

An abstract graphic featuring a stylized cloud shape composed of overlapping translucent blue and white polygons. A network of white lines connects various points across the cloud, with some points highlighted by small colored dots in orange, green, and yellow. The background is a gradient from dark blue on the left to light blue on the right, with several small, glowing white and blue dots scattered throughout.

Business Process Engineering

Wintersemester 2021/2022

Methoden der Softwareentwicklung

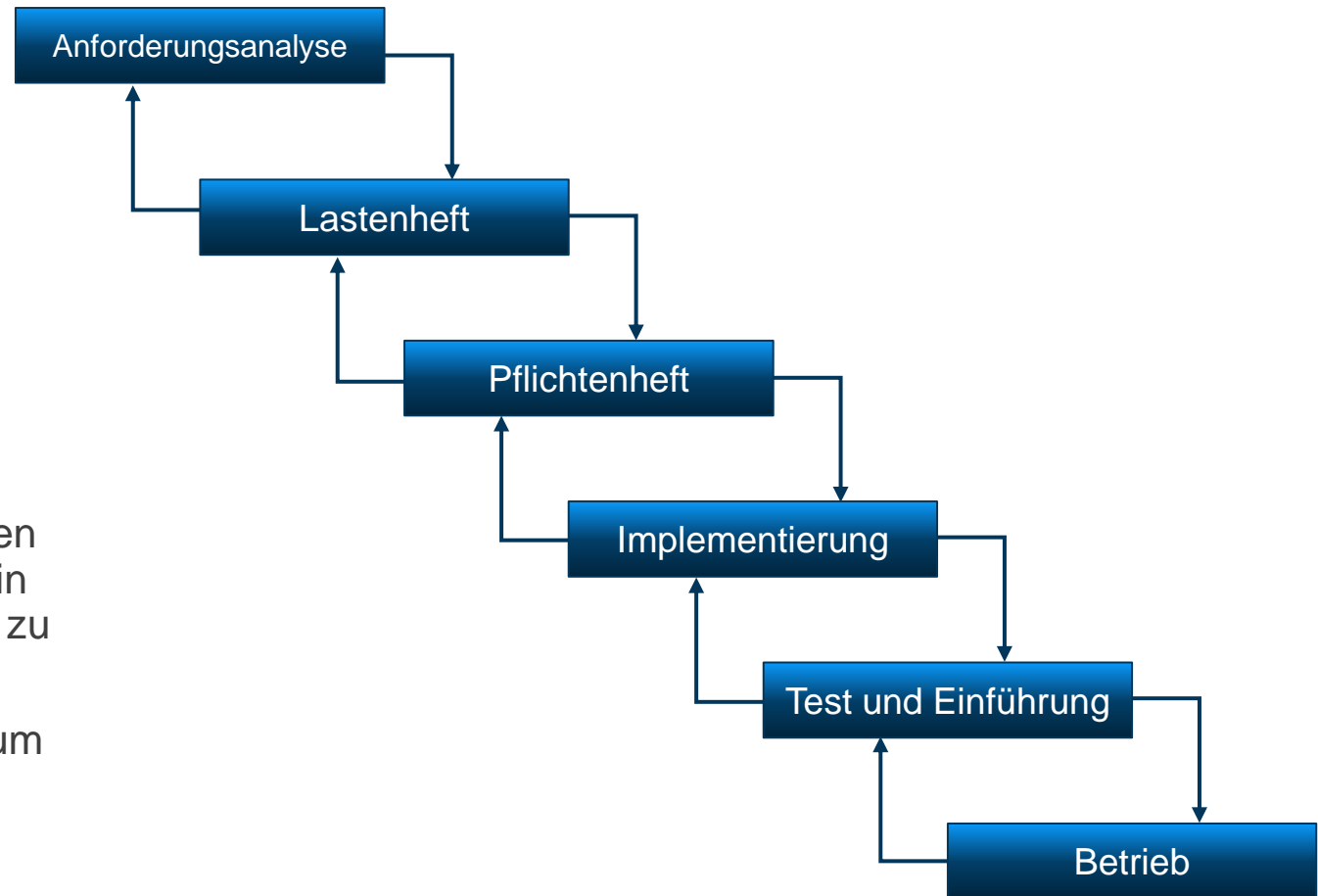
Agenda

- Softwareentwicklungsmethoden
 - Das Wasserfallmodell
 - Das V-Modell
 - Agile Methoden
 - Scrum
- UI Tests
- OData
- SAP Fiori & Demo 2



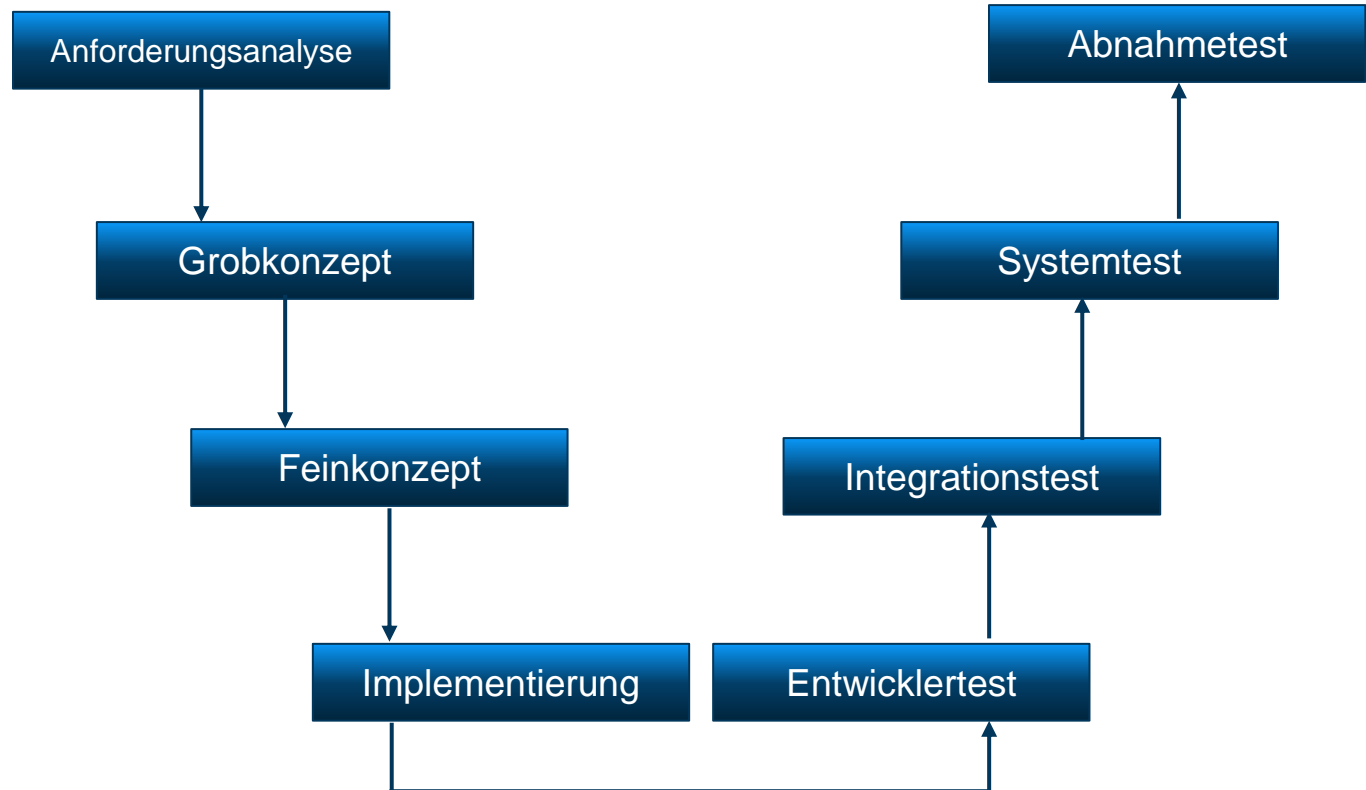
Das Wasserfallmodell (klassisch)

- Ursprung: Bau- und Produktionsprozess
- Die einzelnen Phasen werden linear nacheinander abgearbeitet
 - Lineares Vorgehensmodell
 - i.d.R. sechs Phasen
- Beim Übergang von einer Phase zur nächsten gibt es definierte Ergebnisse, die erreicht sein müssen, um in die nächste Phase wechseln zu können
- Erweiterung mit Rücksprungmöglichkeiten, um ein Problem zu beheben



Das V-Modell (klassisch)

- Basiert auf dem Wasserfallmodell, jedoch für Softwareentwicklungsprojekte entwickelt
- In DE für die Planung und Durchführung von IT-Softwareentwicklungsprojekten der öffentlichen Hand zwingend vorgeschrieben
- Testphasen werden den jeweiligen Realisierungsphasen gegenübergestellt
- Seit 2005: V-Modell XT
 - Orientiert sich mehr an der agilen Softwareentwicklung



Das V-Modell (XT)

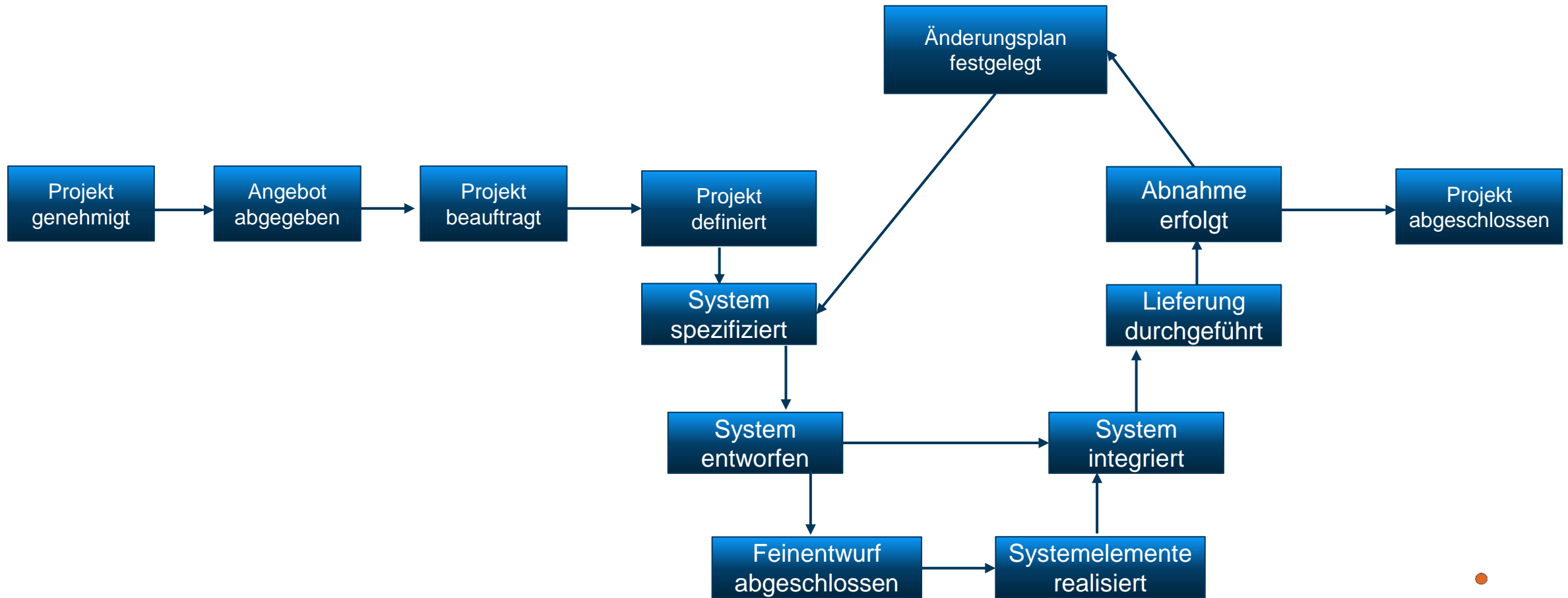
XT (eXtreme Tailoring)

- Extremes Zuschneiden auf das konkrete Projekt
 - angepasstes Vorgehen
- Flexibel nutzbar für Projekte zur (Weiter-)Entwicklung von Systemen
- Bietet keinerlei inhaltliche Unterstützung
 - methoden-neutral

Vorteile

- Gute Planbarkeit von Projekten und Vorhaben
- Hohe Projekttransparenz
- Minimierung der Projektrisiken
- Erhöhung der Qualität
- Verbesserte Kommunikation zwischen Auftraggeber und Auftragnehmer

Das V-Modell (XT)



Agile Methoden

- Orientieren sich an den klassischen Modellen
- Enge Zusammenarbeit mit dem Auftraggeber
- Spezifikation wird während der Umsetzung fortgeschrieben
 - Der Auftraggeber bekommt das was er braucht und nicht das was er irgendwann spezifiziert hat
 - neue Anforderungen können sehr leicht aufgenommen werden
- Beispiele: Scrum oder Kanban



Scrum

- 1933 formalisiert von Jeff Sutherland und Ken Schwaber
- Bekanntestes agiles Vorgehensmodell/ Rahmenwerk
- Relativ wenig vorgegebene Regeln
- Drei Artefakte
 - Product Backlog
 - Sprintbacklog
 - Produktinkrement
- Ziel: komplexe Lösungen in vielen kurzen Iterationen erstellen



Scrum

Sprint

- Kurze Iteration fester Länge (meist drei bis vier Wochen)
- Jeder Sprint soll eine auslieferbare Softwarelösung erzeugen = **Inkrement**
- Mit jedem Sprint wächst der Funktionsumfang der Softwarelösung

Product Backlog

- Verantwortung liegt beim *Product Owner*
- Hält alle bekannten Anforderungen und Arbeitsergebnisse fest, die der Erreichung des Projektziels dienen

Sprint Backlog

- hält Aufgabenmenge überschaubar, wird während des Sprints nicht verändert
- Legt fest, welche Anforderungen in einem Sprint umgesetzt werden
- Nur Anforderungen, die genau genug formuliert wurden (Definition of Ready)
- Kriterien zur Beurteilung der erfolgreichen Fertigstellung des Sprints (Definition of Done)

Transparenz

- Der aktuelle Stand muss für alle sichtbar sein
- Im **Daily Scrum** wird der aktuelle Stand täglich und gemeinsam besprochen

Scrum

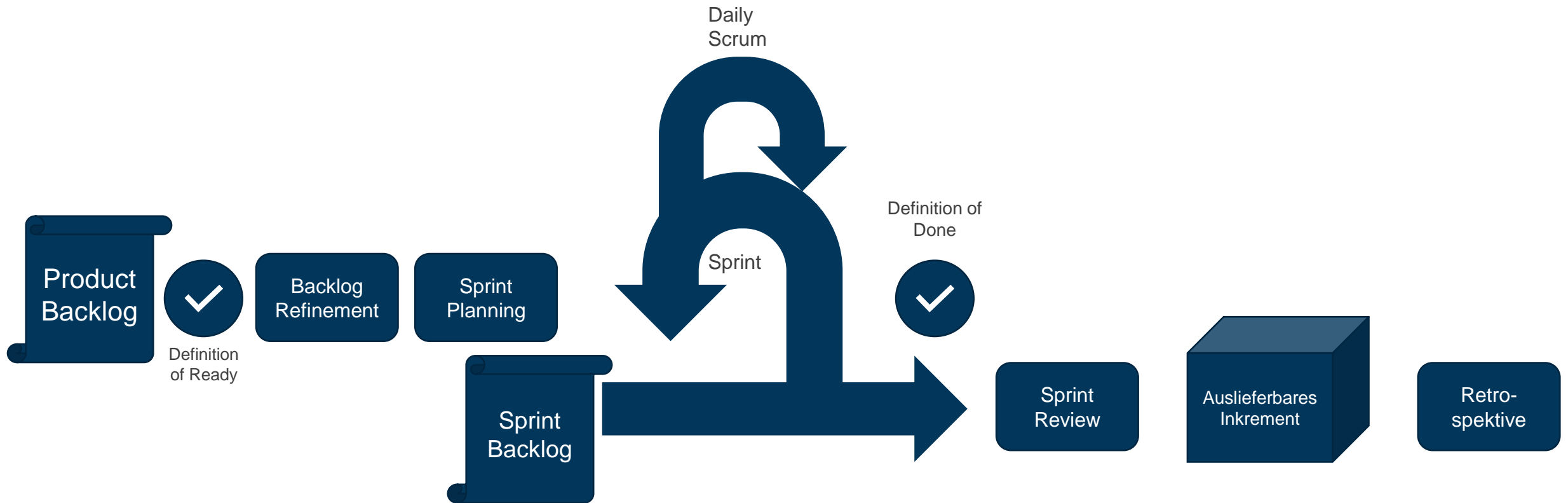
Rituale

- **Backlog Refinement**
 - Verfeinerung der vorhandenen Anforderungen
 - Beseitigung von Unklarheiten
- **Sprint Planning**
 - Die Aufgaben für einen Sprint festlegen, d.h. Erstellen des ***Sprint Backlogs***
- **Daily Scrum**
 - Zum Besprechen der Aufgaben die die einzelnen Teammitglieder an einem Tag erledigt haben
- **Sprint Review**
 - Zum Besprechen der erledigten Aufgaben und des Inkrements
- **Sprint Retrospektive**
 - Nachbesprechung des vergangenen Sprints
 - Verbesserungen für den nächsten Sprint besprechen

Rollen

- **Product Owner**
 - Führt und verantwortet das ***Backlog***
 - Vertritt die Belange des Kunden
- **Scrum Master**
 - Sorgt dafür, dass die ‚Spielregeln‘ eingehalten werden
 - Moderator bei den Meetings
 - Kümmert sich um die Behebung von Störungen
- **Entwicklungsteam**
 - Für die Lieferung der Softwarefunktionalität verantwortlich
 - i.d.R. fünf bis neun Personen
 - Organisiert sich selbst

Der Scrum Prozess



REST

- **Representational State Transfer**
- Paradigma für Softwarearchitektur
 - Definiert die Kommunikation zwischen verteilten Systemen
 - Weit verbreitet
- Es muss folgendes gegeben sein, damit eine Architektur als REST-konform gilt:
 - Client-Server-Architektur
 - Zustandslosigkeit
 - Pufferbarkeit
 - Mehrschichtigkeit
 - Einheitliche Schnittstellen
 - Optional: Programmcode bei Bedarf



OData

- Open Data Protocol
- 2007 Ursprünglich von Microsoft entwickelt
 - v1, v2 und v3 wurden unter der Microsoft Open Specification Promise veröffentlicht
 - V4 wurde von der „Organization for the Advancement of Structured Information Standards“ (OASIS) herausgebracht
- Basiert auf REST
- Unterschied v4 zu v2:
 - Erweiterte Abfragesprache
 - Vereinfachte Syntax
 - Vereinfachte Payloads
 - Standardformat ist JSON
 - XML ist optional

Beispiel: <https://services.odata.org/V2/OData/OData.svc>

SAP Fiori & OData

- Warum sollten bei der Verwendung von Fiori auch OData Services verwendet werden?
 - Volles UI5 Potential ausnutzen
 - Fiori Datenmodell baut auf OData auf
 - Einfacher in der Verwendung als Eigenentwicklung
 - Großer Aufwandsunterschied

Fazit: Einsatz von OData reduziert den Aufwand drastisch!

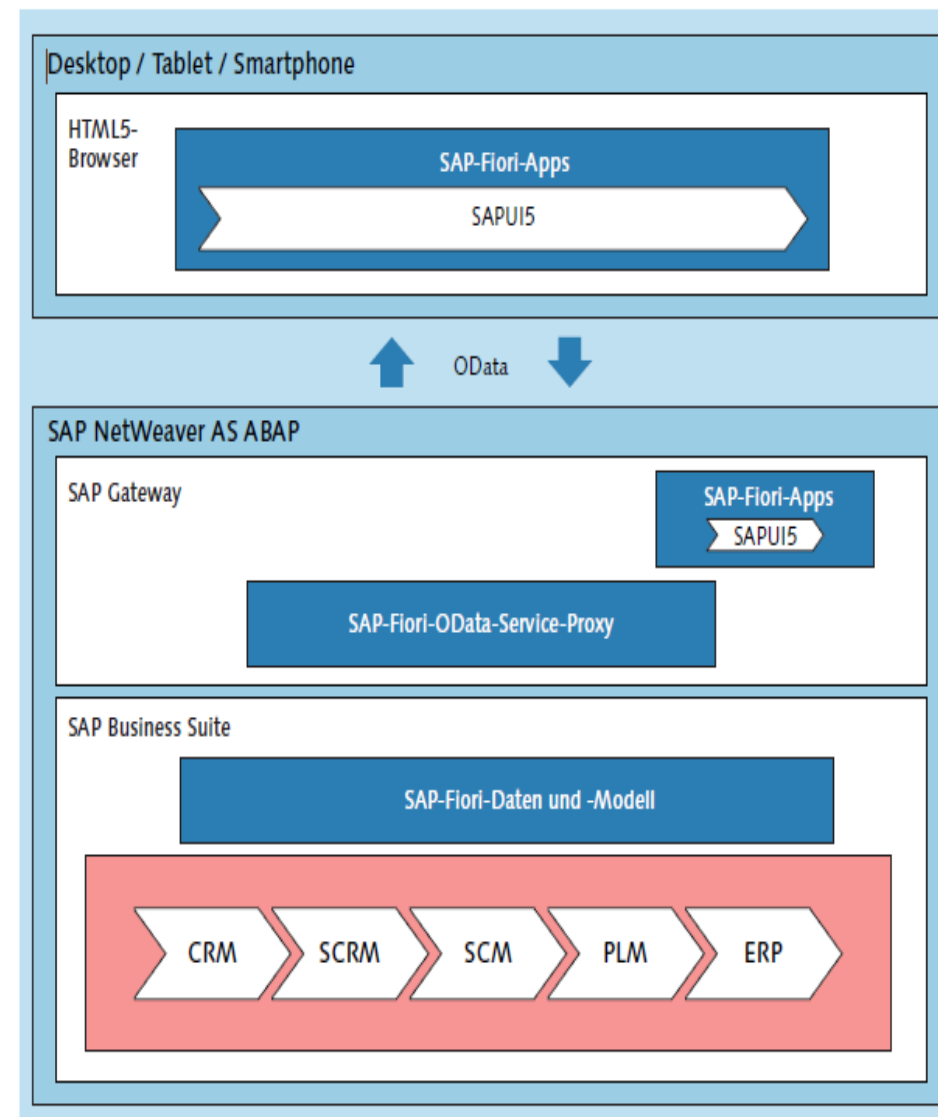


Abbildung 10.2 SAP-Fiori-Architektur

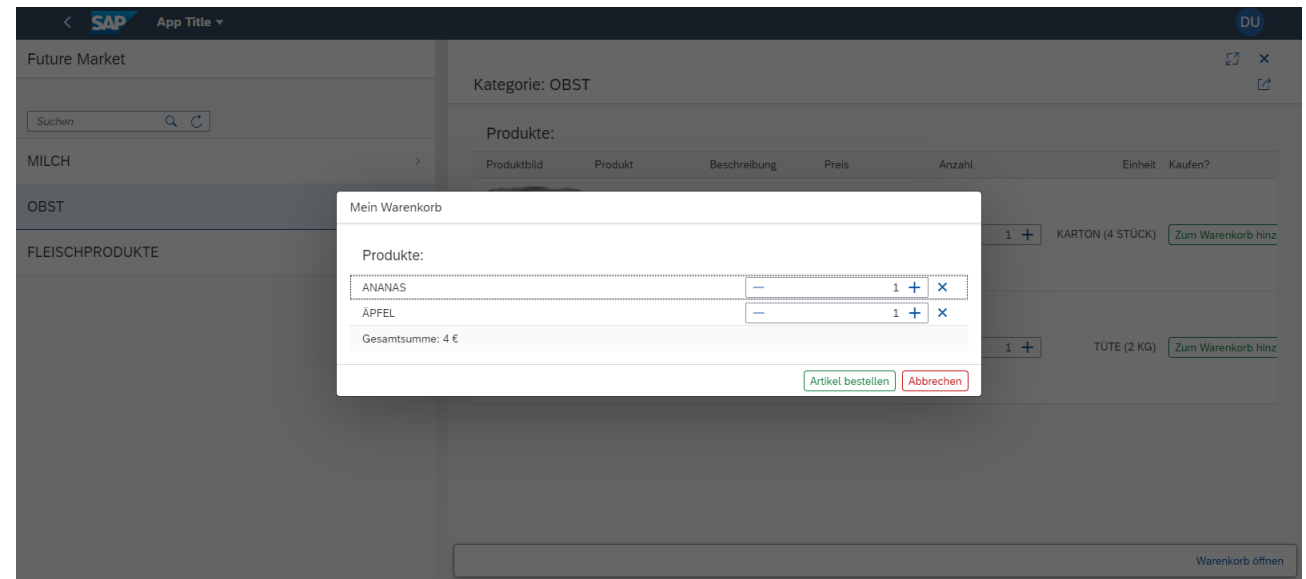
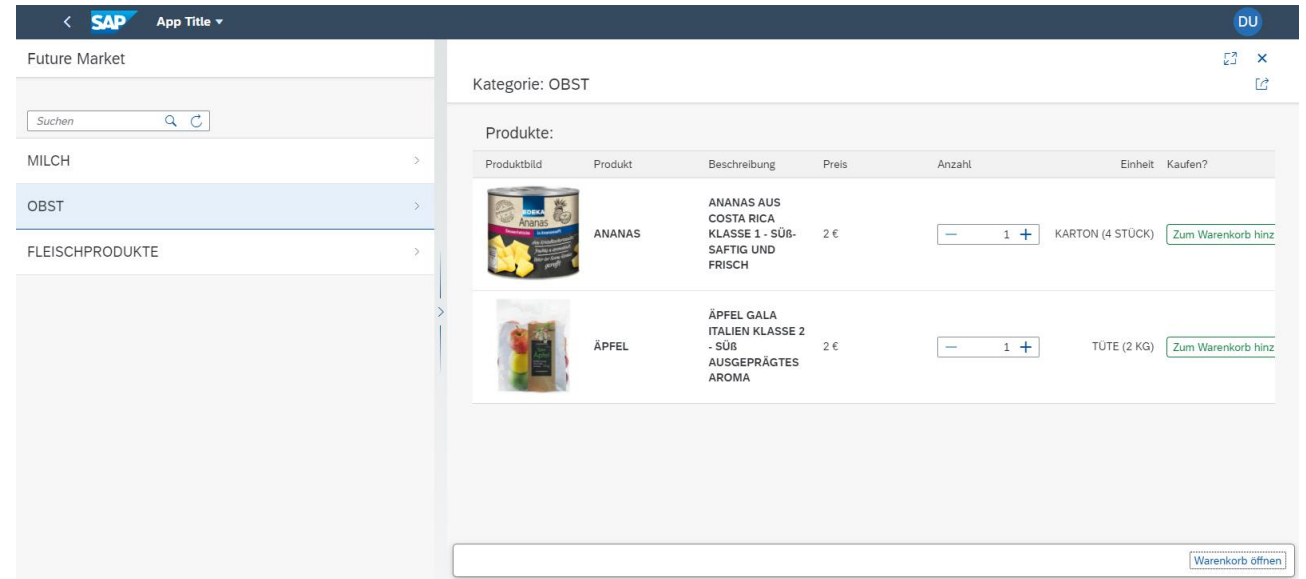
SAP Fiori Demo

Tobias Müller



Aufgabe 8!

- Erweitert die in Aufgabe 7 erstellte Applikation



Quellen

- „*SAP-Testmanagement*“, Buch, SAP Press
- „*OData und SAP Gateways*“, Buch, SAP Press
- <http://docs.oasis-open.org/OData/new-in-OData/v4.01/new-in-OData-v4.01.html>
- https://www.google.com/url?sa=t&source=web&rct=j&url=https://d117h1jjiq768j.cloudfront.net/docs/default-source/next/presentations--wednesday/dci_the-api-debate---graphql-vs-OData.pdf%3Fdownload%3Dtrue&ved=2ahUKEwjgoZTxqIHiAhVb4XMBHUVPCeMQFjAJegQIAhAB&usg=AOvVaw3w4nli04rsxrFk_xgxkTt
- <https://www.progress.com/blogs/whats-new-with-odata-4-odata-2-vs-odata-4>

An abstract graphic of a cloud shape, split vertically. The left side is dark blue and the right side is light blue. The cloud is overlaid with a white network of lines and nodes, some of which are highlighted with small colored dots (orange, green, yellow).

Business Process Engineering

Wintersemester 2021/2022

Dr. Andreas Scharf