



Thema:

**Referenzmodell für die technische Freigabe  
von Software-Schnittstellen in Banken**

**Studienarbeit**

Arbeitsgruppe Wirtschaftsinformatik - Managementinformationssysteme

Themensteller: Prof. Dr. Hans-Knud Arndt

Betreuer: Dipl.-Kfm. Henner Graubitz

vorgelegt von: Chau Le Minh

Abgabetermin: 25. Juli 2008

## Inhaltsverzeichnis

Inhaltsverzeichnis .....	II
Abkürzungsverzeichnis .....	III
Abbildungsverzeichnis .....	IV
1 Einleitung.....	1
1.1 Ausgangssituation .....	1
1.2 Problemstellung.....	1
1.3 Zielsetzung und Aufbau der Studienarbeit.....	2
2 Einführung in die Modellierung .....	3
2.1 Begriffsbestimmung – Modellierung im Allgemeinen .....	3
2.2 Architektur integrierter Informationssysteme (ARIS) / ARIS Toolset .....	6
2.3 Prozessmodellierung mit eEPK.....	8
2.4 Referenzmodellsbegriff.....	10
3 Das Referenzmodell für die Sachsen LB .....	13
3.1 Analyse des Ist-Modells – Softwarefreigabe-Verfahren .....	13
3.2 Entwicklung des Referenzmodells als Soll-Modell .....	14
3.3 Erläuterung zu den Referenzmodellen .....	16
3.3.1 Referenzmodell der DV-Konzepts-Entwurfsphase.....	16
3.3.2 Referenzmodell der technischen Testphase .....	17
3.4 Nutzen der Referenzmodelle .....	18
4 Zusammenfassung und Ausblick .....	19
A Funktionsbaum zu Softwarefreigabe-Verfahren der Sachsen LB .....	20
B Funktionsbaum zur technischen Freigabe von Software-Schnittstellen.....	21
C Referenzmodell für die technische Freigabe von Software-Schnittstellen (DV-Konzepts-Entwurfsphase).....	22
D Referenzmodell für die technische Freigabe von Software-Schnittstellen (Technische Testphase) .....	23
Literaturverzeichnis .....	24

## Abkürzungsverzeichnis

ARIS	Architektur integrierter Informationssysteme
BO	Backoffice
DV	Datenverarbeitung
eEPK	erweiterte Ereignisgesteuerte Prozessketten
FO	Frontoffice
GoB	Grundsätze ordnungsmäßiger Buchführung
GoM	Grundsätze ordnungsmäßiger Modellierung
SLB	Sachsen LB

## Abbildungsverzeichnis

<b>Abb. 2.1:</b> Morphologischer Kasten der Informationsmodellierung.....	4
<b>Abb. 2.2:</b> ARIS Architektur.....	6
<b>Abb. 2.3:</b> Integration der vier Basissichten durch die Steuerungssicht .....	7
<b>Abb. 2.4:</b> Struktur von Modelldatenbanken nach ARIS (vereinfachte Darstellung).....	8
<b>Abb. 2.5:</b> Elemente der eEPK (Darstellung im ARIS Toolset) .....	9
<b>Abb. 2.6:</b> Mit Referenzmodellen verfolgte Zielsetzungen.....	11
<b>Abb. 2.7:</b> Nutzen von Referenzmodellen.....	12
<b>Abb. 3.1:</b> Phasenmodell zur Umsetzung betrieblichen Tatbestände in die Realisierung	15

# 1 Einleitung

## 1.1 Ausgangssituation

Die veränderten Rahmenbedingungen zwingen die Banken ihre Geschäftsmodelle neu anzupassen, d.h. verstärkt dienstleistungsorientiert zu agieren. Heutige Bank-IT-Systeme erfordern deshalb neben den Grundanforderungen wie Sicherheit, Vertraulichkeit, Integrität und Verfügbarkeit flexible und effiziente Systeme, die sich den schnell verändernden Anforderungen aus dem Umfeld anpassen können.<sup>1</sup>

Des Weiteren sind die vielen Systeme innerhalb eines Finanzdienstleistungs-Unternehmens sehr stark miteinander verknüpft, und bilden somit die Systemlandschaft des Unternehmens. Das eine System liefert einerseits Inputdaten für andere Systeme. Andererseits bekommt es auch Daten aus anderen Systemen für eigene Verarbeitung. Die Interoperabilität zwischen den Systemen, die unkomplizierte Erweiterbarkeit von Diensten und die leichte Managebarkeit der IT-Systeme gehören zu den zukünftig immer wichtiger werdenden Anforderungen.

Aus fachlichen Bedürfnissen des Bankgeschäfts werden oftmals neue *Schnittstellen* aus einem System zur Belieferung der Daten an weitere Zielsysteme erwünscht. Solche Schnittstellen sollten sich einfach und möglichst schnell entwickeln lassen. Sie müssen sich auch sehr stabil und sicher verhalten. Um dies zu gewährleisten, müssen alle Anforderungen an die Schnittstellen von vornherein in der Entwurfsphase schon bekannt gemacht werden.

## 1.2 Problemstellung

Im Organisations-Handbuch des Bereichs IT der *Sachsen LB*<sup>2</sup> (SLB) wurde ein sog. *Softwarefreigabe-Verfahren* dokumentiert. Dieses Verfahren beschreibt genau den Prozess, der die sichere und mängelfreie Überführung von Software in den Produktivbetrieb zum Ziel hat. Der Prozessablauf ist von Software abhängig und wird durch den für die Softwarefreigabe verantwortlichen Mitarbeiter aus der Abteilung IT-Organisation anhand einer „*Checkliste zur Softwarefreigabe*“ bestimmt. Das Verfahren findet jedoch nur bei Freigabe von Software im engeren Sinne Anwendung, d.h. Freigabe neuer Software-Systeme oder Freigabe von vorhandenen Softwares aufgrund des Releasewechsels. Trotz deren Häufigkeit verläuft die Freigabe von *Software-Schnittstellen* bisher wegen deren kleineren Umfangs nicht nach einem bestimmten Standardablauf. Die Qualität der Schnittstellen ist deshalb meistens fragwürdig.

---

<sup>1</sup> Vgl. Mücke (2008), S. 7.

<sup>2</sup> ehemalige „Landesbank Sachsen“ mit Sitz in Leipzig

### **1.3 Zielsetzung und Aufbau der Studienarbeit**

Es bedarf einer allgemeingültigen Standardisierung des Prozessablaufs in Form eines Modells von der Entwurfsphase bis zur technischen Testphase und abschließend mit der technischen Abnahme der Schnittstellen. Dieses Modell wird allerdings nicht als Ablösung derzeitigen Softwarefreigabe-Verfahrens in der Bank gesehen, da es sich nur an die Erweiterung vorhandener Softwares durch Schnittstellen richtet.

Um die Problemstellung einer Lösung zuzuführen, war vorgesehen an dem Entwicklungsprozess von Schnittstellen teilzunehmen vom Entwurf des DV-technischen Konzepts bis zur Durchführung des technischen Abnahmetests. Das Ziel war ein Referenzmodell für die technische Freigabe von Software-Schnittstellen in der Bank zu entwickeln.

Um dieses Ziel zu erreichen werden in dieser Arbeit zunächst die theoretischen bzw. technischen Grundlagen von Modellierung beschrieben. Im gleichen Abschnitt werden auch die Modellierungssprache eEPK und das Modellierungswerkzeug „ARIS Toolset“ vorgestellt. Im weiteren Abschnitt werden schrittweise die zu entwickelnden Referenzmodelle konzipiert.

Letzter Abschnitt enthält die Zusammenfassung der Arbeit und einen kleinen Ausblick.

## 2 Einführung in die Modellierung

### 2.1 Begriffsbestimmung – Modellierung im Allgemeinen

Bevor die theoretischen bzw. technischen Grundlagen der Modellierung behandelt werden, ist es zunächst erforderlich, die Bedeutung einiger oft verwendeten Begriffe an dieser Stelle klar zu machen.

#### Modellierung / Modell / Modelltyp

*Modellierung* oder *Modellbildung* ist dem Prozess der Abstraktion gleichgestellt. Unter *Abstraktion* versteht man die Eliminierung von Eigenschaften eines Sachverhalts, welche für eine konkrete Betrachtung dieses Sachverhalts nicht von besonderer Relevanz sind. Das Gegenteil von Abstraktion ist die Konkretisierung.

Das Ergebnis des Modellierens ist ein *Modell*. Es ist ein abstraktes, immaterielles Abbild realer Strukturen bzw. des realen Verhaltens für Zwecke des Modelladressaten. Ein Modell kann damit auch als adäquates, vereinfachendes und idealisierendes Abbild der Realität charakterisiert werden<sup>3</sup>.

Modelle können mit dem Ziel entwickelt werden, Sachverhalte durch Abstraktion so weit zu rekonstruieren, dass sie komplexe oder komplizierte Phänomene *erklären*. Eine andere Zielsetzung der Modellierung ist die Schaffung von *Gestaltungsmodellen*, die Optionen zur (Um-)Gestaltung der Realwelt aufzeigen und damit eine Handlungsverbindung zu ihr besitzen<sup>4</sup>.

Modelle lassen sich aus verschiedenen Sichten (s. *Morphologischer Kasten*) typisieren. Je nach Modellierungszweck wird das Modellierungsergebnis als Modell anders ausgesehen, z.B. bei Modellierung von Geschäftsprozessen ist ein Prozessmodell als Ergebnis zu erwarten; bei Modellierung von Datenstrukturen ein Datenmodell (Datenbankschema); bei Modellierung von betrieblichen Funktionen ein Funktionsbaum; bei Modellierung von Organisationsstrukturen ist ein Organigramm als Ergebnis zu erwarten.

#### Morphologischer Kasten der Informationsmodellierung

Ein morphologischer Kasten der Informationsmodellierung wird als geeigneter Ordnungsrahmen zur Systematisierung von Informationsmodellen vorgeschlagen. Dabei handelt es sich um eine Klassifikation der Merkmalausprägungen nach Merkmalen wie Beschreibungssicht, Beschreibungsebene, Geltungsanspruch, inhaltlicher Individualität

<sup>3</sup> Vgl. Rautenstrauch, Schulze (2003), S. 225.

<sup>4</sup> Vgl. Rautenstrauch, Schulze (2003), S. 226.

und Abstraktionsgrad. Diese Klassifikation trägt zur Auswahl passenden Modells bei Modellierung eines Sachverhalts für mehrere Adressaten bei.

Folgende Abb. 2.1 zeigt die Strukturdimensionen und Ausprägungen der vorgeschlagenen Merkmalklassifikation.

Merkmals-	Ausprägung			
Beschreibungssicht	Daten	Funktionen	Organisation	Prozesse
	Objekte			
Beschreibungsebene	Fachkonzept	DV-Konzept	Implementierungskonzept	
Geltungsanspruch	Istmodell	Sollmodell	Idealmodell	
Inhaltliche Individualität	Unternehmensmodell	Referenzmodell	Mastermodell	
Abstraktionsgrad	Ausprägungsebene	Typebene	Metaebene	Meta-Meta-Ebene

Quelle: Rosemann (1996), S. 22.

**Abb. 2.1:** Morphologischer Kasten der Informationsmodellierung

Die Beschreibungssicht und Beschreibungsebene wurden auch in einem Ordnungsrahmen zusammengefasst, der unter der Bezeichnung *ARIS-Architektur* bekannt geworden ist. Diese Architektur wird im weiteren Unterabschnitt vorgestellt.

### Grundsätze ordnungsmäßiger Modellierung (GoM)

Die *Grundsätze ordnungsmäßiger Modellierung* wurden mit dem Ziel konzipiert die Qualität von Modellen eines Fachkonzepts durch Angabe von Gestaltungsempfehlungen zur bedarfsgerechten Modellerstellung zu erhöhen, die Vielfalt möglicher Modellierungsvarianten und die Subjektivität der Modellierung zu reduzieren und Vergleichbarkeit sowie Integrationsfähigkeit von Modellen zu erhöhen<sup>5</sup>. Die GoM, die in Anlehnung an die *Grundsätze ordnungsmäßiger Buchführung* (GoB) als Leitfaden zur Erstellung eines Modells entwickelt werden, stellen sich somit als Qualitätsmaßstäbe zur Bewertung der Modelle dar.

<sup>5</sup> Vgl. Becker, Rosemann, Schütte (1995)

Im Folgenden werden die sechs Grundsätze ordnungsmäßiger Modellierung kurz charakterisiert<sup>6</sup>.

- Grundsatz der Richtigkeit: Der Grundsatz umfasst zwei Aspekte die *syntaktische* (Konsistenz und Vollständigkeit des Modells zum herangezogenen Metamodell) und die *semantische* (struktur- und verhaltenstreue Beschreibung der Realwelt) Richtigkeit.
- Grundsatz der Relevanz: Der Grundsatz erfordert zum einen die Wiederfindung aller relevanten Aspekte der Realwelt im Modell, d.h. keinen Informationsverlust; zum anderen die Minimalität der enthaltenen Informationen, d.h. Weglassen von Sachverhalten, die unnötig für die Modellierung aus Sicht der Modelladressaten sind. Über diese inhaltlichen Aspekte hinaus, richtet sich der Grundsatz der Relevanz auch an die ausgewählte Modellierungsmethode sowie das verwendete Modellierungswerkzeug.
- Grundsatz der Wirtschaftlichkeit: Dieser Grundsatz bringt andere fünf Grundsätze in einen betriebswirtschaftlichen Zusammenhang. Erstelltes Modell soll eine gute Persistenz (Dauerhaftigkeit) besitzen, d.h. es ist über die Verwendungsdauer ohne Änderungen verwendbar. Sind dennoch Veränderungen erforderlich, dann muss das Modell so flexibel sein, dass die Änderungen ohne großen Aufwand durchführbar sind.
- Grundsatz der Klarheit: Der Grundsatz besagt, dass das Modell sowohl für Modellierer wie Modelladressaten anschaulich sein muss. Klare Modelle müssen strukturiert, intuitiv zugänglich, übersichtlich und lesbar sein.
- Grundsatz der Vergleichbarkeit: Das Wesentliche dieses Grundsatzes ist die identische Anwendung der GoM bei der Erstellung mehrerer Modelle. Vergleichbarkeit ist besonders bei arbeitsteiliger Modellierung relevant, da die Modelle später zu einem Gesamtmodell zusammengeführt werden sollen.
- Grundsatz des systematischen Aufbaus: Der Grundsatz berücksichtigt die Notwendigkeit der Integration der einzelnen Sichten bei Modellierung in getrennten Sichten. Ein sichtenübergreifendes Metamodell ist laut dieser Forderung erforderlich, um die einzelnen Modelle sowohl voneinander abzugrenzen als auch durch ihre gemeinsamen Informationsobjekte in mehreren Sichten zusammenzuführen.

---

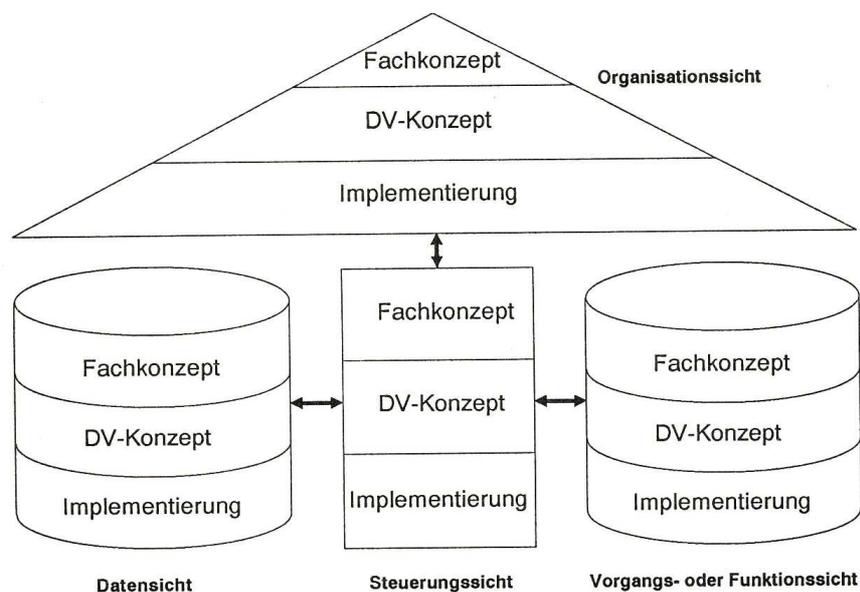
<sup>6</sup> Vgl. Rosemann (1998), S. 6ff.

Die GoM lassen sich in notwendige und ergänzende Grundsätze aufteilen. Zu der notwendigen Gruppe gehören die ersten drei Grundsätze. Sie sind unbedingt einzuhalten, da die Modelle andernfalls wertlos sind. Die letzten drei Grundsätze dienen dagegen der Qualitätsverbesserung der Modelle.

## 2.2 Architektur integrierter Informationssysteme (ARIS) / ARIS Toolset

ARIS ist eine Methodik, die eine ganzheitliche Betrachtung von Geschäftsprozessen anstrebt. Sie unterscheidet fünf Sichten (*Beschreibungssichten*) auf ein Informationssystem: die Organisationssicht, die Funktionssicht, die Datensicht, die Leistungssicht und die Steuerungssicht. Die Steuerungssicht nimmt dabei insofern eine Sonderstellung ein, als sie der Integration der anderen Sichten dient. Neben den verschiedenen Sichten unterscheidet ARIS zwischen den drei Abstraktionsstufen Fachkonzept, DV-Konzept und Implementierung, die als *Beschreibungsebenen* bezeichnet werden<sup>7</sup>.

Abb. 2.2 stellt folgend die Architektur des Informationssystems dar.



Quelle: Scheer (1992), S. 18.

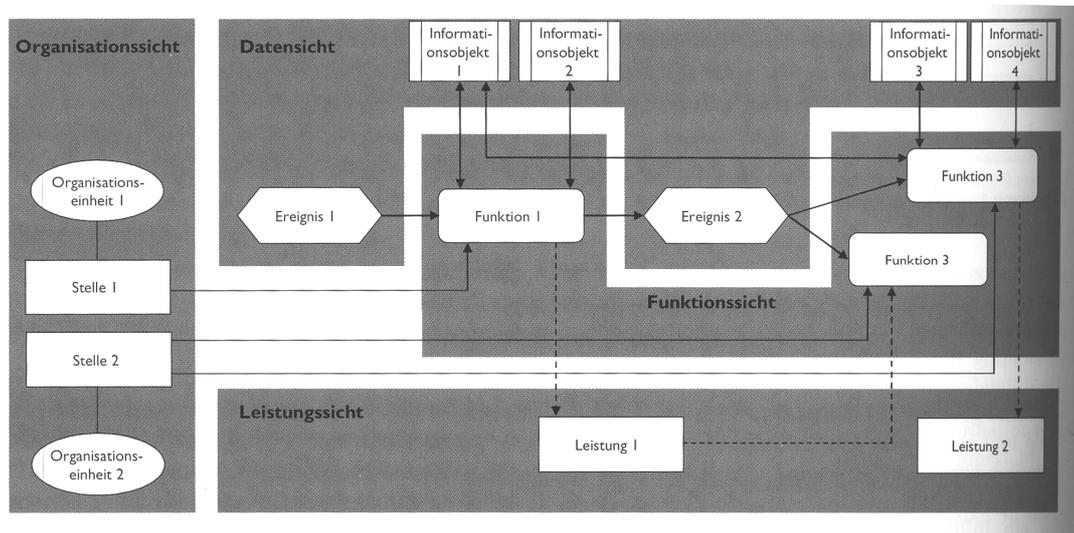
**Abb. 2.2:** ARIS Architektur

Aus dieser Architektur wird deutlich, dass alle Sichten in der Steuerungssicht zusammenlaufen, in der die relevanten Geschäftsprozesse modelliert werden<sup>8</sup>. Der

<sup>7</sup> Vgl. Laudon, Laudon, Schoder (2006), S. 582.

<sup>8</sup> Vgl. Rautenstrauch, Schulze (2002), S. 228.

Begriff *Geschäftsprozess* soll hier jedoch nicht nur den Kontrollfluss, d.h. die zeitlich-logische Reihenfolge der Funktionsausführung umfassen, sondern auch die damit unmittelbar verbundenen Daten-, Organisations- und Ressourcenbeschreibungen (s. Abb. 2.3).



Quelle: Laudon, Laudon, Schoder (2006), S. 584.

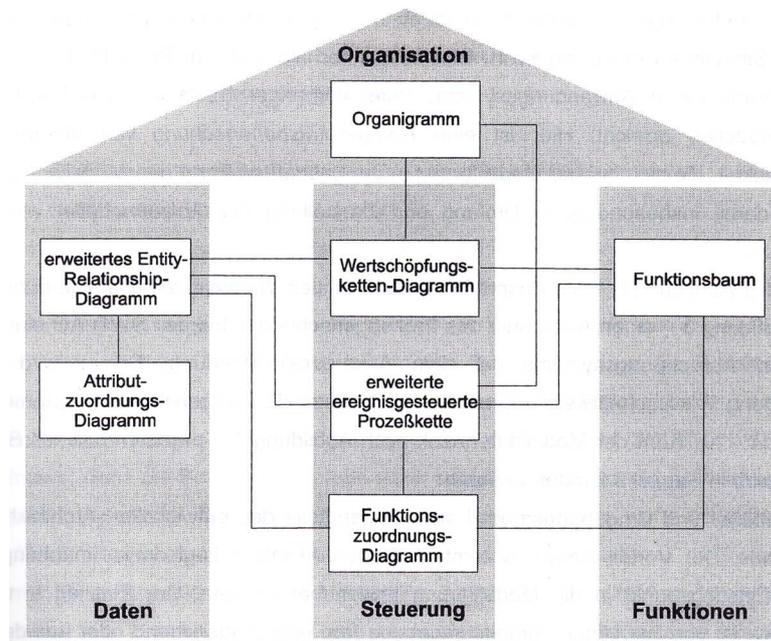
**Abb. 2.3:** Integration der vier Basissichten durch die Steuerungssicht

## Das ARIS Toolset

Als erstes marktfähiges Produkt zur Modellierung und Analyse von Geschäftsprozess ist das im Jahr 1992 von der IDS Scheer GmbH entwickelte ARIS Toolset derzeit Standardsoftware in diesem Marktsegment<sup>9</sup>. Neben der Basisfunktionalität „Modellierung“ mit einem graphischen Modelleditor, eigenen Datenbanken zur Verwaltung von Modellen oder Benutzern bietet das weltweit erfolgreichste für Geschäftsprozessmodellierung entwickelte Werkzeug noch weitere Funktionen zur Optimierung der erstellten Modelle wie Prozesskostenrechnung, Modellauswertung, Reporting, etc. Dieses Tool unterstützt darüber hinaus auch andere Modellierungs-Standards wie BPML, BPMN oder BPEL.

Folgende Abb. 2.4 zeigt einige in der ARIS Architektur vorgesehene bzw. im ARIS Toolset teilweise unterstützte Modelltypen in jeweiliger Sicht.

<sup>9</sup> Vgl. Jost, Wagner (2002), S. 16.



Quelle: Esser, Fidelak, Prescher (1998), S. 47.

Abb. 2.4: Struktur von Modelldatenbanken nach ARIS (vereinfachte Darstellung)

### 2.3 Prozessmodellierung mit eEPK

Die Methode der *Ereignisgesteuerten Prozessketten* (EPK) wurde im Rahmen der *Architektur integrierter Informationssysteme* zur Modellierung von Geschäftsprozessen entwickelt und hat sich schnell sowohl in der Forschung als auch der Praxis als eine Standard-Methode etabliert. EPK basieren auf die Grundidee von Petri-Netzen. Sie sind aber nicht so formal und haben daher in der Praxis vor allem auch im Managementbereich eine große Bedeutung. So wird dieses Modell beispielsweise in weit verbreiteten Produkten wie SAP R/3 unterstützt.

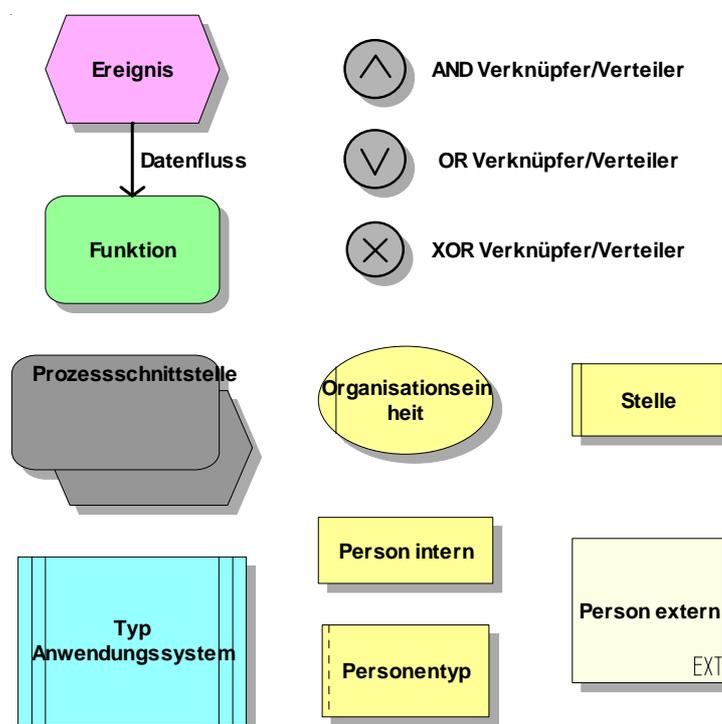
Wesentliches Kennzeichen der EPK ist die Abbildung der zu einem Prozess gehörenden Funktionen in deren zeitlich-logischer Abfolge. EPK sind gerichtete Graphen mit folgenden Grundelementen<sup>10</sup>:

- *Ereignisse* sind ablaufrelevante Zustandsausprägungen, die weder Zeit noch Kosten verbrauchen. Ein EPK muss immer mit einem Ereignis-Knoten starten und enden. Am Ende können mehrere Ereignisse stehen. Ereignisse lösen nachfolgende Funktionen aus und werden durch die Beendigung vorhergehender Funktionen getriggert. Ereignisse werden durch ein Sechseck notiert.

<sup>10</sup> Vgl. Bernroider, Stix (2004), S. 22.

- *Funktionen* transformieren einen Eingangszustand in einen Zielzustand und besitzen Entscheidungskompetenz für den weiteren Prozessverlauf. Nach Beendigung der Funktion wird das darauf folgende Ereignis ausgelöst. Funktionen werden durch ein Runderck notiert.
- *Logische Konnektoren*, an den Teilprozesse zusammenlaufen (Verknüpfen) oder sich aufspalten (Verteiler). Als logische Operationen sind die Konjunktion (UND – alle einlaufenden Kanten müssen aktiv sein bzw. alle ausgehenden Kanten werden aktiv), Disjunktion (Exklusives ODER – genau eine einlaufende Kante muss aktiv sein bzw. genau eine der auslaufenden Kanten kann aktiv werden) und die Adjunktion (ODER – mindestens eine einlaufende Kante muss aktiv sein bzw. mindestens eine der auslaufenden Kanten wird aktiv) zulässig.

Das Grundmodell der EPK kann um weitere semantische Beschreibungselemente (Ressourcen) ergänzt werden. Ein solcher Diagrammtyp wird auch als *erweiterte Ereignisgesteuerte Prozessketten* (eEPK) bezeichnet. Beispiele für Erweiterungen sind die Abbildung von Datenflüssen, Organisationseinheiten oder Anwendungssystemen (s. Abb. 2.4)



**Abb. 2.5:** Elemente der eEPK (Darstellung im ARIS Toolset)

Für die Konstruktion von EPK gelten folgende Regeln<sup>11</sup>:

- Funktionen dürfen nicht direkt mit Funktionen verknüpft werden.
- Ereignisse dürfen nicht direkt mit Ereignissen verknüpft werden.
- Konnektoren dürfen mit Konnektoren verknüpft werden.
- Konnektoren, Ereignisse und Funktionen werden über Kontrollfluss miteinander verbunden.
- Jede Prozesskette muss mit mindestens einem Ereignis beginnen und enden.
- Einem Ereignis darf weder eine disjunktive noch adjunktive Verknüpfung folgen, da Ereignisse keine Entscheidungskompetenz haben (die liegt bei den Funktionen!) und Modelle ohne exogene Informationen erklärbar sein müssen.
- Informationssysteme werden mit Funktionen über Linien verbunden.
- Organisationseinheiten werden mit Funktionen über Linien verbunden.

#### 2.4 Referenzmodellbegriff

Referenzmodelle haben in Theorie und Praxis eine weite Verbreitung gefunden. Ein *Referenzmodell* ergibt sich durch Abstraktion mehrerer unternehmensspezifischer Informationsmodelle (mit dementsprechender „*Allgemeingültigkeit*“) sowie durch den Einbezug theoretischer Erkenntnisse.

Referenzmodelle können ein konkretes Standardanwendungssystem repräsentieren oder auf betriebliche Strukturen und Abläufe, die unabhängig von konkreten Informationssystemlösungen betrachtet werden, fokussieren.

Ein anderer Begriff, der leicht in Verwechslung mit *Referenzmodell* kommt, ist *Vorgehensmodell*. Ein Vorgehensmodell ist ein Ablaufmodell zur Entwicklung von Software, das den Entwicklungsprozess in überschaubare, zeitlich und inhaltlich begrenzte Phasen einteilt. Ziel ist es, die Softwareentwicklung übersichtlicher zu gestalten und in der Komplexität beherrschbar zu machen<sup>12</sup>.

Der Unterschied zwischen einem Vorgehens- und einem Referenzmodell liegt in der Zielsetzung. Vorgehensmodelle sind für Entwicklungs- oder Implementierungsprozesse konzipiert und versuchen, ein optimales Vorgehen für ein bestimmtes Ziel festzulegen.

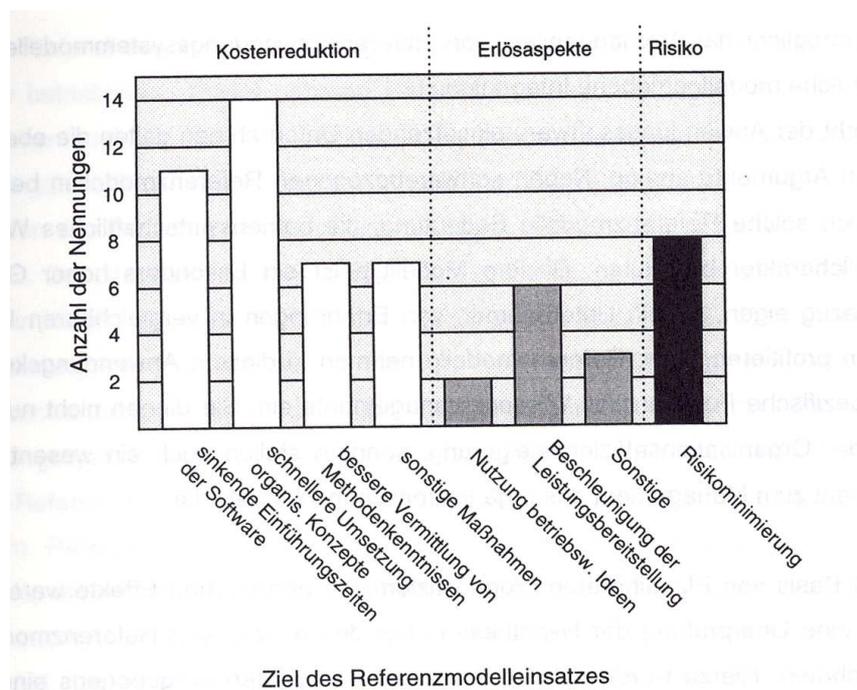
---

<sup>11</sup> Vgl. Rautenstrauch, Schulze (2002), S. 246.

<sup>12</sup> Nach *Wikipedia*

Dieses Ziel lautet in IT-Projekten die Faktoren Zeit und Kosten möglichst gering zu halten sowie ein qualitativ hochwertiges Produkt zu erstellen. Ein Referenzmodell beschreibt dagegen eher, wie ein System oder ein Prozess optimal aufgebaut werden sollte, damit z.B. möglichst einfache Erweiterungen möglich sind oder möglichst geringe Betriebskosten anfallen. Die beiden Modelle sind bei erster Betrachtung voneinander unabhängig, in vielen Situationen ergänzen sie sich oder bilden sogar eine Einheit.

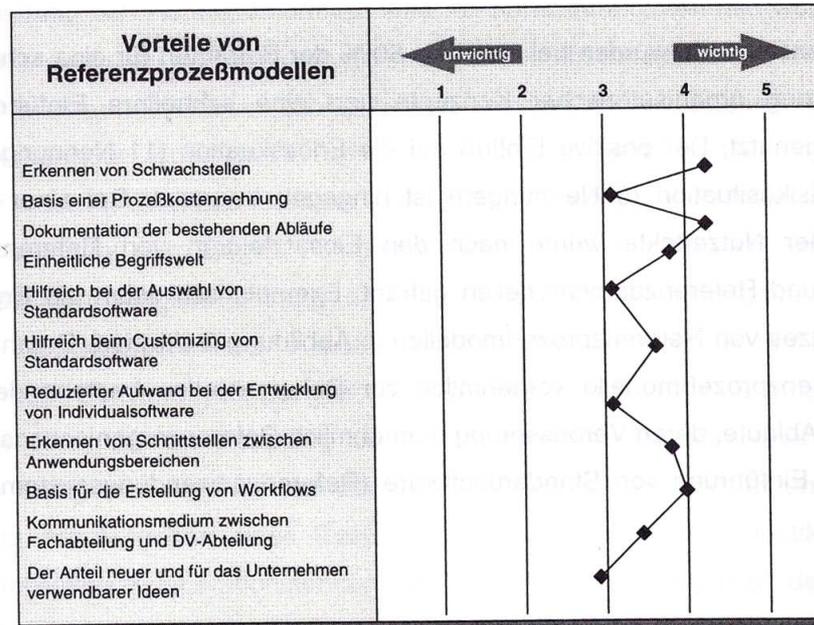
Mehr zu den Einsatzmöglichkeiten bzw. den Zielsetzungen der Referenzmodelle in der Praxis kann man aus der Abb. 2.6 entnehmen.



Quelle: Schütte (1998), S. 66.

**Abb. 2.6:** Mit Referenzmodellen verfolgte Zielsetzungen

Nutzen der Referenzmodelle wird in Abb. 2.7 veranschaulicht.



Quelle: Schütte (1998), S. 66.

**Abb. 2.7:** Nutzen von Referenzmodellen

### 3 Das Referenzmodell für die Sachsen LB

#### 3.1 Analyse des Ist-Modells – Softwarefreigabe-Verfahren

In diesem Abschnitt wird das Referenzmodell, das die theoretischen Grundlagen vom vorigen Abschnitt umsetzt, beschrieben. Anhand einer Ist-Analyse wird zuerst die Problematik in der SLB, welche einer Lösung bedarf, geschildert.

IT-Systeme und die zugehörigen IT-Prozesse laut Anforderung der SLB im *Organisations-Handbuch* des Bereichs IT müssen

- Integrität (Sicherstellung der Datenvollständigkeit und -Richtigkeit sowie Verhinderung von Datenmanipulationen),
- Verfügbarkeit (Schutz vor unbefugter bzw. nicht vorhersehbarer Vorenthaltung von Informationen oder Daten, maximale tolerierbare Ausfalldauer),
- Authentizität (Echtheit bzw. Rechtsgültigkeit der Daten und Ergebnisse) und
- Vertraulichkeit für die Daten und Anwendungen (Schutz vor unbefugter Preisgabe von Informationen)

sicherstellen.

Aus diesem Grund ist bei der Ausgestaltung der IT-Systeme und der zugehörigen IT-Prozesse in der SLB auf gängige Standards abzustellen. Dies beginnt bei der Auswahl und Einführung eines Systems, setzt sich regelmäßig insbesondere bei Updates fort und endet bei Außerbetriebnahmen bzw. Einstellung von Prozessen.

Das im Handbuch umfangreich beschriebene *Softwarefreigabe-Verfahren*, welches die sichere und mängelfreie Überführung von Software in den Produktivbetrieb gewährleistet, steht mit diesen Anforderungen stark in Einklang. Die Regelung gilt grundsätzlich für jede Software, die im Bereich der SLB zum Einsatz kommen soll.

Die im Softwarefreigabe-Verfahren durchzuführenden Verfahrensschritte, inklusive der in die Abstimmung einzubeziehenden Stellen und der zu erstellenden Unterlagen, werden durch den Freigabeverantwortlichen geprüft und festgelegt.

Der Gesamtumfang dieses Verfahrens ist in Anlage A als Funktionsbaum dargestellt. Der Freigabeverantwortliche kann jedoch den Umfang des Verfahrens je nach Situation angemessen kürzen oder ergänzen.

Die IT-Systemlandschaft einer Bank besteht aus einer Menge der stark miteinander im Zusammenhang stehenden Systeme. Der Datenaustausch zwischen diesen Software-Systemen erfolgt über Schnittstellen, die als normale Datenbankobjekte nach einem bestimmten Schema definiert werden.

Es ist meistens keine einzelne Anforderung an die Entwicklung von einer bestimmten Schnittstelle für eine Software zu erwarten, sondern eher sämtliche Anforderungen an mehrere Schnittstellen aus einem oder mehreren Systemen. Solche Anforderungen werden immer mit der Einführung eines neuen Projekts in der Bank gestellt. Im Durchlauf des Projekts sollen alle Schnittstellen implementiert werden.

Eine Software verfügt auch über mehrere Schnittstellen. Falls eine neue Schnittstelle für diese Software angefordert wird, darf sie aber keinen Einfluss auf die Lauffähigkeit anderer vorhandenen Schnittstellen der Software nehmen.

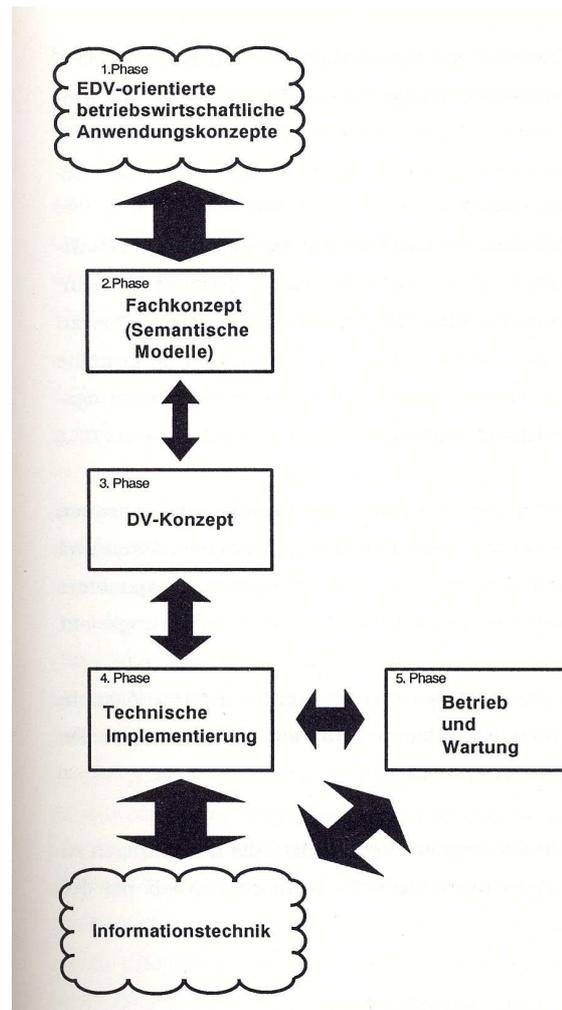
Es ist festzuhalten, dass die Dauer der gesamten Entwicklungszeit von Schnittstellen sehr großen Einfluss auf die Gesamtdurchlaufzeit des Projekts hat. Durch Verringerung der Dauer von der Entwicklungszeit einzelner Schnittstelle und damit der Dauer der gesamten Entwicklungszeit von Schnittstellen wird für das gesamte Projekt einen großen Vorteil von Zeit bzw. Kostenersparnis gebracht.

Die Tatsache, dass der Entwicklungsprozess von einzelnen Schnittstellen viel Zeit kostet, liegt in den bisherigen Ablaufprozess. Es existiert in der Bank noch keinen eigenen Standardablauf für die Freigabe von Software-Schnittstellen. Das vorgegebene Softwarefreigabe-Verfahren ist dagegen zu umfangreich und zeitaufwendig für solche kleinen Schnittstellen. Deshalb wurde es nicht für diesen Fall in Einsatz gebracht. Hier stellt sich auch die Frage nach der Qualitätssicherung der entwickelten Schnittstellen.

Es bedarf daher einer allgemeingültigen Standardisierung dieses Prozessablaufs, welche aber auch auf das Softwarefreigabe-Verfahren der Bank beruht. Damit werden allgemeine Anforderungen an Software-Systeme nicht verletzt.

### **3.2 Entwicklung des Referenzmodells als Soll-Modell**

Zunächst werden die nötigen Schritte zur technischen Freigabe von Schnittstellen grob in einem Funktionsbaum dargestellt (s. Anhang B). Diese Schritte ähneln einigen Phasen des fünfstufigen Phasenmodells zur Implementierung und zum Betrieb des betriebswirtschaftlichen Anwendungskonzepts.



Quelle: Scheer (1992), S. 17.

**Abb. 3.1:** Phasenmodell zur Umsetzung betrieblichen Tatbestände in die Realisierung

Es sind genau sechs Schritte zu verfolgen, bis die Schnittstelle freigegeben werden kann. Der erste Schritt ist ein Fachkonzept zu der Schnittstelle zu erstellen. Wie bereits im vorigen Unterabschnitt bekannt stellt der Fachbereich bei der Einführung in die Projektarbeit ein großes, umfangreiches Fachkonzept zur Verfügung. Dieses Konzept beinhaltet alle nötigen Informationen über das gesamte Projekt bezüglich der Fachlichkeit. Darin sind auch die Anforderungen an die Realisierung mehrerer Schnittstellen zu entnehmen. Die Schnittstellen sind unabhängig voneinander zu implementieren. Um konzentriert mit einer bestimmten Schnittstelle zu arbeiten, ist es deshalb zuerst wichtig, das passende Fachkonzept zu dieser Schnittstelle aus dem großen Dokument in Absprache mit dem Fachbereich zu extrahieren.

Der zweite, auch wichtigste Schritt ist das DV-technische Konzept zu entwerfen. Ein DV-Konzept setzt alle fachlichen Anforderungen aus dem Fachkonzept heraus in DV-technische Anforderungen um, welche die konkrete Implementierung ermöglichen.

Wegen dessen Wichtigkeit in dem ganzen Ablauf verbraucht dieser Schritt auch meiste Arbeitszeit. Deshalb spielt der Einsatz eines Referenzmodells zu diesem Schritt auch sehr wichtige Rolle.

Der nächste Schritt erwartet ein Testkonzept als Ergebnis. Das Testkonzept wird in Absprache mit dem Fachbereich erstellt. Es beschreibt das grundsätzliche Vorgehen und die erwarteten Ergebnisse.

Die Implementierung der Schnittstelle übernimmt das Entwicklerteam. Je nach Anforderungen bzw. Umfang der Schnittstelle wird sie intern oder extern entwickelt.

Der vorletzte Schritt ist der technische Abnahmetest. Hier wird die Schnittstelle grünlich durch den Mitarbeiter der Abteilung IT-Organisation technisch getestet. Bei erfolgreichen Tests kann die Schnittstelle technisch freigegeben werden. Bei fehlerhafter Schnittstelle muss das Entwicklerteam anpassen und Änderungen vornehmen, bevor der Nachtest beginnt. Alle Testergebnisse werden im letzten Schritt protokolliert und dokumentiert.

Nachfolgend werden die Schritte 2 „DV-Konzept erstellen“ und 5 „Technische Tests“ konkret erläutert.

### **3.3 Erläuterung zu den Referenzmodellen**

#### **3.3.1 Referenzmodell der DV-Konzepts-Entwurfsphase**

Dieses Modell beschreibt den allgemeingültigen Prozessablauf zur Erstellung eines DV-technischen Konzepts von Software-Schnittstellen (s. Anhang C).

Es beginnt mit dem Eingang eines Fachkonzepts, das vom Fachbereich erstellt und der Abteilung IT-Organisation als Anforderung an eine neue Schnittstelle übergeben wurde.

Anhand dieses Dokuments sucht der zur Abteilung IT-Organisation zugehörige Mitarbeiter, der für diese Aufgabe verantwortlich ist, einerseits ein Gespräch bei den fachlichen Betreuern aus dem Fachbereich, die eventuell das Fachkonzept entwickelt haben und später die Schnittstelle abnehmen werden. Durch dieses Gespräch sollen im Vorfeld Schwierigkeiten wie fachliche Begriffe, Definitionen beseitigt und über technische bzw. organisatorische Anforderungen geeinigt werden. Der Mitarbeiter informiert sich dennoch konkret über benötigte Informationen über die Schnittstelle (Datenbanken? Datenbankschemata? Tabellen? Datenfelder?).

Andererseits muss dieser Mitarbeiter das Entwicklerteam (intern oder extern) kontaktieren, das die Umsetzung und Implementierung der Schnittstelle übernimmt. Mit

dessen Hilfe stellt er eine Liste der bereit vorhandenen Objekte aus den Systemen, die eventuell für die Schnittstelle relevant sind.

Eine Gegenüberstellung der benötigten bzw. vorhandenen Objekte erfolgt um einen Überblick zu erschaffen. Der Mitarbeiter entwirft ein Datenbankschema für die Schnittstelle und eine Mappingstabelle von benötigten zu vorhandenen Informationen mit entsprechendem Mappingsalgorithmus.

Abschließend stellt er alle wichtigen Informationen in einem Gesamtdokument zusammen und übergibt dem Entwicklerteam für den Realisierungsprozess.

### **3.3.2 Referenzmodell der technischen Testphase**

Dieses Modell beschreibt den allgemeingültigen Prozessablauf zum technischen Abnahmetest von Software-Schnittstellen (s. Anhang D).

Nachdem die Implementierung der Schnittstelle abgeschlossen wird, wird die „neue“ Software mit dieser integrierten Schnittstelle auf der Testumgebung eingesetzt. Der technische Test kann beginnen.

Zunächst wird Zugriffsrecht auf Backoffice (BO)-System beantragt. Dann erfolgt die Prüfung auf Vollständigkeit und Richtigkeit der umgesetzten Schnittstelle, indem das Datenbankschema der Schnittstelle mit dem Schema nach DV-Konzept verglichen wird. Bei Gleichheit setzt der zweite Teil des technischen Tests fort. Sonst wird der Fehler protokolliert und dem Entwicklerteam weitergereicht.

Anhand vom Testkonzept, welches im Abschluss vorigen Prozesses erstellt wurde, werden nun konkrete Testfälle konstruiert. Wichtig an dieser Stelle ist, dass die Testkonstellationen alle Fehlermöglichkeiten abdecken müssen.

Die benötigten Daten werden exportiert und als VORHER-Daten gesichert. Man gibt die konstruierten Testfallbeispiele im Frontoffice (FO)-System ein, lässt den Batchlauf starten, um die Änderungen zu aktivieren.

Danach werden die NACHHER-Daten exportiert. Ein manueller bzw. automatisierter Abgleich der VORHER- gegenüber der NACHHER-Daten findet statt. Es wird nun überprüft, ob die Änderungen wie erwartet entstanden sind. Falls ja, ist es zu protokollieren, dass die Schnittstelle technisch erfolgreich getestet wurde. Andernfalls wird der Fehler protokolliert und dem Entwicklerteam weitergereicht.

### **3.4 Nutzen der Referenzmodelle**

Durch künftigen Einsatz der hier erstellten Referenzmodelle wird erwartet, dass sich der Erstellungsaufwand von DV-technischen Konzepten für die Software-Schnittstellen in der Bank deutlich reduziert bzw. der geplante Zeitrahmen für den technischen Abnahmetest eingehalten wird. Demzufolge kürzt sich auch die Gesamtlaufzeit vom Entwurf bis zur technischen Freigabe der Schnittstellen.

Des Weiteren wird die Qualität des gesamten Ablaufs der Schnittstellen-Freigabe durch Anwendung von Referenzmodellen auch sichergestellt.

## **4 Zusammenfassung und Ausblick**

Modellierung als Methodik bietet die Möglichkeit, komplexe oder komplizierte Sachverhalte einfach und verständlich darzustellen.

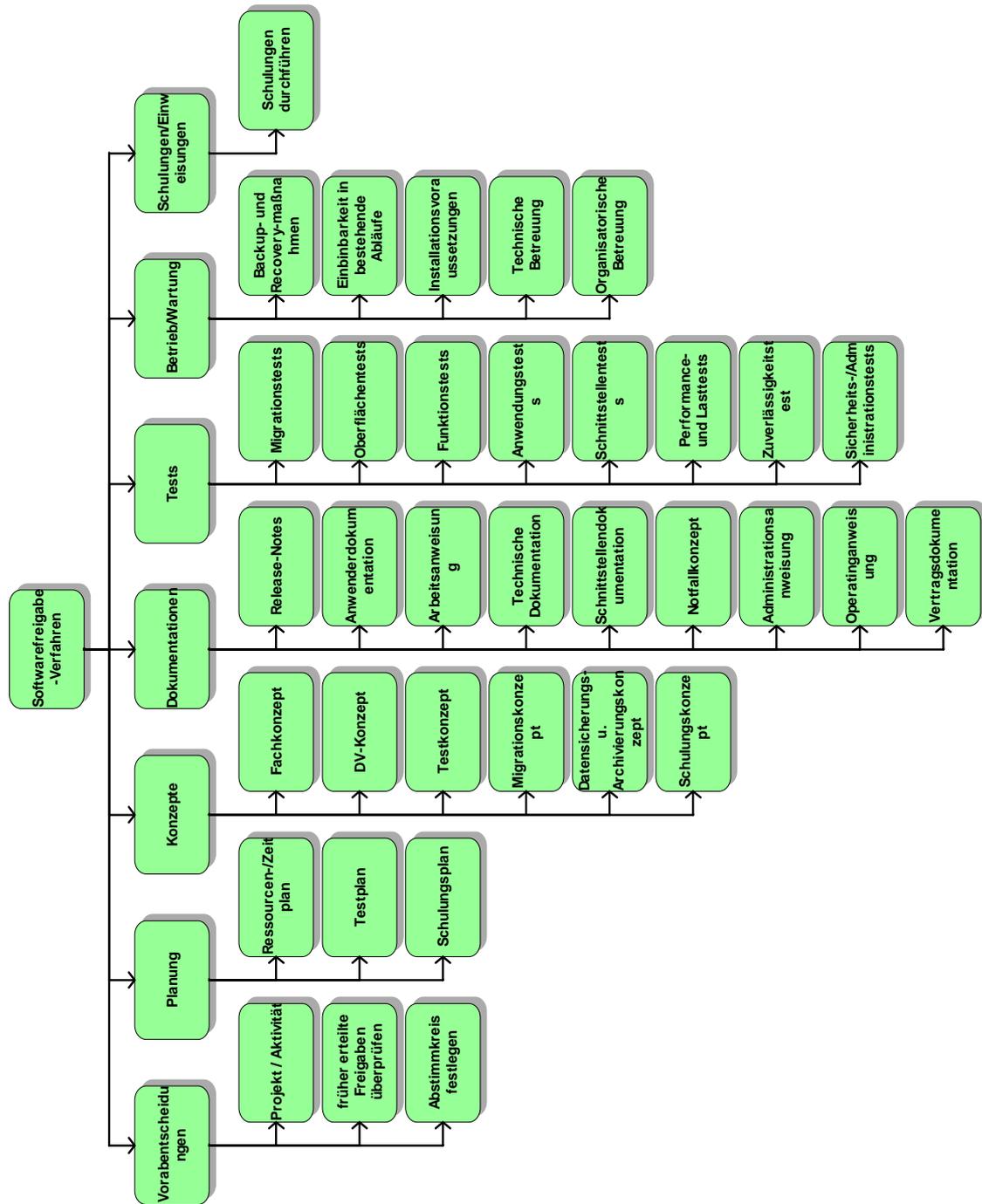
Mit dieser Arbeit möchte ich zuerst einen kleinen Überblick über die Welt der Modellierung durch einige theoretische sowie technische Grundlagen erschaffen. Da Modellierung ein sehr weites Gebiet umfasst, gibt es zu dieser Thematik auch sehr viel Literatur. Einige davon kann man aus dem Literaturverzeichnis als Empfehlungen bei besonderem Interesse zu sich nehmen.

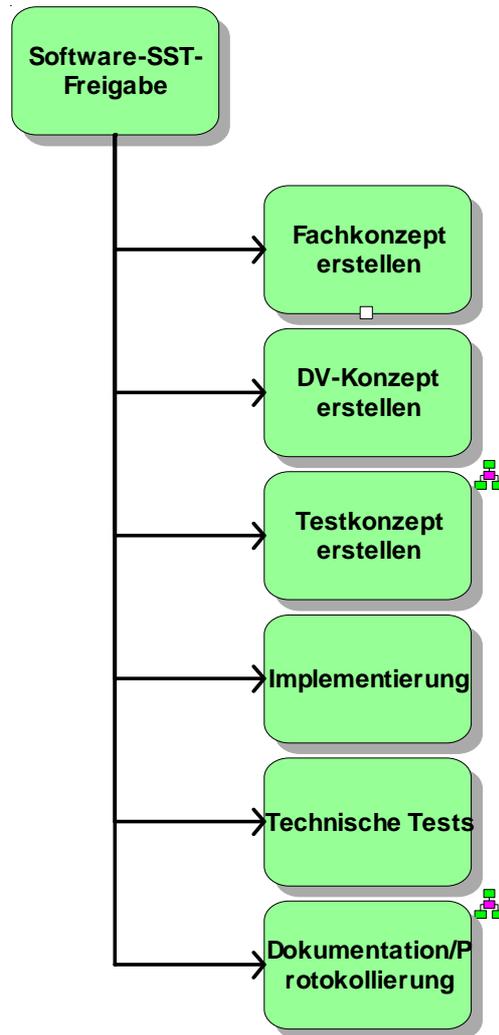
Referenzmodellierung als eine Forschungsrichtung im Bereich Modellierung ist seit langem auch unendliches Interesse vieler großer Unternehmen. Sie haben den Nutzen der Entwicklung solches Modells für eigene unternehmensspezifische Situation erkannt und deshalb auch sehr intensiv für die Referenzmodellierungsprojekte investiert.

Das Referenzmodell als Ergebnis dieser Studienarbeit möchte dem Unternehmen somit auch als ein Fachkonzept für einen automatischen Entwicklungsprozess der Schnittstellen zur Verfügung gestellt werden. Leider existiert die Sachsen LB zum Zeitpunkt dieses Schreibens nicht mehr auf dem Finanzmarkt. Nichtsdestotrotz müssen die hier entwickelten Referenzmodelle auch bestimmte Bedeutung gebracht haben, da der Prozessablauf in jeder Bank fast genauso üblich wie in der Sachsen LB ist.

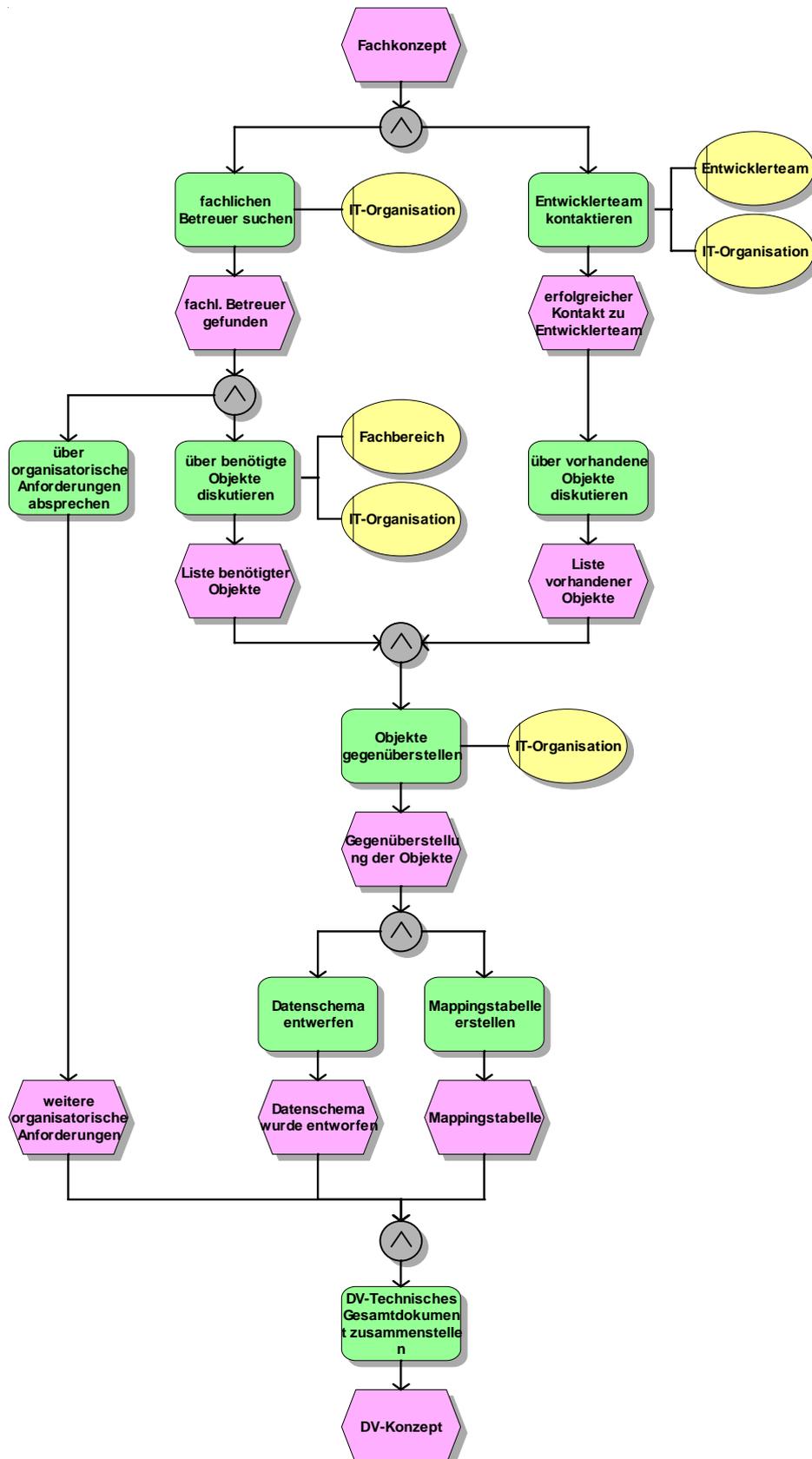
# Anhang

## A Funktionsbaum zu Softwarefreigabe-Verfahren der Sachsen LB

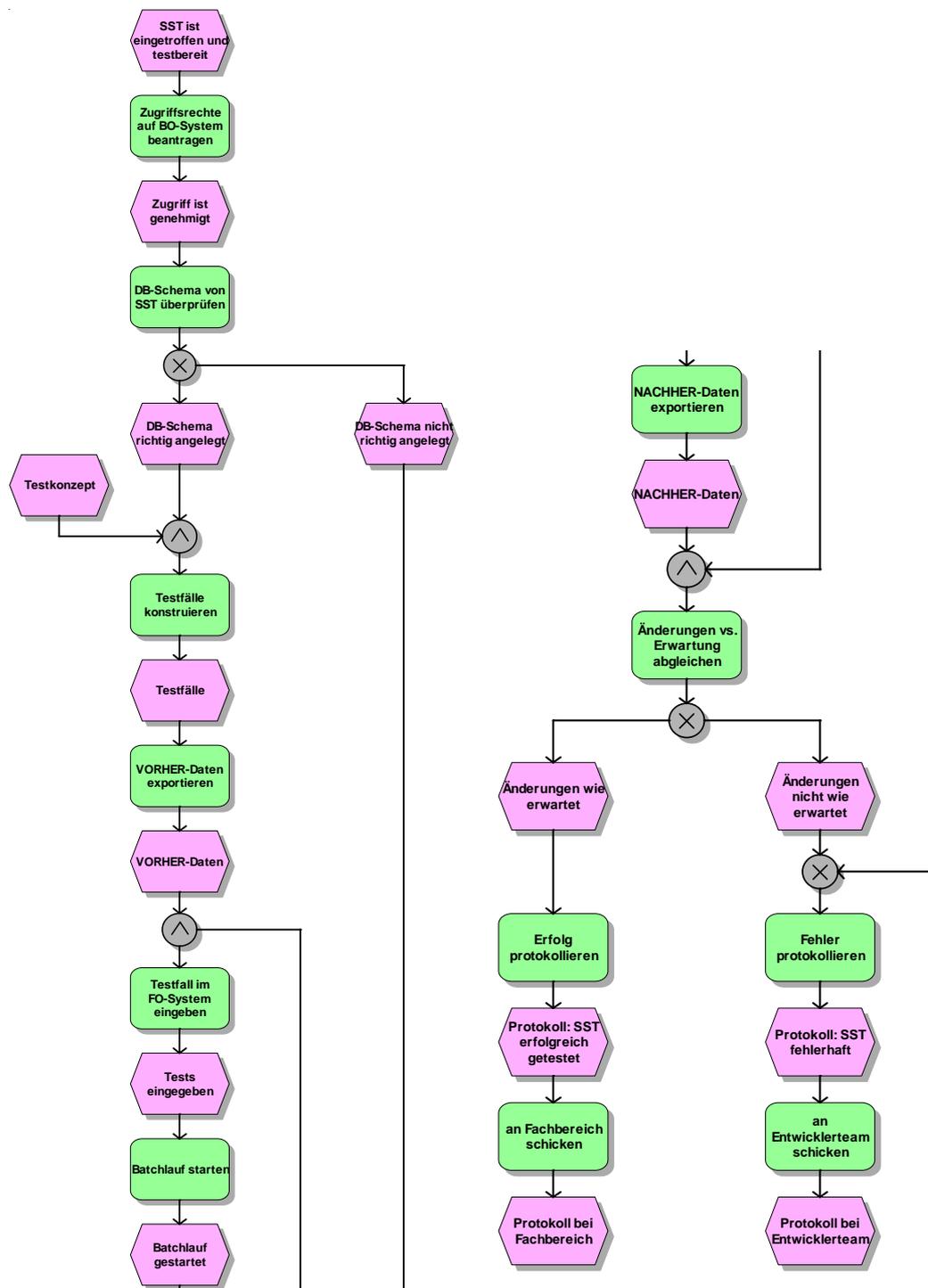


**B Funktionsbaum zur technischen Freigabe von Software-Schnittstellen**

### C Referenzmodell für die technische Freigabe von Software-Schnittstellen (DV-Konzepts-Entwurfsphase)



## D Referenzmodell für die technische Freigabe von Software-Schnittstellen (Technische Testphase)



## Literaturverzeichnis

- Bernroider, E.; Stix V. (2004): Grundzüge der Modellierung – Anwendungen für die Softwareentwicklung. Wien.
- Becker, J.; Rosemann, M.; Schütte, R. (1995): Grundsätze ordnungsgemäßer Modellierung. *Wirtschaftsinformatik* 37, S. 435-445.
- Becker, J. (1998): Referenzmodelle für den Handel. In: Maicher, M.; Scheruhn, H.-J. (1998), S. 85-109.
- Esser, M; Fidelak, M; Prescher, G. (1998): Zwischen Evolution und Revolution: Informationsmodellierung in einem Versorgungsunternehmen. In: Maicher, M.; Scheruhn, H.-J. (1998), S. 37-62.
- Jost, W.; Wagner, K. (2002): Das ARIS Toolset. In: Scheer, A.-W.; Jost, W. (2002), S. 15-32.
- Laudon, K.C.; Laudon, J.P.; Schoder, D. (2006): *Wirtschaftsinformatik – Eine Einführung*. München.
- Maicher, M.; Scheruhn, H.-J. (Hrsg.) (1998): *Informationsmodellierung – Referenzmodelle und Werkzeuge*. Wiesbaden.
- Mücke, A. (2008): *Service Centric Banking als Agentenbasiertes System*. Dissertation, Technische Universität Berlin.
- Rautenstrauch, C.; Schulze, T. (2003): *Informatik für Wirtschaftswissenschaftler und Wirtschaftsinformatiker*. Berlin u. a.
- Rosemann, M. (1996): *Komplexitätsmanagement in Prozeßmodellen – Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung*. Wiesbaden.
- Rosemann, M. (1998): Die Grundsätze ordnungsmäßiger Modellierung – Intention, Entwicklung, Architektur und Multiperspektivität. In: Maicher, M.; Scheruhn, H.-J. (1998), S. 1-21.
- Scheer, A.-W. (1992): *Architektur integrierter Informationssysteme – Grundlagen der Unternehmensmodellierung*. 2. Aufl., Berlin u. a.
- Scheer, A.-W.; Jost, W. (Hrsg.) (2002): *ARIS in der Praxis – Gestaltung, Implementierung und Optimierung von Geschäftsprozessen*. Berlin u. a.
- Schütte, R. (1998): Referenzmodellierung: Anforderungen der Praxis und methodische Konzepte. In: Maicher, M.; Scheruhn, H.-J. (1998), S. 63-84.

## **Abschließende Erklärung**

Ich versichere hiermit, dass ich die vorliegende Studienarbeit selbstständig, ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Magdeburg, den 25. Juli 2008