

Case Story :

Modellgetriebene Softwareentwicklung bei der Rentenzahlung

Referenten:

Holger Benz (Deutsche Post AG, NL Renten Service)
Christoph Schmidt-Casdorff (iks GmbH)

Einführung

- **Erfahrungsbericht über den Einsatz von MDSD**
 - MDSD - Modellgetriebene Softwareentwicklung im Projekt AMIS der NL Renten Service der Deutschen Post AG
 - konkrete Erfahrungen aus einzelnen ausgewählten Bereichen der MDSD
- **Folien mit projektbezogenen Aussagen tragen ‘AMIS’ im Titel**
- **Keine Einführung in MDSD / MDA**

NL Renten Service

NL Renten Service der Deutschen Post AG

- **Zahlung der gesetzlichen deutschen Altersrente**
 - gemäß den gesetzlichen Bestimmungen
 - im Auftrag der Träger der Rentenversicherung

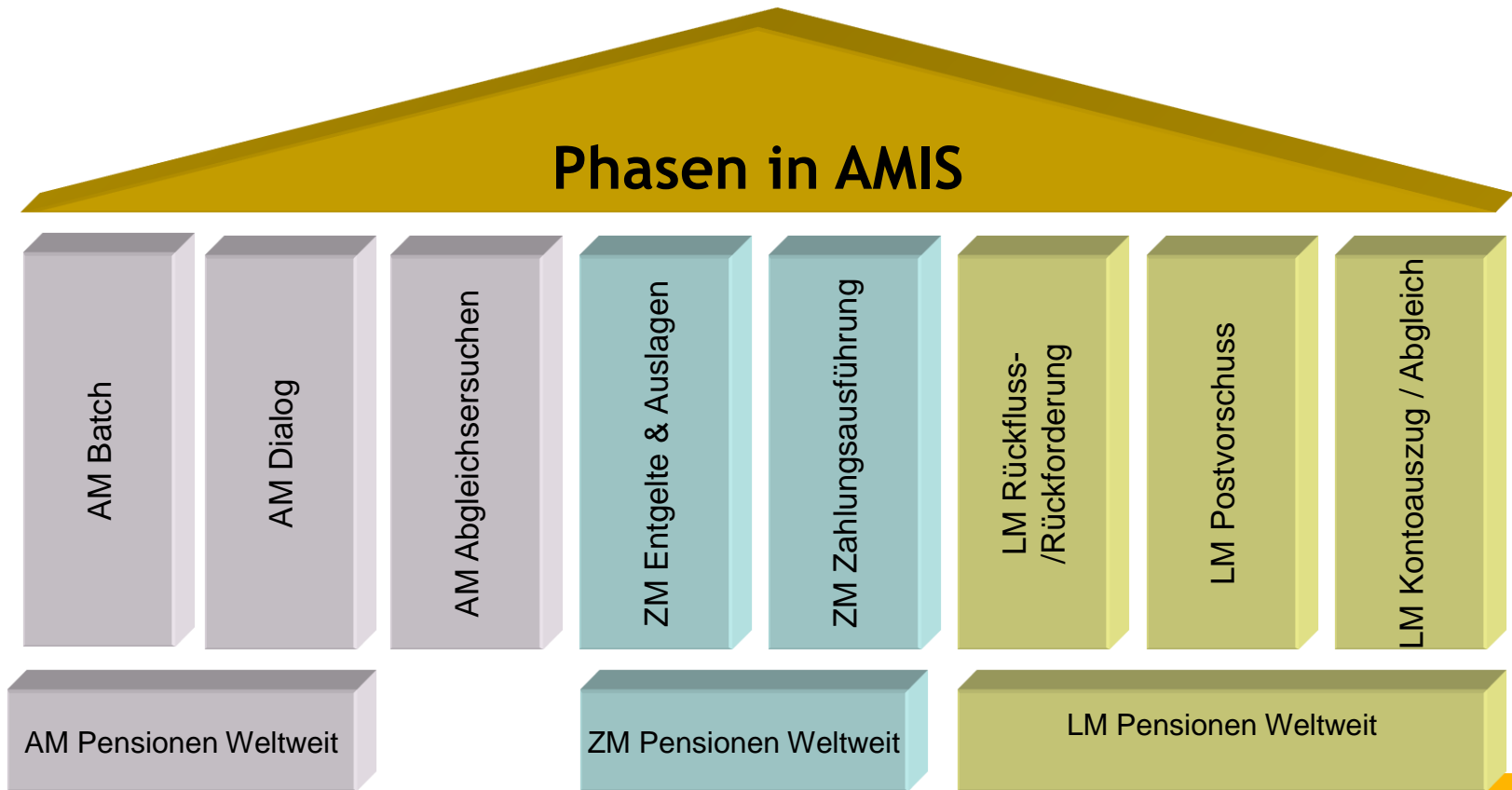
NL Renten Service

- **26 Millionen Konten von ca. 18 Millionen Rentnern**
- **305 Millionen Rentenzahlungen pro Jahr**
- **24 Millionen Rentenberechnungen pro Jahr**
- **17 Millionen Rentenzahlungen ins Ausland pro Jahr**
- **Auszahlung von 195 Milliarden € pro Jahr**

Projektbeschreibung AMIS

- **Änderungsmanagement und Informationsaustausch mit Leistungsträgern (DRV/Bund et al.)**
 - großteils dateibasierte Satzverarbeitung
 - Größenordnung pro Monat:
 - Auflieferung von ca. 0,5 - 1 Mio. Datensätze
 - Versand von bis zu mehreren Mio. Datensätzen an die Leistungsträger

Projektbeschreibung AMIS



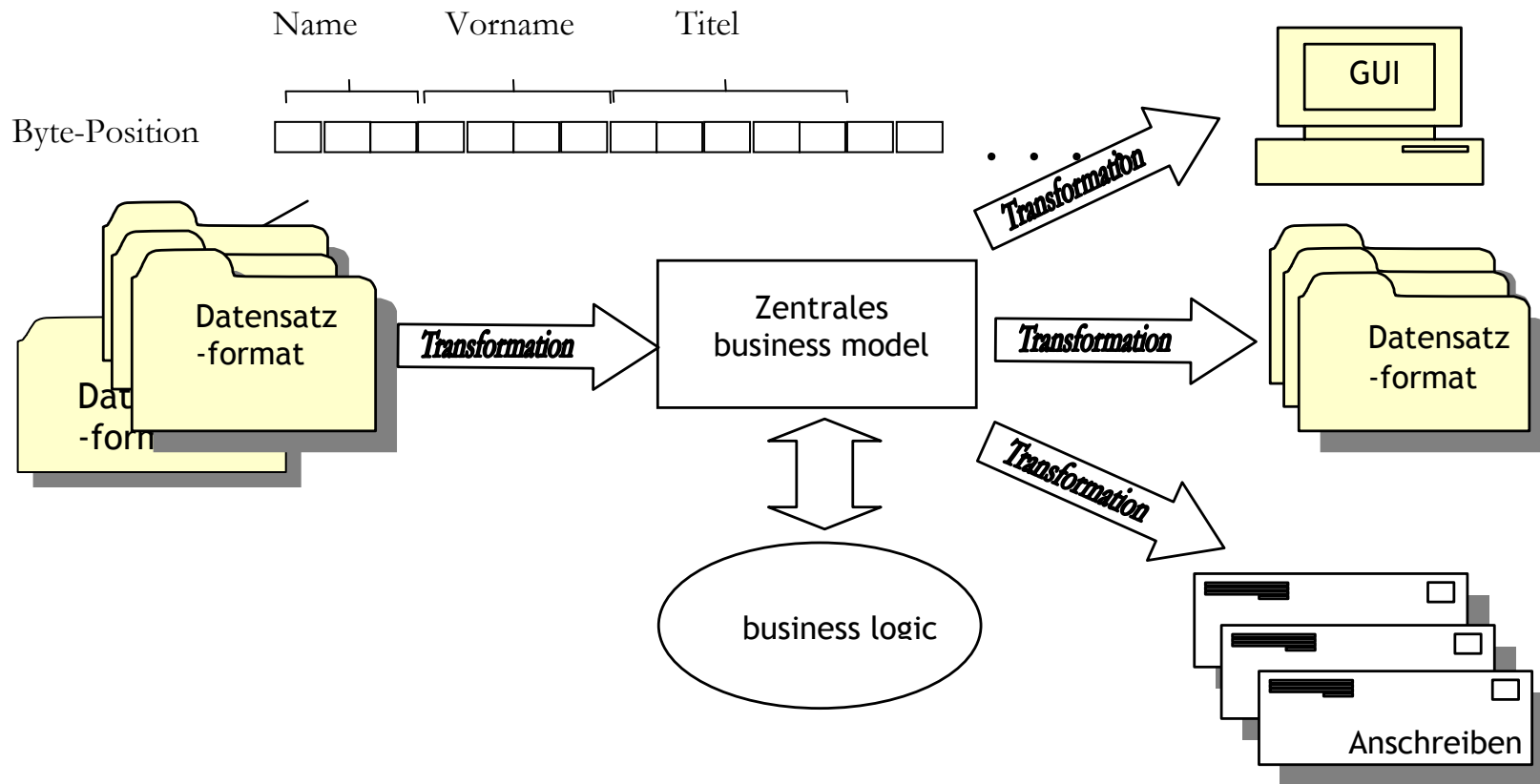
Projektbeschreibung AMIS

- **Phase 1 : AMIS – Veränderungsdienst abgeschlossen und im Wirkbetrieb**
- **Mehrjähriger Zeithorizont des Gesamtprojektes**

Problembeschreibung AMIS

- **Zentrales Businessmodell, welches**
 - durch 3 Datei-/ Datensatzformate gespeist wird
 - in mehr als 15 Datensatzformate transformiert wird
 - in Web-GUIs dargestellt wird
 - in mehr als 10 Kundenanschriften transformiert wird
- **>30 Formate beziehen sich auf das Businessmodell**
 - je Format mehr als 1000 Einzelinformationen

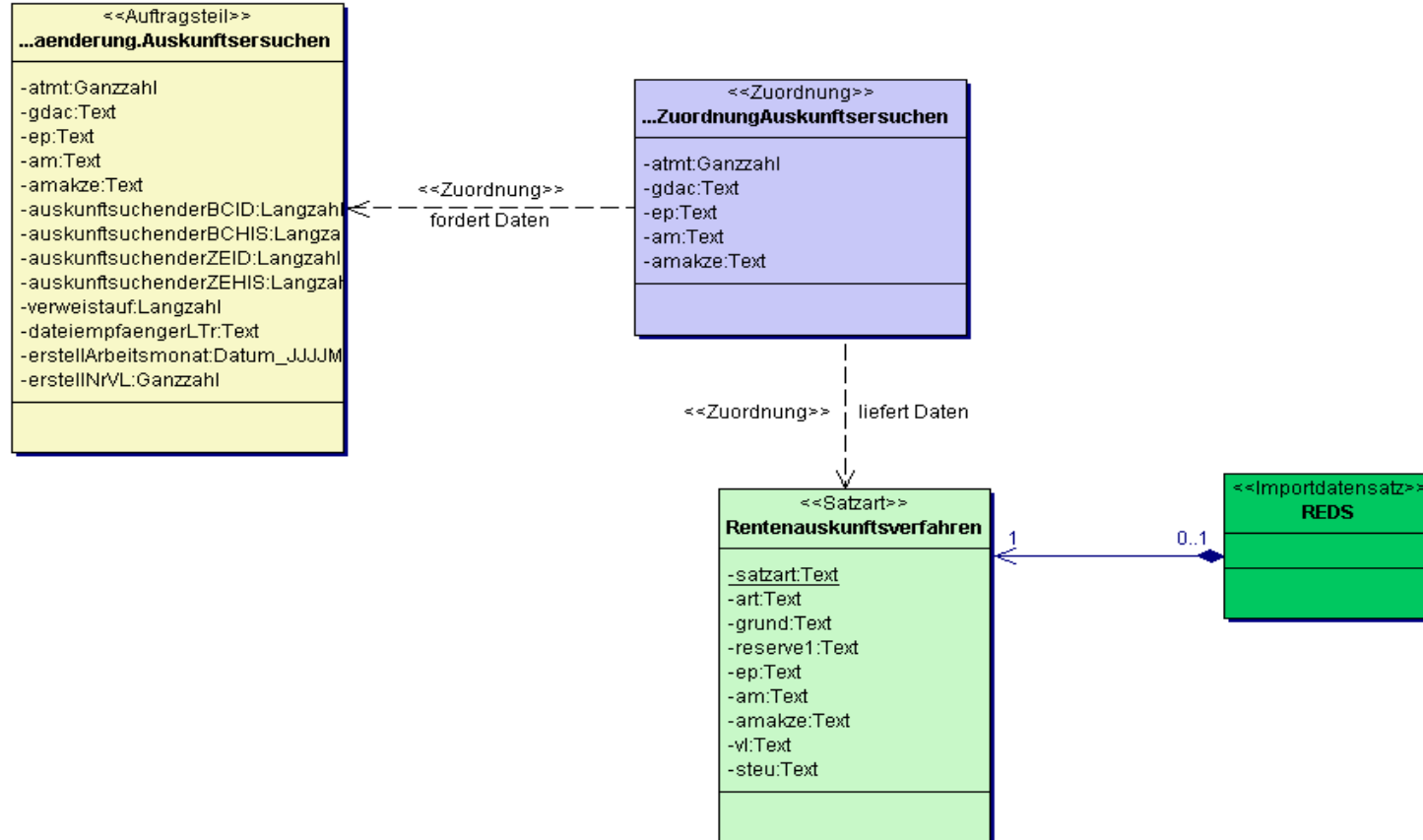
Problembeschreibung AMIS



Motivation für MDSD in AMIS

- **Familie von Softwaresystem mit struktureller Gemeinsamkeit**
 - Gemeinsamkeit ist modellierbar
 - Gemeinsamkeiten sind in jeder Phase (von AMIS) zu finden
- **Zentrales Management der Detailinformationen**
 - Dokumentation und Management von Abhängigkeiten
 - Detailinformationen nicht doppelt in Analyse und Implementierung
- **Große Anforderung an Qualität**
 - Qualitätssicherung durch Generierungsansatz

DSL in AMIS



DSL in AMIS

- **DSL - Metamodell mittels UML**
- **Unsere DSL unterstützt programmierbare Erweiterungen des Modells**

DSL in AMIS

- + **Ermittlung der DSL durch Referenzmodell**
 - Ermittlung des Metamodells (und der DSL) anhand einer Referenzmodellierung
 - damit waren ca. 80% der DSL erfasst
 - die restlichen 20% ergaben sich innerhalb eines Jahres durch neue Anforderungen seitens der Modellierung
 - seitdem ist DSL stabil
- **Prozess zur Ermittlung der DSL war nicht vorgegeben**
 - schlecht planbar

Generatoren in AMIS

- **Generatoren auf nicht ausgereifter DSL aufgesetzt**
 - Generatoren und Framework wurden zu früh parallel zum Referenzmodell umgesetzt
→ führte zu unnötigem Refactoring
 - Geduld mitbringen, bis DSL/Metamodell eine zufriedenstellende Reife erreicht hat

Softwarearchitektur in AMIS

Ermittlung, Validierung und Wartung der Zielplattform

- + Prototyp für Generatoren und Framework
- + Implementierung einer Referenzarchitektur
 - + Validierungen von DSL-Erweiterungen
 - + Referenzarchitektur wird gepflegt

Testverfahren in AMIS

● Modell

- Modellvalidierung (syntaktische Qualitätssicherung)
- derzeit mit eigen-implementierter Validierung

● Generatoren / Framework

- Validierung der DSL/Generatoren durch Referenzmodell und zugehörige Unit-Tests

Testverfahren in AMIS

- **Programmierte Modellerweiterungen**
 - Entwicklung einer Simulationssoftware für aus-/ eingehende Datenformate
- **Generate**
 - Integrations- und Akzeptanztests auf großer Datenmenge
- **Modell**
 - Inhaltliche Qualitätssicherung auf Modellebene

Rollen und deren Anforderungen

Rollen für

● Domäne / Modellierung

- DSL-Analyse
- Modellierung
- Architektur

● Generatoren / Transformationen

- Architektur, Generatorentwicklung, Frameworkentwicklung
- Generierung

Rollen in AMIS

- **Es wurden Rollen in Personalunion besetzt**
 - Architektur, Generatorenentwicklung
 - DSL-Analyse, Modellierung
 - Generierung besetzen spezielle EntwicklerInnen
- **Sehr erfahrene Köpfe für DSL-Analyse, Generatoren- und Frameworkentwicklung**
 - Gefahr von ‚Kopfmonopolen‘ in kleinen Teams
 - + Große Flexibilität gerade in ‘Aufbruchsphase‘

Modellierung in AMIS

- **Modellierung mündet unmittelbar in Software:**
 - + Verlangt große Exaktheit der Analyse
 - Laufzeitaspekte fließen in Modellierung ein
- **Pair-Modelling (Modellierung + Entwicklung)**
- **Betreuung der DSL durch Architektur und DSL-Analyse**
 - + Erweiterungen der DSL werden erkannt
 - + Ausdehnung der Software-Systemfamilie

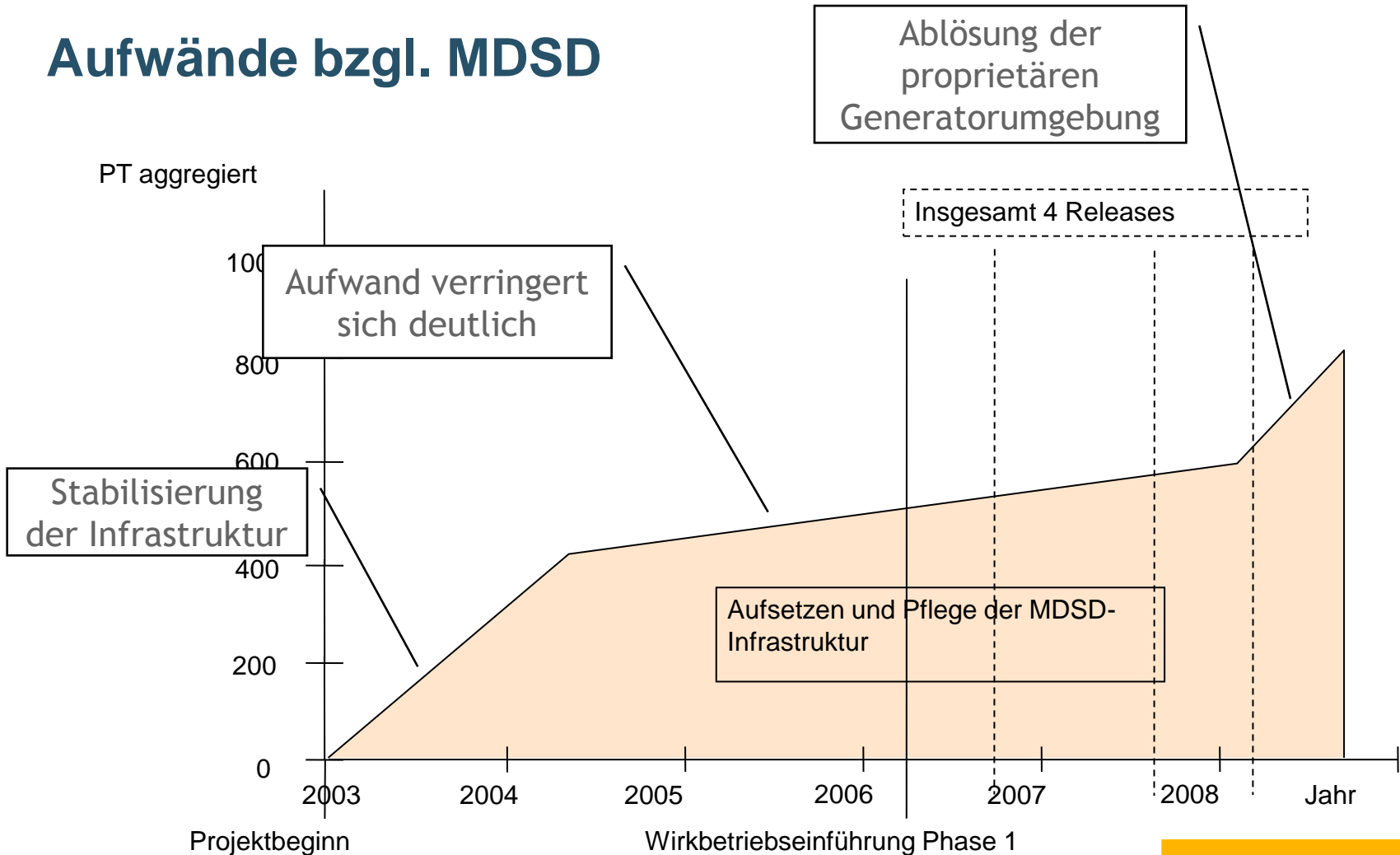
Generierung in AMIS

- **Modell beeinflusst mehrere Teilprojekte daher explizite Generierung:**
 - Generierung obliegt der Entwicklung
 - Generierung ist nicht Bestandteil des Build-Prozesses
 - Versionierung der Generate pro Teilprojekt
 - auf separatem Rechner inkl. lokaler Modultests
 - explizite Freigabe von Versionen der Generate

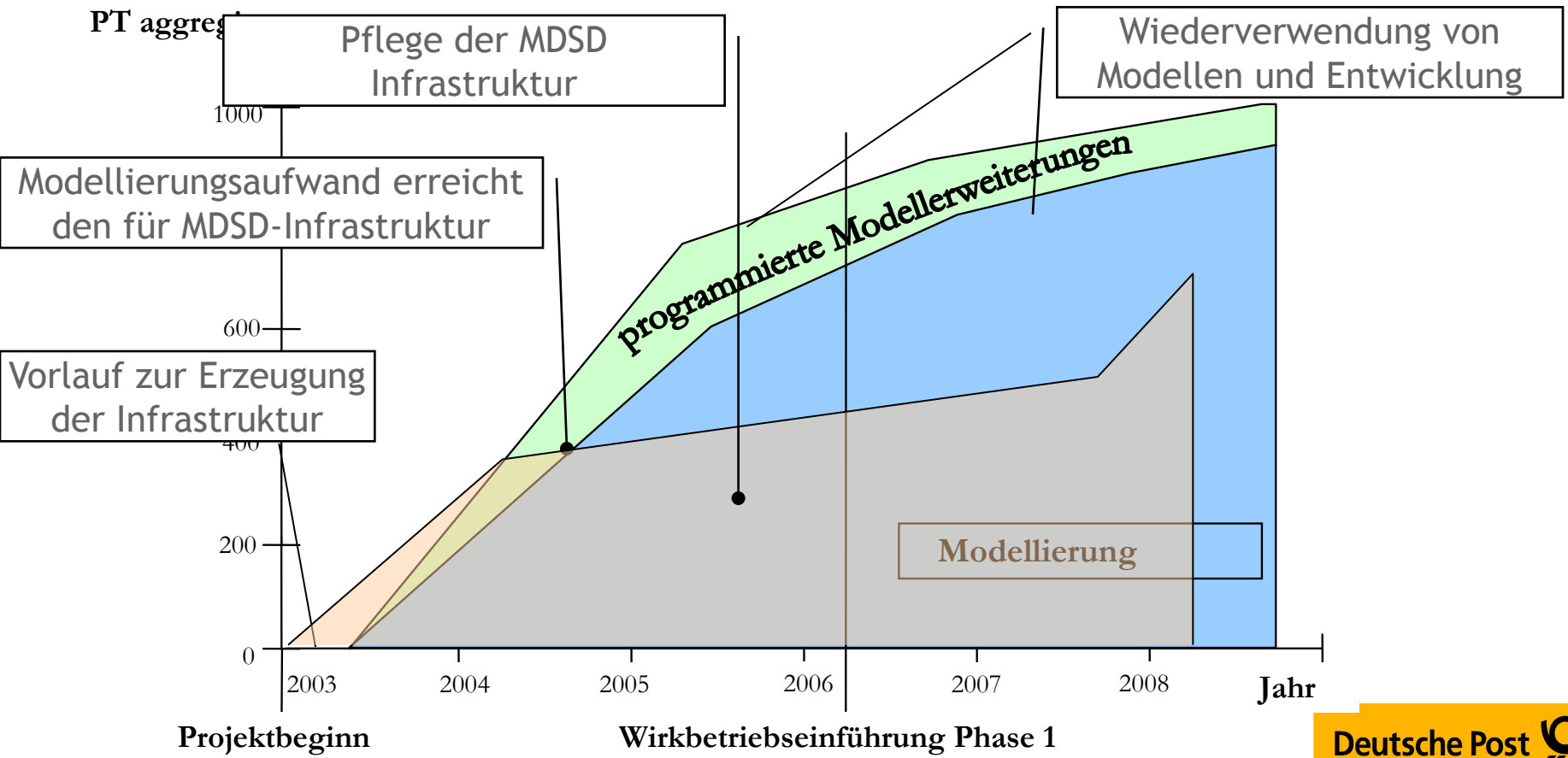
Einsatz von MDSD in AMIS

- **Einsatz von MDSD > 40% (bezogen auf Modelle)**
- **Mengengerüste**
 - derzeit ca. 100.000 Modellelemente, davon < 2% (ca. 1.900) programmatisch ergänzt
- **Kleines Entwicklungs- und Modellierungsteam (< 10 Personen)**
 - davon 3 externe Kräfte

Aufwände bzgl. MDSD



Aufwände bzgl. MDS



Aktueller Stand in AMIS

● Stabilität / Verlässlichkeit

- DSL ist seit über einem Jahr stabil
- Generatoren sind stabil und zuverlässig
- Framework ist bis auf kleinere Änderungen stabil
- Entwicklungsprozess hat sich stabilisiert und bewährt

Risiken in AMIS

- **Methodik ist pflegeintensiv**
 - muß konsequent gelebt werden
 - neigt zu Erosion
 - erfordert Verständnis der Beteiligten am Prozess
 - Weg zurück ist schwer
- **domänen-zentrierte MDSD greift früh im Entwicklungsprozess**
 - systematische Fehler haben gravierende Auswirkungen
- **Bindung der Generatoren an proprietäres Tool**
- **Komplexität der DSL muss begrenzt bleiben**
 - Gefahr unklarer Modellierungsregeln
 - Gefahr impliziter Abhängigkeiten zwischen Modellierungselementen

Sicht des Projektmanagements

● Integration von Modellierung und Implementierung

- erschwert Zuordnung der Verantwortlichkeiten für einzelne Arbeitsergebnisse
 - Termin- und Aufwandskontrolle wird schwieriger
- klassische Wasserfallmethode ist fast unmöglich
 - Iterationen zwischen Modellierung und Entwicklung

Bewertung von MDSD nach 5 Jahren

- **Erfolgreiches Management der Mengen an Detailinformationen**
- **Robuste Software durch Framework-Einsatz**
- **Wartungsaufwand reduziert sich um mehr als 50 %
(konservativ geschätzt)**
 - Vergleich der Aufwände bereits abgelöster Anwendung zu nicht abgelösten Anwendungen

Resümee

In unserem Fall und bei unserer Ausgangssituation hat sich der Einsatz von *Modellgetriebener Softwareentwicklung* gelohnt.



Gesellschaft für
Informations- und
Kommunikationssysteme mbH

www.iks-gmbh.com

Deutsche Post



RENTEN SERVICE

www.deutschepost.de/rentenservice

Deutsche Post



RENTEN SERVICE