

Referenzprofil

Database Developer

Irmhild Rogalla

Dieses Referenzprofil wurde im Rahmen des bmb+f geförderten Projekts
„Arbeitsprozessorientierte Weiterbildung in der IT-Branche“ erarbeitet von:


Fraunhofer
Institut
Software- und
Systemtechnik

Fraunhofer ISST


CDI Deutsche Private
Akademie für Wirtschaft
GmbH

Bildungspartner


SOFTWARE POWERS THE INTERNET™
Oracle Deutschland GmbH

Unternehmenspartner

Danksagung

Diese Profilbeschreibung entstand auf Basis eines Praxisprojekts der Firma Oracle, deren Projektleiter Sascha Nehm wir herzlich für seine fachkundige und umfassende Hilfe danken. Fachlich beratend mitgewirkt haben Dietrich M. Scheringer, CDI und Jörn G. Süß, TU Berlin. Ohne ihre Hilfe hätte dieses Dokument nicht entstehen können.

Inhalt

1	EINFÜHRUNG: REFERENZPROZESSE ALS CURRICULA	4
1.1	EREIGNIS-PROZESS-KETTEN: SYMBOLIK	5
1.2	REFERENZPROZESS UND TEILPROZESSE	7
2	DAS PROFIL: DATABASE DEVELOPER.....	9
2.1	TÄTIGKEITSBESCHREIBUNG	9
2.2	PROFILTYPISCHE ARBEITSPROZESSE.....	10
2.3	PROFILPRÄGENDE KOMPETENZFELDER.....	11
2.4	QUALIFIKATIONSERFORDERNISSE.....	11
2.5	EINORDNUNG INS SYSTEM UND KARRIEREPFADE.....	12
3	REFERENZPROZESS.....	13
3.1	DATABASE DEVELOPING	13
3.1.1	Referenzprozess Datenbankentwicklung	14
3.1.2	Das Beispielprojekt: Entwicklung einer Datenbank für die Vereinfachung der Geschäftsprozesse einer international tätigen Bank	17
3.1.3	Prozesskompass Datenbankentwicklung.....	18
3.1.4	Teilprozesse: Datenbankentwicklung.....	19
3.1.4.1	Unterstützen von Systemanalyse und -design.....	20
3.1.4.2	Mitwirken bei der Festlegung des Entwicklungsrahmens.....	24
3.1.4.3	Überprüfen des Systemdesigns.....	26
3.1.4.4	Einbeziehen und Abgleichen von Standards.....	28
3.1.4.5	Verfeinern der Systementwürfe.....	30
3.1.4.6	Erstellen des Migrationsmodells	32
3.1.4.7	Analysieren der Datenbankzugriffe und Erstellen des Mengengerüsts	34
3.1.4.8	Ableiten der Testfälle	36
3.1.4.9	Erstellen des physischen Datenbankdesigns.....	38
3.1.4.10	Einrichten der Entwicklungsumgebung beim Kunden.....	40
3.1.4.11	Abstimmen der Anwendungslogik im Team.....	42
3.1.4.12	Realisieren der Stored Procedures und Trigger.....	44
3.1.4.13	Programmieren der Software-Bausteine der Zugriffsschicht.....	46
3.1.4.14	Mitarbeiten bei Systemintegration und Systemtests.....	48
3.1.4.15	Mitarbeiten bei der Installation der Datenbank und der Migration der Bestandsdaten	50
3.1.4.16	Mitarbeiten bei der Erstellung von Handbüchern und bei den Nutzerschulungen	53

1 Einführung: Referenzprozesse als Curricula

Das Referenzprojekt des Database Developer verdeutlicht paradigmatisch die diesem Tätigkeitsfeld zu Grunde liegenden Arbeitsprozesse, die mit ihnen verbundenen Ansprüche sowie die daraus resultierenden Anforderungen an Inhalt und Durchführung einer qualitativ hochwertigen Weiterbildung.

Das Referenzprojekt erfüllt mehrere Funktionen:

Aus der Praxis für die Praxis:

Als Abstraktion tatsächlich stattgefundener Projekte und Prozesse bieten die Referenzprozesse eine realistische und leicht nachvollziehbare Abbildung dessen, was die Tätigkeiten eines Database Developer sind.

Prozessorientierung als innovatives „Curriculum“:

Als vollständige Darstellung aller wichtigen Arbeitsprozesse sowie der dazugehörigen Qualifikationen, Tätigkeiten und Werkzeuge bieten die Referenzprozesse die Grundlage für die Weiterbildung zum Database Developer. Alle diese Prozesse müssen - entsprechend den Vorgaben - einmal oder mehrfach durchlaufen werden und ermöglichen dadurch den Weiterzubildenden den arbeitsplatznahen, integrativen Erwerb von relevanten Kompetenzen. Durch den Verbleib im Arbeitsprozess wird nicht nur für die Weiterzubildenden eine hohe Motivation (Arbeit an echten Projekten/Aufgaben) und Nachhaltigkeit erreicht, sondern auch - aus Sicht des Unternehmens - die Kontinuität und Qualität der laufenden Arbeiten gesichert (keine Ausfallzeit durch Seminartage, kein mühsamer Transfer).

Qualitätsstandard für die Weiterbildung:

Als Referenz bieten insbesondere die Teilprozesse und die mit ihnen verbundenen Tätigkeits- und Qualifikationsziele einen Qualitätsmaßstab für die arbeitsprozessorientierte Weiterbildung und die resultierenden Abschlüsse. Vollständige Transparenz und klare Zielvorgaben ermöglichen die qualitativ hochwertige Absicherung auch komplexer Kompetenzen sowie den systematischen Erwerb des notwendigen Erfahrungswissens.

Transferprozesse:

Die Generalisierung des Referenzprojekts aus der Praxis und seine didaktische Anreicherung ermöglichen eine leichte Auswahl angemessener Transferprozesse, deren Bearbeitung die Grundlage der Weiterbildung ist. Transferprozesse sind reale Prozesse, die Referenzprojekte in einer lernförderlichen Umgebung abbilden. Abgeschlossene Transferprozesse auf Basis der hier dargestellten Anforderungen und Qualitätsmaßstäbe sind nicht nur Qualifikationsnachweis des Einzelnen, sondern bilden auch die Basis eines angemesseneren und zielgerichteteren Umgangs mit Geschäfts- und Arbeitsprozessen im Unternehmen.

1.1 Ereignis-Prozess-Ketten: Symbolik

Die Darstellung der Referenzprozesse in Form von Ereignis-Prozess-Ketten¹ ermöglicht einen schnellen Überblick. Vollständigkeit kann leicht überprüft werden, Anpassungen und Modifikationen in Hinblick auf das eigene Unternehmen sind problemlos möglich und Anknüpfungspunkte an andere Prozesse, aber auch zu weiterführenden Informationen ergeben sich automatisch.

Die bei der Darstellung der Referenz- und Teilprozesse verwendete Modellierungssprache stellt eine Anpassung und Weiterentwicklung der klassischen EPK-Modellierung dar:

- Referenz- wie Teilprozesse sind aus der Sicht des jeweiligen Spezialisten, also als Arbeitsprozesse einer Person dargestellt.
- Referenz- wie Teilprozesse stellen in der Regel keinen Geschäftsprozess dar.

Die EPK-Symbole werden hier wie folgt verwendet:

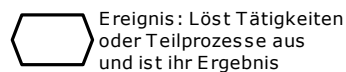
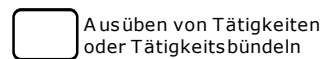
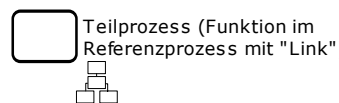


Abbildung A: Grundlegende Symbole der Referenz- und Teilprozessmodelle

Die wichtigsten Symbole sind:


die Tätigkeiten bzw. Tätigkeitsbündel oder Teilprozesse, die mit dem Funktionssymbol dargestellt werden;


die Ereignisse, die Tätigkeiten bzw. Teilprozesse auslösen und Ergebnisse von Teilprozessen sind.

Grundsätzlich gilt: Auf ein Ereignis folgt immer ein Teilprozess bzw. eine Tätigkeit.

Ergebnisse von Tätigkeiten sind sehr oft Dokumente, diese werden dann zusätzlich durch das Dokument-Symbol dargestellt.

¹ vgl. A.-W. Scheer, Wirtschaftsinformatik, Springer 1998

 UND-Verknüpfung

 XOR-Verknüpfung


 ODER-Verknüpfung

Abbildung B: Konnektoren

Wenn Alternativ-Möglichkeiten bestehen, werden Ereignisse und Teilprozesse/Tätigkeiten über Konnektoren (AND, OR, XOR) verbunden. Dabei steht AND für ein verbindendes „und“, OR für ein „oder“, das alle Möglichkeiten offen lässt und XOR für ein „ausschließendes oder“, welches nur einen der angegebenen Pfade ermöglicht.

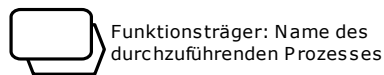


Abbildung C: Schnittstelle

Da die Prozesse aus der Sicht des jeweiligen Spezialisten formuliert werden, sind Schnittstellen zu Prozessen anderer Spezialisten oder zu Entscheidungsprozessen auf höherer Ebene notwendig. Dazu wird das Schnittstellensymbol verwendet. Es steht für Prozesse, die der Spezialist nicht selber durchführt, auf deren Durchführung er aber angewiesen ist. Parallel zu jeder Schnittstelle wird die Tätigkeit dargestellt, die der Spezialist selbst in diesem Zusammenhang ausübt, wie „Beraten bei ...“, „Unterstützen bei ...“ oder „Informieren des ...“.

Alle Prozesse werden durch die Verwendung dieser Symbole klar und einfach strukturiert dargestellt und sind offen für die Übertragung in konkrete Transferprozesse.

1.2 Referenzprozess und Teilprozesse

Der hier vorgestellte Referenzprozess und seine Teilprozesse stellen das Curriculum des Spezialistenprofils Database Developer dar.

Der Referenzprozess erhebt nicht den Anspruch eines Vorgehensmodells, sondern bildet beispielhaft den möglichen Arbeitsprozess und Verlauf eines Projekts auf Spezialistenebene ab.

Er bildet die Grundlage für Weiterbildungen und damit einen Qualitäts-, Niveau- und Komplexitätsmaßstab. Die zugehörigen Teilprozesse sind hier beispielhaft modelliert und stellen eine Möglichkeit der Durchführung dar. Einzelheiten zu den unverzichtbaren Prozessen und Kompetenzfeldern sind hier im Referenzprojekt festgelegt. Die Reihenfolge und die Inhalte der Teilprozesse sind abhängig vom jeweils auszuwählenden Transferprojekt und werden in diesem Zusammenhang festgelegt.

Die Darstellung der Prozesse erfolgt systematisch:

Jeder Prozess wird mit Hilfe von Ereignis-Prozess-Ketten dargestellt. Einem auslösenden Ereignis folgt eine Funktion, die wiederum ein oder mehrere Ereignisse als Ergebnis hat. Ereignisse und Funktionen können mit AND, OR oder XOR, den Konnektoren, verbunden sein.

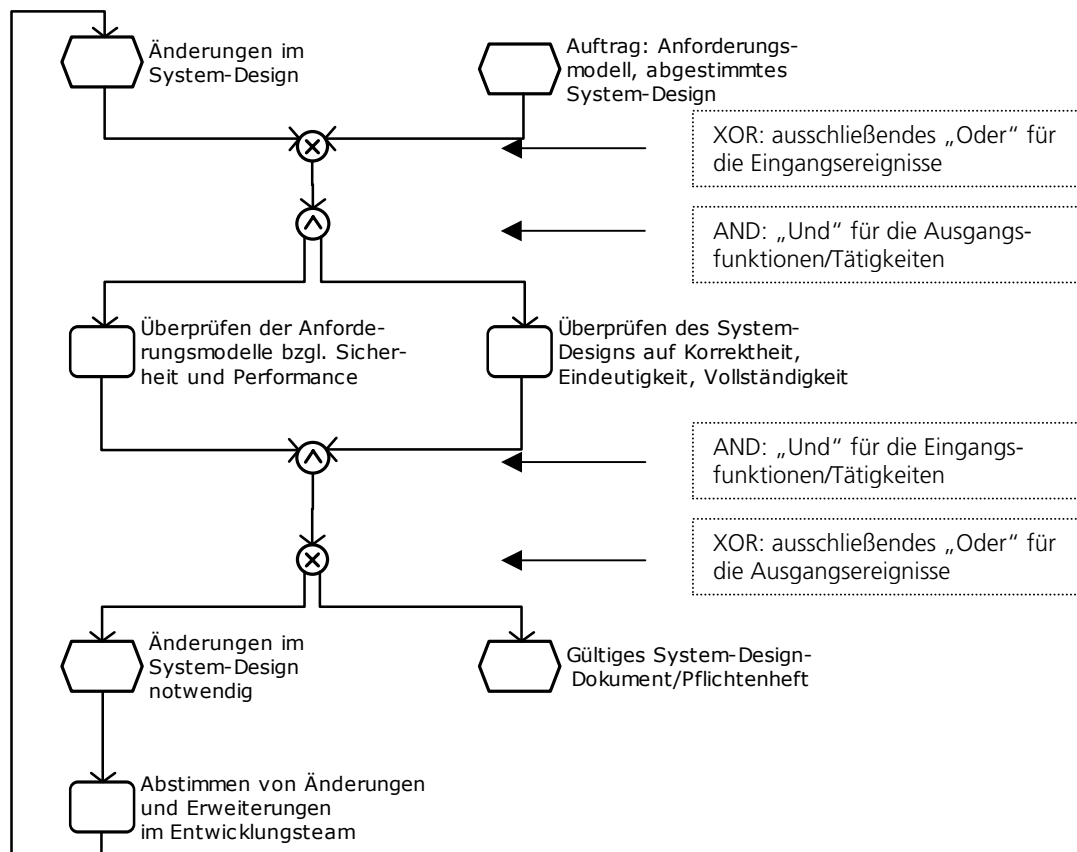


Abbildung D: Beispielprozess (Teilprozess "Überprüfen des Systemdesigns") mit unterschiedlicher Verwendung von Konnektoren

Die Verbindung von Referenzprozess und Teilprozessen erfolgt über die Funktionen des Referenzprozesses:

Jede Funktion im Referenzprozess steht für einen Teilprozess.

Ereignisse, die dem jeweiligen Teilprozess direkt vor oder nachgeordnet sind, sind Anfangs- und Endereignisse der jeweiligen Teilprozesse. Damit stellen die Teilprozesse die Funktionen des Referenzprozesses ausführlich dar, und ein Hin- und Herbewegen zwischen Referenz- und Teilprozessen ist jederzeit problemlos möglich.

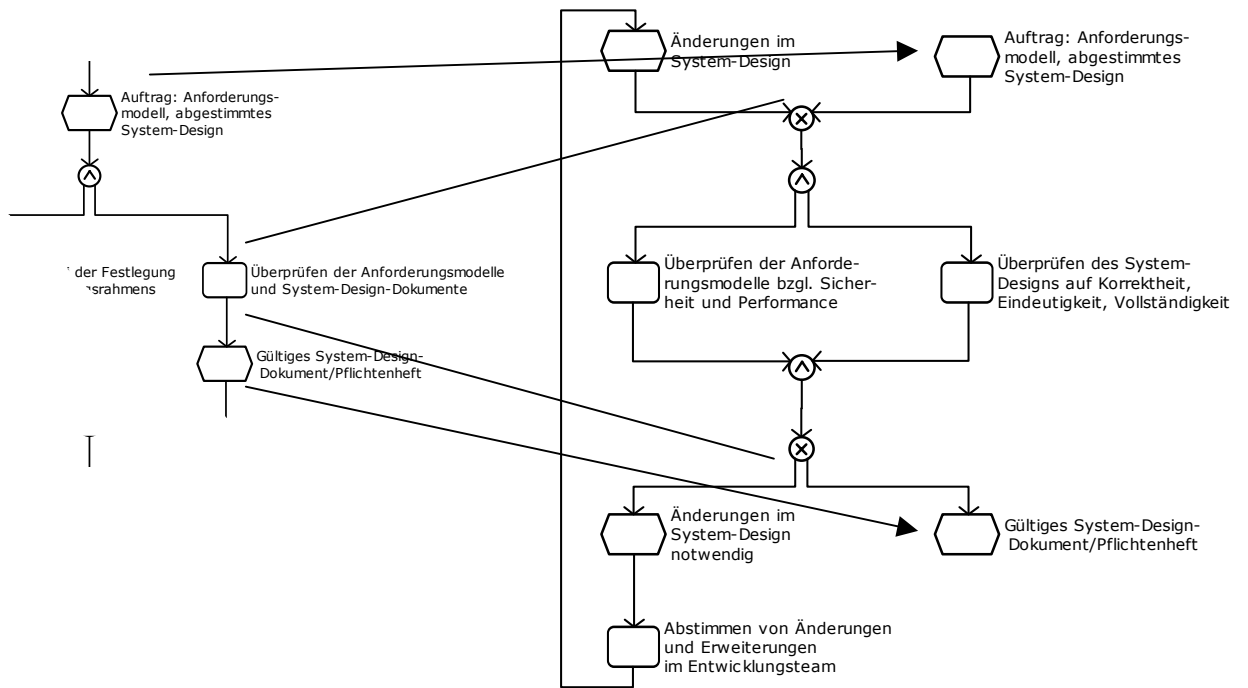


Abbildung E: Ausschnitt aus dem Referenzprozess des Database Developer

Abbildung F: Teilprozess des Database Developer "Überprüfen des Systemdesigns"

Die Teilprozesse stellen so die wesentlichen Teile eines Projekts dar und lassen sich entsprechend auf Transferprojekte übertragen. Den Teilprozessen sind die jeweils wesentlichen Tätigkeiten und Kompetenzfelder zugeordnet.

2 Das Profil: Database Developer

Database Developer² konzipieren und implementieren Datenbanken bedarfsgerecht und wirtschaftlich. Sie nutzen hierzu standardisierte Prinzipien, Methoden, Sprachen und die von den zugrunde liegenden Datenbank-Managementsystemen zur Verfügung gestellten Werkzeuge.

Database Developer optimieren das Leistungsverhalten von IT-Systemen durch einen auf die Systemanforderungen zugeschnittenen Datenbankentwurf. Sie sind darüber hinaus für Maßnahmen zur Unterstützung der Mitführung vorhandener Datenbestände auf neue Systeme zuständig (Migration). Sie unterstützen Unit-, Integrations- und Gesamtsystem-Tests.

2.1 Tätigkeitsbeschreibung

Die zentrale Tätigkeit des Database Developer ist das Programmieren von Datenbanken bzw. Datenbanksystemen, insbesondere in der industriellen Form der Softwareproduktion mit standardisierten Prinzipien, Methoden, Sprachen und Werkzeugen. Hier muss der Database Developer auf eine hohe Benutzerakzeptanz und deutliche Rationalisierung des Erstellungsprozesses achten. Dazu arbeitet er eng mit dem Konfigurationsmanagement und der Qualitätssicherung zusammen um bei den von ihm erstellen Datenbankanwendungen Zuverlässigkeit, Benutzerfreundlichkeit, Effizienz, Wartungsfreundlichkeit, Testbarkeit und Flexibilität zu sichern. Modul-, Programm- und Datenbanksystem-Tests werden vom Database Developer in und mit seiner Arbeit ebenfalls unterstützt. Verantwortlich ist der Database Developer weiterhin für die Optimierung des Leistungsverhaltens (Performance) des Systems, welche schon beim Datenbankentwurf eine Rolle spielen sollte. Ein weiteres wichtiges Tätigkeitsfeld des Database Developer ist die Berücksichtigung und das Mitführen vorhandener Datenbestände auf neue Systeme (Migration). Das Erstellen und Implementieren von Sicherheitskonzepten kommt nur noch am Rande vor: Diese Funktionen werden mehr und mehr bereits in der Anwender- bzw. Middleware-Schicht implementiert und spielen auf der eigentlichen Datenbankebene kaum noch eine Rolle.

Darüber hinaus hat der Database Developer regelmäßigen Kundenkontakt: Von der Beratung nach Kundenanfragen, über die Systementwicklung, die oft beim Kunden und in engem Austausch erfolgt, bis hin zur Anwenderschulung reichen seine Aufgaben. Der Übergang zur Beteiligung an Führungsaufgaben aus dem Bereich des Projektmanagements ist in der Regel fließend, sei es durch Teilprojektleitung oder durch Entscheidungsvorbereitung bzw. -beteiligung für Vorgesetzte (Professionals).

² Das Kapitel 2: „Das Profil: Database Developer“ gibt - mit Ausnahme des Abschnitts 2.1 „Tätigkeitsbeschreibung“ - den offiziellen Text der „Vereinbarung über die Spezialistenprofile im Rahmen des Verfahrens zur Ordnung der IT-Weiterbildung“ vom 25.05.2002 (Bundesanzeiger 105, ausgegeben am 12.06.2002) wieder.

2.2 Profiltypische Arbeitsprozesse

Die im Folgenden beschriebenen Teilprozesse dokumentieren den gesamten profiltypischen Arbeitsprozess des Database Developer. Die Beherrschung dieses Arbeitsprozesses in Verbindung mit den Kompetenzen in den jeweiligen Kompetenzfeldern und der Berufserfahrung bildet die Grundlage für die berufliche Handlungskompetenz.

1. Unterstützen von IT-Systemanalytikern und IT-Systemplanern bei der Systemanalyse und dem System-Design, zum Beispiel durch Erstellen von Prototypen.
2. Mitwirken bei der Festlegung des Entwicklungsrahmens und der Entwicklungsumgebung, der Abschätzung der Aufwände, der Festlegung von Meilensteinen und der Identifizierung von Implementierungsrisiken.
3. Überprüfen von Anforderungsmodellen und System-Design-Dokumenten auf Korrektheit, Eindeutigkeit und Vollständigkeit sowie auf Realisierbarkeit der Systemanforderungen bezüglich Sicherheit und Performance. Abstimmen von Änderungen und Erweiterungen mit IT-Systemanalytikern, IT-Systemplanern und weiteren Spezialisten aus dem Bereich Entwicklung.
4. Einbeziehen und Abgleichen von Standards, um die Einheitlichkeit, Wartbarkeit und Beschreibbarkeit der Systeme und ihrer Handbücher festzulegen.
5. Verfeinern von Systementwürfen durch Abbilden der spezifizierten Systemkomponenten beispielsweise auf Entity-Relationship-Modellen und Systemdatenmodellen, Spezifizieren des dynamischen Verhaltens der Systemkomponenten in Form geeigneter Diagramme.
6. Erstellen von Migrationsmodellen und -werkzeugen, um Datenbestände und Workflow übernehmen zu können.
7. Ableiten von Testfällen und -szenarien aus den Spezifikationen und Bereitstellen von Testdaten für Unit- und Systemtests.
8. Analysieren der Datenbank-Zugriffe und Erstellen des Mengengerüsts als Grundlage für das physische Datenbankdesign, Erstellen einer Prognose über die Entwicklung der Datenmengen und Erwirken von Akzeptanz für diese Prognose.
9. Erstellen des physischen Datenbank-Designs in Abhängigkeit von dem konkreten eingesetzten Datenbank-Managementsystem.
10. Einrichten der Entwicklungsumgebungen bei Kunden sowie Entwickeln und Testen der Datenbanken und Sichern des Produktivbetriebs.
11. Abstimmen mit den Software Developern, um die Plazierung der Anwendungslogik festzulegen: innerhalb der Datenbank (zum Beispiel durch Stored Procedures und Trigger) oder außerhalb der Datenbank in den Software-Bausteinen der Anwendungsschicht.
12. Realisieren der Stored Procedures und Trigger und gegebenenfalls der Software-Bausteine einer Datenbank-Zugriffsschicht als eigentliche Datenbankprogrammierung. Durchführen der Unit-Tests, Festhalten der Testergebnisse.
13. Unterstützen der Systemintegration und der Systemtests bzw. bei kleineren Projekten Durchführen der Systemintegration mit Unterstützung der am Projekt beteiligten Entwickler.
14. Unterstützen von Installationen bei Kunden inklusive der Migration von Bestandsdaten und der Anbindung von Fremdsystemen.
15. Mitarbeiten bei der Erstellung von Handbüchern, Installationsanleitungen und Trainingsmaterialien sowie Durchführung von Schulungen für IT-System- bzw. Datenbankadministratoren.

2.3 Profilprägende Kompetenzfelder

Die Beherrschung der profiltypischen Arbeitsprozesse setzt Kompetenzen unterschiedlicher Reichweite in den nachstehend aufgeführten beruflichen Kompetenzfeldern³ voraus. Den Kompetenzfeldern sind Wissen und Fähigkeiten sowie typische Methoden und Werkzeuge unterschiedlicher Breite und Tiefe zugeordnet.

Grundlegend zu beherrschende, gemeinsame Kompetenzfelder⁴:

- Unternehmensziele und Kundeninteressen,
- Problemanalyse, -lösung,
- Kommunikation, Präsentation,
- Konflikterkennung, -lösung,
- Fremdsprachliche Kommunikation (englisch),
- Projektorganisation, -kooperation,
- Zeitmanagement, Aufgabenplanung und -priorisierung,
- Wirtschaftliches Handeln,
- Selbstlernen, Lernorganisation,
- Innovationspotenziale,
- Datenschutz, -sicherheit,
- Dokumentation, -standards,
- Qualitätssicherung.

Fundiert zu beherrschende, gruppenspezifische Kompetenzfelder:

- Methoden und Werkzeuge der Software-Entwicklung,
- Engineering-Prozesse,
- Systemanalyse,
- Entwicklungsstandards (Leistungsfähigkeit, Sicherheit, Verfügbarkeit, Innovation),
- Qualitätsstandards,
- Datenbanken, Netzwerke.

Routiniert zu beherrschende, profilspezifische Kompetenzfelder:

- Moduldesign, Designmuster
- Datenbankmodellierungsmethoden, -regeln, -verfahren,
- Datenmodelle, -formate, -integrität, -typen,
- Datenbankarchitektur,
- Abfragesprachen,
- Physisches Datenbankdesign.

2.4 Qualifikationserfordernisse

Im Regelfall wird ein hinreichendes Qualifikationsniveau auf der Basis einschlägiger Berufsausbildung oder Berufserfahrung vorausgesetzt.

³ Die Kompetenzfelder werden in der nachfolgenden Auflistung jeweils durch ein zusammenfassendes Stichwort benannt. Da die Weiterbildung zum Spezialisten auf die erfolgreiche Bewältigung zunehmend offener beruflicher Handlungssituationen sowie ganzheitlichen Kompetenzerwerb abzielt, bildet der Kompetenzerwerb einen integralen Bestandteil der Arbeits- und Weiterbildungsprozesse und lässt sich nur im Zusammenhang mit diesen operationalisieren (vgl. dazu die Abschnitte „Kompetenzfelder“ in Kapitel 3.1.4 ff).

⁴ Jeder Spezialist muss in den in diesem Abschnitt genannten Kompetenzfeldern wie „Kommunikation, Präsentation“, „Konflikt-erkennung, -lösung“ usw. ein Niveau erreichen, dass über dem einer Fachkraft liegt. D. h., er muss auch in diesen Feldern zu eigenständigem Handeln in der Lage sein und zum Erreichen des Ziels in dem jeweiligen Feld gegebenenfalls über den Rahmen bekannter Verfahren und Lösungen hinaus gehen können.

2.5 Einordnung ins System und Karrierepfade

Das neue IT-Weiterbildungssystem gibt auf Basis der vier neuen IT-Ausbildungsberufe drei Ebenen für die Weiterqualifizierung vor: Spezialisten, wie auch der Database Developer einer ist, operative und strategische Professionals. Auf der Ebene der Spezialisten existieren eine Reihe verwandter Profile und selbstverständlich kann sich auch der Datenbankentwickler zu einem Professional weiterqualifizieren.

Verwandte Profile

Der Database Developer weist eine Reihe verwandter Profile auf, die sich in drei Gruppen einteilen lassen:

- Advanced Database User, dazu gehören die verschiedenen Administratoren sowie der E Marketing Developer, der E Logistic Developer und nicht zuletzt der Business Systems Adviser.
- Profile, deren Aufgabengebiete, sich mit denen des Database Developer überschneiden können, wie der IT Systems, der Software und der User Interface Developer, der Quality Management Coordinator. Beispielsweise wird der Database Developer bei größeren Anwendungen für den Aufbau der Datenbankschicht zuständig sein, während die Benutzeroberfläche von einem User Interface Developer gestaltet wird.
- Im eigentlichen Sinne verwandte Profile oder Spezialisierungsmöglichkeiten des Datenbankentwicklers sind der Database Administrator sowie der Business System Administrator, der Knowledge Management System Developer sowie der IT Configuration Coordinator.

Aufstiegsqualifizierung

Das Tätigkeitsfeld des Database Developer ist eine ideale Grundlage für Aufstiegsqualifizierungen insbesondere zum IT Business Consultant, mit den Schwerpunkten Erarbeiten und Umsetzen von (auch sehr spezifischen und hochkarätigen) IT-Systemlösungen und zum IT Business Manager mit den Schwerpunkten Koordinieren, Steuern und Unterstützen von Projekten und Prozessen zur Absicherung der jeweiligen Projekt- und Prozessziele.

3 Referenzprozess

Der Referenzprozess des Database Developer ist die Entwicklung einer Datenbankapplikation, vom Systementwurf bis zum endgültigen Systemtest.

3.1 Database Developing

Die Entwicklung der Datenbankapplikation als Referenzprozess des Database Developer besteht - kurz zusammengefasst - aus folgenden ineinandergreifenden Teilen:

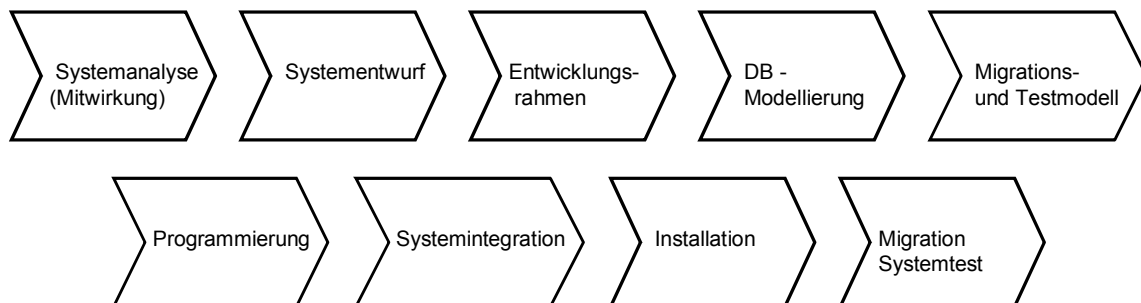


Abbildung G: Zusammenfassung des Referenzprozesses Datenbankentwicklung

Diese Prozesse werden im Folgenden ausführlich dargestellt:

Der Referenzprozess gibt den gesamten Datenbankentwicklungsprozess auf hohem Abstraktionsniveau wieder und ermöglicht so einen Überblick.

Mit den Teilprozessen in den Referenzprozess hineingezoomt. Die Teilprozesse entsprechen damit in etwa der Abbildung von Arbeitsprozessen, sie stellen einen konkreten Tätigkeitsverlauf, einschließlich auslösendem Ereignis und Ergebnis, dar.

Die zur Durchführung der Teilprozesse notwendigen Tätigkeiten und Kompetenzfelder werden jeweils in einem separaten Abschnitt aufgelistet.

Das Praxisprojekt dient als Beispiel zur Konkretisierung und Veranschaulichung. Es ist ein echtes, bereits durchgeführtes Projekt, auf dessen Grundlage die hier dargestellten Referenz- und Teilprozesse entwickelt wurden.

3.1.1 Referenzprozess Datenbankentwicklung

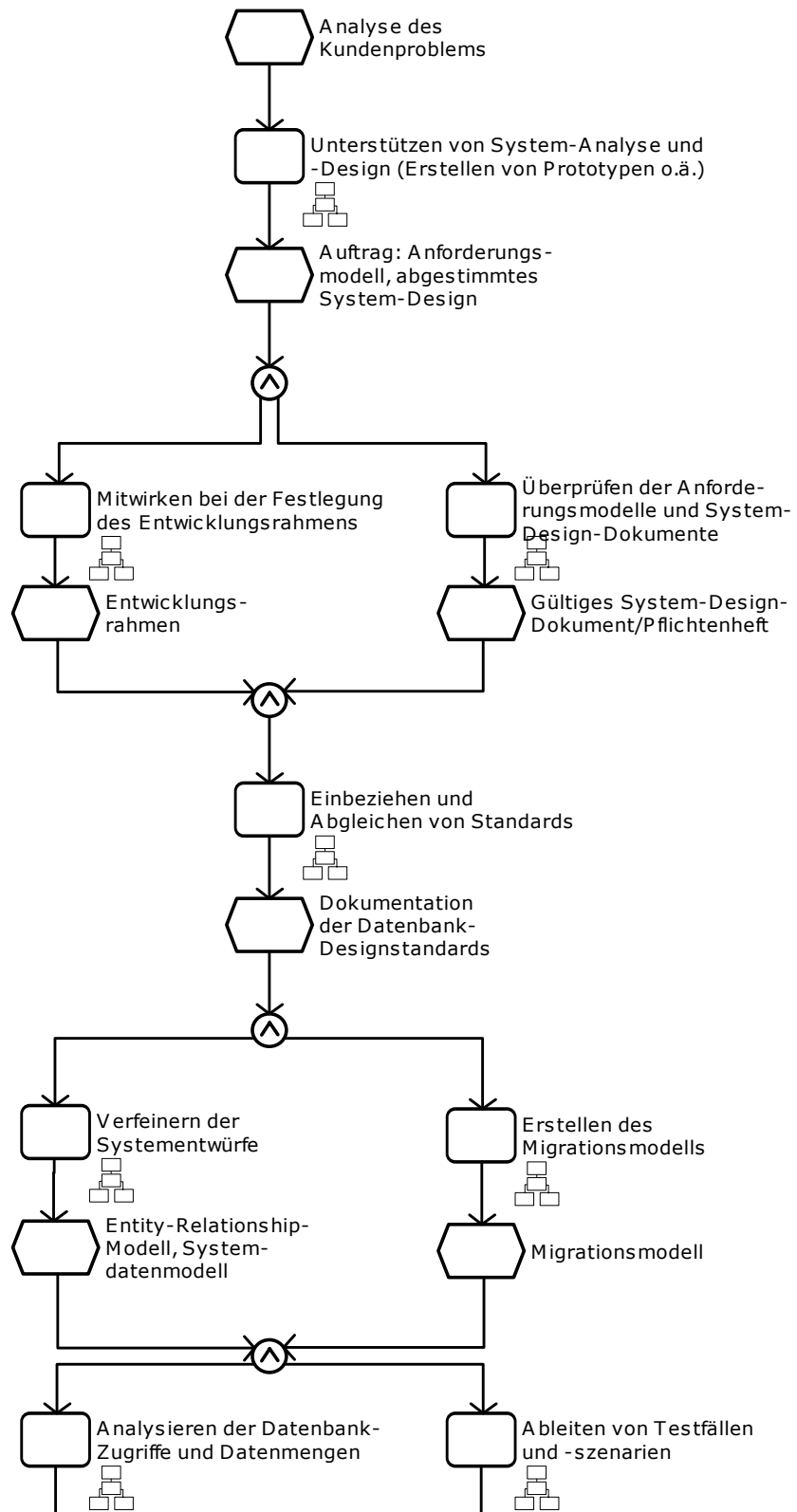


Abbildung: Referenzprozess Database Developer, Teil 1

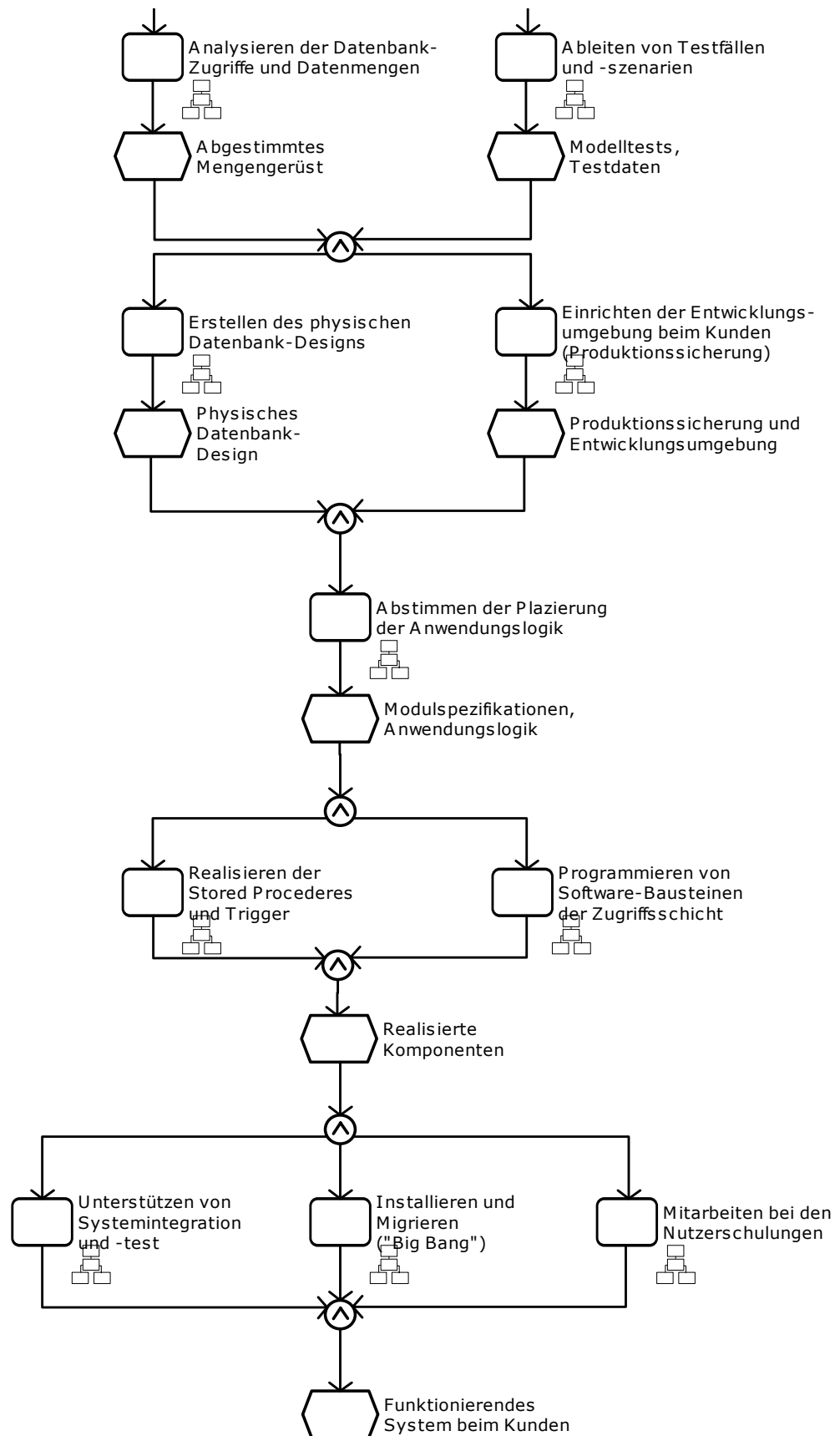


Abbildung: Referenzprozess Database Developer, Teil 2

Der Datenbankentwicklungsprozess ist typisch für ein Developer-Profil: Nach der fachlichen und formalen Festlegung des Projektrahmens geht es mit der Festlegung von Standards, dem Migrationsmodell und der Analyse der Datenbankzugriffe um die Spezifika der Datenbank

bzw. des Datenbankmanagementsystems. Sind auch die Testfälle und das physische Datenbankdesign festgelegt sowie die Entwicklungsumgebung eingerichtet, erfolgt die Programmierung⁵ im engeren Sinne, selbstverständlich in Abstimmung mit dem Team. Nach der Systemintegration, der Installation und Migration sowie der Unterstützung der System-einführung endet das Projekt damit, dass der Kunde ein funktionierendes System hat.

Dieser Prozess läuft in kleinen wie in großen Projekten ab, in denen Datenbankentwicklung eine Rolle spielt. Allerdings wird nicht in jedem Projekt jeder Teilprozess den gleichen Umfang und die gleiche Komplexität haben. So werden in großen Projekten sicherlich mehrere Database Developer zusammenarbeiten und es wird Migrations- und Integrations-spezialisten geben. Kleine Projekte hingegen kann der Datenbankentwickler alleine durchführen, einschließlich der Systemanalyse und der Nutzerschulung.

⁵ In dem hier vorgestellten Praxis- wie Referenzprojekt wird die Programmierung einer relationalen Datenbank zugrunde gelegt. Objektorientierte Datenbankprogrammierung wird damit nicht ausgeschlossen, aber auch nicht explizit beschrieben.

3.1.2 Das Beispielprojekt: Entwicklung einer Datenbank für die Vereinfachung der Geschäftsprozesse einer international tätigen Bank

Der Kunde von Oracle ist die ABN AMRO Bank (Deutschland) AG, mit Niederlassungen in mehreren europäischen und außereuropäischen Ländern. Die Idee hinter dem Projekt besteht in einer Vereinfachung der Geschäftsprozesse, die bei der Kundenbetreuung anfallen. Für die Kundenbetreuung und -verwaltung ist eine Vielzahl von (oft länder-spezifischen) Formularen notwendig, die einen hohen Aktualisierungsbedarf haben und bei deren Benutzung die einschlägigen Vorschriften und Rundschreiben (beispielsweise vom Bundesverband deutscher Banken oder den Landeszentralbanken) beachtet werden müssen. Bislang wurden diese Formulare noch von Hand ausgefüllt und verwaltet, was einen hohen Aufwand mit sich bringt. Außerdem befinden sich einschlägige Handbücher und Rundschreiben oft im Umlauf und daher kann auf sie nicht ohne weiteres zugegriffen werden.

Der Kunde hat sich bereits auf dem Markt nach Software-Lösungen für dieses Problem umgesehen und sich auch bereits von einem anderen Unternehmen beraten lassen. Insbesondere sind die in das System einzubringenden Formulare bereits beschrieben und einige weitere Anforderungen (alle Formulare, Rundschreiben, Handbücher online; Zugriff per Internet auf alle Formulare für alle Nutzer; effiziente und einfache Administration, besonders bez. des Einstellens neuer Formulare und sämtlicher Aktualisierungen; Volltextsuche; einfache Verzeichnisstruktur, angelehnt an Windows-Explorer; automatisches Versionsmanagement) spezifiziert.

Allerdings stellen die bisherigen Angebote den Kunden nicht zufrieden. Die auf dem Markt für diese Probleme vorhandenen Lösungen sind nicht flexibel und komfortabel genug und insbesondere gibt es keine netzwerkfähige Lösung (Internet/Intranet), was eine ganze Reihe von Nachteilen (kein Zugriff auf Online-Handbücher; keine Volltextsuche über Formulare und Handbücher; keine Links zwischen Formularen und zugehörigen Handbüchern; keine Archivierung) mit sich bringt.

3.1.3 Prozesskompass Datenbankentwicklung

1. Unterstützen von Systemanalyse und -design
2. Mitwirken bei der Festlegung des Entwicklungsrahmens
3. Überprüfen des Systemdesigns
4. Einbeziehen und Abgleichen von Standards
5. Verfeinern der Systementwürfe
6. Erstellen des Migrationsmodells
7. Analysieren der Datenbankzugriffe und Erstellen des Mengengerüsts
8. Ableiten der Testfälle
9. Erstellen des physischen Datenbankdesigns
10. Einrichten der Entwicklungsumgebung beim Kunden
11. Abstimmen der Anwendungslogik im Team
12. Realisieren der Stored Procedures und Trigger
13. Programmieren der Software-Bausteine der Zugriffsschicht
14. Mitarbeiten bei Systemintegration und Systemtests
15. Mitarbeiten bei der Installation der Datenbank und der Migration der Bestandsdaten
16. Mitarbeiten bei der Erstellung von Handbüchern und bei den Nutzerschulungen

3.1.4 Teilprozesse: Datenbankentwicklung

Die Teilprozesse geben den Entwicklungsprozess einer Datenbank ausführlich und detailliert wieder. Sie entsprechen so einem realen Kundenprojekt, welches als Grundlage für den Referenz- und die Teilprozesse gedient hat und als Beispiel zur Veranschaulichung beschrieben wird.

Sicherlich werden nicht in jedem Datenbankentwicklungsprojekt alle diese Teilprozesse vorkommen. Insbesondere Prozesse wie das Mitwirken bei der Festlegung des Entwicklungsrahmens oder bei den Nutzerschulungen, sind sehr von der konkreten Einbettung des Projekts abhängig. Ein Database Developer auf der Spezialistenebene sollte allerdings alle diese Prozesse beherrschen. Die Betonung liegt dennoch auf den Prozessen, die für die Datenbankentwicklung zentral sind, angefangen vom Überprüfen der Anforderungsmodelle bis hin zum Installieren und Migrieren.

3.1.4.1 Unterstützen von Systemanalyse und -design

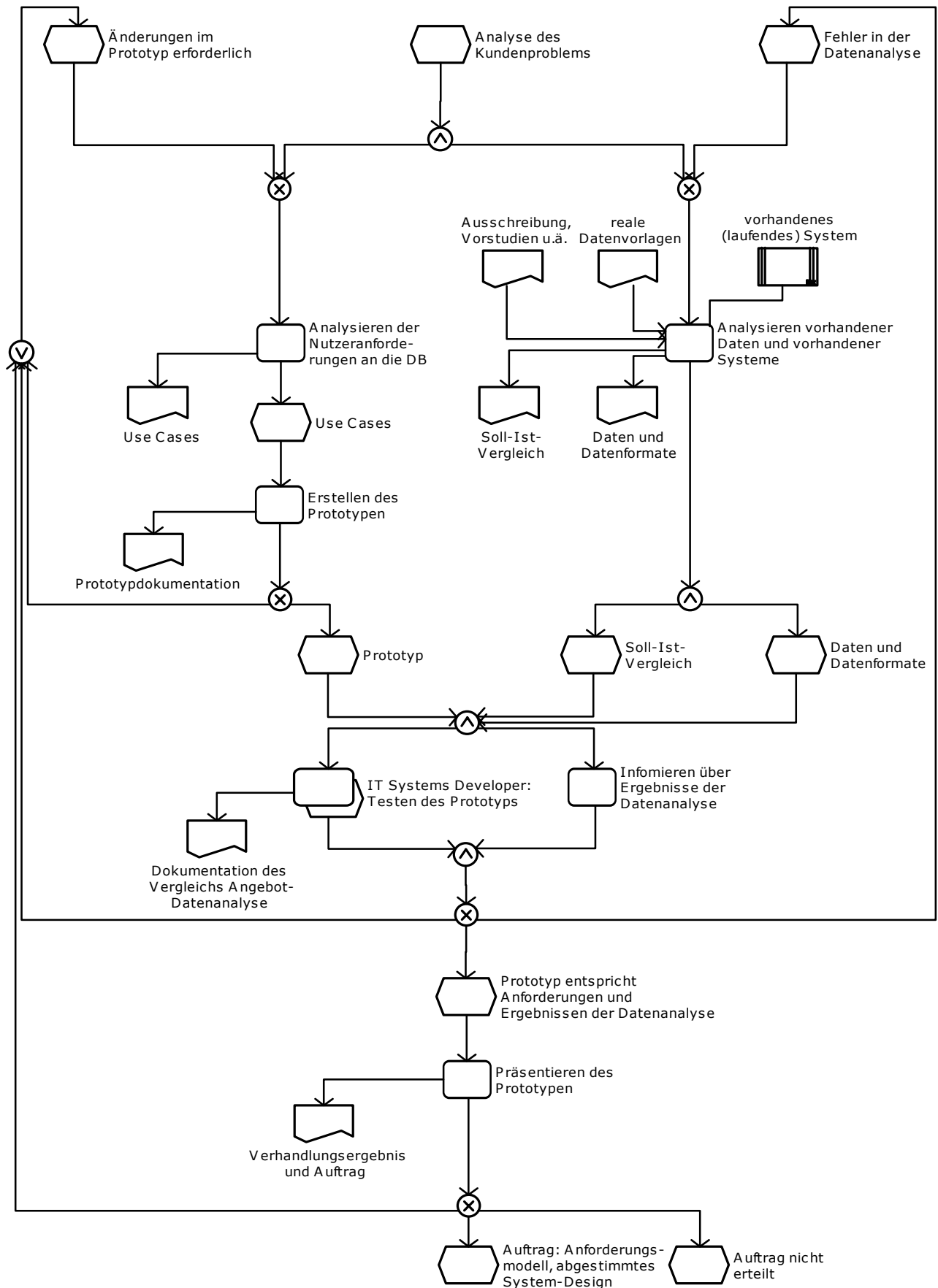


Abbildung 1: Unterstützen von Systemanalyse und -design

3.1.4.1.1 Tätigkeiten: Unterstützen von Systemanalyse und -design

- Analysieren der Nutzeranforderungen an das zu erstellende System aus Datenbank-Sicht, ggf. in Zusammenarbeit mit Systemanalytiker und -designer sowie dem Vertriebsbeauftragten als Kundenbetreuer;
- Dokumentieren der so gewonnenen Use Cases;
- Analysieren des vorhandenen Systems;
- Analysieren der Daten und Datensätze;
- Erstellen des Soll-Ist-Vergleichs zwischen Alt-System und Anforderungen;
- Dokumentieren der Daten und Datenformate;
- Erstellen und Testen des Prototypen (als Wegwerfprodukt) bzw. Mitwirken an der Erstellung des Prototyps;
- Abgleichen der Ergebnisse der Analyse mit den Kundenanforderungen, dem Prototypen und den bisherigen Lösungsansätzen;
- Präsentieren des Prototypen beim Kunden bzw. Mitwirken bei der Präsentation;
- Durchführen von Beratungsgesprächen beim Kunden, Vor- und Nachbereiten dieser Gespräche im Team bzw. fachliche Unterstützung dieser Gespräche;
- Mitwirken bei der Überprüfung der Kundenanforderung in Zusammenhang mit den Aufwänden und Ressourcen sowie der Projektdefinition;
- Mitwirken bei Änderungen des Systemdesigns und des Prototyps;
- Mitwirken bei der Verhandlung des Kostenrahmens.

3.1.4.1.2 Kompetenzfelder: Unterstützen von Systemanalyse und -design

Fähigkeiten/Fertigkeiten

- Projektanalyse überprüfen können;
- Nutzeranforderungen analysieren können;
- Daten und Datensätze analysieren können;
- vorhandenes System analysieren können;
- zu Grunde liegendes Datenmodell aus vorhandener Datenbank (auch ohne Dokumentation) erschließen können;
- Soll-Ist-Vergleichs erstellen können;
- Daten und Datenformate dokumentieren können, unter Anwendung von Standards;
- Geschäftsprozesse, Funktionen, komplexe Strukturen und Unternehmensregeln modellieren können;
- Entity-Relationship-Modell erstellen und in ein Datenbankdesign umsetzen können;
- Datenverwendungen definieren können;
- Prototypen aufgrund der Verhandlungsergebnisse und/oder der Projektanalyse erstellen können oder bei der Erstellung mitwirken können;
- Prototypen testen können (sinnvolle Use Cases, insbesondere für Demonstrationszwecke, erstellen bzw. aussuchen können);
- kurzfristig Änderungen und Kundenwünsche in den Prototypen einpflegen können;
- Mitwirken können bei der Überprüfung der Kundenanforderungen, der Schätzung der Aufwände und Ressourcen;
- sinnvoll im Team mitarbeiten können;
- Präsentieren können bzw. bei Präsentationen sinnvoll mitwirken können;

- Kostenverhandlungen unterstützen können.

Wissen

- Systemdesign, Vorgehensweise der Systementwicklung kennen,
- Datenbank- und Systemarchitektur,
- Datenmodelle,
- Datentypen,
- Datenformate,
- Produktkenntnisse Datenbank,
- Geschäftsprozesse,
- Modellierung,
- Entity-Relationship-Modelle,
- Definition von SQL-Objekten,
- Generierung der DB-Tabellen, Tabellen, Indizes,
- Datenbank-Objekte implementieren können, auch: Fähigkeit zur Umsetzung des ersten Anwendungsdesigns,
- Dokumentationsstandards.
- Grundkenntnisse BWL,
- Software-Entwicklung und Software-Engineering,
- Grundkenntnisse Projektmanagement.

Methoden/Werkzeuge

- Designstrategien,
- Modellierungsregeln, Modellierungsverfahren,
- Umsetzung von ER-Modellen in Systemdatenmodelle,
- Case-Tool für das Design des Prototyps und zur Unterstützung der Analyseprozesse,
- ggf. vorhandene Datenbank zur Erstellung von Use Cases sowie zur Analyse der vorhandenen Datenstrukturen, des Datenmodells,
- Planungs- und Schätzsysteme für Software-Projekte,
- kaufmännische Software.

3.1.4.1.3 Beispiel: Unterstützen von Systemanalyse und -design

Nachdem eine erste Analyse des Problems und der Anforderungen der ABN AMRO vorliegen (vgl. 3.1.2 „Das Beispielprojekt: Entwicklung einer Datenbank für die Vereinfachung der Geschäftsprozesse einer international tätigen Bank“), wird der Kundenbetreuer von Oracle aufgefordert, ein einschlägiges Angebot abzugeben: In zwei Gesprächen werden die Anforderungen des Kunden spezifiziert und ein erster Lösungsansatz entwickelt. Dazu gehören ein öffentliches Portal mit Links zu Internet- und Intranet-Anwendungen, bei dem aber der Zugriff über eine personalisierte Website mit nutzerspezifischen Daten und Informationen (Bookmarks, gespeicherte Formulare, etc.) erfolgt. Der Import der Dokumente wird mit dem iFS (Internet File System) von Oracle realisiert, das Drucken der im Browser ausgefüllten Formulare erfolgt über Postscript- oder pdf-Dateien. Das Nutzungs- bzw. Sicherheitskonzept wird durch die Definition unterschiedlicher Nutzerrollen realisiert. Eine Archivierungsfunktion für alte Dokumentversionen ist ebenfalls notwendig.

Da bei der Bank kein System vorhanden ist, geht es in der Analysephase lediglich um die Analyse der vorhandenen Formulare und der in ihnen enthaltenen Datenfelder. Diese Analyse ist recht aufwändig, da zum einen die vorhandene Vorstudie nicht alle Formulare umfasst, zum anderen ein großer Teil der noch zu analysierenden Dokumente nur in gedruckter Form bzw. als doc-Datei vorliegt. In dieser Phase werden also die Formulare beschrieben, die in

ihnen enthaltenen Daten kategorisiert und gleiche Datenfelder auf unterschiedlichen Formularen identifiziert.

Um den Kunden davon zu überzeugen, dass die von Oracle vorgeschlagene Lösung leistungsfähig ist und den Anforderungen gerecht wird, wird ein Prototyp gebaut. Im vorliegenden Fall sollte damit zum einen bewiesen werden, dass es möglich ist, die Formulare auf dem Bildschirm (im Browser) auszufüllen, dass sich das System komfortabel und einfach bedienen lässt und dass das Erstellen und Einstellen neuer Formulare unproblematisch ist. Für die letzte Anforderung wurde auf das iFS (Internet File System) von Oracle zurückgegriffen, welches dem Nutzer eine ähnliche Oberfläche und vor allem ähnliche Funktionalitäten wie der Windows Explorer zur Verfügung stellt, so dass Löschen, Kopieren, Verschieben und Einfügen neuer Dokumente sehr komfortabel möglich sind.

Bei dem Prototypen handelte es sich um eine für die Präsentation erstellte Lösung, durch die der Kunde schnell überzeugt wurde.

3.1.4.2 Mitwirken bei der Festlegung des Entwicklungsrahmens

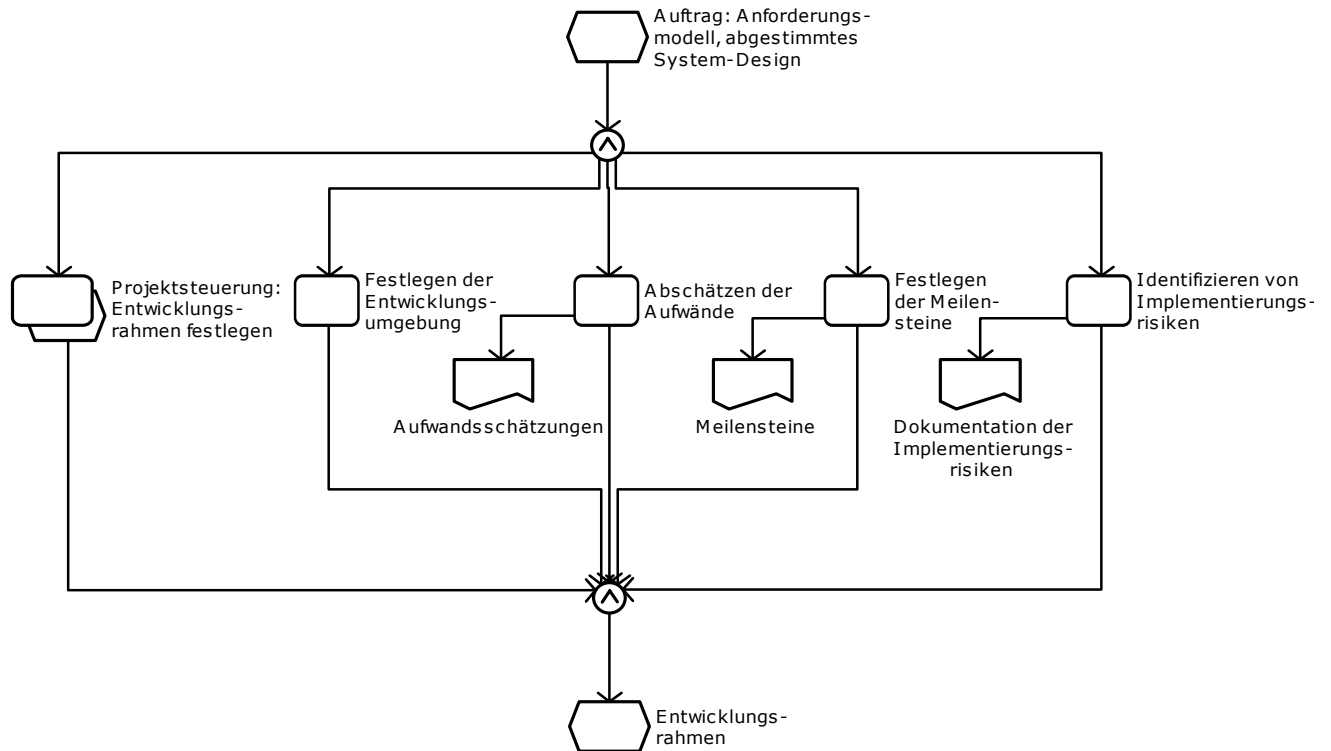


Abbildung 2: Mitwirken bei der Festlegung des Entwicklungsrahmens

3.1.4.2.1 Tätigkeiten: Mitwirken bei der Festlegung des Entwicklungsrahmens

Der Entwicklungsrahmen für ein Datenbankentwicklungsprojekt wird in der Regel von dem zuständigen IT-Projektmanager festgelegt. Dazu gehören das Erstellen der Projektdefinition, einschließlich Bestimmen des Projekttyps, Abschätzen des Projektrisikos, Festlegen der Projektgröße und Beschreibung des Projektumfeldes sowie die Kalkulation der Aufwände und Ressourcen. Durch folgende Tätigkeiten kann der Database Developer an der Festlegung des Entwicklungsrahmens mitwirken:

- Festlegen der Entwicklungsumgebung, einschließlich der daraus folgenden Aufwände;
- Abschätzen der zu erbringenden Aufwände und der Dauer des Projekts im Team;
- Festlegen sinnvoller Meilensteine sowie Zeit- und Aufwandsplanung für diese Meilensteine im Team
- Identifizieren von Implementierungsrisiken (z. B. Performanceproblemen) im Team.

3.1.4.2.2 Kompetenzfelder: Mitwirken bei der Festlegung des Entwicklungsrahmens

Fähigkeiten/Fertigkeiten

- Entwicklungsumgebungen in Bezug auf die Projektaufgaben beurteilen können;
- Aufwände, modul- und projektbezogen, schätzen können;
- sinnvolle Meilensteine festlegen können;
- Aufwände und Ressourcen planen können, Zeitpläne erstellen können;
- zielgerichtete Gespräche im Team führen können;
- bei der Erstellung von Projektdefinitionen mitwirken können;
- bei der Kalkulation von Aufwänden und Ressourcen mitwirken können

- Implementierungsrisiken, die typisch für Datenbankentwicklungen sind, identifizieren können.

Wissen

- Implementierungsrisiken und ihre Vermeidung,
- Grundkenntnisse Projektmanagement in der Software-Entwicklung,
- Grundkenntnisse BWL,
- Kenntnisse Software-Entwicklung und Software- Engineering.

Methoden/Werkzeuge

- Schätzmethoden,
- Planungs- und Schätzsysteme für Software-Projekte,
- kaufmännische Software.

3.1.4.2.3 Beispiel: Mitwirken bei der Festlegung des Entwicklungsrahmens

Nachdem die Anforderungen geklärt sind und ein erster Lösungsansatz - auf Basis von Oracle-Produkten - entwickelt ist, werden zunächst die möglichen Risiken identifiziert. Diese liegen insbesondere in der Vielzahl und Verschiedenheit der zu berücksichtigenden Formulare. Eine Absicherung gegen die Risiken seitens Oracle erfolgte durch die Formulierung entsprechender Annahmen und Absprachen bereits im Angebot bzw. in der Feinspezifikation. Danach werden die Projektdefinition und der Kostenrahmen erstellt, um die Aufwände und Ressourcen kalkulieren zu können. Als Tool verwendet Oracle hierfür den PBM (Project Bridge Modeler) der Firma Applied Business Technology (ABT). Projektdefinition und Kostenrahmen bilden auch die Grundlage für das Projektmanagement, welches bei Oracle mit Hilfe der Project Management Method (PJM) durchgeführt wird. Die Ergebnisse des PBM, insbesondere der Projektplan, werden in das Tool PMW (Project Manager Workbench) eingespeist, welches dann im weiteren Verlauf die aktuellen Daten zur laufenden Planaktualisierung und -qualifizierung liefert.

Zur Festlegung des Entwicklungsrahmens beim Projekt „eForms“ gehörten auch Abmachungen zum Qualitäts- und Konfigurationsmanagement sowie ein festgelegtes Eskalationsverfahren bei Problemen und ein formaler Rahmen für Projektberichte und Reviews.

3.1.4.3 Überprüfen des Systemdesigns

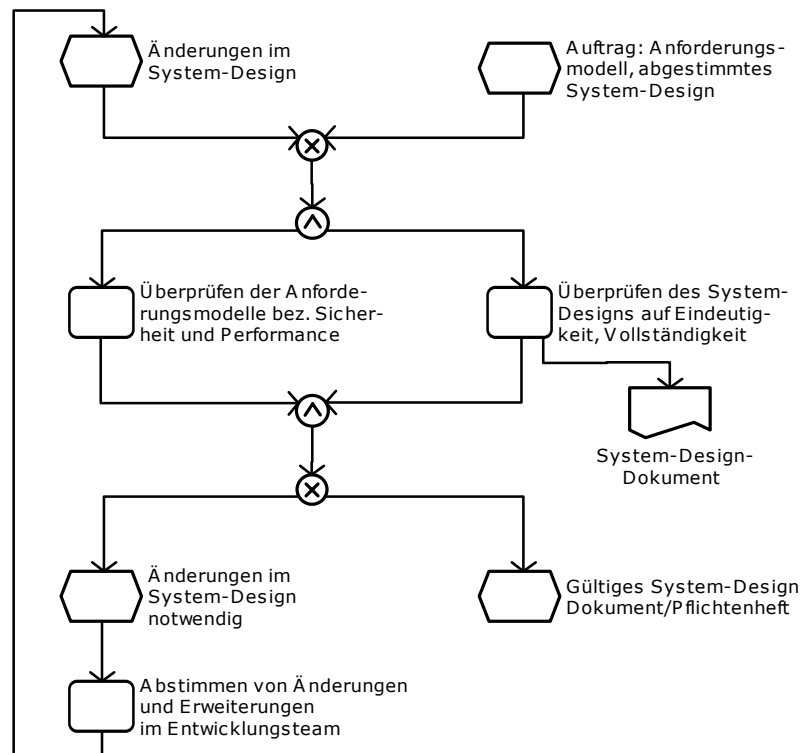


Abbildung 3: Überprüfen des Systemdesigns

3.1.4.3.1 Tätigkeiten: Überprüfen des Systemdesigns

- Überprüfen der Anforderungsmodelle bezüglich Sicherheit und Performance der geplanten Datenbank;
- Überprüfen des System-Designs auf Korrektheit, Eindeutigkeit und Vollständigkeit;
- Falls notwendig: Abstimmen von Änderungen oder Erweiterungen im Entwicklungsteam und mit dem IT Systems Analyst oder IT Systems Developer.

3.1.4.3.2 Kompetenzfelder: Überprüfen des Systemdesigns

Fähigkeiten/Fertigkeiten

- Anforderungsmodelle, System-Design-Dokumente und Pflichtenhefte verstehen können;
- Anforderungsmodelle, System-Design-Dokumente und Pflichtenhefte auf Korrektheit, Eindeutigkeit und Vollständigkeit prüfen können;
- Auswirkungen von Vorgaben auf Sicherheit und Performance der Datenbank abschätzen können;
- Pflichtenheft bzw. System-Design-Dokumente für den Bereich Datenbankentwicklung erstellen können;
- Gespräche im Team führen können, dabei in der Lage sein, die eigene (fachliche) Meinung auch in Konfliktsituationen angemessen zu vertreten.

Wissen

- Software-Entwicklung und Software-Engineering,
- Einflüsse auf Performance von Datenbanken,
- sicherheitsrelevante Aspekte von Datenbanken,

- Kenntnisse Datenbankentwicklungsprozess.

Methoden/Werkzeuge

- Planungs- und Schätzsysteme für Software-Projekte,
- Case Tool (für Datenbank-Design).

3.1.4.3.3 Beispiel: Überprüfen des Systemdesigns

Die Projektbeschreibung von „eForms“ fasst die fachlichen Anforderungen folgendermaßen zusammen: „Bei ABN AMRO soll ein System zur elektronischen Verwaltung von Formularen aufgebaut werden. Dabei sollen sowohl Standardformulare des Bankenvereins (BV), als auch eigens erstellte Formulare abgebildet, elektronisch erfaßt und verwaltet werden. Die Formulare sollen auf dem Bildschirm ausfüllbar sein und ausgedruckt werden können. Die gesamte Aufgabenstellung wird in mehrere Leistungsstufen aufgeteilt, die aufeinander aufbauen:

1. Aufbau eines Systems zur Verwaltung elektronischer Formulare. In dieser Leistungsstufe wird eine Schnittstelle zur Verfügung gestellt, die es erlaubt, die vom BV in elektronischer Form (Postscript) zur Verfügung gestellten Formulare in das neue System zu übernehmen. Über diese Schnittstelle werden weiterhin die internen Formulare des German Instruction Manuals (GIM) in das neue System integriert. [...]“

Bei diesem Projekt war der Database Developer von Anfang an eingebunden. Vor der Beauftragung von Oracle hatte die ABN AMRO bereits eine Studie zu den fachlichen Anforderungen erstellen lassen. Diese beinhaltete Angaben zur technischen Infrastruktur bei der ABN AMRO und zu den in die Datenbank aufzunehmenden Formularen. Die erste Aufgabe des Database Developers war also die Überprüfung dieser Studie auf Plausibilität und Machbarkeit. Daraus wurde auch die Anforderungen an die Sicherheit - die durch das geplante Rollenmodell leicht zu erfüllen waren - und an die Performance abgeleitet, die durch die Verwendung entsprechender Oracle-Produkte als leicht erfüllbar deklariert wurden.

Das Systemdesign wurde in diesem Projekt durch den Database Developer selbst erstellt und beinhaltete die Anforderungen an die Design-Standards und das Datenmodell, die Spezifikation der Schnittstelle zum Laden der Rohdaten, das Datenzugriffs- und Sicherheitskonzept und das Funktionsmodell.

3.1.4.4 Einbeziehen und Abgleichen von Standards

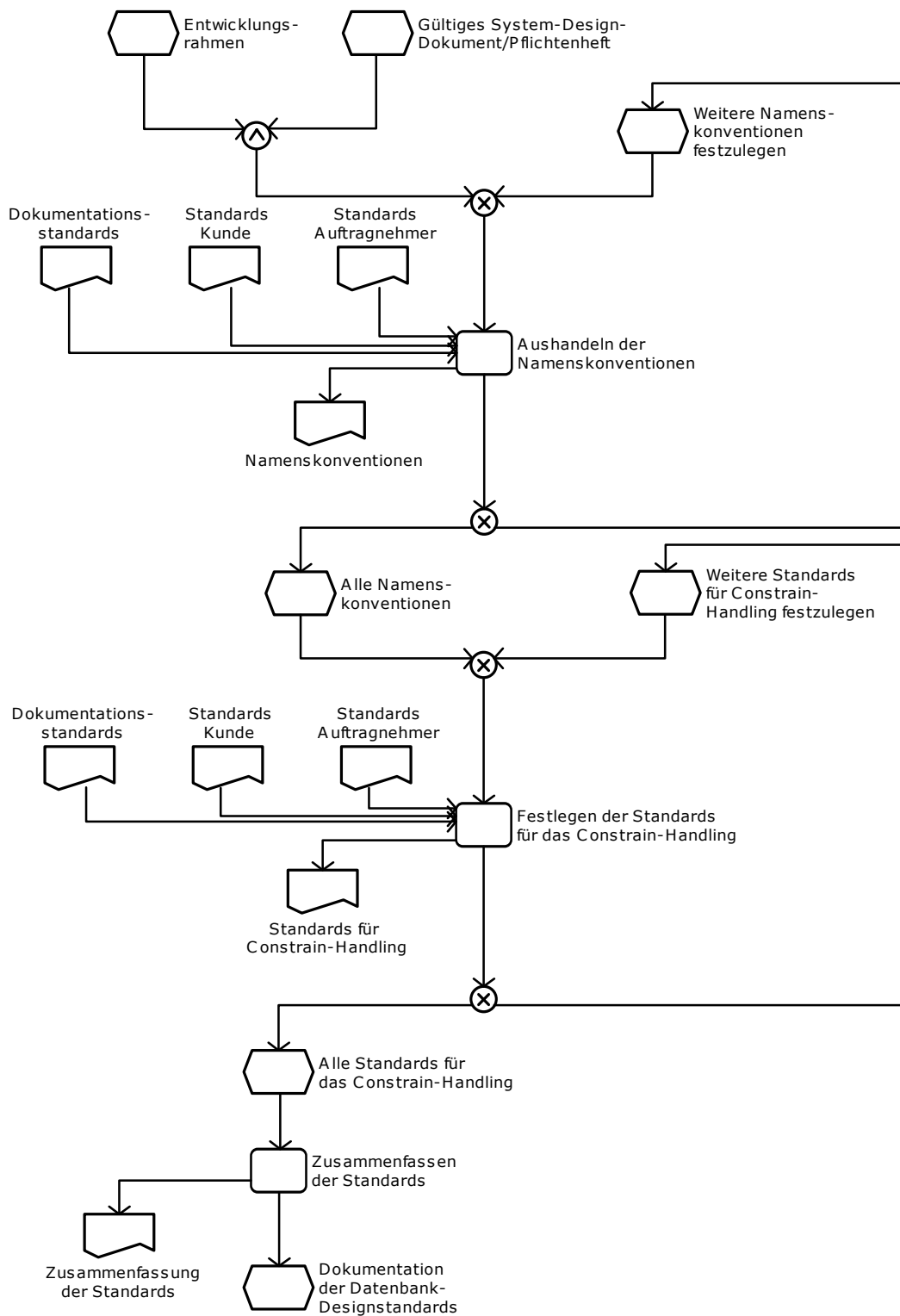


Abbildung 4: Einbeziehen von Standards

3.1.4.4.1 Tätigkeiten: Einbeziehen und Abgleichen von Standards

- Aushandeln der Namenskonventionen;
- Festlegen der Standards für das Constrain-Handling.

3.1.4.4.2 Kompetenzfelder: Einbeziehen und Abgleichen von Standards

Fähigkeiten/Fertigkeiten

- Sinnvoll und angemessen mit Kunden und im Team verhandeln können;
- Standards sinnvoll anwenden können;
- logisch einwandfreie und einheitliche Vorschläge für Konventionen machen können.

Wissen

- Namenskonventionen,
- Constrain-Handling,
- Standards und Standardsetzung.

Methoden/Werkzeuge

- CASE-Tool (für Datenbankdesign).

3.1.4.4.3 Beispiel: Einbeziehen und Abgleichen von Standards

Gerade bei sehr anwendernahen Applikationen wie „eForms“ ist es wichtig, dass Namenskonventionen nicht nur einheitlich, sondern auch für den Kunden verständlich und nachvollziehbar sind. Im speziellen Falle war es so, dass selbst die elektronisch angelieferten Formulare keiner völlig einheitlichen Nomenklatur folgten. Daher konnte und musste eine neue Namensgebung ausgehandelt werden. Da das Anlegen und Einstellen neuer Formulare auch durch den Kunden erfolgen sollte, musste auch auf der Ebene der Feldnamen (beim Anlegen neuer Formulare spielen diese eine entscheidende Rolle) eine sinnvolle Konvention gefunden werden. Dies geschah in einem Aushandlungsprozess zwischen dem Database Developer und den Betroffenen bei der Bank, unter Berücksichtigung des bereits vorhandenen Materials und der bei der Bank gepflegten Standards. Das Aushandeln und Festlegen der Standards dauerte bei diesem Projekt deutlich länger als zunächst geplant, da sich erst in dieser Phase selbst herausstellte, dass die zu erfassenden Formulare noch viel heterogener waren, als zunächst angenommen.

3.1.4.5 Verfeinern der Systementwürfe

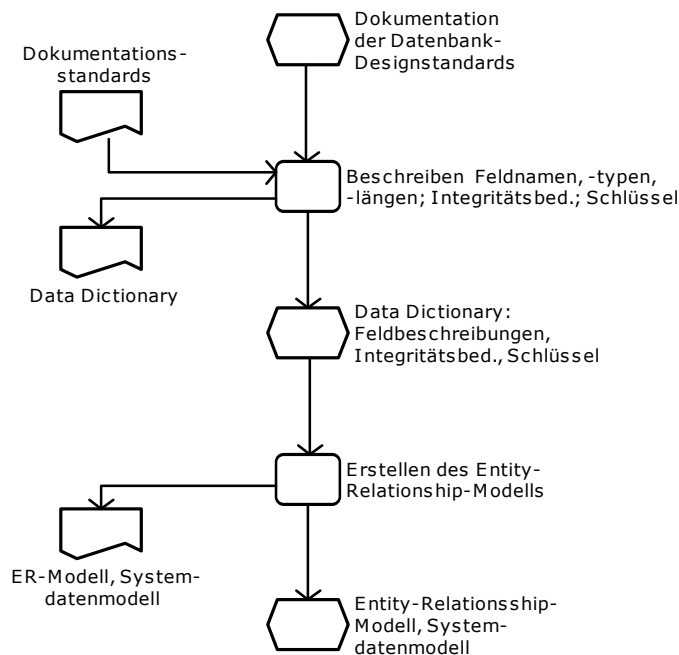


Abbildung 5: Verfeinern der Systementwürfe

3.1.4.5.1 Tätigkeiten: Verfeinern der Systementwürfe

- Beschreiben der Feldnamen, - typen, -längen, -vorkommen, der Integritätsbedingungen, der Schlüssel;
- Erstellen des Entity-Relationship-Modells⁶ und Beschreiben der Nebenbedingungen, die nicht in den Datenbank-Regeln abgebildet werden können;
- Erzeugen des Systemdatenmodells.

3.1.4.5.2 Kompetenzfelder: Verfeinern der Systementwürfe

Fähigkeiten/Fertigkeiten

- Geschäftsprozesse modellieren können;
- Feldnamen, Feldtypen, Feldlängen beschreiben und festlegen können;
- Integritätsbedingungen festlegen können;
- Primär- und Sekundärschlüssel identifizieren und festlegen können;
- Entity-Relationship-Modell erstellen können (nutzeradäquat);
- Datenbank-Objekte implementieren können, auch: Fähigkeit zur Umsetzung des Funktionsmodells in ein erstes Anwendungsdesign.

Wissen

- Datenmodelle,
- Normalisierung, Datenintegrität, Denormalisierung,
- Entity-Relationship-Modelle, einschließlich Querverweise zwischen den Modellen,
- Definition von SQL-Objekten,

⁶ vgl. Anmerkung 5: Auch „Entity-Relationship-Modell“ steht hier nur als eine - allerdings sehr häufige - Möglichkeit der Datenbankmodellierung.

- Generierung der DB-Tabellen, Indizes,
- Datentypen,
- Datenbankarchitektur (allgemein und produktspezifisch),
- Anlegen einer Datenbank, Verwaltung von DB-Strukturen,
- Committed Data,
- Log Data.

Methoden/Werkzeuge

- Designstrategien, auch Verfeinern und Verteilen von Designs,
- Umsetzung von ER-Modellen in Systemdatenmodelle,
- Wiedergewinnung eines Designs (Capture Design) und Round Trip Engineering,
- CASE-Tool (für Modellierung und Erstellung des Systemdatenmodells),
- Administrationstools des Datenbankmanagementsystems.

3.1.4.5.3 Beispiel: Verfeinern der Systementwürfe

Nach der Festlegung der Namenskonventionen konnten die Felder endgültig festgelegt und das Systemdatenmodell erstellt werden. Im Projekt „eForms“ wurde die Erstellung eines Entity-Relationship-Modells übersprungen und gleich das Systemdatenmodell erstellt. Dieses enthielt die notwendigen Tabellen, die Spaltendefinitionen, die Primär- und Sekundärschlüssel sowie die Beziehungen zwischen den Tabellen.

3.1.4.6 Erstellen des Migrationsmodells

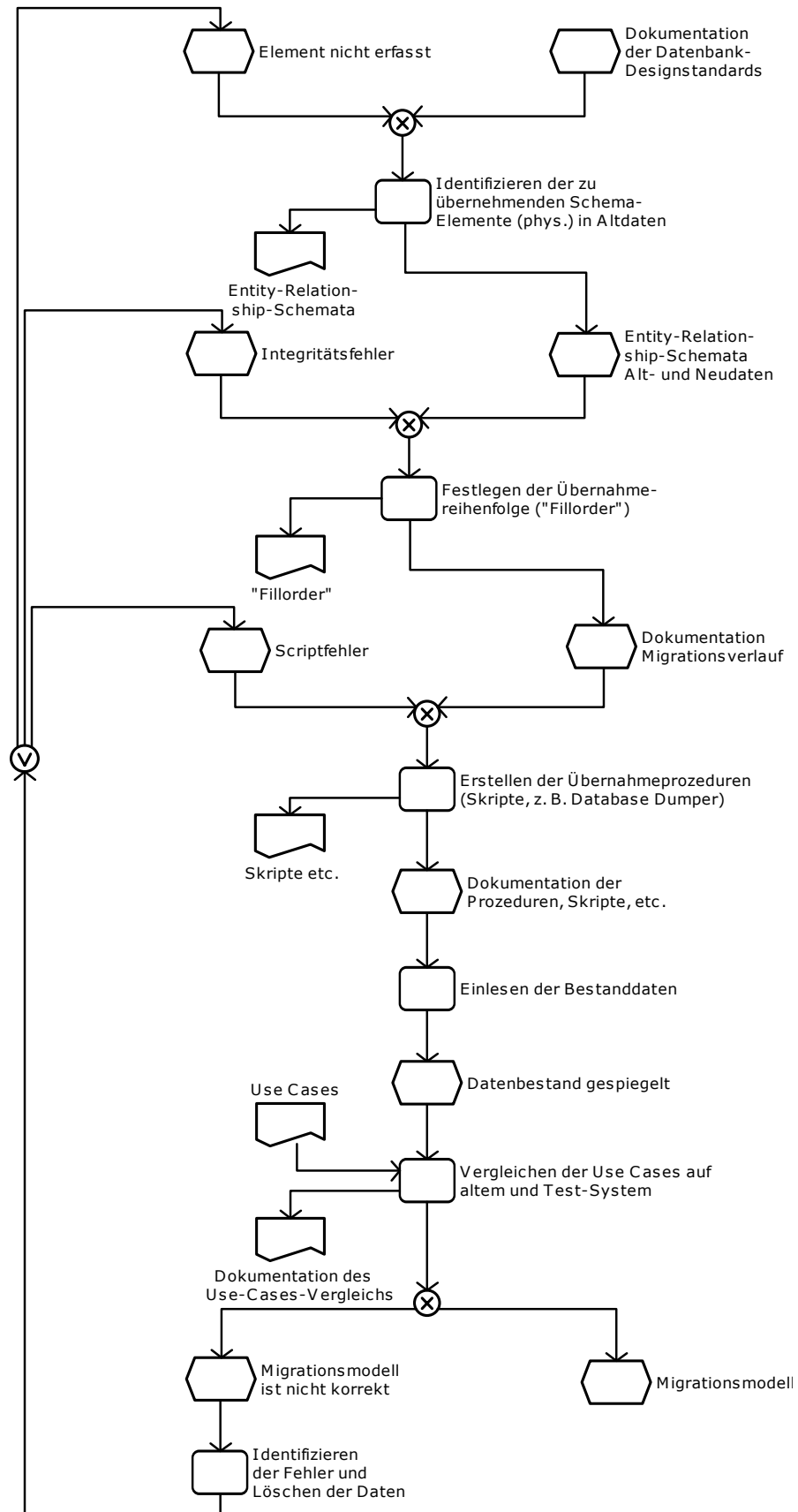


Abbildung 6: Erstellen des Migrationsmodells

Anmerkung: Dieser Migrationsprozess klammert die Datenübernahme aus komplexen, laufenden Systemen bewusst aus. Dazu werden Migrationsspezialisten benötigt. Der hier dargestellte Prozess ist relativ einfach.

3.1.4.6.1 Tätigkeiten: Erstellen des Migrationsmodells

- Identifizieren der zu übernehmenden Schema-Elemente im vorhandenen Datenbestand;
- Festlegen der Übernahmereihenfolge („fillorder“);
- Erstellen von Übernahme-prozeduren (Scripte, z. B. Database Dumper);
- Einlesen der Altdaten (oder einer entsprechenden Auswahl) in das Datenbankmodell des neuen Systems/den Prototypen/die erste Version der Datenbank;
- Vergleichen des Verhaltens von Use Cases im alten und neuen System;
- Identifizieren von Fehlern;
- (ggf.) vollständiges Löschen der Daten im neuen System.

3.1.4.6.2 Kompetenzfelder: Erstellen des Migrationsmodells

Fähigkeiten/Fertigkeiten

- Zugrunde liegendes Datenmodell aus vorhandener Datenbank (auch ohne Dokumentation) erschließen können;
- Daten- und Datenbankmodelle (zusätzlich: alle Fähigkeiten/Fertigkeiten aus 3.1.4.5.2 „Kompetenzfelder: Verfeinern der Systementwürfe“);
- Scripte erstellen können;
- Daten einlesen (lassen) können;
- Testplan erstellen können;
- sinnvolle (Test) Use Cases erstellen können, systematisch vergleichen können;
- Fehler lokalisieren und identifizieren können;
- ggf. mit Testumgebung umgehen können.

Wissen

- Datenmodelle,
- Datenbank- und Systemarchitektur,
- wie 3.1.4.5.2 „Kompetenzfelder: Verfeinern der Systementwürfe“, aber vertiefte Kenntnisse.

Methoden/Werkzeuge

- Datenbanken-Setup und Backup (altes und neues System),
- Datenbanken-Test und -Vergleich mit sinnvollen Use Cases,
- CASE-Tool (für Modellierung),
- (Test) Datenbank.

3.1.4.6.3 Beispiel: Erstellen des Migrationsmodells

Da es bei der ABN AMRO vor dem Projekt keine Datenbank gab, sondern die Formulare von Hand ausgefüllt wurden, war auch kein Datenbestand zu migrieren.

3.1.4.7 Analysieren der Datenbankzugriffe und Erstellen des Mengengerüsts

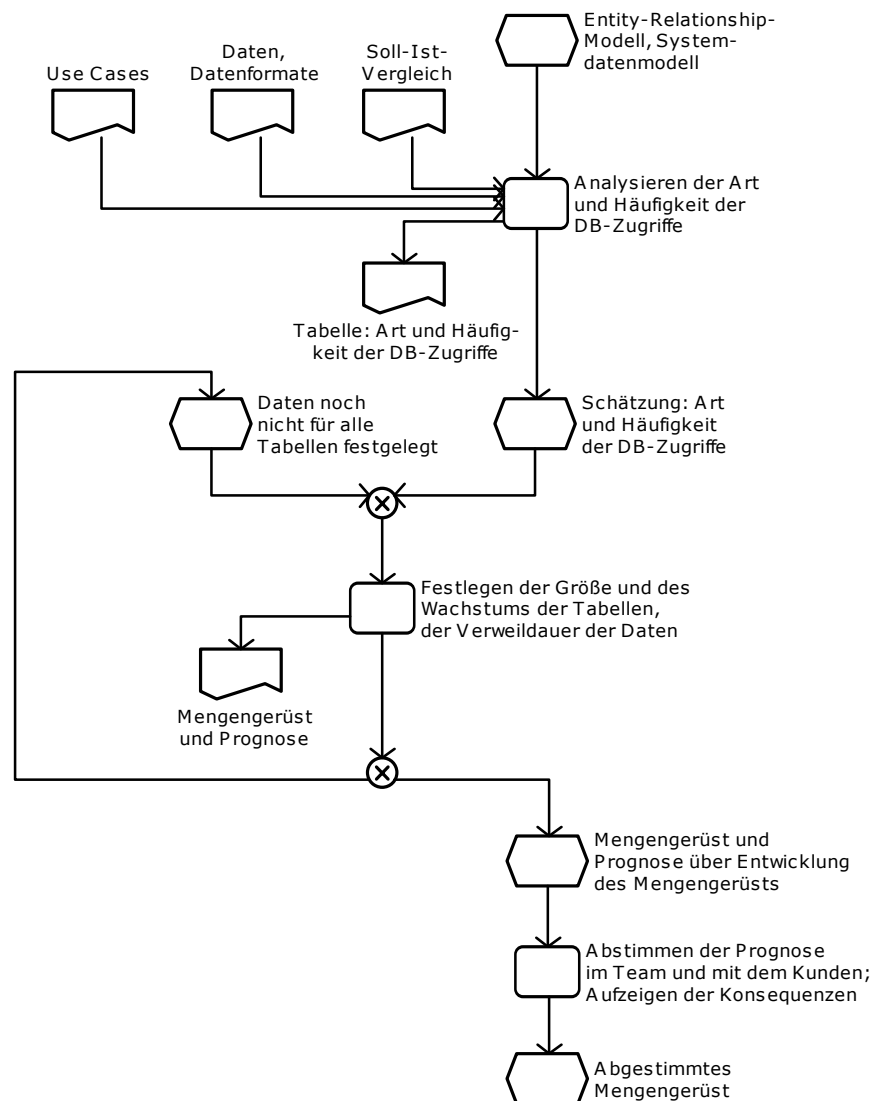


Abbildung 7: Analysieren der Datenbankzugriffe und Erstellen des Mengengerüsts

3.1.4.7.1 Tätigkeiten: Analysieren der Datenbankzugriffe und Erstellen des Mengengerüsts

- Analysieren der Art und Häufigkeit der Datenbank-Zugriffe;
- Festlegen der Größe und des (voraussichtlichen) Wachstums der Tabellen sowie der Verweildauer der Daten;
- Erstellen von Mengengerüst und Kapazitätsplan;
- Abstimmen der Prognose zu Mengengerüst und Kapazitätsplan im Team;
- Abstimmen der Prognose zu Mengengerüst und Kapazitätsplan mit dem Kunden, Aufzeigen von Konsequenzen, gegebenenfalls sinnvollen Kompromiss vorschlagen.

3.1.4.7.2 Kompetenzfelder: Analysieren der Datenbankzugriffe und Erstellen des Mengengerüsts

Fähigkeiten/Fertigkeiten

- Art und Häufigkeit der Datenbankzugriffe beobachten und analysieren können (ggf. mit geeigneten Tools);
- begründete Prognosen über die Veränderungen der Zugriffe durch die Systemänderung abgeben können;

- Voraussichtliche Größe und Wachstum der Tabellen (auch in der Zusammenschau) abschätzen können;
- Mengengerüst und Kapazitätsplan erstellen können;
- begründete Abschätzungen für Datenmengen erstellen und kommunizieren können;
- Konsequenzen aus Mengengerüst und Kapazitätsplan für Datenbank, Datenbankmodell und Hardwareausstattung ableiten und kommunizieren können;
- Konsequenzen und Interdependenzen für und mit Datenmodellen beachten (vgl. 3.1.4.5.2 „Kompetenzfelder: Verfeinern der Systementwürfe“);
- sinnvoll im Team und mit Kunden kommunizieren und diskutieren können;
- Abhängigkeiten zwischen Voraussetzungen, Prognosen und Kosten deutlich machen können;
- Konflikte aushalten und sinnvolle Kompromisse schließen können.

Wissen

- Datenstrukturen,
- Committed Data,
- Log Data,
- Datenbankarchitektur (allgemein und produktspezifisch),
- Anlegen einer Datenbank, Verwaltung von DB-Strukturen,
- Normalisierung, Datenintegrität, Denormalisierung,
- Entity-Relationship-Modelle, einschließlich Querverweisen,
- vertiefte Produktkenntnisse (inklusive Zugriffszeiten).

Methoden/Werkzeuge

- Designstrategien,
- Performancestrategien,
- CASE-Tool (für Analyse und vor allem für Aufstellen des Mengengerüsts),
- gegebenenfalls Tools zur Beobachtung und Analyse der Datenzugriffe,
- (Test) Datenbank.

3.1.4.7.3 Beispiel: Analysieren der Datenbankzugriffe und Erstellen des Mengengerüsts

Beim Aufbau der Datenbank für „eForms“ ging es vor allem darum, abzuschätzen, welche Formulare und welche Formularfelder wie oft benötigt werden und dann die entsprechenden Kapazitäten bereitzustellen. Außerdem werden - aufgrund der gesetzlichen Vorschriften - sämtliche Rundschreiben, Handbücher und alten Formulare gespeichert und somit aufbewahrt. Auch hierfür müssen die entsprechenden Kapazitäten vorhanden sein. Dieses Wissen ließ sich bei den Mitarbeiterinnen und Mitarbeitern der Bank relativ leicht erfragen und es gab auch keine Prognoseschwierigkeiten.

3.1.4.8 Ableiten der Testfälle

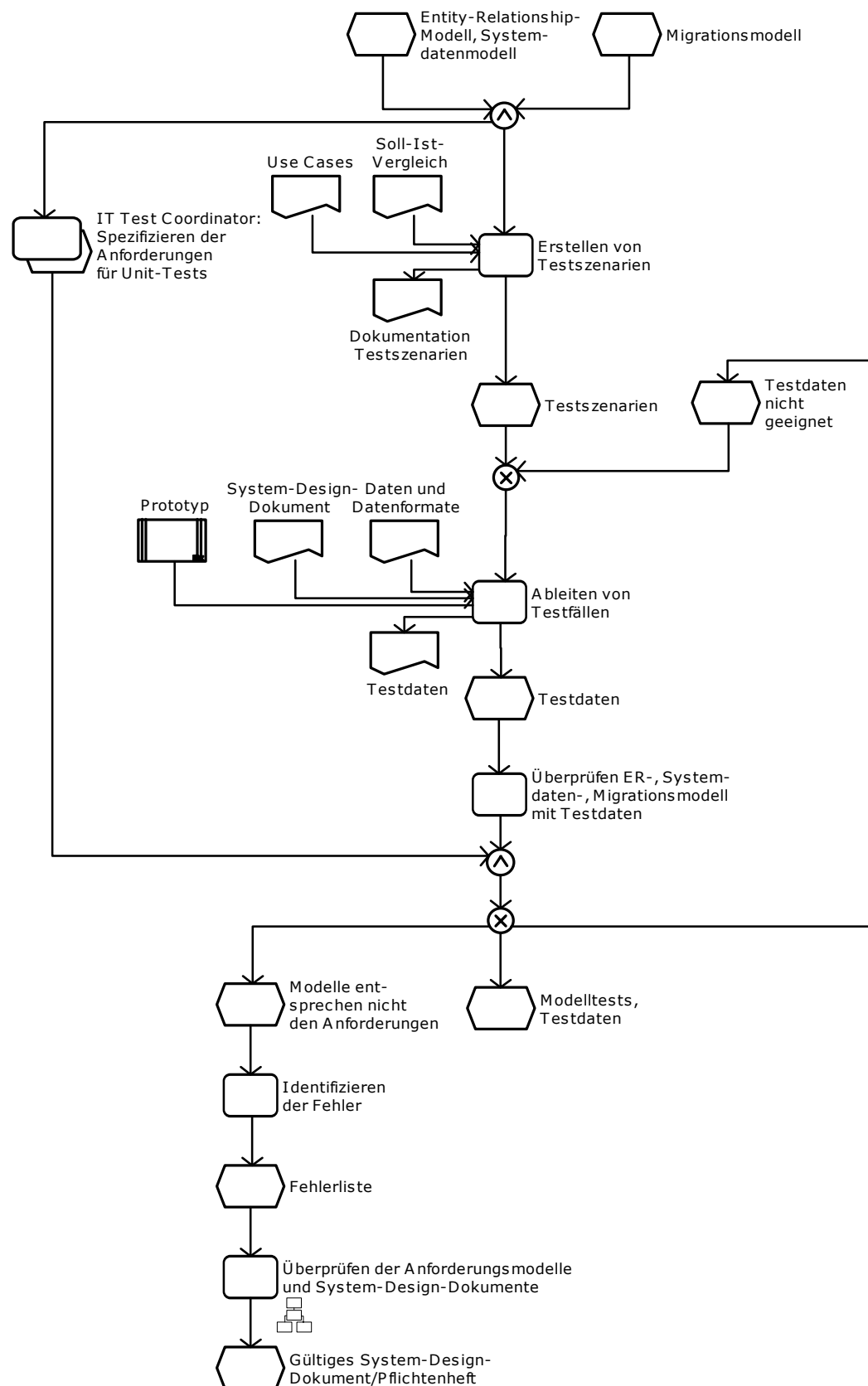


Abbildung 8: Ableiten der Testfälle

3.1.4.8.1 Tätigkeiten: Ableiten der Testfälle

- Erstellen von Testszenarien aus den Use Cases und dem Soll-Ist-Vergleich;
- Ableiten von Testfällen aus den Testszenarien unter Berücksichtigung des Prototypen, der System-Design-Dokumente und der Daten und Datenformate;
- Überprüfen des Entity-Relationship- bzw. Systemdatenmodells und des Migrationsmodells mit den Testdaten;
- Identifizieren der Fehler, wenn die Überprüfung nicht der Spezifikation entspricht;
- enge Zusammenarbeit mit dem IT Test Coordinator bei diesen Tätigkeiten;
- Mit der Fehlerliste muss dann ein früherer Teilprozess, das Überprüfen der System-Design-Dokumente, wieder aufgenommen werden.

3.1.4.8.2 Kompetenzfelder: Ableiten der Testfälle

Fähigkeiten/Fertigkeiten

- Sinnvolle Testszenarien erstellen können, die Schwachstellen der Module aufzeigen können und die Praxis berücksichtigen;
- Kriterien für Black- und White-Box-Tests der Module festlegen können;
- Kriterien für funktionierende Schnittstellen zwischen den Modulen festlegen können;
- Bedingungen für funktionierende Modul Umgebung festlegen können;
- Modelle angemessen überprüfen können;
- Testszenarien, Testdaten und Fehler sinnvoll dokumentieren können;
- mit dem IT Test Coordinator und im Team kooperieren können.

Wissen

- Datenstrukturen,
- Datenbankarchitektur (allgemein und produktspezifisch),
- Schnittstellendefinitionen.

Methoden/Werkzeuge

- Verifikationstechniken für Algorithmen,
- Kenntnisse statistischer Programmanalyse, insbesondere Struktur- und Datenflussanalyse,
- White- und Black-Box-Tests, Testmethoden,
- Case-Tool,
- Testdatenbank,
- Prototyp,
- Testumgebung.

3.1.4.8.3 Beispiel: Ableiten der Testfälle

Im Rahmen der Realisierungsphase von „eForms“ wurden auch Testpläne für die Module und ihre Integration festgelegt. Da es sich bei den neu zu erstellenden Modulen in erster Linie um Schnittstellen für die Ein- und Ausgabe der Daten in die Datenbank handelte, waren die Testspezifikationen und -fälle leicht festzulegen: Die vorhandenen Formulare mussten in die Datenbank eingegeben, in exakter Analogie auf den Bildschirmen dargestellt und ausgedruckt werden können.

3.1.4.9 Erstellen des physischen Datenbankdesigns

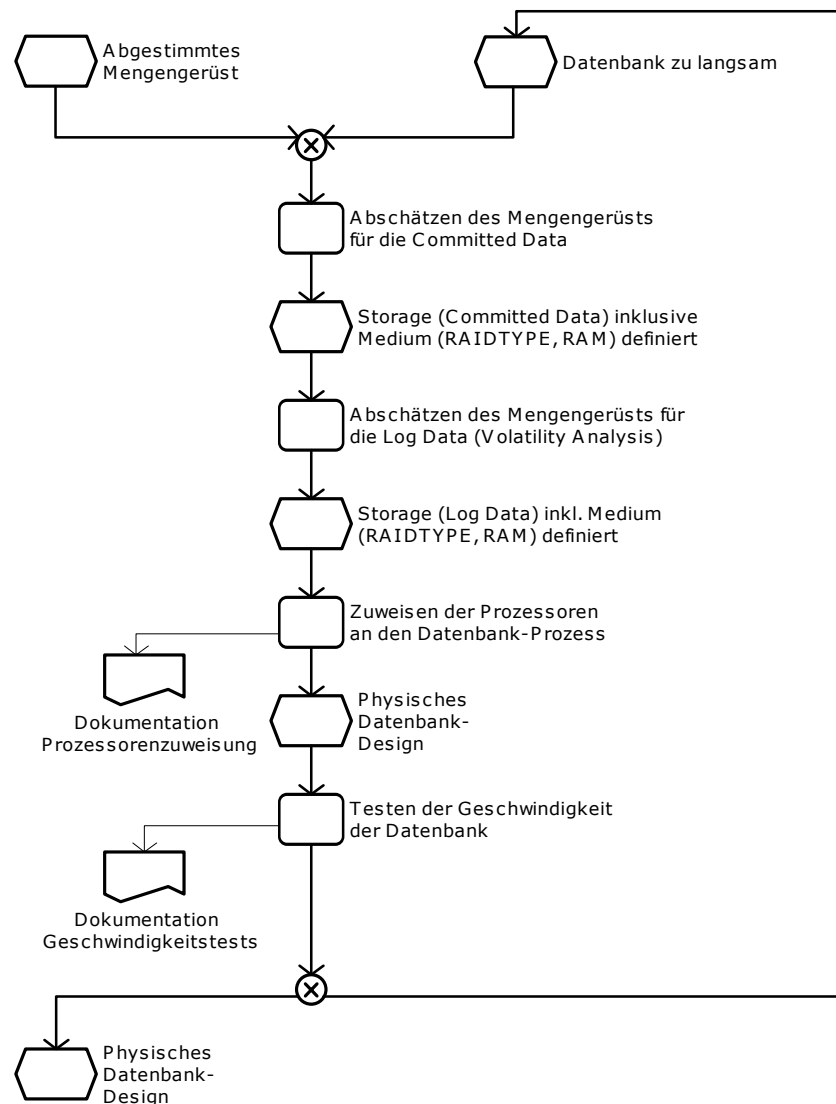


Abbildung 9: Erstellen des physikalischen Datenbank-Designs

3.1.4.9.1 Tätigkeiten: Erstellen des physischen Datenbankdesigns

- Abschätzen des Mengengerüsts für die Committed Data;
- Storage (Committed Data), inklusive des benötigten Mediums definieren;
- Abschätzen des Mengengerüsts für die Log Data (Volatility Analysis);
- Storage (Log Data), inklusive des benötigten Mediums definieren;
- Zuweisen der Prozessoren an den Datenbank-Prozess;
- Testen der Datenbankgeschwindigkeit;
- Schreiben der Installationsskripte.

3.1.4.9.2 Kompetenzfelder: Erstellen des physischen Datenbankdesigns

Fähigkeiten/Fertigkeiten

- Mengengerüste für Committed und Log Data abschätzen können, bezogen auf die konkrete, zu installierende Datenbank;

- Storage (einschließlich Medium) definieren können, bezogen auf die konkrete, zu installierende Datenbank;
- Prozessoren zuweisen können;
- Datenbankgeschwindigkeit testen können (festlegen sinnvoller Testdaten/Use Cases, Test durchführen können);
- Skripte schreiben und dokumentieren können.

Wissen

- Committed Data,
- Log Data,
- Anlegen einer Datenbank, Verwaltung von DB-Strukturen,
- Systemdatenmodelle,
- vertiefte Produktkenntnisse,
- Prozessoren,
- Systemkenntnisse (Betriebssystem),
- vgl. auch 3.1.4.6.2 „Kompetenzfelder: Erstellen des Migrationsmodells“.

Methoden/Werkzeuge

- Performancestrategien,
- CASE-Tool (für Datenbankdesign),
- (Test) Datenbank,
- Script-Editor o.ä.

3.1.4.9.3 Beispiel: Erstellen des physischen Datenbankdesigns

Im konkreten Fall der von Oracle erstellten (statischen) Datenbank WebDB gestaltete sich der Prozess des physischen DB-Designs folgendermaßen:

- Festlegen der Storage-Parameter der DB-Objekte
- Festlegen der Tablespace
- Festlegen der Datafiles
- Festlegen der Controlfiles
- Festlegen der Rollbacksegmente und Redolog-Files
- Festlegen der temporären Segmente
- Beschreiben aller Parameter mit Hilfe von SQL, dies entspricht dem Erstellen der Installationsskripte.

3.1.4.10 Einrichten der Entwicklungsumgebung beim Kunden

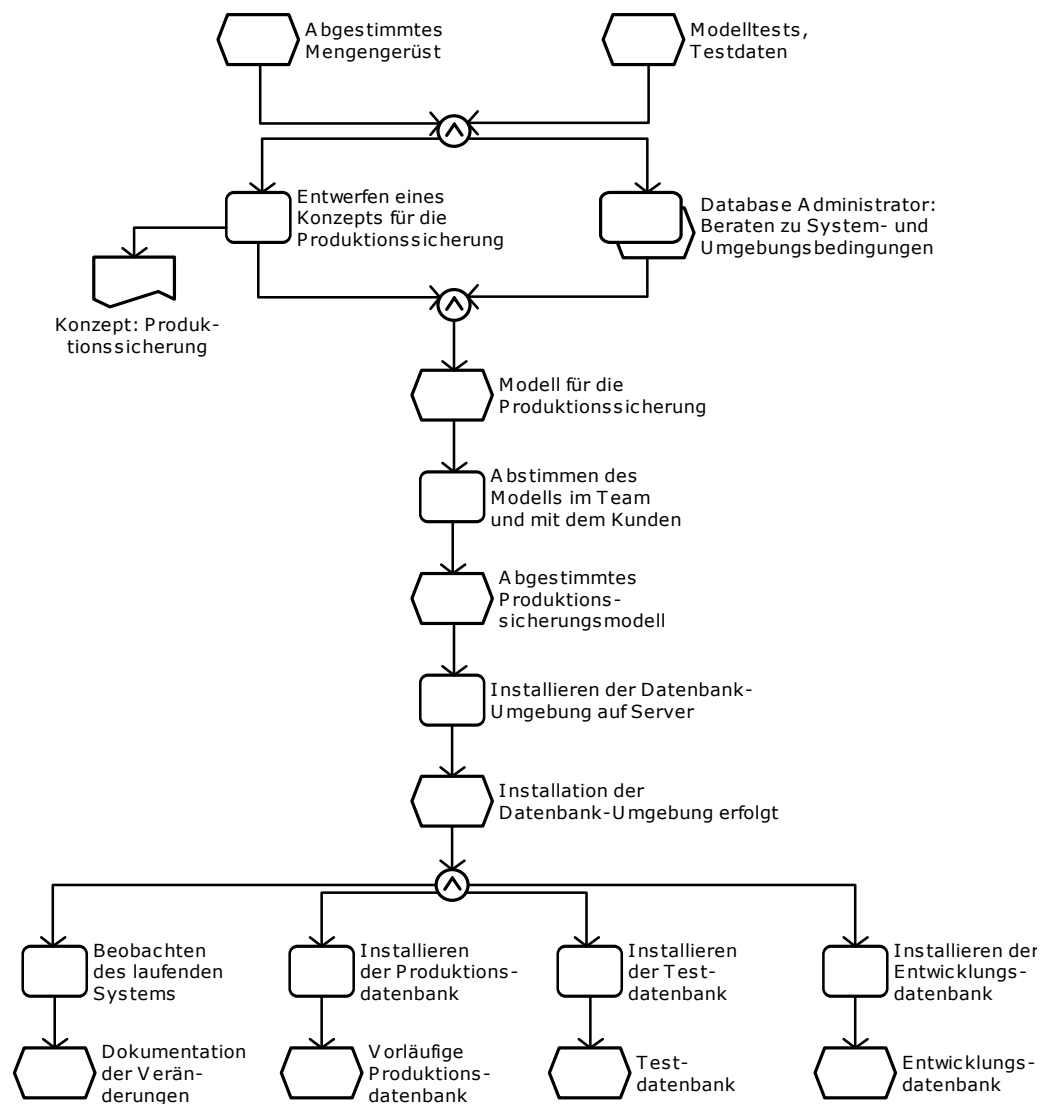


Abbildung 10: Einrichten der Entwicklungsumgebung

Anmerkungen:

Der Teilprozess „Einrichten der Entwicklungsumgebung beim Kunden“ steht im chronologischen Ablauf des Projekts *spätestens* an dieser Stelle. Selbstverständlich ist es möglich, die Entwicklungsumgebung bereits sehr viel früher aufzusetzen. Wird das Projekt nicht beim Kunden erstellt, sondern beim Auftragnehmer, steht die Entwicklungsumgebung gleich zu Projektbeginn zur Verfügung.

Das Einrichten der Entwicklungsumgebung beinhaltet hier auch Produktionssicherung bzw. Sicherstellen der Lauffähigkeit des vorhandenen Systems. Dafür gibt es natürlich viele unterschiedliche Modelle und Möglichkeiten, die an den jeweiligen Bedarf und das Umfeld angepasst werden müssen. Die hier dargestellte Variante ist nur eine denkbare Lösung.

3.1.4.10.1 Tätigkeiten: Einrichten der Entwicklungsumgebung

- Entwerfen eines Konzepts für die Produktionssicherung in enger Zusammenarbeit mit dem Database Administrator;
- Produktionssicherungskonzept mit dem Kunden verhandeln und abstimmen;
- Datenbankumgebung auf dem Server einrichten;
- Beobachten des laufenden Systems, der Use Cases;

- Installieren von Entwicklungs-, Test- und (neuer) Produktionsdatenbank, sowie ggf. weiterer Datenbanken.

3.1.4.10.2 Kompetenzfelder: Einrichten der Entwicklungsumgebung

Fähigkeiten/Fertigkeiten

- Bedingungen für die Produktionssicherung aufnehmen und analysieren können;
- Systembedingungen aufnehmen und analysieren können;
- Konzepte für Produktionssicherung entwerfen und abstimmen können;
- Datenbankumgebung(en) (auf Servern) einrichten können;
- laufende Systeme systematisch und gezielt beobachten können;
- Datenbanken installieren können.

Wissen

- Produktionssicherung,
- Migration,
- Betriebssysteme (Server, Netze),
- Datenbanksysteme,
- Datenbankinstallation und -konfiguration,
- Backup und Recovery.

Methoden/Werkzeuge

- Replikationstechniken,
- Datenbanksystem.

3.1.4.10.3 Beispiel: Einrichten der Entwicklungsumgebung

Da es keine alte Anwendung gab, musste für „eForms“ kein Produktionssicherungsmodell erstellt werden. Allerdings wurde die Anwendung beim Kunden und in enger Zusammenarbeit mit Mitarbeitern der Bank entwickelt. Es gab daher eine Entwicklungsumgebung, die aus insgesamt drei Datenbanken (Entwicklung, Test, Produktion) bestand.

3.1.4.11 Abstimmen der Anwendungslogik im Team

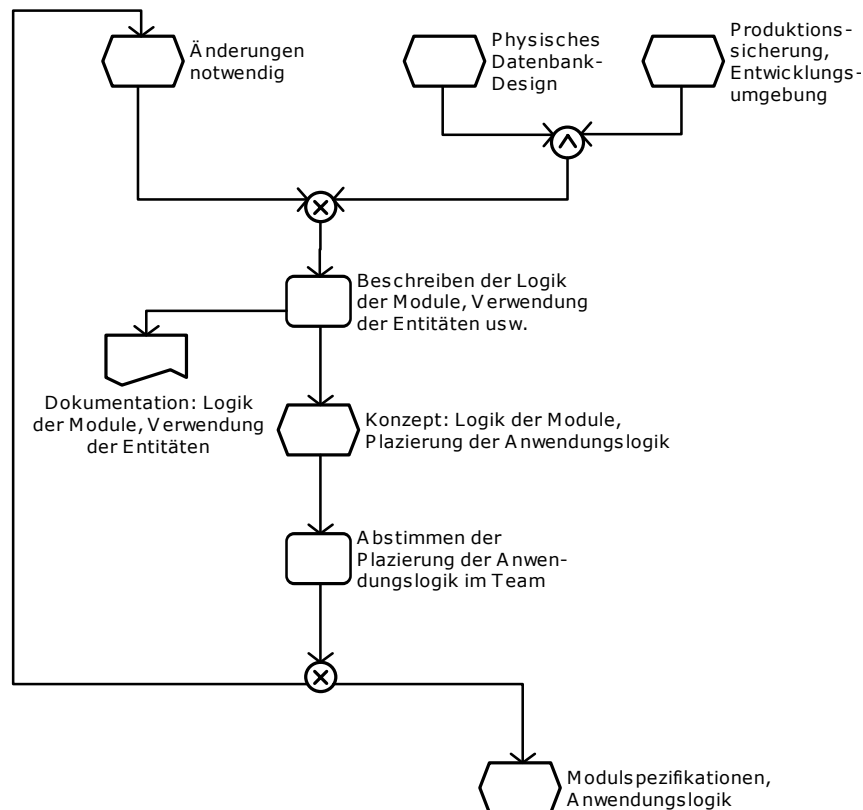


Abbildung 11: Abstimmen der Anwendungslogik

Anmerkung: Der Teilprozess „Abstimmen der Anwendungslogik“ hat zwei wichtige Funktionen. Zum einen dient er der Dokumentation für den weiteren Verlauf des Programmierens und Aufbaus der Datenbank. Zum anderen ermöglicht dieser Prozess, die Erstellung der unterschiedlichen Module auf mehrere Personen zu verteilen, also in den folgenden Prozessen (insbesondere beim Realisieren der Trigger und Stored Procedures) arbeitsteilig zu verfahren. Das Zusammenführen der erstellten Komponenten ist dann Aufgabe des IT Configuration Coordinators.

3.1.4.11.1 Tätigkeiten: Abstimmen der Anwendungslogik

- Beschreiben der Logik der Module, der Verwendung der Entitäten, der Aufruf- und Rückgabeparameter sowie der Schnittstellen zu anderen Programmen;
- Abstimmen der Platzierung der Anwendungslogik im Team.

3.1.4.11.2 Kompetenzfelder: Abstimmen der Anwendungslogik

Fähigkeiten/Fertigkeiten

- Logik der Module etc. sprachlich und grafisch beschreiben und darstellen können;
- Überblick über die gesamte Anwendung, ihren (geplanten) Funktionsumfang und ihre Schnittstellen haben;
- Tiefgreifende SQL- Kenntnisse (bzw. Kenntnisse des verwendeten SQL-Dialekts oder der verwendeten Datenbankabfrage- und Datenbankmanipulationssprache);
- Spezifikationen erstellen können;
- anderen gegenüber die eigene, fachliche Meinung - auch in Konfliktsituationen - begründet und angemessen vertreten können.

Wissen

- Produktionssicherung,

- System- und Datenbankarchitektur,
- System- und Programmschnittstellen,
- vgl. 3.1.4.3.2 „Kompetenzfelder: “ und 3.1.4.5.2 „Kompetenzfelder: Verfeinern der Systementwürfe“
- spezifische Produkt-Kenntnisse,
- Datenbankprogrammierung.

Methoden/Werkzeuge

- SQL (SQL-Dialekte, Datenbankabfrage- und Datenbankmanipulationssprachen),
- CASE-Tool (für Schnittstellendefinitionen etc.),
- Datenbank.

3.1.4.11.3 Beispiel: Abstimmen der Anwendungslogik

Beim Projekt „eForms“ war zwar die eigentliche Datenbankprogrammierung nicht sehr aufwändig, trotzdem sollten die spezifizierten Module von unterschiedlichen Mitarbeitern realisiert werden. Dafür wurden - ausgehend vom System-Design - die Logik der Module, die Verwendung der Entitäten, die Parameter für Aufruf- und Rückgabe und die unterstützenden Module beschrieben und im Team abgestimmt.

3.1.4.12 Realisieren der Stored Procedures und Trigger

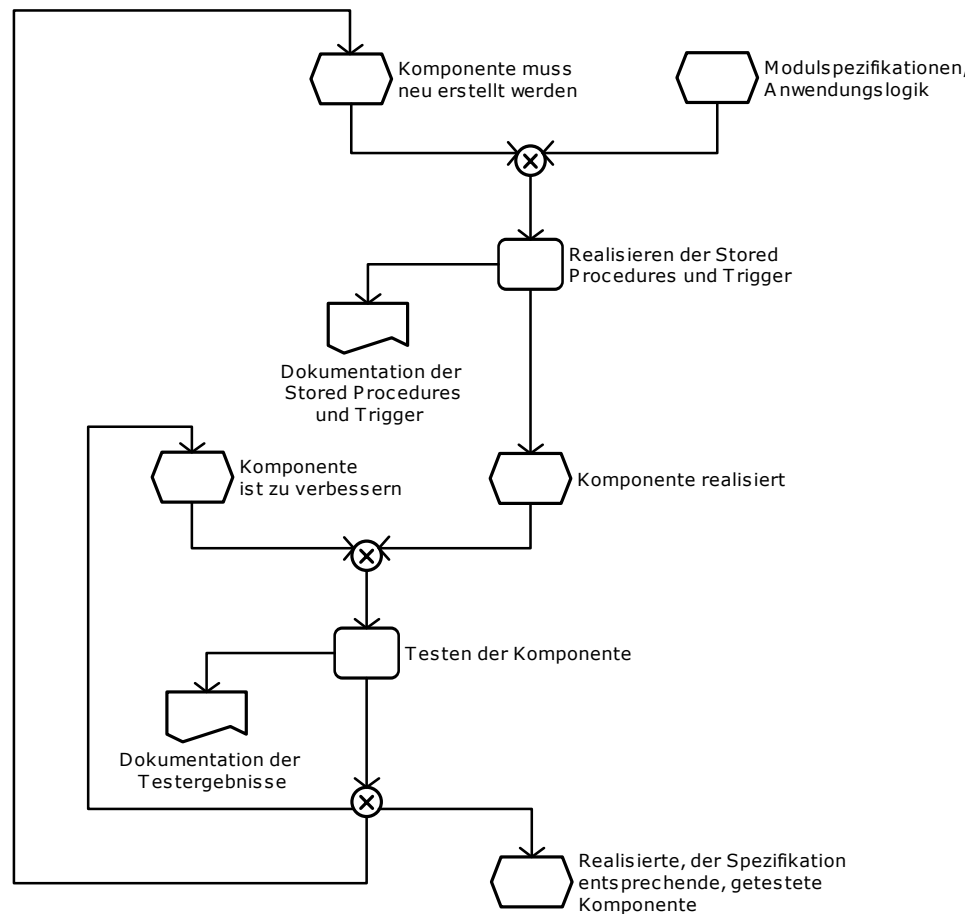


Abbildung 12: Realisieren der Trigger und Stored Procedures

3.1.4.12.1 Tätigkeiten: Realisieren der Stored Procedures und Trigger

- Realisieren der Trigger und Stored Procedures;
- Testen, ob die jeweilige Komponente der Spezifikation entspricht.

3.1.4.12.2 Kompetenzfelder: Realisieren der Stored Procedures und Trigger

Anmerkung: Dieser Teilprozess wurde von allen Beteiligten als der in der Praxis lernhaltigste bezeichnet, da sich der Datenbankentwickler/-programmierer hier bei jedem Projekt auf neue Umgebungen und Bedingungen einstellen und seine Produkte anpassen muss. Um diesen - oft sehr spezifischen - Anforderungen gerecht zu werden, ist die Aneignung neuer Fähigkeiten und Kenntnisse in der Regel unerlässlich. Daher kann aber auch die Beschreibung der zugehörigen Fähigkeiten hier nur sehr generisch und des zugehörigen Wissens nur beispielhaft - damit unvollständig - und ebenfalls generisch geschehen.

Fähigkeiten/Fertigkeiten

- Trigger und Stored Procedures mit angemessenen Methoden und Werkzeugen realisieren können;
- Tests der Komponenten gemäß Spezifikationen (vgl. 3.1.4.8 „Ableiten der Testfälle“) und Anwendungslogik (vgl. 3.1.4.11 „Abstimmen der Anwendungslogik im Team“) durchführen können,
- Komponenten angemessen dokumentieren können.

Wissen

- Datenbankprogrammierung,
- Betriebssysteme, Netzwerke, Netzwerkprotokolle,
- SQL,
- vertiefte Datenbank-Produktkenntnisse,
- höhere Programmiersprache(n),
- verteilte Komponenten und Datenbankanbindung.

Methoden/Werkzeuge

- Datenbankprogrammierung (DDL-Befehle, Entwicklung von Packages, Entwicklung von Triggern, Erweitern des SQL-Sprachumfangs mit benutzerdefinierten Funktionen, Analyse und Fehlersuche, Verwaltung der Programmeinheiten, Zusammenspiel der SQL-Programmeinheiten mit anderen Datenbankobjekten, Verwendung von Standard-Packages, Überblick über Large Objects),
- SQL, ggf. entsprechende Dialekte: Grundlagen, Technologie, Sprachelemente, Einbindung in verwendete Tools, Fehlerbehandlung Datentypen und Exceptions,
- höhere Programmiersprache(n),
- Datenbanksystem,
- Entwicklungs- bzw. Programmierumgebungen.

3.1.4.12.3 Beispiel: Realisieren der Stored Procedures und Trigger

Bei „eForms“ ging es u. a. darum, das System für die Benutzer so komfortabel wie möglich zu gestalten. So wurden z. B. Prozeduren geschrieben, die die Werte der am Bildschirm ausgefüllten Felder der Formulare automatisch mit der vorbereiteten Postscript- bzw. pdf-Datei des Formulars verbanden. Bildschirm-Darstellung wie Postscript-Datei waren in der Datenbank als BLOBs (Binary Large Objects) abgespeichert und mit Metadaten bezüglich der Lage und Größe der auszufüllenden Felder versehen. Die Metadaten erlauben dann die korrekte Zuordnung und Darstellung der vom Nutzer eingetragenen Werte.

3.1.4.13 Programmieren der Software-Bausteine der Zugriffsschicht

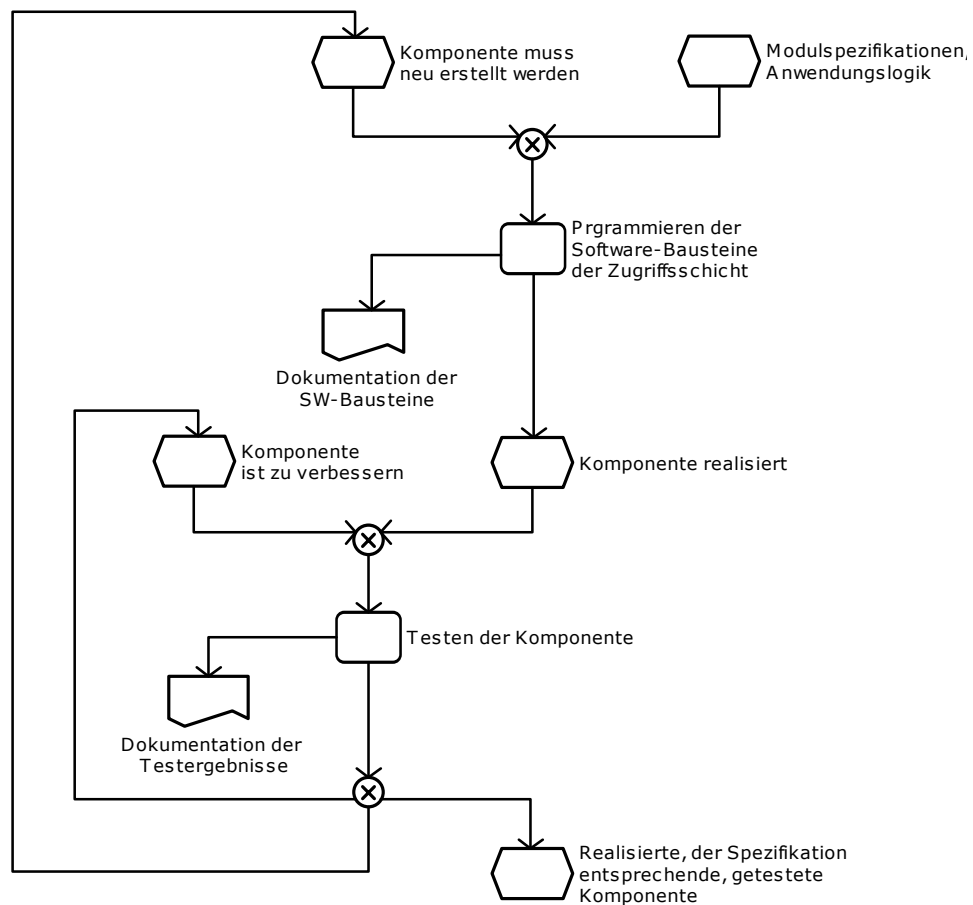


Abbildung 13: Programmieren der Software-Bausteine der Zugriffsschicht

3.1.4.13.1 Tätigkeiten: Programmieren der Software-Bausteine der Zugriffsschicht

- Realisieren der Software-Komponenten der Zugriffsschicht;
- Testen, ob die jeweilige Komponente der Spezifikation entspricht.

3.1.4.13.2 Kompetenzfelder: Programmieren der Software-Bausteine der Zugriffsschicht

Fähigkeiten/Fertigkeiten

- Komponenten der Zugriffsschicht mit angemessenen Methoden und Werkzeugen realisieren können;
- Tests der Komponenten gemäß Spezifikationen (vgl. 3.1.4.8 „Ableiten der Testfälle“) und Anwendungslogik (vgl. 3.1.4.11 „Abstimmen der Anwendungslogik im Team“) durchführen können;
- Komponenten angemessen dokumentieren können.

Wissen

- Betriebssysteme, Netzwerke, Netzwerkprotokolle,
- System- und Programmschnittstellen,
- verteilte Komponenten und Datenbankanbindung.

Methoden/Werkzeuge

- Programmieren in höheren Programmiersprachen,

- Programmieren für Inter-/Intranet, insbesondere Script- und Seitenbeschreibungssprachen,
- Entwicklungs- bzw. Programmierumgebungen,
- Datenbank.

3.1.4.13.3 Beispiel: Programmieren der Software-Bausteine der Zugriffsschicht

Zu den Modulen der Zugriffsschicht gehörte beim Projekt „eForms“ u. a. die Möglichkeit, die am Bildschirm sichtbaren und nur dort ausgefüllten Formulare ausdrucken zu können. Dazu wurde über eine entsprechende Routine eine Postscript-Datei erzeugt und im Dateisystem auf dem Datenbankserver gespeichert. Über einen Daemon-Prozess werden diese Dateien auf die entsprechenden Drucker geroutet. Da die Drucker der ABN AMRO aber kein Postscript- sondern PCL-Drucker sind, mussten sämtliche Dateien auch noch nach pdf konvertiert werden. Dies geschah durch die Verwendung von Ghostscript, da dies keine zusätzlichen Lizenz-Kosten erzeugt.

3.1.4.14 Mitarbeiten bei Systemintegration und Systemtests

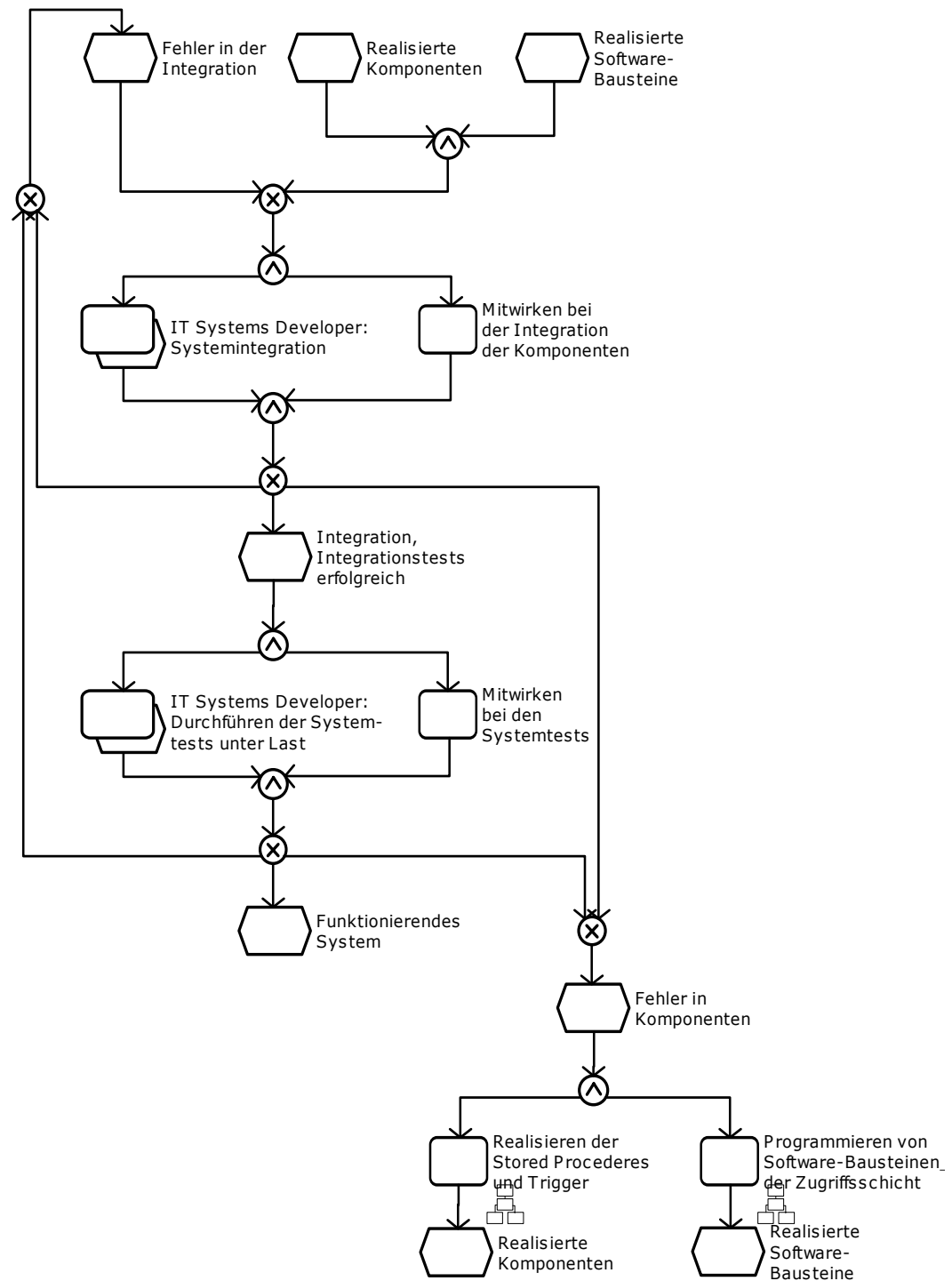


Abbildung 14: Mitarbeit bei Systemintegration und Systemtests

Anmerkung: Dieser Teilprozess wird in jedem konkreten Projekt anders aussehen. In kleinen Projekten, in denen ausschließlich eine Datenbank entwickelt wird, wird der Database Developer die Systemintegration und die Systemtests selber machen. Auf der anderen Seite stehen sehr große Projekte, in denen sich das Mitwirken des Datenbankentwicklers bei Systemintegration und -tests darauf beschränkt, dass er die auftretenden Fehler beheben muss. Bei mittelgroßen Projekten arbeitet der Database Developer mit anderen Teammitgliedern zusammen. Hier hat er auch die Möglichkeit, sich auf Systemintegration oder Tests zu spezialisieren.

3.1.4.14.1 Tätigkeiten: Mitarbeiten bei Systemintegration und Systemtests

- Mitwirken bei der Integration der Komponenten;
- Mitwirken bei den Systemtests;
- Fallen bei der Systemintegration oder den Systemtests Fehler in den Komponenten auf, werden diese durch einen Rücksprung in die entsprechenden Prozesse behoben.

3.1.4.14.2 Kompetenzfelder: Mitarbeiten bei Systemintegration und Systemtests

Fähigkeiten/Fertigkeiten

- Komponenten zu einem Gesamtsystem integrieren und dabei systematisch Fehler identifizieren können;
- Systemtests gemäß Spezifikation durchführen können;
- mit der Testumgebung und automatischen Tests umgehen können;
- ggf. Testdokumentation (gemäß Dokumentationsrichtlinien) schreiben können;

Wissen

- Datenbankmodell,
- Datenbankarchitektur,
- Systemspezifikationen,
- Testspezifikation, Testsystematik, Testmodelle,
- vgl. auch 3.1.4.3.2 „Kompetenzfelder: Überprüfen des Systemdesigns“ bis 3.1.4.13.2 „Kompetenzfelder: Programmieren der Software-Bausteine der Zugriffsschicht“).

Methoden/Werkzeuge

- SQL,
- Datenbanksystem,
- Analysetools,
- ggf. automatisierte Testumgebung.

3.1.4.14.3 Beispiel: Mitarbeiten bei Systemintegration und Systemtests

Da das Beispielprojekt ein eher kleines Projekt war, hatte der Database Developer viel mit Sytemintegration und -tests zu tun: Die Benutzerfreundlichkeit von „eForms“ beinhaltete auch, dass für die Nutzer des Systems keine erneute Anmeldung oder Authentifizierung notwendig sein sollte. Die einmalige Anmeldung am Netzwerk (Windows NT) sollte gleichzeitig die Nutzung des personalisierten „eForms“ Portal ermöglichen. Diese Funktionalität wurde zunächst mit Hilfe eines Java-Applets realisiert, welches auch in der Testumgebung - die nicht in das reguläre Banknetz eingebunden war - problemlos funktionierte. Bei der Bank war das Anmeldeskript dann allerdings nicht lauffähig, da aus Sicherheitsgründen auf den Rechnern bei der Bank keine Java Virtual Machine laufen durfte, so dass das Skript nicht eingesetzt werden konnte, sondern eine neue Lösung gefunden werden musste (die aus Sicherheitsgründen nicht bekannt gegeben wurde).

3.1.4.15 Mitarbeiten bei der Installation der Datenbank und der Migration der Bestandsdaten

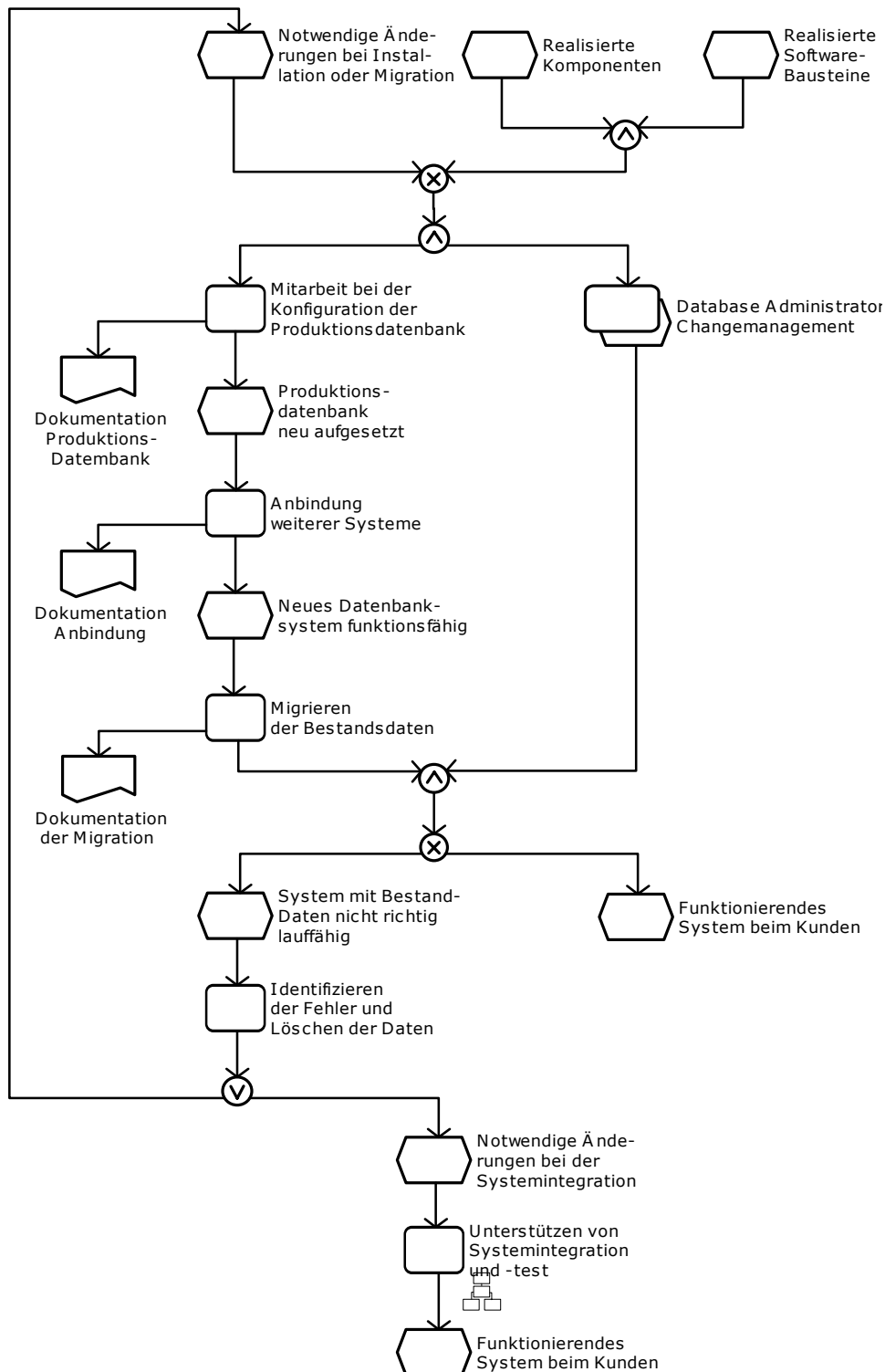


Abbildung 15: Installieren der Datenbank und Migration der Bestandsdaten

Anmerkung: Die Installation und Konfiguration der Datenbank und ihrer Schnittstellen erfolgt sinnvollerweise in enger Zusammenarbeit mit dem beim Kunden zuständigen Administrator. Wie dies konkret aussieht, hängt ebenfalls vom konkreten Projekt ab. Auf jeden Fall muss der Database Developer in der Lage sein, diese Installation und Konfiguration auch selbstständig vorzunehmen. Dies gilt in verstärktem Maße für die Migration. Sie bietet außerdem eine weitere Möglichkeit für den Datenbankentwickler, sich zu spezialisieren, da

die Migration größerer Datenbestände eine hoch komplexe, komplizierte und verantwortungsvolle Aufgabe ist.

Die hier vorgesehene Migration der Bestandsdaten per „Big Bang“ ist nur eine Möglichkeit von mehreren. Selbstverständlich ist die Migration der Bestandsdaten vom gewählten Migrations- sowie Produktionssicherungsmodell abhängig. Die Variante „Big Bang“ steht hier als eine häufig gewählte und leicht darstellbare.

3.1.4.15.1 Tätigkeiten: Mitarbeiten bei der Installation der Datenbank und der Migration der Bestandsdaten

- Installieren und Konfigurieren der endgültigen Produktionsdatenbank (vgl. 3.1.4.10 „Einrichten der Entwicklungsumgebung beim Kunden“) in Zusammenarbeit mit dem Database Administrator;
- Anbinden der Datenbank an die anderen Teile des Systems;
- Konfigurieren der Schnittstellen;
- Migrieren der Bestandsdaten vom vorhandenen System auf die neue Produktionsdatenbank in Zusammenarbeit mit dem Datenbankadministrator;
- Testen der Migration und Identifizieren von Fehlern;
- Falls Fehler bei der Systemintegration gemacht wurden: Mitarbeit bei erneuter Systemintegration.

3.1.4.15.2 Kompetenzfelder: Mitarbeiten bei der Installation der Datenbank und der Migration der Bestandsdaten

Fähigkeiten/Fertigkeiten

- Datenbankumgebungen auf Servern einrichten können;
- laufende Systeme systematisch und gezielt beobachten können;
- Datenbanken in Zusammenarbeit mit dem Administrator installieren können;
- Datenbank in Zusammenarbeit mit dem Administrator konfigurieren können;
- Schnittstellen programmieren können;
- Schnittstellen in Zusammenarbeit mit dem Administrator konfigurieren können;
- Replikationstechniken anwenden können;
- Backup und Recovery durchführen können;
- Migrationstest gemäß Testplan durchführen können (vgl. 3.1.4.6 „Erstellen des Migrationsmodells“ und 3.1.4.8 „Ableiten der Testfälle“);
- Fehler lokalisieren und identifizieren können;
- konstruktiv im Team und auch mit externen Fachkräften zusammenarbeiten können.

Wissen

- wie 3.1.4.10.2 „Kompetenzfelder: Einrichten der Entwicklungsumgebung“,
- wie 3.1.4.13.2 „Kompetenzfelder: Programmieren der Software-Bausteine der Zugriffsschicht“,
- wie 3.1.4.6.2 „Kompetenzfelder: Erstellen des Migrationsmodells“
- wie 3.1.4.8.2 „Kompetenzfelder: Ableiten der Testfälle“.

Methoden/Werkzeuge

- Datenbanken-Setup und Backup (altes und neues System),
- Datenbanken-Test und -Vergleich mit sinnvollen Use Cases,
- Datenbanksystem,
- Analysetools,

- ggf. Entwicklungsumgebung,
- ggf. automatisierte Testumgebung.

3.1.4.15.3 Beispiel: Mitarbeiten bei der Installation der Datenbank und der Migration der Bestandsdaten

Im Rahmen des Einrichtens der Entwicklungsumgebung war bei der ABN AMRO das Datenbanksystem bereits installiert worden. Die Datenbank selbst wurde im Rahmen von „eForms“ neu erstellt.

Da es keine Bestandsdaten gab, die übernommen werden mussten, wurden in dieser Phase alle Formulare in die Datenbank eingepflegt und gegebenenfalls vorher konvertiert.

3.1.4.16 Mitarbeiten bei der Erstellung von Handbüchern und bei den Nutzerschulungen

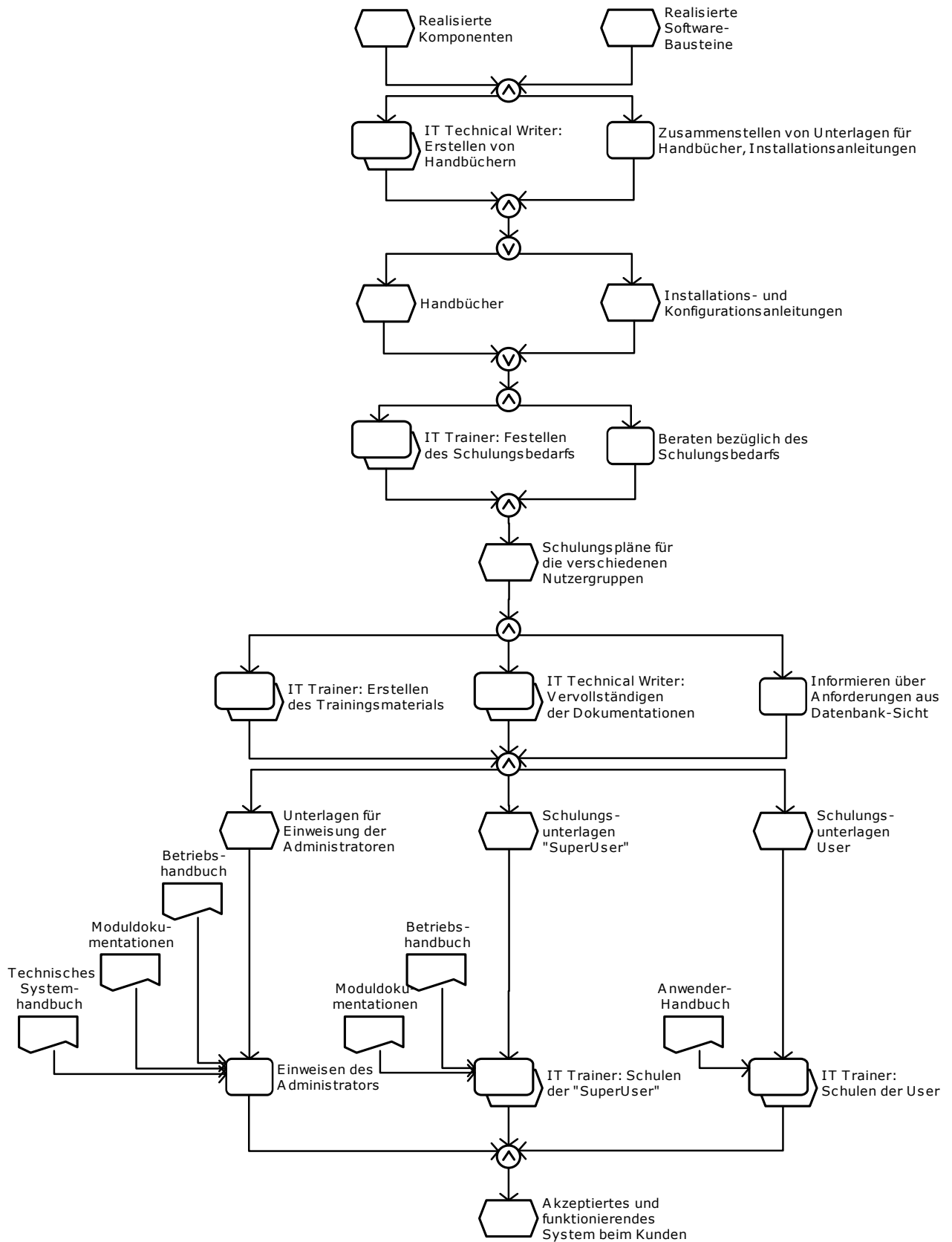


Abbildung 16: Mitarbeiten bei den Nutzerschulungen

3.1.4.16.1 Tätigkeiten: Mitarbeiten bei der Erstellung von Handbüchern und bei den Nutzerschulungen

- Zusammenstellen der Unterlagen für Systemhandbücher, Installations- und Konfigurationsanleitungen sowie Schulungsmaterialien;
- Beraten des IT Trainer bezüglich des Schulungsbedarfs (beispielsweise zu gravierenden Unterschieden in der Bedienung des alten und des neuen Systems);
- Informieren von IT Technical Writer und IT Trainer über relevante Themen für die entsprechenden Unterlagen;
- Einweisen des Administrators in das neue System.

3.1.4.16.2 Kompetenzfelder: Mitarbeiten bei der Erstellung von Handbüchern und bei den Nutzerschulungen

Fähigkeiten/Fertigkeiten

- zielgruppengerechtes Material für den IT Technical Writer und den IT Trainer bereitstellen können;
- Kenntnisse, Fähigkeiten und notwendige Qualifikationen der Nutzer erkennen können;
- fachliche Hinweise zu notwendigen Schulungen geben können;
- fachliche Hinweise zum Schulungsplan geben können;
- fachliche Hinweise zu Dokumentationsstandards geben können;
- Administratoren einweisen und schulen können;
- Kenntnisse aussuchen und vermitteln können, die für die unterschiedlichen Nutzergruppen notwendig und hinreichend sind;
- Informationen angemessen aufbereiten und präsentieren können;
- themengerecht im Team kommunizieren können.

Wissen

- Datenbanksysteme,
- Produktkenntnisse,
- Betriebssysteme, Netzwerke,
- Systematik von Kenntnissen, Metawissen (welche Kenntnisse, die z. B. zur Systemadministration der Datenbank notwendig sind, bauen auf anderen Kenntnissen auf?),
- Projektspezifika aus den Prozessen „Mitwirken bei der Festlegung des Entwicklungsrahmens“, „Einbeziehen und Abgleichen von Standards“, „Verfeinern der Systementwürfe“, „Erstellen des Migrationsmodells“, „Erstellen des physischen Datenbankdesigns“, „Einrichten der Entwicklungsumgebung beim Kunden“, „Abstimmen der Anwendungslogik im Team“, „Realisieren der Stored Procedures und Trigger“, „Programmieren der Software-Bausteine der Zugriffsschicht“, „Mitarbeiten bei der Installation der Datenbank und der Migration der Bestandsdaten“,
- Pädagogik/Didaktik.

Methoden/Werkzeuge

- Datenbanksystem,
- Configuration Management System (CMS).

3.1.4.16.3 Beispiel: Mitarbeiten bei der Erstellung von Handbüchern und bei den Nutzerschulungen

Bei der ABN AMRO musste durch Oracle der Administrator im Umgang mit der Oracle Datenbank geschult werden. Dies geschah zum einen durch ein entsprechendes Grundlagenseminar, zum anderen durch die Einweisung des Administrators in das neue System durch den Database Developer. Daneben war die Schulung der „SuperUser“ nämlich

derjenigen, die neue Formulare vorbereiten und in das System einstellen sollten, Schwerpunkt des Schulungsplans. Hier ergaben sich einmal Anforderungen an das Datenbanksystem, nämlich eine sehr weitgehende Nutzerfreundlichkeit, zum anderen aber ein deutlicher Bedarf an intensiver Einweisung/Schulung.

Entsprechend des Schulungsplans und der gesetzten Schwerpunkte wurde durch den Oracle Database Developer der Administrator eingewiesen. Diese Schulung fand am System statt. Hier wurden aufgrund der vorbereiteten Unterlagen alle wesentlichen Punkte besprochen.

Zum Abschluss der Arbeiten besaß die ABN AMRO das auf ihre Bedürfnisse abgestimmte System „eForms“, mit dem sie sehr zufrieden ist. Es wurde dann sehr schnell ein Aufbau-Projekt vereinbart, dessen Ziel es ist, „eForms“ deutlich zu erweitern. Dazu gehören u. a. die Auswahl, Anzeige und der Druck von archivierten Formularen und Handbüchern, ein Formel-Wizzard zur Unterstützung einer dynamischen Berechnung von Werten innerhalb eines elektronischen Formulars und dynamische Ablage- und Freigabemöglichkeiten für die Aufnahme allgemeiner Handbücher.

Oracle hat das Projekt „eForms“ und die nachfolgenden Projekte genutzt, um ein Projekt-Framework zu erstellen, welches ebenfalls mit großem Erfolg angewendet wird.