

SAXONY MEDIA SOLUTIONS GMBH

PROZESSOPTIMIERUNG UND DIGITALISIERUNG IN DER PRODUKTION UND LOGISTIK

Inhalt

- Vorstellung der SaxMS
 - Unternehmen, Technologie
- Softwareentwicklung Anforderungsaufnahme
 - Entwicklungsprozess, UML, SysML, OCL
- Softwareentwicklung Projektmanagement
 - SCRUM, Produktionsplanung, Softwareentwicklung der vollständige Zyklus
- Historischer Aufbau insb. Softwareentwicklung
 - Beispiele, Methoden, Tools, Architekturen, Software

SaxMS Technologie und Aufstellung

Saxony Media Solutions GmbH

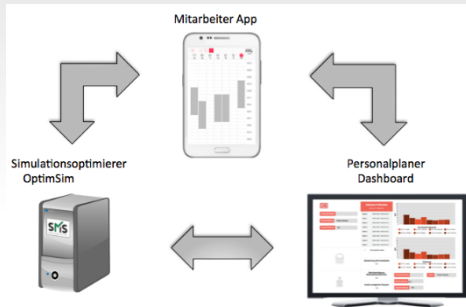
- Softwaregestützte Planung und Steuerung in Produktion und Logistik
- Simulation, simulationsbasierte Optimierung, digitale Zwilling, Solver, AI, maschinelles Lernen, automatische Modellgenerierung
- Zusammenarbeit mit der Technischen Universität Dresden
 - Professur für BWL, insb. Industrielles Management, Technische Logistik insb. Fabrikplanung, Professur für Prozesskommunikation, Professur Modellierung und Simulation
- Team aus Softwareentwicklern, Softwaredesignern und Ingenieuren
- Modernste Technologie,
ausgezeichnet als Deutsche Bahn Innovationsprodukt 2016

Partner



Kompetenzen

Prozessoptimierung



▪ Produktion & Logistik

▪ Komponenten

- » Backend
- » Dashboard
- » Mobile Anwendungen
- » Automatisierung

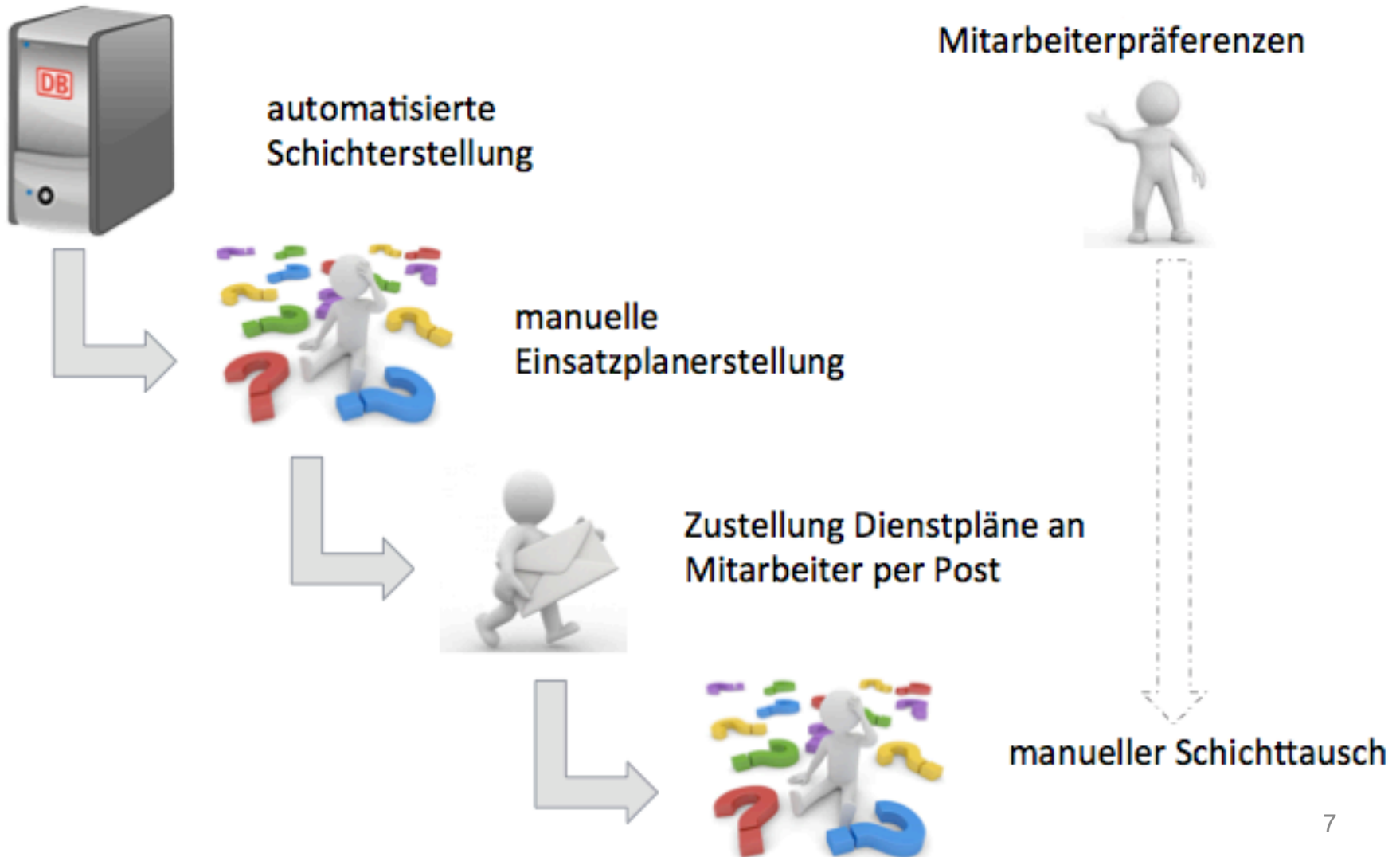
Automatisierte Planung

- **Große Optimierungserfahrung**
- Erarbeitung von Realisierungsvorschlägen und -konzepten
- Erstellung von Lasten- und Pflichtenheften
- Geschäftsprozessmodellierung mit Schwerpunkt kundenspezifischer Anpassung
- Umsetzung kundenspezifischer Anpassungen
- Schulungen und Workshops
- Analyse des Einsparpotenzials und Nutznachweises

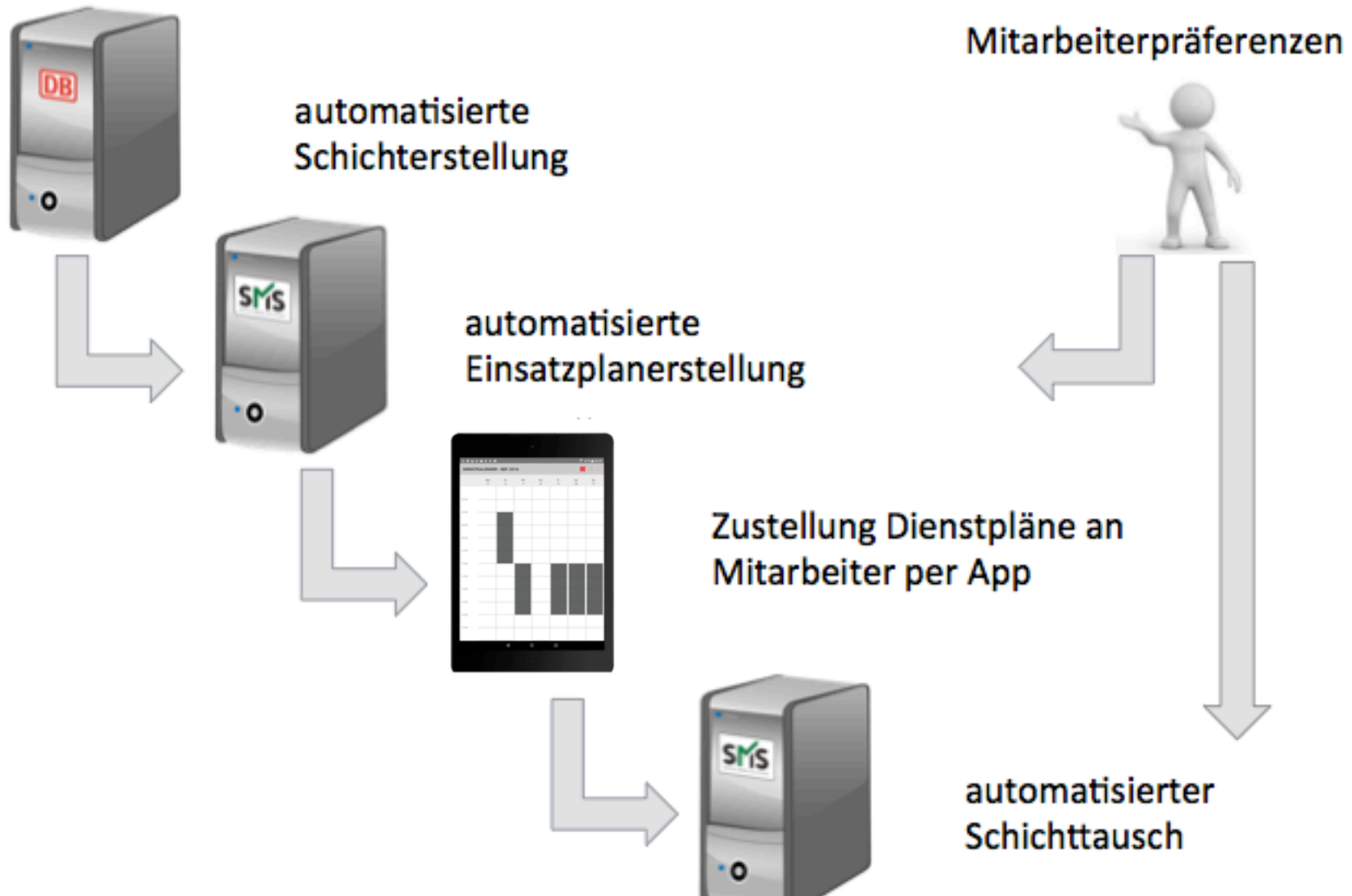
Softwareentwicklung

- **Softwareexperten mit großem Knowhow**
- Erarbeitung von Realisierungsvorschlägen und -konzepten
- Erstellung von Lasten- und Pflichtenheften
- Agile Entwicklung nach Kundenanforderungen
- Backend-, Frontend-, mobile Entwicklung
- Java, C#, Java Script, Hibernate, automatische Codegenerierung
- Schnittstellen- und Datenbankspezialisten

Aktueller Ablauf Personalplanung

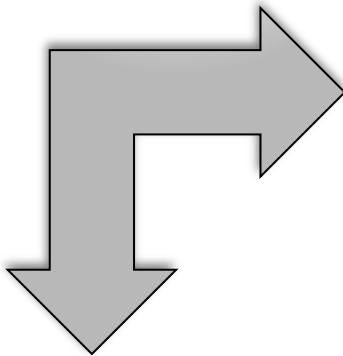


Personalplanung mit MIT

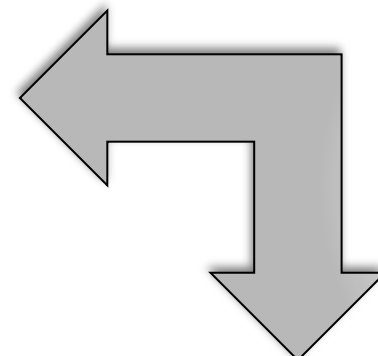


SaxMS-APS-Komponenten

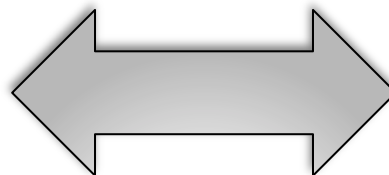
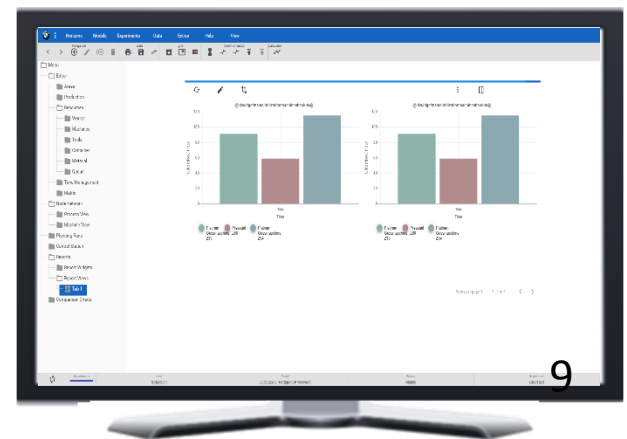
Mobile-Applikation



Simulation - Planung
- Optimierung



APS-
Dashboard



SaxMS-Produkte

Produktion-/Logistikplanung SaxMS APS

- **Produktionsvollzug**
- Ressourcen- und Betriebsmittelmanagement
- Auftragsverwaltung von Auftragsnetzen sowie Stücklisten
- Produktionsvollzugsplanung
- Materialplanung zum genauen Materialbedarfsvollzug
- Lieferterminanalyse
- Wartungsplanung

Personalplanung SaxMS PEP

- **Personaleinsatzplanung**
- Automatische Schichtplanerstellung
- Umplanung mit minimaler Planänderung
- Wechselsvorschläge bei Ausfall
- Manuelle Planänderungen und Bewertung
- Urlaubs- u. Krankheitsverwaltung
- Plananalyse bspw. für Urlaubswünsche
- Stamm- u. Bewegungsdatenverwaltung

Kundenplattform SaxMS CP

- **Bestellvorgänge**
- Anfragen-/Auftragserstellung
- Nachforderungen
- Übersicht über Status der Aufträge
- Einfaches Verwalten der Anfragen/Aufträge für Kunden und Disponenten

Bereiche SaxMS-Software

Disposition



- Personalplanung
- Fahrzeugplanung
- Ausfallbehandlung

Medienbruchfreie Kommunikation



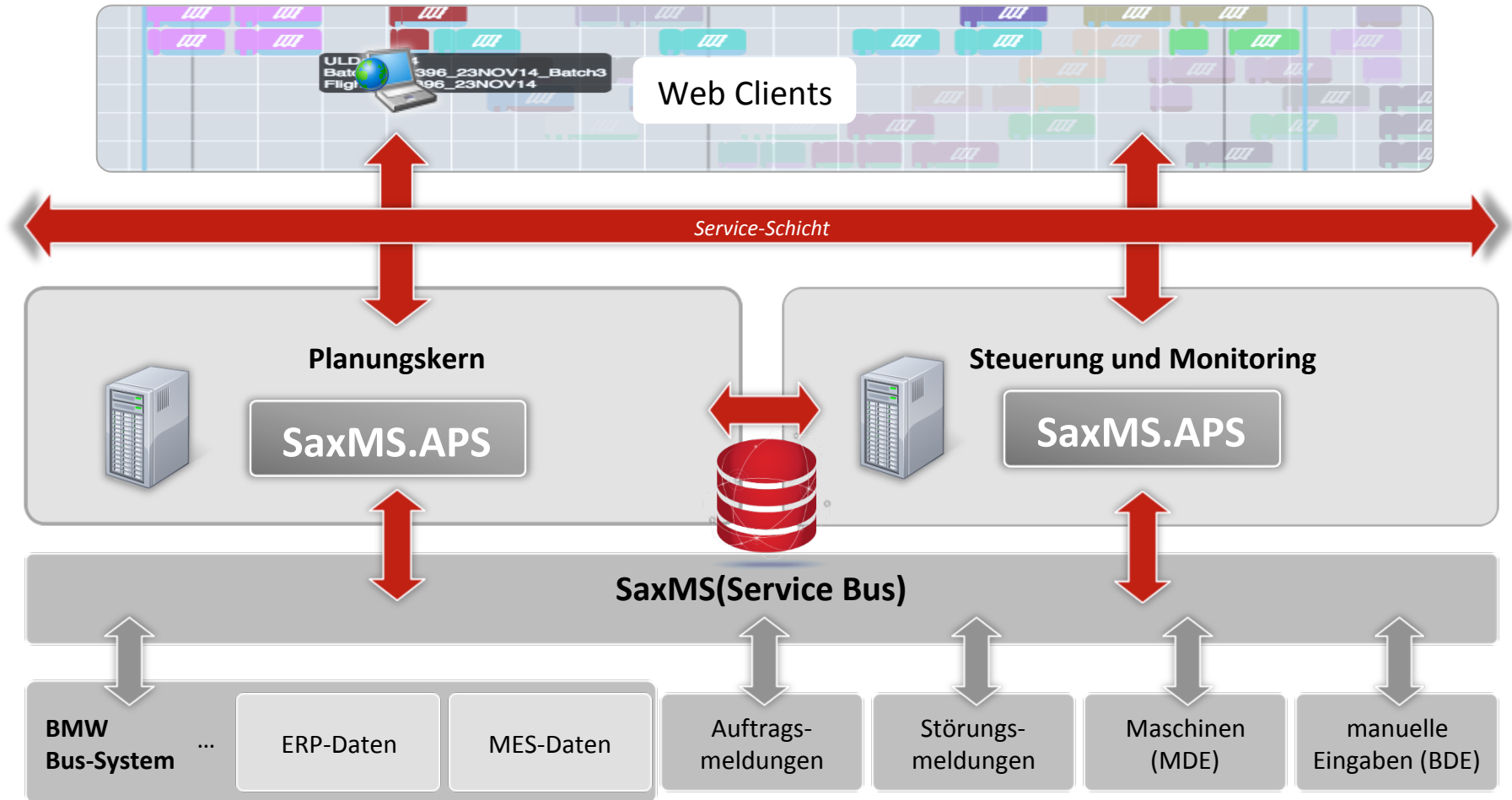
- Dokumente digitalisiert
- Berichte digitalisiert
- Kommunikation digitalisiert

Mitarbeiter

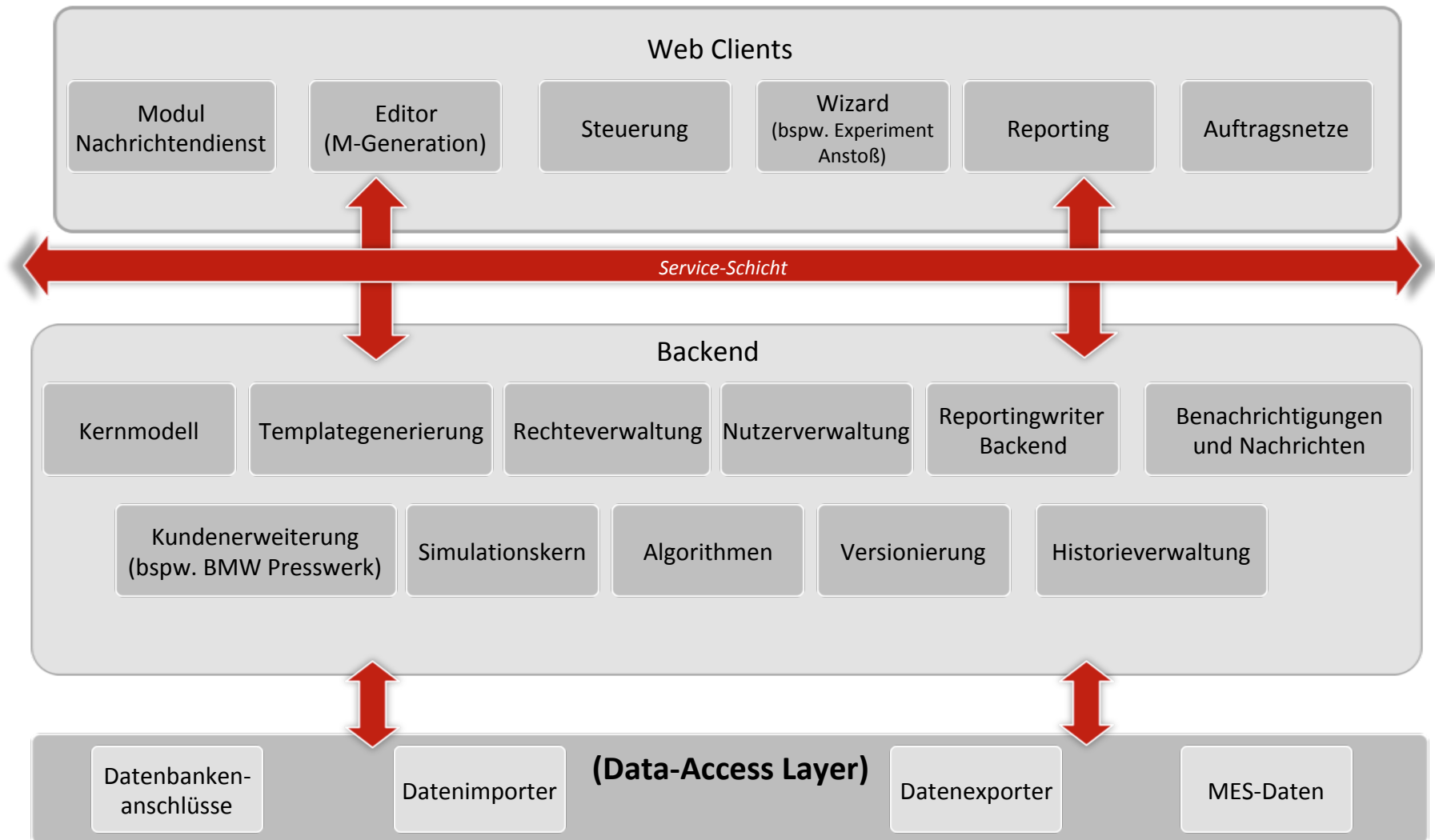


- Einsatzplanung
- Schichttätusche
- Dienstmeldungen

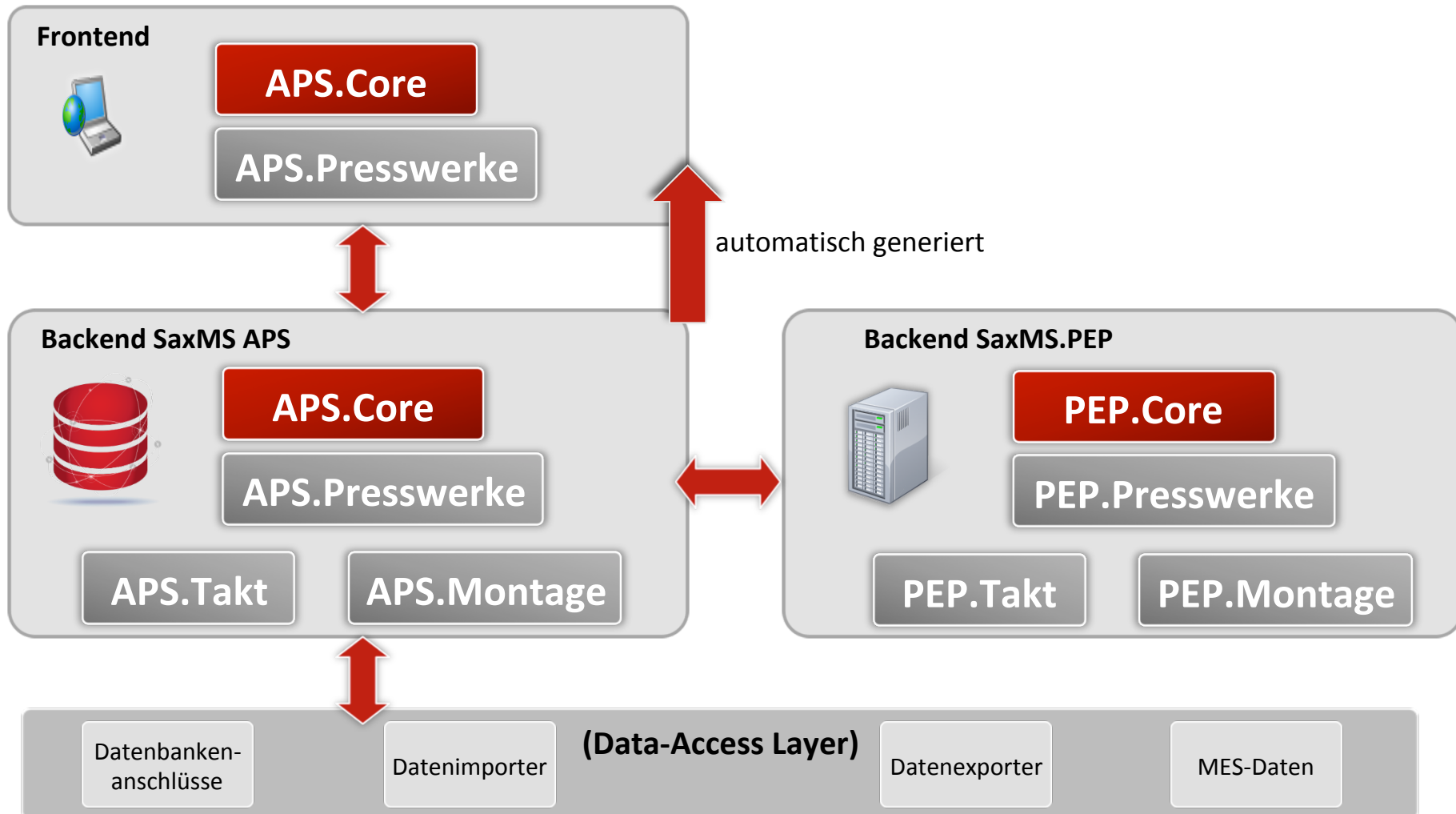
SaxMS-APS-Systemarchitektur



Modularer Aufbau von SaxMS-APS

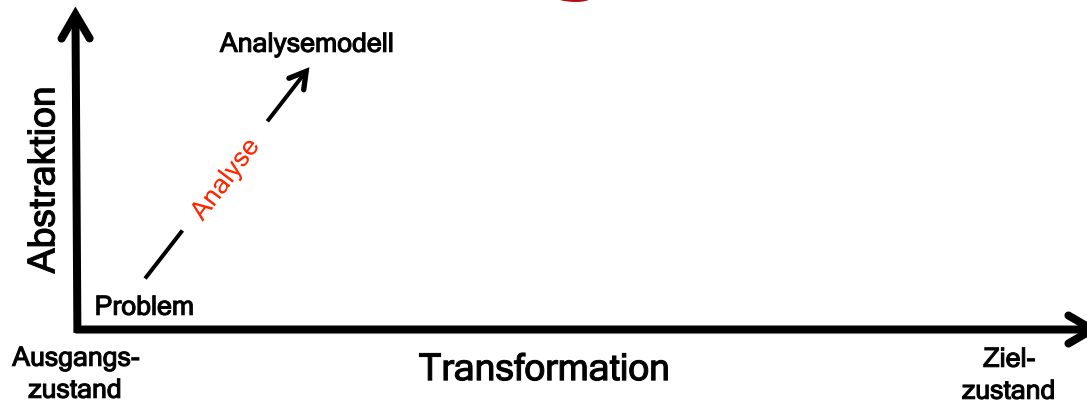


Standardisierte angepasste Lösung



Anforderungsanalyse/Aufnahme bei Softwareentwicklung

Überwindung der Semantischen Lücke

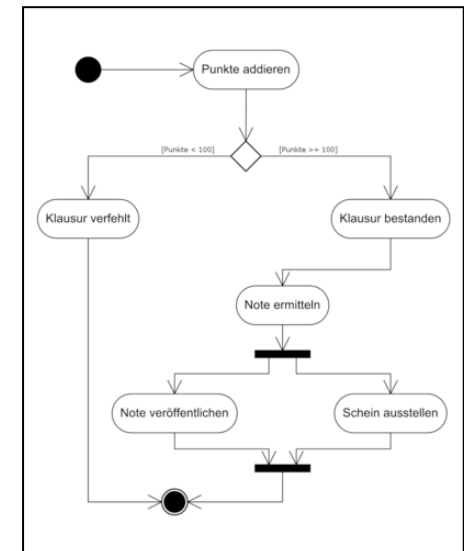
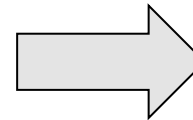


Methodische Unterstützung

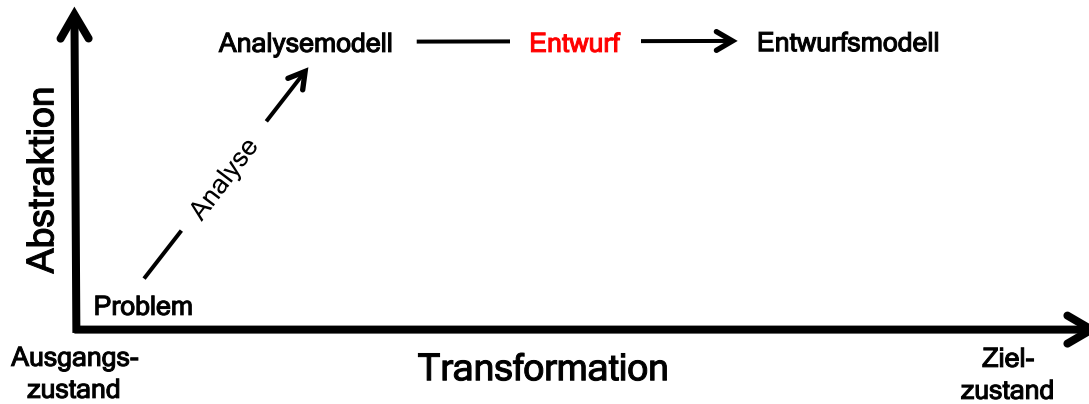
1. Absenkung der Anforderungsebene

→ Verwendung einer (semi-)formalen Sprache zur Anforderungsformulierung

„Bewerte Klausur“



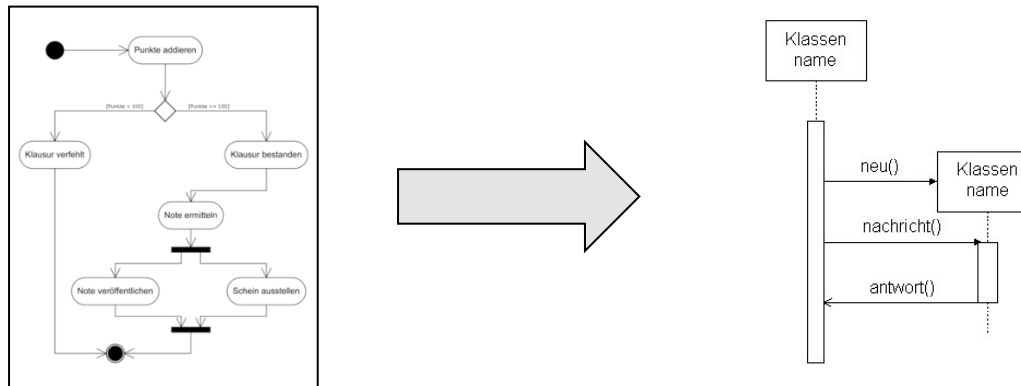
Überwindung der Semantischen Lücke



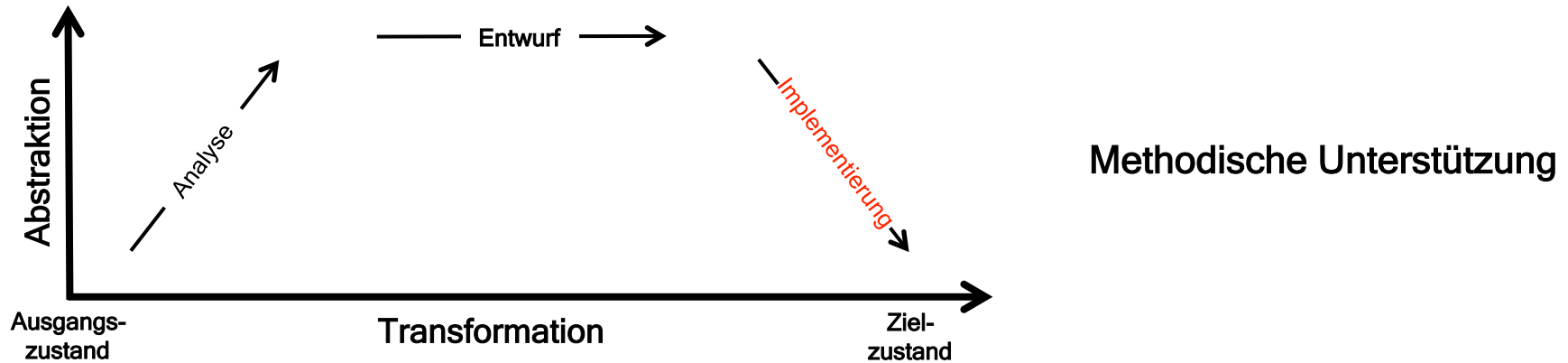
Methodische Unterstützung

2. Transformation der Analysemodelle

→ Verwendung einer (semi-)formalen Sprache zur Lösungsbeschreibung



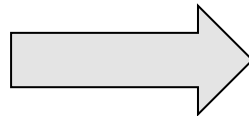
Überwindung der Semantischen Lücke



3. Anhebung der Programmiersprache und Implementierung

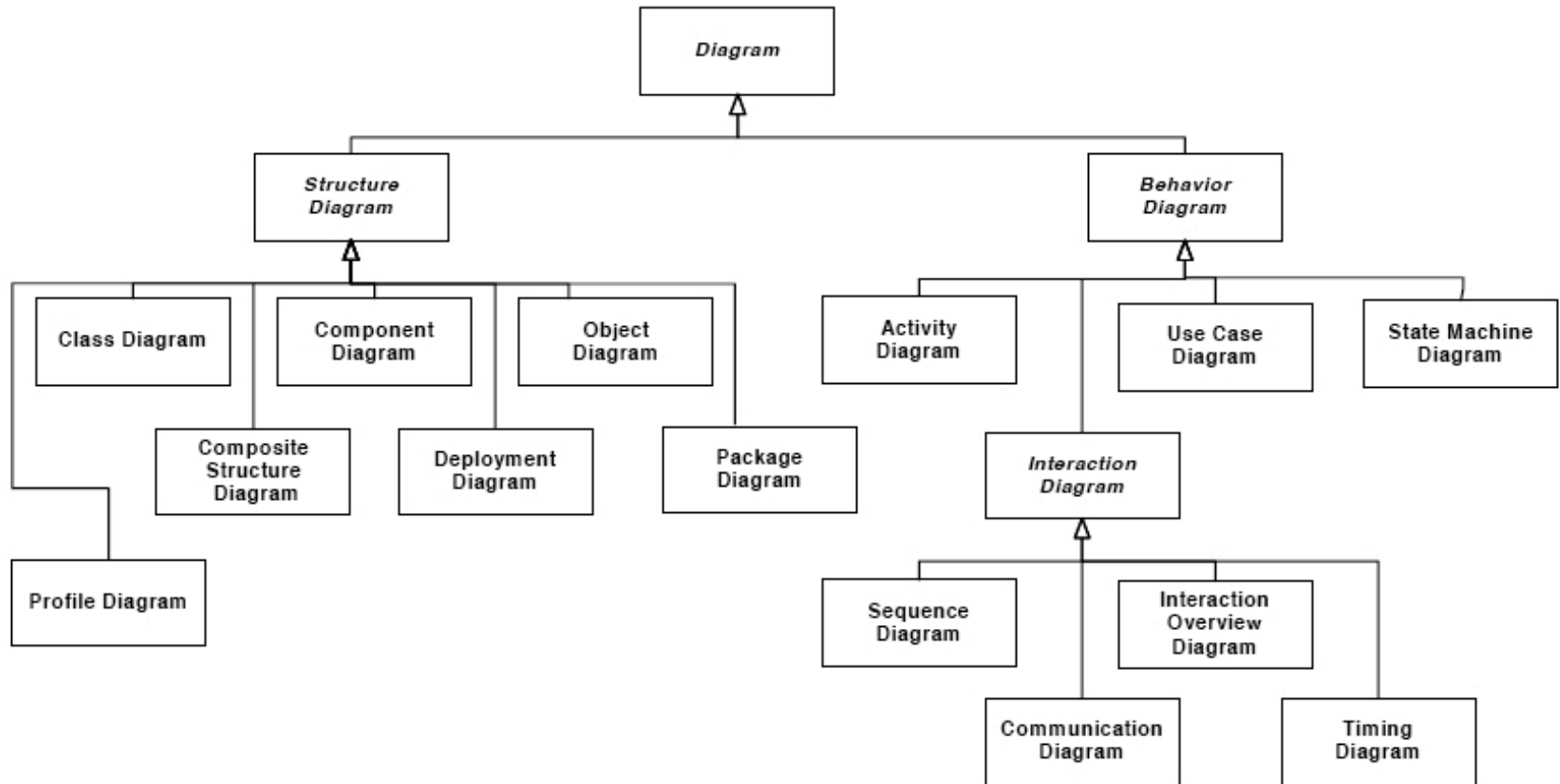
→ Verwendung einer höheren Programmiersprache statt Maschinsprache

```
010011101010
101001110101
011110101011
110011101110
```

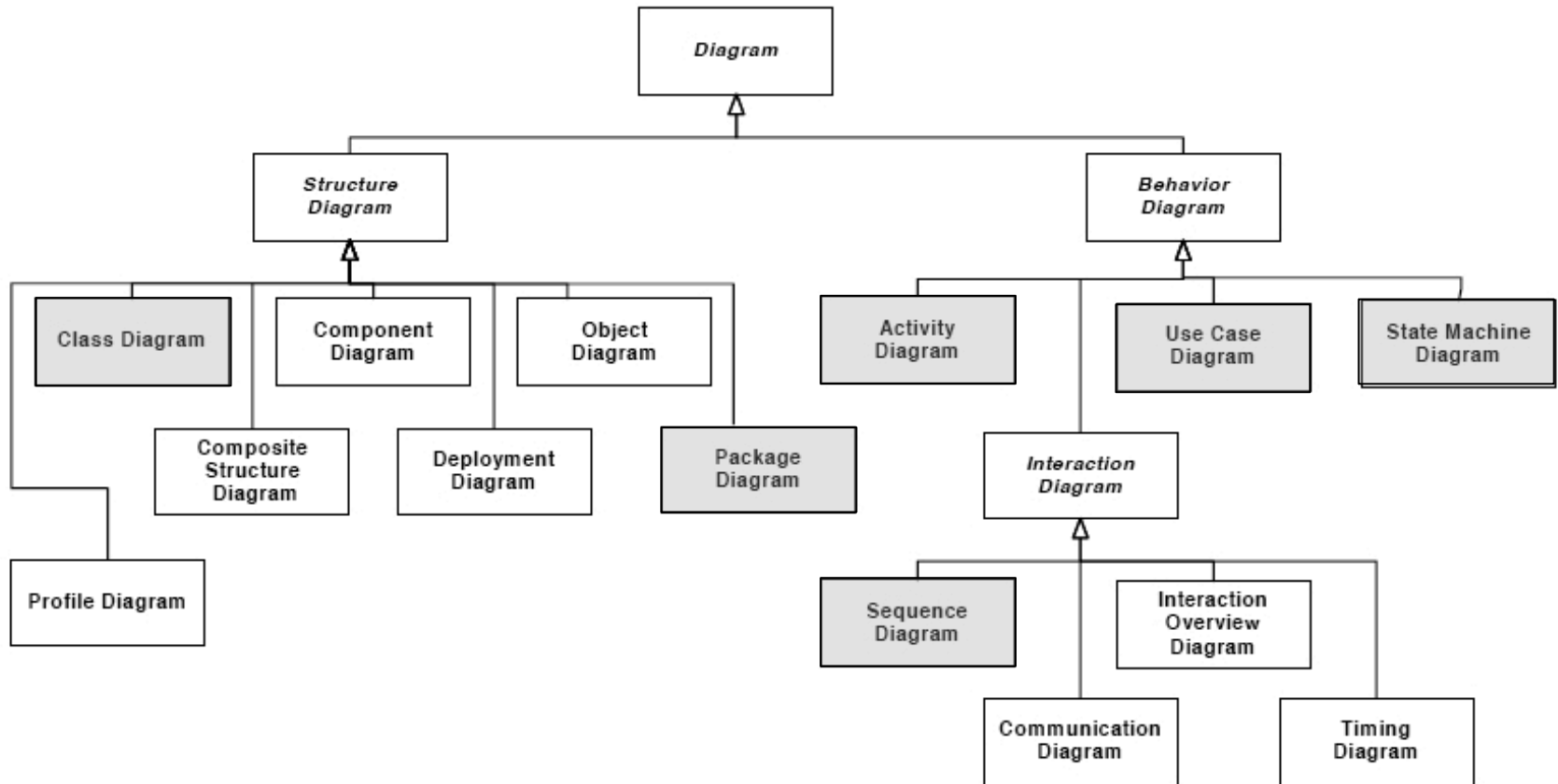


```
public void calculate(){
    int ergebnis = 1;
    for (int i = 1; i < 11; i = i+1) {
        ergebnis = ergebnis * i;
    }
    System.out.println("Ergebnis: " + ergebnis);
}
```

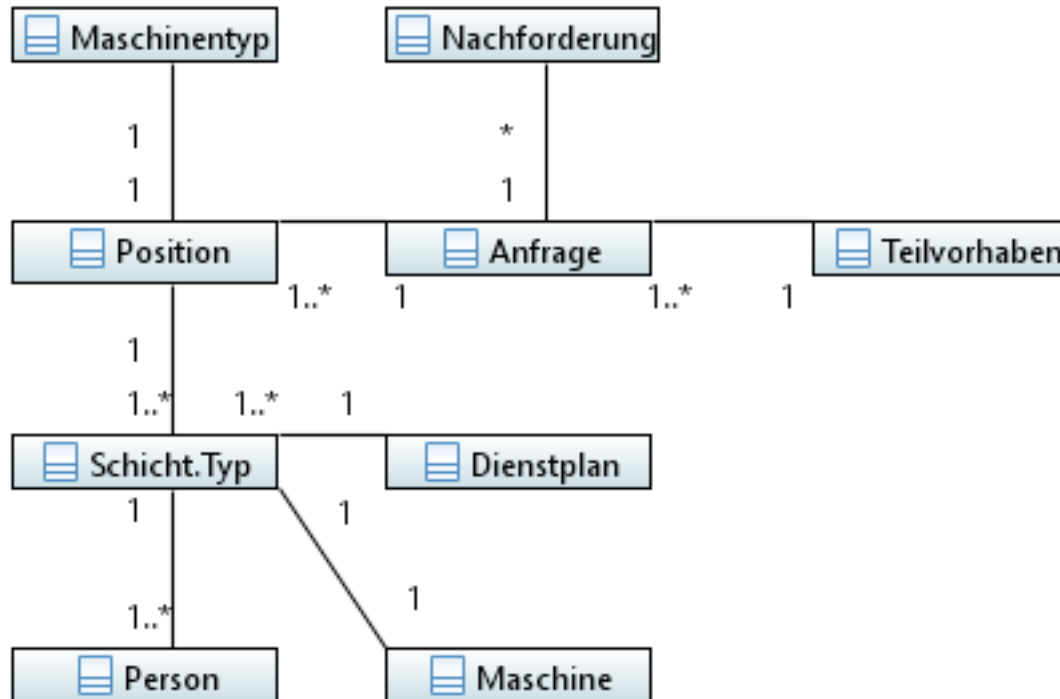
UML 2 (Unified Modeling Language)



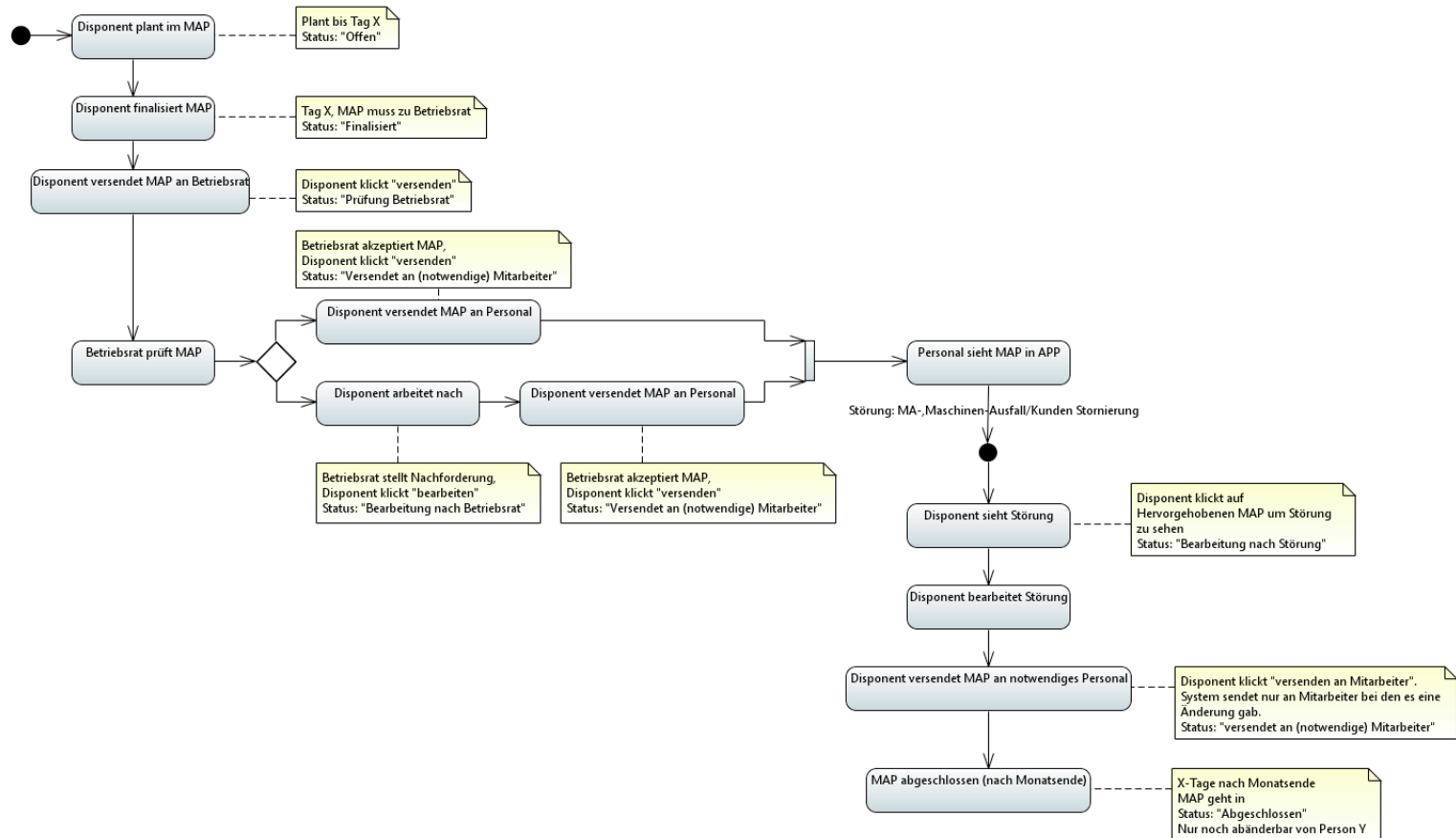
UML 2 (Unified Modeling Language)



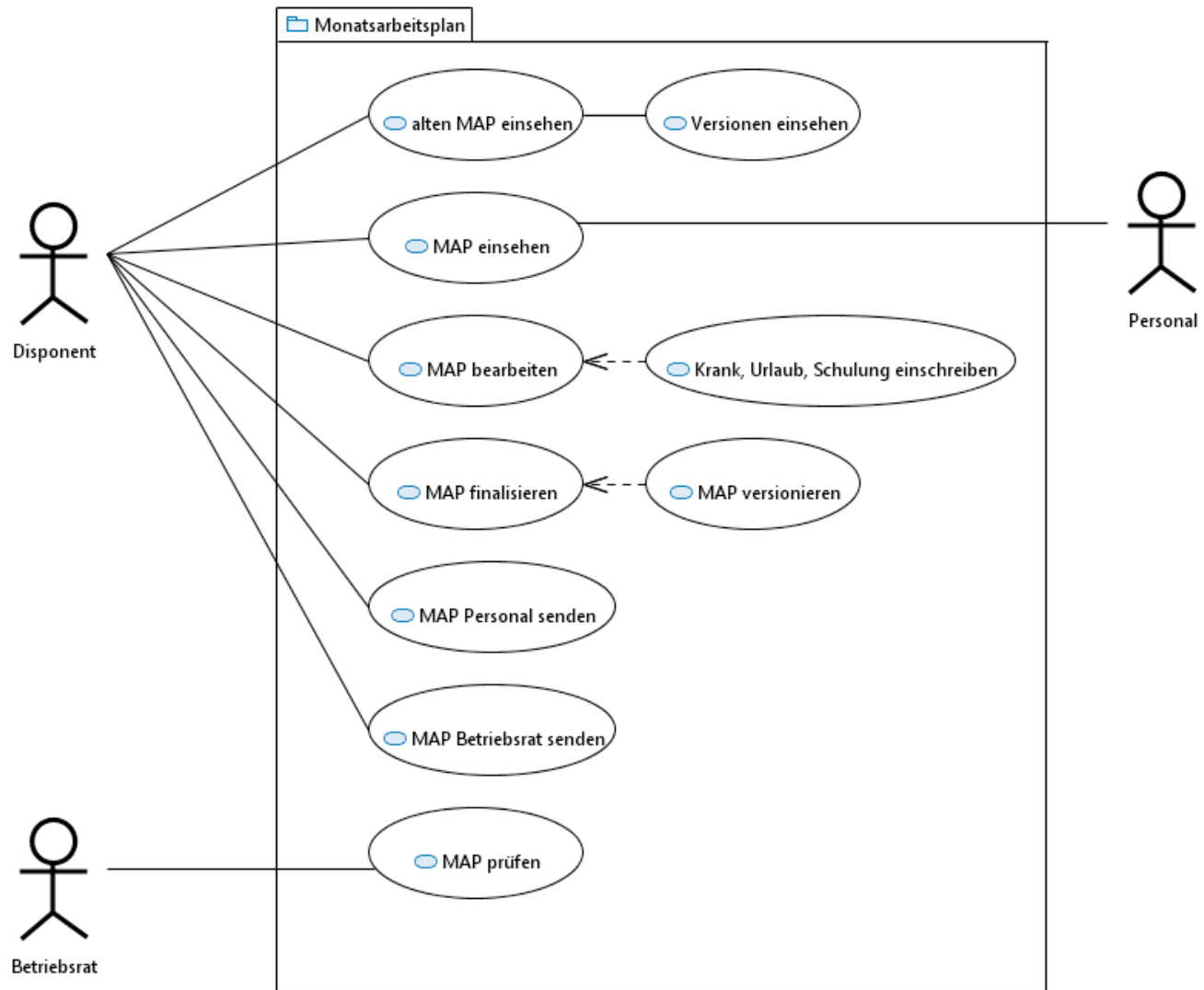
Beispiel Klassendiagramm



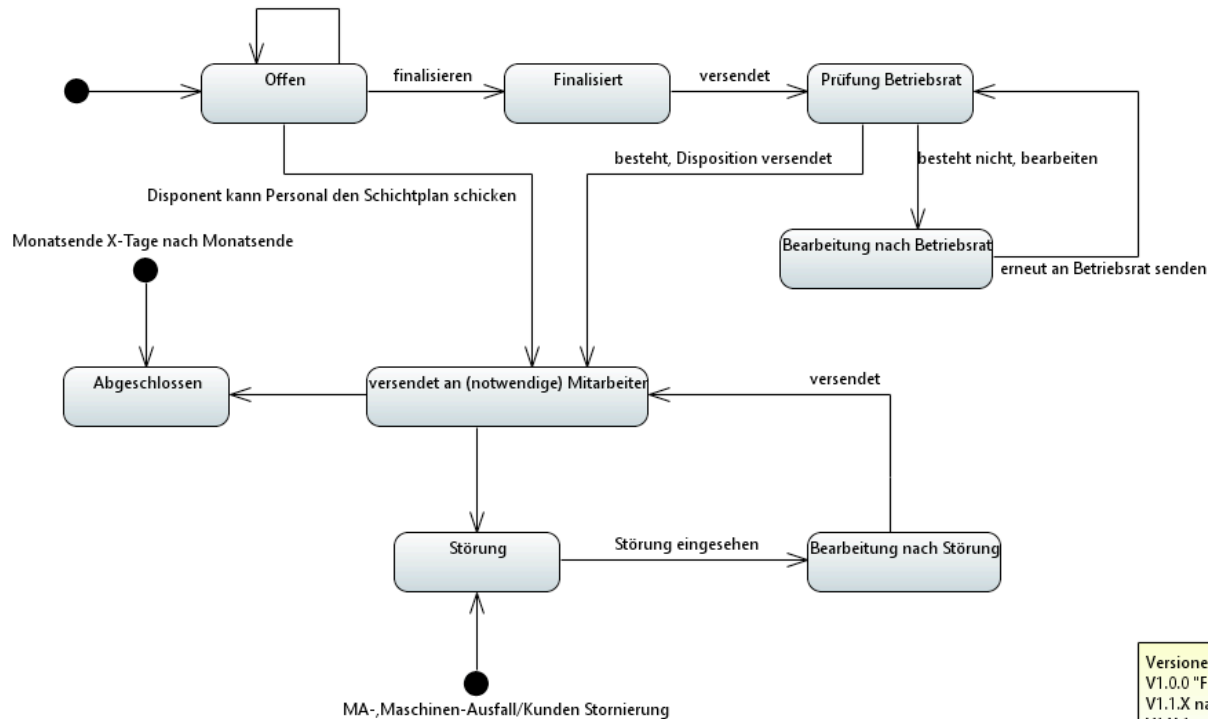
Beispiel User Story Dienstplan



Beispiel Use Case Dienstplan



Beispiel Zustandsdiagramm Dienstplan

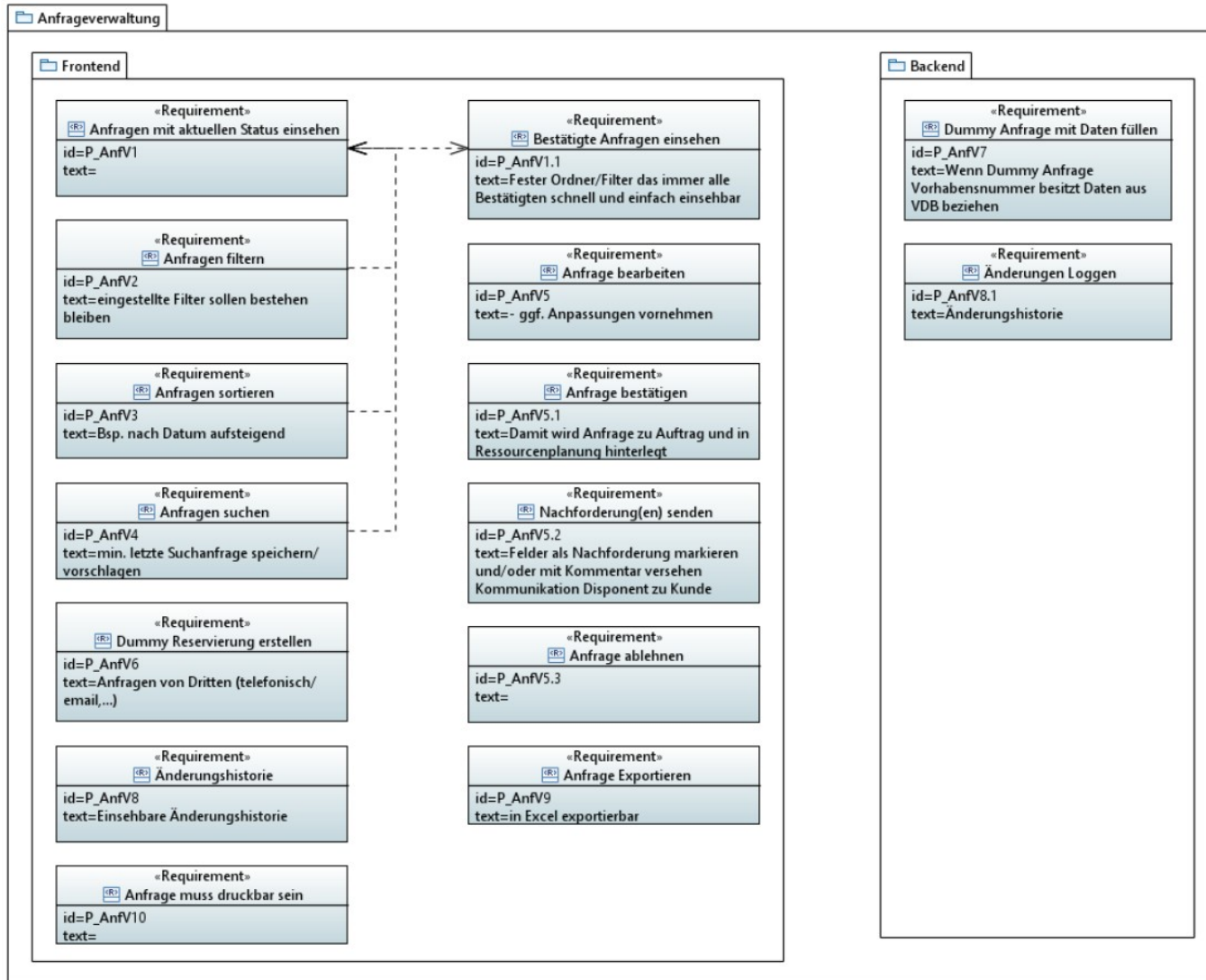


Versionen:
V1.0.0 "Finalisiert"
V1.1.X nach "Prüfung Betriebsrat"
V1.X.1 nach "Bearbeitung nach Störung"

SysML (Systems Modeling Language)



Beispiel Requirement Modellierung

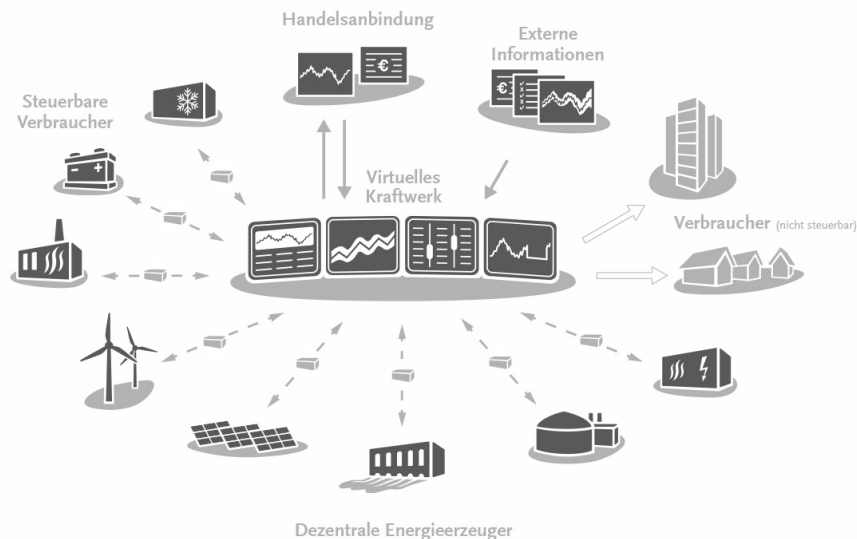
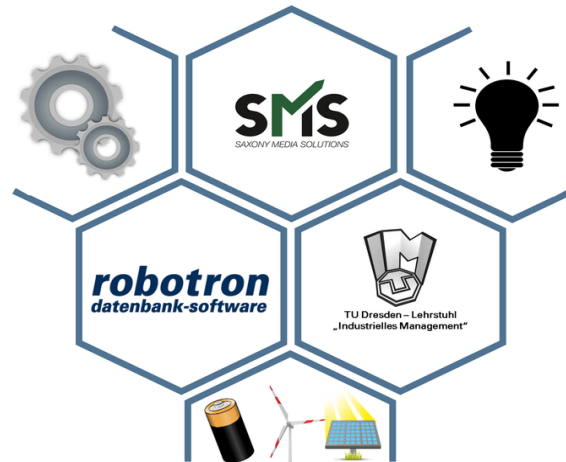


OCL (Object Constraint Language)

- Ergänzt die Unified Modeling Language (UML)
- Formale Sprache für die Definition von Constraints auf UML-Modellen
- Wie UML und SysML standardisiert

Dino4Energy

- Digitale Integration und Optimierung von Energieerzeugung, -verbrauch und -speicherung
- Entwicklung Optimierungsansätze
 - für eine effiziente Anlagensteuerung
 - und eine gewinnmaximierende Vermarktung



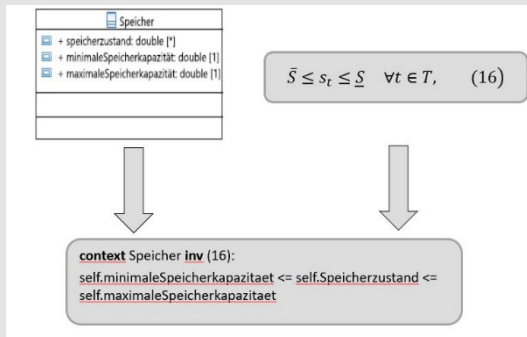
- Echtzeit-Energiemanagement
- Optimierung **virtueller Kraftwerke**
 - Zusammenschaltung von dezentralen Stromerzeugungseinheiten

Optimierung / Solver

Metaheuristik- Solver

Integer-Problem

- Optaplanner
- Objektorientiert
- z.B. OCL (Object Constraint Language)



Mathematische Optimierungssoftware

Mixed-Integer-Problem

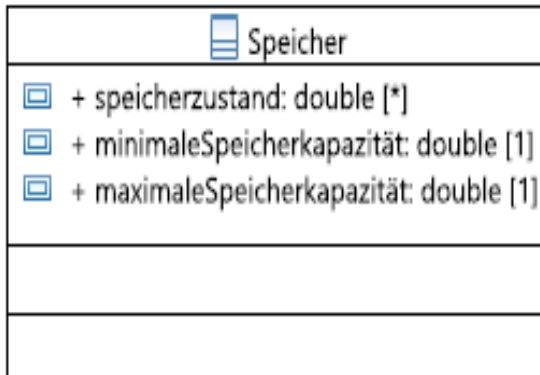
- Gurobi
- CPLEX
- Coin-OR

Lineares Problem

- Mathematische Formulierung (LP-Programmierung)

$$\begin{aligned}
 y_t - y_{t-1} &\leq \bar{V} v_{t-1} + \bar{U} u_t, & \forall t \in \mathcal{T}, \\
 y_{t-1} - y_t &\leq \underline{V} v_t + \underline{W} w_t, & \forall t \in \mathcal{T}, \\
 y_0 &= Y_0.
 \end{aligned}$$

Beispiel OCL

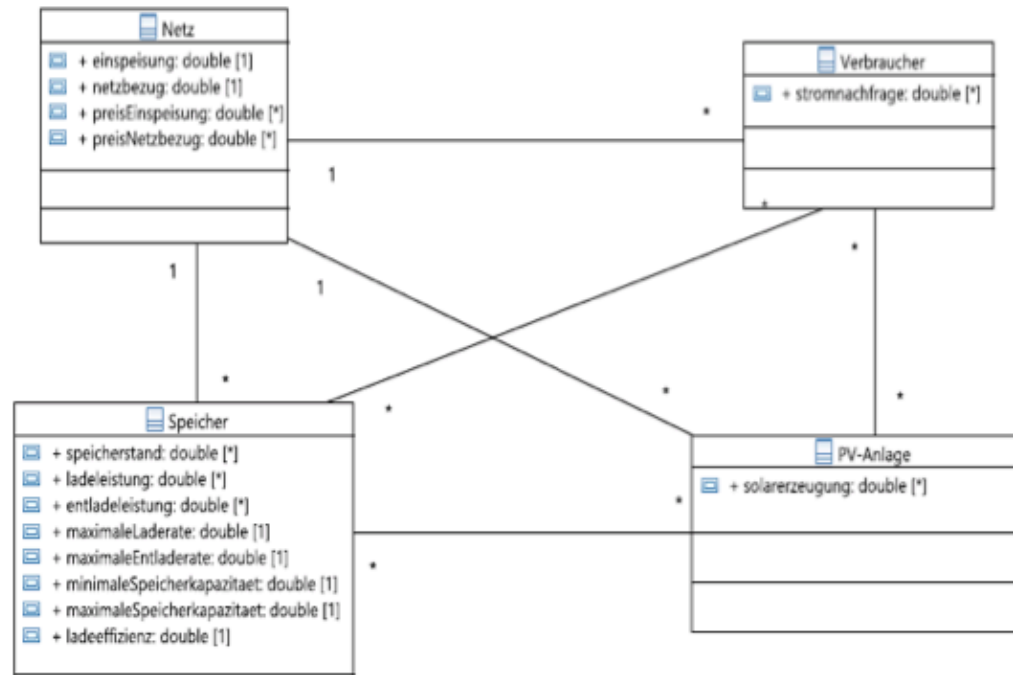


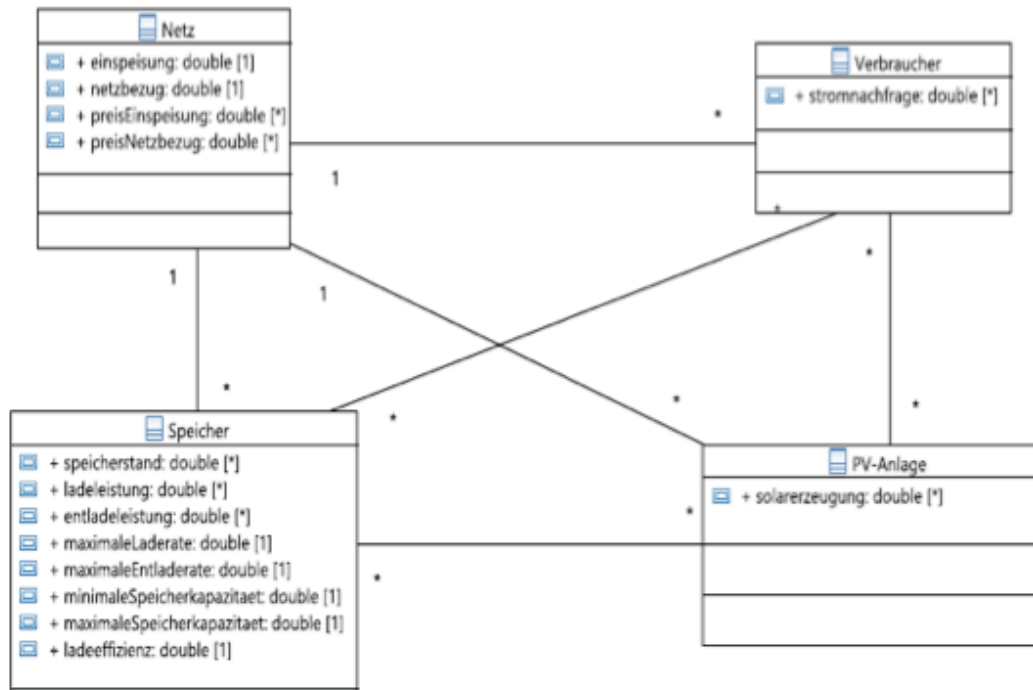
$$\bar{S} \leq s_t \leq \underline{S} \quad \forall t \in T, \quad (16)$$

context Speicher inv (16):

self.minimaleSpeicherkapazitaet <= self.Speicherzustand <=
self.maximaleSpeicherkapazitaet

Beispiel OCL



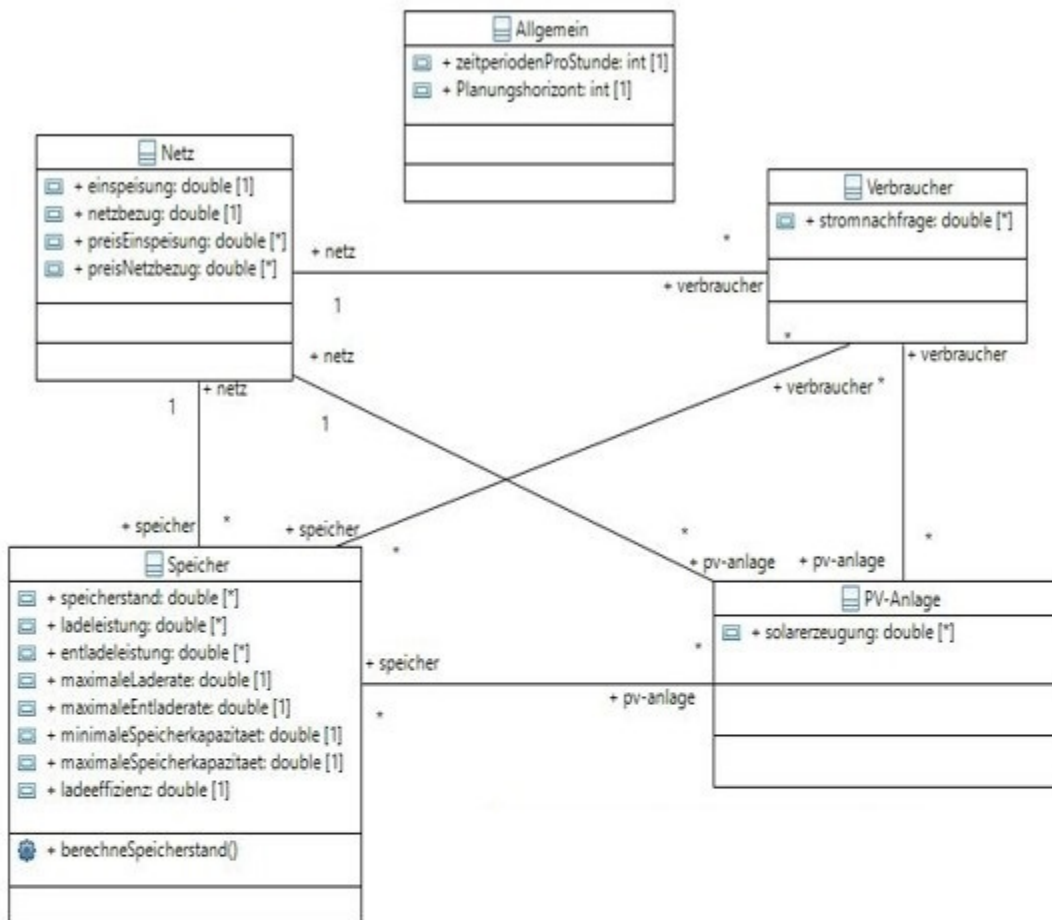


context PV-Anlage **inv** (21):

self.solarerzeugung + speicher.ladeEffizienz*speicher.ladeleistung + netz.netzbezug =
netz.einspeisung + speicher.entladeleistung + verbraucher.stromnachfrage



$$Y_t^S + E^d y_t^d + x_t^- = x_t^+ y_t^c L_t, \quad \forall t \in T. \quad (21)$$



Context Speicher inv (16):
 self.minimaleSpeicherkapazitaet <= self.speicherzustand <= self.maximaleSpeicherkapazitaet
 context Speicher inv (10):
 0 <= self.ladeleistung <= self.maximaleLaderate
 context Speicher inv (11):
 0 <= self.entladeleistung <= self.maximaleEntladerate
 context Speicher inv (19):
 self.ladeleistung <= self.maximaleLaderate - (self.maximaleLaderate / self.maximaleEntladerate) * self.entladeleistung

context Netz inv (5):
 0 <= self.netzbezug
 context Netz inv (6):
 0 <= self.einspeisung

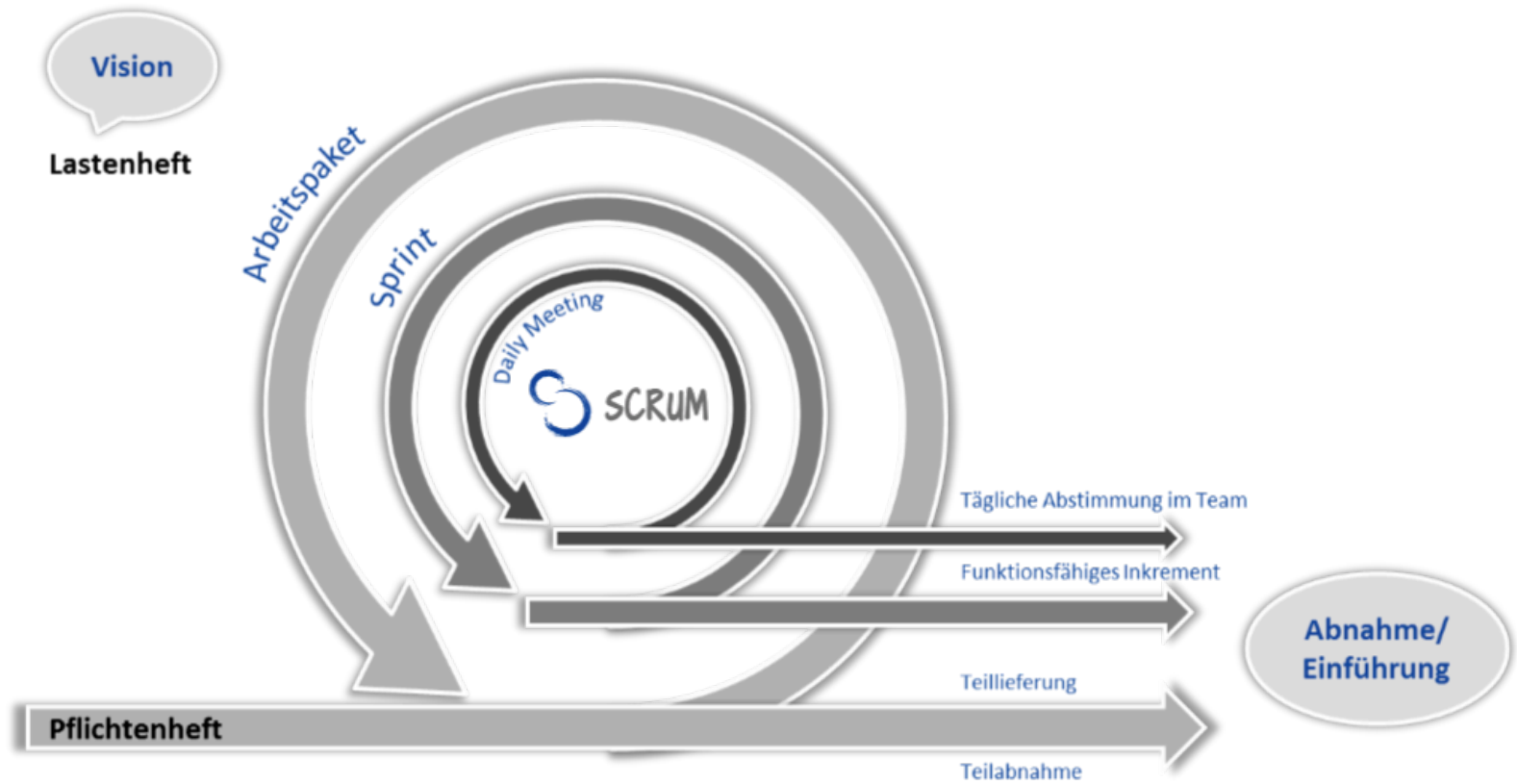
context Netz inv (ZF):
 maximize (self.einspeisung * self.preisEinspeisung --> sum() - self.netzbezug * self.preisNetzbezug --> sum())

context Speicher::berechneSpeicherstand(): Double
 post: speicherstand = speicherstand@pre + self.ladeleistung - self.entladeleistung

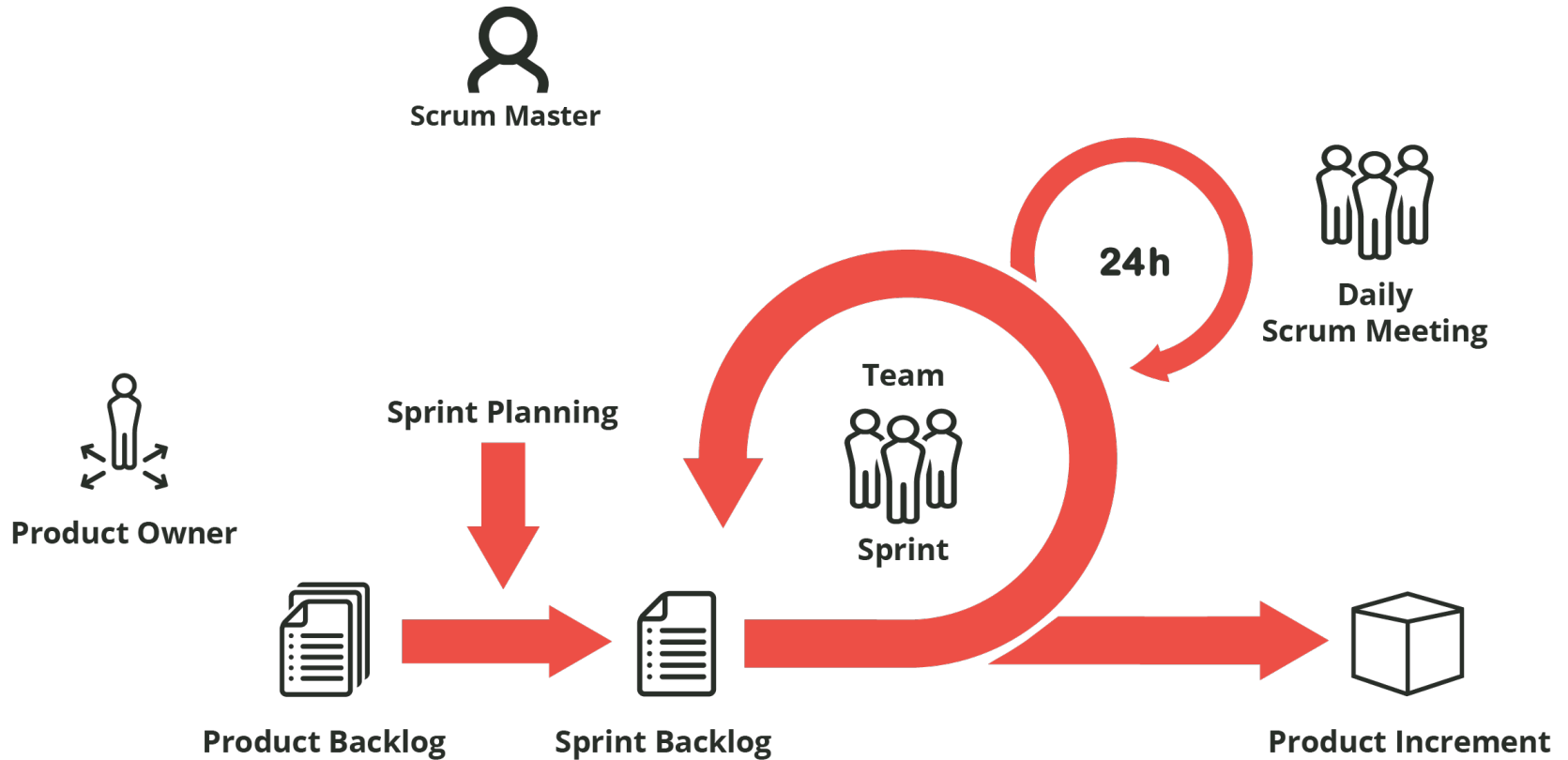
context PV-Anlage inv (21):
 self.solargerzeugung + speicher.ladeEffizienz * speicher.ladeleistung + netz.netzbezug = netz.einspeisung + speicher.entladeleistung + verbraucher.stromnachfrage

Projektmanagement bei Softwareentwicklung

SCRUM



SCRUM



Planung in der Produktion

	Bereich	Verhalten	Horizont
Produktionsplanung	Produktionsprogrammplanung	Welche Produkte sind in welchen Mengen zu produzieren	Strategisch
	Bereitstellungsplanung	Bedarf an Elementarfaktoren für geplante Produkte festlegen	
	Produktionsprozessplanung	Zeitliche, mengenmäßige und räumliche Planung des Produktionsvollzugs	Taktisch
Produktion Steuerung	Auftragsfreigabe Feinterminierung Betriebsdatenerfassung	Aufträge für die Produktion freigeben Ressourcenbereitschaft für die Aufträge überprüfen Aufträge genau einordnen	Operativ

Planung in der Softwaretechnik

	Bereich	Verhalten	Horizont
SW-Entwicklung Planung	Projektplanung	Umzusetzende Meilensteine und dafür zu erfüllende Aufgaben in grober Detaillierung	Strategisch
	Monatsplanung	Planung der kommenden Wochen auf mittlerem Detaillevel um Zielvorgaben zu erreichen	Taktisch
	Sprintplanung	Welche Aufgaben sind in welchem Zeitraum von wem mit welchem Aufwand umzusetzen	Operativ
SW Entwicklung Steuerung	Tägliche Umsetzungsplanung (Daily)	Absprache welche Aufgaben über den Tag wie umgesetzt werden und welche Schnittstellen zwischen den entwickelnden Personen bestehen.	

Vorgeschlagene Projektdurchführung

Bosch	Sebastian Wilhelm Oliver Paul			Workflow Backup																	
				Eventtage	Order	Order															
EFW	Elias Martin P.																				
Bahn		BB Kunden PF																Dispo 0.1		angebundene App.	
	Rico			LongPolling, Template Ref.	Template Ref., Rico K. Template	Dispo	Dispo	Dispo	Dispo	frei	Dispo	Dispo	Dispo	frei	Austauschformat App., Konsistenz Dispo/KP	Abrechnung, Bautageberichte	Rechnungserstellung	Umplanung			
	Sandy			Lernen Templates	Lernen Templates	Dispo	Dispo	Dispo	Dispo	Dispo	Dispo	Dispo	Dispo	Validierung	Validierung, Abrechnung	automatische Benachrichtigung	Umplanung / FM				
Ing	Oliver Franz																				
				Konzpt Datenmodell	Konzpt Planung Dispo Pflichtenheft	Dispo Pflichtenheft	frei	frei		Testprotokoll BB Dispo	Konzept App BB App Pflichtenheft	App Pflichtenheft	frei	frei	Konzept Umplanung						
BE	Max																				
				LiquiBase	LiquiBase, ViewTables APS	ViewTables APS	ViewTables APS														
	Dat																				
	Domenik	BA	BA	BA	BA	BA	BA	BA	BA	TestBahn a&m	TestBahn a&m	TestBahn a&m	TestBahn a&m	TestBahn a&m	TestBahn a&m	TestBahn a&m	TestBahn a&m	TestBahn a&m			
FE	Sebastian Gregor					BB Kunden PF				FE V 0.9										FE V1.0	
			Bahn BF	ZEDAS APP CR	ZEDAS APP CR, Longpolling	Longpolling, Testintegr. Jenkins, Virt. Scroll	Bugfix														
			Tabellen	Tabellen	Tabellen	Tabellen	Tabellen, Charts	Charts	Charts	Charts				App BB	App BB	App BB	App BB	App BB	App BB		
			High Lvl API Gantt	Gugh Lvl API, Bahnbau EFW Features	Bahnbau	Bahnbau	APS														
			Gantt				SD Dispo BB	SD Dispo BB	SD Dispo BB				SD App BB				SD Umplanung BB	SD Umplanung BB			
		KW1 Januar	KW2	KW3	KW4	KW5	KW6 Februar	KW7	KW8	KW9	KW10 März	KW11	KW12	KW13	KW14 April	KW15	KW16	KW17			

Vorgeschlagene Projektdurchführung



Vorgeschlagene Projektdurchführung

14.01. - 18.01. 4 **AKTIV**

3 1 0

...

   ... 14/Jan/19 3:50 PM • 18/Jan/19 6:00 PM [Verknüpfte Seiten](#)

  BOSCH40-87 Workflows from Hist	1.2 HDA 
  BOSCH40-114 Propagate "Delete model" to Models-Db	1.1 Database 
  BOSCH40-42 Implement Connection to Oracle DB	1.1 Database 
  BOSCH40-21 Factory Calendar Analysis	1.2 HDA 

21.01. - 25.01. 6

Sprint starten

...

   ... [Verknüpfte Seiten](#)

  BOSCH40-41 Refactor Database Connection Class	1.1 Database 
  BOSCH40-68 Build project as executable (exe)	1.1 
  BOSCH40-48 Implement Authentication checker	1.2 REST 
  BOSCH40-97 Factory Calendar Import	1.2 APS 
  BOSCH40-101 Use start and end date for BlockingPlan from result	1.2 APS 
  BOSCH40-113 New Hist-Order Concept	1.1 HDA 

+ Vorgang erstellen

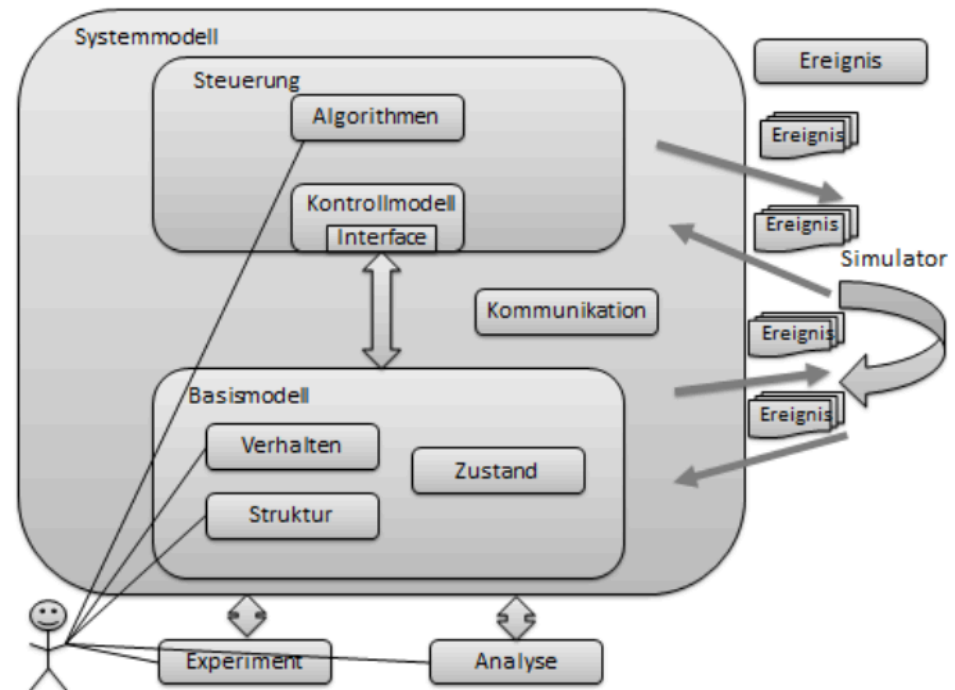
Praxisbericht

Entwicklung unserer Softwareentwicklung

Vorentwicklungen TU Dresden 2005 - 2015

OptimSim

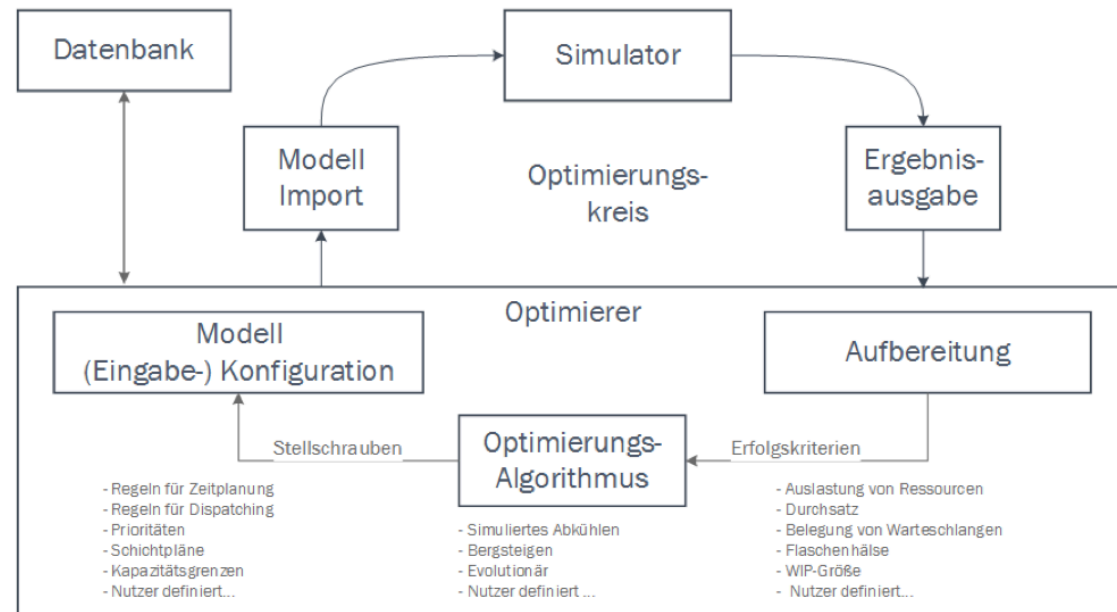
- Simulationsbasiertes Optimierung
- Werkzeug für Prozesse der Produktion und Logistik
- Langjährige Entwicklung an der TU Dresden (Angewandte Informatik) und UniBw München
- Flexibles Modellierungskonzept, überdurchschnittlich schnell und integrierter simulationsbasierter Optimierer
- Forschungsrelevantes Werkzeug



Vorentwicklungen TU Dresden 2005 - 2015

OptimSim

- Simulationsbasiertes Optimierung
- Werkzeug für Prozesse der Produktion und Logistik
- Langjährige Entwicklung an der TU Dresden (Angewandte Informatik) und UniBw München
- Flexibles Modellierungskonzept, überdurchschnittlich schnell und integrierter simulationsbasierter Optimierer
- Forschungsrelevantes Werkzeug

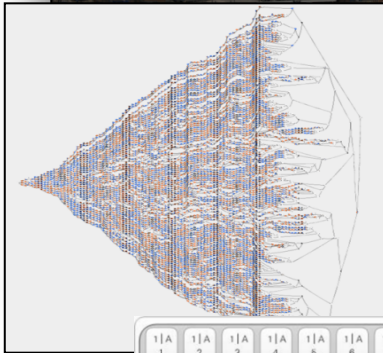


Zur Gründungsidee 2014/2015

Existierende Projektidee durch TU Dresden

- Elbflugzeugwerke Dresden
- Automatische Planung und Steuerung von Produktionsschritten
- Hohe Flexibilität durch Plug-In basierter Simulation

OPTIMIERUNG: Flugzeugwartung

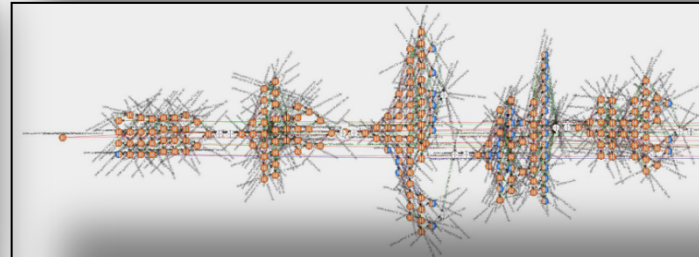


10²⁰⁰⁰ Lösungsvarianten!



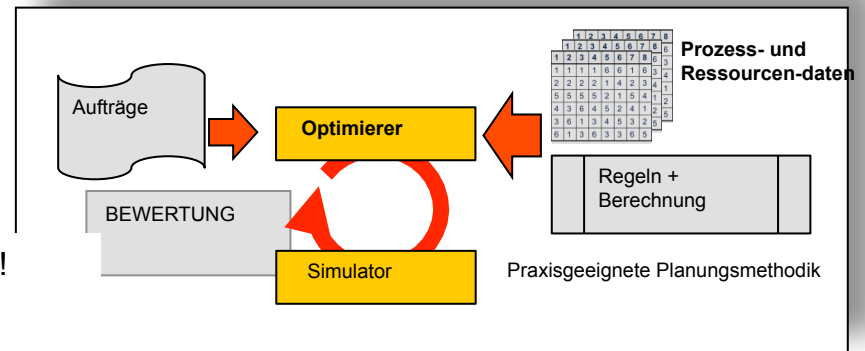
Lösungschromosom!

*Das Ziehen der Zufallszahlen erfolgte mittels des Zufallszahlengenerators von MS Excel 2007.



Gesamtkostenzielfunktion:

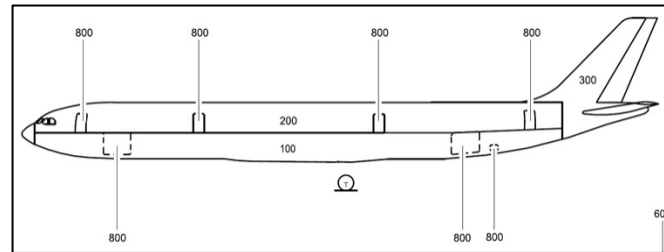
$$K(T) = \left\{ \sum_{i=1}^I \sum_{j=1}^{J_i} \sum_{k \in K} \sum_{t=1}^T \sum_{m=1}^{M_{ij}} [P_{ij} + (c_k^r D_{ijm}^r + c_k^g D_{ijm}^g + c_k^t D_{ijm}^t) r_{ijmk}] x_{ijmt} + z \sum_{i=1}^I \sum_{t=1}^{T-1} W_{it} \right\} + \left\{ \sum_{t=U_1}^{U_I} \sum_{i=1}^I v_i \left[\frac{U_i - g_i}{n} \right] x_{ijmt} - \sum_{t=U_1}^{U_I} \sum_{i=1}^I b_i \left[\frac{g_i - U_i}{n} \right] x_{ijmt} + T * H + \left| \bigcup_{i=1}^I \bigcup_{j=1}^{J_i} t_{ij}^s \right| c_a \right\} \rightarrow \text{Min.}$$



Robuste Ressourcenplanung der Flugzeugwartung und -umrüstung mit OptimSim

Montagerestriktionen

- Arbeitsräume begrenzen
Parallelarbeit (Bsp. Cockpit A330: max. 3 Mitarbeiter)
- OptimSim modelliert Bauräume je Projekt und Flugzeugtyp



Auftragsfreigabe

- Hangar-Stellplätze begrenzen
parallel bearbeitbare Projekte
- OptimSim priorisiert Projekte und steuert deren Einlastung in die Produktion



Robuste Ressourcenplanung der Flugzeugwartung und -umrüstung mit OptimSim

Gründung Dezember 2015

- Erstellung Businessplan:
 - Form?
 - GmbH? GBR?
 - Name?
 - Finanzierung?
 - EXIST? Eigenkapital? Sächsische Aufbaubank? Technologiegründer?
 - Finanzplanung? Liquidität? Rentabilität?
 - Kundenkontakte?
 - Markt? Zugänge?
 - Team?
 - Größe Beginn? Entwicklung? Organisation?

Zur Gründungsidee 2015

Existierende Projektidee durch TU Dresden

- Elbflugzeugwerke Dresden
- Automatische Planung und Steuerung von Produktionsschritten
- Hohe Flexibilität durch Plug-In basierter Simulation

Verfügbares Team

- 2 Personen
- + 1-2 studentische Kontakte als potentielle Frontendentwickler
- + extern 1 MA - Ingenieur TU Dresden

Entwicklung erster Geschäftsmodelle

Vertrieb von Simulationssoftware

- Direkter Verkauf an Anwender
- Direkte produzierende Kunden oder Berater
- Identifikation und Kontaktaufnahme zu potentiellen Anwender schwierig



Koppelung mit Partnern

- Erweiterung von Planungssoftware um automatische Planung
- Identifikation und Kontaktaufnahme zu potentiellen Partnern ohne Basis schwierig



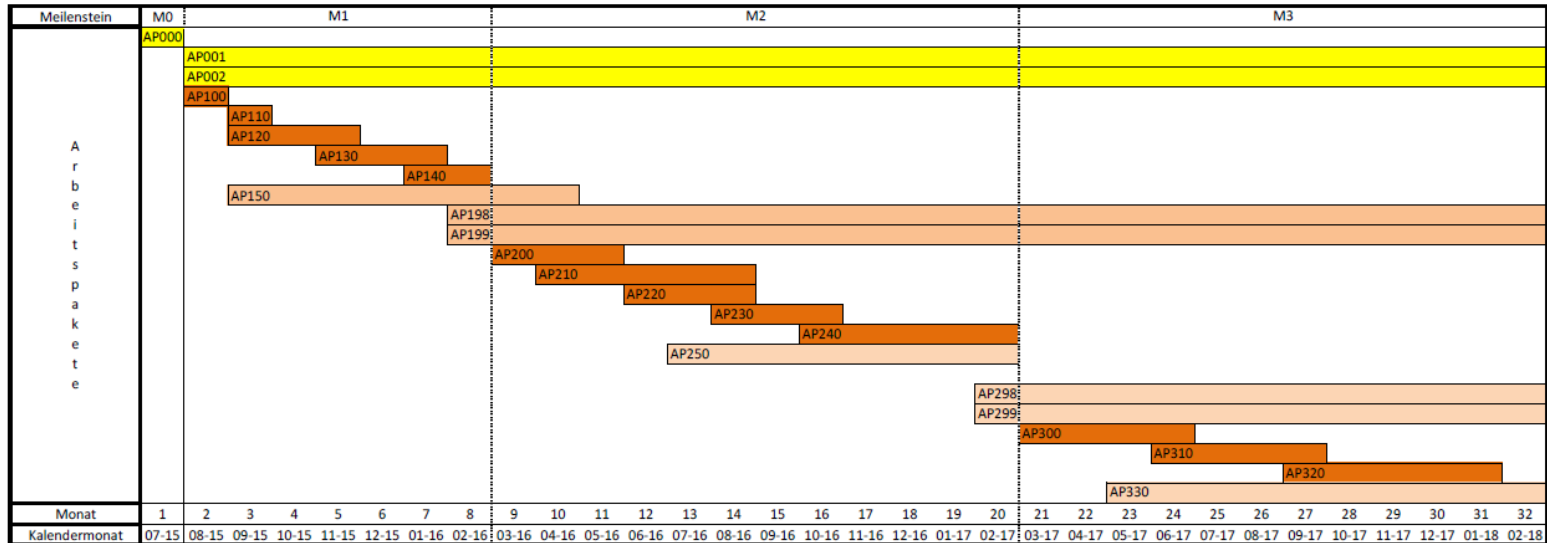
Beratung von Unternehmen

- Aufnahme von Produktionsprozessen
- Simulation derer
- Analyse/Auswertung
- Empfehlung geben
- Erste Vorbetrachtungen vielversprechend



Gründung Dezember 2015

- **Produkt-Entwicklungsplan (1-3 Jahre)**
 - **Klassischer Wasserfallplan nach bestem Wissen erstellt**



Farbskala:	
Organisatorisches zum Geschäftsbetrieb:	Yellow
Marketing, Vertrieb, Kundenservice:	Orange
Entwicklung mit FuE-Anteil:	Light Orange

- **Eingetragene GmbH seit 04.01.2016**

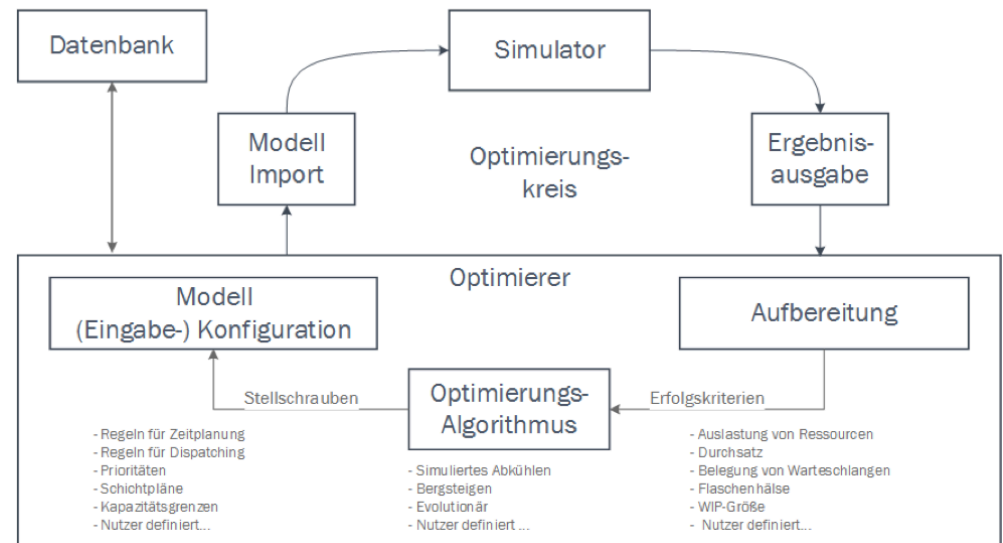
Ausentwicklung Prototyp

Iteration 1

Ausentwicklung Prototyp

Bestandsaufnahme Software

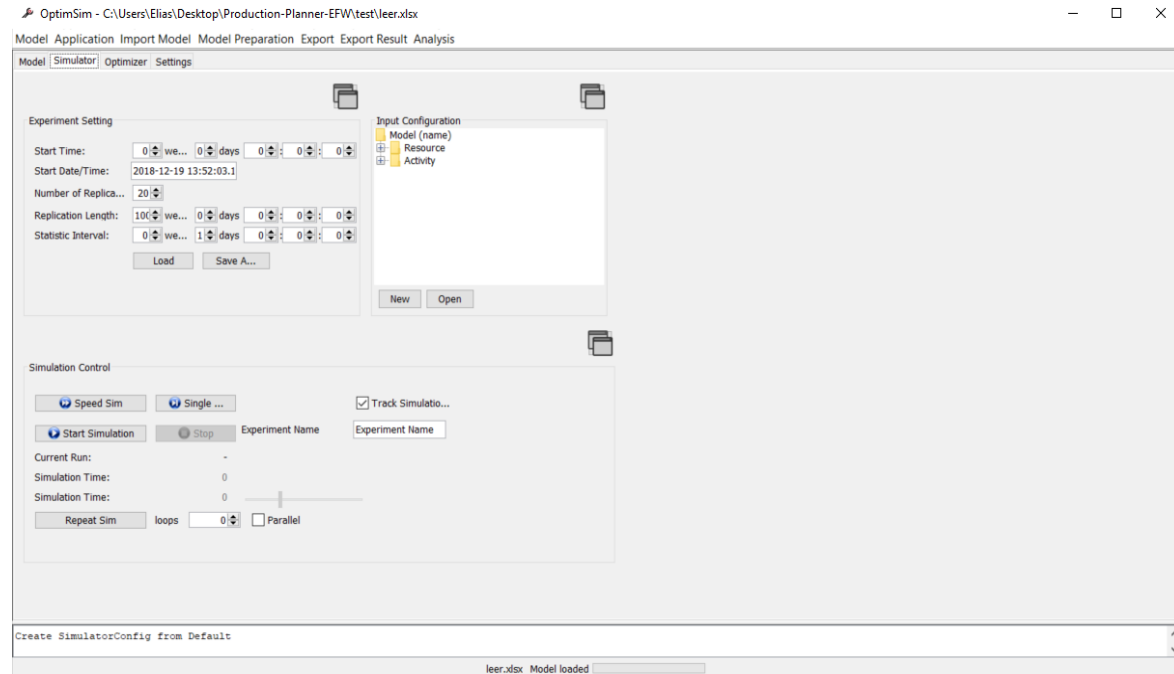
- Simulator
 - Algorithmen in Java
- Optimierer
 - Algorithmen in Java
 - Erlaubt Schlussfolgerungen über „gute“ und „schlechte“ Entscheidungen



Ausentwicklung Prototyp

Bestandsaufnahme Software

- Simulator
 - Algorithmen in Java
- Optimierer
 - Algorithmen in Java
 - Erlaubt Schlussfolgerungen über „gute“ und „schlechte“ Entscheidungen
- Oberfläche
 - Prototypische Bedienoberfläche
 - universitär geprägt,
 - ohne konkrete UseCases entwickelt
 - Native Java Anwendung mit Excel, XML und JSON Schnittstellen (Datenimport / Export)



Ausentwicklung Prototyp

Bestandsaufnahme Software

- Simulator
 - Algorithmen in Java
- Optimierer
 - Algorithmen in Java
 - Erlaubt Schlussfolgerungen über „gute“ und „schlechte“ Entscheidungen
- Oberfläche
 - Prototypische Bedienoberfläche
 - universitär geprägt,
 - ohne konkrete UseCases entwickelt
 - Native Java Anwendung mit Excel, XML und JSON Schnittstellen (Datenimport / Export)
 - (Daten-)Modellerstellung und -Verwaltung mit Hilfe von Excel

Format Version	Manufacture ID	Name	Description	Time-Unit Setting	Fixed General Costs	Administrative costs	Working Days Per Week	Hours Per Working Day
1	TESTMODELL	TESTMODELL	TESTMODELL	1 H	100.00	100.00	5.0	16.0

Resource ID	Resource Type	Name	Description	Priority	Applicable Shifts	Subcontracted Resource	Corresponding Resource	Min Hiring	Sigma	AVG Process Duration	Cost Regular Time	Cost Overtime	Cost Special Shm	Cost Unused Material
1	DUMMY Resource	Workforce	DUMMY Resource	1	0	FALSE	FALSE	1	1	1	1	1	1	1

OrderID	Name	Description	Priority	EarliestStart	Delivery Date	ProductID	Name	Description	Priority	EarliestStart	LatestEnd
1	99998-99-27	99998-99-27	0	30.11.2017 00:00	27.01.2018 00:00	99998-99-27	99998-99-27	99998-99-27	0	01.01.2018 00:00	15.02.2018

Anfang 2016 Prototypenausentwicklung

Iteration 1 – hybride Übergangslösung

- Ziel: Aufbereitung der Planung und Ergebnisse für praxisrelevanten Kontext.
- Grundsatzentscheidung notwendig:
 - Native Anwendung Java basiert oder Webanwendung
- Nativ:
 - + Mögliche Weiterverwendbarkeit Bestandssoftware
 - + Vorkenntnisse vorhanden, geringe Einstiegshürde
 - - Mögliche Hürden bei späterer Integration
 - - Unkenntnis über IT Systemlandschaft bei noch nicht existierenden Kunden
- Web App:
 - + Plattformunabhängigkeit (Zugang erfordert Webbrowser)
 - - Browserkompatibilität
 - + Moderner, Zeitgemäßer Ansatz
 - (-) Neuland, mangelnde Vorkenntnisse
- Entscheidung:
 - Webanwendung



Anfang 2016 Prototypenausentwicklung

Iteration 1 – hybride Übergangslösung

Team

- 3 (1 Backend, 1 Frontend, ½ Organisatorisch, ½ Technische Beratung)

Methode / Tools

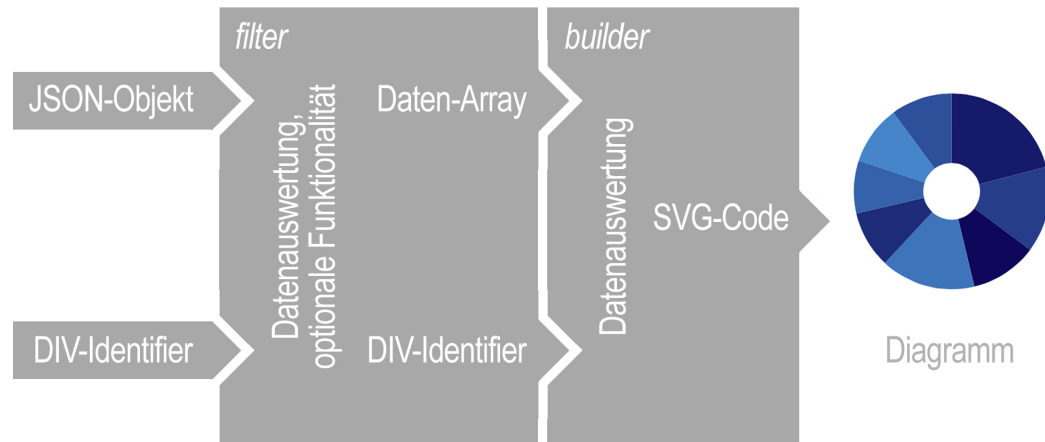
- Agile Entwicklung (Excel & Word), EXTREME Programming, SVN

Aufbau erster Frontend Elemente

- Umsetzung der Auswertungen als Balkendiagramme in eigenen Bibliotheken
- Gantt Plan zur Visualisierung von Abläufen

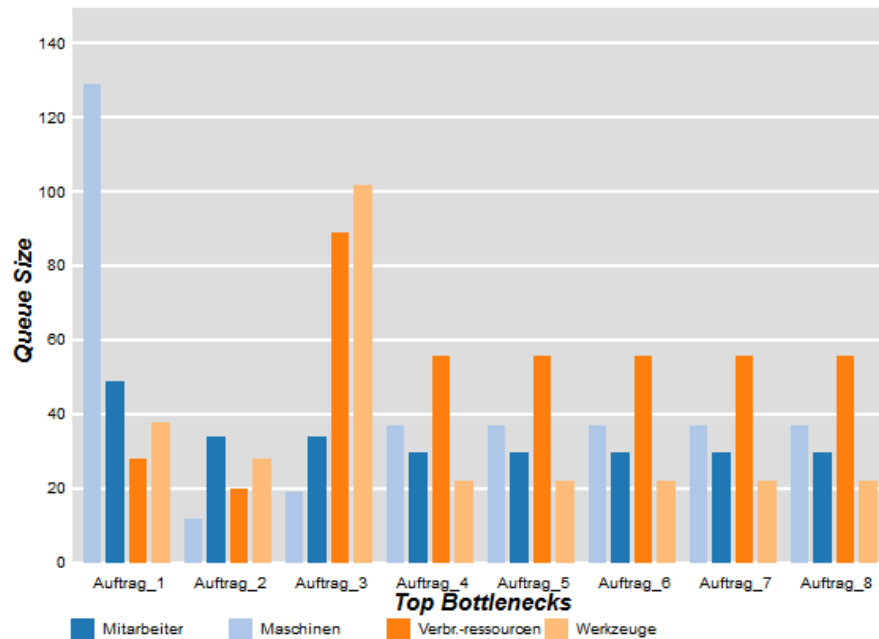
Anfang 2016 Prototypenausentwicklung

Iteration 1 – hybride Übergangslösung



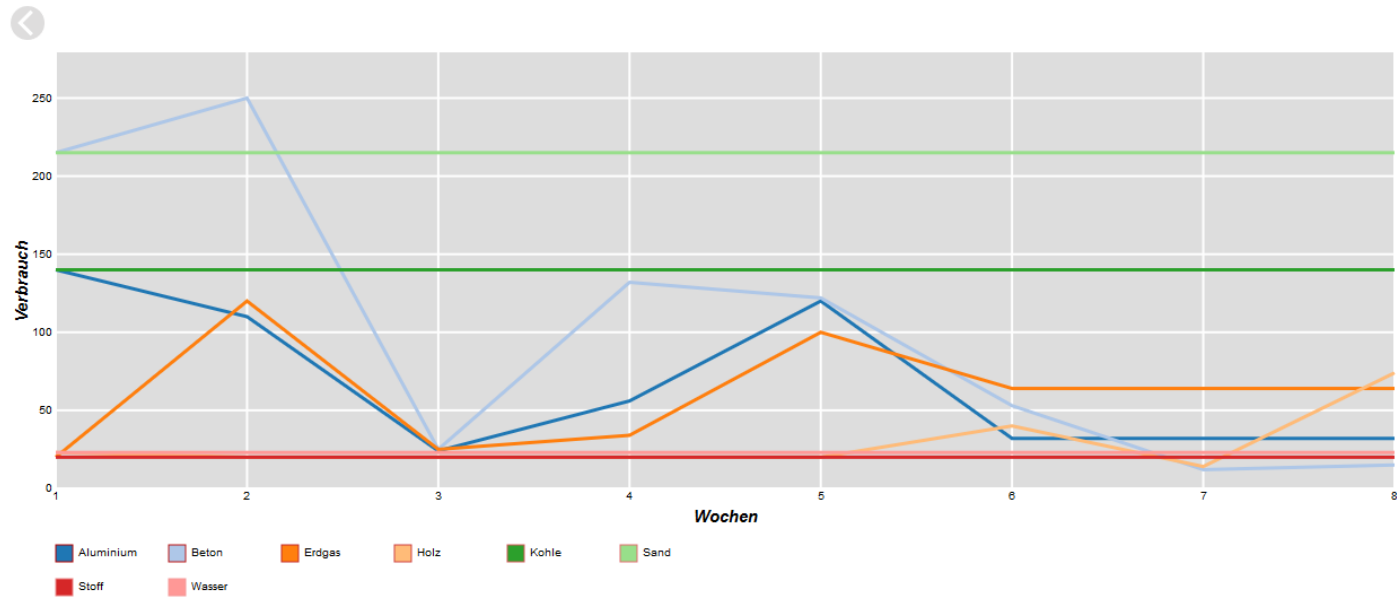
Anfang 2016 Prototypenausentwicklung

Iteration 1 – hybride Übergangslösung



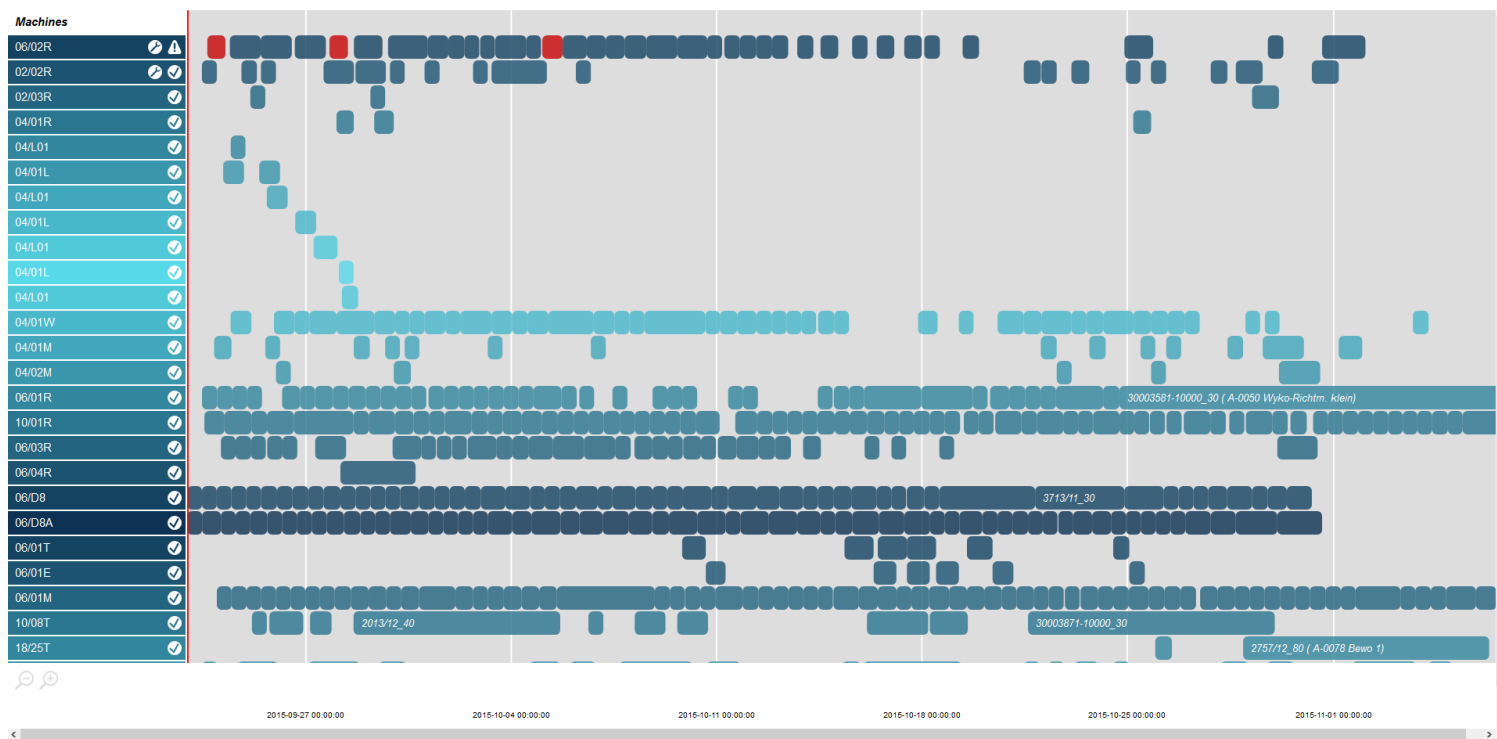
Anfang 2016 Prototypenausentwicklung

Iteration 1 – hybride Übergangslösung



Anfang 2016 Prototypenausentwicklung

Iteration 1 – hybride Übergangslösung



Anfang 2016 Prototypenausentwicklung

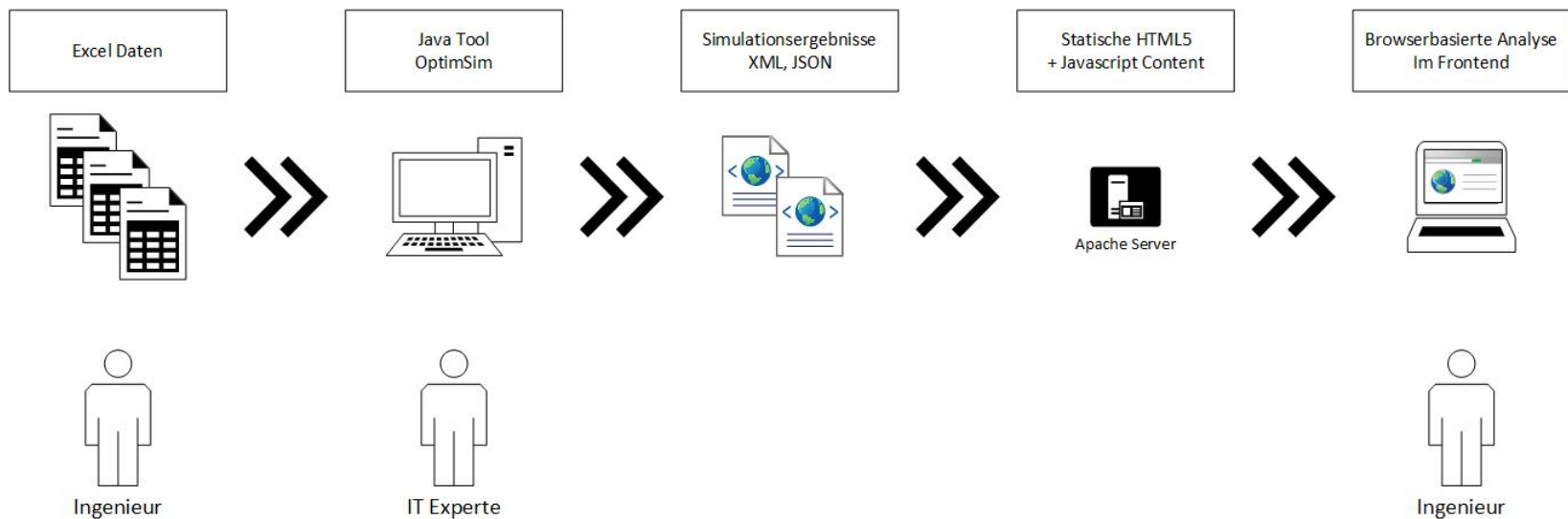
Iteration 1 – hybride Übergangslösung

- Architektur 0.1
- Backend (auf Desktop installierbar)
 - Native Java App
 - Datenverwaltung durch Excel Lösung
 - Nur durch Experten bedienbar
- Frontend
 - Statische HTML5 Website als Dashboard
 - integrierte Javascript Elemente um Charts zu arrangieren
 - Beschränkte Interaktivität, Bswp. Filterung der Ansichten aus Frontend heraus

Anfang 2016 Prototypenausentwicklung

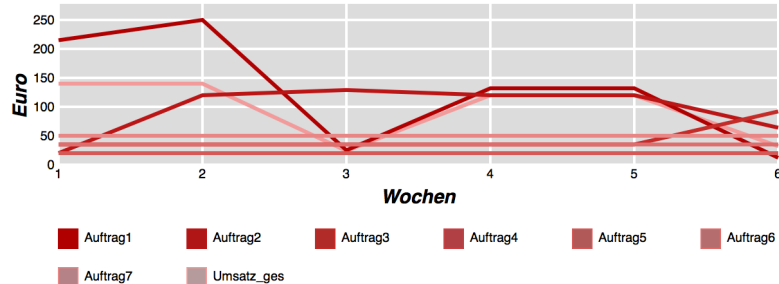
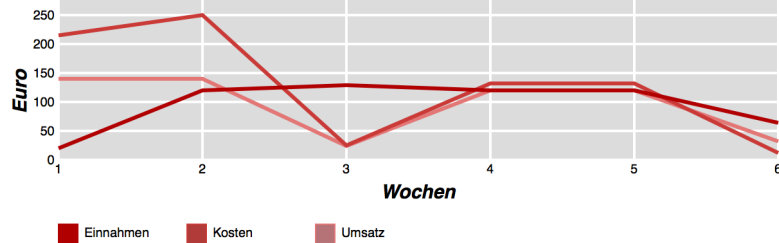
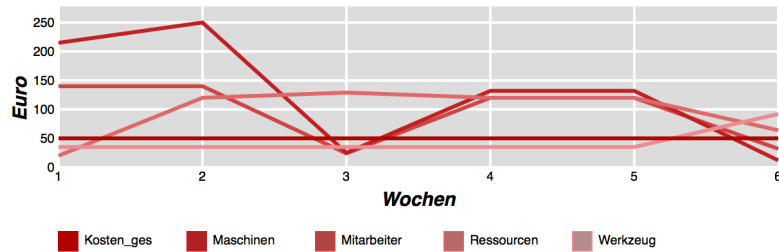
Iteration 1 – hybride Übergangslösung

■ Architektur 0.1



Anfang 2016 Prototypenausentwicklung

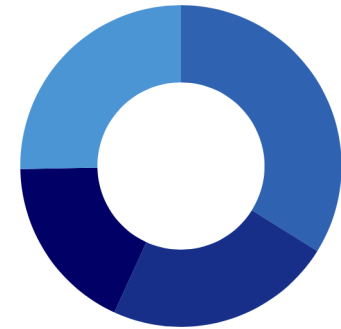
Iteration 1 – hybride Übergangslösung



Zeitraum 8 Wochen

2015-12-7 - 2016-01-31

Week 0	2015-12-07 - 2015-12-13
Week 1	2015-12-14 - 2015-12-20
Week 2	2015-12-21 - 2015-12-27
Week 3	2015-12-28 - 2016-01-03
Week 4	2016-01-04 - 2016-01-10
Week 5	2016-01-11 - 2016-01-17
Week 6	2016-01-18 - 2016-01-24
Week 7	2016-01-25 - 2016-01-31



Maschinen Mitarbeiter Verbr.-ressourcen
Werkzeuge

Gewinn

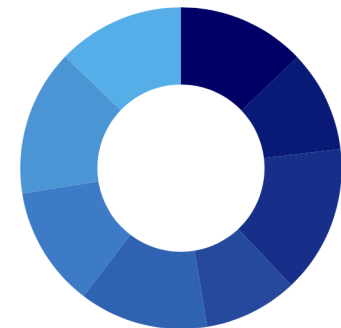
207

Umsatz

234

Kosten

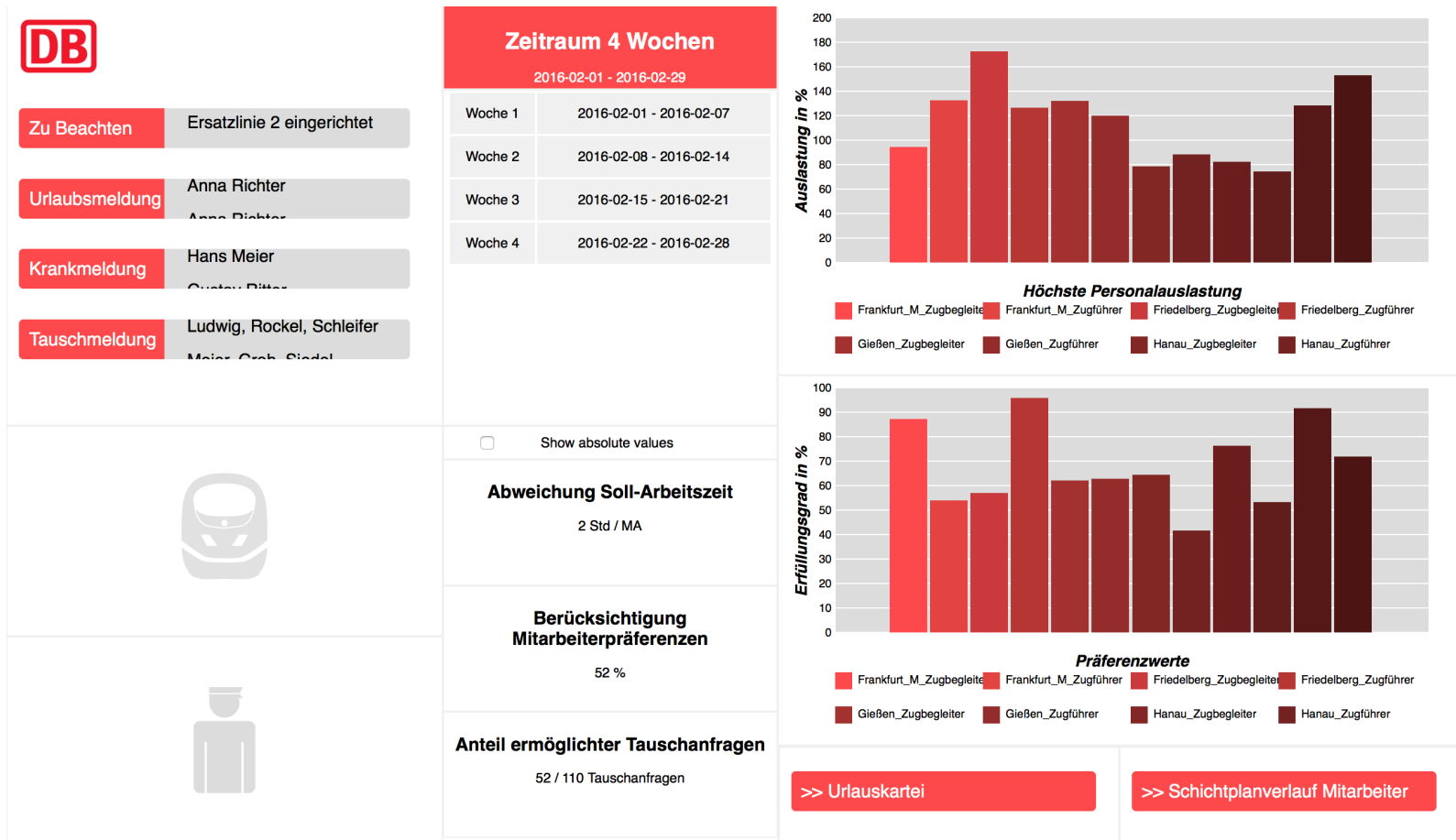
128



Auftrag 1 Auftrag 2 Auftrag 3
Auftrag 4 Auftrag 5 Auftrag 6
Auftrag 7 Auftrag 8

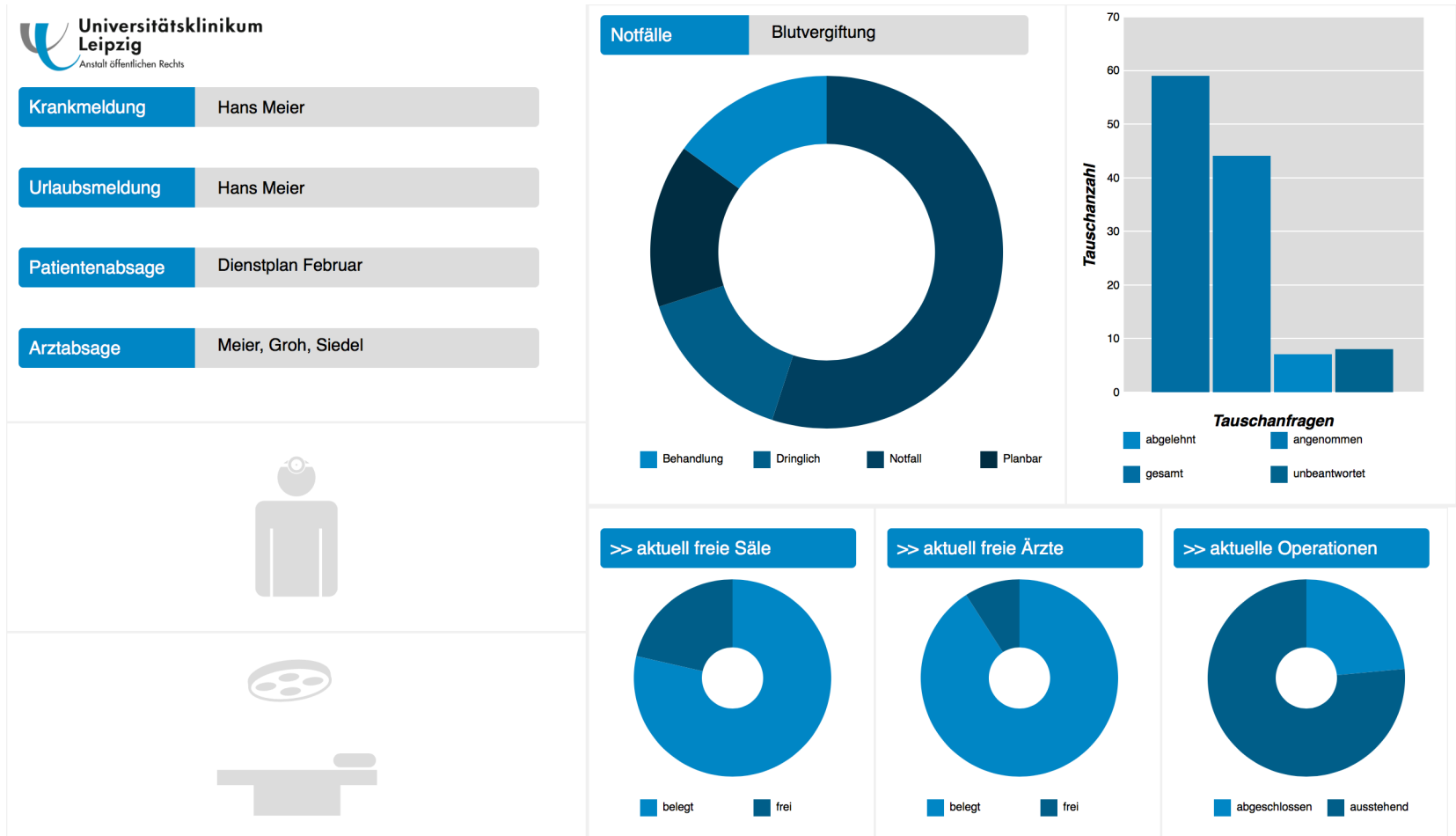
Anfang 2016 Prototypenausentwicklung

Iteration 1 – hybride Übergangslösung



Anfang 2016 Prototypenausentwicklung

Iteration 1 – hybride Übergangslösung



Deutsche Bahn Innovationsprodukt

Parallele Entwicklung eines eigenständigen Produktes

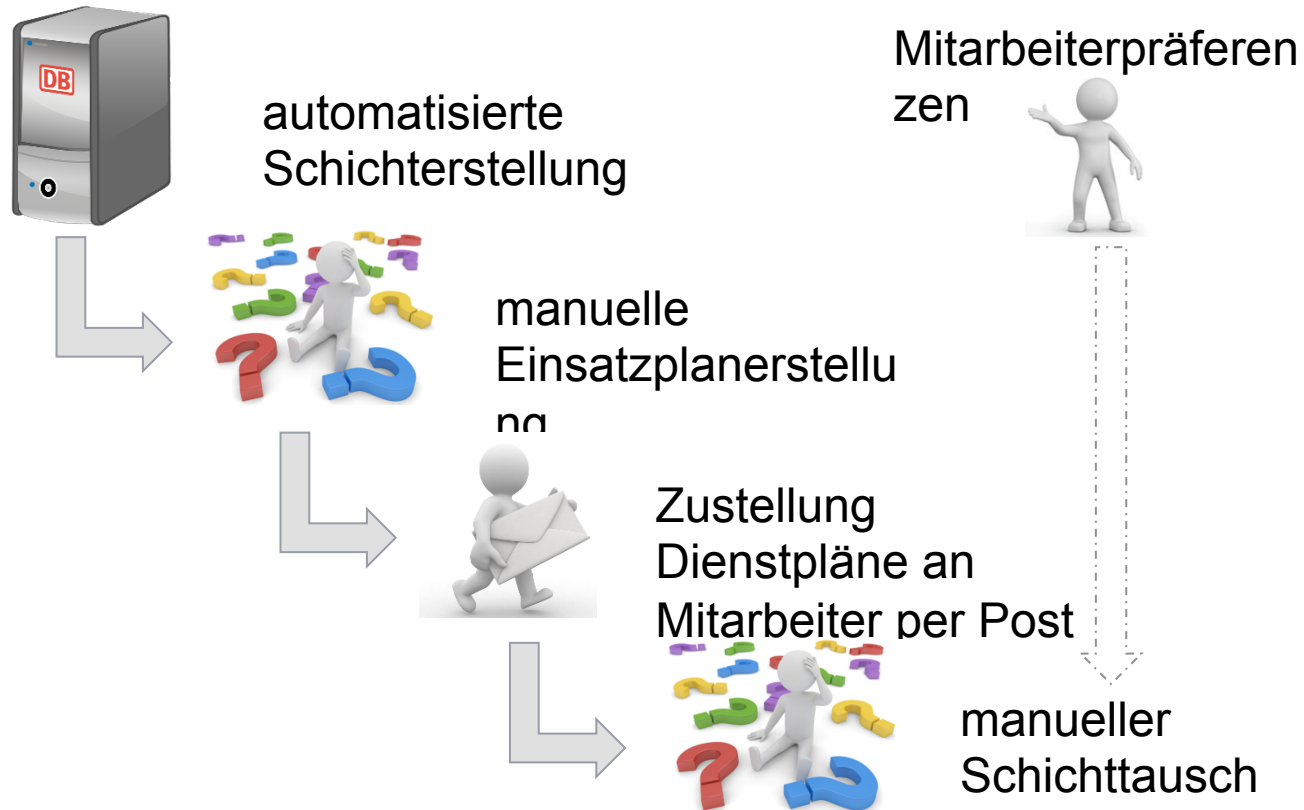
Mitte 2015 – Mitte 2016 DB Innovationswettbewerb

Personalplanung als eigenständige Software

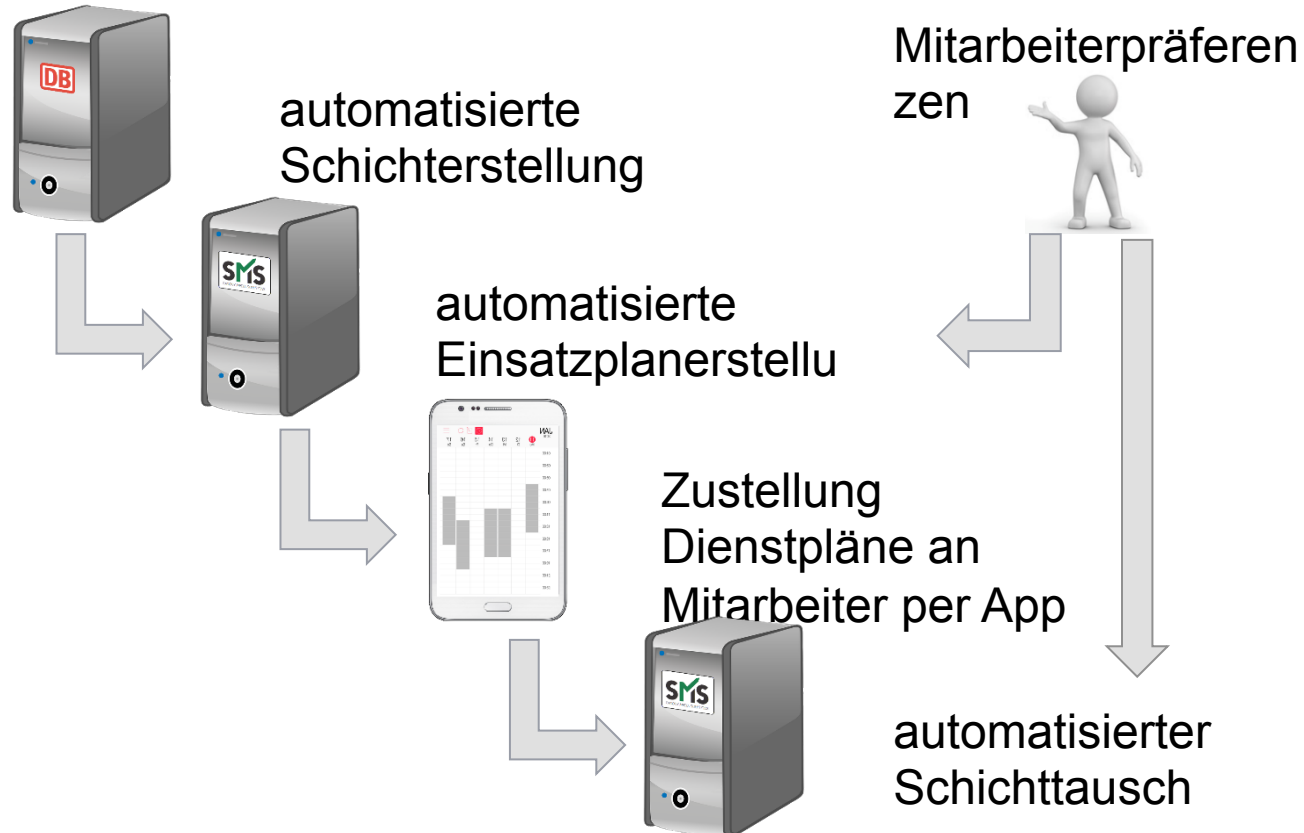
- DB Innovation Challenge
 - Industrieller Wettbewerb 07/2015 – 09/2016
 - 3 Wettbewerbsphasen
 - 133 Konzepte → 25 Prototypen → 10 Produkte → 3 Gewinner
 - Deutsche Bahn Innovationsprodukt



Ist-Zustand der DB Personaleinsatzplanung



Lösungsvorschlag DB Personaleinsatzplanung



Mitte 2015 – Mitte 2016 DB

Innovationswettbewerb

Personalplanung als eigenständige Software

Team:

- 10 (1 Optimierung, 1 ½ BE, 2 FE, 2 Android, 1 ½ ING + ½ Orga, 1 Test)

Methode / Tools:

- Wasserfallmodell (Excel),
- EXTREME Programming,
- **Test**
- Kommunikation toolunterstützt (SVN, Trac)

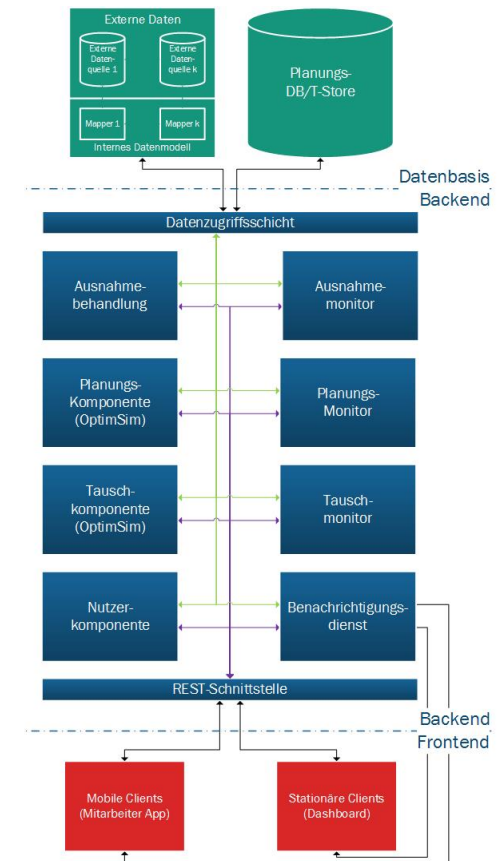
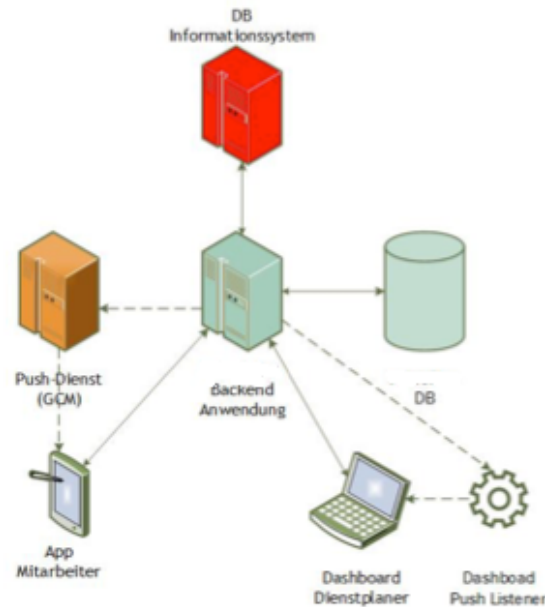
Fokus der Anforderungen

- Optimierung der Dienstpläne
- Funktionen durch Personalplaner bedienbar
- Anlegen von Daten, Editieren von Daten in Datenbank
- Steuerung der Algorithmen
- Konfiguration, Editierbarkeit
- Mitarbeiter App

Mitte 2015 – Mitte 2016 DB Innovationswettbewerb

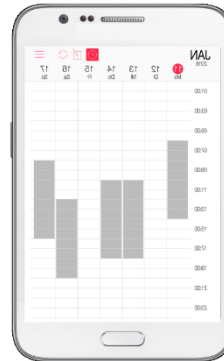
Personalplanung als eigenständige Software

- **Architektur:**
 - Service orientierte
 - WebSchnittstellen REST (JSON)
 - Anbindung Datenbank via Hibernate
 - Push – Benachrichtigungen
 - Interaktive Optimierung
- **Code:**
 - Hoher Anteil an domänenspezifischen Code
 - Wiederverwendbarkeit eingeschränkt
 - Tests manuell (FE) und automatisiert (BE)



Zielsystem

Mitarbeiter App



**Simulationsoptimierer
PersoPlanSim**



**Personalplaner
Dashboard**



Mitte 2015 – Mitte 2016 DB Innovationswettbewerb

Personalplanung als eigenständige Software

ausgezeichnet als
Deutsche Bahn
Innovationsprodukt
2016



Produktversion Simulator

Iteration 2 - Webvariante

Ende 2016 – Simulation und Optimierung Webbasiert

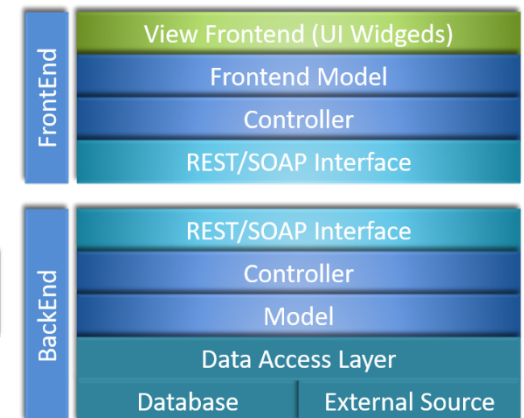
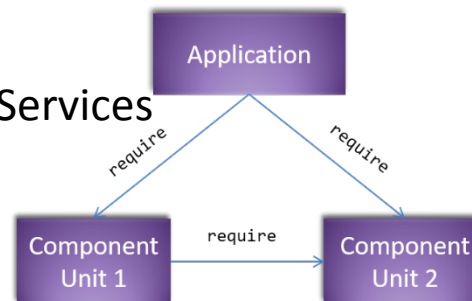
Iteration 2 – webbasierte Simulationssteuerung

- Ziel:
 - Datenbankanbindung
 - Steuerung der Funktionen über Web Oberfläche
 - Ablösung der Java Desktop Anwendung
 - Up-/Download der Excel Mappen
 - Verwendung von Oberflächenelementen aus DB Innovationskontest

Ende 2016 – Simulation und Optimierung Webbasiert

Iteration 2 – webbasierte Simulationssteuerung

- Team:
 - 3 (1 BE, 1 Sim, 1 FE)
- Methode:
 - Agil, Scrum, Jira, Confluence, UML, GIT
- Architektur:
 - Datenbank
 - Anwendung BE
 - Web Services, Konkrete Data Services



Ende 2016 – Simulation und Optimierung Webbasiert

Iteration 2 – webbasierte Simulationssteuerung

22.11.2014 06:00:00 Tester

Live-Modell Langzeitplanung Kurzfristige Grobplanung **Kurzfristige Feinplanung** Steuerung

Newsticker

Ende Aktivität	
Start Aktivität	
Ausnahme Ladeplan	
Ausnahme Ressourcenverfügbarkeit	
Report Schichtende	

Verschiedene Planungshorizonte mit angepasstem Dashboard

Newsticker mit aktuellen Meldungen für den Planer

Versionierung

Live-Modell

Replikation

Bsp1 viele Flüge

Schicht1 zu wenig Stellflächen

Lauf liveTest

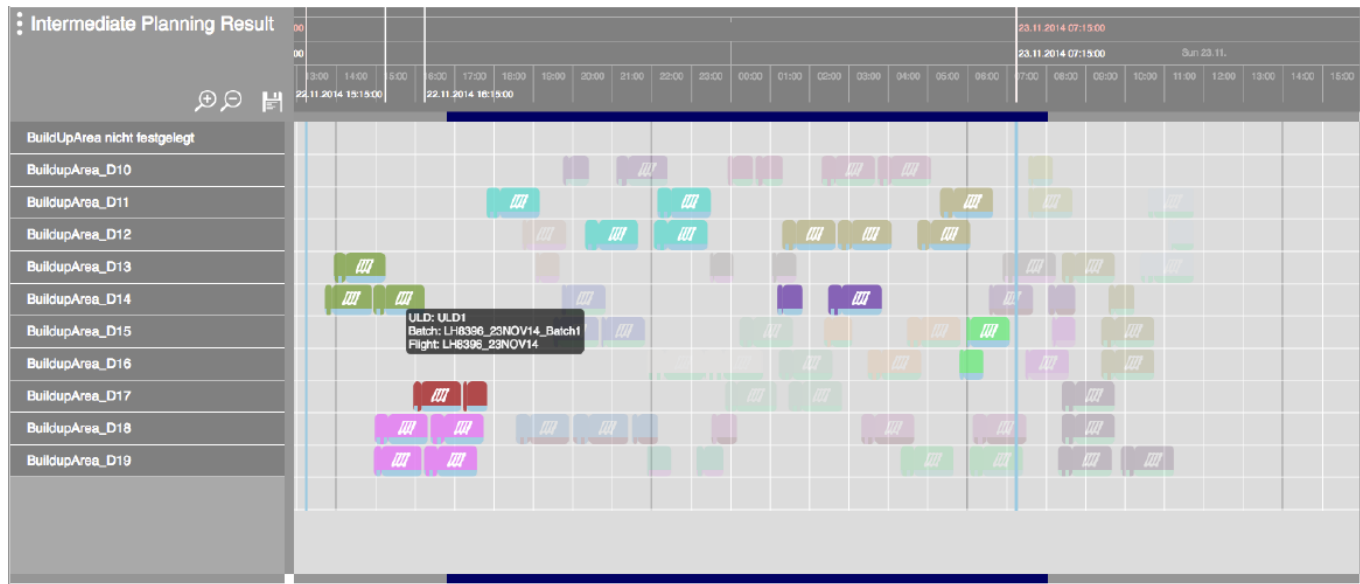
FragenLHCargo

TestMax

Demonstration LH-Cargo

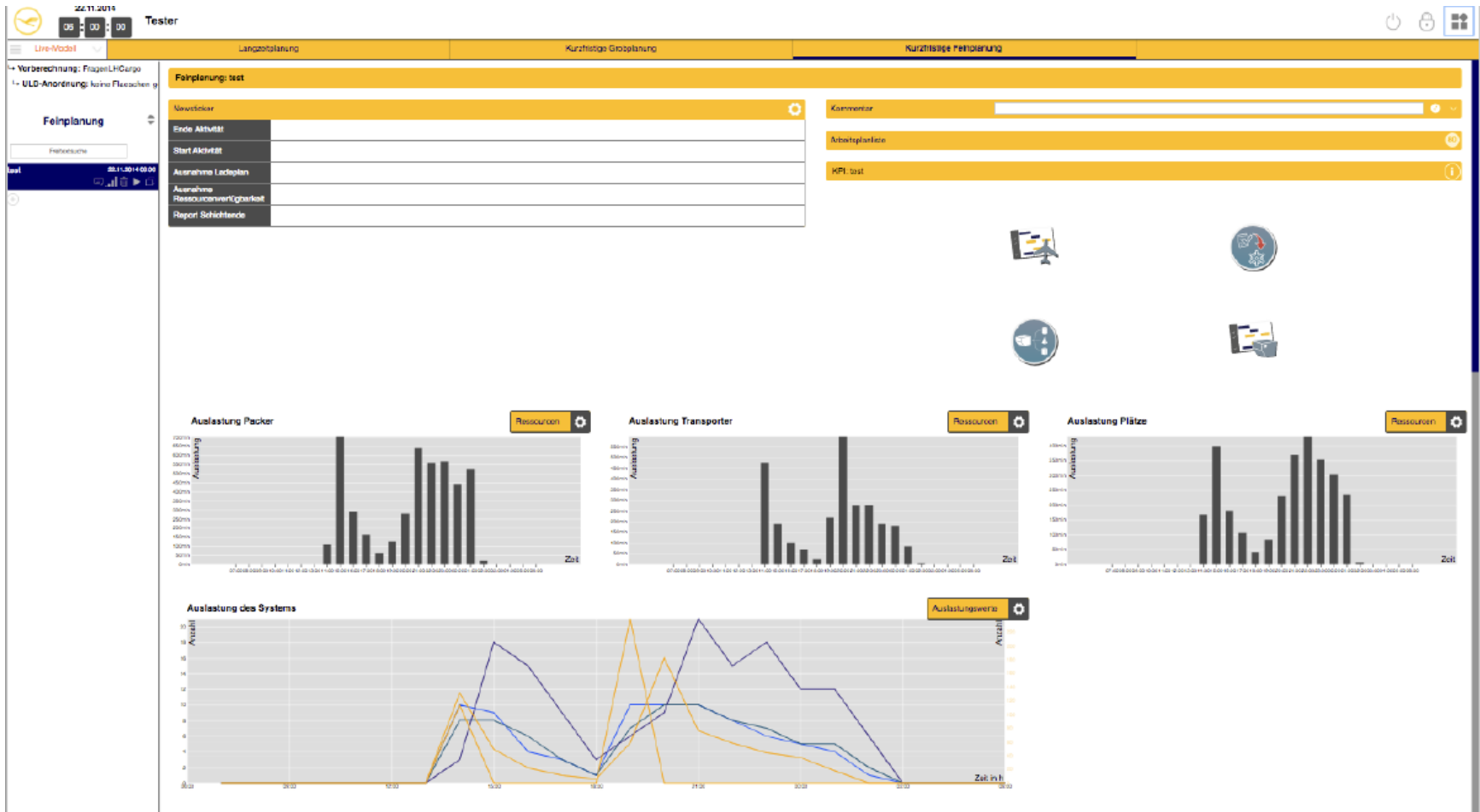
Ende 2016 – Simulation und Optimierung Webbasiert

Iteration 2 – webbasierte Simulationssteuerung



Ende 2016 – Simulation und Optimierung Webbasiert

Iteration 2 – webbasierte Simulationssteuerung



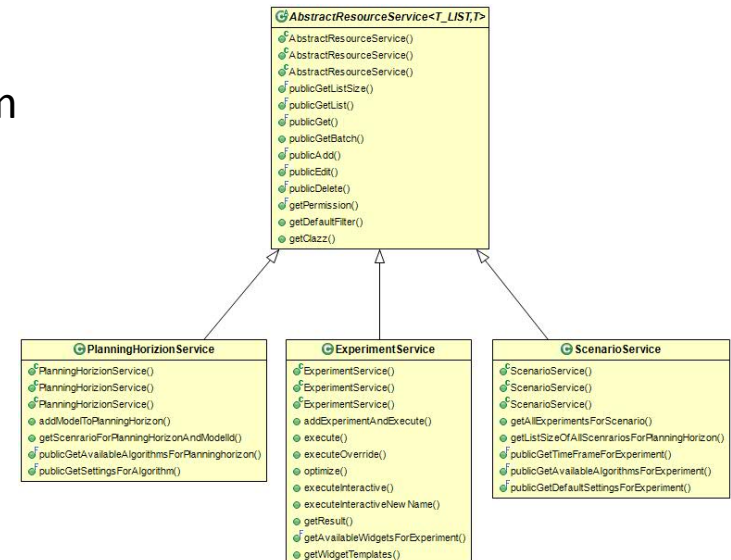
Ende 2016 – Simulation und Optimierung Webbasiert

Iteration 2 – webbasierte Simulationssteuerung

- Aufbau eigener Best-Practice
 - Entwicklung erster Standardschnittstellen um
 - Read(String id)
 - ReadList(String filter, int offset, int limit)
 - Update(Resource r1)
 - Delete(Resource)
 - DB Anbindung
 - Hibernate (ORM Mapping mit Java)
 - Kommunikation
 - Json

geringere Datenmenge im Vergleich zu XML,
validierbar durch Schema, Erweiterbarkeit,
triviale Einbettung in Javascript

- Beobachtung: Gesamte Entwicklungsgeschwindigkeit nimmt ab.



Ende 2016 – Simulation und Optimierung Webbasiert

Iteration 2 – webbasierte Simulationssteuerung

- Ursachen langsamere Entwicklungsgeschwindigkeit:
 - Viele Projekte für kleines Team bei geringer Menge Bestandssoftware
→Hoher Entwicklungsbedarf
 - Wachsendes Team
→Erhöhung Kommunikationsaufwand
→Steigender Einarbeitungsaufwand
 - Agile Entwicklung ohne konkreten UseCase und gesamten Weitblick
→Hypothetische Annahmen
→Verwerfen von Entwicklungen
- Erkenntnis 1: Eigenentwicklung Frontend gerät an Grenzen
- Erkenntnis 2: Wiederverwendbarkeit zwingend notwendig, damals nur bedingt möglich

Fokus auf UseCase

Iteration 3 - APS

2017 Weiterentwicklung und Ausdefinition der Use-Cases

Geschäftsidee APS

- APS
 - Advances Planung and Scheduling
 - Produktion Planung + X

- Was passiert wann und wo?
- Simulation in Experimenten

	Bereich	Verhalten	Horizont
Produktionsplanung	Produktionsprogrammplanung	Welche Produkte sind in welchen Mengen zu produzieren	Strategisch Taktisch Operativ
	Bereitstellungsplanung	Bedarf an Elementarfaktoren für geplante Produkte festlegen	
	Produktionsprozessplanung	Zeitliche, mengenmäßige und räumliche Planung des Produktionsvollzugs	
Produktion Steuerung	Auftragsfreigabe Feinterminierung Betriebsdatenerfassung	Aufträge für die Produktion freigeben Ressourcenbereitschaft für die Aufträge überprüfen Aufträge genau einordnen	

2017 APS

Funktionale Anforderungen

- **Anforderungen:**
 - **Umfangreiches Datenmodell**
(Ressourcen, Verfügbarkeiten, Bedarfe, Aufträge, Arbeitspläne, Rückmeldungen, Zustände, Fehler, Leistungskennzahlen, Zusammenhänge, Restriktionen, Rüstzeiten, Prozessdauern, Stochastik, ...)
 - **Verwaltung aller Daten über Editor**
 - **Vielfältige Funktionen**
(Plan erstellen, Auslastung analysieren, Bedarfe berechnen, Plan optimieren, Manuell eingreifen, Experimente vergleichen, Unregelmäßigkeiten erkennen, Planer über Unregelmäßigkeiten informieren)
 - **Schnittstellenfähigkeit zu externen Systemen**



2017 APS

Nicht funktionale Anforderungen

- Standardprodukt
 - notwendig, um breite Kundenbasis aufbauen und warten zu können.
- Anpassbarkeit
 - Jeder Kunde hat Produktionseigenheiten (Maschinenbauer, Verpacker, Transporte, Säge, Öfen, ...)
 - Fähigkeit zur Customization Voraussetzung für langfristige Stabilität
- Massendatenfähigkeit
- Echtzeitfähigkeit
- Bisherige Methoden nicht anwendbar:
 - Dedizierte Standardservices und Sichten zu bauen, nicht tragbar
- Frage: Wie können Frontend und Backendarbeiten reduziert werden?
- Idee: Modellgeneriertes Frontend & Generisches Modell

Architektur 1.0

Modellgeneriertes Frontend

- Teamgröße 11-15
- Methode:
 - Jira, Confluence, Jenkins (CI), **automatisierte FE-Testings mit cypress**
- Grundarchitektur
 - Umstieg der Frontend-Technologie auf High Level Angular X
 - Aufbau eines High-Level-Kataloges „Templates“
 - Verstärkter Einsatz externer Bibliotheken (Charts)
- Konzept:
 - Definition von Formularen, UI-Elemente, Interaktion und Data-Binding im Backend.
 - Frontend generiert Ansichten aus backend Beschreibung.
- Ziel → Frontend wird im Backend generiert. (Thin Client)

Architektur 1.0

Modellgeneriertes Frontend

Funktionsweise

Grundlage bildet Domänenmodell / Klassenmodell / POJO

- ORM Mapping Hibernate

Templates

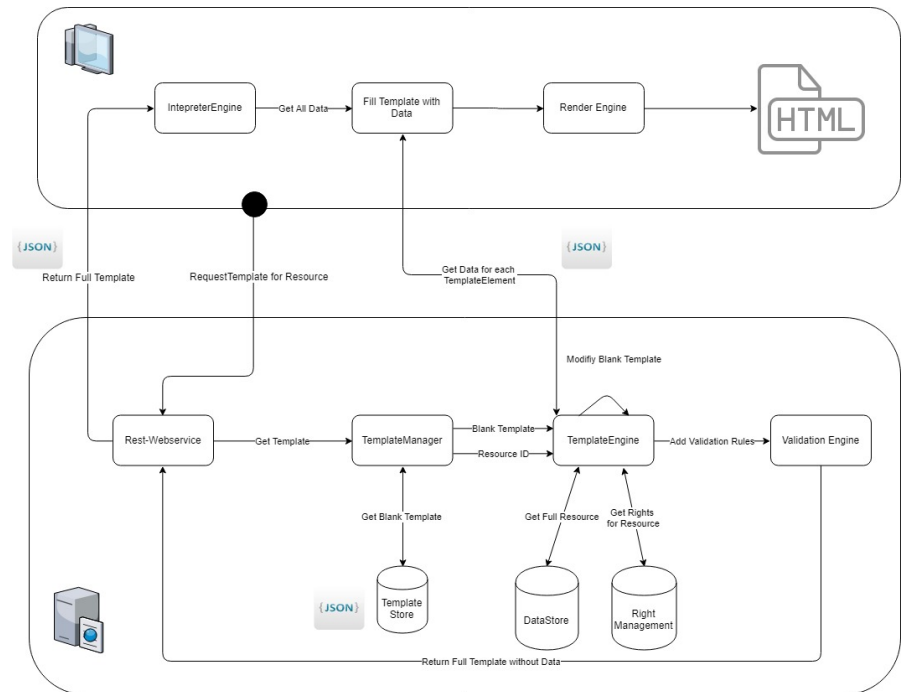
- Beschreibung von UI-Elementen werden als Template im Backend mit JSON
- Data-Binding über XPATH ähnlichen Ausdrücken zur Adressierung im Modell ausgestattet

Template Service:

- Engine zur Interpretiert der Templates im Backend.
- Durchführung von Data-Binding im Backend und Frontend
- Auslieferung an Frontend über wenige REST-Schnittstellen
- Jedes Oberflächenelement ist dann einem Attribut einer Ressource zugeordnet

Template-Service aus Frontend-Sicht

- Engine zum Abruf von Rendern von Templates
- Ein Modul je Template Elementtyp



Architektur 1.0

Modellgeneriertes Frontend

Katalog von Template Elementen

- Formularelemente:
IndexRegister, Buttons, Textfelder, Checkboxes, Tabelle, usw ...
- Strukturelemente
Navigationsbaum, Inhaltsbereich, Header, Menü, Wizzard, ...

Template-Element Services

- Formular Services
Editieren von Textfeldern, Hinzufügen von Zeilen in Tabelle, kopieren, löschen, ...
- Spezielle Interaktive Services Je Oberflächenelement
Vor und nachgelagerte Aktionen, Nachrichte, ...
- Sonstige
Filtern, Suche, Autovervollständigung, ...

Resultat

- Aufbau von Editor und Eingabedialogen erfordert keine Erstellung von Services bzw. Frontendentwicklung mehr.

The screenshot displays the SMS software interface. The top menu bar includes 'Horizonte', 'Modelle', 'Experimente', 'Daten', 'Extras', 'Hilfe', and 'Anzeige'. Below the menu is a toolbar with various icons. The left sidebar contains a 'Menü' section with a tree view showing 'Editor', 'Änderungsprotokoll', 'Bereiche', 'Produktionsprogramm', 'Produkte', 'Aufträge und Workfile', 'Kunden', 'Aufträge' (highlighted), 'Fertigungsaufträge', 'Ressourcen', 'Mitarbeiter', 'Maschinen', 'Werkzeuge', 'Behälter', 'Material', 'Lager', 'Qualifikation', 'Arbeitszeiterwaltung', 'Rüstmatrizen', and 'Szenario'. The main content area is divided into two sections. The top section, titled 'Auftrag', contains a form with fields for 'Name' (POGM9-11943), 'Beschreibung', 'No.' (POGM9-11943), 'Priorität' (1), 'Frühester Start' (01.10.2017 07:10), 'Spätester Ende' (17.09.2018 02:00), and 'Kunde'. The bottom section, titled 'Auftrag Produkte', contains a table with columns 'Produktname', 'Anzahl', 'Frühester Start', 'Spätester Ende', and 'Typ'. The table has one row with the data: 'Product POGM9', '21', '01.01.1970 01:00', '01.01.1970 01:00', and 'Kundenauftrag'. The status bar at the bottom shows 'Model: Model 11.01.2019 17:55:13 11.01.2019 17:55' and 'Experiment: Short'.

Architektur 1.0

Modellgeneriertes Frontend

Katalog von Template Elementen

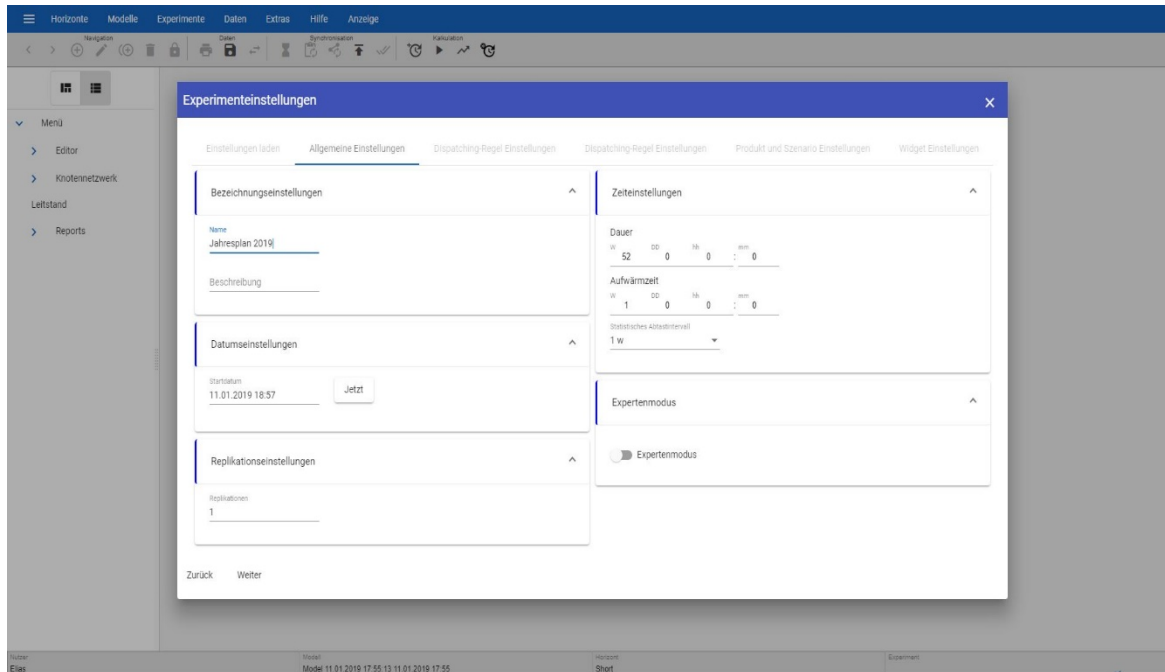
- Formularelemente:
IndexRegister, Buttons, Textfelder, Checkboxes, Tabelle, usw ...
- Strukturelemente
Navigationsbaum, Inhaltsbereich, Header, Menü, Wizzard, ...

Template-Element Services

- Formular Services
Editieren von Textfeldern, Hinzufügen von Zeilen in Tabelle, kopieren, löschen, ...
- Spezielle Interaktive Services Je Oberflächenelement
Vor und nachgelagerte Aktionen, Nachrichte, ...
- Sonstige
Filtern, Suche, Autovervollständigung, ...

Resultat

- Aufbau von Editor und Eingabedialogen erfordert keine Erstellung von Services bzw. Frontendentwicklung mehr.



Architektur 1.0

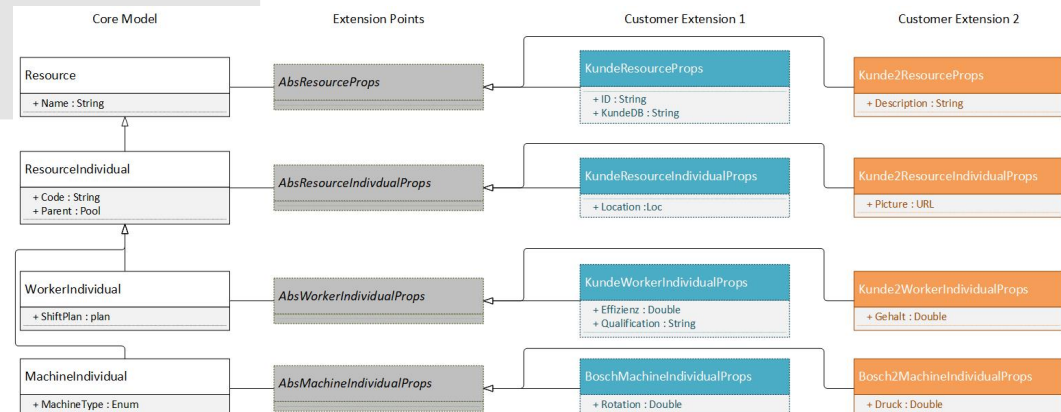
Generisches Modell

Allgemeines Domänenmodell als POJO

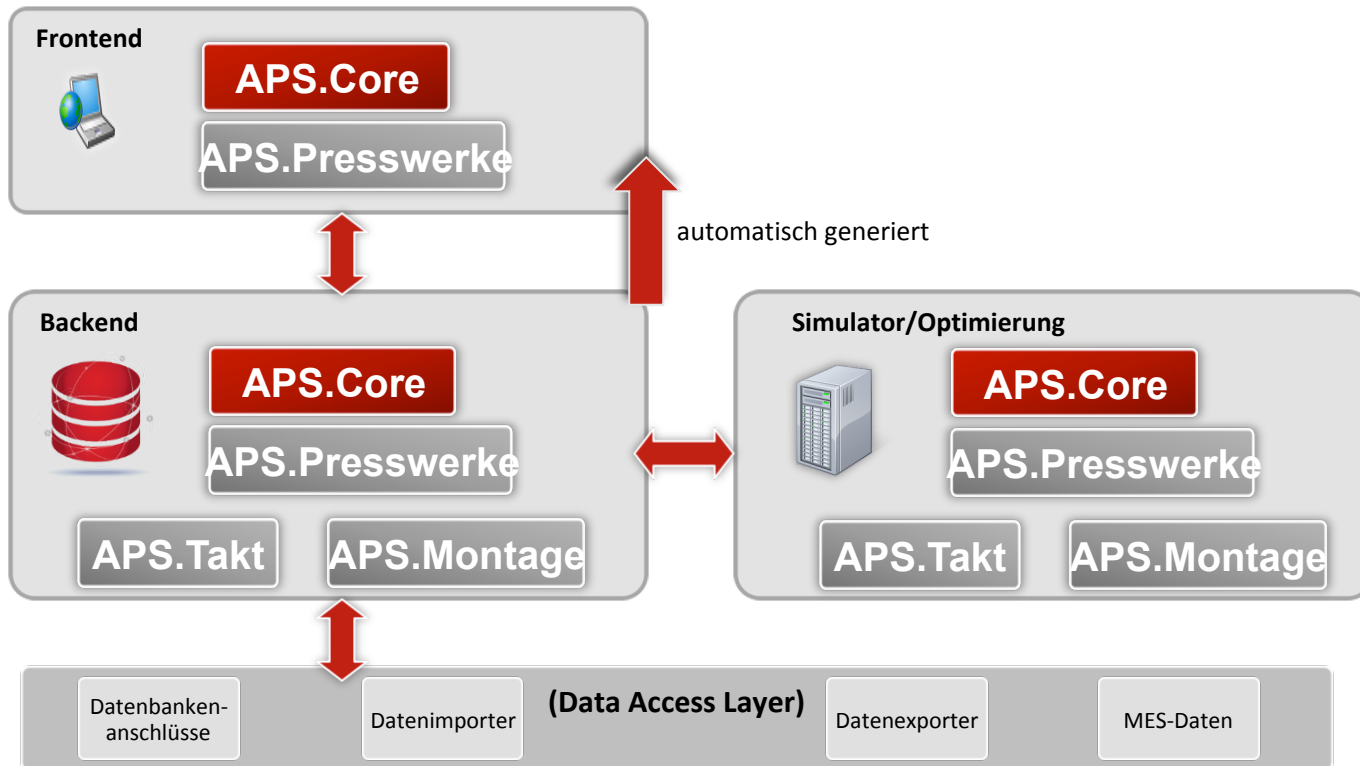
- ORM Mapping Hibernate
- Template API: Auf „Reflections“ basierte Modellverwaltung und Aufbereitung der Template Modellelemente im Backend
- Auslieferung dieser Daten an Frontend

Feste Erweiterungspunkte über „Property Extensions“

- Minimalistische Erweiterungen
- Wartbarkeit



Standardisierte angepasste Lösung



2018 – heute

- Teamgröße Heute 21
- Geplant bis Ende 2019: 27 Mitarbeiter
- Rentabilität der Architekturentscheidungen stellt sich ein
- Aufbau mehrerer Projektteams, die unabhängig arbeiten, bei konstantem Backendteam / Frontendteam.
- Lediglich Katalog wird erweitert
- Iterative Überarbeitung der Elemente

SaxMS Produkte

Produktions-/ Logistikplanung SaxMS APS

- **Produktionsvollzug**
- Ressourcen- und Betriebsmittelmanagement
- Auftragsverwaltung von Auftragsnetzen sowie Stücklisten
- Produktionsvollzugsplanung
- Materialplanung zum genauen Materialbedarfsvollzug
- Lieferterminanalyse
- Wartungsplanung

Personalplanung SaxMS PEP

- **Personaleinsatzplanung**
- Automatische Schichtplanerstellung
- Umplanung mit minimaler Planänderung
- Wechselschläge bei Ausfall
- Manuelle Planänderungen und Bewertung
- Urlaubs- u. Krankheitsverwaltung
- Plananalyse bspw. für Urlaubswünsche
- Stamm- u. Bewegungsdaten-verwaltung

Kundenplattform SaxMS CP

- **Bestellvorgänge**
- Anfragen-/Auftragserstellung
- Nachforderungen
- Übersicht über Status der Aufträge
- Einfaches Verwalten der Anfragen/Aufträge für Kunden und Disponenten

2018 – heute

APS aktuelle Screens

Maschinenübersicht

Maschinen (Einzel)

Schnellsuche Zeile Spalte

✎ ✖ ➕

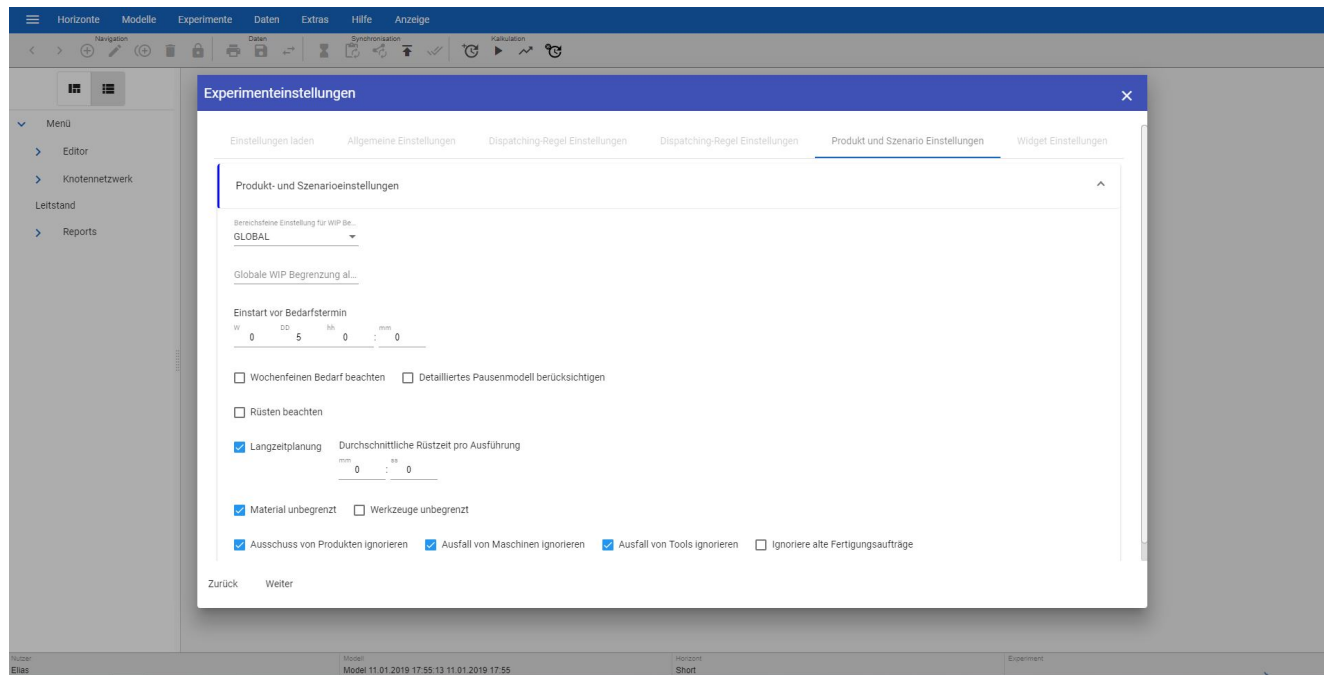
Editieren Löschen Hinzufügen

	Nr.	Maschine	Beschreibung	Maschinentyp	Maschinen Pool
1	01HEQT09	01HEQT09		MultipleChannel	
2	01_CHAM	01_CHAM		Group	
3	08_CHAM	08_CHAM		Group	
4	0FRTXi20	0FRTXi20		Standard	
5	0KOHXK43	0KOHXK43		MultipleChannel	
6	1-040SPC	1-040SPC		Standard	
7	1-AGYRO	1-AGYRO		Group	
8	1-DIEDA	1-DIEDA		Group	
9	1-GURT	1-GURT		Group	
10	1-GYRO	1-GYRO		Group	
11	1-HOTPL	1-HOTPL		Standard	
12	1-MTC	1-MTC		Group	
13	1-RSS201	1-RSS201		Group	
14	1-SGSPC	1-SGSPC		Standard	
15	1-SIOLY	1-SIOLY		Group	
16	1-SMD25X	1-SMD25X		Standard	
17	1-SMGLIE	1-SMGLIE		Standard	
18	1-SMP-LI	1-SMP-LI		Standard	
19	1-SPA-IN	1-SPA-IN		Standard	
20	1-STRAJ	1-STRAJ		Standard	
21	1-VERIF	1-VERIF		Group	
22	10-GURT	10-GURT		Group	
23	10-GYRO	10-GYRO		Group	
24	10-HOTPL	10-HOTPL		Group	
25	10-SGGUR	10-SGGUR		Group	

Nutzer: Elias Modell: 11.01.2019 17:55:13 11.01.2019 17:55 Horizont: Short Experiment

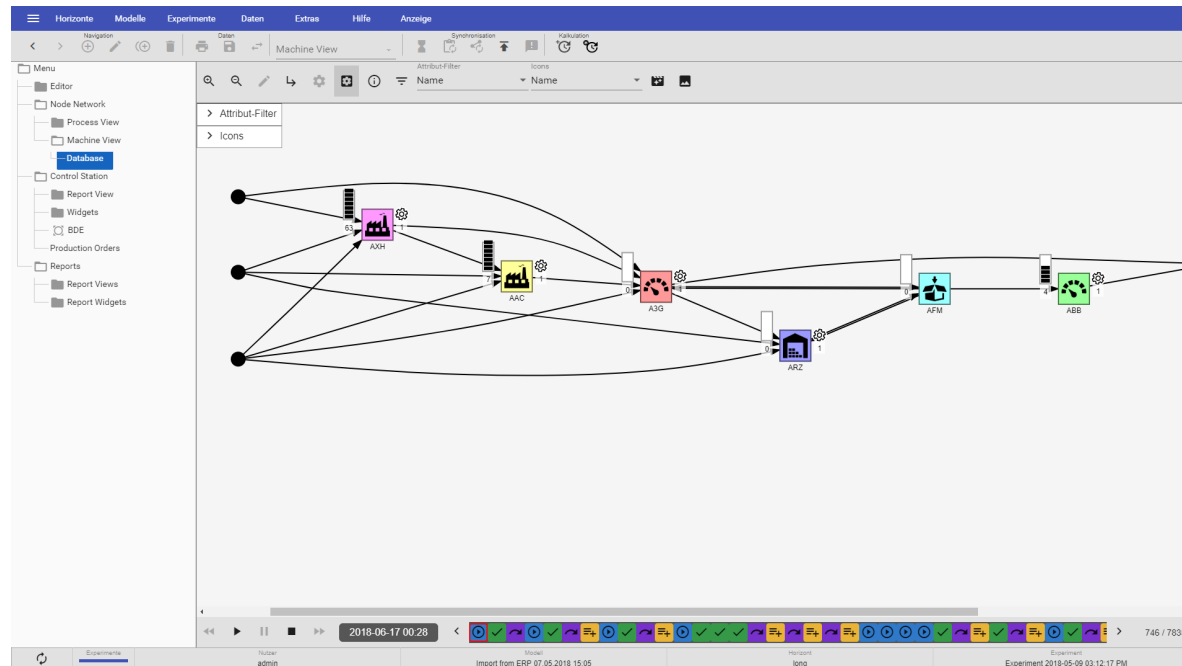
2018 – heute

APS aktuelle Screens



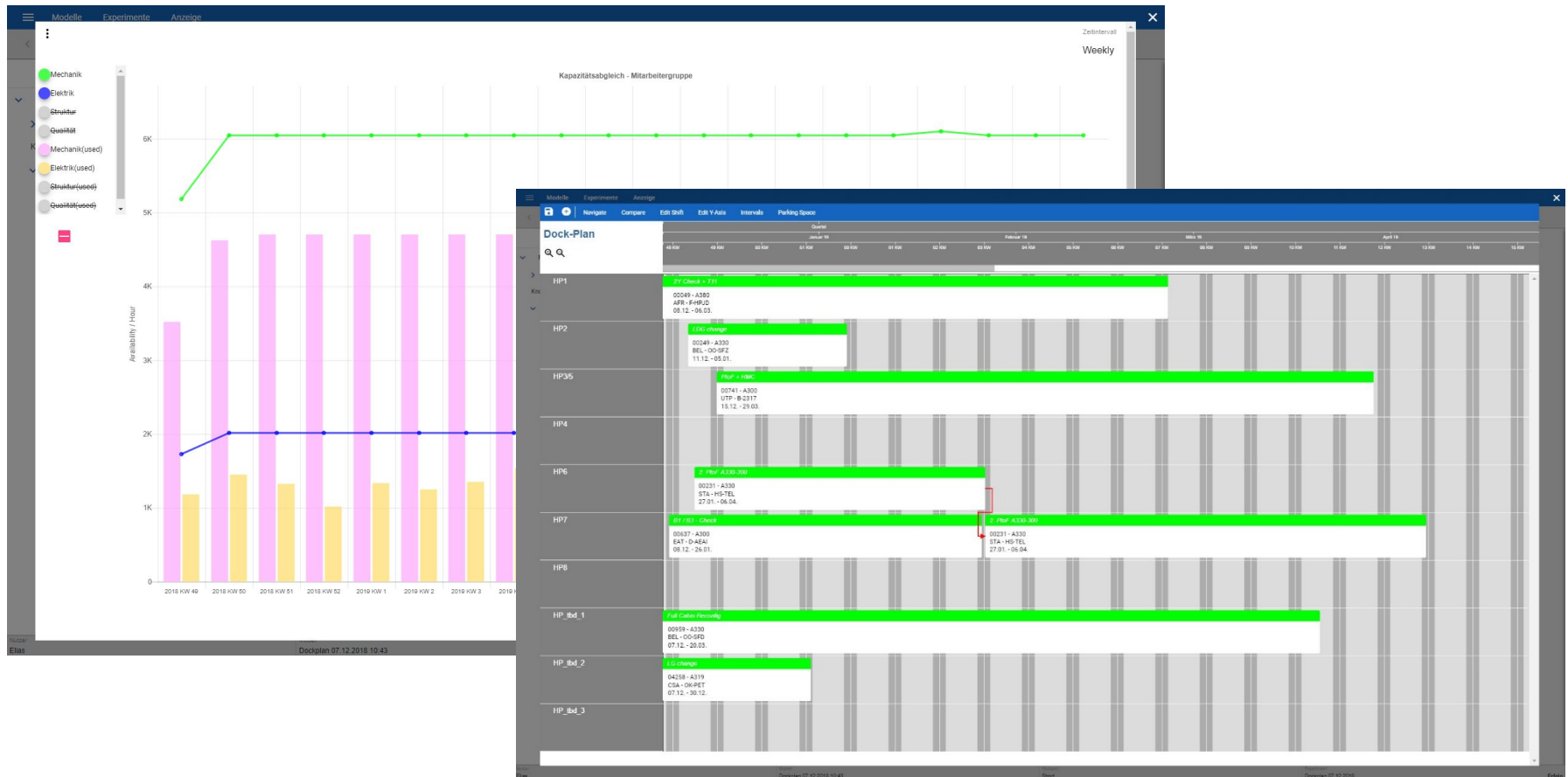
2018 – heute

APS aktuelle Screens



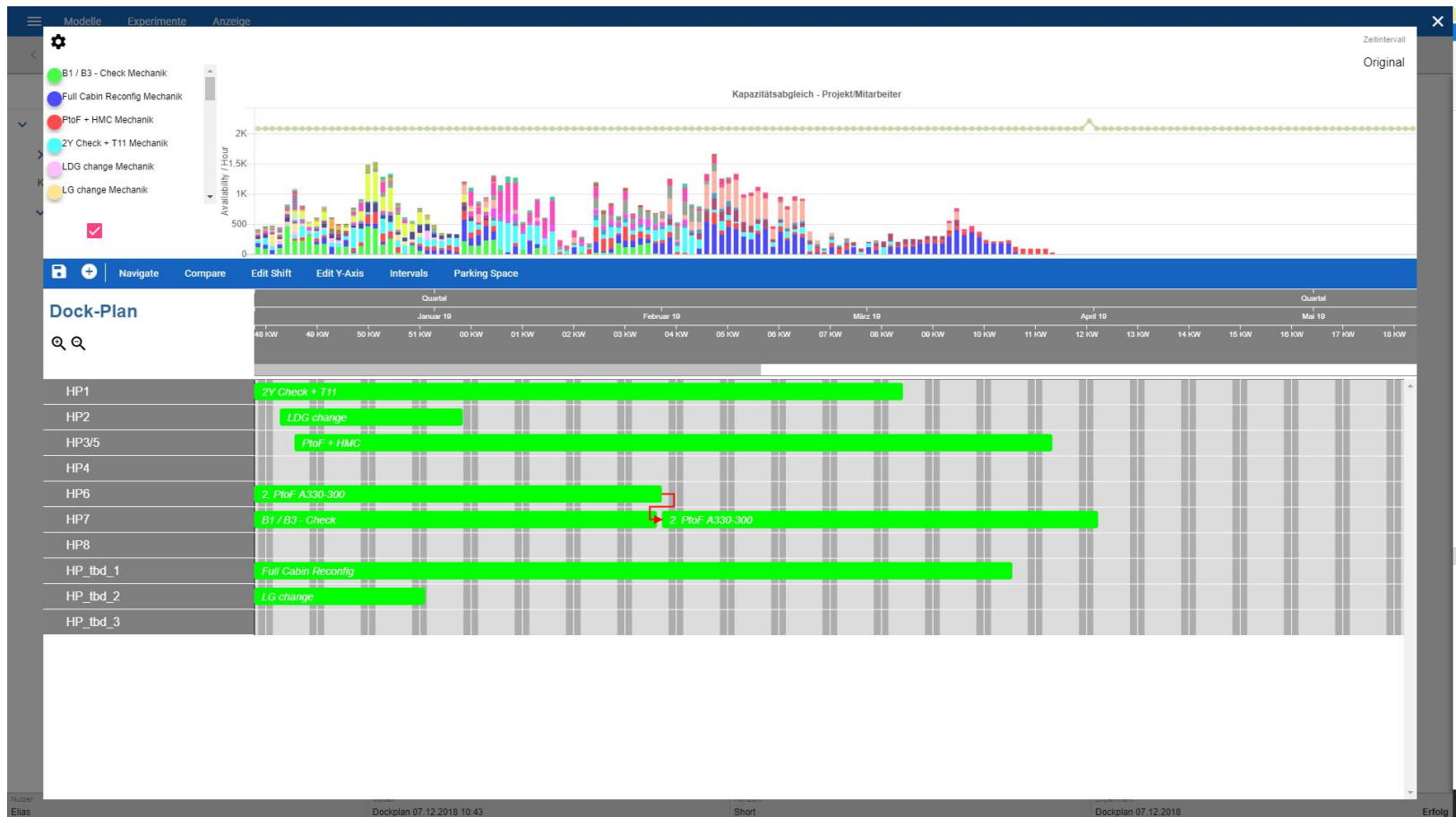
2018 – heute

APS aktuelle Screens



2018 – heute

APS aktuelle Screens



2018 – heute

CP aktuelle Screens

DB Kundenplattform

Entwürfe

anlegen

Hallo

Rico Poehland

Berlin Hbf

960989200263

2100001

Gründung 960989

Sturm 10HG

19.12.2018 16:50

NNK 2

Holger Sanft

18.12.2018 14:00

Sa 002-2019...

Holger Sanft

18.12.2018 13:58

sfdgsfdg

Rico Poehland

18.12.2018 13:31

test

Thomas Litzke

08.01.2019 13:59

TestAntrag

Rico Poehland

20.12.2018 17:22

TS-04

Tom Schädlich

17.01.2019 13:04

Nachtrag 2 T...

Rico Poehland

19.12.2018 16:40

Hallo

↓

Entwurf

erstellt am:

19.12.2018 16:50:15

Schreibstatus:

Bearbeitung möglich

Allgemein

Status

Entwurf

Antragsdatum

19.12.2018

Auftragsbezeichnung

Schaden Gaschwitz-Böhlen

Beinhstelle

Gründung 960989

Antragsnummer

2100001

* Vorhabensnummer

2100001-10-2

* Teilvorhabensnummer

960989200263

* AAR-Nummer

Antragsteller

Rico Poehland

Auftragsverantwortlicher

Sturm 10HG

* Arbeitsverantwortlicher v...

Telefonnummer Arbeitsver...

Ort des Auftrages

Gaschwitz 00000

* Arbeitsort

Berlin Hbf

* Abstellort/Gleisnummer

* Detailinfos Abstellort ZW...

* Infos zu den Fahrplänen

Netzmaßnahme

Startdatum:

Enddatum:

Auftrag

Schnellsuche

Spalte

Antragspositionen

+

+

hinzu

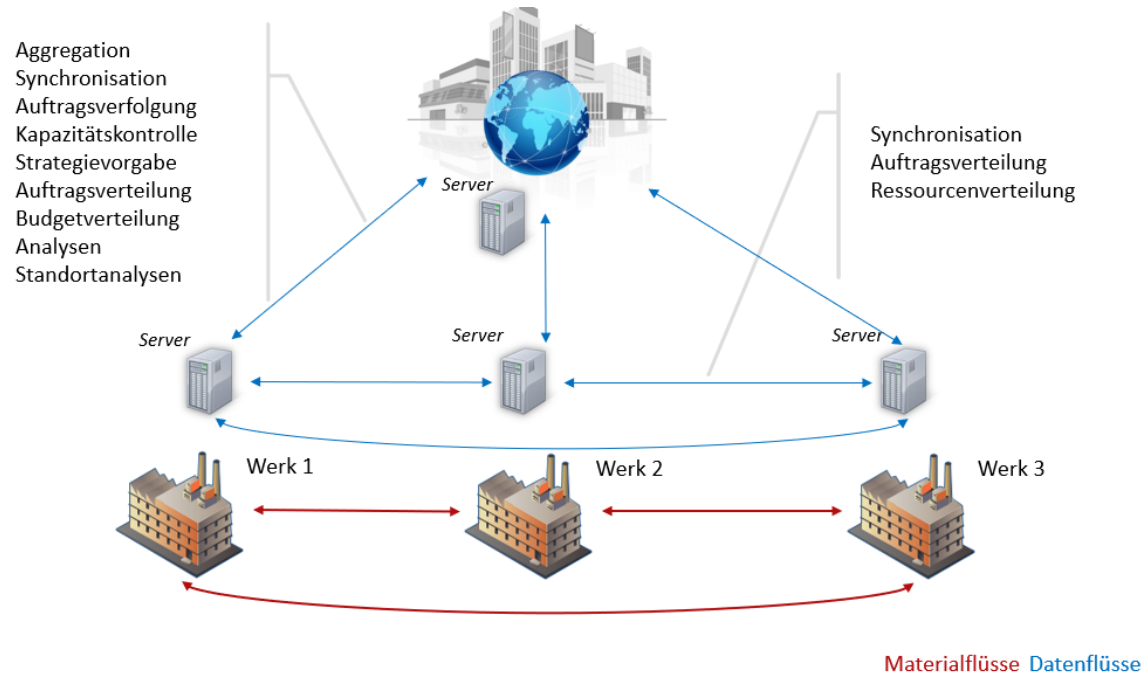
fügen

stornieren

01.10.2018 bis 01.03.2019

Ausblick

SC Planung



FAZIT

Erfolgsfaktoren für erfolgreiche Softwareentwicklung

- Wiederverwendbarkeit & Wartbarkeit im Blick behalten
- Use-Case
 - die Idee muss ausgereift sein.
- Team
 - Mischung aus Überblick und Sp
- Methode
 - Planung
 - Lang, mittel, kurz
 - An Methoden muss man sich h
- Tools
 - Tools passend zu Methoden
- Technik
 - Technik wächst mit Aufgaben und über die Zeit
 - Offen sein für Änderung
- Reflexion
 - Ehrlich und Selbstkritisch



Vielen Dank für Ihre Aufmerksamkeit!

Für vertiefende
Ausführungen sprechen Sie uns an

**Prozesse optimieren und
Kosten reduzieren
in der Logistik und Produktion**

kontakt@saxms.de

**Saxony Media Solutions
GmbH**

Bärensteiner Straße 27/29

01277 Dresden

Tel.: 0351/33221361

Modular, Individuell, Einfach