Architekturen betrieblicher Anwendungssysteme

Bewertung von Anwendungslandschaften

Lehrstuhl für Wirtschaftsinformatik
Prozesse und Systeme
Universität Potsdam

Chair of Business Informatics
Processes and Systems
University of Potsdam

Univ.-Prof. Dr.–Ing. habil. Norbert Gronau
Lehrstuhlinhaber | *Chairholder*

Mail August-Bebel-Str. 89 | 14482 Potsdam | Germany
Visitors Digitalvilla am Hedy-Lamarr-Platz, 14482 Potsdam
Tel +49 331 977 3322

E-Mail ngronau@lswi.de
Web lswi.de

Analyseverfahren für Anwendungslandschaften

Planungsphase

Komplexität von Anwendungslandschaften

Analyseverfahren für Anwendungslandschaften

Planungsphase

Komplexität von Anwendungslandschaften

Ziele der Analyse von Architekturlandschaften

Untersuchungs-bereich	Beschreibung des Analyseverfahrens	Typische Fragestellungen
Abhängigkeiten	Verknüpfte Elemente werden aus der Unternehmensarchitektur selektiert	Welche anderen Elemente sind betroffen, wenn wir die Infrastrukturkomponente X ablösen?
Abdeckung	Abdeckung fachlicher Bereiche, z.B. Prozess/Produktmatrix	Welche Redundanzen oder Lücken gibt es bei der IT-Unterstützung für den Prozess X und das Produkt Y und die Organisationseinheit Z?
Schnittstellen	Analyse der Schnittstellen zw. Anwendungssystemen hstl. Art, Anzahl, Komplexität, Häufigkeit/Aktualität, Performance, Stabilität, Verfügbarkeit	Gibt es Brüche bei der Unterstützung des Prozesses X? Sind produktübergreifende Gemeinsamkeiten in Prozessschritten auch übergreifend gelöst?
Heterogenität	Die Heterogenität der IT-Assets in definierten Einsatzfeldern wird analysiert, z.B. Prozess/Produktmatrix.	Anzahl der Entwicklungslinien pro Einsatzfeld Anzahl der Infrastrukturkomponenten pro Zeile
Komplexität	Anzahl der Komponenten in der Unternehmensarchitektur und Anzahl ihrer Beziehungen	Wie viele Anwendungssysteme mit wie vielen Schnittstellen existieren? Wie viele Infrastruktursysteme mit wie vielen Schnittstellen existieren?

Quelle: Niemann 2005, S. 128-130; Hanschke 2023, S. 27ff.

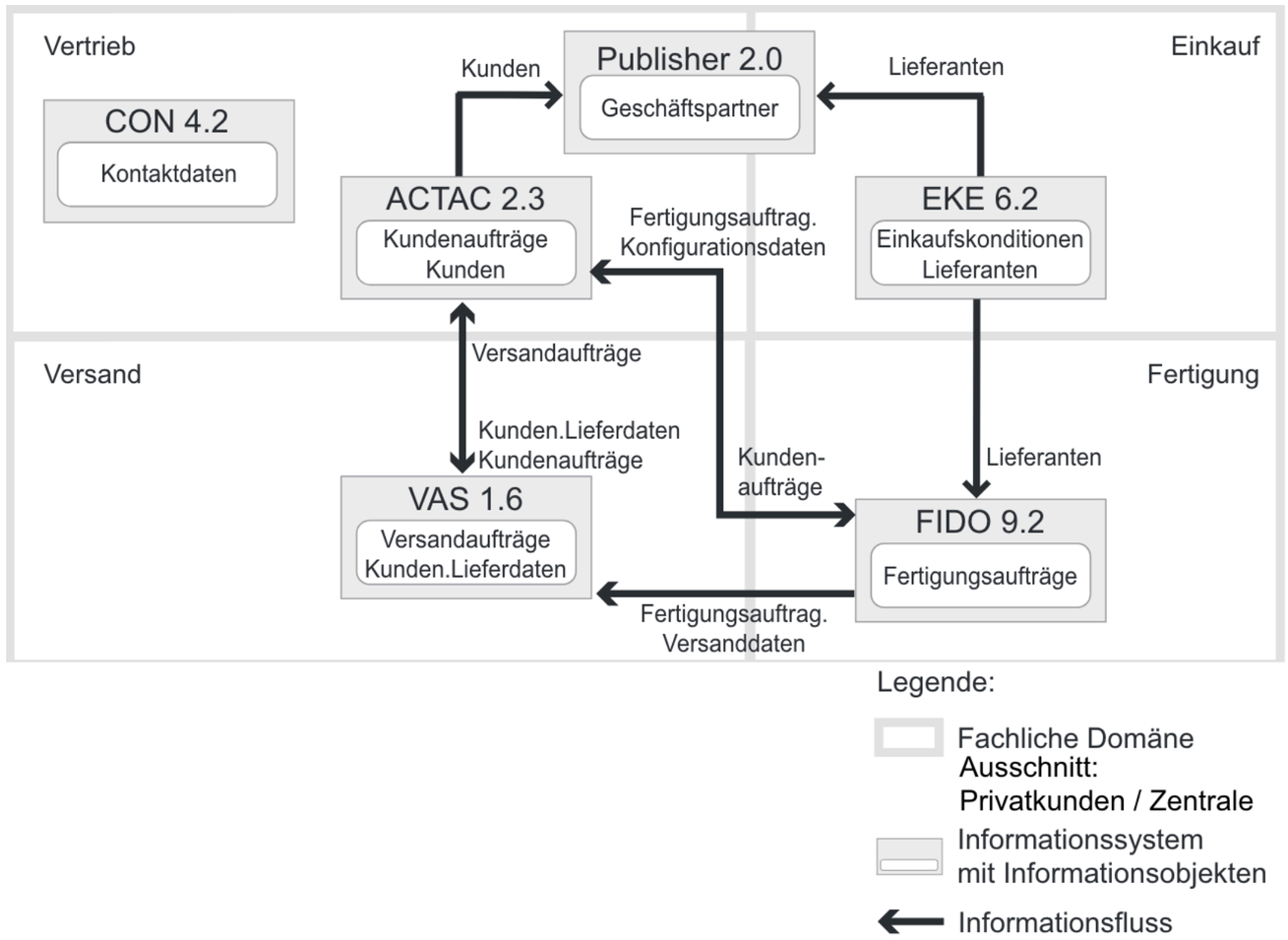
Ziele der Analyse von Architekturlandschaften II

Untersuchungs-bereich	Beschreibung des Analyseverfahrens	Typische Fragestellungen
Konformität	Einhalten von Standards und Ermittlung des Abweichungsgrades. Compliance Rules	Werden existierende Standards eingehalten? Werden die definierten Referenzarchitekturen implementiert? Anteil der Komponenten, die außerhalb des Standards liegen? Werden gesetzliche Vorgaben, Marktstandards und Normen eingehalten?
Kosten	Reporting über kumulierte Erstellungs-, Betriebs- und Wartungskosten.	Welche Kosten sind durchgängig über alle Ebenen der Unternehmensarchitektur mit der IT-Abbildung des Produktes X verbunden?
Nutzen	Nutzenkalkulation z.B. in prozentualem Beitrag zur Erreichung von Unternehmenszielen	Welchen Beitrag zur Erreichung der Unternehmensziele leistet das Anwendungssystem X?

Eine konsolidierte Unternehmensarchitektur ermöglicht umfangreiche Analysen.

Quelle: Niemann 2005, S. 128-130

Analyse der Abhängigkeiten



Mögliche Ergebnisse und Schlussfolgerungen

- Verbindung zwischen verschiedenen Ebenen und Sichten der Anwendungslandschaft
- Verwendung von Managementtools
- Auswirkungen des Wegfalls eines Systems

Abhängigkeitsanalyse ist insbesondere bei Neueinführung oder Entfernung eines Systems grundlegend.

Quelle: Hanschke 2023, S. 382

Abdeckungsanalyse

Aufbau

- Fachliche Gliederung der Anwendungslandschaft
- Produkt/Prozess-Matrix
- Erkennen von Redundanzen und Lücken

Detailuntersuchung bei Lücken

- Gründe für Lücken
- Auswirkung der Lücken auf das Geschäft und die IT (ev. Medienbrüche)
- Entstehende Risiken
- Kosten zur Beseitigung der Lücken

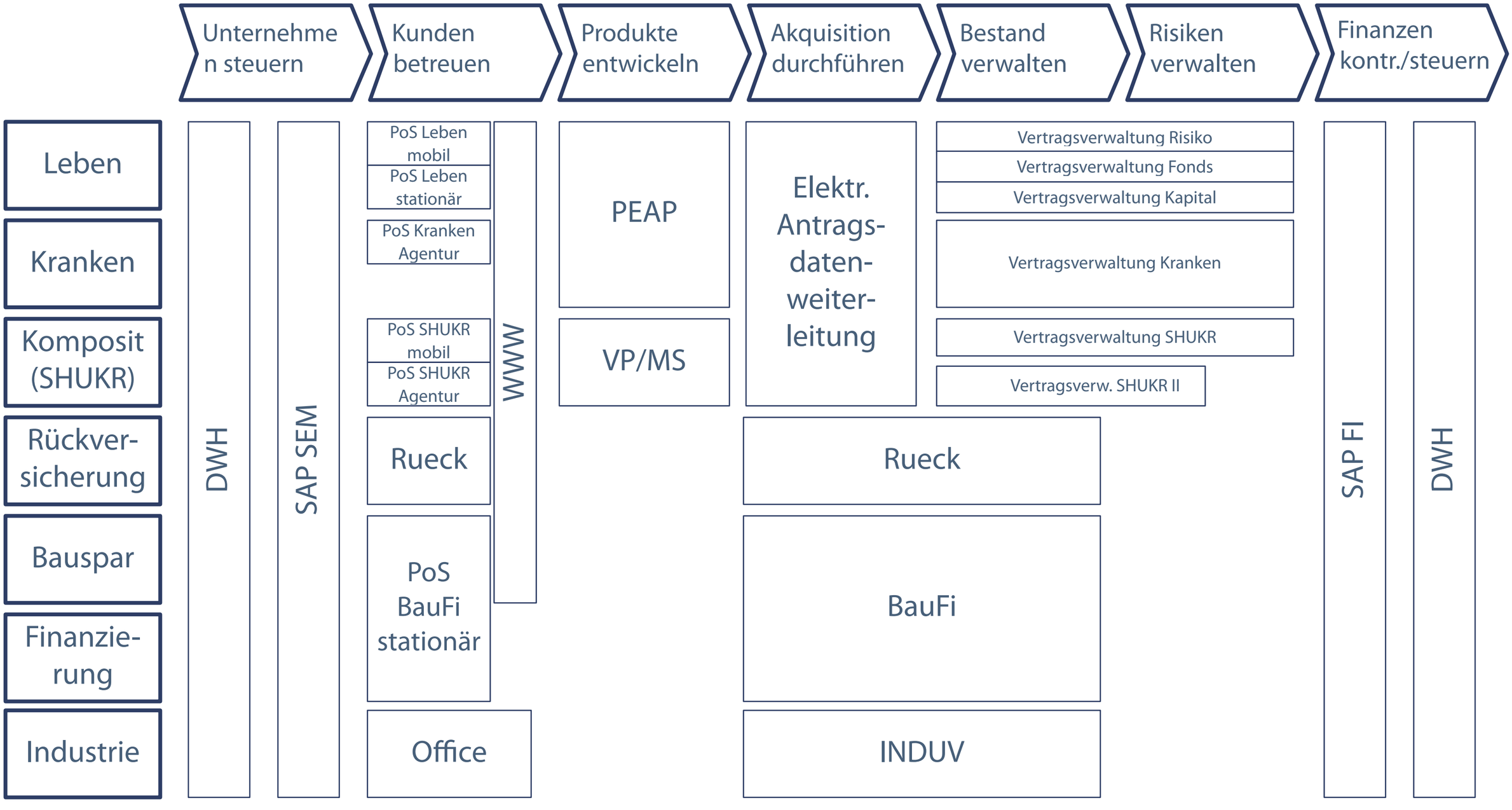
Vertiefende Untersuchung hinsichtlich

- Negativer Auswirkung der Redundanzen auf "Time to Market" bei Einführung neuer Produkte
- IT-Kosten zur Unterhaltung der redundanten Systeme
- Fachliche Argumente für den Erhalt der Redundanzen
- Ist eine Zusammenfassung möglich

Die Abdeckungsanalyse erlaubt das Erkennen von Redundanzen und Lücken in der Anwendungslandschaft.

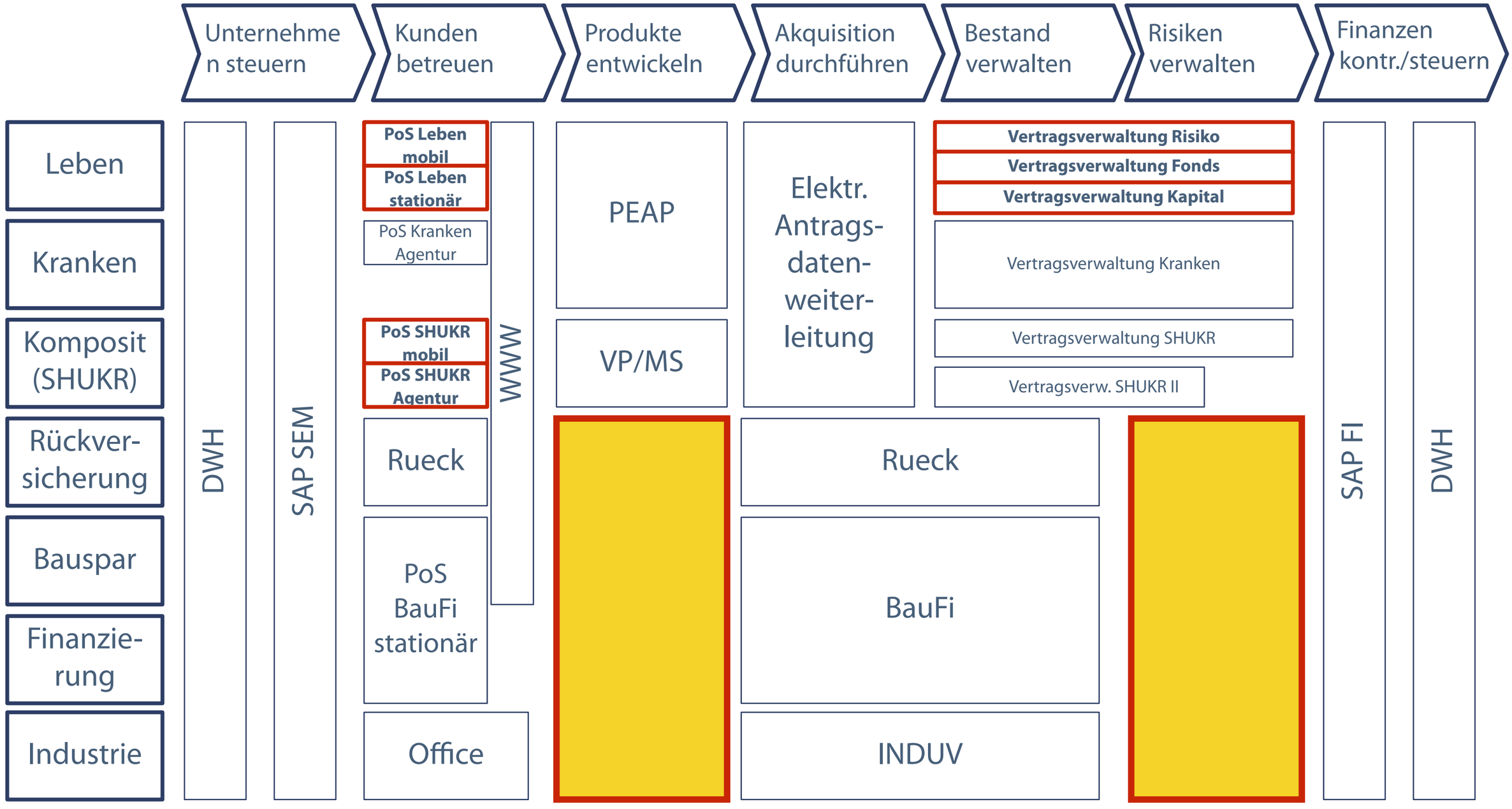
Quelle: Hanschke 2023, S. 627

Abdeckungsanalyse - Redundanzen und Lücken



Quelle: Niemann 2005, S. 132

Abdeckungsanalyse - Redundanzen und Lücken



Quelle: Niemann 2005, S. 132

Analyse der Schnittstellen

Anzahl der Schnittstellen zwischen den Anwendungssystemen

- Maximale Anzahl von Schnittstellen N mit n Systemen:
  N=(n*(n-1))/2

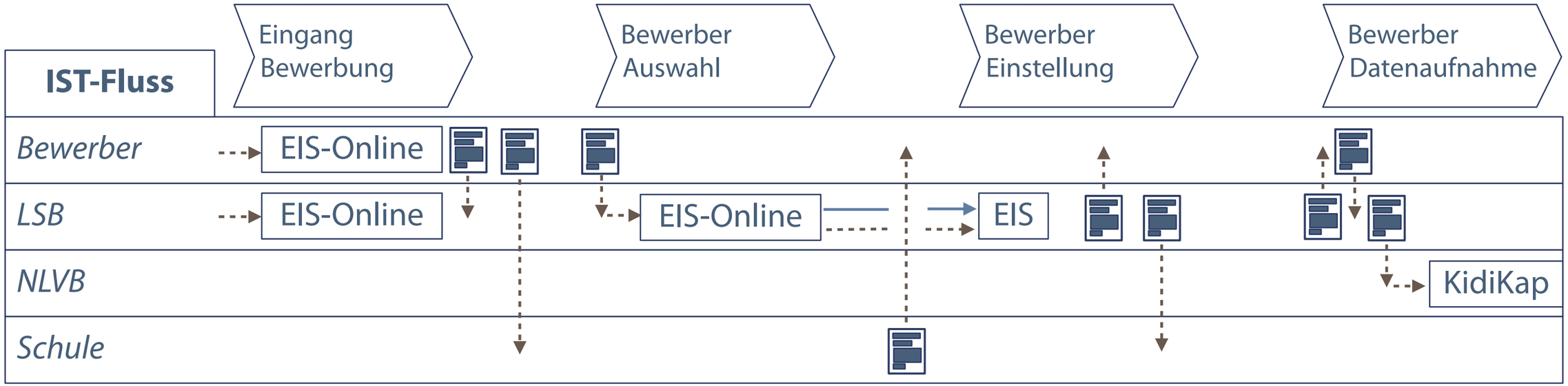
Untersuchung

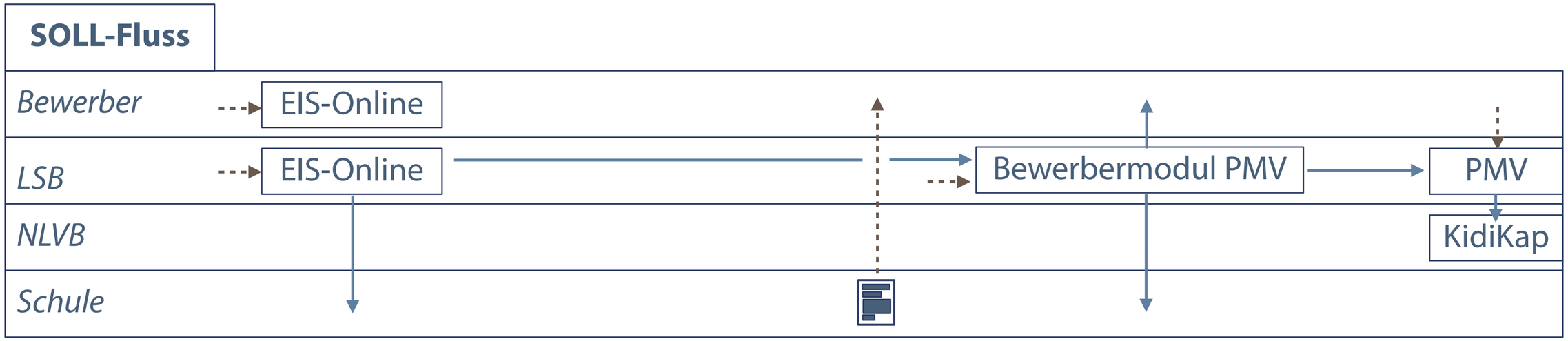
- Entfallen von Systemen durch anderen Zuschnitt
- Zusammenfassung von Systemen
- Überflüssige Systeme eliminieren
- Häufigkeit der Schnittstellennutzung
- Aktualisierungsanforderung
- Stabilitätsanforderungen
- Verfügbarkeitsanforderung

Analysierbar durch Softwarekarte mit Datenfluss als Verbindung

Die Analyse der Schnittstellen gibt Aufschluss über die Notwendigkeit von EAI-Projekten.

Daten- und Dokumentenfluss im Prozess: Neueinstellung in den Schuldienst (Land Niedersachen)



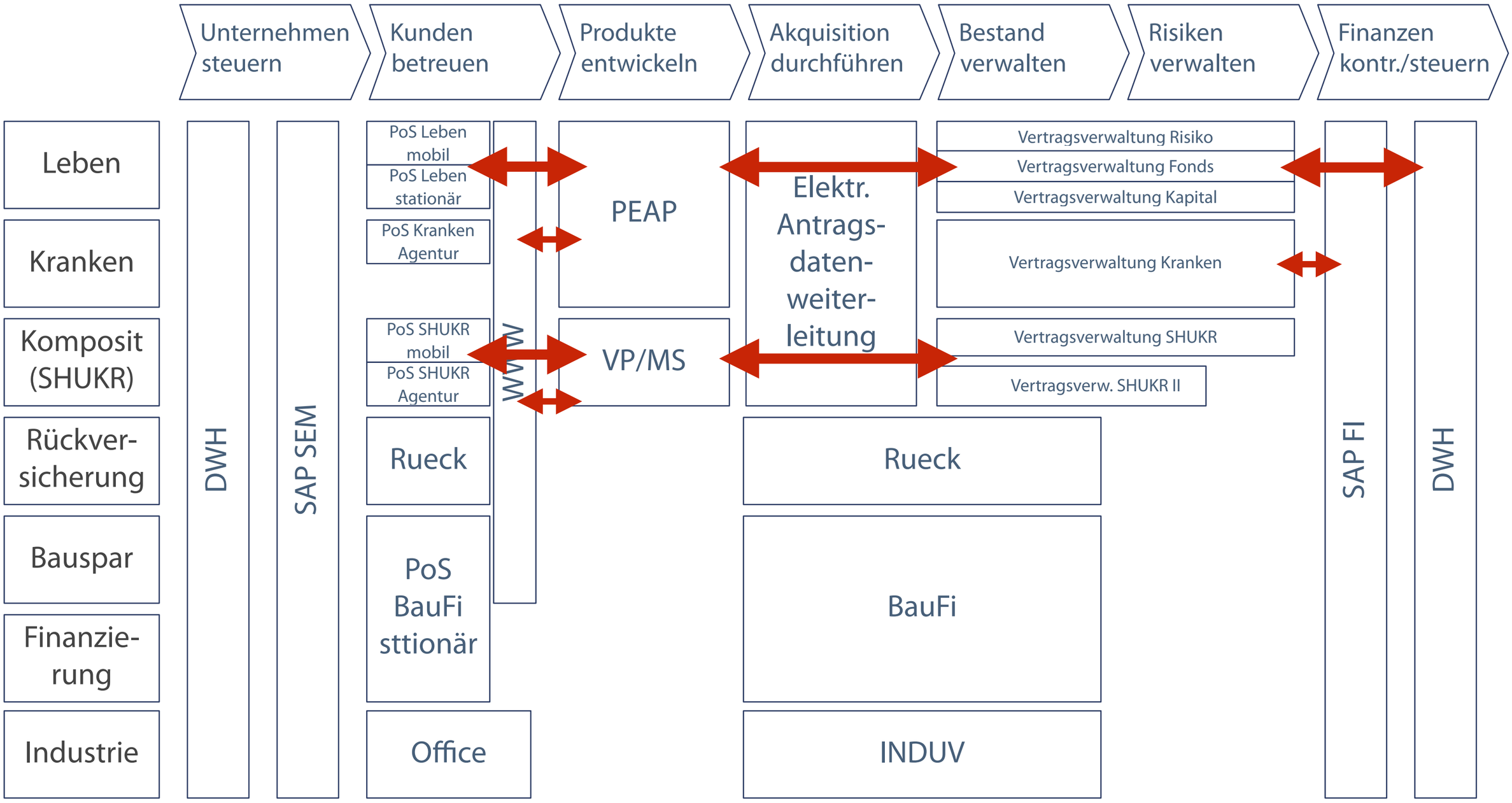


Papierdokumente

Dokument / Datenaustausch per Post / manuelle Eingabe

automatischer Datenaustausch/ elektronische Nachrichten

Schnittstellenanalyse



Quelle: Niemann 2005, S. 132

Analyse der Heterogenität

Ablauf

- Analysemittel ist wieder die Matrix, aber Ersetzung der Produkte durch Organisationseinheiten
- Zuordnung von Anwendungssystemen zu Geschäftsprozessen und Organisationseinheiten
- Analyse auf Entwicklungslinien der Anwendungssysteme
- Hohe Anzahl von Entwicklungslinien pro Zelle = Indikator für Heterogenität
- Zusätzlich Erhebung von Werten zur Verbreitung und absoluter Anzahl der Systeme

Untersuchungen

- Verhältnis von Entwicklungslinien zu Organisationseinheiten
- Begründung ev. durch organisatorische Unterschiede
  Wildwuchs?
- Eigenheiten einzelner Organisationseinheiten, die auf ihr eigenesAnwendungssystem bestehen?

Die Analyse der Heterogenität ist der eigentliche Kern des Architekturmanagements.

Analyse der Heterogenität

	Unternehmen steuern	Kunden betreuen	Produkte entwickeln	Neugeschäft aquirieren	Bestand verwalten	Risiken steuern	Finanzen kontrollieren und steuern
Leben							
Kranken							
Komposit							
Rückver-sicherung							
Bauspar							
Finanzie-rung							
Industrie							

Entwicklungslinie	Symbol						
Cobol		C++		DWS		Office	
Java		Smalltalk		SAP		.NET	

Analyse der Heterogenität im Infrastrukturbereich

Klassifikation aller Infrastruktursysteme:

- Z.B.: Datamanagement, Security Management, Communication, Operating Systems, User Management, and Support, Administration and Operation
- Erkennen von Überpopulation

Erstellung eines „Warenkorbs“ - Gliederung der Systeme in Support Levels:

- Level a: Volle Unterstützung der enthaltenen Infrastrukturkomponenten
- Level b: Volle Unterstützung für Produktion, keine Entwicklungsunterstützung
- Level c: Eingeschränkte Unterstützung für Produktion

Die Verbindung von Heterogenität der Anwendungslandschaft und Infrastrukturkomponenten läßt unnötigen Ballast erkennen.

Analyse der Konformität

Compliance Rules Prüfungen (entsprechen Existenzbedingungen)

- Dokumentation zu definierten Elementen der Unternehmensarchitektur
- Verfahrensbeschreibungen
- Gesicherte Verfahren für Backup, Recovery, Autorisierung etc.
- Security Policy

Konformität für IT-Governance

- Konformität zu den Referenzarchitekturen
- Ausschließlicher Einsatz von im Warenkorb befindlichen Entwicklungswerkzeugen
- Unternehmensinterne Zertifizierung für alle Infrastrukturkomponenten
- Konvergenz der Entwicklungslinien hin zur Referenzarchitektur

Die Prüfung auf Konformität ist eine Kernaufgabe des Architekturmanagements.

Analyse der Kosten

Vorgehen

- Genaue Aufschlüsselung der Kosten nach
  Beschaffungsinvestition, (I)
  Wartungskosten (W)
  Betriebskosten (B)
  Differenziert nach Personalkosten(pk) und
  Sachkosten(sk)
- Initialinvestition herunterbrechen auf jährliche
  Abschreibungskosten (Ad)
- Jährliche Kosten
  $K_{ASi} = ((Isk_{ASi} + Ipk_{ASi}) / Ad) + Wpk_{ASi} + Wsk_{ASi} + Bsk_{ASi} + Bpk_{ASi}$
- Werte pro AS eintragen in Matrix

Die Kostenanalyse hilft bei der Entscheidung zur Projektaktivität.

Kostenanalyse
Beispiel für die initiale Beschaffungsinvestition

Z < 1.000 T EUR
1.000 <= Z < 3.000
Z >= 3.000 T EUR

Unternehmen steuern
Kunden betreuen
Produkte entwickeln
Akquisition durchführen
Bestand verwalten
Risiken verwalten
Finanzen kontr./steuern

Leben
Kranken
Komposit (SHUKR)
Rückver-sicherung
Bauspar
Finanzie-rung
Industrie

DWH
SAP SEM

PoS Leben mobil
PoS Leben stationär
PoS Kranken Agentur
PoS SHUKR mobil
PoS SHUKR Agentur
Rueck
PoS BauFi sttionär
Office

WWW

PEAP
VP/MS

Elektr. Antrags-daten-weiter-leitung

Rueck
BauFi
INDUV

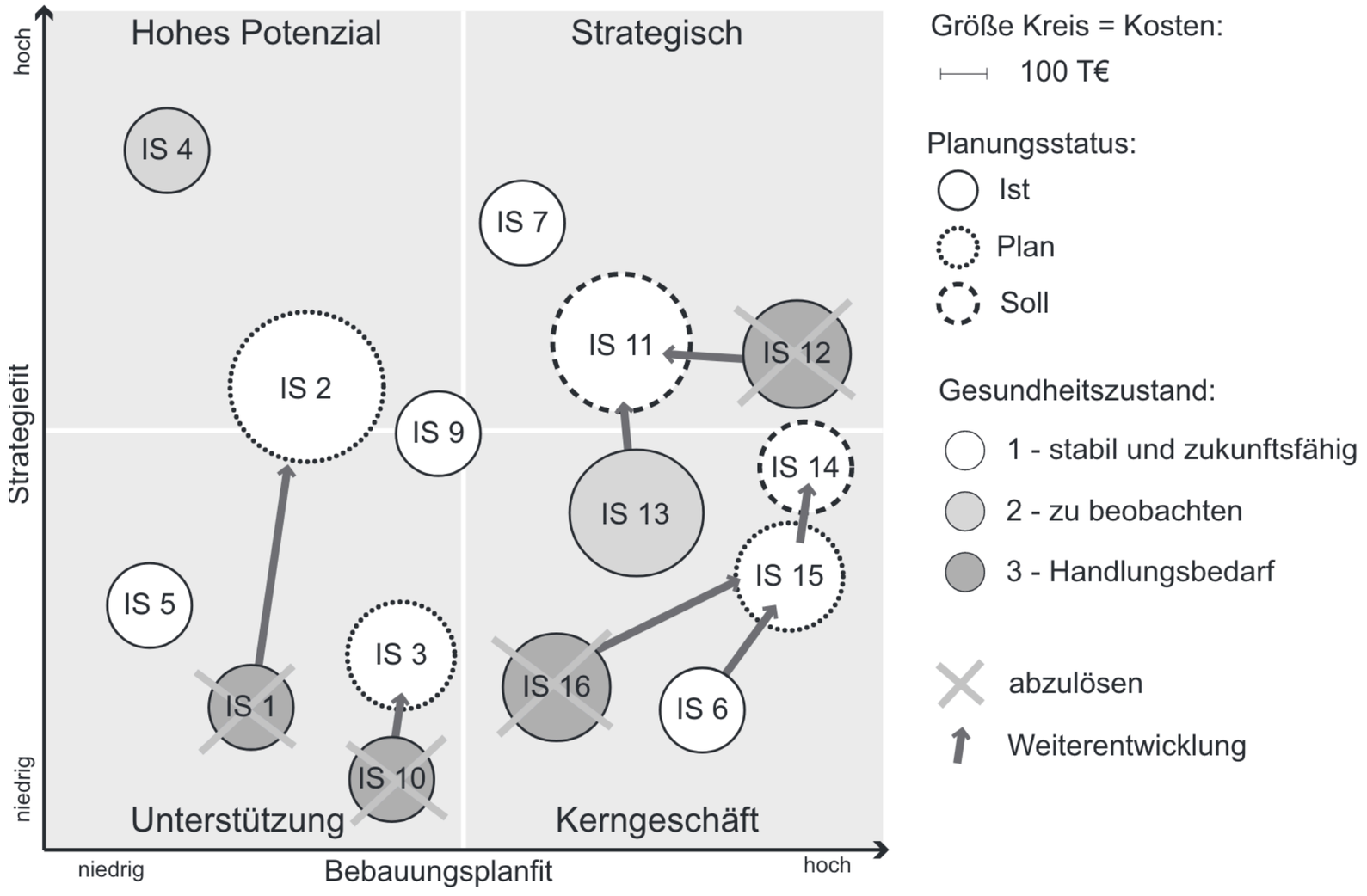
Vertragsverwaltung Risiko
Vertragsverwaltung Fonds
Vertragsverwaltung Kapital
Vertragsverwaltung Kranken
Vertragsverwaltung SHUKR
Vertragsverw. SHUKR II

SAP FI
DWH

Analyse der Kosten/Nutzen und Risikos von IT-Systemen im Bebauungsplan

Strategische Entscheidungshilfe

- Über die Ermittlung der Kosten können strategische Entscheidungen bezüglich der Systeme getroffen werden



Quelle: Hanschke 2023, S. 639f.

Analyse des Nutzens

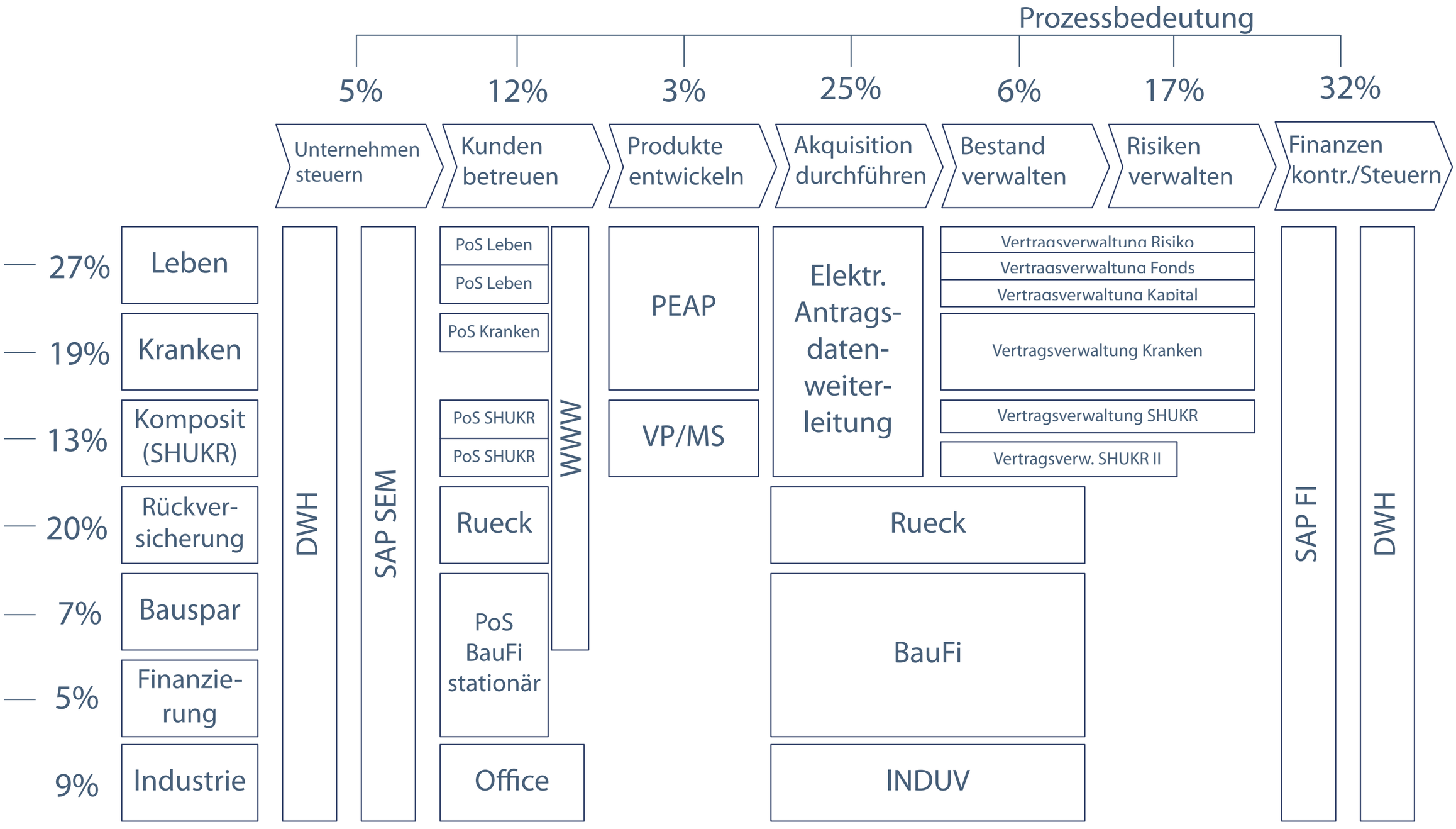
- Nutzenanalyse dient der Priorisierung von zu ergreifenden Maßnahmen
- Kaum Metriken
- Identifikation der Prozesse mit hoher Bedeutung für den Unternehmenserfolg und Produkte mit aktuell hohem Umsatzanteil
- Herleitung über Matrix möglich

Verschiedene Herangehensweisen

- Summe der Investitionen
- Bewertet durch Nutzer oder Betreuer
- Wirkung auf Unternehmensziele
- Unterstützungsgrad für die Geschäftsprozesse
- Bewertung des max. Schadens bei Ausfall

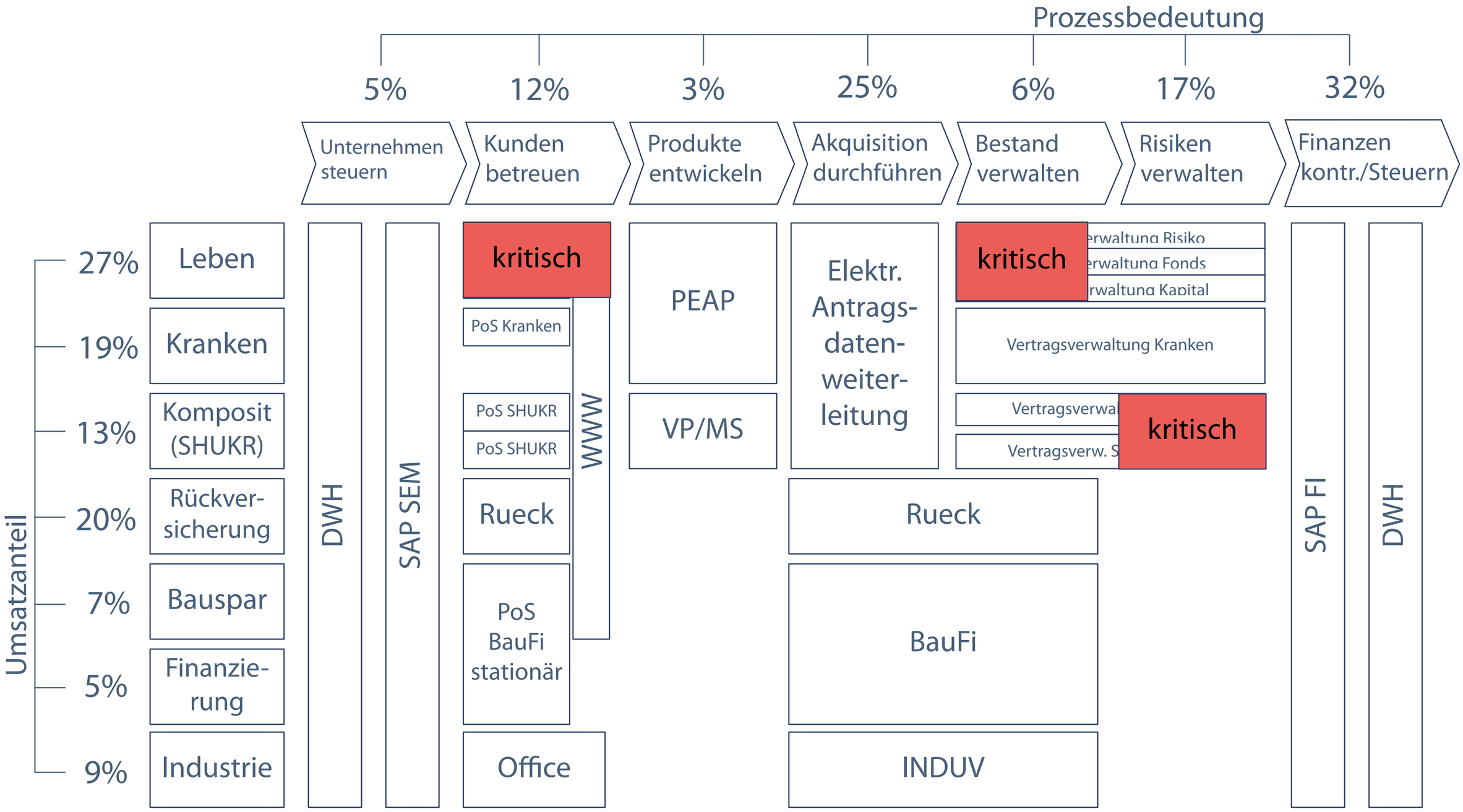
Eine Nutzenanalyse ist auf unterschiedlichen Wegen möglich und kaum operationalisierbar.

Analyse des Nutzens
Beispiel



Quelle: NIemann 2005, S. 132

Analyse des Nutzens
Beispiel



Quelle: Niemann 2005, S. 132

Analyseverfahren für Anwendungslandschaften

Planungsphase

Komplexität von Anwendungslandschaften

IT-Bebauungsplan

Aufbau und Inhalt eines IT-Bebauungsplans

- Planung der Infrastrukturlandschaft (technische Ausrichtung)
- Planung der Anwendungslandschaft (fachliche Ausrichtung)
- Notwendige Ergänzung zum Portfoliomanagement
- Dient der Zukunftssicherheit und Stabilität
- Beseitigung unnötiger Heterogenität
- Erstellt den SOLL-Zustand der Anwendungslandschaft

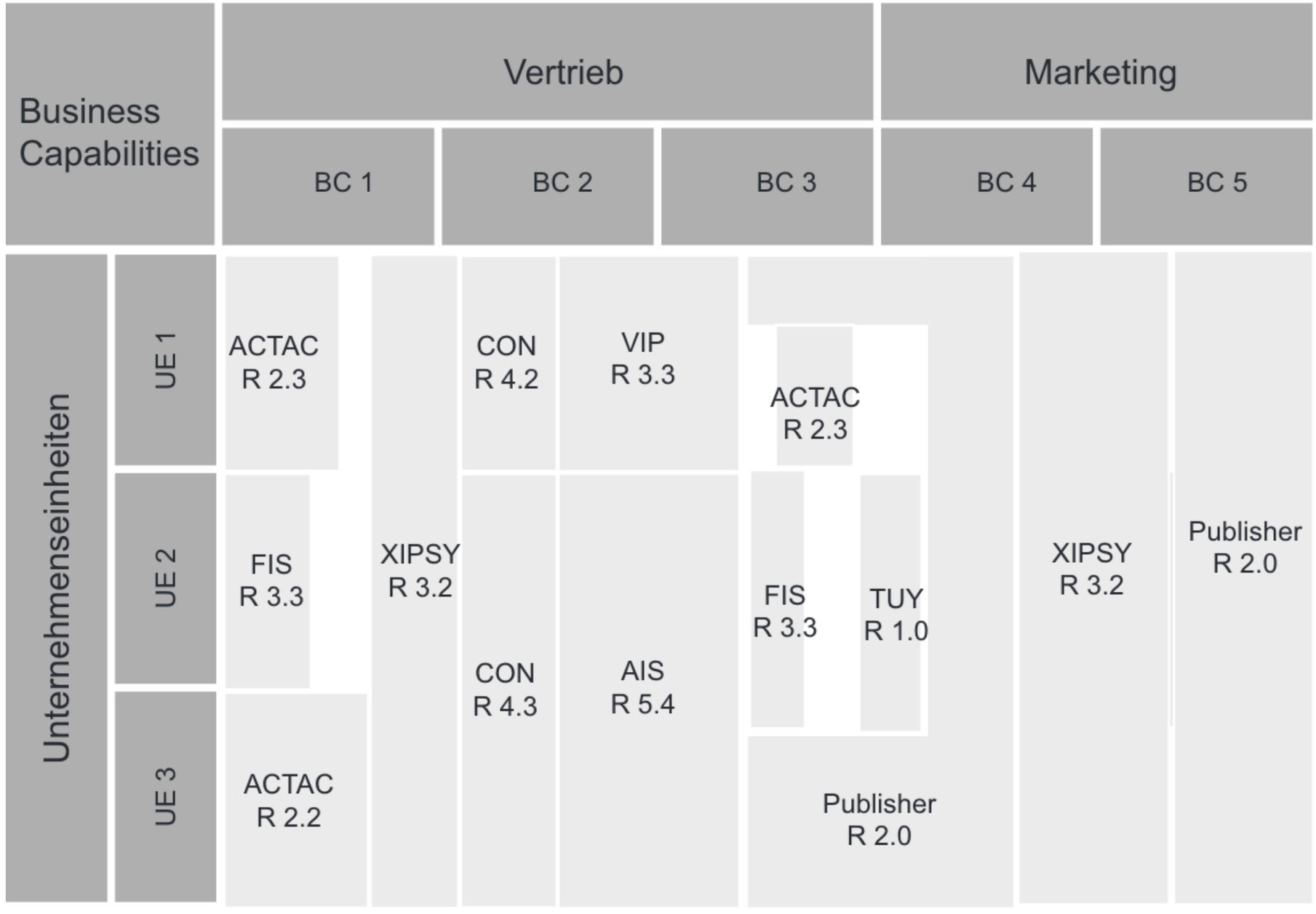
Übersicht zur Gesamtheit der Anwendungssysteme mit Zuordnungen

- Zum Geschäftsprozess/Teilprozess
- Zu den implementierten Geschäftskomponenten
- Zu den beinhalteten Softwarekomponenten
- Zu den genutzten Infrastrukturkomponenten
- Zu den Organisationseinheiten

Nutzung der Erkenntnisse und Ergebnisse der Analysephase

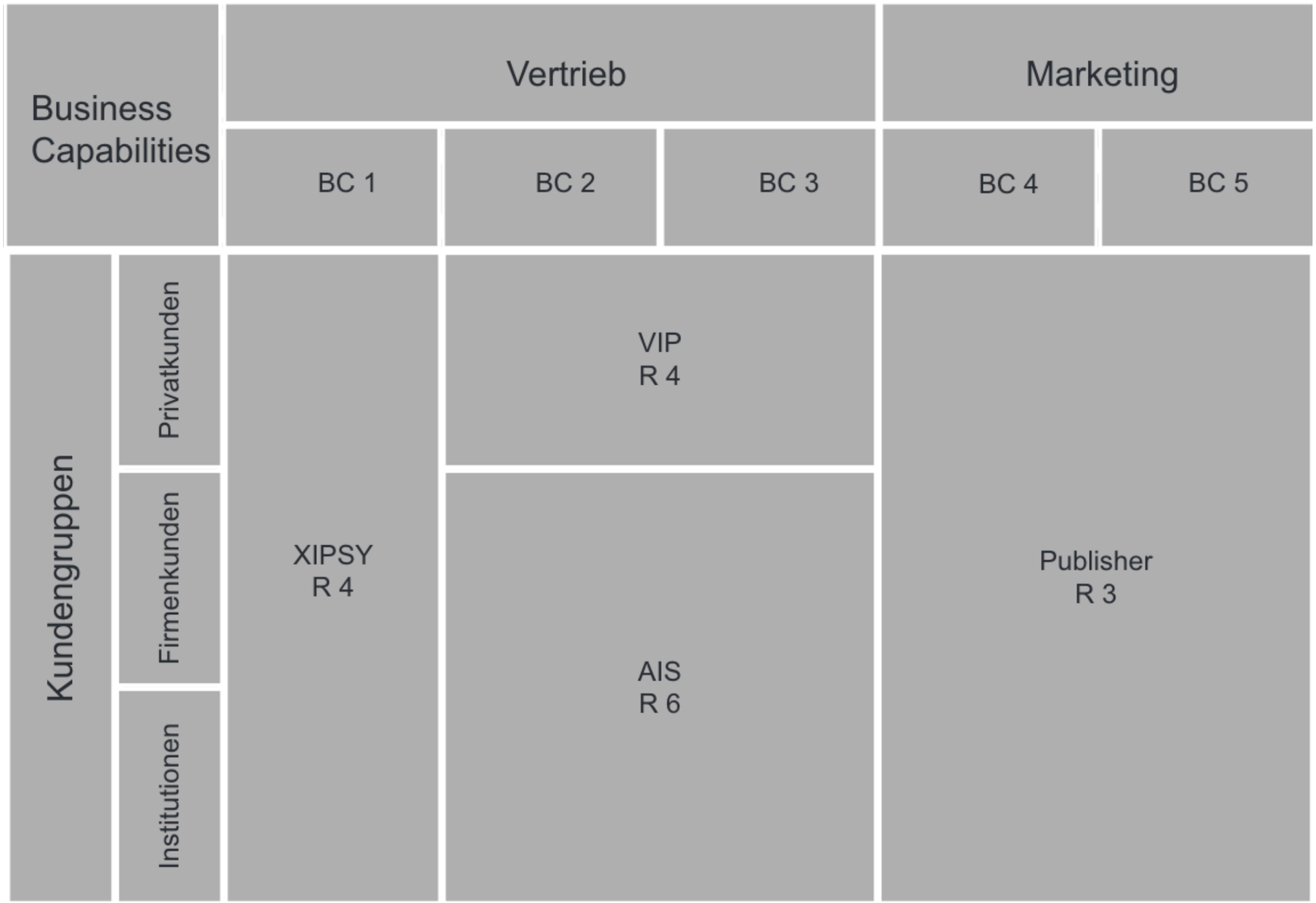
Quelle: Niemann 2005, S. 155

Beispiel eines Bebauungsplans "as is model"



Quelle: Hanschke, S. 451

Beispiel eines Bebauungsplans "as is model"



Quelle: Hanschke, S. 451

Weitere Planungsmethoden

Erstellen von Planungsszenarien

- Beleuchten des Planungsproblems aus verschiedenen Sichtweisen
- Erarbeitung einer risikoarmen und vollständigen Lösung

Alternativszenarien

- Zeitlich parallele Versionen
- Planzustände
- Unterschiedliche Ausprägung in Abhängigkeit von zukünftigen Entscheidungen oder Ereignissen

Historisierung

- Aufbewahrung von Alternativszenarien
- Dokumentation des IST-Zustands
  -> Entscheidung für ein Szenario
  -> SOLL-Zustand

Bildung von Ausschnitten

- Hervorheben bestimmter Teilbereiche
- Erklärungssichten für Stakeholder, wie Vorstand, Bereichsleiter, etc.

Schlußfolgerung, wenn diese nicht benötigt wird, bitte löschen.
Zweite Zeile, wenn nötig.

Quelle: Niemann 2005, S. 159-164

Architekturbewertung

Grundlegendes

- Alle Aktivitäten zur qualitativen oder quantitativen Bestimmung der Qualitätseigenschaften einer Architekturspezifikation
- Überprüfung der Qualität der Architekturspezifikation
- Ergebnis = Nachweis über Erfüllung aller Qualitätsanforderung bzw. Identifikation von nachzubessernden Schwachstellen
- Entwicklungsaufwand, Zeit und Kosten teilweise sehr hoch
- Bedarf an Methoden und Werkzeuge zur Abschätzung der Qualität des Systems gegen die Kunden-Anforderungen während des Evaluierungsprozesses

Die Architekturbewertung dient der Sicherstellung von Qualitätseigenschaften

Definition der Qualitätsmerkmale

Modifizierbarkeit

- Eignung einer Architektur, Anforderungsänderungen möglichst schnell und kostengünstig umzusetzen

Zuverlässigkeit

- Fähigkeit einer Software unter den gegebenen Bedingungen fehlerfrei zu arbeiten

Portabilität

- Anpassbarkeit einer Architektur an eine andere Umgebung (z.B. eine neue Plattform)

Architekturbewertung

Vor- und Nachteile

Vorteile

- Vorrangig ökonomische Nutzen
- Frühe Identifikation von Qualitätsproblemen
- Qualitätsnachweise

Nachteile

- Beschränkte Aussagefähigkeit
- Nur Vorhersage, da Spezifikation unvollständig
- Hoher Aufwand

Trotz des teilweise hohen Aufwands ist eine Architekturbewertung zur Frühwarnung notwendig.

Analyseverfahren für Anwendungslandschaften

Planungsphase

Komplexität von Anwendungslandschaften

Analyse der Komplexität

Wissenswertes

- Kaum Metriken
- Messung der Komplexität ganzer Anwendungslandschaften auf Grund von fehlenden Instrumenten nicht möglich
- Generell gilt: CAL = f (AAS, AIF)
- McCabe-Metrik: Berechnung der inneren Komplexität von Softwaresystemen
- Komplexität K eines Systems ergibt sich aus der Anzahl von Knoten und der Anzahl der Kanten
- Bislang keine Benchmarks für Anwendungslandschaften
- Ermittelte Kennzahl = Indikator für Fortschritt

Die Analyse der Komplexität ist momentan nicht operationalisierbar.

Kostenentwicklung bei Anwendungslandschaften

Kostentreiber

- Stichpunkt 1
- Stichpunkt 2
- Stichpunkt 3

	Kosten	Inanspruchnahme
Hardware	↓30x	↑
Netzwerkverbindungen	↓30x	↑
Lizenzen	→	→
Dienstleistungen	→	↑↑↑

Quelle: McComb2018, S. 31

Kostentreiber

- Komplexität
- Abhängigkeiten
- Integrationsaufwand
- Change Management (Menschen)
- Anpassungsaufwand

Quelle: McComb2018, S. 46

Messung von Komplexität

McCabe Cyclomatic Complexity

- Anzahl der möglichen Pfade durch einen Codeabschnitt

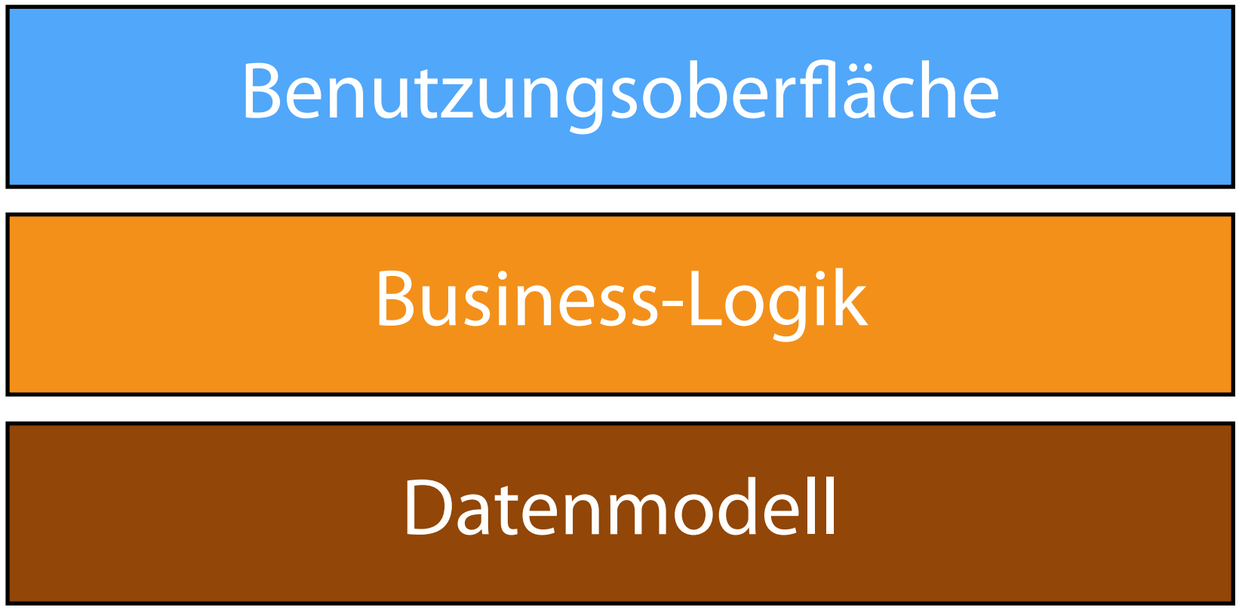
Halstead Metrics

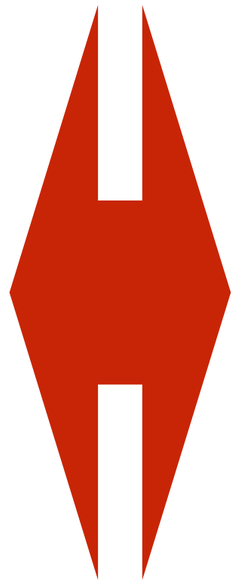
- Größe des Vokabulars
- Anzahl von Operatoren und Operanden

Hier wird die Berechnung der Komplexität von Datentabellen, Programmen usw. benötigt.

Quelle McComb2018, S. 46

„The Application-centric mindset…“





Änderungen betreffen stets alle drei Ebenen, weil Business-Logik Validierung, Sicherheit, Identity Management, Interpretation und Geschäftslogik übernimmt

…führt zu unsinnig hohen Kosten bei der (Weiter-)Entwicklung von Anwendungssystemen

Quelle: McComb2018, S. 59

Komplexität der Anwendungslandschaft

- Notwendige Komplexität **10%**
- Unbeabsichtigte Kompexität **90%**

Berechnung

Zahl der Anwendungen	*	Schemata	*	Code
		↓		↓
		Tabellen und Spalten		pro Schema, also um eine betroffene Tabelle oder Spalte zu bearbeiten

Beispiel

1	*	100 Tabellen, 7500 Attribute	*	10 Mio. Lines of Code = 1300 LOC/Schema

Wichtigste Aufgabe ist die Verringerung der unbeabsichtigten Komplexität!

Quelle: McComb 2018, S. 70

Ein Lösungsansatz: RESTful API

- Application Program Interface (API) das Daten per HTTP verändert
- Aufschlüsselung der Transaktionen mit Webdiensten in Module
- Verteilung der Informationen und Aufgaben auf verschiedene Server
- Software wird für unterschiedlichste Geräte verfügbar gemacht
- Benutzt Representational State Transfer (REST) Paradigma

REST Paradigma

- Uniform interface
- Client-server
- Stateless
- Cache
- Layered system
- Code on demand

HTTP Syntax	Bedeutung
POST	Erstellen
GET	Lesen
PUT	Ersetzen
PATCH	Modifizieren
DELETE	Löschen

Quelle: De, Brajesh 2017

Ansätze zum Management von Anwendungslandchaften

- Relationale Datenbanken
- ERP
- Unternehmensweite Datenmodellierung
- SOA und API's
- Agile Entwicklung
- Data Warehouses/Business Intelligence
- Outsourcing
- Cloud
- Software as a Service

Quelle: McComb 2018, S. 75

Probleme relationaler Datenbanken

- Mangelnde Portierbarkeit
- Fehlende Interoperabilität zwischen verschiedenen Datenbanken
- (Menschliche Einflüsse)
- Steigerung der Komplexität
- Schwierige Skalierbarkeit (Anzahl Tupel, Anzahl Tabellen)

Designkomplexität

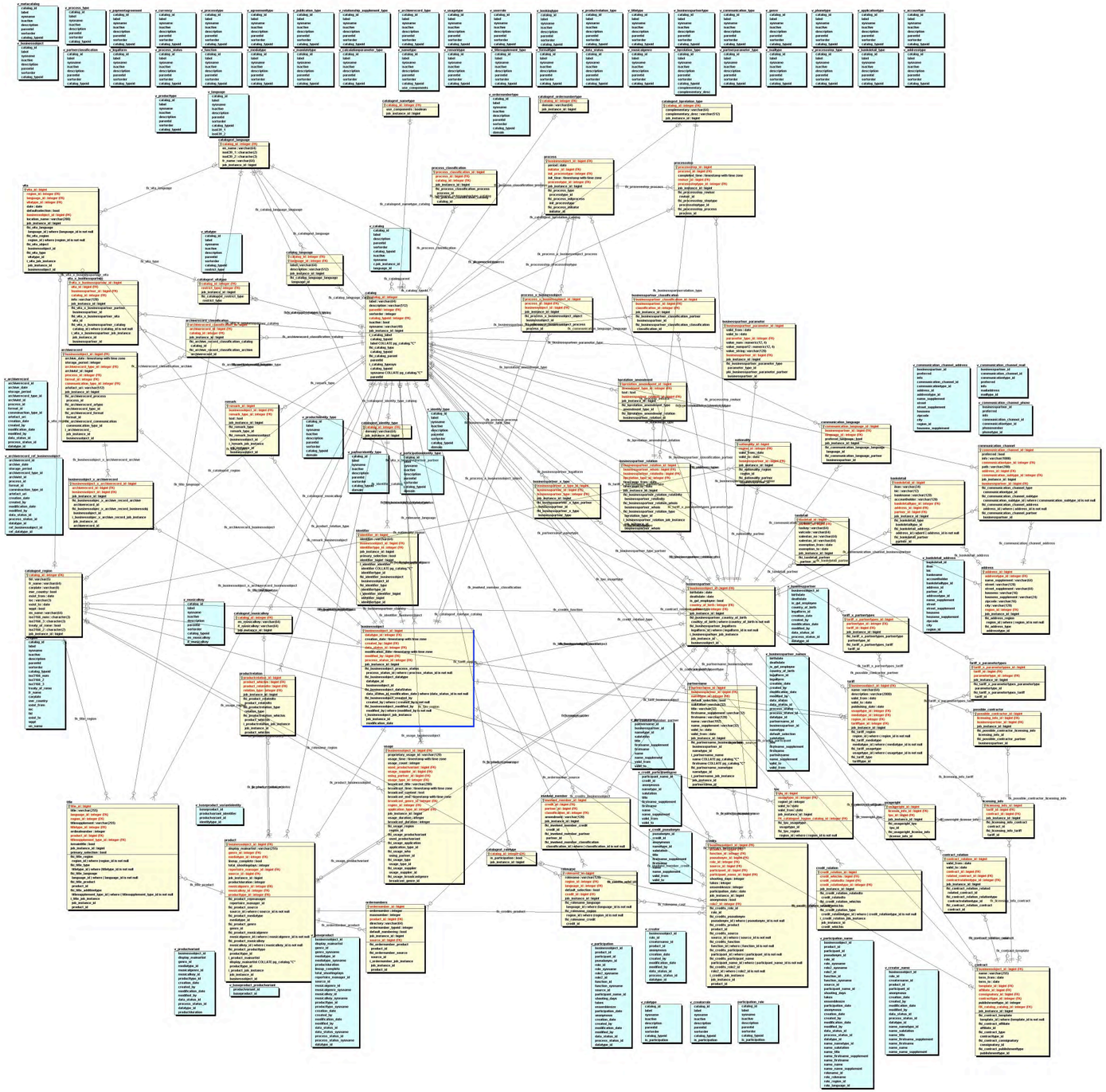
- Keine Vererbungsmechanismen (ähnliche Konzepte in zwei Tabellen)
- Deutliche höhere Zahl von Tabellen erforderlich als eigentlich notwendig

Komplexität des Anwendungscodes

- Impedance Mismatch - Unverträglichkeit zwischen Datenbank und Anwendungsprogrammierung

Quelle: McComb 2018, S. 75

Probleme relationaler Datenbanken



Quelle: Potsdam Consulting

Probleme relationaler Datenbanken
Beispiel

- **SELECT** title, name, (
  SELECT string_agg(trim(both FROM concat(name)),'_|_') FROM DB.credits composer inner **JOIN** names on name_id = partnername_id WHERE […] GROUP […])

- **SELECT** string_agg(eans.identifier,'_|_') as ean
  FROM DB.productrelation pr
  inner JOIN DB.identifier eans on (_id = relatedto or _id=product) WHERE product = id GROUP by product)

- FROM product
  inner **JOIN** DB.businessobject bo on bo.businessobject_id = product.businessobject_id and bo.data_status_id not in (:locked, :deleted, :readonly) and bo.process_status_id = :bpPlausibleId
- **WHERE** product.businessobject_id in (:productIds)
- and product.genre_id <> :genreNullId
  UNION
- SELECT […] FROM product
- inner JOIN […], inner JOIN […] WHERE and id in genre_id **UNION**
- SELECT [...]

Designkomplexität

- In stark normalisierten Datenmodellen sind für komplexe Abfragen zahlreiche Integrationen (JOINs) notwendig um zum gewünschten Ergebnis zu kommen.
- Anzahl Zeilen 180
- Anzahl Tabelle 33
- Anzahl Joins 34

- Bei kombinierten Datenquellen sind Vereinigungen notwendig (UNION)
- Notwendigkeit von Subabfragen um Informationen aus anderen Tabellen zu erhalten (z.B. Name zu ID)
- Die Komplexität kann mittels Zerlegung und Views reduziert werden
- Bei fehlender Optimierung und Indexierung resultieren derartige Abfragen in langer Anfragedauer

Gründe für Standardsoftware

- Einsetzbar ohne Modifikation und mit geringem Customizing
- Eingebaute Logik überschreitet sehr umfassend die eigenen Möglichkeiten zur Entwicklung
- Datenmodell des Unternehmens kann sehr leicht integriert werden
- Kaufpreis unter 1/10 des Entwicklungsaufwnads
- Zeitdruck

Nur wenn diese Merkmale zutreffen, ist lt. McComb Standardsoftware eine gute Wahl.

Quelle: McComb 2018, S. 135

Besondere Schwierigkeiten im Public Sector

- Schwierigkeiten, gute Mitarbeiterinnen und Mitarbeiter anzuwerben
- Übertrieben restriktive Beschaffungsabläufe/-vorschriften
- Anbieter, die sich auf das Ausspielen des Systems spezialisiert haben
- Bereitschaft, Mittelmäßigkeit zu akzeptieren
- Budgetaspekte
- Mangel an Anreizen

Quelle McComb2018, S. 174

Literatur

Ionita, M. T.; Hammer, D. K.; Obbink, H. (2004): Scenario-Based Software Architecture Evaluation Methods: An Overview. Department of Mathematics and Computing Science, Technical University Eindhoven; Department Software Architectures, Philips Research, Netherlands, 2004

Hanschke, I. (2023) .Strategisches Management der IT-Landschaft – Ein praktischer Leitfaden für das Enterprise Architecture Management. Hanser.

Kruchten, P.: Architectural Blueprints—The "4+1" View Model of Software Architecture, IEEE Software 12 (6) November 1995, pp. 42-50

Niemann, K. D. (2005): Von der Unternehmensarchitektur zur IT-Governance: Bausteine für ein wirksames IT-Management. Springer-Verlag Wiesbaden 2005.

Reussner, R.; Hasselbring, W. (Hrsg.) (2008): Handbuch der Software-Architektur. dpunkt 2008.

McComb 2018: McComb, D: Software Wasteland. How the application-Centric Mindset is hobbling our Enterprises, Basking Ridge NJ 2018

De, Brajesh 2017: Designing a RESTful API Interface