

Agenda

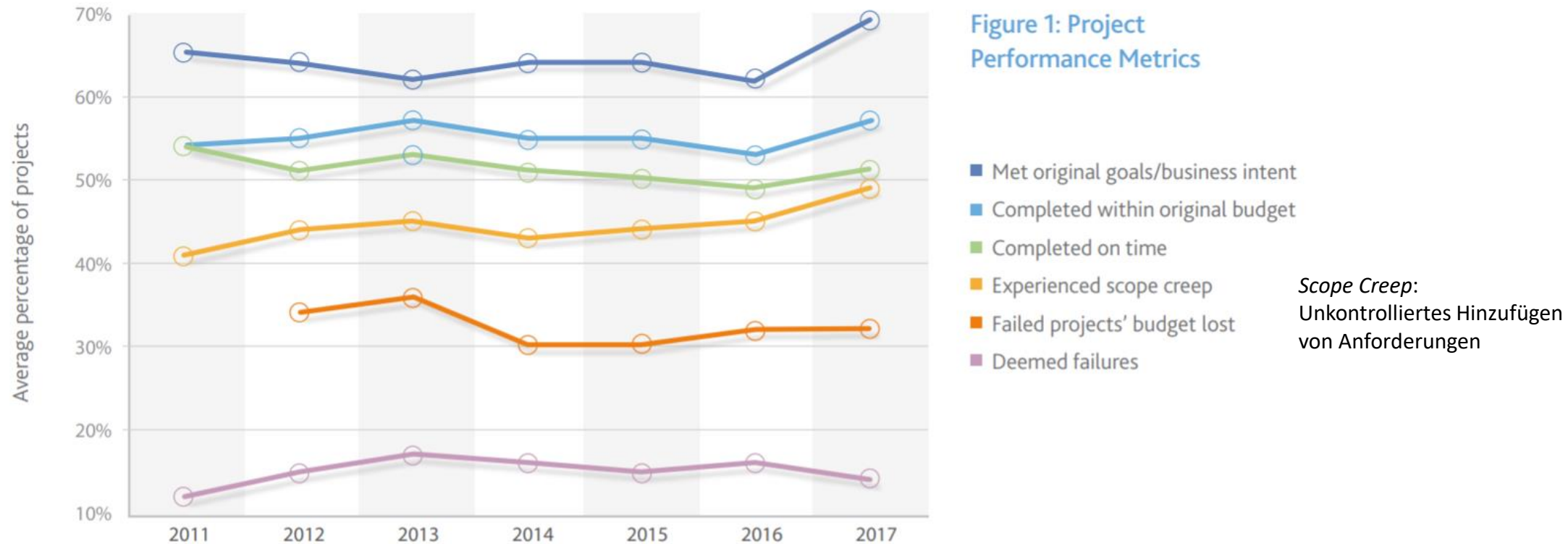
1. FAQ zur Programmierung
2. Übersicht der Themen
3. Grundlagen Softwaretheorie
4. Grundlagen Programmiersprachen
- 5. Software-Entwicklungsprozesse**
6. Software-Entwurftools

> Danach: Praktisches Programmieren (Python)

5. Software-Entwicklungsprozesse

- Problem 1: Software-Projekte können sehr komplex sein
 - Problem 2: Softwareentwicklung wird in (großen) Teams betrieben
 - Problem 3: Akteure des Software-Projekts haben verschiedene Skills, Interessen & Motivation
-
- Lösung: Systematisches Vorgehen! Aber was ist das eigentlich?
 - Planung? Gut aber nicht überplanen
 - Einfach machen lassen? Gut aber Kontrolle auch wichtig

5. Software-Entwicklungsprozesse



5. Software-Entwicklungsprozesse

- Was gehört zum Software-Entwicklungsprozess?
 - Produkte, die entwickelt werden sollen
 - Leistungen, die erbracht werden sollen
 - Software-Lebenszyklus
 - Planung, Qualitätsmanagement, Risikomanagement, Personalplanung, Teammanagement -> auch wichtig aber für Entwickler weniger relevant
- Vertragliche Bestandteile: Verständliches Profil des Projekts
 - Lastenheft: Anforderungen an die Leistungen des Auftragnehmers; Grobes Konzept aber: Überprüfbare Form !!!
 - Pflichtenheft: Arbeitsgrundlage für das Projekt; Detailliert: Welche Funktionalitäten? Welche technische Basis? Wie umgesetzt? Maßstab für zukünftige Beurteilungen -> Überprüfbare Form !!!

5. Software-Entwicklungsprozesse



Anmerkung: Grafik stellt nicht zeitlichen Aufwand dar!

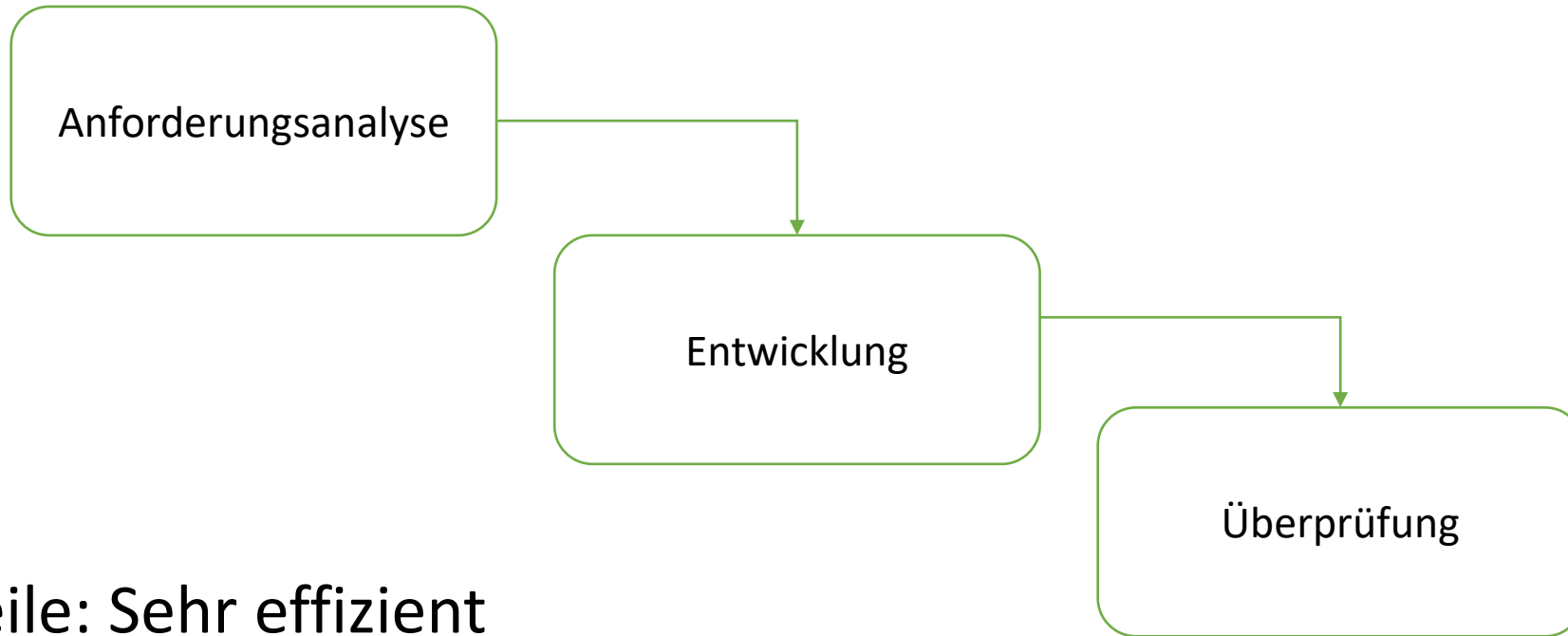
5. Software-Entwicklungsprozesse

Vorgehensmodelle

- Welche Aktivitäten in welcher Reihenfolge?
- Warum „scheitern“ Softwareprojekte?
 - Mangelnde Planung
 - Unpräzise Anforderungen
 - Schlechte Kommunikation
 - Eher selten: Mangelnde Kompetenz der Entwickler
- 2 wichtige Faktoren: Zeit und Geld

5. Software-Entwicklungsprozesse

Vorgehensmodelle: Privates Software-Projekt

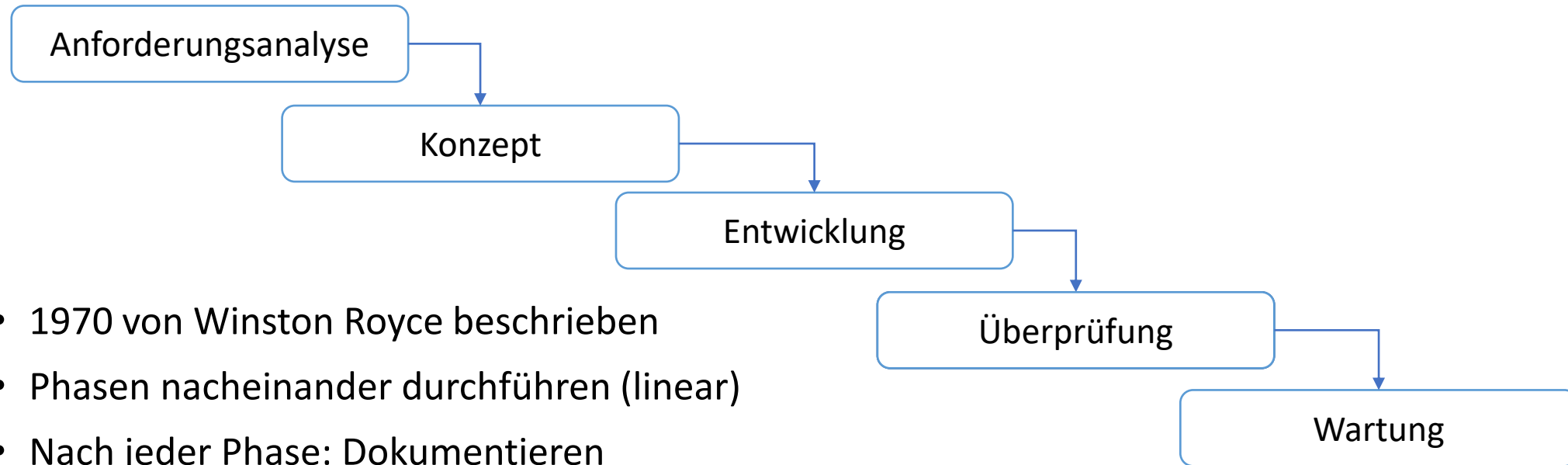


Vorteile: Sehr effizient

Nachteile: Skaliert nicht gut, Kein Konzept, evtl. wenig Überprüfung

5. Software-Entwicklungsprozesse

Vorgehensmodelle: Wasserfallmodell



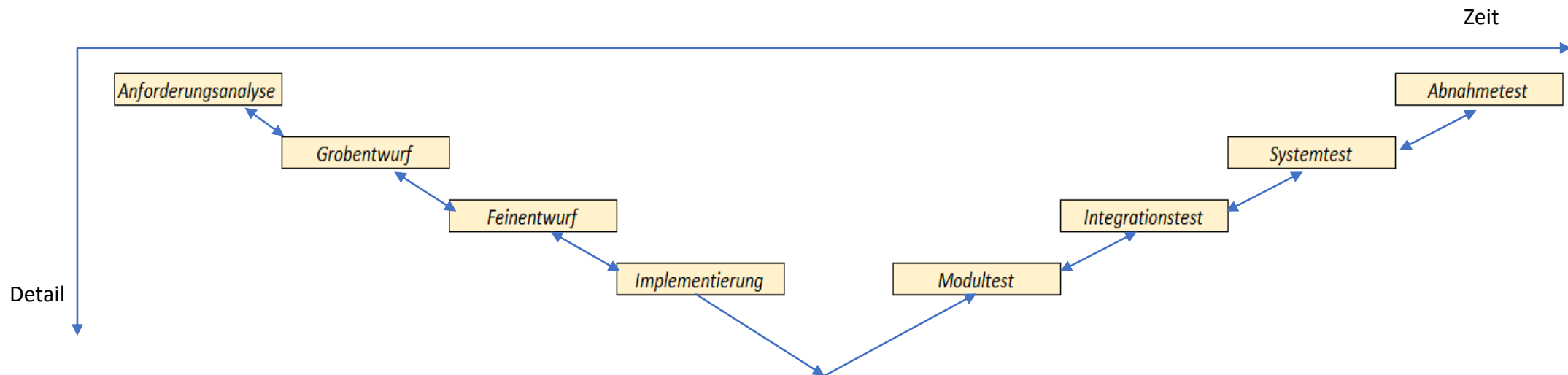
- 1970 von Winston Royce beschrieben
- Phasen nacheinander durchführen (linear)
- Nach jeder Phase: Dokumentieren

Vorteile: Klare Abgrenzung, Einfache Planung und Kontrolle

Nachteile: Klare Abgrenzung praktisch unmöglich; Mangelnde Flexibilität; Rückschritte sind unvermeidlich

5. Software-Entwicklungsprozesse

Vorgehensmodelle: V-Modell



- Aufbauend auf Wasserfallmodell: Phasen nacheinander durchführen
- Dokumentation sehr wichtig
- „Rückkehr“ zu vorherigen Phasen möglich
- Zu jeder Phase (linke Seite) zugehörige Testphase (rechte Seite)
- Vorteile: Verschiedene Abstufungen des Konzepts; Fokus auf Tests
- Nachteile: Fragmentierte Struktur; Veränderungen sind machbar aber schwierig (je detaillierter)

5. Software-Entwicklungsprozesse

Übung zu Vorgehensmodellen

1. Trefft euch mit euren Gruppenmitgliedern in den Nebenräumen (Entsprechend der Gruppennummer, z.B. Gruppe 3 -> Nebenraum 3)
2. Einigt euch auf ein (fiktives) Projekt:
 - Muss kein Software-Projekt sein!
 - Ideen:
 - Etwas, was ein Gruppenmitglied vor Kurzem durchgeführt hat
 - Entwicklung eines neuen Features für eine Webapp
 - ...
3. Wählt eines der beiden Modelle (Wasserfall / V) und bearbeitet folgende Aufgaben:
 - a) Warum habt ihr euch für das Modell entschieden?
 - b) Wie lässt sich das Projekt mit dem gewählten Modell erfolgreich umsetzen?
 - c) Was sind mögliche Fallstricke?
 - d) BONUS: Was müsste sich am Modell ändern, damit euer Projekt noch erfolgreicher wird?