

“Domain Driven Social Cloud Business Technology Provisioning Service”

Individuelle Geschäfts-Infrastruktur
(Software, Prozesse, Projekte)
aus prägnanten Spezifikationen
automatisch bereitstellen

Andreas Leue
Sphenon GmbH, Hamburg



Inhalt

Einführung

Motivation und Ansatz

Software Produktion: Sprachen, Abstraktion, Modelle
EM/OS - Übersicht

Business Design

Service & Brand: fairyloom & businessboard
Business Model: OSS, Tool & Service

Technologie

EM/OS – Architektur & Innovationen



Sphenon: Geschichte

- ★ 2016 fairyloom 1.0, businessboard 2.0, EM/OS 5.0
- 2014-15 Marken & Shopentwicklung
- 2013 EM/OS 4.0, businessboard 1.0 (Full Cycle: 1-3 min.)
- 2012-13 Service Redesign
 - (cloud & browser based, business customers)
- 2011 EM/OS 3.6 (UBPML Support)
- ★ 2011 WWS/Lager SCT Media Solutions
- ★ 2010 EM/OS 3.0 (Complete Executable Models)
- 2009 EM/OS 2.0
- 2008 oomodels.org (100% Java, EM/OS OSS)
- 2004/5 EM/OS 1.0 (SirFace)
- 2002 - Hypovereinsbank (Generator), HLG (VUI, Generator),
- 2008 Rekord AG (VUI), Skymaster (EM/OS 1.0),
- Bosch Versicherungsportal (EM/OS 2.0)
- ★ 2000 Onsecure Versicherungsportal (vollautomatische Produktion)
- 1999 Gründung Sphenon GmbH
- 1998 Generator in perl, XML, mächtige Templatesprache
- 1997 POET WebFactory "Database Publishing" & CMS
- 1995 Prototyp "Medienneutrale Oberfläche"
- 1994 Architektur-Studie "Cyberspace Architecture Project"
- 1989-93 Template-basierter Generator in C, Oberfläche für mehrere Medien

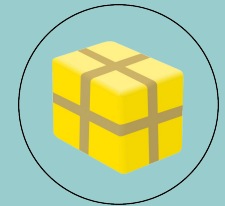
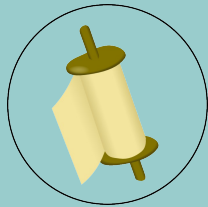
Sphenon: Geschäftsmechanik 2000-2017



Unsere Mission: herausragende Software Lösungen liefern, so schnell, bequem, wirtschaftlich und nachhaltig wie möglich.

Kundenprojekte

Maßgeschneiderte individuelle Applikationen auf Basis von Modellen



EM OS

Enterprise Model Operation Services

Re-Invest

Open Source



Generator



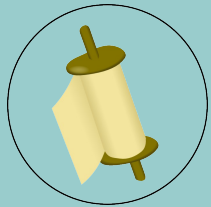
Sphenon: Geschäftsmechanik 2000-2017



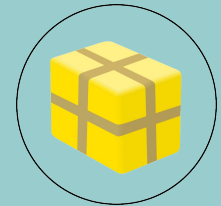
Unsere Mission: herausragende Software Lösungen liefern, so schnell, bequem, wirtschaftlich und nachhaltig wie möglich.

Kundenprojekte

Maßgeschneiderte individuelle Applikationen auf Basis von Modellen



EM OS
Enterprise Model Operation Services



Re-Invest

25 Jahre Reifung durch
Ko-Evolution in Projekten

Open Source



Inhalt

Einführung

Motivation und Ansatz

Software Produktion: Sprachen, Abstraktion, Modelle
EM/OS - Übersicht

Business Design

Service & Brand: fairyloom & businessboard
Business Model: OSS, Tool & Service

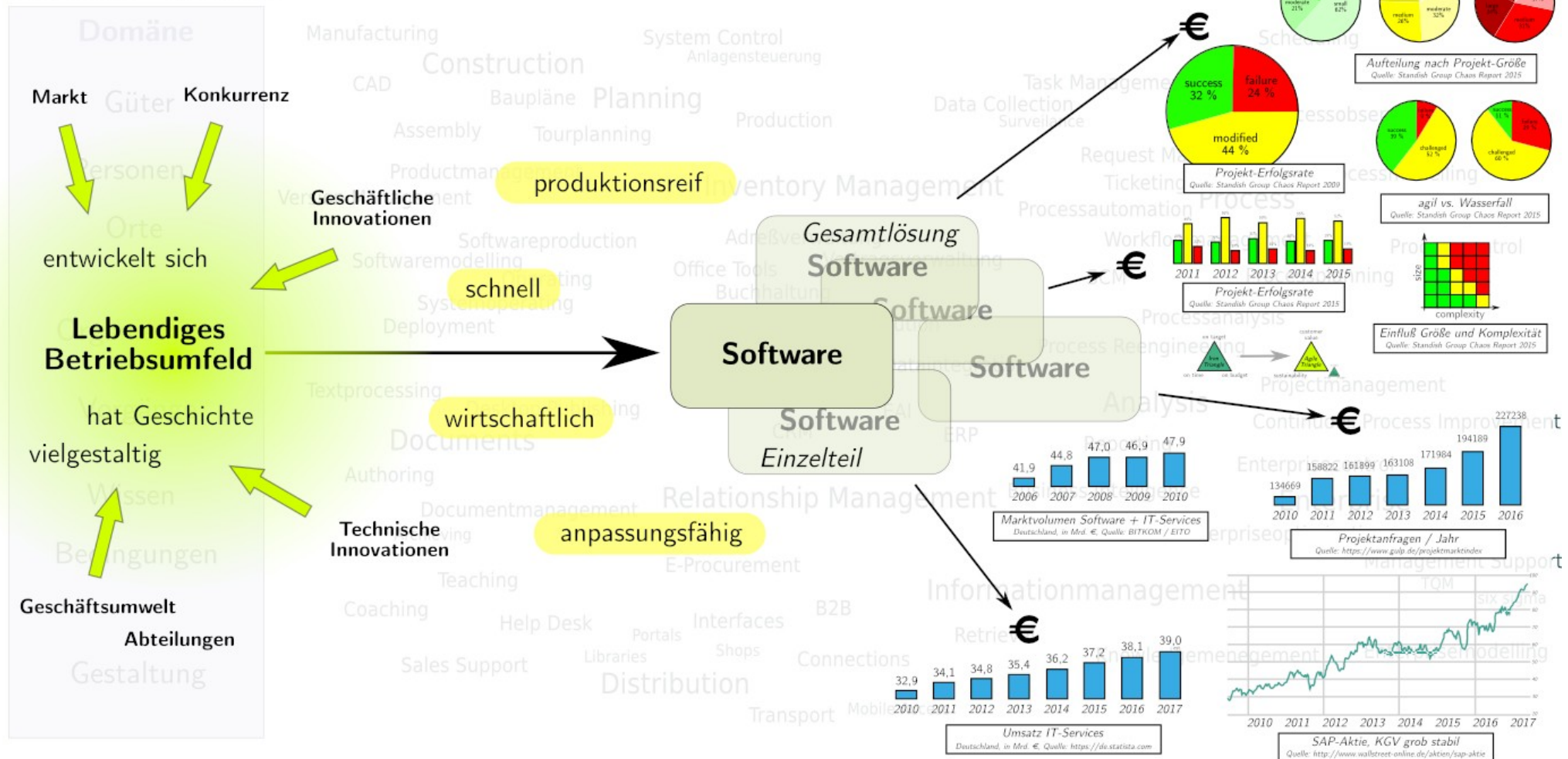
Technologie

EM/OS – Architektur & Innovationen



Geeignete Software für lebendige Unternehmen

Geschäfts-Systeme

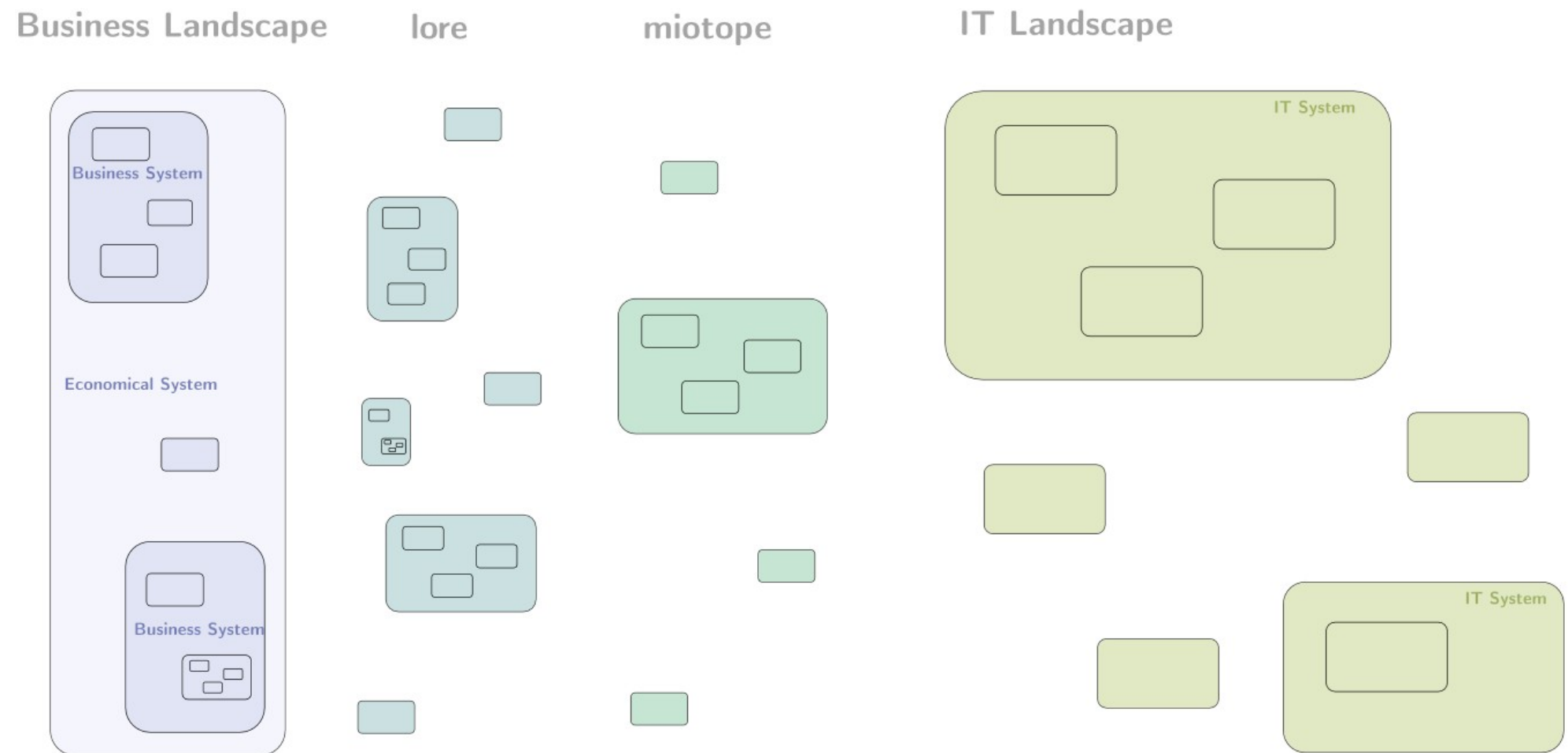


Artefakte

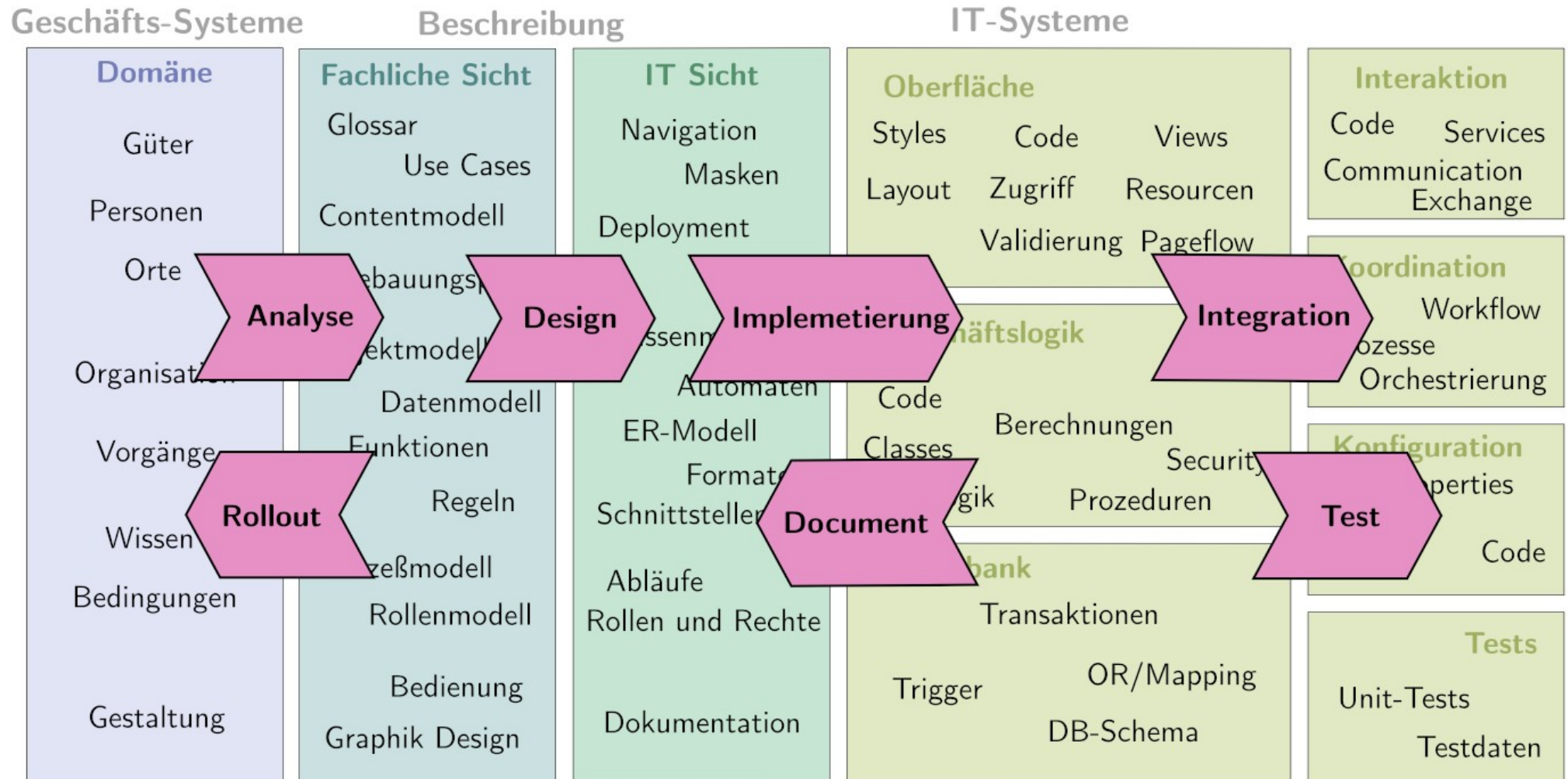
Geschäfts-Systeme	Beschreibung		IT-Systeme		
Domäne	Fachliche Sicht	IT Sicht	Oberfläche		
Güter	Glossar	Navigation	Styles	Code	Views
Personen	Use Cases	Masken	Layout	Zugriff	Ressourcen
Orte	Contentmodell	Deployment	Validierung Pageflow		
Organisation	Bebauungsplan	Klassenmodell	Geschäftslogik		
Vorgänge	Objektmodell	Automaten	Code	Berechnungen	Security
Wissen	Datenmodell	ER-Modell	Classes	Prozeduren	
Bedingungen	Funktionen	Formate	Logik		
Gestaltung	Regeln	Schnittstellen	Datenbank		
	Prozeßmodell	Abläufe	Transaktionen		
	Rollenmodell	Rollen und Rechte	Trigger	OR/Mapping	
	Bedienung	Dokumentation	DB-Schema		
	Graphik Design				
			Interaktion		
			Code Services		
			Communication Exchange		
			Koordination		
			Workflow		
			Prozesse		
			Orchestrierung		
			Konfiguration		
			Properties		
			Code		
			Tests		
			Unit-Tests		
			Testdaten		



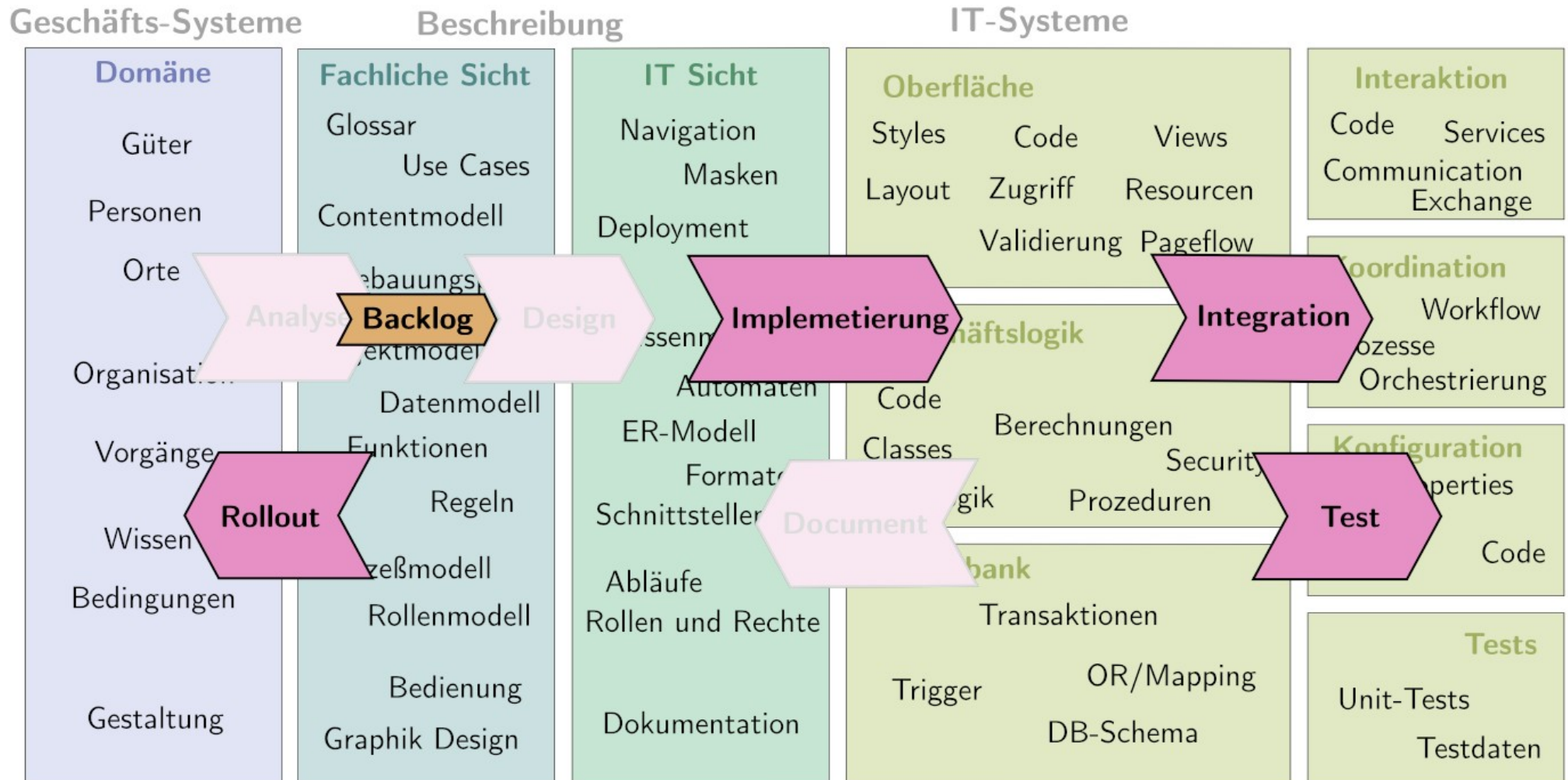
Immer weiter gesteckte Artefakt-Landschaft



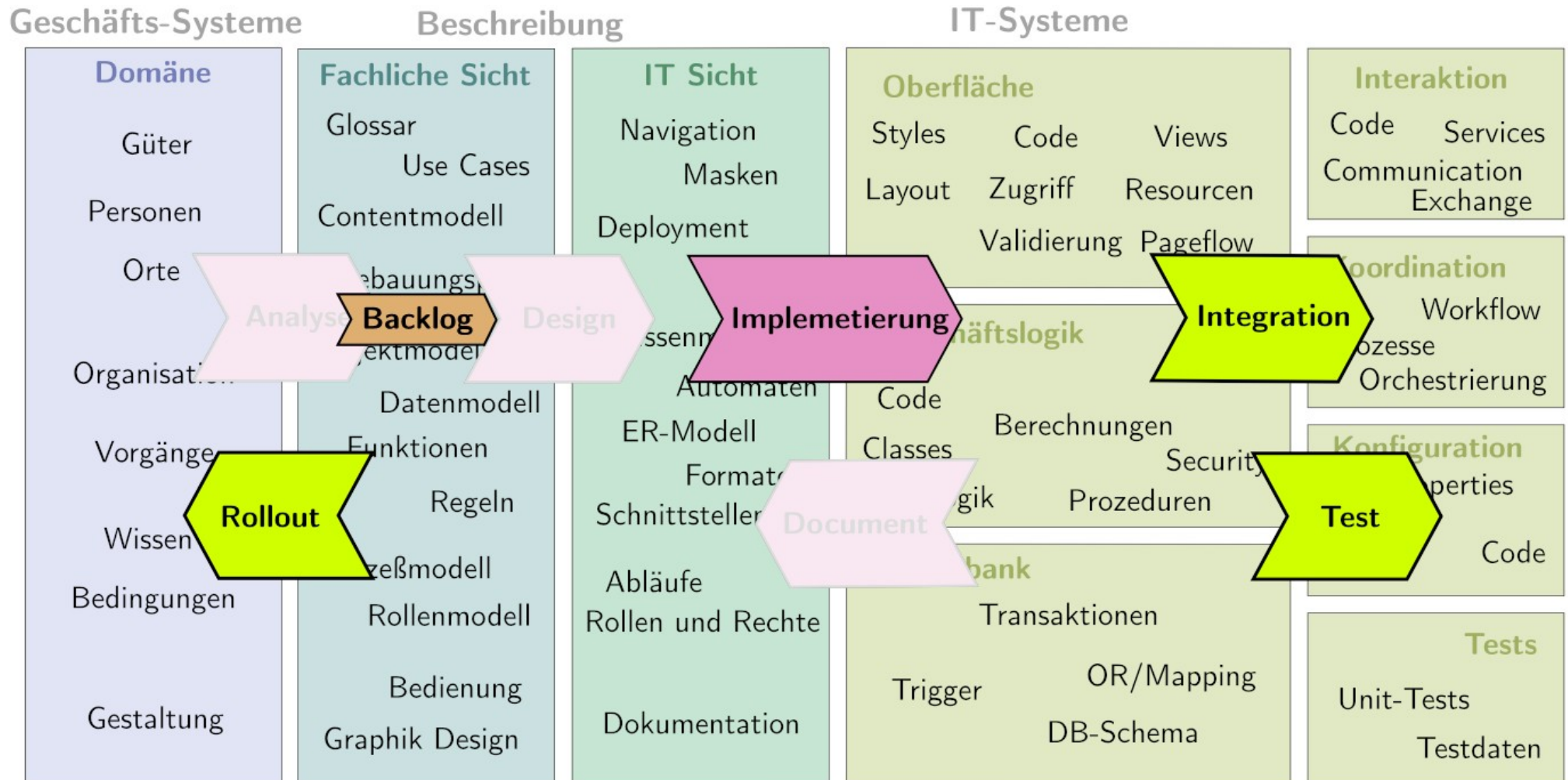
Artefakte und Entwicklung



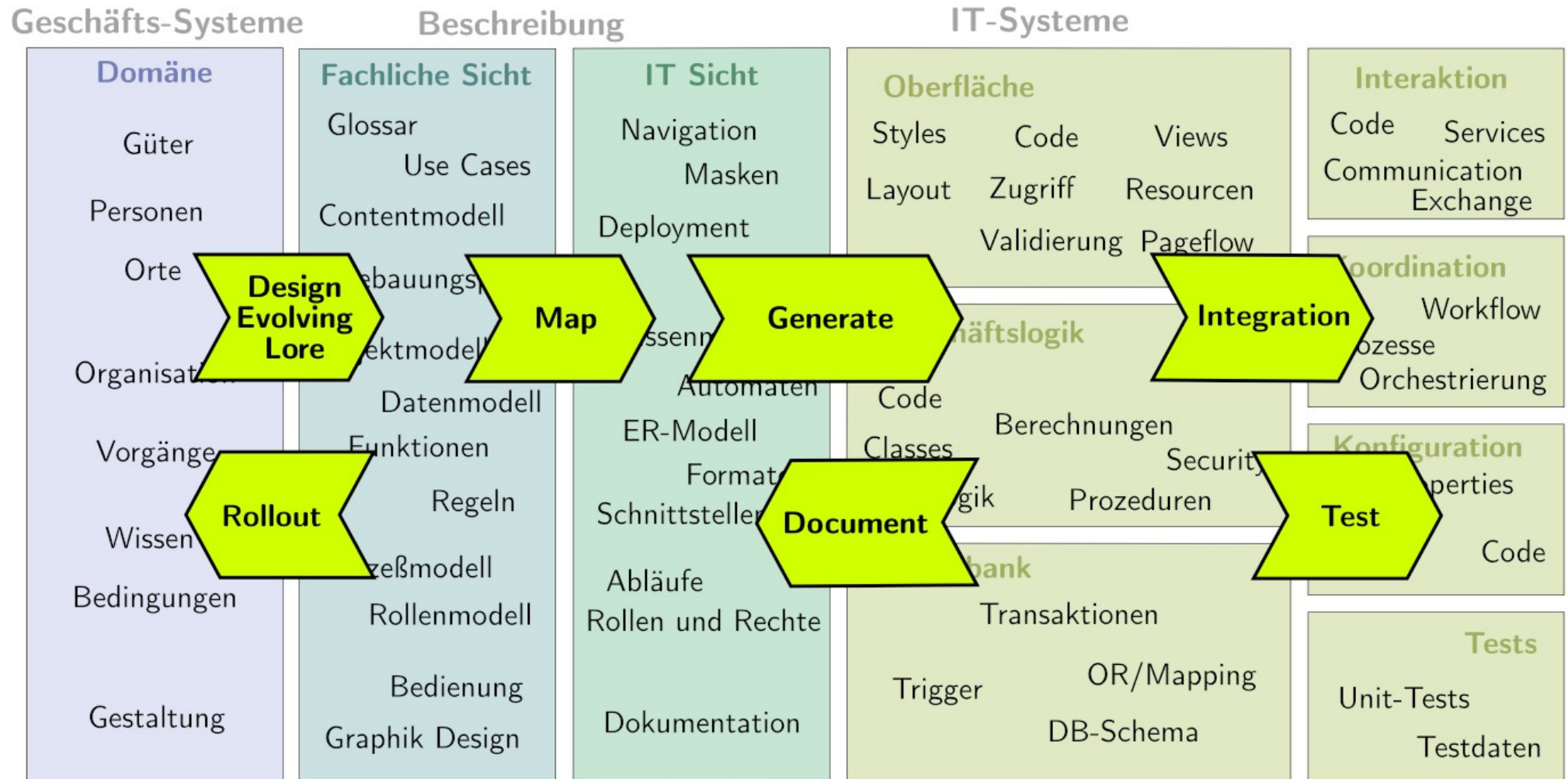
Agile Entwicklung



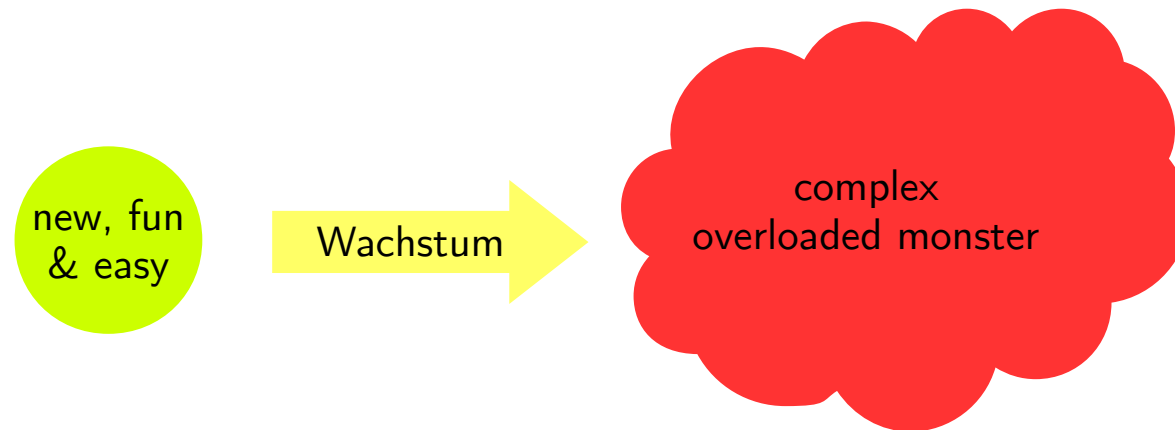
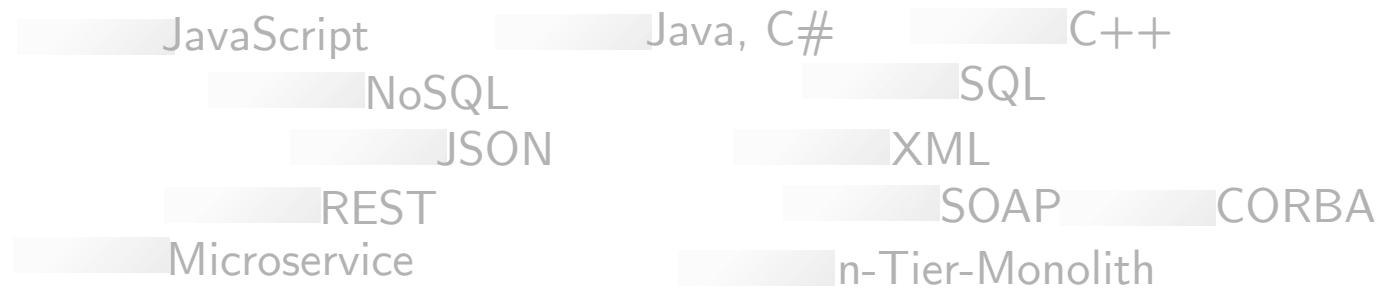
Agile, automatisierte Entwicklung



Agile, voll-automatisierte Entwicklung

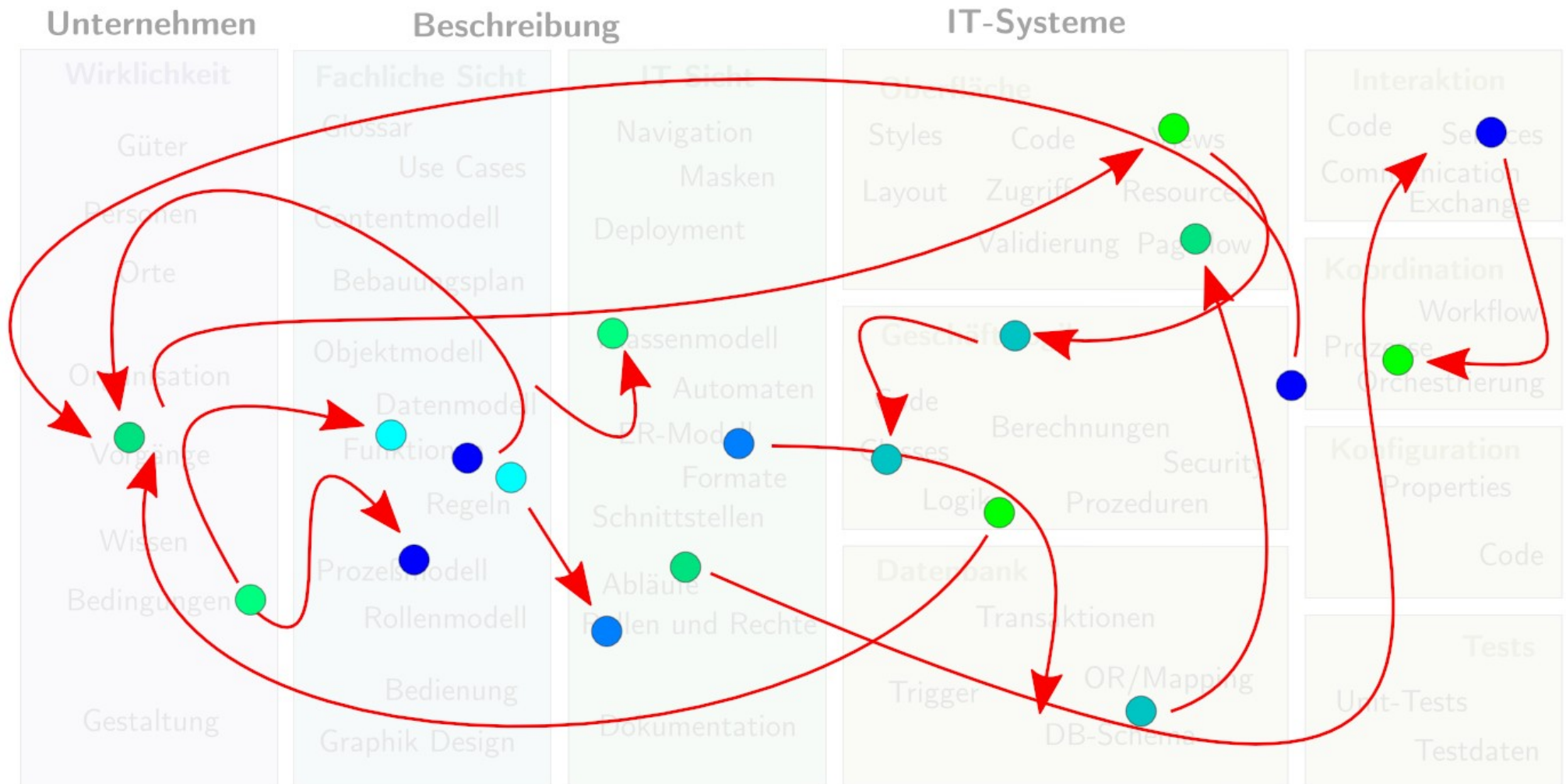


Technology Lifecycle

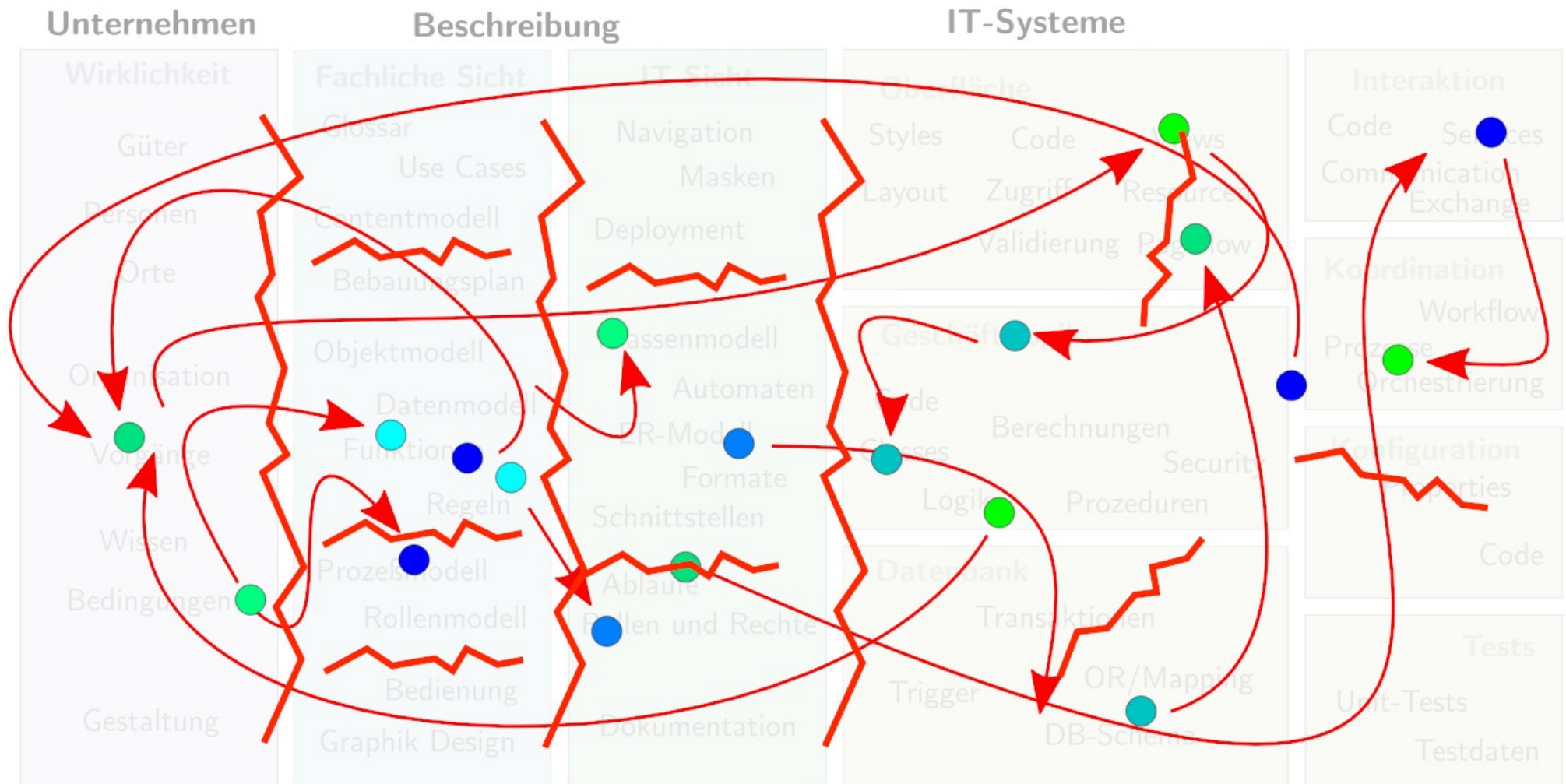


➔ Systeme veralten - Anpassungs-Krisen sind die Regel

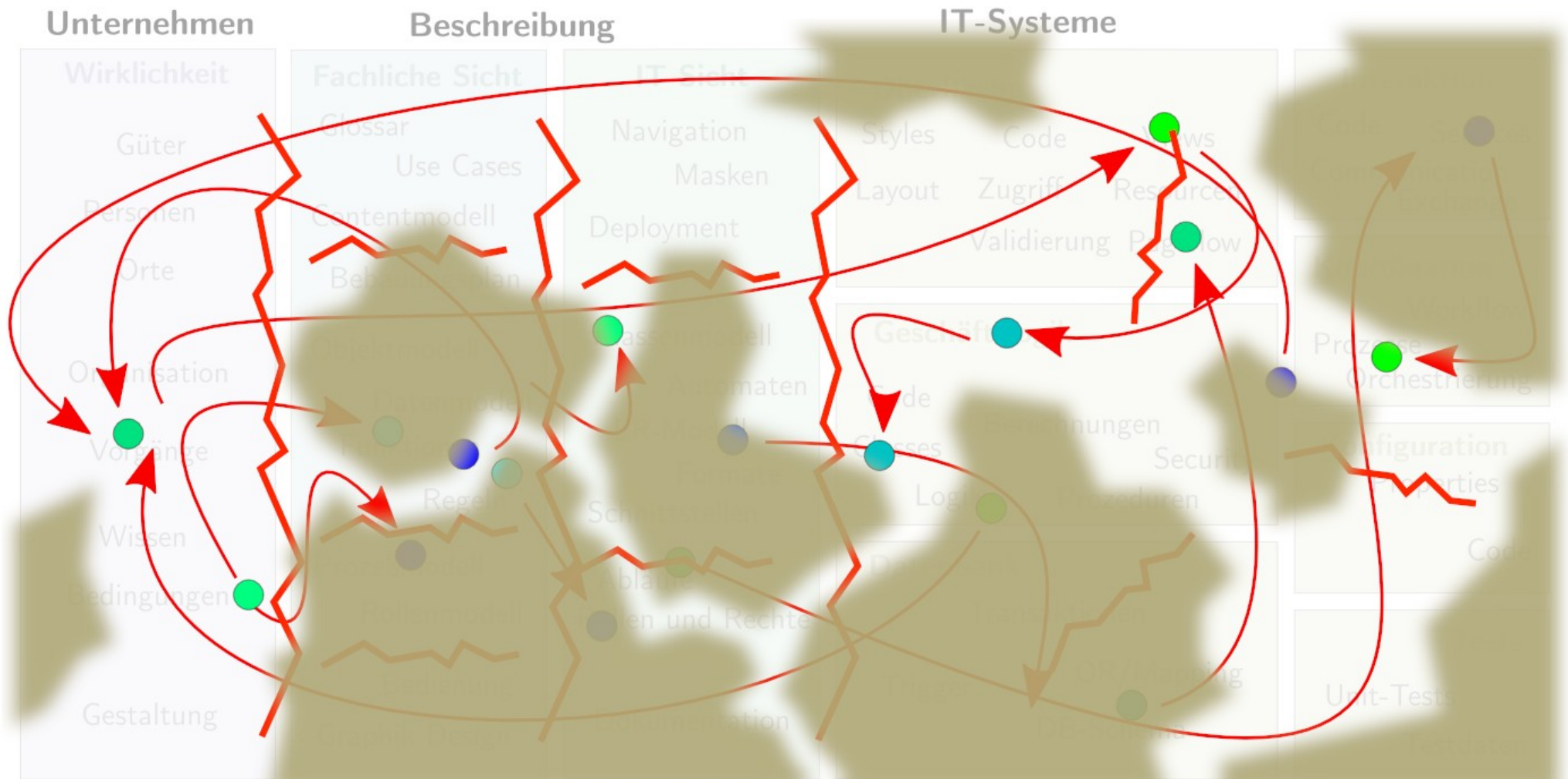
System Lifecycle - Spannungen



System Lifecycle - Spannungen, Brüche

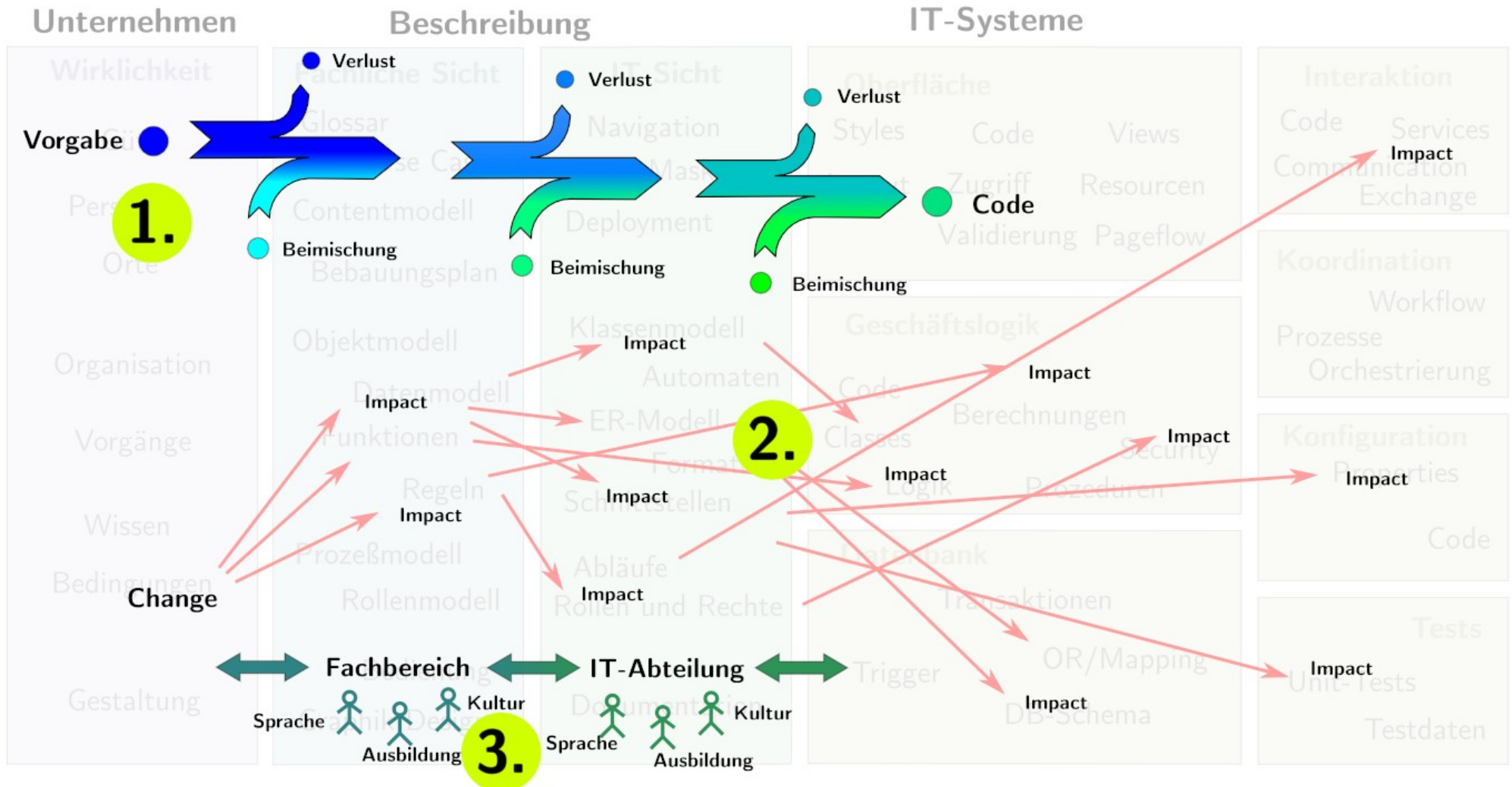


System Lifecycle - Spannungen, Brüche, Erosion



➔ Systeme veralten - Anpassungs-Krisen sind die Regel

Software-Manufaktur

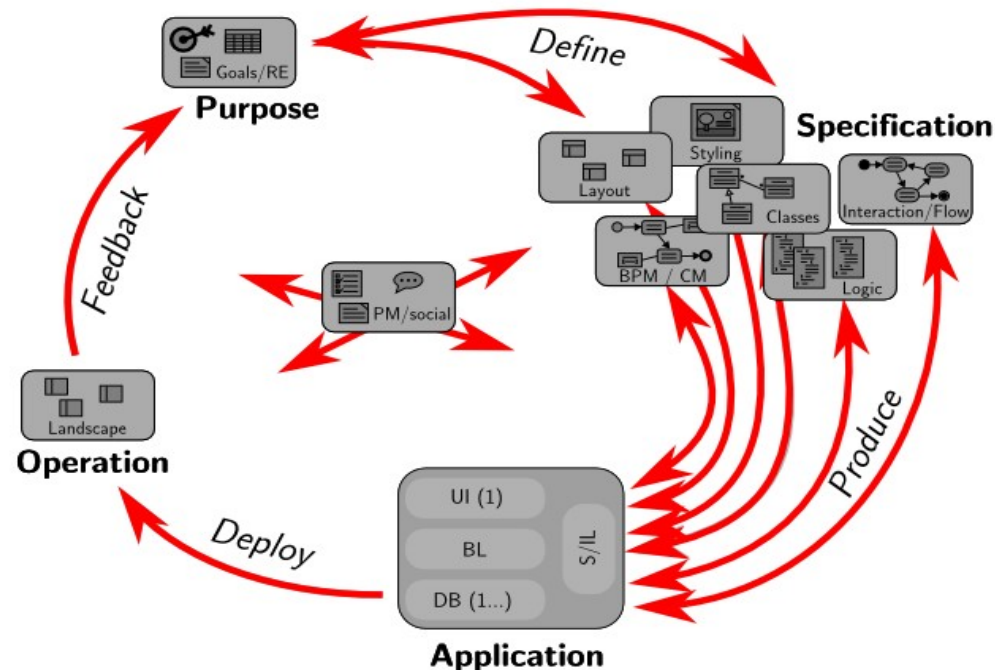


Manuelle Spezifikation und Produktion

sind arbeitsintensiv und fehleranfällig

Lock-in-Effekte

binden an Hersteller, Wissensträger und Technologien

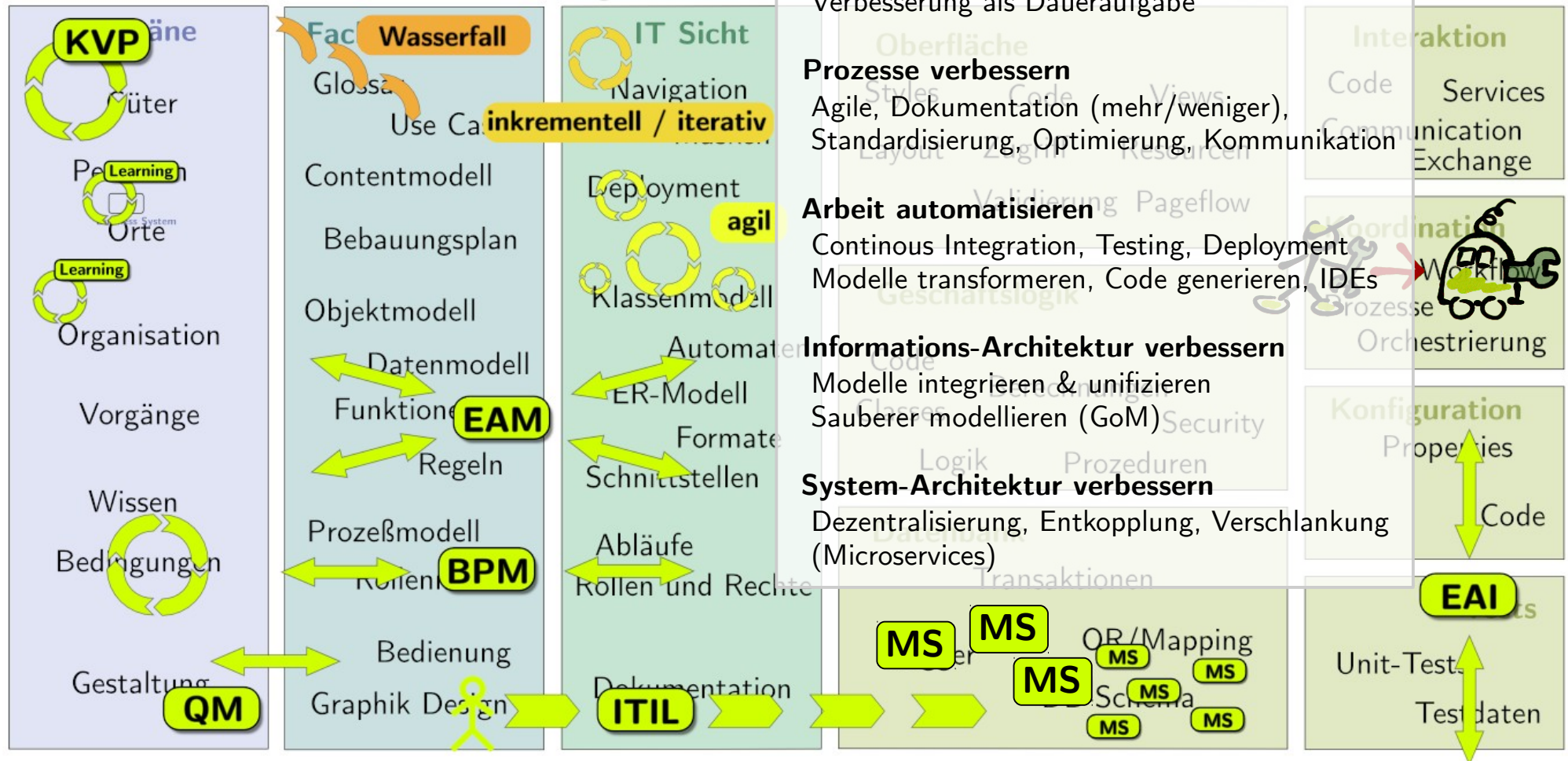


- ➔ Software-Produktion noch immer **riskant** und **teuer**, obwohl
- ➔ Flexibilität und Schnelligkeit **immer wichtiger** wurden und werden

Ansätze zur Verbesserung

Geschäfts-Systeme

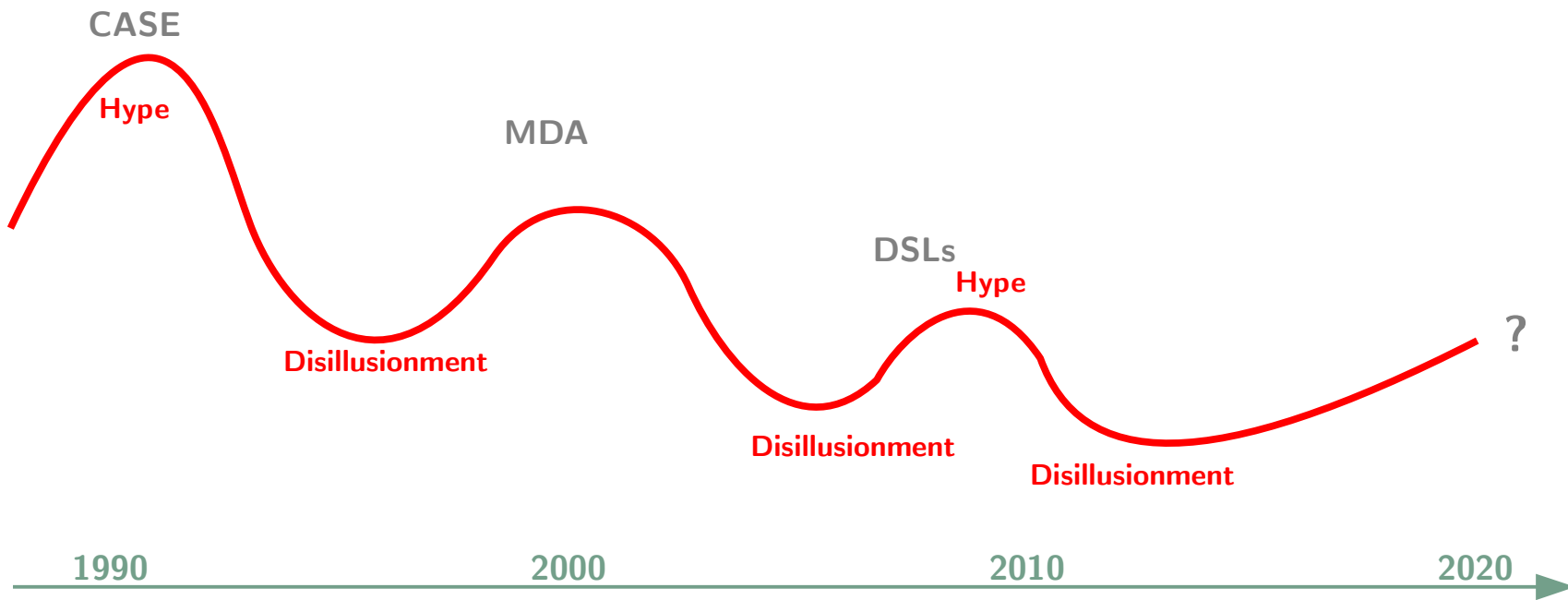
Beschreibung



Mehr als 25 Jahre “Model Driven”

MD Technology

Gartner Hype Cycle

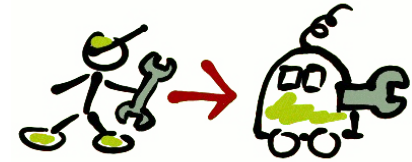


Enlightenment?



1. Generation MDx - CASE...

naiv



Unternehmen

Wirklichkeit

CASE
1990

Personen

Orte

Organisation

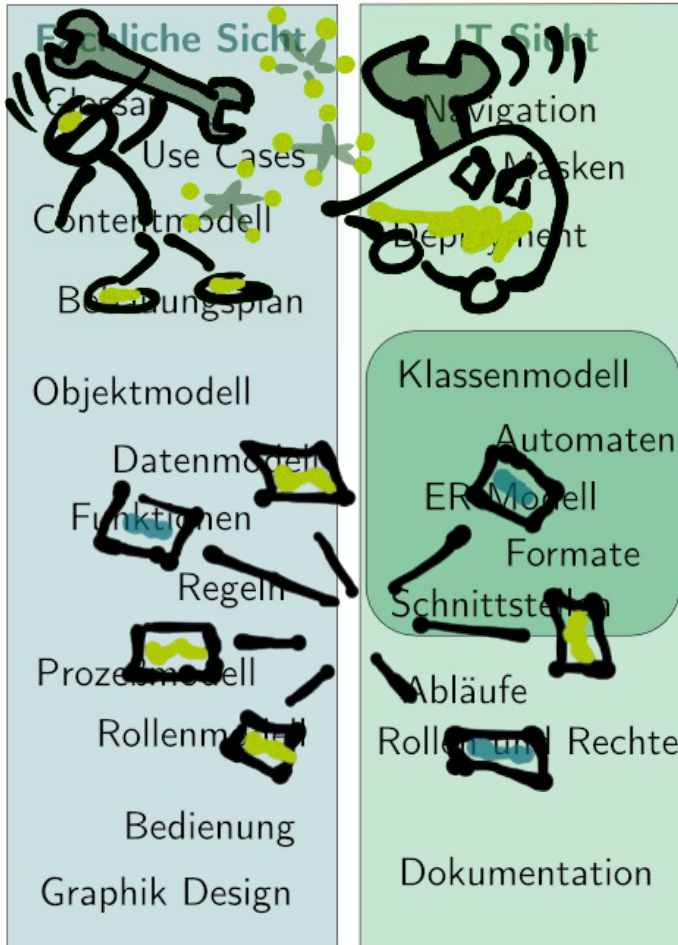
Vorgänge

Wissen

Bedingungen

Gestaltung

Beschreibung



IT-Systeme

Oberfläche

Styles Code Views
Layout Zugriff Ressourcen
Validierung Pageflow

Geschäftslogik

Code
Classes
Logik
Berechnungen
Security
Prozeduren

Datenbank

Transaktionen
Trigger OR/Mapping
DB-Schema

Interaktion

Code Services
Communication
Exchange

Koordination

Workflow
Prozesse
Orchestrierung

Konfiguration

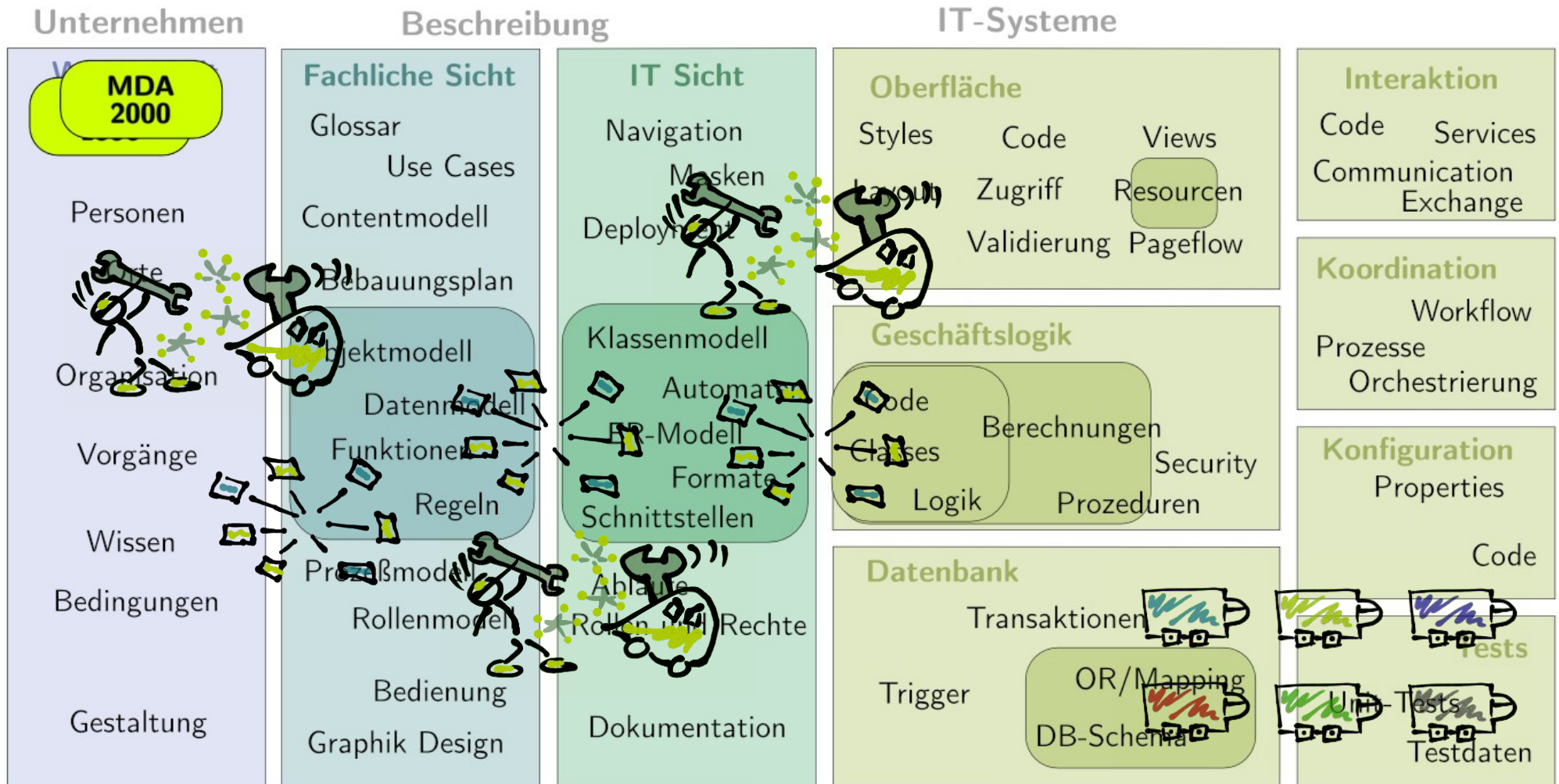
Properties
Code

Tests

Unit-Tests
Testdaten

2. Generation MDx - ...MDA...

Theorie



3. Generation MDx - ...DSLs...

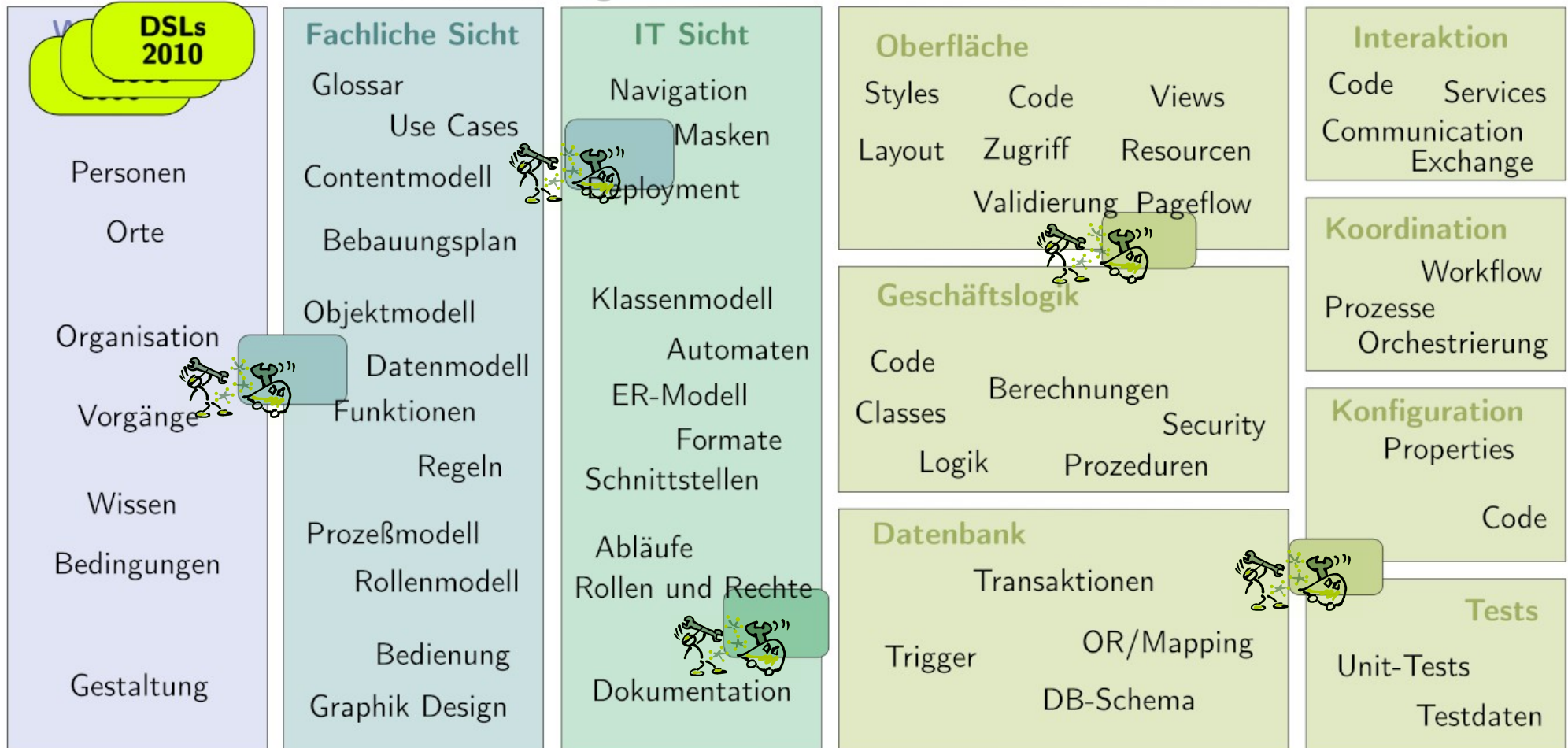
zweischneidige Teillösung



Unternehmen

Beschreibung

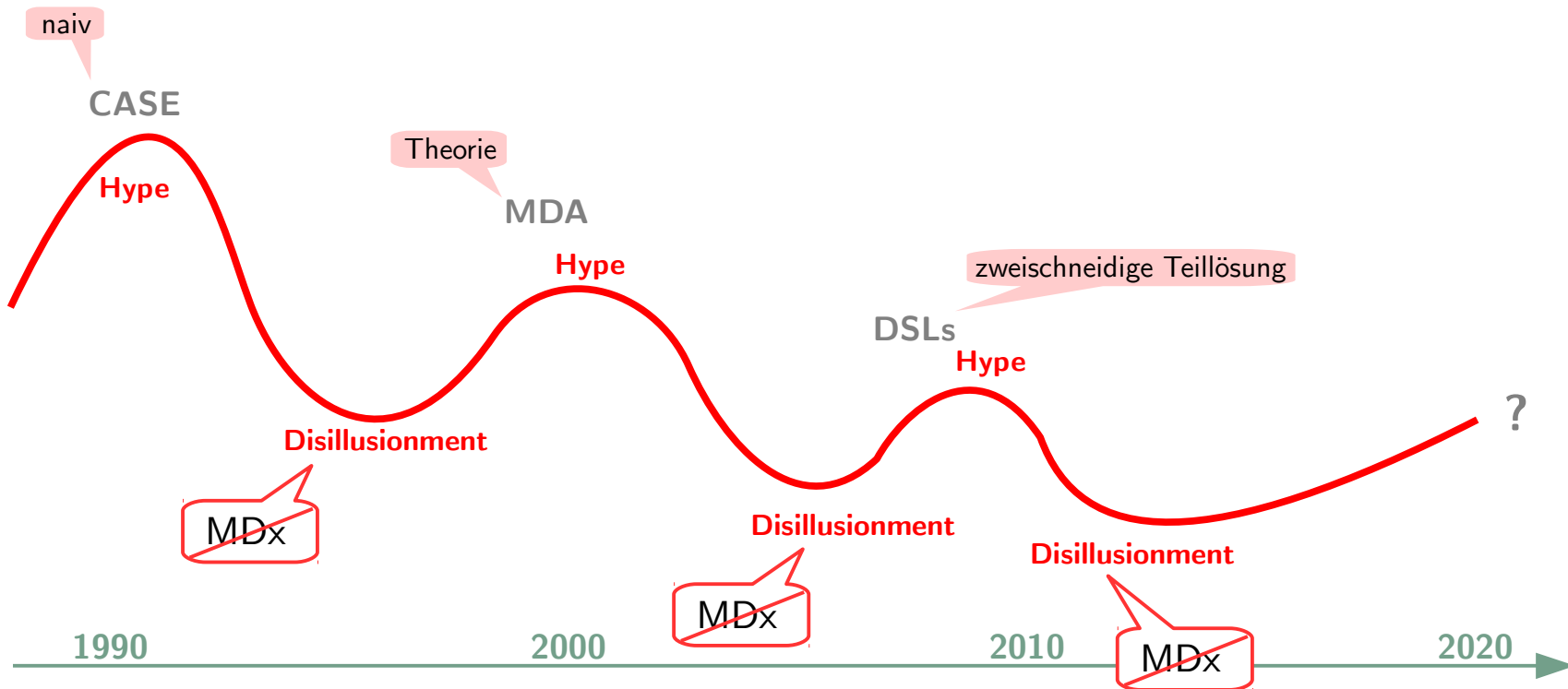
IT-Systeme



Mehr als 25 Jahre “Model Driven”

MD Technology

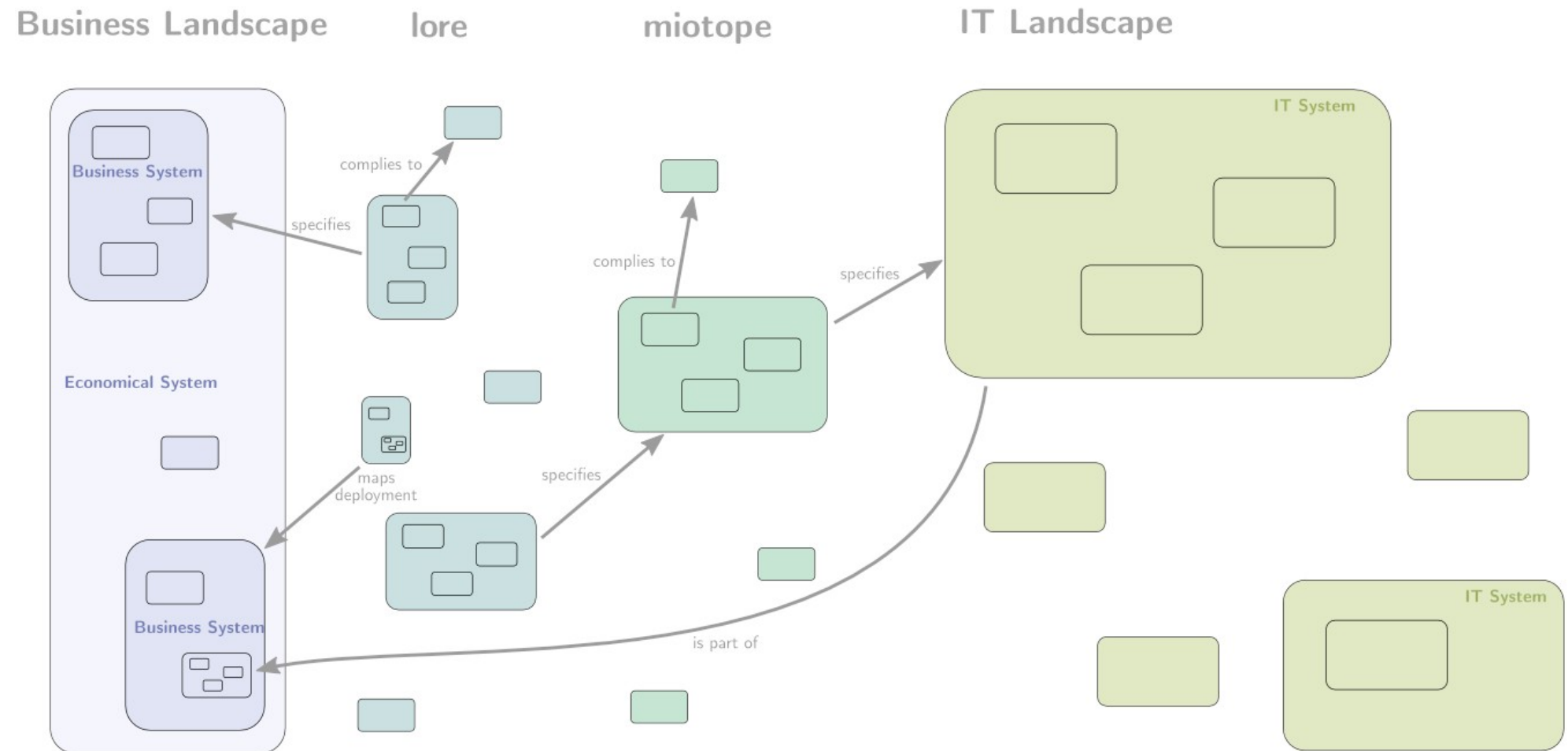
Gartner Hype Cycle



Enlightenment?

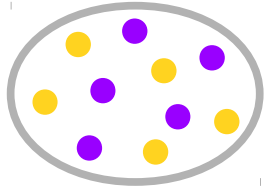


Artefakte und Modellierung

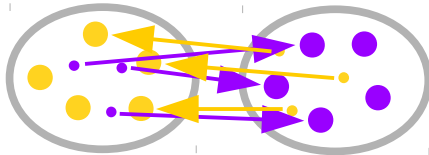


Bewertung Gesamt-System

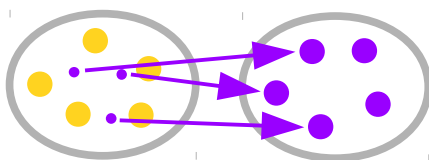
SoC 3 : vermischt



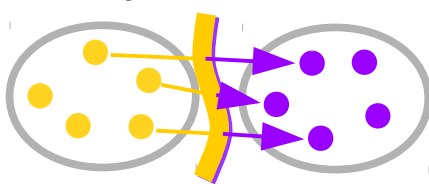
SoC 1 : getrennt, Dependency <-->



SoC 2 : getrennt, Dependency -->

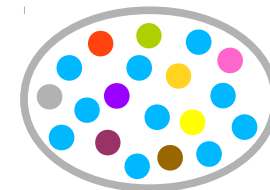


SoC 0 : getrennt, abstract Dependency



- Code
- Komponenten
- Architektur
- Organisation
- Datenbestand
- Deployments
- Akteure
- Unternehmen

DRY 3 : wild verstreut, unwartbar



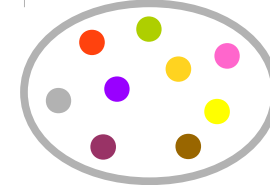
DRY 2 : mehrfach



DRY 1 : genau einmal



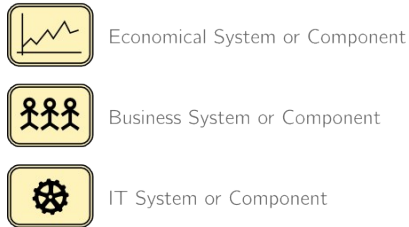
DRY 0 : gar nicht



Modell Klassifikation

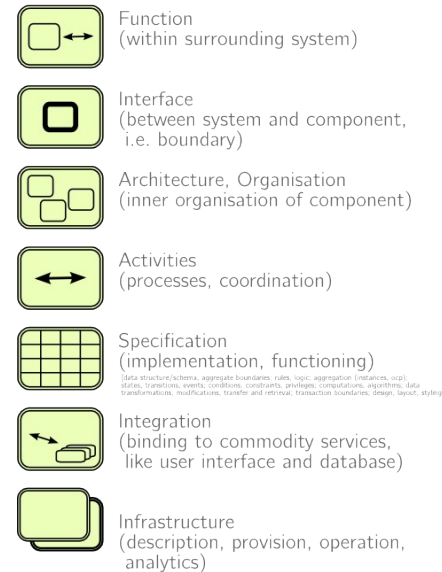
Modelling Object

What is modelled?



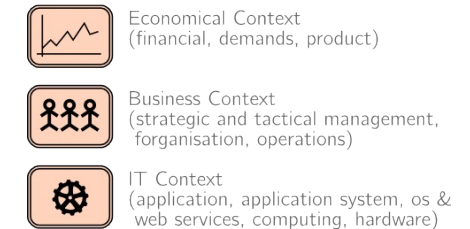
Modelling Aspect

Which part is modelled?



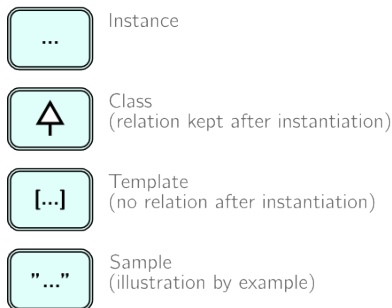
Modelling View

From which perspective modelled, for whom?



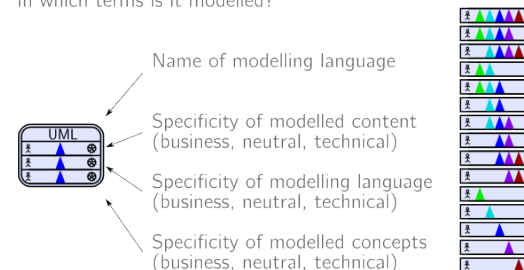
Modelling Genericity

How generalised is the model?



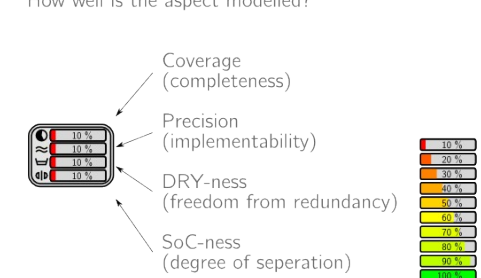
Modelling Language

In which terms is it modelled?



Modelling Quality

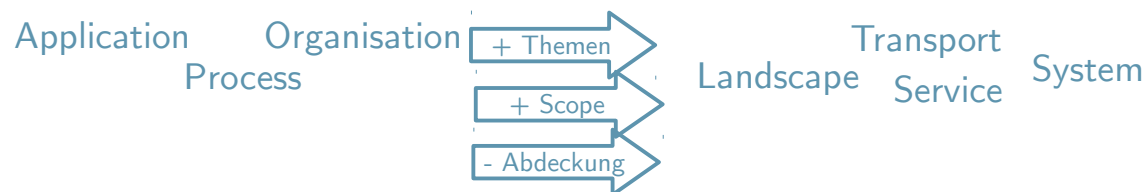
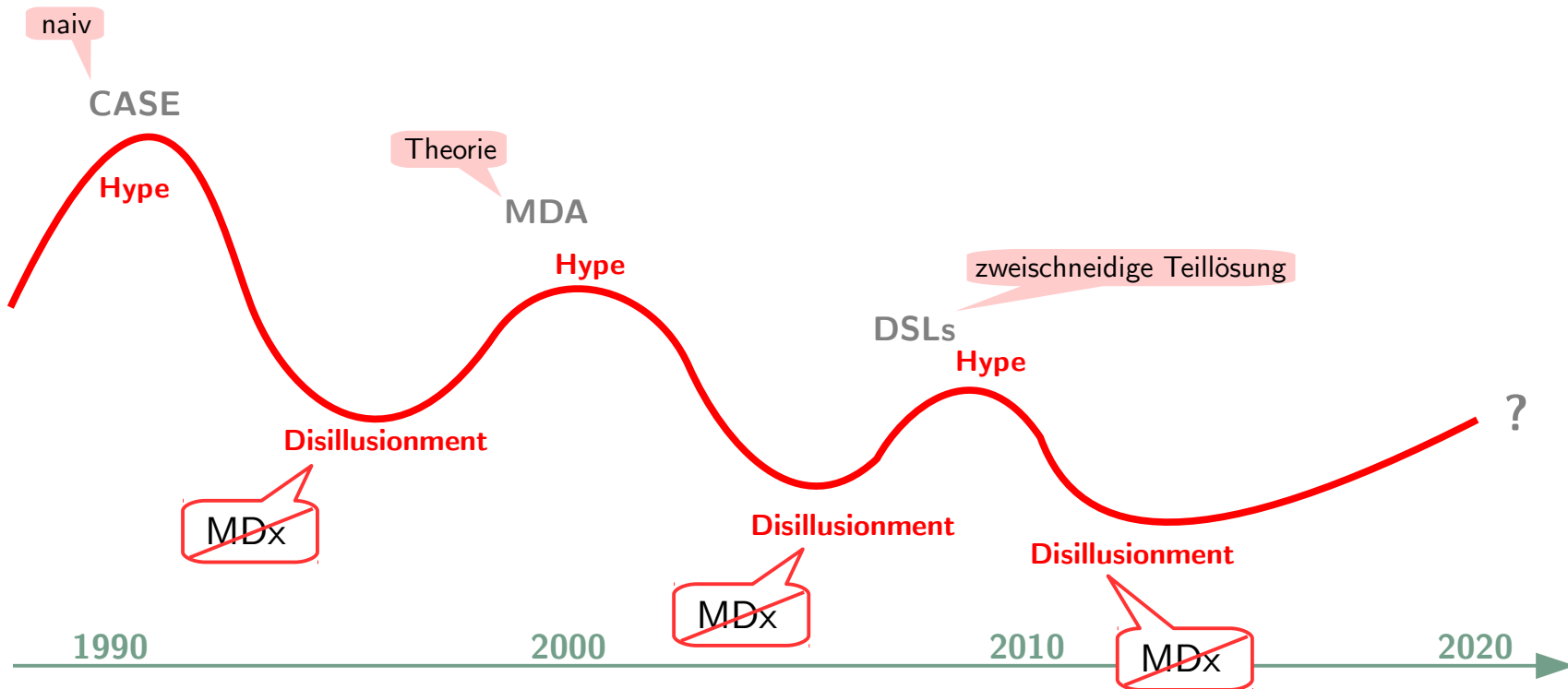
How well is the aspect modelled?



Mehr als 25 Jahre "Model Driven"

MD Technology

Gartner Hype Cycle



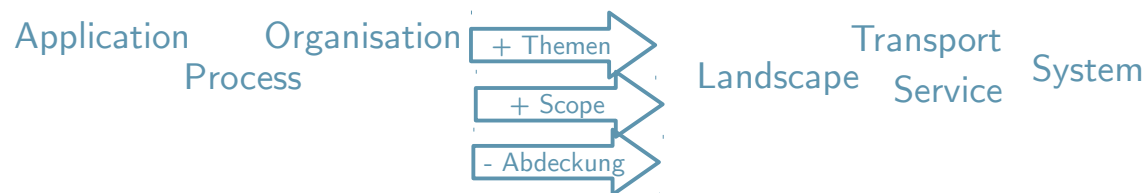
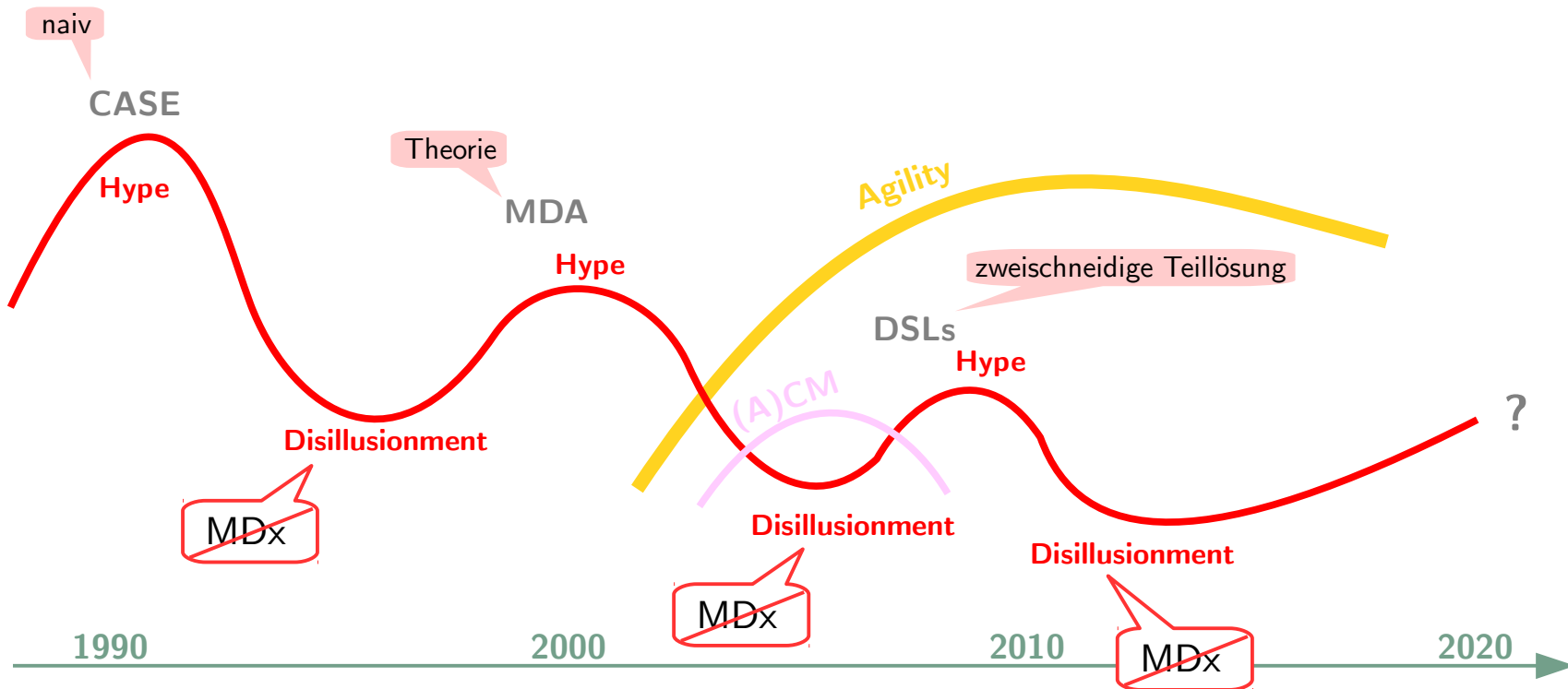
Enlightenment?



Mehr als 25 Jahre "Model Driven"

MD Technology

Gartner Hype Cycle



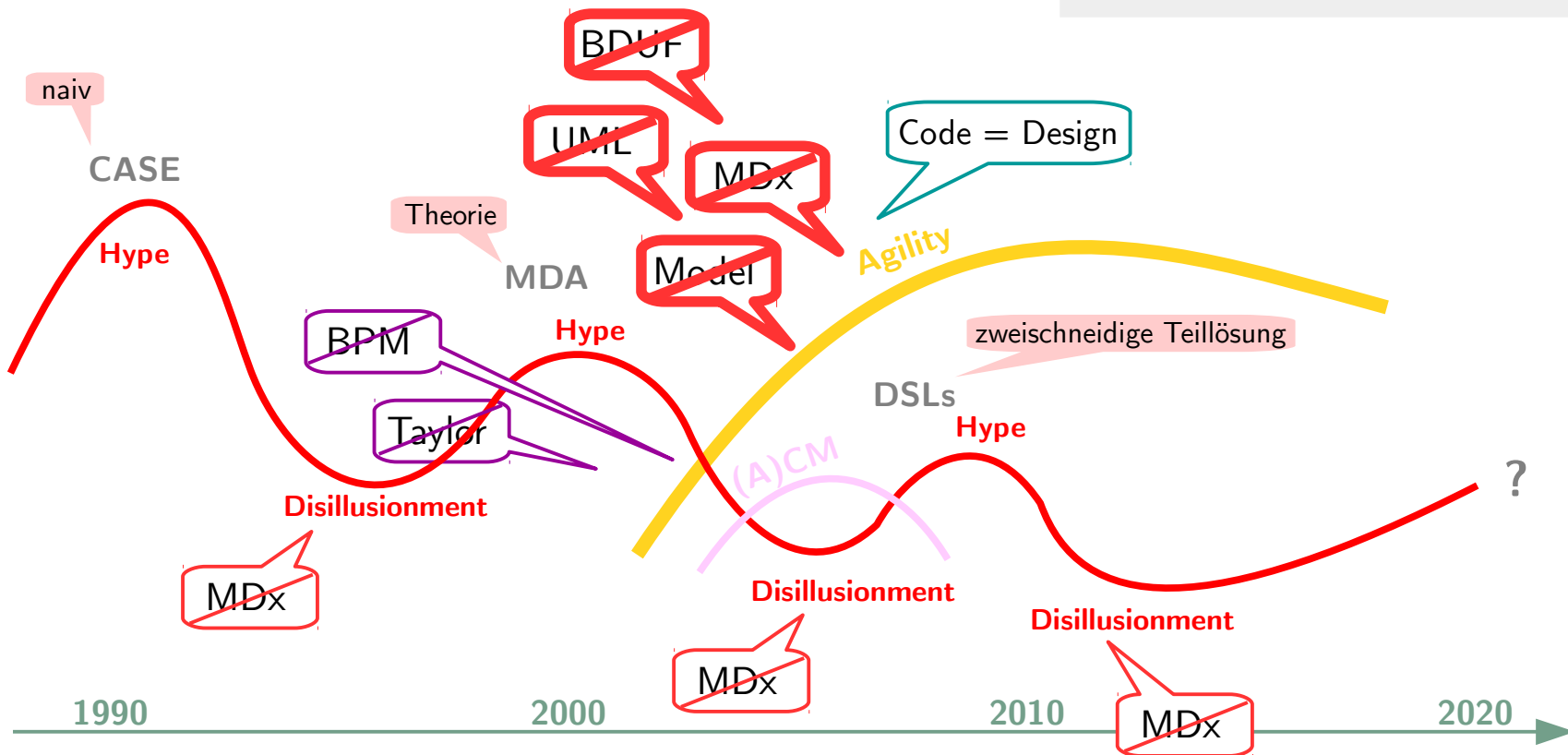
Enlightenment?



Mehr als 25 Jahre “Model Driven”

MD Technology

Gartner Hype Cycle

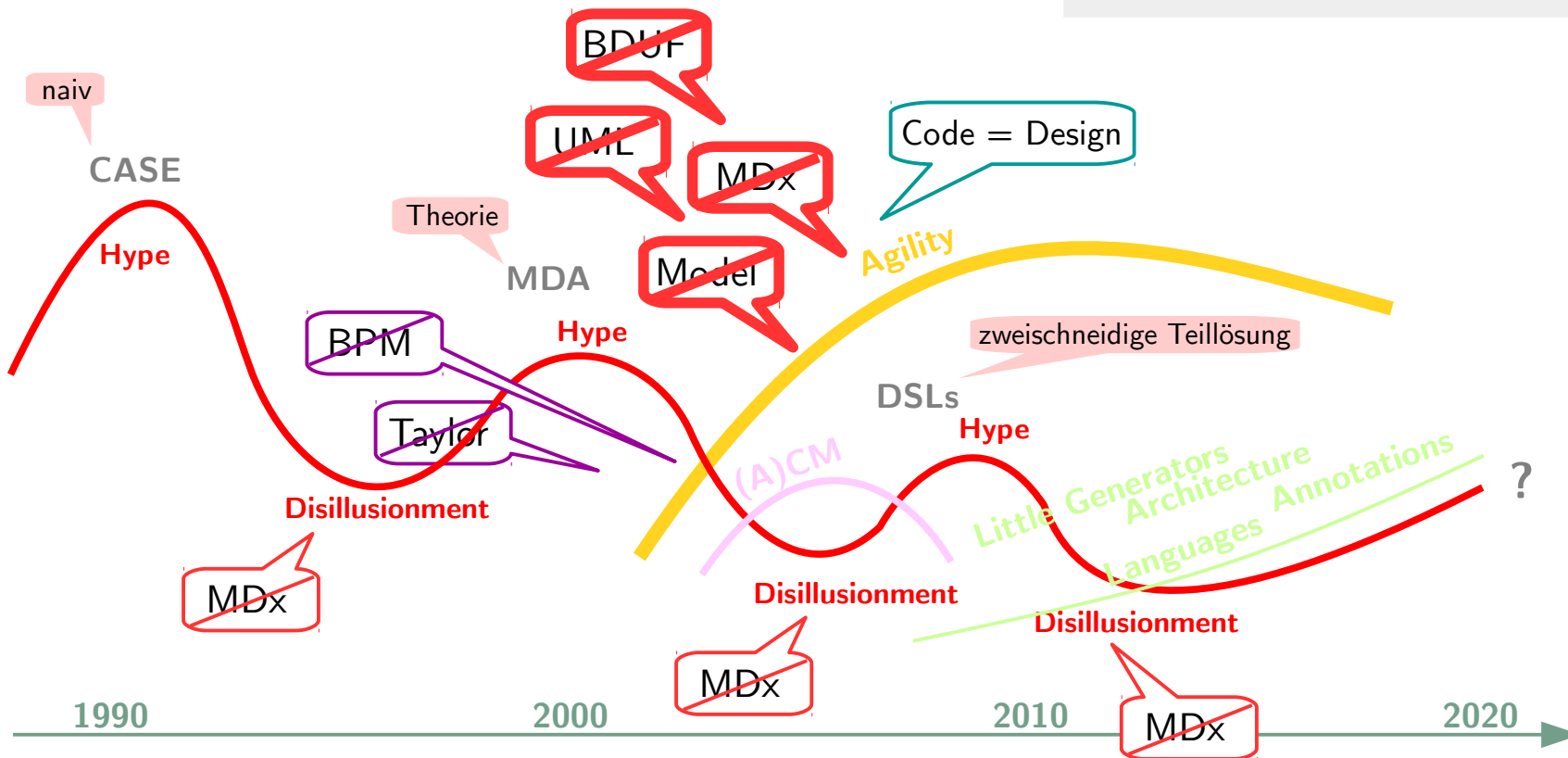


Enlightenment?

Mehr als 25 Jahre “Model Driven”

MD Technology

Gartner Hype Cycle

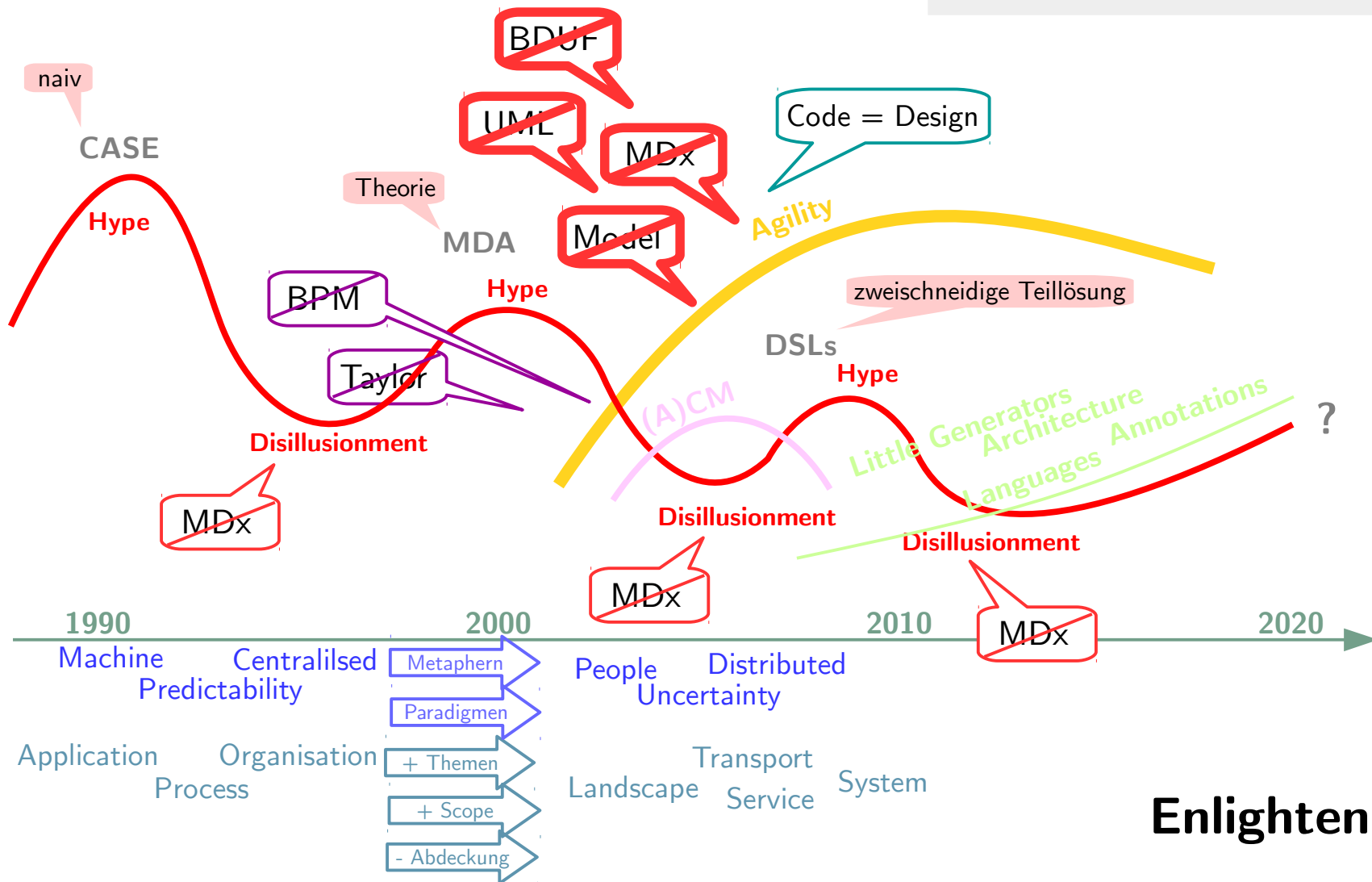


Enlightenment?

Mehr als 25 Jahre "Model Driven"

MD Technology

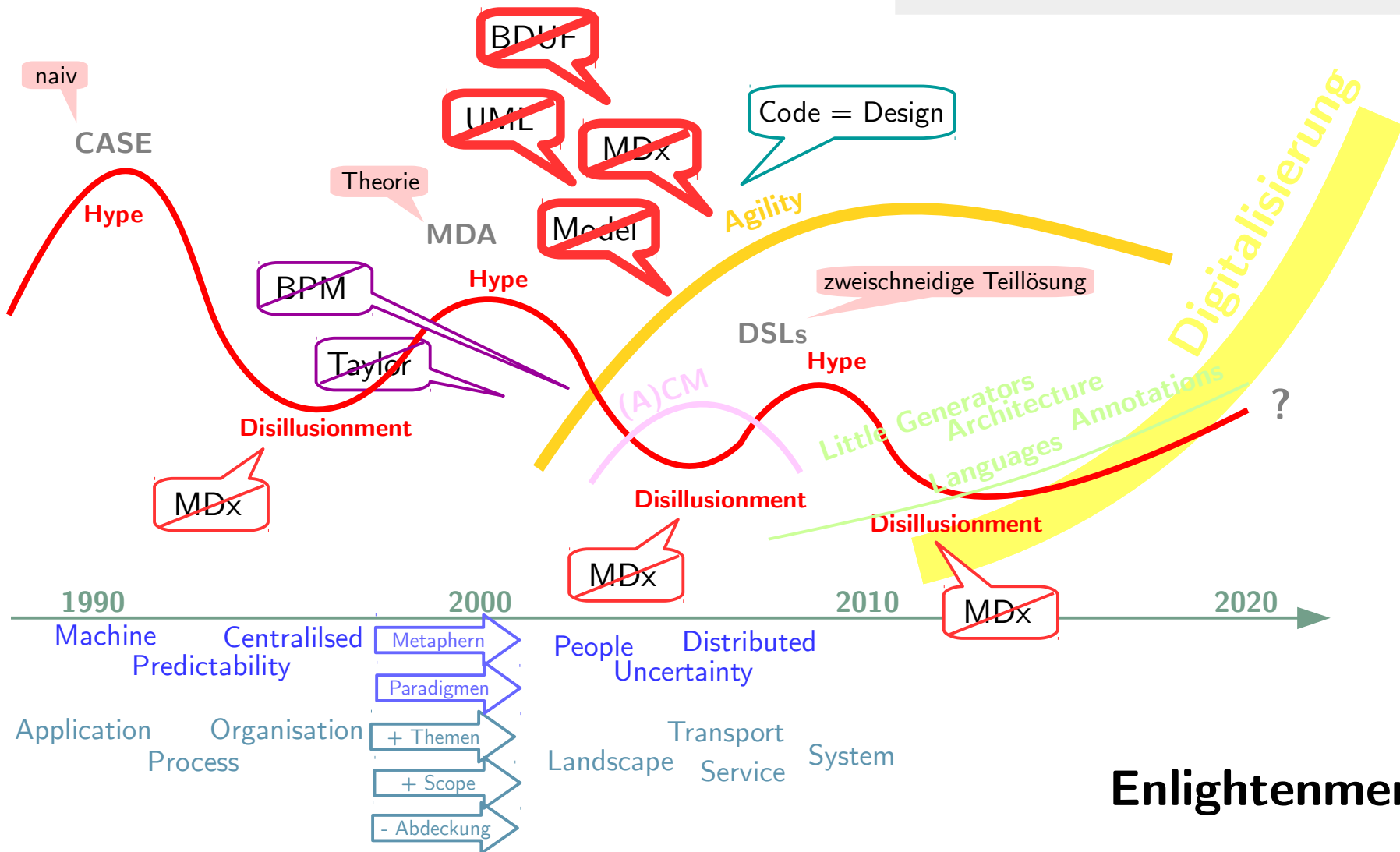
Gartner Hype Cycle



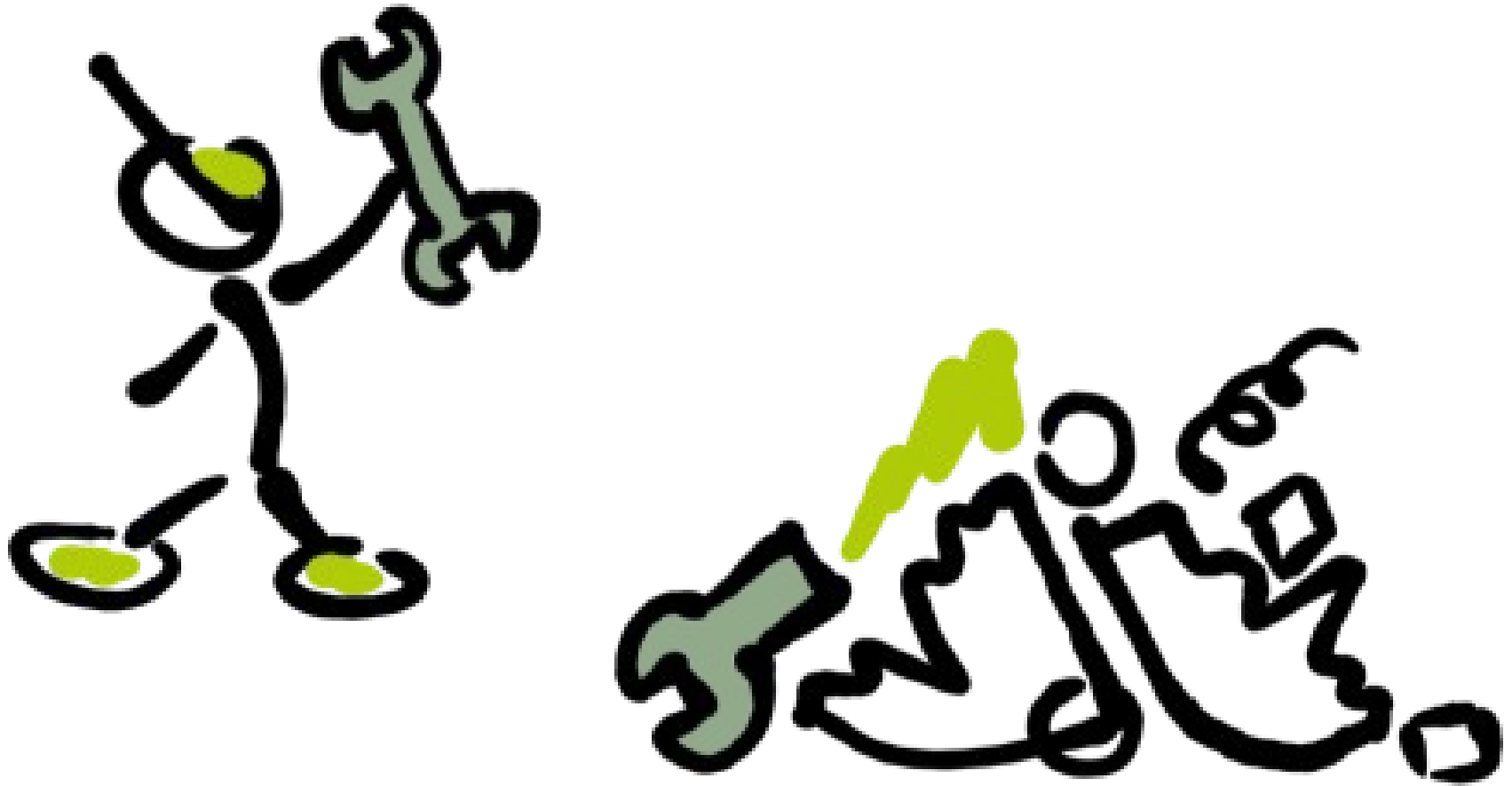
Mehr als 25 Jahre "Model Driven"

MD Technology

Gartner Hype Cycle

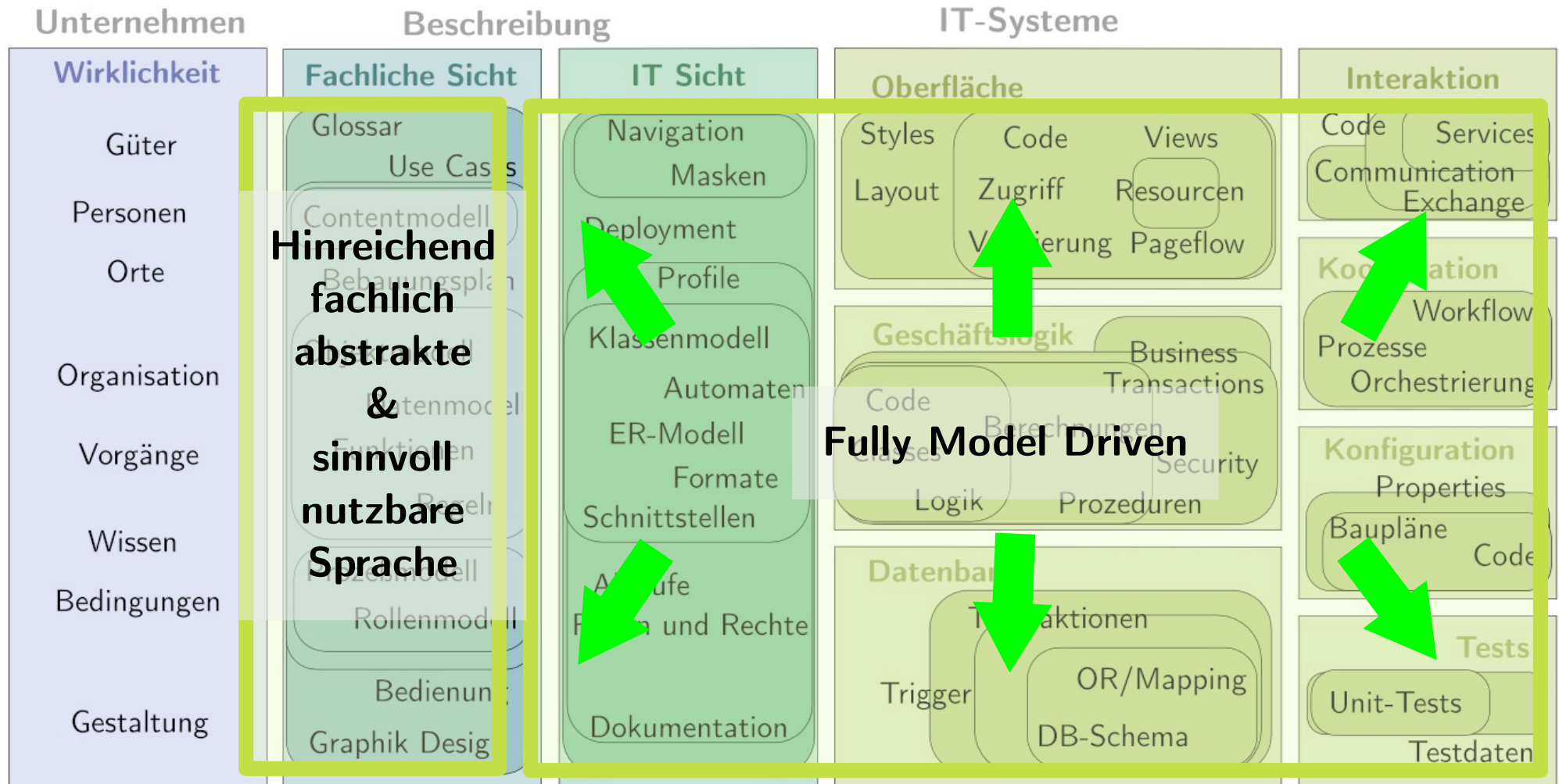


MDx ...auf dem Holzweg?





MDx - nicht-triviales, unerreichtes Moving Target



Inhalt

Einführung

Motivation und Ansatz

Software Produktion: Sprachen, Abstraktion, Modelle
EM/OS - Übersicht

Business Design

Service & Brand: fairyloom & businessboard
Business Model: OSS, Tool & Service

Technologie

EM/OS – Architektur & Innovationen



Kritische Masse:



“...unerreichtes
Moving Target...”
25 Jahre, rührend
naiv unterschätzt

vollautomatischen **Erzeugung**

vom Modell zum Deployment

produktionsreifer
Unternehmenslösungen

kein Nachbearbeiten oder Ergänzen nötig
Fokus: Individualität, Flexibilität und
Skalierbarkeit über ganzen Life-Cycle
keine Modellierung technischer Artefakte
(wie Hilfsklassen oder UI-Widgets)

aus fachlich-prägnanten,

vollständigen und unifizierten,
single-source
Geschäfts-Modellen

UI, BL, DB, S/P innovativ integriert
kein zusätzlicher Code (< 1% Logik im Modell)
textuell, graphisch, in Dateien, DB oder wiki

- ➡ prägnante Spezifikation ist gut **zu handhaben und verständlich**
- ➡ **keine Bindung** an Hersteller, Technologie oder Wissenträger
- ➡ automatische Produktion **spart Zeit und Arbeit**



“...nicht trivial...”

EM/OS – Innovationen

True Model Driven User Interface

Business Logic State Machines

Renderer und Stylesheets

Unified Process Modelling

1% fachliche Logik

Kollaboration & Planung

Landscape-Maps

Hohe Qualität des Modells

Hohe Qualität des Ziel-Systems

Hohe Qualität der Werkzeuge

business board: collaborative online modeller

oomodels: open model repository

fachliche, also nicht-technische **Properties am Fachmodell**

erzeugen medienunabhängiges **Virtual User Interface**

das auf beliebige Frontends abgebildet wird

UI-Interaction fachliche Micro-Processes = medienneutrale Flows

BL-Behaviour/States konsistentes Verhalten und BPM-Verknüpfung

DB-Transaction Control ideale Grundlage für Model Driven Approach

Unabhängig vom Model CD/CI-Styling anwendbar auf beliebige Projekte

Any-UI Web, Desktop, Mobile

UI-Varianten Branding, Devices, Preferences

Prozesse und Objekte verknüpft systematisch via States

Token-freie Workflows weniger störanfällig und vereint BPM und CMM

explizites Material ermöglicht UI-Erzeugung

keinerlei technischer Code kein Glue-Code, kein TX-Code, kein DB-Code

keine UI-Callbacks, kein Init/Exit/Config/Setup

Sprach-Plugins derzeit vor allem Java, offenes Konzept

an jedem Modell-Element jederzeit zur Hand, verständlich

integriertes Project Management verschiedene Verfahren: klassisch oder agil

keine Redundanz

für Deployment automatisch via business board

für Backup Erzeugung von bspw. scripten

für Monitoring Erzeugung von bspw. nagios-Konfiguration

vollständig 100% spezifiziertes System

verständlich lesbar und veränderbar für sowohl Fach- als auch IT-Experten, guter Abstraktionsgrad

DRY, SoC und fein-modular flexibel und teilweise verwendbar und ersetzbar

kontextsensitiv zur Laufzeit automatische Anpassung an Services der Umgebung (UI, DB, APIs)

wartbare Transformationen DRY, SoC und Verständlichkeit von M2M, M2T und Runtime-Aggregation

Alles an einem Ort, Support & Zuarbeit per Mausclick

Powered by Itself EM/OS-App incl. Graphical Modeller !

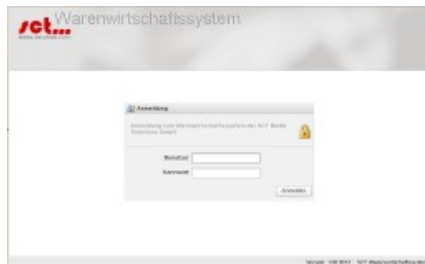
“wikipedia für Modelle”, Community-Exchange

Referenz für App-Collaboration Bounded Context/Transfer Models



Projekte mit EM/OS

Warenwirtschaftssystem



Lagerverwaltung
ca. 180 fachliche Klassen
vollständig Modell-generiert

Produktportal



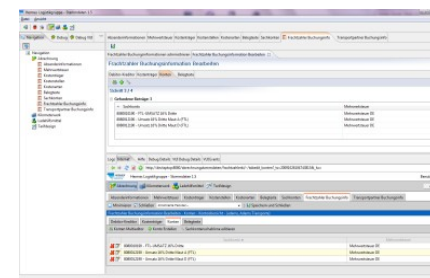
Umfangreicher Katalog
Online CMS
Produkt-Assistent

Online Versicherungsmakler Plattform



Interpretiertes Modell
incl. Content
pixelgenaue Layout Vorgabe

Stammdaten-Administration



Legacy Datenbanken
pixelgenaue Layout Vorgabe
Web-Client & SWT-Client

Weitere Projekte

- Modifizierbare Logistik-Schnittstelle
- Microservice-Multilayer-Generierung
- onsecure (Versicherungsplattform)
- businessboard, fairyloom portal

EM/OS Fakten

Erprobte Ziel-Technologien

Oberflächen

HTML(5), CSS(3, responsive),
JS (extjs, vue.js), SWT/RCP,
Text, Script, WAP, VRML

Datenbanken

JPA, Hibernate, TopLink/eclipselink,
MySQL, PostgreSQL, Oracle, Versant

Services

REST, (SOAP)

Prozesse

UBPML

Office

Excel, PDF, OpenOffice, Reports

Zahlen

Engine

Java Klassen	3696
Objektbaupläne (incl. DynM2M)	428
Templates (M2T)	321
JARs	96
Java Code (Lines)	477928
Java Code (Bytes)	20734236

Kunden-Projekt

Modell Klassen	180
Attribute	551
Operationen	159
Assoziationen	100
Vererbungen	62
Stereotypen	333
Erweiterungen	366



Inhalt

Einführung

Motivation und Ansatz

Software Produktion: Sprachen, Abstraktion, Modelle
EM/OS - Übersicht

Business Design

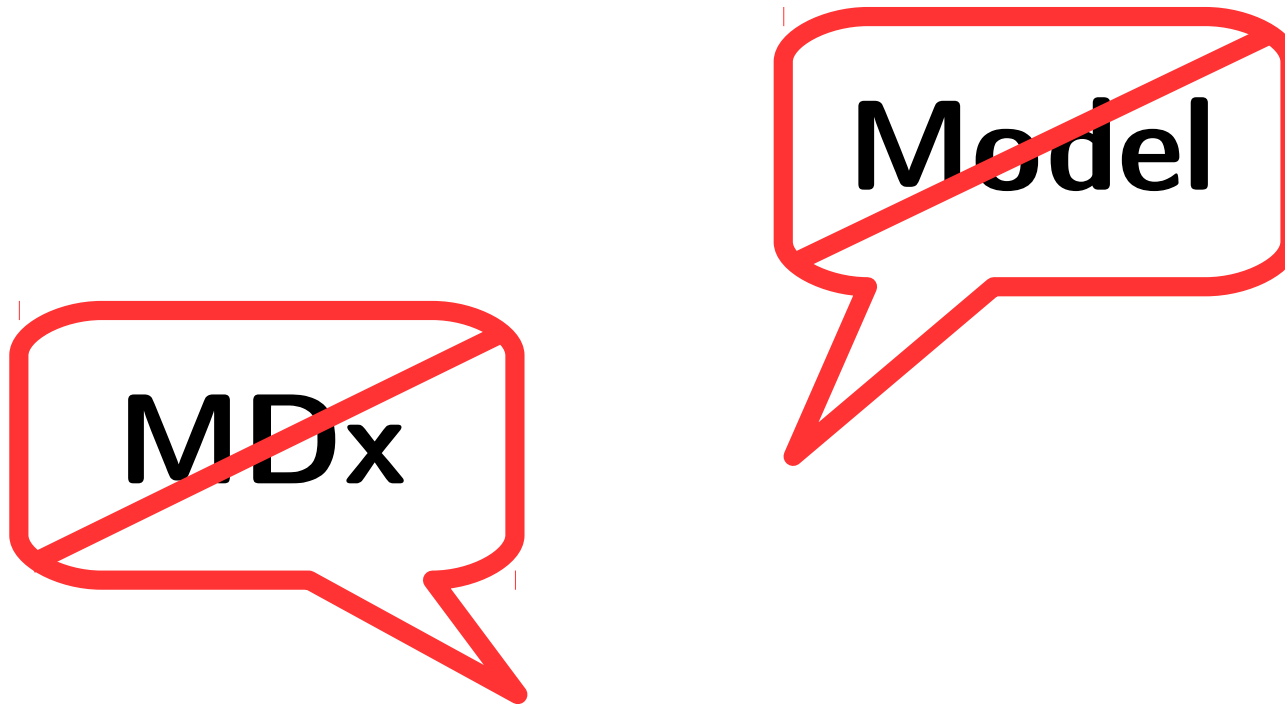
Service & Brand: fairyloom & businessboard
Business Model: OSS, Tool & Service

Technologie

EM/OS – Architektur & Innovationen

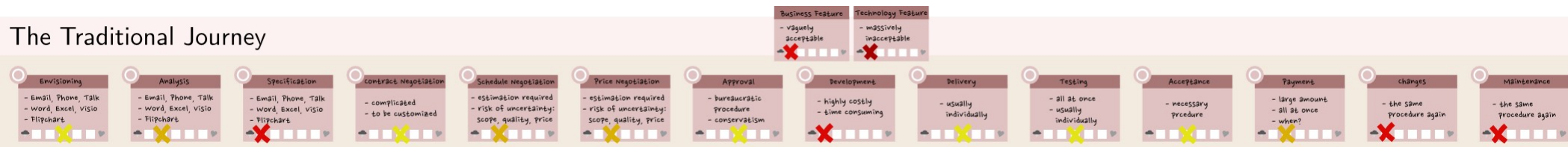


Problem

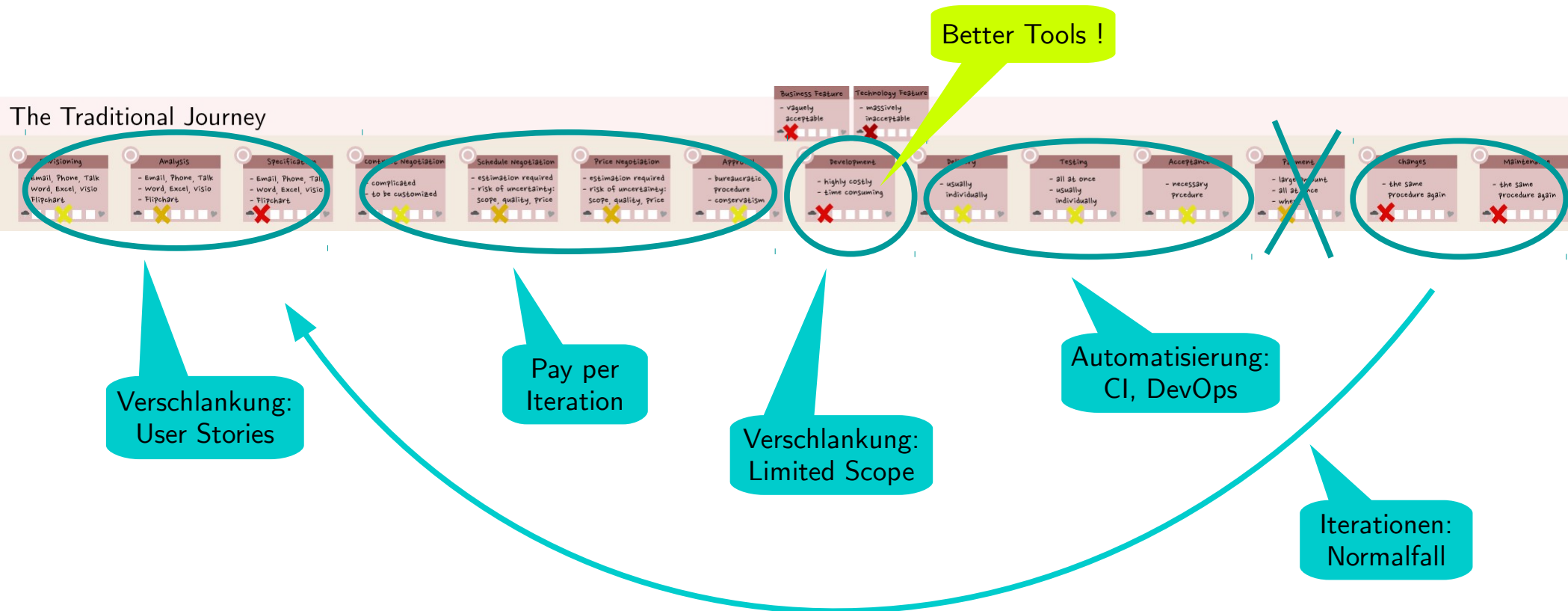


Service Design - Traditional Journey

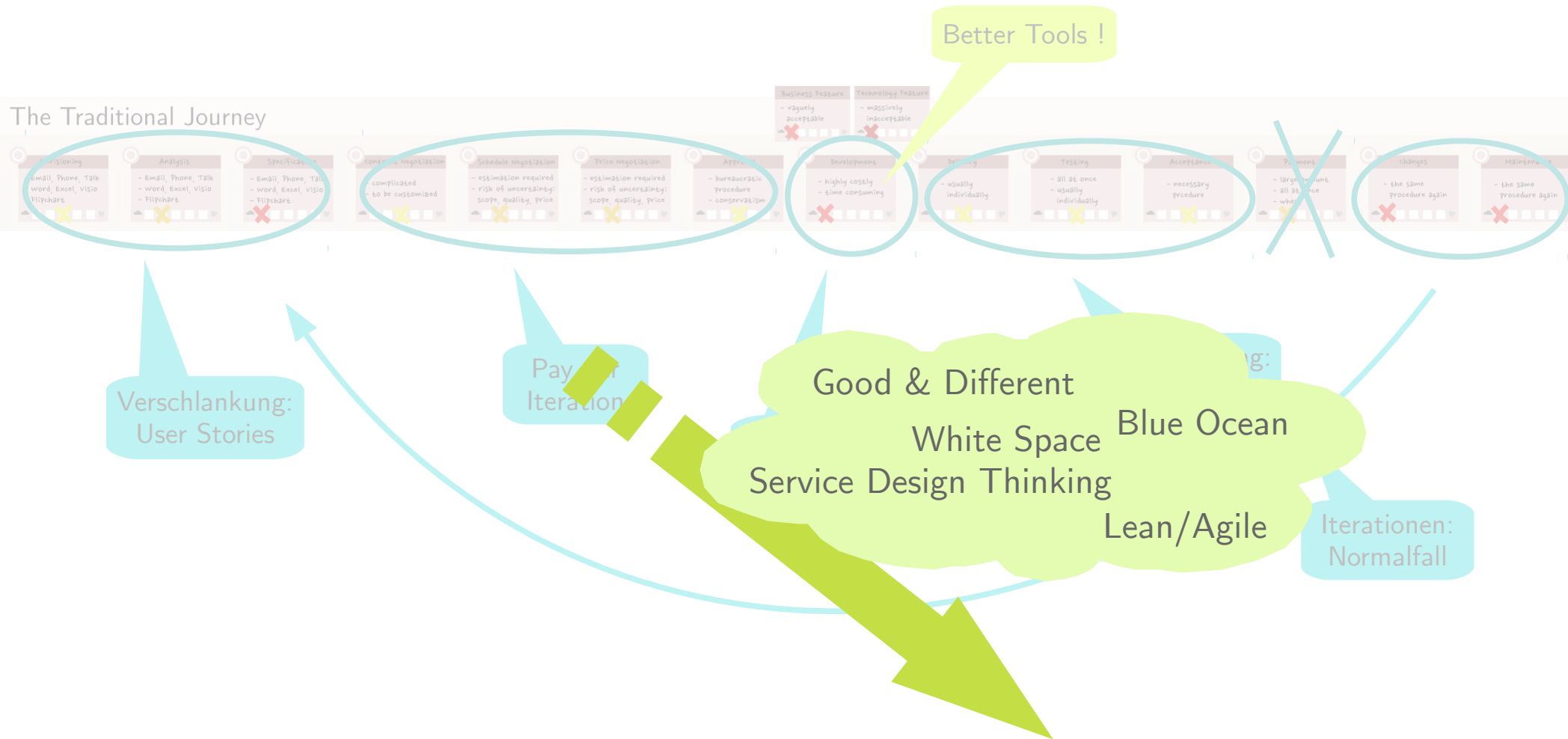
(in extern vergebenen Gewerksaufträgen immer noch vorherrschend)



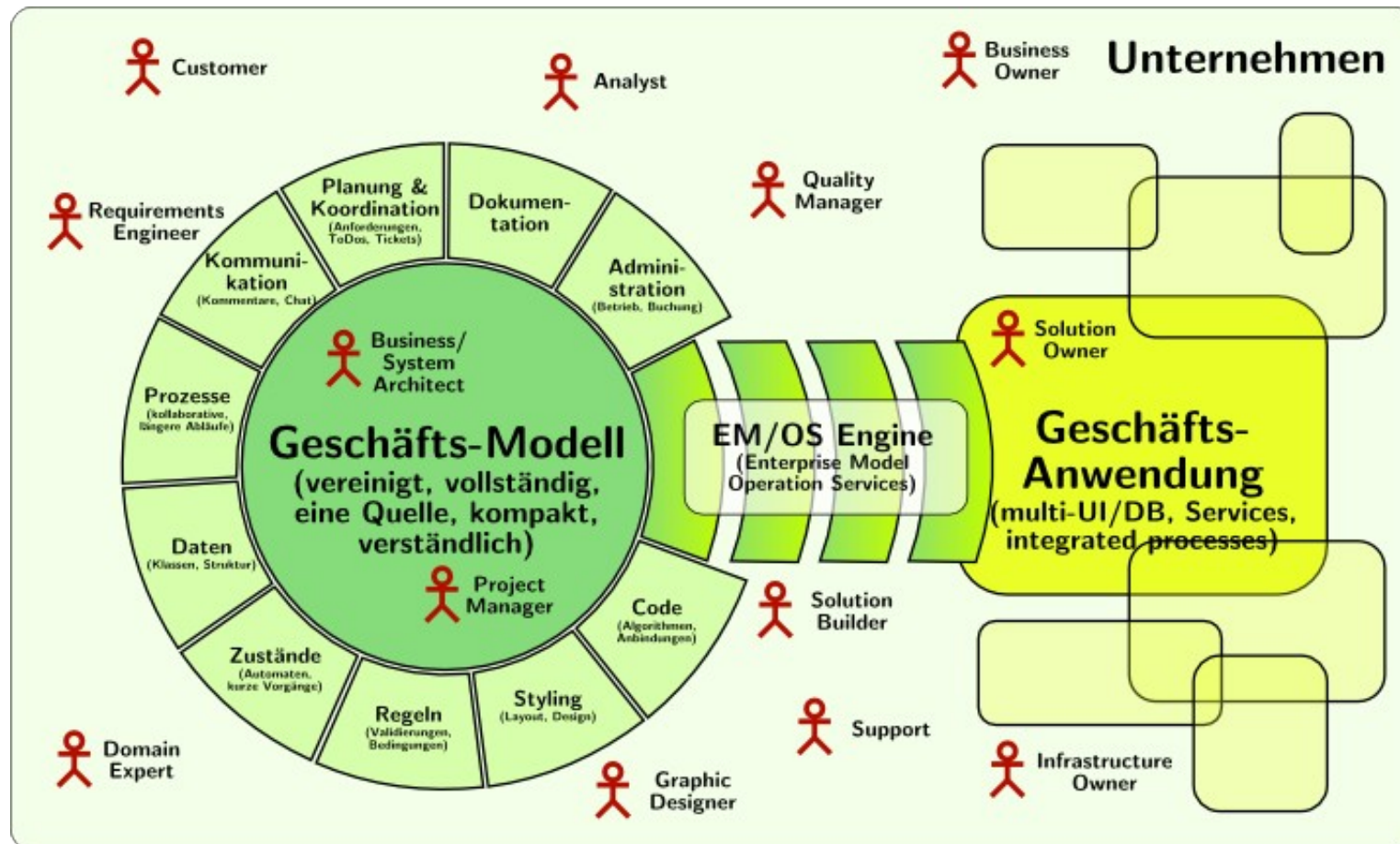
Service Design - Agile Journey



Service Re-Design

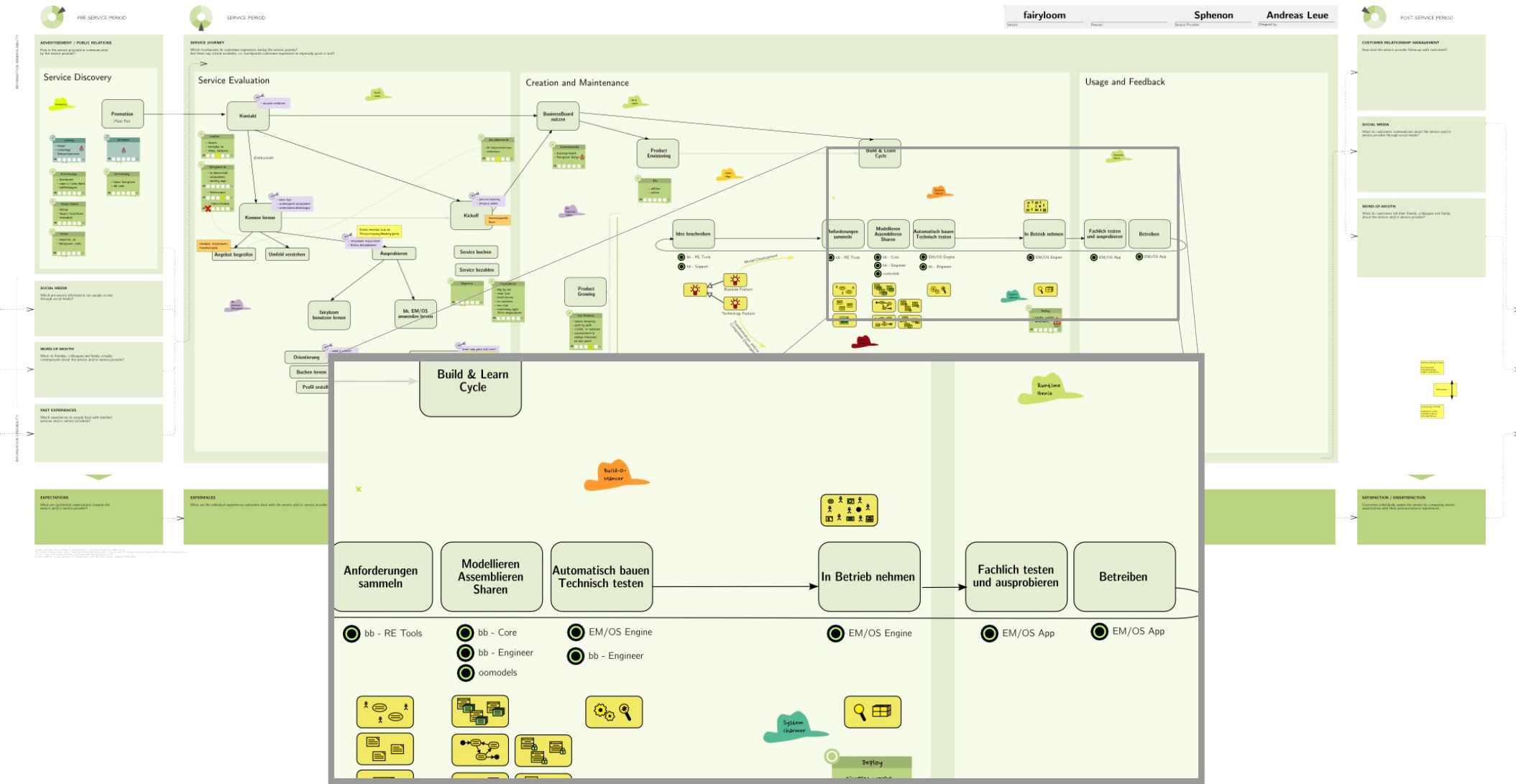


Service Re-Design – 1. Actor & Artefact Scope: businessboard



Service Re-Design - 2. Process Scope: fairyloom

The Customer Journey Canvas



Metaphern und Paradigmen

Machine

Construction

Generation Production

Centralised

Predictable Heavyweight

Engineering

Metal, Screws & Oil

Eternity

Model

Single

Monolithical

Static

Organisation

Application

Craftsman(ship)

Cooperation

Talk

People

Distributed

Social

Agile

Lightweight

Uncertainty

Story

Interacting

Network

Economy

Iterative

Environment

Dynamic

Landscape

Fairytale

Growing

Art

Dwarf

Fairy

Mage

Loom

Magic

Sunlight

Wood & Water

Tales

Recipes

Biology

Lore

Story & Genes & Plan

Evolution

Marke “fairyloom”



Service



Tool



- Brandplay
- Domain
- usw.



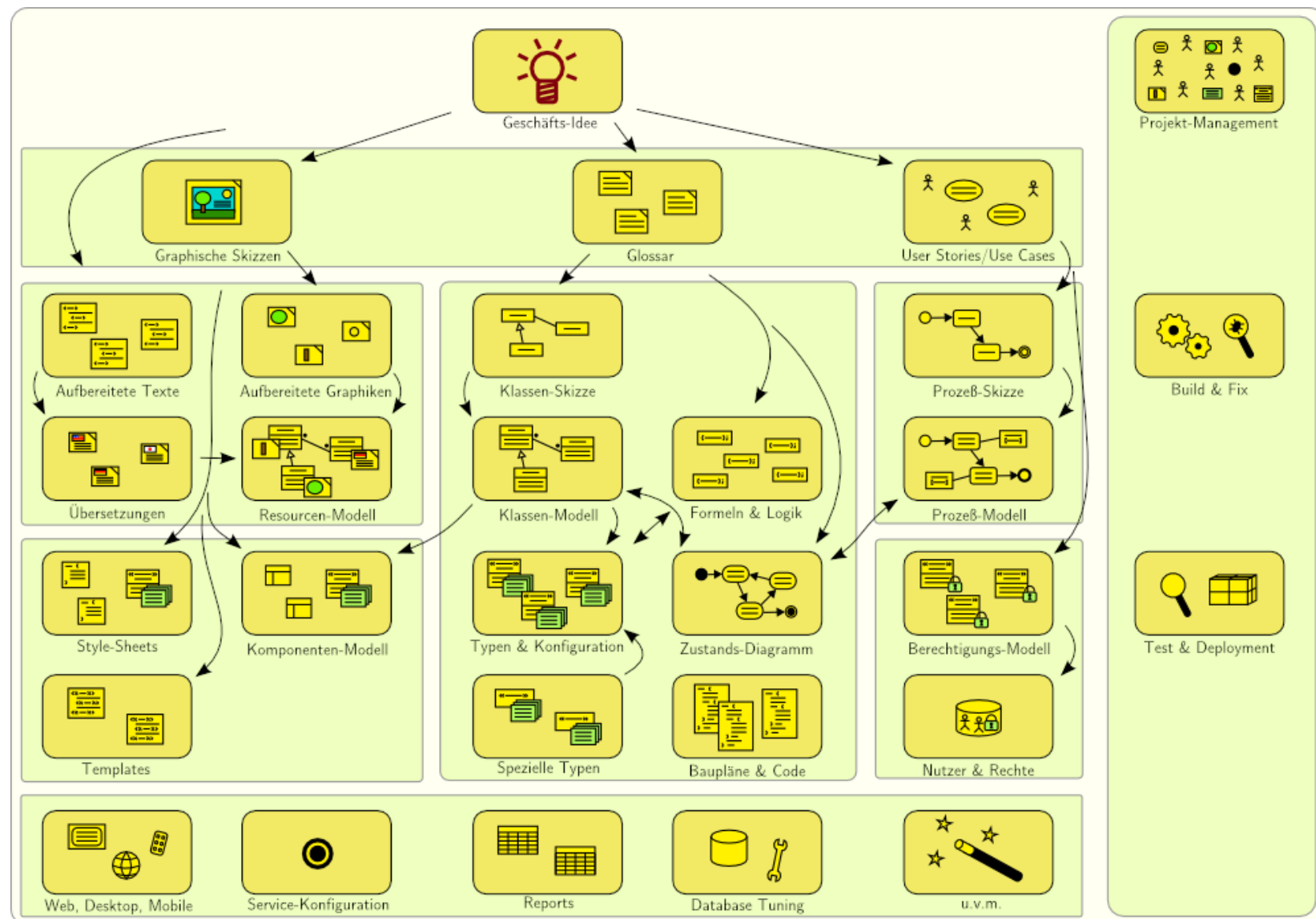
Das

ist ein **Modeller** und ermöglicht

die Bearbeitung des **vereinheitlichten Geschäfts-Modells** (fachlich, single-source)
verteilt in der **Cloud** (anywhere/anytime, desktop & mobile)
und im **Team** (intern/extern)
sowie Dokumentation, Arbeitsplanung und -Koordination
und die **Erzeugung passender Lösungen** mit EM/OS
(und ist selbst eine **100%ige EM/OS-Anwendung**)

- ➔ Spezifikation und Planung **griffbereit an einem Ort**
- ➔ **keine Bindung** an Arbeits-Zeit oder -Ort
- ➔ jederzeit **Einbindung von Experten**
- ➔ **Delegation von Arbeit** per Mausklick

Business Board – EM/OS-Methode

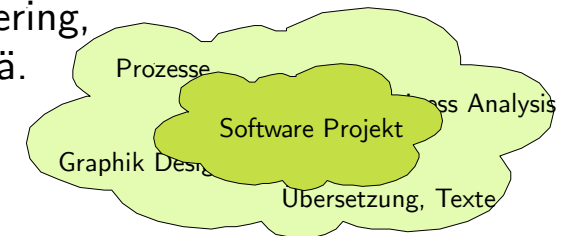




ist ein **Shop** und ein **Service-Portal** in dem

Geschäfts-Kunden ein **businessboard**
und bedarfsweise zugehörigen **Service** mieten
sowie Arbeitspakete **in Auftrag geben** können

künftig können dort zugehörige Dienstleistungen wie Requirements Engineering,
Graphics Design, Texte, Übersetzungen, Anwendungs-Betrieb, Testing u.ä.
angeboten werden



- ➔ Kunden können **per Mausklick ein Projekt initiieren**
- ➔ **keine Bindung** durch Anfangsinvestition (Mietmodell)
- ➔ Projektmitarbeiter müssen **nicht rekrutiert** werden

fairyloom

www.fairyloom.de

Benutzer

.....

Anmelden

Registrieren...

fairyloom

business tools at your service

... versorgt Sie enorm schnell und bequem mit **professionellen Software-Lösungen**, die **genau auf Ihr Geschäft zugeschnitten** sind und sich zudem laufend **selbsttätig an Veränderungen anpassen**. Sie erhalten diese ohne Vertragsbindung, ohne Technologiebindung, ohne Herstellerbindung und als Open Source.

Bedarf im
Geschäftsmodell
festhalten



1



iterativ

verfeinern

und

ausprobieren



2



3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

Passende
Lösung
nutzen

3

Ideal für innovative und
industrielle Unternehmen,
geeignet für alle Branchen

[Kontakt](#) [Impressum](#)


Inhalt

Einführung

Motivation und Ansatz

Software Produktion: Sprachen, Abstraktion, Modelle
EM/OS - Übersicht

Business Design

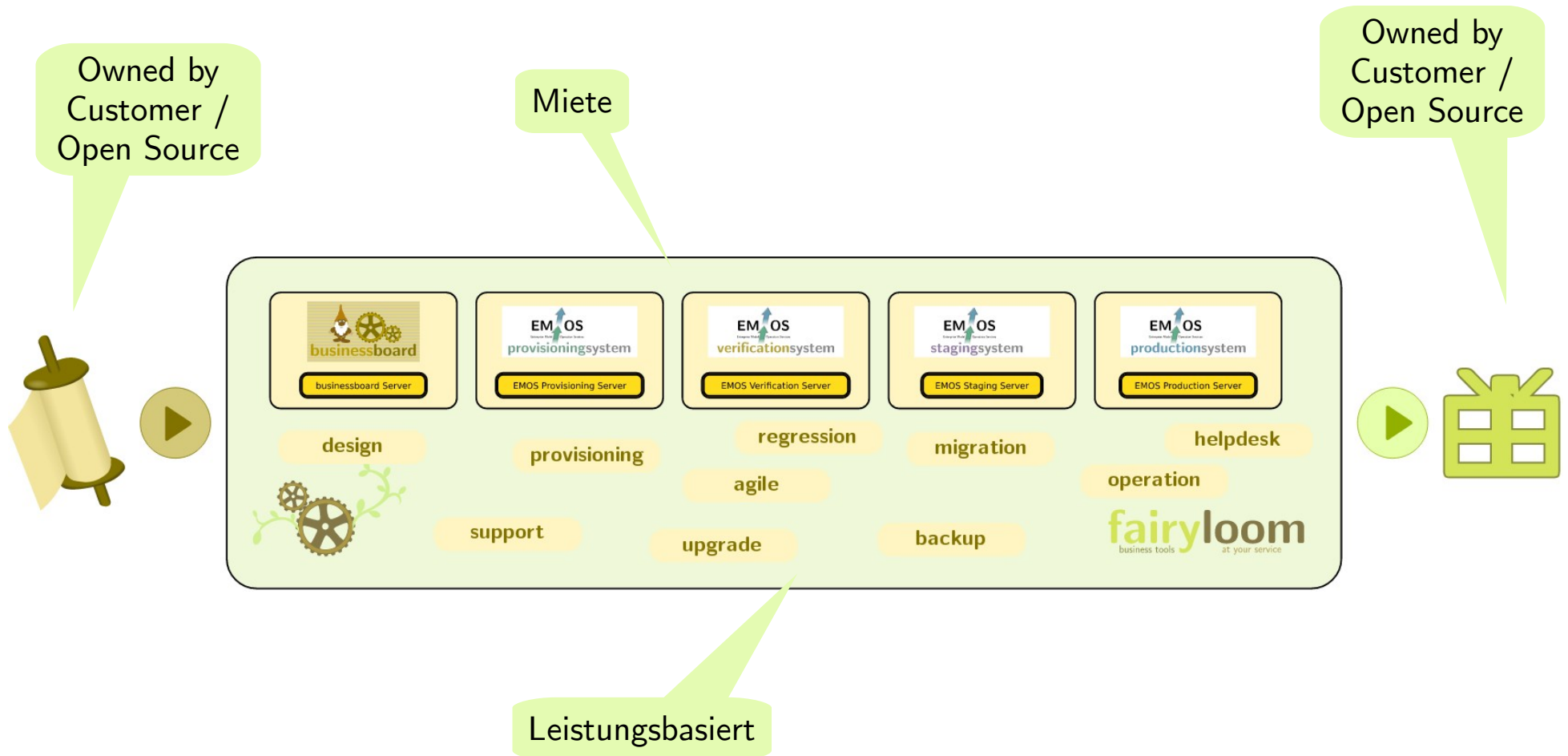
Service & Brand: fairyloom & businessboard
Business Model: OSS, Tool & Service

Technologie

EM/OS – Architektur & Innovationen



Business Model



Business Model Szenarien

+ möglich (einfach)
- Timing? (langsam)

- EM/OS-boardboard-Individual-Projekte,
Finanzierung über Miete & Service (EM/OS-Consulting)

- Marketing- & Vertriebs-Synergie mit fairyloom

+ win/win/win
- weiteres BM (komplex)

- Partnerschaft mit etabliertem Unternehmen

+ Vorhandenes nutzen
- BM-Alignment

- Venture Capital + potentiell schnellstes Wachstum
- Erfahrung, Risiko, Steuerung

- Business Crowd Funded Applications

+ High Potential, Coolness Bonus,
Perfect Timing (Trends!)
- Unbekanntes Terrain

➔ Geschäftliche Optionen



Partnerschaft Szenarien

Projekt- und Vertriebs-Partner

nutzen EM/OS oder das businessboard für Projekte eigener Kunden
oder vermitteln EM/OS oder das businessboard an Kunden gegen Provision

fairyloom-Anbieter

bieten in fairyloom Leistungen als eigenständiger Anbieter an
die der Kunde bequem vom businessboard aus steuern kann
(je nach Leistung als eigenständiges BB-Modul oder integriert in das Software/Prozeß-Modul)
die der Kunde (zumindest teilweise sinnvoll) auf Mietbasis nutzen kann
erleichtern dem Kunden die Zusammenstellung von fairyloom-Angeboten durch enge
technische und organisatorische Kooperation mit anderen Anbietern

Externe Mitarbeiter, Unterauftrags-Nehmer (EM/OS-Experten, Modell-Architekten)

Unterstützen Sphenon bei Projekten für unsere eigenen Kunden

Technologie-Partner (strategisch/taktisch)

Arbeiten auf konzeptioneller Ebene oder bei der Produktentwicklung zusammen.

Marketing-Partner

unterstützen uns und/oder wir sie beim Marketing
sind gerne kreativ, innovativ und am Puls der Zeit – so wie wir uns verstehen



Inhalt

Einführung

Motivation und Ansatz

Software Produktion: Sprachen, Abstraktion, Modelle
EM/OS - Übersicht

Business Design

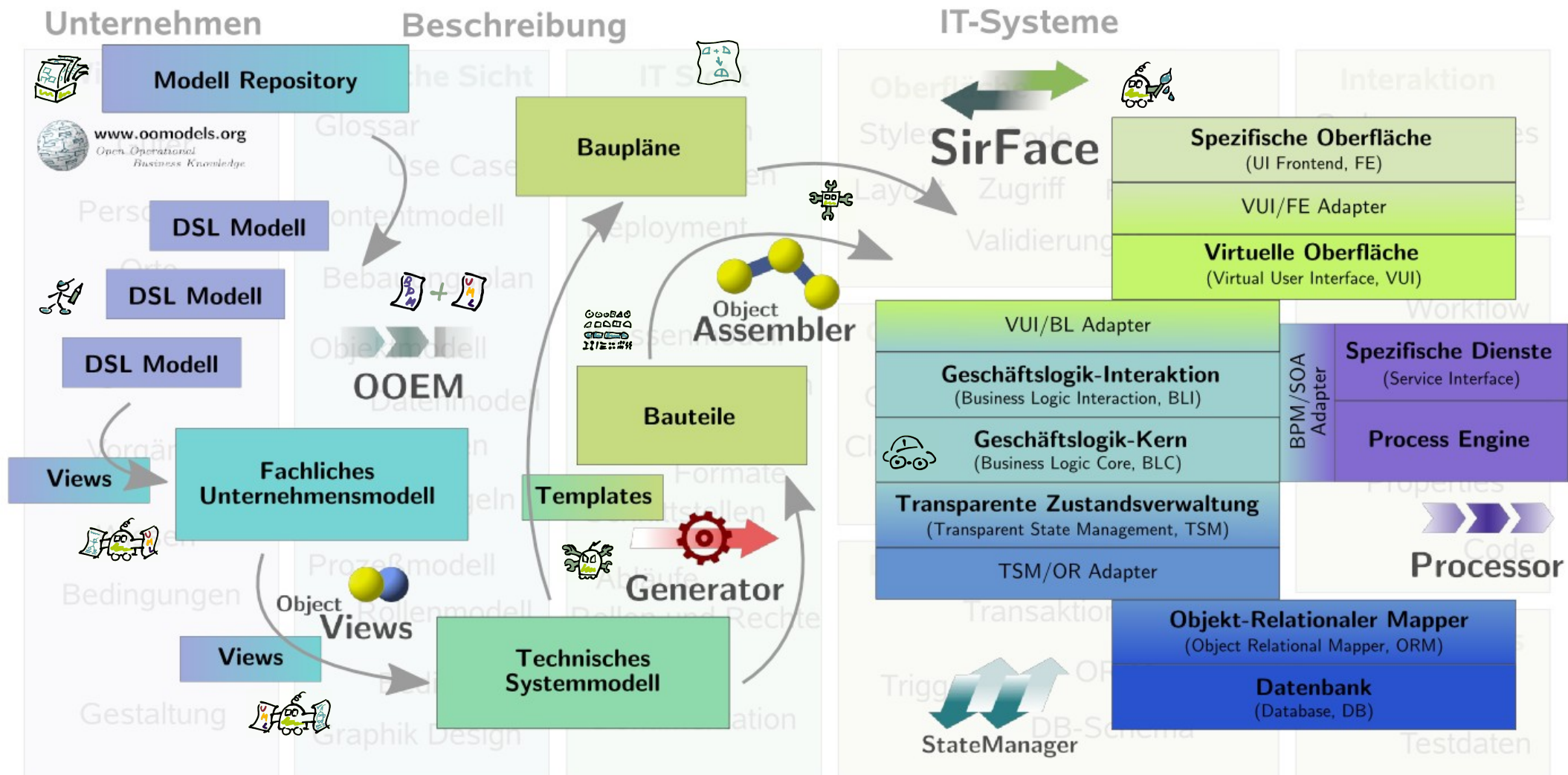
Service & Brand: fairyloom & businessboard
Business Model: OSS, Tool & Service

Technologie

EM/OS – Architektur & Innovationen



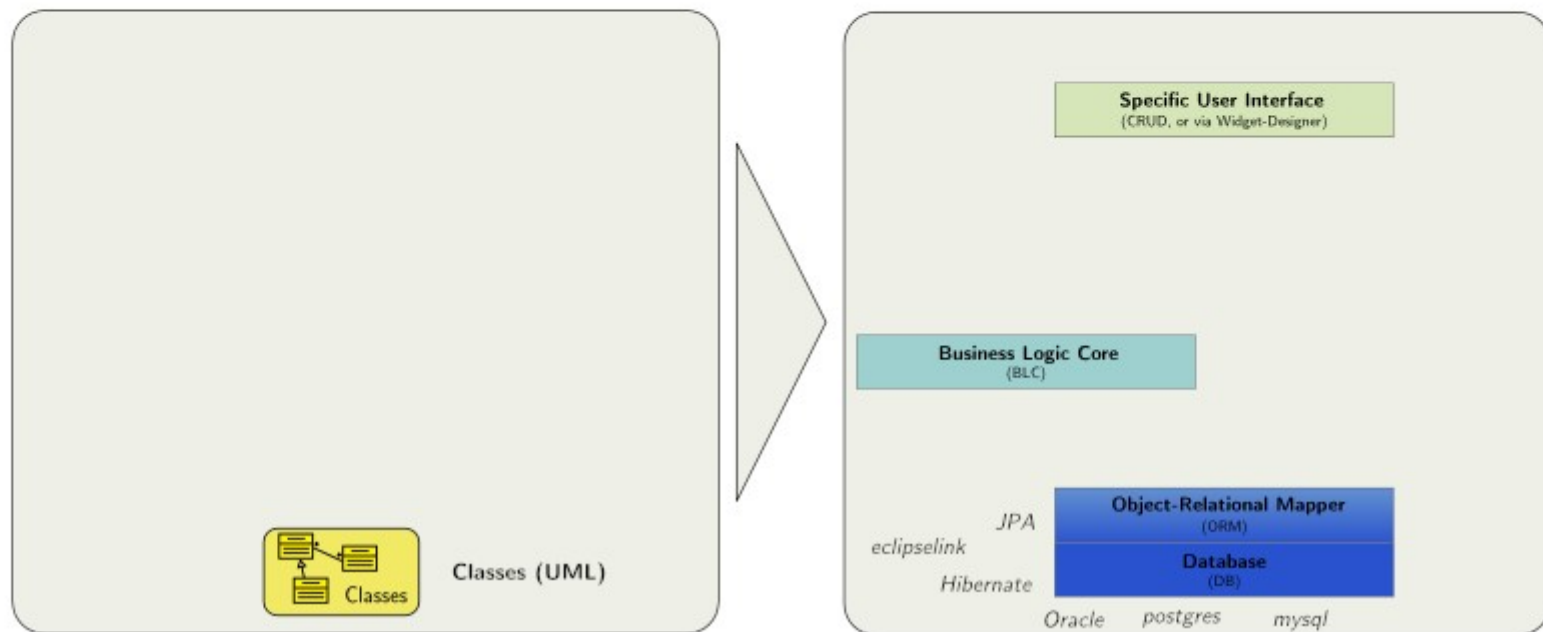
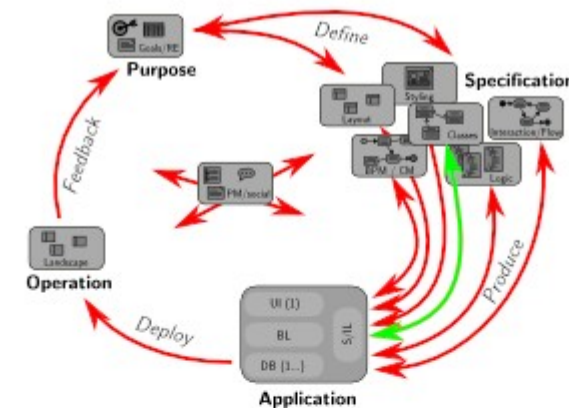
EM/OS Architektur



Model Driven - State of the Art

Core BL
DB Mappings
CRUD UI

sowie BL-/BPM-spezifische Uis via Widget Composer



True Model Driven User Interface

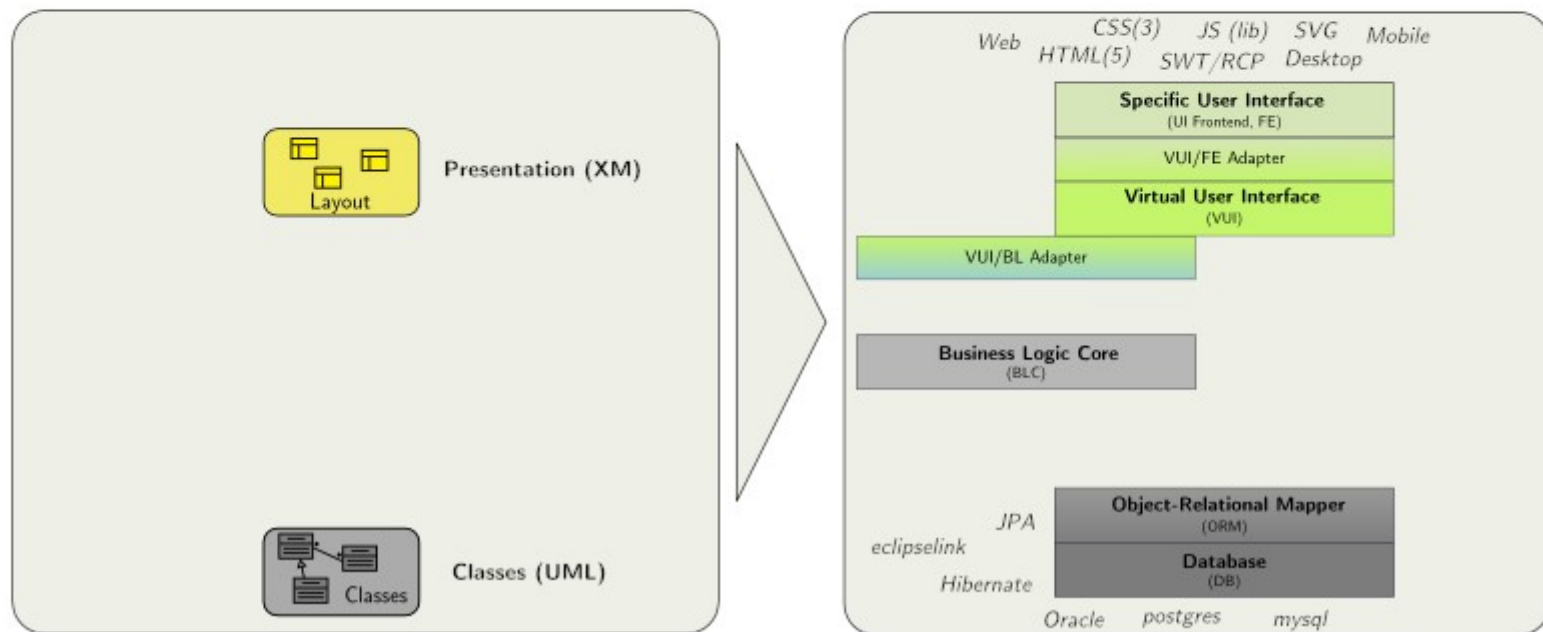
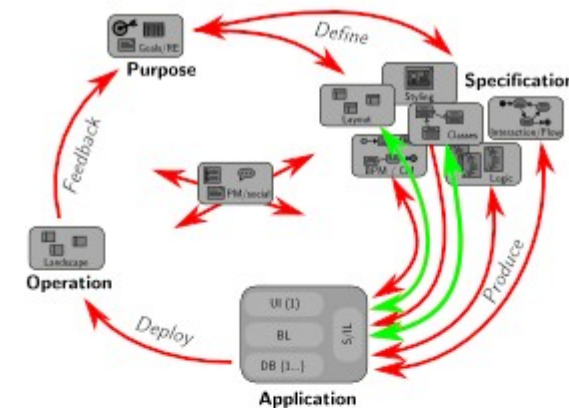
fachliche, also nicht-technische

Properties am Fachmodell

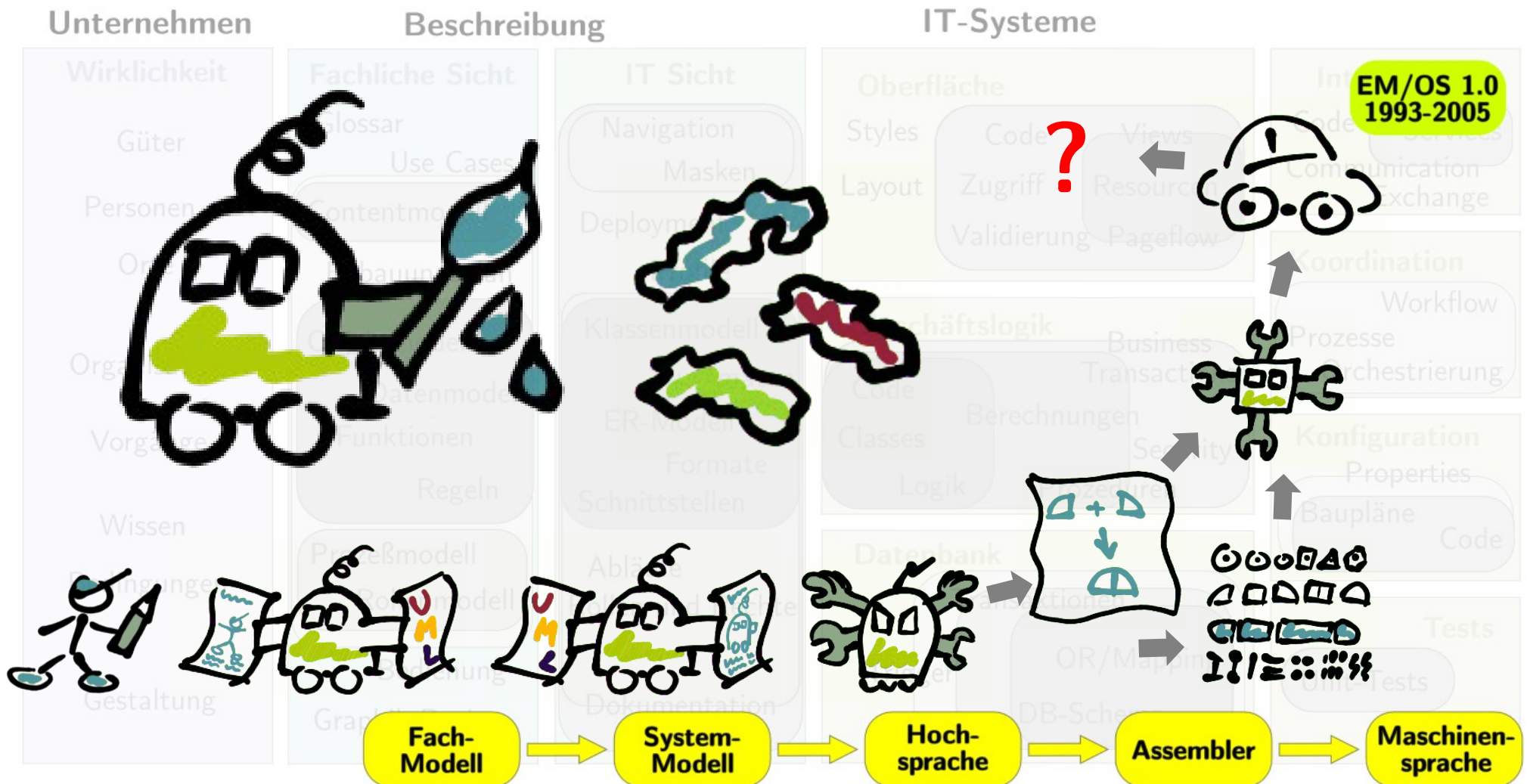
erzeugen medienunabhängiges

Virtual User Interface

das auf beliebige Frontends abgebildet wird

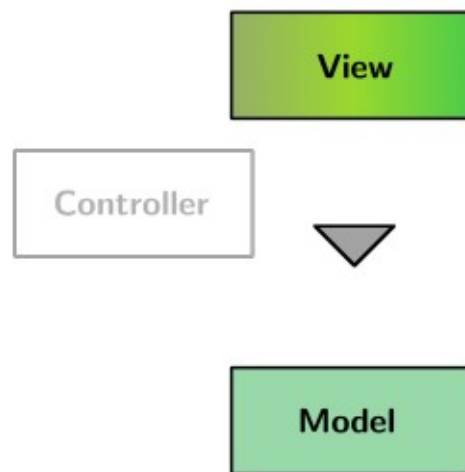


Virtual User Interface mit M3V Entwurfsmuster - www.m3v.org



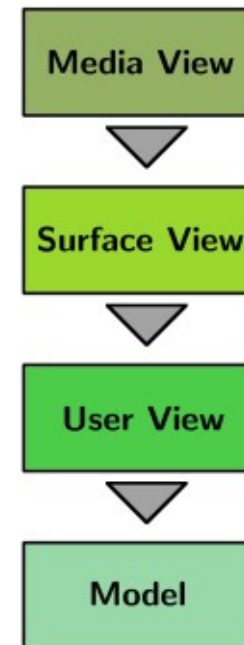
Virtual User Interface mit M3V Entwurfsmuster - www.m3v.org

M3V Muster – 1 Model, 3 Views



MVC

Model-View-Controller

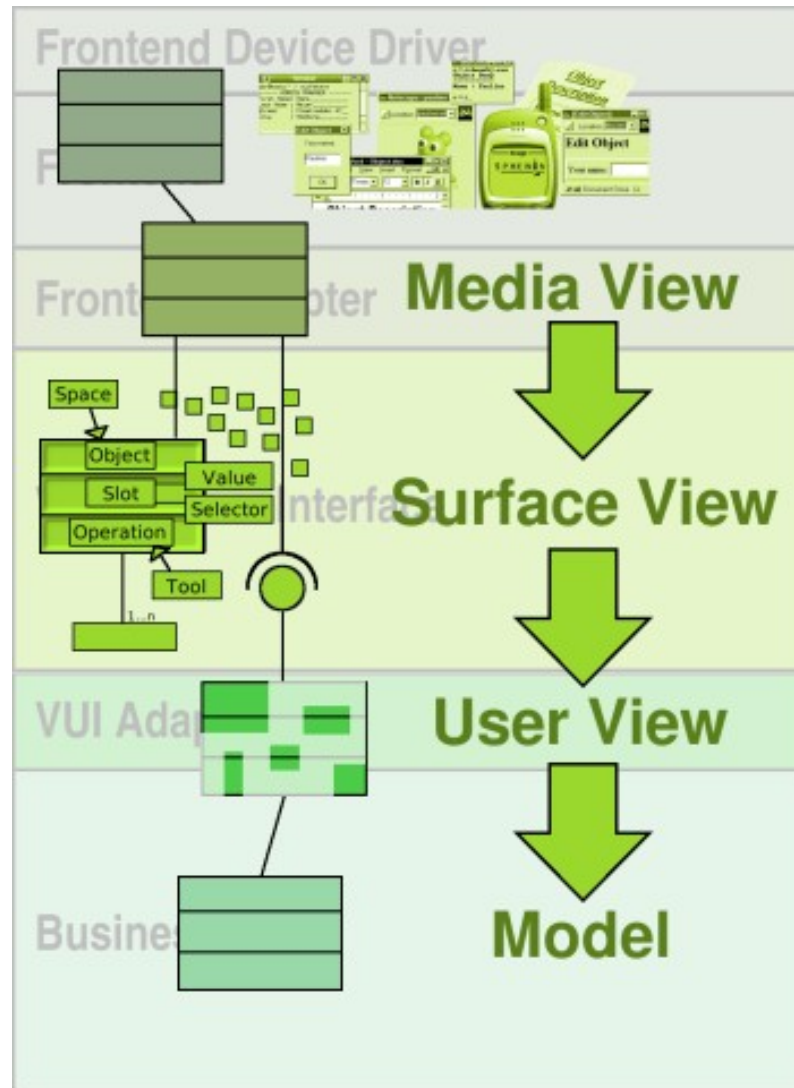


M3V

Model-View-View-View

Virtual User Interface mit M3V Entwurfsmuster - www.m3v.org

M3V Muster – 1 Model, 3 Views



Business Logic State Machines

UI-Interaction

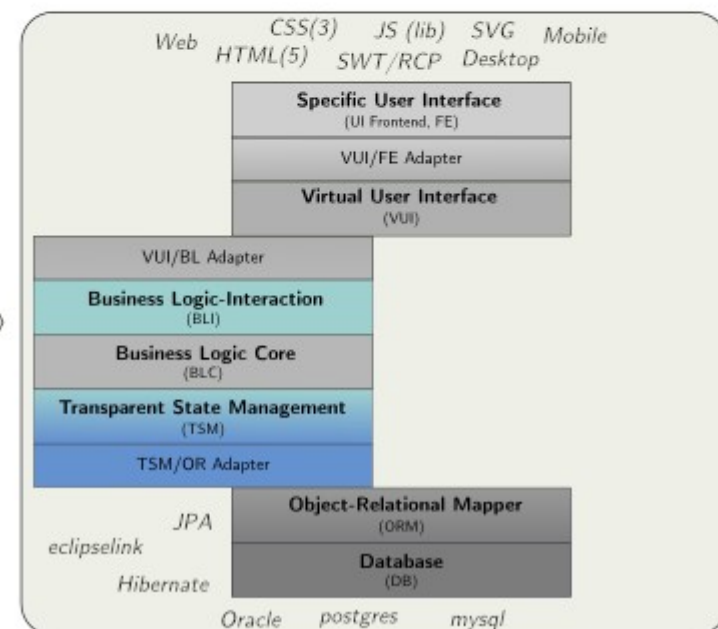
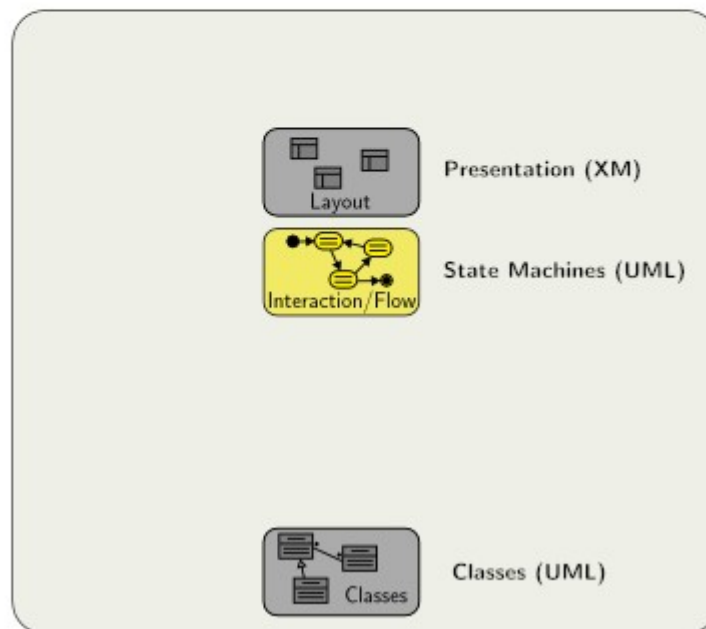
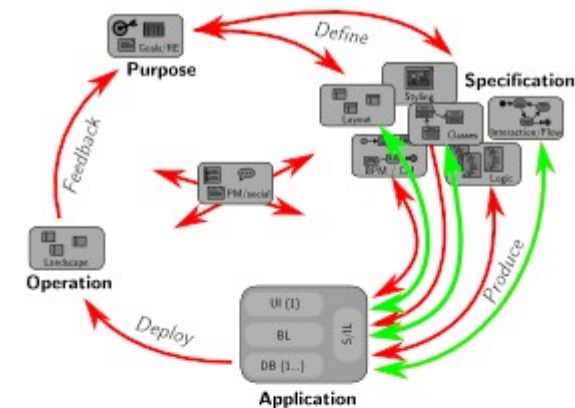
fachliche Micro-Processes = medienneutrale Flows

BL-Behaviour/States

konsistentes Verhalten und BPM-Verknüpfung

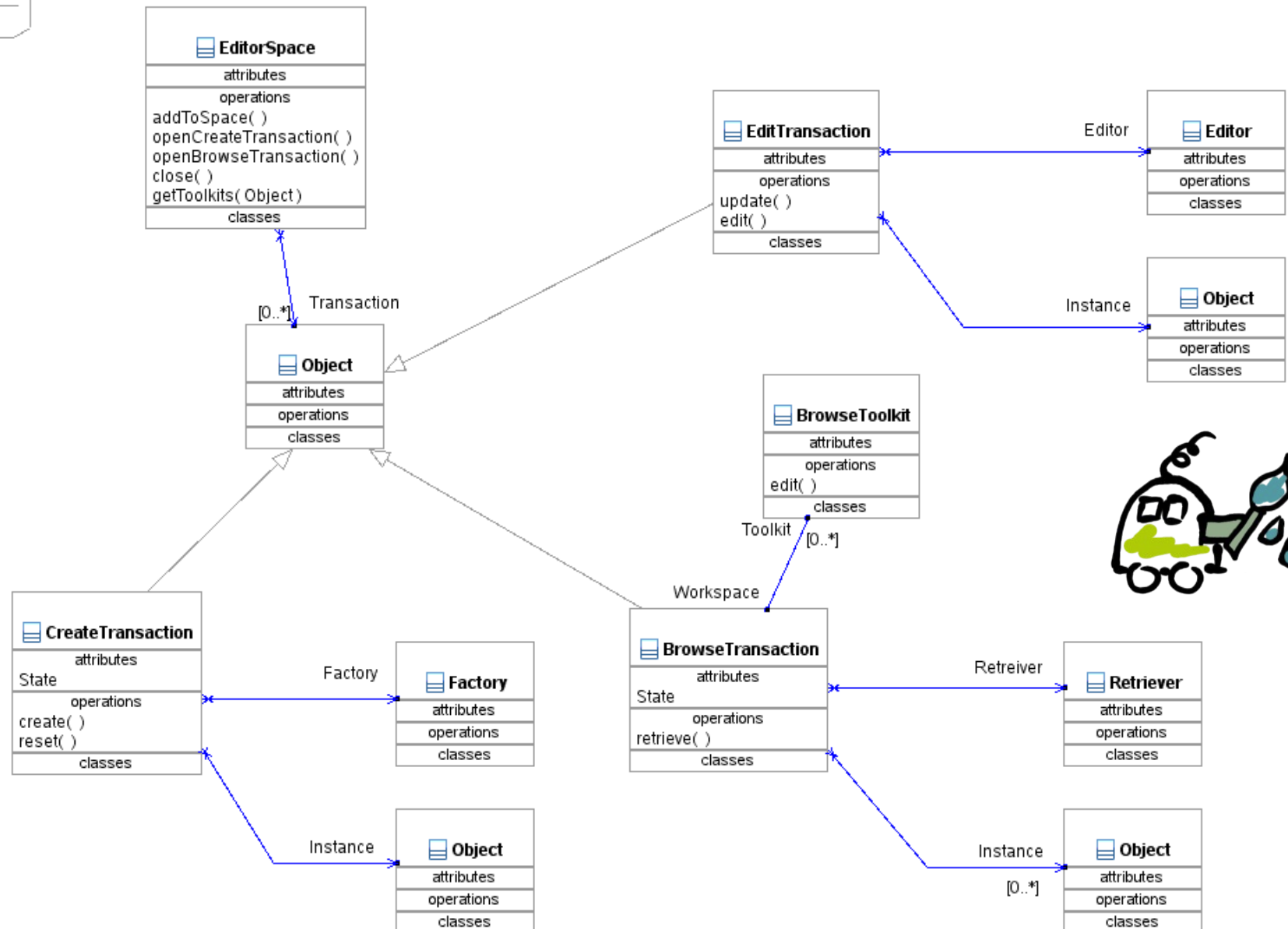
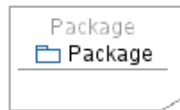
DB-Transaction Control

ideale Grundlage für Model Driven Approach



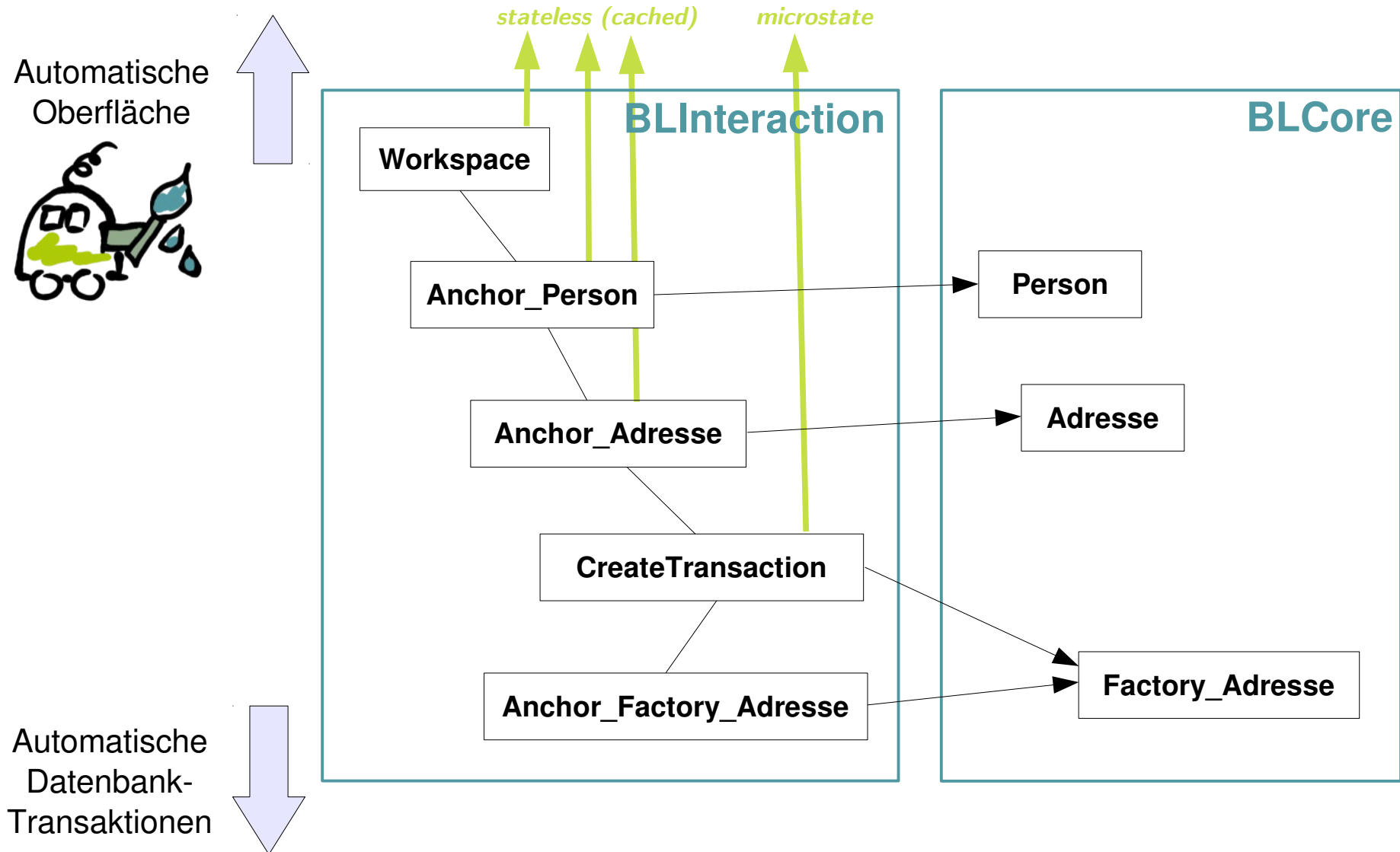
Virtual User Interface mit M3V Entwurfsmuster - www.m3v.org

Business Logic Interaction (BLI) - Objekte des Edit-Dialogs



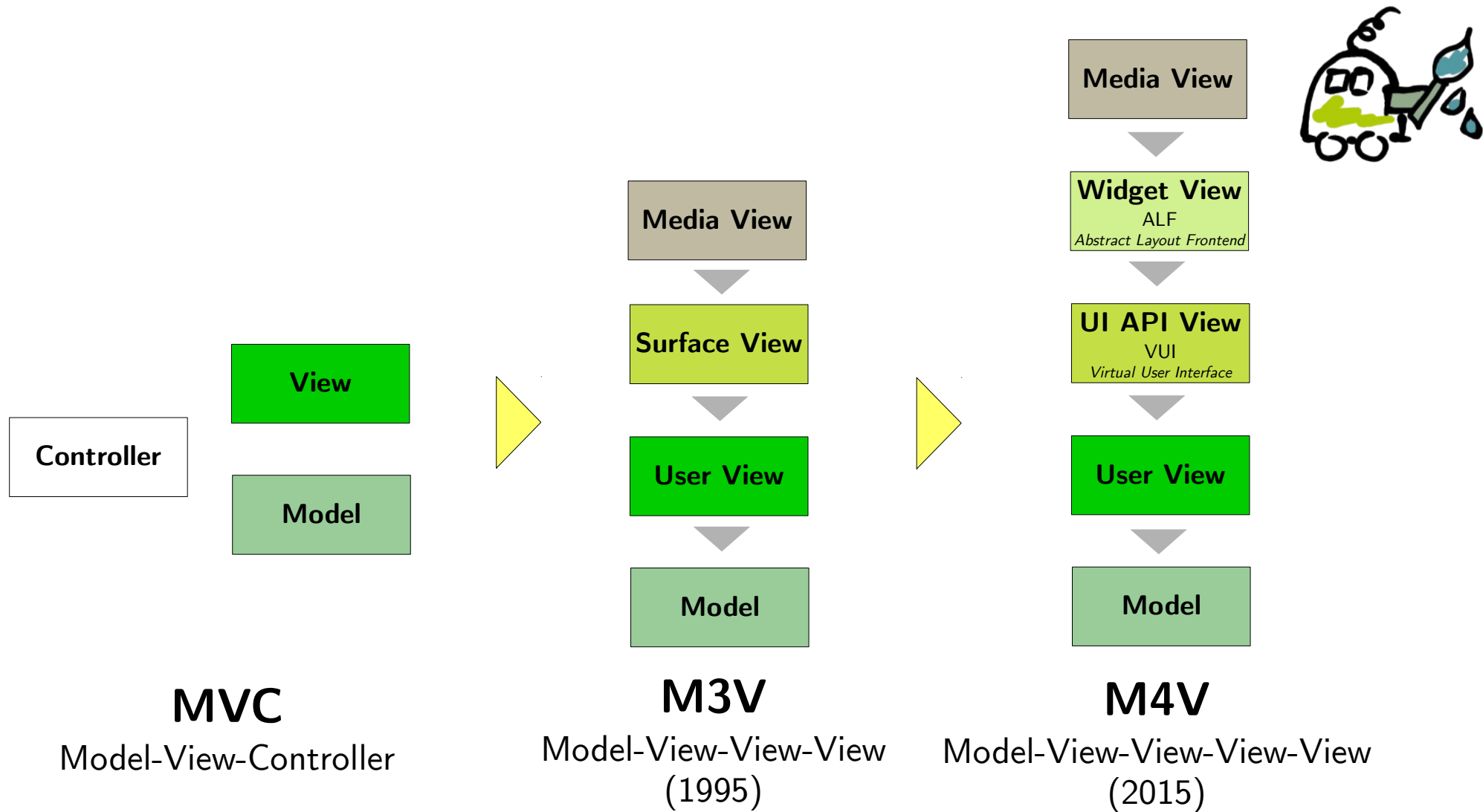
Virtual User Interface mit M3V Entwurfsmuster - www.m3v.org

Business Logic Interaction (BLI) - Arbeits- und Transaktionsgerüst



Von M3V zu M4V

M4V Muster – 1 Model, 4 Views



Renderer und Stylesheets

Unabhängig vom Model

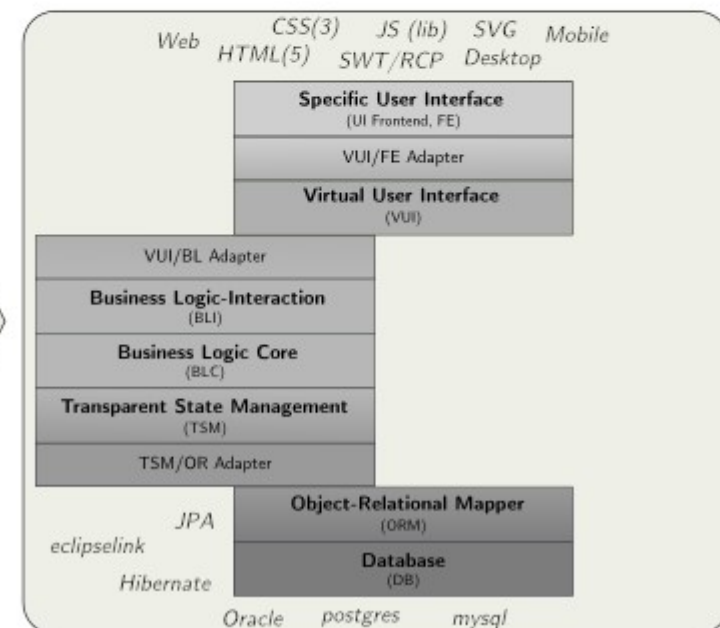
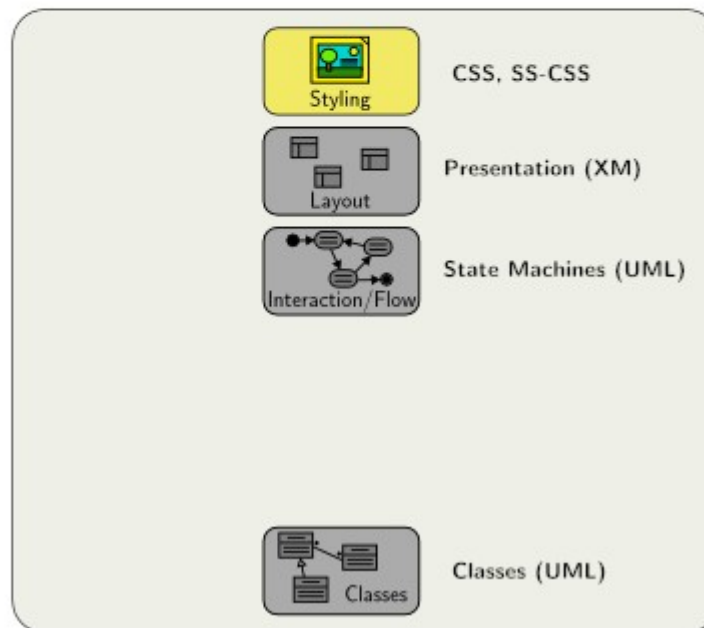
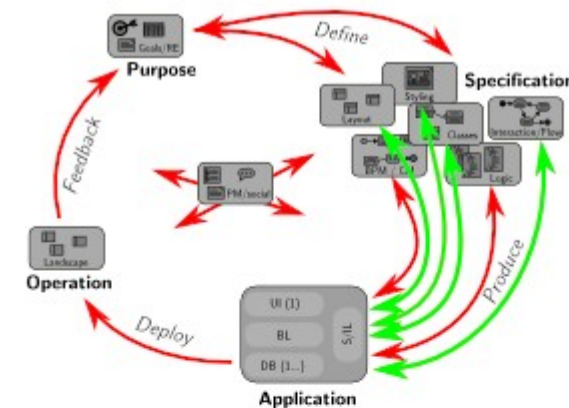
CD/CI-Styling anwendbar auf beliebige Projekte

Any-UI

Web, Desktop, Mobile

UI-Varianten

Branding, Devices, Preferences



Virtual User Interface mit M3V Entwurfsmuster - www.m3v.org

Unternehmen

Beschreibung

IT-Systeme

Wirklichkeit

Oberfläche und Funktion sind völlig (100%) getrennt

Anpassung an Frontend (Web, Desktop, Mobile usw.) zur Laufzeit

Fachliche Sicht

IT Sicht

Oberflächen

EM/OS 1.0
1993-2005



Standardkomponenten



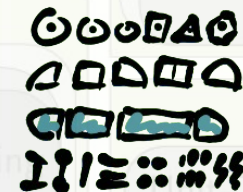
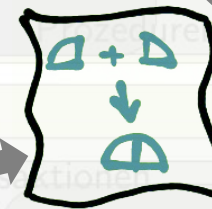
100% BL frei



VUI



100% UI frei



Fach-Modell

System-Modell

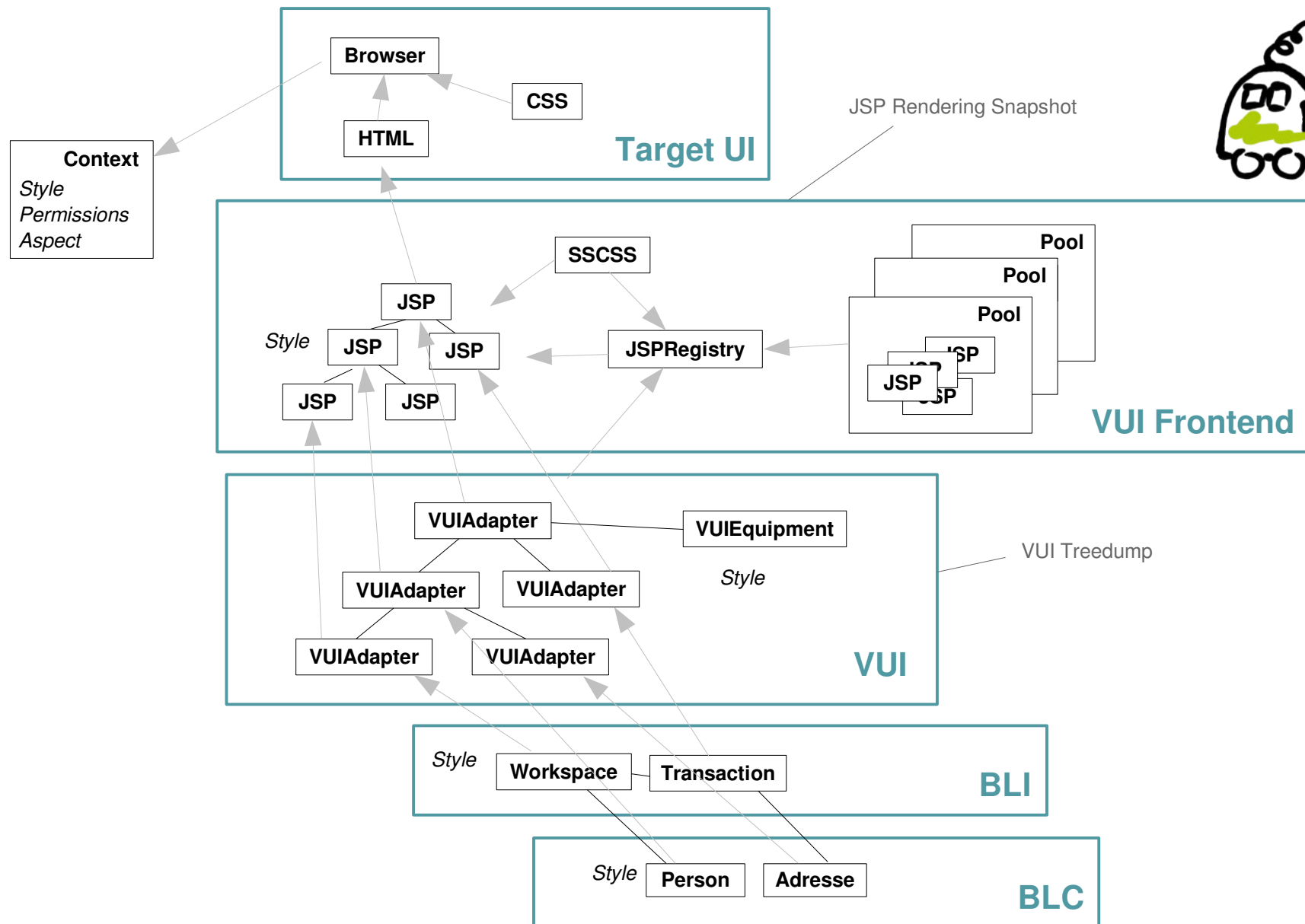
Hochsprache

Assembler

Maschinen-sprache

Virtual User Interface mit M3V Entwurfsmuster - www.m3v.org

Erzeugung der Oberfläche am Beispiel Web



Unified Process Modelling

Prozesse und Objekte verknüpft

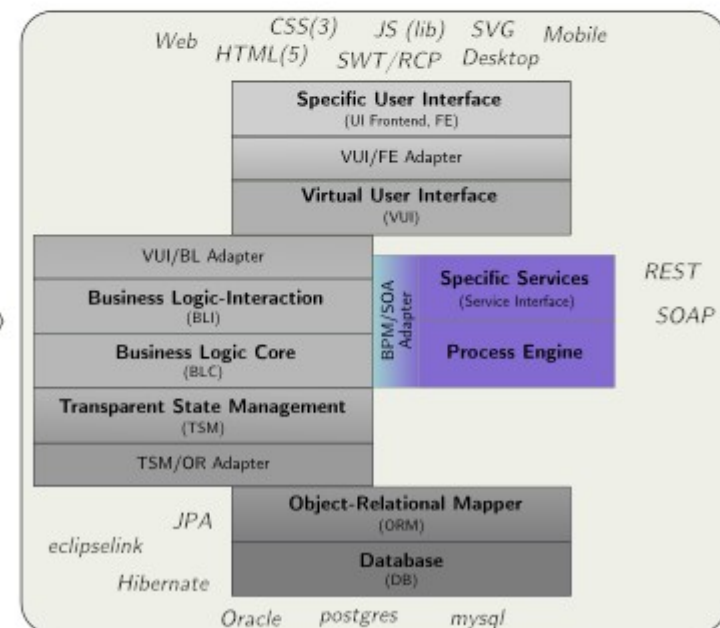
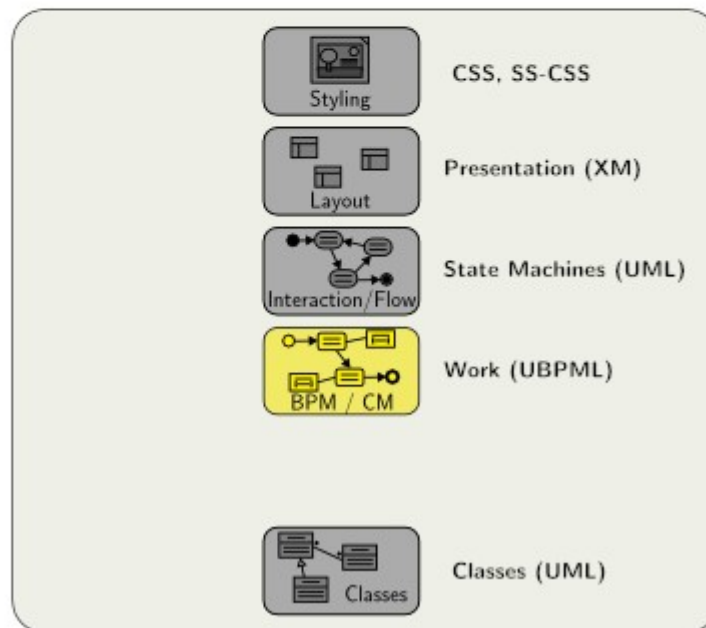
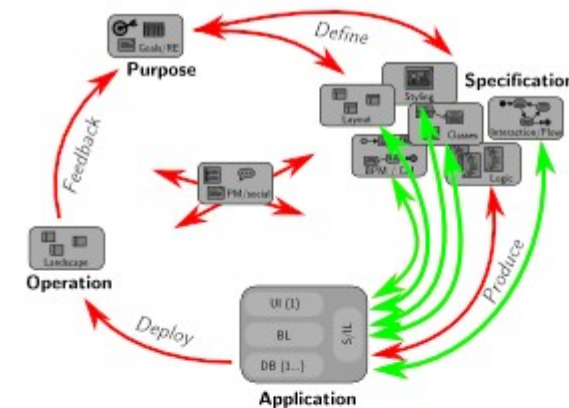
systematisch via States

Token-freie Workflows

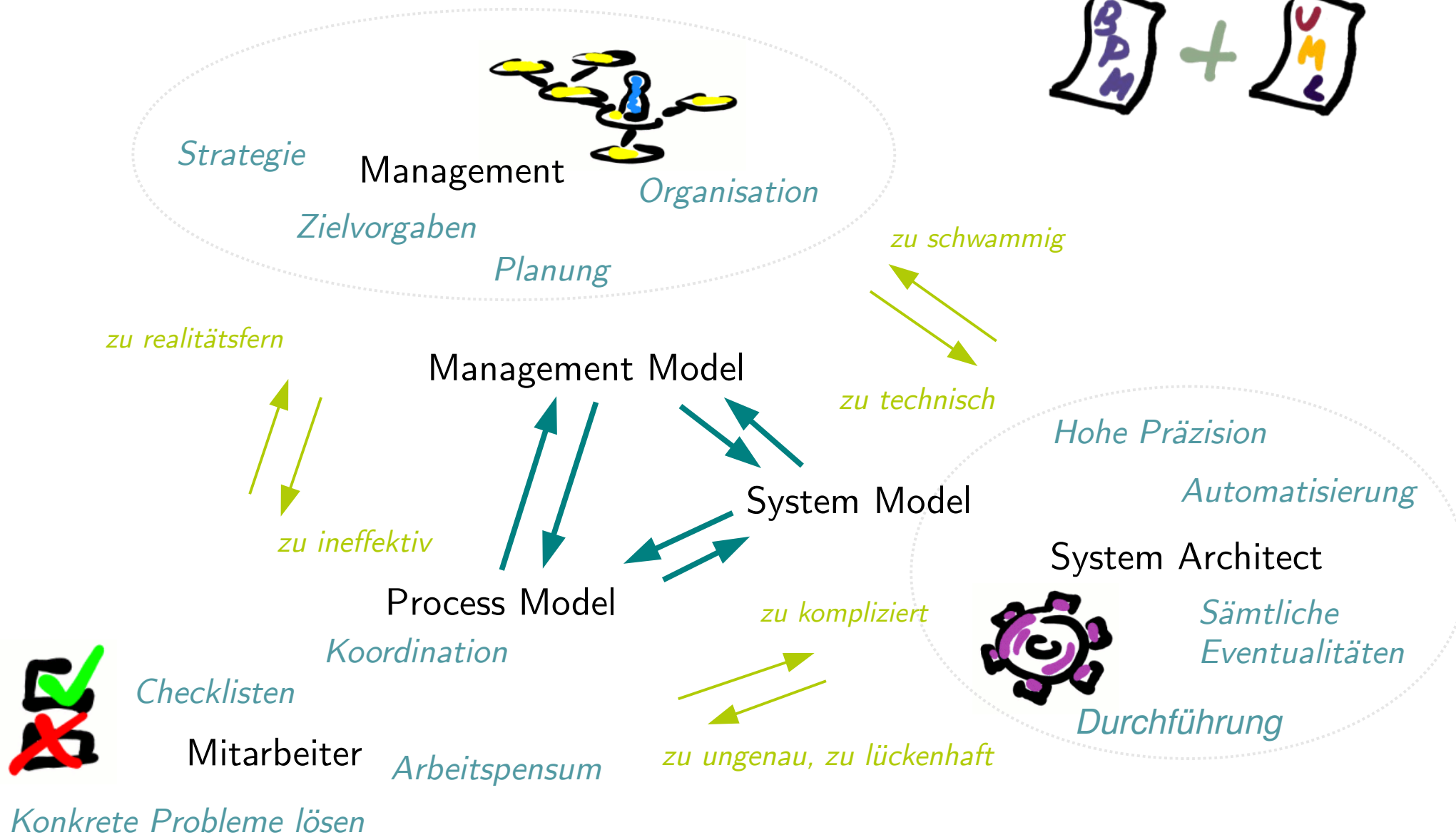
weniger störanfällig und vereint BPM und CMM

explizites Material

ermöglicht UI-Erzeugung



Unifizierte BPM/IT-Modelle - www.oem.org / www.ubpml.org



Unifizierte BPM/IT-Modelle - www.oem.org / www.ubpml.org



Klassen in's Modell

(Zustandsbehaftete Klassen und Objekte)

Weg mit dem Kontrollfluß

(Konstellationen als kausale Zusammenhänge)

Entkopplung der Modellebenen durch OO

(Steps als Objekte, Prozeduren assoziiert)



Unifizierte BPM/IT-Modelle - www.oem.org / www.ubpml.org



Management

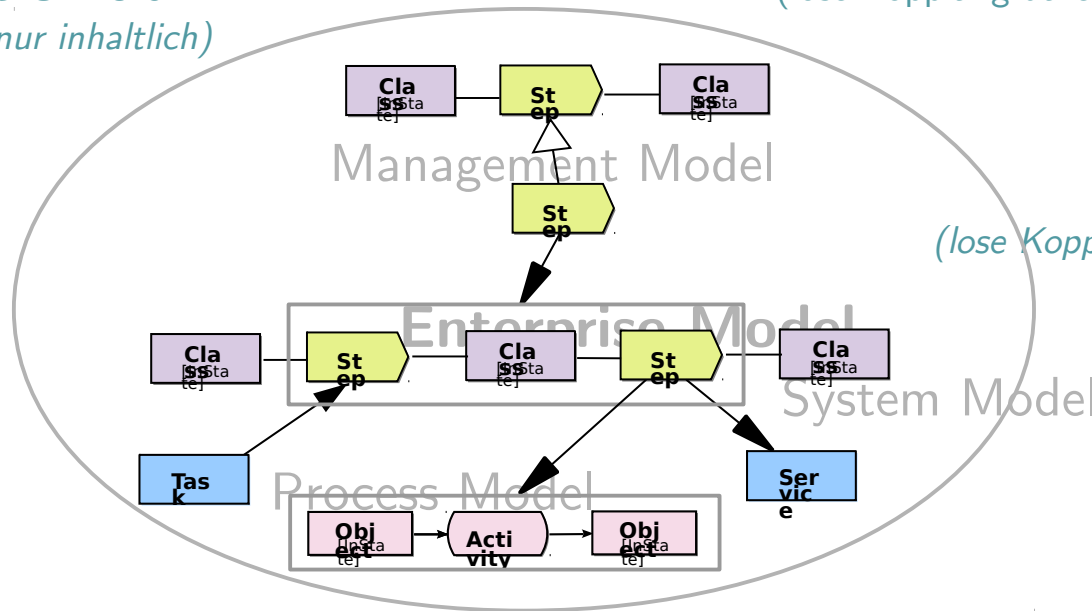


Vererbung

(lose Kopplung durch Abstraktion)

Kein Kontrollfluß

(lose Kopplung: nur inhaltlich)



Verfeinerung

(lose Kopplung durch dynamische Zuordnung)

System Architect



IT Services

(lose Kopplung durch dynamische Zuordnung)



Mitarbeiter

Offen für Erweiterungen

(lose Kopplung durch Prozedur-Abstraktion)

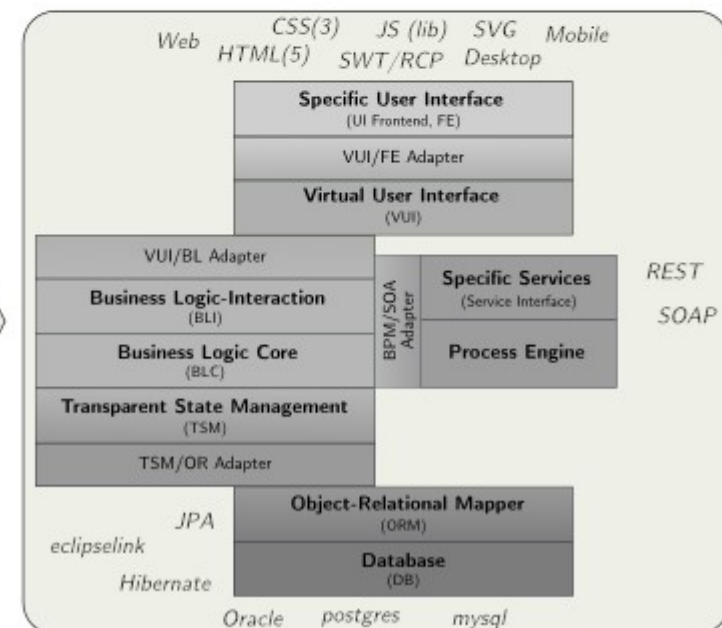
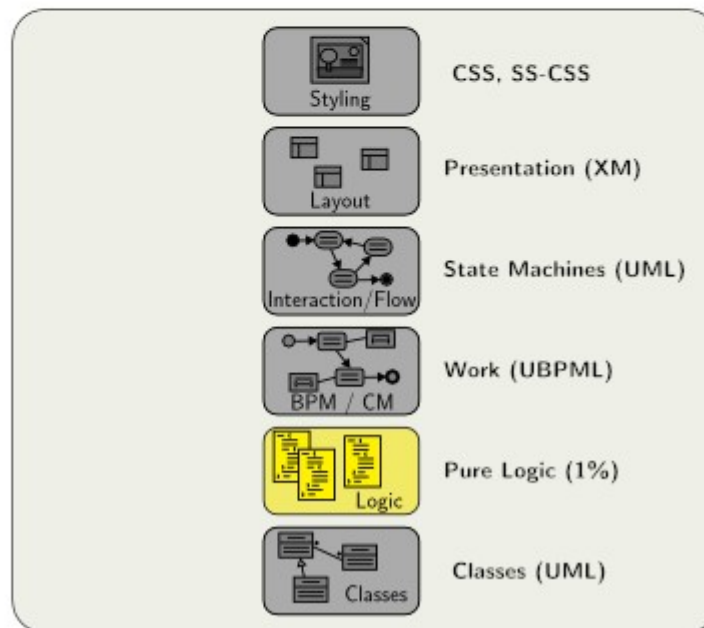
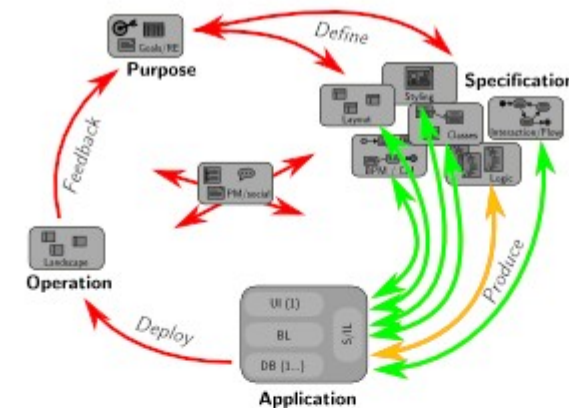
1% fachliche Logik

keinerlei technischer Code

kein Glue-Code, kein TX-Code, kein DB-Code
keine UI-Callbacks, kein Init/Exit/Config/Setup

Sprach-Plugins

derzeit vor allem Java, offenes Konzept



Kollaboration & Planung

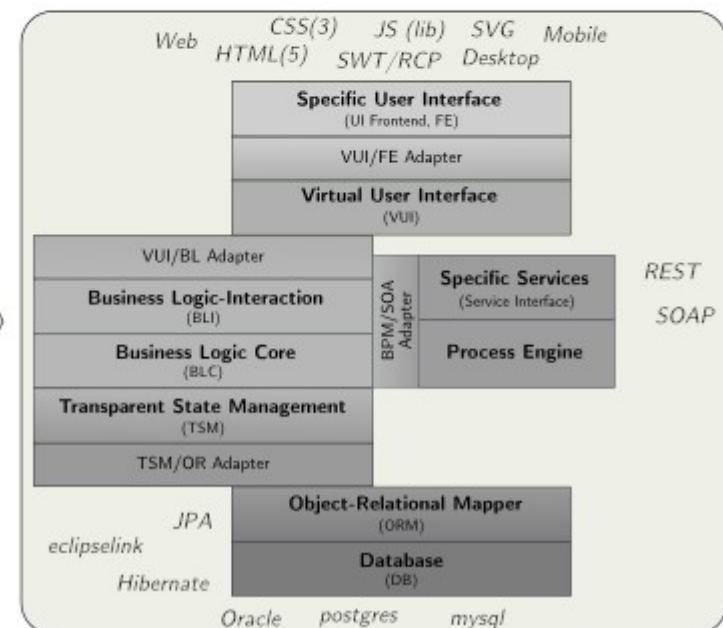
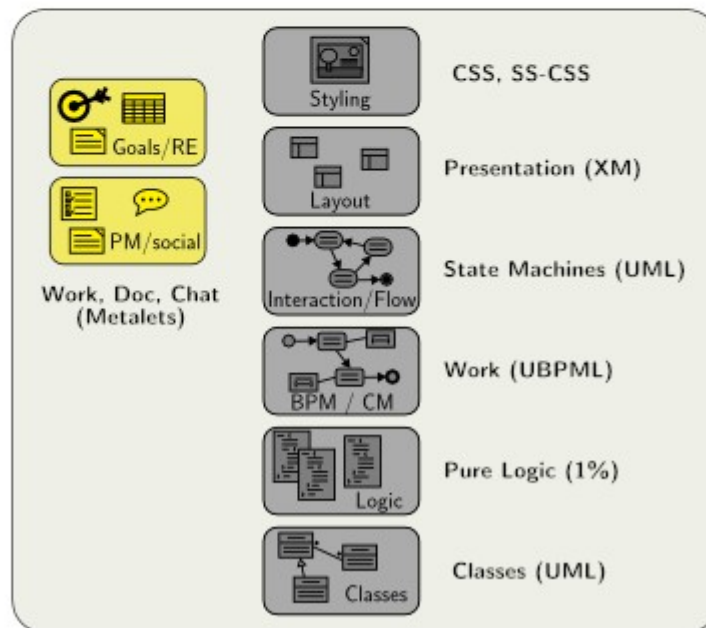
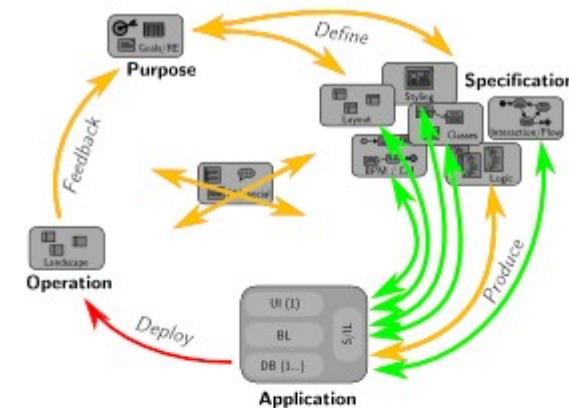
an jedem Modell-Element

jederzeit zur Hand, verständlich

integriertes Project Management

verschiedene Verfahren: klassisch oder agil

keine Redundanz



Landscape-Maps

für Deployment

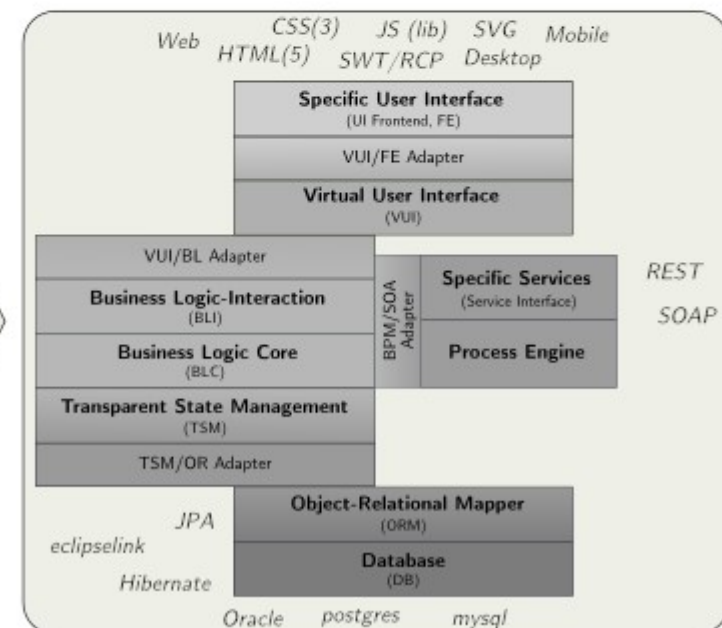
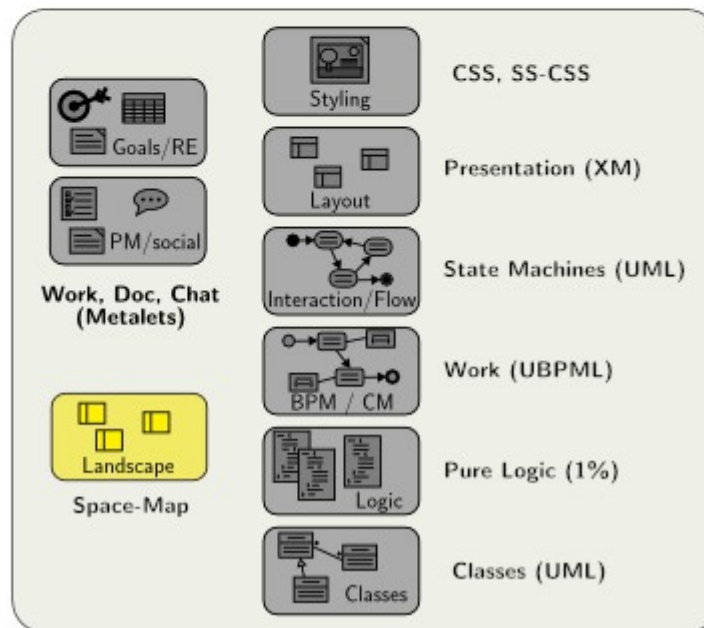
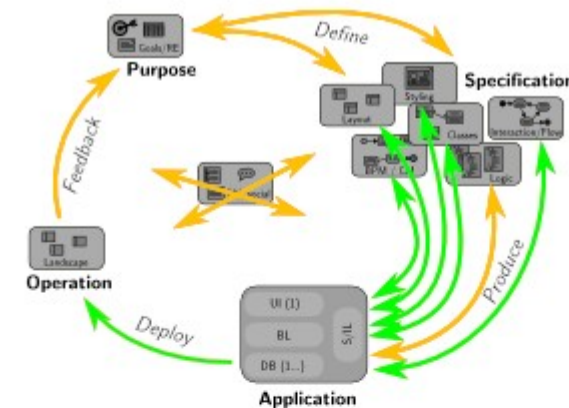
automatisch via business board

für Backup

Erzeugung von bspw. scripten

für Monitoring

Erzeugung von bspw. nagios-Konfiguration



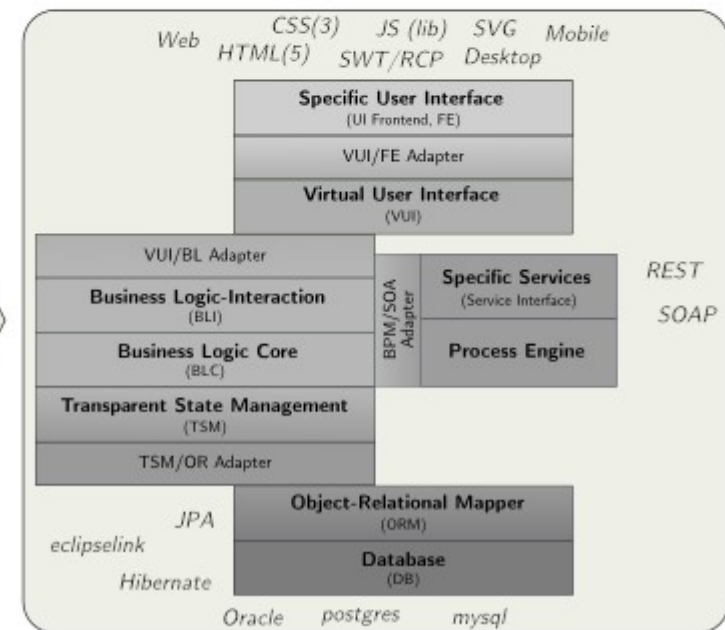
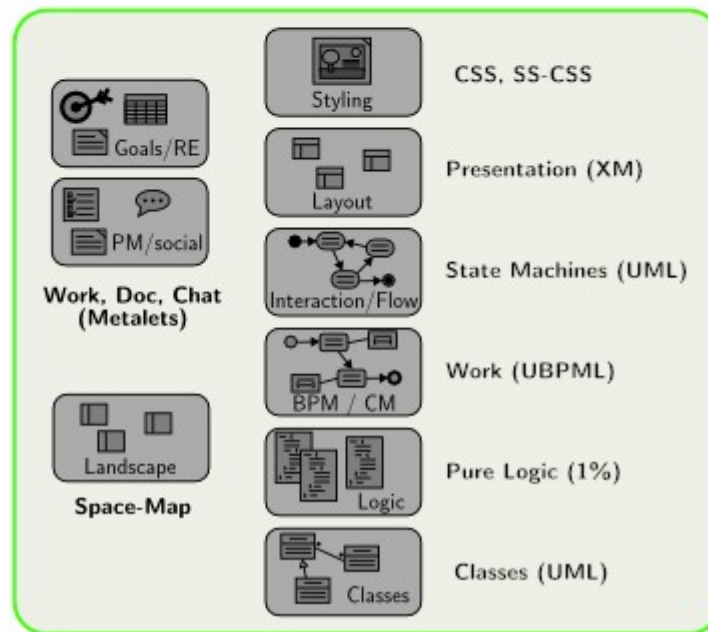
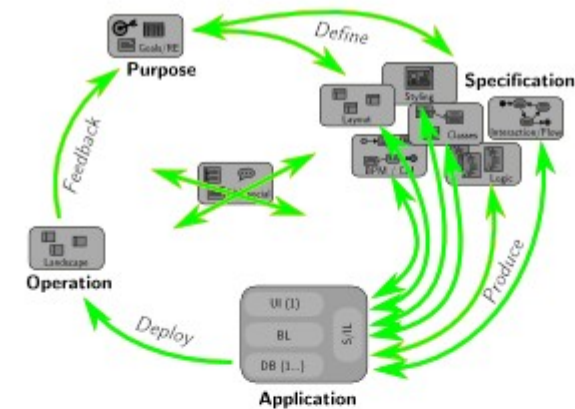
Hohe Qualität des Modells

vollständig

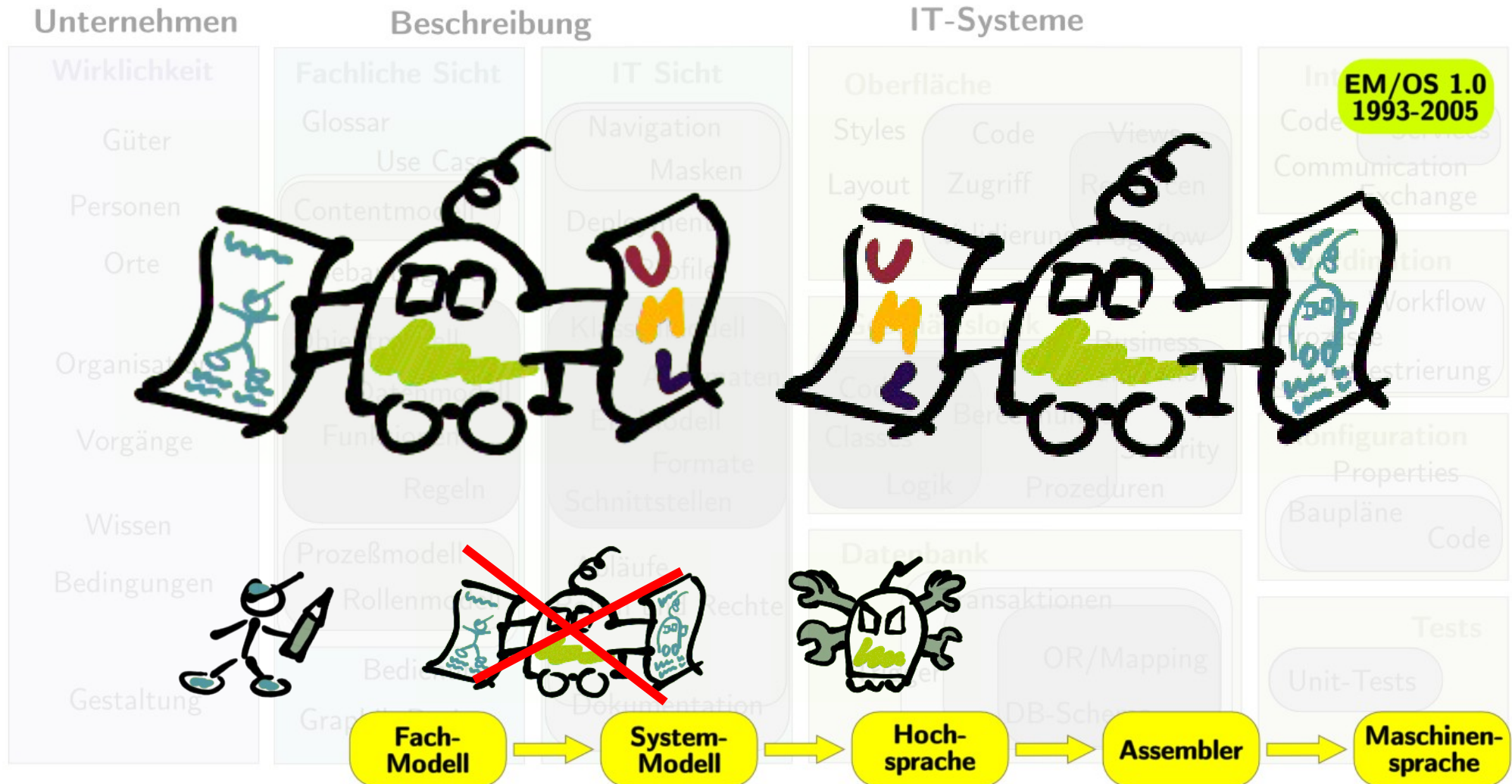
100% spezifiziertes System

verständlich

lesbar und veränderbar für sowohl Fach-
als auch IT-Experten, guter Abstraktionsgrad



Mehrstufige dynamische Modelltransformation



Mehrstufige dynamische Modelltransformation

Völlige Entkopplung
fachlich & technisch

effektiv wieder-
verwendbare Modelle

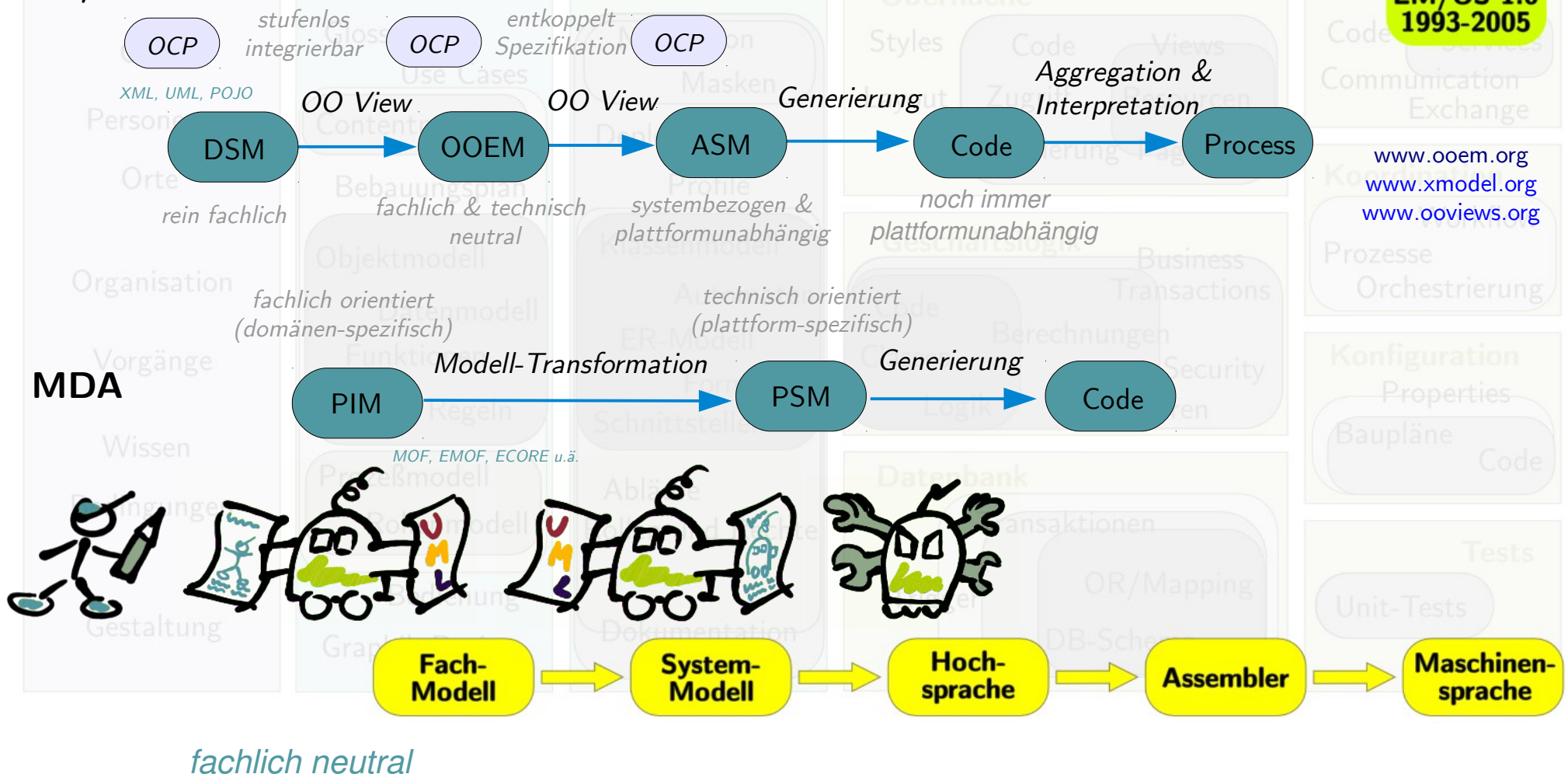
handliche Optionen für
DSLs u.a.: DSM & OCP-
Macros

Überschaubare Themen
auf jeder Stufe

Unternehmen

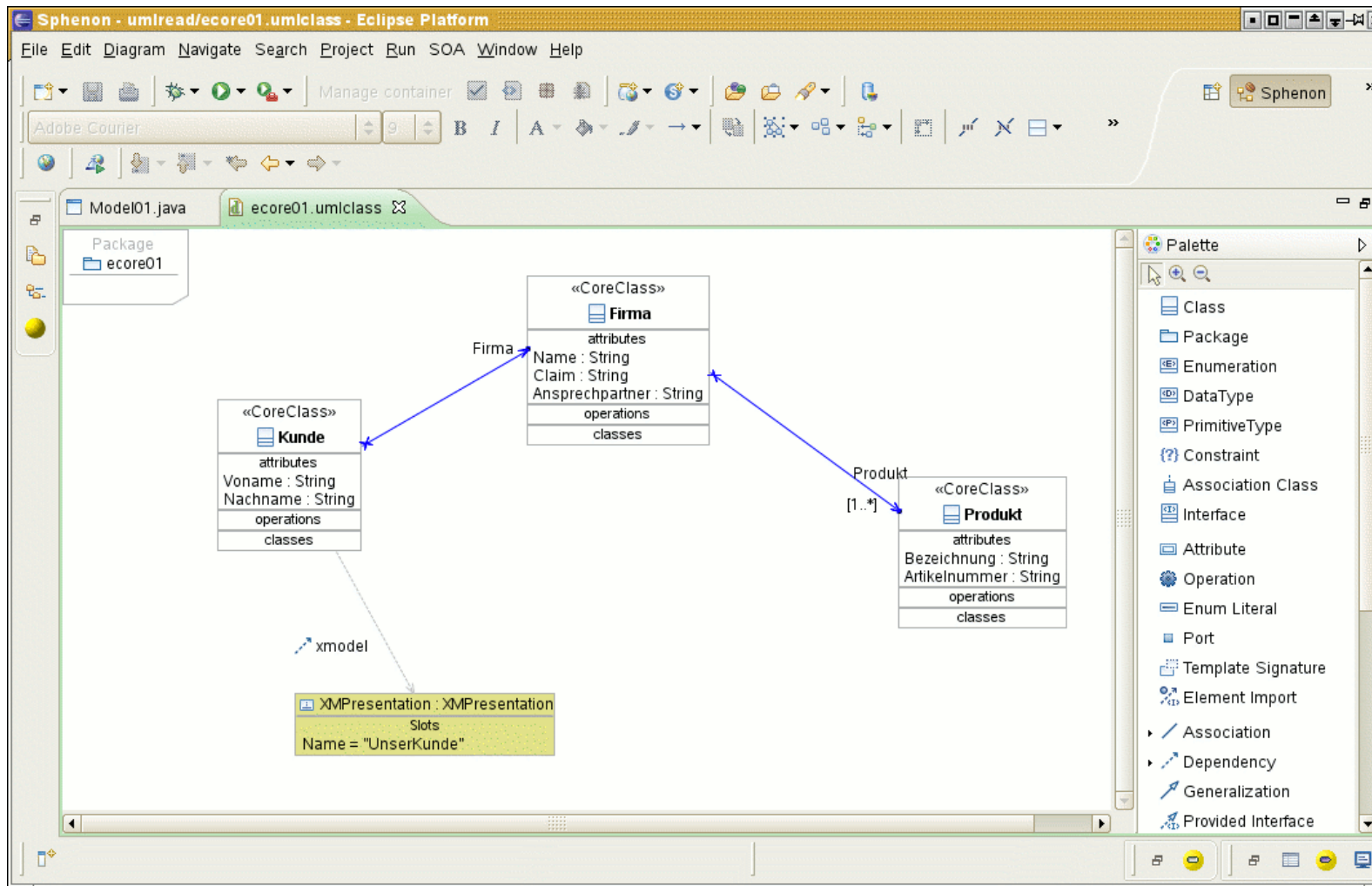
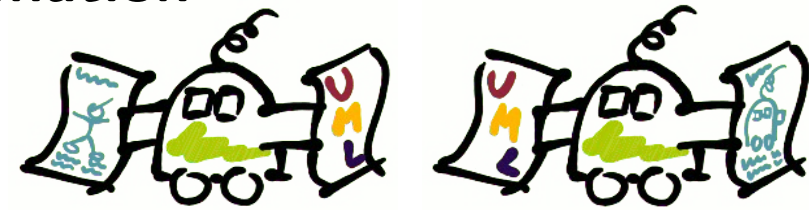
Beschreibung

EM/OS



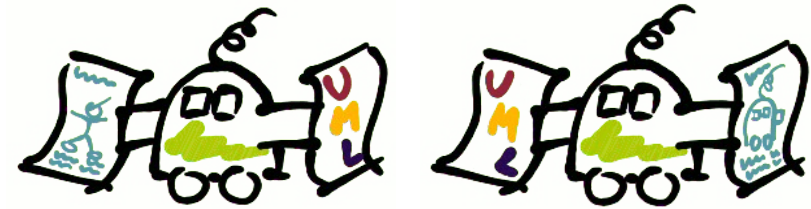
Mehrstufige dynamische Modelltransformation

Eclipse UML 2 Editor



Mehrstufige dynamische Modelltransformation

OOEM 2.0 (XML, mit OCP gelesen)



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<Class>
```

```
  <StereotypeModelImport>org.uml.stereotypes.*</StereotypeModelImport>
```

```
  <ModelImport>org.oomodels.business.foundation.entities.*</ModelImport>
```

```
  <Name>Person</Name>
```

```
  <Base Parent="Entity"/>
```

```
  <Stereotype>CoreClass</Stereotype>
```

```
  <Attribute Name="Name" Type="NameOfPerson"/>
```

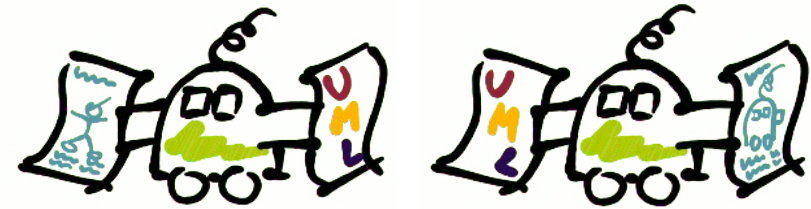
```
  <Attribute Name="Sex" Type="Sex"/>
```

```
  <Attribute Name="Birthday" Type="Date"/>
```

```
</Class>
```

Mehrstufige dynamische Modelltransformation

WIML – Wiki Modeling Language



```
<wiml>
```

```
* Person CoreClass
```

```
> Entity
```

```
A human person
```

```
** Name NameOfPerson
```

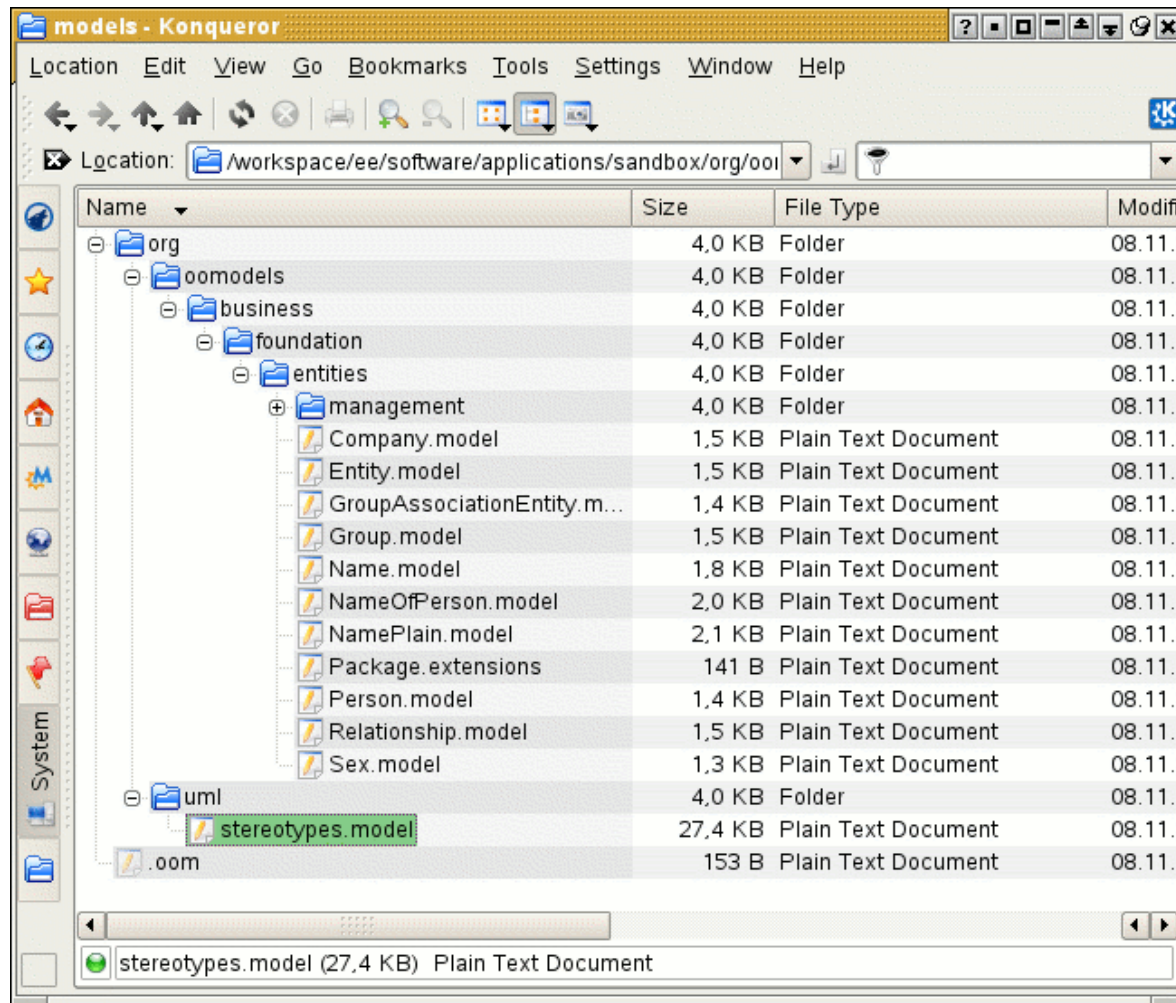
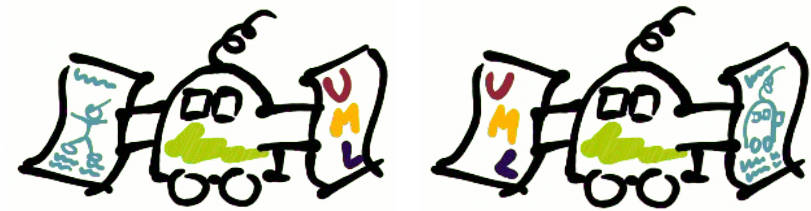
```
** Sex Sex
```

```
** Birthday Date
```

```
</wiml>
```

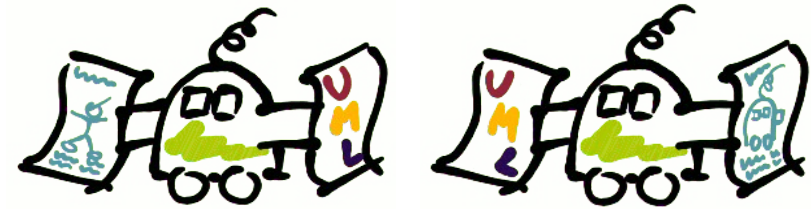
Mehrstufige dynamische Modelltransformation

Teil-Modell-Artefakte im Dateisystem



Mehrstufige dynamische Modelltransformation

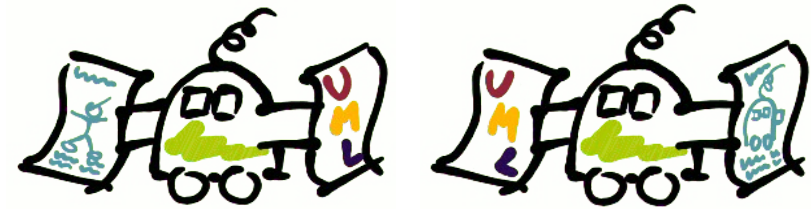
M2M mit OO View mit OCP



```
<?xml version="1.0" encoding="UTF-8"?>
<UMLClass
  SIGNATURE="UMLClass Source, ADCModelOptions Options ..."
  POLYMORPHIC="Class" DEFINE="..." xmlns="...">
  <ModellImport
    FOREACH="m_import : ::*'Source/ModellImports'"
    EXPRESSION="true">m_import</ModellImport>
  <ArtefactOrigin><Track>
    <o1 OID="Origin">com/sphenon/ad/adcore/model/coremodel/UMLClass_Interface</o1>
    <o2 EXPRESSION="true">Source.getPath(context)</o2>
  </Track></ArtefactOrigin>
  <Base FOREACH="base : Source.getBases(context)"
    IF="var ifc = ::[base]'//XModel/Generalization/Interface!; ..."
    >
    <Parent EXPRESSION="true">jspp: prefix + ::[base]'//Property/Parent/Id'</Parent>
  </Base>
  <Name OID="Name" EXPRESSION="true">prefix + Source.getName(context)</Name>
  <Attributes...
```

Mehrstufige dynamische Modelltransformation

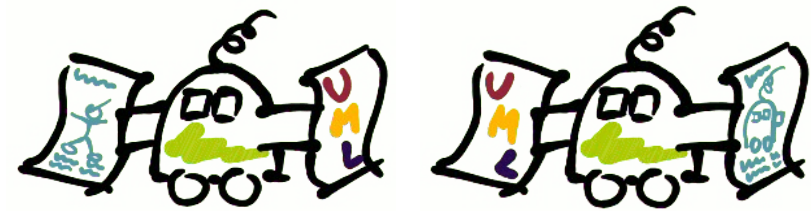
Vermischte Modelle



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
<Package CLASS="Aggregate::com/sphenon/tools/emfumloov/UMLPackage_EMF">  
  <Resource>/DATA/work/eclipse34_ws/umlread/ecore01.uml</Resource>  
</Package>
```

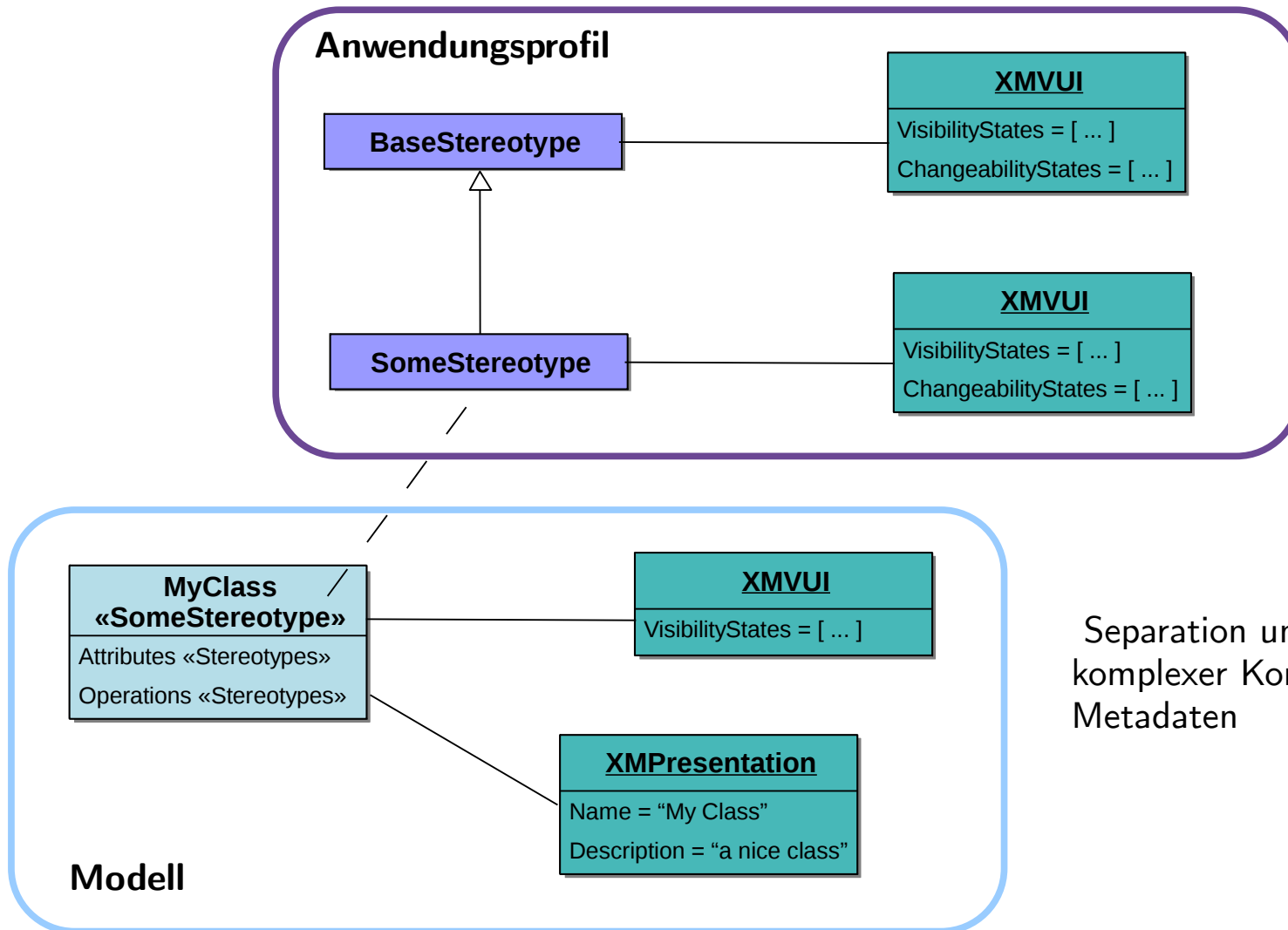
Mehrstufige dynamische Modelltransformation

OOEM View auf Ecore UML



```
<?xml version="1.0" encoding="UTF-8"?>
<UMLPackage
  SIGNATURE="String Resource"
  CLASS="UMLPackage"
  DEFINE="pkg : new Packages.com.sphenon.tools.emfumloov.ECoreAccessor(context, Resource).getPackage(context)"
  xmlns="http://xmlns.sphenon.com/com/sphenon/ad/adcore/model" xmlns:code="code">
  <InternalModelImport>org.uml.stereotypes.*</InternalModelImport>
  <ExternalImport>java.lang.String</ExternalImport>
  <Name EXPRESSION="true">pkg.getName(context)</Name>
  <ArtefactOrigin><Track><o1 EXPRESSION="true">'EMF Resource ' + Resource</o1></Track></ArtefactOrigin>
  <Classes>
    <Class FOREACH="cls : pkg.getClasses(context)">
      <Stereotype FOREACH="sttp : cls.getStereotypes(context)" EXPRESSION="true">sttp.getName(context)</Stereotype>
      <Name EXPRESSION="true">cls.getName(context)</Name>
      <Attributes>
        <Attribute FOREACH="att : cls.getAttributes(context)">
          <Name EXPRESSION="true">att.getName(context)</Name>
          <Type EXPRESSION="true">att.getType(context)</Type>
          <Extensions>
            <XModelExtension FOREACH="xme : att.getXModelExtensions(context)"
              FACTORY="GenericFactory_XModelElement">
              <Class EXPRESSION="true">xme.getClassifiers(context).get(0).getName(context)</Class>
              <Properties EXPRESSION="value">xme.getProperties(context)</Properties>
            </XModelExtension>
```

Feinspezifikation mit Anwendungsprofilen organisieren



Separation und Organisation
komplexer Konfigurationen und
Metadaten

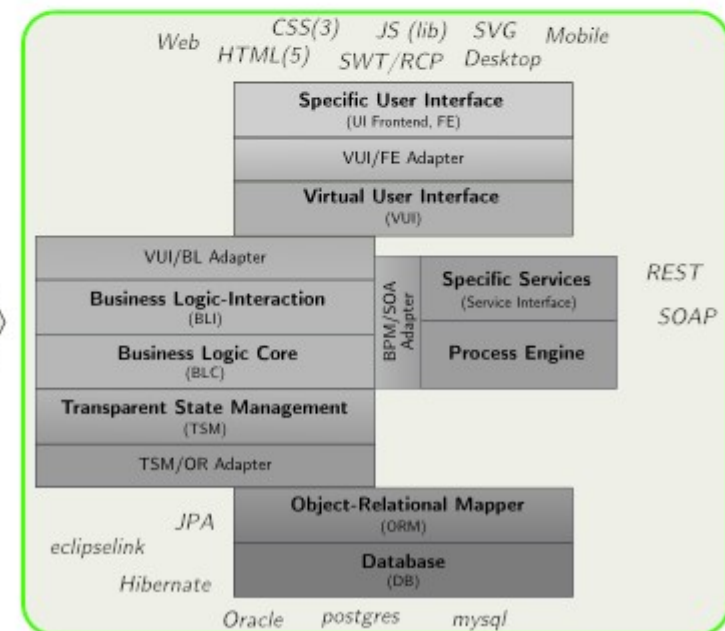
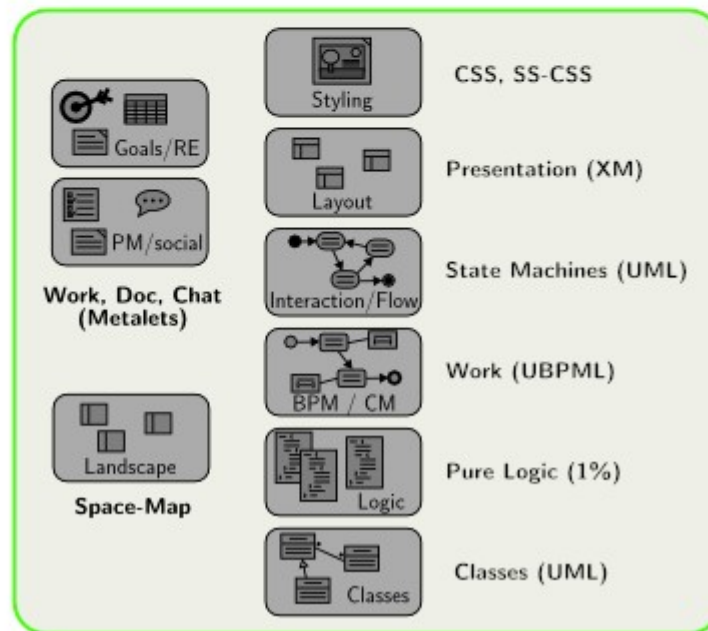
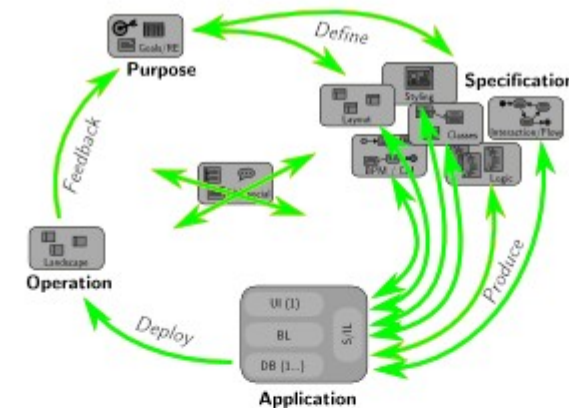
Hohe **Qualität** des Ziel-Systems

DRY, SoC und fein-modular

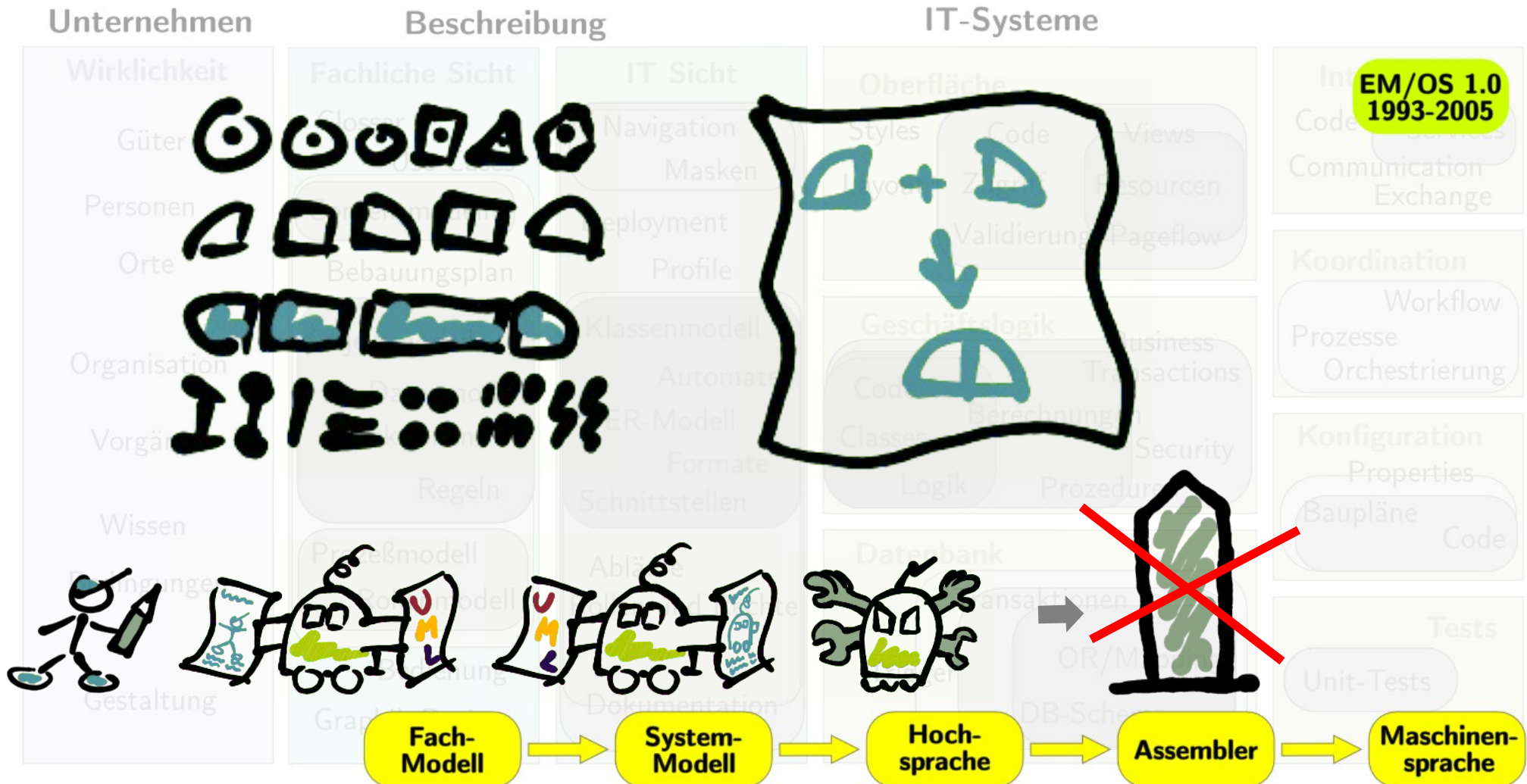
flexibel und teilweise verwendbar und ersetzbar

kontextsensitiv zur Laufzeit

automatische Anpassung an Services der Umgebung (UI, DB, APIs)



Roboter baut keinen Monolithen, sondern Bauteile und Baupläne



Roboter baut keinen Monolithen, sondern Bauteile und Baupläne

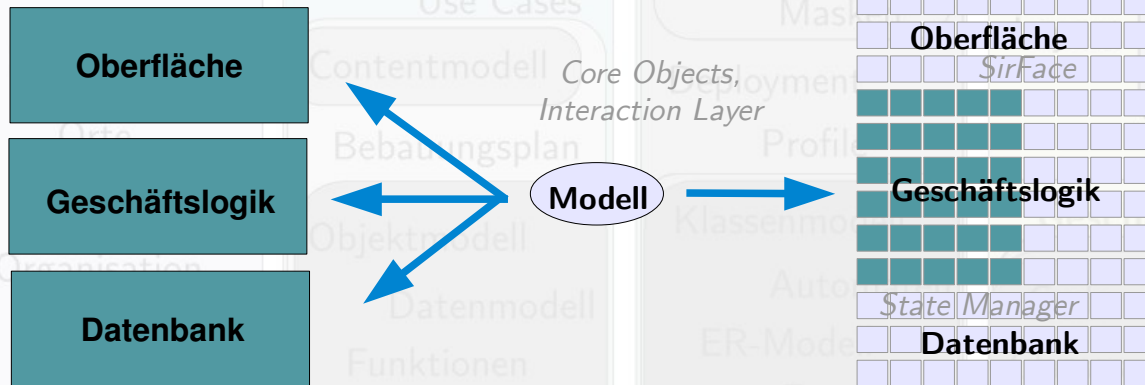
Handgefertigte Bauteile koexistieren problemlos mit generierten Bauteilen und Fertigteilen

Mannigfaltige Möglichkeiten
(Baupläne, abgeleitete Klassen,
Variantenpools, usw.)

Von Konfektionsware zu
Handarbeit mit vielen
Auffangnetzen

EM/OS 1.0
1993-2005

Trennung von Themen: je
feiner, umso formbarer



Traditional Business Layer

BIL - Business Interaction Layer

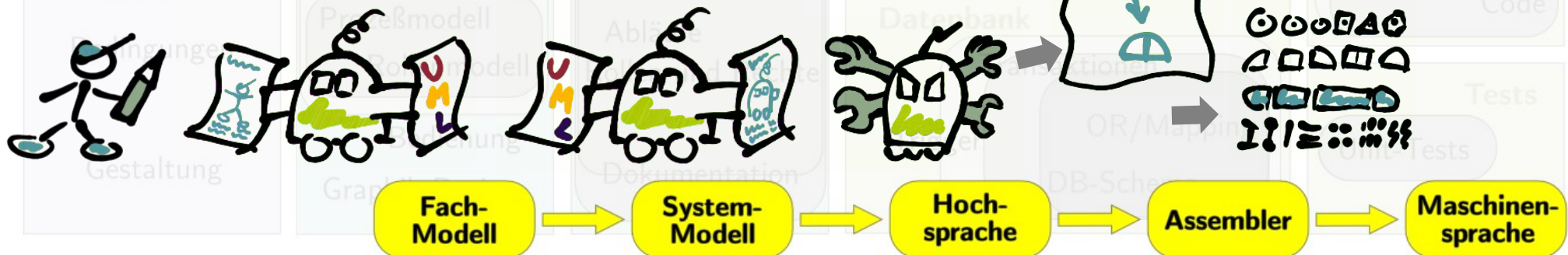
Transactions Workspaces Services Views

BCL - Business Core Layer

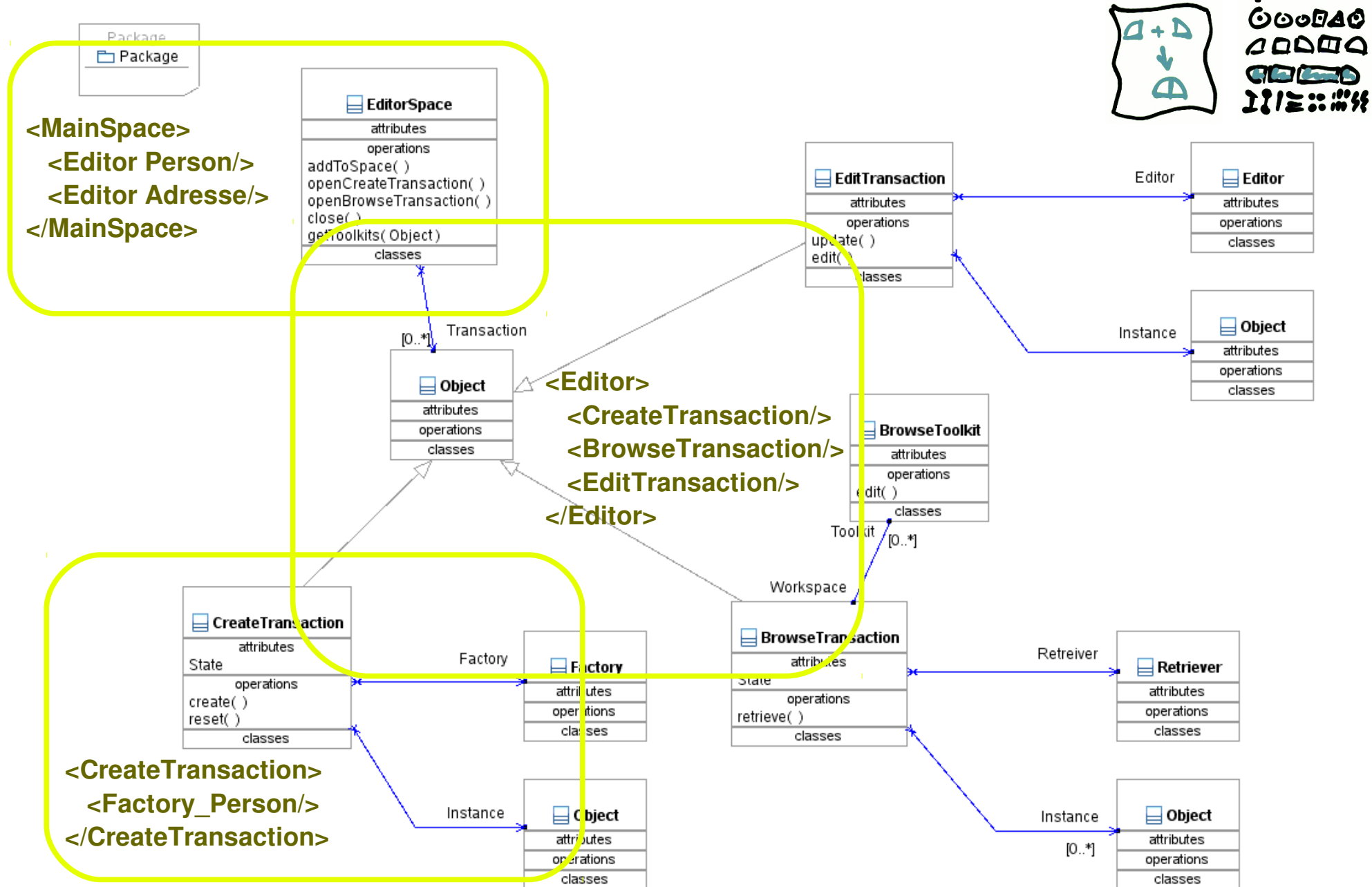
POJOs VUI & TSM Metadata Locating

Klassische Generierung

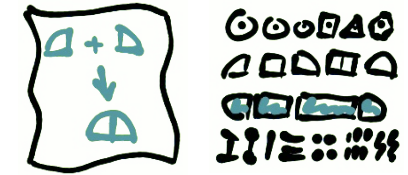
Generierung mit EM/OS



Roboter baut keinen Monolithen, sondern Bauteile und Baupläne



Roboter baut keinen Monolithen, sondern Bauteile und Baupläne



```

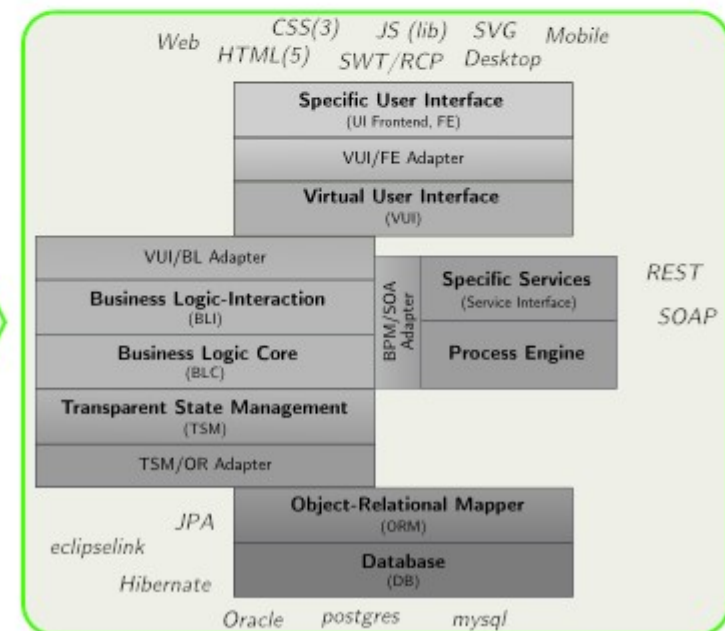
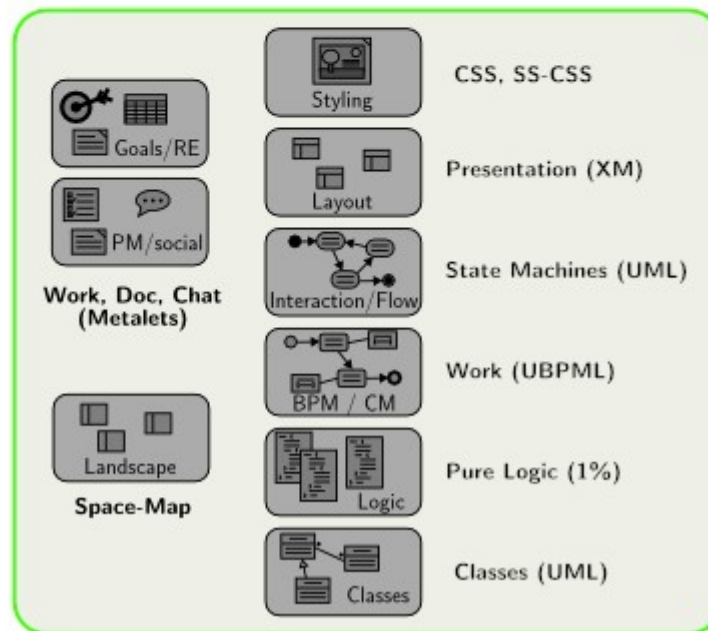
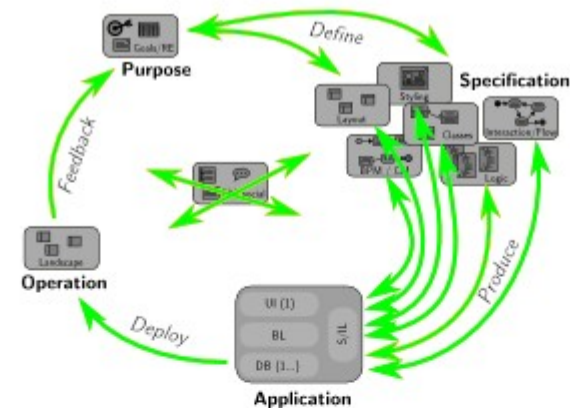
<SubSpace CLASS="VUIEntityObject_EditorSpace" >
  <ADO CLASS="EditorSpace" OID="space">
    <TargetType>org.oomodels.....Company</TargetType>
    <Operations>
      <Operation CLASS="Aggregate::org/.../CreateTransactionStarter_Company">
        <space OIDREF="space"/>
      </Operation>
      <Operation CLASS="Aggregate::org/.../BrowseTransactionStarter_Company">
        <space OIDREF="space"/>
      </Operation>
    </Operations>
  </ADO>
</SubSpace>
  
```



Hohe **Qualität** der Werkzeuge

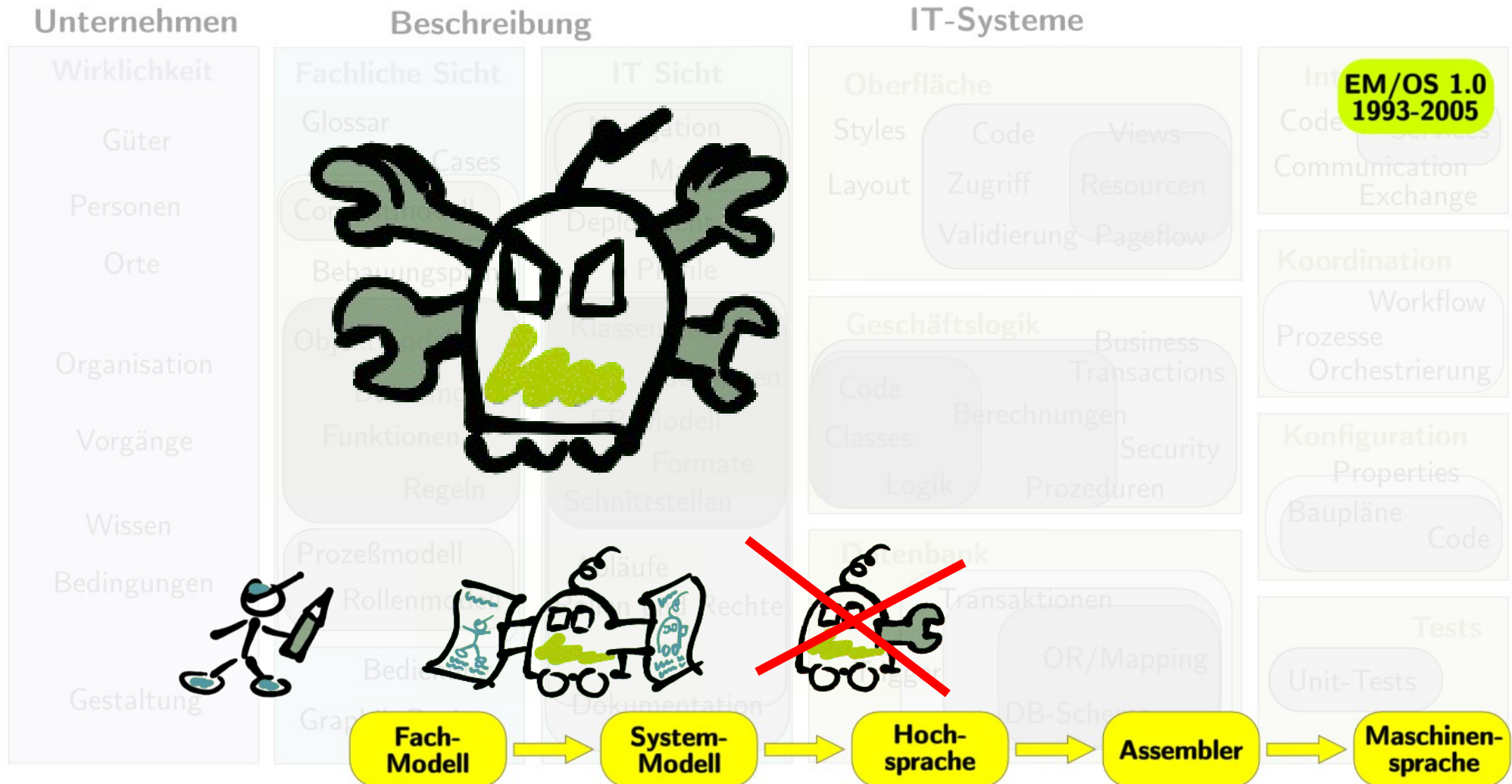
wartbare Transformationen

DRY, SoC und Verständlichkeit von M2M, M2T und Runtime-Aggregation



Hochentwickelter Generator in 4. Generation

www.oogenerator.org (open source)



Hochentwickelter Generator in 4. Generation

www.oogenerator.org (open source)

⌞ G-2.0-uml-1.0 *- coding: utf-8; *-

↳ Attributes

≈ ...

□ (T) Attributes(Object current) □...

⋮ «V(a: || "P/Attributes") { »...

protected <type> <|| "P/Name" ⌘ MC/LCU/JAVAID ▶;

public <type> get <|| "P/Name" ▶ ();

⋮ «}/V»...

□ /(T) □...



Fach-
Modell

System-
Modell

Hoch-
sprache

Assembler

Maschinen-
sprache

EM/OS 1.0
1993-2005

Inte-
code
ommunication
Exchange

oordination
Workflow
prozesse
Orchestrierung

onfiguration
Properties

Baupläne
Code

Tests

Unit-Tests

Hochentwickelter Generator in 4. Generation

www.oogenerator.org (open source)



G-2.0-uml-1.0 -*- coding: utf-8; -*-

↳ Attributes

↳...

□ ⓘ Attributes(Object current) □...

⋮ «V(a: || "P/Attributes") { »...

protected <type> ◀ || "P/Name" ⌘ MC/LCU/JAVID ▶;

public <type> get◀ || "P/Name" ▶ ();

⋮ «}/V»...

□ / ⓘ □...

«V(a: || "Ⓟ/Attributes") { »

Sehr gute Lesbarkeit durch Unicode

Modular wählbare Syntax

OO Templates

Modularisierbarkeit schafft Ordnung

Extrem performant, dynamische Übersetzung

Voller Java Sprachumfang verfügbar

Zahlreiche, konfigurierbare Syntaxelemente

Sehr reife Software



Hochentwickelter Generator in 4. Generation

www.oogenerator.org (open source)



G-2.0-uml-1.0 -*- coding: utf-8; -*-

↳ Attributes

↳...

□ ⓘ Attributes(Object current) □...

⋮ «∀(a:⋮ "P/Attributes") {»...

protected <type> ◀⋮ "P/Name"⌘MC/LCU/JAVID▶;

public <type> get◀⋮ "P/Name"▶ ();

⋮ «}/∀»...

□ /ⓘ □...

<% for(Object a:...

```
public class MyClass<X, Y<X>> {
    public X getX() { ...
```

Sehr gute Lesbarkeit durch Unicode

Modular wählbare Syntax

OO Templates

Modularisierbarkeit schafft Ordnung

Extrem performant, dynamische Übersetzung

Voller Java Sprachumfang verfügbar

Zahlreiche, konfigurierbare Syntaxelemente

Sehr reife Software



Hochentwickelter Generator in 4. Generation

www.oogenerator.org (open source)



G-2.0-uml-1.0 *- coding: utf-8; *-

↳ Attributes

↳...

□ ① Attributes(Object current) □...

⋮ «V(a: || "P/Attributes") {»...

protected <type> ◀|| "P/Name" ⌘ MC/LCU/JAVID ▶;

public <type> get◀|| "P/Name" ▶ ();

⋮ «}/V»...

□ /① □...

G-2.0-uml_state_machine...

□ base ClassTemplate □...

□ ① Features(String myarg) □...

□ → super.Features(myarg) □...

Sehr gute Lesbarkeit durch Unicode

Modular wählbare Syntax

OO Templates

Modularisierbarkeit schafft Ordnung

Extrem performant, dynamische Übersetzung

Voller Java Sprachumfang verfügbar

Zahlreiche, konfigurierbare Syntaxelemente

Sehr reife Software



Hochentwickelter Generator in 4. Generation

www.oogenerator.org (open source)



G-2.0-uml-1.0 -*- coding: utf-8; -*-

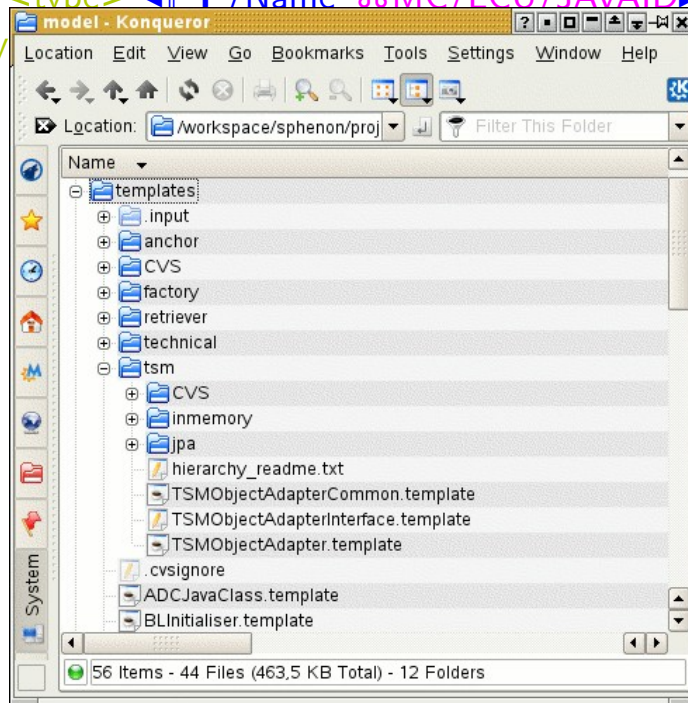
↳ Attributes

□ ⓘ Attributes(Object current) □...

⋮ «V(a:| "P/Attributes") {»...

protected <type> <| "P/Name" &MC/LCU/JAVID>;
public <ty

□ /ⓘ □...



Sehr gute Lesbarkeit durch Unicode

Modular wählbare Syntax

OO Templates

Modularisierbarkeit schafft Ordnung

Extrem performant, dynamische Übersetzung

Voller Java Sprachumfang verfügbar

Zahlreiche, konfigurierbare Syntaxelemente

Sehr reife Software

Hochentwickelter Generator in 4. Generation

www.oogenerator.org (open source)



G-2.0-uml-1.0 -*- coding: utf-8; -*-

→ Attributes

→ ...

□ ① Attributes(Object current) □...

⋮ «V(a: || "P/Attributes") { »...

protected <type> ◀ || "P/Name" ⌘ MC/LCU/JAVID ▶;

public <type> get◀ || "P/Name" ▶ ();

⋮ «}/V»...

□ /① □...

Sehr gute Lesbarkeit durch Unicode

Modular wählbare Syntax

OO Templates

Modularisierbarkeit schafft Ordnung

Extrem performant, dynamische Übersetzung

Voller Java Sprachumfang verfügbar

Zahlreiche, konfigurierbare Syntaxelemente

Sehr reife Software



Hochentwickelter Generator in 4. Generation

www.oogenerator.org (open source)



G-2.0-uml-1.0 -*- coding: utf-8; -*-

↳ Attributes

↳...

□ ⓘ Attributes(Object current) □...

⋮ «∀(a:⋮"P/Attributes") {»...

protected <type> ◀⋮"P/Name"⌘MC/LCU/JAVID▶;

public <type> get◀⋮"P/Name"▶ ();

⋮ «}/∀»...

□ /ⓘ □...

◁mytag(myarg,...)▷

Sehr gute Lesbarkeit durch Unicode

Modular wählbare Syntax

OO Templates

Modularisierbarkeit schafft Ordnung

Extrem performant, dynamische Übersetzung

Voller Java Sprachumfang verfügbar

Zahlreiche, konfigurierbare Syntaxelemente

Sehr reife Software



Hochentwickelter Generator in 4. Generation

www.oogenerator.org (open source)



G-2.0-uml-1.0 **-*- coding: utf-8; -*-**

↳ Attributes

↳...

□ ⓘ Attributes(Object current) □...

⋮ «V(a:| "P/Attributes") {»...

protected <type> ◀| "P/Name" ⚡ MC/LCU/JAVAID ▶;

public <type> get◀| "P/Name" ▶ ();

⋮ «}/V»...

□ /ⓘ □...

Sehr gute Lesbarkeit durch Unicode

Modular wählbare Syntax

OO Templates

Modularisierbarkeit schafft Ordnung

Extrem performant, dynamische Übersetzung

Voller Java Sprachumfang verfügbar

Zahlreiche, konfigurierbare Syntaxelemente

Sehr reife Software

C 1.0 1993...

C++ 1996...

perl 1998..., 1.0 2000

Java 1.0 2002, 2.0 2005, 3.0 2008



Neuer Roboter versteht Bauanleitungen und baut Teile zusammen

Objektbaupläne und der Object Assembler - www.xocp.org (open source)

Unternehmen

baut Objektaggregate aus Bauteilen
anhand von Bauplänen

ehemals: gluecode/hardcoded

breites Anwendungsspektrum

extrem performant (JIT Compiler)

dynamisches Laden und
Neuübersetzen

OO Baupläne

Beschreibung

IT Sicht

```
package my.code;
class JavaClass {
    //...
}
```

Java Source Code

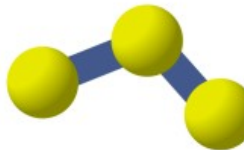
```
<?xml ?>
<MyAggregate>
  <!-- ... -->
</MyAggregate>
```

Object Construction Plan

IT-Systeme

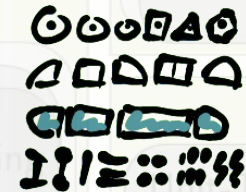
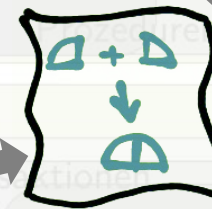
Oberfläche

Object Instance



Object Aggregate

EM/OS 1.0
1993-2005



Fach-
Modell

System-
Modell

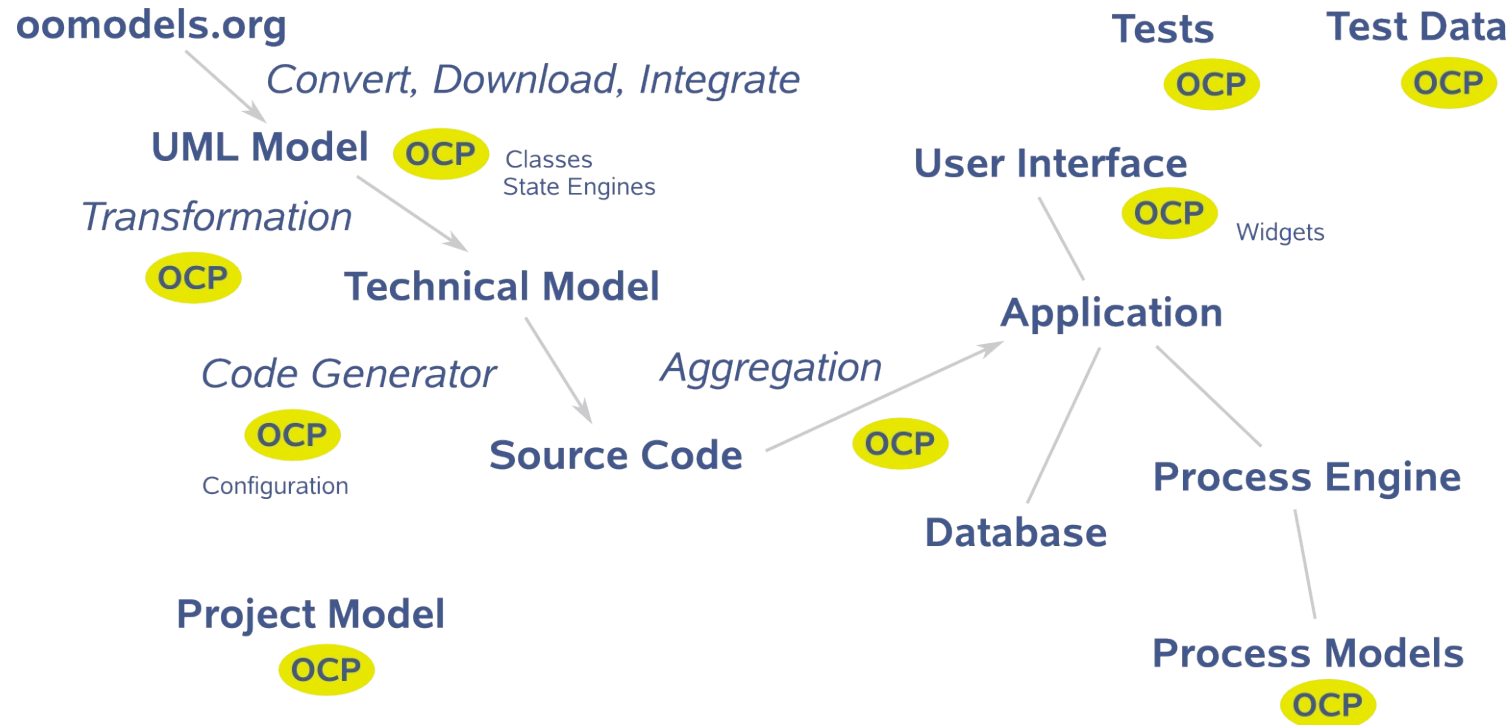
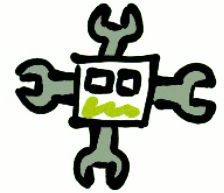
Hoch-
sprache

Assembler

Maschinen-
sprache

Neuer Roboter versteht Bauanleitungen und baut Teile zusammen

Objektbaupläne und der Object Assembler - www.xocp.org (open source)

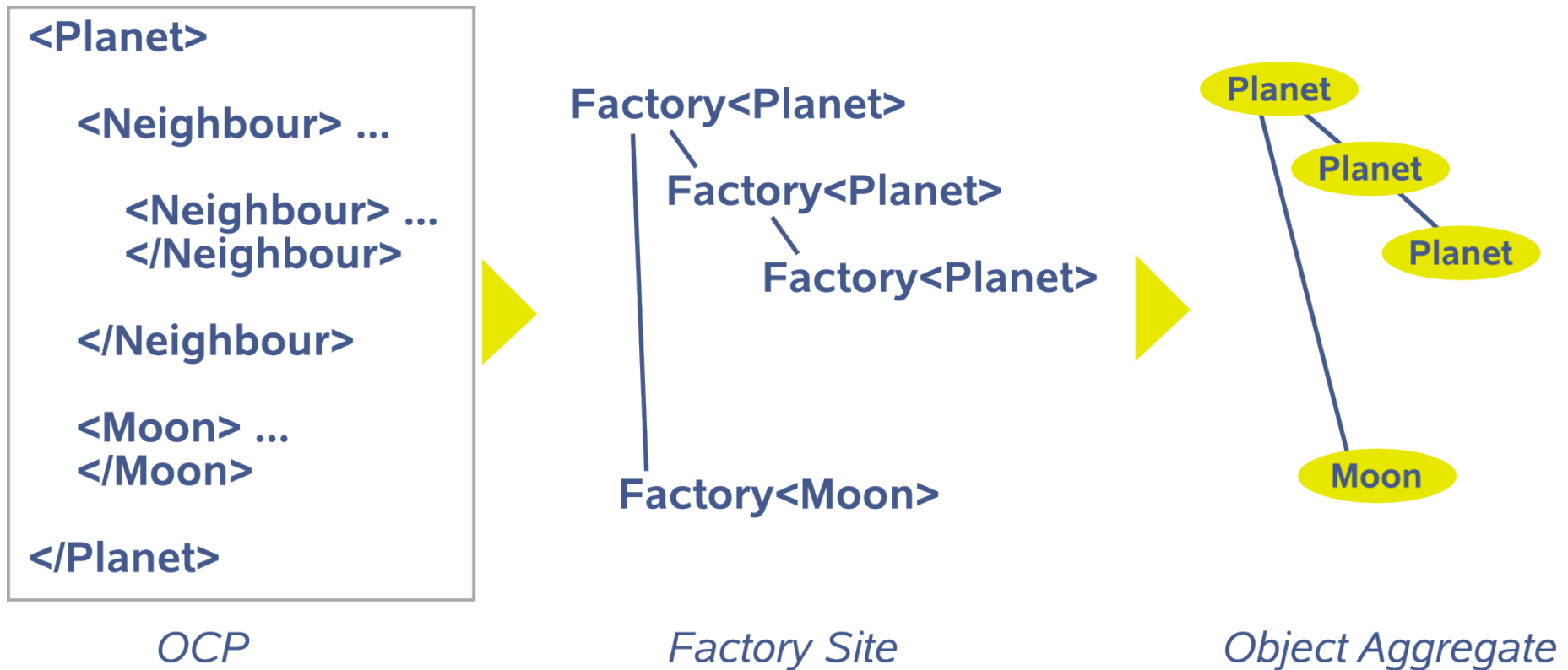
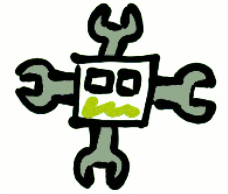


- ▶ 1996: Laden von Testdaten
- ▶ heute: 15% OCP-Anteil, bezogen auf Java



Neuer Roboter versteht Bauanleitungen und baut Teile zusammen

Objektbaupläne und der Object Assembler - www.xocp.org (open source)



Weitere Konzepte

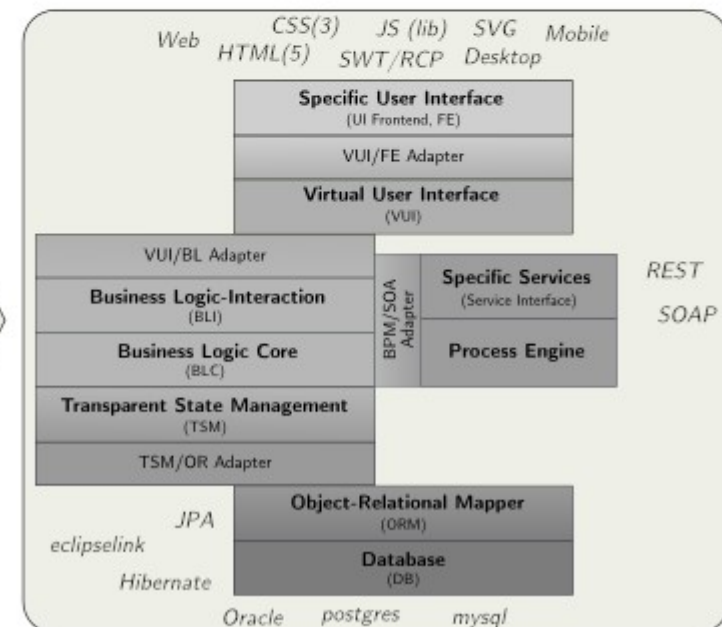
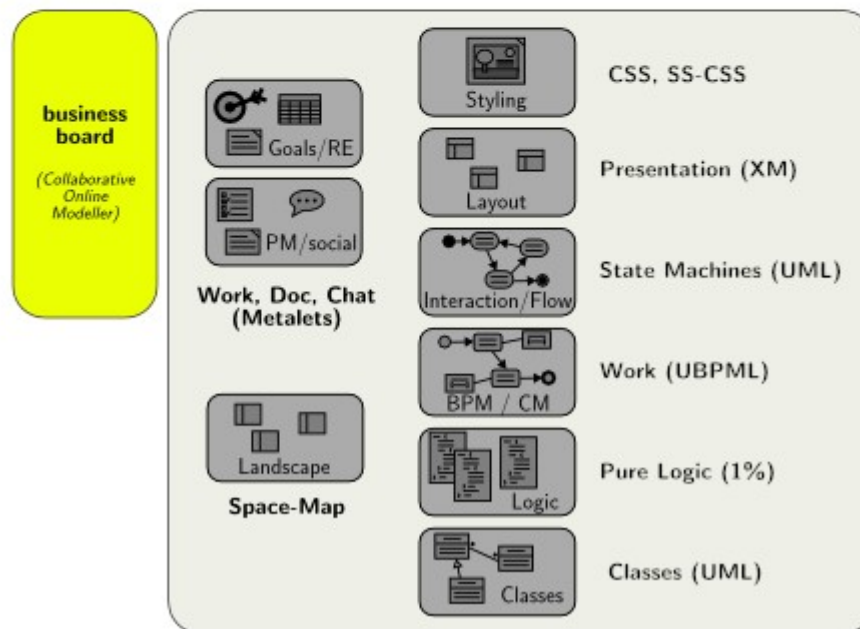
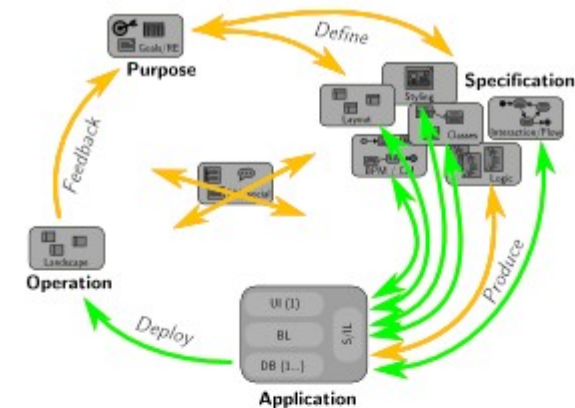
Locator	systematisch erweiterte URLs zur Lokalisierung zahlreicher Ressourcen
Operation	Abstraktion als Grundlage für Services, Scripting, Automatisierung
Space	Abstraktion für Landschaften, Raumaufteilungen, Installationen
Filing/Graph	Abstrakter Zugriff auf annotierte Datei-Ressourcen
Context	Universeller Methoden-Parameter und Object-Member
Execution	Sehr reichhaltigere Variante von Futures, Either & Co.
Exceptions	Sinnvoll kategorisiert
Shell	Integriertes Meta-Scripting zahlreicher Interpreter lokal und remote



business board: collaborative online modeller

Alles an einem Ort
Support & Zuarbeit per Mausklick
Powered by Itself

EM/OS-App incl. Graphical Modeller !



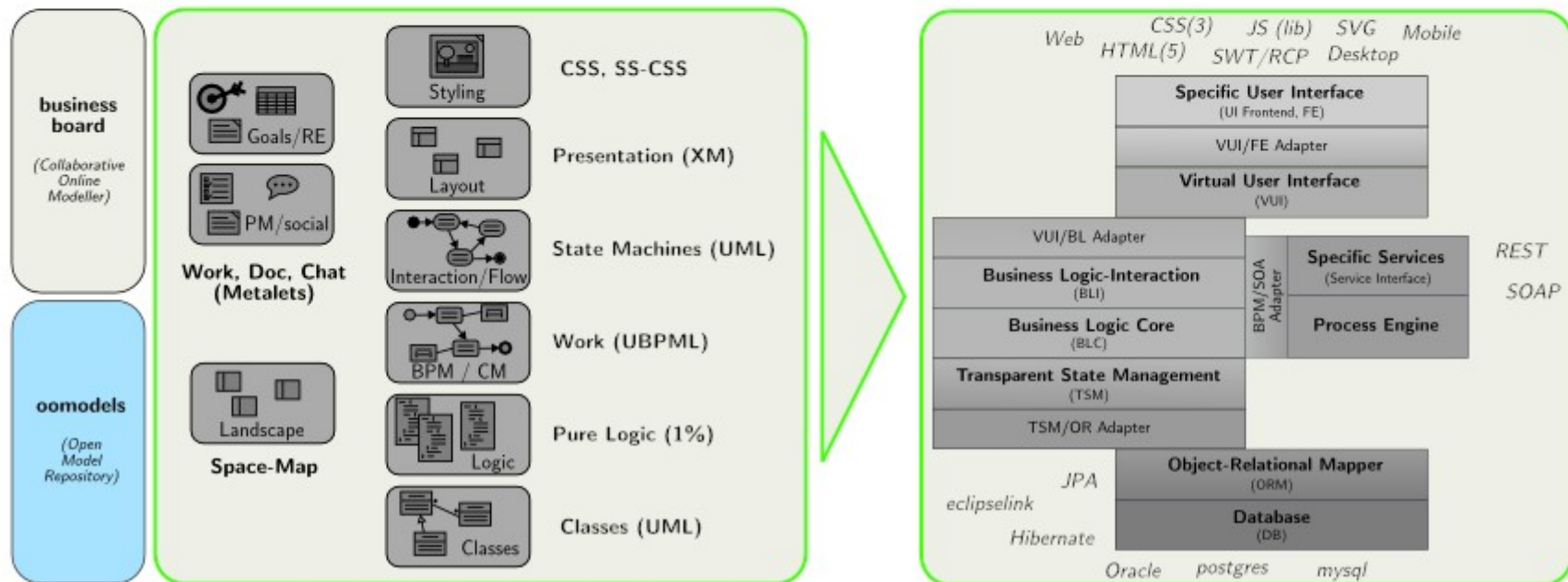
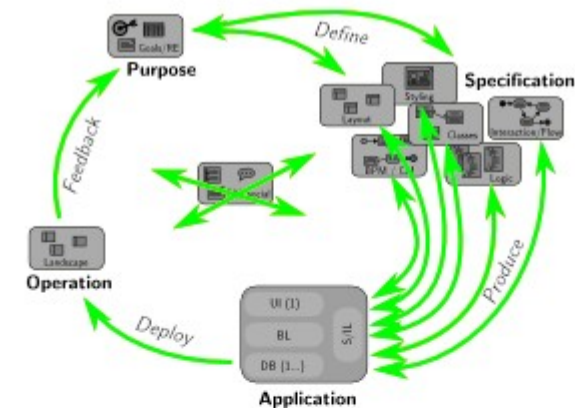
oomodels: open model repository

“wikipedia für Modelle”

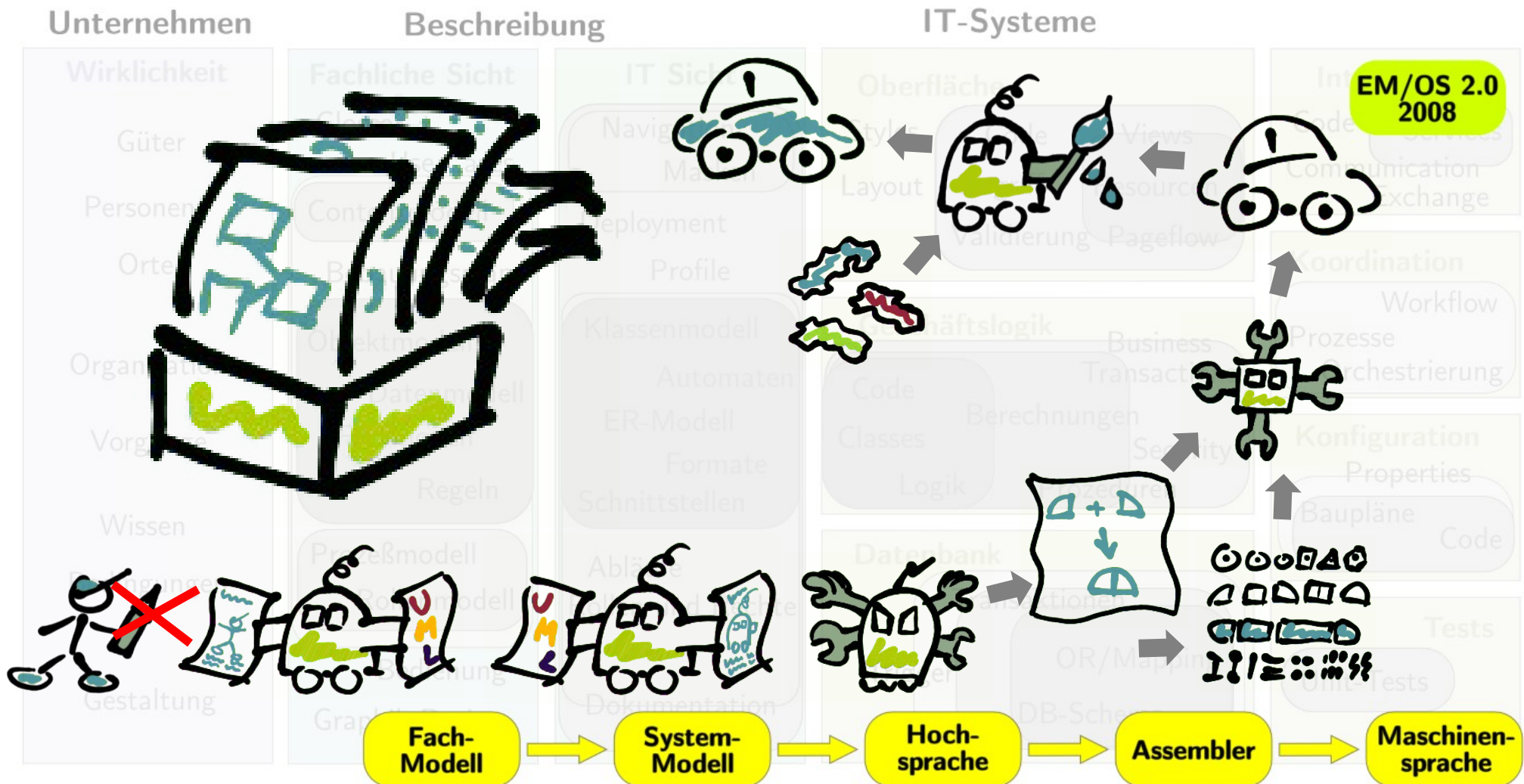
Community-Exchange

Referenz für App-Collaboration

Bounded Context/Transfer Models



Modell Repository - www.oomodels.org



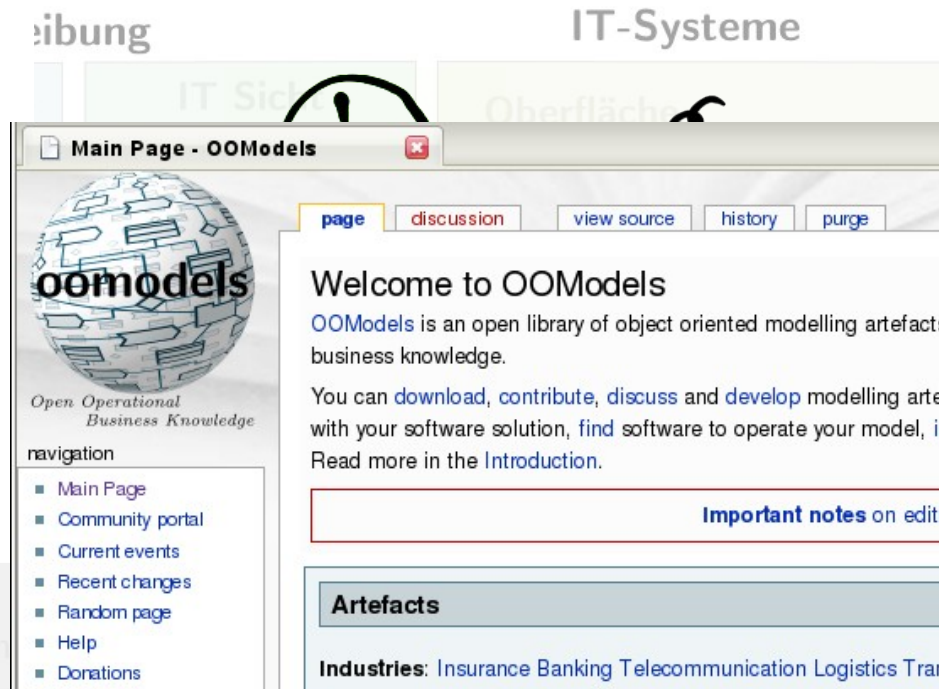
Modell Repository - www.oomodels.org

Frei verfügbares Basis-
Fachwissen: Modelle,
Teilmodelle, Klassen, ...

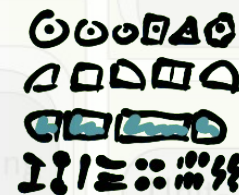
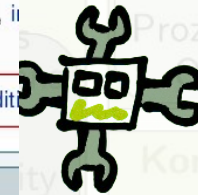
Ready-to-Use: Download,
Online-Konverter, Eclipse-
Plugin, ...

Kollaboratives Arbeiten einer
Architekten-Community

“Wikipedia für IT-Konzepte”



EM/OS 3.0
2010



Fach-
Modell

System-
Modell

Hoch-
sprache

Assembler

Maschinen-
sprache

Modell Repository - www.oomodels.org



Artefact:Model/org/oomodels/sandbox/business/foundation/entities/Person - OOModels - Opera

File Edit View Bookmarks Widgets Tools Help

Artefact:Model/org/oo... X +

http://www.oomodels.org/index.php/Artefact:Model/org/oomodels/sandbox/business/foundation/entities/Person

AndreasLeue my talk my preferences my watchlist my contributions log out

artefact discussion edit history delete move protect watch purge

Artefact:Model/org/oomodels/sandbox/business/foundation/entities/Person

< Artefact:Model | org | oomodels | sandbox | business | foundation | entities

Open Operational Business Knowledge

navigation

- Main Page
- Glossary
- Roadmap
- Recent changes
- Help
- Impressum

search

Go Search

toolbox

- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link

Code [edit]

Entity

△

|

Person
«CoreClass»

A human person

Name	NameOfPerson	« »
Sex	Sex	« »
Birthday	Date	« »

Comments [edit]

Further attributes: title, sex, etc.

Categories: Type:org/oomodels/WIML/1.0 | Domain:business/foundation/entities | Type:org/oomodels/wiki/Model

Classification

Type Type:org/oomodels/WIML/1.0

Domain Domain:business/foundation/entities

Category Type:org/oomodels/wiki/Model

Maturity work in progress

More

Download Code (dev)

Diagram

Namespace (more)

Provides a sandbox for modelling.

It is intended to develop generalised, sufficiently abstract artefacts that can serve as common ground for more specific models.

create new pages

CC BY SA This page was last modified 23:01, 29 October 2009. This page has been accessed 178 times. Content is available under Attribution-Share Alike 3.0 Unported. Privacy policy About OOModels Disclaimers

Powered By MediaWiki

100%

Modell Repository - www.oomodels.org



models

rational
Business Knowledge

age
ry
ap
changes

sum

Search

ks here
changes
file
pages

Editing / www.oomodels.org / www.oomodels.org/sandbox/business/foundation/entities/Entity



```
{{Metabox
| Meta_Type    = Type:org/oomodels/WIML/1.0
| Meta_Domain  = Domain:business/foundation/entities
| Meta_Category = Type:org/oomodels/wiki/Model
| Meta_Download = true
| Meta_Diagram = Artefact:Diagram/org/oomodels/sandbox/business/foundation/entities/Entities
| Meta_AddTypeToCategory  = true
| Meta_AddCategoryToCategory = true
| Meta_Maturity = work in progress
}}
```

== Code ==

```
<wiml>
* Person Core Class
> org/oomodels/sandbox/business/foundation/entities/Entity
  A human person
  ** Name org/oomodels/sandbox/business/foundation/entities/NameOfPerson
  ** Sex org/oomodels/sandbox/business/foundation/entities/Sex
  ** Birthday Date
</wiml>
```

Modell Repository - www.oomodels.org



OOModels Download - Select Format - Opera

File Edit View Bookmarks Widgets Tools Help

OOModels Download ... X

<http://www.oomodels.org/download/?id=Artefact:Model/org/oomodels/sandbox/business/foundation/entities/Person>



Please choose your desired download format:

- ☐ Type:org/oomodels/wiki/ArtefactDescription [info] - (extracted metadata)
- ☐ Type:org/oomodels/WIML/1.0 [info] - (original code, without conversion)
- ☐ Type:org/ooem/UMLClass/Multipart/1.0 [info] - (converted code, via [wiml-1.0-ooem-multipart-1.0])
- ☒ Type:org/ooem/UMLClass/Multipart/1.0/Refactored [info] - (converted code, via [wiml-1.0-ooem-multipart-1.0], [regexp-refactoring])
[add a format]

☒ Please read the [disclaimer](#) and check this box to confirm that you understand and accept it, otherwise do not proceed with downloading. [Download](#)

[\[back to oomodels\]](#) [\[more about downloading\]](#)

100%

Modell Repository - www.oomodels.org



Artefact:Diagram/org/oomodels/sandbox/business/foundation/entities/Entities - OOModels - Opera

File Edit View Bookmarks Widgets Tools Help

Artefact:Diagram/org/o... X

http://www.oomodels.org/index.php/Artefact:Diagram/org/oomodels/sandbox/business/foundation/entities/Entities

AndreasLeue my talk my preferences my watchlist my contributions log out

artefact discussion edit history delete move protect watch purge

Artefact:Diagram/org/oomodels/sandbox/business/foundation/entities/Entities

< Artefact:Diagram | org | oomodels | sandbox | business | foundation | entities

Code [edit]

Classification

Type	Type:org/oomodels/WIDL/1.0
Domain	Domain:it/test
Category	Type:org/oomodels/wiki/Diagram

More

Download Code [\(dev\)](#)

Namespace (more)

Provides a sandbox for modelling.

It is intended to develop generalised, sufficiently abstract artefacts that can serve as common ground for more specific models.

[create new pages](#)

Relationship

End1 : Entity
End2 : Entity

Entity
Name : Name

Group
Name : NamePlain

Company
Name : NamePlain

Person
Name : NameOfPerson
Sex : Sex
Birthday : Date

Name

NamePlain
Name : String

NameOfPerson
Vorname : String
Nachname : String

Categories: Type:org/oomodels/WIDL/1.0 | Domain:it/test | Type:org/oomodels/wiki/Diagram

CC BY SA

This page was last modified 20:31, 8 November 2009. This page has been accessed 40 times. Content is available under Attribution-Share Alike 3.0 Unported. Privacy policy About OOModels Disclaimers

Powered By MediaWiki

Enterprise Knowledge Technology

Modell Repository - www.oomodels.org



Editing Antefact.Diagram/org/oomodels/sandbox/business/foundat

B **I** **Ab** **A** **√n** **W** **—**

```

| Meta_Category      = Type:org/oomodels/wiki/Diagram
| Meta_Download      = true
| Meta_AddTypeToCategory = true
| Meta_AddCategoryToCategory = true
}}

== Code ==

<widl>

        ./Entity                ./Relationship

./Group

        ./Company              ./Name

        ./Person

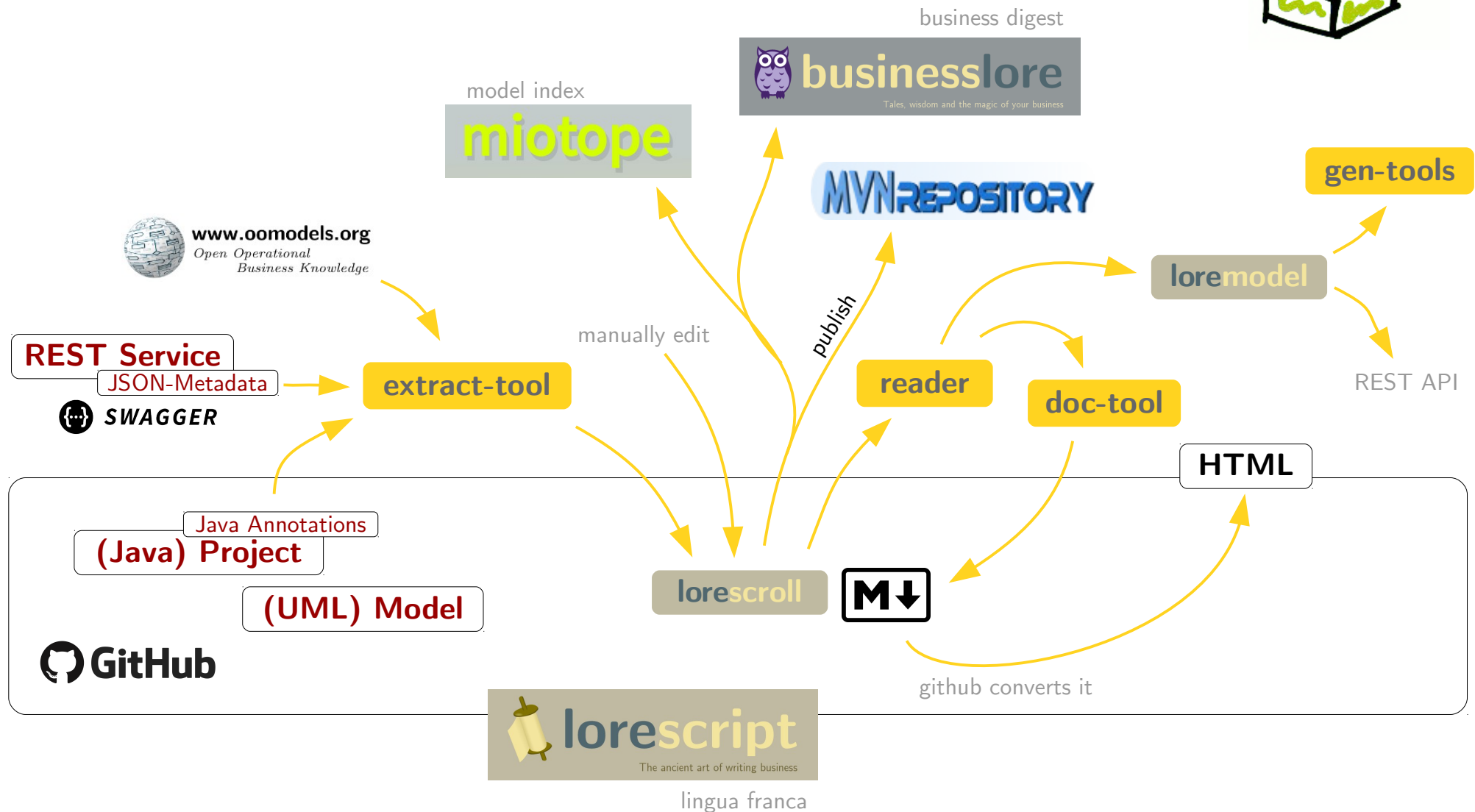
        ./NamePlain            ./NameOfPerson

</widl>

```

Please note that all contributions to OOModels are considered to be released under the Attribution-S

Modell Repository Reloaded - Best Practise Remix



Modell Repository - lorescript.org



paradigm
compliant
terminology



lorescript

The ancient art of writing business

80/20* conceptually complete
(classes, systems, processes...)



Scrolls



Tales



Sigils



Maps



Thesauri



Recipes



Spells

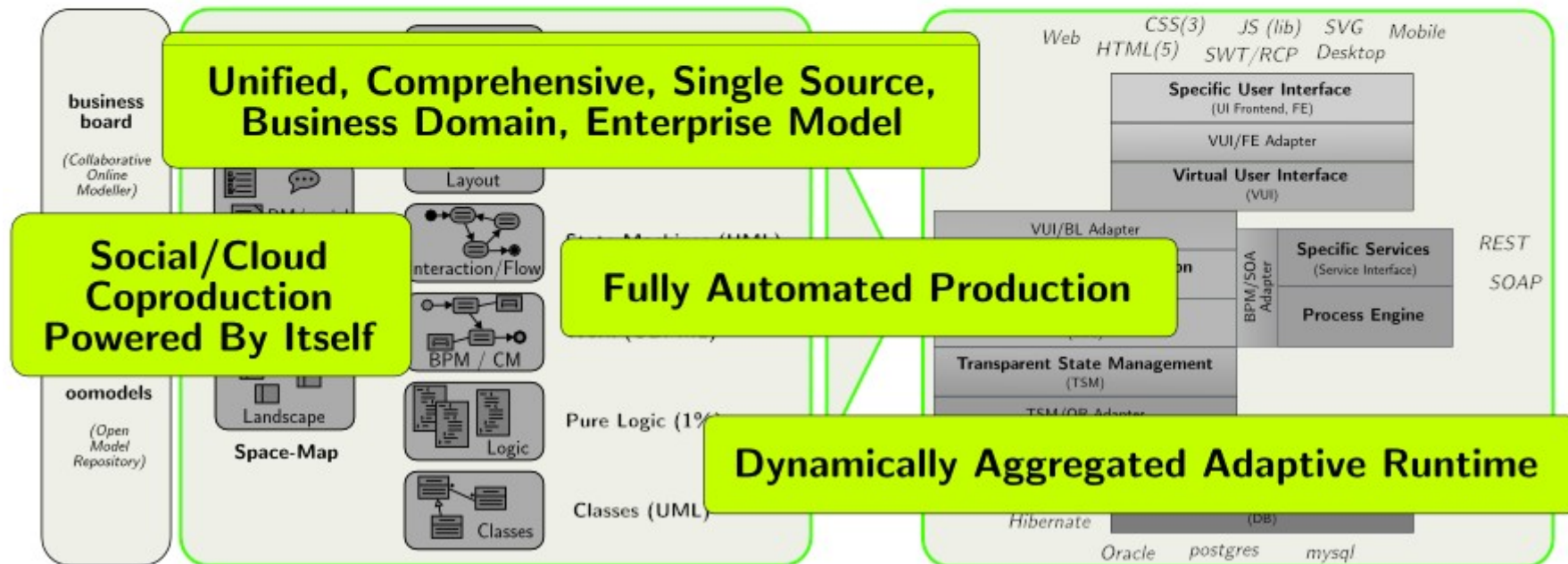
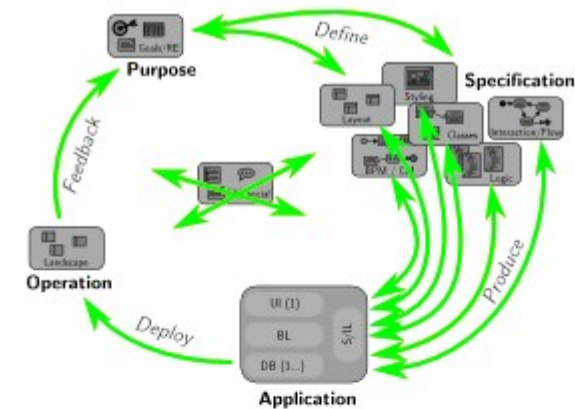
grounded on
best practises
(md, tar, yaml, json)

scaling from
agile sticky notes to
executable model & EA

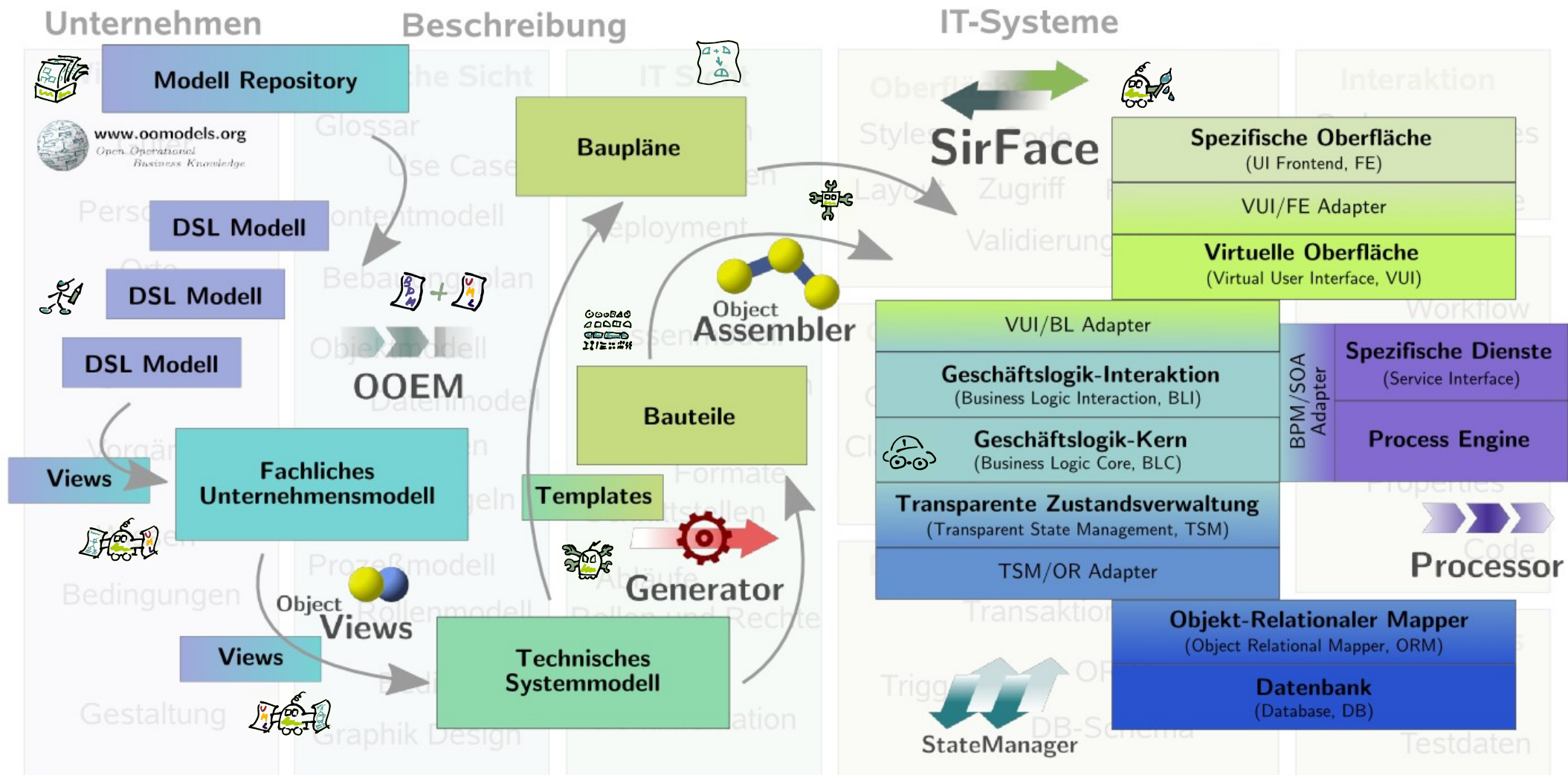
conceptually
open & extensible
(ocp, oorl, ...)



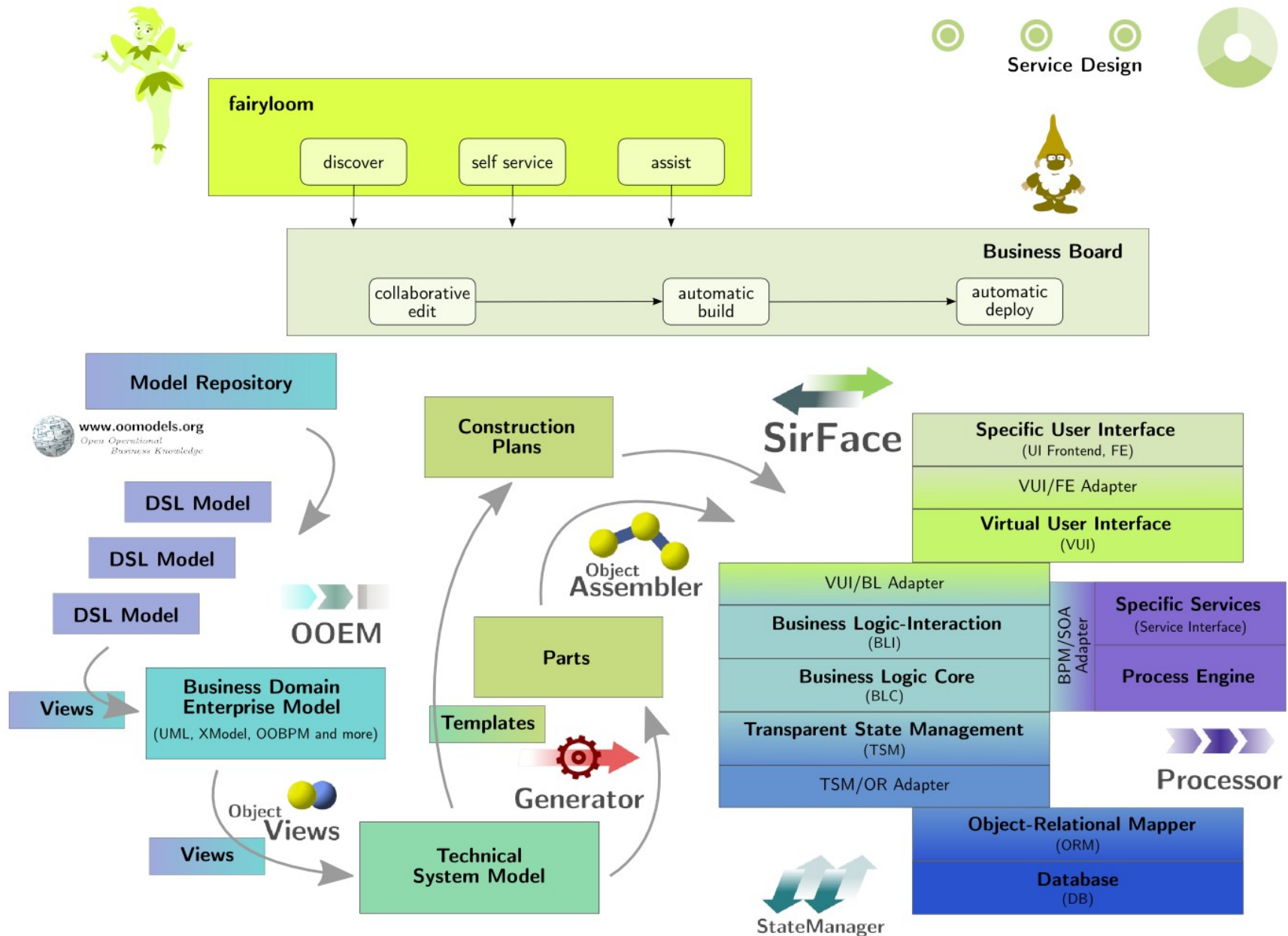
Lifecycle-Excellence



EM/OS Architektur



Product - Tool & Service



Links



www.fairyloom.de

Shop und Service-Portal



board.fairyloom.de

businessboard Demo



www.oomodels.org

Model Repository (Creative Commons)



www.xocp.org

Objektbaupläne (Open Source)



www.oogenerator.org

Generator (Open Source)



www.sphenon.de

Sphenon GmbH



doc.sphenon.de/com/sphenon/company/Sphenon2017.pdf

Präsentation über Sphenon, EM/OS
das businessboard und fairyloom



Technologie-Vergleich

