

IT-Projekt-Management

Dr. The Anh Vuong
email: vuongtheanh@netscape.net
http: www.dr-vuong.de





Vorgehensmodell

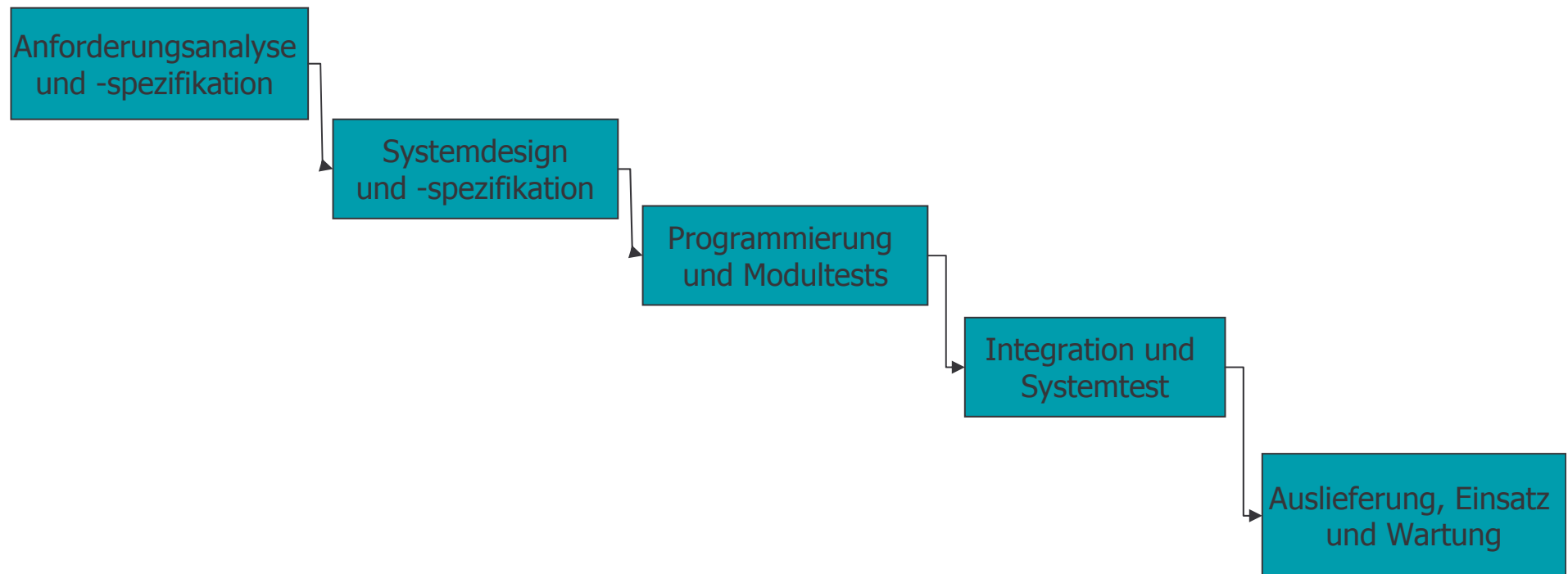


Wasserfallmodell

- 1970 Winston Royce in "Managing the Development of Large Software Systems: Concepts and Techniques."
- Jedes Phase hat ein Start- und Ende mit eindeutigen Ergebnisse Dokumente
- Die Phase des Projekts hängen hintereinander, das Projekt läuft wie der Strom eines Wasserfalls

Wasserfallmodell

- 5 Phasen -





Wasserfallmodell

- 6 Phasen -

- Planung: Erstellung des Lastenhefts, Projektkalkulation und Projektplan. (Engineering)
- Definition: Erstellung des Pflichtenhefts, Produktmodell, GUI-Modell und evtl. schon Benutzerhandbuch. (Analysis)
- Entwurf: UML, Struktogramme (Design)
- Implementierung (Coding)
- Testen (Testing)
- Einsatz und Wartung (Maintenance)



Wasserfallmodell

- Nachteile-

- Sequenz → langsam, planwirtschaft
- Phasentrennung → Nicht Flexibel
- Späte Verfügbarkeit von Produkten → Nicht Aktuell
- Nicht alle Anforderungen abschätzen und alle Lösungen vorher definieren.



V- Modell

- 1986 Bundesministerium für
Verteidigung:
Softwareentwicklungsumgebung für Informationssysteme (SEU-
IS) Softwareentwicklungsumgebung für Waffen- und
Waffeneinsatzsysteme (SEU-WS)
- 1997 Bundesverwaltung – Empfehlung
- 2005 V-Modell XT
<http://www.kbst.bund.de/Anlage306373/anlage306373.pdf>



V-Modell

- Einbindung des Auftraggebers
- Modularisierung: Bausteine , die zusammengestellt werden
- „Weg vom Wie hin zum Was“,
Konzentration auf das erzeugte Produkt.

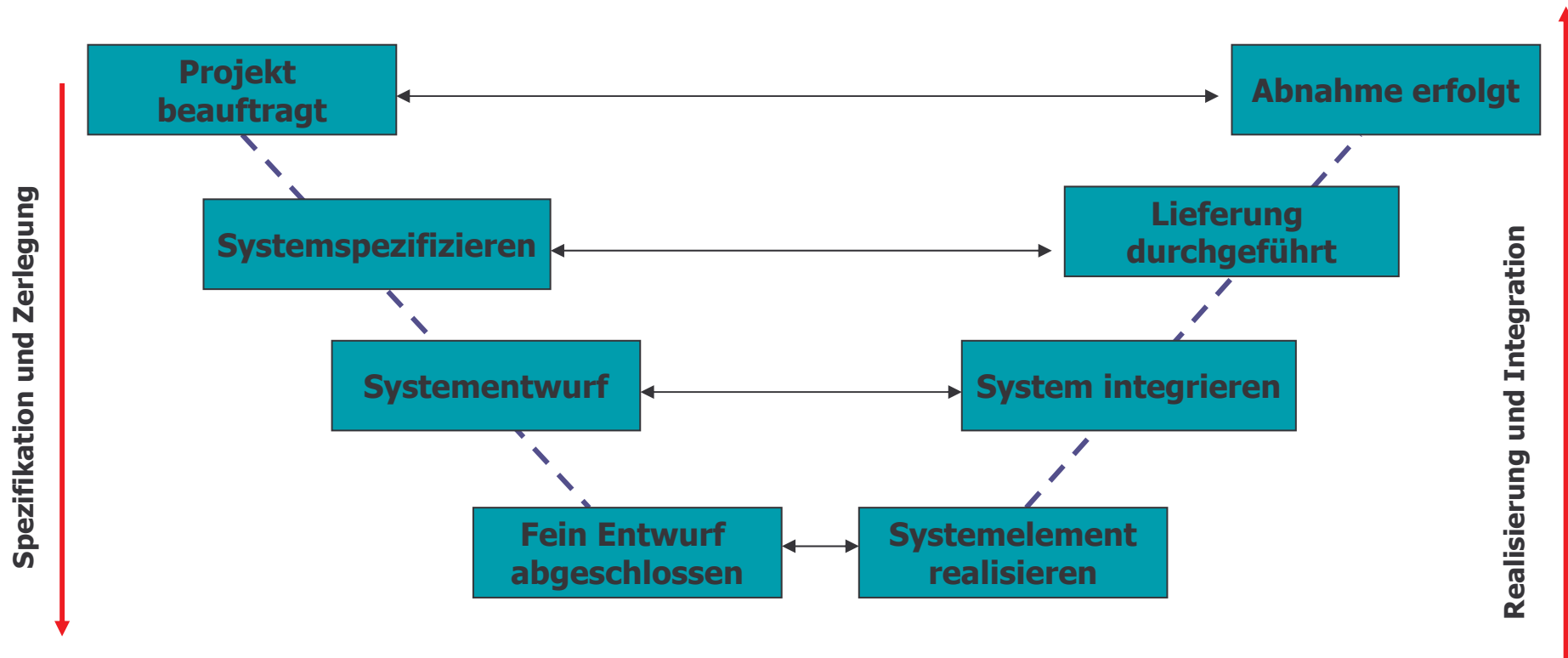


V-Modell

- Dokument in Projekt wird als Produkt bezeichnet.
- Viele Produkte, die eine inhaltliche Zusammenhang haben, bilden sich ein Produktgruppe
- Jedes definierte Produkt durchläuft vier Zustände:
 - geplant
 - in Bearbeitung
 - vorgelegt
 - akzeptiert

V-Modell

- Das V in V-Modell -



Quelle: <http://www.kbst.bund.de/Anlage306373/anlage306373.pdf>



OO Modell

-Eigenschaften-

- Planung von objekt-orientierten Projekten
- Iterativ-inkrementelle Entwicklung
- Prototyping
- Objekt-orientierten Softwareentwicklung
- Dokumentation: UML-Notation



OO Modell

-Planung-

- **Iterationsplanung**

Die Phasen im Entwicklungsprozess (Anforderungsanalyse - Design - Codierung - Test - Abnahme) werden in vielen kleinen Zeitabschnitten wiederholt.

- **Feature Entwicklung**

Modulen von einer Frame-Work werden entwickelt

- **Lauffähige Code als Ergebnismesser**

- **Kunden-Einbildung**

- **Iterative und Inkrementierte Entwicklung**



OO Modell

-Konkrete Modelle-

- Rational Unified Process (RUP)
 - [http:// www.rational.com](http://www.rational.com)
- OEP-Timepacing-Verfahren
 - <http://www.controlchaos.com/>
 - <http://www.oose.de>
- Scrum
 - <http://jeffsutherland.com/scrum/>
 - <http://www.scrumalliance.org/>
- Crystal
 - <http://www.software-kompetenz.de/?23114>
- Extreme Programming (XP)
 - <http://www.frankwestphal.de/ExtremeProgramming.html>



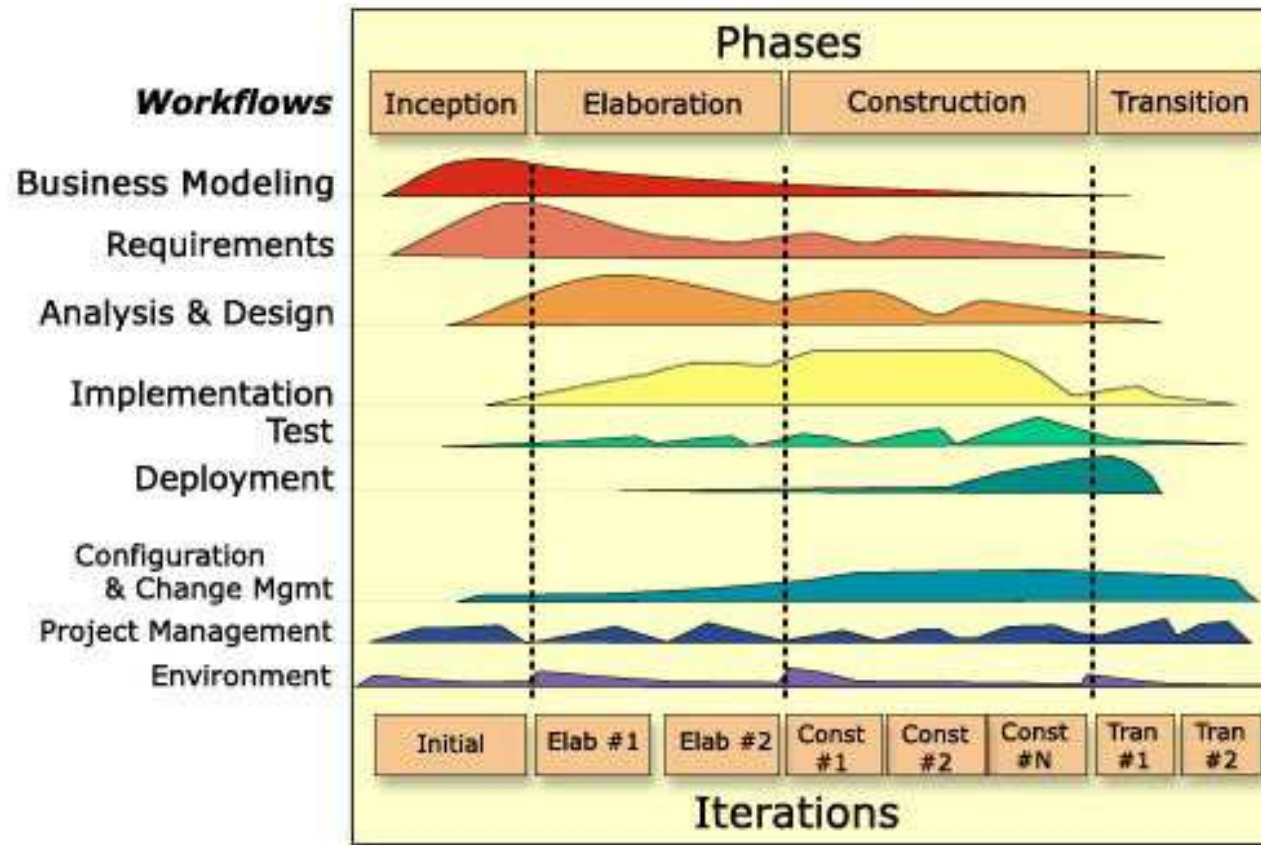
OO Modell

- Rational Unified Process (RUP)- -

- Verwendung der Unified Modelling Language (UML) als visuelle Beschreibungssprache
- Verwendung von Use Cases als Ausgangspunkt des Designs
- Verwendung der Rational Tools als Designwerkzeuge und in der Entwicklungsumgebung
- "Verschmierung" der Phasen des Wasserfallmodells

OO Modell

-Rational Unified Process (RUP)-





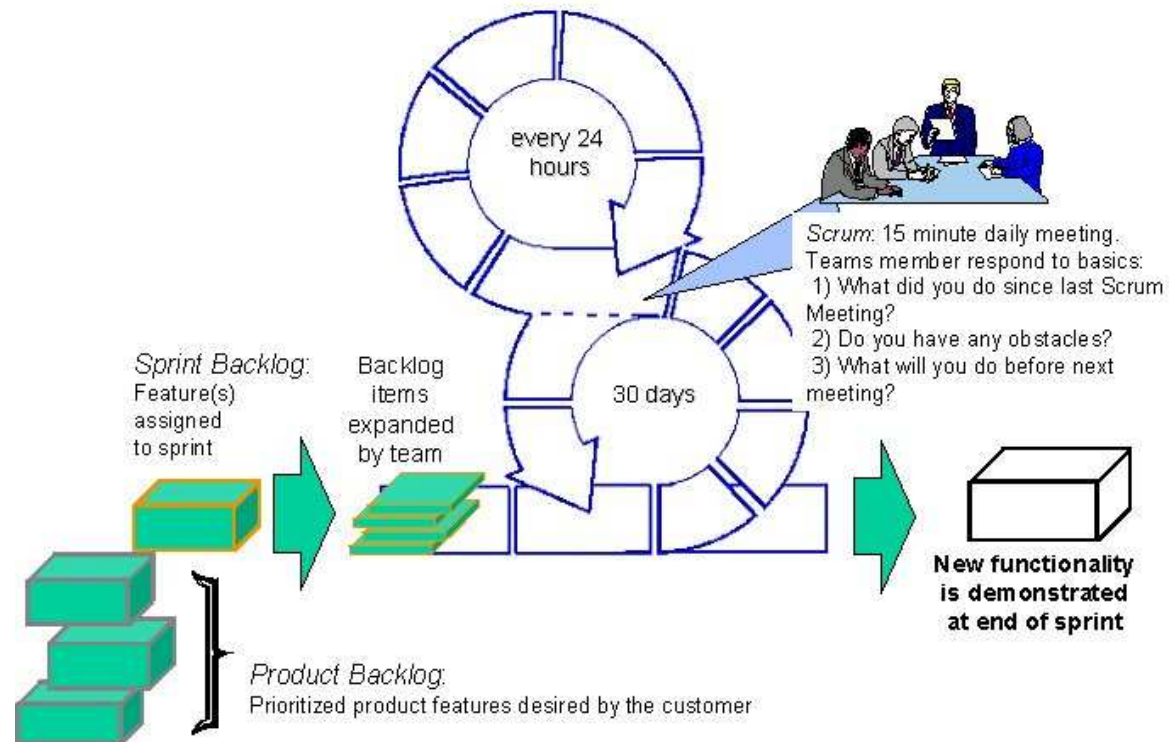
OO Modell

-OEP Timespacing -

- Planung wird bis auf Ebene der Iterationsschritte herunter gebrochen, deren konkrete Feinplanung findet dann in deren unmittelbarem Vorfeld statt
- Zu Beginn einer Iteration findet ein Kick-Off statt
- **Timeboxing**: nach einem vorher fest definierten Zeitabschnitt wird das Ergebnis ausgeliefert - egal wie! Ggf. werden also eher Abstriche am geplanten Umfang als am Zeitrahmen hingenommen.
- Die Ergebnisse werden Reviews unterzogen. Daraufhin werden die Ergebnisse konsolidiert.
- Vorteile:
 - Festen Lieferrhythmus
 - regelmäßigen Kontrollen der Fehlentwicklungen

OO Modell

-SCRUM-



Quelle: www.controlchaos.com



OO Modell

-Extrem Programming-

- Testcode wird vor dem Anwendungscode selbst implementiert
- Häufige, kleine Releases
- Ständiger Kontakt mit dem Anwender: Zweierteams aus Fachexperten und Programmierer arbeiten ständig eng zusammen ("Pair Programming")
- Ausgangspunkt des Designs sind "User Stories"
- Funktioniert gut in kleinen Teams