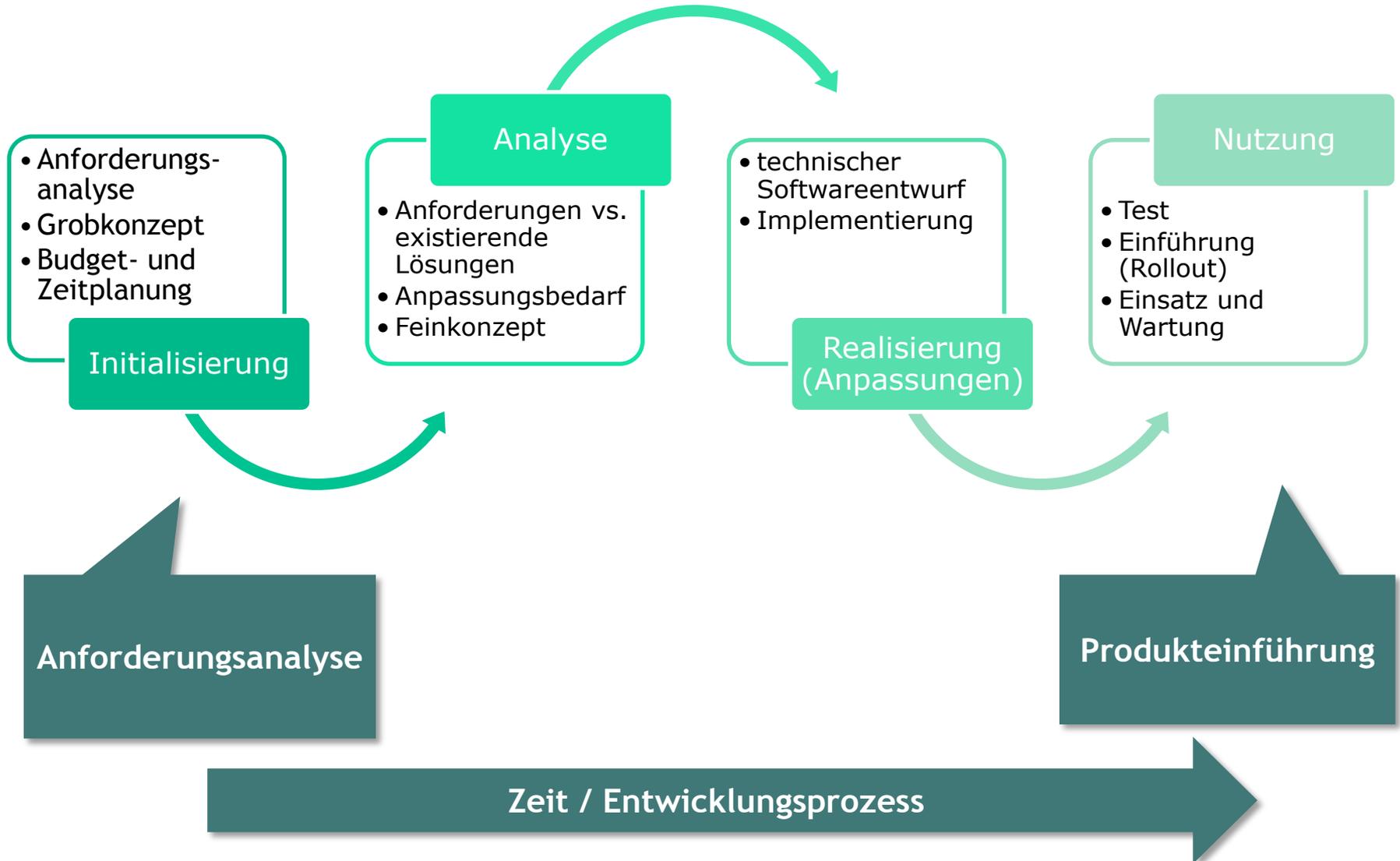


# Aspekte zur Vorgehensweise bei Planung & Einführung eines SDM-Systems

Martin Liebscher  
DYNAmore GmbH

Stuttgart, 2. Juli 2013

# Überblick Phasen der Softwareentwicklung / -beschaffung



# Anforderungsanalyse

---

## ■ Teilnehmer

- Nutzer
- IT-Abteilung
- (Dienstleister)

## ■ Was wird gemacht...

- Kurzdokumentation des IST-Zustandes
  - bestehende Prozesse
  - vorhanden Softwaresysteme
- Sammeln der Nutzeranforderungen (SOLL-Zustand) z.B. durch Interview
  - Dokumentieren von **UserStories**
  - ableiten von **UseCases**
- **nicht fachliche Anforderungen, Systemanforderungen**

## ■ Ergebnis

- Grobkonzept (strukturierte Zusammenstellung) & Checkliste für Anfragen

# Anforderungsanalyse - UserStories

---

## ■ UserStory

- kurze, einfache Beschreibung eines Nutzungsszenarios
- Ausgangssituation, Normaler Vorgang, Probleme, Anmerkungen, Ergebnis
- Quelle (Nutzer) muss mit dokumentieren werden

## ■ Beispiel US1, Person X:

„Zu Versuchen/Simulationen liegen Dokumentationen in Papierform vor. Diese sollen als Scan [PDF] den entsprechenden Versuchen zugeordnet werden können“

**Normal:** Nutzer sucht passenden Versuch, wählt mehrere Dokumente aus einem Verzeichnis aus und ordnet die zuvor ausgewählten Dokumente zu.

**Mögliches Problem:** Die Suche liefert keinen Treffer. Der Versuch/Simulation wurde im System noch nicht angelegt. Das Anlegen eines (Leer-)Versuches soll in diesem Schritt ermöglicht werden.

**Anmerkung:** Die Dokumente können sehr umfangreich sein. Der Importvorgang soll das Arbeiten nicht weiter behindern (ggf. im Hintergrund stattfinden).

**Ergebnis:** Dokumente sind dem Versuch zugeordnet

# Anforderungsanalyse - UseCases

---

## ■ UseCases / Workflows ableiten

- UserStories priorisieren
- Kernfunktionalität herausarbeiten
- nachrangige Funktionalitäten bestimmen und Einordnen
- Grundsätzlich: Die in den UserStories enthaltene Funktionalität detaillieren. Einzelne Arbeitsschritte und eingesetzte Daten genauer beschreiben.

## ■ Charakteristik

- oft kurze formalisierte Darstellung
- einzelne UseCases können Teilschritt eines Szenarios sein.
- ggf. Ablaufdiagramme und Modellierung mittels UML

## ■ Beispiel

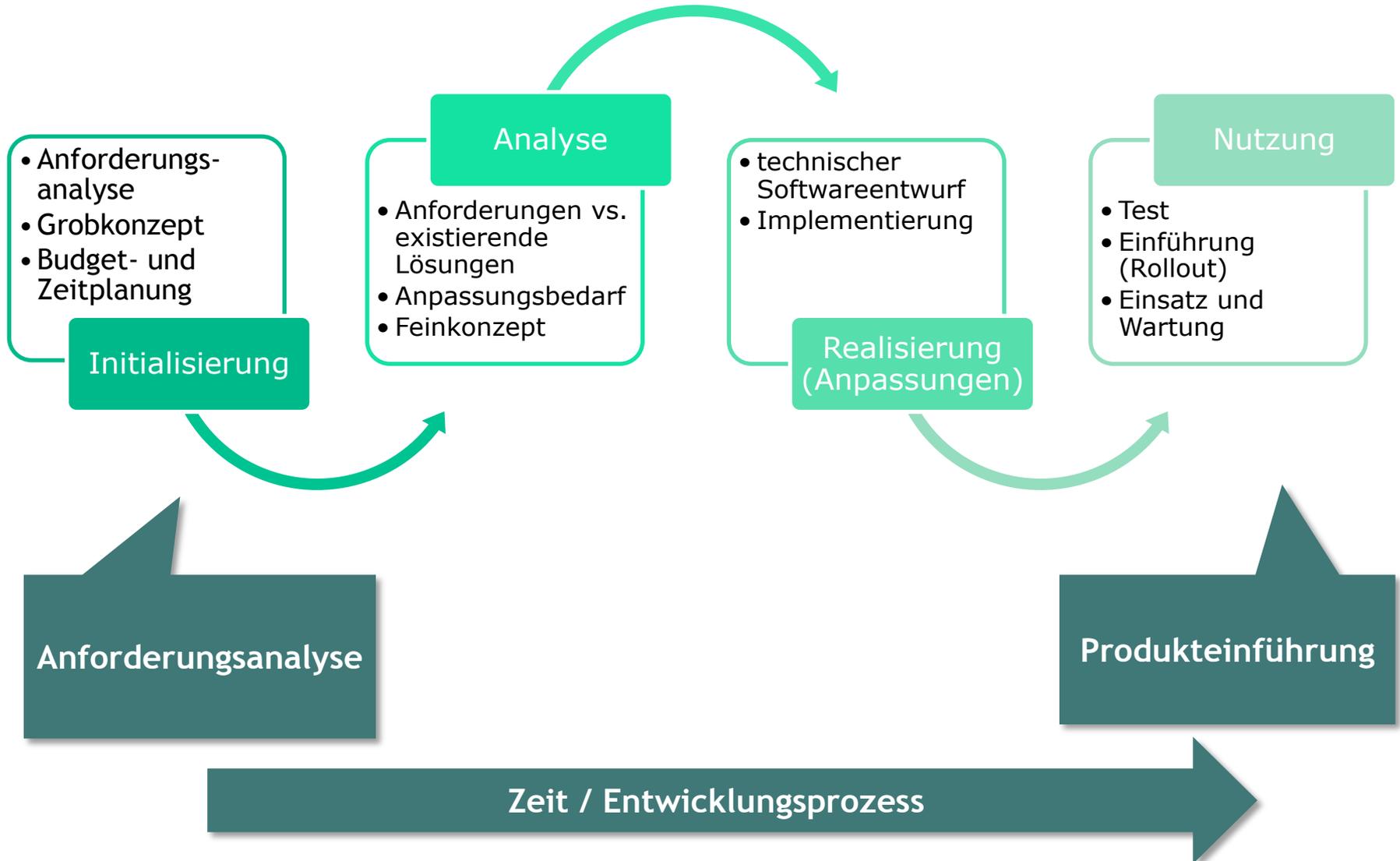
- UC1: Suche einer Simulation / eines Versuchs
  - Nutzer kann in Ansicht xy nach Versuchen mittels Wildcards (?\*) suchen.
  - Auslösen der Aktion über Button, Tastenkombination
  - Eingabe erfolgt in einem separaten Dialog
  - ermittelte Ergebnisse werden in Ansicht xy dargestellt

# Anforderungsanalyse - Nicht-funktionale Anforderungen

---

- Nutzer/Nutzung
  - Wie viele Nutzer sind zu erwarten?
  - Welche Datenmengen und Datentypen sind zu erwarten [kurz und langfristig]
  - Standortübergreifende Nutzung / externer Zugriff?
- Soll die Applikation lokal auf dem Arbeitsplatzrechner laufen oder ist eine reine Serveranwendung mit Webinterface bevorzugt?
- Welche Betriebssysteme/Umgebungen sind zu unterstützen?
- Welche vorhandenen IT-Systeme sollen/müssen (wie) mit eingebunden werden? Schnittstellen?
- Gibt es zur Integration in die IT-Landschaft Vorgaben / Randbedingungen?
  - vorhandene Datenbanksysteme, die genutzt werden sollen (Gründe: Lizenzkosten, Konsistenz, ...)
  - Vorfestlegungen auf Webservice-Umgebungen / Technologien?
- Welche Sicherheitsaspekte (Verschlüsselung, Rechte-Management) sind zu berücksichtigen?
- ...

# Überblick Phasen der Softwareentwicklung / -beschaffung

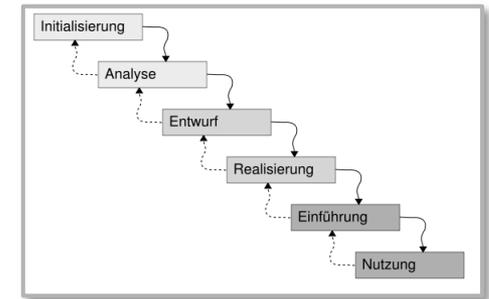


- **Instrument zur Steuerung einer Softwareentwicklung / -beschaffung, von der Konzeption bis zum Produktiveinsatz (zeitlicher Ablauf einzelner Phasen)**
  
- **kein Konsens bzgl. eines optimalen Entwicklungsprozesses**
  - **etabliert: Wasserfallmodell [Royce]**
    - eines der ältesten Modelle (1970)
    - eine starre Abfolge der einzelnen Phasen
  - **aktuell Vielzahl an Weiterentwicklungen**
  - **insbesondere agile Methoden, mit iterativen Vorgehensweisen werden erfolgreich eingesetzt**
  
- **Gegenüberstellung Wasserfallmodell <> agile Vorgehen**

# Wasserfallmodell - Prozessüberblick

## ■ verbreitetes, nicht-iteratives Modell [Royce]

- lineares Vorgehen
- Definition einer Phasenabfolge

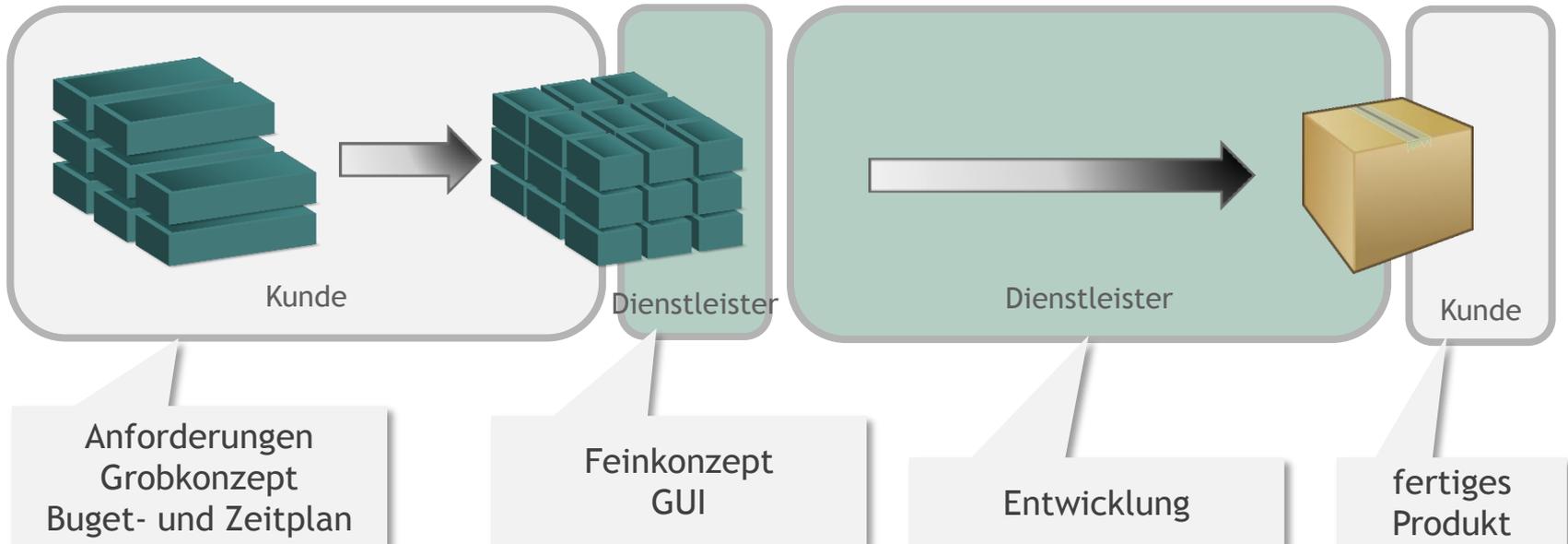


Initialisierung

Analyse

Realisierung

Test, Einführung  
Nutzung



# Entwicklungsstrategien - Wasserfallmodell

---

## ■ Probleme des Wasserfallmodells

- Detaillierte Festlegung zu Projektbeginn schwierig. Anforderungen ergeben sich oft erst bei Nutzung.  
Ergebnis entspricht nicht tatsächlichen Anforderungen.
- Oftmals unverhältnismäßig hoher Aufwand in der Analyse- und Konzeptionsphase
- Fehler werden unter Umständen spät erkannt (Big Bang) und müssen mit erheblichem Aufwand entfernt werden.
- Langer Laufzeit von Softwareprojekten. Software ist bereits zum Zeitpunkt ihrer Einführung inhaltlich veraltet.
- ...

## ■ Warum ist es trotzdem populär?

- zumindest theoretisch
  - definiertes Ergebnis
  - definierter Nutzungsbeginn
  - definierter Festpreis
- Ziele können tatsächlich oft nicht vollständig realisiert werden.

## ■ Wann ist ein Einsatz sinnvoll?

- klar und detailliert definierbare / definierte Anforderungen
- geringe Prozesskomplexität
- genaue Kenntnis der abzubildenden Unternehmensprozesse
- überschaubarer Projektumfang und -laufzeit

## ■ Entwicklung bzw. Integration von SDM-Systemen ist typischer Weise nicht für das Wasserfallmodell geeignet

- SDM-Systeme müssen i.d.R. alle Prozesse vollständig abbilden können
- Nutzung betrifft potentiell „alle“ Entwicklungsschritte (Übergabe CAD, Handling FE-Modelle, Simulation, PostProcessing, Auswertung, Report)
- Neustrukturierung Datenhaltung (-archivierung) und Nutzerinterface muss umgesetzt bzw. gestaltet werden

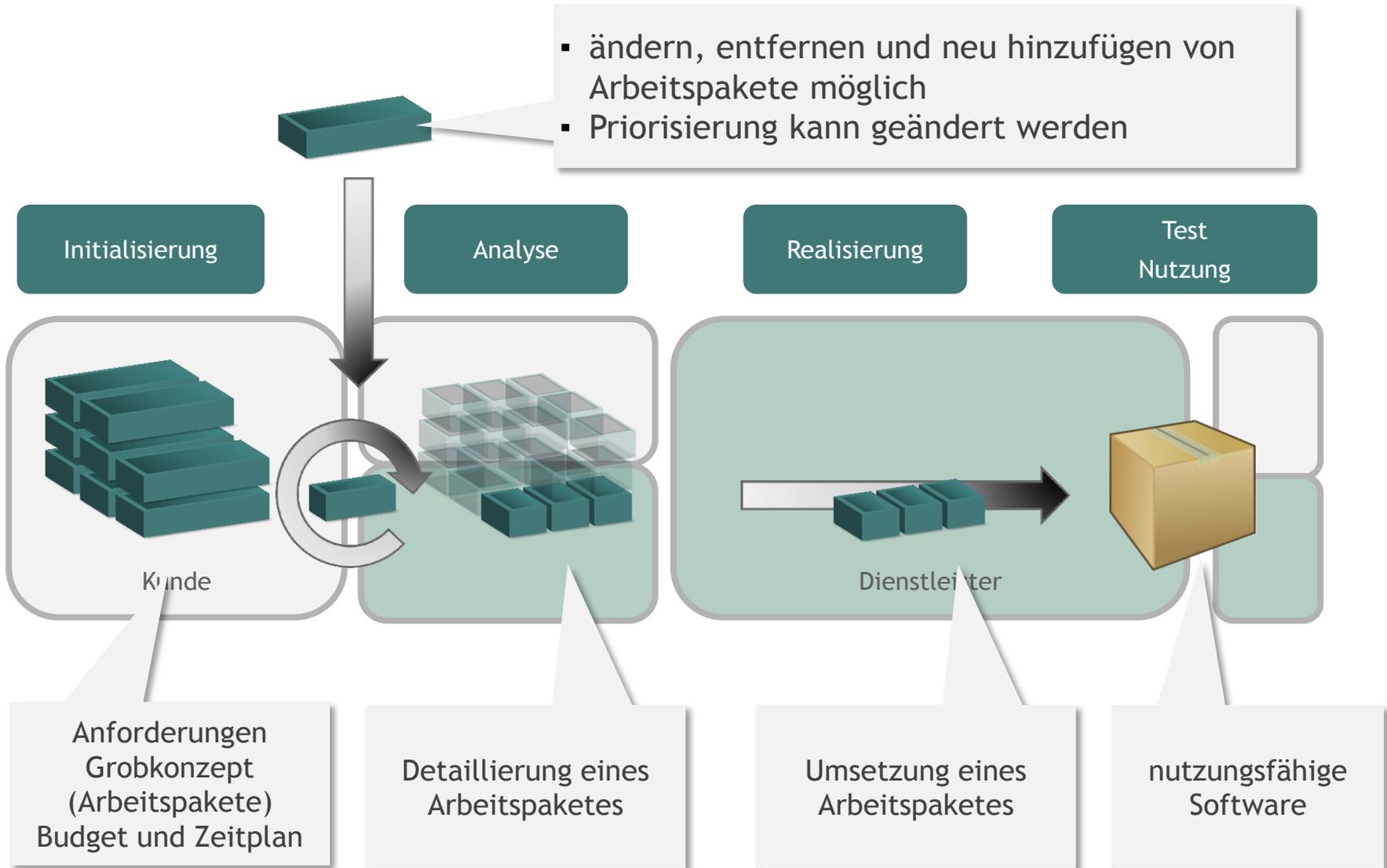
## ■ Grundprinzipien

- Auflösung des starren Entwicklungskonzeptes (vgl. Wasserfall-Modell)
- Abbau der Bürokratisierung, iteratives Vorgehen, mehr Kommunikation
- Reduktion von Konzeptions- und Entwurfsphasen auf ein Minimum
- so früh wie möglich ausführbare Software, in regelmäßigen kurzen Abständen gemeinsame Abstimmung mit dem Kunden bzw. Feedback
- die Adaption von Projektzielen und Produkt-Features ist ausdrücklich Teil des Prozesses und kein Fehler

## ■ Beispiele (eine Vielzahl an Methoden, teilweise in der Definition überschneidend/ähnlich)

- Adaptive Software Development
- Extreme Programming (XP)
- Feature Driven Development
- SCRUM
- ...

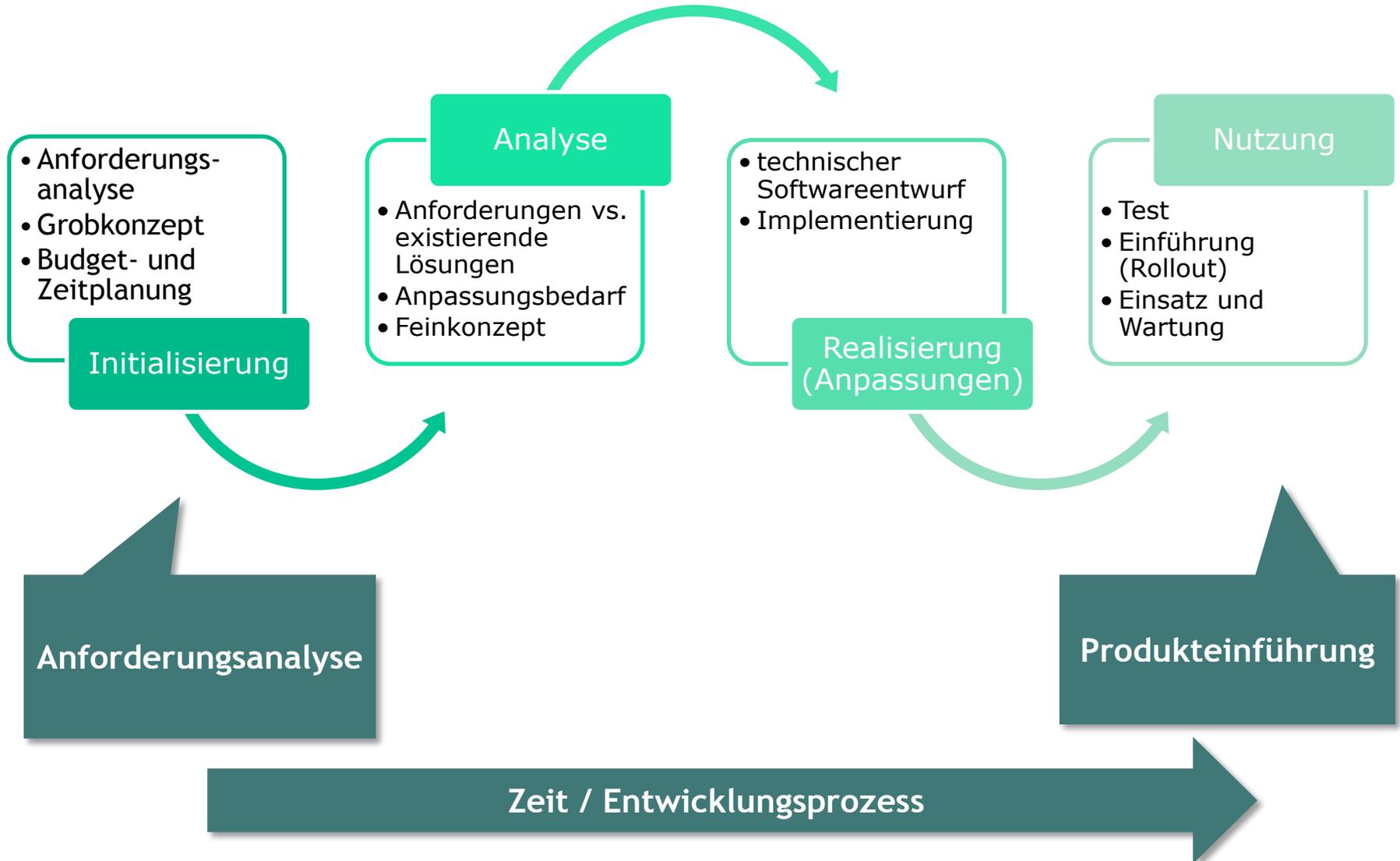
# Agiles Vorgehen - Prozessüberblick [SCRUM]



## ■ Vorteile / Unterschiede / Probleme

- mehrere, kurze Entwicklungsphasen
- Produkt stetig evaluiert werden, bessere Steuerungsmöglichkeiten für den Kunden
- regelmäßig müssen (Richtungs-)Entscheidungen getroffen werden; hoher Einsatz auf Seite des Kunden notwendig; teilweise „Entscheidungszwang“
- Entwurfs- und Konzeptionsaufwand verschiebt/verteilt sich tatsächlich nur in die Entwicklungsphase
- Auswirkungen von Fehlentscheidungen sind geringer und können im Entwicklungsprozess korrigiert werden
- agiles Vorgehen muss bei Vergabe / Beauftragung bereits berücksichtigt werden

# Überblick Phasen der Softwareentwicklung / -beschaffung

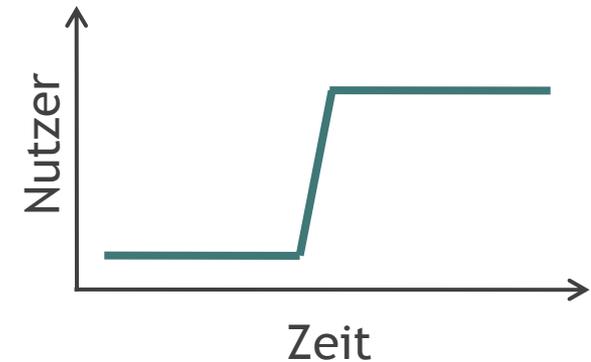


# Produkteinführung

---

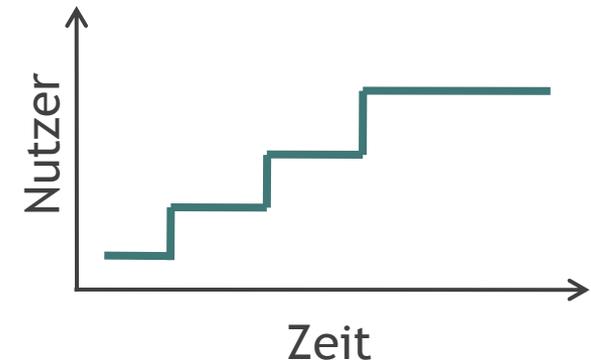
## ■ gleichzeitige Produkteinführung

- zeitgleiche Bereitstellung für alle Nutzer
- hoher Unterstützungsaufwand (Peak)
- Frustration durch Verzögerungen
- Komplikationen > Stillstand



## ■ Stufenplan bei Produkteinführung

- Bilden von Nutzergruppen
- zeitlich versetztes Einführen
- stufenweise Steigerung der Nutzeranzahl
- Unterstützungsaufwand wird zeitlich gestreckt
- Reaktionszeit verbessert
- Komplikationen besser beherrschbar



# Zusammenfassung

---

- die (umfassende) Anforderungsanalyse ist die Basis für Anbieter-bzw. Produktscheidung und Einführungserfolg
- mit Einführung von SDM-System ist i.d.R. eine Produktpassungen (Customization), d.h. Softwareentwicklung, erforderlich
- Wahl der Entwicklungsstrategie hat wesentlichen Einfluss auf den Erfolg
  - keine „beste“ Entwicklungsstrategie
  - im SDM-Umfeld haben agile Vorgehensweisen Vorteile
  - Agiles Vorgehen muss bei Beauftragung bereits berücksichtigt werden
- stufenweise (zeitlich entzerrte) Einführung ist vorteilhaft

Vielen Dank  
für Ihre Aufmerksamkeit!

