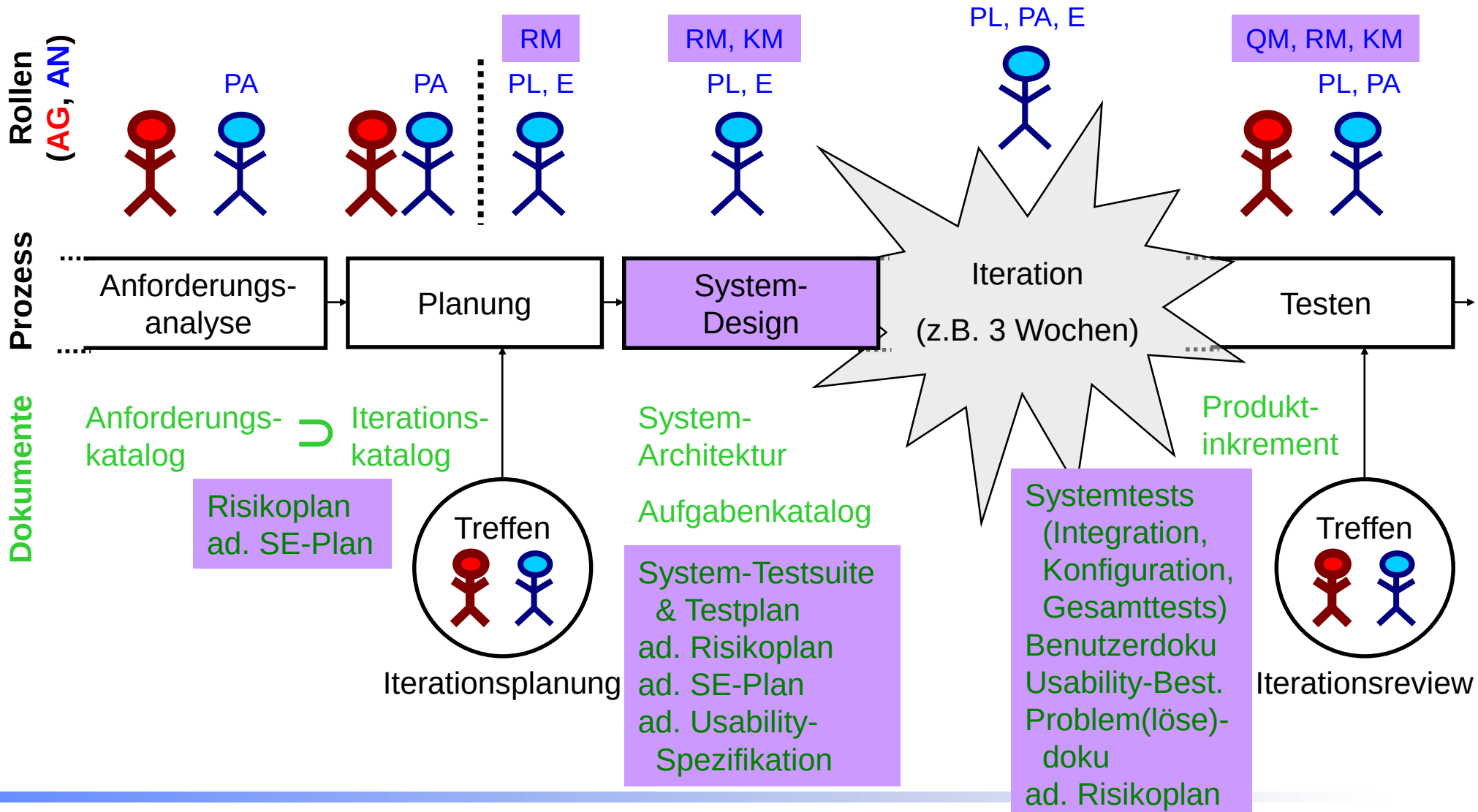
- explizites Systemdesign (Big Picture, Architektur)
- Detaildesign für die Implementierung
- begleitendes Risikomanagement
- umfassende Systemtests (auf Funktionalitätsebene)
- expliziter (dokumentierter) Problemlösungsprozess
- begleitendes Qualitätsmanagement
- bestimmte, für das MPG nötige Rollen

Aber: Keine dieser Aktivitäten ist im Agilen Vorgehen verboten!

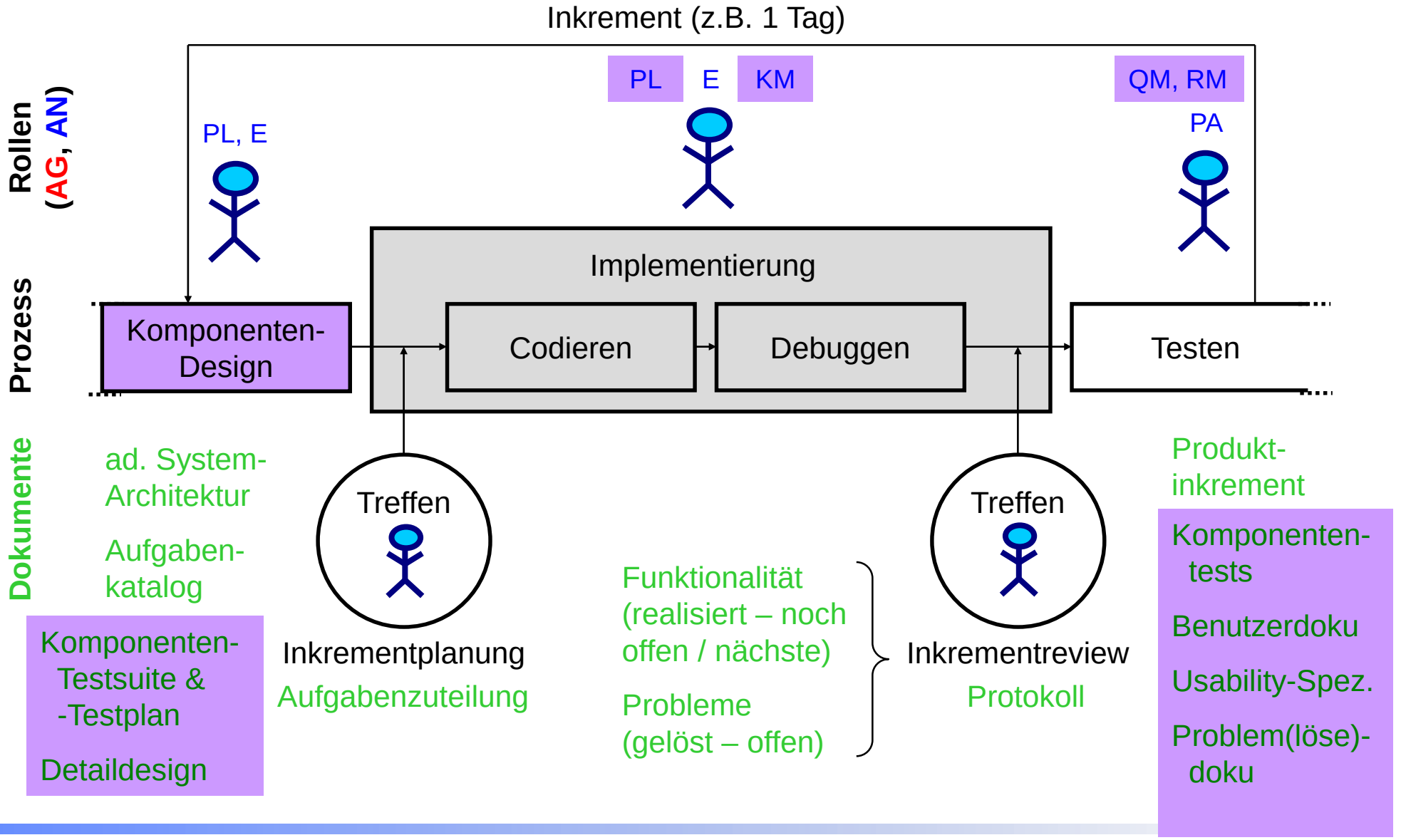
“Zuverlässiger, agiler” Entwicklungszyklus I



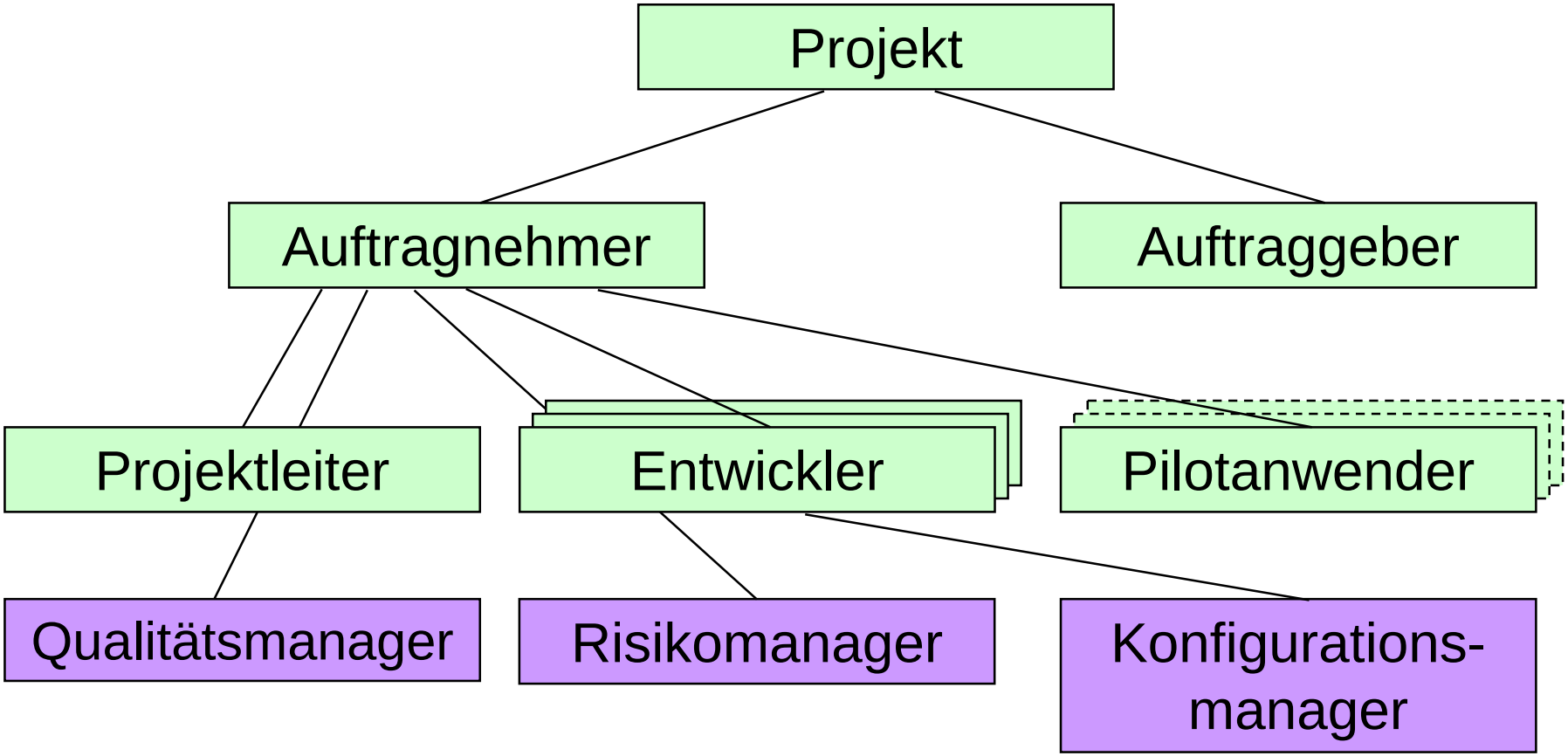
“Zuverlässiger, agiler” Entwicklungszyklus II



“Zuverlässiger, agiler” Entwicklungszyklus III



Erweiterung der Rollen



Aufgabenverteilung QM / RM / KM

Qualitätsmanager: macht Entwicklungsprozess bewusst,
def. Verfahren für Verifikation & Validierung,
plant & prüft Gebrauchstauglichkeit,
ist Ansprechpartner der Benannten Stelle.

Risikomanager: analysiert **Produkt** & identifiziert Risiken,
legt Behandlungsmaßnahmen fest & überprüft,
bewertet Restrisiken & Kunden-Feedback.

Konfigurations-manager: organisiert und verantwortet Konfig.mgmt.,
plant Build-Prozess und Rollout,
koordiniert die Erstellung von System- und Benutzerdokumentation.

Alle kennen die relevanten Normen und halten sich daran!

Motivation für ALM & SLM

Agile Entwicklung zuverlässiger Software am Beispiel Medizinproduktegesetz

erfordert verstärktes
Produktlebenszyklus-Management
und
Servicelevel-Management

Validierung von Medizinprodukten

IEC/EN 60601-1:2005, IEC/EN 60601-1-4:1996 Amendment 1:1999, IEC/EN 60601-1-6:2004, IEC/EN 62366:2006/7

IEC 60601: Medizinische elektrische Geräte: Allgemeine Festlegungen für die Sicherheit

Teil 1: …einschließlich der wesentlichen Leistungsmerkmale

Teil 1-4: Ergänzungsnorm: Programmierbare elektrische medizinische Systeme

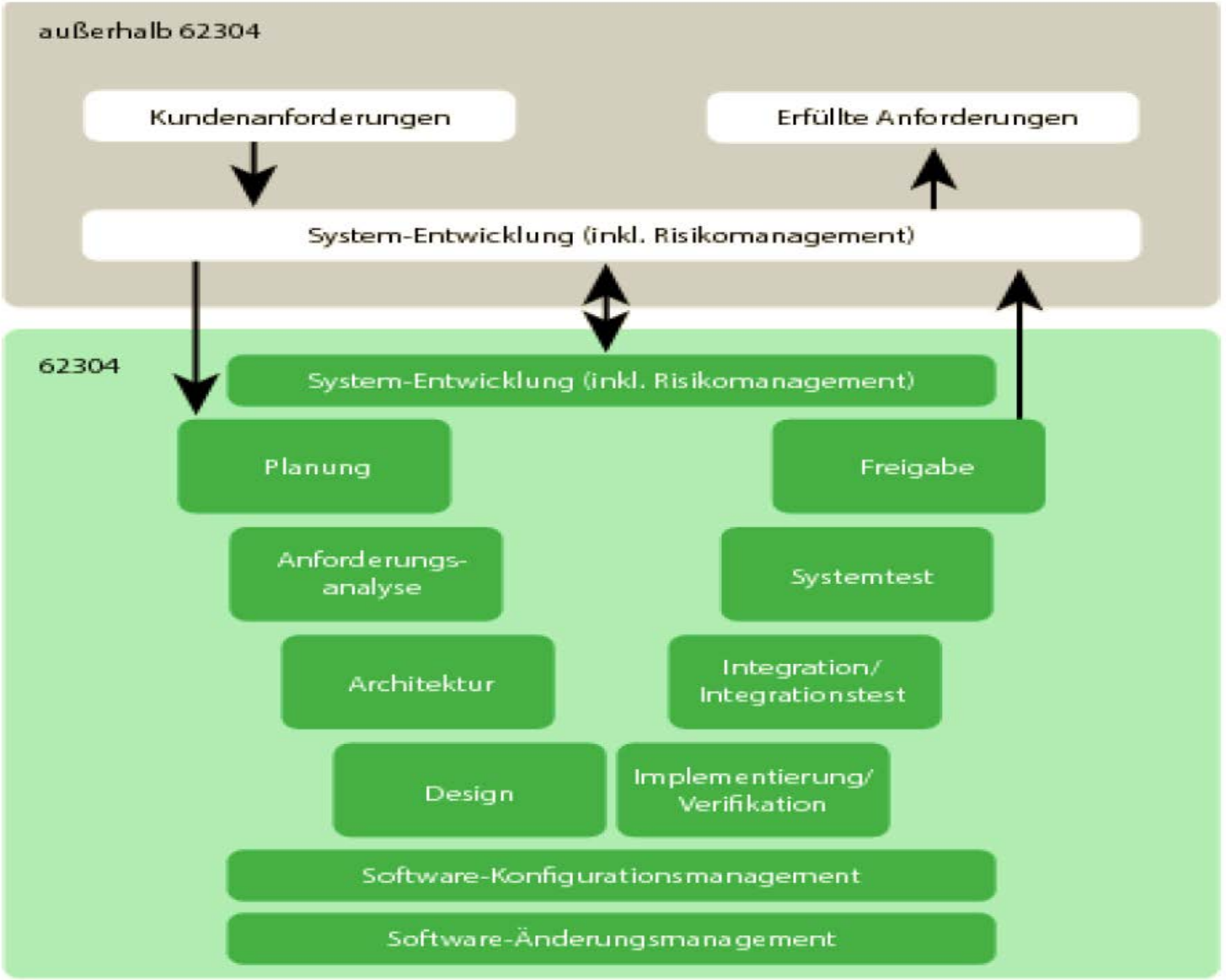
Teil 1-6: Ergänzungsnorm: Gebrauchstauglichkeit

IEC 62366: Medizinprodukte – Anwendung der Gebrauchstauglichkeit auf Medizinprodukte


ÖVE/ÖNORM
EN 62366

„…beschreibt einen **Gebrauchstauglichkeits-orientierten Entwicklungsprozess** und leitet zur Einrichtung und Durchführung dieses **Prozesses** an, um die **Sicherheit** von **Medizinprodukten** sicherzustellen.“ [IEC 62366, Einleitung]

Verbindung zur Validierung



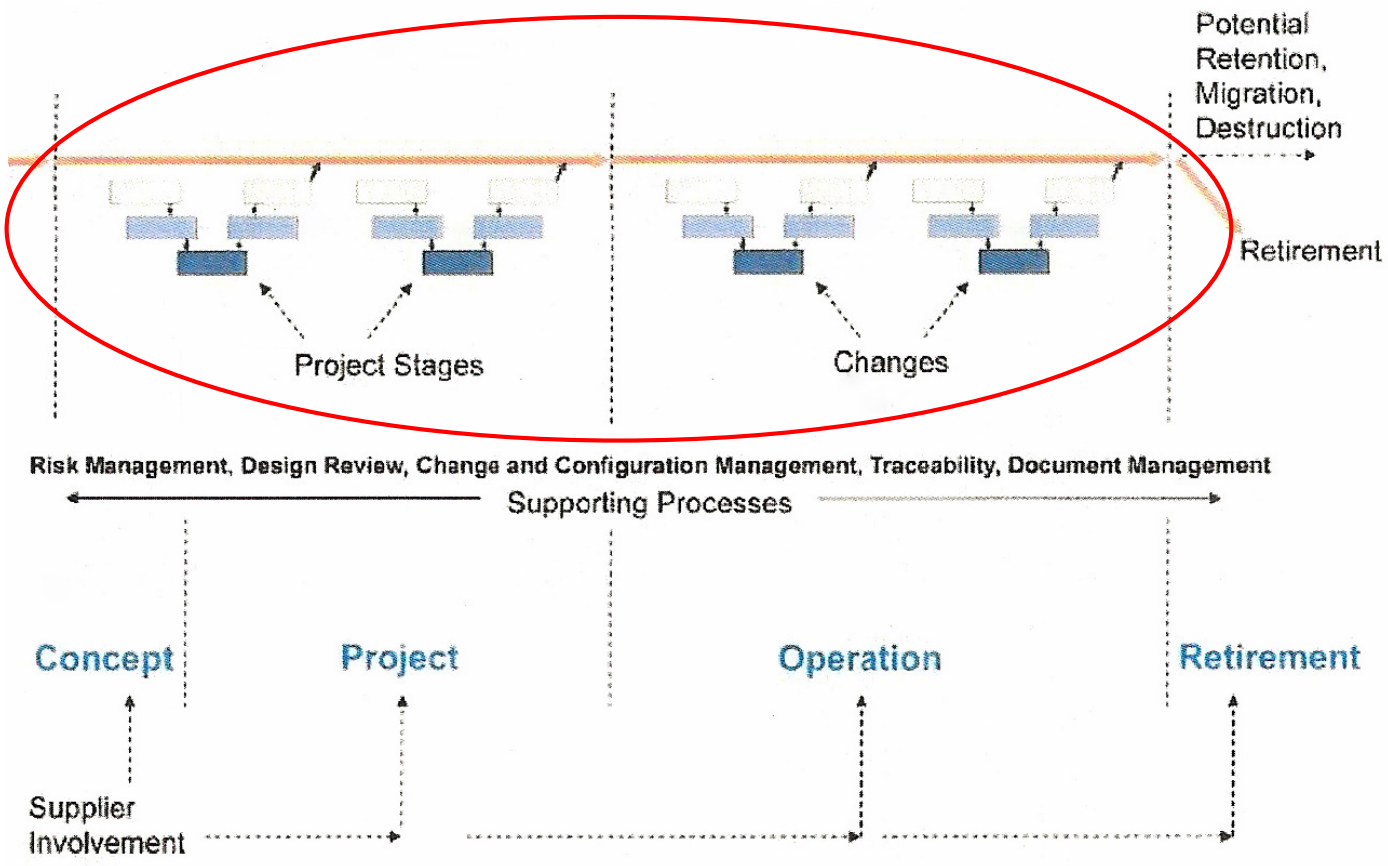
Iterativ-inkrementelles Vorgehen und GAMP 5

Leitfaden: „Good Automated Manufacturing Practice v.5“ (2008)

Phasen „Project“ und „Operation“ verlieren klare Trennung!

Schwerpunkte:

- Betriebsphase
- Migration, Produkt-lebenszyklus
- Risikomanage-ment
- Änderungs-management



Validierung im Rahmen der Usability

Usability (Gebrauchstauglichkeit) gemäß IEC 62366:

- Wer setzt das Produkt ein?
- Wo wird es eingesetzt?
- Wie/wozu wird es verwendet?

Dabei Analyse der **Hauptfunktionen** (als Blackbox) und

Sicherheitsfunktionen (Prüfung unter ungünstigsten Bedingungen)!

Verifikation: feststellen ob die zu Beginn aufgestellten Anforderungen erfüllt werden (IEC 60601).

Validierung: feststellen ob das Ergebnis den beabsichtigten Zweck erfüllt (IEC 60601).

Im sicherheitskritischen Bereich deutlich formaler!

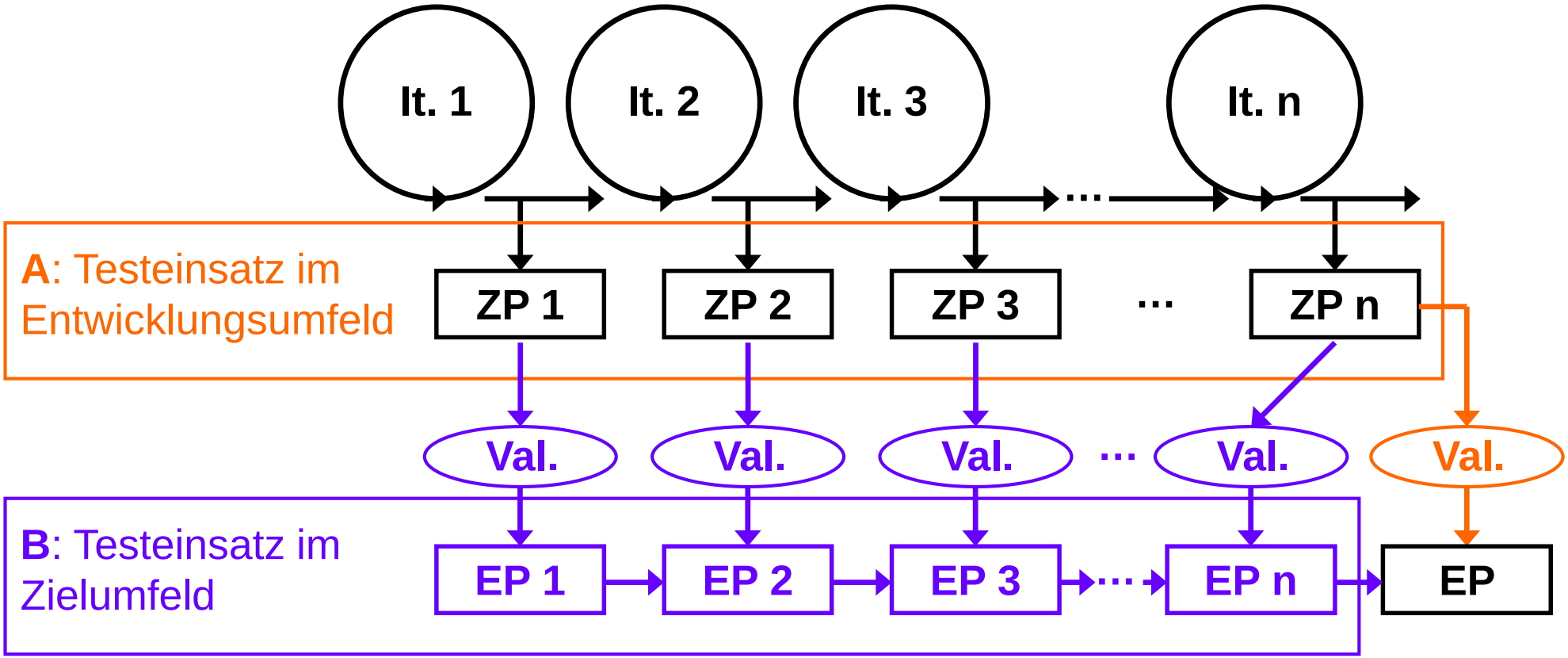
Prozessuale Einbindung der Validierung I

Einbindung in den Prozess ist sinnvoll:

- Validierung an sich **außerhalb der IEC 62304** (auf Produktebene über Produktnormen geregelt).
- Aber eine **retrospektive Validierung** ist meist **teurer** (bis hin zum Einstampfen des Produkts!) als eine **prospektive Validierung.**
- Eine Validierung ist daher am sinnvollsten, wenn sie den Entwicklungsprozess **begleitet** (Validierungsplan!).
- Die **iterativ-inkrementelle Entwicklung** ist **sehr gut** für eine **begleitende Validierung** geeignet!

Prozessuale Einbindung der Validierung II

Zwei Alternativen:



A: < Aufwand, > Val.-Risiko, > Time-to-Market (Kundenferne!)

B: > Kosten, < Val.-Risiko, < Time-to-Market (Kundennähe!)

Zusammenfassung

Agiles Vorgehen bei der Entwicklung von zuverlässiger Software (geregelt durch Normen) ist möglich!

- Die Möglichkeit, auf (Anforderungs-)Änderungen zu reagieren, erlaubt eine **zeitgemäße Softwareentwicklung**.
- Im iterativ-inkrementellen Vorgehen sind nur **Erweiterung der Rollen und der Dokumente** nötig (dann noch agil?).
- **Planung & Reviews** des Modells sind einfach **für RM** und **QM erweiterbar**.
- Ein **umfangreicher Werkzeugeinsatz** (Anforderungen, Tests, …; **KM**) ist essenziell.
- Die laufende Einbindung des Auftraggebers ermöglicht eine **begleitende Validierung**.