



Softwareentwicklung Agil vs Non Agil

Gruppe Napoleon:

Svetlana Marchenko, Charlotte Geiger, Achim Schliebener

Gliederung

A person is captured mid-air, jumping over a horizontal wooden beam into a pool of water. The scene is set at sunset or sunrise, with a warm, golden glow. In the background, a group of people stands on a wooden platform, watching the jump. The water in the foreground is calm, reflecting the person and the beam. The overall atmosphere is one of achievement and celebration.

- Einführung
- Geschichte der Softwareentwicklung
- Non-Agile
- Agile
- Gegenüberstellung
- Fazit

Geschichte

- Konzept der „Datenverarbeitung“ schon sehr alt z.B Abacus
- Jedoch immer Hardware gebunden
- Software benötigt als Grundkonzept „Mehrzweck Prozessor“
- Turing Maschine (1935) + Speicher

Geschichte

- Ca 1950 Assembler als Programmiersprache
- Vorher „Programmierung“ von Computer durch Neuverkabelung
- Apollo Mission in BASIC geschrieben
- Im Rahmen der Apollo Mission Begriff „Software Engineering“ geprägt

Begriff Software Engineering

- „Software Krise“ 1965-1985
- Probleme in der Softwareentwicklung
- Systeme wurde immer komplexer
- Ziele konnten nicht eingehalten werden
- Budget, Zeit, Qualität

Durchbruch

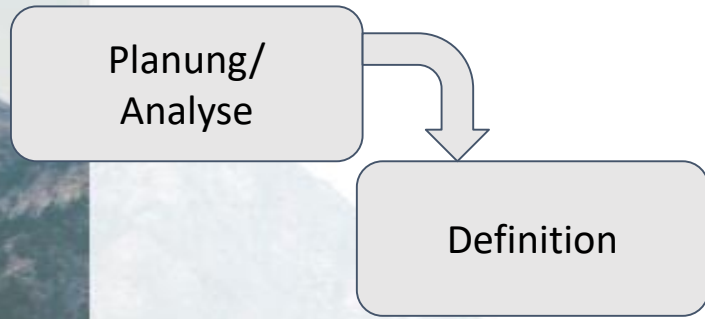
- SE ist der systematische Ansatz bei der Entwicklung
- Theorie Methoden und Werkzeuge werden angewendet um Software zu entwickeln die
- Den Spezifikationen entspricht
- Kosten effizient ist
- Qualität aufweist

Wasserfallmodell

Planung/
Analyse

Resultiert in dem Lastenheft.
Beinhaltet alle Anforderungen des Auftraggebers.
Bsp. Welches Betriebssystem? Welche Hardware? Welche Funktionen?

Wasserfallmodell

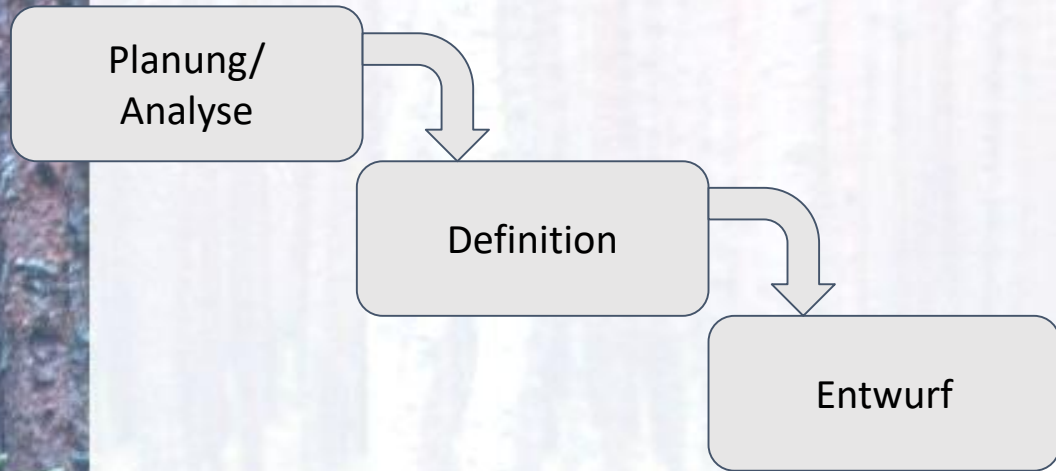


Resultiert in dem Pflichtenheft.

Beinhaltet wie der Auftragnehmer die Anforderungen lösen möchte.

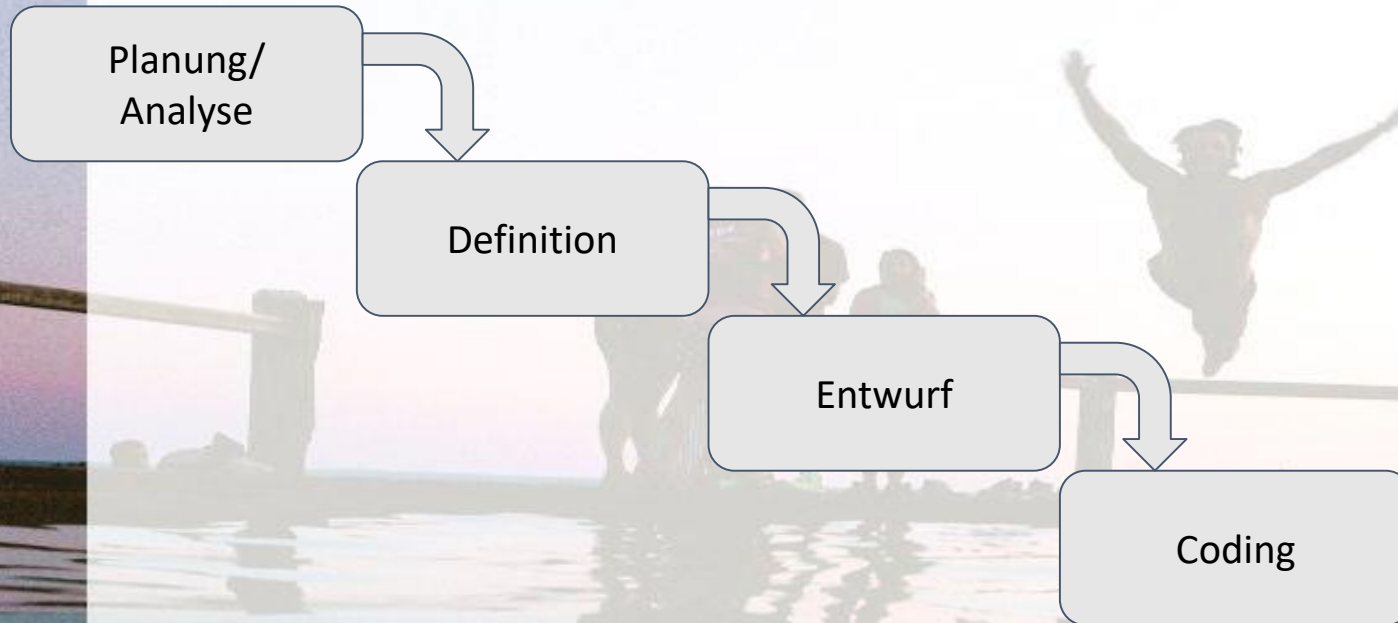
Bsp Zielbestimmungen, Berechtigung, Einsatz, Umgebung

Wasserfallmodell



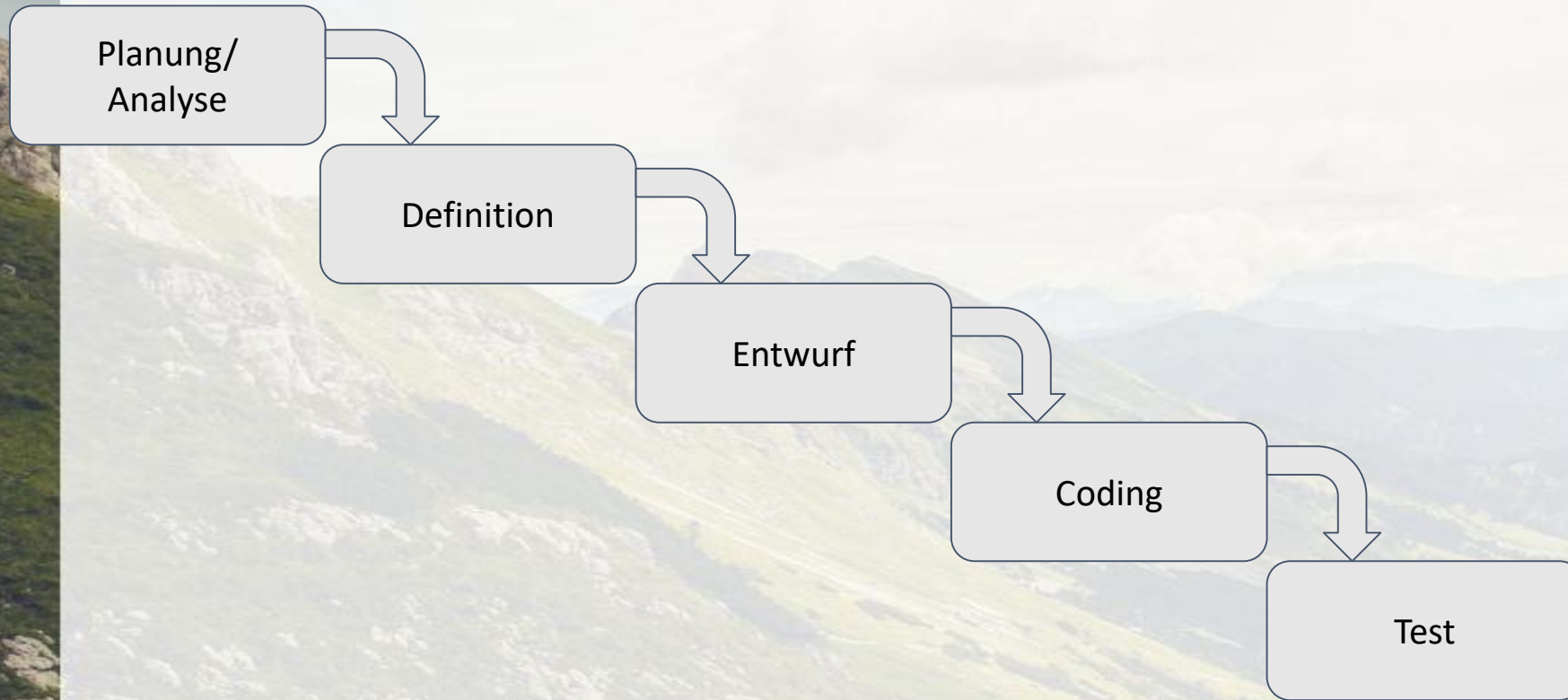
Erstellen von Klassendiagramm

Wasserfallmodell



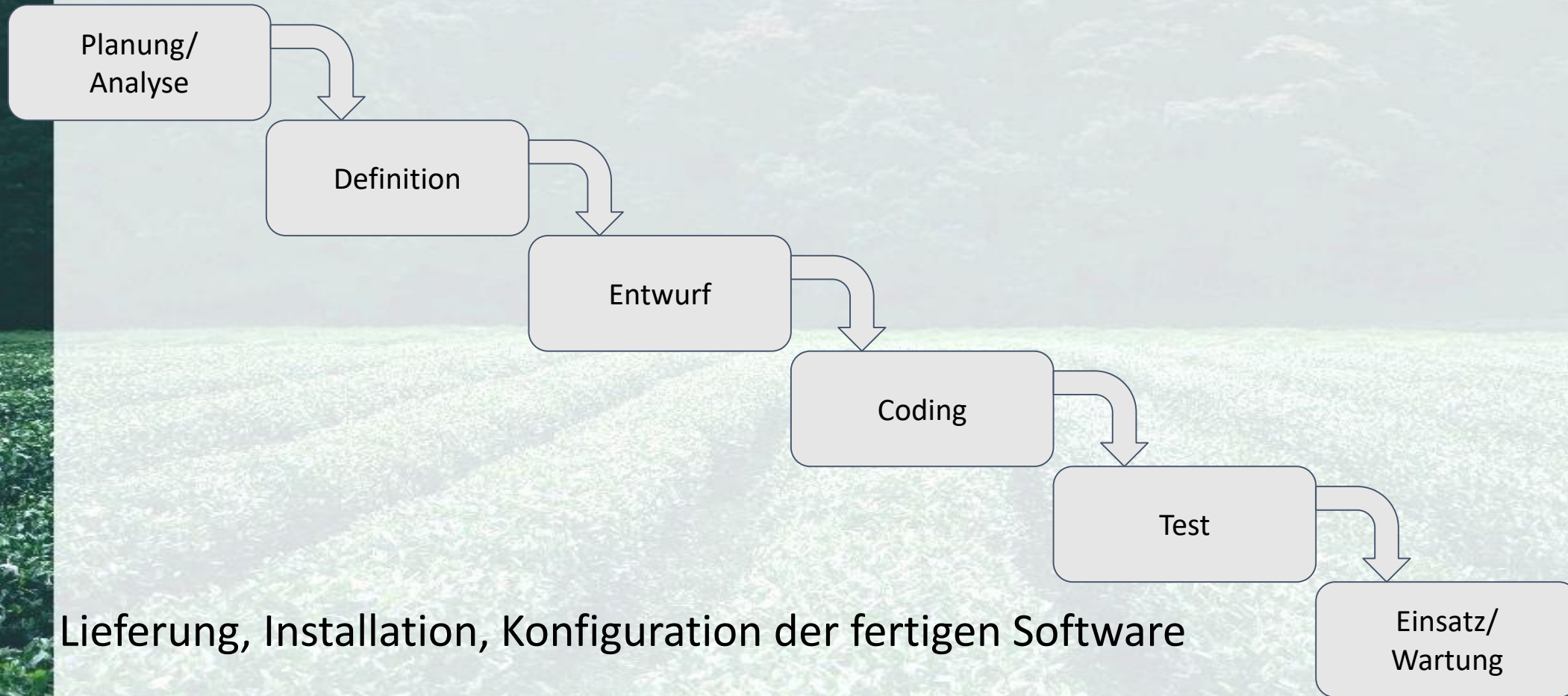
Implementierung der Entwürfe

Wasserfallmodell



Testen/Debuggen des Codes und Dokumentation der Resultate

Wasserfallmodell



Lieferung, Installation, Konfiguration der fertigen Software

Einsatz/
Wartung

Wasserfall

- Aktivitäten sind in der vorgegebenen Reihenfolge und in der vollen Breite vollständig durchzuführen.
- Am Ende jeder Aktivität steht ein fertiggestelltes Dokument, d. h. das Wasserfallmodell ist ein „dokumentgetriebenes“ Modell.
- Der Entwicklungsablauf ist sequenziell; d. h. jede Aktivität muss beendet sein, bevor die nächste anfängt.
- Es orientiert sich am sogenannten Top-down-Verfahren.
- Es ist einfach und verständlich.
- Eine Benutzerbeteiligung ist in der Anfangsphase vorgesehen, anschließend erfolgen der Entwurf und die Implementierung ohne Beteiligung des Benutzers bzw. Auftraggebers. Weitere Änderungen stellen danach Neuaufträge dar.

V-Modell

- Phasen ähnlich dem Wasserfall
- Zusätzlich werden noch Phasen für Qualitätssicherung integriert
- Wie beim Wasserfall muss jede Phase abgeschlossen sein

Zeit

Detailierung

Systemanforderungs-
analyse

System-Architektur

System-Entwurf

Software-Architektur

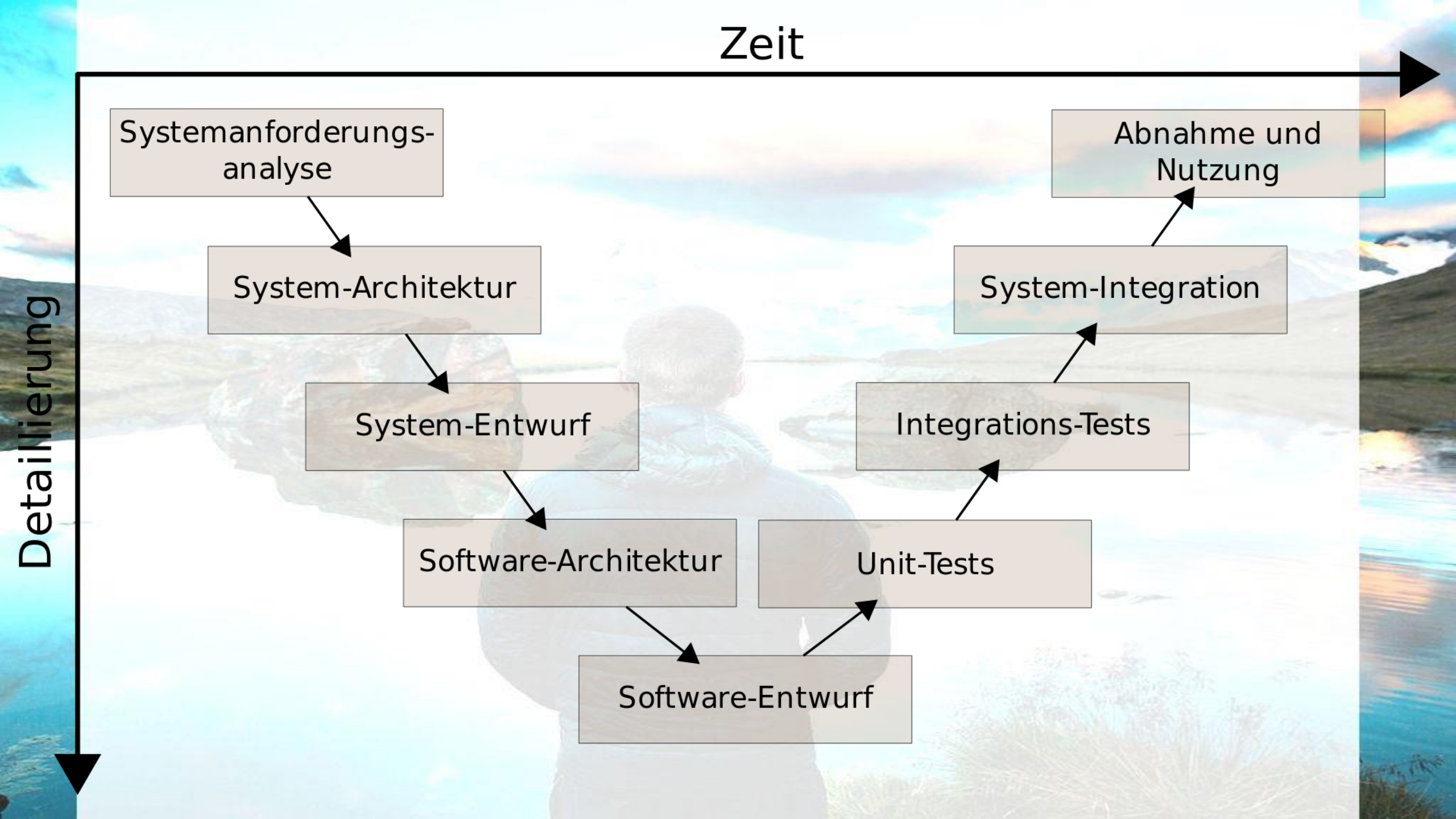
Software-Entwurf

Unit-Tests

Integrations-Tests

System-Integration

Abnahme und
Nutzung



Spiral Modell

- Festlegung von Zielen, Identifikation von Alternativen und Beschreibung von Rahmenbedingungen
- Evaluierung der Alternativen und das Erkennen, Abschätzen und Reduzieren von Risiken, z. B. durch Analysen, Simulationen oder Prototyping
- Realisierung und Überprüfung des Zwischenprodukts
- Planung des nächsten Zyklus der Projektfortsetzung.



Manifesto for Agile Software Development

Individuals and interactions over processes and
tools

Working software over comprehensive
documentation

Customer collaboration over contract
negotiation

Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.



4 Werte
12 Prinzipien

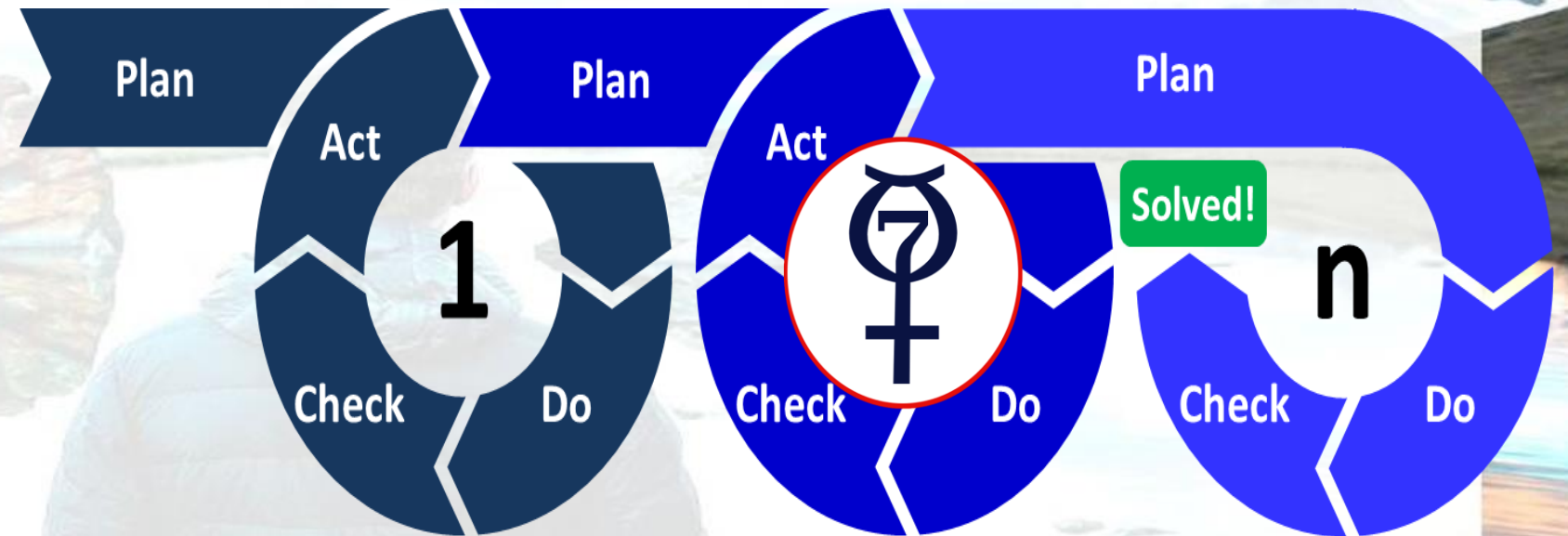


Woher kommt
“Agile
Thinking”?



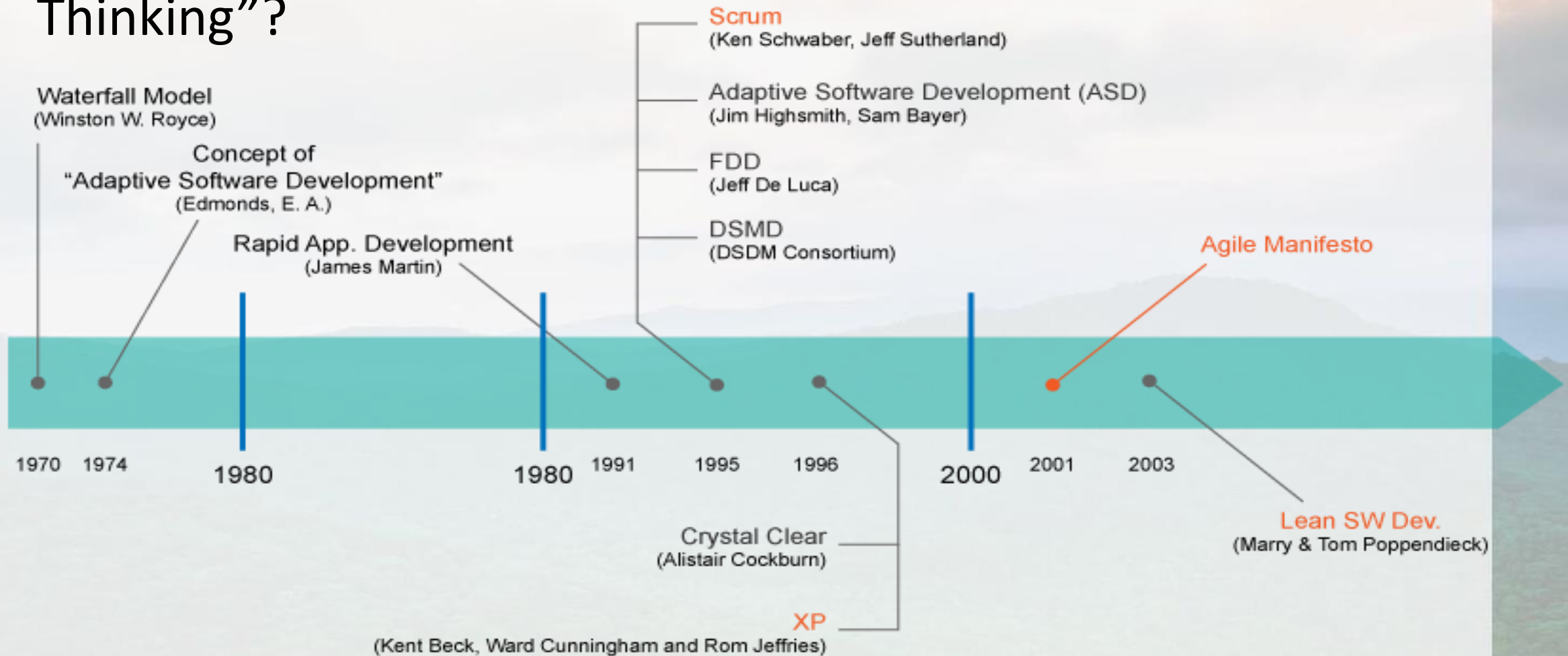
1900

IID IN SOFTWARE

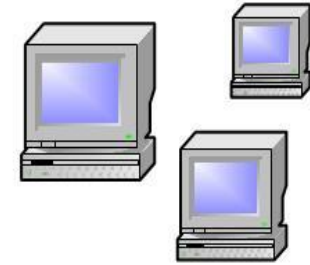
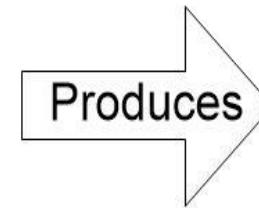
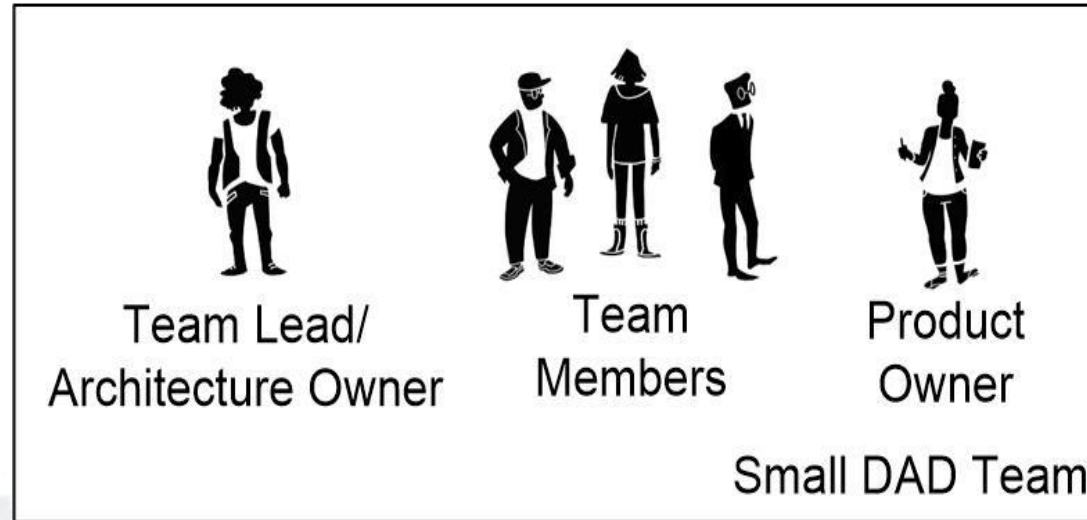


Woher kommt “Agile Thinking”?

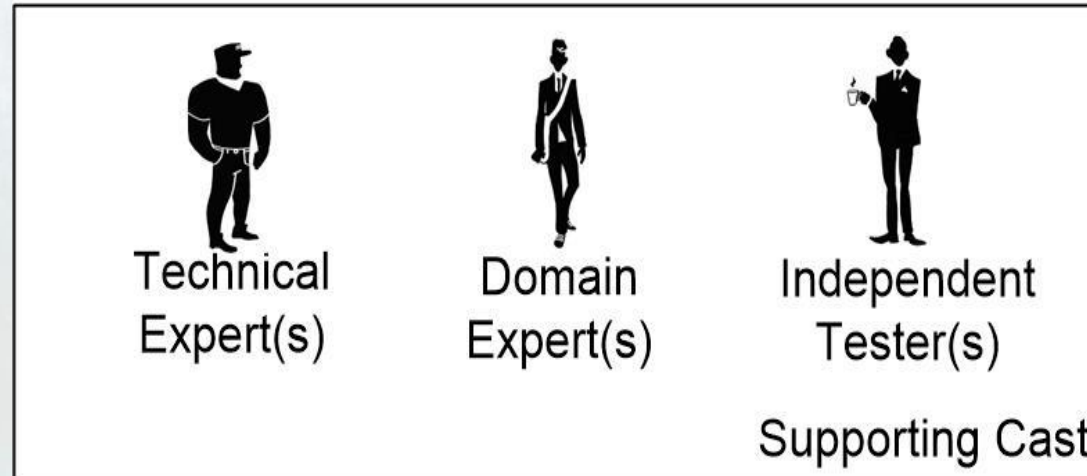
History of Agile



Agile Team



Consumable
Solution



50% der Teammitglieder sind mehr vom Teamerfolg motiviert als von den persönlichen Zielen des Unternehmens (27%) oder des Einzelnen (23%). (Quelle: Atlassian)

Traditional Teams



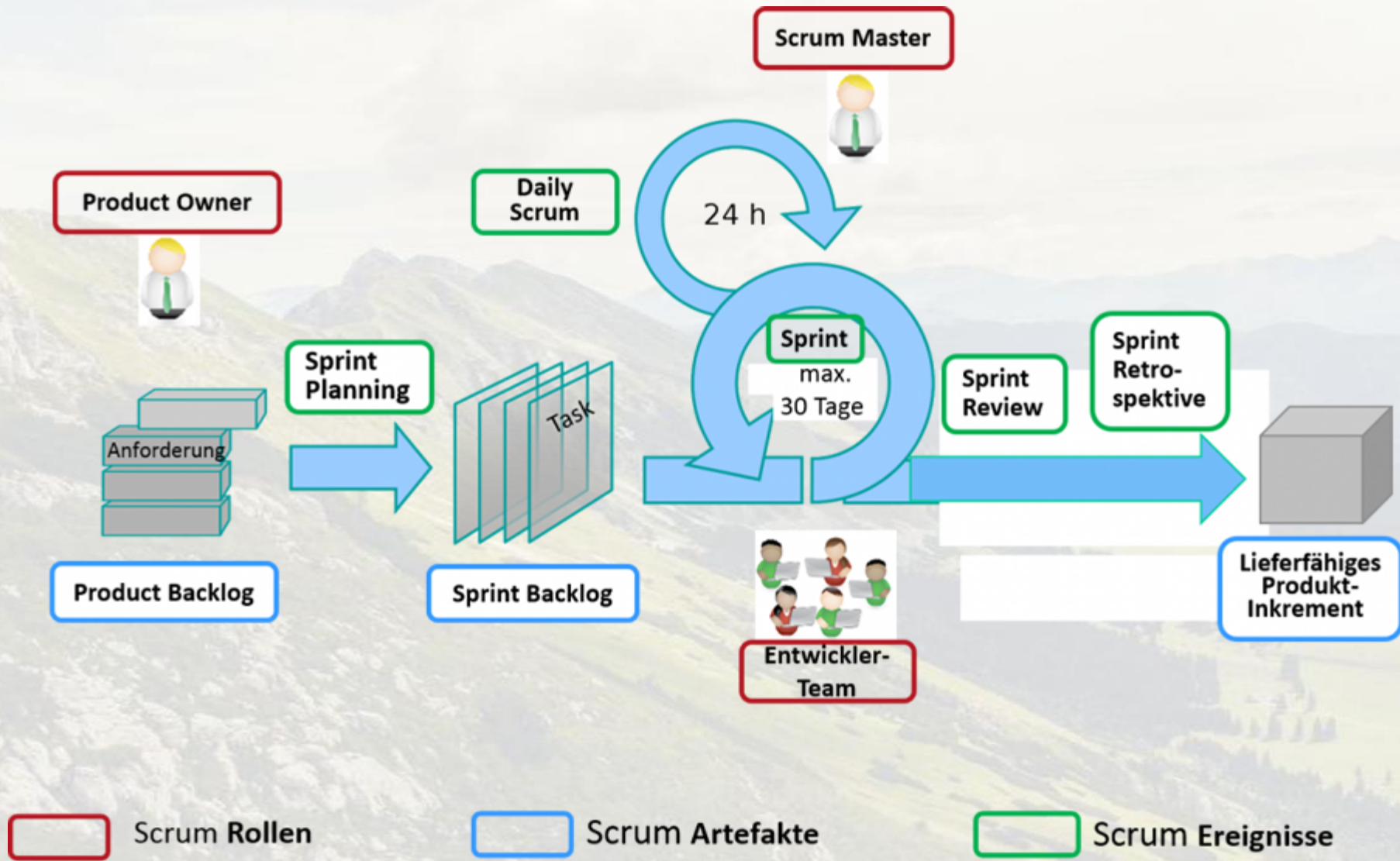
Agile Teams



Scrum

A person is captured in mid-air, jumping over a wooden fence. The scene is set outdoors during sunset or sunrise, with a warm, golden glow. A group of people is standing behind the fence, watching the jumper. The jumper's arms are outstretched in a celebratory gesture. The background shows a line of trees and a clear sky with soft light. The overall mood is one of accomplishment and teamwork.

- Product Owner
- Entwickler
- Scrum Master
- Stakeholder

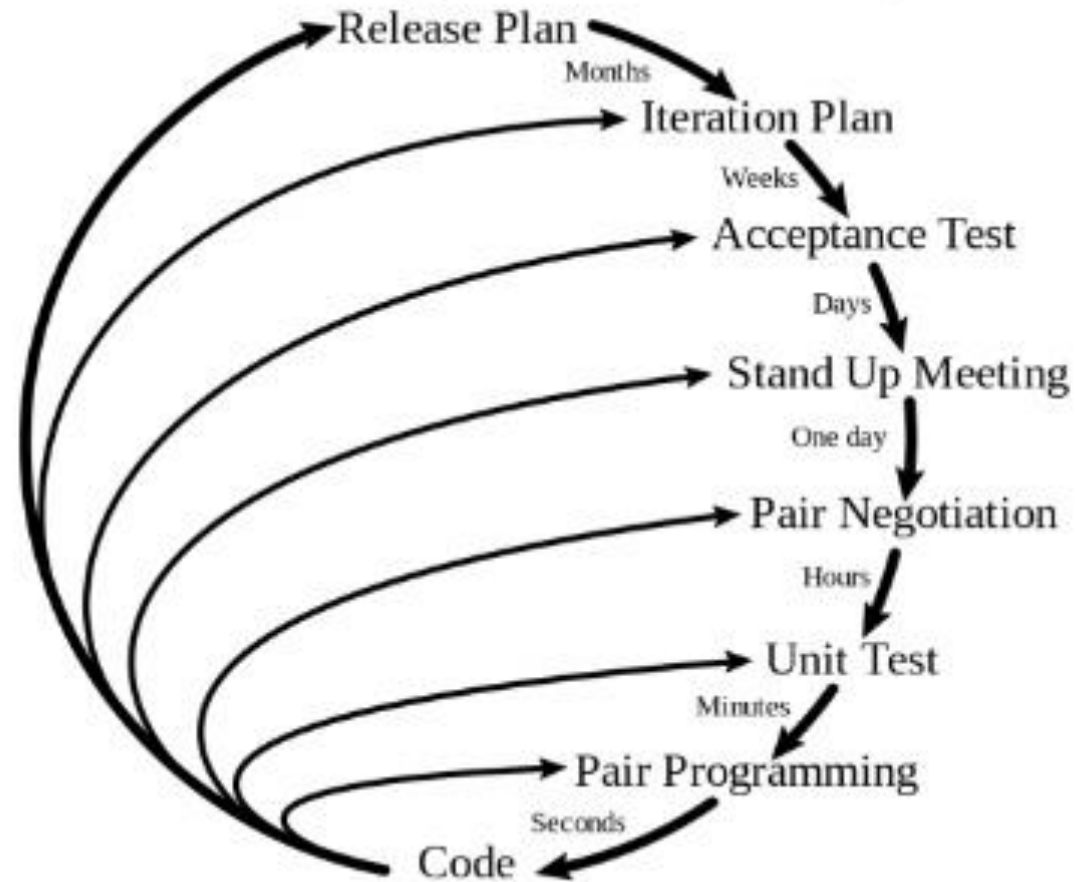


XP

- Kunde kennt zu Beginn der Entwicklung nicht alle Anforderungen
- Programmiererteam hat nicht alle Informationen zu Projektbeginn
- kann daher keine genaue Aufwandsschätzung abgeben
- Prioritäten können sich im Lauf des Projekts ändern
- Für die zu entwickelnden Funktionalitäten werden jeweils die Schritte Risikoanalyse, Nutzenanalyse, die Bereitstellung einer ersten ausführbaren Version (Prototyping) und ein Akzeptanztest durchgeführt.

Extreme Programming (XP)

Planning/Feedback Loops



FDD

- Prozess #1: Entwickle ein Gesamtmodell (Rollen: alle Projektbeteiligte)
- Prozess #2: Erstelle eine Feature-Liste (Rollen: in der Regel nur die Chefprogrammierer)
- Prozess #3: Plane je Feature (Rollen: Projektleiter, Entwicklungsleiter, Chefprogrammierer)
- Prozess #4: Entwurf je Feature (Rollen: Chefprogrammierer, Entwickler)
- Prozess #5: Konstruiere je Feature (Rollen: Entwickler)



**Develop An
Overall Model**



**Build A
Features List**



**Plan By
Feature**



**Design By
Feature**



**Build By
Feature**

Gegenüberstellung

- Non-Agile
 - simpel zu verstehen/umzusetzen
 - Gut zu dokumentieren
 - Änderungen schwer
 - Anforderungen müssen deutlich formuliert sein
- Agile
 - Sehr flexibel
 - Gute Visualisierung der Vorgänge im Projekt
 - Kommunikation +-
 - Einfach zu verstehen schwer umzusetzen

Fazit



Quellen

<https://bildungsbibel.de/wp-content/uploads/2016/10/wasserfallmodell-softwareentwicklung-projektmanagement-beispiel.jpg>

<https://de.wikipedia.org/wiki/V-Modell>

<https://de.wikipedia.org/wiki/Wasserfallmodell>

<https://de.wikipedia.org/wiki/Vorgehensmodell>

<https://de.wikipedia.org/wiki/Spiralmodell>

