



**DR. THOMAS + PARTNER**  
GmbH & Co. KG [www.tup.com](http://www.tup.com)



Vorlesung:



**Institut für Fördertechnik  
und Logistiksysteme**  
Universität Karlsruhe (TH)

# IT-Grundlagen der Logistik 2016

## **Kapitel 6: Softwareentwicklung nach industriellen Maßstäben**

Prof. Dr.-Ing. Frank Thomas  
Dr. Thomas + Partner GmbH & Co. KG, Karlsruhe

Karlsruhe, den 20.07.2016

Integriert

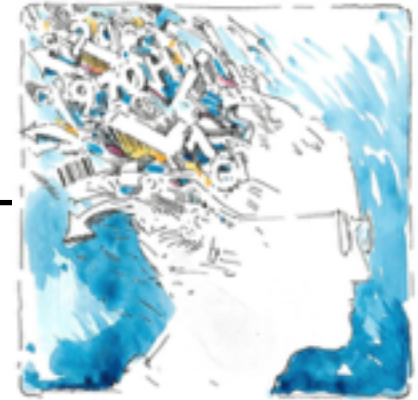
Realisiert

Geplant

1



# Schriftliche Prüfungen



Fachrichtung

Wirtschaftsingenieurwesen (Dauer: 120 min)

**Klausurtermin:**  
**Dienstag, der 20. September 2016 von**  
**08.00 - 10.30 im Gerthsen Hörsaal.**

Die Vorbesprechung zur Klausur findet am Mittwoch,  
den 07. September 2016 um 16.00 Uhr in der  
Firma DR. THOMAS + PARTNER statt.  
(Fragestunde und Stoffabgrenzung)



# Lösungsansatz Adaptive IT

---

In den vorausgegangenen Kapiteln wurde mit der Herleitung der Geschäftsprozessmodule die Basis für die Entwicklung wiederverwendbarer adaptiver IT-Prozessbausteine geschaffen.



# Definition Softwarearchitektur nach IEEE1471

---

Wiederverwendbarkeit, Änderbarkeit und Erweiterbarkeit eines Softwaresystems wird durch die Softwarearchitektur bestimmt.

Die Softwarearchitektur ist die grundlegende Organisation eines Systems und wird verkörpert

- durch ihre Komponenten
- die Beziehungen untereinander und zur Umgebung
- und die Prinzipien, die den Entwurf und die Evolution leiten



# Software-Technik

---

In der Softwaretechnik sind unter anderem zwei Entwicklungen festzustellen:

❑ **Komplexität der Software:**

Der Trend geht dahin, dass Software immer komplexer wird. Daraus folgt eine zunehmende Bedeutung der Softwarearchitektur. Eine „gute“ Softwarearchitektur schafft Transparenz.

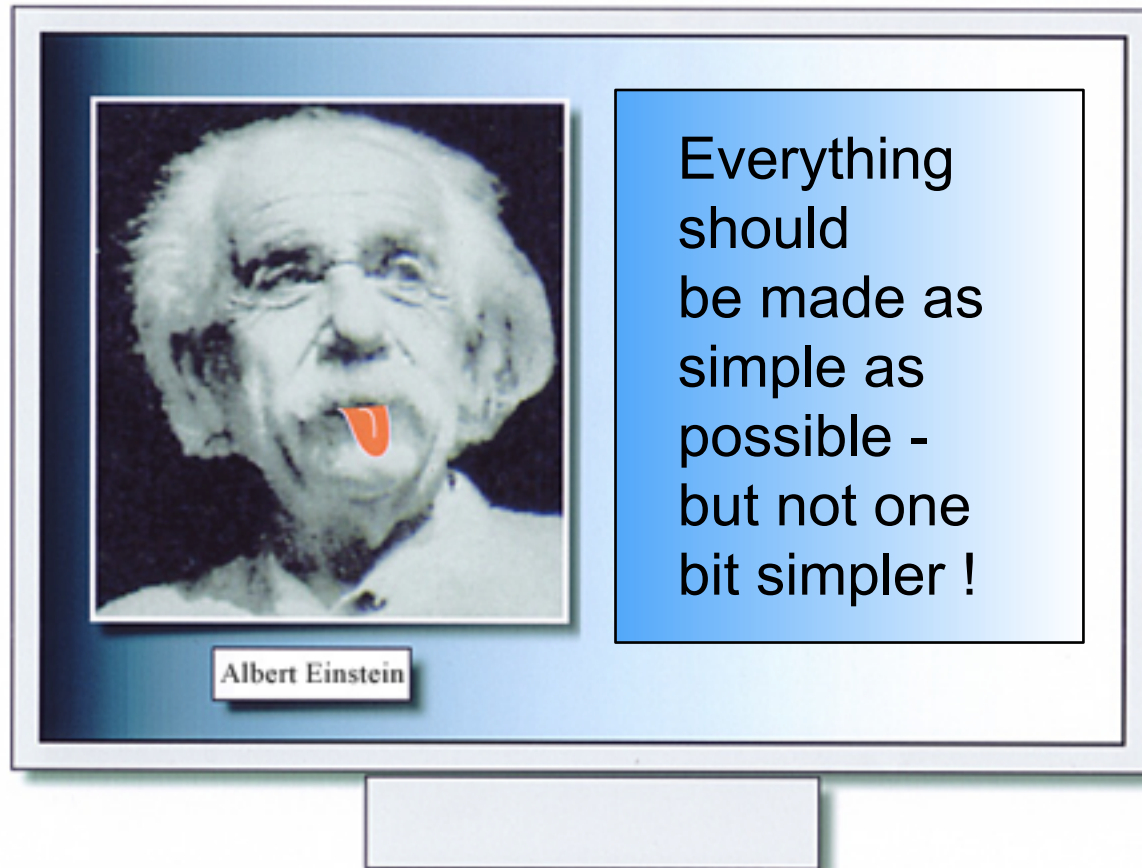
❑ **Software muss sich an ständige Veränderungen anpassen:**

Neue Marktanforderungen oder Kundenwünsche müssen während oder nach der Entwicklungsphase ohne großen Aufwand umsetzbar sein (Entwicklungsprozess mit objektorientierten Werkzeugen). Es kann sonst ein „Big Ball of Mud“ entstehen, eine gewucherte Software. Eine „gute“ Software wirkt dem entgegen.



# Komplexität beherrschen

Hier hilft der Ansatz:





# Objektorientierung

---

In der Softwareentwicklung werden objektorientierte Methoden umgesetzt:

- zur Verbesserung von Produktivität, Wartbarkeit und der Software-Qualität
- damit adaptive IT-Prozessbausteine möglichst oft wiederverwendet werden können. (Objektorientierte Framework)

Die entwickelten Objekte sollen die „reale Welt“ abbilden.



# A warehouse is not a warehouse

---

Die Informatik sorgt nicht für das Verständnis des Problems, sondern gibt Methoden an, auf die dann jedoch die Logistiker angewiesen sind, um ihre Kerngeschäftsprozesse eines WMS einer Lösung zuzuführen.





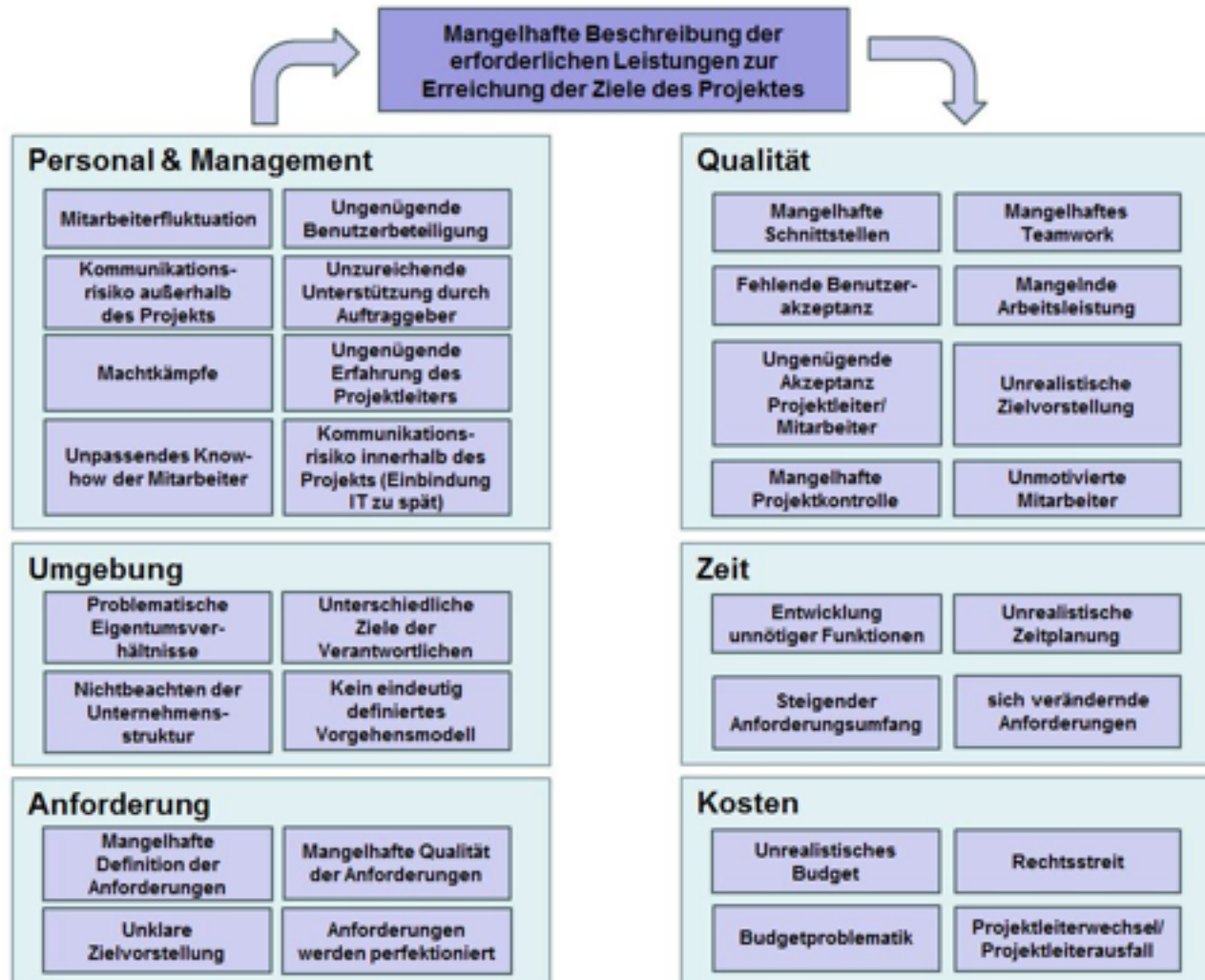
## software follows function

---

**Software follows function** gilt dann, wenn in der Planungsphase alle Projektanforderungen dokumentiert wurden und gemeinsam im interdisziplinären Team aus Logistik-Planern, von dem Kunden/Nutzer und dem IL (Implementierungs-Leiter) unterschrieben wurden. Das Know-How des IL wird frühzeitig in die Prozessgestaltung und Anforderungsaufnahme mit einbezogen.



# Risikokarte





## Risikomanagement in der Pflichtenheft-Phase

---

Die linken Felder „Personal & Management“ „Umgebung“ und „Anforderung“ bilden die Vorgängerrisiken, während der rechte Block möglicher Nachfolger aufdeckt.

Das alleinige Aufdecken von Risiken ist noch nicht ausreichend, um ein erfolgreiches Risikomanagement zu implementieren.



# Qualität der Software-Architektur

---

Als Maß für die Qualität der Software-Architektur gelten:

Mittelbares Maß:

Performance, Sicherheit, Verfügbarkeit und Zuverlässigkeit,  
Robustheit, Funktionsumfang, Benutzbarkeit.

Unmittelbares Maß:

Flexibilität, Testbarkeit, Integrierbarkeit, Wartbarkeit, Änderbarkeit,  
Portierbarkeit, Skalierbarkeit, Wiederverwendbarkeit.

➔ **Frage nach der richtigen Software-Architektur!**



# Softwaretechnik (I)

---

Helmut Balzert definiert Softwaretechnik als zielorientierte Bereitstellung und systematische Verwendung von Prinzipien, Methoden und Werkzeugen für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von umfangreichen Softwaresystemen.

Zielorientiert bedeutet die Berücksichtigung von:

**Kosten, Zeit und Qualität**



## Softwaretechnik (II)

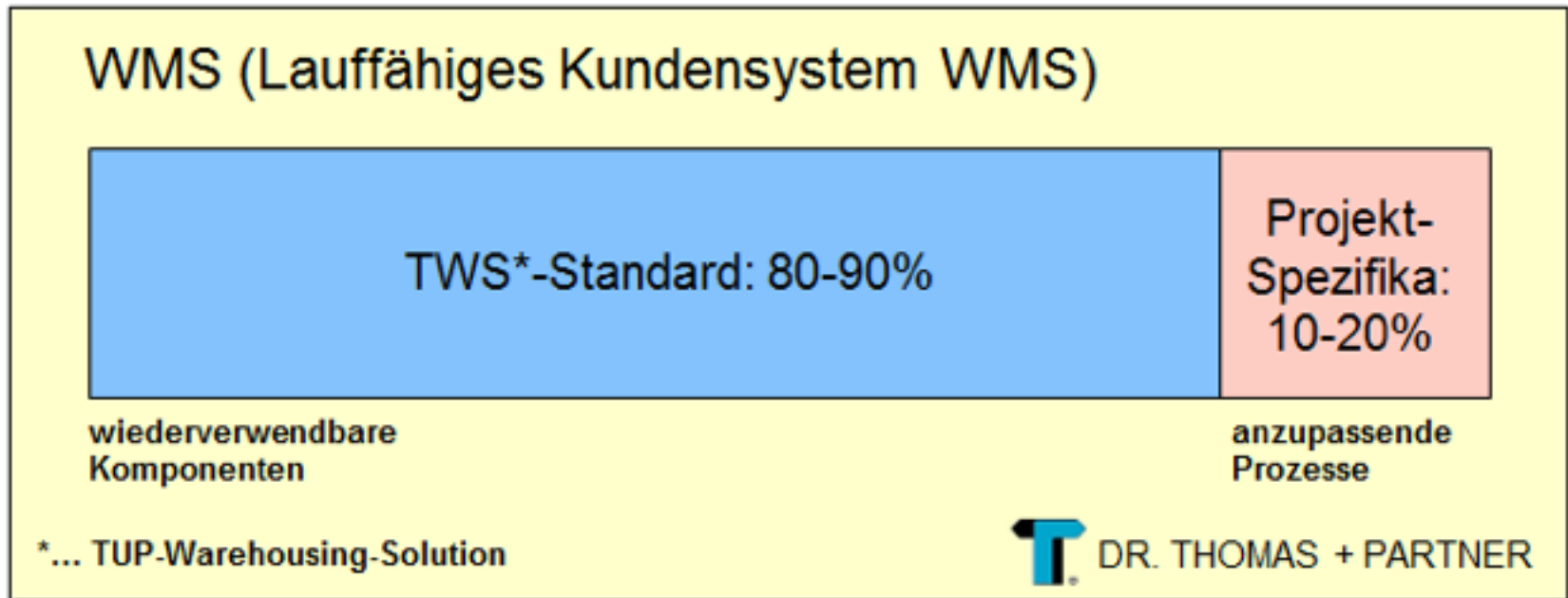
---

Zwei Eigenschaften einer adaptiven IT-Lösung sind für ein lauffähiges Kundensystem notwendig

- ❑ die Wiederverwendung von IT-Prozessbausteinen
- ❑ und die Anpassung „nicht“-gleichartiger Geschäftsprozesse durch Veredelung initiiert sofort die Frage nach der richtigen Softwarearchitektur

# Präferierte Zielvorstellung für ein Lauffähiges Kundensystem

---



Frage nach der richtigen Software-Architektur



# Entwicklungsprozess mit objektorientierten Werkzeugen

---

Entwicklungsprozess mit objektorientierten Werkzeugen:

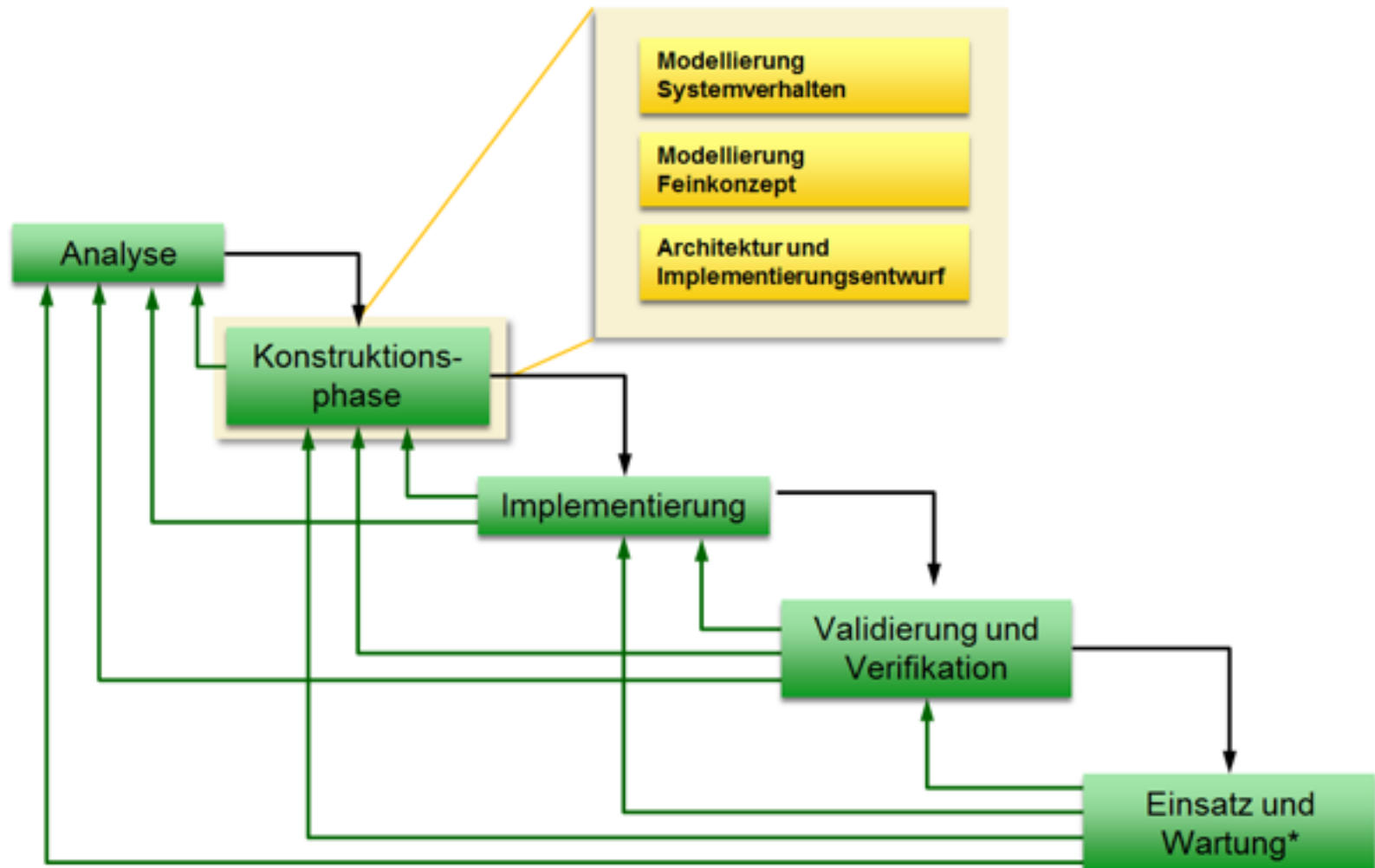
Neu dabei ist:

- ▶ Man verlässt das Wasserfallprinzip
- ▶ Im iterativen Prozess nimmt man in jeder Phase Unvollständigkeiten bewusst in Kauf
- ▶ Die Rückkehr zu jeder Phase wird durch Werkzeuge (case tools) unterstützt, die eine permanente Konsistenzprüfung des Gesamtsystems zulassen





# Iteratives Vorgehensmodell





# Vorgehensmodell: Prinzip der agilen Methoden

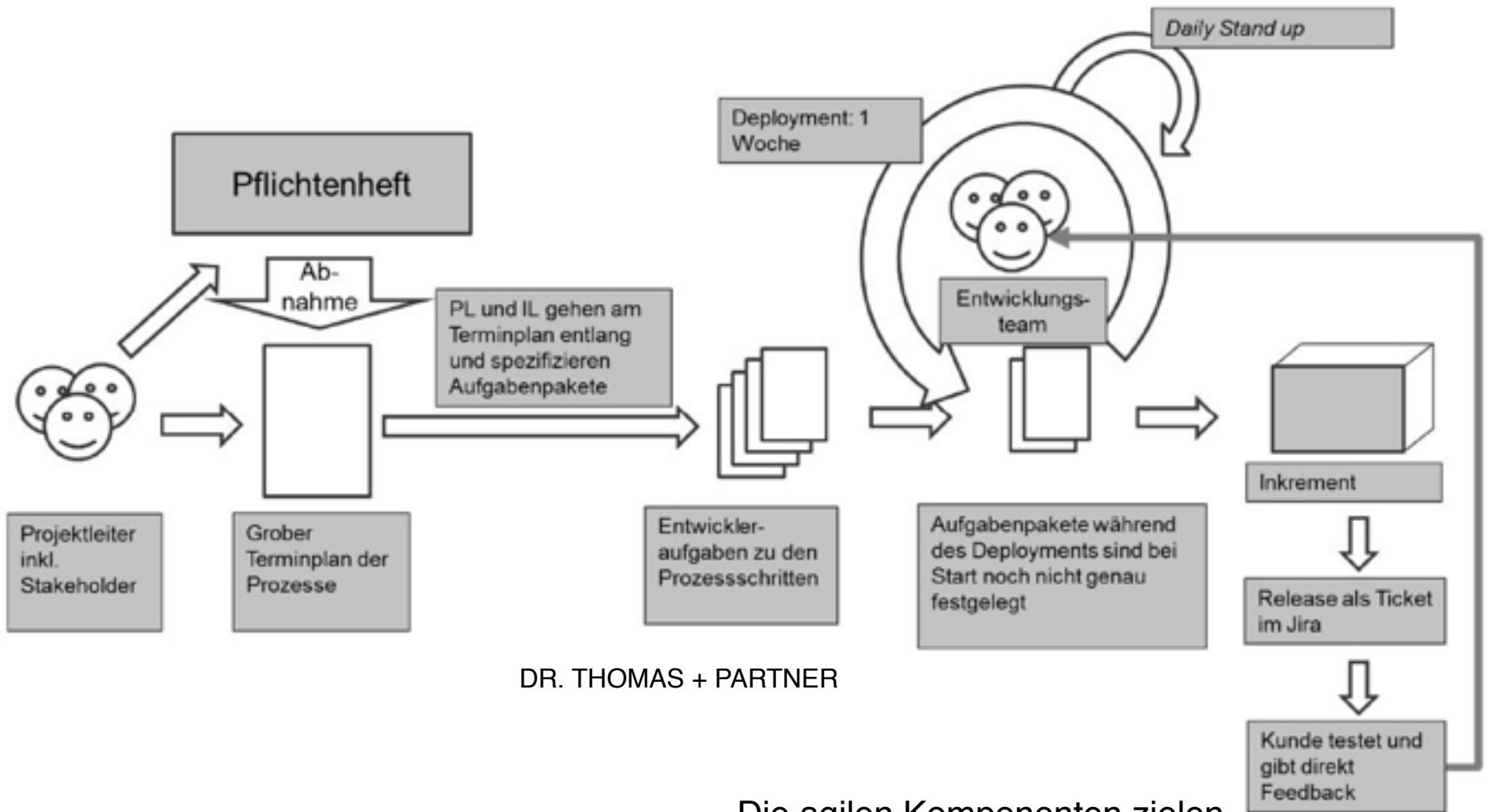
---

Die agilen Komponenten zielen auf die Software-Entwicklung **aber nicht** auf die Anforderungsdefinition.

Die Basis der agilen Vorgehensweise bildet ein vollständiges, gut dokumentiertes Pflichtenheft und unterschriebenes Pflichtenheft wie bei den klassischen Vorgehensmodellen.



# Agile Vorgehensweise bei DR. THOMAS + PARTNER



DR. THOMAS + PARTNER

Die agilen Komponenten zielen auf die Software-Entwicklung ab!



# Erfolg verleiht Flügel

- ▶ das Team macht etwas Spannendes
- ▶ etwas wofür Gehirnschmalz gebraucht wird
- ▶ wo Ideen zählen
- ▶ auf jeden kommt es persönlich an
- ▶ das Team verstärkt sich durch gute, hochmotivierte Mitarbeiter.  
"Wer will keinen Erfolg haben" ?





# Objektorientierung

---

In der Softwareentwicklung werden objektorientierte Methoden umgesetzt:

- zur Verbesserung von Produktivität, Wartbarkeit und der Software-Qualität
- damit adaptive IT-Prozessbausteine möglichst oft wiederverwendet werden können. (Objektorientierte Framework)

Die entwickelten Objekte sollen die „reale Welt“ abbilden.



# Objektorientiertes Strukturmuster - Wiederverwendbare Transportverwaltung

---

Das Standardmodul „Transportverwaltung“ ist die zentrale Instanz zur Überwachung, Beauftragung und Koordination aller Transportaufträge und Transportressourcen.  
(siehe im Skript Kapitel 2.1 und ff.)



# Aufgaben der Materialflusssteuerung

---

Die wichtigste Aufgabe der MFCS ist die Beauftragung von Fördersystemen mit Fahraufträgen in einer Weise, die die Anlage optimal auslastet und die logistischen Prozesse termingerecht bedient.



# Abbild der Förderanlage in der MFCS

---

## Hierarchisches Konzept:

### 1 Modellentwicklung

Fördertechnik wird als Netz von Wegen und Punkten abgebildet

### 2 Rekursives Routing

Aus den statischen Daten wird durch Routing der kürzeste mögliche Weg zum Endziel berechnet

### 3 Die **Routentabelle** wird im Rechner als Tabelle im Hauptspeicher abgelegt

siehe auch Kapitel 2 im Skript





# MFCS: Statisches Abbild der Förderanlage

---

Dem MFCS liegt das statische Abbild der Förderanlage zugrunde. Diese Anlagenmatrix besteht aus folgenden statischen Daten:

- Anlagepunkte, Wege als gerichtete Verbindungen zwischen zwei Punkten
- Fahrplan-Einträge. Hier werden alle anlagemöglichen Transportalternativen aufgelistet



# MFCS: Dynamisches Abbild der Anlagenressourcen

---

Mit der Anlagenmatrix und dem momentan vorliegenden Betriebszustand (dynamisches Abbild der Anlagenressourcen) wird zu einer freien Ressource:

- Förderstrecke, Stellplatz, Fördermittel, Stapler, etc.
- der Weitertransport eines Transportguts beauftragt

Die Kapazität wird bestimmt durch den Typ des Bereichsverwalters bzw. durch die Aufnahmefähigkeit des Endpunktes des aktuellen Weges.



# Funktion Bereichsverwalter

---

Hinter jedem Bereichsverwalter (Transporteurs) stehen standardisierte Module:

- Regalbediengeräte (mehrfachtiefe Einlagerung)
- Palettenfördersysteme,
- Taxibetrieb für Bereiche mit FTE, Elektrohängebahnen, Elektro-Palettenbodenbahnen
- Direktbetrieb von Behälterfördersystemen
- Sorter-Systemen
- sowie Shuttle-Systemen



# Shuttle Systeme

... siehe LOGISTIK KNOWHOW: [www.logistikknowhow.com/shuttle](http://www.logistikknowhow.com/shuttle)

LOGISTIK  
KNOWHOW

Suchen

Aktuelle Themen

Bestandsverwaltung

Codieretechnik

E-Commerce

Einlagerung

Energie und Umwelt

Erfassungssysteme

Forschung

Geschichtliche Entwicklung

Informationssysteme

IT und Software

Kennzahlen

Kommissionierung

Lagerarchitektur

Lagerungstechniken

Materialfluss und Transport

Nachschub

Planung und Organisation eines  
Lagers

Recht in der Intralogistik

## Shuttle



Ein Shuttle ist ein Fahrzeug, das zur automatischen Bedienung von Lagern eingesetzt wird. Ein Shuttle wird in der Lagertechnik auch als Satellitenfahrzeug oder Kanalfahrzeug bezeichnet.

Ein Shuttle ist ein Lagergerät, mit dem Ladeeinheiten ein-, um- und ausgelagert werden können. Die Ein-, Um- und Auslagerleistung eines Lagersystems hängt von folgenden Faktoren des eingesetzten Lagergeräts ab:

- Konstruktion
- Fahrverhalten
- Gangebundenheit
- Lastaufnahmemittel

Ein Shuttle ist entweder ein autonomes Fahrzeug mit einer integrierten Batterie oder aber es ist über eine elektrische Verbindung an das Umsetzgerät angeschlossen, um mit Energie versorgt zu werden.

### Einsatz eines Shuttle-Fahrzeugs

Lagerkanäle, in denen die Ladeeinheiten mehrfach tief eingelagert werden.

### Satellitenregallager

Das Fahrzeug löst sich nach dem Erreichen der x-/z-Koordinate des Lagerfaches vom Regalförderzeug und fährt in das Regalfach ein. Es ist mit einer Hubeinrichtung ausgerüstet und kann selbstständig die Ladeeinheiten auf- und abladen. Die Querträger des Regallagers besitzen zwei horizontale Flächen: Die obere Fläche ist die Auflagefläche für die Ladeeinheit und die untere Fläche wird als Lauffläche für das Shuttle genutzt.

### Kanalregallager

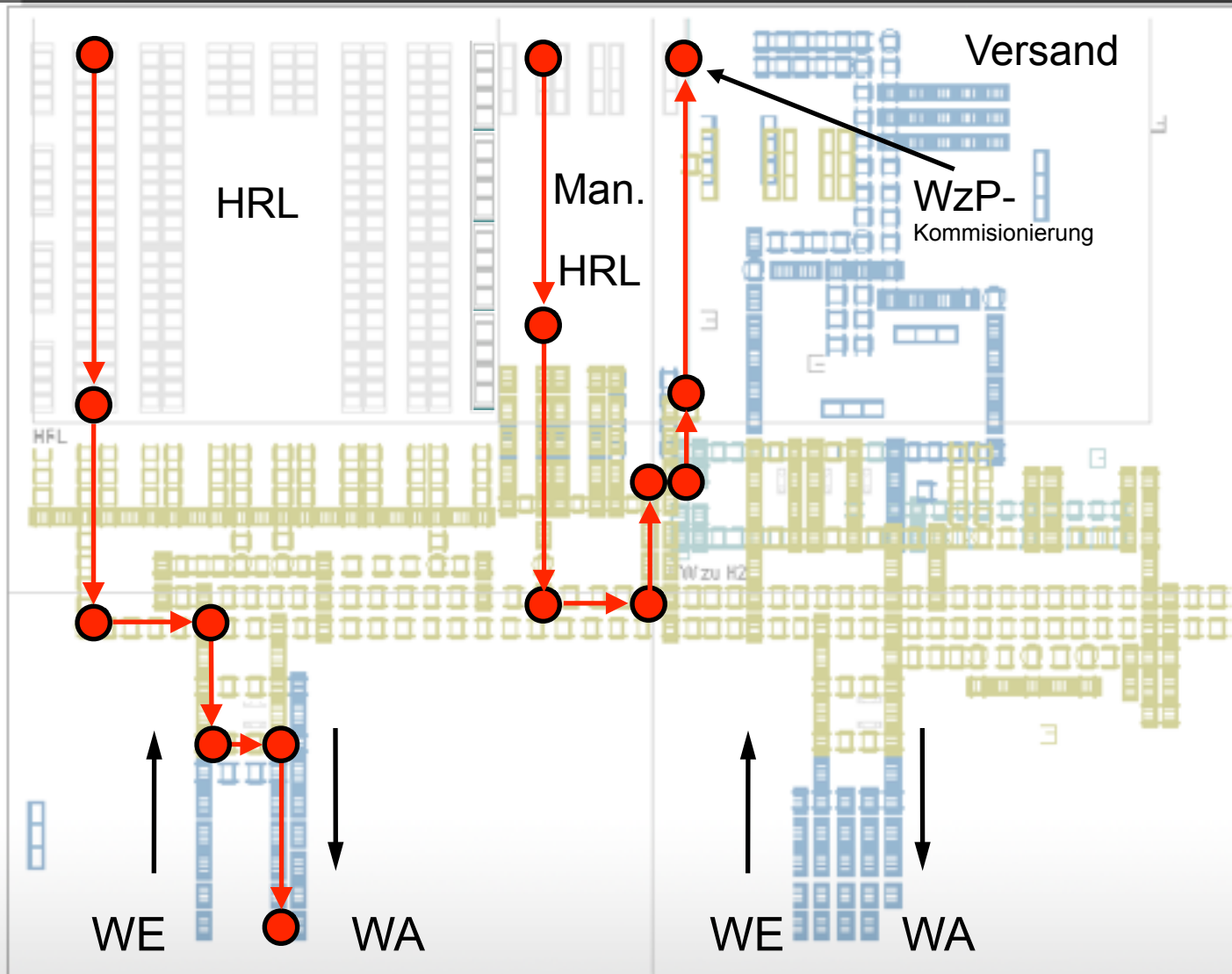
Die Ladeeinheiten des Kanalregallagers können im Kanal verschoben werden. Bei Einschubkanälen werden die Ladeeinheiten von einer Seite ein- und ausgelagert. Bei Durchlaufkanälen werden die Einheiten hingegen von entgegengesetzten Richtungen ein- und ausgelagert. Ein Beispiel für ein Kanalregallager ist ein aktives Durchlaufregallager.

Shuttle-Fahrzeuge können außerdem genutzt werden, um Umschlagvorgänge einzusparen, indem die Ware über ein Schienensystem direkt zum Zielort transportiert wird.

Mehr Informationen finden Sie unter [Automatisches Kleinteilelager](#).



# Anlage (Teilansicht)



Integriert

Realisiert

Geplant



# Parametrierung, Anpassung, Änderung

---

- ❑ Der größte Teil der Funktionen eines MFCS kann entworfen und ausprogrammiert werden, soweit er sich auf die Ressourcenbelegung und die Beauftragung bisher bekannter Fördersysteme bezieht
- ❑ Anlagen-spezifische Anpassung kann in der Laufzeitapplikation durch Parametrierung vorgenommen werden



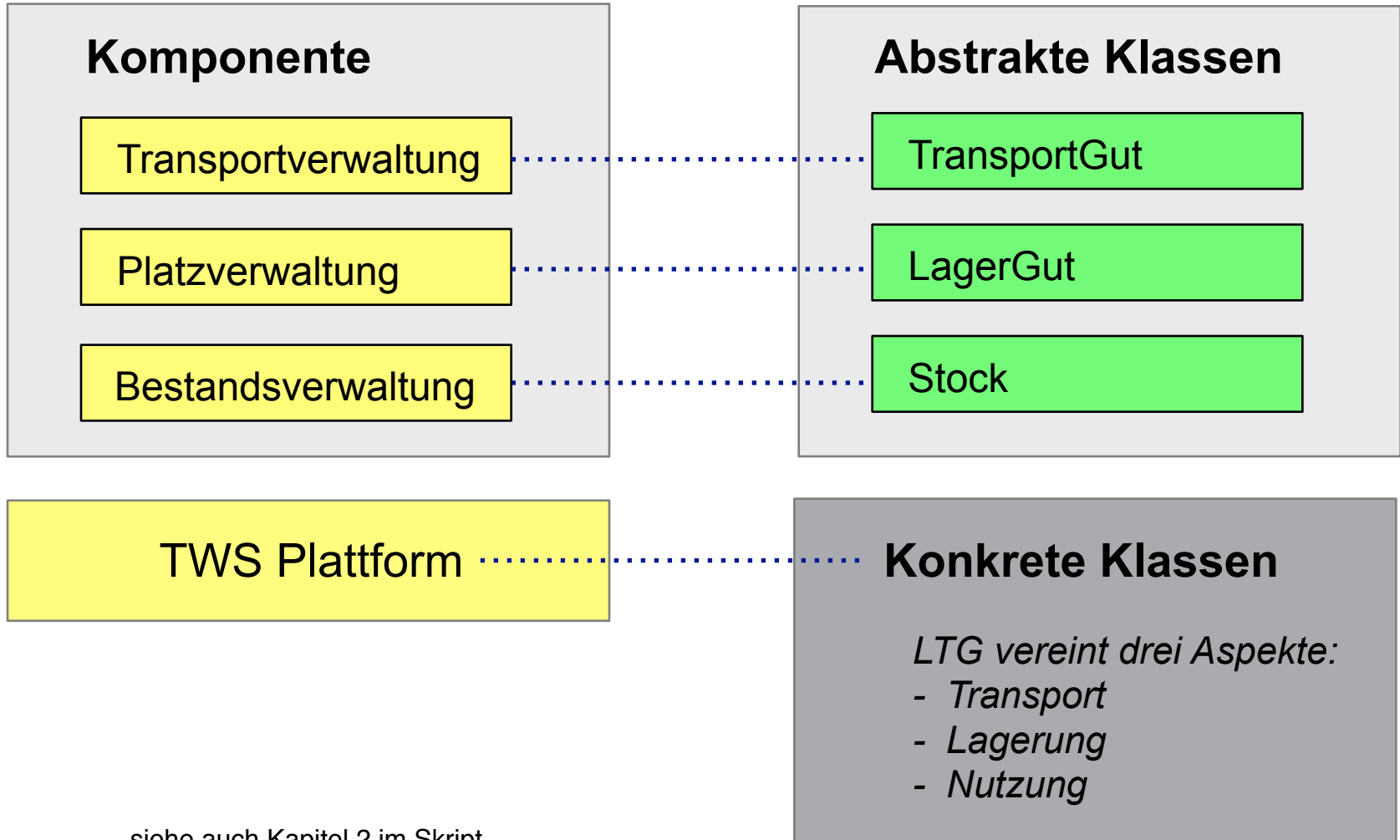
# Projektspezifische Erweiterungen

---

Durch den Einsatz des Entwurfsmuster Brücke wird die Abstraktion der Klassen von der Implementierung entkoppelt um beides unabhängig variieren zu können



# Klasse LTG (LagerTransportGut)



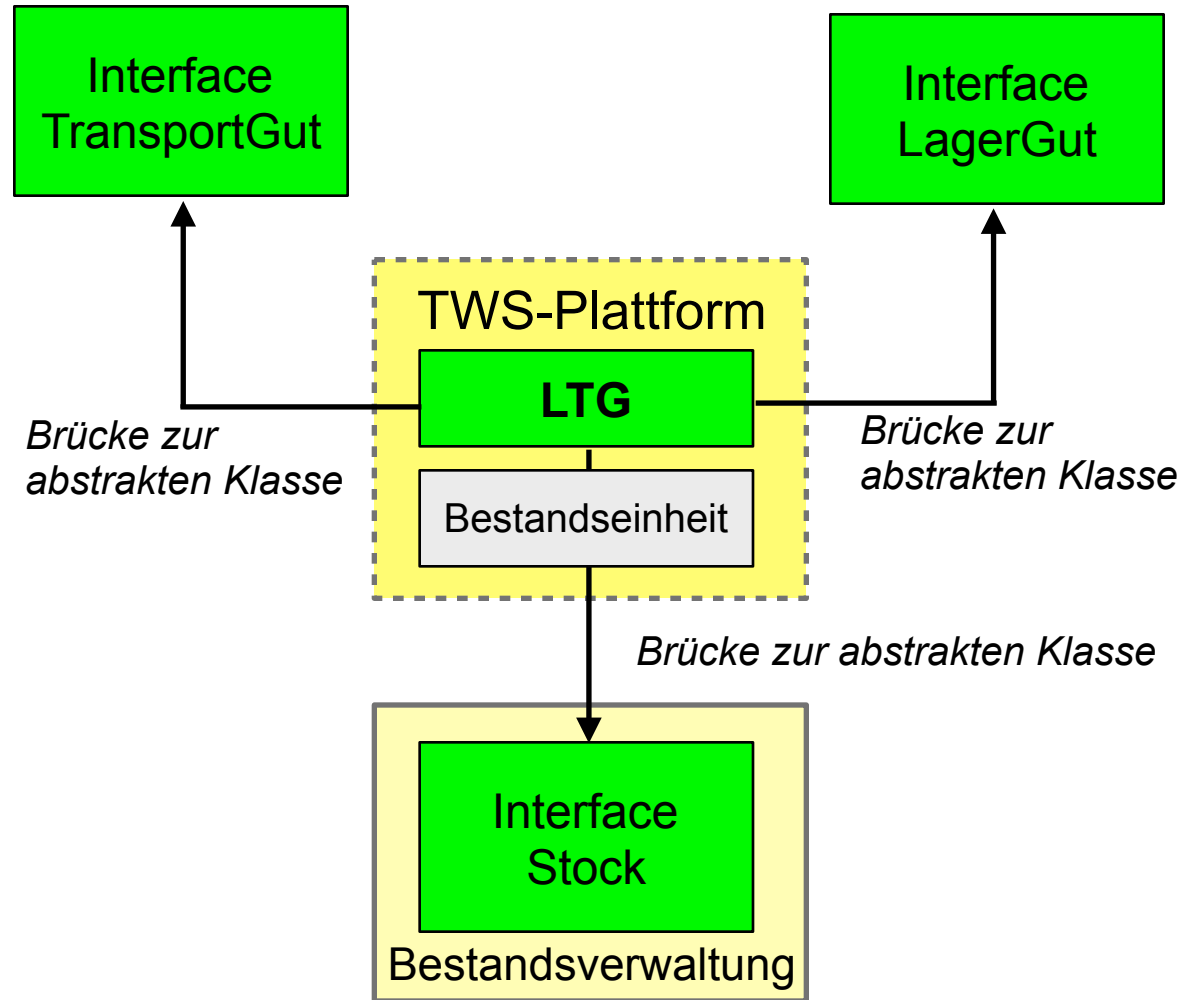
siehe auch Kapitel 2 im Skript





# Verbindung von Transportverwaltung und Platzverwaltung über die Klasse LTG

**Klassendiagramm:  
Anwendung des  
Brückenmusters in  
der Bestandsverwaltung**



siehe auch Kapitel 2 im Skript



**Softwareentwicklung nach industriellen Maßstäben,  
erhöht die Planungsintelligenz bei Intralogistik-Systemen**



# Framework (Rahmen, Gerüst, Skelett)

---

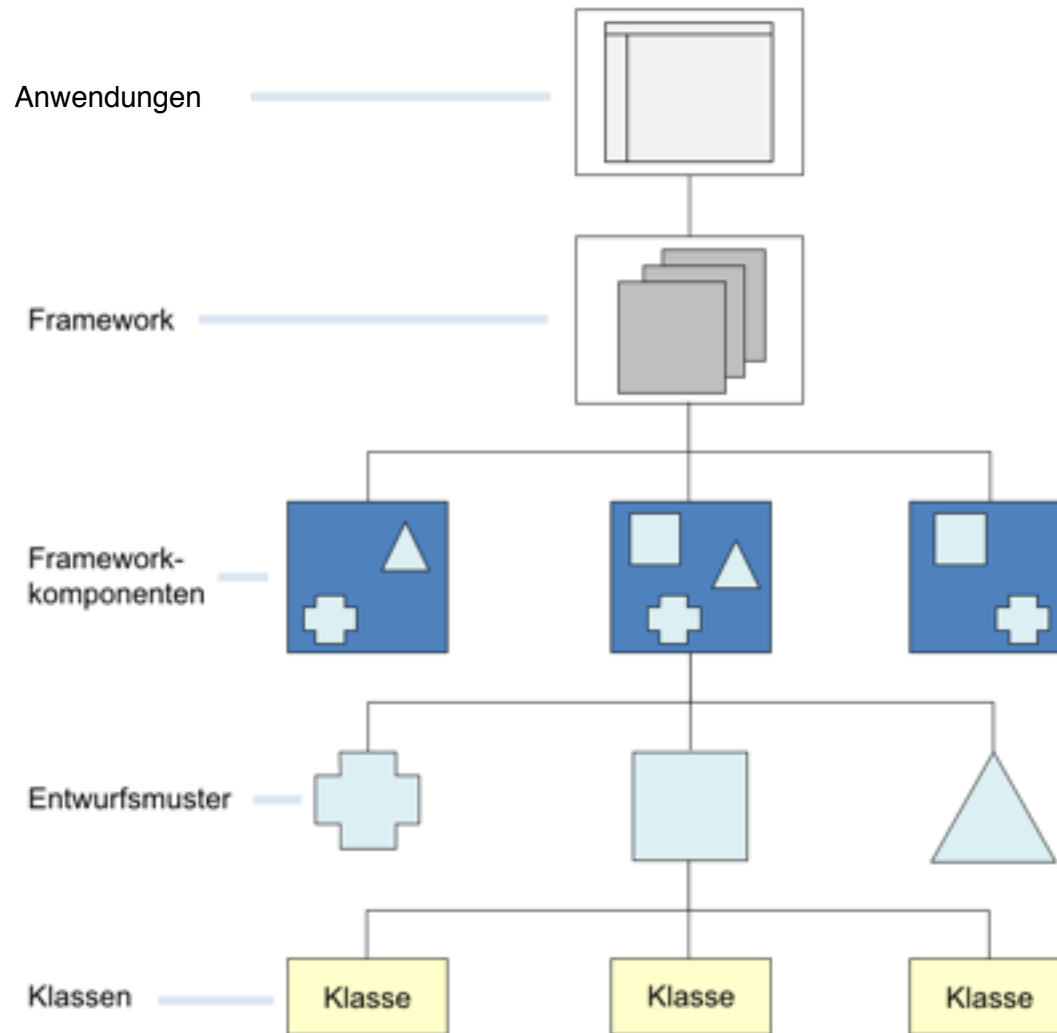
Durch ein objektorientiertes Framework wird eine Software-Architektur für eine Anwendung vorgegeben.

- ❑ Damit werden die Struktur wesentlicher Klassen und Objekte, sowie ein Modell Kontrollfluss, in der Anwendungsdomäne festgelegt (Hollywood-Prinzip)
- ❑ In diesem Sinne werden Frameworks mit dem Ziel einer Wiederverwendung von Architekturen entwickelt und genutzt



# Komponenten des Framework

## Struktureller Aufbau / Beziehungen



Bildquelle: John (1997)



## Frameworks sind in Software gegossene Konzepte und Lösungen

---

Diese Konzepte helfen beim Modellieren der Anwendungsdomäne, die zentralen Objekte Beziehungen und Schnittstellen sind im Framework bereits moduliert.

Für das Entwicklungsteam ergeben sich damit Vorteile.



## Vorteile für das Entwicklungsteam

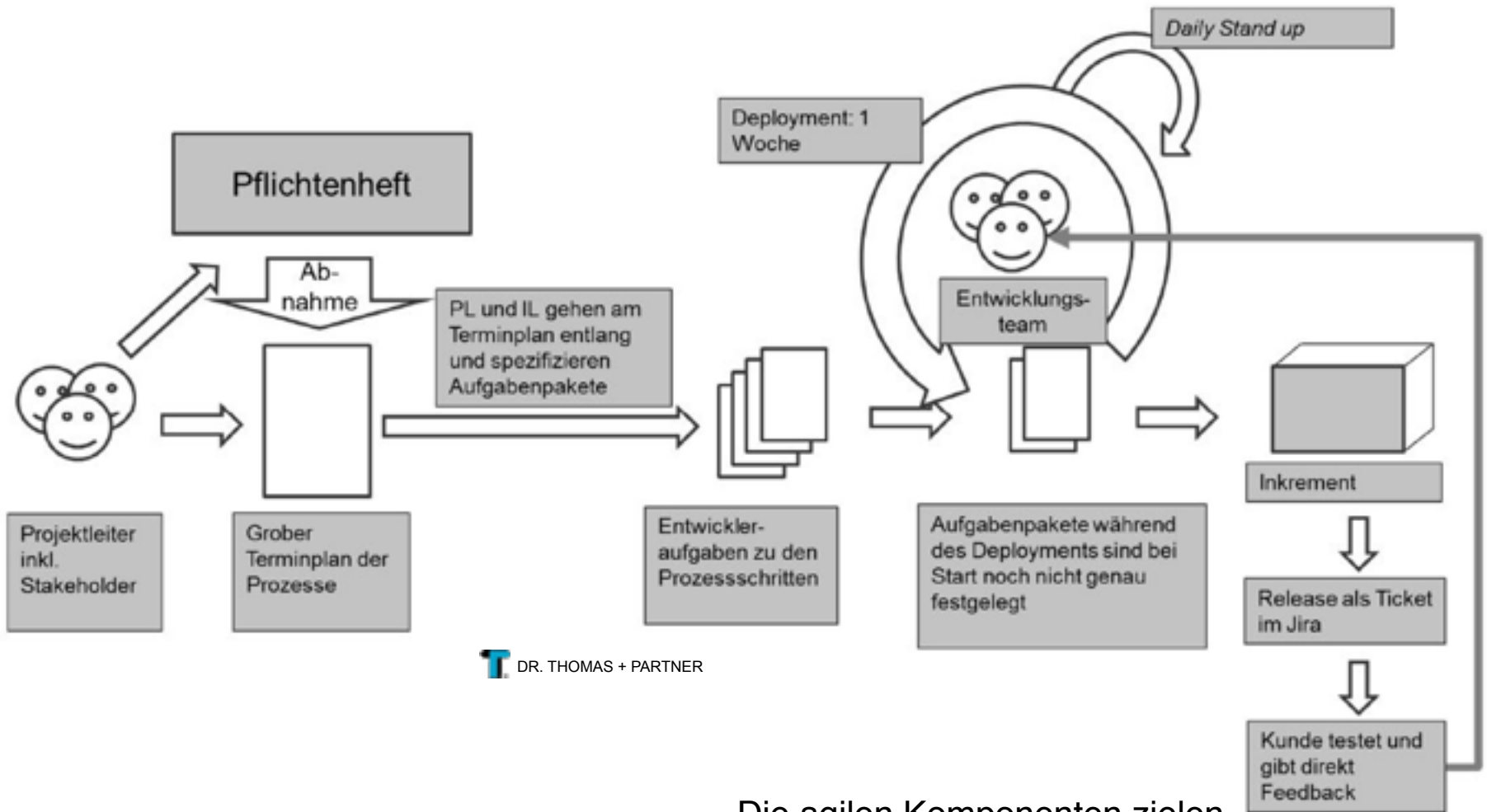
---

Nach dem Vorgehensmodell „Prinzip der agilen Methoden“ können die spezifischen Arbeitspakete entsprechend den Anforderungen an die Entwickler aufgeteilt werden:

- ❑ Für einen mit der Anwendungsdomäne vertrauten Entwickler, führt dies zu einem beschleunigten Projektfortschritt
- ❑ Ein Unerfahrener Entwickler ist in der Lage Anwendungen zu schreiben, ohne die Anwendungsdomäne vollständig verstanden zu haben. Das Grundwissen ist integraler Bestandteil des Framework



# Agile Vorgehensweise bei DR. THOMAS + PARTNER



 DR. THOMAS + PARTNER

Die agilen Komponenten zielen auf die Software-Entwicklung ab!



# Erfolg verleiht Flügel

- ▶ das Team macht etwas Spannendes
- ▶ etwas wofür Gehirnschmalz gebraucht wird
- ▶ wo Ideen zählen
- ▶ auf jeden kommt es persönlich an
- ▶ das Team verstärkt sich durch gute, hochmotivierte Mitarbeiter.  
"Wer will keinen Erfolg haben" ?





## Vorteile einer abstrahierten wiederverwendbaren Anwendungsdomäne

---

- ❑ Frameworks können als Absicherung von Investitionen verstanden werden
- ❑ Erarbeitetes Wissen über das Anwendungsgebiet (hier WMS/MFCS) bleibt in der Firma erhalten
- ❑ Ein gutes Framework ermöglicht Anwendungen zu entwickeln, ohne alle Details der dazu benötigten Ressourcen verstehen zu müssen.





# Frameworks = Components + Pattern

---

Die Struktur gestaltet sich in Form einer Beziehung der Anwendung als spezifischer Instanziierung des Frameworks (z.B. „best practice Komponente“ im WE)

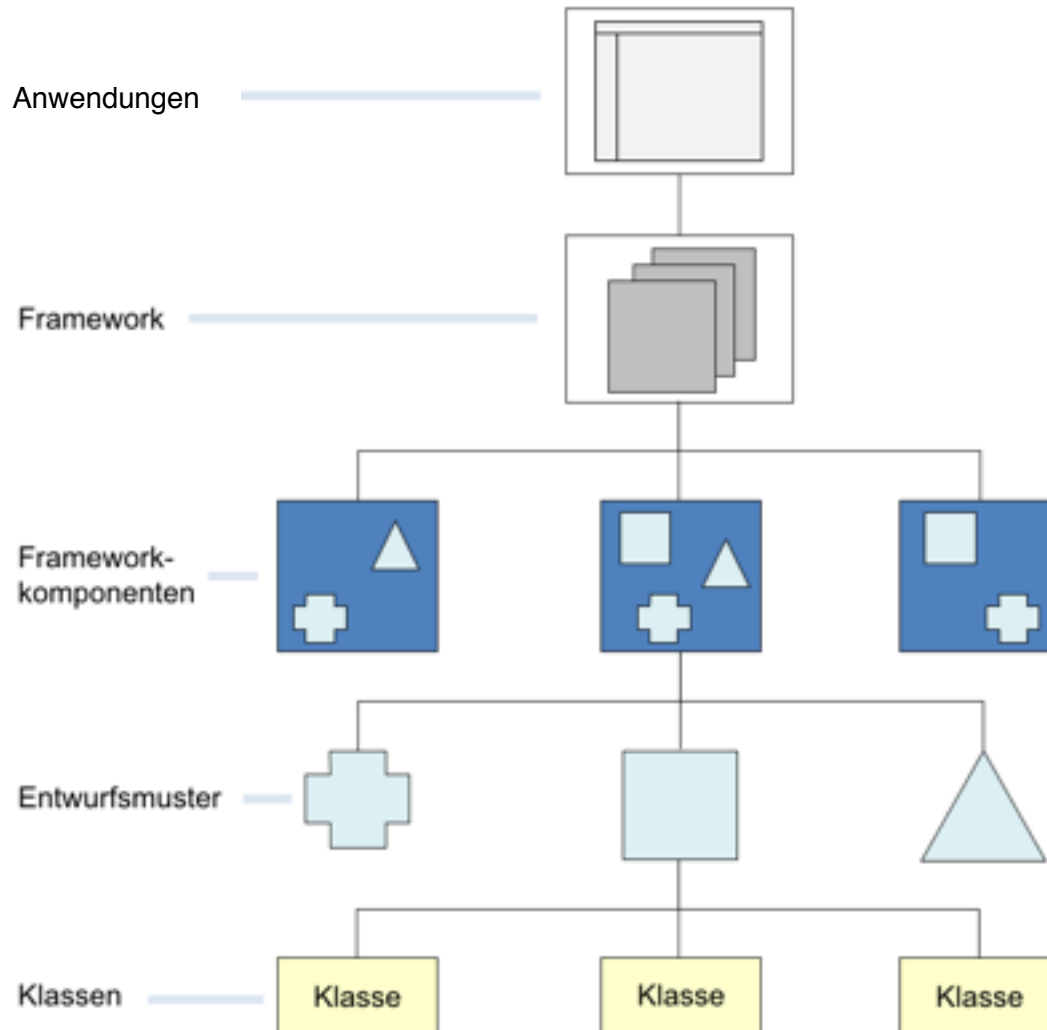
- dem Framework
- den Entwurfsmuster
- den Komponenten Klassen

Klassen und Entwurfsmuster vereinigen sich zu Komponenten. Zusammen bilden alle Komponenten das eigentliche Framework.



# Komponenten des Framework

## Struktureller Aufbau / Beziehungen



Bildquelle: John (1997)

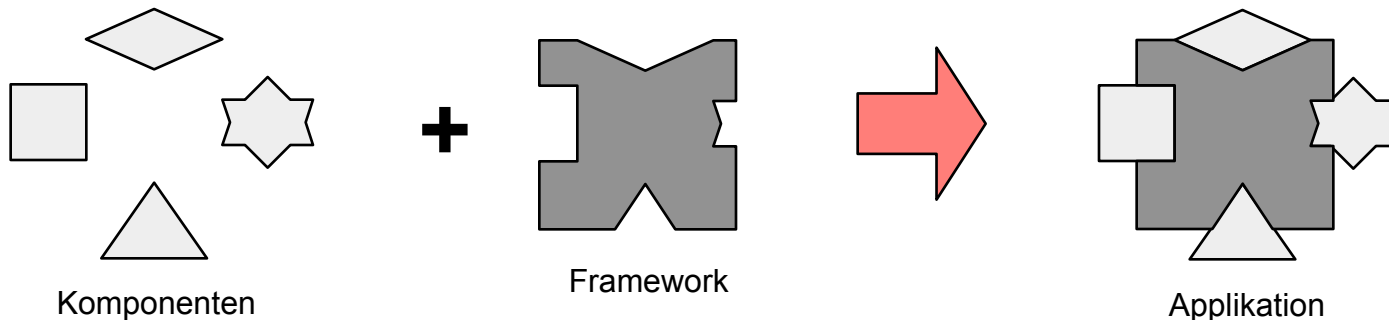


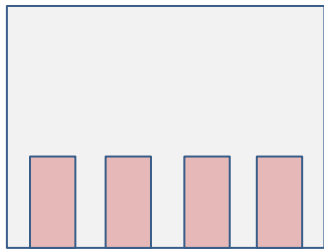
# Frameworks

Um **komplexe objektorientierte adaptive Architekturen** zu beherrschen, gibt es den zentralen Ansatz des **objektorientierten Frameworks** (Rahmenwerk).

Ein Framework im allgemeinen Sinne ist ein wiederverwendbares System, welches in fertige und halbfertige Subsysteme untergliedert ist. Es legt dabei die Struktur dieser Systeme und Subsysteme fest.

- Wiederverwendung von Design
- Zur Entwicklung von ähnlichen Anwendungen



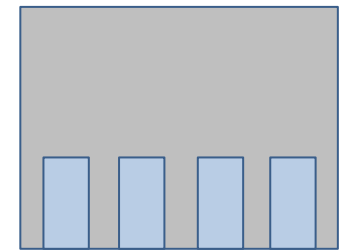


Vorgefertigte Teile

# Klassenbibliothek



# Frameworks



Anwendungsspezifische Teile

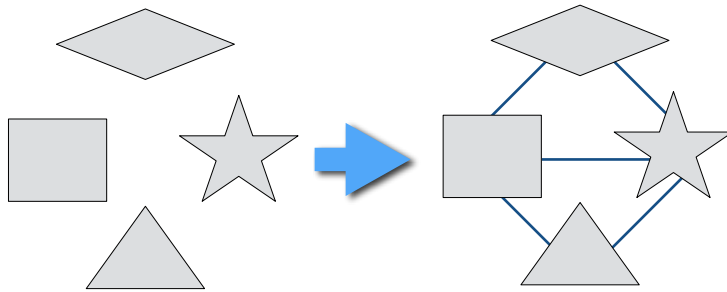
Ablaufsteuerung nicht vordefiniert

Kontrollfluss durch Anwendung

Unabhängige wiederverwendbare Module

Mehrfachverwendung von Funktionalität

Applikationsentwicklung:



Komponenten

Applikation

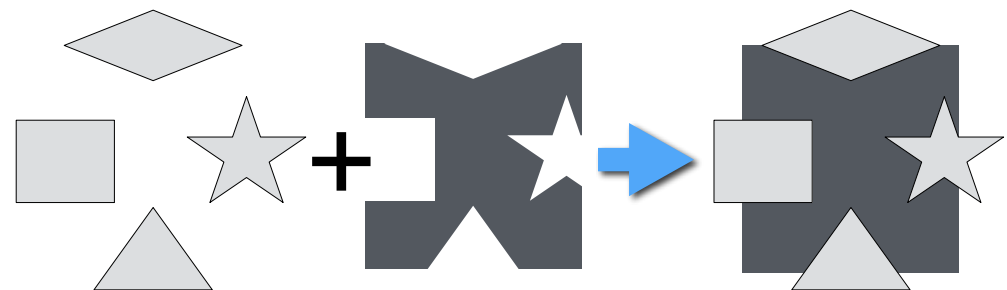
Ablaufsteuerung im wesentlichen vordefiniert

Kontrollfluss durch Framework („Hollywood Prinzip“)

Verbund zusammenhängender Klassen

Mehrfachverwendung von Struktur und Funktionalität

Applikationsentwicklung:



Komponenten

Framework

Applikation



# Softwarearchitektur

---

Eine adaptive IT erfordert eine modulare Softwarearchitektur, die in ihrer Eigenschaft

- skalierbar
- anpassbar
- erweiterbar
- und wiederverwendbar

sein muss.

Ziel ist eine nach den Architektur-Prinzipien ausgerichtete Systemarchitektur.



# Architektur Prinzipien

---

Die Bausteine (Geschäftsprozessmodule) sollten

- nach Aufgabenbereichen getrennt sein  
(Separation of Concerns)
- die Ausprägung von Black-Boxes haben  
(Information-Hiding)
- und einen hohen Abstraktionsgrad aufweisen  
(Abstraktionsprinzip)



# Separation-of-Concerns-Prinzip

---

Wichtigster Einsatz ist die Unterstützung der Modularisierung:

Dabei sind Teile einer anwendungsbestimmten Software -  
z.B.: Im Geschäftsprozessmodul “Zweistufige Kommissionierung”.  
Hier die Aufgabe Mensch-Maschine-Kommunikation während des  
Rundgangs zu identifizieren und als eine wiederverwendbare  
Komponente zu kapseln.

Ein komplexes Geschäftsprozessmodul wird auch aus der Sicht  
„des Prinzip der agilen Methoden“ in verständliche handhabbare  
Arbeitspakete (Komponenten) zerlegt.



## Trennung von anwendungsbestimmter und technikbestimmter Software

---

**A:** Anwendungsbestimmte Software kann immer dann wiederverwendet werden, wenn vorhandene Anwendungslogik ganz oder teilweise benötigt wird.

**T:** Technikbestimmte Software kann immer dann wiederverwendet werden, wenn ein neues - z.B. ein Fördersystem dieselben technischen Komponenten einsetzt.





# Information-Hiding-Prinzip

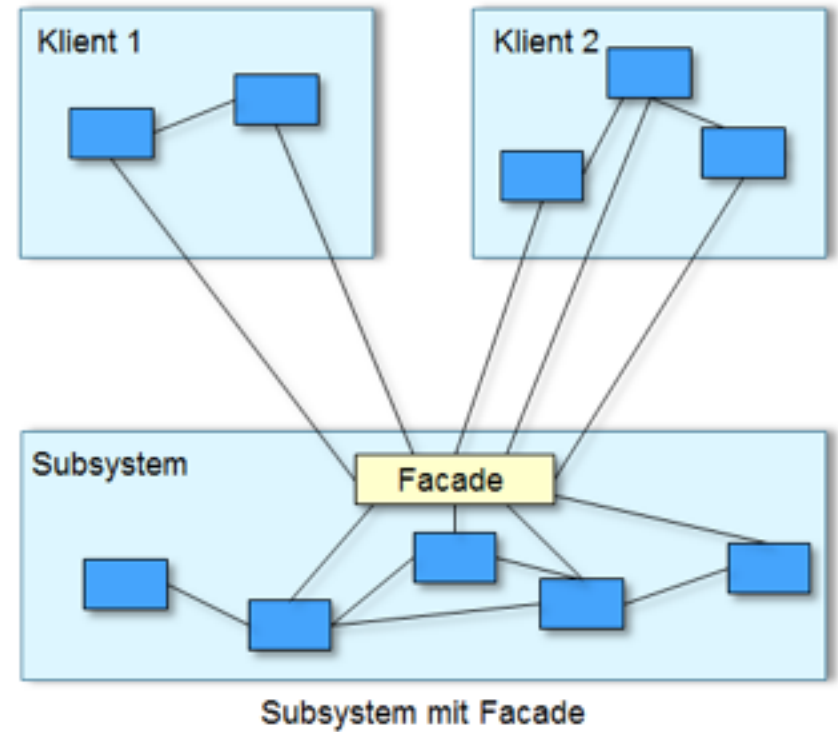
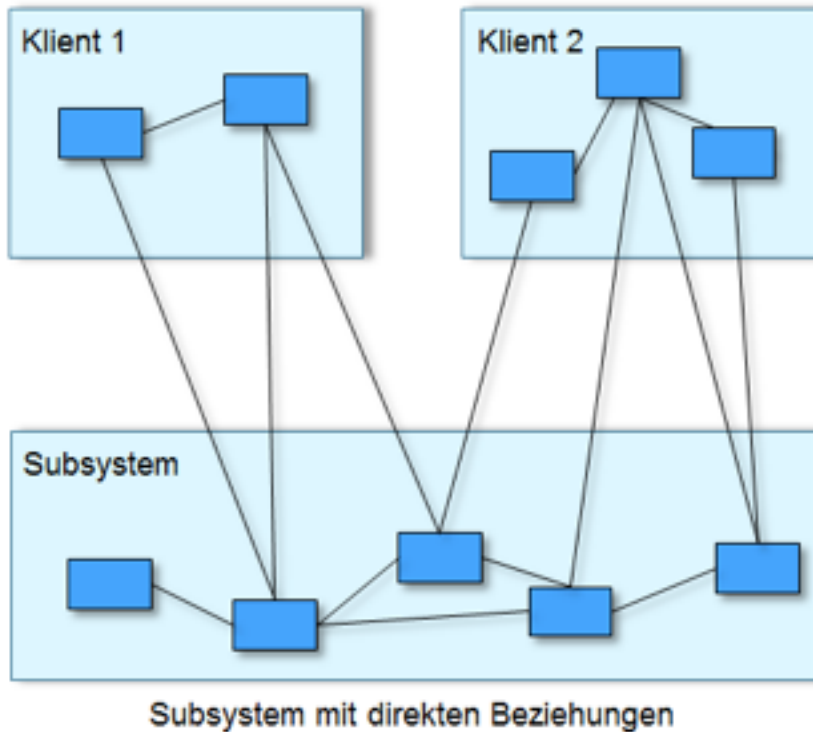
---

Ein Baustein wird gekapselt und ist nach außen durch wohl definierte Schnittstellen bekannt.

- Black-Box-Prinzip, die Interna sind nicht sichtbar, ausschließlich die Schnittstellen damit können Interna quasi beliebig oft geändert werden (MP-Rundgang, EP-Rundgang)
- Unterstützt auch größere Strukturen  
Beispiel: Facade-Entwurfsmuster schützt ein ganzes Subsystem



# Subsystem ohne und mit Facade





# Abstraktionsprinzip

---

Das Prinzip hierbei wichtige Aspekte zu identifizieren und unwichtige Details zu vernachlässigen (Schnittstellenabstraktion)

Trennung von Schnittstelle und Implementierung:

- Damit sich ein Klient auf die Schnittstelle verlassen kann, soll die Schnittstelle separat von der Implementierung betrieben werden
- Anwendung findet das, wenn die Schnittstelle standardisiert ist



# Modularitätsprinzip

---

Eine Softwarearchitektur sollte aus Bausteinen bestehen, die funktionale Verantwortung klar abgegrenzt sind, d.h. leicht austauschbar und in sich geschlossen.

Das Modularitätsprinzip dient der

- Änderbarkeit
- Erweiterbarkeit
- Wiederverwendbarkeit von Bausteinen einer Architektur



# Aufteilung in Anwendungsdomäne und Middleware

In einem ersten Schritt wird die Software getrennt nach dem Aspekt:

- Technikbestimmter Software und
- Anwendungsbestimmter Software

## Lauffähiges Kundensystem

Anwendungsdomäne Intralogistik

Middleware

Die Anwendungsdomäne bewältigt alle Problemfälle der Intralogistik.



# Middleware

---

- ❑ Anwendungsunabhängige Technologie
- ❑ Vermittelt zwischen fachlicher Anwendungssoftware und Betriebssystem / Hardware
- ❑ Hat den größten Wiederverwendungsgrad
- ❑ Die Datenbank ist über die Middleware verbunden.  
Das Modul Persistence hat die Aufgabe in weitgehender transparenter Weise die Verbindung zwischen Anwendung und Datenbank herzustellen



# Middleware

Die Middleware (Zwischenanwendung) bezeichnet eine anwendungs-unabhängige Technologie, die in fachlicher Anwendungs- software und Betriebssystem / Datenbank / Hardware vermittelt.

Sie bietet notwendige und hilfreiche Dienste für die Implementierung und Laufzeit von Anwendungssoftware.

Sie ist aber nicht dem Bereich der Anwendungssoftware zuzuordnen.

Das Modul Persistence hat die Aufgabe, die Verbindung zwischen Anwendung und relationaler Datenbank herzustellen.



Integriert  
Realisiert  
Geplant



# Anwendungsdomäne

---

Nach dem Gesichtspunkt der Wiederverwendung strukturiert sich die Anwendungsdomäne in

- ❑ **Komponenten** (z.B. Best-Practice-Komponente)
- ❑ **TWS-Plattform** ist das Bindeglied der abstrakten unabhängigen Komponenten und konkretisiert die Teilaspekte der Komponenten
- ❑ **Plugins** müssen nicht mehr den Ansprüchen an Flexibilität genügen

Die projektspezifische Software definiert zusammen mit den Komponenten der TWS-Plattform und die Plugins das lauffähige System





# TWS

Dr. Thomas + Partner Warehouse Solutions

## LAUFFÄHIGES KUNDENSYSYEM

### ANWENDUNGSDOMÄNE

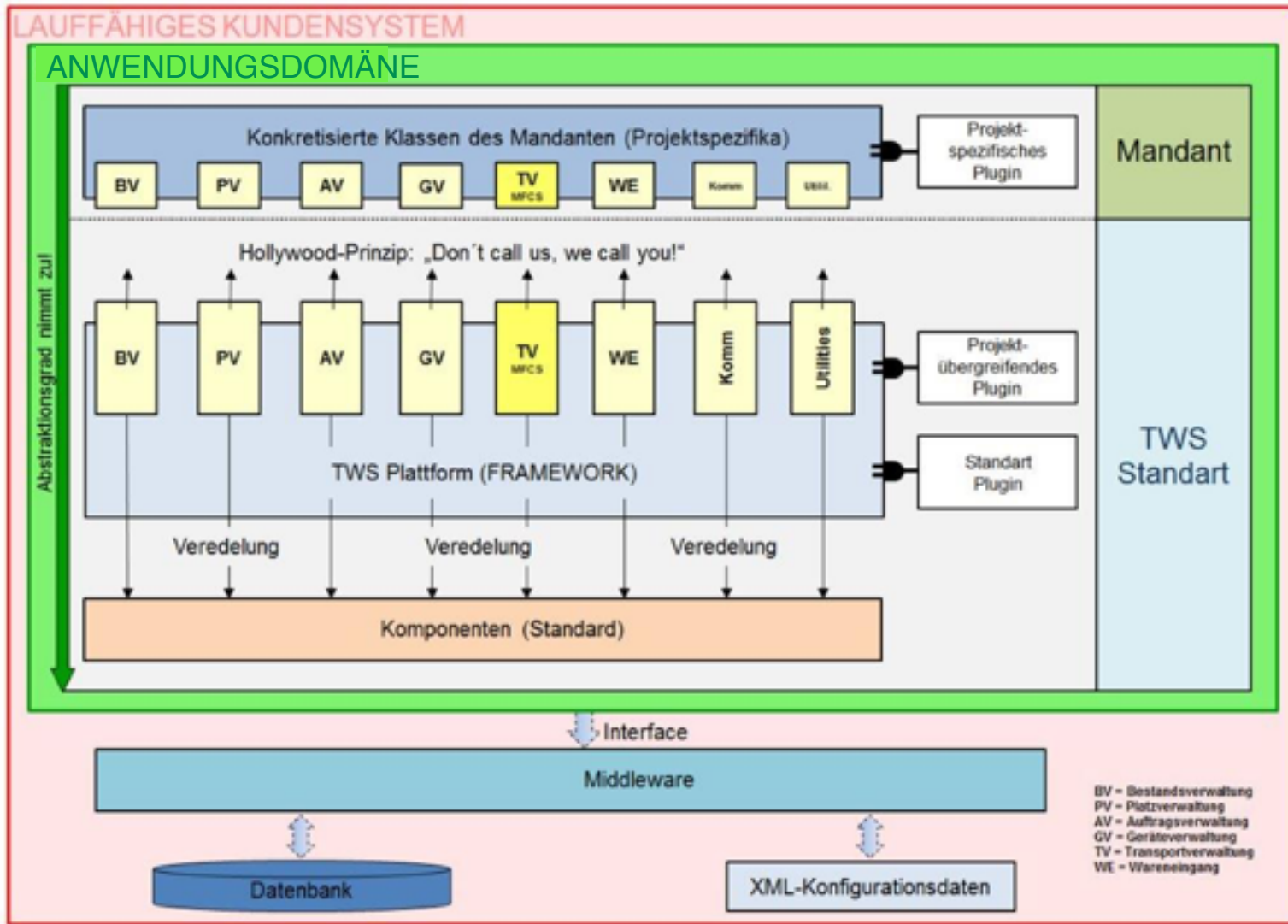
Die Anwendungsdomäne bewältigt alle Aufgaben der Problemfelder innerhalb der Intralogistik.  
Sie strukturiert sich nach dem Grad der Wiederverwendbarkeit, Komponenten, TWS Plattform, Plugins und Profektspezifika.





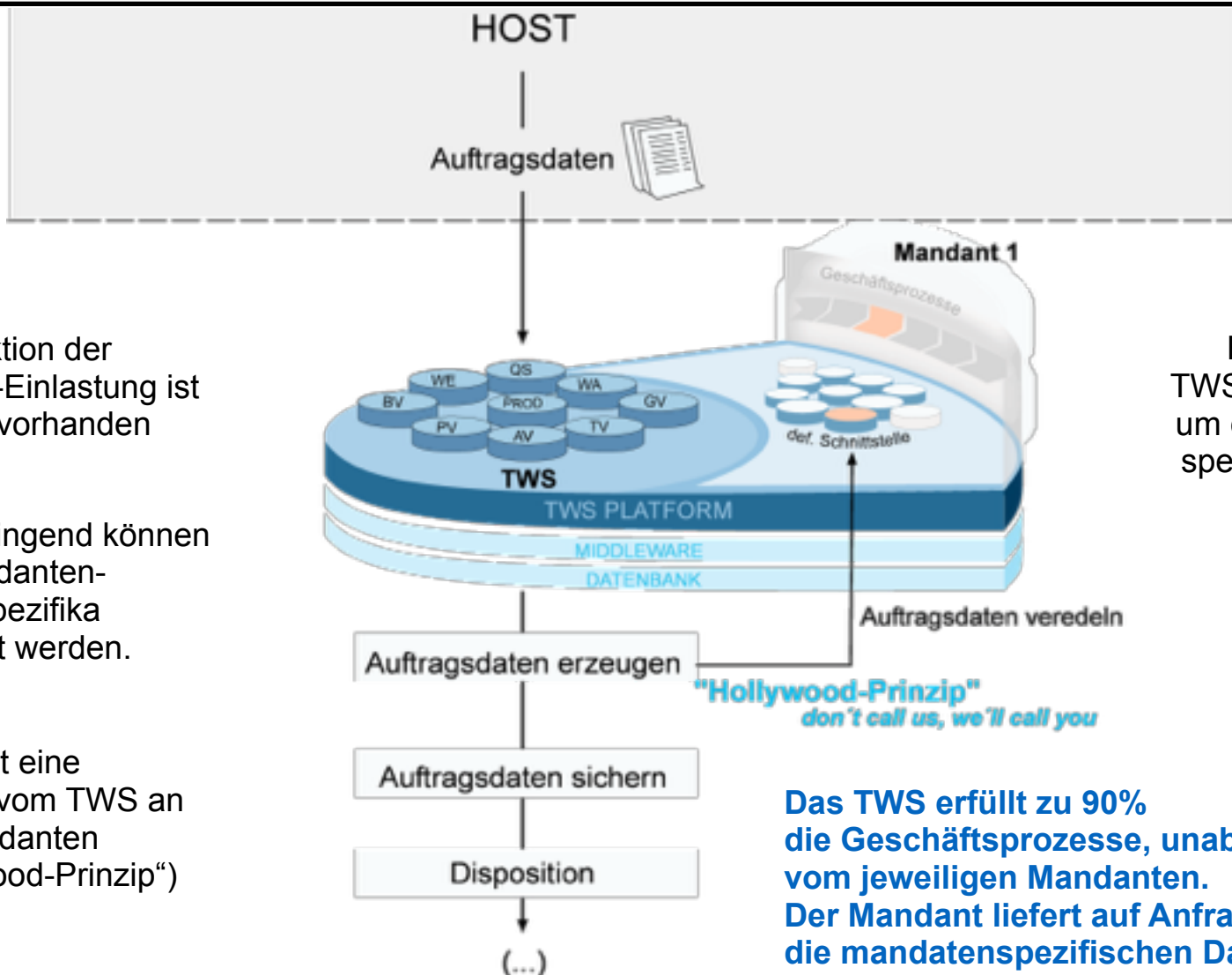
# TWS

## Lauffähiges Kundensystem





# Das TWS und das Hollywood-Prinzip



Die Funktion der Auftrags-Einlastung ist im TWS vorhanden

Nicht zwingend können alle Mandanten-Projektspezifika befriedigt werden.

Es erfolgt eine Anfrage vom TWS an den Mandanten („Hollywood-Prinzip“)

Erweiterung der TWS-Auftragsdaten um die mandanten-spezifischen Daten

Der Mandant als „externer Dienstleister“

**Das TWS erfüllt zu 90% die Geschäftsprozesse, unabhängig vom jeweiligen Mandanten. Der Mandant liefert auf Anfrage die mandatspezifischen Daten.**