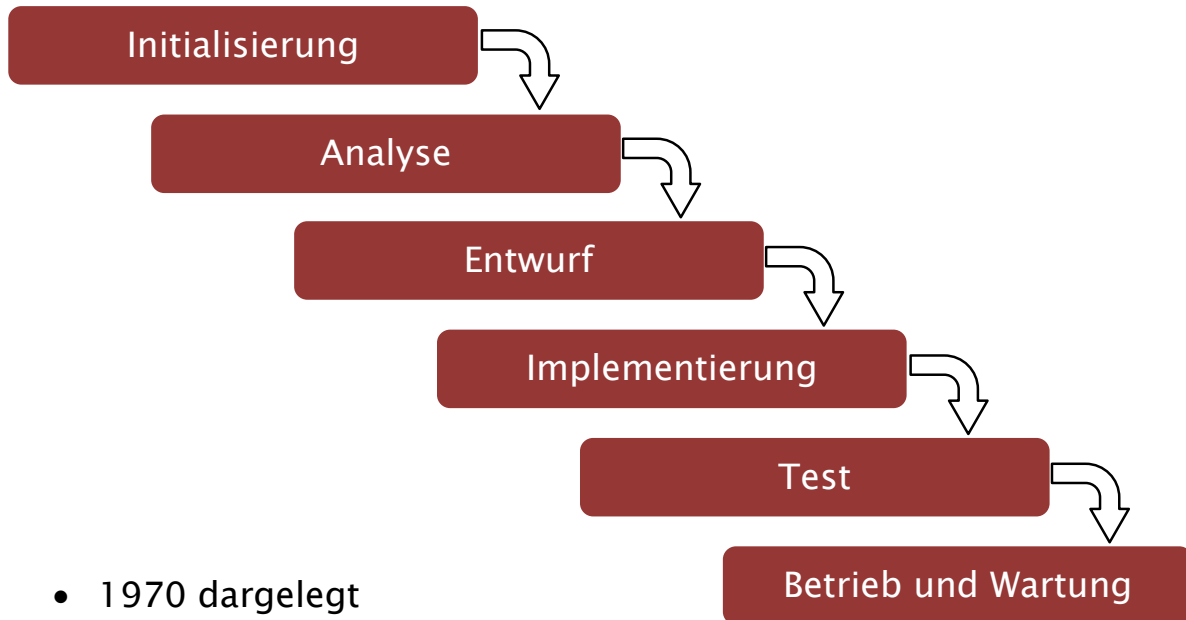


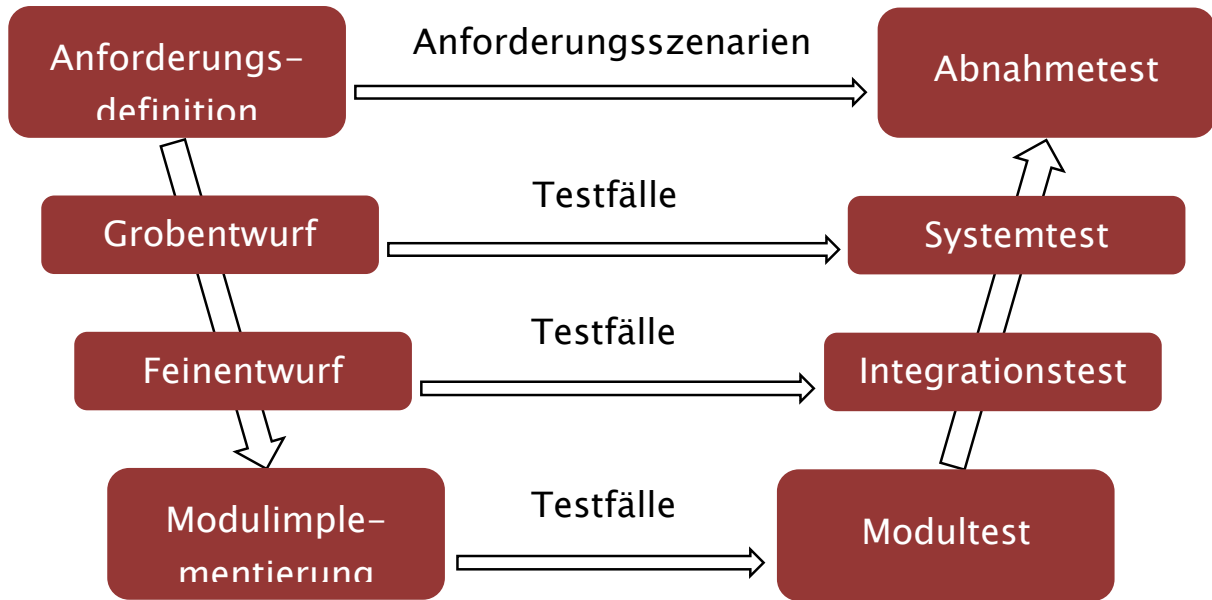
Andere Vorgehensmodelle

Wasserfallmodell



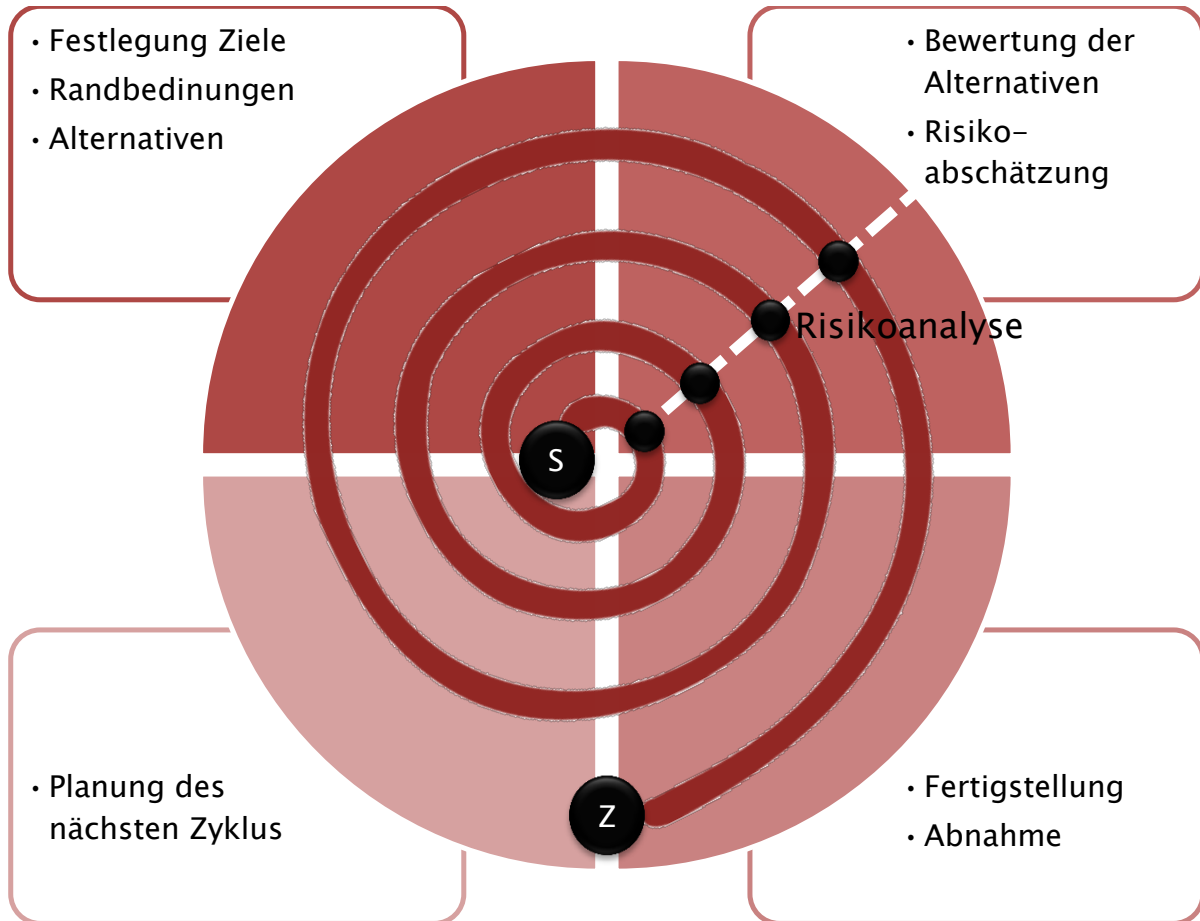
- 1970 dargelegt
- Phase muss erst vollständig abgeschlossen sein, bevor nächste Phase beginnt (**PROBLEM:** Testphase)
- Keine Rücksprungmöglichkeit in klassischer Variante
- Sehr einfaches Modell, behandelt Projektmanagement und -infrastruktur nicht
- Sagt nichts aus über Rollen und Methoden

V-Modell



- Aus den 1980ern
- Tests überprüfen einzelne Phasen, werden parallel entwickelt
- Eher aufwändig

Spiralmodell



- Durchläuft vom Start bis zum Ziel immer wieder gleichartige Zyklen
- Augenmerk wird auf die Ausgestaltung und Bewertung von Alternativen gelegt
- Risikobetrachtung ist Stärke des Modells
- Im ersten Durchgang wird Konzeptionsdokument abgenommen, später die getesteten Softwarekomponenten



Maturareise nach Barcelona. Es soll Freizeitprogramm organisiert werden:

1. Ermitteln Sie vier Risiken: Was kann alles schiefgehen
2. Bewertung jedes Risikos: Einschätzung mit Begründung
3. Vermeidungsstrategien

Extreme Programming

- Aus den 1990ern
- Kann flexibel auf Änderungen im Projekt reagieren
- Bei vielen Änderungen wäre Dokumentation nicht aktuell, deshalb Dokumentation nur im Quellcode
- Kundenbeteiligung
- Viel Kommunikation im IT-Team und zw. IT-Team und Kunde notwendig
- Wegen Refactoring wird Code so einfach wie möglich gehalten
- Gemeinsamer Codebesitz. Mein Code von jedem änderbar. Code muss so geschrieben werden, damit er verstanden wird.
Einhalten von Programmierstandards
- Test als fundamentaler Bestandteil. Automatisierte Tests
- Keine Spezialisierung, jeder Entwickler macht alles
- Pair-Programming
- Kurze Release-Intervalle
- Wenige Rollen: Kunde, Entwickler, Tester, (Projektmanagement ?)

Kritik

- XP nur für kleine Projekte geeignet
- Ablehnung von Dokumentation führt zu Qualitätsverlust
- Pair-Programming ist auf die Dauer viel zu aufwändig