

Entwurf, Entwicklung und Erprobung einer Komponente zur
Erweiterung des Facility-Management-Systems (FMS) am
Deutschen Elektronen-Synchrotron (DESY) um Funktionalitäten zur Unterstützung der Schlüsselverwaltungsprozesse

Hausarbeit zur Diplomprüfung

an der

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN

HAMBURG

Fakultät Design Medien Information

Studiendepartment Information

vorgelegt von

Melanie Wessel

Hamburg, August 2008

1. Prüfer: Prof. Dr. Franziskus Geeb
2. Prüfer: Prof. Dr. Martin Gennis

Inhaltsverzeichnis

Abbildungsverzeichnis.....	iv
Tabellenverzeichnis.....	v
Abkürzungsverzeichnis	v
1 Einleitung	1
2 Projekt.....	2
2.1 Anlass	2
2.2 Ziel und Nutzen des Projektes	3
2.3 Projektablauf.....	4
2.4 Projektbeteiligte.....	5
3 Grundlagen	6
3.1 DESY	6
3.2 Prozessunterstützung durch Informationssysteme	7
3.3 Facility Management	9
3.4 CAFM-Systeme	10
3.5 GIS/FMS – das CAFM-System am DESY	11
3.6 Schlüsselverwaltung	12
4 Werkzeuge und Methoden	14
4.1 Anforderungserhebung und -management	14
4.1.1 Was ist eine Anforderung?	15
4.1.2 Phasen der Anforderungsentwicklung.....	15
4.2 Anforderungserhebung mit der Metaplan-Technik.....	19
4.3 Analyse	20
4.4 Geschäftsprozessanalyse mit der Unified Modelling Language	22
4.4.1 Was ist UML?.....	22
4.4.2 Diagrammtypen der UML	22
4.5 Systementwurf (Design).....	28
4.6 Benchmarking.....	30
4.7 Funktionspunktanalyse	31

5	Analyse	34
5.1	Ist-Zustand	34
5.2	Geschäftsprozessanalyse	37
5.2.1	Die Geschäftsprozesse der Schlüsselverwaltung im Überblick.....	38
5.2.2	Analyse des Schlüsselausgabeprozesses.....	38
5.2.3	Systemumfang des SVS.....	43
5.3	Soll-Zustand.....	44
6	Systemlösung.....	46
6.1	Systementwurf	46
6.2	Implementierungsphase	50
6.3	Systemvorstellung: Das DESY-SVS	51
6.3.1	Der Schlüsselausgabe-Wizard	51
7	Benchmark-Test	54
7.1	Durchführung des Benchmark-Tests	55
7.2	Ergebnisanalyse	57
7.2.1	Gesamtergebnis	57
7.2.2	Bewertung der einzelnen Testanwender.....	58
7.2.3	Bewertung in den Aufgabenblöcken	59
7.2.4	Bewertung in den Szenarien	60
7.3	Zusammenfassung der Ergebnisse	61
8	Kosten- und Aufwandsschätzung für die SVS-Entwicklung.....	62
8.1	Durchführung der Aufwandsschätzung.....	62
8.1.1	Aufwandsschätzung für die Realisierung der Schlüsselausgabefunktionen.....	64
8.1.2	Aufwandsschätzung für die Realisierung der Schlüsselrückgabefunktionen	68
8.1.3	Aufwandsschätzung der gesamten SVS-Entwicklung.....	70
8.2	Ergebnisse der Kostenschätzung	75
9	Ausblick.....	78
10	Fazit und Projektergebnisse	79
	Quellen	82
	Eidesstattliche Versicherung.....	85

Anhang

1	Einleitung	I
2	Prozessanforderungen	I
2.1	Prozessbeteiligte	I
2.2	Funktionale Prozessanforderungen	I
2.2.1	Allgemeine Prozessanforderungen	II
2.2.2	Anforderungen an den Schlüsselausgabeprozess	II
2.3	Randbedingungen	III
2.4	Anforderungen an die Beteiligten zur Prozessumsetzung.....	III
3	Systemanforderungen	III
3.1	Anwenderanforderungen an das Schlüsselverwaltungssystem	IV
3.2	Funktionale Systemanforderungen	V
3.2.1	Allgemeine Systemanforderungen.....	V
3.2.2	Anforderungen an den Schlüsselausgabe-Wizard	V
3.3	Nicht-funktionale Systemanforderungen.....	VII

Abbildungsverzeichnis

Abb. 1: Projektablauf (eigene Darstellung)	4
Abb. 2: Luftbild des DESY-Geländes (Quelle: DESY 2006)	6
Abb. 3: Blick in den Hera-Tunnel (Quelle: DESY 2005 ²)	7
Abb. 4: Umsetzungsmöglichkeiten von Aktivitäten (eigene Darstellung)	8
Abb. 5: Lebensphasen einer Sachressource (nach Nävy 2003, S. 72)	9
Abb. 6: Aspekte des Facility Managements (nach Nävy 2003, S. 4)	10
Abb. 7: Komponenten eines CAFM-Systems (nach Nävy 2003, S. 111).....	11
Abb. 8: GIS/FMS-Architektur (nach Eucker 2005, S. 15; Robben 2004, S. 14).....	12
Abb. 9: DESY-Konzept der Schlüsselverwaltung (eigene Darstellung)	13
Abb. 10: Phasen der Anforderungsentwicklung (nach Sommerville 2005, S. 17)	16
Abb. 11: Anforderungsschablone (nach Rupp 2002, S. 471)	18
Abb. 12: Anforderungsschablone mit Bedingung (nach Rupp 2002, S. 472)	18
Abb. 13: Schritte der Analysephase (eigene Darstellung)	20
Abb. 14: Anwendungsfalldiagramm (eigene Darstellung)	23
Abb. 15: Aktivitätsdiagramm (eigene Darstellung).....	24
Abb. 16: Aktivitätsdiagramm mit Objektfluss (eigene Darstellung)	25
Abb. 17: Klassendiagramm (eigene Darstellung).....	27
Abb. 18: Beispiel einer Systemarchitektur (nach Oestereich 2001, S. 146).....	28
Abb. 19: Vorgehensweise beim Benchmarking (eigene Darstellung).....	30
Abb. 20: Ablauf einer Funktionspunktanalyse (eigene Darstellung).....	31
Abb. 21: Ergebnis der Kartenabfrage	35
Abb. 22: Auslösender Akteur des Prozesses <i>Schlüssel ausgeben</i> (eigene Darstellung)	39
Abb. 23: Ablauf des Schlüsselausgabeprozesses (eigene Darstellung)	40
Abb. 24: Umsetzungsmöglichkeiten der Verwaltungsaktionen (eigene Darstellung)	41
Abb. 25: Ablauf des Schlüsselausgabeprozesses mit IS (eigene Darstellung)	42
Abb. 26: Systemumfang und externe Akteure (eigene Darstellung)	43
Abb. 27: Identifikation der Funktionsklassen (eigene Darstellung)	46
Abb. 28: Systemstruktur der Schlüsselausgabekomponente (eigene Darstellung).....	49
Abb. 29: Gebäude- und Raumauswahl des Schlüsselausgabe-Wizards (Screenshot)	52
Abb. 30: Schlüsselauswahl-Liste des Ausgabe-Wizards (Screenshot).....	53
Abb. 31: Ergebnis der Besitzer-Suche (Screenshot).....	53
Abb. 32: Übersicht der Vorgangsdaten vor Transaktionsspeicherung (Screenshot)	54
Abb. 33: Druckvorschau des Belegs zur Schlüsselausgabe (Screenshot).....	54
Abb. 34: Gesamtergebnis des Benchmark-Tests (eigene Darstellung)	57
Abb. 35: Gesamtergebnis unterteilt nach Testanwendern (eigene Darstellung).....	58
Abb. 36: Systembewertung in den Aufgabenblöcken (eigene Darstellung)	59
Abb. 37: Systembewertung in den Szenarien (Teil I, eigene Darstellung).....	60
Abb. 38: Systembewertung in den Szenarien (Teil II, eigene Darstellung)	61
Abb. 39: Beispielhafte Eingabemaske (Screenshot).....	63
Abb. 40: Beispielhafte Ausgabemaske (Screenshot)	63
Abb. 41: Systemstruktur der Schlüsselausgabekomponente (eigene Darstellung).....	64
Abb. 42: Systemstruktur der Schlüsselrückgabekomponente (eigene Darstellung)	68
Abb. 43: Systemstruktur der Schlüsselstammdatenerfassungskomponente (eigene Darstellung)	72

Tabellenverzeichnis

Tabelle 1: Matrix zur Ermittlung der Funktionspunkte	32
Tabelle 2: Aktionen des Schlüsselausgabeprozesses	40
Tabelle 3: Aktionen des Schlüsselausgabeprozesses mit IS	42
Tabelle 4: Funktionsklassen der Schlüsselausgabekomponente	66
Tabelle 5: Matrix zur Ermittlung der Funktionspunkte	66
Tabelle 6: Funktionsklassen der Schlüsselrückgabekomponente	69
Tabelle 7: Funktionsklassen der Stammdatenerfassungskomponente	73

Abkürzungsverzeichnis

A	Aufwand
A _F	Aufwand zur Realisierung der Schlüsselfundkomponente
A _{ges}	Gesamtaufwand zur Realisierung des SVS
A _{SA}	Aufwand zur Realisierung der Schlüsselausgabekomponente
A _{SA_SR}	Aufwand zur Realisierung der Schlüsselausgabe- und Schlüsselrückgabekomponente
A _{SA_SR*}	Aufwand zur Realisierung der Schlüsselausgabe- und Schlüsselrückgabekomponente ohne Spezifikation und Systemtest
A _{SA_SR_Diff}	Differenz zwischen geschätztem und tatsächlichem Aufwand zur Realisierung der Schlüsselausgabe- und Schlüsselrückgabekomponente
A _{SA_SR_Real}	tatsächlicher Aufwand zur Realisierung der Schlüsselausgabe- und Schlüsselrückgabekomponente
A _{SD}	Aufwand zur Realisierung der Schlüsseldefektkomponente
A _{Spez}	Aufwand zur Spezifikationserstellung
A _{SR}	Aufwand zur Realisierung der Schlüsselrückgabekomponente
A _{SStD}	Aufwand zur Realisierung der Schlüsselstamdatenerfassungskomponente
A _{Test}	Aufwand für die Durchführung eines Systemtests
A _V	Aufwand zur Realisierung der Schlüsselverlustkomponente
A _{V_F_SD_ZD_ZR}	Aufwand zur Realisierung der Komponenten für den Schlüsselverlust, Schlüsselfund, Schlüsseldefekt, Zylinderdefekt und die Zylinderrückgabe
A _{ZA}	Aufwand zur Realisierung der Zylinderausgabekomponente
A _{ZA_ZR}	Aufwand zur Realisierung der Zylinderausgabe- und Zylinderrückgabekomponente
A _{ZA_ZR_Diff}	Differenz zwischen geschätztem und tatsächlichem Aufwand zur Realisierung der Zylinderausgabe- und Zylinderrückgabekomponente
A _{ZA_ZR_Real}	tatsächlicher Aufwand zur Realisierung der Zylinderausgabe- und Zylinderrückgabekomponente
A _{ZD}	Aufwand zur Realisierung der Zylinderdefektkomponente

Abkürzungsverzeichnis

A _{ZR}	Aufwand zur Realisierung der Zylinderrückgabekomponente
A _{ZStD}	Aufwand zur Realisierung der Zylinderstammdatenerfas- sungskomponente
CAD	Computer Aided Design
CAFM	Computer Aided Facility Management
DESY	Deutsches Elektronen-Synchrotron
FLASH	Freier Elektronen Laser Hamburg
FM	Facility Management
FMS	Facility-Management-System
FP	Funktionspunkt
GEFMA	German Facility Management Association
GIS/FMS	Geoinformationssystem / Facility-Management-System
HERA	Hadron-Elektron-Ring-Anlage
ILC	International Linear Collider
IPP	Informationsmanagement, Prozesse, Projekte
IS	Informationssystem
K	Entwicklungskosten
K _{SA_SR_Diff}	Differenz zwischen geschätzten und tatsächlichen Kosten für die Realisierung der Schlüsselausgabe- und Schlüsselrückga- bekomponente
K _{Tag}	Entwicklungskosten pro Tag
K _{ZA_ZR_Diff}	Differenz zwischen geschätzten und tatsächlichen Kosten für die Realisierung der Zylinderausgabe- und Zylinderrückgabe- komponente
OMG	Object Management Group
PETRA	Positron-Elektron-Tandem-Ring-Anlage
REP	Requirements Engineering Process
SVS	Schlüsselverwaltungssystem
TDI	Total Degree of Influence
TP	Team Performance
UFP	Unadjusted Function Point
UFP _{SA}	Unadjusted Function Point der Schlüsselausgabe
UFP _{SR}	Unadjusted Function Point der Schlüsselrückgabe
UFP _{SStD}	Unadjusted Function Point der Schlüsselstammdatenerfassung
UML	Unified Modelling Language
V1	Verwaltungsabteilung 1
VAF	Value added Factor
XFEL	X-ray free-electron laser

1 Einleitung

Die vorliegende Diplomarbeit wurde innerhalb eines Projektes am Deutschen Elektronen-Synchrotron (DESY) in Hamburg erstellt. Ziel dieses Projektes ist die Einführung eines Informationssystems zur Unterstützung der Schlüsselverwaltungsprozesse am DESY.

Bis zur Beendigung des Projektes erfolgte die Organisation von Gebäude- und Raumschlüsseln mit Hilfe eines Systems, das nicht über eine Anbindung an aktuelle Personen-, Gebäude- und Raumdaten verfügt. Zylinderinformationen wurden separat auf analogen Gebäudeplänen gepflegt. Diese mangelnde Vernetzung der relevanten Informationen zur Schlüsselverwaltung erschwerte unter anderem das Ermitteln zutrittsberechtigter Personen zu einem bestimmten Raum. Dieser Zustand sollte durch die Entwicklung einer neuen Systemkomponente und durch ihre Integration in ein existierendes DESY-System, welches zur Verwaltung von Personen-, Gebäude- und Raumdatenbeständen verwendet wird, behoben werden.

Die Informationen zu den Mitarbeitern sowie die Gebäude- und Raumdaten werden bei DESY in einem Geoinformations- und Facility-Management-System (GIS/FMS) dokumentiert und gepflegt. Die Aktualität der Datensätze und der technische Reifegrad des Systems ermöglichten es, eine Komponente zur Verwaltung von Schlüsseln und Schließzylindern in das GIS/FMS zu integrieren. Das Schlüsselverwaltungssystem (SVS) wurde durch ein externes Entwicklerteam realisiert, dem zur technischen Umsetzung ein Anforderungsdokument – angefertigt im Hause DESY – vorlag.

Die Analyse der Schlüsselverwaltungsprozesse ist Inhalt der vorliegenden Diplomarbeit. Sie ist in zehn Kapitel unterteilt. Im Anschluss an die Einleitung wird der Projektrahmen beschrieben. Begonnen wird hierbei mit der Schilderung des Anlasses, welcher die Durchführung des Projektes rechtfertigt. Es folgen Ziel und Nutzen des Projektes sowie die Vorstellung der Projektbeteiligten.

Den dritten Abschnitt der Arbeit bildet das Grundlagenkapitel. Hier wird der Auftraggeber, das Deutsche Elektronen-Synchrotron, skizziert. Außerdem werden die Grundlagen zur Prozessunterstützung mit Hilfe von Informationssystemen erläutert. Ebenfalls Bestandteil des Grundlagenkapitels sind eine Einführung in das Facility Management (FM) sowie eine Vorstellung des GIS/FMS am DESY, welches für FM-Aufgaben eingesetzt wird. Des Weiteren wird der Bereich der Schlüsselverwaltung – eine Teilaufgabe von FM – abgesteckt.

Werkzeuge und Methoden, die innerhalb des Diplomprojektes eingesetzt und angewandt wurden, werden im vierten Kapitel vorgestellt. Hier wird insbesondere auf die Projektphasen *Anforderungserhebung*, *Analyse* und *Systementwurf* eingegangen. Als Hilfsmittel zur Anforderungserhebung fand die Metaplan-Technik Anwendung. Diese sowie die Grundlagen zur Unified Modelling Language (UML), mit Hilfe derer die Analyse der Schlüsselverwaltungsprozesse erfolgte, werden innerhalb dieses Kapitels vorgestellt. Weiterhin werden Methoden zur Durchführung eines Systemleistungs-

vergleichs (Benchmarking) und einer Aufwands- und Kostenschätzung für die Entwicklung des SVS (Funktionspunktanalyse) beschrieben.

Das folgende Analysekapitel setzt sich mit der Identifikation des Ist-Zustandes der Schlüsselverwaltung sowie der Analyse der Schlüsselverwaltungsprozesse auseinander. Im Hinblick auf das Ziel dieses Projektes wird ferner der Soll-Zustand skizziert, der in Form einer Spezifikation dokumentiert wird.

Die Präsentation des Systementwurfes ist Gegenstand des sechsten Kapitels. Weiterhin folgen innerhalb dieses Arbeitsabschnittes die Schilderung der Implementierungsphase und die Vorstellung der Systemlösung.

Kapitel sieben und acht befassen sich mit einem Leistungsvergleich zwischen dem DESY-SVS und einer Kaufsoftware sowie einer Aufwands- und Kostenschätzung für die vollständige Entwicklung einer Systemkomponente zur Verwaltung des Schlüssel- und Zylinderbestandes.

Es folgt in Abschnitt neun das Ausblickkapitel, welches Möglichkeiten zur Weiterentwicklung des SVS aufzeigt und die weiteren Schritte nach Beendigung des Projektes benennt.

Den Abschluss dieser Diplomarbeit bildet das Fazit, welches das Projekt und seine Ergebnisse zusammenfasst und bewertet.

Ein Ausschnitt der Spezifikation, welche von der Autorin der Diplomarbeit erstellt wurde und einem Entwicklerteam als Basis für die technische Umsetzung der Systemkomponente diente, ist der vorliegenden Diplomarbeit als Anhang beigefügt.

2 Projekt

Die folgenden Abschnitte dieses Kapitels bieten einen Überblick über das Projekt zur Entwicklung einer Systemkomponente zur Unterstützung der Schlüsselverwaltungsprozesse am DESY, innerhalb dessen die vorliegende Diplomarbeit entstand. Zunächst wird der Anlass, welcher die Durchführung des Projektes begründet, erläutert. Im Weiteren werden Ziele und Nutzen des Projektes festgesteckt sowie dessen Ablauf skizziert. Zur Komplettierung des Überblicks werden abschließend die Beteiligten und ihre Rollen innerhalb des Projektes vorgestellt.

2.1 Anlass

Auf dem DESY-Campus in Hamburg-Bahrenfeld arbeiten zu Beginn des Projektes ca. 1.600 DESYaner. Hinzu kommen mehr als 3.000 Gastforscher sowie mehrere hundert Auszubildende, Diplomanden, Doktoranden und Nachwuchswissenschaftler

(vgl. DESY 2005¹). Die Zahl der Beschäftigten steigt stetig, ebenso wie die der Gebäude. In den nächsten Jahren werden unter anderem die neuen Experimentieranlagen PETRA III (Positron-Elektron-Tandem-Ring-Anlage III) und XFEL (X-ray free-electron laser) entstehen, für die aus sicherheitstechnischen Gründen besondere Zutrittsregelungen gelten werden. Aus all diesen Faktoren werden in Zukunft erhöhte Anforderungen an die Verwaltung der Gebäude- und Raumschlüssel sowie der Schließanlage resultieren – Anforderungen, die mit Hilfe der Methoden und Werkzeuge, die zu Beginn des Projektes angewendet wurden, nicht erfüllt werden könnten.

Bisher griffen die Verantwortlichen der DESY-Schließanlage und des Schlüssel- und Zylinderbestandes nicht auf zentrale Mitarbeiter- und Gebäude-/Raumdaten zurück, sondern führten diese in einem separaten Schlüsselverwaltungssystem. Gepflegt wurden diese Informationen auf Grund des hohen Aufwandes und da ihre Verwaltung außerhalb des Schlüsselverwaltungsbereiches liegt, nicht. Die Zuständigkeit für die Dokumentation und Pflege der Mitarbeiter- und Gebäudedaten liegt bei einer der Verwaltungsabteilungen. Der Datenbestand, auf den bei der Schlüsselverwaltung zurückgegriffen wurde, war somit inaktuell. Hinzu kommt, dass das Schlüsselverwaltungssystem keine Schnittstelle zu dem Geoinformations- und Facility-Management-System, mit Hilfe dessen die Pflege der Personal- und Gebäudedaten erfolgt, besaß. Dies erschwerte die Erfüllung der Anforderungen an die Schlüsselverwaltung.

Das GIS/FMS wurde vor einigen Jahren am DESY eingeführt. Die darin gespeicherten Informationen sind aktuell und der technische Reifegrad des Systems ermöglicht es, in Zukunft eine Komponente zur Verwaltung des Schlüssel- und Zylinderbestandes zu integrieren.

2.2 Ziel und Nutzen des Projektes

Das Ziel des Projektes ist die Entwicklung und Einführung einer neuen Systemkomponente für das GIS/FMS, die Funktionalitäten zur Unterstützung der Schlüsselverwaltungsprozesse bietet und einen Zugriff auf die zentralen Personal- und Gebäudedaten erlaubt. Mit der Erweiterung des GIS/FMS soll eine Unterstützung der Verwaltungsprozesse und eine Optimierung des Schlüsselmanagements erreicht werden. Die Spezifikation, welche die Anforderungen an die Schlüsselverwaltungsprozesse sowie die GIS/FMS-Erweiterung definiert und einem externen Entwicklerteam als Basis für die technische Umsetzung diente, wurde von der Autorin der vorliegenden Diplomarbeit entwickelt und ist dieser Arbeit in Auszügen beigelegt. Die Arbeit selbst verfolgt das Ziel, den Weg vom Beginn des Projektes bis hin zur Vorstellung einer Systemlösung zu dokumentieren.

Als Datenbasis für die künftige Verwaltung des Schlüssel- und Zylinderbestandes sollen die im GIS/FMS bereits vorhandenen Raum- und Personendaten dienen. Dies wird gegenüber der bisherigen Methode zu einer Verbesserung der Qualität der Verwaltungsdaten führen. Informationen zu Schlüssel- und Zylinderverbleib, bisher getrennt verwaltet, werden mit dem neuen SVS gepflegt und zentral einsehbar sein.

Durch die Integration der Schlüsselverwaltungsdaten in das GIS/FMS wird die Unterdisziplin des Facility Managements sowohl technisch als auch logisch in den FM-Bereich integriert. Gebäude-, Raum-, Personen-, Zylinder- und Schlüsselinformationen werden innerhalb des GIS/FMS miteinander verknüpft und ermöglichen einen schnellen und vollständigen Datenüberblick. Weiterhin soll durch die Zusammenfassung aller verwaltungsrelevanten Daten in einem Informationssystem gewährleistet werden, dass Mitarbeiter, schnell und unkompliziert in den Bereich der Schlüsselverwaltung eingeführt und im Umgang mit dem SVS geschult werden können, so dass für den Schlüsselverwalter, z.B. im Urlaubsfall, Ersatz bereit steht.

2.3 Projektablauf

Das Projekt wurde nach einem systematischen Ablaufplan durchgeführt (siehe Abb. 1). Das aus sechs Schritten bestehende Schema wird im Folgenden kurz vorgestellt. Eine detailliertere Erläuterung der einzelnen Projektphasen erfolgt in den jeweiligen Kapiteln dieser Diplomarbeit.

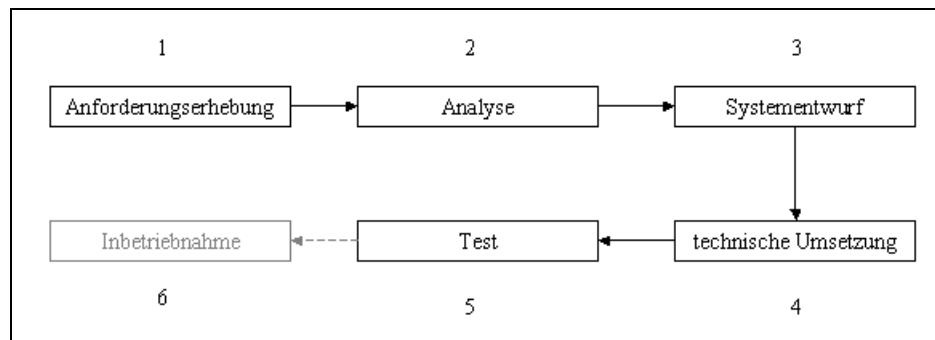


Abb. 1: Projektablauf (eigene Darstellung)

1) Anforderungserhebung

In der ersten Phase des Projektes werden Anforderungen an den Soll-Zustand der Schlüsselverwaltung und die zu entwickelnde Systemkomponente erhoben. Die Anforderungen definieren die Ziele des Projektes und werden in einer Spezifikation komprimiert.

2) Analyse

Im zweiten Schritt folgt die Analysephase. Hier werden die Prozesse der Schlüsselverwaltung in ihre Aktionen zerlegt und eingehend betrachtet. Diese Aufschlüsselung dient als Basis für die nachfolgende Projektphase – den Systementwurf.

3) Systementwurf

Innerhalb dieses Abschnittes wird die Architektur des zu entwickelnden Teilsystems ausgearbeitet. Das Ergebnis dieser Entwurfsphase stellt die Vorlage für den vierten Projektschritt – die technische Umsetzung – dar.

4) Technische Umsetzung

In dieser Phase des Projektes wird an Hand der Spezifikation aus Schritt eins und des Systementwurfs aus Schritt drei die Komponente zur Verwaltung des Schlüssel- und Zylinderbestandes technisch durch ein externes Entwicklerteam realisiert.

5) Systemtest

Im Anschluss an die Entwicklung der Systemkomponente hat ein Systemtest zu erfolgen. Hier werden die Funktionalitäten des Produktes überprüft und gegebenenfalls Korrekturen und Anpassungen vorgenommen.

6) Inbetriebnahme

Nach einer ausführlichen Testphase kann die Systemkomponente in Betrieb genommen werden. Diese Projektphase ist jedoch kein Bestandteil der vorliegenden Diplomarbeit mehr und wird daher in der erläuternden Grafik in Abb. 1 ausgegraut.

2.4 Projektbeteiligte

Das Projekt wurde von der Abteilung V1 (Verwaltung 1) des Deutschen Elektronen-Synchrotrons in Auftrag gegeben und unter der Leitung der Autorin dieser Diplomarbeit durchgeführt. Unterstützend standen der Diplomandin in allen Projektphasen Mitglieder der Gruppe IPP (Informationsmanagement, Prozesse, Projekte) zur Seite. Mitarbeiter der Firma Horstick waren als externe Entwickler für die technische Umsetzung der FMS-Erweiterung verantwortlich.

Die Abteilungsmitglieder von V1 informierten die Auftragnehmer der Gruppe IPP über die Abläufe der Schlüsselverwaltung am DESY und entwickelten Anforderungen an das zu realisierende Schlüsselverwaltungssystem. Die Erhebung der Anforderungen erfolgte durch die Autorin der Diplomarbeit, die von den Projektmitgliedern der Gruppe IPP darin unterstützt wurde. Ebenfalls in dieser Konstellation wurde in der Analyse- und Entwurfsphase zusammengearbeitet. Die Entwickler der Firma Horstick setzten die Systemkomponente zur Verwaltung des Schlüssel- und Zylinderbestandes technisch auf Basis der DESY-Spezifikation, erstellt von der Autorin der vorliegenden Diplomarbeit, um. Das Entwicklerteam stellte dem DESY-Projektteam eine Testversion des SVS bereit, welche von der Diplomandin und Mitgliedern der Gruppe IPP auf ihre Funktionalitäten geprüft wurde. Daten zur Durch-

führung des Systemtests stellten die projektbeteiligten Mitglieder von V1 zur Verfügung.

3 Grundlagen

Das nachfolgende Kapitel beschreibt die Grundlagen derjenigen Themenbereiche, die einen Bestandteil der vorliegenden Diplomarbeit darstellen. Zunächst werden das Deutsche Elektronen-Synchrotron und seine Kerntätigkeiten skizziert. Im Weiteren wird erläutert, inwiefern Prozess durch den Einsatz von Informationssystemen unterstützt werden können. Es folgt eine Definition und Beschreibung des Facility Managements. Eng mit diesem Thema ist die Notwendigkeit von Informationssystemen – CAFM-Systemen – zur Durchführung von FM-Aufgaben verbunden. Das Teilkapitel zum GIS/FMS am DESY liefert hierzu ein Praxisbeispiel. Abschließend folgt eine Definition zum Thema Schlüsselverwaltung.

3.1 DESY

Das Deutsche Elektronen-Synchrotron ist ein Forschungszentrum der Helmholtz-Gemeinschaft mit Standorten in Hamburg und Zeuthen (Brandenburg). Es wurde 1959 gegründet und entwickelt und baut Beschleunigeranlagen, die für die Forschung im Bereich der Teilchenphysik und für die Forschung mit Photonen in Betrieb genommen werden. Wissenschaftler aus aller Welt – jährlich ca. 3.000 Gastforscher aus 45 Nationen – arbeiten mit DESY zusammen. Des Weiteren beteiligt sich das deutsche Forschungszentrum an internationalen Projekten, wie z.B. dem europäischen Röntgenlaser XFEL oder dem Internationalen Linearcollider ILC (vgl. DESY 2007).



Abb. 2: Luftbild des DESY-Geländes (Quelle: DESY 2006)

Forschung wird bei DESY vor Ort betrieben. Unter dem Werksgelände befindet sich neben Experimentieranlagen wie PETRA und FLASH (Freier Elektronen Laser Hamburg) unter anderem ein Teil des sechs Kilometer langen Beschleunigerrings HERA (Hadron-Elektron-Ring-Anlage) (siehe Abb. 2), der im Juni 2007 abgeschaltet wurde. Hier wurden seit 1992 Experimente aus dem Bereich der Teilchenphysik

durchgeführt. Die Auswertung der Ergebnisse wird über das Jahr 2010 hinaus andauern (vgl. DESY 2007).

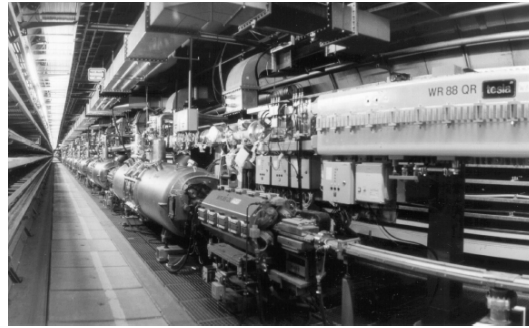


Abb. 3: Blick in den Hera-Tunnel (Quelle: DESY 2005²)

Bis 2013 werden unter dem DESY-Werksgelände zwei weitere Experimentieranlagen entstehen – PETRA III (Inbetriebnahme 2009) und der XFEL (Inbetriebnahme 2013). Die Anlage des XFEL wird mit einer geplanten Länge von ca. drei Kilometern unterirdisch von Hamburg bis nach Schleswig-Holstein reichen und den Wissenschaftlern bei DESY neben PETRA III zur Forschung mit Photonen dienen (vgl. DESY 2007).

3.2 Prozessunterstützung durch Informationssysteme

Das Ziel dieses Projektes ist die Einführung eines Informationssystems zur Unterstützung der Schlüsselverwaltungsprozesse am DESY. Mit dieser Zieldefinition wird gleichzeitig die These aufgestellt, dass Informationssysteme in der Lage sind, die Durchführung von Prozessen zu unterstützen. Bevor im Folgenden diese These auf ihre Belegbarkeit geprüft wird, müssen zunächst einige Grundlagen erklärt werden.

Prozess

Ein Prozess besteht aus einer Menge von Aktivitäten, die parallel oder nacheinander ablaufen, um ein definiertes Ziel zu erreichen (vgl. Grässle/Baumann/Baumann 2007, S. 45 ff). Jeder Prozess hat einen Beteiligten bzw. Akteur oder mehr, welcher diese Aktivitäten durchführt (vgl. Jacobson/Booch/Rumbaugh 2001, S. 25). Die Aktivitäten eines Prozesses können sowohl durch Menschen als auch durch Informationssysteme umgesetzt werden (siehe Abb. 4).

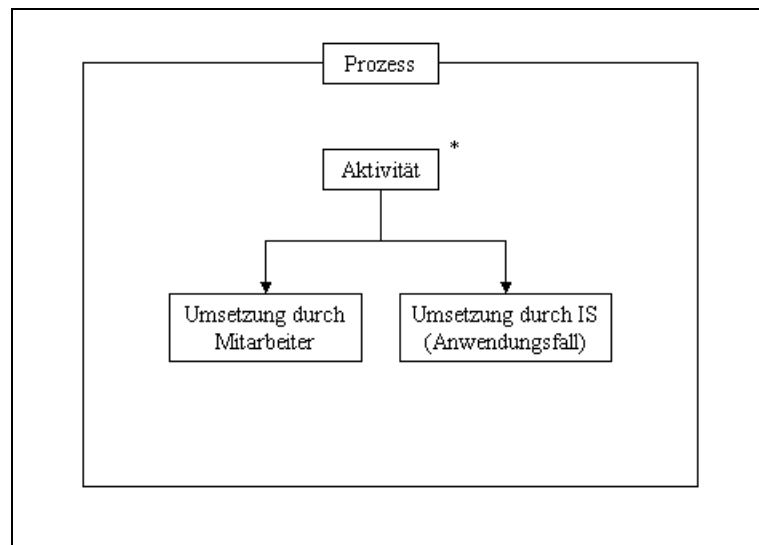


Abb. 4: Umsetzungsmöglichkeiten von Aktivitäten (eigene Darstellung)

Informationssystem

Ein Informationssystem ist ein „System zur Speicherung, Wiedergewinnung, Verknüpfung und Auswertung von Informationen“ (zit. Grässle/Baumann/Baumann 2007, S. 277). Es wird eingesetzt, um statt eines Menschen Aktivitäten auszuführen und um somit die Abwicklung eines Prozesses zu unterstützen. Die Aktionen, die ein Informationssystem durchführt, werden Anwendungsfälle genannt.

Anwendungsfall

Booch, Jacobson und Rumbaugh definieren einen Anwendungsfall als: „A description of a set of sequence actions, including variants, that a system performs that yields an observable result of value to a particular actor“ (zit. Booch/Jacobson/Rumbaugh 2001, S. 432). Ein Anwendungsfall ist demnach eine Abfolge von Aktionen, die ein System durchführt, um einem Anwender ein wahrnehmbares Resultat zu liefern. Eine spezifische Abfolge der Aktionen innerhalb eines Anwendungsfalls wird als Szenario bezeichnet (vgl. Grässle/Baumann/Baumann 2007, S. 286).

Informationssysteme werden demnach eingesetzt, um Akteure bei der Durchführung von Prozessen zu unterstützen und an ihrer statt Aktionen auszuführen. Dieser Automatismus kann in einem Unternehmen dazu führen, dass Mitarbeiter „überflüssig“ und letztendlich von einem Informationssystem verdrängt werden. Diese Einsparung an Personal kann im Endeffekt einen positiven Einfluss auf die Wirtschaftlichkeit ausüben und zu einer Kostenreduktion führen.

Informationssysteme können für unterschiedliche Zwecke betrieben werden. Bei DE-SY wird z.B. das GIS/FMS zur Dokumentation von Geoinformationen und FM-Daten

eingesetzt, während SAP bei der Abwicklung betriebswirtschaftlicher Aufgaben Anwendung findet.

3.3 Facility Management

Facility Management ist eine Management-Disziplin, die eine strategische „Bewirtschaftung, Verwaltung und Organisation aller Sachressourcen innerhalb eines Unternehmens“ zum Ziel hat (zit. Nävy 2003, S. 3). Ohne FM werden diese Sachressourcen (z.B. Grundstücke, Gebäude, Maschinen) oft unter Ausschluss der für sie verantwortlichen Abteilungen betreut. Facility Management hingegen sorgt dafür, dass die Ressourcen zu einer ganzheitlichen Betreuung zusammengeführt werden, was einen zentralen Zugriff auf alle FM-relevanten Informationen innerhalb einer Organisation ermöglicht (vgl. Nävy 2003, S. 3 ff). Diese Transparenz spielt im Facility Management eine entscheidende Rolle. Indem alle Informationen zentral verfügbar und stets aktuell sind, wird erst die Bewirtschaftung, Verwaltung und Organisation der Sachressourcen möglich (vgl. Nävy 2003, S. 6).

Sachressourcen durchlaufen während ihres Lebenszyklus verschiedene Phasen (siehe Abb. 5). Facility Management findet hierbei in jeder einzelnen Lebensphase statt – von der Idee über die Planung, Erstellung, Nutzung bis hin zur Entsorgung der Ressource (vgl. Nävy 2003, S. 5).

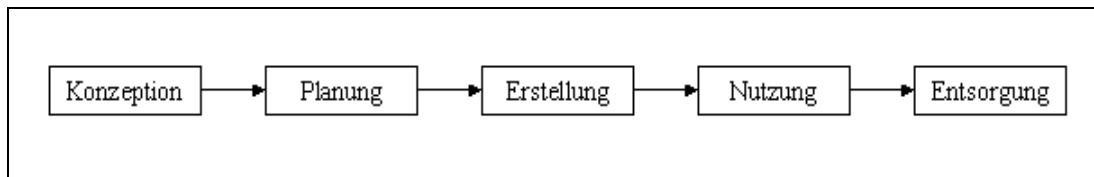


Abb. 5: Lebensphasen einer Sachressource (nach Nävy 2003, S. 72)

Das Konzept des Facility Managements wurde Mitte der 50er Jahre geboren und seitdem stets weiterentwickelt. Ende der 70er Jahre kam FM erstmals in den USA aktiv zum Einsatz. Nach Deutschland gelangte die Management-Disziplin Ende der 80er Jahre mit der Gründung der GEFMA (German Facility Management Association). Sie hat das Ziel, die Praktizierung des Facility Managements in den Unternehmen der Bundesrepublik zu fördern (vgl. Nävy 2003, S. 40 ff).

Unterteilt man Facility Management in seine Unterdisziplinen, ergibt sich ein vielfältiges Bild an Aufgaben. So umfasst FM nicht nur das Management von Gebäuden und Flächen, sondern auch der Kommunikationsanlagen. Abb. 6 stellt diese und weitere Aspekte des Facility Managements grafisch dar.

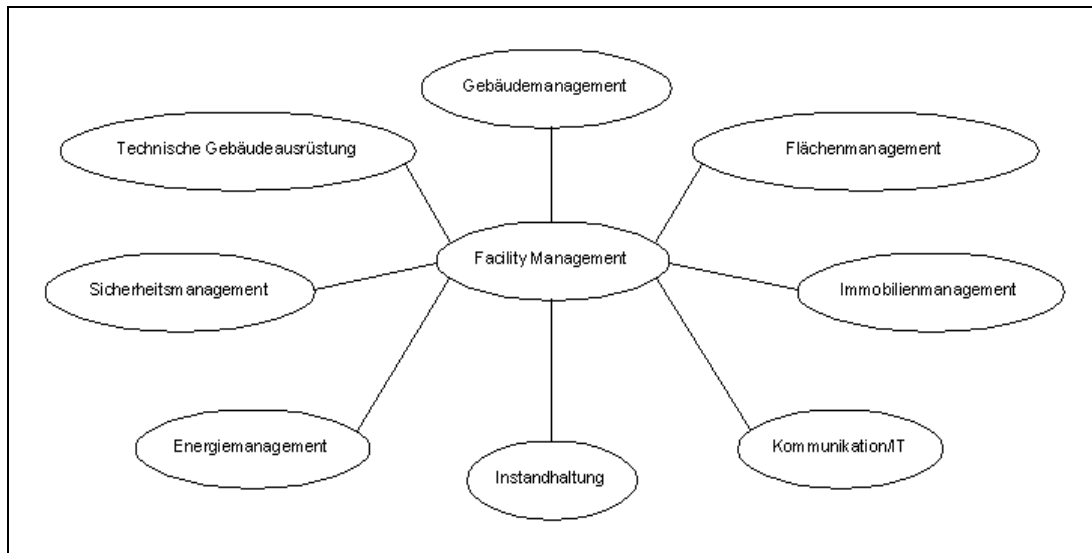


Abb. 6: Aspekte des Facility Managements (nach Nävy 2003, S. 4)

3.4 CAFM-Systeme

Um Facility Manager bei der Erfüllung der in Abb. 6 dargestellten Aufgaben zu unterstützen, werden Informationssysteme eingesetzt. Diese Aufgabenbewältigung mit Hilfe von Computern verleiht FM auch den Namen *Computer Aided Facility Management* (CAFM). Abgeleitet hiervon werden die FM-unterstützenden Informationssysteme auch *CAFM-Systeme* genannt. Diese begleiten die Sachressourcen durch alle Phasen ihres Lebenszyklus und ermöglichen allen Beteiligten durch die zentrale Speicherung aller FM-relevanten Informationen einen zeitgleichen Zugriff auf redundanzfreie Daten (vgl. Nävy 2003, S. 67 ff).

CAFM-Systeme setzen sich hauptsächlich aus zwei Komponenten zusammen: auf der einen Seite aus einer Datenbank, die alphanumerische Daten speichert und auf der anderen Seite aus dem Computer Aided Design (CAD), welches für die Verarbeitung von grafischen Elementen verantwortlich ist (siehe Abb. 7).

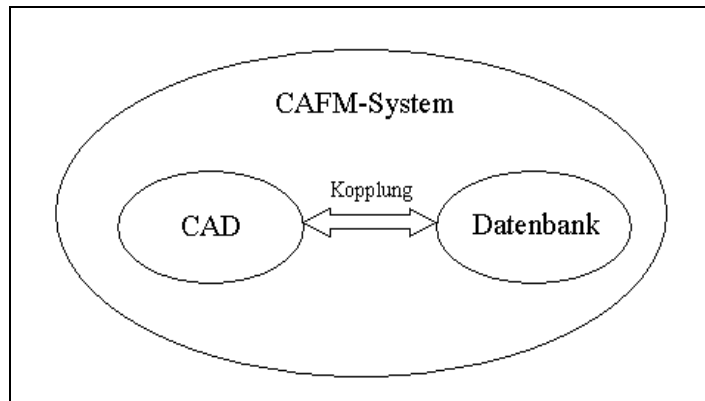


Abb. 7: Komponenten eines CAFM-Systems (nach Nävy 2003, S. 111)

Beide Komponenten sind miteinander verknüpft, so dass eine Änderung der grafischen Daten im CAD gleichzeitig eine Aktualisierung der alphanumerischen in der Datenbank nach sich zieht. Dies garantiert ein Arbeiten mit redundanzfreien Daten und reduziert das Fehleraufkommen (vgl. Nävy 2003, S. 69 ff).

3.5 GIS/FMS – das CAFM-System am DESY

Bei DESY werden die Aufgaben des Facility Managements in Unterstützung mit dem GIS/FMS umgesetzt. Es ist ein kombiniertes Geoinformations- und Facility-Management-System und wird bei DESY von verschiedenen Abteilungen für unterschiedliche Zwecke verwendet – unter anderem zur Planung und Dokumentation von Gebäuden und Anlagen. Das GIS/FMS beherbergt Sachdaten sowie Bauzeichnungen, Übersichtskarten und Landkarten. Alphanumerische und geografische Daten sind hierbei in getrennten Datenbanken abgelegt. Des Weiteren ist in dem GIS/FMS das sogenannte Raumbuch integriert, welches Flächenmanagement (z.B. Berechnung von Reinigungsflächen oder freier Flächen für die Aufstellung neuer Experimentierstände) ermöglicht (vgl. Eucker 2005, S. 14 ff).

Den technischen Aufbau des GIS/FMS beschreibt Abb. 8. Sach- oder geografische Daten werden durch die Fachanwender in die jeweilige Benutzeroberfläche eingetragen. Zu den Systemnutzern gehören unter anderem die Bauabteilung oder die Verwaltung. Gespeichert werden die Daten über das DESY-Intranet in der jeweiligen Datenbank. Auch der Abruf der im GIS/FMS dokumentierten Informationen ist ausschließlich über das DESY-Intranet möglich.

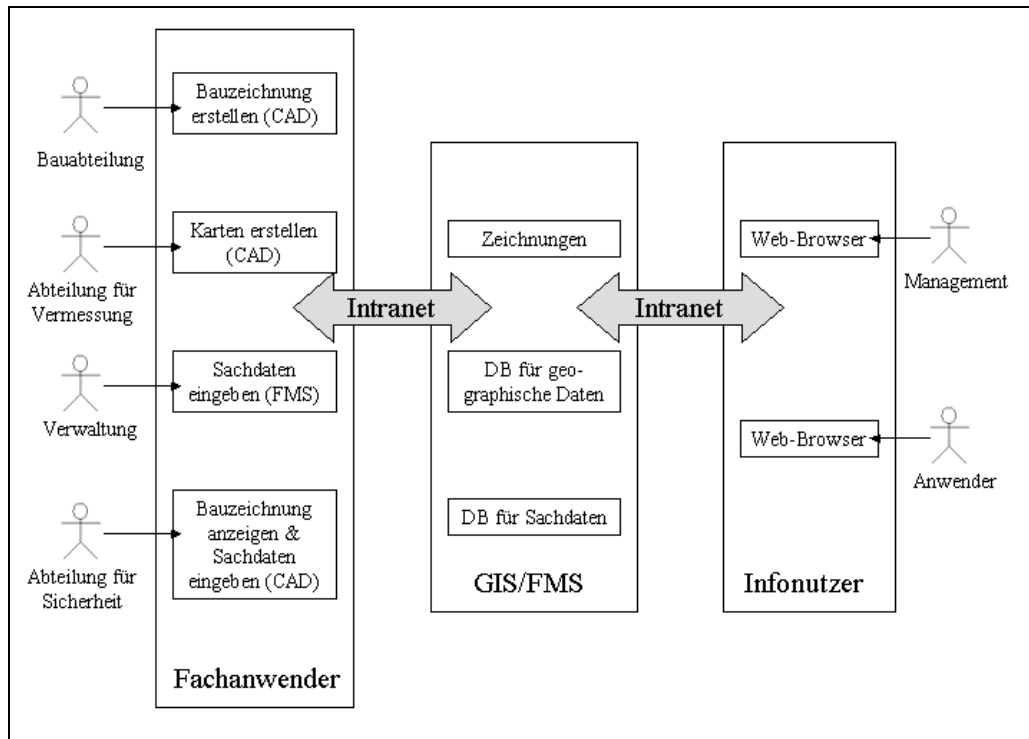


Abb. 8: GIS/FMS-Architektur (nach Eucker 2005, S. 15; Robben 2004, S. 14)

Das GIS/FMS basiert auf Standardsystemkomponenten, die im Auftrag von DESY um Funktionen ergänzt wurden. Die technische Umsetzung der von DESY spezifizierten Zusatzfunktionen erfolgte unter anderem von der Firma Horstick (vgl. Eucker 2005, S. 16).

3.6 Schlüsselverwaltung

Eine der Teilaufgaben des Facility Managements ist die Schlüsselverwaltung. Sie zählt zum Sicherheitsmanagement, ist jedoch auch eng mit dem Gebäudemanagement verknüpft (vgl. Nävy 2003, S. 254). Der Bereich der Schlüsselverwaltung lässt sich wiederum in weitere Unteraufgaben teilen. Im Fall des Schlüsselmanagements bei DESY wurden in Absprache mit den Verantwortlichen dieses Arbeitsbereiches sieben Hauptaufgaben zu seiner Erfüllung definiert (siehe Abb. 9). Dazu gehören unter anderem die Erfassung von Schlüssel- und Zylinderstammdaten sowie die Ausgabe und Rücknahme von Schlüsseln und Zylindern. Um die Schlüsselverwalter in der Durchführung dieser Aufgaben zu unterstützen, werden auch in diesem Bereich des Facility Managements CAFM-Systeme eingesetzt (vgl. Nävy 2003, S. 256).

Abb. 9 fasst den Bereich der Schlüsselverwaltung, wie DESY ihn definiert, grafisch zusammen.

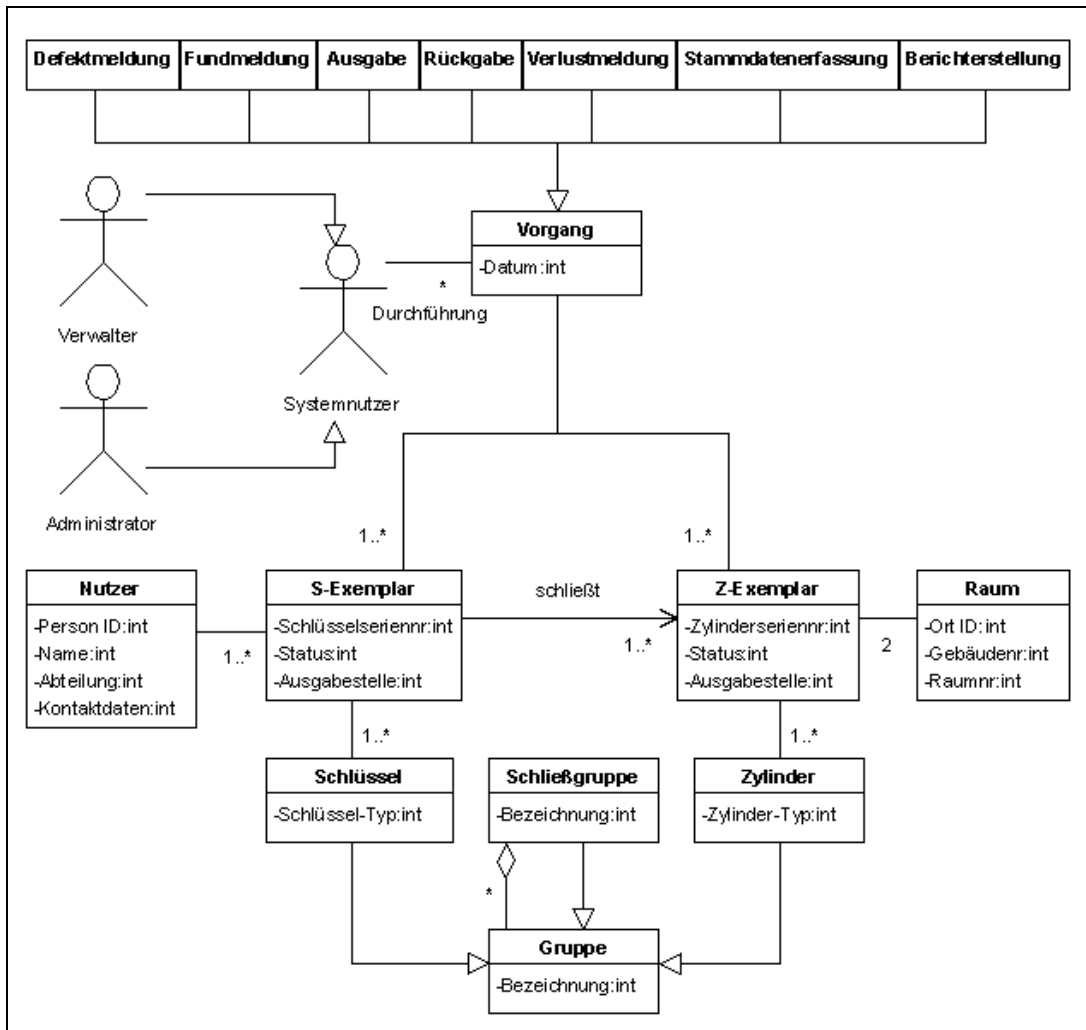


Abb. 9: DESY-Konzept der Schlüsselverwaltung (eigene Darstellung)

Die Aufgaben bzw. Vorgänge der Schlüsselverwaltung werden von einem Systemnutzer durchgeführt. Dieser kann sowohl ein Verwalter als auch ein Administrator sein. Ein Vorgang kann sich sowohl auf Schlüssel- als auch auf Zylinderexemplare beziehen. Beide besitzen eine Seriennummer, einen Status sowie eine Ausgabestelle – Informationen, die individuell für ein Exemplar gelten. Schlüsselexemplare können zu einem Schlüssel, Zylinderexemplare zu einem Zylinder zusammengefasst werden. Diese Gruppierung ermöglicht die Zusammenfassung gleichartiger Schlüssel und Zylinder zu einem Typen (z.B. Einzelschlüssel oder Einzelzylinder). Je nach Bedarf können somit Hierarchien gebildet werden, bei denen der Elternschlüssel die Schließereigenschaften aller seiner Kinderschlüssel vereint. Den Kopf der DESY-Hierarchie stellt der Generalschlüssel, der alle Zylinder der DESY-Anlage schließen kann. Unter ihm folgen Obergruppen-, Gruppen-, Einzel- und Zentralschlüssel, die – von oben nach unten innerhalb der Schließhierarchie – einzelne, immer kleinere Schließkreise schließen.

Diese Definition setzt voraus, dass ein Schlüsselexemplar ein Zylinderexemplar oder mehrere schließen kann. Weiterhin legt das DESY-Modell der Schlüsselverwaltung fest, dass ein Nutzer ein Schlüsselexemplar oder mehr besitzen kann und ein Zylinderexemplar zwei Räumen zugeordnet ist.

Diese Verknüpfungen zwischen Schlüssel- und Zylinderexemplar, Zylinderexemplar und Raum sowie Schlüsselexemplar und Nutzer sind notwendig, um eine korrekte und umfassende Durchführung der Schlüsselverwaltungsaufgaben zu gewährleisten. Erst mit der Vernetzung dieser Informationen ist z.B. eine Überwachung von Zutrittsregelungen möglich (vgl. May 2004, S. 30).

Neben der Organisation von Raumschlüsseln gehört zum Schlüsselmanagement auch die Verwaltung von Schlüsseln für Spinde, Tore, Fenster, Alarmkästen etc. Im Fall des Projektes am DESY einigte man sich jedoch darauf, zunächst nur die Verwaltung von Raumschlüsseln anzustreben.

4 Werkzeuge und Methoden

Um einen Lösungsansatz für die GIS/FMS-Erweiterung zu entwickeln, wurden innerhalb der ersten drei Projektphasen (siehe Abb. 1, Kapitel 2.3) verschiedene Werkzeuge verwendet und Methoden angewandt. Ziel dieses Kapitels ist es, diese vorzustellen und zu erläutern. Zur Bewältigung der ersten Projektphase werden zunächst die Grundlagen zur Erhebung von Anforderungen und deren Management vorgestellt. Als Methode zur Anforderungserhebung wurde die Metaplan-Technik angewandt, die ebenfalls innerhalb dieses Kapitels umrissen wird. Weiterhin werden das Vorgehen zur Durchführung einer Prozessanalyse sowie Grundlagen zur Entwicklung eines Systementwurfes geschildert. Als Werkzeug zur Analyse der Schlüsselverwaltungsprozesse fand die Unified Modelling Language (UML) Anwendung. Der entsprechende Kapitelabschnitt definiert die UML und stellt einzelne ihrer Elemente vor. Ebenfalls Bestandteil dieses Kapitels ist eine Erläuterung zur Benchmarking-Methode, welche innerhalb des Projektes zur Leistungsbewertung der entwickelten Schlüsselverwaltungskomponente angewandt wurde. Außerdem wird die Funktionspunktanalyse – ein Hilfsmittel zur Abschätzung des Arbeitsaufwandes sowie der Kosten eines Projektes – vorgestellt.

4.1 Anforderungserhebung und -management

Um mit dem Ergebnis des Projektes den Wünschen des Auftraggebers gerecht zu werden, müssen zu Beginn ihre Anforderungen an das zu entwickelnde Produkt erhoben werden. Das folgende Unterkapitel geht zunächst näher auf den Begriff der Anforderung ein. Anschließend werden die einzelnen Phasen der Anforderungsentwicklung vorgestellt.

4.1.1 Was ist eine Anforderung?

Bevor auf das Thema des Anforderungsmanagements eingegangen wird, muss zunächst der Begriff *Anforderung* erklärt werden. Nach Chris Rupp ist eine Anforderung „eine Aussage über eine zu erfüllende Eigenschaft oder zu erbringende Leistung eines Produktes, eines Prozesses oder der am Prozess beteiligten Personen“ (zit. Rupp 2002, S.13). Anforderungen an ein Produkt können sich zum Beispiel an ein Informationssystem richten. Diese Systemanforderungen sind wiederum in zwei weitere Kategorien zu unterteilen:

- funktionale Anforderungen
- nicht-funktionale Anforderungen

Funktionale Anforderungen beschreiben, welche Aktionen ein System durchführen soll (vgl. Rupp 2002, S. 173). Hierbei wird nicht darauf eingegangen, WIE ein System etwas leisten soll, sondern WAS das System zu leisten hat. Nicht-funktionale Anforderungen hingegen machen eine Aussage bezüglich der Systemeigenschaften – das WIE. Mit ihnen können Vorgaben hinsichtlich Entwicklungsumgebung oder zu verwendender Programmiersprachen getroffen werden (vgl. Booch/Jacobson/Rumbaugh 2001, S. 447). Den Projektbeteiligten dienen Anforderungen vor allem als Kommunikations- und Wissensgrundlage.

4.1.2 Phasen der Anforderungsentwicklung

Bevor mit der Entwicklung von Anforderungen begonnen werden kann, müssen dem Projekt zunächst klare Ziele gesetzt werden. Erst mit definierten Projektzielen vor Augen können Anforderungen zwecksgerichtet erhoben werden (vgl. Rupp 2002, S. 20).

Der Prozess der Anforderungsentwicklung wird auch Requirements-Engineering-Process (REP) genannt. Dieses Verfahren kapselt eine Reihe von Aktivitäten, die den Projektbeteiligten dabei helfen, die zu erbringenden Leistungen eines Systems herauszukristallisieren und diese in einem Anforderungsdokument zusammenzufassen (vgl. Sommerville 2005, S. 16). Der Prozess der Anforderungsentwicklung unterteilt sich nach Ian Sommerville in sechs Phasen (siehe Abb. 10):

- Anforderungserhebung
- Analyse der Anforderungen
- Bestätigung der Anforderungen durch die Auftraggeber
- Verhandlung
- Formulierung der Anforderungen / Dokumentationsphase
- Management der Anforderungsänderungen

(nach Sommerville 2005, S.16 ff)

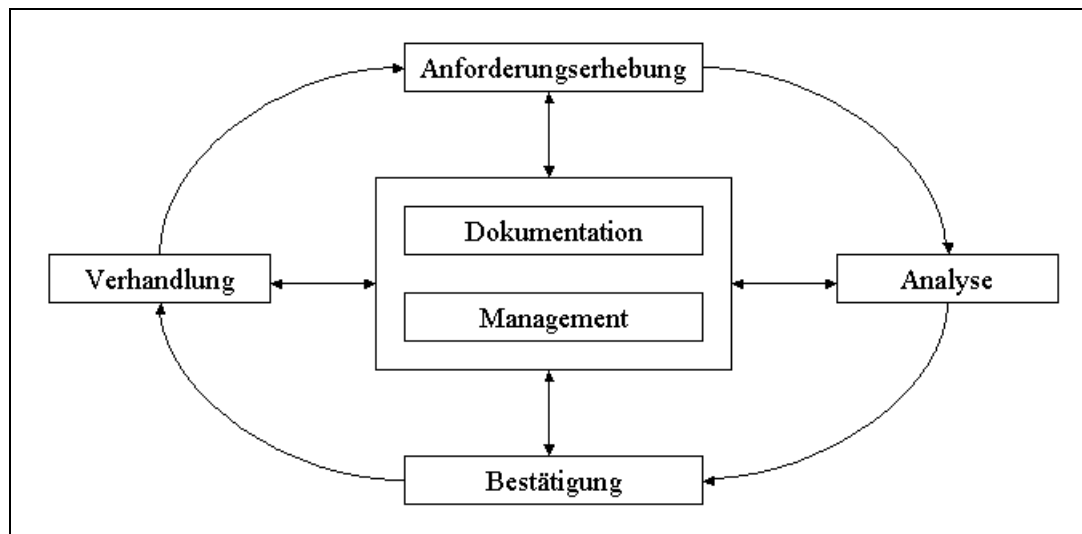


Abb. 10: Phasen der Anforderungsentwicklung (nach Sommerville 2005, S. 17)

Sommerville betont, dass es sich bei der Entwicklung der Anforderungen um einen iterativen Prozess handelt, bei dem die einzelnen Phasen mehrmals durchlaufen werden (vgl. Sommerville 2005, S. 17). Die Darstellung des REP in Abb. 10 unterstreicht diese These.

In den folgenden Abschnitten wird näher auf die einzelnen Phasen der Anforderungsentwicklung eingegangen.

Anforderungserhebung

Die Erhebung von Anforderungen kann auf unterschiedliche Arten und mit Hilfe verschiedener Methoden erfolgen. Welche hierbei die „richtige“ bzw. effektivste ist, hängt von Faktoren wie Zeit, Geld und Motivation der Kunden ab (vgl. Rupp 2002, S. 92). Von Interviews über Brainstorming-Verfahren bis hin zu Fragebögen oder Feldbeobachtungen ist hier vieles möglich. Je nach Projektsituation sollte eine geeignete Technik gewählt werden. Das angewandte Erhebungsverfahren innerhalb dieses Projektes wird in Kapitel 4.2 erläutert.

Analyse

In der Analysephase werden die Zusammenhänge zwischen den erhobenen Anforderungen hergestellt. Diese Betrachtung erfolgt oftmals mit Hilfe einer grafischen Darstellung. Durch die Modellierung der Anforderungen ist es möglich, die Qualität der Spezifikation aufzudecken. Die beiden Prozesse *Anforderungserhebung* und *Modellierung* sollten daher parallel stattfinden (vgl. Rupp 2002, S. 68 ff). Damit bei Änderungen im Laufe des Projektes sowohl die Anforderungen als auch die entsprechenden Punkte im Modell angepasst werden, muss eine bidirektionale Verbindung zwischen beiden bestehen. Widersprüchen und Redundanzen kann so entgegengesteuert werden (vgl. Rupp 2002, S. 70).

Bestätigung und Verhandlung

In der Bestätigungsphase werden die Anforderungen den Auftraggebern präsentiert, um sich ihrer Korrektheit, Verständlichkeit und Vollständigkeit zu vergewissern. Dies stellt sicher, dass Analytiker und Auftraggeber die selben Ziele verfolgen und dass Missverständnisse zwischen beiden Parteien ausgeräumt werden (vgl. Rupp 2002, S. 120; S. 122). In der Verhandlungsphase geht es darum, aus den unterschiedlichen Ansichten und Ansprüchen der Auftraggeber eine gemeinsame Linie zu bilden (vgl. Sommerville 2005, S. 17).

Dokumentation

Die Menge der Anforderungen, die auf diesem Wege innerhalb eines Projektes formuliert werden, bilden ein Anforderungsdokument bzw. eine Spezifikation. Diese dient während des Projektes unter anderem für eine abschließende Produktabnahme, z.B. für einen Systemtest. Verhält sich das System gemäß einer Anforderung, kann diese als erfüllt festgehalten werden. Die Spezifikation dient jedoch nicht nur als Anleitung für einen Systemtest. Auch Systemänderungen erfolgen ausschließlich auf Basis einer abgeänderten Spezifikation (vgl. Rupp 2002, S. 14).

Um zu gewährleisten, dass der Anforderungskatalog qualitativ hochwertig ist, sind an jede Anforderung Qualitätsansprüche zu stellen. Einige dieser Kriterien sind im Folgenden aufgelistet:

- Vollständigkeit
- Korrektheit
- Konsistenz
- Testbarkeit
- Gültigkeit
- Verstehbarkeit
- Umsetzbarkeit
- Notwendigkeit
- Eindeutigkeit

(vgl. Rupp 2002, S. 24)

Anforderungen haben allerdings nicht nur den oben aufgelisteten Qualitätskriterien zu genügen, sondern auch einer vorgeschriebenen Satzkonstruktion zu entsprechen. Diese syntaktische Anforderungsschablone, so Chris Rupp, reduziert Formulierungsfehler und „ermöglicht es, jeder Anforderung eine ähnliche Struktur aufzuprägen“ (zit. Rupp 2002, S. 232).

Abb. 11 stellt ein Muster zur Formulierung von Anforderungen dar. Hierbei wird deutlich, dass eine Anforderung verschiedene Verbindlichkeitsgrade aufweisen kann. Eine Pflicht wird durch MUSS, ein Wunsch durch SOLL und eine Absicht durch WIRD ausgedrückt. Des Weiteren kann eine Anforderung nutzer- bzw. systemorientiert sein.

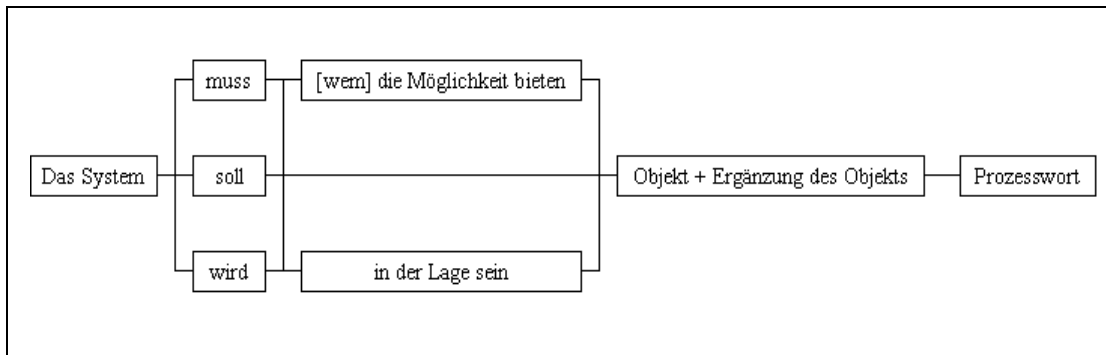


Abb. 11: Anforderungsschablone (nach Rupp 2002, S. 471)

Vollständige Anforderungen gemäß der Schablone in Abb. 11 können folgendermaßen aussehen:

Das Schlüsselverwaltungssystem muss dem Nutzer die Möglichkeit bieten, verfügbare Schlüsselexemplare zu identifizieren.

Das Schlüsselverwaltungssystem soll in der Lage sein, Daten zu importieren.

Anforderungen können außerdem unter Einbeziehung von Bedingungen formuliert werden. Eine entsprechende Schablone findet sich in Abb. 12.

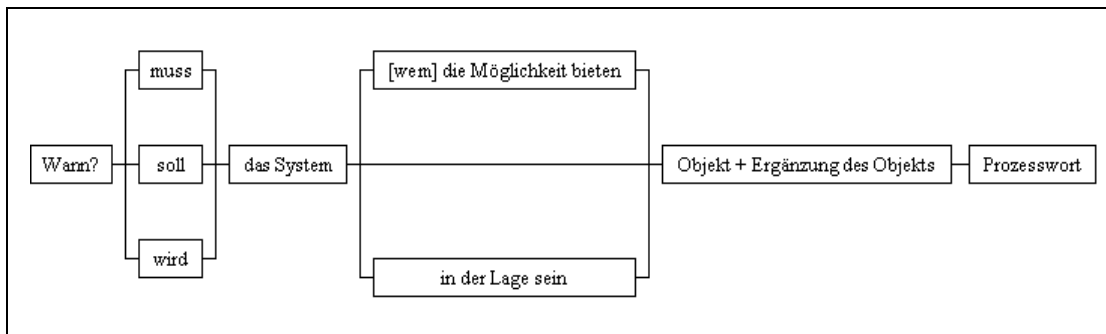


Abb. 12: Anforderungsschablone mit Bedingung (nach Rupp 2002, S. 472)

Folgende Anforderungen dienen als Beispiel für eine Formulierung mit Hilfe der Anforderungsschablone in Abb. 12:

Nach der Speicherung der Prozessdaten soll der Schlüsselausgabe-Wizard dem Systemnutzer die Möglichkeit bieten, sich eine Übersicht des durchgeführten Vorgangs anzeigen zu lassen.

Wurde ein Prozess erfolgreich durchgeführt, soll das System eine Übersicht des durchgeführten Vorgangs ausdrucken.

Beide Schablonen gelten nur für die Formulierung von funktionalen und nicht-funktionalen Anforderungen. Für das Ausformulieren von Anforderungen an Prozes-

se oder für an einem Prozess beteiligte Personen müssen andere Muster herangezogen werden.

Management

Da während eines Projektes oftmals große Mengen an Anforderungen entwickelt werden, ist ein definiertes Verfahren notwendig, um diese beherrschbar zu machen. An diesem Punkt setzt das Anforderungsmanagement an. „[...] es ermöglicht, die oftmals unüberschaubare Anzahl an Anforderungen komplexer Projekte zu beherrschen“ (zit. Rupp 2002, S. 15). Anforderungsmanagement dient zur Vermeidung von Fehlern in der Analysephase eines Projektes und erhöht die Qualität der Anforderungen (vgl. Rupp 2002, S. 12 ff). Die Voraussetzung für eine hohe Qualität der Anforderungen setzt jedoch voraus, dass auch dessen Änderungen – gerade in komplexen Projekten – bewältigt werden. Hierzu bedient man sich Requirements-Management-Techniken. Die richtige Herangehensweise ist hierbei entscheidend darüber, ob die Projektbeteiligten trotz fortlaufender Änderungen an dem Anforderungsdokument stets auf die selbe Arbeitsgrundlage zurückgreifen. Anforderungsmanagement ermöglicht ein paralleles Arbeiten aller Projektbeteiligten mit der Spezifikation und garantiert einen einheitlichen Informationsstand (vgl. Rupp 2002, S. 15). Professionelles Anforderungsmanagement kann demnach maßgeblich zu einem erfolgreichen Projektabschluss beitragen (vgl. Rupp 2002, S. 12 ff).

4.2 Anforderungserhebung mit der Metaplan-Technik

Innerhalb des DESY-Projektes zur Einführung eines SVS wurde zur Anforderungserhebung die Metaplan-Technik angewandt. Sie ist jedoch nicht nur für die Erhebung von Anforderungen geeignet, sondern als moderierte Gruppenarbeitsmethode vielfältig einsetzbar. Mit Hilfe einer Kartenabfrage werden Ideen und Aspekte zu einem vorgegebenen Thema gesammelt. Den Teilnehmern der Arbeitsgruppe werden Aufgaben bzw. Fragen präsentiert, die in einem festgesteckten Zeitrahmen zu bearbeiten sind. Jedes Gruppenmitglied notiert seine Ideen auf Karten. Dabei ist zu beachten, dass auf jeder Karte stets nur ein Gedanke festgehalten wird. Am Ende der Bearbeitungszeit werden die einzelnen Ideen vorgelesen und zur Gruppendiskussion freigegeben. Ist ein Gedanke nicht für alle Arbeitsgruppenteilnehmer nachvollziehbar, wird er von dem Verfasser erläutert. Die Karten werden an eine Pinwand gesteckt und unter Einbeziehung aller Teilnehmer thematisch sortiert. Der komplette Verlauf des Workshops wird von einem Moderator koordiniert und begleitet. Diese Arbeitsmethode dient dazu, alle Gruppenmitglieder in die Ideenfindung zu involvieren und ein möglichst breites Spektrum an Ideen und Meinungen zu erzielen (vgl. Donert/Sterzenbach 1999, S. 74-76).

4.3 Analyse

Im Anschluss an die Anforderungserhebung folgt die Analyse. In dieser Projektphase findet eine tiefere Auseinandersetzung mit den Anforderungen statt, was dabei hilft, einen detaillierteren Blick auf den Problembereich zu erhalten und diesen zu verstehen (vgl. Booch/Jacobson/Rumbaugh 2001, S. 174; Hitz/Kappel 1999, S. 178). Die Analyse ist Voraussetzung, um später ein Design bzw. einen Systementwurf anfertigen zu können. Außerdem können Fehler in den Anforderungen durch ihre genaue Betrachtung innerhalb dieser Projektphase frühzeitig erkannt und behoben werden, bevor sie bei der Erstellung der Systemarchitektur weiterverarbeitet werden.

In der Erhebungsphase werden die Anforderungen in natürlicher Sprache formuliert, um den Bedürfnissen der Kunden gerecht zu werden. In der Analysephase kann auf andere, präzisere Ausdrucksmöglichkeiten – z.B. die UML (siehe Kapitel 4.4) – zurückgegriffen werden (vgl. Booch/Jacobson/Rumbaugh 2001, S. 173 ff). Als Resultat steht ein Analyse-Modell, welches sich durch einen gewissen Abstraktionsgrad auszeichnet und möglichst wenige Details aufweist. Zu diesem Zeitpunkt ist das Modell noch frei von jeglichen Umsetzungsmöglichkeiten. In der Design-Phase dient es als Grundlage für die Anfertigung eines Systementwurfs (vgl. Booch/Jacobson/Rumbaugh 2001, S. 176 ff).

Dem vollständigen Analyse-Modell nähert man sich in mehreren Schritten. Man wandert hierbei vom Groben ins Feine (siehe Abb. 13).

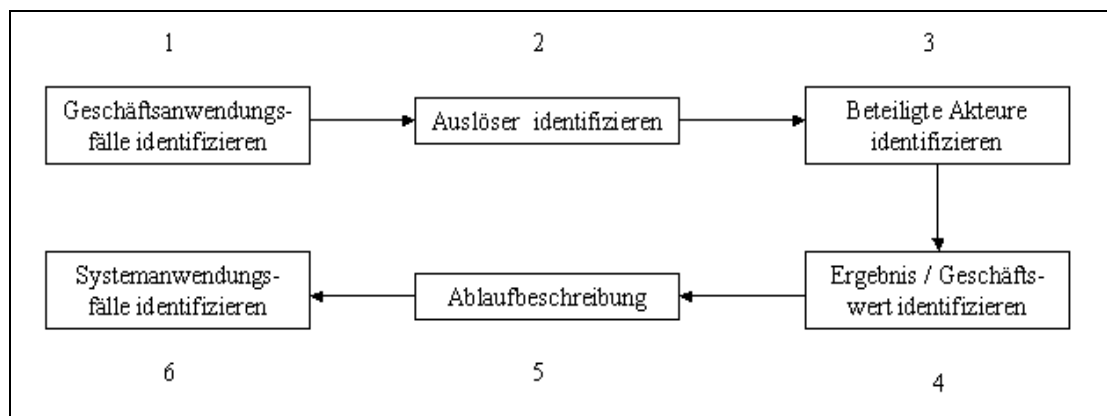


Abb. 13: Schritte der Analysephase (eigene Darstellung)

1) Identifikation der Geschäftsanwendungsfälle

Im ersten Schritt werden die Geschäftsanwendungsfälle des zu betrachtenden Problembereiches identifiziert. Ein Geschäftsanwendungsfall ist ein Prozess innerhalb einer betrieblichen Organisationsstruktur, an dessen Ende das Erreichen eines betrieblichen Ziels steht (vgl. Grässle/Baumann/Baumann 2007, S. 46). Mit diesem Schritt wird ein Überblick zur Thematik geschaffen, während die folgenden Schritte auf eine nähere Betrachtung der einzelnen Geschäftsanwendungsfälle abzielen.

2) **Auslösenden Akteur identifizieren**

Im zweiten Schritt folgt die Bestimmung des auslösenden Akteurs des betrachteten Geschäftsanwendungsfalls. Der auslösende Akteur repräsentiert eine Rolle für einen agierenden Teilnehmer innerhalb eines Prozesses, der diesen aktiv einleitet.

3) **Beteiligte Akteure identifizieren**

Neben dem auslösenden Akteur wird ein Geschäftsanwendungsfall durch weitere Prozessteilnehmer durchgeführt. Diese Akteure werden innerhalb des dritten Schrittes bestimmt.

4) **Ergebnis / Geschäftswert identifizieren**

Jeder Geschäftsanwendungsfall hat mit der Erbringung eines Geschäftswertes zu enden. Dieses Prozessergebnis ist in Schritt vier zu ermitteln.

Die Durchführung dieser vier Schritte gewährleistet, dass die Zusammenhänge des Problembereiches erkannt wurden (vgl. Oestereich 2001, S. 96 ff; S. 129 ff). Eine Ausdifferenzierung der einzelnen Geschäftsanwendungsfälle erfolgt mit Hilfe der folgenden beiden Schritte.

5) **Ablaufbeschreibung**

In diesem Schritt wird der Ablauf eines Geschäftsanwendungsfalls dargestellt. Hierbei ist darauf zu achten, dass der Prozess durch eine Aktion des auslösenden Akteurs begonnen wird. Des Weiteren muss berücksichtigt werden, dass Prozessabläufe nicht immer linear sind, sondern durch Ausnahmen und Verzweigungen einen anderen Weg als den Standardablauf verfolgen können. Um sich nicht in unnötige und sich möglicherweise verändernde Details zu verlieren, werden die Aktionen eines Geschäftsanwendungsfalls möglichst allgemein und abstrakt formuliert (vgl. Oestereich 2001, S. 101 ff). In dieser Phase der Analyse wird zunächst vollständig lösungsfrei modelliert.

6) **Identifikation der Systemanwendungsfälle**

Welche Aktionen eines Geschäftsanwendungsfalls letztendlich durch ein Informationssystem unterstützt werden sollen, wird innerhalb dieses Schrittes durch die Identifikation der Systemanwendungsfälle bestimmt (vgl. Oestereich 2001, S. 109 ff).

Ebenfalls Teil der Analysephase ist die strukturelle Darstellung des Problembereiches. Hierbei werden zunächst die Geschäftsklassen identifiziert und schließlich die Beziehungen zueinander bestimmt (vgl. Oestereich 2001, S. 120 ff). Ein Beispiel für eine derartige Darstellung findet sich in Abb. 9 in Kapitel 3.6.

4.4 Geschäftsprozessanalyse mit der Unified Modelling Language

Zur Analyse der Geschäftsprozesse der Schlüsselverwaltung wurde innerhalb dieses Projektes die Unified Modelling Language verwendet. Der folgende Kapitelabschnitt klärt zunächst, was sich hinter der UML verbirgt. Anschließend folgt eine Vorstellung ausgewählter Diagrammtypen sowie ihrer Elemente, die innerhalb dieses Projektes Anwendung fanden. Im Einzelnen werden Anwendungsfall-, Aktivitäts- und Klassendiagramm vorgestellt.

4.4.1 Was ist UML?

Die Unified Modelling Language ist eine von der Object Management Group (OMG) standardisierte grafische Sprache. Ursprünglich wurde sie von Grady Booch und James Rumbaugh, später auch von Ivar Jacobsen, als Notation zur objektorientierten Analyse entwickelt (vgl. Oose 2007). Die erste standardisierte Version der UML wurde 1996 von der OMG veröffentlicht (vgl. Grässle/Baumann/Baumann 2007, S. 19). Die UML ermöglicht es, die Artefakte eines Systems zu visualisieren, zu spezifizieren, zu konstruieren und zu dokumentieren. Weiterhin kann sie zur Modellierung von Geschäftsprozessen eingesetzt werden (vgl. ISO/IEC 2005, S. 2).

Ein Problembereich, seien es Geschäftsprozesse oder Informationssysteme, kann aus unterschiedlichen Sichten dargestellt werden. Für jede Sicht gibt es einen Diagrammtypen, um die hervorzuhebenden Aspekte darzustellen (vgl. Booch/Jacobson/Rumbaugh 2001, S. 23 ff). Dabei müssen alle Diagramme der unterschiedlichen Sichten in Beziehung zueinander stehen, da sie den selben Problembereich, jedoch aus unterschiedlichen Perspektiven, darstellen (vgl. Hitz/Kappel 1999, S. 6).

Seit Anfang des Jahres 2007 steht UML in der Version 2.1.1 zur Verfügung (vgl. OMG¹ 2007).

4.4.2 Diagrammtypen der UML

Wie in dem vorangegangenen Kapitelabschnitt erwähnt, kennt die UML verschiedene Diagrammtypen – insgesamt dreizehn an der Zahl – ,die jeweils einen anderen Fokus auf den dargestellten Sachverhalt legen (vgl. OMG² 2007). Im Folgenden werden diejenigen Diagramme und ihre Elemente, die zur Analyse des Schlüsselverwaltungs-konzeptes innerhalb dieses Projektes verwendet wurden, vorgestellt.

Anwendungsfalldiagramm

Anwendungsfalldiagramme zeigen Anwendungsfälle, Akteure und deren Beziehungen zueinander. Sie geben einen Überblick über die Aktionen, die von einem Anwendungsfall umfasst werden, geben jedoch keine Information darüber, in welcher zeitli-

chen Reihenfolge diese Aktionen ablaufen oder welcher Akteur diese ausführt (vgl. Grässle/Baumann/Baumann 2007, S. 64; Oestereich 2001, S. 196).

Akteure werden in Anwendungsfalldiagrammen durch Strichmännchen dargestellt. Jeder Akteur steht für eine Rolle, die von einem Beteiligten eines Anwendungsfalls eingenommen werden kann. Akteure lösen Anwendungsfälle, die durch Ellipsen dargestellt werden, aus (vgl. Grässle/Baumann/Baumann 2007, S. 65). Der Anwendungsfall *Schlüssel ausgeben* wird beispielsweise durch den Akteur *Gebäudenutzer* ausgelöst (siehe Abb. 14).

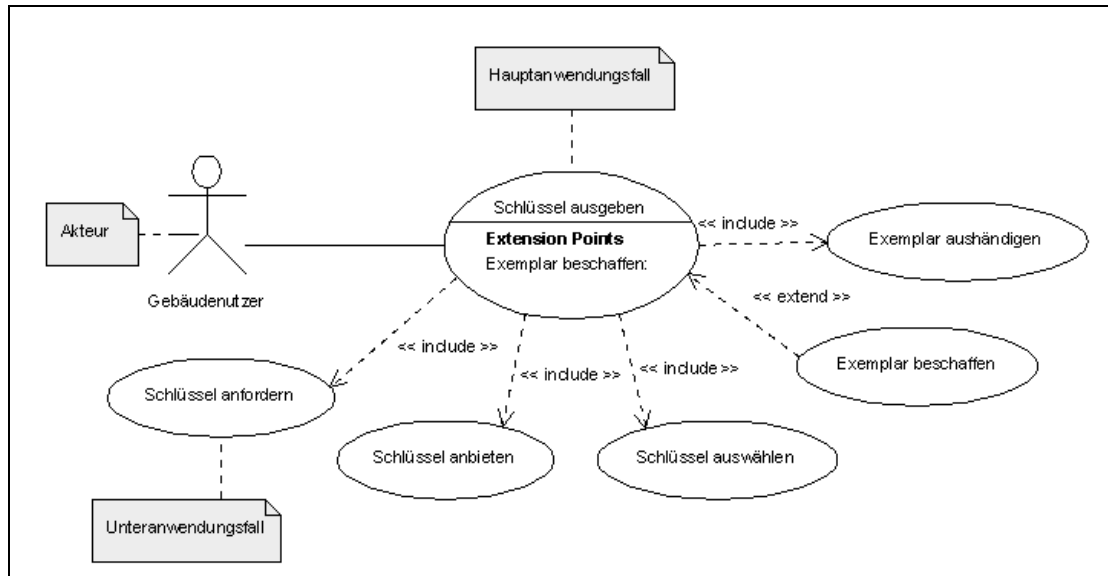


Abb. 14: Anwendungsfalldiagramm (eigene Darstellung)

Ein Anwendungsfall kann in einzelne Unteranwendungsfälle aufgegliedert werden. (vgl. Grässle/Baumann/Baumann 2007, S. 62). Diese werden dem Hauptanwendungsfall durch include- oder extend-Beziehungen zugeordnet (vgl. Oestereich 2001, S. 199 ff). Die include-Beziehung sagt aus, dass einem Hauptanwendungsfall ein Unteranwendungsfall angehört (vgl. Oestereich 2001, S. 207). Hiermit kann demnach beschrieben werden, aus welchen Teilschritten sich ein Anwendungsfall zusammensetzt, was ein detaillierteres Bild über den betrachteten Bereich liefert. Der Anwendungsfall *Schlüssel ausgeben* umfasst beispielsweise die Anwendungsfälle *Schlüssel anfordern*, *Schlüssel anbieten* etc. (siehe Abb. 14). Die extend-Beziehung hingegen drückt aus, dass der Hauptanwendungsfall in einem speziellen Szenario um einen Anwendungsfall erweitert wird (vgl. Oestereich 2001, S. 207). In dem Beispieldiagramm in Abb. 14 wird der Anwendungsfall *Schlüssel ausgeben* durch den Unteranwendungsfall *Exemplar beschaffen* erweitert. Bei der Durchführung der Schlüsselabgabe muss demnach in einigen Fällen ein Schlüsselexemplar beschafft werden, um den Prozess erfolgreich beenden zu können.

Aktivitätsdiagramm

Ein Aktivitätsdiagramm stellt die zeitliche Abfolge der Aktionen eines Prozesses dar. Als Aktion ist hierbei ein einzelner Schritt innerhalb eines Prozesses zu verstehen (vgl. Grässle/Baumann/Baumann 2007, S. 79 ff). Neben Arbeitsabläufen stellen Aktivitätsdiagramme Zuständigkeiten dar. Diese werden durch Partitionen oder auch sogenannte *Schwimmbahnen*, denen Aktionen zugeteilt werden, verdeutlicht (vgl. Grässle/Baumann/Baumann 2007, S. 84). Im Folgenden werden die einzelnen Elemente eines Aktivitätsdiagramms vorgestellt. Eine entsprechende grafische Darstellung findet sich in Abb. 15.

Start- und Endpunkt eines Prozesses werden in einem Aktivitätsdiagramm durch entsprechende Kreissymbole markiert (vgl. Hitz/Kappel 1999, S. 153). Aktionen werden durch Rechtecke mit abgerundeten Ecken dargestellt. Ist eine Aktion durchgeführt, wird mit einem von ihr ausgehenden Pfeil eine weitere Aktion ausgelöst. Aktivitätsdiagramme sind weiterhin in der Lage, Alternativabläufe darzustellen. Dies ist möglich, indem der ausgehende Pfeil einer ausgeführten Aktion zu einem Entscheidungspunkt, dargestellt durch eine Raute, führt. Von ihm aus kann ein Ablauf in unterschiedliche Richtungen fortgesetzt werden. Treffen zwei Abläufe an einem Punkt wieder zusammen, wird dies durch eine Raute mit mehreren eingehenden Pfeilen dargestellt (vgl. Oestereich 2001, S. 290).

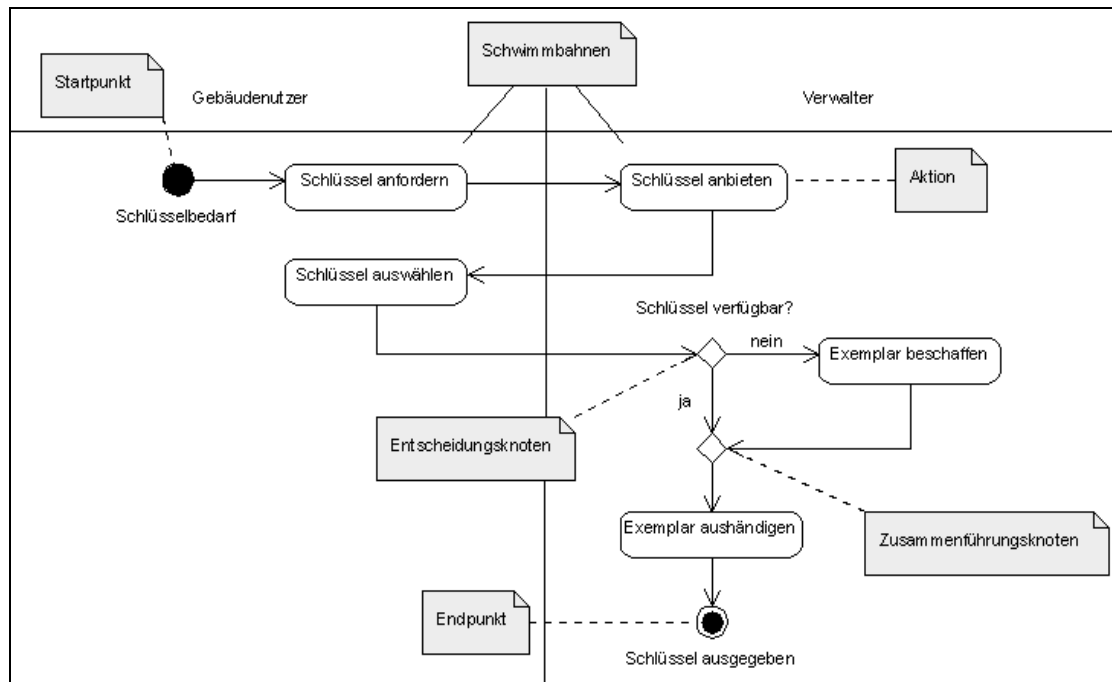


Abb. 15: Aktivitätsdiagramm (eigene Darstellung)

Das Beispielaktivitätsdiagramm in Abb. 15 lässt sich wie folgt lesen: Besteht ein Bedarf, fordert ein Gebäudenutzer einen Schlüssel an. Ein Verwalter bietet Schlüssel an und der Gebäudenutzer wählt einen davon aus. Ist der ausgewählte Schlüssel nicht

verfügbar, wird ein Exemplar durch den Verwalter beschafft. Wurde ein Exemplar beschafft oder ist eines im Lager befindlich, händigt der Verwalter dieses aus.

Prozesse können bei Bedarf mit einem Objektfluss dargestellt werden (siehe Abb. 16). Hierunter sind ein- und ausgehende Informationen bei der Durchführung von Aktionen zu verstehen. Diese werden durch rechteckige Objekte dargestellt. Ein Objekt kann in einem Aktivitätsdiagramm mehrfach verarbeitet werden. In diesem Fall wird es durch seinen jeweiligen Zustand – in eckigen Klammern dargestellt – ergänzt. (vgl. Grässle/Baumann/Baumann 2007, S. 81 ff; Hitz/Kappel 1999, S. 158 ff).

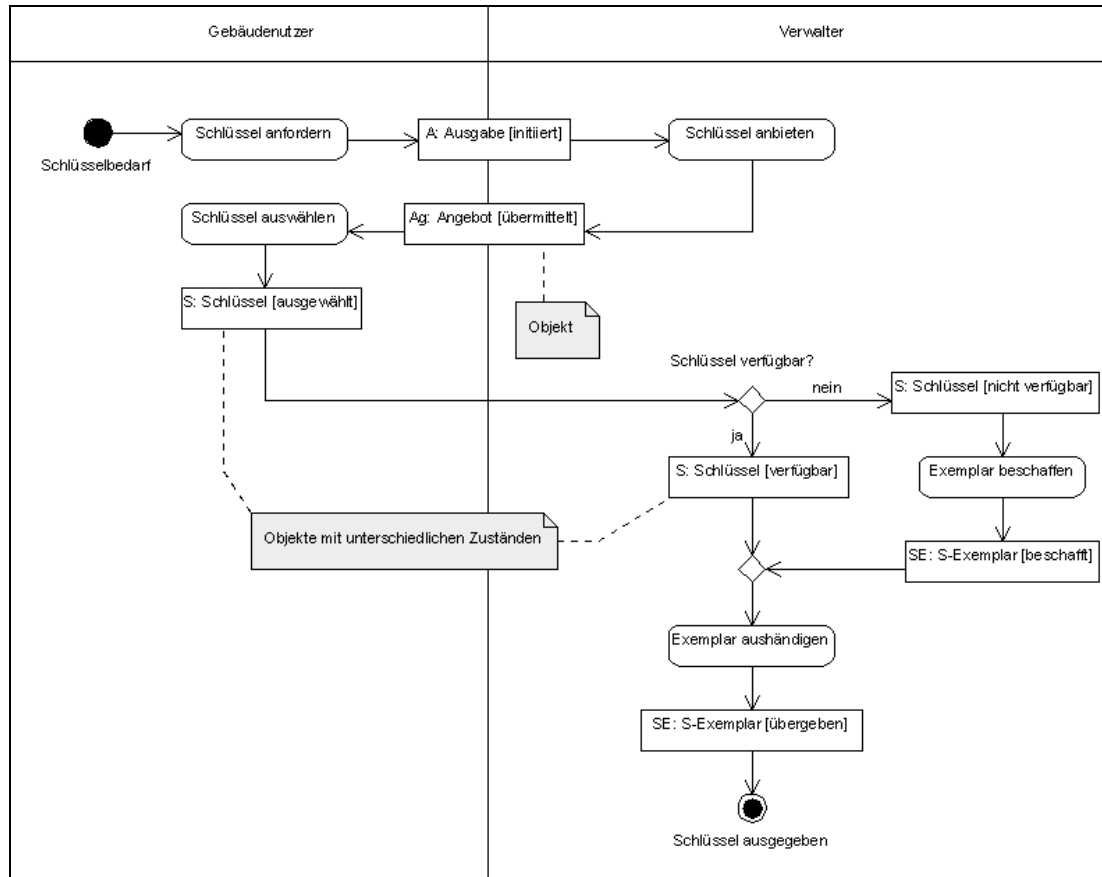


Abb. 16: Aktivitätsdiagramm mit Objektfluss (eigene Darstellung)

Die Aussage des Aktivitätsdiagramms in Abb. 16 ist folgende: Besteht ein Bedarf, fordert ein Gebäudenutzer einen Schlüssel an und initiiert so eine Ausgabe. Ein Verwalter bietet Schlüssel an und übermittelt dem Gebäudenutzer ein Angebot. Dieser wählt einen Schlüssel aus. Ist der ausgewählte Schlüssel nicht verfügbar, beschafft der Verwalter ein Exemplar. Ist das Schlüsselexemplar beschafft, händigt der Verwalter dieses aus. Dasselbe geschieht bei Verfügbarkeit des Schlüssels. Wurde das Exemplar übergeben, gilt die Schlüsselausgabe als beendet.

Klassendiagramm

Klassendiagramme stellen Klassen und ihre Beziehungen zueinander dar (vgl. Oestereich 2001, S. 326). Mit ihnen können sowohl die Strukturen eines Geschäftssystems als auch die eines Informationssystems – also ein Systementwurf – verdeutlicht werden (vgl. Grässle/Baumann/Baumann 2007, S. 107; S. 160). Beide Diagramme stehen in Beziehung zueinander und zeigen die Informationen des zu betrachtenden Bereiches aus unterschiedlichen Perspektiven (vgl. Grässle/Baumann/Baumann 2007, S. 255). Klassendiagramme werden weiterhin zur Begriffsklärung eingesetzt.

Ein Klassendiagramm besteht – wie der Name schon sagt – zum einen aus Klassen. Diese stehen für ein Konzept, eine Menge von Personen oder Dingen mit denselben Eigenschaften (vgl. Grässle/Baumann/Baumann 2007, S. 161). Klassen werden durch Rechtecke dargestellt. Jede Klasse besitzt einen Namen und wird zusätzlich durch Attribute und Operationen ergänzt (vgl. Oestereich 2001, S. 209). Attribute beschreiben die Eigenschaften einer Klasse, während die Operationen ihr Verhalten kennzeichnen (vgl. Oestereich 2001, S. 40).

Klassen sind über Beziehungen verschiedenster Art miteinander verbunden. Die einfachste Beziehung zwischen zwei Klassen wird Assoziation genannt. Sie kann mit einem Zusatz versehen werden, der die Art der Beziehung näher beschreibt. Ohne eine Assoziation stehen zwei Klassen nicht in Verbindung zueinander und können somit auch nicht miteinander kommunizieren (vgl. Oestereich 2001, S. 262).

Die Generalisierung ist eine weitere Art der Beziehung. Sie drückt aus, dass Klassen mit den gleichen Merkmalen zu einer gemeinsamen Klasse zusammengefasst werden. Generalisierungen werden als Pfeile dargestellt und als ist-Beziehung gelesen (vgl. Grässle/Baumann/Baumann 2007, S. 157). Z.B.: Eine Ausgabe ist ein Vorgang (siehe Abb. 17).

Sind zwei Klassen durch eine Aggregation miteinander verbunden, wird dadurch ausgedrückt, dass eine der beiden Klassen sich zum Teil aus der anderen zusammensetzt. Eine Aggregation wird wie eine Assoziation dargestellt, jedoch unter dem Zusatz einer Raute auf Seiten der übergeordneten Klasse. Diese Art der Beziehung wird als „besteht aus“ gelesen (vgl. Oestereich 2001, S. 280). Z.B.: Eine Schließgruppe besteht aus mehreren Gruppen (siehe Abb. 17).

Assoziationen und Aggregationen können mit einem Zusatz versehen werden: der Multiplizitätsangabe. Sie sagt aus, wie viele Objekte miteinander in Beziehung stehen (vgl. Oestereich 2001, S. 262). Multiplizitätsangaben können mit unterschiedlichen Werten belegt werden:

0...1 Ein Objekt steht zu keinem oder einem Objekt in Beziehung

1 Es besteht eine Verbindung zu genau einem Objekt einer Klasse

- * Es besteht eine Beziehung zwischen einem Objekt und vielen Objekten einer Klasse
- 1...* Ein Objekt kann mit einem Objekt oder mehreren Objekten in Beziehung stehen

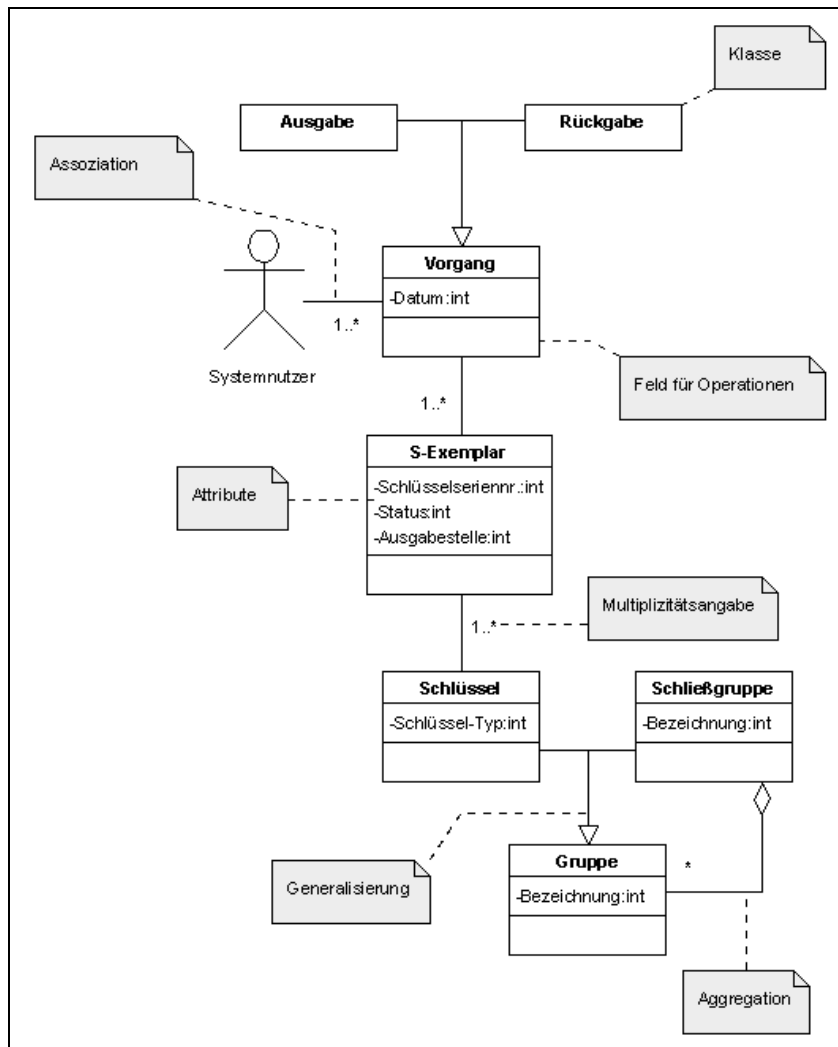


Abb. 17: Klassendiagramm (eigene Darstellung)

Abb. 17 zeigt ein Beispiel eines Klassendiagramms mit den oben angeführten Elementen und lässt sich folgendermaßen lesen: Ausgaben und Rückgaben sind Vorgänge. Ein Systemnutzer führt einen Vorgang oder mehrere durch. Jeder Vorgang besitzt ein Datum und bezieht sich auf ein Schlüsselexemplar oder mehr. Ein Schlüsselexemplar besitzt die Attribute *Schlüsselseriennummer*, *Status* und *Ausgabestelle*. Ein Schlüsselexemplar oder mehr können zu einem Schlüssel gruppiert werden. Dieser besitzt das Attribut *Typ*. Schlüssel und Schließgruppen sind Gruppen. Eine Schließgruppe setzt sich aus mehreren Gruppen zusammen. Beide besitzen eine Bezeichnung.

4.5 Systementwurf (Design)

Unter einem Design ist die Entwicklung eines Systemkonzeptes zu verstehen, welches dazu dient, ein definiertes Problem zu lösen. Die Entwicklung hat unter Einbeziehung erhobener nicht-funktionaler Anforderungen und Rahmenbedingungen zu erfolgen (vgl. Oestereich 2001, S. 145; Booch/Jacobson/Rumbaugh 2001, S. 215). In der Design-Phase wird unter anderem entschieden, auf welcher Entwicklungsumgebung gearbeitet oder welche Programmiersprache zur Umsetzung verwendet werden soll (vgl. Hitz/Kappel 1999, S. 180; S. 257).

Das Design wird auf Basis des Analyse-Modells entwickelt. Die Design-Phase stellt einen Übergang zwischen der Ausarbeitung und der Konstruktionsphase dar (vgl. Booch/Jacobson/Rumbaugh 2001, S. 215 ff). Als Ergebnis entsteht ein Design-Modell, welches die Struktur bzw. Architektur des zu entwickelnden Systems darstellt und einen Entwurf für die letztendliche Implementierung liefert (vgl. Booch/Jacobson/Rumbaugh 2001, S. 265). Es gibt verschiedene Arten von Architekturen, die aus unterschiedlichen Elementen bestehen können (vgl. Oestereich 2001, S. 145). Im Folgenden werden einige Beispielbestandteile genannt, die auch in dem System, welches innerhalb dieses Projektes entwickelt wurde, wiederzufinden sind. Eine grafische Darstellung dieser beispielhaften Architektur findet sich in Abb. 18.

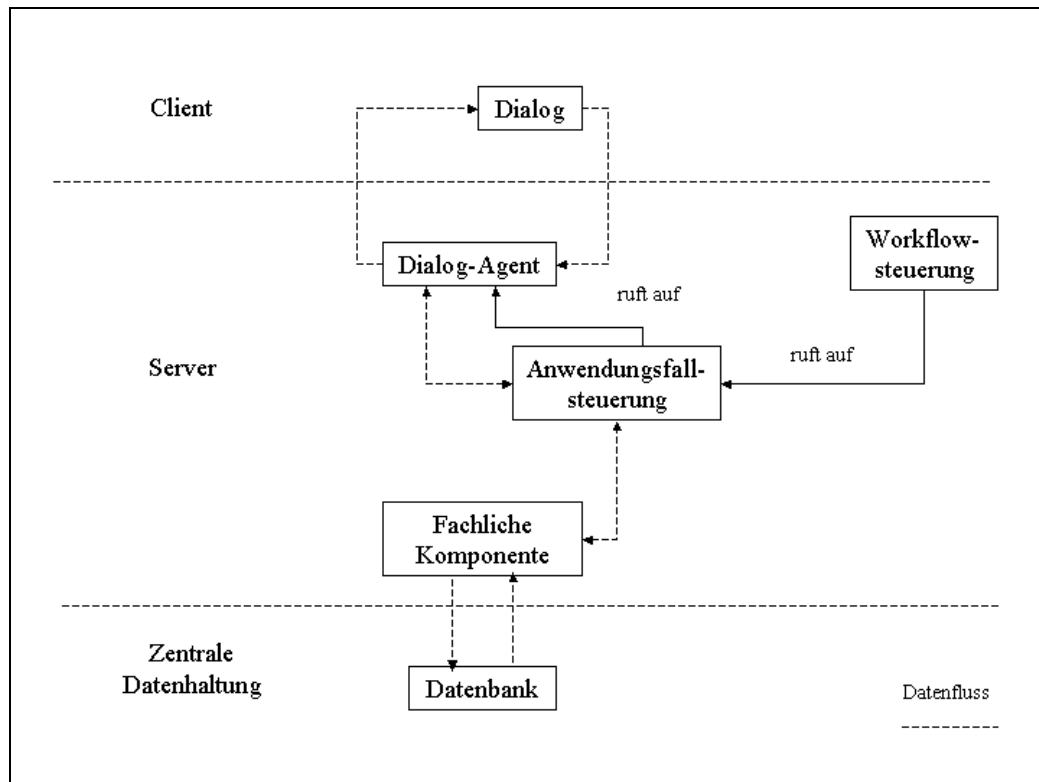


Abb. 18: Beispiel einer Systemarchitektur (nach Oestereich 2001, S. 146)

Dialog

Der Dialog ist eine Interface-Klasse und stellt eine Schnittstelle zum Systemnutzer (Client) dar. Über dieses Element erfolgen Eingaben durch den Client sowie die Darstellung von Informationen am Bildschirm. Der Dialog setzt sich jedoch nicht nur aus Elementen zusammen, die für den Anwender am Monitor sichtbar sind. Weiterhin kann eine Steuerungsklasse hinter dem Dialog stehen, die ihn koordiniert (vgl. Oestereich 2001, S. 146). Dialoge können unterschiedliche Erscheinungsformen haben. Einige ermöglichen dem Nutzer eine freie Kommunikation. In anderen Fällen werden benutzerführende Dialoge – sogenannte *Wizards* – eingesetzt (vgl. Oestereich 2001, S. 175).

Dialog-Agent

Der Dialog-Agent ist eine Steuerungsklasse und dient als Vermittler zwischen der Dialogsteuerung und den Steuerungselementen auf Seiten des Servers. Der Dialog-Agent ist auf einen bestimmten Dialog angepasst, nimmt Daten von ihm entgegen und leitet diese weiter. Umgekehrt leitet der Dialog-Agent Daten an die Dialogsteuerung weiter (vgl. Oestereich 2001, S. 147).

Anwendungsfallsteuerung

Eine Anwendungsfallsteuerung ist ebenfalls eine Steuerungsklasse und für die Koordinierung von reihenfolglich festgelegten Abläufen eines Anwendungsfalls oder mehrerer Anwendungsfälle verantwortlich. Sie aktiviert bzw. deaktiviert Dialoge und dient als weiterleitende Instanz zwischen Dialogsteuerung und Workflowsteuerung bzw. den fachlichen Komponenten (vgl. Oestereich 2001, S. 147). Als Anwendungsfallsteuerung im Bereich der Schlüsselverwaltung kann z.B. die IS-unterstützte Dokumentation einer Schlüsselausgabe verstanden werden.

Workflowsteuerung

Eine weitere Steuerungsklasse innerhalb der beispielhaften Systemarchitektur aus Abb. 18 ist die Workflowsteuerung. Sie „initiiert, überwacht und steuert die Abarbeitung eines Geschäftsvorfalles“ (zit. Oestereich 2001, S. 146). Falls zwischen den Anwendungsfällen eines Geschäftsvorfalles zeitliche Spannen liegen, sorgt die Workflowsteuerung dafür, dass „der aktuelle Zustand der Bearbeitung für die Zeit der Unterbrechung“ gespeichert wird (zit. Oestereich 2001, S. 148). Im Fall der Schlüsselverwaltung kann als Workflowkomponente ein Ausgabe- oder Rückgabe-Wizard verstanden werden.

Fachklassen

Die fachlichen Komponenten stehen für die Fachklassen – inklusive der Attribute und Operationen – eines Geschäftsbereiches. Fachlich zusammengehörende Klassen werden in der Design-Phase zu fachlichen Komponenten zusammengefasst (vgl. Oestereich 2001, S. 148-150). Im Beispiel der Schlüsselverwaltung würde aus den Fach-

klassen *Schlüssel* und *Schlüsselexemplar* die fachliche Komponente *Schlüsselbestand* werden. Diese Komponente kann der Anwendungsfallsteuerung Daten wie Schlüsselseriennummer, Schlüsseltyp, Status oder Ausgabestelle liefern. Die fachlichen Komponenten werden auch als Informationsobjekte bezeichnet.

4.6 Benchmarking

Benchmarking ist ein Instrument der Wettbewerbsanalyse, welches dazu dient, Produkte, Prozesse, Organisationsstrukturen und Strategien des eigenen Unternehmens mit denen der marktführenden Konkurrenz zu vergleichen. Es ermöglicht, Stärken und Schwächen gegenüber Spitzenunternehmen zu identifizieren, um Leistungslücken systematisch zu schließen (vgl. 4managers; Sabisch/Tintelnot 1997, S. V).

Benchmarking erfolgt üblicherweise in einem fünfstufigen Verfahren (siehe auch Abb. 19):

- 1) Auswahl des zu untersuchenden Objektes (Produkt, Prozess...)
- 2) Auswahl des Vergleichsunternehmens bzw. Vergleichsobjektes
- 3) Datengewinnung
- 4) Informationsanalyse: Feststellung von Leistungslücken und ihrer Ursachen
- 5) Umsetzung der Erkenntnisse

(nach Karlöf/Östblom 1994, S. 86; 4managers)

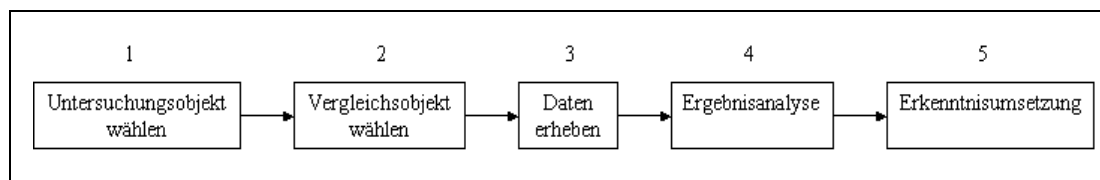


Abb. 19: Vorgehensweise beim Benchmarking (eigene Darstellung)

1) Untersuchungsobjekt wählen

Zunächst muss entschieden werden, was benchmarked werden soll. Hierbei kann es sich um ein spezielles Produkt, einen Prozess, eine Strategie, eine Projekt- oder Organisationsstruktur handeln.

2) Vergleichsobjekt wählen

Im zweiten Schritt ist ein Vergleichsobjekt zu wählen. Dies kann ein Objekt innerhalb des Unternehmens sein (internes Benchmarking), ein Objekt eines branchenähnlichen Unternehmens (externes Benchmarking) oder ein Objekt eines Unternehmens jeder Branche (funktionales Benchmarking) (vgl. Karlöf/Östblom 1994, S. 62-67).

3) Daten erheben

Es folgt die Phase der Informationsbeschaffung. Dies kann auf unterschiedlichen Wegen erfolgen: über Fragebögen, Interviews, Recherche nach allgemeinen Wirtschaftsinformationen etc. Wichtig ist, dass hier sowohl zum Konkurrenzunternehmen als auch zur eigenen Organisation Informationen gesammelt werden (vgl. Karlöf/Östblom 1994, S. 139; S. 145-155).

4) Ergebnisanalyse

In diesem Schritt werden die Informationen analysiert, Leistungslücken identifiziert und interpretiert. Oft kann es sein, dass die Gesamtergebnisse der beiden verglichenen Objekte wegen unterschiedlicher betrieblicher Voraussetzungen stark voneinander abweichen. Diese unterschiedlichen Bedingungen müssen identifiziert und in die Bewertung des Ergebnisses einbezogen werden (vgl. Karlöf/Östblom 1994, S. 166 ff).

5) Erkenntnisumsetzung

Im letzten Schritt des Benchmarking sollen die identifizierten Lücken mit dem Ziel einer Leistungssteigerung geschlossen werden (vgl. Karlöf/Östblom 1994, S. 185 ff).

4.7 Funktionspunktanalyse

Die Funktionspunktanalyse wurde Ende der 70er Jahre von dem IBM-Mitarbeiter Alan Albrecht entwickelt (vgl. Hürten 1999, S. 12). Sie ist eine Methode, um den Aufwand für ein zu realisierendes Informationssystem vor dessen Entwicklung abzuschätzen (vgl. Bundschuh/Peetz/Siska 1991, S. 28). Bei dieser Aufwandsschätzung werden zwei Faktoren betrachtet: Leistungsumfang und Komplexität. Durchgeführt wird die Funktionspunktanalyse anhand von sechs Einzelschritten (siehe Abb. 20).

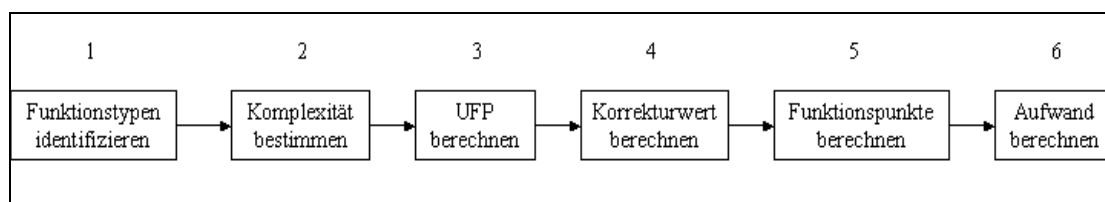


Abb. 20: Ablauf einer Funktionspunktanalyse (eigene Darstellung)

1) Funktionstypen identifizieren

Im ersten Schritt werden die sogenannten Funktionstypen des zu entwickelnden Informationssystems identifiziert. Hiervon gibt es fünf an der Zahl:

- Eingabe (Interface-Klasse)
- Ausgabe (Interface-Klasse)
- Speicher (Informationsobjekt)
- Rechnung (Steuerungsklasse)
- Schnittstelle / Export

(nach Bundschuh/Peetz/Siska 1991, S. 40)

2) Komplexität bestimmen

Im zweiten Schritt werden die identifizierten Funktionstypen hinsichtlich ihrer Komplexität beurteilt. Der Schwierigkeitsgrad ist in drei Stufen aufgebaut:

- einfach
- mittel
- komplex

(vgl. Bundschuh/Peetz/Siska 1991, S.40)

Aus folgender Matrix lassen sich für jeden Funktionstypen eines gewissen Schwierigkeitsgrades die entsprechenden Funktionspunkte ablesen. Eine einfache Eingabe weist beispielsweise 3 Funktionspunkte auf, während eine komplexe Rechnung 15 Funktionspunkte umfasst.

Tabelle 1: Matrix zur Ermittlung der Funktionspunkte

Funktionstyp	Schwierigkeitsgrad		
	einfach (-)	mittel (o)	komplex (+)
Eingabe	3	4	6
Ausgabe	4	5	7
Speicher	3	4	6
Rechnung	7	10	15
Schnittstelle / Export	5	7	10

(nach Bundschuh/Peetz/Siska 1991, S. 40)

3) UFP berechnen

Addiert man die einzelnen Funktionspunkte, erhält man die Summe *Unadjusted Function Points* (UFP) – eine noch nicht an das Entwicklungsumfeld angepasste Gesamtzahl der Funktionspunkte. Eine mittlere Eingabe (4) und ein komplexer Speicher (6) ergeben zum Beispiel einen UFP-Wert von 10.

4) Korrekturwert berechnen

Des Weiteren muss in die Aufwandsabschätzung der Einfluss des Entwicklungsumfeldes einbezogen werden. Hierbei werden 14 Merkmale betrachtet, die mit einem Einflussgrad von 0-5 (0 = kein Einfluss; 5 = starker Einfluss) bewertet werden. Die Summe dieser Bewertungen ergibt den sogenannten *Total Degree of Influence* (TDI) – eine Zahl, die den Gesamteinfluss des Entwicklungsumfeldes repräsentiert. Der TDI kann in folgende Formel eingefügt werden, um einen Korrekturwert (VAF – Value added Factor) für die Summe *UFP* zu bestimmen:

$$\text{VAF} = (\text{TDI} * 0,01) + 0,65$$

(vgl. Hürten 1999, S. 25; Bundschuh/Peetz/Siska 1991, S. 40 ff).

5) Funktionspunkte berechnen

Mit Hilfe des Korrekturwertes können die Funktionspunkte (FP) berechnet werden:

$$\text{FP} = \text{UFP} * \text{VAF}$$

(vgl. Hürten 1999, S. 40)

6) Aufwand berechnen

Um den zeitlichen Aufwand zur Entwicklung eines Informationssystems bestimmen zu können, muss der Realisierungsaufwand für einen Funktionspunkt bekannt sein. Dieser Aufwand kann jedoch nicht pauschal gesetzt werden, sondern hängt von vielen Faktoren, z.B. Qualifikation der Entwickler oder Programmiersprache, ab. Außerdem ist zu definieren, ob der festgelegte Aufwand zur Umsetzung eines Funktionspunktes Anforderungsentwicklung und Systemtest inkludiert (vgl. Glinz 2004, S. 57). Die Anzahl der Funktionspunkte, die ein Entwickler an einem Tag umsetzt, wird im Folgenden als Team Performance (TP = FP pro Tag) bezeichnet. Der zeitliche Aufwand (A) lässt sich durch folgende Formel berechnen:

$$A = \text{FP} / \text{TP}$$

Um den Zwischenschritt der Funktionspunktberechnung [FP = UFP * VAF] zu sparen und gleich die Berechnung des Aufwandes vornehmen zu können, werden diese und die Formel zur Berechnung des Aufwandes kombiniert:

$$A = \text{UFP} * \text{VAF} / \text{TP}$$

Entsprechend dieser Verkürzung der Rechnung werden in Kapitel 8 zur Berechnung des Aufwandes und der Kosten des DESY-SVS die Schritte 5 und 6 zusammengefasst.

5 Analyse

In den folgenden Kapitelabschnitten werden die einzelnen Schritte der Analysephase innerhalb des Projektes vorgestellt. Begonnen wird mit der Schilderung des Ist-Zustandes hinsichtlich der Durchführung der Schlüsselverwaltung am DESY. Zu diesem Zweck fand zu Beginn des Projektes ein Meeting mit den Auftraggebern der Abteilung V1 statt. Im Weiteren werden die Geschäftsprozesse der Schlüsselverwaltung definiert und eine exemplarische Analyse an Hand des Schlüsselausgabeprozesses durchgeführt.

5.1 Ist-Zustand

Zum Einstieg des Projektes fand ein Meeting mit den Auftraggebern der Abteilung V1 statt. Anwesend waren der Abteilungsleiter, ein Facility Manager und der hauptverantwortliche Schlüsselverwalter. Das Zusammentreffen diente vor allem zur Klärung der Projektziele und des -umfangs. Außerdem wurde es genutzt, um den Ist-Zustand der Schlüsselverwaltung am DESY zu umreißen und um einige Anforderungen an den Soll-Zustand zu entwickeln. Als Hilfsmittel fand hierzu die in Kapitel 4.2 vorgestellte Metaplan-Technik Anwendung. Den Projektbeteiligten der Abteilung V1 wurden folgende Leitfragen zur Erhebung des Ist-Zustandes und der Anforderungen an den Soll-Zustand gestellt:

- 1) Welche Aufgaben zum Thema Schlüsselverwaltung fallen in Ihren Tätigkeitsbereich?
- 2) Welche Informationen sind für die Erfüllung dieser Aufgaben erforderlich?
 - Welche Informationen erhalten Sie?
 - Welche Informationen geben Sie weiter?
- 3) Welche positiven Aspekte weist Ihre jetzige Arbeitsweise auf?
- 4) Was würde Ihnen bei der Bearbeitung Ihrer Aufgaben helfen und was ist verbesserungswürdig?

Den projektbeteiligten Mitgliedern von V1 stand eine vordefinierte Bearbeitungszeit für das Notieren ihrer Antworten und Ideen zur Verfügung. Im Anschluss folgte eine thematische Gruppierung der Ergebnisse der Kartenabfrage im Plenum. Hieraus ergab sich folgendes Bild:

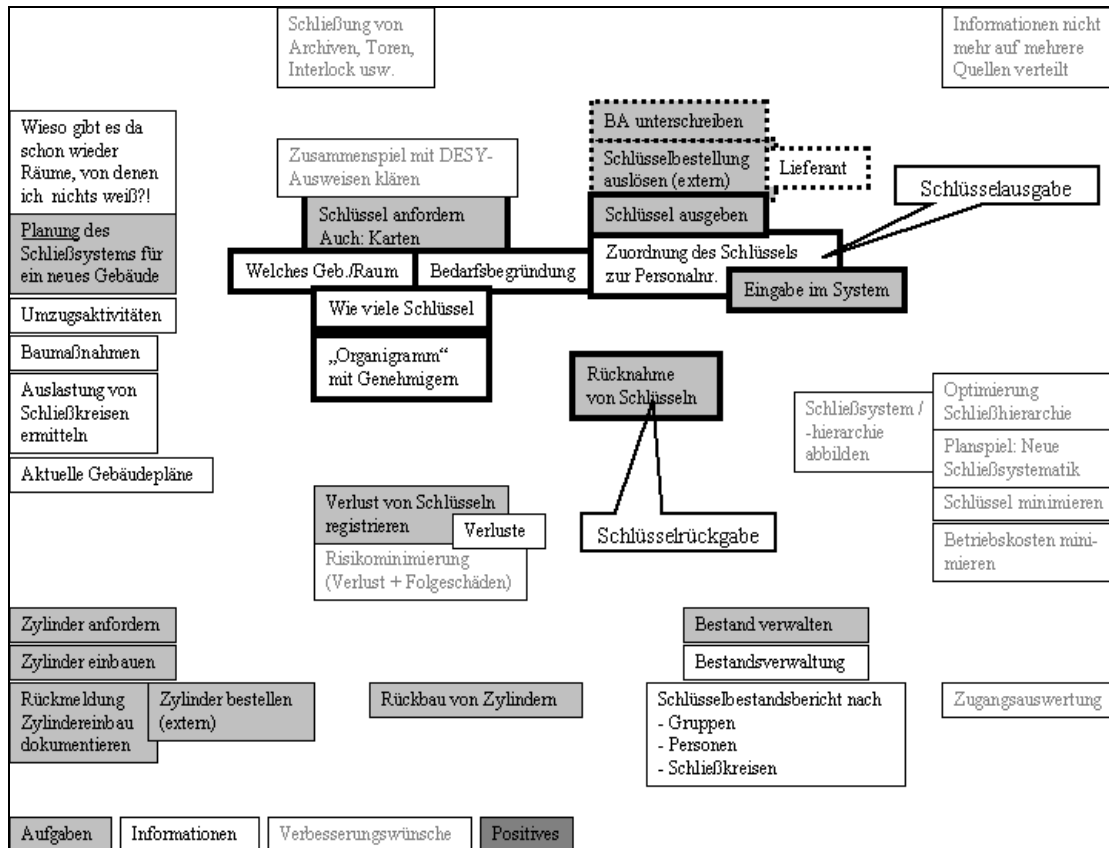


Abb. 21: Ergebnis der Kartenabfrage

1) + 2) Aufgaben der Schlüsselverwaltung und zu verarbeitende Informationen

Mit Hilfe der Metaplan-Technik konnten die Auftraggeber der Abteilung V1 den Projektmitgliedern der Gruppe IPP einen ersten Überblick zu den Aufgaben der Schlüsselverwaltung darbieten. Während der Gruppendiskussion erfolgte eine Erläuterung der Aufgaben und der dazugehörigen Informationen. Hierbei konnten einige der Aufgaben zu einer übergeordneten Tätigkeit zusammengefasst werden. So umfasst die Ausgabe von Schlüsseln beispielsweise auch eine entsprechende Eingabe im System. Die Aufgaben und Informationen der Schlüsselausgabe und -rückgabe beispielsweise, die logisch und thematisch zusammengehören, wurden in Abb. 21 entsprechend angeordnet und mit einem fettgedruckten Rahmen versehen.

Andere Aufgaben sind ebenfalls übergeordneten Tätigkeiten zuzuordnen, werden jedoch nur in Ausnahmefällen ausgeführt. So werden z.B. Schlüsselbestellungen nur dann getätigt, wenn entsprechende Schlüssel nicht mehr im Lager befindlich sind. Entsprechende Punkte innerhalb des Schlüsselausgabeprozesses wurden in Abb. 21 mit einem gestrichelten Rahmen versehen.

Weiterhin ist als Ergebnis der Gruppenarbeit festzustellen, dass einige Aspekte der Schlüsselverwaltung ausreichend umrissen wurden, andere hingegen recht spärlich. Im Falle der Schlüsselausgabe konnte mit Hilfe der Metaplan-Technik ein Bild der

Aufgaben und der zu verarbeitenden Informationen abgedeckt werden. Als Gegenbeispiel konnte das Feld der Schlüsselrückgabe noch nicht ausreichend abgesteckt werden. Um ein vollständiges Bild der Schlüsselverwaltung am DESY zu erhalten, waren im weiteren Projektverlauf Einzelgespräche mit dem hauptverantwortlichen Schlüsselverwalter erforderlich.

Weiterhin ist auffällig, dass einige Aufgaben, wie beispielsweise das Anfordern von Schlüsseln oder der Ein- und Rückbau von Zylindern, nicht in den Bereich des Schlüsselverwalters fallen, sondern innerhalb eines entsprechenden Prozesses von anderen Akteuren durchgeführt werden. Wiederum andere Aufgaben, wie das Verwalten des Bestandes, beziehen sich auf den gesamten Themenbereich und können daher nicht als Kerntätigkeiten bezeichnet werden. Der angeführte Prozess *neues Schließsystem planen* kann wiederum nicht als essentielle Schlüsselverwaltungsaufgabe bezeichnet werden und findet lediglich bei Umzugsaktivitäten oder Baumaßnahmen Anwendung. In der weiteren Phase der SVS-Entwicklung wurde daher auch dieser Prozess nicht weiter berücksichtigt. Als Hauptaufgaben können gemäß des Ergebnisses der Auftraggeberbefragung zunächst folgende festgehalten werden:

- Schlüssel ausgeben
- Schlüssel zurücknehmen
- Verlust von Schlüsseln registrieren
- Zylindereinbau dokumentieren
- Rückbau (bzw. Rücknahme) von Zylindern

3) + 4) Positive Aspekte des Ist-Zustandes und Verbesserungswünsche

Als ein weiteres auffälliges Ergebnis der Kartenabfrage ist festzustellen, dass die Abteilungsmitglieder von V1 keine positiven Aspekte bezüglich ihrer derzeitigen Arbeitsweise zu bemerken hatten. Dies lässt auf einen erheblichen Verbesserungsbedarf schließen. Gleichzeitig zeigten sich die Auftraggeber von V1 bezüglich ihrer Verbesserungswünsche recht verhalten. Es wurden wenig konkrete Ansätze für den Soll-Zustand geschaffen – die meisten Aspekte kamen in der Plenumdiskussion auf, die bei der Ideenfindung eine förderliche Wirkung hatte. Einige dieser Verbesserungswünsche überschritten sich jedoch nicht mit den Zielen des Projektes. So können eine neue Schließsystematik oder eine Optimierung der Schließhierarchie durch die Einführung einer Systemkomponente, welche auf die Unterstützung der Schlüsselverwaltungsprozesse abzielt, nicht verwirklicht werden. Folgende Verbesserungswünsche konnten dagegen innerhalb des Projektes berücksichtigt werden:

- Informationen nicht mehr auf mehreren Quellen verteilt → Bündelung in einem IS
- Schließsystem / -hierarchie abbilden
- Zugangsauswertung
- Zusammenspiel mit DESY-Ausweisen klären

Hinsichtlich des Ist-Zustandes ergab sich aus einer offenen Gesprächsrunde mit den Auftraggebern von V1 folgende Skizze:

- Die Gebäudepläne liegen dem Verwalter in Papierform vor und beinhalten relativ genaue Informationen über die in den Türen eingebauten Zylinder.
- Die Pflege der Pläne erfolgt händisch.
- Die Gebäudepläne werden vom Schlüsselverwalter teils selbst erstellt, teils liegen Kopien der Abteilung Zbau vor.
- Für die Schlüsselausgabe wird das System *Big Key* verwendet.
- Die Zylinderausgabe wird nicht über *Big Key* verwaltet, sondern in den Gebäudeplänen festgehalten.
- Es werden nur Gebäudeschlüssel verwaltet – keine Tor-, Schrankschlüssel usw. Dies wird in Zukunft auch nicht angestrebt.
- Um die verfügbare Schlüsselstückzahl herauszufinden, zählt der Verwalter im Schlüsselschrank nach.
- Der DESY-Bestand weist 15.000 Schlüssel und 3.500 Zylinder auf (bei 2.000 Mitarbeitern).

Zusammenfassung der Ergebnisse

Die Ergebnisse des Brainstormings und der anschließenden Diskussion innerhalb des ersten Meetings mit den Auftraggebern stellten einen ersten Projektansatz dar, der jedoch keinen Anspruch auf Vollständigkeit hat. Er bot allerdings einen Anhaltspunkt, mit dem sich weiter arbeiten und das grobmaschige Bild der Ansprüche an die Schlüsselverwaltung am DESY verdichten ließ. Dies erfolgte, wie bereits erwähnt, in Einzelgesprächen mit dem hauptverantwortlichen Schlüsselverwalter.

Weiterhin war auffällig, dass die Metaplan-Technik von den Auftraggebern wenig begeistert angenommen wurde und das erhoffte Aufkommen an Ideen und Anregungen ausblieb. Dies ist vor allem darauf zurückzuführen, dass die Interessenslage unter den Auftraggebern stark voneinander abwich. Das Projekt wurde durch eine Auftragserteilung Seitens der Abteilungsleitung initiiert, während der hauptverantwortliche Schlüsselverwalter wenig überzeugt von der Notwendigkeit zur Entwicklung einer GIS/FMS-Erweiterung für die Schlüsselverwaltung und der Abschaffung des derzeitigen SVS im selben Zug war. An dieser Stelle musste im weiteren Projektverlauf Überzeugungsarbeit durch die Diplomandin und die Projektmitglieder der Gruppe IPP geleistet werden. Dies wurde ebenfalls durch Einzelgespräche, bei denen die Abteilungsleitung nicht anwesend war, erreicht und indem besonders intensiv auf die Anforderungen des hauptverantwortlichen Schlüsselverwalters eingegangen und mit ihm gemeinsam das DESY-Konzept der Schlüsselverwaltung definiert wurde.

5.2 Geschäftsprozessanalyse

Wie in Kapitel 4.3 dargestellt, beginnt die Analysephase im ersten Schritt mit der Identifikation der Geschäftsprozesse. Als ein Geschäftsprozess wird ein Prozess innerhalb einer betrieblichen Organisationsstruktur definiert, der ein betriebliches Ziel zum Zweck hat (vgl. Grässle/Baumann/Baumann 2007, S. 46). In den folgenden Unterkapiteln wird zunächst ein Überblick der zu analysierenden Geschäftsprozesse der

Schlüsselverwaltung dargeboten und eine exemplarische Analyse an Hand des Schlüsselausgabeprozesses vorgenommen. Dabei wird die in Kapitel 4.3 vorgestellte Vorgehensweise verwendet. Abschließend werden der Umfang des zu entwickelnden SVS und der Soll-Zustand dargestellt.

5.2.1 Die Geschäftsprozesse der Schlüsselverwaltung im Überblick

Ein Teil der Schlüsselverwaltungsprozesse konnte bereits aus den Ergebnissen der Kartenabfrage herausgefiltert werden (siehe Kapitel 5.1). Andere wurden durch Einzelgespräche mit dem hauptverantwortlichen Schlüsselverwalter ergänzt. Insgesamt lassen sich unter dem Begriff der Schlüsselverwaltung aus DESY-Sicht folgende elf Geschäftsprozesse zusammenfassen:

- Schlüssel erfassen
- Schlüssel ausgeben
- Schlüssel zurückgeben
- Schlüsselverlust melden
- Schlüsselfund melden
- Schlüssedefekt melden
- Zylinder erfassen
- Zylinder ausgeben
- Zylinder zurückgeben
- Zylinderdefekt melden
- Bericht erstellen

Da die oben aufgelisteten Geschäftsprozesse in dieser groben Form nicht analysierbar sind, wird im Folgenden der Geschäftsprozess *Schlüssel ausgeben* exemplarisch weiter aufgegliedert. Auf die Analyse der übrigen zehn Prozesse der Schlüsselverwaltung wird in dieser Diplomarbeit nicht näher eingegangen. Sie erfolgte jedoch nach demselben hier an dem Geschäftsprozess *Schlüssel ausgeben* vorgestellten Muster.

5.2.2 Analyse des Schlüsselausgabeprozesses

Im Folgenden wird der Prozess der Schlüsselausgabe gemäß der Vorgehensweise aus Kapitel 4.3 aufgegliedert und analysiert.

1) Identifikation der Geschäftsanwendungsfälle

Zunächst hat eine Identifikation der Geschäftsanwendungsfälle zu erfolgen. Dies geschah bereits in Kapitel 5.2.1. Der im Folgenden betrachtete Geschäftsanwendungsfall ist dementsprechend der Prozess *Schlüssel ausgeben*.

2) Auslösenden Akteur identifizieren

Wurden die Geschäftsprozesse identifiziert, ist der auslösende Akteur zu bestimmen. Dieser repräsentiert eine Rolle für einen agierenden Teilnehmer innerhalb eines Prozesses, der diesen aktiv einleitet. Der Schlüsselausgabeprozess wird von einem Akteur namens *Gebäudenutzer* ausgelöst (siehe Abb. 22).

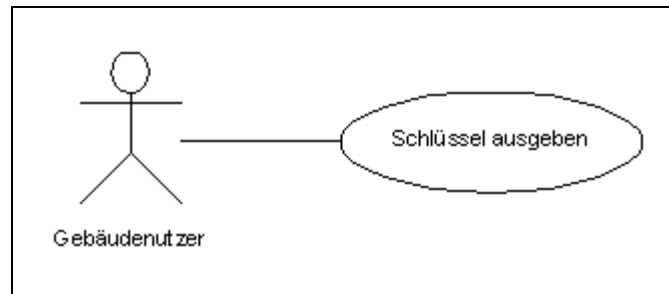


Abb. 22: Auslösender Akteur des Prozesses *Schlüssel ausgeben* (eigene Darstellung)

3) Beteiligte Akteure identifizieren

Ein weiterer beteiligter Akteur bei der Durchführung des zu betrachtenden Geschäftsprozesses ist der *Verwalter*. Er reagiert auf die Einleitung des Prozesses durch den Gebäudenutzer.

4) Ergebnis / Geschäftswert identifizieren

Für den Gebäudenutzer als auslösenden Akteur hat sich bei Abschluss des Prozesses ein Geschäftswert zu ergeben. Im Fall der Schlüsselausgabe wäre dies der Erhalt eines Schlüsselexemplars, das seinen Anforderungen entspricht.

5) Ablaufbeschreibung

In diesem Schritt folgt die Aufgliederung des Geschäftsprozesses *Schlüssel ausgeben*. Hierbei ist auf einen angemessenen Detaillierungsgrad zu achten. Weiterhin sollen die Aktionen noch keine Schlussfolgerung auf die Umsetzung zulassen – sie sollen lösungsfrei sein. Des Weiteren ist es erforderlich, den Prozess durch eine Aktion des auslösenden Akteurs – hier des Gebäudenutzers – beginnen zu lassen. Im Einzelnen umfasst der Prozess folgende Aktionen:

Tabelle 2: Aktionen des Schlüsselausgabeprozesses

Nr.	Akteur	Aktion
1	Gebäudenutzer	Schlüssel anfordern
2	Verwalter	Schlüssel anbieten
3	Gebäudenutzer	Schlüssel auswählen
4	Verwalter	Exemplar aushändigen
5	Gebäudenutzer	Exemplar entgegennehmen
6	Verwalter	Ausgabe dokumentieren
7	Gebäudenutzer	Ausgabe bestätigen
Ausnahmen		
Schlüssel nicht verfügbar / beschaffen		
3a	Verwalter	Exemplar beschaffen (weiter mit 4.)
Schlüssel nicht verfügbar / nicht beschaffen		
3b	Gebäudenutzer	Auswahl ändern (weiter mit 3.)

Grafisch wird der Prozess in einem Aktivitätsdiagramm in Abb. 23 dargestellt. Es macht Zuständigkeiten durch Partitionen deutlich und zeigt ein- und ausgehende Informationen, die bei der Durchführung von Aktionen übertragen werden.

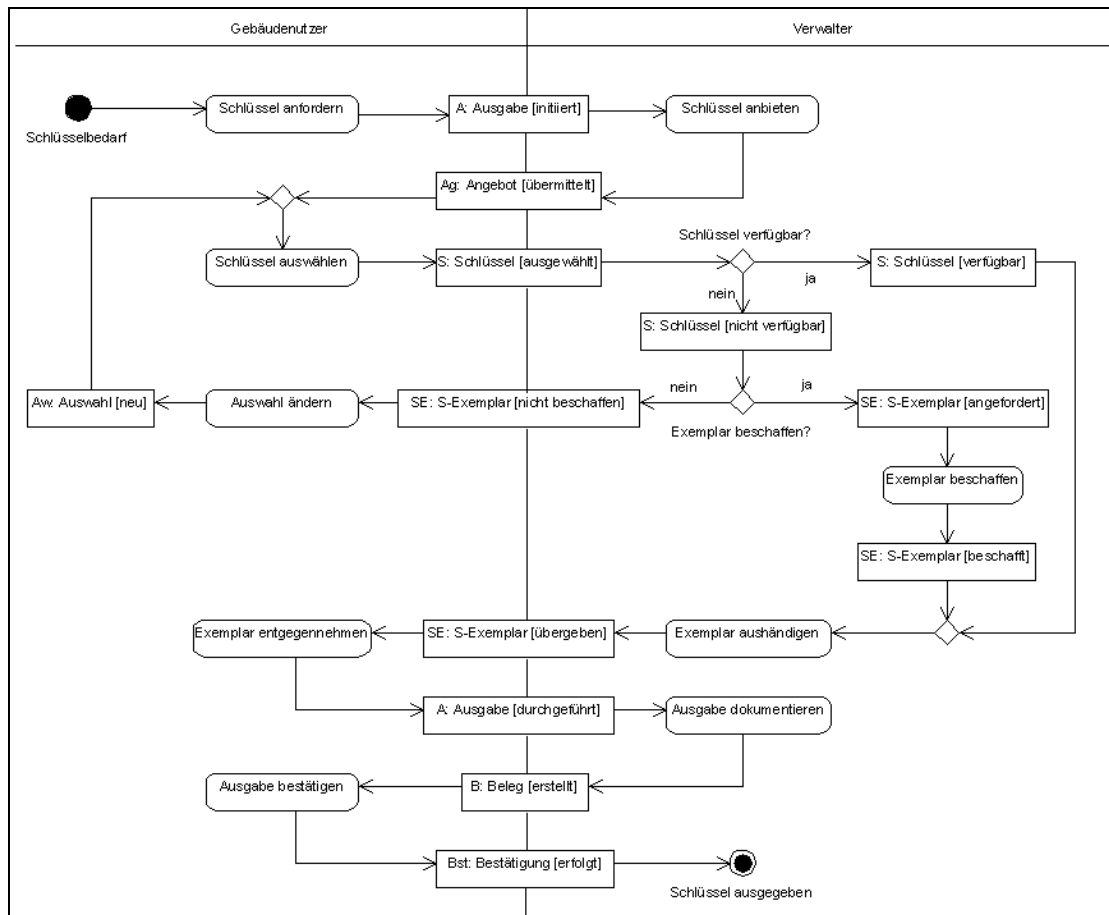


Abb. 23: Ablauf des Schlüsselausgabeprozesses (eigene Darstellung)

6) Systemanwendungsfälle identifizieren

Bei der Bestimmung des Ablaufs des Geschäftsprozesses *Schlüssel ausgeben* wurde zunächst auf die Einbeziehung von Umsetzungsmöglichkeiten verzichtet. Im folgenden Analyseschritt erfolgt nun eine Betrachtung der Aktionen auf Verwalterseite hinsichtlich ihrer Ausführbarkeit durch ein Informationssystem. Hierbei ist festzustellen, dass die Aktionen *Schlüssel anbieten* und *Ausgabe dokumentieren* von einem IS durchgeführt werden können. Abb. 24 fasst diesen Sachverhalt grafisch zusammen.

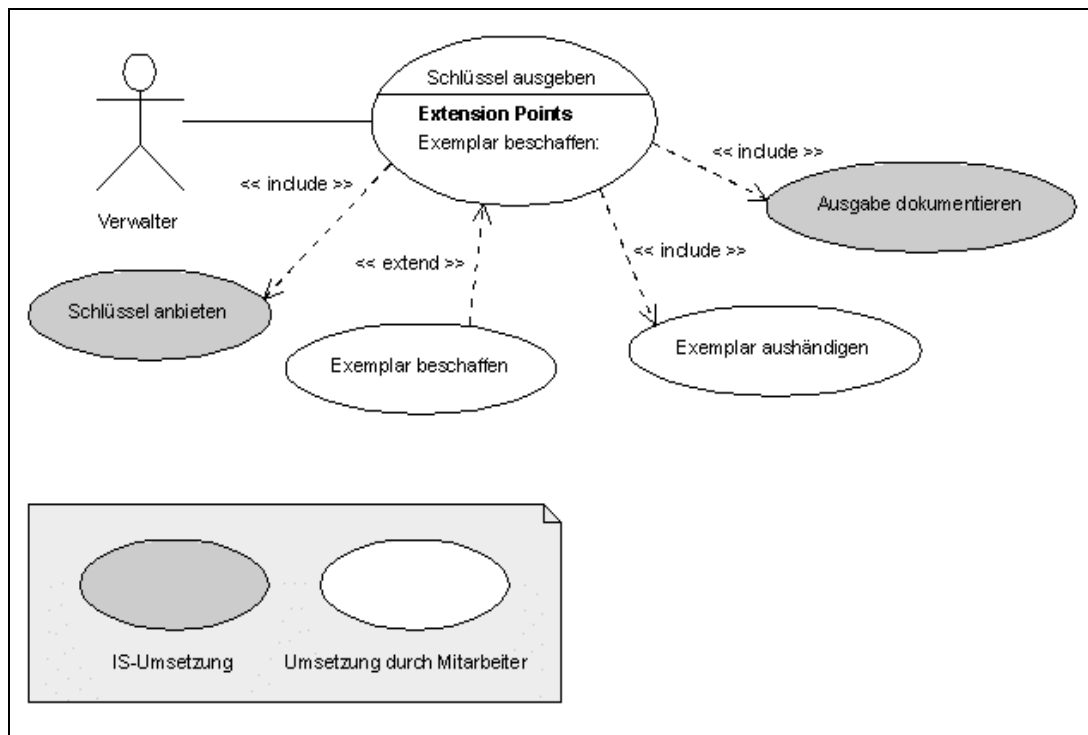


Abb. 24: Umsetzungsmöglichkeiten der Verwaltungsaktionen (eigene Darstellung)

Das in Abb. 23 dargestellte Szenario ist unter Einbeziehung der Umsetzungsmöglichkeit der Prozessaktionen durch ein IS um einen weiteren Akteur – ein Informationssystem – zu erweitern. Aus diesem Schritt ergibt sich folgendes abgeändertes Szenario:

Tabelle 3: Aktionen des Schlüsselausgabeprozesses mit IS

Nr.	Akteur	Aktion
1	Gebäudenutzer	Schlüssel anfordern
2	Verwalter	Anforderung weiterleiten
3	IS	Schlüssel anbieten
4	Verwalter	Angebot weiterleiten
5	Gebäudenutzer	Schlüssel auswählen
6	Verwalter	Exemplar aushändigen
7	Gebäudenutzer	Exemplar entgegennehmen
8	Verwalter	Dokumentation einleiten
9	IS	Ausgabe dokumentieren
10	Verwalter	Beleg weiterleiten
11	Gebäudenutzer	Ausgabe bestätigen
Ausnahmen		
Schlüssel nicht verfügbar / beschaffen		
5a	Verwalter	Exemplar beschaffen (weiter mit 6.)
Schlüssel nicht verfügbar / nicht beschaffen		
5b	Gebäudenutzer	Auswahl ändern (weiter mit 5.)

Die grafische Darstellung des Szenarios unter Einbeziehung des IS erfolgt in Abb. 25:

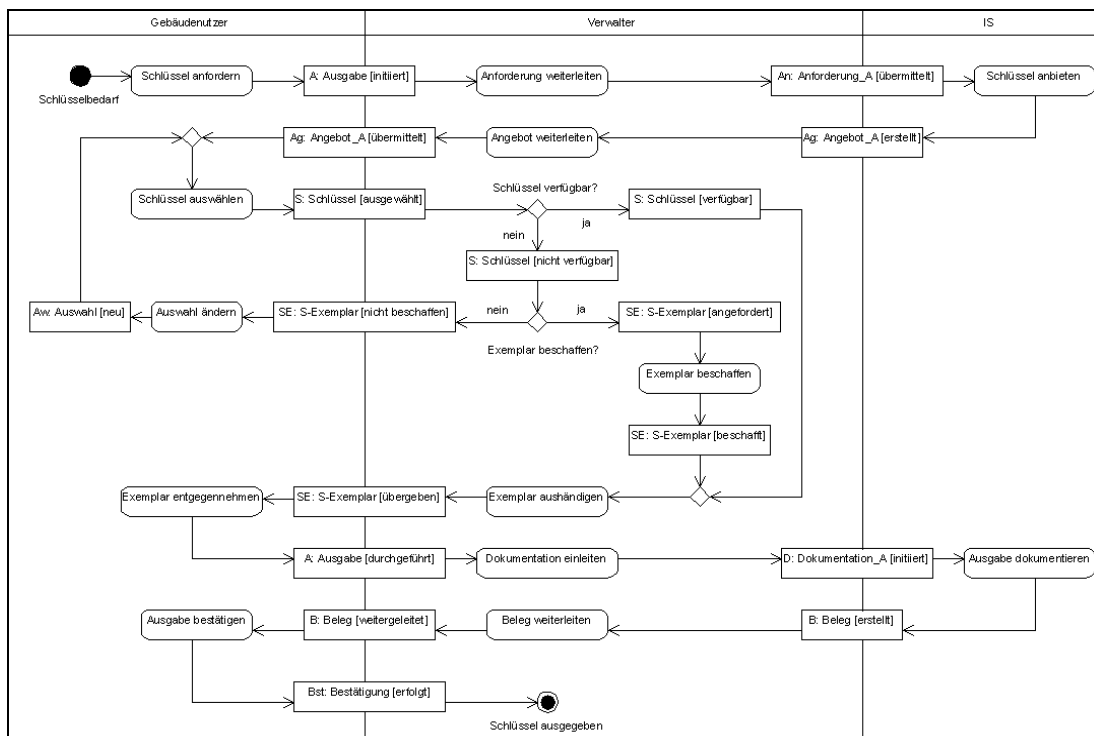


Abb. 25: Ablauf des Schlüsselausgabeprozesses mit IS (eigene Darstellung)

Abbildung 24 zeigt, dass die Aktionen *Schlüssel anbieten* und *Ausgabe dokumentieren* mit Hilfe eines IS umgesetzt werden können, während die Aktionen *Schlüssel beschaffen* und *Exemplar aushändigen* weiterhin von Mitarbeitern durchzuführen sind. Auffällig bei der grafischen Darstellung des abgeänderten Szenarios in Abb. 25 ist, dass es bei der Erweiterung um das IS auch um Aktionen auf Seiten des Verwalters zu erweitern ist. Dies ist notwendig, da der Verwalter als Vermittler zwischen Gebäudenutzer und Informationssystem fungieren soll. Im Folgenden sollen unter dem Hauptanwendungsfall *Schlüssel ausgeben* ausschließlich die IS-umgesetzten Aktionen verstanden werden.

5.2.3 Systemumfang des SVS

Im Eingang dieses Kapitels wurde erwähnt, dass sich der Begriff der Schlüsselverwaltung neben der Schlüsselausgabe aus weiteren Prozessen zusammensetzt. Der vorherige Abschnitt zeigt, dass einige Aufgaben innerhalb des Schlüsselausgabeprozesses durch ein Informationssystem übernommen werden können. Die übrigen Schlüsselverwaltungsprozesse wurden innerhalb des Projektes auf dieselbe Art und Weise analysiert – mit dem Ergebnis, dass ausnahmslos für jeden einzelnen entsprechende Funktionen innerhalb des SVS zu entwickeln sind. Ausgehend davon, dass unter den eingangs definierten Prozessen ausschließlich die Aktionen, welche IS-unterstützt umzusetzen sind, verstanden werden, ergibt sich folgender Systemumfang:

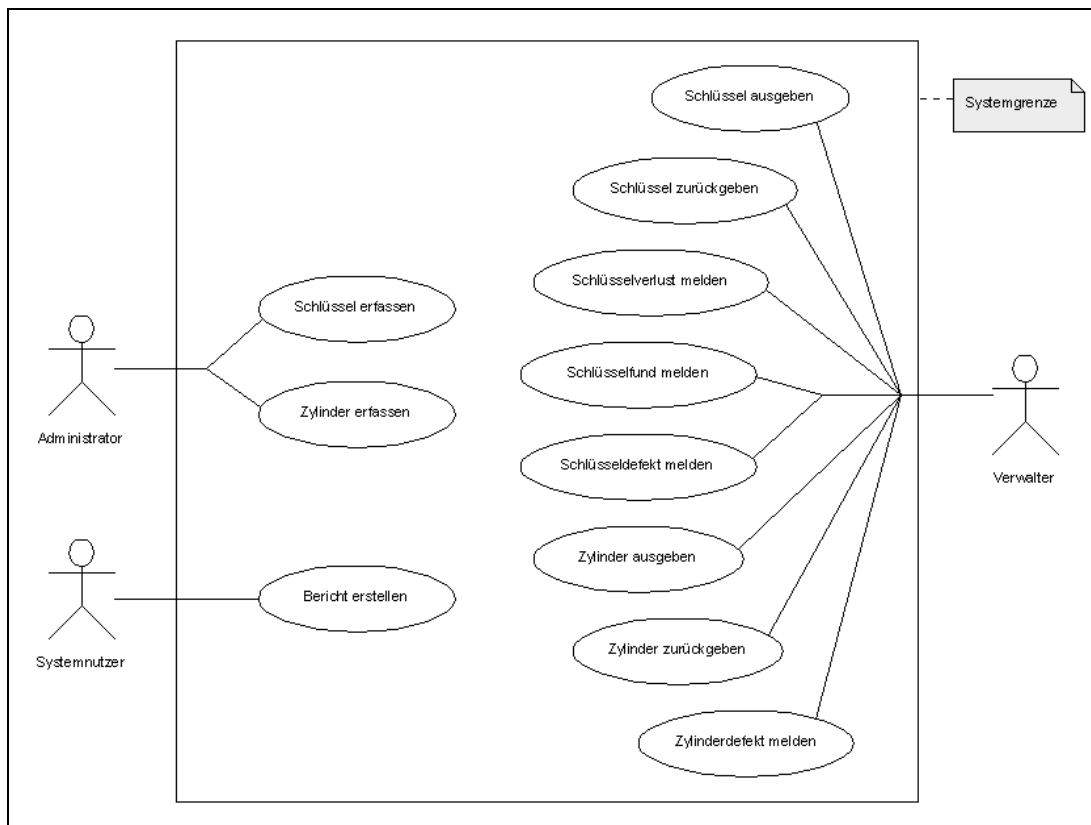


Abb. 26: Systemumfang und externe Akteure (eigene Darstellung)

Abb. 26 stellt den Umfang der zu entwickelnden Systemkomponente dar und verdeutlicht, dass sie bei der Durchführung aller Schlüsselverwaltungsprozesse dienlich sein soll. Entsprechende Funktionen sind durch ein Entwicklerteam zu realisieren. Die Anwendungsfälle der Schlüsselverwaltung werden von einem Administrator, Systemnutzer und Verwalter, die als externe Akteure außerhalb der Systemgrenze liegen, mit Hilfe des SVS durchgeführt. Systemnutzer im Allgemeinen können Berichte erstellen. Abbildung 9 in Kapitel 3.6 zeigt, dass sowohl Administrator als auch Verwalter Systemnutzer sein können. Abb. 16 zeigt, dass beide in der Lage sind, Berichte anzufertigen. Allerdings ist es nur Administratoren erlaubt, Schlüssel und Zylinder zu erfassen, während Verwalter Prozesse wie Ausgaben, Rückgaben und Verlustmeldungen mit Hilfe des Systems verarbeiten dürfen. Das System führt die Aktionen nicht alleine aus, sondern auf Veranlassung und in Kommunikation mit den externen Akteuren.

5.3 Soll-Zustand

Der Soll-Zustand der Schlüsselverwaltung ergab sich innerhalb des Projektes zum einen aus Gesprächen mit den Auftraggebern der Abteilung VI und zum anderen aus der Analyse der Schlüsselverwaltungsprozesse. Das Erreichen des Soll-Zustandes stellt das letztendliche Ziel dieses Projektes dar. Damit dies gelingt, wurden Anforderungen an die Schlüsselverwaltung und die zu entwickelnde Systemkomponente formuliert und in einer Spezifikation gebündelt. Hierfür wurden die in Kapitel 4.1.2 vorgestellten Anforderungsschablonen verwendet. Da die Projektmitglieder mit fortschreitender Projektdauer stets neue Erkenntnisse im Hinblick auf das zu entwickelnde Schlüsselverwaltungssystem gewannen, wurde die Spezifikation über den Projektverlauf hinweg stetig gepflegt und erweitert. Es waren nicht nur Anforderungen hinzuzufügen, sondern auch zu verfeinern, zu ändern oder sogar zu entfernen. Die hierbei entstandene Spezifikation lieferte den externen Entwicklern der Firma Horstick eine detaillierte Anforderungsliste, welche ihnen als Basis für die technische Umsetzung und die Realisierung der FMS-Erweiterung diente. Neben ihrer Funktion als Anforderungsdokument stellte die Spezifikation eine einheitliche Wissensbasis zum Thema Schlüsselverwaltung für alle Projektbeteiligten dar. Dazu wurde ihr eine Definition der FM-Disziplin beigelegt. Die gesamte Spezifikation unterteilt sich in fünf Kapitel:

- 1) Konzept der Schlüsselverwaltung
- 2) Prozessüberblick
- 3) Prozessdefinitionen
- 4) Prozessanforderungen
- 5) Systemanforderungen

Das erste Kapitel definiert ein einheitliches Verständnis zum Thema Schlüsselverwaltung und beinhaltet unter anderem das Klassendiagramm aus Kapitel 3.6. Im zweiten Kapitel wird kurz umrissen, welche Prozesse innerhalb der Schlüsselverwaltung anfallen und von welchen Akteuren diese ausgeführt werden. Zur grafischen Erklärung

ist das Systemkontextdiagramm (Abb. 26) aus Kapitel 5.2 diesem Abschnitt der Spezifikation beigelegt. Kapitel drei geht näher auf die einzelnen Prozesse ein. Die Abläufe werden anhand von Aktivitätsdiagrammen und Szenariotabellen wie in Kapitel 5.2.2 dargestellt. Auch Ausnahmen und Erweiterungen innerhalb der Prozesse werden hier erläutert. Das darauffolgende Kapitel definiert die Anforderungen an die Prozesse. Jeder einzelne wird hier hinsichtlich vierer Punkte aufgeschlüsselt:

- Prozessbeteiligte
- Funktionale Prozessanforderungen
- Randbedingungen
- Anforderungen an die Beteiligten zur Prozessumsetzung

Zunächst werden die beteiligten Akteure und ihre Rollen innerhalb des jeweiligen Prozesses genannt. Es folgt eine Auflistung der funktionalen Anforderungen an die Prozesse. Hierin wird definiert, welche Ergebnisse ein formulierter Prozess zu leisten hat, was seine Ziele und diejenigen einzelner Aktionen innerhalb des Prozesses sind. Die Randbedingungen beschreiben, welche Maßgaben erfüllt sein müssen, damit die Prozesse ausgeführt werden können. Sie gelten innerhalb dieses Projektes für alle Prozesse und werden nicht für jeden einzelnen angepasst. Die Anforderungen an die Beteiligten zur Umsetzung eines Prozesses spezifizieren die zu erbringenden Leistungen der Akteure, um einen Prozess erfolgreich durchführen zu können und somit wiederum den Anforderungen an den Prozess gerecht zu werden.

Das Kapitel der Systemanforderungen unterteilt sich ebenfalls in einzelne Unterabschnitte:

- Anwenderanforderungen
- Funktionale Systemanforderungen
- Nicht-funktionale Systemanforderungen

Die Anwenderanforderungen definieren, welche Leistungen die Systemlösung seinen Nutzern bieten soll, damit diese einen Prozess erfolgreich durchführen können. Hier wird ganz speziell der Dialog zwischen Anwender und System angesprochen, während die funktionalen Anforderungen ausschließlich auf die zu erbringenden Leistungen und Fähigkeiten des Systems anspielen. Unter den nicht-funktionalen Systemanforderungen werden Maßgaben zur inneren Beschaffenheit des Systems zusammengefasst (siehe auch Kapitel 4.1.1).

Die Spezifikation ist auszugsweise dieser Diplomarbeit angehängt. Da die ersten drei Kapitel *Konzept der Schlüsselverwaltung*, *Prozessüberblick* und *Prozessdefinition* bereits in den vorherigen Kapiteln dieser Diplomarbeit ihre Erwähnung fanden, wurde darauf verzichtet, sie ihr ein weiteres Mal im Anhang hinzuzufügen. Die beigelegte Spezifikation beschränkt sich auf die beiden Kapitel *Prozessanforderungen* und *Systemanforderungen*. Allerdings beziehen sich diese nicht auf alle elf Schlüsselverwaltungsprozesse, sondern nur auf denjenigen der Schlüsselausgabe, da ausschließlich dieser als Referenz innerhalb dieser Diplomarbeit dient.

6 Systemlösung

Das folgende Kapitel stellt die Systemlösung zur Unterstützung der Schlüsselverwaltungsprozesse am DESY vor. Zunächst wird der Entwurf einer Systemarchitektur für die Schlüsselausgabe vorgestellt. Weiterhin werden die Zusammenarbeit mit den Entwicklern erläutert und schrittweise die Systemfunktionen zur Unterstützung des Schlüsselausgabeprozesses präsentiert.

6.1 Systementwurf

Bevor mit der technischen Umsetzung der Anforderungen und der Entwicklung begonnen werden konnte, hatte eine Ausarbeitung der Systemarchitektur zu erfolgen. Als Basis wurde hierfür auf die nicht-funktionalen Systemanforderungen und auf das aus der Analysephase resultierende Modell zurückgegriffen. Im Fall des Schlüsselausgabeprozesses erfolgte die Entwicklung der Systemarchitektur ausgehend von dem Aktivitätsdiagramm (Abb. 25) aus Kapitel 5.2.2. Abbildung 27 zeigt schrittweise die Erhebung der Systemelemente, die später in einem Klassendiagramm zusammenzufassen sind.

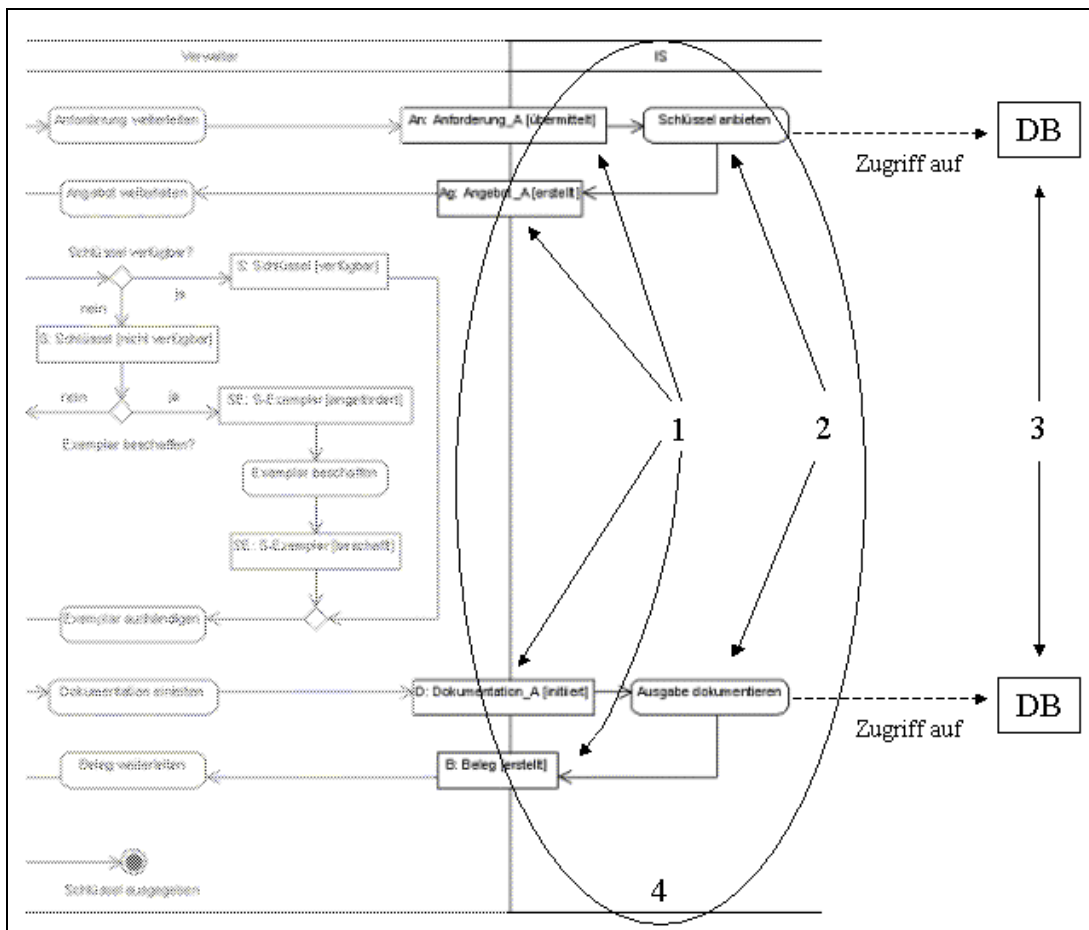


Abb. 27: Identifikation der Funktionsklassen (eigene Darstellung)

1) Ermittlung der Interface-Klassen

Zunächst werden die auf der Partitions Grenze zwischen Verwalter und IS liegenden Klassen betrachtet. Sie stellen eine Schnittstelle zwischen den beiden Akteuren dar und bilden die Interface-Klassen. Klassen dieses Typs können zwischen Eingabe- oder Ausgabeklasse differenziert werden. Die drei folgenden Klassen stellen Eingabemasken dar:

- Anforderung_A
- Angebot_A
- Dokumentation_A

Demgegenüber bilden die drei folgenden Klassen jeweils eine Ausgabemaske:

- Angebot_A
- Dokumentation_A
- Beleg

Es ist auffällig, dass die beiden Klassen *Angebot_A* und *Dokumentation_A* sowohl Eingabe- als auch Ausgabemasken sind. Im Folgenden ist diese Kombination aus Eingabe und Ausgabe getrennt und als zwei unterschiedliche Elemente zu behandeln.

In der Regel lässt sich aus dem Aktivitätsdiagramm an der Richtung des Informationsflusses erkennen, ob es sich bei den Interface-Klassen um Ausgabe- oder Eingabemasken handelt. Geht die Information vom Verwalter zum Informationssystem, muss es sich unweigerlich um eine Eingabeklasse handeln. In diesem Fall ist die Modellierung allerdings nicht detailliert genug, um zu erkennen, dass es sich bei den Klassen *Angebot_A* und *Dokumentation_A* um eine Kombination aus Eingabe und Ausgabe handelt. Andernfalls wäre die Ablaufübersicht auch nicht mehr nachvollziehbar, sondern von zu vielen Details überschwemmt. Bei Bedarf können jedoch einige Ausschnitte mit größerer Feinheit modelliert werden, um eben diese Details zu verdeutlichen.

2) Ermittlung der Steuerungsklassen

In diesem Schritt werden die Klassen identifiziert, welche die Interface-Klassen steuern. Um diese ausfindig zu machen, orientiert man sich an den IS-umgesetzten Prozessen *Schlüssel anbieten* und *Ausgabe dokumentieren*. Ausgehend hiervon können folgende Steuerungsklassen bestimmt werden:

- Schlüsselanbieter_A
- Dokumentar

Die beiden Steuerungsklassen rufen die Eingabe- bzw. Ausgabemasken auf und lassen sie am Bildschirm erscheinen.

Die Klasse *Schlüsselanbieter_A* benötigt Daten, um diese an die Interface-Klassen weiterleiten zu können. Die Klasse *Dokumentar* hingegen soll Informationen zur Speicherung weitergeben. An dieser Stelle sind daher zwei weitere Steuerungsklassen notwendig:

- Schlüsselsuche
- Vorgangsspeicher

Die *Schlüsselsuche* fordert Informationen zu Schlüsselexemplaren von einer Datenbank an und leitet diese an den *Schlüsselanbieter_A* weiter. Der *Vorgangsspeicher* nimmt die Vorgangsdaten vom *Dokumentar* entgegen und sorgt dafür, dass diese in einer entsprechenden Datenbank abgelegt werden.

3) Ermittlung der Speicherklassen

Die Steuerungsklasse *Schlüsselsuche* muss auf eine Datenbank mit Informationen zum Schlüsselbestand zugreifen, um ihre Aufgabe zu erfüllen. Zu diesem Zweck wird eine Speicherklasse eingeführt. Ebenso soll der *Vorgangsspeicher* auf eine Datenbank zugreifen und in ihr entsprechende Daten ablegen. Auch hierfür wird eine Speicherklasse eingeführt. Insgesamt sind folgende Datenbankklassen notwendig:

- Schlüsselbestand
- Vorgangsbestand

Die Speicherklasse *Schlüsselbestand* beherbergt die Stammdaten aller DESY-Schlüssel, während mit Hilfe der Speicherklasse *Vorgangsbestand* Informationen zu durchgeführten Transaktionen dokumentiert werden.

4) Ermittlung der Hauptsteuerungsklasse

Des Weiteren muss es eine Steuerungsklasse geben, welche den kompletten Ablauf der Schlüsselausgabe innerhalb des Informationssystems koordiniert und die Steuerungsklassen *Schlüsselanbieter_A* und *Dokumentar* aufruft:

- Ausgabe-Wizard

Aus dieser Analyse heraus ergibt sich folgende Systemstruktur:

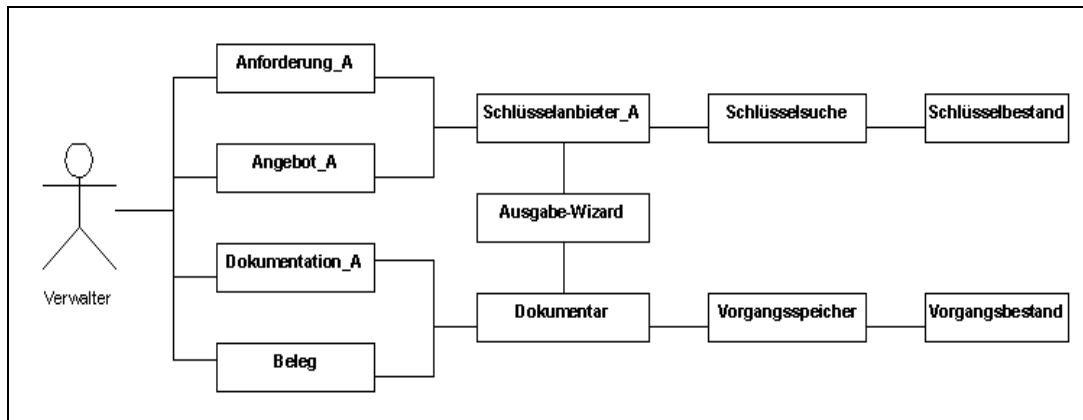


Abb. 28: Systemstruktur der Schlüsselausgabekomponente (eigene Darstellung)

Das Diagramm lässt sich wie folgt interpretieren: Ein Verwalter greift auf die Interface-Klassen *Anforderung_A*, *Angebot_A*, *Dokumentation_A* und *Beleg* zu. Über Eingabeklassen gibt er Daten ein bzw. bekommt Informationen durch Ausgabeklassen angezeigt. Die Interface-Klassen werden von den Steuerungsklassen *Schlüsselanbieter_A* und *Dokumentar* kontrolliert. Diese wiederum koordiniert die Steuerungsklasse *Ausgabe-Wizard*. Der *Schlüsselanbieter_A* aktiviert die Interface-Klasse *Anforderung_A*, welche am Bildschirm erscheint. Um eine Suchanfrage nach Schlüsseln abzugeben, trägt der Verwalter die Daten des Zugangswunsches (Gebäude und Raum) in die Interface-Klasse *Anforderung_A* ein. Sie gibt die Informationen weiter an den *Schlüsselanbieter_A*. Dieser ruft die *Schlüsselsuche* auf und übergibt ihr die entgegengenommenen Daten. Die *Schlüsselsuche* greift auf Grund der Gebäude- und Raumdaten auf entsprechende Informationen aus dem *Schlüsselbestand* zu und leitet passende Schlüsselexemplare weiter an die Steuerungsklasse *Schlüsselanbieter_A*, welche die Interface-Klasse *Angebot_A* aktiviert. Diese zeigt dem Verwalter die Informationen der ermittelten Schlüsselexemplare an. *Angebot_A* stellt gleichzeitig eine Eingabemaske dar und nimmt die Auswahl eines Schlüsselexemplars vom Verwalter entgegen. Diese Information wird an den *Schlüsselanbieter_A* weitergeleitet, welcher dem *Ausgabe-Wizard* die Erfüllung seiner Aufgaben meldet. Der *Ausgabe-Wizard* ruft im nächsten Schritt den *Dokumentar* auf, welcher die Interface-Klasse *Dokumentation_A* aktiviert. Über die Eingabemaske *Dokumentation_A* tätigt der Verwalter eine Dateneingabe, indem er den Empfänger des ausgewählten Schlüsselexemplars angibt. Die Informationen werden an die Steuerungsklasse *Dokumentar* weitergeleitet, welche die Klasse *Vorgangsspeicher* aktiviert und ihr die Daten übergibt. Der *Vorgangsspeicher* legt diese im *Vorgangsbestand* ab. Der *Vorgangsspeicher* bestätigt dem *Dokumentar* die Erfüllung seiner Aufgabe, woraufhin dieser die Interface-Klasse *Dokumentation_A* zur Bestätigung der Durchführung für den Verwalter aufruft. Abschließend aktiviert der *Dokumentar* die Interface-Klasse *Beleg*, die dem Verwalter abschließend die Informationen des durchgeführten Prozesses anzeigt.

Da sich die in Abb. 28 dargestellte Systemarchitektur ausschließlich auf den Schlüsselausgabeprozess bezieht, wurde hier lediglich ein Systemausschnitt modelliert. Um ein umfassendes Design zu erhalten, waren für die übrigen Schlüsselverwaltungsprozesse nach dem gleichen Schema Systemelemente und -struktur zu modellieren. Der

Grundaufbau variiert hierbei nicht, lediglich die Systemklassen waren entsprechend der Prozesse anzupassen, wobei einige durchaus wiederverwendet werden konnten.

6.2 Implementierungsphase

Die Erweiterung des FMS fand nicht innerhalb DESYs statt, sondern wurde an eine externe Firma – Horstick – vergeben, die für den Betrieb und die Anpassungen des GIS/FMS verantwortlich ist. Da es sich bei dem SVS um eine Komponente handelt, die in das GIS/FMS integriert werden und dieses erweitern soll, wurde Horstick folglich mit der technischen Realisierung betraut. Die Implementierung der Komponente zur Verwaltung des Schlüssel- und Zylinderbestandes erfolgte auf Basis einer von DESY erstellten Spezifikation.

Zu Beginn des Projektes fand ein Meeting mit den zuständigen Entwicklern der Firma Horstick statt, um Projektumfang und -zeitplan sowie die technischen Rahmenbedingungen zu klären. Außerdem wurde das Meeting mit dem Entwicklerteam zu Anfang der Projektphase genutzt, um ein einheitliches Verständnis zum Thema Schlüsselverwaltung sowohl auf DESY- als auch auf Entwicklerseite zu schaffen. Weiterhin wurde der Ist-Zustand vorgestellt und im Gegenzug erläutert, wie DESY sich die Zukunft im Hinblick auf die Schlüsselverwaltung vorstellt. Um den Entwicklern der Firma Horstick zusätzlich einen Eindruck von den Nutzerwünschen und der erwarteten Performance im Hinblick auf das SVS zu ermöglichen, wurde ihnen eine Systemsimulation auf Microsoft-Powerpoint-Basis präsentiert. In einer anschließenden Diskussion wurde entschieden, dass es am sinnvollsten ist, die Funktionalitäten jedes Verwaltungsprozesses in einem benutzerführenden Wizard zu kapseln. Dies soll zum einen verhindern, dass im SVS Gebäude- oder Personendaten geändert werden oder dass der Verwalter bei der Durchführung eines Prozesses die Angabe notwendiger Daten vergisst. Insgesamt sollen die einzelnen Wizards zur Fehlervermeidung beitragen, die Datenkonsistenz fördern und den Schlüsselverwalter schrittweise durch den jeweiligen Prozess führen.

Es folgte die Entwicklungsphase auf Basis der von DESY erstellten Spezifikation. Für Testzwecke wurden jedoch nicht alle Funktionalitäten umgesetzt, sondern zunächst nur diejenigen zur Unterstützung der Schlüsselausgabe und -rückgabe. Diese wurden den Projektmitgliedern DESYs von den Entwicklern der Firma Horstick in einem Workshop vorgestellt und diskutiert. Änderungswünsche der DESY-Projektmitglieder konnten aufgenommen und im weiteren Projektverlauf umgesetzt werden. Des Weiteren wurden Unstimmigkeiten und Missverständnisse zwischen den Projektmitgliedern, die sich möglicherweise auf das ganze Projekt erstreckt und ein Risiko für einen erfolgreichen Abschluss dargestellt hätten, durch die anfängliche Entwicklung zweier Prozesse vermieden bzw. konnten rechtzeitig beseitigt werden. Außerdem konnten offene Fragen zur Schlüsselverwaltung während des Workshops unter den Projektmitgliedern oder durch Rückmeldung mit den zuständigen Schlüsselverwaltern der Abteilung VI geklärt werden.

Für die Projektmitglieder auf DESY-Seite stellte der Workshop eine erste technische Einführung im Umgang mit dem SVS dar. Auf dieser Wissensbasis war es ihnen später möglich, das SVS mit Testdaten zu befüllen und den Schlüsselausgabe- und -rückgabeprozess zu erproben.

Schrittweise erhielten die Entwickler der Firma Horstick die Spezifikationen für die weiteren Schlüsselverwaltungsprozesse und setzten diese technisch um. Alle Funktionalitäten wurden von den Projektteilnehmern DESYs hinsichtlich der spezifizierten Anforderungen überprüft und auf etwaige Entwicklungsfehler getestet und von den Entwicklern entsprechend überarbeitet. Die Projektmitglieder näherten sich somit schrittweise dem fertigen Produkt.

6.3 Systemvorstellung: Das DESY-SVS

Aus der Implementierungsphase entstand als Ergebnis eine Systemlösung entsprechend der DESY-Spezifikation. Das SVS wurde als neue Komponente in das bestehende GIS/FMS integriert. Da das System zur Verwaltung des Schlüssel- und Zylinderbestandes sensible Informationen enthält, die nur durch autorisierte Systemnutzer einsehbar und änderbar sein sollen, war die Einrichtung einer Benutzeranmeldung erforderlich. Jeder Schlüsselverwalter des Deutschen Elektronen-Synchrotrons besitzt entsprechend ein Nutzerkonto für das SVS mit einer bestimmten Rechtszuweisung. Hierdurch kann z.B. entschieden werden, dass ein Systemnutzer zwar Schlüsselexemplare zurücknehmen, jedoch keine ausgeben darf. Des Weiteren werden ihm durch die Zuordnung seiner Ausgabestelle Transaktionsrechte für Schlüsselexemplare seines Verantwortungsbereiches eingeräumt. Die Durchführung von Vorgängen mit Exemplaren einer anderen Ausgabestelle wird das System dem Verwalter somit unterbinden. Im Folgenden werden das SVS und seine Funktionalitäten genauer vorgestellt. Da diese Diplomarbeit als Referenz-Prozess stets auf die Schlüsselausgabe zurückgreift, steht folgerichtig der Schlüsselausgabe-Wizard im Fokus der nachstehenden Systemvorstellung.

6.3.1 Der Schlüsselausgabe-Wizard

Bevor einer der Wizards aktiviert und genutzt werden kann, ist zunächst das SVS zu starten. Nach erfolgreicher Benutzeranmeldung hat der Systemnutzer die Möglichkeit, für die Durchführung eines Prozesses aus der linken Navigationsleiste einen entsprechenden Wizard aufzurufen. Ein Klick auf den *Schlüsselausgabe*-Button öffnet den Wizard, welcher den Schlüsselverwalter durch den Schlüsselausgabeprozess leitet.

Jeder Wizard setzt sich aus mehreren Eingabe- und Ausgabemasken zusammen. Im Fall des Schlüsselausgabe-Wizards ist dies an den Reiterkarten *Suchen*, *Auswahl*, *Besitzer* und *Speichern* zu erkennen (siehe Abb. 29). Auf diesem Weg hat der Systemnutzer die Möglichkeit, innerhalb einer Schlüsselausgabe zurückzuspringen, um ge-

gebenenfalls seine Angaben zu ändern. Das Vorspringen auf eine Reiterkarte ist jedoch nicht möglich und wurde von den Entwicklern der Firma Horstick entsprechend umgesetzt. Dies verhindert, dass der Schlüsselverwalter einige Schritte innerhalb eines Prozesses überspringt und so die Angabe notwendiger Daten vergisst.

Bei der Durchführung einer Schlüsselausgabe startet der Wizard auf der Reiterkarte *Suchen*. Der Verwalter hat im ersten Schritt von dem Gebäudenutzer einen Zugangswunsch zu erfragen. Von dieser Information ausgehend, kann der Verwalter mit Hilfe des SVS ermitteln, welcher Schlüssel den Zugangswunsch des Klienten befriedigt. Hierfür sind in der Suchmaske des Schlüsselausgabe-Wizards die Informationen *Gebäudennummer*, *Etage*, *Raum 1*, alternativ *Raum 2* und ebenfalls alternativ *Schlüsseltyp* anzugeben (siehe Abb. 29). Durch die Angabe des Schlüsseltypen kann die Suche gleich zu Beginn der Schlüsselermittlung eingegrenzt werden, denn sollte der Gebäudenutzer keine Berechtigung für den Besitz eines Generalschlüssels haben, ist ein entsprechendes Angebot überflüssig. Als Variante können die Suchfelder auch mittels Markierung eines Punktes in dem Navigationsbaum automatisch ausgefüllt werden. Per Mausklick auf den Button *Suchen* wird die Anfrage an das Informationssystem übergeben, um passende Schlüssel zu ermitteln.

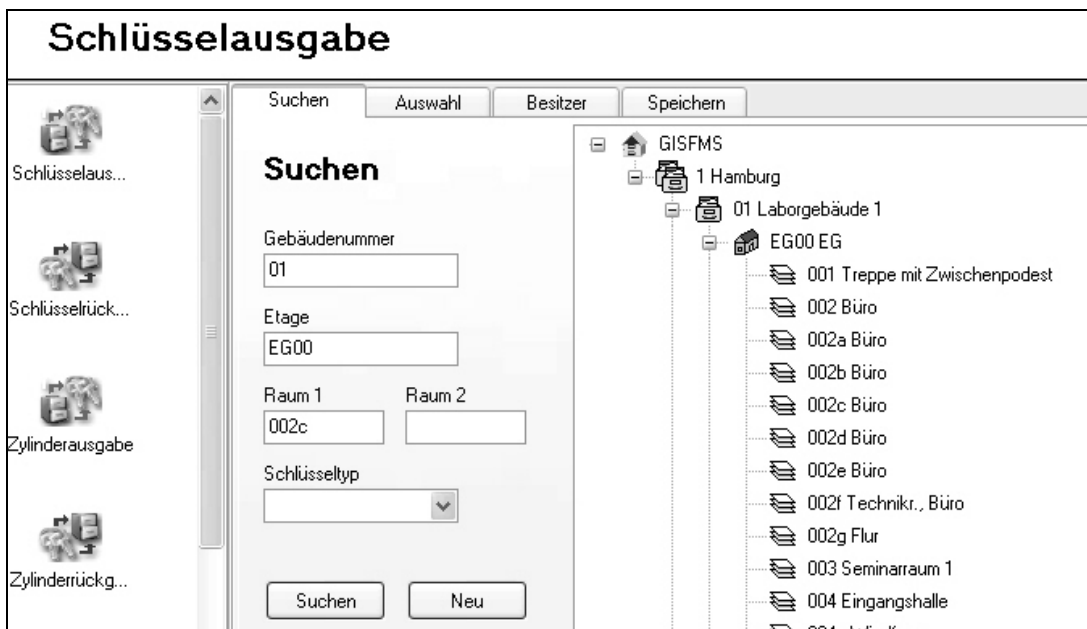


Abb. 29: Gebäude- und Raumauswahl des Schlüsselausgabe-Wizards (Screenshot)

Wurde die Suchanfrage abgeschickt, zeigt der Schlüsselausgabe-Wizard im nächsten Schritt auf der Reiterkarte *Auswahl* ein entsprechendes Suchergebnis an. Dem Verwalter wird eine Liste verfügbarer Schlüssel angeboten. Als zusätzliche Informationen liefert ihm das System die Schlüsselseriennummer, die Typangabe und die Ausgabestelle. Auf der rechten Seite des Fensters erhält der Schlüsselverwalter eine Übersicht mit der Anzahl noch verfügbarer Schlüsseltypen. In dem Beispiel aus Abb. 30 sind zur Suchanfrage vier Einzelschlüssel, jedoch weder Generalhauptschlüssel, Obergruppenschlüssel, Gruppenschlüssel oder Zentralschlüssel vorhanden. Diese An-

zeige ermöglicht dem Verwalter das rechtzeitige Erkennen einer notwendigen Schlüsselnachbestellung.

Hat der Gebäudenutzer einen oder mehrere Schlüssel ausgewählt, kann der Verwalter ein entsprechendes Exemplar oder mehrere markieren und gelangt per Mausklick auf den Button *auswählen* zum nächsten Ausgabe-Schritt. Wählt der Schlüsselverwalter ein Exemplar aus, das nicht zu seiner Ausgabestelle gehört, erscheint eine Meldung, welche ihm mitteilt, dass für diese Aktion keine Benutzerrechte bestehen.

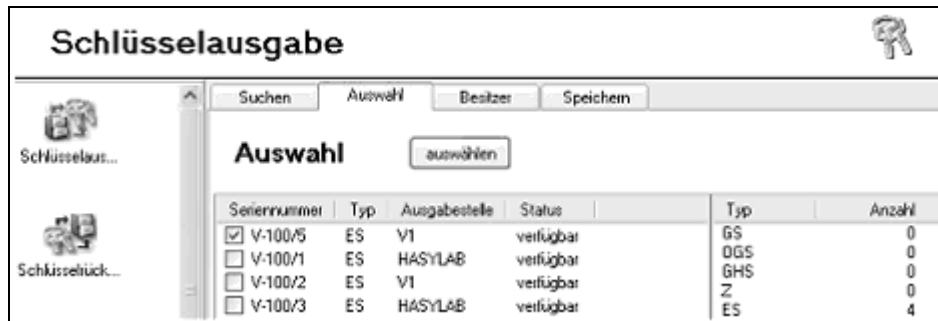


Abb. 30: Schlüsselauswahl-Liste des Ausgabe-Wizards (Screenshot)

Hat der Verwalter ein Schlüsselexemplar oder mehrere zur Ausgabe ausgewählt, kann er diese an den Empfänger übergeben. Zur Dokumentation des Vorgangs ist eben dieser Empfänger im nächsten Schritt auf der Reiterkarte *Besitzer* anzugeben (siehe Abb. 31). Um den zukünftigen Besitzer zu finden, ist die Durchführung einer Suche notwendig. Diese kann in den Kategorien *Name*, *Vorname* und *Gruppe* vorgenommen werden. Gibt der Schlüsselverwalter wie in dem Beispiel in Abb. 31 unter Vorname *Anna* ein und aktiviert die Filterfunktion, wird ihm eine Liste mit gespeicherten Personen, die den Vornamen *Anna* besitzen, angezeigt. Im folgenden Schritt ist der Empfänger aus der Suchergebnisliste zu markieren. Per Mausklick auf den Button *auswählen* gelangt der Schlüsselverwalter zur nächsten Maske des Schlüsselausgabe-Wizards.

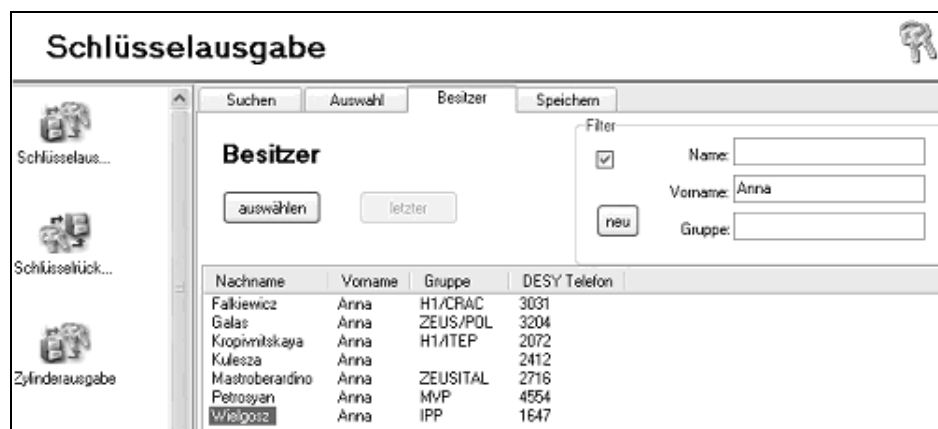


Abb. 31: Ergebnis der Besitzer-Suche (Screenshot)

Auf der Maske in Abb. 32 werden dem Schlüsselverwalter vor der Durchführung der Transaktion abschließend die Daten des Vorgangs angezeigt. Dazu gehören zum einen die Schlüsselinformationen wie *Seriennummer*, *Typ* und *Ausgabestelle*. Zum anderen wird dem Verwalter eine Übersicht der Empfängerdaten geboten. Aktiviert der Verwalter den Button *speichern*, wird die Transaktion abgeschlossen und der Vorgang in der Datenbank gespeichert. Der Status des Schlüsselexemplars wird hierbei in *ausgegeben* geändert, damit es nicht für eine weitere Ausgabe angeboten wird.

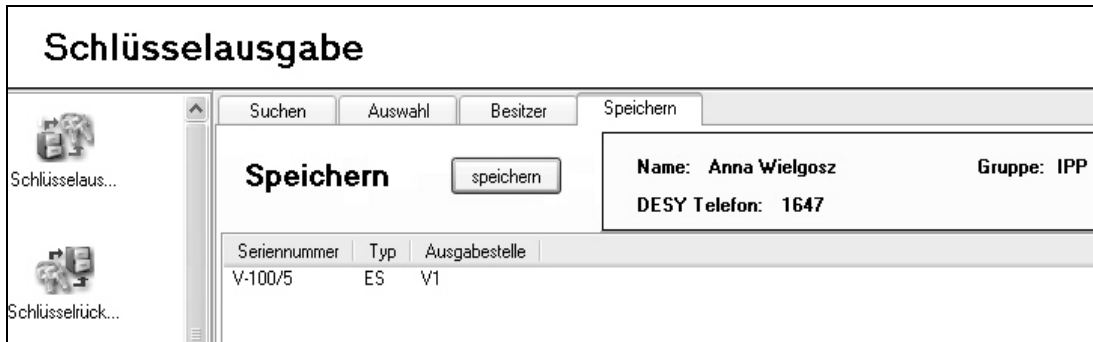


Abb. 32: Übersicht der Vorgangsdaten vor Transaktionsspeicherung (Screenshot)

Eine vollständige Übersicht des Vorgangs lässt sich auf Wunsch als Beleg für den Schlüsselempfänger ausdrucken. Unter Angabe des Datums werden hier Schlüsselseriennummer, Art des Vorgangs, Schlüsseltyp, Gebäude- und Raumdaten sowie Name und Gruppe des Empfängers aufgelistet (siehe Abb. 33).

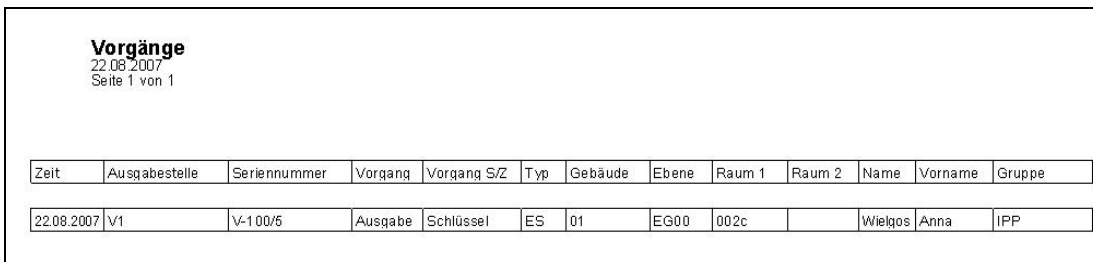


Abb. 33: Druckvorschau des Belegs zur Schlüsselausgabe (Screenshot)

Durch eine Unterschrift kann der Empfänger die Korrektheit der Angaben bestätigen. Diese Anerkennung des durchgeführten Vorgangs ist die Voraussetzung für die erfolgreiche Beendigung des Prozesses. Der Verwalter hat nun die Möglichkeit, einen weiteren Prozess durchzuführen und hierfür einen entsprechenden Wizard zu starten.

7 Benchmark-Test

Inhalt des folgenden Kapitels ist ein Leistungsvergleich zwischen dem DESY-SVS und einer Kaufsoftware. Ziel hierbei ist die Erkenntnis darüber, mit welchem der beiden Systeme die Projektziele erfüllt werden können und welches von ihnen den Ansprüchen DESYs genügen kann. Hierbei wurde gemäß des fünfstufigen Verfahrens

von Karlöf und Östblom (siehe Kapitel 4.6) vorgegangen. Der erste Kapitelabschnitt schildert die Herangehensweise bei der Durchführung des Benchmark-Tests. Anschließend folgen eine Analyse der Ergebnisse des Leistungsvergleichs sowie eine Zusammenfassung derselbigen.

7.1 Durchführung des Benchmark-Tests

Das ausschnittsweise in Kapitel 6.3 vorgestellte SVS wurde auf Basis eines ausgearbeiteten Anforderungskatalogs von der Firma Horstick entwickelt und als Erweiterung in ein bestehendes Facility-Management-System integriert. Im Vorfeld war ein entsprechender Aufwand zur Anforderungserhebung, Analyse der Schlüsselverwaltungsprozesse und zur Entwicklung der Spezifikation zu leisten. Stellt sich die Frage, ob eine kundenspezifische Systementwicklung Aufwand und Kosten rechtfertigt oder ob der Erwerb einer Kaufsoftware ebenso ausreicht, um die Kundenwünsche zu befriedigen. Zur Beantwortung dieser Frage wurde ein Leistungsvergleich mit Hilfe der Benchmarking-Methode durchgeführt.

Hiermit ist der erste Punkt nach Karlöf und Östblom geklärt:

1) Untersuchungsobjekt wählen

Untersucht werden soll das maßgeschneiderte Schlüsselverwaltungssystem des Deutschen Elektronen-Synchrotrons.

2) Vergleichsobjekt wählen

Als Vergleichsobjekt wurde das System *Key Organizer* der Firma AIDeX Software ausgewählt. Der *Key Organizer* steht interessierten Kunden als Demoversion mit vollständiger Funktionalität, jedoch in der Anzahl der Datensätze auf 20 Schlüssel und 20 Personen begrenzt, zur Verfügung (vgl. AIDeX¹). Je nach Bedarf kann der *Key Organizer* mit unterschiedlicher Datensatzkapazität für einen Preis von ca. 80€ bis ca. 550 € erworben werden. Sollten Größe des Unternehmens oder des Schlüsselbestandes zunehmen, ist der Erwerb eines Software-Upgrades möglich (vgl. AIDeX²). AIDeX zählt mehr als 500 Unternehmen, Behörden und Organisationen in Deutschland, Österreich und der Schweiz zu seinen Kunden, darunter den Landesverband der Betriebskrankenkassen Baden-Württemberg BKK, den Arbeiter-Samariter-Bund in Offenbach, das Kaufhaus Saturn und Reader's Digest Deutschland (vgl. AIDeX³).

3) Daten erheben

Die Datengewinnung zu den beiden Produkten erfolgte über einen Fragenkatalog. Dieser deckte folgende vier Aufgabenbereiche und deren Unterpunkte ab:

1. Datenmodell

- 1.1 Stammdaten
- 1.2 Relationen

2. User Interface / Ergonomie

- 2.1 Benutzerhilfe / Verständlichkeit
- 2.2 Ergonomie

3. Prozesse

- 3.1 Schlüssel ausgeben
- 3.2 Schlüssel zurückgeben

4. Systemarchitektur

- 4.1 Rollen und Rechte
- 4.2 Schnittstelle zu anderen Systemen / Import
- 4.3 Berichte

Der Katalog bestand aus 41 Aufgaben, die von den Anforderungen aus der Spezifikation abgeleitet und hinsichtlich derer beide Systeme getestet wurden. Der Aufgabenkatalog diente als Anleitung, um die beiden Systeme *Key Organizer* und *DESY-SVS* einem Testpublikum vorzustellen. Nacheinander wurden alle Aufgaben von einem Vertreter des Entwicklerteams zunächst mit dem einen, dann mit dem anderen System durchgeführt. Das Testpublikum mit einer Stärke von vier Personen vergab für die Erfüllung der Aufgaben Punkte von 1 bis 4. Die Aufschlüsselung der Bewertung lautet wie folgt:

- 1 – nicht erfüllt
- 2 – erfüllt
- 3 – gut erfüllt
- 4 – sehr gut erfüllt

Das Bewertungsspektrum wurde bewusst grobmaschig gehalten, um den Testanwendern bei ihrer Bewertung eine eindeutige Tendenz zu entlocken. In der Auswertung der einzelnen Bewertungen wurden den Punkten prozentuale Werte zugeordnet.

- 1 – 0 %
- 2 – 50 %
- 3 – 85 %
- 4 – 100 %

Dies ermöglichte bei der Zusammenfassung der einzelnen Bewertungen zu einer Gesamtbeurteilung eine bessere Einschätzung der Akzeptanz der beiden Systeme bei den Testanwendern.

Da den Aufgabenblöcken, Unteraufgabenblöcken und Aufgaben nicht die gleiche Wichtigkeit zuzuordnen ist, wurde im Vorfeld der Systempräsentation eine Gewichtung der einzelnen Aufgabenkatalog-Segmente vorgenommen. Ziel dieser Vorge-

hensweise war, dass als weniger wichtig einzuordnende Aufgaben oder Aufgabenblöcke einen geringeren und besonders wichtige einen höheren Anteil zum Gesamtergebnis beitragen.

4) Ergebnisanalyse

Die Analyse der gewonnenen Daten aus Punkt 3 erfolgt in Kapitel 7.2.

7.2 Ergebnisanalyse

Im Zuge der Durchführung eines Leistungsvergleichs ist folglich eine Analyse der Ergebnisse vorzunehmen. Die Resultate dieser Betrachtung werden in den nachstehenden Unterkapiteln vorgestellt. Zunächst wird auf das Gesamtergebnis des Leistungsvergleichs eingegangen. Es folgen differenzierte Betrachtungen hinsichtlich der Bewertungen der einzelnen Testanwender sowie der Ergebnisse in den einzelnen Aufgabenblöcken und Szenarien.

7.2.1 Gesamtergebnis

Als Ergebnis der Datengewinnung lag eine Leistungsbewertung der beiden Vergleichsobjekte *Key Organizer* und *DESY-SVS* durch vier Testanwender vor. Aus den einzelnen Bewertungsergebnissen wurde ein Gesamtergebnis errechnet, das sich in Grafik 34 widerspiegelt.

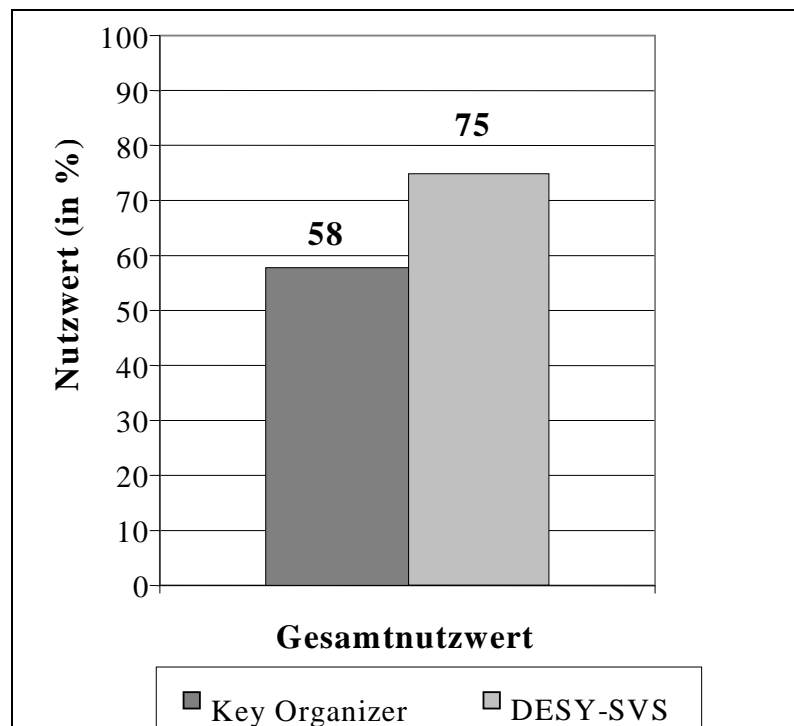


Abb. 34: Gesamtergebnis des Benchmark-Tests (eigene Darstellung)

Es ist zu erkennen, dass die Kaufsoftware *Key Organizer* bei den Testanwendern eine Akzeptanz von 58 % erreicht, während das nach DESY-Wünschen entwickelte SVS der Firma Horstick 75 % Anwenderakzeptanz erzielen kann. Somit liegt die Bewertung beider Systeme über der 50%-Grenze, was ein entscheidender Indikator dafür ist, dass die Projektziele mit beiden Produkten erreicht werden könnten und ein Scheitern des Projektes ausgeschlossen werden kann. Dennoch – es darf nicht übersehen werden, dass das Produkt *Key Organizer* die 50%-Schwelle nur mit knappen 8 % übersteigt und ein Erreichen der Projektziele mit dem SVS der Firma Horstick bei 75 % leistungsmäßiger Bewertung eher gewährleistet werden kann.

7.2.2 Bewertung der einzelnen Testanwender

Eine Gesamtbeurteilung der Systeme fasst die Bewertungen der einzelnen Testanwender zusammen. Eine Betrachtung der Einzelergebnisse ermöglicht, Differenzen in der Bewertung unter den Testanwendern zu erkennen. Die Aufspaltung des Gesamtergebnisses gibt Aufschluss darüber, ob es sich in den einzelnen Bewertungen der Testanwender widerspiegelt oder einen Mittelwert darstellt, der völlig unterschiedliche Anwenderbewertungen zu einer vereinigt. Abbildung 35 stellt das Ergebnis der Einzelbewertungen der vier Testanwender grafisch dar.

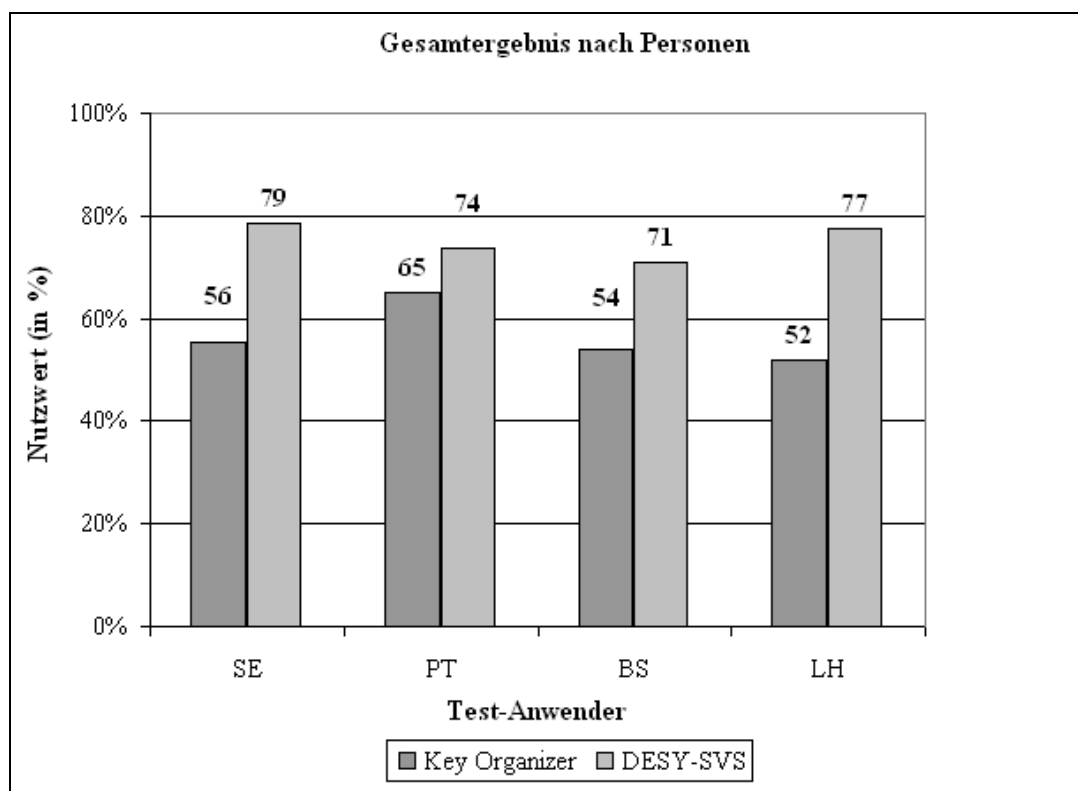


Abb. 35: Gesamtergebnis unterteilt nach Testanwendern (eigene Darstellung)

Ein Vergleich der Werte aus den Abbildungen 34 und 35 ergibt, dass die Bewertungen der Testanwender alle relativ nahe am Gesamtergebnis liegen und die Streuung

der Bewertungen recht gering ist. Die Key-Organizer-Bewertungen der Anwender weichen mit maximal 7 % nach oben und 6 % nach unten ab. Bei der Bewertung des DESY-SVS liegt die maximale Abweichung nach oben und unten bei 4 %. Das Gesamtergebnis kann die Bewertung jedes einzelnen Testanwenders gut repräsentieren.

7.2.3 Bewertung in den Aufgabenblöcken

Zur Feststellung von Defiziten (Punkt 4 nach Karlöf und Östblom, siehe Kapitel 4.6) wird eine Betrachtung der Nutzwerte unterteilt nach Aufgabenblöcken vorgenommen. Ausgehend hiervon können Schwachstellen und Stärken der beiden Systeme identifiziert und genauer untersucht werden. Um die zu beseitigenden Leistungslücken besser eingrenzen zu können, erfolgt im nächsten Schritt eine entsprechende Analyse (siehe Abb. 36). Dies kann bis auf die unterste Ebene – eine Betrachtung der einzelnen Aufgaben – geführt werden. Eine derart feingliedrige Analyse wurde in dieser Arbeit jedoch nicht vorgenommen.

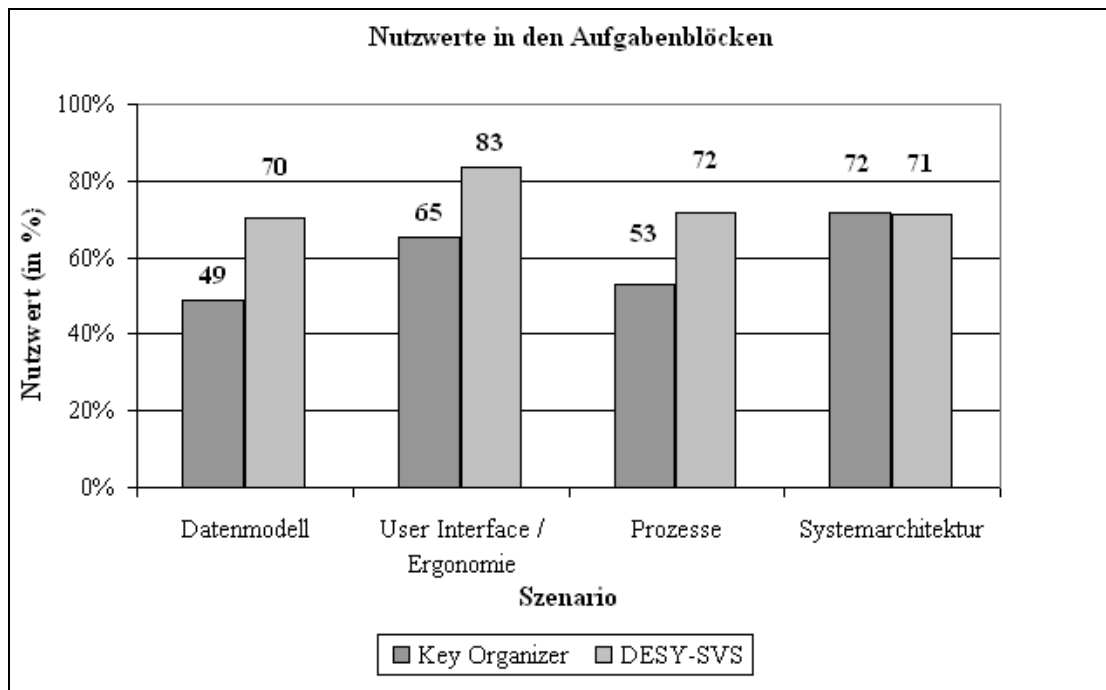


Abb. 36: Systembewertung in den Aufgabenblöcken (eigene Darstellung)

Der Grafik in Abbildung 36 ist zu entnehmen, dass das SVS der Firma Horstick in allen Bereichen eine um ca. 20 % höhere Akzeptanz bei den Testanwendern erzielen kann. Lediglich im Aufgabenblock *Systemarchitektur* wurde der *Key Organizer* um 1 % besser als das DESY-SVS bewertet. Gleichzeitig ist zu bemerken, dass das SVS der Horstick-Entwickler in allen Bereichen über der 50%-Grenze liegt und ein Scheitern des Projektes somit ausgeschlossen werden kann. Anders hingegen sieht es bei der Kaufsoftware *Key Organizer* aus. Das Schlüsselverwaltungssystem kann die Testanwender mit seiner Leistung in den Aufgabenblöcken *Datenmodell* (49 %) und

Prozesse (53 %) nicht überzeugen und verfehlt die 50%-Grenze bzw. überschreitet sie nur knapp. Bei dem DESY-SVS ist davon auszugehen, dass die Leistungsfähigkeit im Laufe des Projektes noch steigerungsfähig ist, da einige Punkte aus dem Anforderungskatalog zum Zeitpunkt des Benchmarkings noch nicht umgesetzt waren.

7.2.4 Bewertung in den Szenarien

In den Grafiken 37 und 38 werden die Nutzwerte durch eine Analyse der einzelnen Szenarien einer noch detaillierteren Betrachtungsweise unterzogen. Hier lassen sich Schwachstellen gezielter identifizieren.

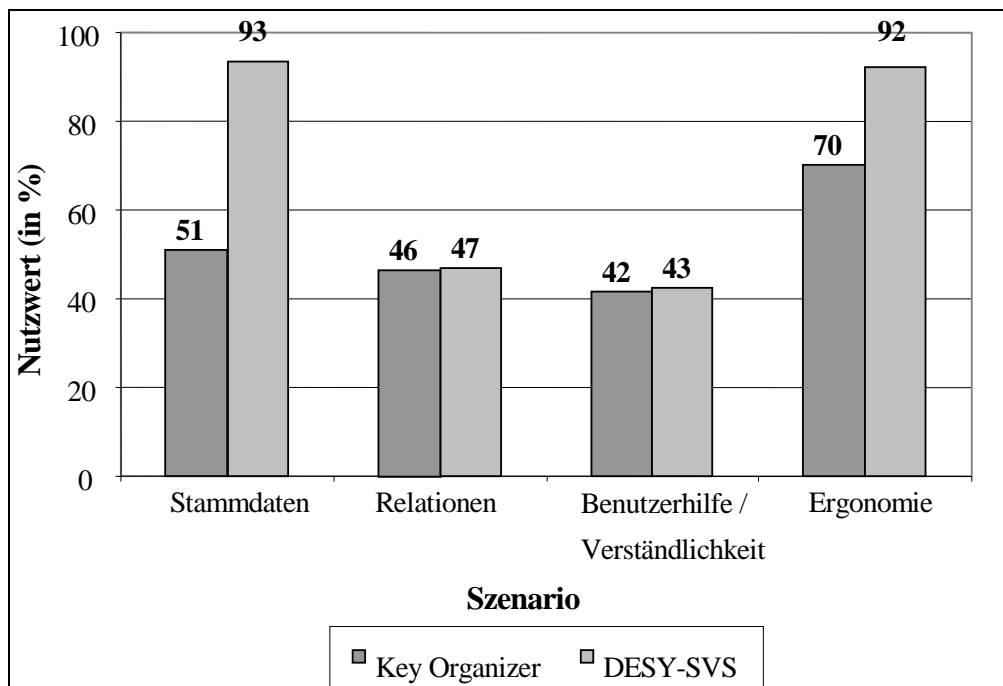


Abb. 37: Systembewertung in den Szenarien (Teil I, eigene Darstellung)

Aus Grafik 37 lässt sich ablesen, dass beide Systeme die Punkte *Relationen* und *Benutzerhilfe / Verständlichkeit* aus Sicht der Testanwender nicht erfüllen können. Hier muss es bei dem DESY-SVS innerhalb des weiteren Projektverlaufes zu Nachbesserungen kommen, wohingegen dies bei der Kaufsoftware nicht möglich ist. Die Projektziele könnten mit dem *Key Organizer* in diesen Szenarien sowie bei den Stammdaten nicht erfüllt werden. In den Kategorien *Stammdaten* und *Ergonomie* kann das DESY-SVS die Testanwender überzeugen. Die Akzeptanz dieser Szenarien erreicht über 90 % und liegt mit 40 % (*Stammdaten*) bzw. 20 % (*Ergonomie*) deutlich vor der Kaufsoftware *Key Organizer*. Dieser Leistungsvorsprung kann dahingehend begründet werden, dass das DESY-SVS auf Kundenwünsche angepasst wurde und somit den Anforderungen eher gerecht werden kann, als die universell verwendbare Software *Key Organizer*.

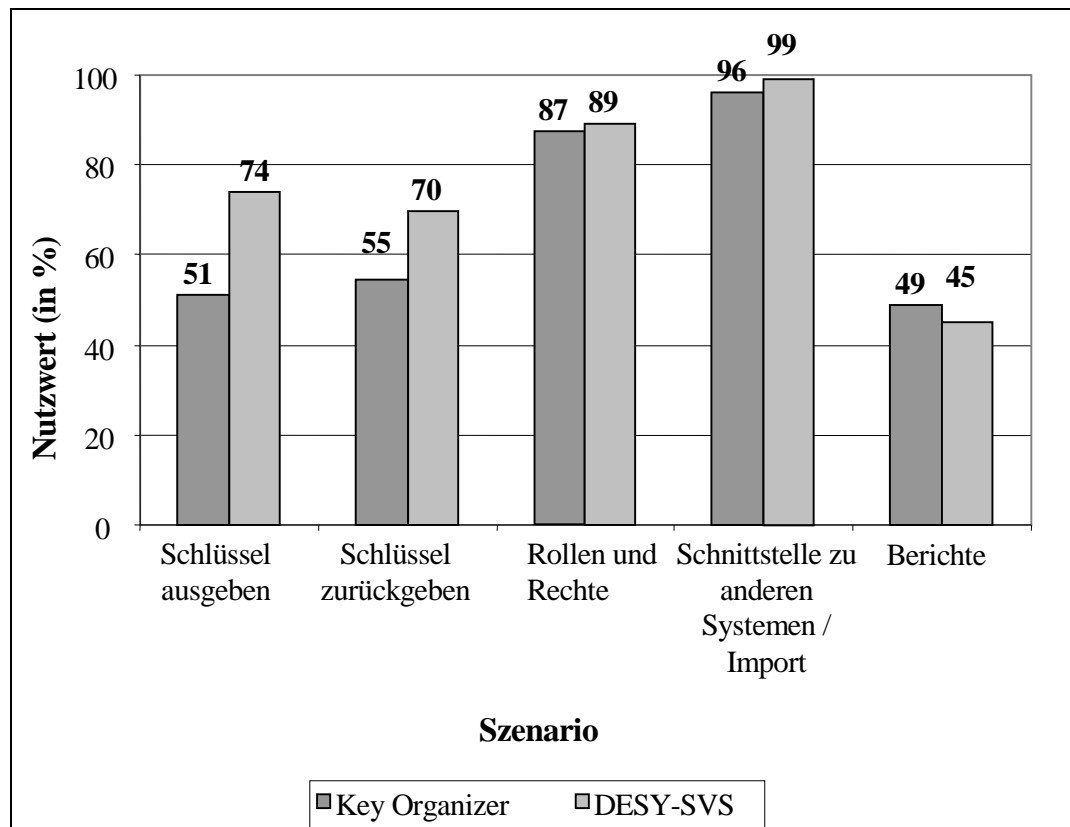


Abb. 38: Systembewertung in den Szenarien (Teil II, eigene Darstellung)

Abbildung 38 zeigt, dass das Produkt der Firma Horstick in den Szenarien *Schlüssel ausgeben* und *Schlüssel zurückgeben* bei den Testanwendern eine Akzeptanz von über 70 % erreicht und das Testpublikum in den Kategorien *Rollen und Rechte* sowie *Schnittstelle zu anderen Systemen / Import* nahezu komplett zufrieden stellen kann. Eine deutliche Schwachstelle ist jedoch in der Kategorie *Berichte* zu erkennen. Allerdings ist hierzu zu bemerken, dass dieser Aufgabenbereich der Firma Horstick zum Zeitpunkt des Benchmarkings noch nicht als Spezifikation vorlag und somit noch nicht umgesetzt war.

7.3 Zusammenfassung der Ergebnisse

Abschließend kann zusammengefasst werden, dass das maßgeschneiderte SVS bereits in einem frühen Entwicklungsstadium durchaus mit einer Kaufsoftware mithalten und es in seiner Leistung sogar überbieten kann. In der Gesamtbetrachtung wird deutlich, dass das DESY-SVS die Projektziele in jedem Fall, die Kaufsoftware jedoch nur minimal erreichen kann. Eine differenziertere Leistungsbetrachtung ermöglicht die Identifizierung von Schwachstellen in den einzelnen Aufgabenblöcken und Szenarien. Im Falle des DESY-SVS können Mängel erkannt und in der weiteren Systementwicklung behoben werden.

Die Ergebnisse des Benchmark-Tests müssen dennoch kritisch betrachtet werden. Bei den Testanwendern handelt es sich zwar um unvoreingenommene Personen, die jedoch nur geringes Fachwissen im Bezug auf den Bereich der Schlüsselverwaltung besitzen. Sie sind nicht ausreichend mit der Materie vertraut, um die Leistungen der Systeme wirklich beurteilen zu können. Hinzu kommt, dass die Gewichtung der Aufgabenblöcke und Szenarien von Projektteilnehmern der Gruppe IPP vergeben wurde, die zwar ein gewisses Beurteilungsvermögen zum Thema Schlüsselverwaltung besitzen, letztendlich aber keine Facility-Management-Experten oder Schlüsselverwalter sind.

8 Kosten- und Aufwandsschätzung für die SVS-Entwicklung

Entscheidet man sich für den Erwerb und den Einsatz einer Kaufsoftware, können die Kosten im Vorfeld mühelos eingeholt werden. Bei einer maßgeschneiderten Entwicklung auf Basis einer Spezifikation ist entweder ein Pauschalpreis festzulegen oder die Entwickler sind nach Aufwand zu vergüten. In diesem Fall entschied sich DESY für eine aufwandsbezogene Entlohnung. Um einen genauen Kostenüberblick zu erhalten und das zugestandene Projektbudget nicht zu überschreiten, war im Vorwege eine Abschätzung der Entwicklungskosten notwendig. Dies erfolgte auf Basis einer Funktionspunktanalyse. Die Durchführung einer entsprechenden Aufwands- und Kostenschätzung stellt den Inhalt dieses Kapitels dar. Es wird die Herangehensweise erläutert und eine Abschätzung des Aufwandes für die Entwicklung der Schlüsselausgabekomponente durchgeführt. In den weiteren Kapitelabschnitten folgen eine Abschätzung zur Realisierung der Schlüsselrückgabefunktionalitäten sowie der gesamten SVS-Entwicklung. Abschließend wird eine Ergebnisbetrachtung zur Kosten- und Aufwandsschätzung vorgenommen.

8.1 Durchführung der Aufwandsschätzung

Entsprechend der in Kapitel 4.7 beschriebenen Herangehensweise wird in diesem Kapitel eine Aufwandsschätzung für die Entwicklung der Schlüsselausgabefunktionen vorgenommen. Zunächst soll die zu erwartende Team Performance – die Anzahl der Funktionspunkte, die der Entwickler der Firma Horstick voraussichtlich an einem Tag umsetzen kann – abgeschätzt werden. In der Praxis können oftmals Erfahrungswerte verwendet werden. Da diese jedoch nicht vorlagen, wurde die Team Performance mit Hilfe eines praktischen Beispiels ermittelt. Als Basis dienten hierfür eine Eingabe- (Abb. 39) und eine Ausgabemaske (Abb. 40).

Die Beispiel-Eingabe besitzt zwei Eingabefelder. Sie kann somit als einfach eingestuft und gemäß der Matrix in Kapitel 4.7 mit 3 Funktionspunkten versehen werden.

Jugendherbergssuche

Kleiner Hinweis:
Sie müssen nicht alle Felder ausfüllen, um ein Suchergebnis zu erhalten.

Name oder Ort der JH:

Bundesland:

Abb. 39: Beispielhafte Eingabemaske (Screenshot)

Die Ausgabemaske liefert fünf Datensätze mit je drei Feldern. Sie kann ebenfalls als einfach eingestuft werden und erhält gemäß der Matrix in Kapitel 4.7 4 Funktionspunkte.

Das Ergebnis Ihrer Jugendherbergssuche

- 🏠 Online-Buchung (nicht für Gruppen mit mehr als 6 Personen)*
- 🏠 Reservierungsanfrage und teilweise Online-Buchung
- 🏠 Telefon, Fax, E-Mail

Saarland

- 🏠 **Hohenburg-Jugendherberge Jugendgästehaus Homburg/Saar**
66424 Homburg
(Lvb Rheinland-Pfalz / Saarland)
- 🏠 **Jugendherberge an der Saarschleife Jugendgästehaus Dreisbach**
66693 Mettlach-Dreisbach
(Lvb Rheinland-Pfalz / Saarland)
- 🏠 **Europa-Jugendherberge Jugendgästehaus Saarbrücken**
66123 Saarbrücken
(Lvb Rheinland-Pfalz / Saarland)
- 🏠 **Schaumberg-Jugendherberge Jugendgästehaus Tholey**
66636 Tholey
(Lvb Rheinland-Pfalz / Saarland)
- 🏠 **Jugendherberge Weiskirchen**
66709 Weiskirchen
(Lvb Rheinland-Pfalz / Saarland)

Abb. 40: Beispielhafte Ausgabemaske (Screenshot)

Mit einer einfachen Eingabe- (3 FP) und einer einfachen Ausgabemaske (4 FP) ergibt sich folgende Rechnung für den Wert *UFP*:

$$UFP = 3 + 4$$

$$UFP = 7$$

Es wird geschätzt, dass für die Umsetzung der beiden Masken zwei Arbeitstage benötigt werden ($A = 2$). Ausgehend von dieser Annahme wird die Team Performance (FP pro Tag) berechnet:

$$TP = UFP / A$$

$$TP = 7 / 2$$

$$TP = 3,5$$

$$UFP = 7$$

$$A = 2$$

Dieser Wert [$TP = 3,5$] wird in der Berechnung des Aufwandes für die Entwicklung der Schlüsselverwaltungsfunktionen im Folgenden als Orientierung verwendet.

8.1.1 Aufwandsschätzung für die Realisierung der Schlüsselausgabefunktionen

Der folgende Kapitelabschnitt stellt schrittweise die Berechnung des Aufwandes zur Realisierung der Schlüsselausgabefunktionen vor. Hierzu wird das in Kapitel 4.7 definierte Schema angewandt.

1) Funktionstypen identifizieren

Bevor mit der Aufwandsschätzung begonnen werden kann, müssen zunächst alle vorkommenden Funktionstypen identifiziert werden. Dies geschieht ausgehend von dem in Kapitel 6.1 erstellten Systementwurf (siehe auch Abb. 41).

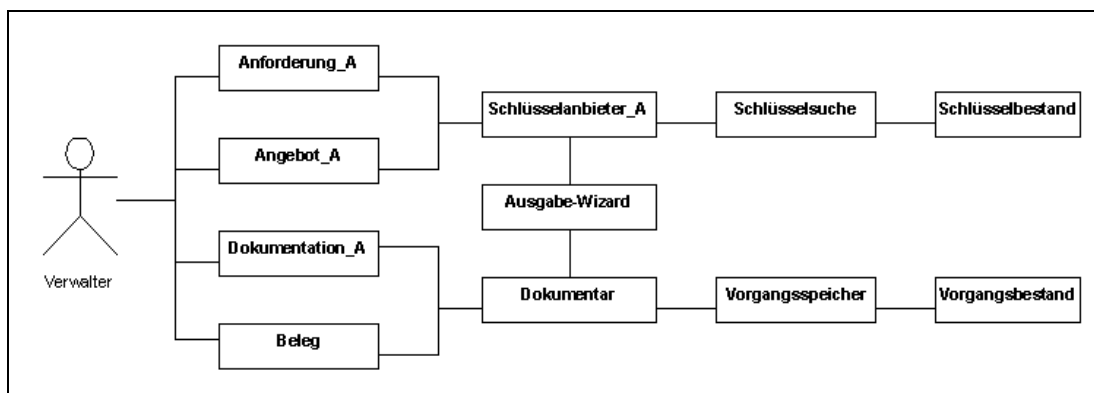


Abb. 41: Systemstruktur der Schlüsselausgabekomponente (eigene Darstellung)

Insgesamt vier Ein- und Ausgabemasken bilden bei der Systemstruktur der Schlüssel-
ausgabe die Schnittstelle zum Klienten:

- Anforderung_A
- Angebot_A
- Dokumentation_A
- Beleg

Koordiniert wird das Verhalten der Interface-Klassen von den Steuerungsklassen. Die
in Abb. 41 dargestellte Systemstruktur weist zwei solcher Klassen auf:

- Schlüsselanbieter_A
- Dokumentar

Zwei weitere Klassen sind für das Suchen von Schlüsseln und das Speichern von
Vorgängen verantwortlich:

- Schlüsselsuche
- Vorgangsspeicher

Beide Steuerungsklassen greifen zur Umsetzung ihrer Aufgaben auf Datenspeicher
zu:

- Schlüsselbestand
- Vorgangsbestand

Der komplette Prozess wird von einer weiteren Steuerungsklasse koordiniert:

- Ausgabe-Wizard

2) Komplexität bestimmen

Alle vier Eingabe- bzw. Ausgabemasken verfügen über eine geringe Anzahl an Da-
tenfeldern und werden daher als einfach in ihrer Komplexität eingestuft. Auch die
Komplexität der Interface-Steuerungsklassen *Schlüsselanbieter_A* und *Dokumentar*
wird als einfach definiert. Die beiden Steuerungsklassen *Schlüsselsuche* und *Vor-
gangsspeicher* werden mit einer hohen Komplexität eingestuft, da sie eine größere
Menge an Informationen zu verarbeiten haben. Die Komplexität der Speicherklassen
Schlüsselbestand und *Vorgangsbestand* wird als hoch ausgezeichnet, da sowohl die
Anzahl der Datensätze als auch die der Datenfelder relativ hoch ist. Die Komplexität
der Klasse *Ausgabe-Wizard* wird als gering eingestuft. Werden die Komplexitäten
aller Klassen aus Abb. 41 zusammengefasst, ergibt sich folgende Matrix:

Tabelle 4: Funktionsklassen der Schlüsselausgabekomponente

	Anforderung_A	Angebot_A	Dokumentation_A	Beleg	Schlüsselanbieter_A	Dokumentar	Schlüsselsuche	Vorgangsspeicher	Asugabe-Wizard	Schlüsselbestand	Vorgangsbestand	Anzahl		
												-	o	+
Eingabe	-	-	-									3	0	0
Ausgabe		-	-	-								3	0	0
Speicher										+	+	0	0	2
Rechnung					-	-	+	+	-			3	0	2
Export												0	0	0

3) UFP berechnen

Um die Summe der Funktionspunkte zu bilden, wird die Tabelle aus Kapitel 4.7 zu Rate gezogen. Mit Hilfe dieser Tabelle wird einem Funktionstypen eines gewissen Schwierigkeitsgrades eine bestimmte Anzahl an Funktionspunkten zugeordnet.

Tabelle 5: Matrix zur Ermittlung der Funktionspunkte

Funktionstyp	Schwierigkeitsgrad		
	einfach (-)	mittel (o)	komplex (+)
Eingabe	3	4	6
Ausgabe	4	5	7
Speicher	3	4	6
Rechnung	7	10	15
Schnittstelle / Export	5	7	10

Es ergibt sich folgende Rechnung zur Bestimmung der Gesamtfunktionspunkte *UFP*:

$$UFP_{SA} = (3 * 3) + (3 * 4) + (2 * 6) + (3 * 7) + (2 * 15)$$

$$UFP_{SA} = 84$$

Die Realisierung der Funktionalitäten zur Unterstützung der Schlüsselausgabe umfasst insgesamt 84 Funktionspunkte.

4) Korrekturwert berechnen

Zur Bestimmung des Einflusses des Entwicklungsumfeldes werden alle 14 Merkmale mit einem Durchschnittswert von 3 bewertet. Daraus ergibt sich folgender Wert für den *TDI*:

$$\text{TDI} = 14 * 3$$

$$\text{TDI} = 42$$

Mit Hilfe des *TDI* kann der Korrekturwert *VAF* errechnet werden. Es ergibt sich folgende Rechnung:

$$\text{VAF} = (\text{TDI} * 0,01) + 0,65$$

$$\text{VAF} = (42 * 0,01) + 0,65$$

$$\text{VAF} = 1,07$$

$$\text{VAF} \approx 1,1$$

5) + 6) Aufwand berechnen

Die Team Performance wird entsprechend der Rechnung eingangs dieses Kapitels übernommen:

$$\text{TP} = 3,5 \text{ FP pro Tag}$$

Die Aufwandsbestimmung für die Entwicklung der Klassen der Schlüsselausgabekomponente erfolgt mit der Formel:

$$A_{SA} = \text{UFP}_{SA} * \text{VAF} / \text{TP}$$

$$A_{SA} = 84 * 1,1 / 3,5$$

$$A_{SA} = 26,4 \text{ Tage (inkl. Spezifikation und Test)}$$

$$\text{UFP}_{SA} = 84$$

$$\text{VAF} = 1,1$$

$$\text{TP} = 3,5$$

Um die Funktionalitäten zur Unterstützung des Schlüsselausgabeprozesses zu realisieren, ist ein Aufwand (A_{SA}) von 26,4 Tagen nötig. Diese Zahl umfasst sowohl die Entwicklung als auch den Aufwand für die Erstellung einer Spezifikation und die Durchführung eines Systemtests. Der pure Entwicklungsaufwand wird in Kapitel 8.1.3 aufgeschlüsselt.

8.1.2 Aufwandsschätzung für die Realisierung der Schlüsselrückgabefunktionen

Ebenso wie im vorangegangenen Unterkapitel wird im Folgenden der Aufwand zur Realisierung der Schlüsselrückgabefunktionen abgeschätzt. Es wird weiterhin nach dem in Kapitel 4.7 definierten Verfahren vorgegangen.

1) Funktionstypen identifizieren

Um die Funktionstypen der Schlüsselrückgabe zu bestimmen, ist, wie in Kapitel 8.1.1 beschrieben, ein entsprechender Systementwurf als Basis zu verwenden. Ausgehend von einem entsprechenden Aktivitätsdiagramm, ergibt sich ein Systemdesign für die Schlüsselrückgabekomponente gemäß Abb. 42.

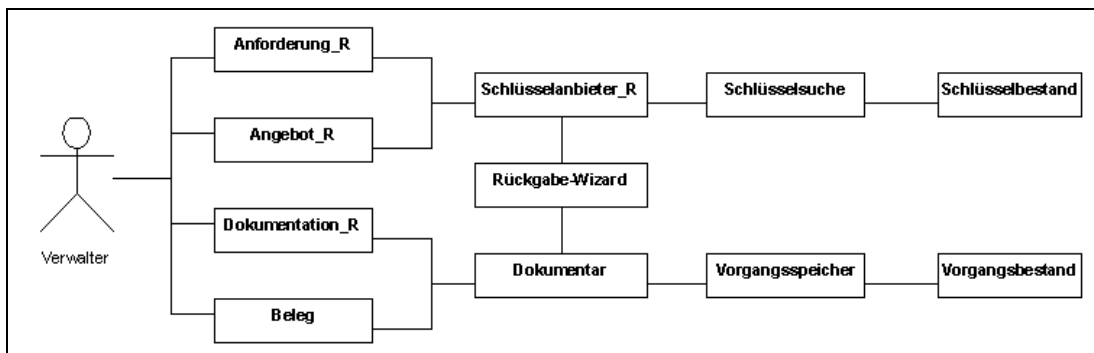


Abb. 42: Systemstruktur der Schlüsselrückgabekomponente (eigene Darstellung)

Bei einem Vergleich der beiden Systemarchitekturen in Abb. 41 und 42 ergibt sich, dass folgende für die Umsetzung der Schlüsselausgabekomponente entwickelten Elemente für die Realisierung der Rückgabekomponente wiederverwendet werden können:

- Beleg
- Dokumentar
- Schlüsselsuche
- Vorgangsspeicher
- Schlüsselbestand
- Vorgangsbestand

Die Interface-Klassen der Schlüsselausgabekomponente können dagegen nicht für die Schlüsselrückgabefunktionalität wiederverwendet werden. Für diesen Prozess sind separate Ein- und Ausgabemasken zu entwickeln:

- Anforderung_R
- Angebot_R
- Dokumentation_R

Als weitere zu entwickelnde Elemente für die Schlüsselrückgabe verbleiben die Steuerungsklassen:

- Schlüsselanbieter_R
- Rückgabe-Wizard

2) Komplexität bestimmen

Die drei Interface-Klassen *Anforderung_R*, *Angebot_R* und *Dokumentation_R* der Schlüsselrückgabekomponente weisen eine geringe Anzahl an Datenfeldern auf und werden daher mit einer geringen Komplexität ausgezeichnet. Auch die beiden Elemente *Schlüsselanbieter_R* und *Rückgabe-Wizard* werden mit einer geringen Komplexität versehen. Zusammenfassen ergibt sich folgende Matrix mit der Gesamtzahl zu entwickelnder Klassen für die Umsetzung des IS-gestützten Rückgabeprozesses:

Tabelle 6: Funktionsklassen der Schlüsselrückgabekomponente

	Anforderung_R	Angebot_R	Dokumentation_R	Beleg	Schlüsselanbieter_R	Dokumentar	Schlüsselsuche	Vorgangsspeicher	Rückgabe-Wizard	Schlüsselbestand	Vorgangsbestand	Anzahl		
												-	o	+
Eingabe	-	-	-									3	0	0
Ausgabe		-	-									2	0	0
Speicher												0	0	0
Rechnung					-				-			2	0	0
Export												0	0	0

3) UFP berechnen

Um die Gesamtfunktionspunkte gemäß der Tabelle aus Kapitel 4.7 zu berechnen, wird folgende Formel angewandt:

$$UFP_{SR} = (3 * 3) + (2 * 4) + (2 * 7)$$

$$UFP_{SR} = 31$$

Bei der Entwicklung der Funktionalitäten zur Unterstützung des Schlüsselrückgabeprozesses sind 31 Funktionspunkte zu realisieren.

4) Korrekturwert berechnen

Der Einfluss des Entwicklungsumfeldes und die Team Performance lauten weiterhin:

$$\text{VAF} = 1,1$$

$$\text{TP} = 3,5 \text{ FP pro Tag}$$

5) + 6) Aufwand berechnen

Trägt man die Werte UFP , VAF und TDI in die Formel zur Bestimmung des Aufwandes ein, ergibt sich folgende Rechnung:

$$A_{SR} = UFP_{SR} * VAF / TP$$

$$A_{SR} = 31 * 1,1 / 3,5$$

$$A_{SR} = 9,7 \text{ Tage (inkl. Spezifikation und Test)}$$

$$UFP_{SR} = 31$$

$$VAF = 1,1$$

$$TP = 3,5$$

Insgesamt ist ein Aufwand (A_{SR}) von 9,7 Tagen für die Realisierung der Schlüsselrückgabefunktionalitäten notwendig. Dieser Wert umfasst sowohl die Entwicklung als auch den Aufwand für die Erstellung der Spezifikation und einen abschließenden Systemtest.

8.1.3 Aufwandsschätzung der gesamten SVS-Entwicklung

Um eine Aufwandsschätzung für die gesamte Systementwicklung vornehmen zu können, müssen die Aufwände für die Umsetzung der weiteren Prozessfunktionalitäten der Schlüsselverwaltung bestimmt werden.

S-Verlust, S-Fund, S-Defekt, Z-Defekt, Z-Rückgabe

Auf Grund der ähnlichen Struktur der Aktivitätsdiagramme wird davon ausgegangen, dass die folgenden Aufwände dem zur Realisierung der Schlüsselrückgabefunktionalitäten entsprechen:

- Schlüsselverlustbearbeitung (A_V)
- Schlüsselfundbearbeitung (A_F)
- Schlüsseldefektbearbeitung (A_{SD})
- Zylinderdefektbearbeitung (A_{ZD})
- Zylinderrückgabe (A_{ZR})

Für diese fünf Prozesse ergibt sich ausgehend von der Berechnung des Aufwandes für die Realisierung der Schlüsselrückgabekomponente folgender Ansatz:

$$A_{V_F_SD_ZD_ZR} = A_{SR}$$

$$A_{SR} = 31 * 1,1 / 3,5$$

$$A_{SR} = 9,7$$

$$A_{V_F_SD_ZD_ZR} = (31 * 1,1 / 3,5) * 5$$

$$A_{V_F_SD_ZD_ZR} = 48,5 \text{ Tage (inkl. Spezifikation und Test)}$$

Die oben aufgelisteten fünf Prozesse der Schlüsselverwaltung nehmen für ihre Realisierung einen Aufwand ($A_{V_F_SD_ZD_ZR}$) von 48,5 Tagen in Anspruch. Dieser Wert steht jedoch nicht allein für den Entwicklungsaufwand, sondern umfasst weiterhin den Aufwand für die Erstellung der Spezifikation und die Durchführung eines Systemtests.

Zylinderausgabe

Ebenfalls auf Grund der ähnlichen Struktur der Aktivitätsdiagramme wird ausgehend von der Schlüsselausgabe ein äquivalenter Realisierungsaufwand für die Funktionen der Zylinderausgabe (A_{ZA}) angenommen. Für diesen Prozess gilt folgende Rechnung:

$$A_{SA} = A_{ZA}$$

$$A_{SA} = 84 * 1,1 / 3,5$$

$$A_{SA} = 26,4$$

$$A_{ZA} = 84 * 1,1 / 3,5$$

$$A_{ZA} = 26,4 \text{ Tage (inkl. Spezifikation und Test)}$$

Als Aufwand (A_{ZA}) für die Umsetzung der Zylinderausgabefunktionen ergibt sich aus der obigen Rechnung ein Wert von 26,4 Tagen – ebenfalls inklusive Spezifikationserstellung und Systemtest.

Stammdatenerfassung

Als letzter Prozess ist in die Berechnung des Entwicklungsaufwandes die Stammdatenerfassung für Schlüssel- und Zylinderexemplare hinzuzufügen. Da die Prozesse einander entsprechen, wird hier exemplarisch der Aufwand für die Erfassung der Schlüsselstammdaten berechnet. Klassendiagramm 43 stellt die zu entwickelnden Systemklassen vor.

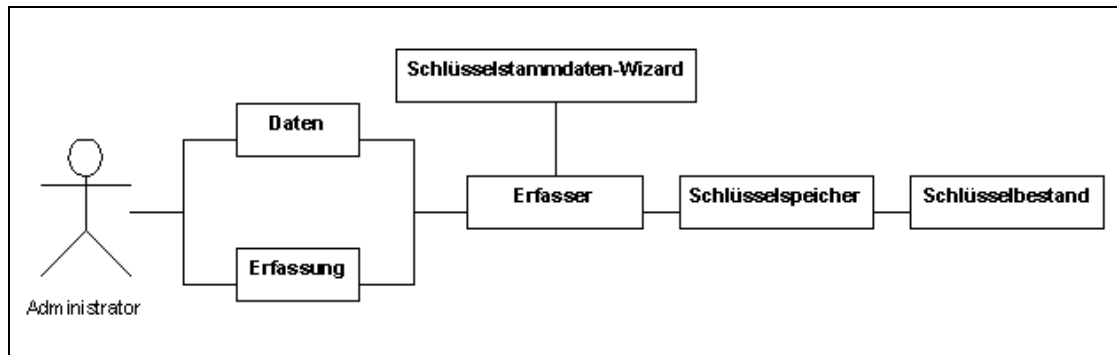


Abb. 43: Systemstruktur der Schlüsselstammdatenerfassungskomponente (eigene Darstellung)

1) Funktionstypen identifizieren

Um die Komponente zur Unterstützung der Schlüsselstammdatenerfassung realisieren zu können, ist die Entwicklung folgender Klassen notwendig:

- Daten (Interface-Klasse)
- Erfassung (Interface-Klasse)
- Erfasser (Steuerungsklasse)
- Schlüsselstammdaten-Wizard (Steuerungsklasse)
- Schlüsselspeicher (Speicher)

2) Komplexität bestimmen

Die Klasse *Schlüsselbestand* wurde bereits zur Realisierung der vorher beschriebenen Prozesse entwickelt und kann daher für die Aufwandsschätzung der Schlüsselstammdatenerfassung unberücksichtigt bleiben.

Den Interface-Klassen *Daten* und *Erfassung* wird auf Grund einer geringen Anzahl an Datenfeldern jeweils eine geringe Komplexität zugewiesen. Die Steuerungsklassen *Erfasser* und *Schlüsselstammdaten-Wizard* zeichnen sich durch eine niedrige, die Klasse *Schlüsselspeicher* durch eine hohe Komplexität aus.

Folgende Matrix zeigt einen Überblick über die zu entwickelnden Klassen und deren Komplexität.

Tabelle 7: Funktionsklassen der Stammdatenerfassungskomponente

							Anzahl		
	Daten	Erfassung	Erfasser	Schlüsselstammdaten-Wizard	Schlüsselspeicher	Schlüsselbestand	-	o	+
Eingabe	-						1	0	0
Ausgabe		-					2	0	0
Speicher							0	0	0
Rechnung			-	-	+		2	0	1
Export							0	0	0

3) UFP berechnen

Die Funktionspunkte ergeben sich durch die Berechnung folgender Formel:

$$UFP_{SStD} = (1 * 3) + (1 * 4) + (2 * 7) + (1 * 15)$$

$$UFP_{SStD} = 36$$

Für die Realisierung der Komponente zur Unterstützung des Schlüsselstammdatenerfassungsprozesses sind entsprechend der obigen Rechnung 36 Funktionspunkte umzusetzen.

4) Korrekturwert berechnen

Der Einfluss des Entwicklungsumfeldes sowie die Team Performance gelten für die Realisierung der Stammdatenerfassungsfunktionalitäten ebenso wie für die vorher bestimmten Komponenten.

$$VAF = 1,1$$

$$TP = 3,5 \text{ FP pro Tag}$$

5) + 6) Aufwand berechnen

Die Aufwandsschätzung für die Entwicklung der Komponente zur Unterstützung der Schlüsselstammdatenerfassung ergibt sich aus der Berechnung folgender Formel:

$$A_{SSID} = UFP_{SSID} * VAF / TP$$

$$A_{SSID} = 36 * 1,1 / 3,5$$

$$A_{SSID} = 11,3 \text{ Tage (inkl. Spezifikation und Test)}$$

$$UFP_{SSID} = 36$$

$$VAF = 1,1$$

$$TP = 3,5$$

Für die Umsetzung des Prozesses zur Erfassung der Schlüsselstammdaten werden gemäß der obigen Rechnung 11,3 Tage (A_{SSID}) benötigt.

Es wird davon ausgegangen, dass für die Realisierung der Zylinderstammdatenerfassung der selbe Aufwand (A_{ZSID}) von 11,3 Tagen gilt.

Berechnung des Gesamtaufwandes zur Entwicklung des DESY-SVS

Um den Gesamtaufwand (A_{ges}) für die Entwicklung eines Systems zur Verwaltung von Schlüsseln und Zylindern zu berechnen, sind folgende Aufwände zu addieren:

$$A_{SA} = 26,4$$

$$A_{SR} = 9,7$$

$$A_{V_F_SD_ZD_RZ} = 48,5$$

$$A_{ZA} = 26,4$$

$$A_{SSID} = 11,3$$

$$A_{ZSID} = 11,3$$

$$A_{ges} = A_{SA} + A_{SR} + A_{V_F_SD_ZD_ZR} + A_{ZA} + A_{SSID} + A_{ZSID}$$

$$A_{ges} = 26,4 + 9,7 + 48,5 + 26,4 + 11,3 + 11,3$$

$$A_{ges} = 136,6 \text{ Tage (inkl. Spezifikation und Test)}$$

Um das Schlüsselverwaltungssystem vollständig zu entwickeln, sind 136,6 Tage nötig. Diese Zahl beinhaltet neben dem zeitlichen Aufwand für die technische Umsetzung weiterhin die Aufwände für die Erstellung einer Spezifikation (A_{Spez}) sowie für die Durchführung eines Systemtests (A_{Test}). Es wird geschätzt, dass pro Prozess

- 2,5 Tage für die Zusammenstellung der Spezifikation (A_{Spez})
- 3 Tage für das Testen des Systems (A_{Test})

aufgewendet werden müssen. Insgesamt fallen diese Tätigkeiten für zehn Prozesse an und sind von A_{ges} abzuziehen. Daraus resultiert folgende Rechnung:

$A = A_{\text{ges}} - (10 * A_{\text{Spez}}) - (10 * A_{\text{Test}})$	$A_{\text{ges}} = 136,6$
$A = 136,6 - (10 * 2,5) - (10 * 3)$	$A_{\text{Spez}} = 2,5$
$A = 81,6 \text{ Tage}$	$A_{\text{Test}} = 3$

Als Gesamtaufwand (A) für die Entwicklung des SVS ergeben sich 81,6 Tage. Damit wäre der zeitliche Rahmen für die Systementwicklung abgeschätzt. Davon ausgehend können die Kosten (K) für eine entsprechende Entwicklung der GIS/FMS-Komponente errechnet werden.

$K = A * K_{\text{Tag}}$	$A = 81,6$
$K = 81,6 * 500$	$K_{\text{Tag}} = 500$
$K = 40.800 \text{ €}$	

Bei einem Betrag von 500 € Entwicklungskosten pro Tag (K_{Tag}) ergibt sich eine Summe von 40.800 €, die dem Entwicklerteam der Firma Horstick für ihre Arbeit zugestanden wird.

8.2 Ergebnisse der Kostenschätzung

Um das DESY-SVS zu realisieren, ist gemäß der Berechnung in Kapitel 8.1.3 ein Aufwand (A) von 81,6 Entwicklertagen nötig. Damit kostet die Realisierung des Systems bei einem Tagessatz von 500 € für den Entwickler 40.800 €. Diese Zahl ist jedoch nicht mit den Projektkosten insgesamt gleichzustellen. Bei einer Gesamtbetrachtung der Projektkosten sind weiterhin Fahrt- und Übernachtungskosten des Entwicklerteams zur Besprechung des Projektrahmens, zur Durchführung eines Workshops und zur Vorstellung eines ersten SVS-Entwurfes zu berücksichtigen. Außerdem wird das Projekt nicht nur durch erbrachte Leistungen von Seiten der Entwickler zu einem Abschluss geführt. Auch im Hause des Deutschen Elektronen-Synchrotrons werden Arbeitskräfte mit dem Projekt betraut. So sind die Projektmitglieder DESYs unter anderem für die Anforderungserhebung, die Erstellung der Spezifikation und für den abschließenden Test des Systems zuständig und werden mit einer entsprechenden Vergütung entlohnt.

Das Ergebnis der Kostenschätzung für die Systementwicklung kann keineswegs als ein reeller Maßstab betrachtet, sondern muss hinterfragt werden. Die Rechnung basiert auf vielen unsicheren Angaben wie z.B. dem Value added Factor und der geschätzten Umsetzung der Funktionspunkte pro Tag – der Team Performance. Die

Kostenschätzung kann daher nur als Schätzung bezeichnet werden, die durchaus von den reellen Entwicklungskosten abweichen kann. Je nachdem, auf welche Sprache zur technischen Umsetzung zurückgegriffen wird, können bereits deutlich mehr oder erheblich weniger Funktionspunkte pro Tag realisiert werden als in der Kostenschätzung angenommen. Um eine genauere Abschätzung der Kosten und der Umsetzungsdauer für eventuell kommende Systemerweiterungen vornehmen zu können, ist innerhalb dieses Projektes ein Vergleich zwischen angenommener und tatsächlich benötigter Entwicklungszeit vorzunehmen.

Die Dauer für die technische Umsetzung der Funktionalitäten des Schlüsselausgabe- und Schlüsselrückgabeprozesses inklusive Spezifikation und Systemtest wurde innerhalb der Kostenschätzung in Kapitel 8.1.1 und 8.1.2 wie folgt berechnet:

$$A_{SA} = 26,4$$

$$A_{SR} = 9,7$$

Für beide Prozesse zusammen ergibt sich folgender Aufwand (A_{SA_SR}):

$$A_{SA_SR} = 26,4 + 9,7$$

$$A_{SA_SR} = 36,1$$

Um den puren Entwicklungsaufwand zu erhalten, sind folgende Werte abzuziehen:

- 2,5 Tage für die Erstellung der Spezifikation (A_{Spez})
- 3 Tage für die Durchführung des Systemtests (A_{Test})

Davon ausgehend ergibt sich für die technische Umsetzung der Funktionalitäten für die Schlüsselausgabe und Schlüsselrückgabe folgender geschätzter Aufwand $A_{SA_SR^*}$:

$$A_{SA_SR^*} = A_{SA_SR} - (2 * A_{Spez}) - (2 * A_{Test})$$

$$A_{SA_SR^*} = 36,1 - (2 * 2,5) - (2 * 3)$$

$$A_{SA_SR^*} = 25,1$$

$$A_{SA_SR} = 36,1$$

$$A_{Spez} = 2,5$$

$$A_{Test} = 3$$

Exklusive der Spezifikationserstellung und des Systemtests ergibt sich ein purer Entwicklungsaufwand ($A_{SA_SR^*}$) von 25,1 Tagen für die Realisierung des Schlüsselausgabe- und Schlüsselrückgabeprozesses. Reell hat das Entwicklerteam für die Realisierung der Schlüsselausgabe- und Schlüsselrückgabefunktionalitäten jedoch 15 Tage ($A_{SA_SR_Real}$) benötigt. Gegenüber der Kostenschätzung ergibt dies eine Differenz ($A_{SA_SR_Diff}$) von 10 Entwicklertagen.

$A_{SA_SR_Diff} = A_{SA_SR*} - A_{SA_SR_Real}$ $A_{SA_SR_Diff} = 25,1 - 15$ $A_{SA_SR_Diff} = 10,1$	$A_{SA_SR*} = 25,1$ $A_{SA_SR_Real} = 15$
--	--

Ausgehend von dieser Abweichung zwischen Realität und Aufwandsschätzung ergibt sich bei der Realisierung zweier Prozesses bereits eine Kostendifferenz ($K_{SA_SR_Diff}$) von 5.050 €

$K_{SA_SR_Diff} = A_{SA_SR_Diff} * K_{Tag}$ $K_{SA_SR_Diff} = 10,1 * 500$ $K_{SA_SR_Diff} = 5.050$	$A_{SA_SR_Diff} = 10,1$ $K_{Tag} = 500$
--	---

Weiterhin muss beachtet werden, dass einige Funktionsklassen sich zwar formal voneinander unterscheiden, aber prinzipiell kopiert und minimal abgeändert werden können. Innerhalb der oben durchgeführten Kostenschätzung wurden diese jedoch als neu zu entwickelnde Klassen betrachtet und gingen somit vollständig in die Gesamtrechnung ein. An diesem Punkt kann es in der Realität daher zu weiteren zeitlichen und somit kostenmäßigen Einsparungen gegenüber der Schätzrechnung kommen. So benötigten die Entwickler der Firma Horstick für die Umsetzung der Funktionen des Zylinderausgabe- und Zylinderrückgabeprozesses 6 Tage ($A_{ZA_ZR_Real}$), während die dazu äquivalenten Funktionen für die Schlüsselausgabe- und Schlüsselrückgabeprozesse innerhalb von 15 Tagen realisiert waren. Die Kostenschätzung veranschlagte für die technische Umsetzung der Prozesse hingegen 25,1 Tage (A_{ZA_ZR}). Hier schlägt demnach bereits erneut eine Differenz ($A_{ZA_ZR_Diff}$) von 19,1 Entwicklertagen zu Buche.

$A_{ZA_ZR_Diff} = A_{ZA_ZR} - A_{ZA_ZR_Real}$ $A_{ZA_ZR_Diff} = 25,1 - 6$ $A_{ZA_ZR_Diff} = 19,1$	$A_{ZA_ZR} = 25,1$ $A_{ZA_ZR_Real} = 6$
--	--

Diese Differenz gegenüber der Schätzung führt erneut zu einer Kostenabweichung, die bei den Funktionen zur Unterstützung des Zylinderausgabe- und Zylinderrückgabeprozesses 9.550 € ($K_{ZA_ZR_Diff}$) beträgt.

$K_{ZA_ZR_Diff} = A_{ZA_ZR_Diff} * K_{Tag}$ $K_{ZA_ZR_Diff} = 19,1 * 500$ $K_{ZA_ZR_Diff} = 9.550$	$A_{ZA_ZR_Diff} = 19,1$ $K_{Tag} = 500$
--	---

Abschließend betrachtet werden die tatsächlichen Entwicklungskosten durch die Summierung der Abweichungen in den einzelnen Prozessen um einige Tausend Euro von der Schätzung abweichen. Diese Erfahrungswerte machen es jedoch möglich, bei Systemerweiterungen in Zukunft eine treffendere Rechnung bezüglich des benötigten Aufwandes und der zu vergütenden Systementwicklung vorzunehmen. Dennoch ist zu beachten, dass es sich stets um eine Schätzung handeln wird, mit der man das reale Ergebnis im Vorfeld zwar simulieren, aber niemals garantieren kann.

9 Ausblick

Innerhalb dieses Projektes wurden Komponenten für die Unterstützung von zehn Schlüsselverwaltungsprozessen umgesetzt und eine davon – die Schlüsselausgabekomponente – in der vorliegenden Diplomarbeit mit Hilfe von Screenshots vorgestellt. Die Erstellung von Berichten wird mit einem speziellen Reporting-Tool erfolgen. Hier war daher keine Entwicklungsarbeit notwendig. Doch obwohl das System mit den erwähnten Funktionalitäten ausgestattet wurde und das Projekt abgeschlossen werden kann, sind für die Zukunft Systemerweiterungen notwendig, da mit diesen zehn Funktionen lediglich die Grundlagen der Schlüsselverwaltung abgedeckt werden. Dies wurde ganz bewusst entschieden, um den zeitlichen Rahmen des Projektes überschaubar zu halten und um das System möglichst schnell einsetzen zu können. Weniger notwendige Features sollen nach und nach hinzugefügt werden und das SVS stetig erweitern und optimieren. Spätestens wenn das SVS in der Abteilung V1 zum Einsatz kommt und das bisherige System *Big Key* ablöst, werden Wünsche zu Systemerweiterungen aufkommen. Doch bereits während des Projektes wurde die zukünftige Notwendigkeit einiger weiterer Systemeigenschaften diskutiert.

Für die Schlüsselausgabe ist ein Anforderungsschein auszufüllen, der von dem Empfänger bei der Schlüsselübergabe vorzuzeigen ist und ihn befähigt, einen Schlüssel zu erhalten. Anzahl der angeforderten Schlüssel, Schlüsseltyp, Gebäude und Raum sowie Name des Empfängers werden hier angegeben. Die Unterschrift des jeweiligen Abteilungsleiters garantiert für die Richtigkeit der Angaben. In Zukunft könnte dies ebenfalls über das SVS geschehen und die papierernen Anforderungsscheine der Vergangenheit angehören lassen. Weiterhin könnten an die DESY-Mitarbeiter über das GIS/FMS, an welches das DESY-SVS angebunden ist, Berechtigungen für den Erhalt von Schlüsseln vergeben werden. Ist ein Mitarbeiter z.B. nicht qualifiziert, einen Schlüssel für eine Experimentierhalle zu erhalten, wird in diesem Fall eine entsprechende Ausgabe von dem System verweigert.

Mit der Dokumentation eines Vorgangs liefert die erste Version des SVS bereits eine Basis, um später eine Historienfunktionalität zu integrieren. Diese soll Übersichten zu den Aktivitäten einzelner Mitarbeiter oder Abteilungen, sowie Historien zu Schlüssel-exemplaren ermöglichen. Auch Abfragen zu Ausgaben oder Verlusten innerhalb eines Zeitraumes sollen mit Hilfe der Vorgangsspeicherung angefertigt werden können. Ein weiteres Feature, das vor allem für die V1-Abteilungsleitung, die nur in geringem Maße in die Schlüsselverwaltung involviert ist, interessant und nützlich wäre,

ist die grafische Abbildung der Schließhierarchie. Tiefe und Breite derselbigen können mit Hilfe einer optischen Darstellung gerade in der Führungsebene der Abteilung V1 besser erkannt werden.

Weiterhin könnte in Zukunft eine Erweiterung des Datenmodells notwendig sein, da für den in Planung befindlichen XFEL auch eine Dokumentation der Türen vorgesehen ist. Die Schnittstelle würde hier demnach nicht mehr zwischen Zylinder und Raum liegen, sondern zwischen Zylinder und Tür, wobei die Tür zwei Räumen zuzuordnen ist. Diese Erweiterung wurde bereits in dem ersten Meeting mit den Entwicklern in Aussicht gestellt. Eine Abänderung des Datenmodells in Zukunft wurde demnach bereits eingeplant, in die Spezifikation aufgenommen und die Systemstruktur entsprechend aufgebaut, so dass Anpassungen keine Schwierigkeiten darstellen werden.

Im Falle der Zylinderausgabe kommt momentan jeder registrierte DESY-Mitarbeiter als Empfänger in Frage. Die Übergabe soll jedoch ausschließlich an DESYaner mit dem Attribut *Hausmeister* erfolgen. Eine entsprechende Erweiterung soll ebenfalls im Zuge einer Systemoptimierung erfolgen. Da das GIS/FMS bereits über die Möglichkeit verfügt, Rollen wie z.B. den Hausmeisterposten an Personen zu vergeben, wurde eine Basis für diese Systemerweiterung bereits geschaffen.

Diese und andere Systemerweiterungen werden diskutiert und in Zukunft entsprechend ihrer Priorität nach und nach hinzugefügt. Als direkter Schritt nach Abschluss des Projektes folgen jedoch zunächst eine Vorstellung des SVS bei den Auftraggebern der Abteilung V1 sowie eine Anwenderschulung für den hauptverantwortlichen Schlüsselverwalter und seine Vertreter. Für die finale Systemübergabe und den Beginn der Inbetriebnahme sind die aktuellen Daten aus *Big Key* zu exportieren und in das DESY-SVS zu importieren.

Intensive Gespräche mit den Auftraggebern ergaben jedoch, dass auch auf Verwaltungsebene Änderungen erfolgen müssen. So wird einem Mitarbeiter, der DESY verlässt, in der Regel ein Laufzettel ausgehändigt, der vollständig abgearbeitet werden muss, bevor dieser das Gelände verlassen darf. Unter anderem muss er alle ihm ausgehändigten DESY-Schlüssel zurückgeben. Allerdings erhalten nur diejenigen Mitarbeiter einen Laufzettel, die auch in der Verwaltung gemeldet sind. So kommt es häufig vor, dass Gastforscher versehentlich die Abgabe ihrer Schlüssel vergessen und diese nie wieder in den DESY-Bestand zurückfließen. Mit der Zeit geht somit eine nicht unerhebliche Anzahl an Schlüsseln verloren, was weiterhin ein Sicherheitsrisiko mit sich bringt und zu einem theoretisch stark überhöhten Schlüsselbestand führt.

10 Fazit und Projektergebnisse

Ziel des Projektes, innerhalb dessen Rahmen die vorliegende Diplomarbeit entstand, war die Entwicklung einer Komponente zur Unterstützung der Schlüsselverwaltungsprozesse und ihre Integration in ein bestehendes Facility-Management-System am

Deutschen Elektronen-Synchrotron. Bisher erfolgte die Verwaltung des Schlüsselbestandes mit Hilfe eines separaten Informationssystems, wobei der Zylinderbestand nicht in das IS integriert war, sondern analog gepflegt wurde. Dieses Vorgehen war nicht mehr zukunftssträftig und im Zuge wachsender Gebäude- und Mitarbeiterzahlen wäre ein qualitativ hochwertiges Schlüsselmanagement nur unter erschwerten Bedingungen durchzuführen.

Zu Beginn des Projektes wurden die Anforderungen der Auftraggeber an das neue System zur Verwaltung des Schlüssel- und Zylinderbestandes erhoben und in einer Spezifikation komprimiert. Diese wurde im weiteren Verlauf des Projektes stets angepasst und entsprechend neuer Erkenntnisse und daraus resultierender Anforderungen abgeändert. Die vollständige und endgültige Fassung der Spezifikation diente einem externen Entwicklerteam als Basis für die technische Umsetzung der Schlüsselverwaltungsfunktionalitäten. Ein Systementwurf verdeutlichte zusätzlich die gewünschte Struktur des SVS. Entworfen wurde das Systemdesign auf Basis eines Analysemodells, welches sich mit dem Aufbau der Schlüsselverwaltungsprozesse auseinandersetzte und diese definierte.

Ein Team aus Entwicklern der Firma Horstick wurde mit der technischen Umsetzung der DESY-Anforderungen betraut. Um die Entwicklungsdauer gering zu halten und den Diskussionsrahmen auf ein geringes Maß einzuschränken, wurden zunächst nur die Anforderungen an die Schlüsselausgabe- und Schlüsselrückgabefunktionen realisiert. Das Projektteam des Deutschen Elektronen-Synchrotrons führte bei Vorlage der Demoversion einen Systemtest durch und prüfte die Umsetzung der Anforderungen. Hierfür wurden den Projektmitgliedern von seinen Auftraggebern Testdaten zur Verfügung gestellt. Hatte das SVS mit den ersten beiden Funktionalitäten eine gewisse Stabilität erreicht, wurde ein Leistungsvergleich mit einer Kaufsoftware durchgeführt. Das SVS wurde einem Team aus vier Testanwendern vorgestellt und von ihnen hinsichtlich der Erfüllung vorgegebener Aufgaben bewertet. Das Ergebnis dieses Leistungsvergleiches ergab, dass mit dem maßgeschneiderten SVS selbst in einem frühen Entwicklungsstadium die Chance auf Erfüllung der Projektziele höher sind als mit einer standardisierten Kaufsoftware, welche den DESY-Ansprüchen nur annähernd gerecht werden konnte. Mit Hilfe dieser Leistungsbewertung konnte die Entwicklung eines nach DESY-Wünschen ausgerichteten SVS gerechtfertigt werden. Außerdem ermöglichte der Leistungsvergleich das Aufdecken von Schwachstellen bei dem DESY-SVS, welche in den fortlaufenden Projektphasen behoben wurden.

Um einen Überblick der anfallenden Entwicklungskosten und der Dauer der technischen Umsetzung für die Schlüsselverwaltungsfunktionalitäten und für eventuell in Zukunft folgende Systemerweiterungen zu simulieren, wurde eine entsprechende Abschätzung mit Hilfe der Funktionspunktanalyse vorgenommen. Durchgeführt wurde sie ausgehend von den Systementwürfen für die einzelnen Prozesskomponenten. Ein Vergleich der Aufwandsschätzung mit der realen Entwicklungsdauer ergab jedoch, dass die Funktionspunktanalyse tatsächlich nur Schätzwerte hervorbringt, die teils stark von der Wirklichkeit abweichen. Der Grund hierfür war, dass die Rechnung auf vielen unsicheren Faktoren beruhte und nicht auf Erfahrungswerte zurückgegriffen werden konnte. Mit Hilfe der Dokumentation der tatsächlichen Entwicklungsdauer

für die Realisierung der einzelnen Prozesskomponenten können jedoch für künftig folgende Systemerweiterungen erheblich präzisere Schätzungen hinsichtlich des zeitlichen Aufwandes zur Umsetzung und der daraus resultierenden Kosten vorgenommen werden.

Bereits ohne Systemerweiterungen ist das DESY-SVS am Ende des Projektes einsatzbereit. Es wurde mit allen Funktionalitäten, die nötig sind, um eine Verwaltung von Schlüsseln technisch zu unterstützen, ausgestattet. Mit erfolgter Übergabe hat der hauptverantwortliche Schlüsselverwalter nun die Möglichkeit, auf aktuelle und vollständige Gebäude- und Personendaten zuzugreifen und diese als notwendige Basis für die Schlüsselverwaltung zu verwenden. Auch die Dokumentation des Zylinderbestandes und ihres Verbleibes ist nun mit Hilfe des DESY-SVS zu erfüllen. Durch den Einsatz von benutzerführenden Wizards wird die Fehlerquote innerhalb des Informationsbestandes reduziert und die Qualität der Schlüsselverwaltung erhöht. Mit der Zeit werden jedoch Wünsche zu Systemerweiterungen aufkommen. Vor allem mit der Inbetriebnahme des XFEL wird eine Erweiterung des Datenmodells um Türen notwendig sein. Dies wird auch das SVS betreffen, da die Schnittstelle dann nicht mehr zwischen Zylinder und Raum, sondern zwischen Zylinder und Tür liegen wird. Hätte DESY sich mit dem Erwerb einer Kaufsoftware begnügt, wäre eine Erweiterung des Systems kaum möglich gewesen. Da es sich bei dem SVS jedoch um eine maßgeschneiderte Anfertigung handelt, sind zukünftige Anpassungen möglich und im Zuge betrieblicher Veränderungen kaum zu vermeiden.

Quellen

4managers

4managers: Benchmarking – Die Lücke zum Besten schließen

URL: <http://www.4managers.de/themen/benchmarking/>

[letzter Zugriff am 17.08.2007]

AIDeX¹

AIDeX Software: Download – Kostenlos testen, jetzt!

URL: <http://www.aidex.de/software/key-organizer/download.html>

[letzter Zugriff am 19.08.2007]

AIDeX²

AIDeX Software: Preise & Bestellung – Preisliste 2007

URL: <http://www.aidex.de/software/key-organizer/preise.html>

[letzter Zugriff am 19.08.2007]

AIDeX³

AIDeX Software: Referenzen

URL: <http://www.aidex.de/software/key-organizer/referenzen.html>

[letzter Zugriff am 19.08.2007]

Booch/Jacobson/Rumbaugh 2001

Grady Booch, Ivar Jacobson, James Rumbaugh: The unified software development process, 4. Auflage, Addison-Wesley 2001

Bundschuh/Peetz/Siska 1991

Manfred Bundschuh, Willi Peetz, Reinhard Siska: Aufwandschätzung von DV-Projekten mit der Function-Point-Methode – Grundlagen, Fallstudien, Lehrmaterial, Verlag TÜV Rheinland, Köln 1991

DESY 2005¹

Deutsches Elektronen-Synchrotron: Zahlen und Fakten, Stand: 15.06.2005

URL: http://zms.desy.de/ueber_desy/zahlenfakten/index_ger.html

[letzter Zugriff am 17.07.2007]

DESY 2005²

Deutsches Elektronen-Synchrotron: Pressebereich, Bilderarchiv: Teilchenphysik

URL: http://zms.desy.de/e548/e550/e6025/index_ger.html

[letzter Zugriff am 01.02.2008]

DESY 2006

Deutsches Elektronen-Synchrotron: DESY besichtigen

URL: http://zms.desy.de/ueber_desy/besichtigungen/index_ger.html

[letzter Zugriff am 01.02.2008]

DESY 2007

Deutsches Elektronen-Synchrotron: DESY Information „Kurz gefasst – Forschung bei DESY“

Donnert/Sterzenbach 1999

Rudolf Donnert, Marion Sterzenbach: Präsentieren – gewusst wie – Praktischer Leitfaden für Vortrag, Moderation und Seminar unter Einsatz neuer Medien, 2. überarbeitete Auflage, Lexika Verlag, Würzburg 1999

Eucker 2005

Silke Eucker: Entwicklung und Einführung eines Verfahrens zur Erstellung und Pflege von Flucht- und Rettungsplänen auf Grundlage des computergestützten Facility Managements (CAFM) am Deutschen Elektronen-Synchrotron (DESY), Diplomarbeit 2005, HAW Hamburg

Glinz 2004

Martin Glinz: Software Engineering I, Vorlesungsskript WS 2004/2005, Universität Zürich

URL:

http://www.ifi.uzh.ch/serg/teaching/discontinued_courses/software_engineering_i/#c277

[letzter Zugriff am 21.08.2007]

Grässle/Baumann/Baumann 2007

Patrick Grässle, Henriette Baumann, Philippe Baumann: UML 2 projektorientiert, 4.aktualisierte Auflage, Bonn 2007

Hitz/Kappel 1999

Martin Hitz, Gerti Kappel: UML @ Work: Von der Analyse zur Realisierung, dpunkt.verlag, Heidelberg 1999

Hürten 1999

Robert Hürten: Function-Point Analysis – Theorie und Praxis, Expert Verlag, Renningen-Malmsheim 1999

ISO/IEC 2005

ISO/IEC: Unified Modelling Language Specification, Version 1.4.2, 2005

URL: <http://www.omg.org/docs/formal/05-04-01.pdf>

[letzter Zugriff am 13.08.2007]

Karlöf/Östblom 1994

Bengt Karlöf, Svante Östblom: Das Benchmarking Konzept – Wegweiser zur Spitzenleistung in Qualität und Produktivität, Verlag Vahlen, München 1994

May 2004

Michael May: IT im Facility Management erfolgreich einsetzen – Das CAFM-Handbuch, Springer-Verlag, Berlin Heidelberg 2004

Nävy 2003

Jens Nävy: Facility Management – Grundlagen, Computerunterstützung, Systemeinführung, Anwendungsbeispiele, 3. Auflage, Springer-Verlag, Berlin Heidelberg 2003

Oestereich 2001

Bernd Oestereich: Objektorientierte Softwareentwicklung – Analyse und Design mit der Unified Modeling Language, 5. völlig überarbeitete Auflage, Oldenbourg Wissenschaftsverlag, München Wien 2001

OMG¹ 2007

Object Management Group: UML Resource Page, zuletzt aktualisiert am 02.01.2007

URL: <http://www.uml.org/>

[letzter Zugriff am 13.08.2007]

OMG² 2007

Object Management Group: Introduction to OMG's Unified Modelling Language, zuletzt aktualisiert am 03.04.2007

URL: http://www.omg.org/gettingstarted/what_is_uml.htm

[letzter Zugriff am 13.08.2007]

Oose 2007

Oose Innovative Informatik GmbH: UML Historie, 2007

URL: http://www.oose.de/uml_historie.htm

[letzter Zugriff am 14.08.2007]

Robben 2004

Andrea Robben: Einführung eines Geoinformations- und Facility Management Systems (GIS/FMS) am DESY, Präsentation auf dem Autodesk AnwenderTreff Düsseldorf am 30.06. 2004

Rupp 2002

Chris Rupp: Requirements-Engineering und -Management – Professionelle, iterative Anforderungsanalyse für die Praxis, 2. überarbeitete Auflage, Carl Hanser Verlag, München, Wien 2002

Sabisch/Tintelnot 1997

Helmut Sabisch, Claus Tintelnot: Benchmarking – Weg zu unternehmerischen Spitzenleistungen, Schäffer-Poeschel Verlag, Stuttgart 1997

Sommerville 2005

Ian Sommerville: Integrated Requirements Engineering: A Tutorial In: IEEE Software, Januar / Februar 2005

Eidesstattliche Versicherung

Eidesstattliche Versicherung

Ich versichere, die vorliegende Arbeit selbstständig ohne fremde Hilfe verfasst und keine anderen Quellen und Hilfsmittel als die angegebenen benutzt zu haben. Die aus anderen Werken wörtlich entnommenen Stellen oder dem Sinn nach entlehnten Passagen sind durch Quellenangaben kenntlich gemacht.

Ort, Datum

Unterschrift

Anhang: Spezifikation für die Schlüsselausgabekomponente

1 Einleitung

Die folgende Spezifikation ist ein Auszug des Anforderungsdokumentes, welches die Entwickler der Firma Horstick als Basis für die technische Umsetzung der Schlüsselverwaltungsprozesse erhielten. In der Originalfassung enthält die Spezifikation zunächst eine Erläuterung des Schlüsselverwaltungskonzeptes so wie es innerhalb des Hauses DESY gilt. Dieses Kapitel soll gewährleisten, dass alle Projektmitglieder auf der selben Wissensbasis arbeiten. Weiterhin umfasst die Original-Spezifikation einen Überblick der Schlüsselverwaltungsprozesse sowie Definitionen zu den einzelnen. Im Folgenden werden jedoch lediglich die Kapitel zu den Prozess- und Systemanforderungen wiedergegeben. Zum einen wird auf die Beifügung der vollständigen Spezifikation verzichtet, da einige Elemente Bestandteil der Diplomarbeit sind und an dieser Stelle nicht wiederholend hinzugefügt werden sollen. Zum anderen erscheint eine Ausführung der Definitionen und Anforderungen aller Schlüsselverwaltungsprozesse nicht angebracht, da lediglich einer – der Schlüsselausgabeprozess – innerhalb der vorliegenden Diplomarbeit behandelt wurde. Die folgenden Seiten befassen sich demnach ausschließlich mit dem Prozess der Schlüsselausgabe.

2 Prozessanforderungen

Die Schlüsselverwaltung gliedert sich in insgesamt elf Teilprozesse. Im Folgenden werden die Rollen der Beteiligten dieser Prozesse erläutert sowie die funktionalen Anforderungen und Randbedingungen an den ausgewählten Prozess definiert. Anforderungen an die Beteiligten zur Umsetzung des Prozesses bilden den Abschluss dieses Kapitelabschnitts.

2.1 Prozessbeteiligte

Der Prozess *Schlüssel ausgeben* hat als beteiligte Akteure *Gebäudenutzer*, *Verwalter* und *Informationssystem*. Der Gebäudenutzer ist der auslösende Akteur des Geschäftsanwendungsfalls. Der Verwalter reagiert auf die Einleitung der Transaktion, führt diese in Unterstützung mit dem Informationssystem durch und pflegt in demselben die Schlüsselexemplarsinformationen, so dass Bestandsübersichten stets ein korrektes Abbild der Realität darstellen.

2.2 Funktionale Prozessanforderungen

Innerhalb dieses Abschnitts werden die funktionalen Anforderungen an die Schlüsselverwaltungsprozesse definiert. Sie werden hinsichtlich allgemeiner Anforderungen und spezieller Anforderungen unterschieden. Die allgemeinen Anforderungen zielen

auf alle Schlüsselverwaltungsprozesse ab, während die speziellen Anforderungen zum Schlüsselausgabeprozess in einem extra Abschnitt behandelt werden.

2.2.1 Allgemeine Prozessanforderungen

Die Prozesse sollen den Erfolgsfall einer Transaktion darstellen.

Die Prozesse sollen die Arbeitsschritte festlegen, die zur Aufgabenerfüllung notwendig sind.

Die Prozesse sollen die Reihenfolge der Arbeitsschritte festlegen.

Die Prozesse sollen Zuständigkeiten unter den Beteiligten festlegen.

Die Prozesse sollen ausschließlich von autorisierten Schlüsselverwaltern durchgeführt werden.

Die Prozesse sollen sicherstellen, dass die Daten der durchgeführten Transaktionen dokumentiert werden, um diese nachhaltig nachvollziehen zu können.

2.2.2 Anforderungen an den Schlüsselausgabeprozess

Eine Verfügbarkeitsprüfung soll sicherstellen, dass ausschließlich im Lager befindliche Schlüssel zur Ausgabe angeboten werden.

Eine Verfügbarkeitsprüfung soll es ermöglichen, nicht vorrätige Schlüssel zu identifizieren.

Eine Verfügbarkeitsprüfung soll es ermöglichen, die Notwendigkeit zur Nachbestellung festzustellen.

Der Prozess soll sicherstellen, dass ausgegebene Schlüsselexemplare entsprechend dokumentiert werden.

Die Dokumentation der Ausgabe soll die Möglichkeit zur Kontrolle des Exemplarsverbleibs sicherstellen.

Eine Bestätigung über die korrekte Dokumentation der durchgeführten Schlüsselausgabe durch den Gebäudenutzer soll Dateninkonsistenz vermeiden.

2.3 Randbedingungen

Die Durchführung der Schlüsselverwaltungsprozesse und deren Dokumentation sollen in Unterstützung durch G-Info in der Version 2007, welches eine Plattform innerhalb des GIS/FMS ist, erfolgen.

G-Info 2007 soll mit Funktionen zur Unterstützung der Schlüsselverwaltungsprozesse erweitert sein.

Die Schlüsselexemplare des DESY-Bestandes sollen mitsamt ihren Attributen in G-Info angelegt sein.

Die Zylinderexemplare des DESY-Bestandes sollen mitsamt ihren Attributen in G-Info angelegt sein.

Die Klassen der Schlüsselverwaltung sollen mit den an diesen Kontext angrenzenden Themen *Raum/Gebäude* und *Personen* verbunden sein.

Autorisierte Schlüsselverwalter sollen einen Zugang zu G-Info haben.

2.4 Anforderungen an die Beteiligten zur Prozessumsetzung

Der Verwalter soll Schlüssel auf Verfügbarkeit prüfen.

Der Verwalter soll verfügbare Schlüssel zur Ausgabe anbieten.

Der Verwalter soll nicht verfügbare Schlüssel beschaffen.

Der Verwalter soll das auszugebende Schlüsselexemplar direkt an den Gebäudenutzer übergeben. Eine Übergabe an Dritte soll vermieden werden.

Der Verwalter soll die Ausgabe dokumentieren.

Der Verwalter soll den Schlüsselexemplaren bei der Dokumentation der Ausgabe den jeweiligen Empfänger zuordnen.

Der Gebäudenutzer soll die Korrektheit der Angaben bestätigen

3 Systemanforderungen

Die von DESY identifizierten Schlüsselverwaltungsprozesse sollen in Unterstützung mit G-Info 2007 durchgeführt und dokumentiert werden. G-Info ist in das GIS/FMS integriert und wird bei DESY u.a. zur Dokumentation und Pflege von Gebäude- und Raumdaten eingesetzt. Um mit G-Info in Zukunft auch die Prozesse der Schlüssel-

verwaltung unterstützen zu können, sind Systemerweiterungen notwendig, die im Folgenden spezifiziert werden. Zunächst soll jedoch darauf eingegangen werden, welchen Nutzen das Schlüsselverwaltungssystem (SVS) aus Anwendersicht erfüllen muss. Es folgt eine Definition der Anforderungen an die zu entwickelnden Systemerweiterungen. Diese gliedern sich in funktionale und nicht-funktionale.

3.1 Anwenderanforderungen an das Schlüsselverwaltungssystem

Wizards sollen den SVS-Nutzer durch eine vordefinierte Prozesskette führen und verhindern, dass dieser von ihr abweicht.

Das SVS soll dem Nutzer die Möglichkeit bieten, Bestandsübersichten anzufertigen.

Das SVS soll dem Nutzer die Möglichkeit bieten, einem Zugangswunsch entsprechende Schlüssel auf ihre Verfügbarkeit zu prüfen.

Das SVS soll dem Nutzer die Möglichkeit bieten, die Anzahl verfügbarer Schlüsseltypen zu ermitteln, um diese ggf. zu beschaffen.

Das SVS soll dem Nutzer die Möglichkeit bieten, verfügbare Schlüsselexemplare zu identifizieren.

Das SVS soll dem Nutzer die Möglichkeit bieten, Schlüsselexemplare Besitzern zuzuordnen.

Das SVS soll dem Nutzer die Möglichkeit bieten, Prozesse mit mehr als einem beteiligten Schlüsselexemplar durchzuführen.

Das SVS soll dem Nutzer die Möglichkeit bieten, durchgeführte Transaktionen zu dokumentieren.

Wizards sollen den SVS-Nutzer dazu veranlassen, alle für die Dokumentation eines Prozesses notwendigen Informationen anzugeben.

Ein Schlüsselausgabe-Wizard soll den SVS-Nutzer veranlassen, eine Schlüsselausgabe ausschließlich mit Besitzerzuordnung zu dokumentieren.

Wizards sollen dem SVS-Nutzer die Möglichkeit bieten, Transaktionen zu dokumentieren, ohne den finalen Status des Schlüsselexemplars manuell ändern zu müssen.

Das SVS soll dem Nutzer die Möglichkeit bieten, Transaktionen nachhaltig zu speichern.

3.2 Funktionale Systemanforderungen

In den folgenden Kapitelabschnitten werden zunächst die allgemeinen Systemanforderungen definiert. Es folgen Anforderungen an einen Schlüsselausgabe-Wizard, welcher den Schlüsselverwalter durch den dazugehörigen Prozess führen soll.

3.2.1 Allgemeine Systemanforderungen

Das SVS soll in der Lage sein, Daten zu importieren.

Das SVS soll in der Lage sein, Daten zu exportieren.

Die Durchführung aller identifizierten Schlüsselverwaltungsprozesse soll durch entsprechende Wizards unterstützt werden.

Entscheidet sich der Nutzer für die Durchführung eines Prozesses, soll das SVS den dafür vorgesehenen Wizard öffnen.

Das SVS soll dem Nutzer die Möglichkeit bieten, Suchen durchzuführen.

Das SVS soll dem Nutzer die Möglichkeit bieten, Suchen mit Filterfunktion oder Trunkierung durchzuführen.

Die Wizards sollen in der Lage sein, Schlüsselseriennummern auf Formatzulässigkeit zu prüfen.

Schlüsselseriennummern werden bei DESY nach folgenden Mustern vergeben:

V	-	Z	/1	(Gebäudeingangsschlüssel)
V	-	106	/1	(Einzelschlüssel; nach `` max. 4 Zahlen)
GS	-	V1	/1	(Gruppenschlüssel)
OGS	-	V	/1	(Obergruppenschlüssel)
GHS			/1	(Generalhauptschlüssel)

Die Wizards sollen den SVS-Nutzer daran hindern, Aktionen durchzuführen, ohne zuvor erforderliche Angaben gemacht zu haben.

3.2.2 Anforderungen an den Schlüsselausgabe-Wizard

Der Schlüsselausgabe-Wizard soll den SVS-Nutzer nach folgenden Werten fragen, um Schlüssel für einen Zugangswunsch zu ermitteln:

- Gebäudenummer
- Etage
- Raumnummer
- Schlüsseltyp

Die Angabe des Schlüsseltypen soll optional sein.

Der Schlüsselausgabe-Wizard soll dem SVS-Nutzer die Option bieten, Schlüssel für weitere Räume zu ermitteln.

Der Schlüsselausgabe-Wizard soll den SVS-Nutzer zur Bestätigung seiner Eingaben auffordern.

Hat der Nutzer seine Eingabe bestätigt, soll das SVS eine Suche entsprechend der eingetragenen Werte durchführen.

Das SVS soll die Suche nur auf Schlüsselexemplare anwenden, die den Status *verfügbar* haben.

Der Schlüsselausgabe-Wizard soll eine Trefferliste mit zutreffenden Schlüsselexemplaren anzeigen.

Der Schlüsselausgabe-Wizard soll folgende Schlüsselexemplarsattribute anzeigen:

- Schlüsselseriennummer
- Schlüsseltyp
- Ausgabestelle
- Status

Der Schlüsselausgabe-Wizard soll neben einer Trefferliste mit Schlüsselexemplaren eine Liste mit der Anzahl verfügbarer Schlüsseltypen anzeigen.

Der Schlüsselausgabe-Wizard soll den SVS-Nutzer zu einer Auswahl auffordern.

Das SVS soll dem SVS-Nutzer ausschließlich Auswahl und Bearbeitung von Schlüsselexemplaren, die in seinem Verantwortungsbereich liegen, ermöglichen.

Hat der SVS-Nutzer das auszugebende Schlüsselexemplar ausgewählt, soll der Schlüsselausgabe-Wizard den SVS-Nutzer zur Angabe eines neuen Besitzers auffordern.

Das SVS soll prüfen, ob der Besitzer bereits ein Schlüsselexemplar besitzt, das auf den Zugangswunsch passt.

Der Schlüsselausgabe-Wizard soll den SVS-Nutzer zur Bestätigung der Ausgabe auffordern.

Das SVS soll die Transaktion ausschließlich mit Besitzerzuordnung durchführen.

Das SVS soll die Transaktion dokumentieren.

Das SVS soll dem Exemplar den zuvor durch den SVS-Nutzer angegebenen Besitzer zuordnen

Das SVS soll den Exemplarsstatus in *ausgegeben* ändern.

Das SVS soll alle Daten des durchgeführten Vorgangs mit dem aktuellen Datum in einer Historie speichern.

Der Schlüsselausgabe-Wizard soll den SVS-Nutzer über eine erfolgreiche Transaktionsdurchführung informieren.

Nach der Speicherung der Prozessdaten, soll der Schlüsselausgabe-Wizard dem SVS-Nutzer eine Übersicht des durchgeführten Vorgangs anzeigen.

Die Übersicht des durchgeführten Vorgangs soll folgende Informationen enthalten:

- Transaktion
- Datum der Transaktion
- Schlüsselseriennummer
- Schlüsseltyp
- Ausgabestelle
- Vorname des Besitzers
- Nachname des Besitzers
- Gruppe des Besitzers

Der Schlüsselausgabe-Wizard soll dem SVS-Nutzer die Option zum Belegdruck anbieten.

3.3 Nicht-funktionale Systemanforderungen

Die Schlüsselverwaltungsprozesse sollen durch das GIS/FMS unterstützt werden.

Das DESY-SVS soll ein Datenmodell gemäß der dargestellten Beziehungen in Abb. 9 (Kapitel 3.6) aufweisen.

Die Themengebiete *Gebäude/Raum* und *Personen* sind ein Bestandteil der des GIS/FMS und grenzen an das Konzept der Schlüsselverwaltung an. Diese Themengebiete sollen mit den neu hinzukommenden Themengebieten zur Schlüsselverwaltung verknüpft sein.

Das DESY-SVS soll so konzipiert sein, dass Änderungen und Erweiterungen möglich sind. (Verwaltung von Schrankschlüsseln, Bindung der Schlüssel an Objekte statt an Räume: Türen, Zapfsäulen, Schränke usw.)

Die Schlüsselverwaltungsprozesse sollen durch benutzerführende Wizards unterstützt werden.

Das SVS soll den Zugriff zu Schlüsselverwaltungsfunktionen ausschließlich über Wizards zulassen.

Eingaben über die Standardoberfläche des SVS sollen unterbunden werden.

Wizards sollen zur Sicherung der Datenkonsistenz dienen.

Wizards sollen den Nutzer zur Eingabe notwendiger Daten zwingen.

Wizards sollen die Durchführung widersprüchlicher Aktionen vermeiden. (Ein ausgegebener Schlüssel muss einer Person zugeordnet werden. Ein zurückgegebener Schlüssel darf keinen Besitzer haben etc.)

Das SVS soll über eine Benutzerautorisierung verfügen.

Die Wizards sollen über Auswahllisten verfügen.

Auswahllisten sollen die Verwendung unterschiedlicher Begriffe (ausgeliehen, ausgegeben) sowie Rechtschreibfehler vermeiden und Vereinheitlichung sicherstellen.

Folgende Auswahllisten sind vorgesehen:

- Mitarbeiter (bereits im SVS angelegt)
- Gebäude (bereits im SVS angelegt)
- Räume (bereits im SVS angelegt)
- Schlüsseltyp:
 - Z (Gebäudeeingangsschlüssel)
 - ES (Einzelschlüssel)
 - GS (Gruppenschlüssel)
 - OGS (Obergruppenschlüssel)
 - GHS (Generalhauptschlüssel)