Kapitel 2
Lineare Prozessmodelle

1 Einführung

Die **zentrale Fragestellung** für das Prozessmodell ist, wie man zweckmäßigerweise bei der Erstellung einer Software vorgeht.

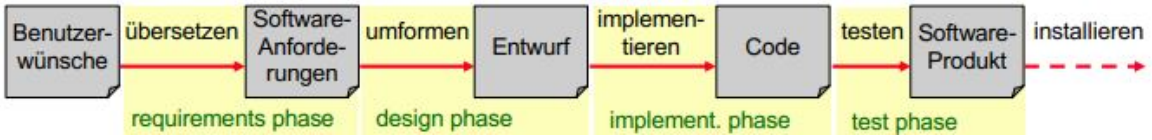
Lebenszyklus:
Idee/Wunsche ⇒ Entwicklung ⇒ Gebrauch ⇒ Wartung ⇒ Ausmusterung

Entwicklungszyklus:
Entwurf ⇒ Realisierung von Komponenten ⇒ Zusammebau ⇒ Prüfung

2 Prozess und Preozessmodelle

Ein **Prozess** ist eine Sequenz von Schritten, die für einen bestimmten Zweck ausgeführt werden.

Beispiel eines Software-Entwicklungsprozess / Prozessmodell:



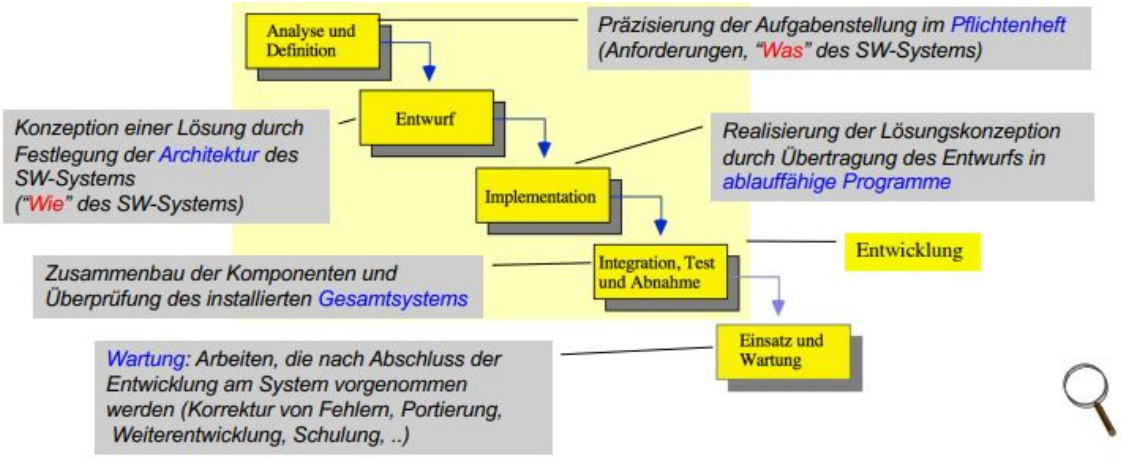
Ein **Prozessmodell** ist eine Muster-Struktur für Organisation und Durchführung eines Software-Entwicklungs-Prozesses.
Bei einem Softwareentwurf geht man (noch immer gängig) mit dem *Code and fix* Modell vor, also Codierung und Korrektur im Wechsel mit Ad-hoc-Tests als einzige bewusste Tätigkeiten der Software-Entwicklung. Dies ist allerdings **kein** Prozessmodell.
Nach IEEE-Standart ist der Inhalt eines Prozessmodells folgendermaßen:

- Reihenfolge des Arbeitsablaufs
- Durchzuführende Aktivitäten
- Definition der Teilprodukte
- Verantwortlichkeiten, Rollenverteilung und Kompetenzen
- Notationen, Vorgehensweisen und Werkzeuge
- Struktur und Merkmale der Dokumente
- Prüf- und Fertigstellungskriterien

3 Lineare Vorgehensweisen

Aktivitäten-orientierte Prozessmodelle - Wasserfallmodell

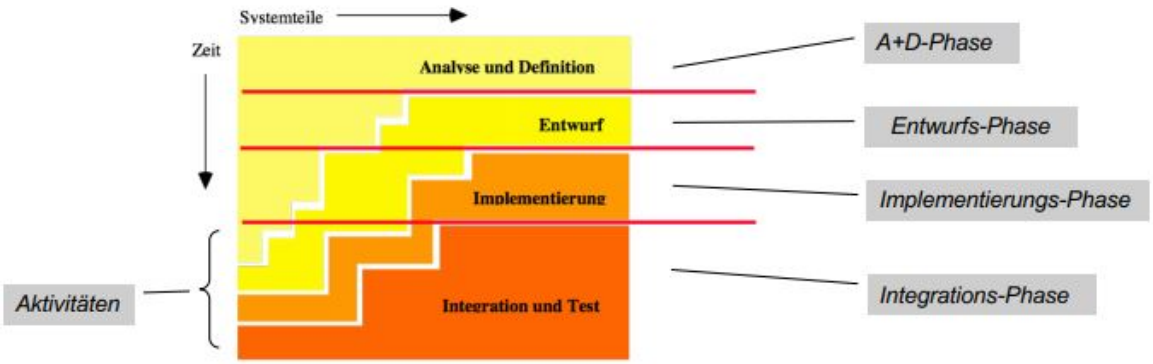
Das Wasserfallmodell ist ein streng sequentielles „Einbahnstraßenmodell“ und ist **Aktivitätenorientiert**, statt nach Zielen oder Ergebnissen.



Vorteile	Nachteile
Erstellung der Software in kleinen, überschaubaren, wohldefinierten Schritten	Idealvorstellung, die nicht der Realität entspricht
Projektüberwachung und Fortschrittskontrolle an festen Messpunkten	SW-Entwicklung verläuft meist nicht linear, Prüfung am Ende zu spät

Ergebnis-orientierte Prozessmodelle - Phasenmodell

Beim Phasenmodell wird der Entwicklungsprozess in Abschnitte („Phasen“) gegliedert mit definierten Meilensteinen, welche durch inhaltliche Kriterien und nicht durch Zeitpunkte definiert sind.
Eine Phase ist hierbei ein Intervall zwischen zwei Meilensteinen. Phasen sind charakterisiert durch erzielte Ergebnisse, und nicht durch Aktivitäten (wesentlicher Unterschied zum Wasserfallmodell).

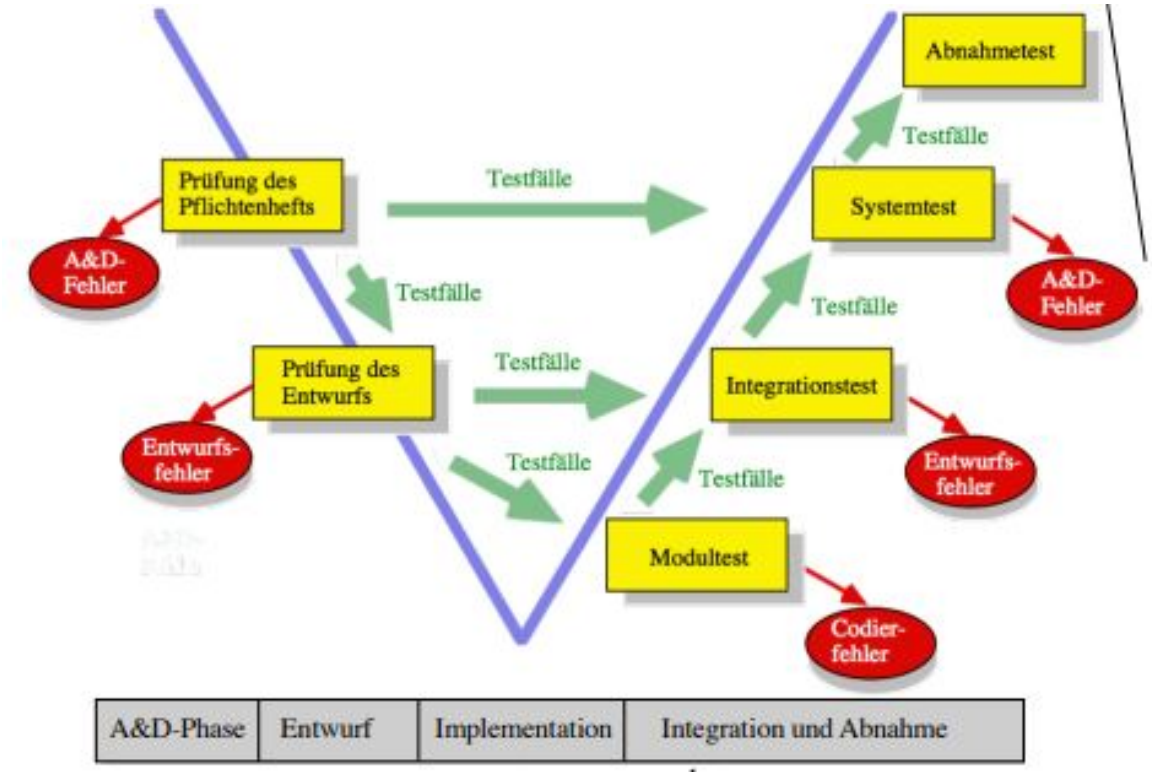


Hierbei werden die Phasen sequentiell durchlaufen, wobei eine Rückkehr zu einer früheren Phase nicht vogesehen ist. Die Phasenbezeichnung gibt jeweils nur die Haupttätigkeit an.

Vorteile:
Klar organisierter Ablauf (wie beim Wasserfallmodell) und eine relativ einfache Projektführung (anhand von Meilensteinen).

V-Modell

Grundidee: Beim V-Modell werden das Phasenmodell und Qualitätssicherung integriert.



Aufgebaut ist das V-Modell folgendermaßen:

- Von links nach rechts erstreckt sich die Zeitachse
- Von oben nach unten erstreckt sich der Grad der Detaillierung
- Die einzelnen Phasen sind unterhalb des „V“ aufgelistet.
- In der Spitze erfolgt die Implementierung, die anschließend auf der rechten Seite gegen die entsprechenden Spezifikationen der linken Seite getestet wird

Die Wesentlichen Erkenntnisse dieses Modells belaufen sich auf die verschiedenen Abstraktionsebenen des Entwicklungsprozesses, und, dass Fehler im Allgemeinen nur auf der Ebene gefunden werden, auf der sie gemacht wurde.

V-Modell 97

Das V-Modell 97 ist eine Weiterentwicklung der Grundidee und ist nicht nur für Software, sondern auch für software-gestützte Systeme. Weitere Weiterentwicklungen:

- Verschiedene Aspekte der Standardisierung
- Neben Entwicklung und Qualitätssicherung (QS) auch weitere Submodelle mit definierten Zusammenhänge zwischen den Submodellen.
- verschiedene Abstraktionsebenen

Außerdem werden *Aktivitäten*, *Produkte* und *Zustände* des Entwicklungsprozesses festgelegt.

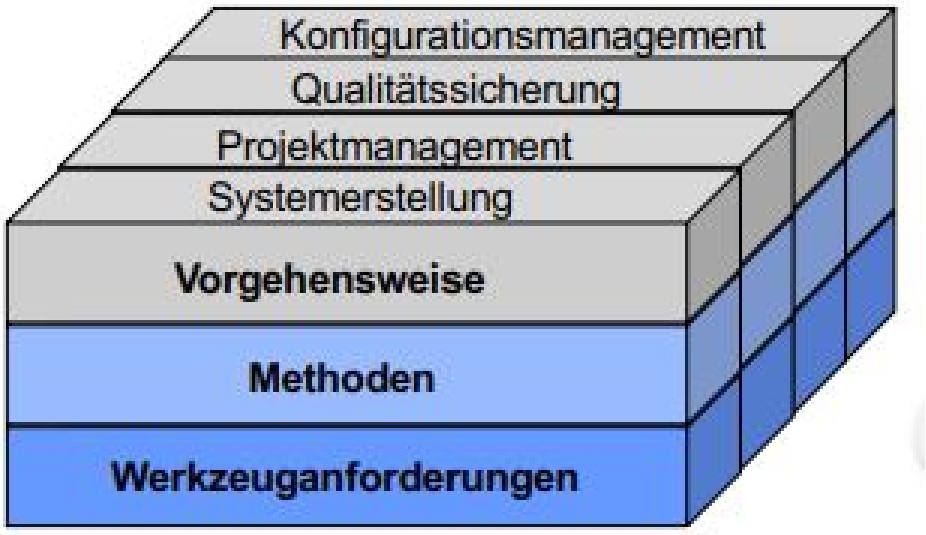
Standardisierung:
Ziel der Standardisierung ist die bessere Planbarkeit von IT-Projekten sowie die Verbesserung der Qualität und Kommunikation zwischen Auftraggeber und -nehmer.

Beim V-Modell 97 gibt es 4 Submodelle:

- Systemerstellung (SE)
- Projektmanagement (PM)
- Qualitätssicherung (QS)
- Konfigurationsmanagement (KM)

Für jedes dieser Submodelle gibt es damit 3 Aspekte:

- Vorgehensweise: **Was** ist zu tun?
- Methoden: **Wie** ist etwas zu tun?
- Werkzeuganforderungen: **Womit** ist etwas zu tun?



V-Modell XT

Das V-Modell XT ist ein Produktzentrierter Ansatz („Integriertes Produktmodell“) und wird gegliedert in

- **Produkte**: stehen im Mittelpunkt, haben definierte Abhängigkeiten und sind (Zwischen-)Ergebnisse eines Projekts
- **Aktivitäten**: dienen dazu, um Produkte zu erzeugen oder zu bearbeiten
- **Rollen**: beschreiben zusammengehörige Aufgaben, Verantwortlichkeit und benötigte Fähigkeiten

Submodelle, Abstraktionsebenen und Erzeugnisstruktur im Wesentlichen wie bei V-Modell 97.

Unterschiede von V-Modell XT zu V-Modell 97:

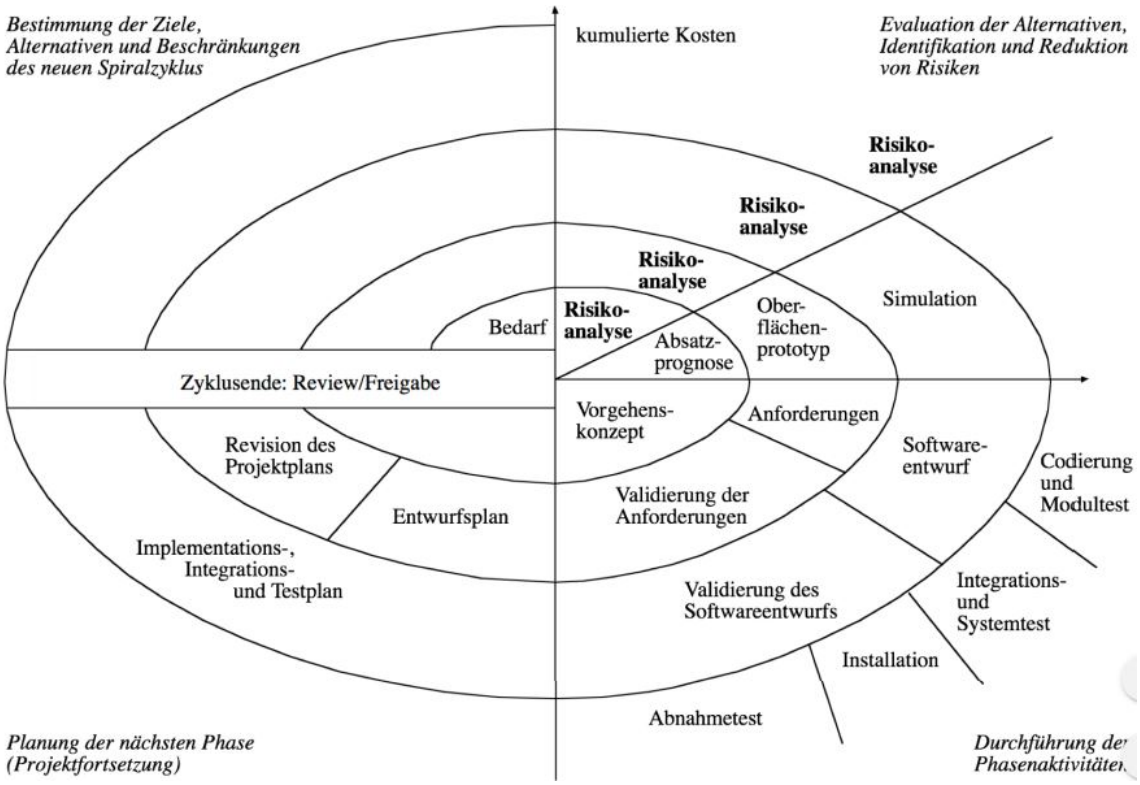
- Legt Rahmenvorgaben fest und bietet dadurch größere *Flexibilität* (bzgl. konkreter Vorgehensweise), zum Beispiel
  - Inkrementell
  - Komponentenbasiert
  - Agil
- Legt für jedes Produkt eindeutig *Rollen/Verantwortlichkeiten* fest
- Unterscheidet *verschiedene Projekttypen*
- Außerdem ist das Problem- und Änderungsmanagement ein separates Submodell

Vorteile	Nachteile
Standardisierte Abwicklung von Systemerstellungsprojekten	Ohne geeignete Werkzeugunterstützung nicht handhabbar (entsprechendes Werkzeug gibt es)
Integrierte Darstellung durch SE, QS, PM und KM	Gefahr von „Software-Bürokratie“ für kleine und mittlere Systeme
Generisches Vorgehensmodell mit definierten Möglichkeiten zur Anpassung $\Rightarrow$ tailoring	
Gut geeignet für große Systeme mit langer Lebensdauer und Outsourcing	

Spiralmodell

Das Spiralmodell nach Boehm ist ein *Generisches Modell* mit Vorteilen für das Projektmanagement.
In jedem Entwicklungsschritt (= eine Spiralwindung) gibt es folgende Phasen:

- Zielbestimmung, Alternativen
- Evaluation der Alternativen, Erkennung und Reduktion vo Risiken
- Entwicklung und Validation des jeweiligen Zwischenproduktes
- Review des Zwischenproduktes und Planung der Folgeaktivität



Vorteile	Nachteile
Fehler und ungeeignete Alternativen werden frühzeitig eliminiert	Hoher Managementaufwand
Flexibler Prozessablauf	