Vorlesung Softwaretechnik I (SS 2024)

4. Prozessmodelle

Prof. Dr. Jens Grabowski

Tel. 39 172022

grabowski@informatik.uni-goettingen.de

Vorgehensmodell versus Prozessmodell

- Vorgehensmodelle …
  - sind Schablonen für das Vorgehen in Softwareprojekten.
  - geben Projektleitern/Entwicklern Hinweise welche Tätigkeiten als nächstes auszuführen sind.
  - machen jedoch keine Aussagen über
    - die personelle Organisation,
    - die Dokumentation und ihre Gliederung, oder
    - die Verantwortlichkeiten für Aktivitäten und Dokumente

    in einem Softwareprojekt.

- Prozessmodelle …
  - sind konkrete Implementierungen von Vorgehensmodellen.

Vorgehensmodell versus Prozessmodell

- Prozessmodelle machen (in der Regel) Aussagen zu
  - Organisation, Verantwortlichkeiten und Rollenverteilung;
  - Struktur und Merkmale der Dokumente;
  - einzusetzende Verfahren;
  - auszuführende Schritte der Entwicklung, ihre Reihenfolge und ihre Abhängigkeiten (Vorgehensmodell);
  - Projektphasen, Meilensteine und Prüfkriterien;
  - Notationen und Sprachen;
  - Werkzeuge.

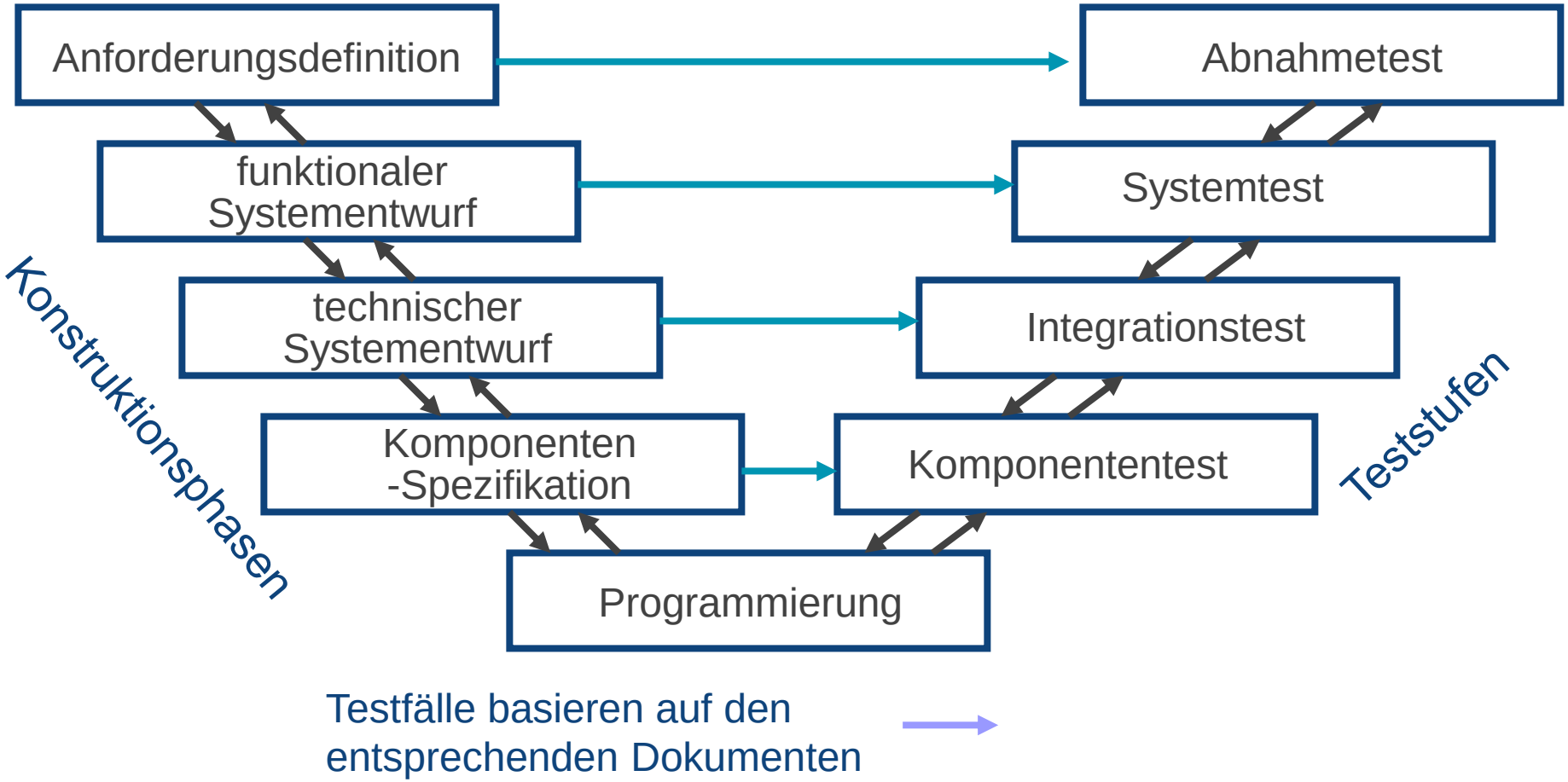
Inhalt

- **V-Modell und V-Modell XT**
- Unified Process
- Agile Prozesse
  - Extreme Programming (XP)
  - Scrum
  - Agile Prozesse – Pro und Kontra
- Lernziele

V-Modell und V-Modell XT – Geschichte

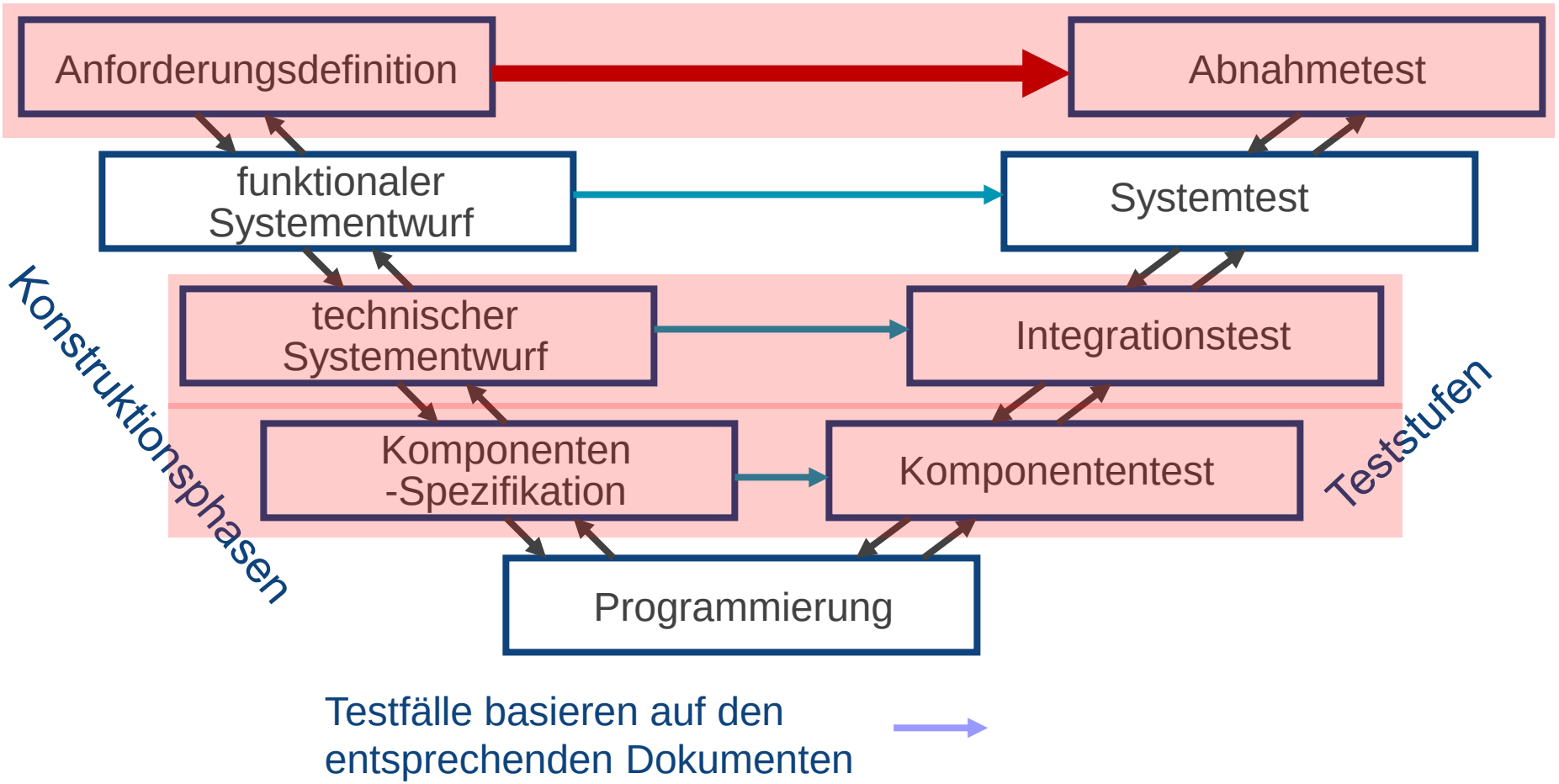
- Ursprüngliche Idee ist ein auf dem Wasserfallmodell basierendes Vorgehensmodell von Barry W. Boehm (1979)
- **V-Modell** vom Bundesministerium für Verteidigung weiterentwickelt.
- Seit 1992 per Erlass Entwicklungsstandard bei der Bundeswehr
- 1995/96 Entwicklung/ einer „zivilen Variante“ des V-Modells
- 1997: Veröffentlichung des **V-Modells 97**
  - Inkrementelle Entwicklung, OO-Entwicklung, koordinierte Entwicklung von Soft- und Hardware
- 2004: **V-Modell XT** (eXtreme Tailoring)
  - Einbindung des Auftragnehmers, stärkere Modularisierung, stärkere Orientierung in Richtung agiler und inkrementeller Ansätze.

V-Modell

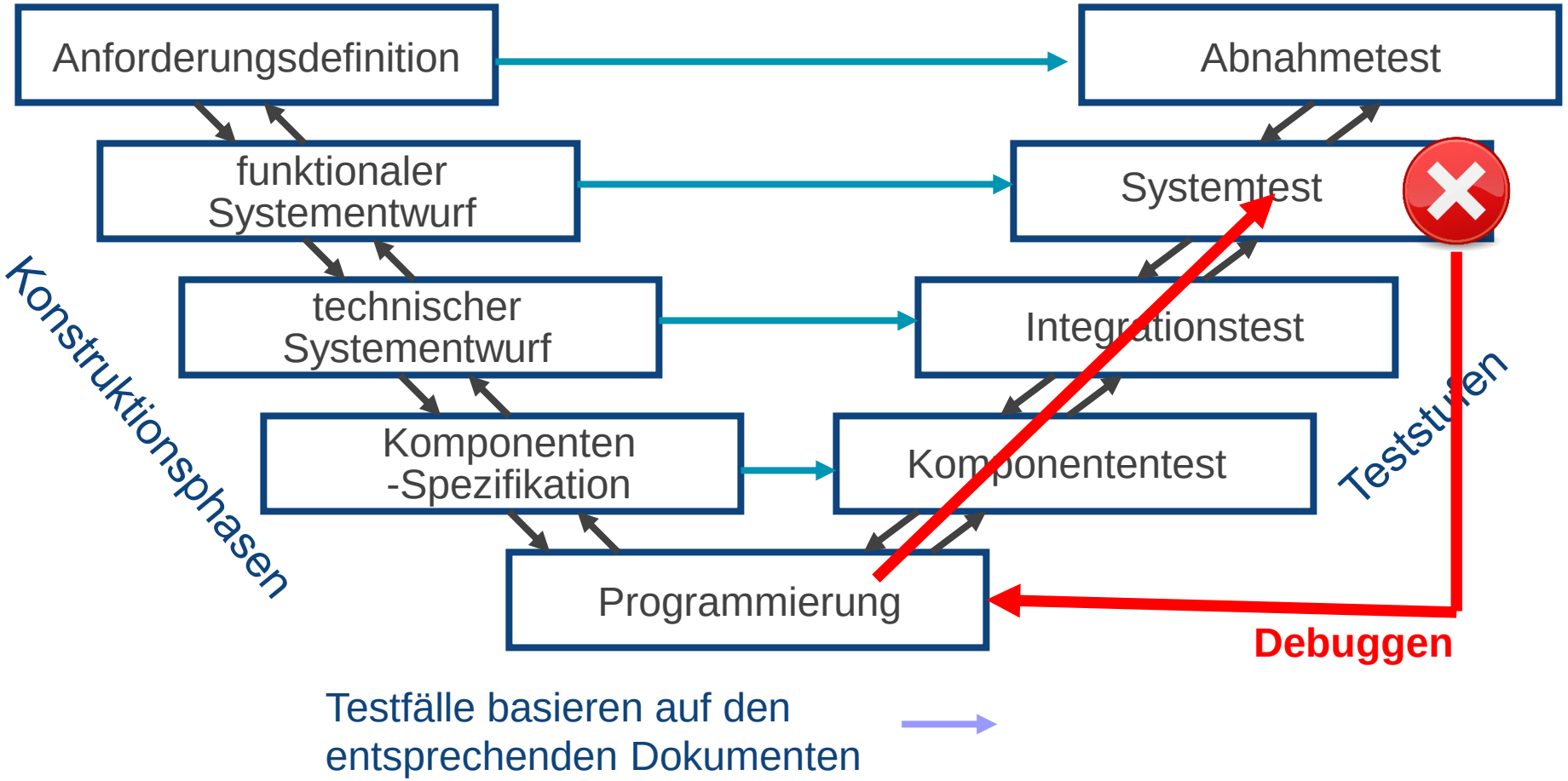


Testfälle basieren auf den entsprechenden Dokumenten

V-Modell

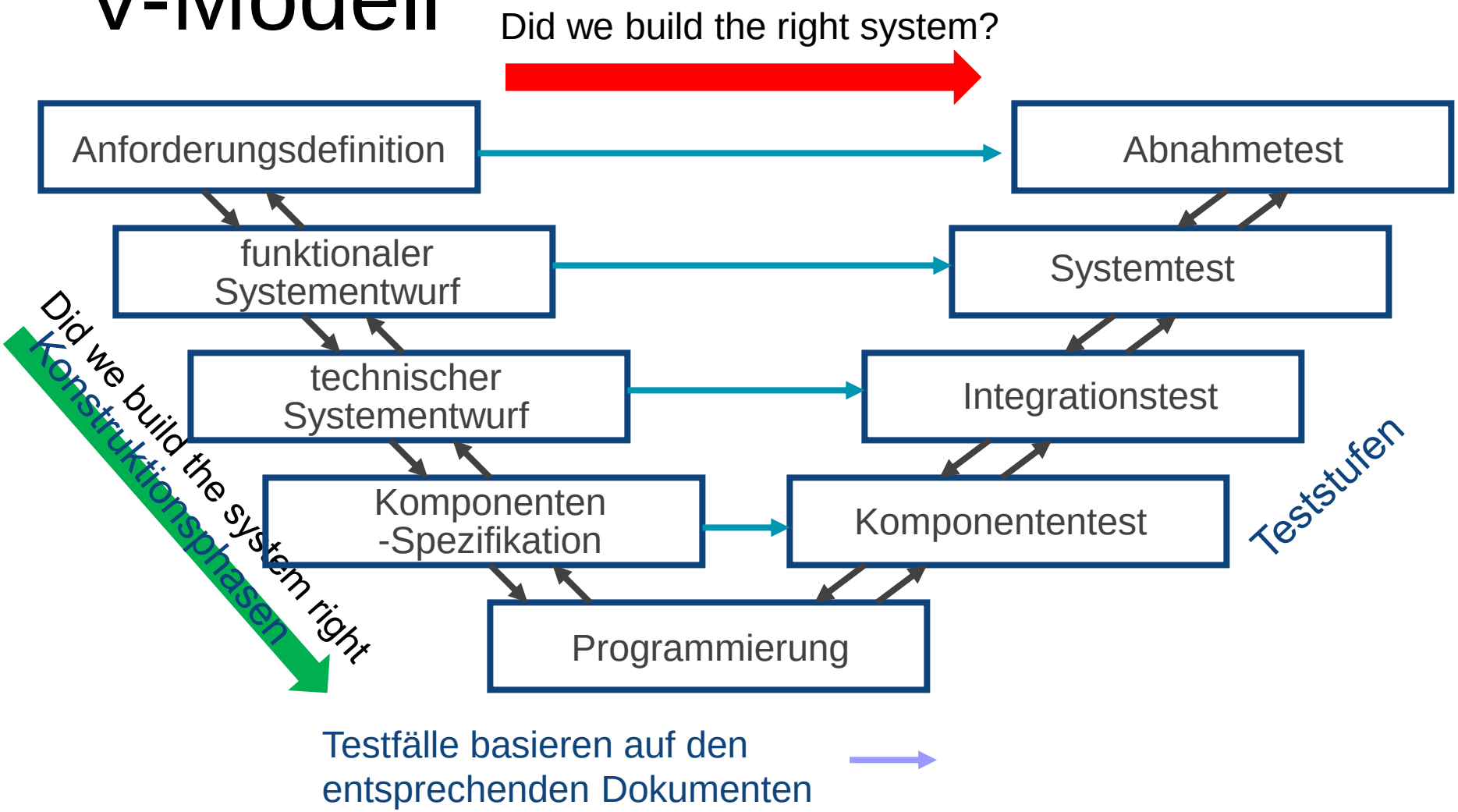

Anforderungsdefinition
Abnahmetest
funktionaler Systementwurf
Systemtest
technischer Systementwurf
Integrationstest
Komponenten -Spezifikation
Komponententest
Programmierung
Konstruktionsphasen
Teststufen
Testfälle basieren auf den entsprechenden Dokumenten

V-Modell



Testfälle basieren auf den entsprechenden Dokumenten

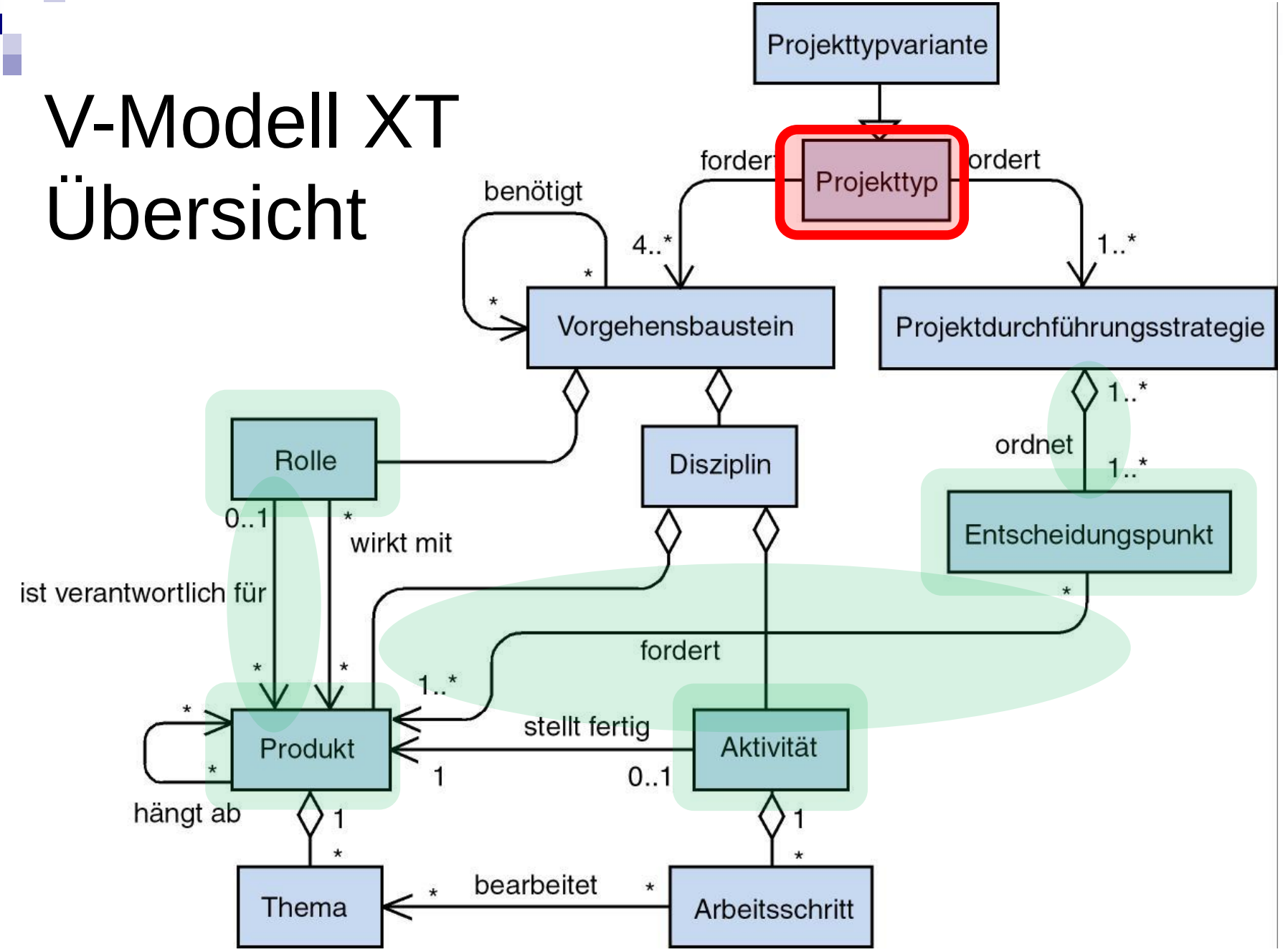
V-Modell


Did we build the right system?
Anforderungsdefinition
Abnahmetest
funktionaler Systementwurf
Systemtest
technischer Systementwurf
Integrationstest
Komponenten -Spezifikation
Komponententest
Programmierung
Did we build the system right
Konstruktionsphasen
Teststufen
Testfälle basieren auf den entsprechenden Dokumenten

V-Modell XT – Eigenschaften

- Weiterentwicklung des Standard-Phasenmodells.
- Unterteilt Projekt in Projektabschnitte (= Phasen), die in einem Entscheidungspunkt (= Meilenstein) enden.
- Unterstützt verschiedene Projekttypen.
- Unterscheidet zwischen Auftragnehmer- und Auftraggeberprojekten.
- Integriert projektbegleitende Tätigkeiten (insbesondere Qualitätssicherung, Konfigurationsverwaltung und Projektmanagement).
- Unterstützt traditionelle, inkrementelle, komponentenbasierte und agile Entwicklungsprozesse.
- Erweiterbar und anpassbar.

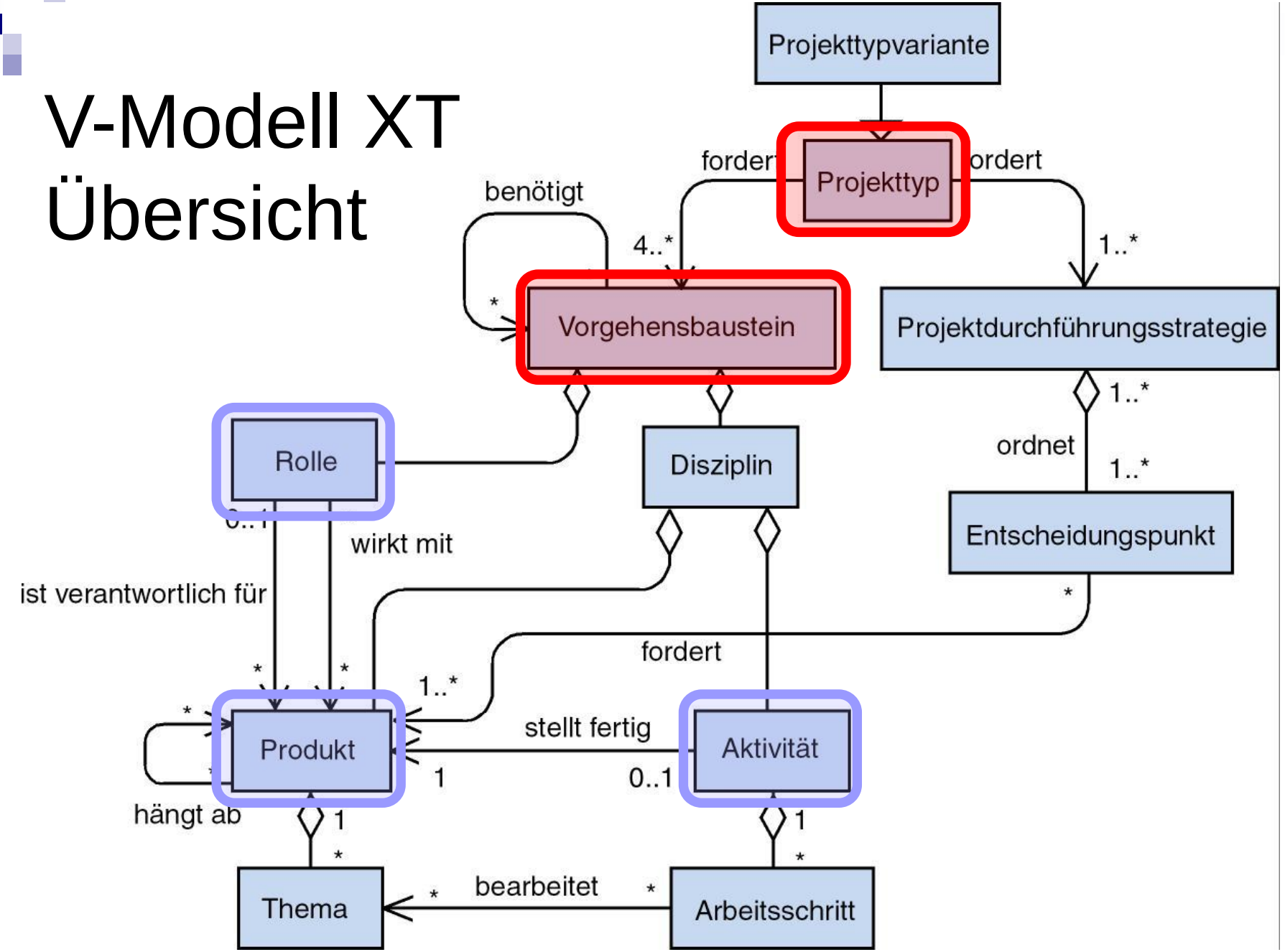
V-Modell XT Übersicht



V-Modell XT – Projekttypen

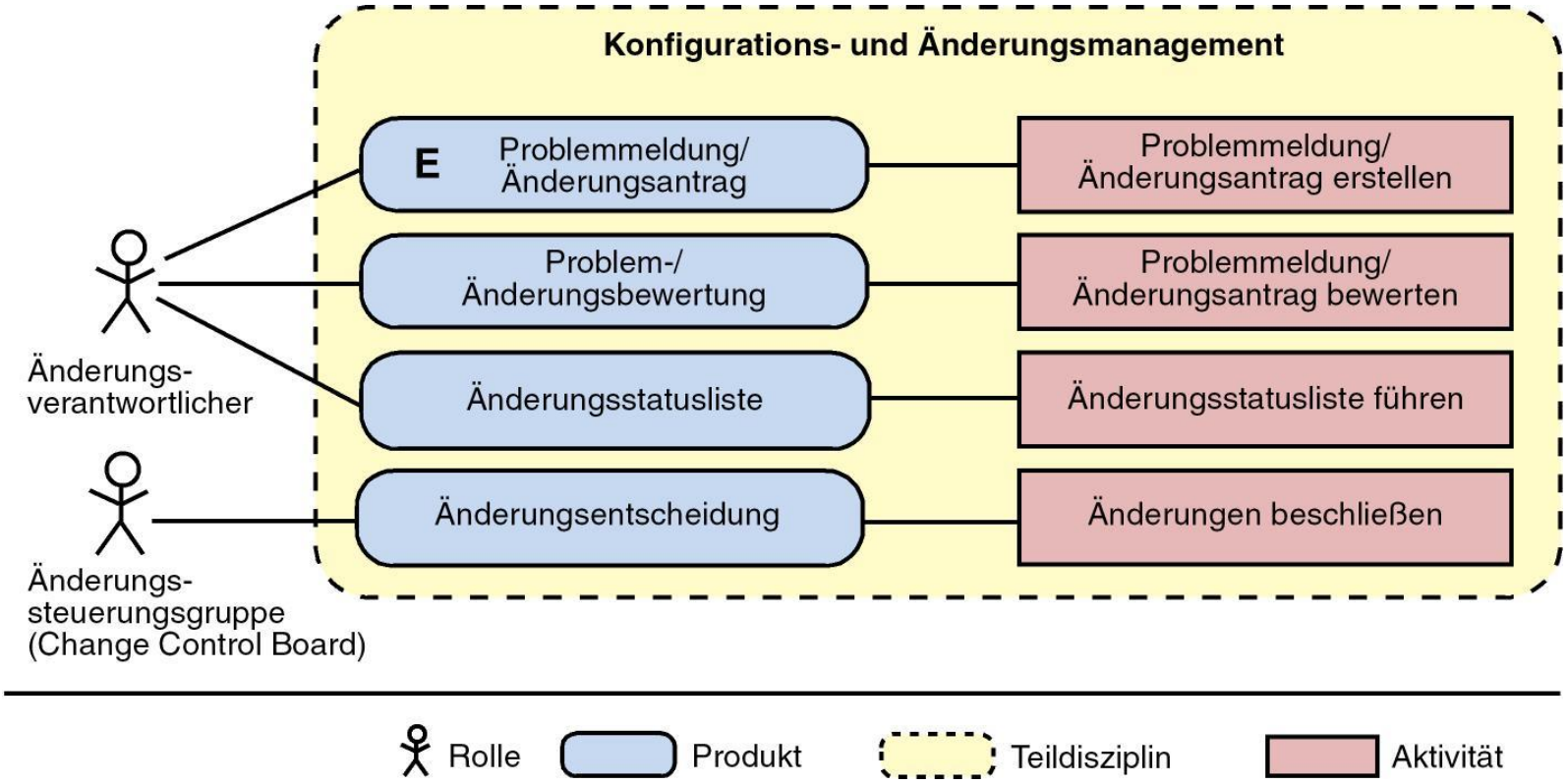
- 3 Grundlegende Projekttypen
  - PT 1: Systementwicklungsprojekt (AG)
    - aus der Sicht des **A**uftrag**G**ebers
  - PT 2: Systementwicklungsprojekt (AN)
    - aus der Sicht des **A**uftrag**N**ehmers
  - PT 3: Systementwicklungsprojekt (AG / AN)
    - Keine Trennung zwischen AG und AN notwendig (z.B. AG und AN kommen aus derselben Organisation)

V-Modell XT Übersicht



V-Modell XT –

Vorgehensbaustein (Problem- und Änderungsmangement)



V-Modell XT – Aktivitäten, Produkte, Disziplinen

- Aktivitäten, Arbeitsschritte
  - Aktivitäten bearbeiten oder erstellen Produkte.
  - Aktivitäten können in Arbeitsschritte gegliedert sein.
  - Nicht-untergliederbare Aktivitäten und Arbeitsschritte werden en bloc durchgeführt.
  - Das V-Modell XT definiert mehr als 90 Aktivitäten.
- Produkte, Themen
  - Produkte sind Ergebnisse und Zwischenergebnisse von Projekten
  - Themen gliedern Produktgruppen.
  - Insgesamt kennt das V-Modell XT über 100 Produkte.
- Disziplin
  - Gruppe von inhaltlich eng zusammenhängenden Produkten und Aktivitäten, welche die Produkte erstellen

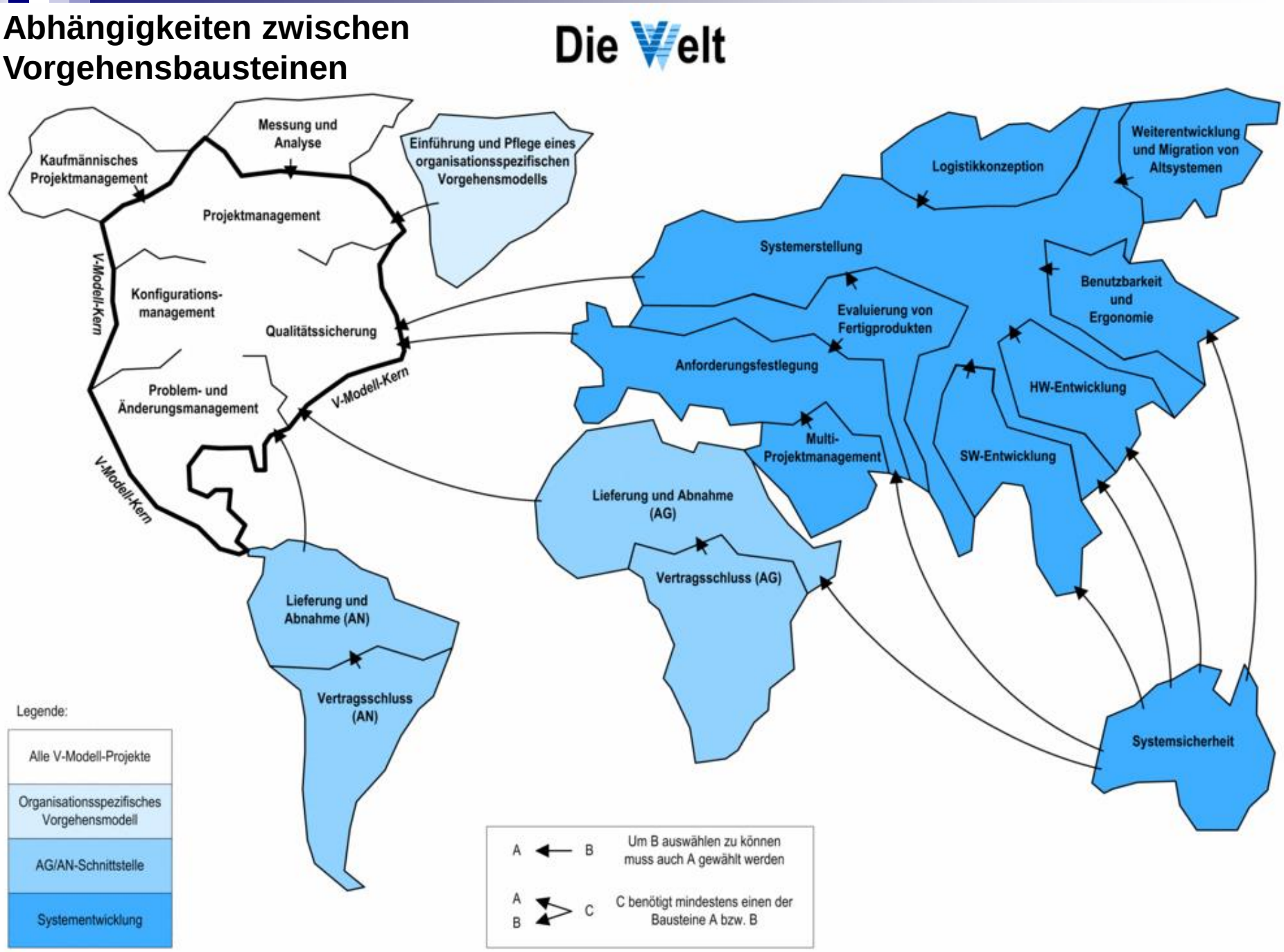
V-Modell XT – Rollen und Vorgehensbausteine

- Rollen
  - zusammengehörige Aufgaben, Verantwortlichkeiten und Fähigkeiten.
  - werden Produkten zugeordnet (jedem Produkt ist genau eine Rolle).
  - Insgesamt kennt das V-Modell XT 32 Rollen.
- Vorgehensbausteine
  - sind die zentralen Einheiten des V-Modell XT.
  - fassen inhaltlich abhängige Rollen, Produkte und Aktivitäten zusammen.
  - können voneinander abhängig sein.
  - 4 obligatorische Vorgehensbausteine, für alle Projekttypen:
    - Projektmanagement
    - Qualitätssicherung
    - Problem und Änderungsmanagement
    - Konfigurationsmanagement.

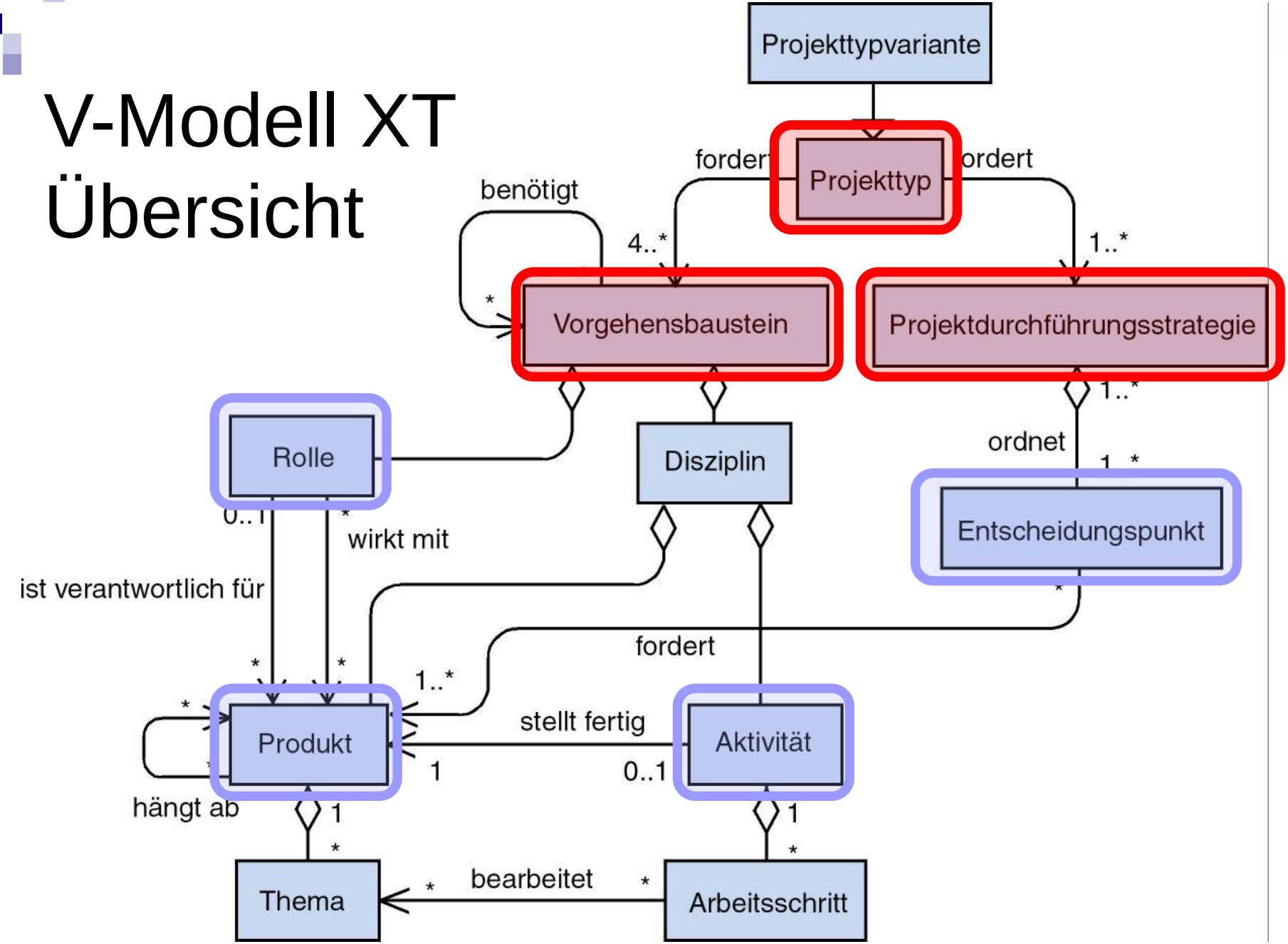
V-Modell XT – Vorgehensbausteine

- Vorgehensbausteine
  - sind die zentralen Einheiten des V-Modell XT.
  - fassen inhaltlich abhängige Rollen, Produkte und Aktivitäten zusammen.
  - können voneinander abhängig sein.
  - 4 obligatorische Vorgehensbausteine, für alle Projekttypen:
    - Projektmanagement
    - Qualitätssicherung
    - Problem und Änderungsmanagement
    - Konfigurationsmanagement.

Abhängigkeiten zwischen Vorgehensbausteinen



V-Modell XT Übersicht

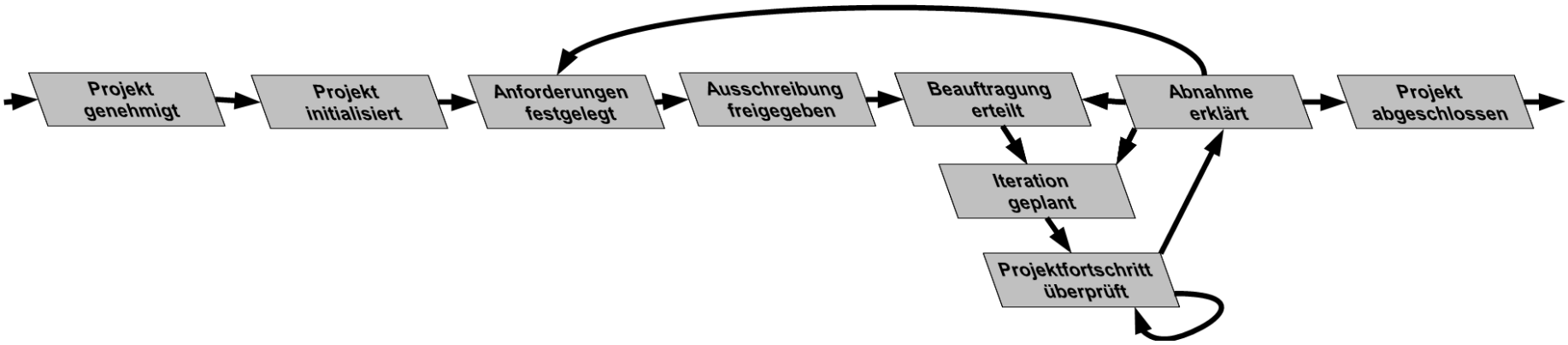


V-Modell XT

- Projektdurchführungsstrategie
  - ordnet eine Menge von zusammengehörenden Entscheidungspunkten und gibt deren zeitliche Reihenfolge vor.
  - schafft den Rahmen, um ein Projekt geordnet und nachvollziehbar durchzuführen.
  - liefert die Grundlage für die Projektplanung.
- Entscheidungspunkte
  - entsprechen den Meilensteinen im Phasenmodell.
  - teilen das Projekt in Projektabschnitte (Phasen) ein.
  - definieren Zeitpunkte im Projekt an denen entschieden wird, ob der nächste Projektabschnitt begonnen wird.
    - Hierfür definiert jeder Entscheidungspunkt die Produkte, die am Entscheidungspunkt erstellt sein müssen und deren Bewertung die Grundlage der Entscheidung ist.
    - Das V-Modell XT definiert 21 Entscheidungspunkte.

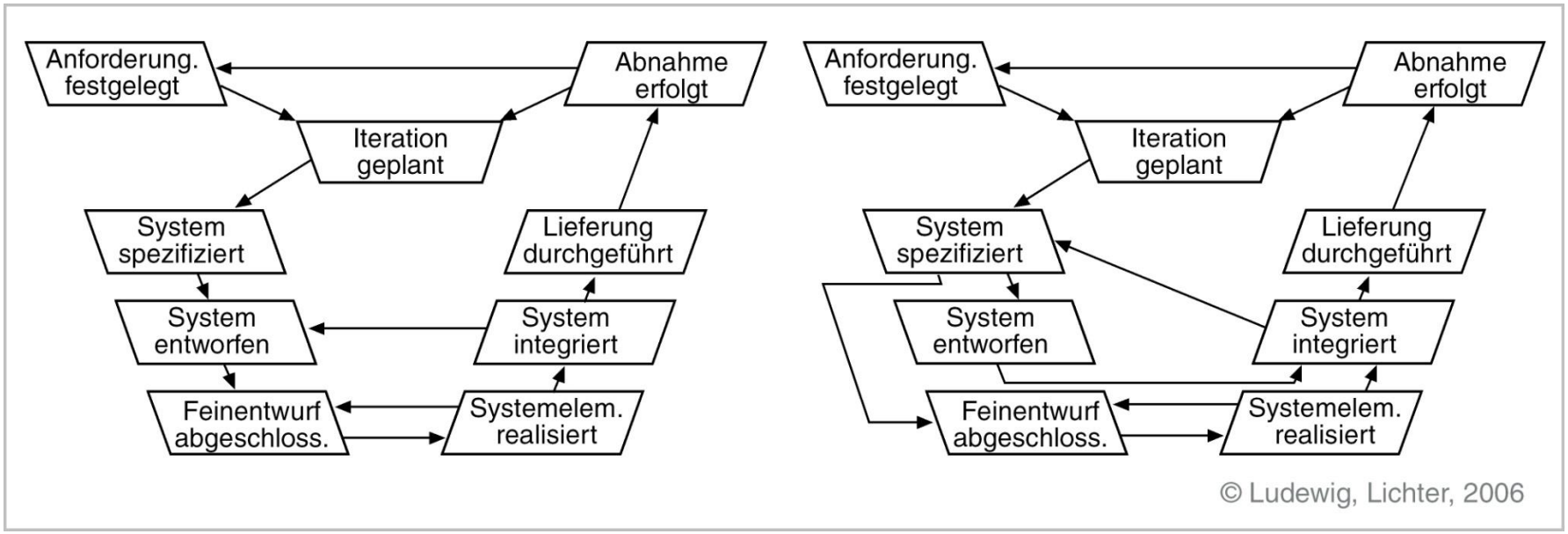
V-Modell XT – Projektdurchführungsstrategien

- Beispiel Projektdurchführungsstrategie:
  - AG-Projekt mit einem Auftragnehmer

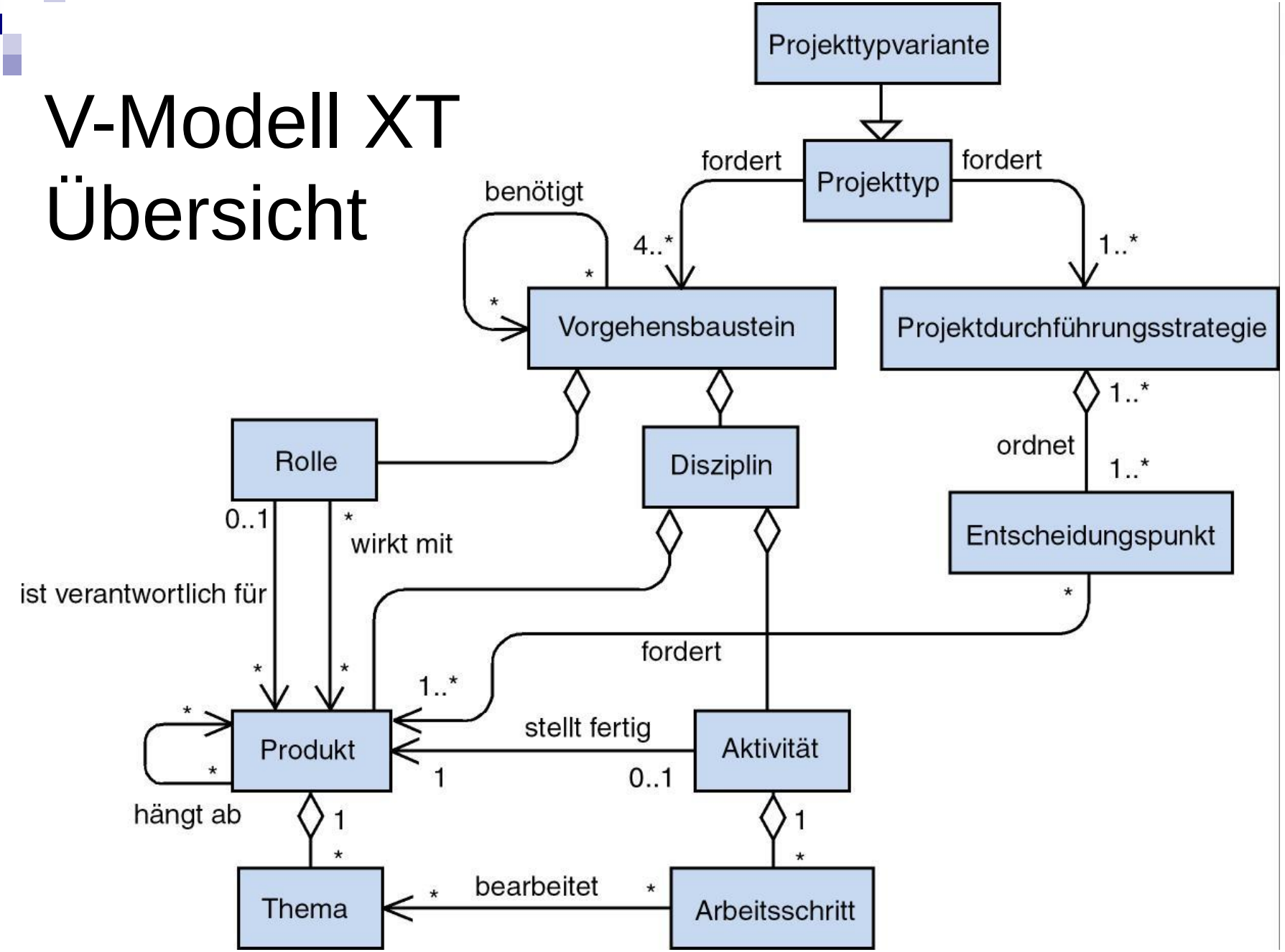


V-Modell XT – Ablaufbausteine

- Inkrementelle und komponentenbasierte Entwicklung (vereinfacht)



V-Modell XT Übersicht



V-Modell XT - Zusammenfassung

- Für jedes Projekt sind Vorgehensbausteine und Projektdurchführungsstrategien festgelegt.
- Eine Projektdurchführungsstrategie ordnet Entscheidungspunkte an, die erreicht werden müssen.
- An jedem Entscheidungspunkt müssen definierte Produkte fertig gestellt sein.
- Ein Vorgehensbaustein fasst alle Rollen, Produkte und Aktivitäten zusammen, die notwendig sind, um eine zentrale Projektaufgabe zu lösen.
- Vorgehensbausteine bauen aufeinander auf; jedes Projekt muss mindestens vier Kern-Vorgehensbausteine enthalten.
- Produkte und Aktivitäten sind gruppiert und können gegliedert sein. Jedes Produkt wird durch genau eine Aktivität fertig gestellt. Produkte hängen voneinander ab.
- Rollen sind für Produkte verantwortlich und wirken an der Erstellung von Produkten mit.

V-Modell XT – Tailoring

- Tailoring beschreibt die Anpassung des V-Modell XT auf ein konkretes Projekt.
- Tailoring geschieht durch Angabe von
  - Projektgegenstand (z.B. eingebettetes System, Hard- oder Software)
  - Rolle im Projekt (Auftraggeber, Auftragnehmer)
  - 9 weitere Projektmerkmale, beinhalten z.B.
    - Systemzyklusausschnitt (Entwicklung, Wartung, Migration)
    - Einsatz von Fertigprodukten
    - Vorhandensein von Bedienoberflächen
    - Risikoeinstufung
    - Usw.

V-Modell XT – Tailoring

- Ergebnis des Tailoring
  - Optionale und verpflichtende Vorgehensbausteine
  - Projektdurchführungsstrategie
- Tailoring wird durch den Projektassistenten (Software) unterstützt.
- Tailoring legt Produkte, Aktivitäten, Entscheidungspunkte und Reihenfolge von Entscheidungspunkten fest, sodass hierauf aufbauend die einzelnen Tätigkeiten geplant werden können.

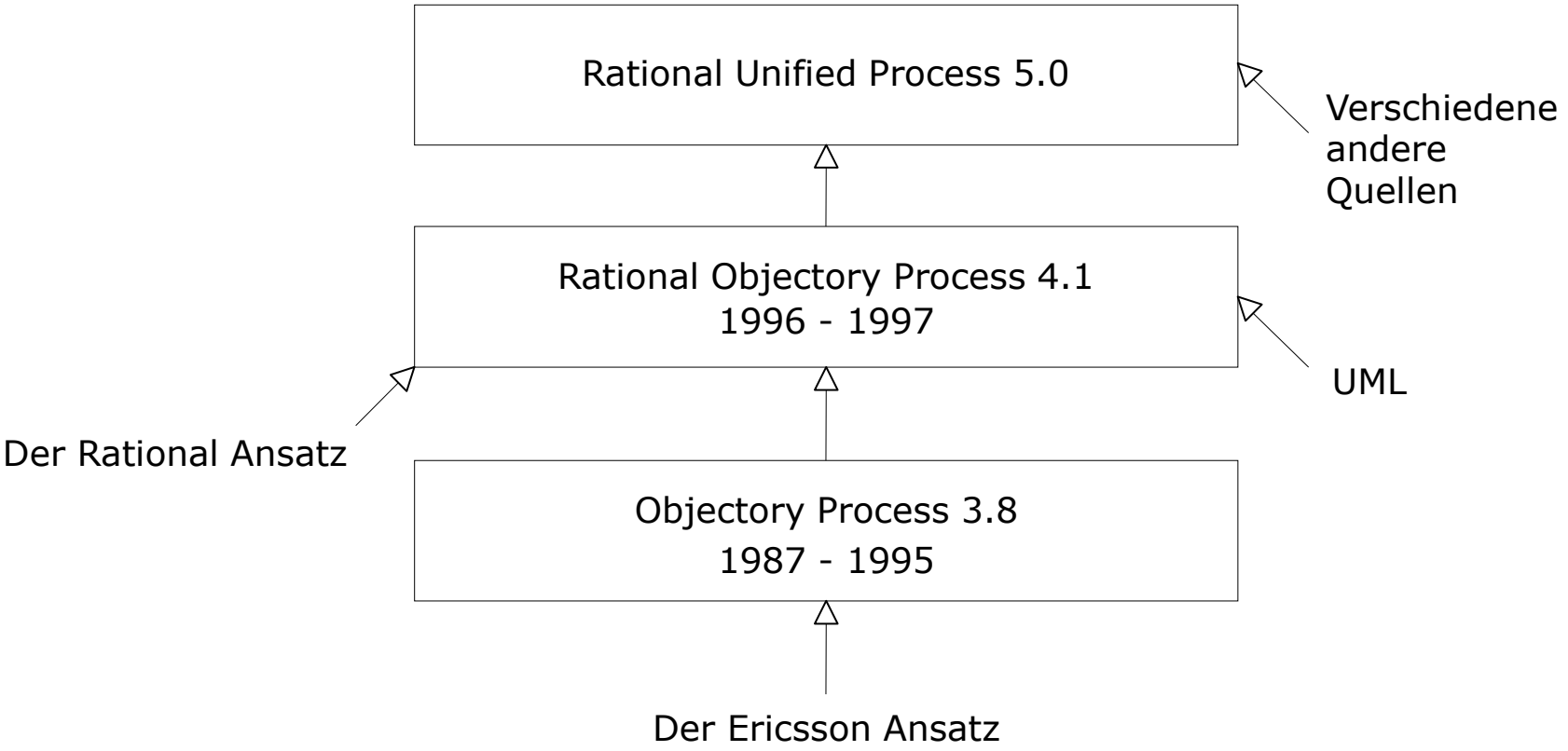
V-Modell XT – Bewertung

- Positiv
  - 4 Kernbausteine (Problem und Änderungsmanagement, Projektmanagement, Konfigurationsmanagement, Qualitätssicherung) bleiben im Bewusstsein des Managements.
  - Modell ist öffentlich und kann ohne Lizenzkosten benutzt werden.
  - Modell ist generisch und bietet Unterstützung um es unternehmens- und projektspezifisch anzupassen.
- Problematisch
  - Sehr großer Umfang von Produkten, Aktivitäten, Rollen und Projektdurchführungsstrategien. Tailoring kostet recht viel Aufwand und dadurch ist das V-Modell XT nur bedingt für mittlere und kleine Projekte geeignet.
  - V-Modell XT ohne erhebliche Anpassungen ist aufgrund der vielen Produkte sehr schwerfällig. Auch nach dem Tailoring werden sehr viele Produkte gefordert.
  - Die in den Projektdurchführungsstrategien modellierten Abläufe der Entwicklungsstrategien sind zum Teil diskutabel.

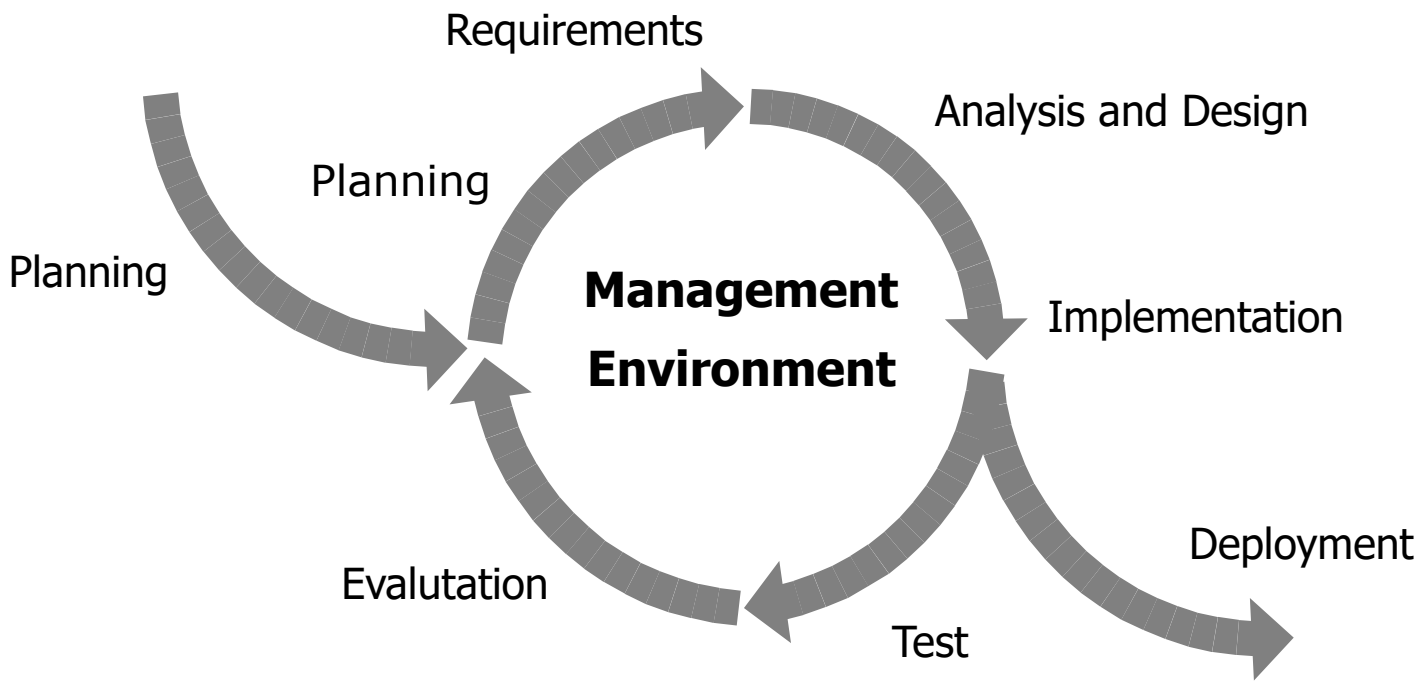
Inhalt

- V-Modell und V-Modell XT
- **Unified Process**
- Agile Prozesse
  - Extreme Programming (XP)
  - Scrum
  - Agile Prozesse – Pro und Kontra
- Lernziele

Unified Process – Geschichte



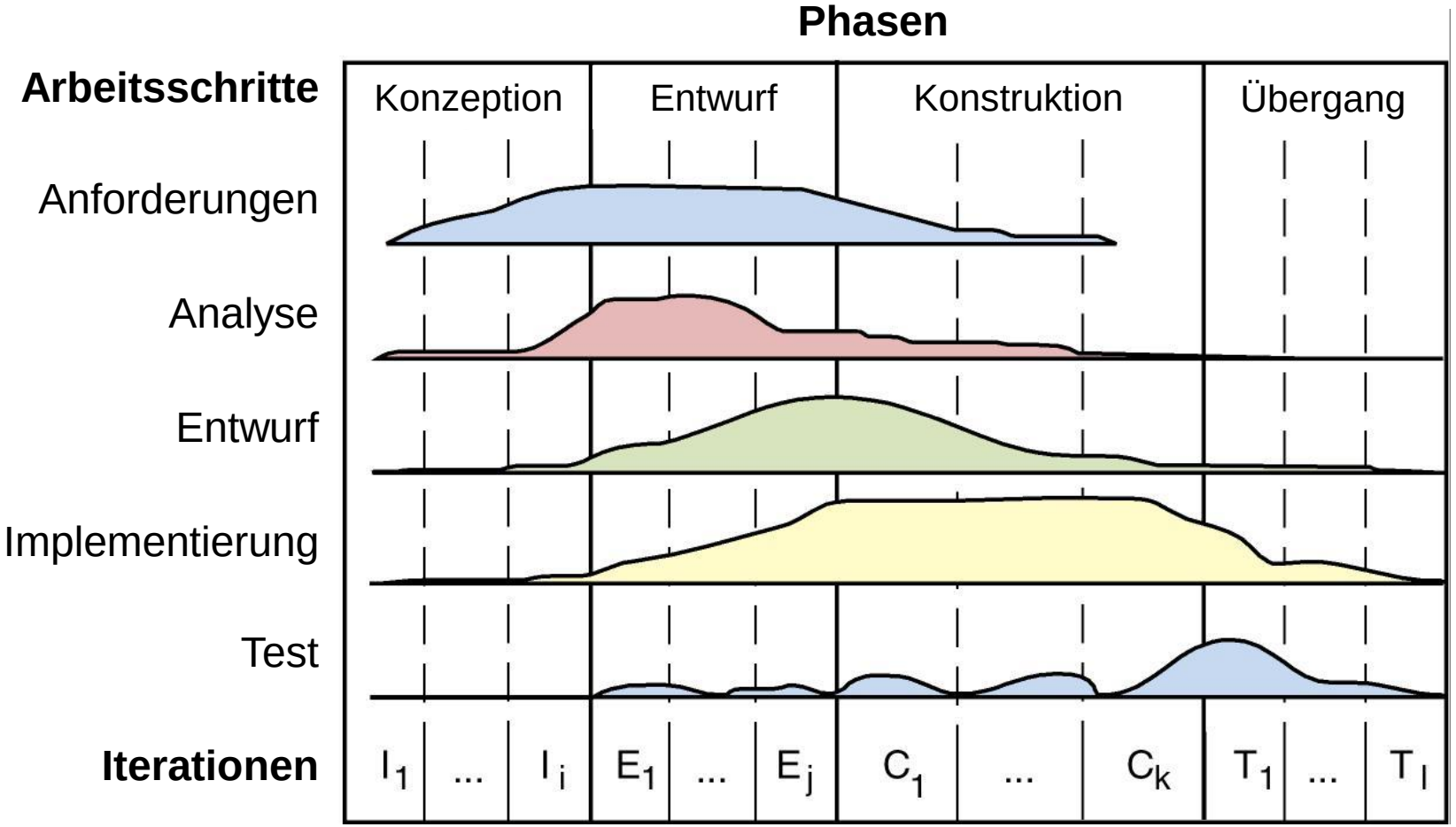
Iterativ und Inkrementell



Der Unified Process …

- … eine Sammlung und Weiterentwicklung von Best Practices.
- … ein objektorientierter Software-Entwicklungs-Prozess (OO-SWEP).
- … verwendet UML zur Darstellung der Modelle.
- … Anwendungsfall-gesteuert
  - Anforderungen werden in Form von Anwendungsfällen beschrieben
- … Architektur-zentriert
  - Wesentlicher Erfolgsfaktor eines Software-Systems
  - Auswirkungen für gesamtes System (Effizienz ...)
  - Schnittstellen
- … iterativ und inkrementell
  - kleine Schritte
  - geringes Risiko
  - kleine Rückschläge

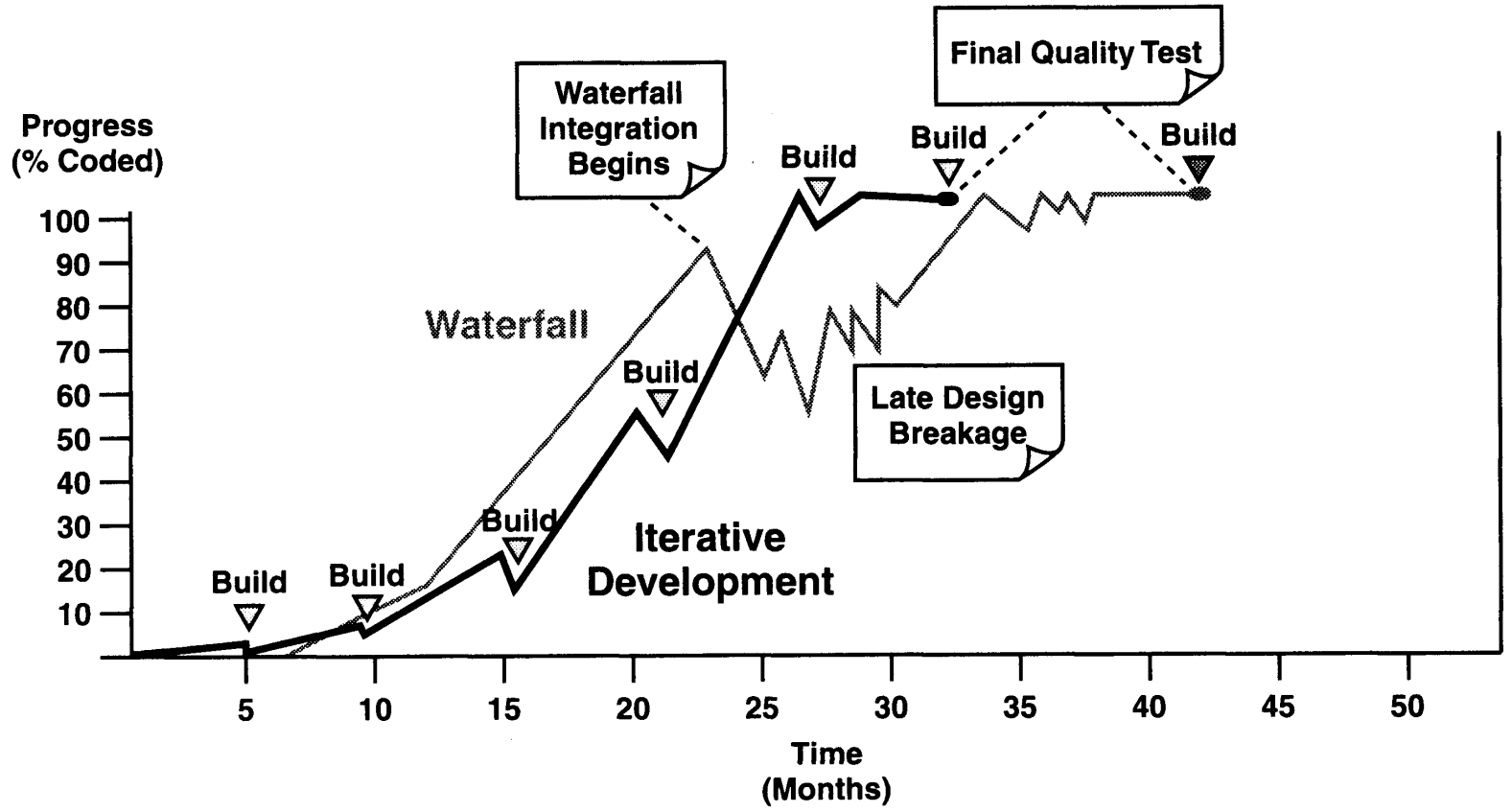
Phasen im Unified Process



Projektgröße und Iterationen

Komplexität des Projekts	Summe Iterationen	Iteratio-nen Beginn	Iterationen Ausarbeitung	Iterationen Konstruktion	Iterationen Umsetzung
Niedrig	3	0	1	1	1
Normal	6	1	2	2	1
Hoch	9	1	3	3	2

Vergleich Unified Process und Wasserfall-Modell



Unified Process – Bewertung

- **Voraussetzungen für den Einsatz**
  - Ausgezeichnete Konfigurations- und Änderungsverwaltung
    - Iterative und inkrementelle Ansätze führen dazu, dass sich die Arbeitsergebnisse in jeder Iteration ändern.
  - Gute Projektmanagement-Fertigkeiten
    - Planung von Anzahl und Dauer der Iterationen erfordert viel Erfahrung.
  - Kenntnis objektorientierter Konzepte und Notationen.
- **Positiv**
  - Gute Darstellung des Prozesses, hoher Detaillierungsgrad der Beschreibung
- **Problematisch**
  - Schwierige Anpassung an spezielle Gegebenheiten einer Organisation
    - Prozess ist sehr detailliert beschrieben und dadurch sind die Prozesselemente sehr stark miteinander vernetzt.

Inhalt

- V-Modell und V-Modell XT
- Unified Process
- **Agile Prozesse**
  - Extreme Programming (XP)
  - Scrum
  - Agile Prozesse – Pro und Kontra
- Lernziele

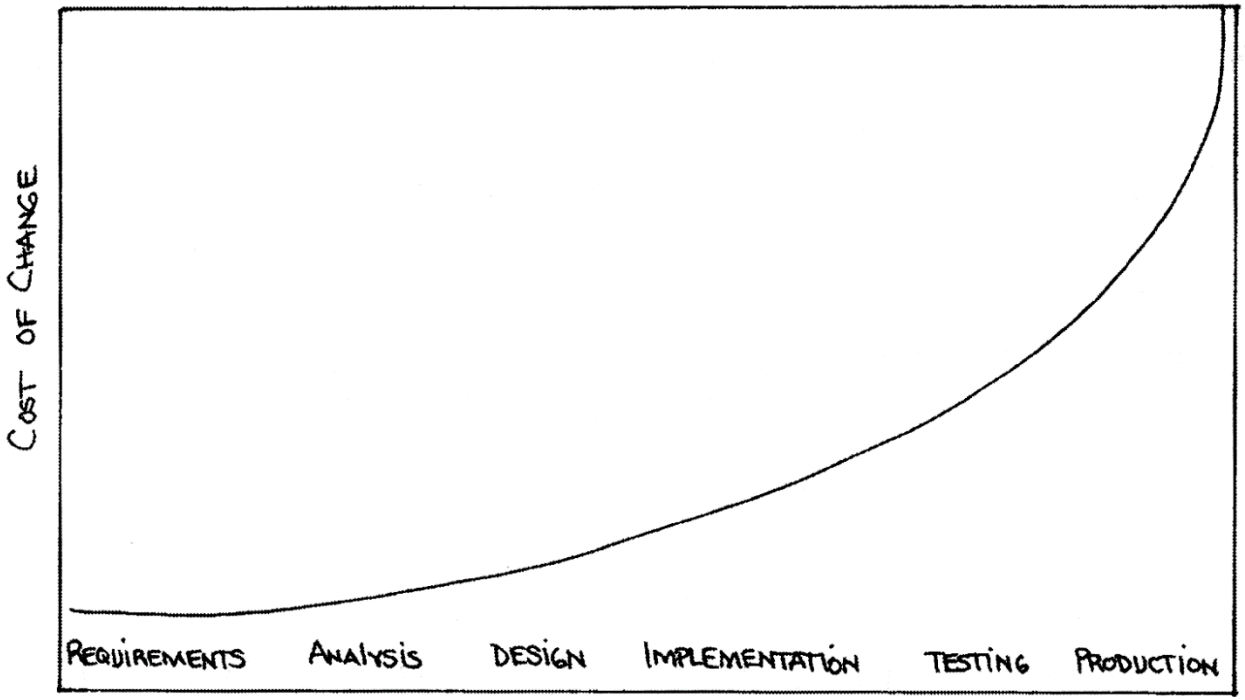
Schwergewichtige Software-Entwicklungsprozesse

- Viel Planung:
  - Umfassende Anforderungsdefinitionen, Spezifikationen, Meilensteinpläne usw., die vor Beginn der Implementierung angefertigt werden.

- Aber: Planung stimmt mit Wirklichkeit doch nie überein!
  - Oft sind die Anforderungen nur scheinbar klar und ändern sich noch im Laufe des Projekts.
  - Häufig ergeben sich unerwartet Verzögerungen.

Schwergewichtige Software-Entwicklungsprozesse

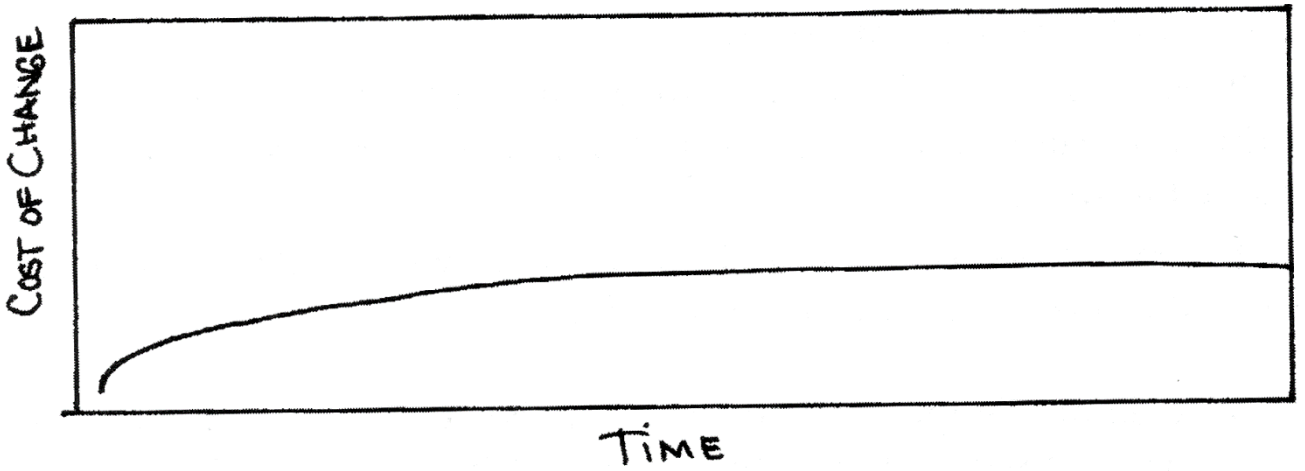
- Jahrzehnte alter Erfahrungswert:
  - Aufwand & Kosten zur Änderung eines Programms steigen exponentiell mit der Zeit (Entwicklungsphase).
    - Weil bei Änderungen in später Phase auch alle Planungsdokumente aus früheren Phasen angepasst werden müssen. ⇒ Früh viel & gut planen!



Quelle: Kent Beck: *Extreme Programming Explained.* Addison-Wesley, 1999

Was wäre wenn…
…die Aufwandskurve flach wäre?

- Dann wären späte Änderungen nicht teurer als frühe!



Quelle: Kent Beck: *Extreme Programming Explained.* Addison-Wesley, 1999

- Möglichkeit, um flache Aufwandskurve zu erreichen: Ballast abwerfen, z.B.
  - Änderungen systematisch durchführen,
  - Tests, die nach jeder Änderung durchgeführt werden müssen, automatisch statt manuell ausführen,
  - Statt Planungsdokumente anzupassen, diese soweit wie möglich reduzieren.
    - Z.B. statt Anforderungsdefinitionen & Spezifikationen für Kunden: regelmäßig dem Kunden Zwischenversionen der Software vorführen.

Agile Software-Entwicklungsprozesse

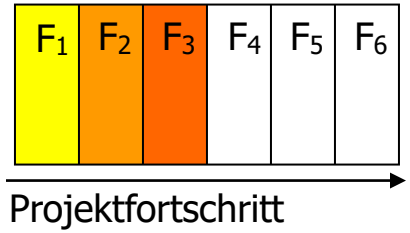
- … können auch nicht zaubern, aber:
  - Haben den Anspruch, gut mit Änderungen und vagen Anforderungen umgehen zu können.
  - Soviel Planung wie nötig, so wenig Planung wie möglich.
    - Agil bedeutet aber nicht: „einfach drauf los programmieren”!
  - Mündliche statt schriftliche Kommunikation.
    - Keine schwergewichtigen Dokumente, die aktualisiert werden müssen.
      - Der Quelltext (inkl. automatisierte Tests) ist das zentrale Dokument.
    - Das gesamte Team arbeitet zusammen in einem Raum.

Agile Software-Entwicklungsprozesse

- Häufiges Ausliefern von lauffähigen Versionen („Release”).
  - Frühzeitig lauffähige Software durch inkrementelle Entwicklung.

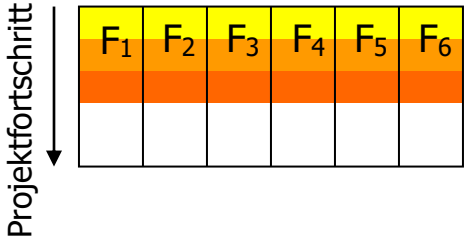
F_i=Funktionalität i

Inkrementell:



⇒ Bereits 1. Inkrement lauffähig.

Nicht-inkrementell:



⇒ System erst am Ende lauffähig.

  - Ständige Einbeziehung des Kunden.
    - Kunde erlebt keine bösen Überraschungen.
- Bekannteste agile Software-Entwicklungsprozesse:
  - Extreme Programming, Scrum.

Inhalt

- V-Modell und V-Modell XT
- Unified Process
- Agile Prozesse
  - **Extreme Programming (XP)**
  - Scrum
  - Agile Prozesse – Pro und Kontra
- Lernziele

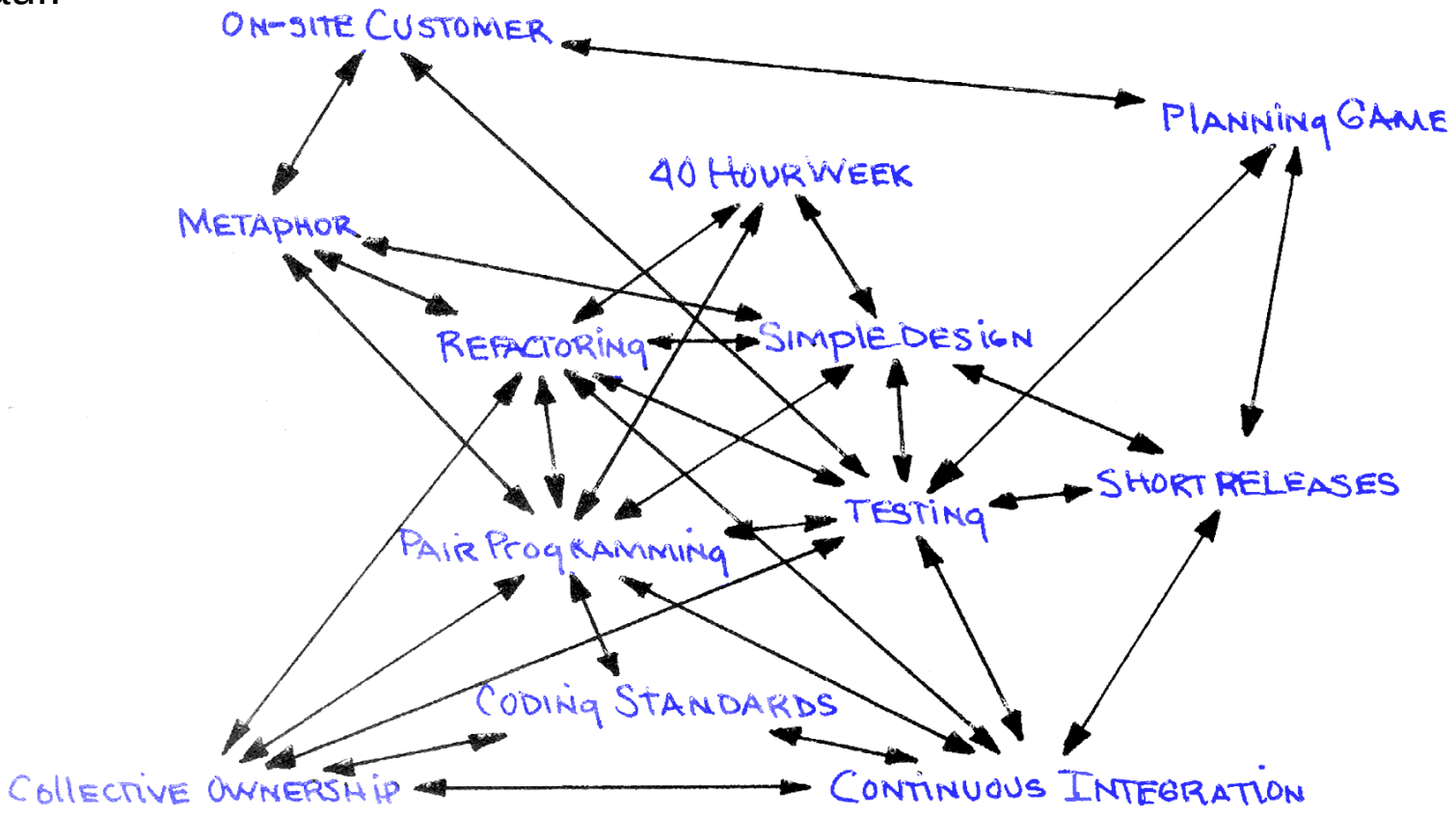
Extreme Programming (XP)

- 1996 entwickelt und erstmals eingesetzt.
  - Einzelne XP-Techniken sind jedoch älter, aber zuvor noch nie aufeinander abgestimmt eingesetzt worden.

- Hat nichts mit extremer Risikobereitschaft zu tun, sondern damit, dass als gut erkannte Prinzipien „bis zum Extrem" gesteigert werden:
  - Wenn Testen gut ist, wird fortlaufend getestet.
  - Wenn Code Reviews, bei denen Quelltext von jemand anderem als dem Autor überprüft wird, gut sind, wird in Paaren programmiert, um kontinuierliche Code Reviews zu erreichen.

Die XP-Techniken – Überblick

- Die 12 Techniken ergänzen sich bzw. fangen die Schwächen der anderen auf:



Ausgewählte XP-Techniken

- **Planungsspiel**
  - Kunde beschreibt Funktionalitäten anhand von „User Stories”.
  - Das Team schätzt Implementierungsaufwand von Funktionalitäten, Kunde priorisiert „User Stories”.
- **Testen**
  - Automatisierte Tests=Ausführbare Spezifikation, d.h. Test wird entwickelt, bevor Implementiert wird (Testgetriebene Entwicklung).
  - Tests als Sicherheitsnetz bei Änderungen.
- **Einfaches Design**
  - Keine (falschen) Annahmen über die Zukunft.

Ausgewählte XP-Techniken

- **Refactoring**
  - Quelltext wird laufend systematisch „aufgeräumt”,
    d.h. die interne Struktur ohne Änderung der Funktionalität verbessert.
  - Sorgt dafür, dass Design einfach bleibt.
- **Programmieren in Paaren**
  - Kontinuierliches Code-Review erhöht Code-Qualität.
  - Dynamisch wechselnde Paarkonstellation sorgt für Verbreitung von Wissen.
- **Gemeinsame Verantwortlichkeit**
  - Jeder darf jeden Quelltext ändern.
  - Tests als Sicherheitsnetz.
  - Man muss nicht warten, bis „Besitzer” eines Quelltexts Fehler darin behebt.
  - Verbreitung von Wissen.

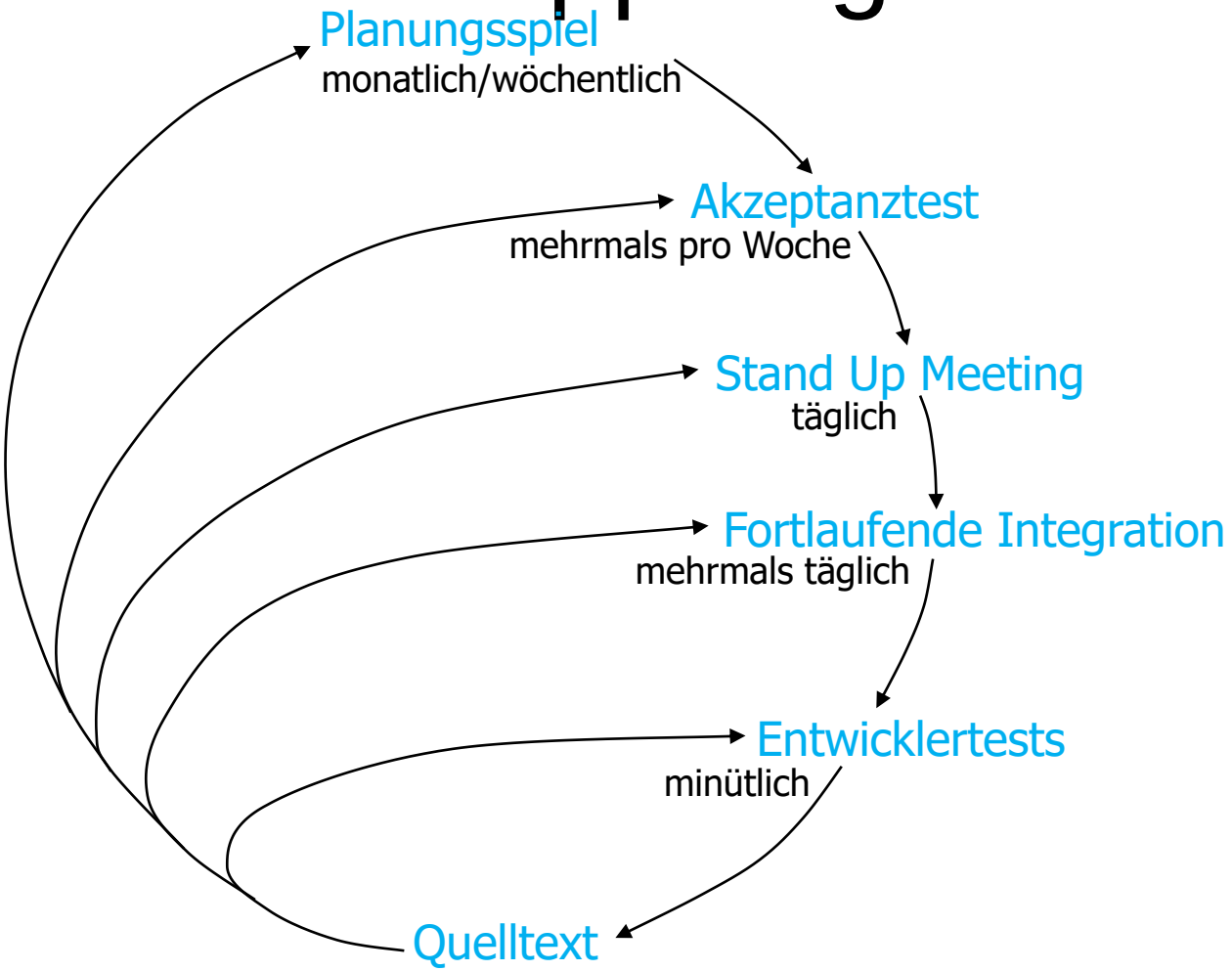
Ausgewählte XP-Techniken

- **Fortlaufende Integration**
  - Sobald alle Tests innerhalb der lokalen Arbeitsversion eines Paares laufen, werden die Änderungen in den zentralen Quelltext integriert.
  - Man stellt sofort fest, wenn Module nicht mehr zusammenpassen.

- **Nachhaltiges Tempo**
  - Keine Überstunden! Unter Zeitdruck lässt man gerne Qualität außer acht.
  - Bei Termindruck ist etwas anderes schief gelaufen: Ursache suchen!
  - Notfalls in neuer Planungsspiel-Runde Funktionalität einschränken.

Der eigentliche Prozess: Planung und Rückkopplung

- Ständige Rückkopplung ermöglicht schnell gegenzusteuern und Planung anzupassen.



Inhalt

- V-Modell und V-Modell XT
- Unified Process
- Agile Prozesse
  - Extreme Programming (XP)
  - **Scrum**
  - Agile Prozesse – Pro und Kontra
- Lernziele

Scrum – Historie

- **1986**: Hirotaka Takeuchi und Ikujiro Nonaka führen den Begriff „Scrum“ im Kontext der Produktentwicklung ein
- **Frühe 1990er**: Ken Schwaber und Jeff Sutherland, John Scumniotales, Jeff McKenna entwickelten parallel einen Ansatz den sie Scrum nannten (ähnlich zum heutigen Scrum)
- **1995**: Sutherland und Schwaber präsentierten ihren Ansatz auf der OOPSLA und starteten eine gemeinsame Kollaboration
- **2001**: Schwaber und Mike Beedle schreiben das Buch „Agile Software Development with Scrum“
- **2002**: Schwaber et al. gründet die „Scrum Alliance“ (*Certified Scrum*)
- **2009**: Schwaber et al. gründet „**scrum.org**“ (*Professional Scrum*)
- **Seit 2010**: „The Scrum Guide“ ist öffentlich

Scrum

- Aufteilung
  - von **Personen**: in kleine, funktionsübergreifende, selbstorganisierte Teams
  - von **Arbeit**: in konkrete Auslieferungspakete, Sortierung nach Priorität
  - von **Zeit**: in kurze Iterationen fester Länge (1-4 Wochen), auslieferungsfähiger Code nach jeder Iteration

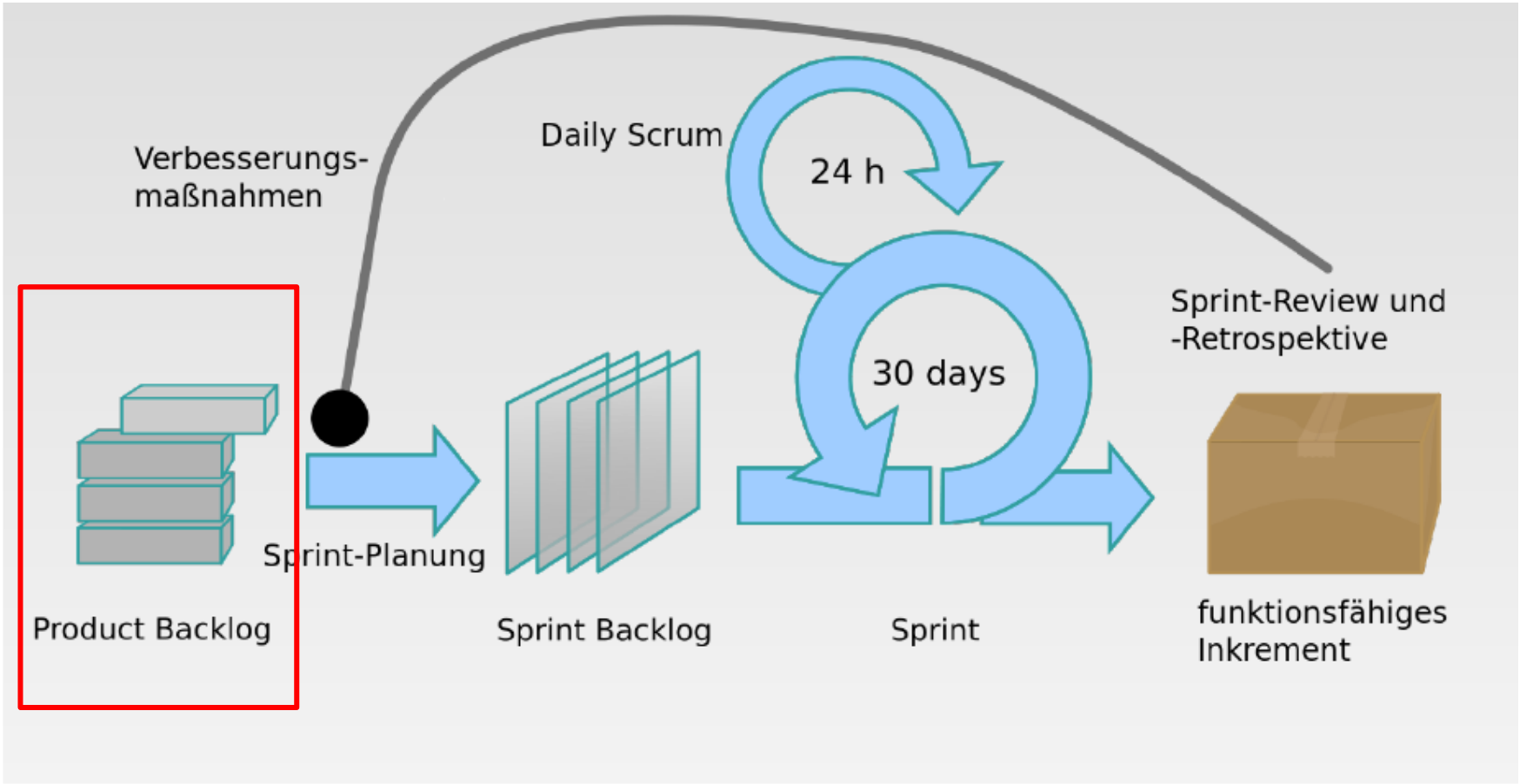
Scrum-Rollen (1)

- Product Owner
  - Anforderungsbeschreibung und –management
    - Entwickeln und Verwalten des Product Backlogs
    - Priorisierung der Features
  - Releasemanagement
  - Stakeholder-Management
- Team
  - autonom, selbstorganisiert
    - legt Arbeitsschritte und –organization selbst fest
    - entscheidet über Anzahl umzusetzender Anforderungen in einem Inkrement
  - Verschiedene Kompetenzen

Scrum-Rollen (2)

- Scrum-Master (vom Team bestimmt)
  - Unterstützung des Teams
  - Einführung Scrum Regeln (Coach)
  - Sicherstellung der Zusammenarbeit von Team und Product Owner
  - Beseitigung von Hindernissen
  - Verbesserung von Entwicklungspraktiken
  - Verhindern von Anti-Patterns in der Arbeit mit Scrum

Scrum-Prozess



Product-Backlog

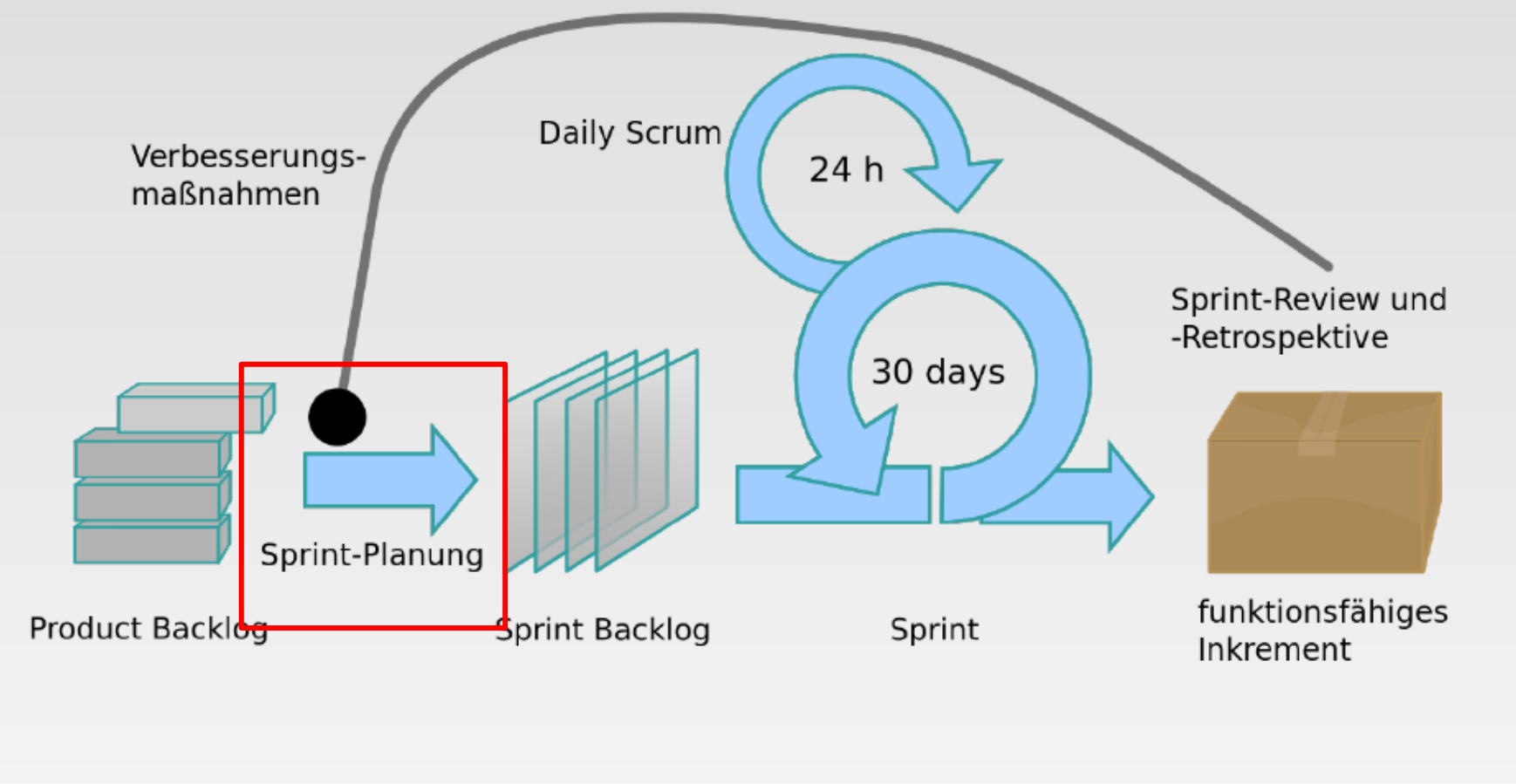
- Kern von Scrum
  - Enthält alle bekannten funktionale und nichtfunktionalen Anforderungen (sortiert nach Priorität) → Meist in Form von User Stories
  - Beinhaltet weitere Arbeitsergebnisse (z.B. Aufsetzen der Test- und Entwicklungsumgebung)
- Keine Aktivitäten
- Pflege durch Product Owner

Product-Backlog

Beispiel

PRODUCT BACKLOG (example)					
ID	**Name**	**Imp**	**Est**	**How to demo**	**Notes**
1	Deposit	30	5	Log in, open deposit page, deposit €10, go to my balance page and check that it has increased by €10.	Need a UML sequence diagram. No need to worry about encryption for now.
2	See your own transaction history	10	8	Log in, click on "transactions". Do a deposit. Go back to transactions, check that the new deposit shows up.	Use paging to avoid large DB queries. Design similar to view users page.

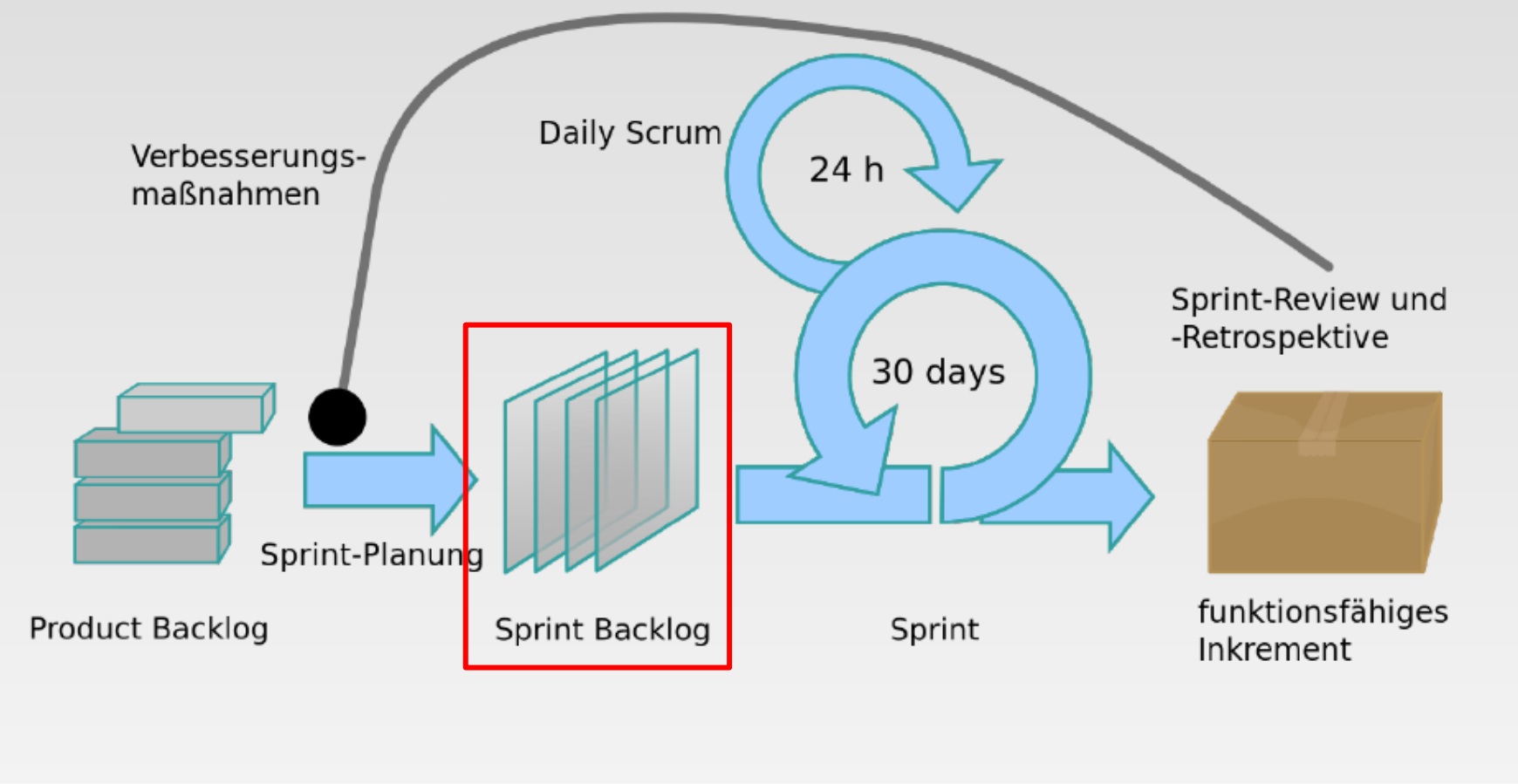
Scrum-Prozess



Sprint Planung

- Alle Rollen sind erforderlich
- Findet zu Beginn eines jeden Sprints statt
- Beantwortung von 2 Fragen:
  - Was soll im kommenden Sprint entwickelt werden?
  - Welche Aufgaben sind zur Lieferung des Product-Backlog-Eintrags nötig?
- Ergebnis: Sprint-Backlog

Scrum-Prozess



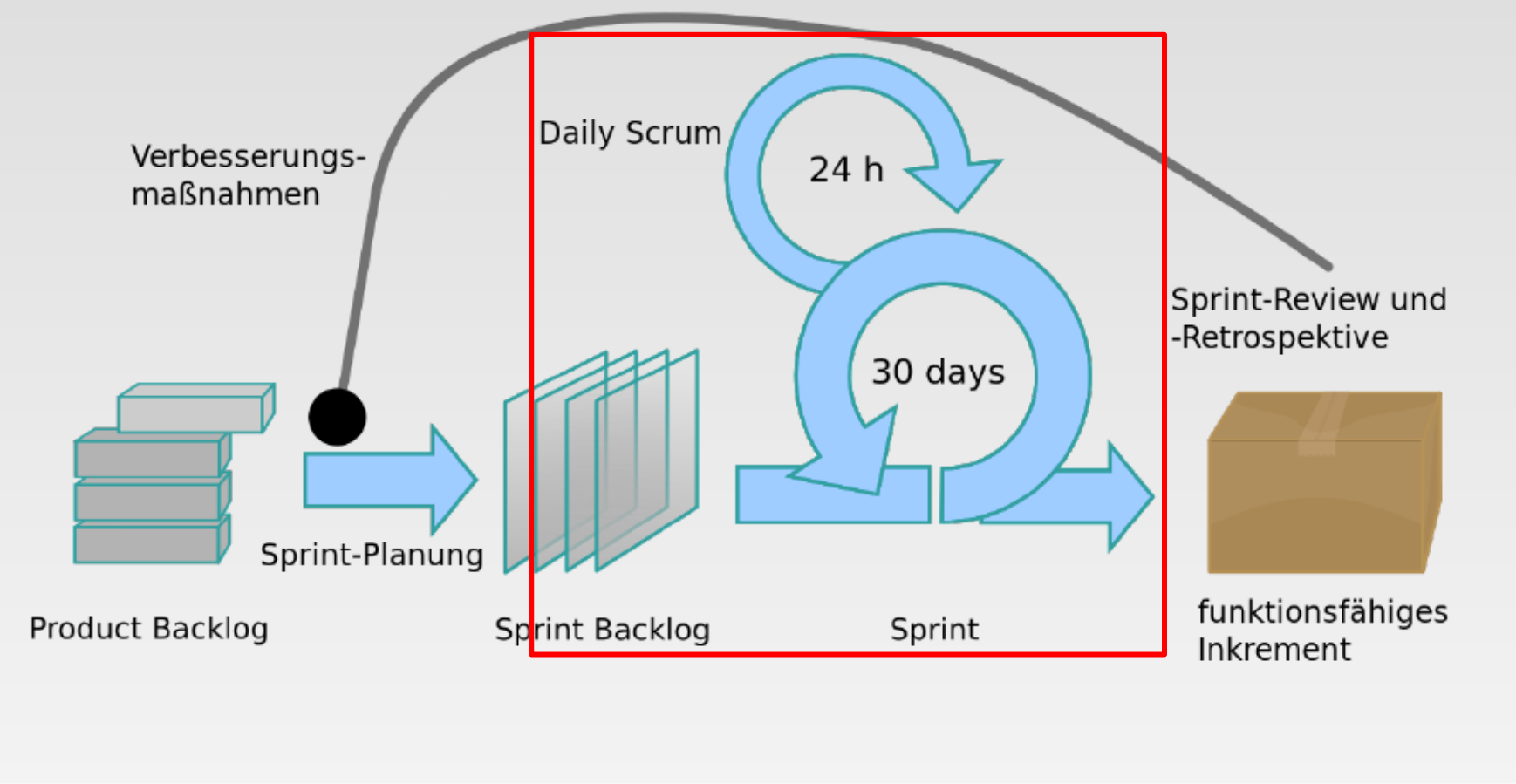
Sprint Backlog

- Liste an Features (User Stories) die in einem Sprint entwickelt werden sollen
- Sortiert nach Priorität
- Enthält auch Aufgaben (Tasks), welche vom Team zur User Story erstellt wurden
- Kann auch Abschätzungen bezüglich des Aufwandes für Tasks beinhalten.

Sprint Backlog
Beispiel

User Story	Tasks	Day 1	Day 2	Day 3	Day 4	Day 5	...
As a member, I can read profiles of other members so that I can find someone to date.	Code the ...	8	4	8	0		
	Design the ...	16	12	10	4		
	Meet with Mary about ...	8	16	16	11		
	Design the UI	12	6	0	0		
	Automate tests ...	4	4	1	0		
	Code the other ...	8	8	8	8		
As a member, I can update my billing information.	Update security tests	6	6	4	0		
	Design a solution to ...	12	6	0	0		
	Write test plan	8	8	4	0		
	Automate tests ...	12	12	10	6		
	Code the ...	8	8	8	4		

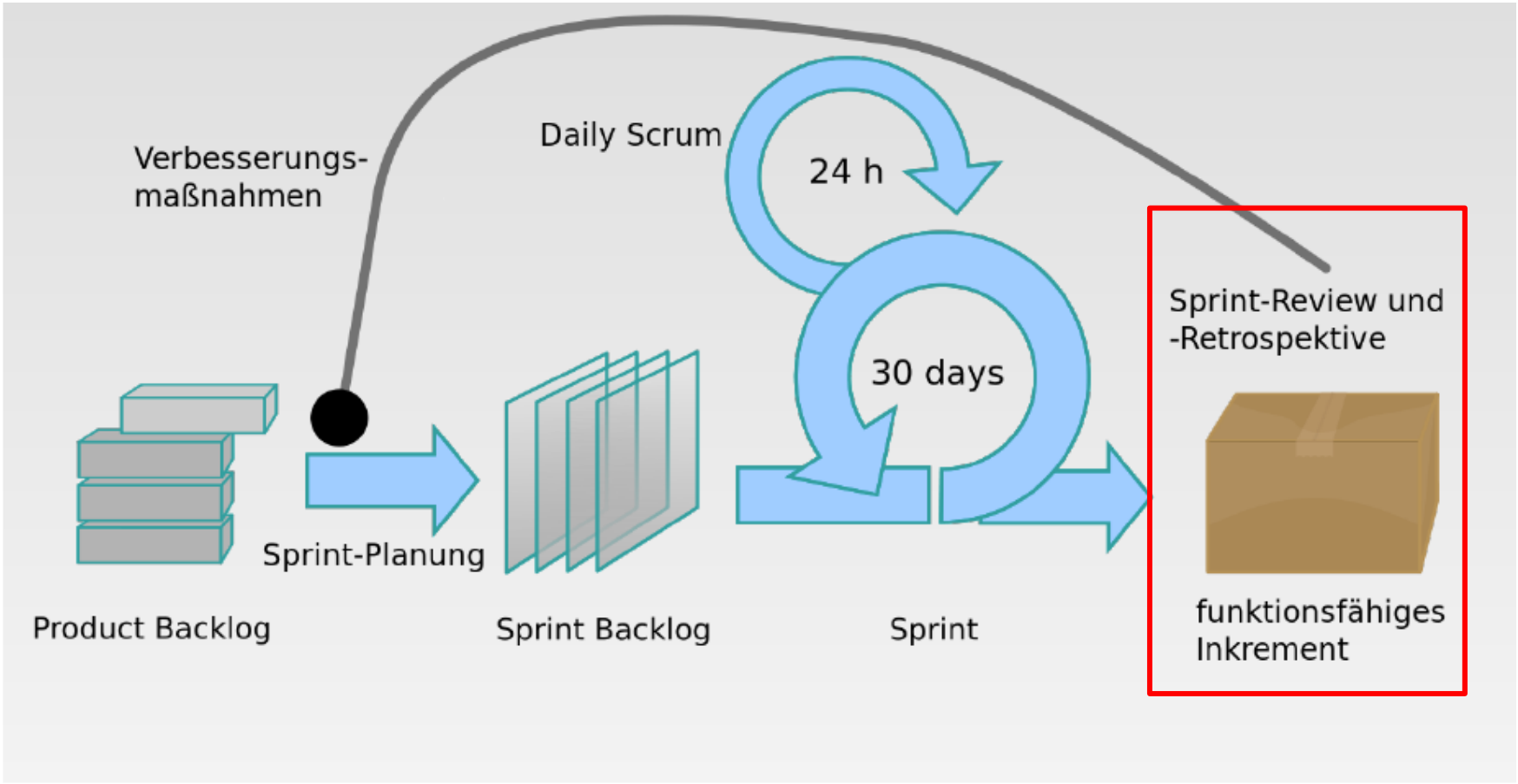
Scrum-Prozess



Sprint und Daily Scrum

- Sprint
  - Abarbeiten der Einträge im Sprint-Backlog
- Daily Scrum
  - Tägliches **kurzes** Treffen des Entwicklerteams
  - Oftmals Product Owner und Scrum Master anwesend
  - Jeder berichtet von folgenden Dingen
    - Was habe ich gestern erledigt?
    - Woran werde ich heute arbeiten?
    - Bin ich durch irgendetwas blockiert?

Scrum-Prozess



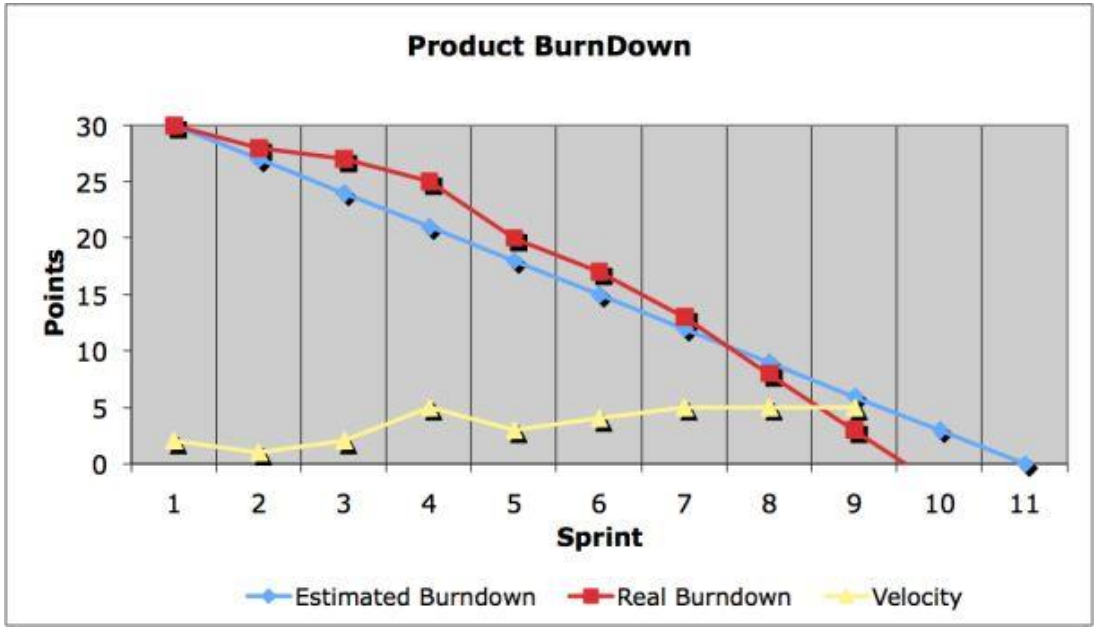
Sprint Review

- Alle Rollen anwesend
- Check ob Ziel mit dem Inkrement erreicht wurde am Ende eines Sprints
  - Demovorführung
- Wichtig: Kunde / Anwender sollten anwesend sein!

Sprint Retrospektive

- Alle Rollen anwesend
- Team überprüft Arbeitsweise (Effizienz) am Ende einer Iteration
- Dokumentation und Planung von Verbesserungsmaßnahmen

Burn-Down Charts



- Velocity: #Komplettierte Story Points
- Nur komplettierte Stories zählen

Scrum Kostenschätzung

Beispiel: Planning Poker

- Vor jedem Sprint wird die Komplexität der anstehenden User Stories geschätzt
- Workshop mit folgenden Regeln
  1. Product Owner präsentiert die User Story
  2. Diskussion der Story
  3. Jedes Teammitglied bekommt ein Kartenset und wählt nun die Karte mit der geschätzten Komplexität und legt sie auf den Tisch
  4. Alle Team-Mitglieder drehen die Karten simultan um
  5. Höchste und niedrigste Bewertungen werden erläutert
  6. Rückfragen und kurze Diskussion
  7. Erneutes Schätzen
  8. Schritte 3 bis 7 werden max. 3 mal wiederholt, danach: Mittelwert (oder höchster)

Scrum Kostenschätzung

Beispiel: Planning Poker

Punkteskala (Fibonacci-Reihe):

0	kein Aufwand
1	sehr kleiner Aufwand
2	kleiner Aufwand = 2 × sehr kleiner Aufwand
3	mittlerer Aufwand = sehr kleiner + kleiner Aufwand
5	großer Aufwand = kleiner + mittlerer Aufwand
8	sehr großer Aufwand = mittlerer + großer Aufwand
13	riesiger Aufwand = großer + sehr großer Aufwand

Inhalt

- V-Modell und V-Modell XT
- Unified Process
- Agile Prozesse
  - Extreme Programming (XP)
  - Scrum
  - **Agile Prozesse – Pro und Kontra**
- Lernziele

Agile Entwicklungsprozesse – Pro und Kontra

- Agile Software-Entwicklungsprozesse, z.B. Extreme Programming, Scrum:
  - Besonders geeignet für:
    - Vage, sich ändernde Anforderungen.
  - Weniger geeignet für:
    - Große Projekte (>12 Entwickler).
      - Mündliche Kommunikation skaliert nicht beliebig.
    - Sicherheitskritische Systeme, die nachprüfbare Spezifikationen erfordern.

Inhalt

- V-Modell und V-Modell XT
- Unified Process
- Agile Prozesse
  - Extreme Programming (XP)
  - Scrum
  - Agile Prozesse – Pro und Kontra
- **Lernziele**

Lernziele

- Wie stehen Vorgehens- und Prozessmodelle zueinander?
- Welche Prozessmodelle gibt es und wie unterscheiden sie sich?
- Welche Vor- und Nachteile bieten die verschiedenen Prozessmodelle?
- Worin unterscheiden sich Agile und Schwergewichtigen Prozessmodelle?