

- 01 Einführung
- 02 Prozessmodelle
 - 02.1 Softwarelebenszyklus
 - 02.2 Basis-Vorgehensmodelle
 - 02.3 Monumentale Vorgehensmodelle
 - 02.4 Agile Vorgehensmodelle
- 03 Konfigurationsmanagement
- 04 Requirements Engineering
- 05 Modellierung
- 06 Qualitätsmanagement

- Softwarelebenszyklus
 - Schritte („Phasen“), die innerhalb eines Softwareentwicklungsprojekts durchlaufen werden
- Zentrale Frage:
 - Wie kommt man von diesen allgemeinen Schritten zu einem konkreten Vorgehen in einem Projekt?



Was ist das Ziel jeder Softwareentwicklung?

02 Prozessmodelle / 02.2 Basis-Vorgehensmodelle

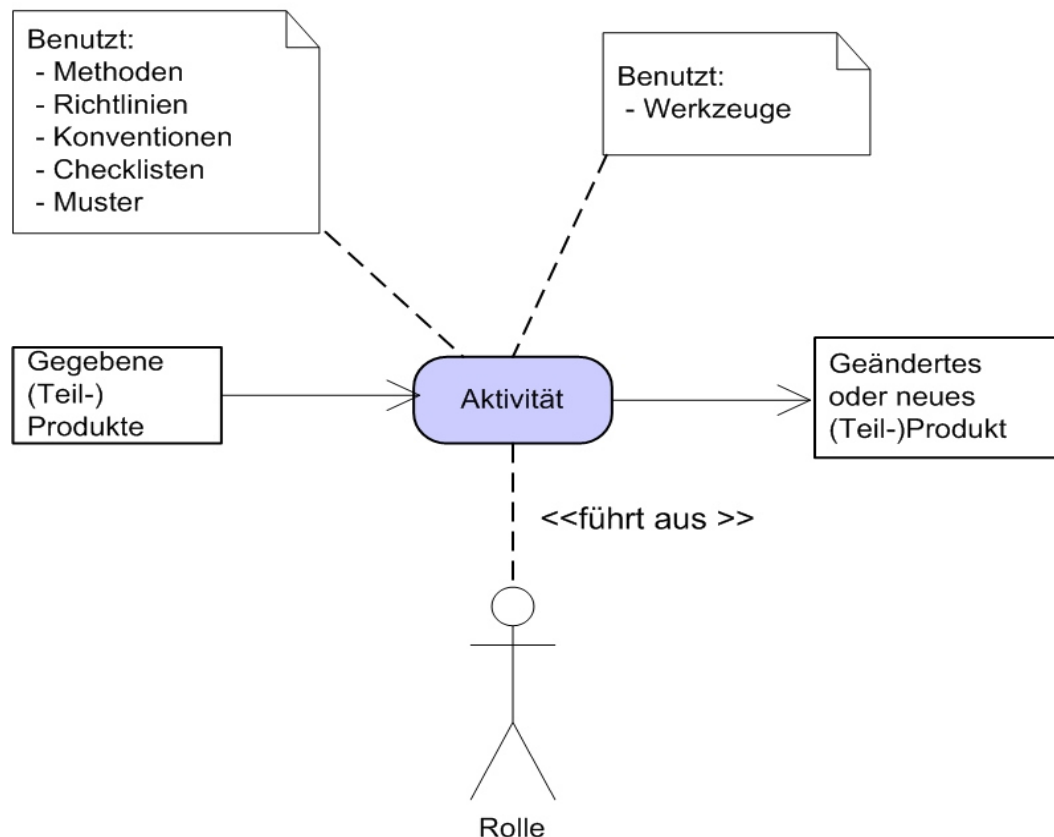
- Softwareprodukte **termin- und kostengerecht** in der **definierten Qualität** zu erstellen
 - Finales Produkt – Software-Lösung
 - Sämtliche Artefakte, die für die Herstellung dieser Lösung relevant sind
- Zusammenarbeit von *einigen wenigen* bis *sehr vielen* Menschen
 - Softwareerstellung muss **organisiert** werden
 - ➔ Festlegung des **organisatorischen Rahmen** erfolgt in **Prozessmodellen**
 - In der Praxis haben sich verschiedene **Vorgehensmodelle** etabliert



- **Strategie für die Projektdurchführung**
- **Organisatorischer Rahmen**, der festlegen sollte:
 - Reihenfolge des Arbeitsablaufs
 - Jeweils durchzuführende Aktivitäten
 - Definition der Teilprodukte (inkl. Layout und Inhalt)
 - Fertigstellungskriterien (Qualitätsniveau)
 - Notwendige Mitarbeiterqualifikationen
 - Verantwortlichkeiten und Kompetenzen
 - Anzuwendende Standards, Richtlinien, Methoden und Werkzeuge

Basiselement eines Softwareprozesses

02 Prozessmodelle / 02.2 Basis-Vorgehensmodelle



Quelle: in Anlehnung an Balzert 2009, S. 443

- Erste Modelle: Softwareentwicklung auf Projektebene
 - Grobgranular
 - Geben auf der Ebene von Phasen an, wie die Reihenfolge und Inhalte der Phasen aussehen soll
 - Basismodelle
- Weiterentwicklung
 - Monumentale / schwergewichtige Modelle
 - Umfangreiche und detaillierte „to-do-Modelle“
 - Gegenbewegung: Agile / leichtgewichtige Modelle

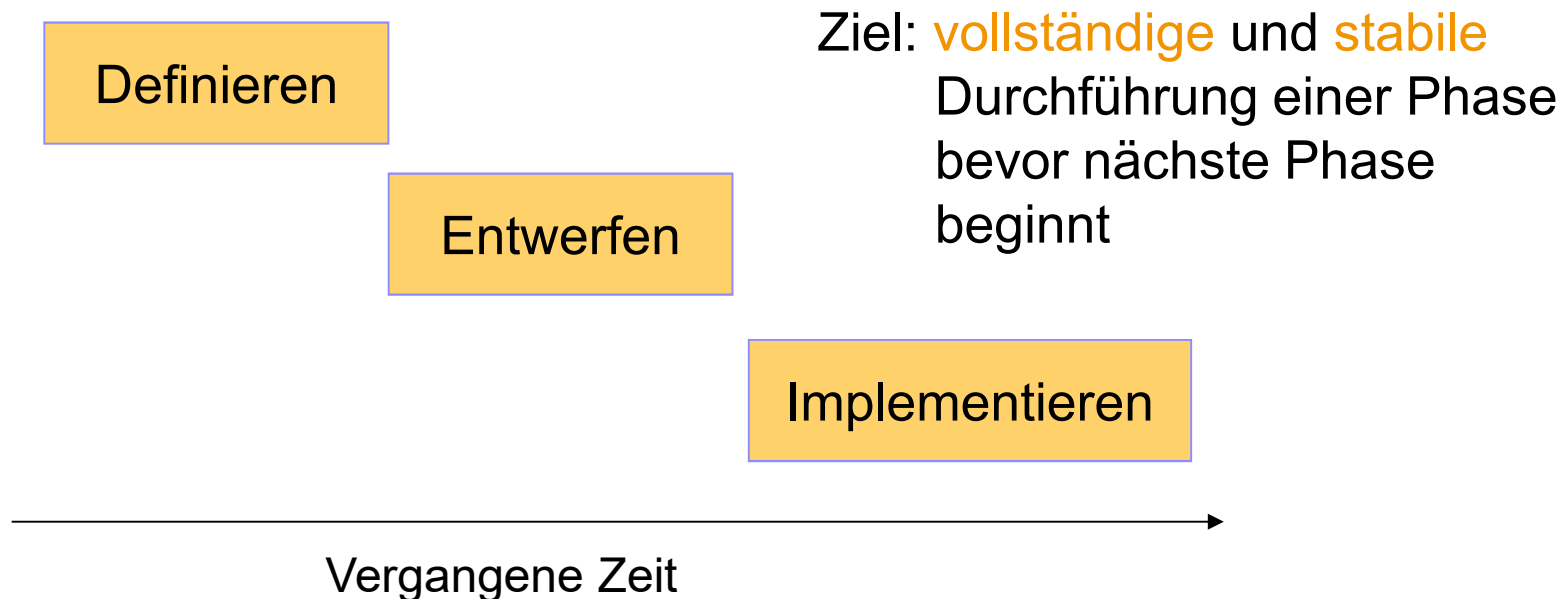


- Auswahl
 - Sequenzielles Modell
 - Nebenläufiges Modell
 - Inkrementelles Modell
 - Evolutionäres Modell
 - V-Modell
 - Spiralmodell

Sequenzielles Modell

02 Prozessmodelle / 02.2 Basis-Vorgehensmodelle

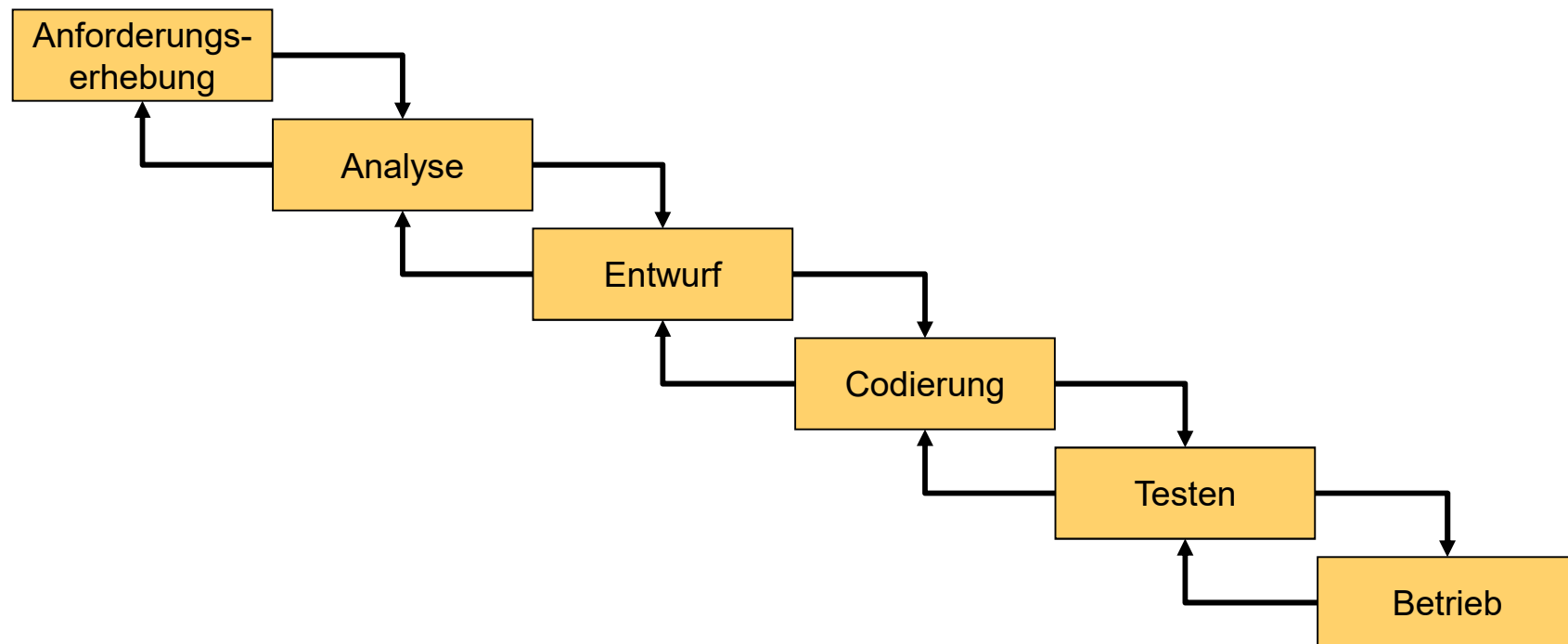
- Softwareentwicklung wird in Phasen gegliedert, die sequenziell hintereinander ablaufen
- Phase beginnt erst wenn Vorgängerphase vollständig abgeschlossen ist



Wasserfall-Modell (1)

02 Prozessmodelle / 02.2 Basis-Vorgehensmodelle

- Bekannteste Ausprägung
- Erweiterung um Rückkopplungsschleifen zwischen den Stufen



● Charakteristische Eigenschaften

- Jede Aktivität ist in der richtigen Reihenfolge und vollen Breite durchzuführen
- Am Ende jeder Aktivität steht ein fertiggestelltes Dokument
- Jede Aktivität muss beendet sein bevor die nächste beginnt
- Vorangegangene Phase muss vollständig abgeschlossen und freigegeben sein, bevor die nächste Phase gestartet werden kann
- Benutzerbeteiligung nur in der Definitionsphase vorgesehen – Entwurf und Implementierung erfolgen ohne Beteiligung der Benutzer bzw. Auftraggeber



Bewertung Wasserfall-Modell

02 Prozessmodelle / 02.2 Basis-Vorgehensmodelle

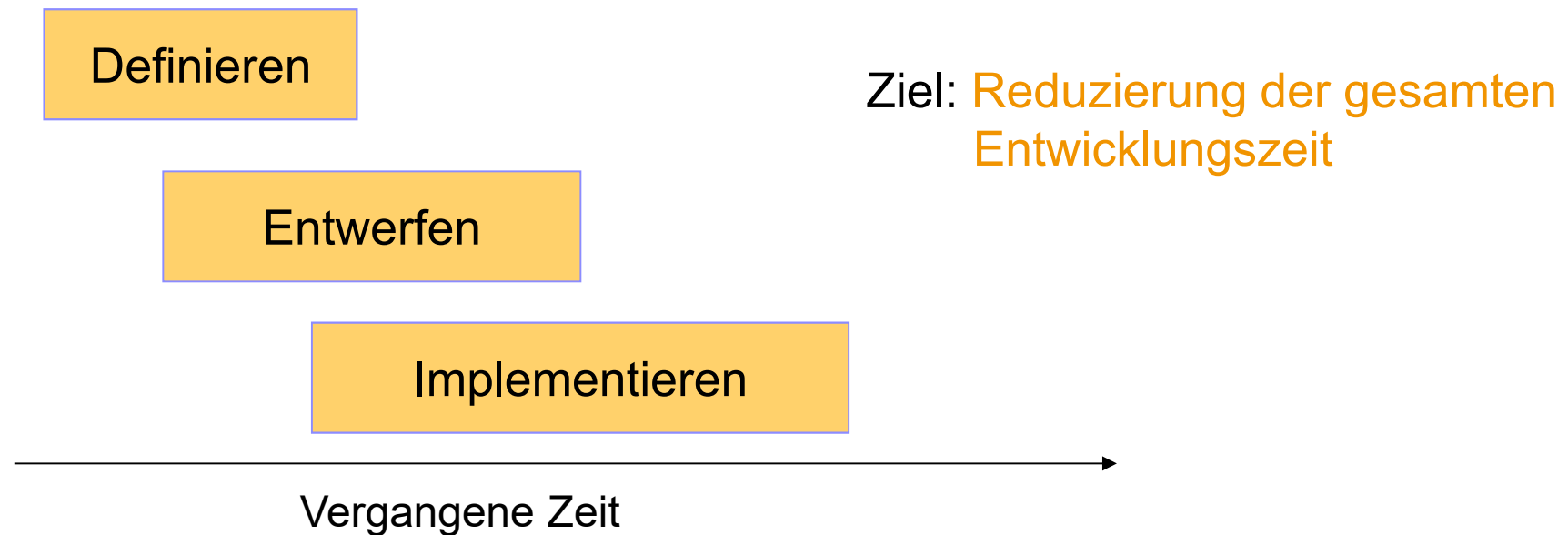
- 😊 Beitrag zu diszipliniertem und kontrollierbarem Prozessablauf
- 😊 Gute Planbarkeit durch klar abgegrenzte Phasen
- 😞 Nicht immer sinnvoll alle Entwicklungsschritte in voller Breite, vollständig und sequenziell durchzuführen
- 😞 Wenig Flexibilität – Reaktion auf Änderungen schwierig
- 😞 Gefahr: Dokumentation wichtiger als eigentliches System
- 😞 Finale SW-Lösung steht erst spät zur Verfügung
→ späte Fehlererkennung



- Zielsetzung: Reduzierung der Gesamtentwicklungszeit
- Ansatz
 - Phasenüberlappendes Arbeiten und Rückkopplungen zwischen den Phasen
 - Nachfolgeteam beginnt bereits wenn es vom Vorgängerteam geeignete Informationen erhalten hat
 - Paralleles Arbeiten, ggf. Überarbeitung bei entsprechender Information
- Voraussetzung
 - Gute Kommunikationsmöglichkeiten und ausreichende Kapazitäten für die Überarbeitungen

Nebenläufiges Modell (2)

02 Prozessmodelle / 02.2 Basis-Vorgehensmodelle



Bewertung Nebenläufiges Modell

02 Prozessmodelle / 02.2 Basis-Vorgehensmodelle

- 😊 Optimale Zeitausnutzung
- 😊 Vorgängerteams erhalten frühzeitig Rückmeldung, ob die Anforderungen bzw. Entwürfe umsetzbar sind
- 😞 Risiko, dass die grundsätzlichen und kritischen Entscheidungen zu spät getroffen werden
- 😞 Nachfolgerteam fängt zu früh an, obwohl Informationen unzureichend sind
- 😞 Hohe Anforderungen an Kommunikation (räumlich verteilte Teams!)

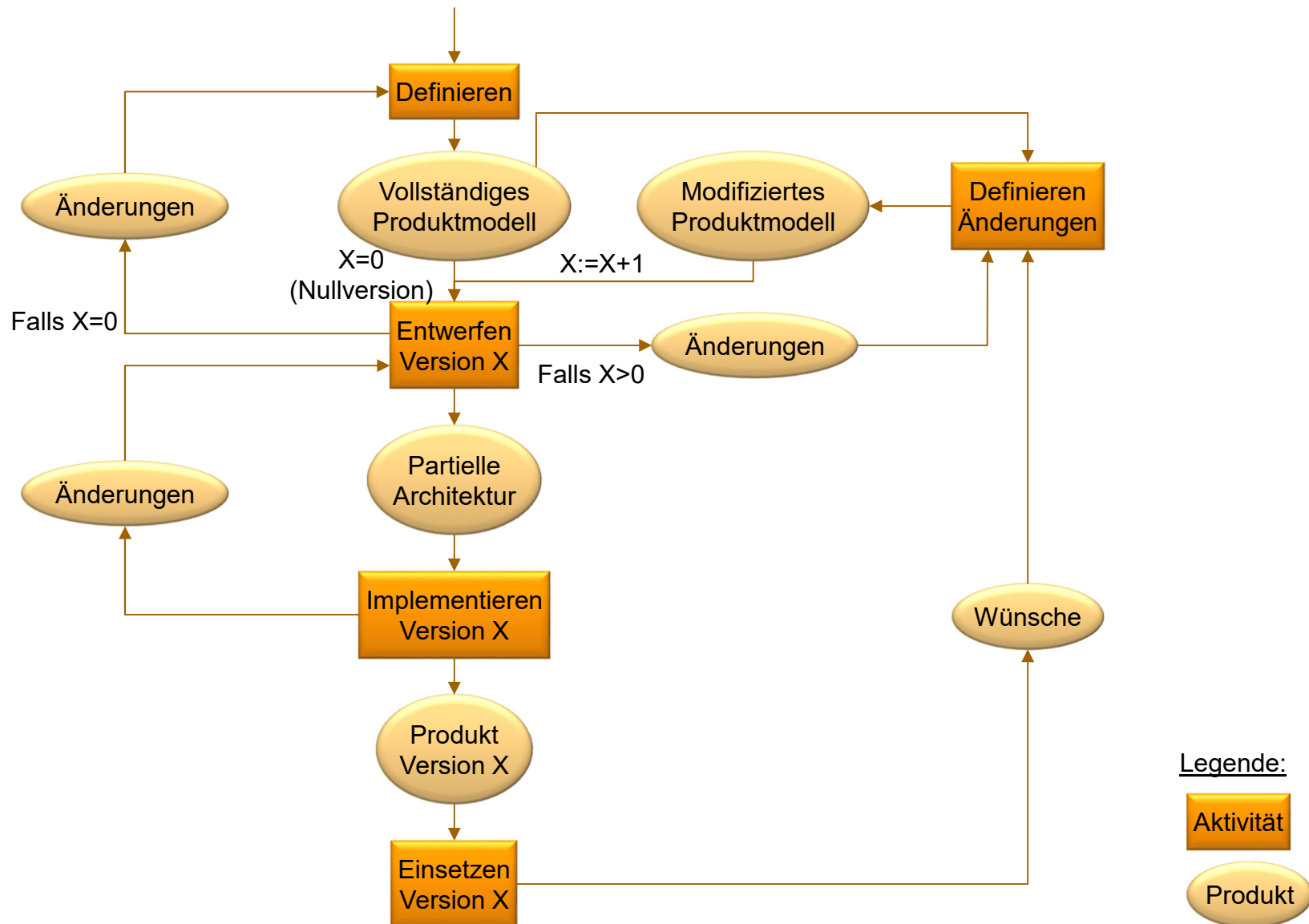


● Ansatz

- Möglichst vollständige Erfassung und Modellierung der Anforderungen an das Produkt
- Zerlegung des Produkts
 - Aufbau aus Teilprodukten
 - oder schalenförmiger Aufbau
- Erst wird nur das erste Teilprodukt bzw. nur der Kern entworfen und implementiert
- AG bekommt Version 0 ausgeliefert und kann sie bereits einsetzen
- Einsatzerfahrungen werden bei Version 1 berücksichtigt

Inkrementelles Modell (2)

02 Prozessmodelle / 02.2 Basis-Vorgehensmodelle



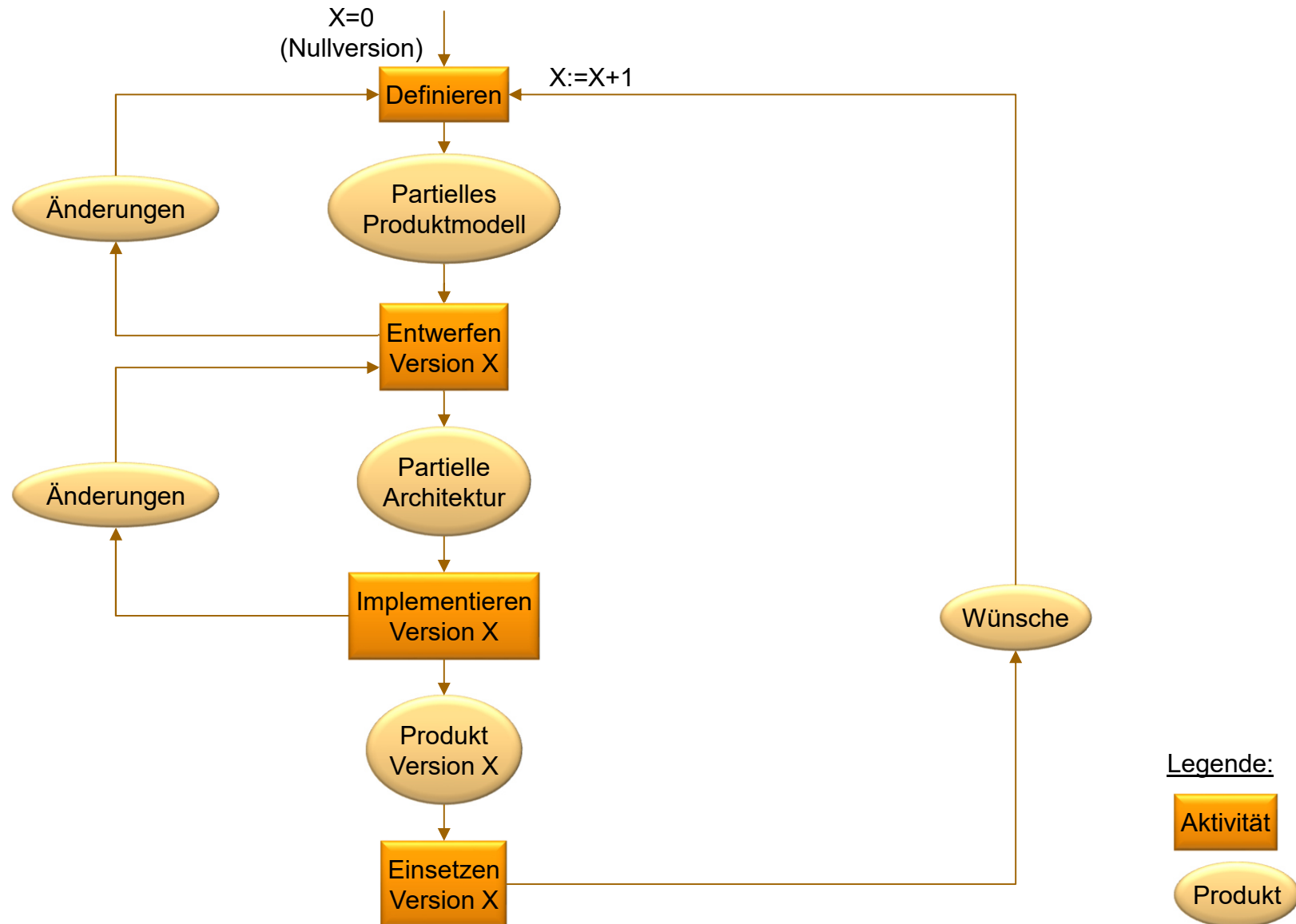
- 😊 Von Beginn eine vollständige Anforderungsdefinition
→ gezielte Aufteilung in Teilprodukte und Auswahl einer geeigneten Systemarchitektur
- 😊 AG erhält in kurzen Zeitabständen einsatzfähige Produkte
- 😊 Risiko bzgl. falscher Architekturentscheidung wird minimiert
- 😞 Ggf. lange Definitionsphase (alle Anforderungen müssen erfasst, diskutiert, modelliert und verabschiedet werden)
- 😞 Auftreten von geänderten Kundenanforderungen oder massiven Änderungen (Architektur, Entwurf)

- Ausgangspunkt: **Kern- und Muss-Anforderungen** des AG
 - Legen Produktkern oder ein Teilprodukt fest
 - Nur Produktkern / Teilprodukt werden entworfen und umgesetzt
 - **Nullversion** wird an AG ausgeliefert
- AG sammelt Erfahrungen mit Nullversion und ermittelt daraus Anforderungen für eine erweiterte Version
- Implementierung wird über viele Versionen hinweg verfeinert bis ein angemessenes System entstanden ist



Evolutionäres Modell (2)

02 Prozessmodelle / 02.2 Basis-Vorgehensmodelle



- 😊 AG erhält in kurzen Zeitabständen einsatzfähige Produkte (frühzeitiger Erfahrungsgewinn)
- 😊 Spezifikation wird nach und nach erstellt
- 😞 Gefahr, dass Systemarchitektur komplett überarbeitet werden muss (übersehene Kernanforderungen)
- 😞 Gefahr, dass Nullversion nicht flexibel genug ist sich an ungeplante Evaluationspfade anzupassen

● Ausgangssituation

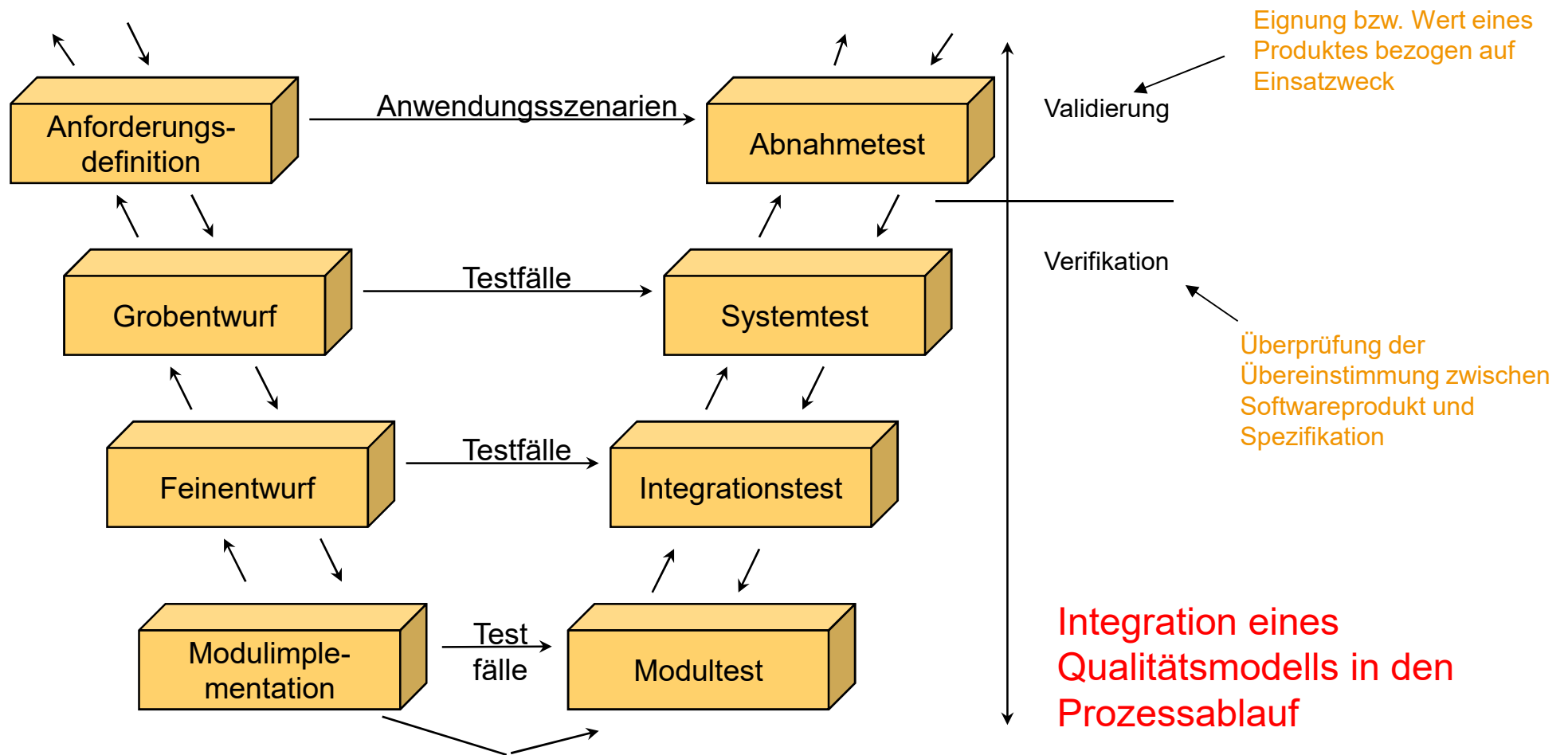
- Ergänzung der Entwicklungsphase mit Inspektionsaktivität für Qualitätssicherung i.d.R. nicht ausreichend

● Ansatz

- In spezifizierenden Phasen werden Vorgaben für die realisierenden Phasen festgelegt
 - Anwendungsszenarien für die Produktabnahme (Anwendersicht)
 - Integrationstestfälle (Architektursicht)
 - Modul-/Komponententest (Implementierungssicht)
- Integrierte Qualitätssicherung
 - Nach jeder Phase Anwendung von Methoden wie Review oder Inspektion

V-Modell (2)

02 Prozessmodelle / 02.2 Basis-Vorgehensmodelle



- Prozessmodell mit integriertem Qualitätsmodell
- Starres Modell – festgelegter, nicht änderbarer Ablauf
- 😊 Systematische Struktur – gute Planung
- 😞 Umgang mit Fehlern und Änderungen
(späte Erkennung → sehr hohe Aufwände für Korrektur)

- **Metamodell,**
 - das hilft in jeder Phase einer Softwareentwicklung
 - das geeignete Prozessmodell zu finden.
- Für jedes Teilprodukt sind vier zyklische Schritte zu durchlaufen:
 - (1) Definition von Zielen und Alternativen
 - (2) Einschätzen des Risikos
 - (3) Entwicklung und Durchführung von Test und Evaluierungen der aktuellen Ergebnisse
 - (4) Planung der nächsten Phase

02 Prozessmodelle / 02.2 Basis-Vorgehensmodelle



Bewertung Spiralmodell

02 Prozessmodelle / 02.2 Basis-Vorgehensmodelle

- 😊 Periodische Überprüfung und ggf. erneute Festlegung des Prozessmodells in Abhängigkeit von den Risiken
- 😊 Flexibles Modell (Integration verschiedener Modelle)
- 😊 Fehler und ungeeignete Alternativen werden frühzeitig eliminiert
- 😞 Hoher Managementaufwand, da viele Entscheidungen zu treffen sind
- 😞 Für kleine und mittlere Projekte weniger gut geeignet
- 😞 Wissen über das Identifizieren und Managen von Risiken nicht weit genug verbreitet



- Welches Basismodell würden Sie für die Entwicklung folgender Software-Systeme auswählen?
 - (1) System, das bei einem Auto ein Antiblockiersystem steuert
 - (2) Buchhaltungssystem, das in einem Unternehmen das bestehende System ersetzt
 - (3) Interaktives System für Bahnpassagiere, das auf Bahnhöfen die Abfahrtszeiten von Zügen findet
 - (4) Virtual-Reality-System für Kraftwerke

