

Vom Fachbereich Maschinenwesen der Universität Duisburg-Essen

zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs genehmigte Dissertation

Konzeption eines Software-Support-Management-Systems (SSMS) zur

Verbesserung von kundenorientierter Softwarepflege und

Weiterentwicklung sowie zur Wissensdistribution.

Vorgelegt von

Jan Herud

Mainz

Referent: Priv.Doz. Dr.-Ing. habil. F. Lobeck

Korreferent: Univ.-Prof. Dr.-Ing. R. Tracht

Vorsitzender: Univ.-Prof. Dr.-Ing. E. Sauer

Tag der mündlichen Prüfung: 21.03.2007

1. EINLEITUNG	5
1.1. SOFTWARELEBENSZYKLUS UND SUPPORT IM UMFELD VON IT-ANWENDUNGEN	5
1.2. AUFGABENSTELLUNG UND REALISIERUNG.....	6
2. SOFTWARE-SUPPORT-MANAGEMENT HEUTE	8
2.1. ALLGEMEINES	8
2.2. AUFGABEN DES SUPPORTS IM UMFELD VON CAD-ANWENDUNGEN	10
2.2.1. <i>Aufgaben im Umgang mit Softwareproblemen</i>	10
2.2.2. <i>Aufgaben bei Implementierung und Wissensdistribution</i>	13
2.2.3. <i>Aufgaben im Rahmen kundenspezifischer Softwareanpassung</i>	13
2.3. ANALYSE EINIGER PRAXISBEISPIELE	14
2.3.1. <i>Beispiel 3D-CAD-Standalone-Produkt SolidWorks</i>	16
2.3.2. <i>Beispiel CAD-Zusatzapplikation Varbox</i>	22
2.3.3. <i>Beispiel kundenspezifische Softwareanpassung</i>	27
2.3.4. <i>Beispiel kundenspezifische Funktionserweiterung der Software</i>	31
2.4. BEURTEILUNG DER DEFIZITE DER ANALYSIERTEN PRAXISBEISPIELE.....	34
2.4.1. <i>Bearbeitete Prozesse</i>	34
2.4.2. <i>Qualitätssicherung</i>	36
2.4.3. <i>Eingesetzte Tools</i>	38
2.4.4. <i>Wissensdistribution</i>	41
2.4.5. <i>Informationsmanagement</i>	42
2.5. ZUSAMMENFASSUNG DER DEFIZITE	43
3. FORDERUNGEN AN EIN SOFTWARE-SUPPORT-MANAGEMENT-SYSTEM.....	48
3.1. UMFANG DES SSMS.....	48
3.2. ANFORDERUNGEN AN DAS DATENMANAGEMENT	49
3.2.1. <i>Organisation und Verwaltung von Kundeninformationen</i>	49
3.2.2. <i>Aufbau und Inhalt von Dokumentation</i>	50
3.2.3. <i>Verwaltung von Experten-Wissen</i>	51
3.2.4. <i>Übernahme und Verwaltung von Bestandsdaten</i>	53
3.3. ANFORDERUNGEN AN DAS PROZESSMANAGEMENT.....	53
3.3.1. <i>Bearbeitung von Anfragen</i>	54
3.3.2. <i>Verifikation von Bugs</i>	56
3.3.3. <i>Vorqualifizierung von VVs</i>	56
3.3.4. <i>Externe Kommunikation</i>	58
3.3.5. <i>Interne Kommunikation</i>	59
3.3.6. <i>Dokumentation und Erfassung von Expertenwissen</i>	59
3.3.7. <i>Qualitätssicherung</i>	60
3.4. ANFORDERUNGEN AN DIE BENUTZEROBERFLÄCHE	61
3.5. ANFORDERUNGEN AN DIE ANBINDUNG UND INTEGRATION DES SSMS	62
3.6. ANBINDUNG AN VORHANDENE SYSTEME ZUR UNTERSTÜTZUNG DES SOFTWARESUPPORTS	64
3.7. ZUSAMMENFASSUNG DER ANFORDERUNGEN	65
4. AKTUELLE TECHNOLOGIEN ZUR VERBESSERUNG DES SOFTWARE-SUPPORTS	70
4.1. ABGRENZUNG ZU PLM-SYSTEMEN	70
4.2. ABGRENZUNG ZU CRM-SYSTEMEN	73
4.3. ABGRENZUNG ZU DATENBANKSYSTEMEN.....	74
4.4. AUSWAHL DER TECHNOLOGIEN UND WERKZEUGE FÜR DAS NEUE SSMS-KONZEPT	75
5. KONZEPT FÜR EIN RECHNERGESTÜTZTES SSMS	78
5.1. AUFBAU DES SSMS	78
5.2. CLIENT-SERVER-ARCHITEKTUR.....	80
5.3. PROZESSDEFINITION.....	82
5.3.1. <i>Erfassung von Kundeninformationen</i>	84
5.3.2. <i>Erfassung und Pflege von Wissensdatenbank-Artikeln</i>	87

5.3.3.	<i>CALL-Handling</i>	88
5.4.	DATENHALTUNG UND FUNKTIONSDEFINITION.....	91
5.4.1.	<i>Kontextbetrachtung und Funktionsbestimmung</i>	91
5.4.2.	<i>Datenhaltung von Kundeninformationen</i>	93
5.4.3.	<i>Datenhaltung von CALL-Informationen</i>	99
5.4.4.	<i>Datenhaltung von Expertenwissen</i>	108
5.4.5.	<i>Datenhaltung von Applikationsdateien</i>	115
5.5.	SOFTWAREARCHITEKTUR DES SSMS.....	125
5.5.1.	<i>Interner Aufbau des DB-Moduls</i>	126
5.5.2.	<i>Klassen- und Objektstruktur des SSMS</i>	128
5.5.3.	<i>Auszug aus dem Klassendiagramm</i>	128
5.5.4.	<i>Auszug aus dem Objektdiagramm</i>	131
5.6.	DIE BENUTZERSCHNITTSTELLE DES SSMS-CLIENT.....	133
6.	ZUSAMMENFASSUNG	137
6.1.	AUSBlick.....	138
7.	LITERATURVERZEICHNIS	140
8.	ABBILDUNGSVERZEICHNIS	142
9.	ANHANG	143
9.1.	FRAGENKOMPLEX DER ANALYSIERTEN PRAXISBEISPIELE.....	143
9.2.	PROZESSDIAGRAMME.....	145
9.3.	KATEGORIENLISTE.....	152
10.	ABKÜRZUNGSVERZEICHNIS	155

Vorwort und Danksagung

Die Idee für diese Arbeit entstand während meiner Tätigkeit als Supportingenieur für das 3D-CAD-System SolidWorks bei einem Systemhaus für CAE / PLM-Lösungen. Der alltägliche Umgang mit Fragestellungen zur Implementierung und dem Betrieb derartig komplexer Applikationen bei Kunden unterschiedlichster Branchen sowie die zentrale Rolle des Supports in diesem Kontext führten zu dem Wunsch, diese Prozesse näher zu beleuchten. Dabei wurden bald vielfältige Verbesserungspotentiale sowohl hinsichtlich der Qualität, als auch der Wirtschaftlichkeit der Supportprozesse erkennbar. Entstanden ist schließlich das Konzept eines umfassenden IT-Systems - eines Software-Support-Management-Systems. Dieses soll künftig mittelständischen Systemhäusern eine einheitliche, solide und funktionale Basis für Ihre Supportprozesse bieten.

Herrn Privatdozent Dr.-Ing. F. Lobeck danke ich für sein stets offenes Ohr, seine Hilfsbereitschaft und seine tatkräftige Unterstützung beim Schreiben der Arbeit sowie für die Übernahme des Erstgutachtens.

Herrn Univ.-Prof. Dr.-Ing. R. Tracht danke ich für das entgegengebrachte Interesse und die Übernahme des Korreferats.

Herrn Univ.-Prof. Dr.-Ing. H.J. Stracke danke ich für die Anregungen und Beratung bei der Ausarbeitung dieser Dissertation.

Ich möchte außerdem Herrn P. Miklitza von der DPS Software GmbH für seine vielfältige Unterstützung bei der Ausfertigung der vorliegenden Arbeit meinen Dank aussprechen.

Ein besonderer Dank geht an meine liebe Barbara, für Ihre fachliche und moralische Unterstützung, aber auch für Ihre Geduld und Nachsicht während der Erstellung dieser Arbeit.

Ich danke schließlich meinen Eltern Klaus und Heidi Herud dafür, das Sie mir die Ausbildung ermöglicht und mir die Werte vermittelt haben, die mich bis zu der Erstellung dieser Arbeit geführt hat.

Wiesbaden, im Februar 2007

Jan Herud

1. Einleitung

Der Begriff **Software-Lifecycle-Management (SLM)** umfasst die informationstechnologischen Methoden zur Abbildung des Lebenszyklus einer Software und der dabei ablaufenden Prozesse. Ein Softwarelebenszyklus wird dabei wie folgt definiert [15]:

„Als Softwarelebenszyklus wird die gesamte Lebensdauer eines Software-Produktes von seiner Entwicklung über seinen Betrieb bis hin zu seiner „Außer-Betriebnahme“ bezeichnet.“

Obwohl die Begrifflichkeit des Lebenszyklus gemeinhin im Kontext zu materiellen Produkten wie Konsumgütern oder Maschinen genannt wird, so deutet diese Definition bereits darauf hin, dass auch Software - obwohl immateriell - im Sinne eines SLM als Produkt betrachtet wird und damit einem solchen Zyklus genauso unterliegt wie jedes materielle Produkt.

Gemäß der VDI-Richtlinie 2221 [1] wird diesem Umstand auch innerhalb der folgenden allgemeinen Produktdefinition Rechnung getragen:

„Erzeugnis, das als Ergebnis des Entwickelns und Konstruierens hergestellt oder angewendet wird. Das können materielle (z.B. Maschinen, Verfahren) oder auch immaterielle Erzeugnisse (z.B. Programme) sein.“

Ergänzend dazu wird Software nach IEEE / ANSI [14] definiert als:

“Computer programs, procedures, rules, and possibly associated documentation and data pertaining to the operation of a computer system.”

(Computerprogramme, Prozeduren, Regeln und ggf. dazugehöriger Dokumentation und Daten mit Bezug auf den Betrieb eines Computersystems - Anm. d. Verf.).

1.1. Softwarelebenszyklus und Support im Umfeld von IT-Anwendungen

Der Software-Support erfüllt gegenwärtig vielfältige Aufgaben. Im Rahmen dieser Arbeit sollen nicht der Support von Betriebssystemen oder Office-Software betrachtet werden, sondern es soll vielmehr untersucht werden, wie der Support im Rahmen von komplexen Anwendungen wie PDM¹, PLM², PPS³ und speziell bei CAD⁴ / CAM⁵-Applikationen in ein SLM eingebunden wird.

¹ PDM – Product Data Management

² PLM – Product Lifecycle Management

³ PPS - Produktions- Planungs- und -Steuerungssystem

⁴ CAD – Computer Aided Design

⁵ CAM - Computer Aided Manufacturing

1.2. Aufgabenstellung und Realisierung

In der wissenschaftlichen Arbeit von [8] wurde ein SLM-System konzipiert, welches bereits den Support in den Lebenszyklus einbindet. In dem dort formulierten Ansatz übernimmt der Support primär die Aufgabe des Testens neuer bzw. überarbeiteter Programme unter praxisnahen Bedingungen, da davon ausgegangen wird, dass eine Entwicklungsabteilung aufgrund ihrer Voreingenommenheit dazu nicht in der Lage ist. In diesem Punkt ist der von [8] gewählte Ansatz für ein SLM nicht umfassend genug, da der Support nur im unmittelbaren Zusammenhang mit einer Fehlerbehebung im Rahmen der Qualitätssicherung gesehen wird. Insbesondere im Umfeld von CAD-/CAM-Anwendungen gehen die Aufgaben des Supports jedoch wesentlich darüber hinaus. Der Support nimmt zwar auch beim Testen neuer Version bestehender Softwareprodukte oder beim Testen gänzlich neuer Programme eine Schlüsselrolle ein, in erster Linie bildet der Support jedoch die Schnittstelle zwischen Endkunden und der Entwicklung im Rahmen der kontinuierlichen Weiterentwicklung und Verbesserung bestehender Software. Dies gilt insbesondere im Bezug auf deren

- Qualität,
- Ergonomie und
- Funktionalität.

Wesentlich in diesem Zusammenhang ist zudem, dass der Support über die applikationsspezifischen Kernkompetenzen des Unternehmens verfügt. So können Entwickler aufgrund ihrer Position die Anforderungen an ein neues Produkt oftmals nur schwer beurteilen. Durch seine Kundennähe und den täglichen, anwendungsorientierten Umgang mit komplexen Softwareprodukten verfügt der Support über die wesentlich günstigere Ausgangssituation, um Anforderungen an neue Softwareprodukte aus Sicht der Endkunden zu formulieren und darüber hinaus das für den Unternehmenserfolg und den Kundenerfolg notwendige Expertenwissen zu erwerben. Dieses Wissen kann der Support wiederum unmittelbar zur Unterstützung und Schulung der Endkunden hinsichtlich des Handlings und der Implementierung von Software einsetzen. Zudem spielt dieses Wissen eine wesentliche Rolle bei der Eingrenzung und Behebung von Softwarefehlern und muss deshalb auch von diesem Standpunkt aus betrachtet, künftig einen größeren Einfluss auf die Entwicklung und Pflege von Softwareprodukten im Sinne eines SLM nehmen.

Die Mehrung, Erhaltung und Vermittlung von Expertenwissen sind von daher grundlegende Aufgaben des Supports und müssen zu den höchsten Zielen eines Softwareherstellers oder

Systemhauses gehören. Für die Lösung dieser Aufgaben steht dem Support eine ganze Reihe von kommunikations- und informationstechnischen Werkzeugen zur Verfügung. Hierbei handelt es sich jedoch überwiegend um Insellösungen, die keinen ganzheitlichen Ansatz verfolgen und nicht vollständig in ein SLM eingebunden sind. Bei diesen Lösungen sind inkonsistente Datenerfassung auf der einen Seite, mangelhafte Effizienz und ein erheblicher Informationsverlust auf der anderen Seite die zwangsläufige Folge.

Im Rahmen dieser Arbeit soll aus den genannten Gründen ein System konzipiert werden, welches diesen Anforderungen im Kontext des SLM gerecht wird. Insbesondere sollen in diesem System die unterschiedlichen Prozesse bei der Umsetzung von Verbesserungsvorschlägen, der Verifizierung, Klassifizierung, Dokumentation und der Beseitigung von Fehlern sowie bei der Erfassung, Pflege und Bereitstellung von Expertenwissen innerhalb der Supportorganisation und für den Endkunden abgebildet werden. Ausgehend von einer umfassenden Darstellung und Analyse der gegenwärtigen Prozesse im Software-Support-Management werden die Anforderungen abgeleitet, denen ein künftiges SSMS in jeglicher Hinsicht genügen muss. Bevor das Konzept entwickelt werden kann, müssen jedoch zunächst die aktuellen zur Verfügung stehenden Technologien dahingehend analysiert werden, in wie weit diese zur Verbesserung des Software-Supports genutzt bzw. verbessert werden können. Auf der Grundlage ausgewählter Technologien und Werkzeuge wird schließlich das Konzept für ein rechnergestütztes SMSS basierend auf einem objektorientierten Entwicklungsansatz dargelegt und detailliert beschrieben. Für eine beispielhafte Umsetzung ist das hier entwickelte Konzept zu komplex, um dieses im Rahmen einer Dissertation von einer Person umsetzen zu können. Die Zusammenfassung und der anschließende Ausblick sollen zeigen, welche Entwicklungspotentiale für zukünftige Integrationen noch identifizierbar sind.

2. Software-Support-Management heute

Bevor eine gezielte Analyse der gegenwärtigen Praxis im Software-Support-Management in Abhängigkeit von der Interaktion zwischen Softwareentwicklung, Support und Endkunden bzw. Anwendern erfolgen kann, muss zunächst geklärt werden, welche Aspekte in diesem Kontext analysiert werden sollen.

Zur Verbesserung des allgemeinen Verständnisses werden der Klärung dieser Frage einige Erläuterungen zu grundlegenden Prozessen und Voraussetzungen sowie ein Überblick über die zu analysierenden Abläufe und die in diesem Zusammenhang relevanten Rahmenbedingungen vorangestellt.

2.1. Allgemeines

Gegenwärtig werden bei der Integration des Software-Supports in ein SLM-System unterschiedliche Strategien verfolgt. Diese sind eng verknüpft mit den Vertriebsstrukturen der jeweiligen Softwarehersteller. Auf der einen Seite ist der Direktvertrieb ein verbreitetes Konzept zur Vermarktung von Software im CAD-Markt. Hier sind sowohl der Vertrieb als auch der Support im herstellenden Unternehmen lokalisiert. Diese Konstellation ist überwiegend im Bereich des Highrange-CAD-Marktes und bei regional operierenden Softwareherstellern anzutreffen. Auf der anderen Seite wird eine Reihe von Softwareprodukten im CAD-Umfeld über indirekte Vertriebskanäle vermarktet, wobei ein Softwarehändler sowohl die vertrieblichen Aktivitäten zwischen dem Hersteller und dem Endkunden koordiniert und den überwiegenden Teil des Supports leistet als auch als primärer Ansprechpartner für die Endkunden positioniert ist. Diese zweite Konstellation ist im Mainstream-CAD-Markt weit verbreitet. Es bleibt zu erwähnen, dass neben den beiden genannten Strukturen auch gemischte Konstellationen im Vertrieb von CAD-Produkten vorkommen. So können für ein Software-Produkt sowohl direkte als auch indirekte Vertriebswege nebeneinander existieren, wobei jedoch in der überwiegenden Zahl der Fälle Support und Vertrieb im selben Unternehmen angesiedelt sind. Dies ist allein schon deshalb sinnvoll, weil die auf den Verkaufsvorgang folgenden Prozesse während der Implementierung immer wieder vertriebliche Tätigkeiten beinhalten und somit eine enge Zusammenarbeit zwischen Support- und Vertriebsorganisation erfordern.

Aufgrund der starken Verbreitung bezieht sich der Inhalt der vorliegenden Arbeit auf die Untersuchung des Mainstream-CAD-Marktes, die Ergebnisse lassen sich jedoch unter Berücksichtigung der individuellen Gegebenheiten auch auf andere Segmente des CAD- bzw. IT-Marktes übertragen.

Bevor die Aufgaben des Supports näher analysiert werden, muss zunächst der Begriff „Support“ geklärt werden. Der Support (Anglizismus, v. engl. *support* = die Stütze; *to support* = unterstützen) ist eine problemorientierte Beratungstätigkeit. Das Ziel dieser Tätigkeiten ist die Bearbeitung und Lösung von Supportanfragen („Tickets“) interner oder externer Kunden vor Ort, via eMail, Telefon oder mit Hilfe anderer Kommunikationsmittel. Zur Differenzierung von Kompetenzen ist der Support in der Regel hierarchisch aufgeteilt. In den meisten Fällen kommt ein dreistufiges System zum Einsatz:

- Der First-Level-Support (auch **User Help Desk** bzw. UHD) ist die erste Anlaufstelle für alle eingehenden Unterstützungsfragen der Kunden. Ein Mitarbeiter bearbeitet diese nach seinem Kenntnisstand weitestgehend selbständig. Das Ziel ist das schnelle Lösen einer möglichst großen Anzahl von Problemen, was durch Zuhilfenahme von Wissensdatenbanken ermöglicht wird.
- Der Second-Level-Support unterstützt den First-Level-Support durch eine Weiterbildung am Arbeitsplatz (engl. *training-on-the-job*) und durch die Übernahme komplexer Anfragen. Neu erarbeitete Lösungen werden gegebenenfalls in Wissensdatenbanken eingepflegt, um das Wissen für den First-Level-Support nutzbar zu machen. Übersteigt die Komplexität einer Anfrage das Know-How oder die technischen Möglichkeiten des Second-Level-Supports, so wird diese an den Third-Level-Support weitergeleitet (eskaliert).
- Der Third-Level-Support setzt sich aus Spezialisten einzelner Fachabteilungen bzw. des Herstellers zusammen und stellt so die höchste Eskalationsstufe innerhalb einer Supportorganisation dar [16].

Wie bereits angedeutet, ist das Supportteam eines Softwareherstellers oder eines Softwarevertriebsunternehmens erster Ansprechpartner für die Anwender und bildet somit das Bindeglied zwischen dem Vertriebsteam und der Softwareentwicklung bzw. zwischen Softwareentwicklung und Endkunde. Da im CAD-Markt die Anwendungen durchweg technischer Natur sind, ist die komplexe Implementierung einer Software beim Kunden ohne das Expertenwissen und die fundierten technischen Kenntnisse des Supports nicht zu bewältigen. Darüber hinaus hat der Support nach dem Verkauf eines Softwareproduktes den umfangreichsten Kontakt mit den Kunden, da diese mit fachlicher Kompetenz bei auftretenden Problemen unterstützt werden müssen. Der Support stellt dadurch oftmals als erstes fest, ob auf Kundenseite ein zusätzlicher Bedarf an Produkten, Schulung oder Beratung vorliegt und kann diese Informationen an den Vertrieb weiterleiten. Daraus lässt sich die Forderung ableiten, dass die vertrieblichen Strukturen eines Softwareunternehmens oder Systemhauses ebenfalls bei der Konzeption eines SSMS als Erweiterung eines SLM berücksichtigt werden müssen.

Im Anschluss sollen die Prozesse bei der Verarbeitung von Softwareproblemen durch den Support in allen Phasen der Geschäftsbeziehung zwischen Softwarehersteller und Endkunden analysiert werden. Darüber hinaus soll auch ermittelt werden, wie die dabei anfallenden Daten und wie das sich zwangsläufig mehrende Expertenwissen auf Seiten des Supports erfasst und verteilt werden.

2.2. Aufgaben des Supports im Umfeld von CAD-Anwendungen

Wie bereits erwähnt, ist der am häufigsten anzutreffende Vertriebskanal im Mainstream-CAD-Markt eine Konstellation mit indirekten Vertriebswegen vom Softwarehersteller über einen Softwarehändler (VAR) zum Endkunden. Hierbei lässt sich das Tätigkeitsfeld des Supports in drei Hauptbereiche einteilen:

- Unterstützung der Kunden und der Hersteller bei der Behebung von Softwarefehlern.
- Beratung der Kunden hinsichtlich der Softwareimplementierung und Distribution von (softwarerelevantem) Expertenwissen.
- Erstellung von Anforderungskatalogen für kundenspezifische Softwareanpassungen und Zusatzapplikationen.

Häufig treten die VAR auch als Softwareentwickler auf, etwa bei der Programmierung von Zusatzapplikationen oder im Rahmen kundenspezifischer Software-Anpassungen, dem so genannten Customizing. Die grundsätzlichen Aufgaben des Supports werden dadurch nicht berührt, es verkürzen sich jedoch die Prozessketten, bedingt durch die Tatsache, dass es hier nur noch eine einstufige Supporthierarchie gibt.

2.2.1. Aufgaben im Umgang mit Softwareproblemen

Bezogen auf ein Mainstream-CAD-Produkt übernimmt der VAR den 1st-Level-Support und ist damit die Anlaufstelle für alle softwaretechnischen Probleme seiner Kunden. Diese Softwareprobleme lassen sich dabei in 3 Kategorien einordnen und zwar in solche,

1. die durch Unkenntnis oder unsachgemäße Bedienung einer Software entstehen oder
2. die durch Fehler im Quellcode einer Software auftreten oder
3. die aufgrund funktioneller Beschränkungen eines Programms auftreten

Stößt der Endkunde softwareseitig auf ein Problem, so informiert dieser das Support-Team seines Händlers. Der Support entscheidet - soweit möglich - zu welcher der oben genannten Kategorien das Problem gehört. Abhängig davon werden unterschiedliche Prozesse zur Lösungsfindung initiiert. Sind Problem und Lösung bekannt, so übermittelt der 1st-Level-Support des Händlers die Lösung direkt an den Kunden und schließt im Erfolgsfall die Anfrage ab. Ist die Lösung nicht in einer Form bekannt, in der sie umgehend gegenüber dem Kunden kommuniziert werden kann, so verifiziert, klassifiziert und dokumentiert der 1st-Level-Support zunächst das Problem.

Im Rahmen der Verifizierung wird zunächst festgestellt, ob und wie das Problem zu reproduzieren ist. Darüber hinaus werden vom Support alle notwendigen Informationen über die eingesetzte Hard- und Softwareplattform des Kunden gesammelt und evtl. erforderliche Daten angefordert.

Während der Klassifizierung des Problems wird ermittelt, in welchem funktionellen Bereich der Software das Problem angesiedelt ist und wie stark die Produktivität des Kunden durch das Problem beeinträchtigt wird.

Die anschließende Dokumentation soll sicherstellen, dass jeder im Rahmen des Problemlösungs-Prozesses beteiligte Mitarbeiter – also Entwickler, Vertriebsbeauftragte oder andere Support-Mitarbeiter - Zugriff auf all jene Informationen und Daten bekommen, die notwendig sind, um das Problem nachzuvollziehen und den Fehler lokalisieren zu können.

Anhand der gesammelten Informationen versucht ein Supportmitarbeiter zunächst mit Hilfe von Softwaredokumentationen, Wissensdatenbanken und seines Support-Teams eine Lösung für das Problem zu ermitteln. Kann der 1st-Level-Support das Problem nicht lösen, oder wird das Problem in die letzten beiden Kategorie eingestuft, so übermittelt der 1st-Level-Support die Ergebnisse der Voruntersuchung an den Support des Softwareherstellers. Dieser 2nd-Level-Support übernimmt den Fall und verifiziert das Problem anhand der übermittelten Dokumentation und der zur Verfügung gestellten Daten erneut. Zusätzlich gleicht dieser das Ergebnis mit dem Inhalt von Wissensdatenbanken ab, um herauszufinden, ob das Problem bekannt ist und ob gegebenenfalls bereits eine Lösung existiert. Ordnet der 2nd-Level-Support das Problem in die Kategorie 1. ein, wird also das Problem durch Unkenntnis oder einen Bedienfehler verursacht, so erhält der 1st-Level-Support umgehend eine Mitteilung hierüber. Die Dokumentation und Vermittlung der kompletten Lösung für den Endkunden obliegt anschließend wiederum dem 1st-Level-Support.

Wenn sich herausstellt, dass das Problem durch einen Softwarefehler verursacht wurde, so übermittelt der 2nd-Level-Support die Ergebnisse seiner Untersuchung an die Softwareentwicklung. Der 1st-Level-Support erhält zusammen mit einer eindeutigen Kennung die Bestätigung, dass es sich bei dem geschilderten Problem um einen Softwarefehler handelt.

Die Kennung ermöglicht von diesem Zeitpunkt an die Abfrage des Bearbeitungsstatus für die Kundenanfrage. Nach der Beseitigung des Fehlers durch die Entwicklung testet der 2nd-Level-Support die überarbeitete Software hinsichtlich des Problems und informiert den 1st-Level-Support über das Testergebnis. Verlaufen die Tests erfolgreich, so wird die fehlerbereinigte Version der Software schließlich im Rahmen eines so genannten Patches oder Servicepacks an den Kunden übermittelt. Bewertet der 2nd-Level-Support das Problem hingegen als funktionale Beschränkung der Software und nicht als Fehler, so wird vom 1st-Level-Support ein Verbesserungsvorschlag aus der Problembeschreibung abgeleitet und an die Softwareentwicklung übermittelt. Der Support des VAR begleitet somit den Kunden im gesamten Problemlösungsprozess.

Der Trend, dass die VAR in zunehmendem Maße selber als Softwareentwickler auftreten, ergibt sich aus den Anforderungen der Kunden nach maßgeschneiderten Lösungen und Anpassungen von Standardsoftware für Ihre speziellen Anwendungen. Des Weiteren entwickeln die VAR Softwareprodukte für regionale Märkte, die für die global agierenden Hersteller von CAD-Produkten aus unterschiedlichen Gründen nicht hinreichend attraktiv sind. Die Gründe für die mangelnde Attraktivität solcher Nischenlösungen sind vielfältig. Zum einen streben die Hersteller von Mainstream-CAD-Produkten - wie die Bezeichnung bereits vermuten lässt - nach universellen Lösungen, die einen entsprechend großen Absatzmarkt, (den sog. Mainstream) bedienen können. Zum anderen sind die Hersteller bestrebt, die endlichen Ressourcen an Programmierern für die Entwicklung des Kernprodukts einzusetzen und sich damit auf die eigenen Kernkompetenzen zu konzentrieren. Aus diesen Gründen stellen die meisten CAD-Anwendungen eine mehr oder minder breit angelegte Programmierschnittstelle zur Verfügung, die einen Zugriff auf eine ganze Reihe von Funktionen der Software gestattet.

Die Frage, ob und wie weit ein CAD-Hersteller sein Produkt selber an regionale Märkte anpasst oder Nischenlösungen entwickelt, ist in höchstem Maße eine strategische. Aufgrund der schlankeren Kostenstruktur, die eine Konzentration auf die sukzessive Verbesserung und den Ausbau der Kernfunktionen eines CAD-Systems ermöglicht, hat sich im Mainstream-CAD jedoch die beschriebene Methodik durchgesetzt und bildet ein Hauptunterscheidungsmerkmal zu den Highrange-Systemen, wo die Zusatz-Applikationen oft aus einem Hause kommen oder zumindest unter einer gemeinsamen Führung programmiert werden. Die Grenzen zwischen beiden Ansätzen sind jedoch fließend.

2.2.2. Aufgaben bei Implementierung und Wissensdistribution

Der Support nimmt beim Aufbau und der Distribution von applikationsrelevantem Wissen eine Schlüsselposition ein. Zu applikationsrelevantem Wissen ist hier das Wissen bezüglich des Handling und des Funktionsumfangs der vertriebenen Software zu zählen, sowie jene Erkenntnisse bezüglich der notwendigen Rahmenbedingungen für die erfolgreiche Implementierung eines Softwareproduktes in eine vorhandene, auszubauende oder neu zu schaffende IT-Infrastruktur. Mit applikationsrelevantem Wissen sind jedoch auch Kenntnisse über Strategien zur Umsetzung kundenspezifischer Anforderungen an die zu erzeugenden Applikationsdaten gemeint. Letzteres setzt neben fundierten technischen Kenntnissen zum Verständnis der gestellten Anforderungen auch ein gewisses Maß an Erfahrungen voraus. Wissensdatenbanken unterstützen den Support bei der Bewältigung der im Rahmen dieser Prozesse auftretenden Probleme. Im Einzelnen lassen sich bei den Aufgaben des Supports bezüglich der Wissensdistribution folgende Bereiche unterscheiden:

- Vermittlung von Wissen bezüglich der Implementierung der Software in die IT-Infrastruktur des Kunden unter Berücksichtigung vorhandener Hard- und Software.
- Vermittlung von Grundwissen und Fertigkeiten im Umgang mit der Software durch Schulungsmaßnahmen in Trainingszentren eines Händlers oder bei dem Kunden vor Ort.
- Erstellung von Richtlinien zur Erzeugung strukturierter Produktdaten unter Berücksichtigung von kundenspezifischen Anforderungen.
- Ausbau des Grundwissens in Bezug auf kundenspezifische Anforderungen durch ergänzende Schulungsmaßnahmen und Beratung zur Optimierung des Softwareeinsatzes.

2.2.3. Aufgaben im Rahmen kundenspezifischer Softwareanpassung

Als primärer Ansprechpartner der Endkunden bei technische Fragen übernimmt der Support die Bedarfsermittlung für kundenspezifische Softwareanpassungen, indem er zunächst bewertet, ob sich eine bestimmte Kundenanforderung mit vorhandenen Funktionen der eingesetzten Software umsetzen lässt, oder ob die gewünschte Funktion durch die eigene Entwicklungsabteilung programmiert werden muss. Ist Letzteres der Fall, erstellt der Support ein grobes Anforderungsprofil für die gewünschte Funktionalität und vergibt gegebenenfalls Prioritäten für die verschiedenen Unterfunktionen. Dieses Anforderungsprofil übermittelt der

Support anschließend an den Vertrieb und die Entwicklung. Die nimmt ihrerseits Kontakt zum Kunden auf, um eine Feinabstimmung der Anforderungen im Rahmen der Erstellung eines Pflichtenheftes vorzunehmen. Parallel dazu wird die Umsetzbarkeit der Anforderungen im Zusammenhang mit dem zur Verfügung stehenden Kostenrahmen bewertet. Wird der Auftrag erteilt, so programmiert die Entwicklung die geforderte Funktion nach den Vorgaben des Kunden. Diese neu entwickelte Funktion wird anschließend im Sinne des SLM vom Support auf ihre Praxistauglichkeit und eventuelle Fehler getestet. Außerdem überprüft dieser die Dokumentation auf Konsistenz. Nach Programmfreigabe erfolgt die Übermittlung des compilierten Codes und der dazu gehörigen Dokumentation an den Kunden, gefolgt von einer Einweisung in die programmierten Funktionen.

2.3. Analyse einiger Praxisbeispiele

Im Folgenden soll am Beispiel der 3D-CAD-Anwendung SolidWorks dargestellt werden, wie die unterschiedlichen Supportprozesse bei der Interaktion zwischen Kunde, VAR und Softwarehersteller im Detail ablaufen und welche Werkzeuge dabei zum Einsatz kommen. Bei der Analyse ist nicht so sehr die Anzahl der befragten Unternehmen entscheidend, sondern vielmehr die Komplexität der unterschiedlichen Applikationen und deren Entwicklungskonstellationen, die wiederum in engem Zusammenhang mit den Vertriebswegen stehen. Die gewählten Beispiele decken die häufigsten Entwicklungskonstellationen ab, so dass eine hinreichend allgemeingültige und gleichzeitig umfassende Aussage über die momentane Arbeitsweise getroffen werden kann.

Die Auswahl der CAD-Software SolidWorks und von Applikationen aus dem Umfeld dieser Anwendung bedeutet nicht, dass eine Analyse anderer CAD-Programme nicht zu ähnlichen Ergebnissen führen würde. Diese Auswahl wurde in erster Linie deshalb getroffen, da der Verfasser über umfangreiche Erfahrungen mit den CAE⁶-Anwendungen im Umfeld von SolidWorks verfügt und dadurch einen guten Zugang in dieses Marktsegment hat, wodurch eine breit angelegte Analyse durchgeführt werden kann. Darüber hinaus finden sich im Umfeld des Kernprodukts von SolidWorks eine breite Palette von Softwareprodukten, die unmittelbar in die SolidWorks-Benutzerumgebung eingebettet sind und über einen Zugriff auf interne Funktionen verfügen, die über die öffentliche Programmierschnittstelle (API) nicht angesprochen werden können.

Der Großteil der für SolidWorks erhältlichen Zusatzapplikationen wird jedoch über die API an das Kernprodukt angebunden und von eigenständigen Unternehmen entwickelt. Nicht selten wird ein spezielles Produkt dabei für unterschiedliche 3D-CAD-Anwendungen angeboten –

⁶ CAE – Computer Aided Engineering

die Gründe dafür liegen auf der Hand und sind wirtschaftlicher Natur. Damit eine optimale Integration gewährleistet ist, sind durch den Hersteller umfangreiche Kriterien und Reglements für die offizielle Akkreditierung als Zusatzanwendung vorgegeben. Eine Applikation erhält seitens des Herstellers die höchste Akkreditierung, wenn die erzeugten Daten vollständig und bidirektional assoziativ zu den nativen SolidWorks-Daten sind. Zudem muss die Applikation selbst vollständig in die Benutzeroberfläche von SolidWorks eingebunden sein (sog. „single-window Integration“).

Am Ende der Palette an Applikationen im Umfeld einer CAD-Anwendung steht das so genannte Customizing. Dies beginnt bei der Implementierung von Makros und einzelnen Befehlen, der Automatisierung von kundenspezifischen Standardabläufen bis hin zu komplexeren Programmen wie z.B. Postprozessoren für CAM-Integrationen. Sämtlichen im Rahmen des Customizing erstellten Programmen ist jedoch gemeinsam, dass diese für einen speziellen Anwendungsfall und in der Regel nur für einen einzelnen Kunden programmiert werden, anstatt für einen breiten Markt.

Somit finden sich im Umfeld von SolidWorks alle denkbaren Konstellationen im Zusammenspiel zwischen Kunde, Support und Entwicklung. Diese werden im Folgenden anhand einer Reihe von Beispielen hinsichtlich der Integration des Softwaresupports in das SLM untersucht und dokumentiert. Die ausgewählten Unternehmen sind mit eigenen Mitarbeitern in den Organisationsbereichen Entwicklung, Vertrieb und Support im Umfeld von Mainstream-3D-CAD-Anwendungen tätig und bearbeiten das folgende Spektrum:

- Entwicklung einer Standalone 3D-CAD-Lösung SolidWorks. (vgl. Kap. 2.3.1)
- Entwicklung der Zusatzapplikation Varbox. (vgl. Kap. 2.3.2)
- Durchführung von kundenspezifischen / maschinenspezifischen Softwareanpassungen. (vgl. Kap. 2.3.3)
- Programmierung von kundenspezifischen Funktionserweiterungen (vgl. Kap. 2.3.4)

Obwohl die einzelnen zu analysierenden Konstellationen sehr unterschiedlich sind, kann für die gemeinsame Schnittmenge ein einheitlicher Fragenkomplex vorab formuliert werden (vgl. Kapitel 9.Anhang). Diesem liegt im Ansatz das Verständnis eines SSMS als Erweiterung eines SLM-Systems zu Grunde. Im Kontext der vorherigen Untersuchungen sollen dabei folgende Punkte analysiert werden: Unternehmen, Software, Aufgabenverteilung, Prozessmanagement und Qualitätssicherung, Informationsmanagement, Wissensmanagement und Datenhaltung.

2.3.1. Beispiel 3D-CAD-Standalone-Produkt SolidWorks

Unternehmen

Die 3D-CAD-Software SolidWorks ist ein Produkt der SolidWorks Corp. mit Sitz in Concord, Massachusetts, USA. Zurzeit beschäftigt das Unternehmen weltweit ca. 600 Mitarbeiter⁷. Die Firma ist seit 1997 zu 100% in Besitz von Dassault Systemes S.A. Die SolidWorks Corp. ist in vielen Ländern der Welt mit Dependancen vertreten, entwickelt wird im Stammhaus in Concord / USA, in Cambridge / UK und in Pune / Indien. Von den rund 600 Mitarbeitern sind ca. 220 in der Entwicklung tätig.

Software

SolidWorks ist nach eigenen Angaben das zurzeit führende 3D-CAD-System im Mainstream-CAD-Markt mit rund 600.000 installierten Lizenzen⁸. Das Produkt wird weltweit vertrieben und ist derzeit in 14 Sprachen erhältlich. Zielmärkte sind die mechanische Konstruktion für Industrie, Medizin, Wissenschaft, Bildung, Technologie, Transport und Konsumgüter. SolidWorks ist eine Standalone-Anwendung, die auf dem Betriebssystem Microsoft-Windows aufsetzt. Die Software ist eigenständig, verwendet jedoch in Teilbereichen Funktionen von MS-Excel, MS-Word und MS-Access. Für die Geometrie-Erzeugung nutzt SolidWorks den Parasolid-Modellierkern. Dabei handelt es sich um einen 3D-CAD-Volumenmodellierkernel, der von der Firma Unigraphics Solutions lizenziert wird. Auf Parasolid setzen neben SolidWorks noch über 60 weitere CAD-, FEM⁸- und CAM-Systeme verschiedener Anbieter auf.

Aufgabenverteilung

Der 1st-Level-Support wird von rund 400 Vertriebspartnern (VAR) weltweit übernommen. Der 2nd-Level-Support wird durch das Supportteam von SolidWorks geleistet. Ein klar abgegrenzter 3rd-Level-Support existiert so gesehen nicht, seine Aufgaben werden vom 2nd-Level-Support mit abgedeckt. Hauptaufgabe des 2nd-Level-Supports ist die Unterstützung der Händler in funktionalen Fragen, die Verifizierung von Softwarefehlern sowie die Initiierung der entsprechenden Prozesse zu deren Behebung. Die regionalen Niederlassungen von SolidWorks beschäftigen ebenfalls eigene Supportmitarbeiter, diese sind jedoch vorwiegend in der Vertriebsunterstützung der regionalen Händler tätig. Die Supporttätigkeiten der VAR und hier insbesondere das so genannte CALL-Handling, also die

⁷ Stand 2-2007

⁸ FEM – Finite Elemente Methode

Abwicklung von Serviceanfragen für die CAD-Anwendung zeigt die Abbildung 2.3-1.

Im Umfeld der Vorausentwicklung und bei Softwaretest spielt der Support bei SolidWorks nur noch eine untergeordnete Rolle. Hier wird inzwischen der überwiegende Teil der Funktionalitäten von Servicepacks und neuen Releases automatisch mit Hilfe von Makros und Skripts getestet. Die Supportorganisation ist in Teams nach funktionalen Schwerpunkten der CAD-Software aufgeteilt. Die Anfragen der Kunden werden vom 1st-Level-Support entweder per Webclient und Internetportal des Herstellers oder per eMail bzw. telefonisch an den 2nd-Level-Support übermittelt. Bei Anfragen über das Internetportal wird automatisch ein **Service-Request (SR)** generiert und mit einer Nummer (SR-Nummer) versehen, welche im weiteren Verlauf der Bearbeitung eine einfache Verfolgung und Identifikation der Anfragen für alle Beteiligten ermöglicht. Das Internetportal bildet einen Zugang zum CRM⁹-System des Softwareherstellers. In diesem System werden sämtliche kundenrelevanten Informationen und die SR-Nummer in einer Datenbank verwaltet. Bei Anfragen per eMail oder Telefon müssen die Anfragen von den Supportmitarbeitern des 2nd-Level-Supports per Hand in die Datenbank des CRM-Systems eingepflegt und mit einer SR-Nummer versehen werden.

Prozessmanagement und Qualitätssicherung

Kundenanfragen werden vom Support des VAR nach funktionalen Themengebieten und Dringlichkeit kategorisiert. Über den Themenschlüssel erfolgt automatisch die Zuweisung der Anfrage in eine Queue¹⁰ des jeweiligen Kompetenzteams beim Hersteller. Ein freier Mitarbeiter des Teams weist sich selbst die Anfrage aus der Queue zu und legt damit seine Zuständigkeit für den weiteren Bearbeitungsprozess fest. Der Status der Anfrage ändert sich damit von „*Pending*“ zu „*Open*“. Gegebenenfalls benötigte Applikationsdateien werden, so der Händler diese nicht bereits bei der Erstellung der Anfrage auf den Server des Portals geladen hat, über diesen beim Endkunden angefordert. Umfangreiche Datensätze, etwa von umfassenden 3D-Baugruppen, werden unter der SR-Nummer auf einem FTP-Server gespeichert. Die Firma SolidWorks hat zum Zwecke der Fehlerdokumentation ein eigenes Softwaremodul in die Applikation integriert, die bei Bedarf jene Benutzer- und Programmaktivitäten als Video aufzeichnet, welche zu einem Fehler führen. Zudem sammelt die Applikation alle benötigten System-Informationen und verpackt diese zusammen mit den betroffenen Applikationsdaten in ein ZIP¹¹-Archiv. Der 2.Level-Support verifiziert das in der Anfrage geschilderte Problem anhand der zur Verfügung gestellten Daten und leitet dies bei Verdacht auf einen Softwarefehler (Bug) an die zuständige Entwicklungsabteilung weiter.

⁹ CRM - Customer Relationship Management System

¹⁰ Queue - engl. Warteschlange, Pufferinhalt eines sequentiellen Anfragespeichers

¹¹ ZIP - Dateiformat für die komprimierte Speicherung von Dateien

Bestätigt die Entwicklung das Problem als Fehler im Quellcode der Software, so wird diesem ein **Software-Performance-Report** (SPR) innerhalb des SLM-Systems der Entwicklung zugeteilt und mit dem Datenbankeintrag der Serviceanfrage, also dem SR, im Partnerportal verlinkt. Der SPR wird anschließend vom 2.Level-Support mit einer unverbindlichen Perspektive für die Lösung des Problems per eMail an den Support des VAR und von diesem an den Endkunden übermittelt.

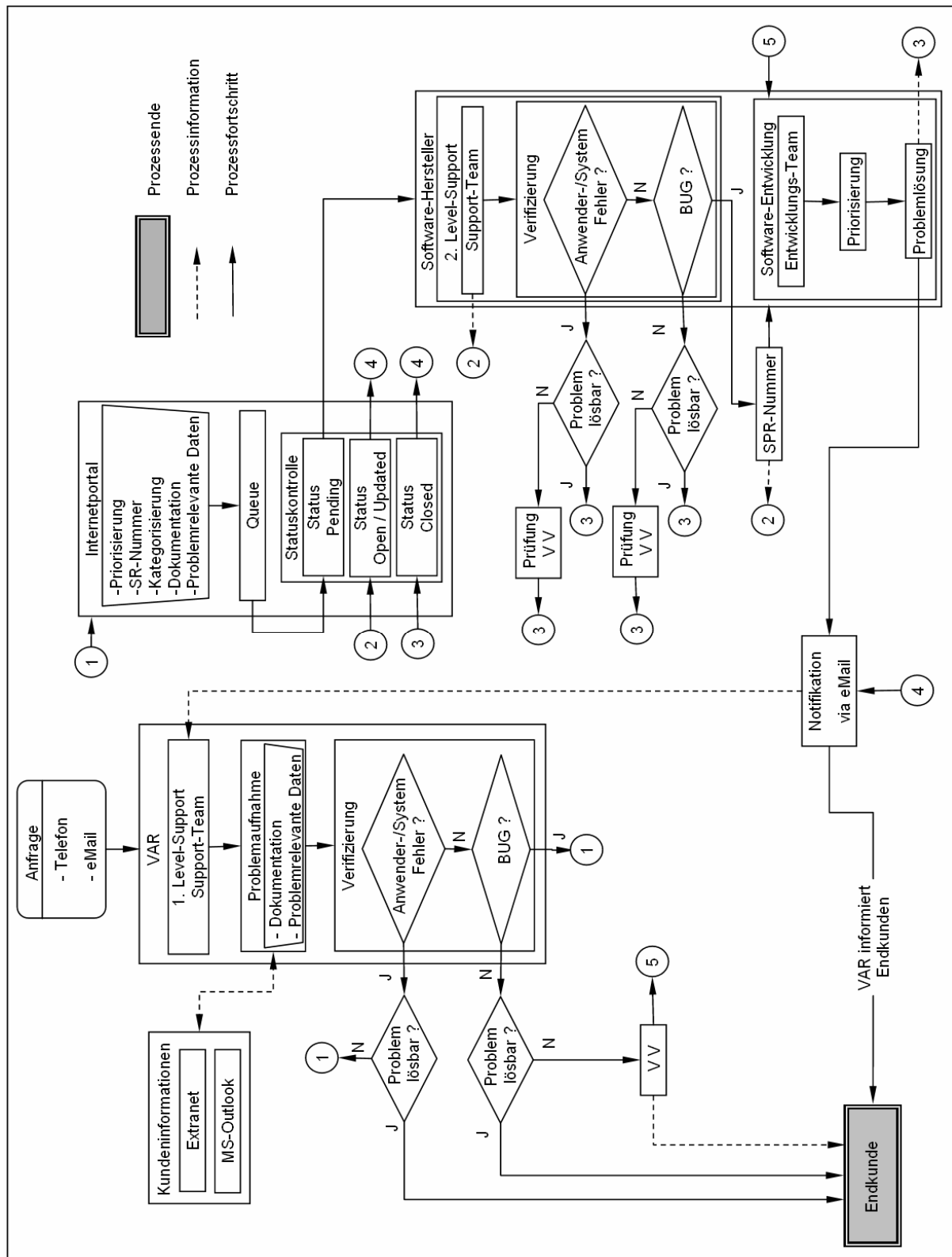


Abbildung 2.3-1: CAD-Anwendung (Fallbeispiel 1)

Der zeitliche Rahmen zur Behebung eines Bugs wird generell von unterschiedlichen Faktoren beeinflusst:

- Durch das Priorisieren des Bugs durch den Händler.
- Durch die Häufigkeit, mit der ein Bug von unterschiedlichen Endkunden gemeldet wird (Customer Hits).
- Durch die von der Entwicklung festgelegte Art und Schwere des Problems (Nature of Problem). Diese Priorität ist wiederum von unterschiedlichen Faktoren abhängig, etwa davon, ob durch den Bug eine zuvor vorhandene Funktion nicht mehr nutzbar ist oder ob das Problem zu einem Absturz der Software führt.

Hat ein Entwickler einen Fehler lokalisiert und behoben, so wird der geänderte Code von ihm auf einen eigenen Server des SLM geladen und dadurch von ihm freigegeben. Zusammen mit den revidierten Quellcodefragmenten der anderen Entwickler wird hieraus alle 24 Stunden automatisch ein Servicepack generiert, auf die entsprechende CAD-Software angewendet und von der Qualitätssicherung (QS) per Makros und Skripts ebenfalls automatisch getestet. Findet die QS keine Fehler, so folgen noch einige manuelle Tests bis zur Freigabe im Status „*Early-Visibility*“, einem Beta-Testprogramm, in welches ausgewählte Endkunden eingebunden sind. Tauchen während des Betatests durch die Endkunden keine Mängel auf, so wird das Servicepack von der QS freigegeben und veröffentlicht, also zum Download im Subskriptionskunden-Bereich der SolidWorks-Homepage bereitgestellt.

Zu jedem Servicepack wird eine Liste mit den behobenen SPR veröffentlicht. Anhand dieser Liste können somit Händler und Kunden nachvollziehen, ob ein gemeldeter Bug innerhalb eines Servicepacks behoben wurde. Anhand der SR-Nummer hingegen kann vom Händler außerdem zu jeder Zeit der Status einer Anfrage abgerufen werden. Ist ein Fall abgeschlossen, so wird der Status der Anfrage vom 2.Level-Support in „*Closed*“ geändert. Ergeben sich neue Erkenntnisse zu einem Problem oder stellt sich heraus, dass es nach einem Servicepack oder Releasewechsel erneut auftritt, so kann der Händler den Status einer Anfrage jederzeit wieder in „*Open*“ ändern und damit den gesamten Mechanismus der Fehlerverifizierung und –behebung erneut initiieren.

Zusätzlich zu den hier vorgestellten Mechanismen tritt wöchentlich ein Development-Team aus den Projektleitern der unterschiedlichen Entwicklungsteams zusammen, um die dringlichsten Probleme zu besprechen, Prioritäten zu setzen und das Vorgehen zu deren Lösung zu koordinieren. Auf die Tagesordnung dieser Sitzungen können auch die Supportmitarbeiter der regionalen Dependancen Einfluss nehmen und somit bestimmten Problemen eine zusätzliche Gewichtung verleihen, etwa wenn ein bestimmter Bug bei einem regionalen Großkunden schwere Produktivitätseinbußen verursacht und einer

schnellstmöglichen Lösung bedarf.

Die Eingabe von **Verbesserungsvorschlägen (VV)** erfolgt losgelöst vom Kundenportal über einen Link im Subskriptionsbereich der SolidWorks-Homepage. Während sämtliche problembezogenen Aspekte durch die Händler an den Support des Softwareherstellers übermittelt werden, erfolgt die Anfrage für VV direkt durch den Endkunden. Dabei wird der VV durch den Endkunden in vorgegebene Kategorien nach Funktionsmodulen kategorisiert. Die Weiterleitung der VV erfolgt aufgrund dieser Einstufung von der Webseite direkt an den zuständigen Produktmanager in Form von automatisch generierten eMails. Es sind keinerlei Methoden zur Nachverfolgung von VV hinterlegt. Das priorisieren der VV erfolgt hier offiziellen Angaben zufolge entsprechend der Häufigkeit mit der eine bestimmten Funktionalität Nachgefragt wird, aber auch aufgrund der Machbarkeit innerhalb eines vorgegebenen Kostenrahmens und der von der Vorausentwicklung festgelegten Ziele für zukünftige Versionen.

Informationsmanagement

Das Kundenportal von SolidWorks stellt neben einer Umgebung für das Absetzen von Serviceanfragen auch eine Reihe von Kundeninformationen bereit. So sind hier notwendige Daten zur Firma des Kunden, wie Name, Ansprechpartner oder Lieferadresse gespeichert. Zusätzlich werden alle Informationen über Anzahl und Art der Softwarelizenzen, zugehörige Seriennummern und Freischaltcodes bereitgestellt. Die Händler können jedoch nur die Informationen der von Ihnen betreuten Kunden abrufen, während die Supportmitarbeiter von SolidWorks Zugriff auf die Daten aller Kunden haben.

Wissensmanagement

SolidWorks stellt seinen Wartungskunden auf der eigenen Homepage eine Datenbank mit häufig gestellten Fragen (FAQ¹²s) zu verschiedenen Themen zur Verfügung. Auf den Inhalt kann über eine Suchmaske mittels Volltextsuche oder über eine Themenauswahl zugegriffen werden. Die FAQs werden vom 1st-Level-Support erstellt und gepflegt. Es handelt sich dabei in erster Linie um allgemeine Abhandlungen und Detail-Informationen zu bestimmten Funktionen. Verfahrensweisen zu bestimmten Problemstellungen oder Tipps zur Implementierung und Abbildung spezifischer Dokumentstrukturen finden sich nur in Ausnahmefällen. Da ein Revisionsmechanismus fehlt, sind die Beiträge inhaltlich teilweise erheblich veraltet oder gänzlich überholt und somit auf aktuelle Softwareversionen nicht mehr anwendbar. Darüber hinaus stellen die regionalen Dependancen Ihren Händlern

¹²FAQ - engl. Frequently asked Questions

regelmäßig Newsletter mit aktuellen Informationen zu bestimmten Themen zu. In der Regel handelt es sich dabei um die Behebung von auftretenden Problemen im Zusammenhang mit aktuellen Servicepacks oder Versionen oder um anwendungsorientiertes Applikations-Knowhow.

Datenhaltung

Die Fa. SolidWorks setzt im Umfeld von Support und Entwicklung unterschiedliche IT-Systeme ein, unter anderem die Software Siebel CALL-Center von Siebel Systems. Es handelt sich hierbei um eine datenbankbasierte CRM-Lösung mit einer in einen Webbrowser integrierten Zugriffsebene, einem sog. Webportal. Alle supportrelevanten Kundendaten werden, mit Ausnahme großer Applikations-Datensätze, in der Datenbank des Siebel-CRM verwaltet. Solche umfangreichen Datensätze werden in der Regel mittels FTP¹³ auf einem separaten Server unter der SR-Nummer abgelegt und sind nicht direkt mit der Datenbank des CRM verknüpft. Zugriffsrechte auf alle Supportbereiche der Datenbank werden mittels der Siebel-Software zugeteilt. In der Software-Entwicklung verwendet SolidWorks eine selbst entwickelte Datenbanklösung auf Oracle-Basis. Beide Datenbanken sind nur im Rahmen von aus SR resultierenden SPR verlinkt.

2.3.2. Beispiel CAD-Zusatzapplikation Varbox

Unternehmen

Das Normteile- und Utility-Paket Varbox ist ein Produkt eines VAR. Dieses mittelständische Unternehmen beschäftigt rund 92 Mitarbeiter¹⁴, betätigt sich ausschließlich auf dem deutschsprachigen Markt (Deutschland und Österreich) und ist mit 8 Niederlassungen in der gesamten Bundesrepublik vertreten. Von den 92 Mitarbeitern sind rund 35 im Support und 4 in der Softwareentwicklung tätig.

Software

Die Software Varbox ist ein Normteile- und Utility-Paket als Zusatzapplikation für SolidWorks. Diese ist als Client-Server-Anwendung ausgelegt, wobei die Client-Software vollständig in die Benutzeroberfläche von SolidWorks integriert ist. Bestandteil der Varbox-Software ist außerdem ein eigenständiges Administrationsprogramm, der Varbox-Wizard.

¹³ FTP : File Transfer Protocol

¹⁴ Stand 2-2007

Die Serveranwendung besteht im Wesentlichen aus einem Datenverzeichnis, welches die generierten Normteile und deren Vorlagen enthält. Darüber hinaus beinhaltet die Serveranwendung eine Datenbank, mit deren Hilfe sämtliche Parameter und Metadaten der Normalien gepflegt und bereitgestellt werden. Zurzeit sind rund 1500 Softwarelizenzen bei ca. 1000 Kunden im Einsatz. Neben der erweiterbaren Datenbank mit den gängigsten Normteilen für den Maschinenbau stellt die Varbox eine Integration für ein umfangreiches Normalienpaket eines Drittanbieters und eine Reihe von Zusatzfunktionen, so genannte Utilities, für den Konstruktionsalltag bereit. Als Datenbanksystem kommt MS-Access zum Einsatz, programmiert wird in C++. Die Anbindung an SolidWorks wird mit Hilfe der SolidWorks-API realisiert. Die Anforderungen an die Systemumgebung decken sich mit jenen von SolidWorks.

Aufgabenverteilung

Der 1st Level Support wird von den Supportmitarbeitern des VAR (vgl. Abbildung 2.3-2) übernommen. Ein 2nd-Level-Support existiert nicht. Hauptaufgabe des Supports ist auch hier die Beratung der Endkunden in funktionalen Fragen, die Verifizierung von Softwarefehlern und die Initiierung von Prozessen zu deren Beseitigung. Daneben wird der Support bei Betatests eingesetzt.

Die Supportorganisation ist in PreSales- und Postsales-Support aufgeteilt. Der Postsales-Support ist wiederum in eine CAD- und eine CAM-Sparte untergliedert. Darüber hinaus haben einzelne Supportmitarbeiter erweiterte Kompetenzen in unterschiedlichen Spezialgebieten erworben, wie beispielsweise bei FEM- und PDM-Anwendungen oder bezüglich der Administration der Varbox, und verfügen in diesen Gebieten über einen Experten-Status. Bei Kundenanfragen in den entsprechenden Themengebieten werden diese Experten hinzugezogen falls der Bearbeiter sie nicht selbst beantworten kann.

Die Anfragen werden per Telefon oder eMail an das CALL-Center, die sog. Hotline des Supports gerichtet. Im ersten Fall wird die Anfrage über eine zentrale Hotline-Nummer an die Telefonanlage des CALL-Centers und von dort automatisch auf einen freien Hotlinemitarbeiter weitergeschaltet. Anfragen per eMail werden an ein zentrales Postfach geleitet, welches von allen Supportmitarbeitern eingesehen werden kann. Von dort aus werden sie von den Hotlinemitarbeitern chronologisch abgearbeitet. Anfragen, deren Bearbeitung den Zeitrahmen von wenigen Minuten übersteigt, werden unter einer automatisch generierten Nummer sowie unter Angabe der Firma und des Anfragestellers, der Problemkategorie und des Bearbeiters in den Helpdesk des Supports eingepflegt.

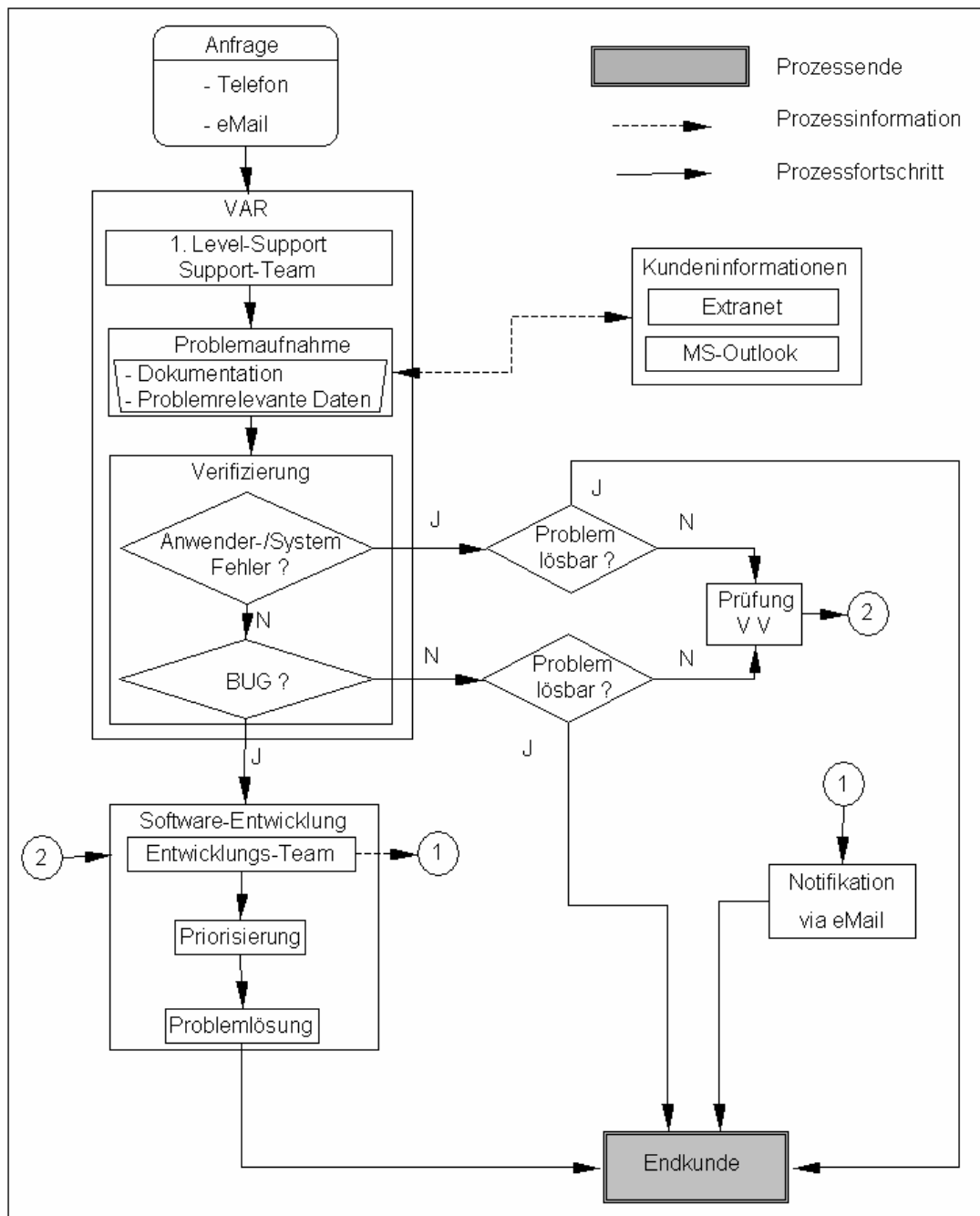


Abbildung 2.3-2: CAD-Anwendung (Fallbeispiel 2)

Beim Helpdesk handelt es sich um eine vom Systemhaus in Eigenregie programmierte Erweiterung von MS-Outlook, die am Standort des CALL-Centers mit der Telefonanlage gekoppelt ist. Dadurch wird es möglich, bei einem Anruf automatisch den dazugehörigen Kontaktdatensatz des Anrufers aus der Adressdatenbank von Outlook aufzurufen. Bei Großkunden gibt es außerdem einen Key-Account-Supporter, der Ansprechpartner für sämtliche Fragen und Probleme des Kunden rund um die CAD-Applikation und deren Zusatzanwendungen ist. Der Key-Account-Supporter koordiniert die Abarbeitung der Anfragen und kommuniziert die Ergebnisse mit dem Kunden.

Für Betatests der Varbox gibt es keinerlei festgelegte Prozeduren oder Zuständigkeiten. Nachdem eine neuen Softwareversion mittels einer eMail an alle Supportmitarbeiter durch die Entwicklung freigegeben wurde, kann prinzipiell jeder Supporter diese installieren und testen. Die Teilnahme an den Tests erfolgt jedoch ausschließlich auf freiwilliger Basis.

Prozessmanagement und Qualitätssicherung

Die Anfragen der Kunden bezüglich der Varbox werden von einem Supportmitarbeiter zunächst als Freitext mit Datum und Uhrzeit unter der Kategorie „Varbox“ in den Helpdesk eingetragen und der Status auf „*In Bearbeitung*“ geändert. In manchen Fällen, vor allem bei Großkunden, erfolgt eine Anfrage auch direkt an die zuständigen Entwickler.

Anschließend werden die Informationen und benötigten Daten von dem Kunden angefordert, welche benötigt werden, um das Problem zu verifizieren. Hierzu zählen Informationen über das verwendete OS¹⁵, die eingesetzte SolidWorks-Version und Varbox-Version mit den entsprechenden Servicepacks sowie eventuell betroffene Applikationsdateien.

Die zu sammelnden Daten und Informationen sind in einer Verfahrensanweisung in Form eines Textdokuments spezifiziert. Die Informationen über die Softwareumgebung des Kunden werden zusammen mit der Problembeschreibung im Helpdesk abgelegt, Applikationsdateien hingegen in einer Ordnerstruktur unter Firma und Datum auf dem Firmenserver. Jeder Supportmitarbeiter hat sowohl auf die Anfragen im Helpdesk, die so genannten CALLs, als auch auf die Kundendaten auf dem Server einen vollständigen Zugriff, Daten und Informationen können also beliebig ergänzt und verändert werden.

Kann die Anfrage vom Supportmitarbeiter anhand der gesammelten Informationen nicht abgeschlossen werden oder liegt die Vermutung nahe, dass es sich um einen Fehler im Quellcode der Varbox handelt, so wird - je nach Verfügbarkeit - ein Experte oder ein Entwickler hinzugezogen und der Fall an diesen übergeben, indem im Helpdesk die Zuständigkeit geändert wird. Der neue Bearbeiter wird über diesen Vorgang mit einer automatischen eMail informiert. Die Entwickler priorisieren, kategorisieren und verifizieren die Probleme anschließend intern und pflegen diese in das Software-Projekt-Management-System ein, auf das sämtliche Entwickler über ein Webportal Zugriff haben. Als Management-System für die Entwicklung kommt Fogbugz von FogCreek Software, eine SQL-basierte Datenbanklösung für Fehlerverfolgung und Zieldefinition in der Softwareentwicklung, zum Einsatz. Außerhalb der Entwicklung erfolgt eine Nachverfolgung von Anfragen nur über deren Status im Helpdesk. Die Quellcode-Verwaltung für neue Releases, Servicepacks und Hotfixes geschieht unter der Verwendung von Visual

¹⁵ OS - Operating System

SourceSafe, einem datenbankbasierten Versionskontrollsystem von Microsoft. Ein Zugriff hierauf besteht ebenfalls nur innerhalb des Entwicklungsteams. Eine Rückmeldung bei Kundenanfragen erfolgt in der Regel direkt zwischen der Entwicklung und dem Endkunden. Die Entwicklungsabteilung schließt letztendlich auch den CALL ab.

Die Eingabe von VV für die Varbox geschieht über unterschiedliche Kanäle wie Vertrieb und Support oder direkt von den Endkunden an die Entwicklung. Eine Vergabe von Prioritäten für VV erfolgt innerhalb der Entwicklung und in Absprache mit der Geschäftsleitung in Bezug auf den Aufwand zur Umsetzung und auf die Vereinbarkeit mit den strategischen Zielen für das Softwareprodukt. Außerdem hat die wirtschaftliche Bedeutung des Kunden für den VAR, gemessen an der Kunden-Installationsbasis, einen starken Einfluss auf die Priorität eines VV. Die VV werden von der Entwicklung in das Entwicklungs-Management-Tool Fogbugz eingepflegt. Eine Nachverfolgung von VV durch Externe ist nicht möglich, der Status eines VV kann bestenfalls über ein persönliches Gespräch mit dem Entwickler ermittelt werden.

Informationsmanagement

Der VAR setzt eine selbst entwickelte SQL-Datenbank mit Webportal, das so genannte Extranet, für das Informationsmanagement innerhalb des Unternehmens ein. Innerhalb dieses Extranets sind unter anderem auch Informationen über Kundennamen, Ansprechpartner, Anzahl und Art der Softwarelizenzen, sowie die dazugehörigen Seriennummern und Wartungsinformationen hinterlegt. Über eine Suchmaske kann die Datenbank (DB) sowohl nach Firmennamen als auch nach Seriennummern durchsucht werden. Für die Varbox existiert darüber hinaus eine eigene Zone, welche spezifische Informationen wie Varbox-Lizenz- und Katalogcodes, assoziierte SolidWorks-Seriennummern, Kundennamen und –nummern sowie Wartungsinformationen zur Varbox enthält und mit der Seriennummern-Datenbank verlinkt ist. Hier werden alle Kunden aufgelistet, die einen Softwarewartungsvertrag für die Varbox abgeschlossen haben.

Wissensmanagement

Für die Implementierung und Anwendung der Varbox sind eine Reihe von FAQs in Form von Textdokumenten im Extranet des VAR verfügbar und somit allen Support-Mitarbeitern zugänglich. Darüber hinaus beinhaltet das Extranet ein Kundenportal für Varboxkunden, wo kundenrelevante Informationen als PDF hinterlegt sind, etwa Hinweise zu Besonderheiten bei einem Upgrade von einem Release auf das Nächste.

Datenhaltung

Die Kundendaten werden in unterschiedlichen Datenbanken gehalten:

- Die Adress- und Kontaktdaten werden in einer allgemein zugänglichen Adressdatenbank in MS-Outlook gespeichert und gepflegt.
- Vertriebsseitig werden die Adress- und Kontaktdaten zusammen mit anderen vertriebsrelevanten Informationen parallel in einem Kontaktdatenmanagementsystem namens ACT! von Sage Software gepflegt. Bei diesem System hat jeder Vertriebsmitarbeiter seine eigene lokale ACT!-Datenbank.
- Zusätzlich werden die Kundendaten zusammen mit den Informationen über SolidWorks- und Varbox-Lizenzen in einer Seriennummerndatenbank innerhalb des Extranets des VAR und in einer dazu verlinkten Varbox-Lizenzdatenbank gehalten.
- Applikationsdateien der Kunden werden lokal in einer Datenbank des bearbeitenden Entwicklers abgelegt und nach Abschluss einer Anfrage gelöscht.

2.3.3. Beispiel kundenspezifische Softwareanpassung

Als Beispiel für die kundenspezifische Anpassung von Standardsoftware wird die Programmierung von Postprozessoren für die in SolidWorks integrierten CAM-Applikationen CAMWorks und SolidCAM analysiert (vgl. Abbildung 2.3-3).

Unternehmen

Im Rahmen der fortschreitenden Integration aller entwicklungs- und fertigungsrelevanten Prozesse gewinnt die Kopplung von CAD und CAM ein ständig wachsendes Gewicht. Dabei spielt die kundenspezifische Softwareanpassung eine ähnlich gewichtige Rolle wie bei der Integration eines PDM- oder PLM-Systems. So wird die vollständige Implementierung eines CAM-Systems erst durch eine Softwareanpassung möglich. Der untersuchte VAR hat innerhalb der letzten Jahre umfangreiche Erfahrungen und Kompetenzen bei diesen Integrationsprozessen gewonnen und zählt damit zu den führenden Anbietern von CAD-CAM-Lösungen für SolidWorks auf dem deutschsprachigen Markt.

Software

Die hier betrachteten Produkte SolidCAM und CAMWorks sind beide von SolidWorks mit dem höchsten Akkreditierungsgrad ausgezeichnet. Damit ist gewährleistet, dass bei

sämtlichen CAM-Daten die vollständige Assoziativität zu den bearbeiteten SolidWorks-Dateien erhalten bleibt und die Applikationen in das Userinterface von SolidWorks integriert sind. Die von beiden Produkten bereitgestellten Funktionalitäten ermöglichen die Generierung von NC-Programmen für die Fertigungsverfahren Fräsen, Drehen, Dreh-Fräsen (Drehen mit angetriebenen Werkzeugen) sowie für das Drahterodieren.

Dazu wird zunächst innerhalb von SolidWorks ein maßstäbliches, virtuelles 3D-Modell eines Bauteils modelliert. Nach der Festlegung eines Koordinatensystems und den Rohteilabmessungen bestimmt der Anwender unter Angabe einer Vielzahl von Fertigungsparametern und der gewünschten Werkzeuge, mit welchen Bearbeitungsstrategien die unterschiedlichen Geometrien des Bauteils gefertigt werden sollen. Anschließend werden diese Strategien simuliert und - wenn das Ergebnis den Anforderungen entspricht - unter der Verwendung eines Postprozessors (PP) ein NC-Programm für eine spezifische Werkzeugmaschine generiert. Mittels eines Bussystems oder per Diskette wird das so erzeugte NC-Programm schließlich auf die Werkzeugmaschine übertragen, wo das Bauteil gefertigt wird.

Ein PP besteht im Wesentlichen aus einem Textdokument, das neben dem Programmkopf mit allgemeinen Maschinen- und Bauteilinformationen detaillierte Anweisungen enthält, wie die simulierten Werkzeugwege, Drehzahlen und Vorschübe in die Programmiersprache der Maschinensteuerung zu übersetzen sind.

Aufgabenverteilung

Aufgrund der speziellen Anforderungen in der CAD-CAM-Kopplung wird im Vertrieb, Support und der PP-Entwicklung ein eigenes Mitarbeiterteam innerhalb eines CAM-Competence-Centers eingesetzt. Sämtliche Anfragen von CAM-Kunden werden innerhalb dieses Teams abgearbeitet. Die Aufträge zur Programmierung eines PP gehen parallel mit dem Kauf der CAM-Software einher, da ein angepasster PP für die Inbetriebnahme eines CAD-CAM-Systems unverzichtbar ist. Mit der Programmierung und Inbetriebnahme von PP ist derzeit ein einziger Mitarbeiter betraut. Dieser legt die Anforderungen in Absprache mit dem Kunden fest und programmiert den PP in Eigenregie mit Hilfe des Editors UltraEdit von IDM Computer Solutions Inc.

Bei Problemen erhält der Programmierer Unterstützung vom Softwarehersteller. Nach Abschluss der Programmierarbeiten nimmt der Programmierer den PP beim Kunden vor Ort in Betrieb und testet diesen anhand eines repräsentativen Kundenbauteils. Aufgrund des hohen Integrationsgrades sind die beschriebenen CAM-Systeme nur gemeinsam mit einer SolidWorks-Installation zu betreiben, so dass beides in einem Kontext betrachtet werden muss.

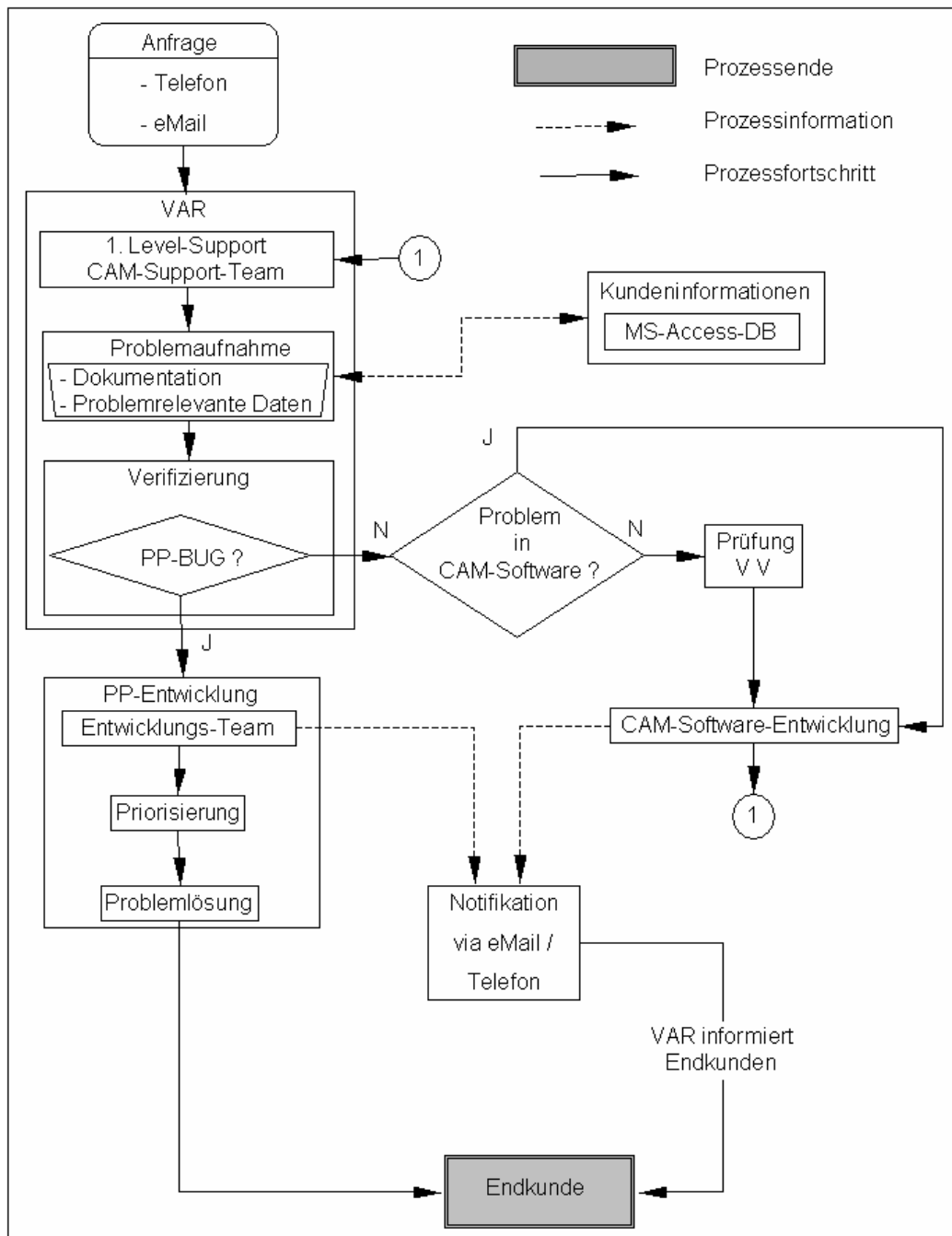


Abbildung 2.3-3: CAD-Anwendung (Fallbeispiel 3)

Oftmals werden diese Systeme deshalb als so genannte Bundles aus einer SolidWorks- und einer CAM-Lizenz verkauft. Der Support für Anfragen bezüglich der SolidWorks-Applikation wird zu ca. 75 % vom CAM-Support abgewickelt. Komplexere Fragestellungen und Softwarefehler in Bezug auf das CAD-System werden per eMail oder Telefon an den SolidWorks-Support des VAR weitergeleitet und von diesem bearbeitet.

Prozessmanagement und Qualitätssicherung

Kundenanfragen zum Thema CAM werden zunächst über die Telefonanlage automatisch an einen Mitarbeiter des CAM-Competence-Centers weitergeleitet. Falls die Frage des Kunden per eMail gestellt wurde, wird diese in ein eigenes Postfach innerhalb von MS-Outlook transferiert. Vereinzelt werden Anfragen auch an die Mitarbeiter der CAM-Vertriebsteams gestellt und von diesen per eMail oder telefonisch an die Mitarbeiter des CAM-Supports weitergeleitet. Der zuständige Supportmitarbeiter sammelt die erforderlichen Informationen und Daten beim Kunden und beantwortet die Anfrage anschließend per eMail oder telefonisch. Stellt sich eine Anfrage als Fehler im PP dar, so wird diese zusammen mit einer Problembeschreibung, den Softwaredaten und Kundeninformationen per eMail zur Verifizierung an den Programmierer weitergeleitet. Dieser übernimmt anschließend die vollständige Abwicklung der Anfrage für den Endkunden. Stellt sich ein Problem als Fehler in der CAM-Software dar, so wird dieser von den Supportmitarbeitern verifiziert und zusammen mit einer Problembeschreibung und den betroffenen Applikationsdateien per eMail an den Hersteller der CAM-Software gesendet. Eine informationstechnische Unterstützung der Prozessabwicklung in Form eines Managementsystems auf Seiten des VAR existiert nicht.

Informationsmanagement

Die Kontaktdaten der Kunden werden im Adressverzeichnis von MS-Outlook, in der Seriennummern-Datenbank des Extranets und parallel dazu in einer in Eigenregie programmierten Access-Datenbank des CAM-Teams gehalten. In dieser Access-Datenbank sind auch alle detaillierten Informationen zur Installationsbasis, ausgelieferten PP, Seriennummern und Softwarelizenzen aufgeführt.

Auf Informationen zu den in den CAD-CAM-Bundles enthaltenen SolidWorks-Lizenzen haben die Mitarbeiter des CAM-Competence-Centers durch das Siebel-Partnerportal von SolidWorks keinen Zugriff. Diese Lizenzen werden vom Hersteller der CAM-Software bei SolidWorks erworben und an die VAR zusammen mit der CAM-Software weiterverkauft. Somit sind diese SolidWorks-Lizenzen dem Hersteller der CAM-Software und nicht dem VAR als Händler zugeordnet. Für die VAR von SolidWorks sind jedoch aus Gründen des Wettbewerbsschutzes nur die Informationen ihrer eigenen Kunden über das Partnerportal zugänglich, nicht aber jene die von anderen VAR betreut werden. Da die SolidWorks-Lizenzen jedoch vom Hersteller der CAM-Software eingekauft und als Teil eines CAD-CAM-Paketes über die Händler an die Kunden vertrieben werden, werden dem Endverkäufer die SolidWorks-Lizenzen seiner CAM-Kunden im Partnerportal von SolidWorks nicht angezeigt. Da der Endverkäufer aber sowohl den Support für die CAM- wie auch für die gleichzeitig mitveräußerte CAD-Software übernehmen muss, macht dies das Anlegen von

Serviceanfragen zur CAD-Software im Partnerportal unmöglich.

Wissensmanagement

Innerhalb des CAM-Competence-Centers existiert derzeit kein Wissensmanagement.

Datenhaltung

Alle Kundendaten und PP werden lokal auf den Arbeitsstationen der Mitarbeiter des CAM-Supports innerhalb einer Ordnerstruktur mit Firmennamen und Datum abgelegt. Die angesprochene Access-Datenbank mit den Lizenzinformationen der Kunden liegt auf einem Server und wird von allen Mitarbeitern des CAM-Supports gemeinsam genutzt und gepflegt.

2.3.4. Beispiel kundenspezifische Funktionserweiterung der Software

Hier wird beispielhaft eine kundenspezifische Funktionserweiterung in SolidWorks mittels einer Makroprogrammierung analysiert (vgl. Abbildung 2.3-4).

Unternehmen

Die Automatisierung von Standardprozessen bei der mechanischen Konstruktion mit Hilfe von CAD-Anwendungen ist eine etablierte Methode zur Erhöhung der Effizienz. Diese spielt heute eine zunehmende Rolle, vor allem dort, wo große Produktivitätsschübe durch die Einführung eines 3D-CAD-Systems eine Weile zurückliegen, das System solide implementiert ist und Verbesserungen mit den vorhandenen Funktionen bzw. durch Konfiguration der Software kaum noch zu erreichen sind. Hier können durch Programmierung von spezifischen Makros fehlende Funktionen geschaffen oder auch ganze Abläufe automatisiert werden.

Software

Die Makros werden in der Regel als Add-Ins ausgeführt und in VBA oder C++ programmiert. Die Funktionsumfänge sind sehr unterschiedlich und richten sich zum einen nach den Kundenanforderungen und zum anderen nach dem zur Verfügung stehenden Kostenrahmen. Zu den häufigsten Aufgaben gehört die Generierung komplexer Geometrien z.B. optische Linsen anhand vorgegebener Parameter oder die Erzeugung von Layouts.

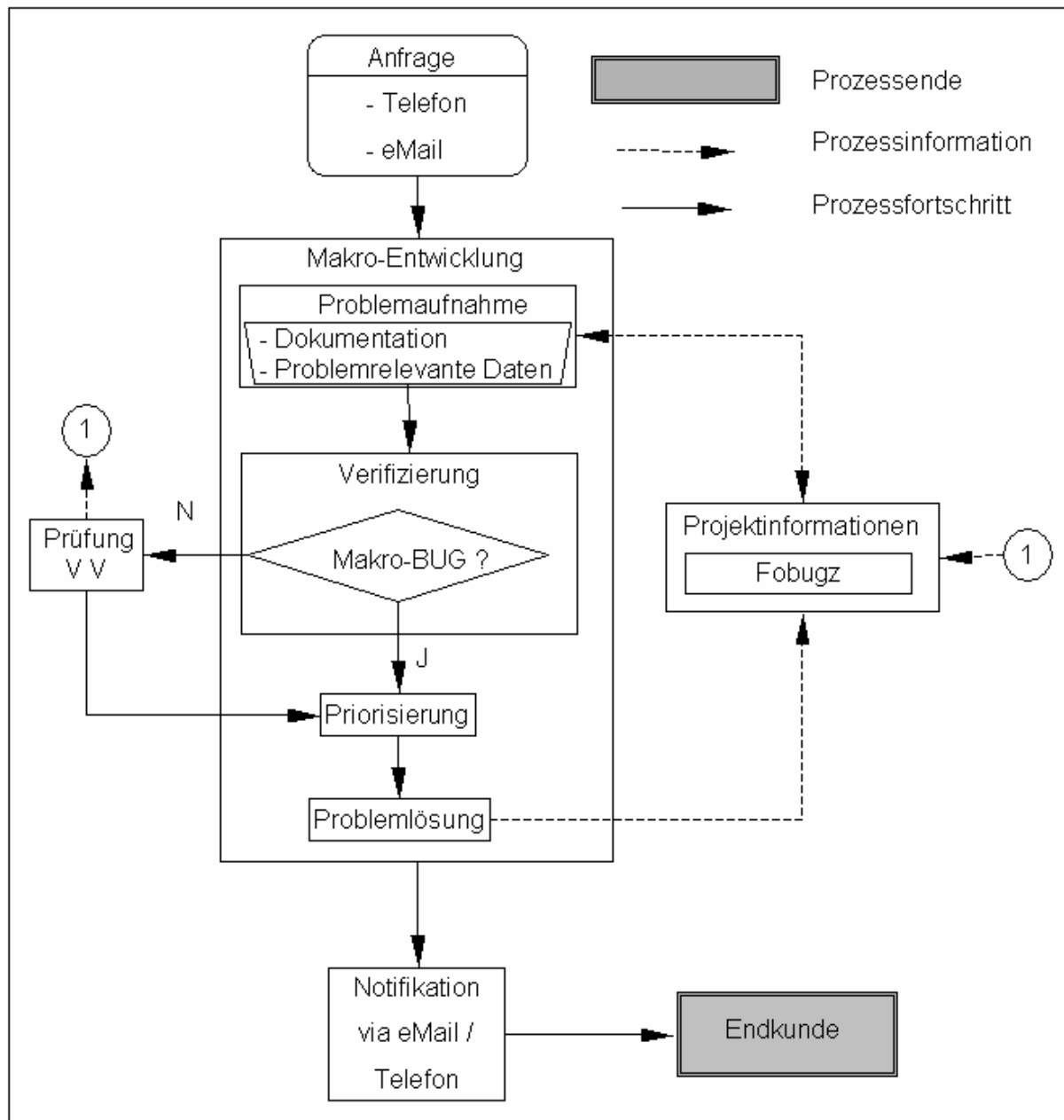


Abbildung 2.3-4: CAD-Anwendung (Fallbeispiel 4)

Außerdem können wiederkehrende Kombinationen von Funktionsaufrufen automatisiert werden, deren manuelle Ausführung entsprechend zeitaufwändig ist.

Gemeinsam ist allen Programmen, dass diese nur für einen einzigen Kunden erstellt werden und dass im Gegensatz zu den vorher genannten Fällen der Quellcode in der Regel Bestandteil des Lieferumfangs ist und somit in den Besitz des Kunden übergeht. In diesem Fall ist eine weitere Anforderung, dass die Gestaltung des Codes mit dem Kunden abgeglichen werden muss, damit dieser das Programm gegebenenfalls eigenständig weiterentwickeln und pflegen kann.

Aufgabenverteilung

In der Regel wird das gesamte Projekt ohne Einbindung des Supports abgewickelt. Auch die Betatests werden ausschließlich von den Entwicklern mit Bezug auf die im Pflichtenheft mit dem Kunden vereinbarten Funktionalitäten durchgeführt. Der Kunde stellt hierzu für gewöhnlich Referenzmodelle in Form von 3D-CAD-Daten zur Verfügung, anhand derer später die Funktionalität überprüft und die Software abgenommen wird.

Prozessmanagement und Qualitätssicherung

Der Auftrag zur Programmierung eines Makros oder Add-Ins geschieht über verschiedene Wege, meist über den Vertrieb oder die Hotline des Supports. Dieser stimmt auch mit dem Kunden zusammen ein Pflichtenheft ab. Bei Auftragserteilung wird ein Projekt im Entwicklungsmanagementsystem Fogbugz angelegt, in dem sämtliche Entwicklungsschritte dokumentiert werden. Betrachtet der Entwickler alle Anforderungen des Pflichtenhefts im Zusammenhang mit den Referenzdaten erfüllt, gibt er den Code frei und übergibt diesen an den Kunden. Ist der Kunde mit der Umsetzung einverstanden, nimmt dieser den Code ab und das Projekt ist abgeschlossen.

Oftmals kommen jedoch nach der Implementierung der im Pflichtenheft festgelegten Funktionen weitere Wünsche hinzu, oder der Kunde ist mit der Art und Weise der Implementierung unzufrieden, so dass gegebenenfalls mehrere Freigabezyklen durchlaufen werden müssen, bis ein Projekt endgültig abgeschlossen werden kann. Verbesserungsvorschläge und Fehlerreports werden vom Kunden per eMail oder telefonisch direkt an die Entwicklung gemeldet und im Dialog mit diesem Team implementiert bzw. behoben. Sämtliche VV und Bugs werden ebenfalls im Entwicklungsmanagementsystem Fogbugz dokumentiert und darin gepflegt. Die Sicherung der unterschiedlichen Versionen des Quellcodes erfolgt wiederum in MS-Visual SourceSafe.

Informationsmanagement

Die Möglichkeiten des Zugriffs auf Kundeninformationen und deren Verwaltung deckt sich mit der in Kapitel 2.3.1 beschriebenen Zugangswegen und Methoden. Allerdings werden die Informationen zu den kundenspezifischen Add-Ins oder Makros nur innerhalb des Entwicklungsmanagementsystems gehalten und sind somit für das übrige Supportteam nicht einsehbar.

Wissensmanagement

Für die entwickelten Add-Ins und Makros existiert keinerlei Wissensmanagement.

Datenhaltung

Sämtliche entwicklungsrelevanten Daten werden in Fogbugz bzw. SourceSafe gespeichert. Referenz- bzw. Applikationsdateien der Kunden werden lokal innerhalb eines Dateisystems des projektverantwortlichen Entwicklers gespeichert.

2.4. Beurteilung der Defizite der analysierten Praxisbeispiele

Bei der Beurteilung der Defizite der analysierten Praxisbeispiele fällt auf, dass sich die Szenarien in ihren Abläufen stark ähneln, wenn auch die Komplexität der Prozesse unterschiedlich ausfällt. Im Anschluss werden die untersuchten Aspekte wie folgt zusammengefasst.

2.4.1. Bearbeitete Prozesse

Während der Bestandsaufnahme wurde das Prozessmanagement im Kontext zur Qualitätssicherung analysiert, was aufgrund der gegenseitigen Verflechtung beider Prozesse auch sinnvoll erscheint. Um jedoch ein möglichst differenziertes Bild der Gesamtsituation entwerfen zu können, werden in der abschließenden Zusammenfassung beide Punkte getrennt dargestellt.

Die Prozessabläufe der unterschiedlichen Praxisbeispiele präsentieren sich oberflächlich betrachtet als recht homogen. Im Detail werden jedoch gravierende Unterschiede z.B. bezüglich des Prozessmanagements deutlich. Die grundlegende Prozesskette Anfrage-Verifizierung-Problemlösung findet sich zwar überall wieder, jedoch ist der Umfang der in den verschiedenen Beispielen involvierten Abteilungen, der Prozessschritte und die Anzahl der verwendeten Tools höchst unterschiedlich. So findet sich im Beispiel der Entwicklung der Standalone-Anwendung SolidWorks auf der Ebene des 2nd-Level-Supports eine umfangreiche Unterstützung dieser Prozesskette durch das in das Siebel-Partnerportal integrierte Prozessmanagement, diese endet jedoch mit der Schnittstelle zum 1st-Level-Support: Ein Großteil der relevanten Daten und die Korrespondenz mit dem Endkunden werden hier parallel gehalten. Die Notwendigkeit hierzu resultiert zum einen aus der

sprachlichen Barriere – die Korrespondenz mit dem Kunden erfolgt in der jeweiligen Landessprache, im vorgestellten Beispiel auf Deutsch, während über das Partnerportal von SolidWorks mit dem 2nd-Level-Support stets auf Englisch kommuniziert wird. Auf der anderen Seite ergibt sich die parallele Datenhaltung aus der Notwendigkeit, dass der 1st-Level-Support die Applikationsdateien des Kunden für die Untersuchung durch den 2nd-Level-Support aufbereitet, die Originaldaten jedoch bis zum Abschluss des CALLs beim VAR gespeichert werden müssen.

Prozessübergreifend existieren keine einheitlichen Workflows¹⁶. So wird bei der kundenspezifischen Funktionserweiterung der Software der Support nur im Rahmen der Auftragsvergabe eingebunden, im weiteren Verlauf der auftragsspezifischen Kundenbeziehungen hingegen ist dieser nicht involviert, was in erster Linie einen Bruch im Informationsfluss darstellt: Geht eine Support-Anfrage bezüglich der gelieferten Software vom Endkunden ein, so ist dem Support oftmals schon die bloße Existenz dieses Produktes unbekannt, Hilfestellung kann deshalb ausschließlich durch den projektverantwortlichen Entwickler geleistet werden.

Die Abarbeitung von Hotlineanfragen verläuft auf der Ebene des 2nd-Level-Supports der Firma SolidWorks aufgrund der umfangreichen informationstechnologischen Unterstützung des eingesetzten CALL-Management-Systems sehr effektiv. Vereinfacht wird die Disposition der Anfragen aus dem Vertriebskanal dadurch, dass diese ausschließlich über das Webportal gestellt werden. Auf der Ebene des 1st-Level-Supports - hier im Beispiel von SolidWorks und der Varbox - besteht durch das Vorhandensein der unterschiedlichen Anfragekanäle jedoch stets das Risiko, dass die selbe Anfrage von mehreren Support-Mitarbeitern gleichzeitig abgearbeitet wird, da eine Echtzeiterfassung der CALLs durch das träge und unergonomische Handling des Helpdesk als Erweiterung von MS-Outlook nicht möglich ist.

Dies mag aus der Sicht des Endkunden zunächst weniger problematisch erscheinen, jedoch machen sich gerade in Zeiten hoher Auslastung der Hotline, die aus solchen Doppelbelegungen resultierenden Produktivitätseinbußen in der CALL-Abwicklung letztendlich auch bei den Kunden in Form längerer Antwortzeiten negativ bemerkbar. Die Trägheit des Systems und der hohe Aufwand zur CALL-Erfassung innerhalb des Helpdesk verleiten zudem die Support-Mitarbeiter gerade beim zeitkritischen Hotline-Support zu einer lückenhaften Erfassung, welche im Einzelfall sogar vollends unterbleibt.

Der Status eines CALL oder SPR ist innerhalb des Partnerportals von SolidWorks gut nach zu verfolgen, während aufgrund der eingeschränkten Suchroutinen innerhalb von MS-Outlook das Nachvollziehen eines CALL im Helpdesk des VAR sehr zeitraubend ist. Für alle

¹⁶ Workflow = engl. Arbeitsablauf, vordefinierte Abfolge von Aktivitäten

vorgestellten Beispiele ist die Nachverfolgbarkeit von VV sowohl aus Sicht der Kunden als auch aus der Sicht des Supports verbesserungswürdig. Hierzu ist zu bemerken, dass einzig die Entwicklung der Firmen SolidWorks und SolidCAM eindeutige Kennungen für die eingereichten VV vergeben, was aber letztlich nur einer Empfangsbestätigung gleichkommt. Bei den VV für die PP-Programmierung und den kundenspezifischen Softwareerweiterungen sind diese Punkte als weniger kritisch zu bewerten, da hier jegliche Leistungen der Entwicklungsabteilung kostenpflichtig sind und von daher eine Ausführung in beiderseitigem Interesse zeitnah erfolgt. Für alle anderen Fälle kann aus den genannten Gründen pauschal formuliert werden, dass verbindliche Aussagen zur Implementierung der gewünschten Funktionalitäten stets unterbleiben.

Letztendlich bleibt festzustellen, dass nur im Falle der Softwareentwicklung von SolidWorks Ansätze zu einer durchgängigen Verkettung aller relevanten Bereiche im Sinne eines SLM vorhanden sind, die Kommunikation zwischen VAR und Endkunden jedoch nicht in diesen Prozess integriert ist.

2.4.2. Qualitätssicherung

Obschon der Begriff im Rahmen der vorausgegangenen Betrachtungen bereits mehrfach Verwendung fand, sollen hier im Vorfeld zur Beurteilung der Defizite im Bereich der Qualitätssicherung kurz der Begriff der *Qualität* und des *Qualitätsmanagements* definiert werden:

Qualitätsbegriff nach DIN ISO 9000

Nach der Norm EN ISO 9000:2005 ist *Qualität der Grad, in dem ein Satz inhärenter Merkmale und Anforderungen erfüllt wird*. Entscheidend für die Qualität eines Produktes nach ISO 9000 ist nicht sein Preis oder die Qualität der verwendeten Materialien. Entscheidend ist, dass die gestellten Anforderungen an das Produkt erfüllt werden. Dabei müssen sowohl die objektiv messbaren Eigenschaften und Vorgaben des Produzenten als auch die subjektiven Erwartungen der Kunden erfüllt werden. Nach der IEC 2371 ist Qualität die *Übereinstimmung zwischen den festgestellten Eigenschaften und den vorher festgelegten Forderungen einer Betrachtungseinheit*. Der Begriff Qualität ist grundsätzlich für sämtliche Arten von Produkten und Dienstleistungen anwendbar [16].

Definition des Qualitätsmanagements

Der Begriff Qualitätsmanagement (QM) bezeichnet einen Teilbereich des funktionalen Managements. Das Ziel ist die Optimierung von Arbeitsabläufen oder von

Produktionsprozessen unter der Berücksichtigung von materiellen und zeitlichen Kontingenten sowie der Qualitätserhalt von Produkten bzw. Dienstleistungen und deren Weiterentwicklung. Hierbei von Belang sind etwa die Optimierung von Kommunikationsstrukturen, professionelle Lösungsstrategien, die Erhaltung oder Steigerung der Zufriedenheit von Kunden oder Klienten sowie der Motivation der Belegschaft, die Standardisierungen bestimmter Handlungs- und Arbeitsprozesse, Normen für Produkte oder Leistungen, Dokumentationen, Berufliche Weiterbildung, Ausstattung und Gestaltung von Arbeitsräumen [16].

Bei der Betrachtung der unterschiedlichen Beispiele wird deutlich, dass die Qualitätssicherung in Bezug auf die Softwareprodukte recht unterschiedlich gehandhabt wird, obwohl in allen Fällen dieselbe Zielsetzung angestrebt wird. Bei SolidWorks wird die Qualität der entwickelten Software zunächst automatisch geprüft und anschließend in fest terminierten Betatests verifiziert. Die Qualitätssicherung für die freigegebenen Releases erfolgt anschließend zentral über das VAR-Portal und wird dabei den Anforderungen an Nachverfolgbarkeit und Prozessmanagement im Sinne der Qualitätssicherung weitestgehend gerecht. Die qualitätsorientierte Funktionsüberprüfung bei der Entwicklung der Varbox hingegen ist rudimentär und lückenhaft, da weder eine automatische Überprüfung der Software-Funktionen erfolgt, noch eine feste Integration von Supporttechnikern zur manuellen Überprüfung vorgesehen ist. In diesem Kontext ist auch der fehlende Zugang der Supporttechniker zum Projektmanagementsystem der Entwicklung zu bemängeln, so erfolgt die gesamte Dokumentation letztendlich mittels unabhängiger Textdateien. Der Umfang der Qualitätssicherung ist somit nicht klar definiert und letztendlich auf dem individuellen Engagement einzelner Supporttechniker begründet.

Im Falle der kundenspezifischen Funktionserweiterung der CAD-Software fehlt eine Qualitätssicherung durch Dritte gänzlich, da hier die Funktionen des Entwicklers und des Testers in einer Person vereint sind. Dieser Zustand wird lediglich dadurch etwas entschärft, dass das Funktionsspektrum der zu entwickelnden Programme für gewöhnlich gering ist und sowohl die Funktionsüberprüfung als auch die Endabnahme unter zuvor fest definierten Rahmenbedingungen bzw. anhand fest definierter Datensätze erfolgt. Trotzdem bleibt festzustellen, dass eine fehlende Trennung von Entwicklung und Testumgebung den Anforderungen an eine Qualitätssicherung kaum gerecht werden kann und somit das Risiko für umfangreiche, unendgeldliche Nacharbeiten im Anschluss an die Auslieferung sehr hoch ist.

In Abgrenzung zu den übrigen Beispielen gestaltet sich die Qualitätssicherung bei der kundenspezifischen Softwareanpassung im Rahmen der PP-Programmierung deutlich anders. Dies liegt im Wesentlichen in der Natur des PP begründet, da dieser letztendlich nur eine definierte Menge von Programmbefehlen und Parametern in eine andere

Programmiersprache übersetzt. Zwar tritt auch hier wieder die unerwünschte Konstellation der Vereinigung von Entwickler und Tester in einer Person auf, diese ist jedoch im Gegensatz zu den übrigen Fällen weniger kritisch zu bewerten, da das Programm beim Kunden vor Ort auf einer zuvor spezifizierten Werkzeugmaschine in Betrieb genommen wird. Das Programm läuft dabei in einer fixen Peripherie von Systemkomponenten und sämtliche nachträgliche Änderungen am Leistungsspektrum des PP sind für den Kunden kostenpflichtig. All diese Faktoren lassen die fehlende Integration der Qualitätssicherung in ein einheitliches SLM-Konzept im Falle der PP-Programmierung weniger kritisch erscheinen als in den übrigen Fällen.

Im Unterschied zu den Softwareprodukten existieren derzeit im Bezug auf die Qualität der Supportvorgänge selbst - also der durch die Ausübung des Supports erbrachten Dienstleistungen - keinerlei Mechanismen zu deren Sicherung. Die Überprüfung von Kundenzufriedenheit oder die Bewertung der Effizienz des CALL-Handlings fehlt ebenso, wie eine detaillierte Definition des Leistungsumfangs und die Überprüfung der Einhaltung der im Rahmen des CALL-Handlings ablaufenden Prozesse. Dabei scheitert die Durchführung eines QM bei dieser Art von Dienstleistung bereits an der fehlenden Vorgabe von Qualitätsmerkmalen bezüglich des Umfangs und zeitlichen Rahmens dieser Maßnahme. Eine Definition der zu durchlaufenden Prozesse fehlt ebenfalls. Diese Umstände schließen somit eine wie auch immer geartete Qualitätssicherung für die Dienstleistung „*Softwaresupport*“ gemäß der zuvor formulierten Definition von QM aus. Der beschriebene Zustand bedingt hingegen die Erbringung einer vereinbarten Leistung aufgrund einer rein subjektiven Definition von Support. Damit sind Differenzen aufgrund unterschiedlicher Auslegungen dieser Definitionen sowohl seitens des Systemhauses als auch des Endkunden vorprogrammiert.

Analog zu den Defiziten der zu bearbeiteten Prozesse (vgl. Kapitel 2.4.1) muss auch bei der Qualitätssicherung der Schluss gezogen werden, dass lediglich im Falle der Entwicklung von SolidWorks eine hinreichende Integration in ein SLM-System gegeben ist.

2.4.3. Eingesetzte Tools

Als Software zur Verwaltung der unterschiedlichen Daten und für das Prozessmanagement kommen bei den verschiedenen Beispielen die unterschiedlichsten Tools (Software-Applikationen), teilweise in auch Kombination, zum Einsatz. Diese Software-Systeme lassen sich wie folgt charakterisieren:

Siebel-CALL-Center

Das Siebel-System zeichnet sich durch ein umfangreiches, datenbankbasiertes CALL-Handling-Modul aus. Die Applikation kann auf die meisten gängigen Datenbanksysteme aufgesetzt und an die unterschiedlichsten Anforderungen angepasst werden. Das Tool verfügt über ein Webportal und eine umfangreiche Rechteverwaltung. Weiterhin gestattet dieses Softwaresystem die Definition von Workflows im Rahmen eines Prozessmanagements. Im Rahmen der Möglichkeiten des zugrunde liegenden Datenbank-Managementsystems können auch Verknüpfungen zu anderen Systemen, wie z.B. den Entwicklungsdatenbanken, CRM-, PLM- oder auch SLM-Systemen eines Unternehmens erfolgen.

MS Outlook

MS-Outlook ist eine umfangreiche eMail-Client-Software von Microsoft und Teil des Microsoft Office-Paketes. Es unterstützt die Verwaltung mehrerer eMail-Konten über eine Oberfläche. Diese können über POP3¹⁷, IMAP4¹⁸ oder Exchange-RPC¹⁹ angesteuert werden. Neben der eMail-Funktion ist die Terminverwaltung die im betrieblichen Umfeld am intensivsten genutzte Funktion von Outlook. Über einen Exchange-Server können automatisch Besprechungen und andere Termine mit mehreren Teilnehmern geplant werden, in dem der Outlook-Client die Terminkalender aller Teilnehmer vergleicht und mögliche Terminüberschneidungen anzeigt. Gleichzeitig können unter Outlook Ressourcen (z. B. Räume oder Geräte) reserviert werden. PDA²⁰s, Pocket PCs oder auch Smartphones lassen sich mittels des Hilfsprogramms ActiveSync einfach und komfortabel mit Outlook synchronisieren, so dass es dem Benutzer ermöglicht wird, auch unterwegs Termine einzutragen bzw. sich an Termine erinnern zu lassen. Outlook lässt sich mittels VBA und OLE automatisieren und erweitern [16].

MS-Access

Microsoft Access ist ein Datenbanksystem, das die Verwaltung von Daten und die Entwicklung von Datenbankanwendungen ermöglicht. MS-Access ist Bestandteil des Microsoft Office Pakets von Microsoft und unterstützt nur SQL-92. MS Access existiert

¹⁷ POP3 – Post Office Protocol Vers. 3, Übertragungsprotokoll für eMails

¹⁸ IMAP4 – Internet Message Access Protocol Vers. 4, Protokoll für die Verwaltung von eMail-Konten

¹⁹ RPC - Remote Procedure Call, Prozedur-Fernaufruf

²⁰ PDA – Personal Digital Assistant

derzeit in der Version MS-Access 2003 und wird im Gegensatz zu anderen Office-Programmen von Microsoft nur für Windows angeboten. Access speichert alle Daten einer Datenbank in eine einzelne Datei des eigenen mdb-Dateiformates ab, dies schließt sowohl Elemente der Oberfläche als auch die Datenbanktabellen ein. Es ist allerdings möglich, die Tabellendefinitionen und den Datenbestand sowie die Oberfläche in verschiedenen Dateien zu halten (sog. front- bzw. backend). Beim Einbinden bzw. Verknüpfen von externen Tabellen können auch Access-fremde Formate über ODBC²¹ angesprochen werden.

Im Gegensatz zu früheren, PC-basierenden Datenbanksystemen unterstützt MS-Access das relationale Datenbankmodell. Um von extern auf Access-Datenbanken zuzugreifen, eignet sich am besten die ebenfalls von Microsoft entwickelte ODBC-API. Ferner kann auf Access-Datenbanken auch von anderen Programmiersprachen, z.B. Delphi, VB, etc. durch den Einsatz von ADO²² zugegriffen werden. Dazu muss Access nicht installiert sein. Ab Windows 2000 ist ADO ein Bestandteil des Betriebssystems.[16]

Extranet

Das Extranet ist eine auf einer SQL-Datenbank basierende Informationsquelle des hier analysierten VAR. Der Zugriff erfolgt per Web-Access über einen Webbrowser. Mit Hilfe des Extranet kann jeder Mitarbeiter des VAR die darin enthaltenen Informationen von einem beliebigen Ort aus über das Internet mit Hilfe von Benutzerkennung und Passwort abrufen. Die Datenbank enthält eine Reihe von spezifischen Lizenzinformationen zu den CAD- und CAM-Kunden des VAR. Die Datenbank verfügt über eine integrierte Benutzerverwaltung mit fein differenzierbaren Zugriffsrechten. Somit kann der Zugriff auf bestimmte Bereiche oder Teile davon präzise gesteuert werden.

Fogbugz

Fogbugz von Fog Creek Software ist ein webbasiertes Projektmanagement-System für Software-Entwicklungs-Teams. Der Zugriff erfolgt über den Internet-Explorer oder MS-Outlook.

Visual SourceSafe

MS Visual SourceSafe wurde von Microsoft zur Versions- und Revisionsverwaltung in Softwareprojekten entwickelt und verfügt dadurch auch über eine gute Anbindung an andere

²¹ ODBC - Open Database Connectivity, standardisierte, SQL-basierende Datenbank-Schnittstelle

²² ADO - Active Data Objects, Windows-Komponente zum Aufbau von Datenbankverbindungen

Microsoft-Entwicklungstools wie z.B. MS Visual Studio. Die Ablage der Versionen erfolgt in einer Datenbank, so dass jeder beliebige Entwicklungsstand reproduziert werden kann. Neben dem Ein- und Auschecken von Dokumenten und Quellcodes ist eine Zugriffsteuerung zur Sperrung ausgecheckter Dateien implementiert. Eine benutzerspezifische Zugriffsteuerung fehlt jedoch ebenso wie die Möglichkeit, Workflows zur Lenkung des Entwicklungsprozesses zu implementieren. Verlinkungen zwischen unterschiedlichen Dateien sind ebenfalls nicht vorgesehen.

2.4.4. Wissensdistribution

Die Distribution von Wissen im Bezug auf alle Anwendungsbereiche der untersuchten Software-Produkte wird in unterschiedlicher Form realisiert. Die Firma SolidWorks bietet Ihren Vertriebspartnern und Endkunden eine heterogene Menge von Quellen für das Anwenderwissen. Hier ist zum einen die unvollständige und nur zum Teil aktuelle FAQ-Datenbank zu nennen, zum anderen die unregelmäßigen Newsletter zu aktuellen Problematiken. Daneben beinhaltet die SolidWorks-Software eine umfangreiche Online-Hilfe. Nicht zuletzt obliegt es den VAR den Endkunden Anwenderwissen im Rahmen von Seminaren oder über die Hotline zu vermitteln. Die VAR organisieren ihr Wissen teilweise in eigenen Datenbanken und versenden daraus turnusmäßig willkürlich Auszüge hieraus an Ihre Kunden. Internes Wissen wird im Wesentlichen indirekt über den Helpdesk dokumentiert. Im Großen und Ganzen sind hier lediglich Ansätze zu einer umfassenden, einheitlichen Datenbasis zu erkennen, die sowohl dem Softwarehersteller als auch den Vertriebspartnern und nicht zuletzt den Endkunden einen umfangreichen Zugang zu Anwender- und Expertenwissen ermöglicht.

Für die Varbox existieren lediglich vereinzelte und größtenteils veralteten FAQs. So sind auch die Supportmitarbeiter des VAR bei Problemen auf die Online-Hilfe der Varbox angewiesen, welche naturgemäß keine Problemfälle abbildet, sondern immer von einer idealen Implementierung ausgeht. Tiefer greifendes Wissen kann letztendlich nur durch direkten Kontakt zu den Entwicklern erworben werden, was bei akuten Problemen allein durch die Abwesenheit des Entwicklungspersonals zu einer unüberwindlichen Hürde werden kann. Daneben macht eine solche Konstellation den Support und damit auch den Vertrieb der Varbox durch andere Vertriebspartner von SolidWorks problematisch und somit unattraktiv.

In den Bereichen der kundenspezifischen Funktionserweiterung und –Anpassung der CAD-Software begünstigt die ausschließliche Abwicklung der Projekte durch die Software-Entwicklung das völlige Fehlen von dokumentiertem Wissen. Neben der Verkomplizierung des laufenden Supports bedingt dies ein nicht unerhebliches unternehmerisches Risiko für

das Systemhaus, da im Falle eines Personalwechsels in der Entwicklung die Kundenunterstützung in Bezug auf das gelieferte Produkt nahezu unmöglich wird. So finden sich im Punkt der Wissensdistribution in allen Beispielen mehr oder minder ausgeprägte Defizite im Bezug auf einen umfassenden Ansatz wie er dem SLM zugrunde liegt.

2.4.5. Informationsmanagement

Im Bereich des Informationsmanagements treten bei den unterschiedlichen Beispielen zum Teil erhebliche Defizite zu Tage. Für die SolidWorks-Entwicklung ist die Organisation von Kundendaten, Lizenzinformationen und anfragespezifischen Dokumentationen, sowie den 1st-Level-Support und den 2nd-Level-Support in sofern ausreichend, als dass sämtliche Informationen zentral über die Siebel-Datenbank gepflegt und von allen Beteiligten abgerufen werden können. In den übrigen Fällen ist der Zugriff auf diese Informationen für den Support, vor allem durch die parallele Datenhaltung kompliziert und zeitraubend. Der Wert dieser Informationen wird zudem durch inkonsistente Datenhaltung signifikant gemindert. Dies wird bereits bei der Betrachtung des 1st-Level-Support für SolidWorks durch den VAR deutlich. Die Kundeninformationen werden zum einen im Extranet, zum anderen in MS-Outlook gehalten. Der Support hat zudem Zugriff auf die Kundeninformationen von SolidWorks in der Siebel-Datenbank. Der Vertrieb hält die Kundendaten in unabhängigen ACT!-Datenbanken. Der Vollständigkeit halber sei erwähnt, dass die Administration die Kundendaten für die Auftragsabwicklung ebenfalls in einer eigenen, unabhängigen Datenbank hält und pflegt.

Informationen zu Kundenanfragen werden vom Support zum einen in MS-Outlook, zum anderen im Siebel-Partnerportal hinterlegt. Insbesondere der mit dem Einpflegen der Anfragen in Outlook verbundene Aufwand führt häufig zu unvollständigen, wenig aussagekräftigen Eintragungen, zudem unterbleibt die Eintragung oftmals völlig. Eine einheitliche Plattform für Pflege und Zugriff auf sämtliche Kundeninformationen fehlt also vollends.

Die zuvor aufgeführten Defizite finden sich auch in den übrigen Beispielen in mehr oder minder ausgeprägter Form wieder. Bestimmte Informationen, z.B. aus Bereichen der kundenspezifischen Funktionserweiterung der CAD-Software, der Varbox-Programmierung oder erweiterte Lizenzinformationen zur Varbox sind für den Support aus unterschiedlichen Gründen nicht verfügbar – entweder weil dieser keinen Zugriff auf das Entwicklungsmanagementsystem hat, oder weil diese Informationen überhaupt nicht dokumentiert werden.

Das Beispiel der kundenspezifischen Softwareanpassung in Form von PP-Programmierung zeigt zwar die gemeinsame Nutzung einer Access-Datenbank durch Entwicklung und CAM-

Support, in welcher alle relevanten Informationen enthalten sind, diese ist jedoch von den anderen Datenbanken abgekoppelt, so dass die übrige Supportmannschaft keinen Zugriff darauf hat. Die dadurch fehlenden Zugriffsmöglichkeiten des CAD-Supports auf die notwendigen Kunden- und Lizenzinformationen verkomplizieren den gesamten Supportprozess bei Überschneidungen zwischen CAD- und CAM-Support, welche durch den hohen Integrationsgrad der CAM-Software in die SolidWorks-Anwendung häufig vorkommen. Nicht zuletzt ist durch die Isolierte Datenhaltung auch hier stets die Gefahr von Redundanzen und inkonsistenten Daten im Bezug auf die übrigen Datenbanken gegeben.

2.5. Zusammenfassung der Defizite

Vor der Formulierung der Forderungen an ein SSMS werden die gegenwärtig zu verzeichnenden Defizite zusammengefasst. Hierbei fließen nun nicht nur die Defizite der analysierten Praxisbeispiele, sondern auch die Erfahrungen des Verfassers aus seiner Tätigkeit im Softwaresupport sowie das Ergebnis von zahlreichen Begegnungen und Diskussionen mit Softwareentwicklern, Vertriebsbeauftragten und Anwendern ein.

Für den Software-Support existieren derzeit eine Reihe von Tools, die ähnlich dem hier vorgestellten Siebel CALL-Center aus dem CRM hervorgegangen sind, z.B. MySAP CRM von SAP. Obwohl die Analyse gezeigt hat, dass das Siebel CALL-Center den alltäglichen Anforderungen im Software-Support gerecht werden kann, so ergeben sich doch im vorliegenden Fall zwei wesentliche Nachteile:

- Die Anschaffung, Konfiguration und Programmierung des Systems ist für kleinere Unternehmen, wie sie im CAD-Vertrieb und- Support anzutreffen sind, aufgrund der damit verbundenen hohen Kosten nicht wirtschaftlich.
- Die Vertriebsstrukturen im CAD-Markt bedingen häufig eine Trennung von Hersteller und VAR und somit auch eine räumliche und organisatorische Trennung von 1st- und 2nd-Level-Support. Eine globale, organisationsübergreifende Lösung ist dadurch mit den verfügbaren Tools in der Regel nicht realisierbar.

Die Vielzahl der verwendeten Tools und Datenbanksysteme in den analysierten Beispielen führt zwangsläufig zu einer inkonsistenten Datenbasis und redundanter Datenerfassung im Systemhaus. Fehlende Workflows erhöhen zusätzlich den Aufwand und die Fehleranfälligkeit bei der Erfassung und Abwicklung einzelner CALLs. Der Support wird hierdurch wesentlich in seinen Fähigkeiten beschränkt, zeitnah und effizient auf Kundenanfragen reagieren zu können. Dadurch ergeben sich nicht nur Mehrkosten für das

Systemhaus, sondern auch die Gefahr einer latenten Unzufriedenheit beim Endkunden durch verlängerte Antwortzeiten für die jeweiligen Anfragen. Die unzureichende oder fehlende Nachverfolgbarkeit des Bearbeitungsstandes eines jeden CALLs innerhalb des Supportmanagements des Systemhauses machen es für den Support schwierig, dem Endkunden gegenüber verbindliche Aussagen über den für die Lösung eines Problems benötigten Zeitraum zu treffen. Da für VVs in keinem Fall eine Aussage über den Zeitpunkt von deren Umsetzung getroffen werden kann, tritt dieses Problem hier noch verschärft auf.

Im Rahmen der ständig wachsenden Komplexität von Konstruktionsaufgaben und damit auch der CAE-Prozesskette bei gleichzeitig wachsendem Zeit- und Kostendruck durch den globalen Wettbewerb auf das produzierende Gewerbe, muss sich auch der Software-Support diesen Anforderungen stellen und seinerseits effizientere Methoden zur Unterstützung seiner Kunden entwickeln, um so letztendlich dem Kunden möglichst zeitnah Lösungen für sämtliche innerhalb dieser Prozesskette auftretende Softwareprobleme zu liefern.

Die Betrachtung der den Support umfassenden Bereiche offenbart eine Vielzahl von Verbesserungspotentialen sowohl innerhalb der einzelnen Betätigungsfelder als auch bereichsübergreifend in Bezug auf deren Verzahnung sowie auf die entsprechenden Prozesse und Workflows. So ergibt sich aus der Lösung eines Problems nicht nur die für die Fortführung des dadurch unterbrochenen Produktentwicklungsprozesses notwendige Information für das Kundenunternehmen sondern auch eine für das Systemhaus eine wertvolle Erkenntnis. Diese kann für das Bewältigen künftiger, ähnlich gearteter Fragestellungen genutzt werden. Hinzu kommt die durch die vertrieblichen Aktivitäten eines Systemhauses ständig wachsende Anzahl von zu betreuenden Kundenarbeitsplätzen. Dies bedingt wiederum eine steigende Anzahl von Support-Mitarbeitern zur Betreuung dieser Kunden. Sowohl die Neu- als auch die Bestandskunden des Systemhauses setzen die CAD-Software und die dazugehörigen Applikationen mit individuellen, an die jeweiligen Produktanforderungen angepassten Arbeitsweisen ein. Allen Supportmitarbeitern muss deshalb die funktionale Bedeutung solcher Arbeitsweisen im Rahmen der Einarbeitung in die Supporttätigkeit möglichst effizient vermittelt werden. Außerdem muss für den Supporttechniker transparent gemacht werden, bei welchem Kunden diese Arbeitsweisen zum Einsatz kommen. Zudem besteht auch hier ein wirtschaftliches Risiko für das Systemhaus, durch Fluktuation innerhalb des Supports wertvolle Informationen zu Anwendungen und deren Implementierung zu verlieren und die sich dadurch öffnenden Kompetenzlücken nicht zeitnah schließen zu können.

Aus all diesen Punkten ergibt sich zwangsläufig der Bedarf für eine effektive Dokumentation und Distribution von anwendungs- und implementierungsspezifischem Wissen durch ein integriertes Wissensmanagementsystem. Außerdem müssen durch das Systemhaus Anreize zur gezielten Dokumentation von Wissen in Form von themenspezifischen Abhandlungen

geschaffen werden, obgleich dies nicht die vorrangige Aufgabe eines SSMS sondern eher der Unternehmensleitung sein muss. Da die Dokumentation von applikationsspezifischem Wissen aber ein existentielles Ziel eines Systemhauses sein muss, ist auch diesem Aspekt im Rahmen der Betrachtung des Software-Support-Managements ein angemessener Stellenwert einzuräumen. Neues Wissen, welches im Rahmen von Beratungstätigkeiten oder sonstigen, zu Implementierung und Anwendung der entsprechenden Applikationen gehörigen Tätigkeiten des Supports gewonnen wird, sollte schließlich auch in die durch Anwenderschulungen betriebene Wissensdistribution des Supports an die Endkunden einfließen. Dazu ist eine laufende Integration dieser Erkenntnisse in die Schulungs-Dokumentationen notwendig. Nicht zuletzt wird die Schaffung einer kundenseitigen Zugriffsebene auf ein Wissensmanagementsystem auch den Support in seiner täglichen Arbeit entlasten, da die Kunden innerhalb dieses Systems selber nach Problemlösungen suchen könnten und somit ein Teil der Anfragen an den Support gar nicht erst gestellt wird.

Angesichts allgegenwärtiger personeller Unterbesetzung und Ressourcenknappheit muss für die Gewichtung eines CALLs neben den inhaltlichen Aspekten und dem Grad der Beeinträchtigung des Konstruktionsablaufs, also des dadurch verursachten wirtschaftlichen Schadens für den Kunden, auch die wirtschaftliche Bedeutung dieses Kunden für das Systemhaus selbst eine Rolle spielen. So ist aus ökonomischer Sicht ein Kunde mit 50 Softwarelizenzen durch die dadurch generierten Lizenz- und Wartungsgebühren für das Systemhaus von größerer Bedeutung, als ein Kunde mit 2 Softwarelizenzen. In Zeiten hoher Auslastung einer Supportorganisation muss diesem Umstand durch priorisieren der Anfragen nicht nur gemäß dem negativen Einfluss auf die Produktivität des Kunden, sondern auch nach der Bedeutung dieses Kunden für das Systemhaus Rechnung getragen werden. Dies bedingt die Forderung nach der unmittelbaren Visualisierung dieses Aspektes für den Support bei der Beurteilung jeder Serviceanfrage.

Schließlich muss zur Bewertung der Abwicklung eines CALLs nicht nur die dazu benötigte Zeit sondern auch die Qualität der gelieferten Lösung betrachtet werden. Dazu zählt sowohl der zur Lösung zu betreibende Aufwand als auch dessen Effektivität. Die Qualität eines Lösungsansatzes z.B. in Bezug auf eine Änderung an einem 3D-Design steht also in engem Kontext mit der Effizienz, mit der eine solche Änderung gemäß der vom Support gelieferten Lösung durchgeführt werden kann. Daneben muss in diese Bewertung aber auch die Qualität der vom Support geführten und an den Kunden gelieferte Dokumentation für den gesamten Änderungsvorgang einfließen. Im Kontext des allgegenwärtigen Kosten- und Zeitdrucks ist es für einen Endkunden häufig nicht von primärer Bedeutung, *ob* eine Lösung für ein Problem geliefert werden kann, sondern *wann* diese Lösung verfügbar ist.

Die Beobachtung des CAE-Marktes lässt eine starke Tendenz zu integrierten Software-Systemen für die gesamte Prozesskette „Produktentwicklung“ erkennen, innerhalb derer die

eigentliche CAD-Software zwar einen bedeutenden Stellenwert einnimmt, aber in der auch andere Applikationen, wie Fertigungsintegration (CAM), Simulation (Mehrkörperdynamik, FEM, CFD²³) oder PDM zunehmend an Stellenwert gewinnen. Damit steigen in gleichem Maße auch die Anforderungen an den Software-Support eines jeden Anbieters derartiger Systemlösungen.

In den vergangenen Jahren hat das 3D-CAD für die Produktentwicklung unbestritten an Bedeutung gewonnen. Da gegenüber dem 2D-CAD ein wesentlich realistischeres, virtuelles Produktmodell geschaffen werden kann, können Konstruktionsfehler z.B. durch Kollisionsprüfungen oder 3D-Kinematik-Analysen, bereits am virtuellen Modell vermieden werden, wodurch eine deutliche Erhöhung der Produktivität durch Verkürzung der Entwicklungszyklen zu erreichen ist. Wenn das Ziel der Unterstützung der mechanischen Konstruktion durch 3D-CAD eine Produktivitätssteigerung durch eine Konzentration des Konstrukteurs auf die Umsetzung der eigentlichen Konstruktionsaufgabe mittels Bereitstellung eines realitätsnahen Produktmodells ist, so muss das Ziel des Software-Supports für das 3D-CAD und dessen Peripherie die Konzentration auf die Lösungsfindung für die in diesem Kontext auftretenden Probleme an sich, und nicht auf das Sammeln, Zusammenführen und Abgleichen von problem- und kundenrelevanten Informationen sein. Aufgrund dieser zeitaufwändige Vorgänge jedoch und der parallel dazu kontinuierlich wachsenden Komplexität der Prozesse rund um die Produktentwicklung mit Hilfe von 3D-CAD ist eine Verbesserung sämtlicher Prozesse seitens des Supports durch eine integrative, informationstechnische Lösung dringend geboten.

Bei den untersuchten Beispielen basiert die Einhaltung der notwendigen Prozessfolge während der Abarbeitung eines CALLs im Wesentlichen auf der Erfahrung des jeweiligen Support-Technikers, da diese gegenwärtig nicht durch Workflows innerhalb der IT-Infrastruktur des Systemhauses abgebildet wird, d.h. die zu durchlaufenden Prozesse bei der Bearbeitung von Anfragen verlaufen nach Schemata, welche gegenwärtig nur in Form von Erfahrungswerten vermittelt und angewandt werden. Definierte Prozessabläufe sind jedoch notwendiger Bestandteil der Problemlösung, da Ihre Einhaltung großen Einfluss auf den zeitlichen Verlauf und den Erfolg der Lösungssuche hat. Deshalb ergeben sich aus der momentanen Situation nicht nur Verzögerungen in der CALL-Abwicklung und bei der Lösungsfindung sondern zusätzlich auch erschwerte Bedingungen für die Einarbeitung neuer Supportmitarbeiter in einem expandierenden Betätigungsfeld. Somit muss bei der Beurteilung der Defizite der gegenwärtigen Praxis sowohl die Vernetzung der unterschiedlichen Datenbasen als auch die Steuerung des Zugriffs auf und des Einpflegens von Daten durch definierte Workflows im Rahmen eines umfassenden Prozessmanagements

²³ CFD – Computational Fluid Dynamics

betrachtet werden.

Im Rahmen der derzeitigen Lösung ist seitens des Supports nur bei einem direkten Anschluss an das Unternehmensnetzwerk des Systemhauses ein Zugriff auf alle kundenrelevanten Daten möglich. Da jedoch häufig Beratungstermine bei Kunden vor Ort stattfinden, ist eine umfassender Zugriff auf sämtliche Informationen, z.B. mit Hilfe des Internets wünschenswert, würden sich doch dadurch eventuelle, im Rahmen der Beratungstätigkeit auftretende Problemstellungen in der selben Art und Weise bearbeiten lassen, wie es bei einem direkten Zugang zum Firmennetzwerk möglich ist.

3. Forderungen an ein Software-Support-Management-System

Nach der umfangreichen Analyse des Ist-Zustandes und den daraus resultierenden Defizite sollen nun die Forderungen an ein SSMS formuliert werden. Dabei gilt es insbesondere die kommunikativen Schwächen der vorhandenen Lösungen zu beseitigen und gleichzeitig die Durchlaufzeiten der CALLs zu beschleunigen. Ein ganzheitlicher Ansatz soll die unterschiedlichen Datenplattformen zusammenführen, um die Vielzahl der Redundanzen zu eliminieren. Letztendlich soll das SSMS sowohl für den Support und das Systemhaus als auch für den Endkunden eine effizientere Lösung zur Bewältigung der alltäglich auftretenden Probleme bei dem Einsatz von 3D-CAD-Programmen und der damit assoziierten Softwareperipherie bieten.

3.1. Umfang des SSMS

Die Notwendigkeit der Einführung eines SSMS folgt aus den Anforderungen der unterschiedlichen Bereiche rund um die Distribution und den Support von Softwareprodukten. Im vorliegenden Fall sind dies komplexe Anwendungen aus dem CAD-Bereich. Hierbei kann der Support keinesfalls getrennt von der Auftragsabwicklung oder Vertrieb betrachtet werden, da sämtliche Bereiche sich gegenseitig beeinflussen. Das folgende Beispiel soll dies verdeutlichen: Die Akquirierung eines Neukunden durch den Vertrieb bedingt in der Regel auch die Einbindung des Supports in die Beratungsmaßnahmen, da die Anforderungen an die Software häufig komplex sind. Durch das Hinzuziehen eines Supporttechnikers wird sichergestellt, dass die Anforderungen an das zu beschaffende CAD-System auch technisch umsetzbar sind. Gegebenenfalls werden in diesem Stadium auch Softwareentwickler hinzugezogen, um die Machbarkeit von Customizing-Maßnahmen zu beurteilen.

Im Falle eines Kaufs wickelt die Auftragsabwicklung den administrativen Teil dieses Vorganges, wie z.B. Angebotserstellung, Auftragsbestätigung, Rechnungsstellungen und Zahlungseingänge ab. Anschließend wird der gegenwärtige Bearbeitungsstatus an den Vertrieb übermittelt.

Bei Vertragsabschluss werden vom Kunden nicht nur die Softwareprodukte, sondern auch Consulting- und Schulungsmaßnahmen zur Inbetriebnahme der Software erworben. Darüber hinaus gibt der Kunde eventuell erforderliche Programmierarbeiten in Auftrag. Die Initiierung dieser Maßnahmen erfolgt maßgeblich durch die Auftragsbearbeitung in Abhängigkeit von der Auftragsbestätigung durch den Kunden und in Absprache mit den zuständigen Support-, Entwicklungs- und Vertriebsmitarbeitern. Schließlich folgt während und nach der

Implementierungsphase die Anwender-Unterstützung durch den Support, um das tägliche Arbeiten mit der Software zu optimieren oder im Notfall den Betrieb wieder herzustellen.

Obwohl dies nur ein Ausschnitt aus den Prozessen rund um das Tätigkeitsfeld eines Systemhauses ist, so wird doch die intensive Verflechtung aller Bereiche rund um die Distribution und den Support komplexer Softwareprodukte deutlich. Wenn auch eine erste Umsetzung des SSMS nicht all diese Bereiche umfassen muss, so sind diese doch bei der Formulierung der Anforderungen bereits zu berücksichtigen und die Möglichkeiten zu deren nachfolgender Integration vorzusehen. Der im folgenden dargestellte, erster Ansatz für das SSMS soll sich zunächst auf das Kerngebiet des Softwaresupports beschränken, muss jedoch hinreichend flexibel und ausbaufähig sein, um die nachträgliche Einbindung angrenzender Organisationsbereiche zu ermöglichen. Um dies zu gewährleisten, ist das SSMS als modulares System zu konzipieren. Es muss jedoch stets betont werden, dass ein wie auch immer geartetes SSMS nicht den Supporttechniker mit dessen individuellen Fähigkeiten ersetzen soll und kann. Vielmehr soll ein solches System durch die Bereitstellung unterschiedlichster Funktionen den Supportmitarbeiter in dessen Tätigkeiten unterstützen und dabei Routinetätigkeiten mit Hilfe von Workflows automatisieren.

3.2. Anforderungen an das Datenmanagement

Da dem Datenmanagement eine zentrale Rolle bei der Bearbeitung von Kundenanfragen zukommt, muss hier eine einheitliche Zugriffsebene auf sämtliche Informationen geschaffen werden. Die gewachsene Struktur eines Systemhauses verkompliziert jedoch vor allen Dingen die Vereinheitlichung der unterschiedlichen, gleichwohl gewachsenen Datenbasen. Letztendlich bieten sich nur zwei Möglichkeiten, um dieses Defizit auszuräumen:

- Ein einheitlicher Zugriff auf die unterschiedlichen Datenbanken mit Hilfe eines gemeinsamen Portals oder
- die Zusammenführung aller vorhandenen Datenpools in eine einzige Datenbank bei Bereitstellung einer umfassenden Zugriffsebene.

3.2.1. Organisation und Verwaltung von Kundeninformationen

Die in mehreren Datenbanken vorhandenen Kundeninformationen müssen zunächst so konsolidiert werden, dass redundante Informationen eliminiert, vorhandene Informationen aktualisiert und fehlende Informationen ergänzt werden. Zu den zu aktualisierenden Informationen zählen dabei z.B. die detaillierte Auflistung sämtlicher beim Kunden

vorhandenen CAD-Applikationen inklusive der dazugehörigen Lizenzinformationen, aber auch Kunden- und Lieferadresse sowie Ansprechpartner. Zu ergänzen wäre beispielsweise das Betätigungsfeld oder das Produktspektrum eines Kunden oder Informationen zur IT-Infrastruktur mit PPS- bzw. ERP-Systemen. Der komplette Datenbestand muss anschließend zwingend in ein einziges Datenbanksystem zusammengeführt werden, um die Schaffung einer einheitlichen Zugriffsebene zu ermöglichen. Die Organisation der Daten muss in einer vorgegebenen Struktur in Form von Kategorien erfolgen, um die Suche und das Filtern nach bestimmten Informationen zu vereinfachen und zu beschleunigen.

3.2.2. Aufbau und Inhalt von Dokumentation

Der Aufbau und Inhalt künftiger Dokumentationen muss stets die Nachvollziehbarkeit einzelner Kundenanfragen gewährleisten und deren Bearbeitungsstatus für alle Beteiligten deutlich visualisieren. Deshalb muss der zu realisierende Aufbau der Dokumentation von CALLs, VVs und Bugs zweckmäßigerweise mittels einer datenbankähnlichen, vordefinierten, hierarchischen Struktur erfolgen. Diese muss die bereits im Siebel-Portal oder sonstigen Herstellerportalen vorhandenen Kategorien abbilden, um eine möglichst reibungslose Übertragung der Daten für die SR zu ermöglichen. Für weitere Anwendungsfälle und Applikationen neben der zu betreuenden CAD-Software ist diese Struktur entsprechend zu ergänzen und zu erweitern. Ziel der Kategorisierung muss es sein, den Anteil von Freitexten bei Problembeschreibungen auf ein Minimum zu reduzieren. Hierdurch soll eine schnelle Erfassung der Anfragen und eine effiziente Suche innerhalb der Datenbank ermöglicht werden. Die Anfragen müssen außerdem mit einem Bearbeitungsstatus versehen werden, um Mehrfachbearbeitungen durch unterschiedliche Support-Mitarbeiter zu verhindern. Um die Transparenz für sämtliche Involvierten zu verbessern sind die Anfragen durch die Vergabe einer eindeutigen, alphanumerischen Kennung zu kennzeichnen.

Des Weiteren ist das Priorisieren von CALLs und Bugs mittels einer ergänzenden, gestaffelten Kennung vorzusehen, z.B. nach der Art von Schulnoten, welche den Grad der Produktivitätseinbußen beim Kunden durch den beschriebenen Fehler / das Problem wiedergeben. Auch hierbei erscheint eine Orientierung an den vorhandenen Strukturen des Siebel-Partnerportals von SolidWorks als sinnvoll. Sämtliche Dokumentationen müssen mit den Kundeninformationen des Fragestellers verknüpft werden, damit sich der Supporttechniker schnell einen umfassenden Überblick über die Gesamtzahl und den Status aller CALLs eines Kunden verschaffen kann. Durch diese Maßnahmen wird auch dem so genannten „*Problemfremden*“ ein Zugriff auf sämtliche, problemrelevanten Informationen und ein schneller Überblick über die gesamte Sachlage ermöglicht.

3.2.3. Verwaltung von Experten-Wissen

Bei der Beurteilung des Wissensmanagements muss sowohl der Erfassung als auch der Suche nach problemrelevanten Informationen in gleicher Weise Beachtung geschenkt werden. Prinzipiell sind bei der Wissenserfassung drei unterschiedliche Bereiche zu beachten:

- Problemspezifisches oder fehlerspezifisches Wissen, welches durch die Bearbeitung von Hotline-Anfragen im Supportalltag erworben wird. Dazu zählt z.B. die Erkenntnis, dass eine Funktionsbeschränkung oder Fehlfunktion einer Applikation eventuell durch einen Softwarefehler und nicht durch falsche Bedienung oder durch eine ungeeignete Systemkonfiguration hervorgerufen wird.
- Implementierungsspezifisches Wissen, das im Rahmen der Integration von Applikationen in eine Systemumgebung zur Anwendung kommt bzw. dadurch erweitert wird. Dazu zählen Kenntnisse über die Rahmenbedingungen einer Integration in Bezug auf Betriebssystem, Netzwerkplattform, evtl. vorhandene Applikationen oder der Distributionstyp von Applikationslizenzen.
- Anwenderspezifisches Wissen, welches Strategien zur Umsetzung unterschiedlicher Konstruktionsszenarien umfasst. Hier werden umfangreiche Informationen darüber gespeichert, welche Vorgehensweisen beim Einsatz der Software, z.B. bei Änderung von Parametern, zu einem definierten Verhalten der generierten Daten führen, welche im Vorhinein innerhalb eines funktionellen Rahmens bestimmt wurden. So ist bei der Modellierung von Freiformgeometrien für Konsumgüter eine andere Vorgehensweise angeraten, als bei der Darstellung einer komplexen Sondermaschine.

Der Übergang zwischen den unterschiedlichen Bereichen ist mitunter fließend. Bei der Analyse des gegenwärtigen Zustandes fällt jedoch auf, dass die Dokumentation von Wissen, wenn überhaupt, in höchst heterogener Form erfolgt. Hierbei kommen MS-Word-Dokumente ebenso zum Einsatz, wie einzelne, nicht kategorisierte Datenbankeinträge in Form von Freitexten im Extranet, mangelhaft gepflegte und oftmals nicht aktuelle FAQs in Form von Newslettern oder die bunt gemischten FAQs der Firma SolidWorks auf deren Homepage. Keines dieser Instrumente kann der wachsenden Komplexität der Applikationen und den Kundenanforderungen in der Produktentwicklung bei ständig zunehmenden Termindruck gerecht werden. Hier muss ein effizientes, dynamisches und applikationsübergreifendes System zur Wissensdistribution Abhilfe schaffen, um nicht zuletzt auch in Zukunft die Handlungsfähigkeit des Supports zu gewährleisten.

Es wurde hinreichend belegt, dass die Bedeutung von dokumentiertem Wissen für den Fortbestand und die Handlungsfähigkeit eines Systemhauses kaum überschätzt werden kann. Diese Dokumentation ist die Basis für die Effizienz künftiger Supporteinsätze und bedeutet dabei gleichzeitig die Unabhängigkeit des Systemhauses von personeller Fluktuation innerhalb seiner Supportorganisation. Anzustreben ist daher die vollständige Integration einer Wissensdatenbank in das SSMS mit den Möglichkeiten, zum einen fehlerspezifische CALLs direkt aus dem CALL-Handling-System zu übernehmen und zum anderen zu bestimmten Themengebieten des Integrations- und Anwenderwissens Artikel zu verfassen. Die Durchsuchung dieser Datenbank muss mittels hochdifferenzierbarer Abfragen über eine Suchmaschine ermöglicht werden.

Sinnvoll erscheint in diesem Kontext ein Ansatz, wie er bereits in Form der Wiki-Engines verfolgt wird, deren prominentester Vertreter die Online-Enzyklopädie wikipedia [16] ist: Bei diesem Ansatz hat ein beliebiger User die Möglichkeit, innerhalb eines erweiterbaren Grundgerüsts Essays zu bestimmten Thematiken zu verfassen. Diese können anschließend von jedem anderen Usern bearbeitet, korrigiert und erweitert werden. Der ursprüngliche Verfasser wird über jede Änderung informiert und moderiert den Artikel gewissermaßen. Dadurch wird gewährleistet, dass ein „*roter Faden*“ innerhalb eines Themas beibehalten wird und Ergänzungen nicht zu stark vom diesem abschweifen bzw. gegebenenfalls in weiterführende Artikel umgeleitet werden. Da das Wiki-Prinzip bereits erprobt ist und damit seine Tauglichkeit und sein Leistungspotential unter Beweis gestellt hat, ist es nahe liegend, eine solche Engine als Basis für das Wissensmanagementsystem des SSMS zu verwenden.

Im Rahmen eines Lebenszyklus müssen die Wissens-Beiträge regelmäßig überprüft und überarbeitet werden, spätestens zu jedem Versionswechsel der betroffenen Software. Hierbei sind bestimmte Beiträge potentiell häufiger von Änderungen betroffen als andere. So ist problemspezifisches oder fehlerspezifisches Wissen am häufigsten Veränderungen unterworfen, die auch während relativ kurzer Zeiträume innerhalb einer Softwareversion im Rahmen der Veröffentlichung von Servicepacks auftreten. Aus diesem Grund ist es sinnvoll, auch das Bugtracking mit in diesen Prozess zu integrieren. Innerhalb des SSMS müssen deshalb alle Beiträge entsprechend kategorisiert werden, um die turnusmäßige Überprüfung zu beschleunigen.

Implementierungsspezifisches Wissen ist im Laufe eines Releases aufgrund der konstant bleibenden Systemanforderungen und Implementierungsroutinen kaum einer Änderung unterworfen. Die ständige Entwicklung von Hardwareperipherie, Betriebssystemen und Office-Produkten bedingt jedoch auch hier Veränderungen, die in der Regel jeweils zu einem Releasewechsel greifen. Eine entsprechende Kennzeichnung von Artikeln zu diesem Thema muss also bei einem Releasewechsel eine Aufforderung des SSMS zur Überprüfung der Inhalte durch den Support nach sich ziehen.

Anwenderspezifisches Wissen ist am wenigsten von grundlegenden Veränderungen betroffen. Hier könnte das SSMS den Lebenszyklus unterstützen, indem der Verfasser von Artikeln, welche z.B. mehr als 1 Jahr nicht modifiziert wurden, turnusmäßig an die Überprüfung der Inhalte oder auch an das Schließen oder Entfernen der Artikel erinnert werden.

Da der Stellenwert einer Wissensdatenbank für das Systemhaus sehr hoch sein muss, die Dokumentation jedoch für die betroffenen Mitarbeiter vordergründig eine Zusatzbelastung darstellt, müssen im Rahmen der Einführung eines SSMS gezielt Anreize für die Supportmitarbeiter zur Erfassung dieses Wissens geschaffen werden, um diese nicht der Eigeninitiative des betroffenen Personals zu überlassen und somit gleichwohl eine nur schleppende Entwicklung der Wissensdatenbank in Kauf zu nehmen.

3.2.4. Übernahme und Verwaltung von Bestandsdaten

Die Übernahme von Bestandsdaten aus sämtlichen Bereichen in ein einheitliches Datenbanksystem ist der Schaffung einer einheitlichen Zugriffsebene auf die unterschiedlichen, bestehenden Bestandsdaten aus den bereits genannten Gründen vorzuziehen, wobei eine Konsolidierung aller Daten oberste Priorität haben muss. Sämtliche Redundanzen sind im Rahmen der Übernahme durch einen gegenseitigen Abgleich zu eliminieren und assoziierte Daten zu einer eindeutigen Datenbasis zusammenzuführen. Die Qualität der somit geschaffenen Datenbasis hat größten Einfluss auf den verfügbaren Informationsstand und damit auch auf die Handlungsfähigkeit sowohl der Supportorganisation als auch des Systemhauses bezüglich der Effizienz bei der Bearbeitung von Anfragen und der Nutzung von Vertriebspotentialen im vorhandenen Kundenstamm.

3.3. Anforderungen an das Prozessmanagement

Ein vordefiniert Ablauf von untereinander abhängigen Prozessen ermöglicht es dem Support-Techniker, sich auf die eigentlichen Aufgaben der Problemlösung zu konzentrieren. Gegenwärtig ist ein solcher Ablauf bei der Problembewältigung gleichwohl nur rudimentär oder überhaupt nicht vorhanden. Deshalb muss in allen Bereichen des CALL-Handlings eine definierte Struktur für die unterschiedlichen Stufen der Problembewältigung geschaffen werden.

3.3.1. Bearbeitung von Anfragen

Eine vorrangige Aufgabe bei der Abarbeitung von Anfragen ist die Vermeidung von Mehrfachbearbeitungen durch Mitarbeiter des Supports zwecks Schonung der personellen Ressourcen des VAR. Während die Anfragen über unterschiedliche Kanäle, hauptsächlich telefonisch oder per eMail, gestellt werden, besteht vor allen Dingen bei telefonischen Mehrfachanfragen durch einen Kunden die Gefahr, dass unterschiedliche Supportmitarbeiter ein- und dieselbe Anfrage bearbeiten, wenn der erste CALL des Kunden nicht zeitnah erfasst wurde. Ein Beispiel soll diese Problematik verdeutlichen:

Supporttechniker „A“ erhält im Rahmen einer Anwender-Hotline einen Anruf vom Kunden „Z“ bezüglich eines Problems bei der Anwendung einer Applikation, welches diesen bei der Fortführung seiner Arbeit behindert. Support-Mitarbeiter „A“ kann dieses Problem nicht direkt im Verlauf des Gesprächs lösen, erbittet deshalb vom Kunden „Z“ etwas Zeit zur Lösungsfindung und sagt einen Rückruf zu. Wenn der Supporter „A“ nun die gewonnene Zeit für die Lösungssuche anstatt für das Einpflegen des CALLs in den Helpdesk investiert, oder unmittelbar ein weiterer Anruf über die Anwender-Hotline bei Supporter „A“ eingeht, so besteht die Gefahr, dass bei einem erneuten Anruf des seinerseits unter Termindruck stehenden Kunden „Z“ die Telefonanlage diesen an einen anderen Support-Techniker „B“ weiterleitet. Dieser beginnt nun seinerseits wiederum mit der Lösungssuche, da er von der vorausgegangenen Anfrage desselben Kunden aufgrund der fehlenden CALL-Dokumentation keine Kenntnis hat.

Auf den ersten Blick scheint die Lösung für dieses Problem in der unmittelbaren Erfassung eines jeden CALLs zu liegen. Die Praxis zeigt jedoch, dass mit den gegenwärtig zur Verfügung stehenden Mitteln für die CALL-Erfassung aufgrund des unergonomischen Handlings und der zu durchlaufenden, zeitraubenden Vorgänge die Forderung nach einer solchen Echtzeiterfassung in Zeiten hoher Auslastung nicht praktikabel ist. Erschwerend kommt hinzu, dass selbst im Falle der erfolgten Dokumentation eines CALLs vom Kunden „Z“, der Supporter „B“ erst sehr aufwändig innerhalb des Helpdesks nach einem eventuell schon vorhandenen Eintrag über diesen CALL vom Kunden „Z“ suchen muss, da effiziente und performante Suchalgorithmen fehlen. Zwar erfolgt die Kennzeichnung eines Vorgangs mit einer CALL-Nummer innerhalb des Helpdesks, diese wird jedoch nicht an den Kunden übermittelt, so dass bei Rückfragen kein Bezug darauf genommen werden kann. Diese Konstellation bedeutet nicht nur für den Kunden sondern auch für das Systemhaus eine nicht akzeptable Produktivitätseinbuße bei der Disposition von Anfragen.

Das SSMS muss also durch eine möglichst zeitnahe und effiziente Erfassung von

Kundenanfragen die Mehrfachbearbeitung ein- und desselben CALLs durch unterschiedliche Supportmitarbeiter in jedem Fall vermeiden helfen. Hierzu muss im Falle einer Kundenanfrage die Telefonnummer von der Telefonanlage an das SSMS übermittelt und der entsprechende Kundendatensatz dort im Interface dargestellt werden. Anschließend muss der Workflow für die Problemaufnahme vom Supporttechniker initialisiert. Die Problemaufnahme muss dabei innerhalb vordefinierter Kategorien und Prioritäten erfolgen. Mit Hilfe dieser Einordnung kann bei durchschnittlicher Auslastung der Hotline die Abarbeitung der CALLs sequentiell erfolgen, während in Zeiten hoher Auslastung die Abarbeitung auf Grundlage der Priorität geschehen kann. Dies wiederum gewährleistet, dass schwerwiegende Probleme, also solche, welche die Produktivität des Kunden stark beeinträchtigen, in jedem Falle zeitnah bearbeitet werden. Nur die beschriebene, zeitnahe Erfassung von Anfragen lassen sich Mehrfachbearbeitungen und die daraus folgenden Redundanzen und Produktivitätseinbußen vermeiden.

Bei Anfragen, die per eMail an den Support gestellt werden, muss ebenfalls eine möglichst direkte Übernahme in das SSMS erfolgen, damit auch hier eine Priorität festgelegt werden kann. Das Queing der Anfragen, also die Festlegung der Reihenfolge, in der die einzelnen Anfragen abgearbeitet werden, und die Zuweisung an die aktiven Hotline-Mitarbeiter muss anschließend anhand dieser Priorität erfolgen. Die Zuweisung der Anfragen zu einzelnen Support-Mitarbeitern wird anschließend entweder mittels Notifikation per eMail und Weiterleitung durch das SSMS oder durch einen Disponenten mit den entsprechenden Berechtigungen bewerkstelligt.

Das SSMS muss schließlich die Möglichkeiten bieten, mit Unterstützung definierbarer Workflows Anfragen zu unterschiedlichen Applikationen gemäß den Anforderungen des jeweiligen Entwicklungs- oder Kompetenzteams zu formatieren, zu konfigurieren und an die Verantwortlichen zu übermitteln. Um den Aufwand der manuellen Übertragung von eMail-Anfragen in das SSMS mittelfristig zu verringern, muss für Wartungskunden ein Webportal mit Anbindung an das SSMS bereitgestellt werden. Hier muss ein Workflow mit einer Benutzerführung integriert werden, welche die Auswahl der Problemkategorie und Priorität sowie die Erstellung der Problembeschreibung beinhaltet und die Möglichkeit zum Upload problemrelevanter Daten bietet. Durch ein solches Portal wird nicht nur der Hotline-Support durch den Wegfall der manuellen Übertragung von Anfragen in das SSMS entlastet, sondern auch ein Mehrwert für die Endkunden geschaffen, da CALLs auch außerhalb der Geschäftszeiten der Hotline, also 24 Stunden und 7 Tage die Woche, abgesetzt werden können.

3.3.2. Verifikation von Bugs

Die Verifikation von Bugs erfolgt gegenwärtig nach Erfahrungswerten. Der Prozess lässt sich dabei in einem sequentiellen Schema abbilden und ist somit prädestiniert für die Integration in das SSMS mit Hilfe eines Workflows. Die erforderlichen Schritte umfassen die Reproduktion eines Fehlers anhand von Kundendaten und der betroffenen Software-Version, sowie die Eingrenzung auf einen bestimmten Anwendungsfall bzw. auf eine bestimmte Hardwareplattform. Ist der Fehler lokalisiert, so muss aus dem CALL ein Bugreport mit allen notwendigen Informationen und Daten generiert und gemäß den Formatierungskonventionen des Softwareherstellers an diesen übermittelt werden. Damit ist die Verifikation von Bugs ein spezieller Fall des CALLs, wobei sich dessen Natur mitunter aber auch erst im Rahmen der Analyse erschließt. Im Rahmen der Analyse muss auch überprüft werden, ob der Fehler durch die Konfiguration der Systemumgebung hervorgerufen wird, da die Unterstützung der Kunden in diesen Fragen gegenwärtig nicht in die Zuständigkeit des Applikationssupports fällt. Schließlich muss überprüft werden, ob es sich um einen Fehler im Quellcode der Applikation oder um einen Bedienungsfehler durch den Endkunden handelt. Das SSMS muss diese Vorgänge als Sonderfall der Anfrage-Bearbeitung innerhalb des zuvor beschriebenen, allgemeinen CALL-Handling-Workflows abbilden. Wie bei der allgemeinen Abarbeitung und Dokumentation von Anfragen ist auch hier die Definition von hinreichenden, umfassenden und differenzierten Kategorien seitens des SSMS notwendig.

3.3.3. Vorqualifizierung von VVs

Bei der Vorqualifizierung von VVs, falls diese an den 1.Level-Support herangetragen und nicht direkt vom Kunden an den Hersteller übermittelt werden, ist in erster Linie zu prüfen, ob die geforderte Funktionalität durch den Einsatz bestehender Funktionen realisiert werden kann. Die Effizienz dieses Vorgangs beruht jedoch in erster Linie auf den durch Erfahrung erworbenen, applikationsspezifischen Kompetenzen des jeweiligen Support-Mitarbeiters und lässt sich damit kaum durch ein SSMS unterstützen.

Die Prioritäten bei der Abarbeitung von VVs durch den Softwarehersteller werden primär durch drei Faktoren beeinflusst und zwar durch:

- die Häufigkeit, mit der eine Funktionalität von bestehenden Kunden nachgefragt wird (die so genannten „Customer-Hits“),
- die evtl. Verbesserung der Wettbewerbssituation des Produktes durch Bereitstellung dieser Funktionen und
- den Aufwand zu dessen programmiertechnischer Umsetzung.

Der Support kann also die Erfolgsaussichten eines VVs in erster Linie dadurch verbessern, indem er dafür sorgt, das VV mit ähnlichem Inhalt zusammengeführt werden, um die „Customer-Hits“ dieser Anfragen zu erhöhen. Aus diesem Grund ist die Aufgabe des SSMS in diesem Kontext darin zu sehen, im Rahmen von Workflows für die Dokumentation der VV Funktionen zu implementieren, die es dem Support-Techniker ermöglichen, zu verifizieren ob ein derartiger VV bereits gestellt wurde, um schließlich die Aussicht auf eine Umsetzung dieses VV durch einen weiteren „Customer-Hit“ zu verbessern.

Obwohl die VV von Seiten des Herstellers zum Teil mit Nummernschlüsseln versehen werden, ist eine Suche nach bestehenden VV innerhalb des Herstellerportals nicht möglich. Deshalb müssen die gestellten VV seitens des Systemhauses in einer eigenen Datenbank als Teil des SSMS gespeichert und mit dem Nummernschlüssel des Herstellers gekennzeichnet werden. Dadurch hat der Support die Möglichkeit, einen neuen VV mit den bereits abgesetzten abzugleichen und bei Übereinstimmung den entsprechenden Inhalt in einen neuen VV zu kopieren und an den Hersteller zu übermitteln. Durch den identischen Wortlaut wird somit auch die Anzahl der „Customer-Hits“ zu einem VV und dadurch gleichzeitig die Aussicht auf Umsetzung der geforderten Funktionalität erhöht.

Ein SSMS muss also sowohl die Struktur für das Absetzen der VV abbilden, als auch die Suchalgorithmen bereitstellen, um bereits formulierte VV zu einem bestimmten Themenbereich zu identifizieren. Dabei müssen die thematischen und applikationsspezifischen Kategorisierungen der einzelnen Softwarelieferanten berücksichtigt werden, um die Übermittlung der VV aus dem SSMS an die Softwareentwicklung möglichst effizient zu gestalten.

3.3.4. Externe Kommunikation

Die externe Kommunikation umfasst sämtliche Kommunikationswege von Kundenanfragen an den Support und an die Entwicklung. Sie enthält damit sämtliche, in Bezug auf den Kunden, öffentlichen Aspekte eines CALLs. Im Gegensatz dazu werden die nicht öffentlichen bzw. internen Aspekte einer Anfrage innerhalb des Supports und zwischen dem Support und der Entwicklung gegenüber der externen Kommunikation differenziert betrachtet und im folgenden Abschnitt behandelt. Eine solche, getrennte Betrachtung ist deshalb sinnvoll, da innerhalb einer Supportorganisation einer Anfrage aus unterschiedlichen Gründen zuweilen ein anderer Stellenwert eingeräumt werden muss, als dies gegenüber dem Kunden mit seinem isolierten Standpunkt vertreten werden kann. Zu diesen Gründen ist beispielsweise die wirtschaftliche Bedeutung eines Kunden für das Systemhaus (z.B. aufgrund des vorhandenen Lizenzvolumens) zu zählen, welche sich im Hinblick auf die Vergabe von Prioritäten bei der Bearbeitung von Anfragen auswirken kann. Ein Problem hat für den jeweils Betroffenen naturgemäß eine zwar immer individuelle, jedoch stets hohe Priorität. Dieser Einschätzung kann jedoch bei äußerer Betrachtung und unter Berücksichtigung der personellen Ressourcen im Support nicht immer durch eine entsprechend zeitnahe Lösung Rechnung getragen werden. Der Trennung von interner und externer Kommunikation muss das SSMS deshalb im Rahmen der Dokumentation durch die definierter Kennzeichnung von Korrespondenz Rechnung tragen.

Im Falle eines Softwarefehlers dokumentiert der Support diesen Fehler, leitet diesen anschließend zusammen mit der Dokumentation an die Entwicklung weiter und übermittelt abschließend die gefundene Lösung an den Kunden. Damit bildet der Support den zentralen Kommunikationsknoten bei der Bewältigung von CALLs und bündelt diesbezüglich sämtliche spezifischen Informationen. Innerhalb dieses Prozesses muss die Kommunikation von und zum Kunden bezüglich ein- und derselben Anfrage im Idealfall über einen einzigen Ansprechpartner beim Support erfolgen. Somit muss der Kunde nicht bei jeder Nachfrage zum Status einer Anfrage oder Übermittlung ergänzender Information die gesamte Problematik von neuem Erläutern. Gleichzeitig bleibt hierdurch innerhalb des Supports der Überblick über den Status einer Problemsituation gewahrt. Da dies in der Praxis bisher nicht gewährleistet ist, führt dieser Umstand zur Forderung nach einer hinreichend detaillierten Dokumentation von Anfragen, damit sich jeder Involvierte stets ein umfassendes Bild über den bisherigen Verlauf und den gegenwärtigen Status einer Anfrage verschaffen kann. Darüber hinaus muss der ursprüngliche Autor, z.B. bei einer vorübergehenden Abwesenheit, ständig über den zwischenzeitlichen Fortschritt in der Bearbeitung eines CALLs informiert werden. Dadurch ist gewährleistet, dass der Betroffene bei einer möglichen Eskalation des

CALLs gegebenenfalls sofort eingreifen kann. Damit alle in einen CALL Involvierten bei einer Statusänderung oder einer Aktualisierung einer Anfrage unmittelbar in Kenntnis gesetzt werden können, muss diese Funktion zweckmäßigerweise über eine automatische Notifikation per eMail durch das SSMS umgesetzt werden.

3.3.5. Interne Kommunikation

Die interne Kommunikation umfasst im Gegensatz zur externen nur die Kommunikationswege innerhalb der Supportorganisation sowie zwischen dem Support und der Entwicklung, also sämtliche Korrespondenz, die den Kunden aus unterschiedlichen Gründen nicht mit einbezieht. Die möglichen Gründe hierfür wurden bereits dargelegt und sind mannigfaltig. Während die Risiken der Verletzung dieser Trennungsebene durch den ursprünglich involvierten Supporttechniker relativ gering und auch aufgrund des individuellen Charakters einer jeden Anfrage im Rahmen eines Prozessmanagements kaum abzufangen sind, so muss die Verhinderung einer solchen Verletzung durch Dritte mit Hilfe einer strikten Kategorisierung der dokumentierten Korrespondenz im SSMS garantiert werden. Diese Kategorisierung ermöglicht Außenstehenden einen hinreichend differenzierten Überblick über den anfragespezifischen Informationsaustausch sowohl innerhalb der Supportorganisation, als auch zum Hersteller und zum Endkunden.

3.3.6. Dokumentation und Erfassung von Expertenwissen

Wie bereits beschrieben, wird im Rahmen der Supporttätigkeit eines Systemhauses eine Vielzahl von applikationsspezifischen Kompetenzen erworben. Um diese künftig für alle im Rahmen von Implementierung und Support Involvierten nutzbar zu machen, müssen diese Kenntnisse konsequent dokumentiert und gepflegt werden. Dabei muss der Aufwand sowohl für die Dokumentation neuer Erkenntnisse als auch die Erweiterung und Pflege bestehender Dokumentationen möglichst gering gehalten werden.

Wissen in jeder Form wird ständig erweitert, relativiert und revidiert. Dies trifft in besonderem Maße auf applikationsspezifisches Wissen zu. Ein Beispiel soll dies verdeutlichen:

In der Vergangenheit musste das Anfügen von Stücklistensymbolen in einer Zusammenbauzeichnung manuell erfolgen, im aktuellen Softwarerelease jedoch erfüllen Automatismen diesen Zweck. War es in vergangenen Releases nicht möglich, Metadaten mit diesen Stücklistensymbolen ohne Programmieraufwand zu verlinken, so steht gegenwärtig eine Funktion innerhalb der Standardsoftware zur Verfügung, die eine solche Verlinkung mittels einfacher Konfiguration ermöglicht.

Hiermit wird deutlich, dass ein Wissensbestand immer in einem zeitlichen Kontext existiert und ständig aktualisiert werden muss. Damit erscheint es zweckmäßig, die Dokumentation dieses Wissens in eine Art Lebenszyklus zu integrieren, der an den des jeweiligen Softwareproduktes gekoppelt ist. Hierdurch wird eine ständige Überprüfung, Aktualisierung und Anpassung der Erkenntnisse in Bezug auf das sich weiterentwickelnde Softwareprodukt gewährleistet.

Die Erfassung und Erweiterung von Wissen in der Wissensdatenbank des Systemhauses, sei es durch Übernahme von Erkenntnissen aus dem täglichen Supportalltag oder aus Kundenanfragen heraus, soll möglichst effizient und unkompliziert erfolgen. Das SSMS muss die Erfüllung dieser Ziele durch Abbildung eines Lebenszyklus für das Bestandswissen und durch die Schaffung von Workflows zur Erfassung, Integration und Pflege neuer Erkenntnisse gerecht werden. Dabei muss die Erfassung von Erkenntnissen innerhalb derselben Kategorien erfolgen, wie sie auch beim CALL-Handling zum Einsatz kommt. Dies ermöglicht eine automatische Filterung und Anzeige von Einträgen der Wissensdatenbank zum Thema eines aktuellen CALLs, dadurch auch dem Supporttechniker einen direkten Überblick über eventuell vorhandene Lösungsansätze bieten und somit die Abwicklung der Anfrage signifikant beschleunigen.

3.3.7. Qualitätssicherung

Der Begriff Qualitätssicherung (QS) ist im Zusammenhang mit der gegenwärtigen Praxis im Software-Support kaum mit Inhalten gefüllt. Lediglich ein Teil der Entwicklungsprozesse der hier betrachteten Softwareprodukte weisen Merkmale einer Qualitätssicherung auf, während diese im Bezug auf die Supportdienstleistung gänzlich fehlen. Die derzeitigen Strukturen würden zudem die Schaffung und Durchführung einer solchen Qualitätssicherung aufgrund der mangelnden Transparenz und Nachverfolgbarkeit einzelner Anfragen kaum ermöglichen. Ein SSMS kann hier nur mit Hilfe von mittels Workflows implementierten und definierten Prozessen sowie durch statistische Auswertungen abgeschlossener Anfragen ein Qualitätsmanagement ermöglichen bzw. unterstützen. Die Definition der qualitätsrelevanten Prozesse sowie der Qualitätsmerkmale von Software und Supportdienstleistungen muss jedoch im Vorfeld außerhalb des SSMS erfolgen und ist nicht Gegenstand dieses Konzeptes. Eine zusätzliche Methode zur QS stellen regelmäßige, stichprobenartige Kundenbefragungen bezüglich der Abwicklung einzelner CALLs dar. Dazu versendet das SSMS nach definierten Regeln automatische eMails an die betroffenen Kunden. In diesen eMails werden die Kunden aufgefordert, mit Hilfe eines vordefinierten, interaktiven Bewertungsbogens, die Qualität der Supportdienstleistung in einem konkreten Fall zu

beurteilen. Die Ergebnisse dieser Untersuchungen können anschließend im Rahmen eines QM-Audits ausgewertet werden, um anschließend geeignete Maßnahmen zur Verbesserung der identifizierten Defizite abzuleiten.

Das von den Unternehmen angestrebte Ziel einer stetigen Produktivitätssteigerung verlangt auch im Softwaresupport nach einer ständigen und nachhaltigen Verbesserung. Aus diesem Grund ist auch eine mit der Einführung eines SSMS parallel verlaufende Integration eines Qualitätsmanagements unabdingbar. Daraus lässt sich die Forderung ableiten, das als Voraussetzung für die Qualitätssicherung die Definition der im Support ablaufenden Prozesse und das Spektrum der Dienstleistungen zwangsläufig Teil eines Einführungskonzeptes für ein SSMS sein muss. Das SSMS muss die von diesen Definitionen abgeleiteten Workflows und Verifizierungs-Tools beinhalten, um die Einhaltung der Prozessabläufe zu gewährleisten und Verbesserungspotentiale im Supportprozess zu visualisieren.

3.4. Anforderungen an die Benutzeroberfläche

Die Benutzeroberfläche (GUI) und die Bedienungslogik eines SSMS muss sich möglichst dicht an gängigen Softwarelösungen des MS-Office-Umfeldes anlehnen, da auch die bisher betreuten Softwareprodukte und eingesetzten Tools durchweg WINDOWS-basierend sind. Hierdurch wird nicht nur die Integration in die bestehende WINDOWS-Systemumgebung vereinfacht, sondern auch die Einarbeitungszeit aller Beteiligten für das neue System verkürzt und das Risiko für Bedienungsfehler minimiert. Zu den Merkmalen eines solchen Interfaces zählen primär WINDOWS-konforme Menüstrukturen und Icons, Tastenkürzel, Kontextmenüs und Hilfefunktionen. Das GUI muss über eine Startmaske den schnellen und intuitiven Zugriff auf sämtliche Kundeninformationen gestatten und stets einen möglichst aussagekräftigen Überblick über den Status des Kunden gewährleisten. Der Status kann beispielsweise durch so genannte „Emoticons“ (z.B. ☺ ☹) und entsprechende Farbgebung visualisiert werden.

Da die Erfassung des Kundenstatus durch den Support-Techniker im Falle einer telefonischen Anfrage innerhalb weniger Sekunden erfolgen muss, sollte die Startmaske dabei folgende Informationen in kompakter Form bereitstellen bzw. durch geeignete Mittel visualisieren:

Wirtschaftlich relevante Informationen für das Systemhaus:

- Hat der Kunde einen Software-Wartungsvertrag oder ist dieser durch andere Umstände zur kostenfreien Inanspruchnahme der Hotline / des Supports berechtigt?

- Handelt es sich um einen Bestandskunden oder um einen Interessenten bei der Durchführung von Softwaretest im Rahmen der Entscheidungsfindung bei der Anschaffung eines neuen CAD-Systems?
- Wie wichtig ist der Kunde für das Systemhaus unter Berücksichtigung des Umfangs an vorhandenen Softwarelizenzen oder als Referenzkunde in einem Marktsegment?

Status-Relevante Informationen zum Kunden:

- Wie viele CALLs des Kunden sind gegenwärtig in Bearbeitung?
- Wie sind diese CALLs priorisiert?
- Welcher Art waren die bisherigen Anfragen?

Qualitätsrelevante und subjektive Informationen zum Kunden:

- Wie viele Anfragen hat der Kunde bereits an die Hotline gestellt?
- Wie lange war der durchschnittliche Bearbeitungszeitraum bis zum Abschluss dieser CALLs?
- Wie schätzen die mit dem Kunden in Kontakt getretenen Support-Mitarbeiter die Zufriedenheit des Kunden und dessen Leidensfähigkeit ein?
- Wirkte der Kunde bei vorausgegangenen Kontakten verärgert oder kam es zu Eskalationen im Rahmen einzelner Anfragen?

Zusätzlich zu diesen Kundeninformationen muss das GUI eine schnelle und unkomplizierte Initiierung der Prozesse für Dokumentation, Kategorisierung und Vergabe von Prioritäten bieten. Der Ergonomie und der Benutzerführung bei der CALL-Erfassung muss im Interesse eines möglichst effizienten Ablaufs eine hoher Stellenwert eingeräumt werden, um den Support-Techniker so wenig wie möglich von der eigentlichen Lösungsfindung abzulenken. Eine weitere Konkretisierung der Gestaltung des GUI wird im Rahmen des Konzeptes erfolgen, da diese stark vom Umfang der letztendlich implementierten Funktionen abhängt.

3.5. Anforderungen an die Anbindung und Integration des SSMS

In der vorausgegangenen Analyse wurden die unterschiedlichen Datenbank-Strukturen der zur Unterstützung des Supports eingesetzten DV-Systeme beschrieben. Dabei ist

augenfällig, dass die vorhandene IT-Infrastruktur im Wesentlichen auf WINDOWS-basierende Anwendungen beschränkt ist und auch die Arbeitsplatzrechner durchgehend mit WINDOWS-Betriebssystemen arbeiten. Daraus ergibt sich als erste Forderung für die Anbindung und Integration eines SSMS dessen Integrationsfähigkeit in die vorhandene, WINDOWS-basierende IT-Infrastruktur, z.B. mit Office-Anwendungen (Outlook, Internet-Explorer / Mozilla-Firefox) oder vorhandene Datenbanksysteme. Ein dem SSMS zugrunde liegendes Datenbankmanagementsystem muss aus diesem Grund auch WINDOWS-kompatibel sein. Vorhandene Datenbank-Inhalte müssen mit vertretbarem Aufwand in das SSMS überführt werden können. Dazu müssen im Rahmen der Implementierung des SSMS geeignete Schnittstellen definiert und falls nicht schon verfügbar, auch programmiert werden.

Außerdem muss das SSMS eine hohe Flexibilität in Bezug auf dessen Anpassbarkeit an die spezifischen Anforderungen eines Systemhauses aufweisen. Das SSMS muss hierzu eine möglichst einfache Definition und Ergänzung von Workflows ermöglichen, um Prozessabläufe abbilden zu können. Außerdem muss sowohl die Gestaltung und Modifikation der Zugriffsebene (GUI) als auch die Anpassung von Struktur und Inhalt der zugrunde liegenden Datenbank problemlos durchführbar sein.

Die Programmierung eines SSMS muss vorzugsweise in etablierten Hochsprachen wie z.B. VBA, SQL, C++, Java, oder den .NET-Standards erfolgen, um die laufende Pflege möglichst einfach zu gestalten und um auf bereits vorhandene Literatur und Dokumentationen zurückgreifen zu können. Außerdem soll dadurch eine Erweiterungen und Anpassungen durch die eigenen IT-Fachleute des Systemhauses ermöglicht werden.

Der Zugriff auf das SSMS muss sowohl mittels LAN / Intranet als auch über das Internet möglich sein, was eine Client-Server-Struktur voraussetzt. Die Supporttechniker können dadurch auch während Ihrer Einsätze bei Kunden vor Ort Anfragen und Informationen direkt in das System einpflegen und ergänzen oder auf die Wissensdatenbank zugreifen.

Das System muss über eine Benutzerverwaltung zur hinreichend fein differenzierbaren Definition von Zugriffsrechten verfügen. Diese Benutzerverwaltung muss sowohl die Definition von Gruppen wie z.B. Support, Vertrieb, Auftragsabwicklung oder Systemadministration zulassen, als auch anwenderspezifische Zugriffsrechte abbilden können, da einzelne Personen unter Umständen mehrere Funktionen erfüllen, also beispielsweise sowohl vertriebsorientierte als auch supportorientierte Funktionen wahrnehmen.

Hohe Anforderungen sind an die Gesamtperformance des Systems in Form kurzer Antwortzeiten und schneller Datenübertragung zu stellen. Die Gewährleistung der Echtzeiterfassung von Informationen innerhalb des SSMS, z.B. im Rahmen einer telefonischen Kundenanfrage, muss in jedem Falle sichergestellt werden. Hierbei darf weder der Gesprächsverlauf gestört oder der Abarbeitungsprozess verzögert noch der

Supporttechniker von der Gesprächsführung abgelenkt werden. Da eine schlechte Performance des SSMS für jeden Anwender im Rahmen der Anfragebearbeitung sofort als Behinderung des Arbeitsablaufs spürbar wird, ist diese auch hinsichtlich der Etablierung und Akzeptanz des Systems bei den Anwendern innerhalb der Supportorganisation ein höchst kritischer Punkt.

3.6. Anbindung an vorhandene Systeme zur Unterstützung des Softwaresupports

In den vorausgegangenen Ausführungen wurde verdeutlicht, dass das Wissensmanagement, das Prozessmanagement, das Qualitätsmanagement und das Datenmanagement sowie die vertriebsorientierten Aktivitäten, die Auftragsabwicklung, die Softwarepflege und das CALL-Handling aufgrund der gegenseitigen Verflechtungen der einzelnen Bereiche innerhalb einer einzigen Plattform zusammengeführt und verwaltet werden müssen. Darüber hinaus müssen in diesem Zusammenhang externe Systemkomponenten und Tools betrachtet werden, die aus unterschiedlichen Gründen nicht direkt in das SSMS integriert sondern nur über Schnittstellen angebunden werden können.

Dazu zählt in Bezug auf den Support von SolidWorks in erster Linie das Siebel-Portal, sowie die seitens der Applikations-Hersteller bereitgestellten Systeme, auf deren Nutzung der Support nicht verzichten kann, deren Definition und Gestaltung sich jedoch der Einflussnahme des Systemhauses entzieht. Erstrebenswert ist die Schaffung einer einheitlichen Zugriffsebene für alle Systemkomponenten, einschließlich der externen Systeme mit Hilfe der direkten Verlinkung dieser Tools in das SSMS. Das primäre Ziel ist dabei die Erhöhung der Produktivität durch die Abbildung von Standardprozessen in Workflows und einer möglichst effizienten, verlustarmen Datenübertragung für Serviceanfragen im Rahmen des Bug-Handlings. Da dies aus Gründen fehlender Zugriffsberechtigungen für die Webservices der Hersteller in absehbarer Zeit nicht umsetzbar ist, muss zumindest versucht werden, den Arbeitsablauf bei der Datenübertragung zu harmonisieren. Dies könnte beispielsweise in der Form umgesetzt werden, dass das Aufrufen, Authentifizieren und Anmelden an diese Systemen mit Hilfe einer einzigen Schaltfläche im SSMS erfolgt, anstatt die einzelnen Schritte jedes Mal manuell auszuführen. Eine weitere Möglichkeit zur Harmonisierung bestünde darin, die benötigten Dokumentationen und Daten aus dem SSMS automatisch entsprechend der Anforderungen des jeweiligen Systems formatiert in eine Zwischenablage (Clipboard) zu übertragen, von wo aus sie bequem und schnell in die entsprechenden Felder des Herstellerportals kopiert werden können.

3.7. Zusammenfassung der Anforderungen

Im Folgenden sollen die Anforderungen an ein SSMS zusammengefasst und mit Prioritäten versehen werden, um durch verbesserte Transparenz bezüglich der Verflechtung und Interaktionen zwischen den unterschiedlichen, supportrelevanten Bereichen eines Systemhauses die Ableitung eines Konzeptes zu vereinfachen.

Umfang des SSMS:

- Das SSMS muss modular aufgebaut sein.

Datenmanagement:

- Bestehende Kundendaten müssen konsolidiert, kategorisiert und möglichst verlustfrei in das SSMS migriert werden.
- Die Erfassung neuer Anfragen muss möglichst in Echtzeit erfolgen.
- Für die Erfassung der Anfragen sind geeignete (problem-, implementierungs- und anwendungsspezifische) Kategorien und unterschiedliche Prioritäten zu definieren.
- Jeder CALL muss mit einer alphanumerischen Kennung versehen werden.
- Der aktuelle Bearbeitungsstatus einer Anfrage ist mittels definierter Kennungen zu visualisieren.

Dokumentation von Expertenwissen:

- Anzustreben ist die vollständige Integration einer Wissensdatenbank (KB) in das SSMS.
- Es muss möglich sein, die Quintessenz von fehlerspezifischen CALLs direkt aus dem CALL-Handling-System in die KB zu übernehmen.
- Außerdem müssen Artikel zu bestimmten Themengebieten des Integrations- und Anwendungswissen erfasst werden können. Sinnvoll erscheint in diesem Kontext ein Ansatz, wie er bereits in Form der Wiki-Engines verfolgt wird.

- Für die Durchsuchung der Datenbank muss eine Suchmaschine bereitgestellt werden, welche das Durchsuchen der DB mit hoch differenzierbaren Abfragen ermöglicht.
- Der Kategorisierung von Einträgen muss die identische Struktur wie beim CALL-Handling zu Grunde liegen.
- Eine Verknüpfung mit dem CALL-Handling ist anzustreben. Dabei läuft während der CALL-Erfassung automatisch eine Suchroutine für relevante Artikel zur Kategorisierung des aktuell in Bearbeitung befindlichen CALLs mit und stellt das Suchergebnis in geeigneter Form da.
- Das SSMS muss der Erfüllung dieser Ziele sowohl durch die Abbildung eines Lebenszyklus für Bestandswissen als auch durch die Implementierung von Workflows zur Erfassung, Integration und Pflege neuer Erkenntnisse gerecht werden.

Prozessmanagement:

- Das in das SSMS integrierte Prozessmanagement muss so gestaltet sein, dass Mehrfachbearbeitungen einer Anfrage unmöglich sind. Des Weiteren sind Workflows für:
 - die CALL-Erfassung,
 - das Queing anhand von CALL-Prioritäten und
 - die Formatierung von Anfragen und Dokumentationen gemäß der Anforderungen der zuständigen Entwicklungsteams zu implementieren.
- Das Prozessmanagement muss außerdem fortlaufende, alphanumerische Kennungen für jede Anfrage generieren und diese automatisch sowohl an den Bearbeiter als auch an den jeweiligen Kunden übermitteln.

Bugverifikation:

- Die Verifikation von Bugs muss durch das SSMS als Sonderfall der Anfrage-Bearbeitung innerhalb des allgemeinen CALL-Handling-Workflows abgebildet werden.

- Die Definition einer hinreichenden und umfassend differenzierten Kategorisierung im Hinblick auf die Anforderungen der Entwicklungsteams ist Seitens des SSMS ebenfalls zu realisieren.

Verbesserungsvorschläge:

- Im Rahmen der Workflows für die Dokumentation von VV sind Funktionen zu schaffen, welches es dem Support-Techniker ermöglichen, zu verifizieren, ob ein gleichartiger VV bereits gestellt wurde.
- Das SSMS muss sowohl die Struktur für das Absetzen der VV abbilden als auch die Suchalgorithmen bereitstellen, um bereits formulierte VV zu einem bestimmten Themenbereich zu lokalisieren.

Externe Kommunikation:

- Der Trennung von interner und externer Kommunikation muss das SSMS bei deren Dokumentation durch entsprechende Kennzeichnungen der Korrespondenz Rechnung tragen.
- Der ursprüngliche Ersteller einer Anfrage ist über jede Bearbeitung des CALLs von Dritten automatisch durch das SSMS zu informieren. Diese Funktion ist über eine automatische Versendung einer Notifikation per eMail umzusetzen.

Interne Kommunikation:

- Die Aufgabe des SSMS muss die strikte Kategorisierung der Korrespondenz während der Dokumentation sein.

Qualitätsmanagement:

- Das SSMS muss mittels Workflows die für das QM relevanten Prozesse implementierten und die statistische Auswertungen abgeschlossener Anfragen im Rahmen eines Qualitätsmanagements ermöglichen bzw. unterstützen. Die Definition dieser Prozesse sowie der Qualitätsmerkmale der Softwareprodukte und der Supportdienstleistungen erfolgt im Vorfeld außerhalb des SSMS.

- Zur Beurteilung der Kundenzufriedenheit müssen stichprobenartige Befragungen zur Abwicklung von einzelnen CALLs in Form automatischer eMails erfolgen.
- Das SSMS muss die notwendigen Verifizierungs-Tools beinhalten, um die Einhaltung der Prozessabläufe zu gewährleisten und die Verbesserungspotentiale im Supportprozess zu visualisieren.

Benutzerschnittstelle (GUI):

- Die Bedienungslogik des SSMS muss sich an den Softwarelösungen des MS-Office-Umfeldes anlehnen.
- Das GUI muss mit einer Startmaske einen schnellen und intuitiven schnell Zugriff auf sämtliche Kundeninformationen ermöglichen.
- Das GUI muss einen möglichst aussagekräftigen Überblick über den Status des Kunden in folgenden Bereichen gewähren:
 - Wirtschaftlich relevante Informationen für das Systemhaus.
 - Status-Relevante Informationen zum Kunden.
 - Qualitätsrelevante und subjektive Informationen zum Kunden.
- Das GUI muss eine schnelle, unkomplizierte Initiierung der Prozesse für Dokumentation, Kategorisierung und Vergabe von Prioritäten beim CALL-Handling ermöglichen.
- Der Ergonomie und der Benutzerführung bei der CALL-Erfassung muss im Interesse eines möglichst effizienten Ablaufs eine hohe Priorität eingeräumt werden.

Anforderungen an die Anbindung und Integration des SSMS:

- Gute Integrationsfähigkeit in die vorhandene, WINDOWS-basierende IT-Infrastruktur, z.B. Office-Anwendungen (Outlook, Internet-Explorer / Mozilla-Firefox) oder vorhandene Datenbanksysteme. Ein dem SSMS zugrunde liegendes Datenbanksystem muss aus diesem Grund WINDOWS-kompatibel sein.
- Vorhandene Datenbank-Inhalte müssen mit vertretbarem Aufwand in das SSMS überführt werden können.

- Definition und Programmierung geeigneter Schnittstellen für die Anbindung an bestehende IT-Infrastruktur; Das SSMS muss eine hohe Flexibilität im Bezug auf die Anpassbarkeit an die spezifischen Anforderungen eines Systemhauses in folgenden Bereichen aufweisen:
 - Einfache Definition und Ergänzung von Workflows zur Abbildung von Prozessen.
 - Gestaltung und Modifikation der Zugriffsebene (GUI).
 - Anpassung von Struktur und Inhalt des zugrunde liegenden Datenbank-Systems.
- Die Programmierung des SSMS muss vorzugsweise in etablierten Hochsprachen wie z.B. VBA, SQL, C++, JAVA oder .NET-Standards erfolgen.
- Der Zugriff auf das SSMS muss sowohl über einen per Intranet / LAN mit der Datenbank verbundenen Computer als auch über das Internet möglich sein.
- Das System muss über eine Benutzerverwaltung verfügen.
- Hohe Anforderungen sind an die Gesamtperformance des Systems in Form von kurzen Antwortzeiten und schneller Datenübertragung zu stellen.

Anbindung an vorhandene Systeme zur Unterstützung des Softwaresupports:

- Anzustreben ist eine einheitlichen Zugriffsebene auf alle, auch externe, Systeme mit Hilfe der direkten Verlinkung dieser Tools in das SSMS.
- Ziel ist stets die Erhöhung der Produktivität des Supports durch Integration von Workflows und der möglichst effizienten und verlustarmen Gestaltung der Datenübertragung z.B. für Serviceanfragen im Rahmen des Bug-Handlings. Dazu ist der Workflow bei der Datenübertragung zu harmonisieren.
- Die Daten aus dem SSMS müssen automatisch und gemäß den Anforderungen des jeweiligen Systems geordnet in eine Zwischenablage übertragen werden, von wo aus sie in die entsprechenden Felder des Herstellerportals kopiert werden können.

4. Aktuelle Technologien zur Verbesserung des Software-Supports

Aus der Formulierung der Anforderungen erwächst bereits die Erkenntnis, dass ein SSMS eine gewisse Komplexität haben muss. Um eine Differenzierung zu bestehenden Systemen zur Unterstützung des Software-Supports zu ermöglichen, sollen diese im Anschluss untersucht werden. Durch diese Differenzierung soll anschließend eine geeignete Basis für das SSMS determiniert werden. In erster Linie muss dabei festgestellt werden, ob verfügbare Systeme als Basis zur Umsetzung der Anforderungen geeignet sind oder ob eine individuelle Lösung geschaffen werden muss.

Bei der Betrachtung der im vorausgegangenen Kapitel aufgeführten Forderungen wird deutlich, dass ein Teil der Funktionalitäten mittels kommerzieller Softwaresysteme aus den Bereichen PLM, SLM und CRM umsetzbar sein könnten, da hier ähnliche Aufgabenstellungen gegeben sind. Aus diesem Grund sollen diese Systeme dahingehend untersucht werden, ob sie als Teil des SSMS eingesetzt und in der entsprechenden Richtung erweitert werden können.

4.1. Abgrenzung zu PLM-Systemen

Die Produktfamilie der PLM-Systeme hat sich aus den EDM- / PDM-Systemen entwickelt [2] und wird heute im Zusammenhang mit den unterschiedlichsten Datenmanagementsystemen verwendet. Die Grenzen zwischen den unterschiedlichen Definitionen dieser Systeme und deren Leistungsumfang sind fließend und variieren stark. Sowohl für EDM als auch für PDM existiert eine Vielzahl von Interpretationen von *Electronic Document Management*, *Engineering Data Management* bis hin zu *Product Data Management*. Letztendlich hat sich für die gesamte Produktfamilie der EDM- und PDM-Systeme der Begriff des *Product Lifecycle Managements* etabliert. Da in wachsendem Maße nicht nur die Verwaltung der Produktdaten sondern auch die Abbildung des Produktlebenszyklus innerhalb eines solchen Systems an Bedeutung gewinnt, soll diesem Umstand mit der Integration des Begriffs „Lifecycle“ in die Produktbeschreibung Rechnung getragen werden.

Ein PLM-System ist grundsätzlich als IT-Lösungs-System zu verstehen, mit dem alle Daten, die bei der Entstehung, Lagerhaltung und dem Vertrieb eines Produkts anfallen, einheitlich gespeichert, verwaltet und abgerufen werden. Dabei greifen im Idealfall alle Bereiche bzw. Systeme, die mit einem Produkt in Berührung kommen, auf eine gemeinsame Datenbasis zu: Von der Planung (PPS/ERP) über die Konstruktion (CAD) und der Fertigung (CAM) bis zur Qualitätssicherung, dem Controlling, Vertrieb und Service [16]. Vordergründig betrachtet,

entspricht also die Definition eines solchen Systems den Anforderungen eines SSMS. Die folgende Untersuchung soll hierüber Klarheit verschaffen.

Bei PLM-Systemen findet sich in der Regel ein modularer Aufbau, der auf der obersten Ebene Datenmanagement und Prozessmanagement beinhaltet. Nun ist der Bereich des Datenmanagements bei PLM-Systemen in der Regel stark aufgeprägt und umfassend, da hier vor allen Dingen bei der Verwaltung von 3D-CAD komplex referenzierte Datenstrukturen organisiert werden müssen. Um dabei den Ansprüchen der gegenwärtigen 3D-CAD-Systeme gerecht zu werden, müssen neben den Einzelteil-, Baugruppen- und 2D-Zeichnungsdateien auch deren Abhängigkeiten untereinander abgebildet werden. Zusätzlich können neben den eigentlichen CAD-Daten noch assoziierte Metadaten und nahezu jegliche Art von Dokumenten innerhalb eines geschützten Archivs verwaltet und bei Bedarf aus- und eingechecked werden. Daneben wird der Klassifizierung ein großer Stellenwert eingeräumt, da eine Unterteilung des Dokumentenspektrums eine unabdingbare Voraussetzung für das Durchsuchen und die Wiederverwendung der gespeicherten Dateien und Informationen darstellt.

Darüber hinaus stellt die Unterstützung bzw. Abbildung eines Lebenszyklus für sämtliche mit einem Produkt in Zusammenhang stehenden Dokumente eine zentrale Aufgabe von PLM-Systemen dar. Einen wachsenden Stellenwert nimmt in diesem Zusammenhang auch die Abbildung von Prozessen durch Workflows ein.

Obwohl sich beim Vergleich der Anforderungen an ein SSMS mit den Funktionen eines PLM-Systems zahlreiche Überdeckungen feststellen lassen, treten jedoch auch einige Defizite zu Tage, die gegen die Verwendung eines PLM-Systems für die Umsetzung eines SSMS sprechen. Gegenwärtig verfügen PLM-Systeme über umfangreiche und leistungsfähige Schnittstellen zu den gängigen CAD-Systemen, um die darin erzeugten Daten und deren gegenseitige Referenzen abbilden und verwalten zu können. Die Entwicklung dieser Systeme und die daraus resultierende Funktionalität ist in erster Linie zweifelsohne auf die Bereiche Entwurf, Konstruktion und Fertigung fokussiert. Da im Rahmen des hier betrachteten Software-Supports auch 3D-CAD-Dateien eine gewisse Rolle spielen, mag dies zunächst als Vorteil erscheinen. Die hoch entwickelten Kernfunktionen eines solchen Systems im Bereich des Revisionsmanagements für derartige Dateien werden jedoch im Softwaresupport kaum benötigt. Zudem hat der Großteil der anfallenden Informationen und Daten nicht den Charakter von CAD-Daten, so dass geeignete Schnittstellen erst entwickelt werden müssten. Zwar existieren auch Schnittstellen zu Office-Applikationen wie MS-Word oder MS-Outlook, diese können jedoch den Anforderungen an ein SSMS nicht gerecht werden. Da zudem der Entwicklungsfokus von PLM-Systemen schwerpunktmäßig im Bereich der Handhabung von produkt- und fertigungsrelevanten Daten zu sehen ist, ist eine Weiterentwicklung der Schnittstellen zu Office-Produkten im notwendigen Rahmen nicht zu

erwarten.

Da die erforderlichen Schnittstellen tief in die Strukturen des PLM-Systems eingebettet werden müssen, um eine hinreichende Funktionalität zu gewährleisten, würde dies zudem umfangreiche Eingriffe und Veränderungen am Software-Kern des Systems bedeuten. Die derzeitig verfügbaren PLM-Systeme verfügen jedoch nicht über eine ausreichend offene API, um von dritter Seite solche grundlegende Modifikationen durchführen zu können. Zwar sind derartige Systeme darauf ausgelegt, durch die Bereitstellung entsprechender Programmierschnittstellen eine Anpassung der Funktionalitäten an die individuellen Bedürfnisse der Endkunden zu ermöglichen bzw. zu vereinfachen, diese Schnittstellen bieten jedoch nicht den notwendigen Umfang, um das System auf grundsätzlich andere Problemstellungen wie sie ein SSMS erfordert, anzupassen, da hier andere Algorithmen zu Grunde gelegt werden müssen.

Ein weiterer Punkt, der einer Verwendung von PLM-Systemen als Grundlage eines SSMS entgegensteht, ist die diesen Systemen zugrunde liegende Klassenstruktur. Die vorhandene, hierarchische Struktur enthält entsprechend dem Einsatzgebiet Klassen für CAD-Dokumente wie 2D-Zeichnungen, Baugruppen, Einzelteile, Normteile aber auch Text- und Bilddateien mit Bezug zum Produkt oder dessen Spezifikationen. Diese Hierarchie müsste nahezu komplett durch auf das SSMS zugeschnittene Klassen ersetzt werden, was wiederum tief greifende Veränderungen des Softwarekerns bedingen würde. Eine ähnliche Situation ist im Bereich des Workflow- und Prozess-Managements vorzufinden, da die vorhandenen Strukturen ausschließlich auf den Lebenszyklus von Produktdaten ausgerichtet sind und somit den Anforderungen an ein SSMS nicht entsprechen.

Neben diesen funktionalen Aspekten können aber auch wirtschaftliche Faktoren in diesem Kontext nicht unberücksichtigt bleiben. So sind die Kosten für die Anschaffung und die laufenden Wartungsgebühren für ein PLM-System sehr hoch, zumal ein großer Anteil der bereitgestellten PLM-Funktionen für ein SSMS nicht benötigt wird. Außerdem fließt ein Grossteil der durch die Wartungsgebühren finanzierte Entwicklungsressourcen gerade in solche Funktionen wie das Revisionsmanagement und die Schnittstellen für CAD-Daten, während die Aussichten für die Neu- und Weiterentwicklung für die für ein SSMS relevanten Schnittstellen aufgrund deren geringer Bedeutung für das allgemeine PLM-Klientel gering sind. Die Wirtschaftlichkeit einer Nutzung von PLM als Basis eines SSMS ist somit insbesondere bezüglich des Supports im Umfeld von Midrange-CAD-Anwendungen durch kleine und mittelständische Systemhäuser mehr als fragwürdig.

4.2. Abgrenzung zu CRM-Systemen

Ein wesentlicher Bestandteil der Anforderungen an ein SSMS ist die Verwaltung von Kundendaten und- Transaktionen. Diesen Forderungen wird gegenwärtig im Markt zunehmend mit Hilfe von datenbankgestützten IT-Systemen zur Organisation von Kundenbeziehungen – kurz CRM (**C**ustomer **R**elationship **M**anagement) - Rechnung getragen.

Die Entwicklung von CRM-Systemen resultiert aus den Anforderungen des Marktes die Kundenbindung mittels individualisierter Kundenansprache zu verbessern bzw. zu erhöhen. Zu diesem Zweck werden sämtliche Kundendaten und alle Transaktionen eines Unternehmens mit diesen Kunden in einer Datenbank gespeichert. Die Daten werden so integriert und aufbereitet, dass jeder in eine Transaktion involvierte Mitarbeiter des Unternehmens zu jeder Zeit den notwendigen Zugang zu diesen Daten hat. CRM kann somit als eine übergreifende, IT-gestützte Geschäftsstrategie verstanden werden, deren Hauptziel der Ausbau und die Pflege dauerhafter, profitabler Kundenbeziehungen ist. Die Ziele des CRM decken sich damit weitgehend mit den Zielen eines jeden Unternehmens - mittels CRM soll der Marktanteil eines Unternehmens erhöht und die Kundenzufriedenheit gesteigert werden. Durch die zentrale Erfassung der Daten werden Kosten reduziert und der Service verbessert.

Heutige CRM-Systeme bieten neben den genannten Funktionen eine fein strukturierte Benutzerverwaltung mit den entsprechenden Zugriffsrechten und eine Zugriffsebene über einen Internet-Browser. Bei Betrachtung dieser Funktionen lässt sich eine ganze Reihe von Überdeckungen mit den Anforderungen an ein SSMS feststellen, vor allen Dingen im Bereich der Organisation und der Bereitstellung von Kundeninformationen und- Transaktionen. Damit scheinen CRM-Systeme eine ideale Ausgangsbasis für eine SSMS darzustellen.

Bei einer genauen Betrachtung der Funktionalität wird allerdings deutlich, dass der enorme Fokus auf die hier vorab genannten Punkte in Bezug auf die Anforderungen eines SSMS als Nachteil anzusehen ist, denn die notwendige Verwaltung von 3D-CAD-Daten mit ihren komplexen, gegenseitigen Referenzen ist innerhalb der Klassenstruktur eines CRM nicht vorgesehen. Diese Defizite in der Klassenstruktur von CRM-System ließen sich zwar mittels entsprechender Eingriffe in den Quellcode eliminieren, wobei allerdings der dafür notwendige Aufwand einen erheblichen Kostenaufwand erfordern würde. Selbst unter Vernachlässigung dieses Aspektes stellt sich jedoch die Frage, ob die Vielzahl der innerhalb des SSMS zu verknüpfenden Datenquellen bzw. zu integrierenden Daten allein durch Migration derselben und durch die Konfiguration eines vorhandenen CRM-Systems innerhalb eines wirtschaftlichen Rahmens umsetzbar ist. Die Anbieter dieser Systeme beziehen einen wesentlichen Teil der Wertschöpfung nicht aus dem Vertrieb des Systems, sondern aus dem

Entgelt für die programmiertechnische Anpassung dieser Systeme an die individuellen Anforderungen Ihrer Kunden.

Wie bei den PLM-Systemen sind auch bei den CRM-Systemen die Anschaffungskosten und die notwendigen Anpassungen zu kostenintensiv. Zwar bieten die derzeitigen CRM-Systeme neben umfangreichen Möglichkeiten zur Konfiguration des jeweiligen Systems auch Schnittstellen zu dessen Programmierung an, aber die hierzu bereitgestellten API-Schnittstellen sind in der Regel nicht öffentlich, so dass programmiertechnische Modifikationen nur vom Anbieter des Systems selbst durchgeführt werden können. Dies gilt in gleichem Maße für das, für ein SSMS essentielle Workflow-Management. Sämtliche Workflows müssen durch den Anbieter des CRM-Systems implementiert und bei Bedarf auch modifiziert werden, was den zeitmäßigen, aber auch den kostenmäßigen Rahmen für die Umsetzung schwer kalkulierbar gestalten würde. Neben den entstehenden Kosten würde dies eine erhebliche, langfristige Abhängigkeit des CAD-Systemhauses von den Ressourcen und Kompetenzen eines CRM-Anbieters bedeuten, da jede Änderung am SSMS nur durch den Anbieter durchgeführt werden kann.

Im Fazit bedeutet dies, dass sowohl die Kosten als auch die zu erwartenden, umfangreichen programmiertechnischen Anpassungen gegen den Einsatz eines CRM-Systems sprechen, da hier eine Neuentwicklung eines SSMS für diesen Markt geringere Kosten verursacht und darüber hinaus den gestellten Anforderungen besser entspricht. Diese Betrachtungen machen deutlich, dass aus heutiger Sicht die Modifikation bestehender CRM-Systeme für ein SSMS zwar technisch möglich, aber wirtschaftlich wenig sinnvoll sind.

4.3. Abgrenzung zu Datenbanksystemen

Aus den Anforderungen an ein SSMS lässt sich als zentraler Bestandteil die zentrale, dynamische Speicherung und Verwaltung großer Datenmengen ableiten. Zur Erfüllung einer solchen Funktion haben sich gegenwärtig im Bereich der technischen Datenverarbeitung die relationalen Datenbanken etabliert. Die Ablage der Daten geschieht hier innerhalb einzelner Zellen tabellenartiger Strukturen, wobei die Zellen mit Hilfe von übergeordneten Verknüpfungen miteinander in Beziehung gesetzt werden. Zwar existieren als Organisationsform für Datenbanksystem auch objektorientierte Ansätze, hier hat sich jedoch im Gegensatz zum relationalen Datenbankmodell noch kein Windows-kompatibler Standard etablieren können, so dass diese im Hinblick auf die geforderte Integrationsfähigkeit in eine vorhandene, WINDOWS-basierende IT-Struktur nicht in Frage kommen.

Relationale Datenbanken haben im Zusammenspiel mit den entsprechenden Zugriffsebenen, den Datenbank-Management-Systemen, zusammengefasst die folgenden Eigenschaften:

Die relationalen **Datenbank-Management-Systeme** (RDBMS) sind in der Lage, große Datenmengen zu verwalten und den Zugriff auf die enthaltenen Informationen mit guter Performance zu gewährleisten. Die Trennung von Zugriffsebene und Datenbank wird innerhalb einer Client-Server-Struktur abgebildet, so dass auch Zugriffe über ein Netzwerk möglich sind. Die Zugriffsebene ermöglicht neben dem Abruf der Daten auch eine differenzierte Verwaltung von benutzerspezifischen Zugriffsrechten. Die gegenwärtig etablierten RDBMS sind außerdem in der Lage, den zeitnahen Zugriff mehrerer Benutzer auf den selben Datensatz zu koordinieren, wobei jeweils der erste Zugriff einem Benutzer die Rechte an dem Datensatz reserviert, während die übrigen Anwender nur Leserechte für den Datensatz erhalten (sog. „record locking“). Damit erfüllen RDBMS bereits eine wesentliche Anforderung an ein SSMS, den multiplen Zugriff auf Informationen bei gleichzeitiger Wahrung der Datenintegrität.

Zudem existiert mit der Programmiersprache SQL ein Bestand von Kommandos zur Erstellung, Manipulation und Abfrage von allen gängigen RDBMS. Es stehen auf der Basis von SQL auch objektorientierte Erweiterungen zur Verfügung, so dass dem Ansatz eines Komponenten basierenden Aufbaus eines SSMS entsprochen und zudem die Abbildung eines Datenmodells vereinfacht wird. Trotzdem bleibt festzuhalten, dass alle Operationen auf der Grundlage einer RDBMS auf der Ebene diskreter Datensätze stattfinden müssen, so dass der Aufbau der Domäne komplexer ist, als dies unter Verwendung einer objektorientierten Datenbank der Fall wäre.

4.4. Auswahl der Technologien und Werkzeuge für das neue SSMS-Konzept

Zur Umsetzung des SSMS-Konzeptes kommen eine Reihe von Softwaretools und Programmiersprachen zum Einsatz, welche im Folgenden kurz dargestellt werden sollen. Dabei wird das SSMS-System für die Auswahl der Tools in vier Bereiche untergliedert:

- Clientinterface und GUI
- Webserver
- SSMS-Serversoftware
- Datenbanksystem

Für die Programmierung des Clientinterface und des GUI kommt zweckmäßigerweise ASP.NET zum Einsatz, wobei das GUI als HTML-Seite ausgeführt und die Anwendungslogik in Form von C#-Code hinterlegt wird. Die Programmierung mit ASP bietet dabei gegenüber reinem HTML oder PHP eine Reihe von Vorteilen. Schon in der Bezeichnung ASP (für **A**ctive

Server Pages) kommt zum Ausdruck, das der innerhalb einer ASP-Seite neben dem HTML enthaltene Scriptcode serverseitig ausgeführt und die Seite anschließend an den Client geschickt wird. Dies führt zu einer verbesserten und vom Client-PC unabhängigen Performance. Auf der anderen Seite ermöglicht ASP durch das Code-Behind-Konzept, also den für den User unsichtbar in der ASP-Seite enthaltenen Scriptcode, COM-Objekte aufzurufen und mit Hilfe dieser Objekte beispielsweise Datenbankabfragen auszuführen und die HTML-Seite im Kontext zum Abfrageergebnis zu modifizieren. Mit den genannten Eigenschaften empfiehlt sich ASP.NET damit als beste Wahl für die Programmierung dynamischer Webseiten. Dabei ist ASP.NET eine objektorientierte Programmiersprache, die auf das .NET-Framework von Microsoft und die dazugehörige CLR²⁴ aufsetzt. Theoretisch könnte von daher für den Scriptcode auch eine andere .NET-Sprache, wie z.B. VB.NET verwendet werden, was aufgrund der verbesserten Performance und durchgängigen Objektorientierung von C# jedoch unterbleibt. Die Gestaltung der das GUI bildenden Webseite geschieht jedoch ausschließlich mit Hilfe von HTML.

Für die Kommunikation des SSMS-Servers mit den Clientrechnern via HTTP-Protokoll, also mit Hilfe eines Webbrowsers, wird zunächst ein Webserver verwendet. Dieser muss jedoch in den seltensten Fällen eigens zu diesem Zwecke angeschafft werden, da die Existenz eines solcher Webserver in Systemhäusern der IT-Branche als gegeben angesehen werden kann. Da es neben der Integration in Windows-Server-OS Portierungen von ASP auch für Apache und sonstige gängige Webserver gibt, ist hier eine weitestgehende Flexibilität gewährleistet.

Die unterschiedlichen Module der SSMS-Server-Software sollten in einer objektorientierten Programmiersprache umgesetzt werden, die bei guter Performance möglichst umfangreichen Schnittstellen zu anderen Systemen bietet und dabei auch Mehrfachvererbung unterstützt. Diesen Anforderungen lassen sich am besten mit C++ und C# als .NET-Variante bzw. Nachfolger von C++ umsetzen. Durch die Nähe zur Maschinensprache bietet C eine unübertroffene Performance und durch die starke Verbreitung im Bereich der Applikationsprogrammierung verfügt die überwiegende Zahl von Anwendungen über Schnittstellen oder API in dieser Programmiersprache.

Für die Wahl des Datenbanksystems sind neben SQL auch DBMS wie Oracle, DB2, PostgreSQL oder Microsoft SQL geeignet. Lediglich MySQL soll wegen der fehlenden Möglichkeit zur automatischen Integritätsprüfung nicht eingesetzt werden, während MS-Access durch dessen Größenbeschränkungen und die nur bedingt mögliche Portierbarkeit nicht verwendet werden soll. Die Wahl des Datenbanksystems ist somit nicht auf eine bestimmte Lösung festgelegt, sondern kann unter Berücksichtigung bereits vorhandener

²⁴ CLR = Common Language Runtime, gemeinsame Laufzeitbibliothek der .NET-Sprachen

Server, deren Betriebssysteme und DBMS-Lizenzen erfolgen.

Als Tool für die Datenkomprimierung und -dekomprimierung auf den Clients kommt aufgrund von dessen starker Verbreitung WinRAR zum Einsatz. Dieses Programm kann mit Hilfe von Befehlszeilenparametern ohne Benutzereingriff durch das SSMS ausgeführt werden. Die Spezifikation der zu integrierenden Übersetzungssoftware bedarf einer eingehenden Prüfung und Praxistests, welche den Rahmen dieser Arbeit jedoch sprengen würden.

5. Konzept für ein rechnergestütztes SSMS

Im Folgenden soll nun unter der Berücksichtigung der Anforderungen aus Kapitel 3. das Konzept für ein SSMS erarbeitet werden. Ein SSMS besteht aus verschiedenen Komponenten, welche zwar auf einem einzigen Rechner installiert werden können, aber aufgrund der Natur dieses Systems eher auf unterschiedlichen, miteinander in Verbindung stehenden Computern lokalisiert sind. Aufgabe des Systems ist es, die Tätigkeit der unterschiedlichen Supporttechniker zu organisieren und zu koordinieren. Sämtliche supportrelevanten Tätigkeiten innerhalb eines Systemhauses sollen durch dieses System auf einer einheitlichen Plattform zusammengeführt werden, um alle notwendigen Informationen jedem involvierten Mitarbeiter zur gegebenen Zeit zur Verfügung zu stellen. Bedingt durch die Vielzahl der potentiellen Nutzer, deren räumlicher Trennung und im Sinne einer möglichst effektiven Wart- und Skalierbarkeit muss dieses System modular aufgebaut sein.

5.1. Aufbau des SSMS

Innerhalb der unterschiedlichen Software-Module des SSMS müssen Funktionen realisiert werden, welche zentral auf einem Rechner lokalisiert sind und solche, die von einzelnen Arbeitsplätzen außerhalb des Zentralrechners ausgeführt werden und mit dem Zentralrechner über eine Datenleitung kommunizieren. Eine solche Konstellation wird häufig als Client-Server-Architektur bezeichnet, genau genommen, handelt es sich jedoch, wie im Folgenden aufgezeigt, um eine n-Tier-Architektur (vgl. Abbildung 5.1-1) mit mehreren Schichten:

Präsentationsschicht:

Client: Benutzerschnittstelle, Bereitstellung des GUI, Anmeldung des Users am System und Ausführen von Operationen auf den Daten des Systems.

Webserver: Ermöglicht den Zugriff auf das SSMS über einen Internetbrowser.

Logikschicht:

SSMS-Server-Software: Verarbeitet alle Operationen und koordiniert den Zugriff auf sämtliche Daten. Beinhaltet außerdem die Logik und die Benutzerverwaltung des Systems.

Datenschicht.

Dateisystem (Vault): Dient zur Ablage sämtlicher Applikationsdateien.

Relationale Datenbank (DB): Speicherung sämtlicher supportrelevanter Informationen sowie deren Beziehungen untereinander.

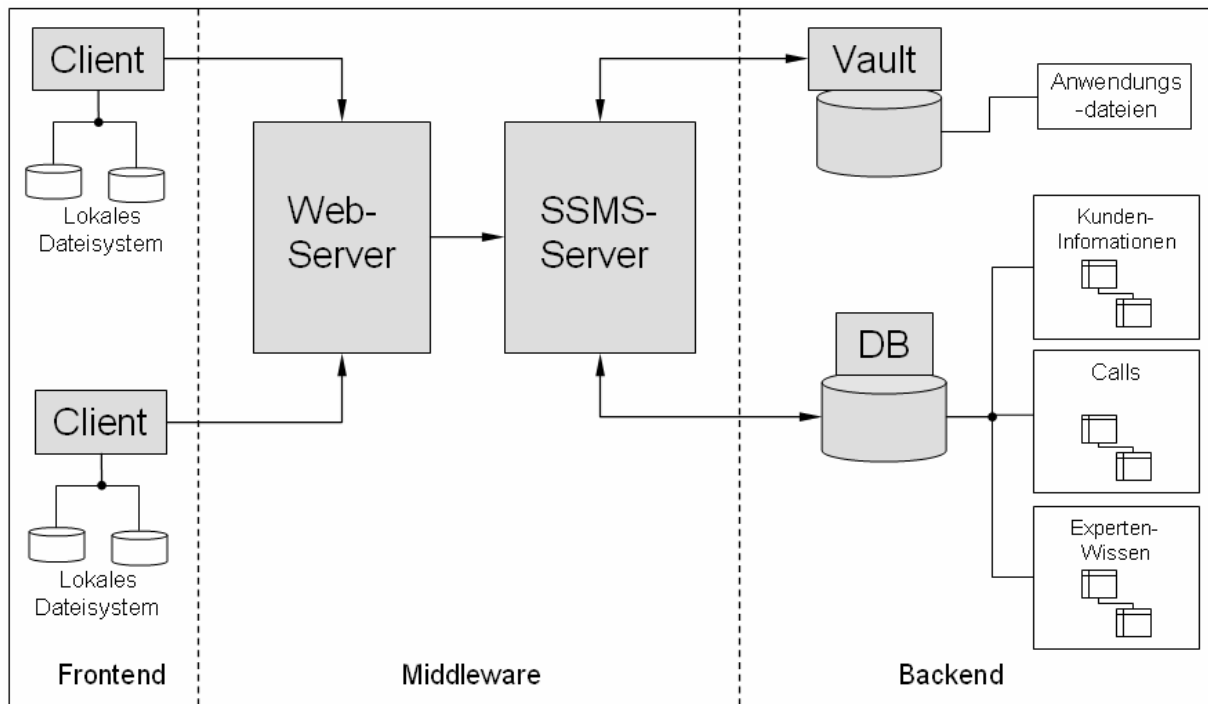


Abbildung 5.1-1: n-Tier-Architektur des SSMS

Im vorliegenden Fall bildet die Präsentationsschicht des Systems ein Userinterface auf dem Client. Dieses Interface wird als Webanwendung gestaltet, um eine größtmögliche Plattform-unabhängigkeit zu gewährleisten, den Installationsaufwand auf den Clientcomputern so gering wie möglich zu gestalten und die Wartbarkeit des Systems zu optimieren. Die Clientsoftware bildet die Schnittstelle zwischen den Anwendern, dem SSMS-Server und den Daten. Über dieses Interface führt der Anwender mit Hilfe der Serversoftware vielschichtige Operationen mit den Daten der Datenschicht aus.

Die SSMS-Server-Software beinhaltet die gesamte Anwendungslogik (Logikschicht) und nimmt somit eine Schlüsselposition ein. Dieses Programm ist der zentrale Kommunikationsknoten des Systems und formuliert aus den Anfragen der Arbeitsplatzrechner die Operationen für den Zugriff auf die Datenschicht.

Die Datenschicht der Architektur wird durch eine Datenbank und ein Dateisystem gebildet. In der Datenschicht werden die vier unterschiedlichen, im Rahmen des Softwaresupports anfallenden Datengruppen gespeichert:

- Kundeninformationen
- CALL-Informationen
- Expertenwissen
- Applikationsdateien

Während Kundendaten, CALL-Informationen und Expertenwissen in der Datenbank gespeichert werden, ist das Dateisystem (Vault) der Ort für die Ablage von anfrage-spezifischen Applikationsdateien, die mit den betreuten Anwendungen im Kontext zu den Kundenanfragen generiert wurden.

Der Zugriff der Clients auf Datenbank und Vault über einen Webserver ermöglicht die Verteilung sämtlicher Systemkomponenten auf unterschiedlichen Computern, was eine gute Performance durch die Verteilung der Ressourcen gewährleistet. Neben den genannten Punkten bietet die n-Tier-Architektur Vorteile bezüglich Wartbarkeit und Skalierbarkeit des Systems, da einzelne Soft- und Hardwarekomponenten bei Bedarf ausgetauscht oder erweitert werden können, ohne das gesamte System verändern zu müssen. Die in diesem Abschnitt dargestellte Architektur des Systems wird in Kapitel 5.5 im Anschluss an die Darstellung der Datenhaltung weiter präzisiert.

5.2. Client-Server-Architektur

Der zentrale Bestandteil des SSMS ist die Serversoftware, welche den Zugriff auf die Daten des SSMS steuert. Der Zugriff geschieht über ein lokales Netzwerk oder das Internet, so dass die Server-Software, die Datenbank und das Vault auf beliebigen Rechnern innerhalb dieses Verbundes lokalisiert sein können (vgl. Abbildung 5.2-1). Die Computer der Anwender verwenden den SSMS-Client, welcher die Kommunikation zwischen Anwender, seinem lokalen Dateisystem und der Server-Software ermöglicht. Prinzipiell dient die Clientsoftware nur zur Eingabe von Abfragen und Manipulation von Daten sowie zur Darstellung der Abfrageergebnisse. Da die Clientsoftware als Webseite gestaltet wird und somit innerhalb eines beliebigen Internetbrowsers dargestellt werden kann, muss zwischen Client und SSMS-Server ein Webserver geschaltet werden. Da ein solcher Web-Server für die Systemhäuser der CAD-Branche aus einer Vielzahl von Gründen kaum entbehrlich ist, kann von dem Vorhandensein eines solchen Servers innerhalb der IT-Infrastruktur eines CAD-Systemhauses ausgegangen werden, so dass dadurch kaum zusätzlicher Kosten- oder Programmieraufwand zu erwarten ist. Die Interpretation der Abfragen und Steuerung von Workflows sowie die dynamische Aktualisierung des Client-Interfaces findet auf Ebene des

SSMS-Servers bzw. des Webservers statt, um die Clientanwendung möglichst schlank und Ressourcen schonend zu halten. Diese Konstellation gewährleistet zudem in Bezug auf den Client-Computer eine weitgehende Unabhängigkeit der Performance des SSMS von der Leistungsfähigkeit und prinzipiell auch dem Betriebssystem der Arbeitsplatzrechner.

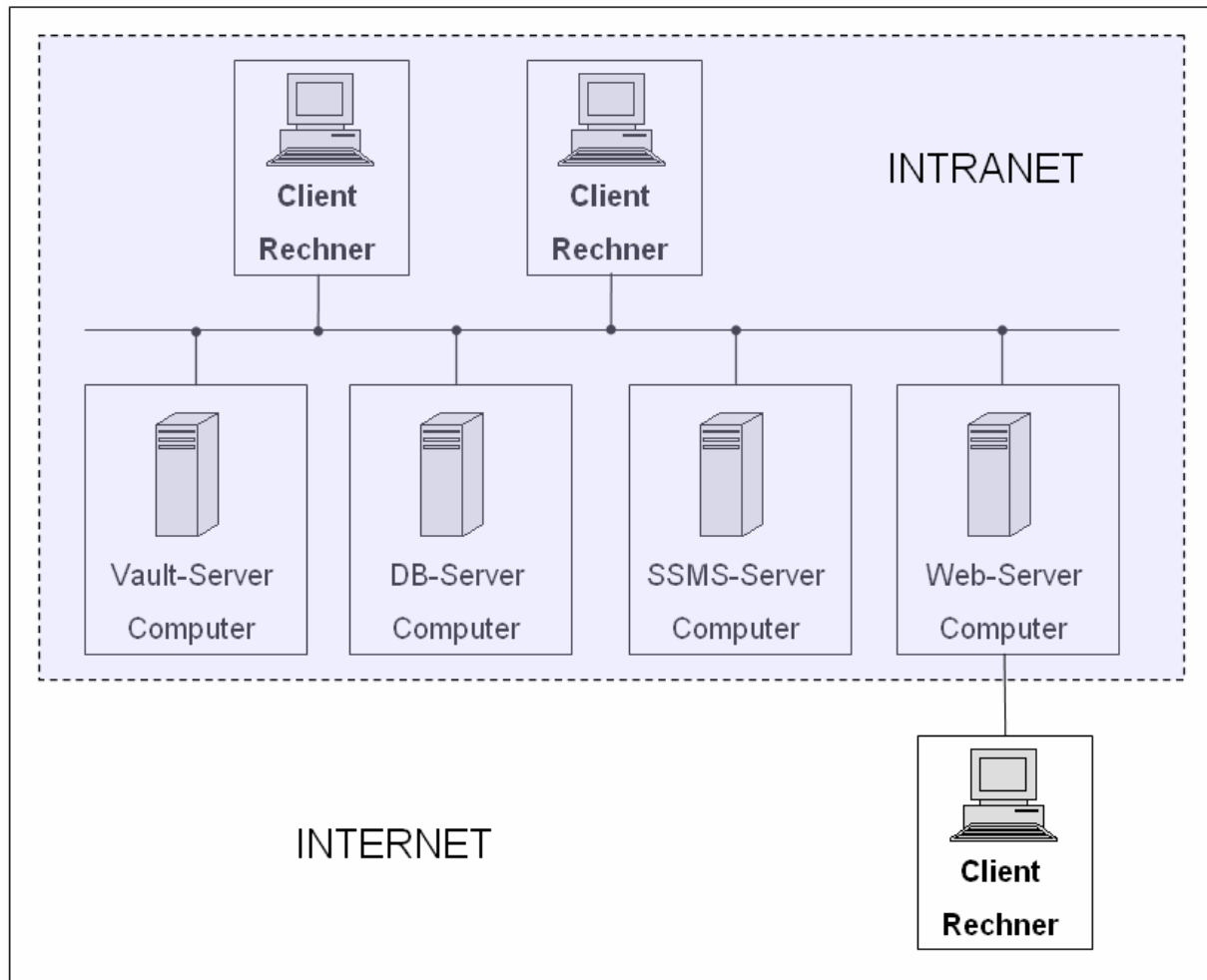


Abbildung 5.2-1: Netzwerkstruktur des SSMS

Die gewählte Architektur bedingt eine ständige Intranet- oder Internetverbindung der Clientrechner, welche zudem über eine geeigneter Geschwindigkeit und Bandbreite verfügen muss. Sie ermöglicht jedoch auch die Erfüllung der Forderung nach einer Zugänglichkeit des SSMS von Orten außerhalb des internen Netzwerkes eines Systemhauses. Aus Gründen der Sicherheit gegen unbefugten Zugriff werden Möglichkeiten zur Bereitstellung einer verschlüsselten Verbindung zwischen Client und Server (SSL²⁵ / VPN²⁶) berücksichtigt.

²⁵ SSL = engl. Secure Sockets Layer, Verschlüsselungsprotokoll für Datenübertragung im Internet

²⁶ VPN = engl. Virtual Private Network, virtuelle Erweiterung eines Netzwerkes, z.B. über das Internet

5.3. Prozessdefinition

Um festzulegen, in welcher Form die geforderten Funktionen in das SSMS implementiert werden, müssen diese Funktionen zunächst in einen Kontext mit den Prozessen innerhalb des Supports gestellt werden. Hierbei werden im Rahmen einer **Prozessdefinition** (PD) die wesentlichen Supportprozesse unter der Annahme dargestellt, das ein SSMS bereits vorhanden und in den Prozess eingebunden ist. Das SSMS selbst wird dabei jedoch stets als eine Art „Blackbox“²⁷ behandelt, deren innere Vorgänge im Rahmen der PD nicht beschrieben werden. Die Beschreibung dieser Vorgänge erfolgt erst in Kapitel 5.4.

Für die PD werden sämtliche Abläufe innerhalb des Supports aus einer prozessorientierten Sicht betrachtet, um später daraus die notwendigen **Funktionen**, **Module** und **Schnittstellen** des SSMS abzuleiten. Bei einer solchen Funktion kann es sich z.B. um das Abspeichern eines CALLs handeln. Einzelfunktionen werden später sinnvoll in Software-Module zusammengefasst. Bei Schnittstellen kann es sich um

- Mensch-Maschine-Schnittstellen (z.B. ein GUI),
- Mensch-Mensch-Schnittstellen (z.B. eine Telefonverbindung zur Übermittlung von verbalen Informationen zwischen Kunden und Support-Techniker) oder
- Maschine-Maschine-Schnittstellen (zwischen verschiedenen System- bzw. Softwarekomponenten)

handeln. In der Prozessdefinition wird also grundsätzlich festgelegt,

- wer (Rollen),
- was (Aufgabe),
- wann (Prozess)
- in welcher Form (Umgebung)

zu erledigen hat. Hierzu werden die unterschiedlichen Arbeitsabläufe, die innerhalb des Softwaresupports wiederholt abgearbeitet werden müssen, strukturiert in Prozessdiagrammen abgebildet. Diese Art der Darstellung ermöglicht eine übersichtliche Abbildung komplexer Abläufe bei gleichzeitiger Gewährleistung einer hinreichend hohen Informationsdichte.

²⁷ Blackbox = engl. „schwarze Kiste“, ein Objekt dessen innerer Aufbau und Funktionsweise unbekannt sind

Ein abzubildender Prozess könnte beispielsweise wie folgt aussehen:

Öffnet ein Benutzer mit Hilfe des Clients einen bestimmten CALL mit Leseberechtigung, so formuliert zunächst der Client daraus eine Anfrage und übermittelt diese an die SSMS-Server-Software. Die Serversoftware verarbeitet die Anfrage des Clients zu einer Datenbankabfrage und generiert aus dem Ergebnis eine HTML-Seite. Die generierte HTML-Seite wird anschließend wieder an den Client übertragen und dort mit Hilfe eines Internetbrowsers dargestellt. Grob betrachtet führt der User die Operation „Lesen“ auf einem definierten Datensatz aus. Neben der Funktion „Lesen“ sind natürlich auch Operationen wie „Bearbeiten“, „Speichern“, „Suchen“ und ähnliche Aktionen in Bezug auf diesen Datensatz denkbar.

Die Prozessdiagramme dienen im Rahmen der Konzepterstellung auch zur Definition von Workflows. Dabei kann ein Workflow als die definierte Abfolge von rechnergestützten Aktivitäten interpretiert werden. Im vorliegenden Fall sollen diese Abläufe Teil einer Prozessteuerung werden, die den User durch die unterschiedlichen Prozesse lenkt und bestimmte, wiederkehrende Funktionen automatisch ausführt. In Bezug auf ein SSMS kann ein solcher Workflow für den Anwender sichtbar durch eine Benutzerführung z.B. bei der Erfassung eines CALLs integriert werden. Workflows können aber auch im Hintergrund des Systems durch bestimmte Ereignisse, so genannte Events, initiiert werden. So könnte z.B. nach der Erfassung einer Anfrage – gesteuert durch die Änderung des CALL-Status zu „Erfasst“ - eine automatische Notifikation per eMail an den entsprechenden Ansprechpartner beim Kunden versendet werden.

In Kapitel 2.3 wurde bereits der allgemeine Ablauf des CALL-Handlings gemäß der gegenwärtigen Praxis abgebildet, was die Komplexität dieses Prozesses gut veranschaulicht. Für die nachfolgende Prozessdefinition gilt es, zunächst die unterschiedlichen Prozesse gegeneinander abzugrenzen und anschließend in einer geeigneten Form grafisch darzustellen. Für die vorliegende PD-Darstellung wird die EPK-Notation nach Becker ff. [17] gewählt (vgl. Abbildung 5.3-1), da diese Darstellung eine hohe Informationsdichte bei einer guten Lesbarkeit ermöglicht. Die Prozess-Diagramme werden von oben nach unten gemäß dem Verlauf der Pfeile gelesen. Die Darstellung eines Prozesses erfolgt dabei als Abfolge von Ereignissen und Funktionen. Verzweigungen sind durch entsprechende Junktoren²⁸ „ODER“ (inklusiv und exklusiv) und „UND“ dargestellt und führen stets zu einem Ereignis. Der Gesamtprozess ist mit Hilfe von Prozessschnittstellen in Unterprozesse aufgeteilt. Komplexe Funktionen sind - wo notwendig - in Einzelfunktionen unterteilt und werden als „verfeinerte Funktion“ dargestellt. Zur besseren Orientierung werden Prozessbeginn und Prozessende mit entsprechenden Symbolen gekennzeichnet und die Prozessschnittstellen

²⁸ Junktors = (von lat. iungere für verknüpfen, verbinden) Logischer Operator

am Anfang und am Ende eines Unterprozesses hervorgehoben. Um in den nachfolgenden Abschnitten einfacher auf Funktionen und Ereignisse innerhalb der Diagramme Bezug nehmen zu können, sind diese durchgehend nummeriert.

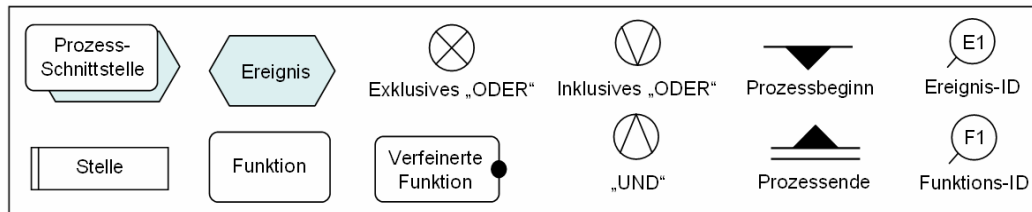


Abbildung 5.3-1: EPK-Legende

Im Folgenden werden nun zunächst die Prozessdiagramme für die Prozesse

- des Anlegens von Kundeninformationen und (vgl. Abbildung 5.3-2)
- beim Verfassen von KB-Artikeln (vgl. Abbildung 5.3-3)

dargestellt, da diese die Voraussetzung für die anschließende Formulierung der Prozesse des

- CALL-Handling (vgl. Abbildung 9.2-1 bis Abbildung 9.2-7)

sind. Um die Übersichtlichkeit des vorliegenden Konzeptes zu verbessern, sind die auf das Anlegen der Kundeninformationen folgenden Prozessdiagramme im Anhang der Arbeit aufgeführt (vgl. Kapitel 9.2).

5.3.1. Erfassung von Kundeninformationen

Das Erfassen, Pflegen und Ergänzen von Kundeninformationen im SSMS ist eine unverzichtbare Voraussetzung für alle kundenspezifischen Prozesse innerhalb des Supports. Dies folgt aus der Bedingung, dass für einen Kunden kein CALL aufgenommen werden kann, solange dessen Informationen nicht in das SSMS eingepflegt worden sind. Die Kundeninformationen geben dem Supporttechniker Auskunft über den Wartungsstatus und das Lizenzaufkommen beim Kunden und sind der Ausgangspunkt für die nachfolgenden Tätigkeiten im Rahmen des CALL-Handlings. Zunächst ist die Migration der bestehenden Kundendaten in das SSMS eine klare Forderung aus Kapitel 3.7. Eine Prozessdefinition ist hier jedoch nicht notwendig, da es sich um einen einmaligen Vorgang handelt, der keine Funktion des SSMS darstellt. Die fortwährende Akquisition neuer Kunden hingegen bedingt sehr wohl die Definition eines Prozesses für die Erfassung der relevanten Informationen über diese Kunden in der Zeit nach der Bestandsdaten-Migration. Die Abbildung 5.3-2 zeigt das

Prozessdiagramm des gesamten Vorgangs „Kundeninformationen erstellen und bearbeiten“ (vgl. F1).

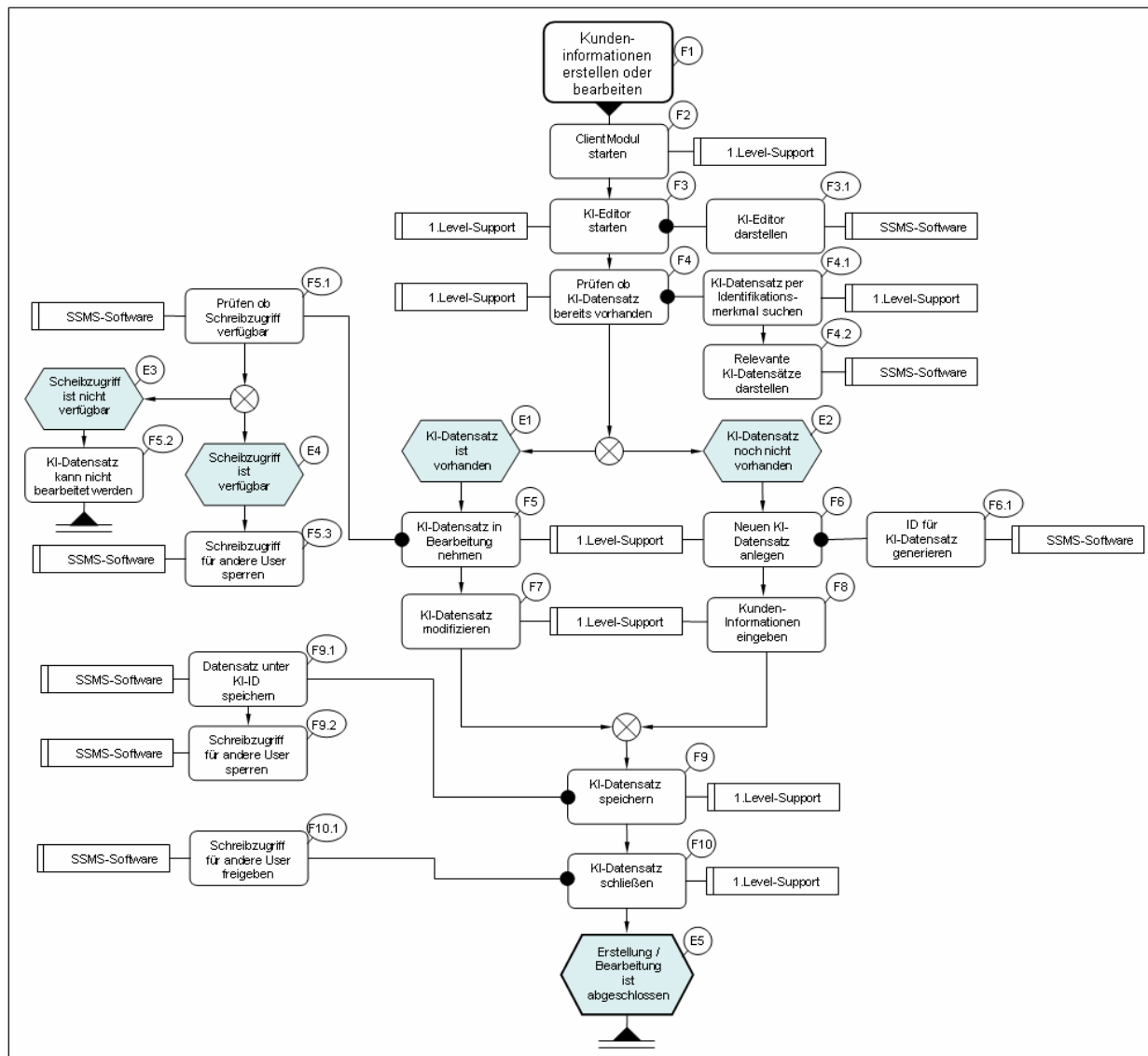


Abbildung 5.3-2: PD für die Erstellung und Bearbeitung von Kundeninformationen

Beispielhaft sollen nun die ersten Schritte des Diagramms (Abbildung 5.3-2) beschrieben werden, um die Interpretation der übrigen PD-Diagramme zu vereinfachen:

Zunächst ruft der Mitarbeiter des 1.Level-Supports (Stelle) das Client-Interface im Browser seines Arbeitsplatz-Rechners (vgl. F2) auf. Von dort aus startet der Anwender (Stelle) den Kunden-informations-Editor (KI-Editor, vgl. F3), die SSMS-Software (Stelle) stellt daraufhin das Interface des KI-Editors im Browser auf dem Client-Rechner dar (vgl. F3.1). Es folgt die Überprüfung des Datenbestandes auf bereits vorhandene Datensätze zu dem Kunden (vgl. F4) durch den Support-Mitarbeiter (Stelle). Die Definition des Prozessschrittes F4 erfolgt als verfeinerte Funktion mit den Unterfunktionen F4.1 und F4.2, welche sequentiell ablaufen.

Der 1.Level-Support (Stelle) durchsucht das SSMS anhand eines eindeutigen Identifikationsmerkmals, z.B. dem Firmennamen (vgl. F4.1), woraufhin das SSMS (Stelle) all jene Datensätze im Client-Interface darstellt, auf die dieses Merkmal zutrifft (F4.2).

Die Überprüfung kann prinzipiell zu zwei Ergebnissen bzw. Ereignissen führen: Ein Datensatz zu dem Kunden ist bereits vorhanden (vgl. E1) oder er ist noch nicht vorhanden (vgl. E2). Da nicht beide Ereignisse zusammen auftreten können, handelt es sich bei der Verzeigung um ein exklusives „*ODER*“ was durch die entsprechende Symbolik verdeutlicht wird. Es wird also nur einer der beiden Prozesszweige weiter verfolgt.

Das Speichern des Datensatzes (vgl. F9) durch den 1.Level-Support (Stelle) im weiteren Verlauf des Schaubildes muss jedoch in jedem Fall erfolgen, so dass vor dem Speichern die beiden Prozesszweige wieder zusammengeführt werden, und zwar ebenfalls mit dem exklusiven „*ODER*“-Junktor, da wie bereits dargelegt, nicht beide Prozesszweige zur gleichen Zeit durchlaufen werden können.

5.3.2. Erfassung und Pflege von Wissensdatenbank-Artikeln

Die Zusammenfassung von Erkenntnissen in der Wissensdatenbank des SSMS ist ebenfalls eine unverzichtbare Voraussetzung für ein erfolgreiches und effizientes CALL-Handling und deshalb auch eine Forderung aus Kapitel 3.7. Das Erfassen von KB-Inhalten kann gemäß der Anforderungsdefinition auf zweierlei Arten erfolgen: Zum einen als Quintessenz fehler-spezifischer CALLs – dieser Prozess ist somit Teil des CALL-Handling-Prozesses, zum zweiten mit dem Verfassen integrations- und anwendungsspezifischer Artikel durch die Supporttechniker des Systemhauses. Dies ist folglich ein vom CALL-Handling entkoppelter, separater Prozess der in der Abbildung 5.3-3 definiert wird.

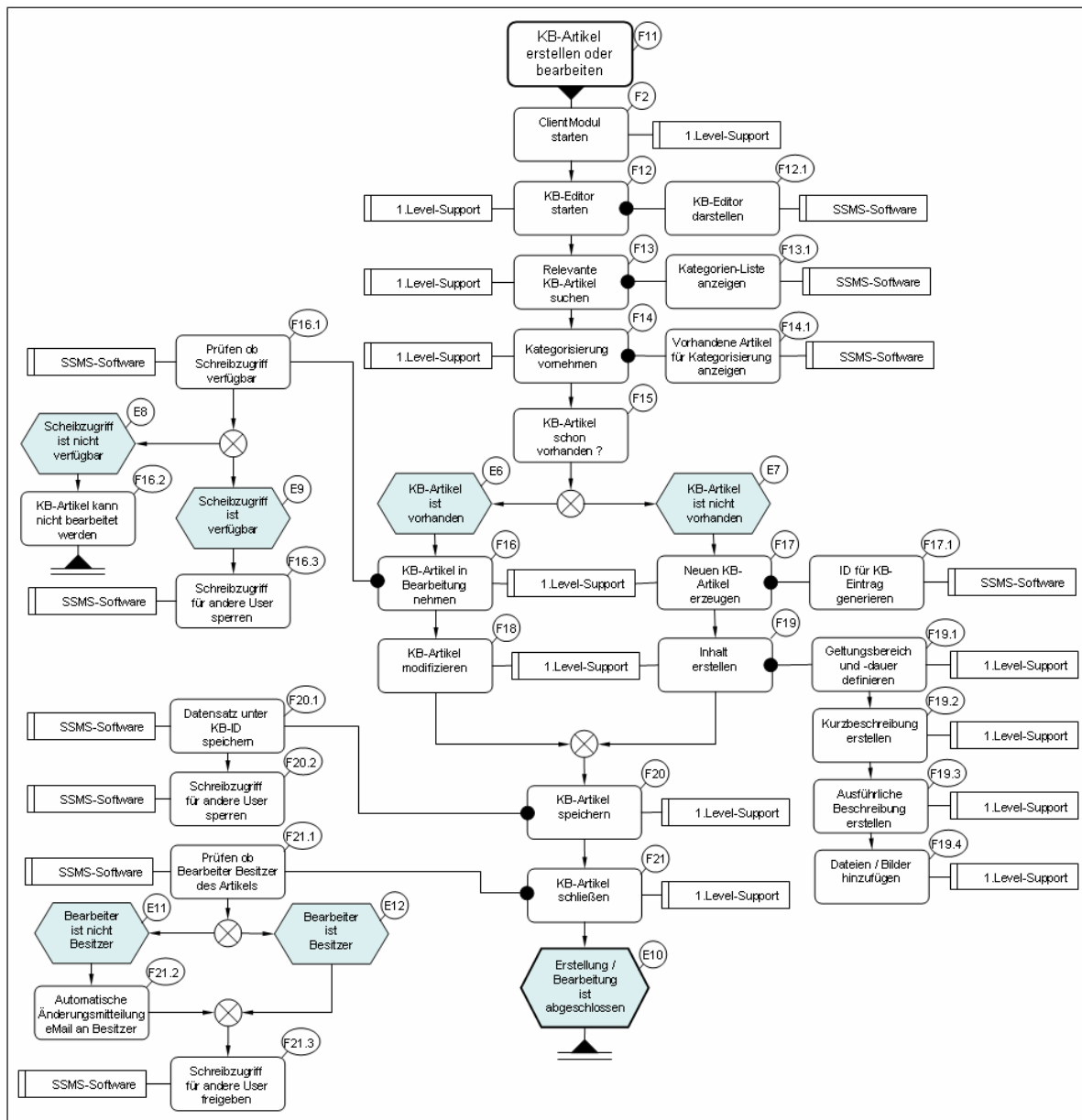


Abbildung 5.3-3: PD für die Erstellung und Bearbeitung von KB-Artikeln

5.3.3. CALL-Handling

Der Prozess CALL-Handling wird für die Darstellung innerhalb des Prozessdiagramms in mehrere Unterprozesse aufgeteilt, um die unterschiedlichen Varianten bezüglich des Ablaufes einer CALL-Bearbeitung abbilden zu können. Der Gesamtprozess des CALL-Handlings ist dazu wie folgt unterteilt (vgl. Abbildung 5.3-4):

- Telefonischen CALL entgegennehmen, (vgl. Abbildung 9.2-1)
- eMail-CALL-Erfassung und Queuing, (vgl. Abbildung 9.2-2)
- neuen CALL erfassen, (vgl. Abbildung 9.2-3)
- CALL bearbeiten (vgl. Abbildung 9.2-4)
- CALL verifizieren, (vgl. Abbildung 9.2-5)
- CALL verfolgen und (vgl. Abbildung 9.2-6)
- CALL-Handling abschließen. (vgl. Abbildung 9.2-7)

Im Folgenden soll der grobe Ablauf innerhalb dieses Gesamtprozesses beschrieben werden. Detaillierte Informationen zu den einzelnen Unterprozessen sind den entsprechenden Abbildungen im Anhang zu entnehmen. Der Gesamtprozess des CALL-Handling beginnt stets mit dem Eingang einer Anfrage per Telefon (vgl. F22) oder per eMail (F41). In beiden Fällen kann bereits zu diesem Zeitpunkt eine Anfrage abgeschlossen werden (vgl. F33), wenn z.B. der Anfrager keinen Anspruch auf Supportdienstleistungen durch das Systemhaus hat. Ist die Anfrage noch nicht im SSMS erfasst worden, so folgt dies im nächsten Schritt (vgl. F40). Im Rahmen der Erfassung werden alle notwendigen Informationen und Daten zum aktuellen CALL gesammelt und für jeden Support-Mitarbeiter zugänglich dokumentiert.

Ist ein CALL erfasst, wird dessen Bearbeitung initiiert (vgl. F35). Hierzu wird zunächst geprüft, in welchem der drei möglichen Zustände („erfasst“, „in Bearbeitung“ und „geschlossen“) sich der CALL gegenwärtig befindet. In Abhängigkeit von diesem Status wird entweder die Verifikation (vgl. F73) veranlasst oder das CALL-Handling abgeschlossen, z.B. wenn der CALL bereits verifiziert und geschlossen wurde oder wenn der CALL gegenwärtig durch einen anderen Support-Mitarbeiter bearbeitet wird.

Die CALL-Verifikation zielt auf die Ermittlung der Problemursache, wobei der CALL zunächst anhand der Dokumentation vom Support-Mitarbeiter nachvollzogen und in Abhängigkeit vom Ergebnis dieser Untersuchung entweder geschlossen oder zur Nachverfolgung (vgl. F84) bestimmt wird.

Hierzu wiederum ein Beispiel:

Hat der Support-Mitarbeiter im Rahmen der Verifikation festgestellt, dass ein Problem auf einen Softwarefehler zurückzuführen ist, so wird dies an den Software-Hersteller übermittelt und der CALL zur Nachverfolgung bestimmt, da die Antwort des Herstellers und somit die Problemlösung noch aussteht. Ist die Lösung für ein Problem, gleich welcher Art, jedoch bekannt, so schließt der Support-Mitarbeiter den CALL und übermittelt die Problemlösung an den Fragesteller. Die Verfolgung einer Anfrage erfolgt durch den Softwarehersteller. In Abhängigkeit von dessen Untersuchungsergebnissen wird der CALL anschließend vom 1.Level-Support wieder in Bearbeitung genommen oder zum Abschluss gebracht (vgl. F33). Der Abschluss-Prozess führt in Abhängigkeit von einem Schlüsselereignisses entweder zur Ablehnung der Anfrage (wenn der Kunde keinen Wartungsvertrag für die Applikation abgeschlossen hat) oder, nach Übermittlung des Untersuchungsergebnisses bzw. der Problemlösung an den Kunden, durch Setzen des CALL-Status auf „geschlossen“ zu deren Schließung.

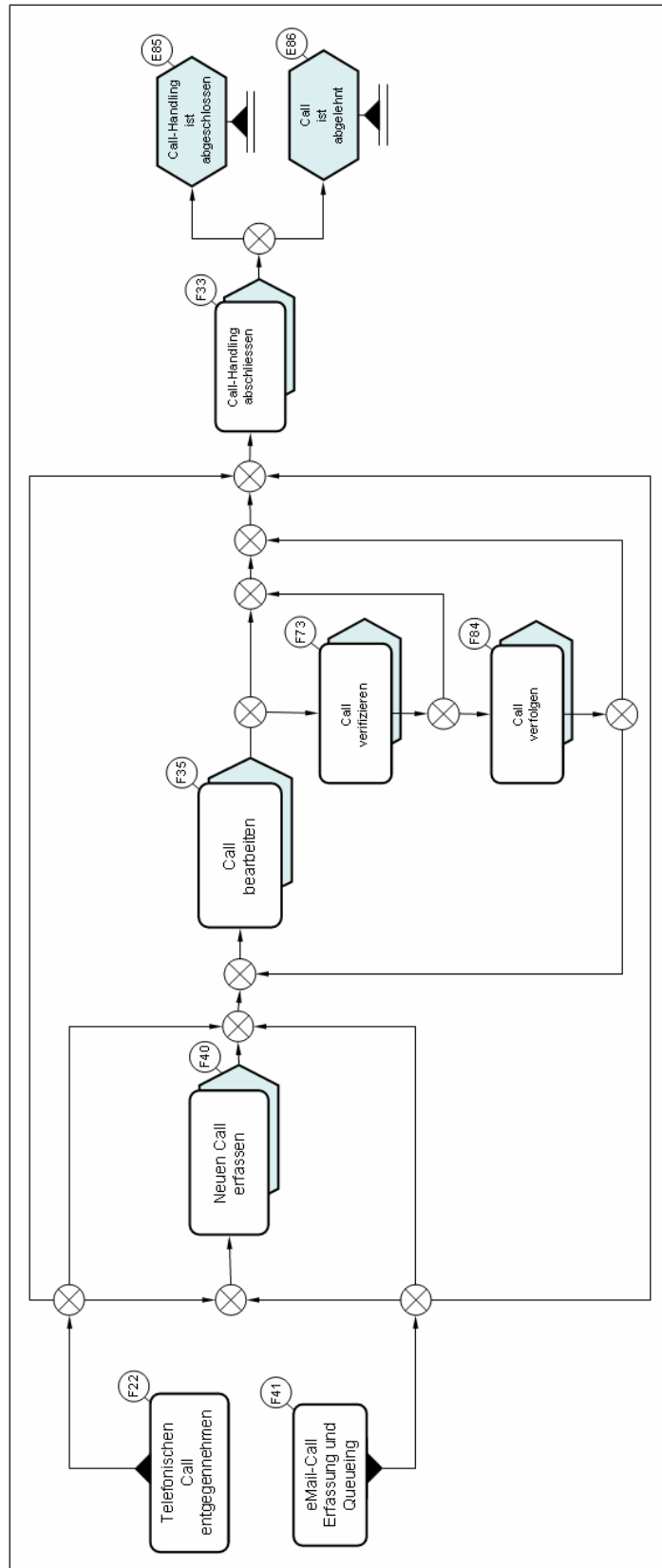


Abbildung 5.3-4: PD Gesamtprozess CALL-Handling

5.4. Datenhaltung und Funktionsdefinition

Die Datenhaltung beschreibt die Bereitstellung und Speicherung sämtlicher, innerhalb des Supportprozesses anfallenden Daten. Hierzu zählen Kunden- und CALL-Informationen ebenso wie Expertenwissen und die Applikationsdateien der betreuten Softwareprodukte. Eine Betrachtung der Prozessdefinitionen des vorherigen Abschnittes unter diesem Aspekt macht deutlich, dass ein wesentlicher Teil der Interaktionen zwischen Anwender und SSMS einen funktionellen Bezug zur Datenhaltung aufweist. Im Folgenden sollen diese Prozesse näher betrachtet und nach Anwendungsfällen gegliedert, dargestellt werden. Für die unterschiedlichen Operationen werden Datenfluss-Diagramme erstellt, um den Weg der Daten durch das System zu beschreiben und schließlich die zu implementierenden Funktionen und Module des SSMS daraus abzuleiten.

Während in den Prozessdiagrammen das gesamte SSMS quasi als Blackbox darstellt wurde, mit welcher der Benutzer zu unterschiedlichen Zeitpunkten des Supportprozesses interagiert, werden die an diesen Schnittstellen zum SSMS ablaufenden Prozesse nunmehr präzisiert. Somit stellt die Definition der Datenhaltung eine Verfeinerung bzw. Fortführung der entsprechenden Positionen innerhalb der Prozessdiagramme dar und ist damit der nächste, logische Schritt zu einem hinreichend konkreten Konzept für das SSMS. Diese Darstellung der Datenhaltung innerhalb des SSMS führt zu diesem Zweck zur Spezifikation einzelner Funktionsgruppen in Form von Pseudomodulen. Diese Pseudomodule repräsentieren lediglich nach funktionalen Gesichtspunkten gruppierte Einzelfunktionen des Systems, welche anschließend zur Definition der zu realisierenden Software-Module herangezogen werden. Die softwaretechnische Umsetzung eines solchen Pseudomoduls in Form von dessen Konkretisierung in einzelne Softwarebausteine wird in Kapitel 5.5 im Anschluss an die Datenhaltung beschrieben.

5.4.1. Kontextbetrachtung und Funktionsbestimmung

Im Rahmen der Kontextbetrachtung wird aus den Prozessdiagrammen (vgl. Kapitel 5.3) zusammenfassend abgeleitet, welche Operationen der User im Verlauf unterschiedlicher Interaktionen mit dem SSMS mit den betroffenen Daten durchführt. Die Prozessdiagramme zeigen, dass die auf den unterschiedlichen Datengruppen ausgeführten Operationen vielfältig sein können. Um für jeden Anwendungsfall die im System ablaufenden Prozesse definieren zu können, sollen zunächst für jede Gruppe von Daten die damit ausführbaren Operationen aufgelistet werden. Unter einer „Daten-Art“ ist in diesem Kontext, ein dem kausalen Zusammenhang nach verwandter Informationstyp, z.B. „Expertenwissen“, zu

verstehen. Wie bereits in Kapitel 5.1 dargestellt, lassen sich vier verschiedene Arten von Daten unterscheiden, die innerhalb eines SSMS verwaltet werden müssen:

- Applikationsdateien
- Kundeninformationen
- Expertenwissen
- CALL-Informationen.

Im Rahmen dieser Verwaltung werden die Applikationsdateien aufgrund ihrer Beschaffenheit als umfangreiche MS-Windows-Dateien innerhalb eines Dateisystems gespeichert, während die übrigen Daten, welche nicht als Windows-Dateien vorliegen, sondern den Charakter von diskreten Datensätzen haben, in einer Datenbank gehalten werden. Die Abbildung 5.4-1 zeigt schematisch die aus den Prozessdiagrammen abgeleiteten Operationen, welche diese Daten verwenden. Das SSMS-System ist als Zusammenfassung aller darin enthaltenen Softwarepakete und deren Module zu verstehen. Zu diesen Paketen zählen:

- Webbrowser und lokales Dateisystem des Clients.
- Der Webserver.
- Die SSMS-Server-Software, bestehend aus Datenbank-Modul (DB-Modul), Wissensdatenbank-Modul (KB-Modul), Notifikations-Modul (NoteModul), Vault-Modul, Clipboard-Modul (ClipModul) und Übersetzungs-Software.
- DBMS und die damit verwaltete Datenbank.
- Das Dateisystem (Vault)

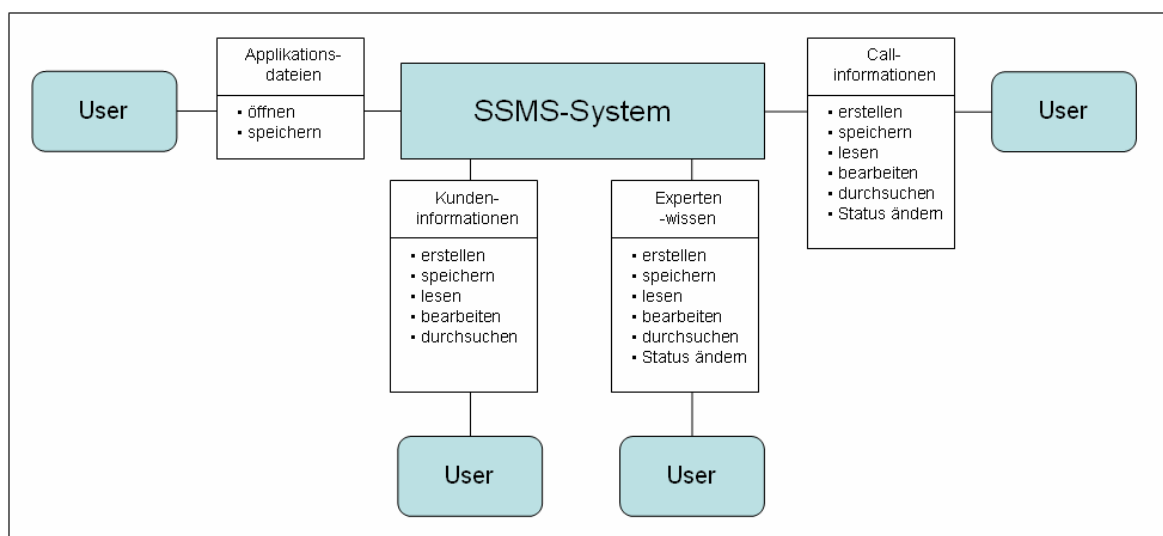


Abbildung 5.4-1: Kontextdiagramm der zu verwaltenden Daten

5.4.2. Datenhaltung von Kundeninformationen

Die Datenhaltung von Kundeninformationen unterscheidet sich grundlegend von der bei Applikationsdateien (vgl. Kapitel 5.4.5) angewendeten Methode, da hier nur ein Austausch von Datensätzen zwischen Client-Interface und Datenbank stattfindet. Die Kundeninformationen sind der zentrale Ausgangspunkt für alle supportrelevanten Tätigkeiten des Systemhauses, ihr Inhalt entscheidet schon zu Beginn des CALL-Handlings darüber, ob ein CALL überhaupt bearbeitet werden darf (vgl. F32 in Abbildung 9.2-1 und in Abbildung 9.2-2). Ferner kann ein CALL nur erfasst werden, wenn der entsprechende Kundendatensatz im System vorhanden ist (vgl. F30 in Abbildung 9.2-1 und in Abbildung 9.2-2). Zwar ist die Erfassung von Kundendatensätzen nicht die primäre Aufgabe des Supports, sie kann jedoch aus den bereits genannten Gründen nicht aus dem vorliegenden Konzept ausgeklammert werden. Aus der Abbildung 5.4-1 gehen die grundsätzlich ausführbaren Operationen für die Kundeninformationen hervor, dazu gehören:

- Erstmaliges Erfassen von Kundeninformationen mit Hilfe einer geeigneten Benutzerschnittstelle.
- Bearbeiten von Kundendatensätzen.
- Speichern von Kundeninformationen in der Datenbank des SSMS.
- Durchsuchen der Datenbank nach Kundeninformationen anhand von Identifikationsmerkmalen und Darstellen der Suchergebnisse.

In diesem Kapitel werden zunächst die Datensichten der Bearbeitung und das Speichern von Kundendaten beschrieben. Dabei befasst sich Kapitel 5.4.2.1 mit der Bereitstellung der Kundeninformationen zum Zwecke der Bearbeitung, während der darauf folgende Speichervorgang in Kapitel 5.4.2.2 behandelt wird.

Die eigentliche Erfassung im Sinne einer manuellen Eingabe von Kundendaten in eine Maske des GUI wird in Kapitel 5.6 in Zusammenhang mit der Beschreibung des SSMS-Clients behandelt. Dies ist deshalb sinnvoll, da die Erfassung primär eine Frage des Umfangs der zu speichernden Informationen und der Gestaltung der Eingabemaske an sich ist.

5.4.2.1. Bearbeiten von Kundendatensätzen

Die Bearbeitung von Kundendatensätzen wird immer dann notwendig, wenn Kundeninformationen geändert oder ergänzt werden müssen, etwa wenn der Lizenzpool eines Kunden erweitert wird oder neue Mitarbeiter zum CAD-Anwenderteam eines Kunden hinzukommen (vgl. E15 in Abbildung 9.2-1 und in Abbildung 9.2-2). Zunächst muss vor der eigentlichen Bearbeitung jedoch der entsprechende Datensatz aus dem SSMS aufgerufen werden. Da bei einem Anruf oder einer eMail-Anfrage eines dem SSMS unbekanntem Ansprechpartners die Kundendaten von dessen Arbeitgeber (also dem Kunden) nicht automatisch aufgerufen werden können, muss die Suche nach dem zugehörigen Datenstamm manuell erfolgen. Zudem muss der Datensatz des Ansprechpartners zunächst zum Kundendatensatz ergänzt werden, bevor der CALL im SSMS überhaupt erfasst werden kann (vgl. F31 in Abbildung 9.2-1 bzw. F49 in Abbildung 9.2-2). Um also den Mitarbeitern eines Kunden jederzeit optimalen Support bieten zu können und die Konsistenz der Kundendaten im Systemhaus zu gewährleisten, muss das SSMS Methoden bereitstellen, die es den Support-Mitarbeitern ermöglichen, diese Informationen effizient im System ergänzen und pflegen zu können.

Zum Zwecke der Bearbeitung sucht der Support-Mitarbeiter zunächst manuell den Stammdatensatz des Kunden anhand eines eindeutigen Identifikationsmerkmals, z.B. der Firmierung oder einer Lizenznummer und ruft diesen auf (vgl. Kapitel 5.4.2.3). Anschließend initiiert er die Bearbeitung des Kundendatensatzes. Da innerhalb einer Datenbank ein Datensatz stets nur von einem einzelnen User bearbeitet werden kann, sendet der Client nach dieser Initiierung die Information über die Bearbeitung an das DBMS²⁹, so dass dieses den Datensatz für die Bearbeitung durch andere User sperrt. Der Support-Mitarbeiter nutzt für das Aufrufen und Editieren des Datensatzes ein spezielles Subinterface, den Kundeninformations-Editor (KI-Editor, vgl. F3 in Abbildung 5.3-2), innerhalb des Client-Interfaces des SSMS (vgl. Abbildung 5.4-2). Um einen Ansprechpartner hinzuzufügen oder Informationen zu diesem zu ergänzen, initiiert der User diesen Vorgang mit Hilfe der entsprechenden Funktion des SSMS (z.B. „Kontakt hinzufügen“ bzw. „Kundendatensatz bearbeiten“). Sobald der Support-Mitarbeiter die notwendigen Informationen eingegeben hat, initiiert er den Speichervorgang (vgl. Kapitel 5.4.2.2) und schließt die Eingabemaske. Anschließend hebt das DBMS den exklusiven Schreibschutz für den Kundendatensatz auf und der Bearbeitungsvorgang ist abgeschlossen.

²⁹ DBMS = Datenbank-Management-System, Zugriffsebene einer Datenbank

5.4.2.2. Speichern von Kundeninformationen in der Datenbank des SSMS-Systems

Die Betrachtung des Speicherns von Kundeninformationen beginnt mit der Initiierung des Speichervorgangs durch den Anwender (vgl. Abbildung 5.4-2). Nach dem Abschluss der Eingabe über die entsprechende Funktion des Clients wird die Webseite mit der Eingabemaske an den Webserver übermittelt und die enthaltenen Informationen dort einer Plausibilitätsprüfung unterzogen. Im Rahmen dieser Prüfung wird ermittelt, ob alle notwendigen Informationen angegeben wurden. Bei Unvollständigkeit wird eine Meldung an den User übermittelt, mit der Aufforderung, sämtliche Pflichtangaben auszufüllen. Diese Prüfung ist unumgänglich, um die Generierung einer konsistenten Datenbasis zu gewährleisten.

Aus dem gleichen Grund bleibt der Kundendatensatz für die Bearbeitung durch andere SSMS-Anwender vom das System gesperrt (vgl. F5.1 und F9.2 in Abbildung 5.3-2). Die Plausibilitätsprüfung wird anschließend wiederholt und erst nach deren Erfolg werden die Daten in Form eines DataSets³⁰ (z.B. eines Objektes, oder einer XML³¹-Datei) an das Datenbank-Modul übergeben. Das DB-Modul generiert nun aus dem DataSet ein SQL-Statement und sendet dieses an die Datenbank, wo die Kundeninformationen unter einer eindeutigen Kennung (Kunden-ID), sowie Datum und Uhrzeit gespeichert werden.

³⁰ DataSet = Eine im Hauptspeicher verwaltete zusammengehörige Datenmenge in tabellarischer Struktur

³¹ XML = Extensible Markup Language, engl. Erweiterbare Auszeichnungssprache

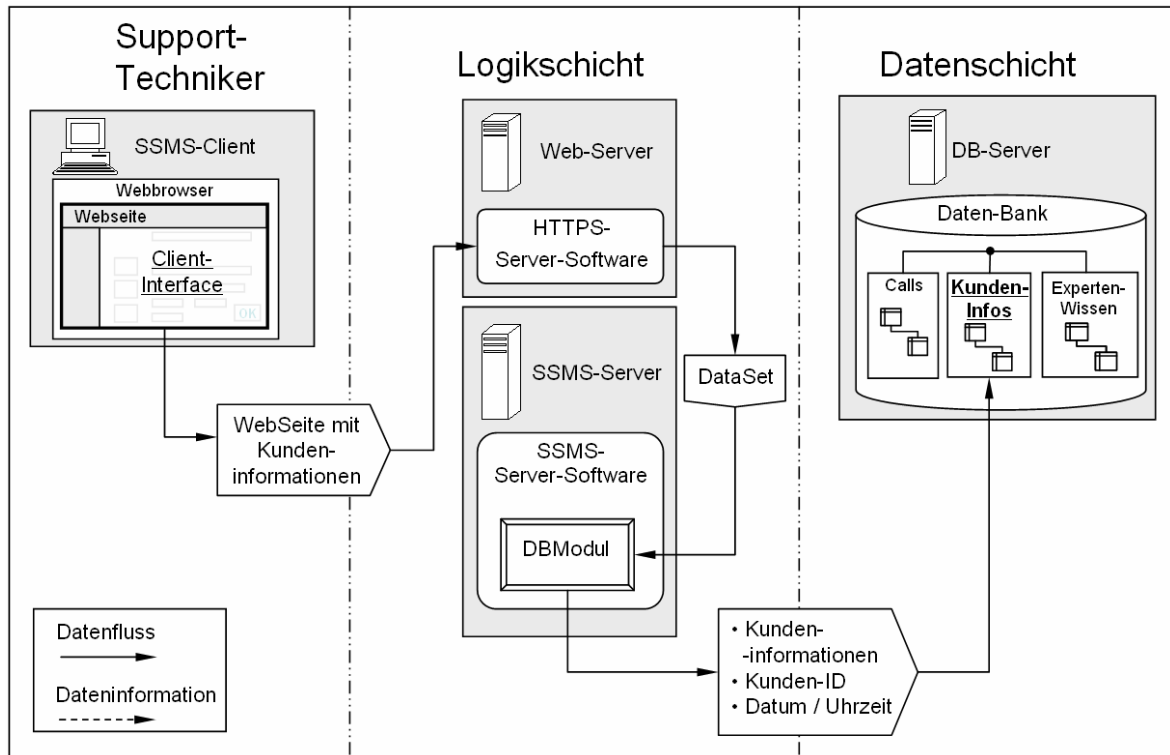


Abbildung 5.4-2: Datensicht des Speicherns von Kundeninformationen

5.4.2.3. Durchsuchen der Datenbank nach Kundeninformationen und Darstellen der Suchergebnisse

Das Durchsuchen der Datenbank nach Kundeninformationen kann aufgrund der dem SSMS zugrunde liegenden Datenbankstruktur prinzipiell über sämtliche im Datensatz hinterlegten Attribute und deren Kombinationen erfolgen (vgl. Abbildung 5.4-3). In nachfolgenden Suchläufen kann das Suchergebnis durch Angaben von zusätzlich Identifikationsmerkmalen zusätzlich noch weiter eingeschränkt werden. Zu den eindeutigen Identifikationsmerkmalen zählt beispielsweise die Telefonnummer (vgl. F25 in Abbildung 9.2-1) oder die eMail-Adresse (vgl. F44 in Abbildung 9.2-2) eines Ansprechpartners bei einem spezifischen Kunden. Das SSMS durchsucht im Falle einer Kundenanfrage die Datenbank zunächst anhand dieser, durch die Telefonanlage bzw. den eMail-Client übermittelten Informationen nach dem zugehörigen Kundendatensatz und stellt diesen anschließend im Client-Interface dar.

Kann aufgrund der dezentralen Supportstruktur des Systemhauses die Übermittlung der Telefonnummer durch die Telefonanlage nicht erfolgen (z.B. weil der CALL vom Standort der Telefonanlage aus über eine einheitliche Nummer an den Standort des Clients weitergeleitet wurde) oder fehlen entsprechende Informationen zu diesen Merkmalen in der Datenbank (z.B. weil die eMail-Adresse eines Ansprechpartners bis dato noch nicht bekannt ist), so muss der Supporttechniker den Suchvorgang mit Hilfe von anderen, eindeutigen Merkmalen, wie z.B. Vor- und Nachnamen, manuell initiieren.

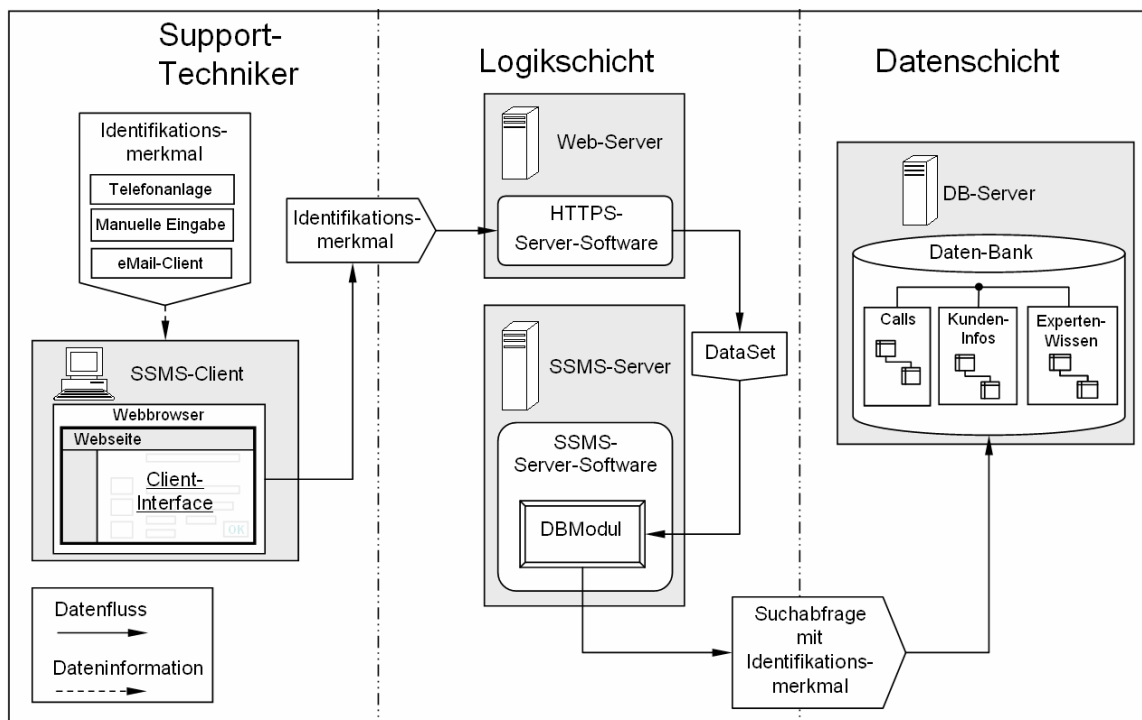


Abbildung 5.4-3: Datensicht des Durchsuchens der Datenbank nach Kundeninformationen

Für die Datensicht des Suchvorgangs ist es zunächst nicht relevant, ob die Suchanfrage über den Client manuell oder durch Übergabe von Merkmalen mit Hilfe der Telefonanlage oder des eMail-Clients automatisch gestartet wird. Festzuhalten bleibt an dieser Stelle, dass für die eMail-CALL-Erfassung und das anschließende Queuing eine Schnittstelle für das Clientinterface geschaffen wird, welcher die automatische Erkennung des Adressaten einer im Hotline-Postfach eintreffenden eMail beinhaltet. Ferner muss eine Schnittstelle zur Telefonanlage geschaffen werden, welche die über die zentrale Hotline-Nummer eingehenden Anrufnummern an den SSMS-Client des jeweiligen Hotlinemitarbeiters übergibt.

Nachdem ein Identifikationsmerkmal an den SSMS-Client übergeben wurde, wird diese Information über das Webinterface bzw. die das Interface bildende Webseite an den HTTPS-Server übermittelt. Dieser wandelt die Anfrage wiederum in ein DataSet um und übergibt dieses anschließend an das DB-Modul. Das DB-Modul formuliert aus dem DataSet die eigentliche Datenbankabfrage über das Attribut und dessen Wert in Form eines SQL-Statements und schickt dieses an die Datenbank. Das Datenbanksystem identifiziert nun jenen Datensatz, in welchem der Wert des entsprechenden Attributs mit dem Identifikationsmerkmal übereinstimmt.

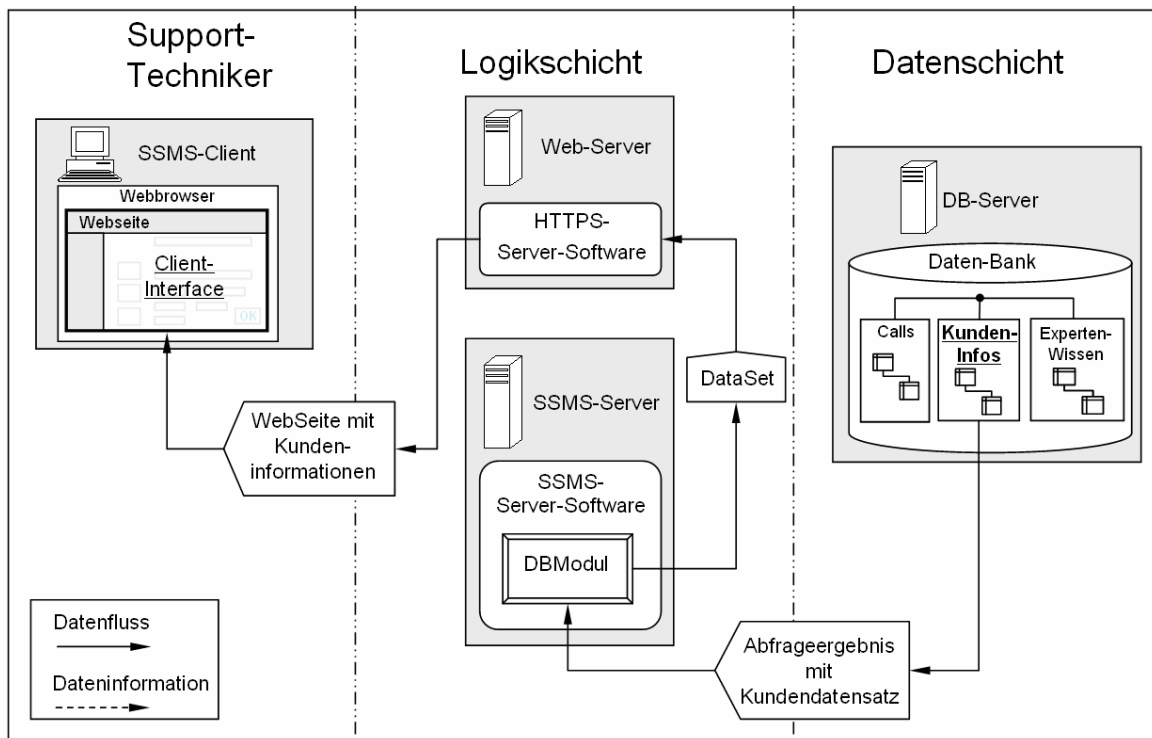


Abbildung 5.4-4: Datensicht der Darstellung von Abfrageergebnissen für Kundeninformationen

Das Ergebnis der Abfrage wird vom Datenbanksystem an das DB-Modul übergeben (vgl. Abbildung 5.4-4), welches aus dem Inhalt des Abfrageergebnisses nun wiederum ein DataSet formuliert und dieses an den Webserver übermittelt. Hier wird der Inhalt des DataSets wieder in die Webseite des Client-Interfaces integriert und an den SSMS-Client übermittelt, wo sie dem User innerhalb des Internetbrowsers dargestellt wird.

5.4.3. Datenhaltung von CALL-Informationen

Die Erfassung und Haltung von CALL-Informationen zählt zu den Kernfunktionen des SSMS. Im Rahmen der Prozessdefinition wurden die hierbei durch das SSMS bereitzustellenden Funktionalitäten ermittelt und dargestellt (vgl. Abbildung 9.2-3 und Abbildung 9.2-4). In der Abbildung 5.4-1 sind die aus der Prozessdefinition abgeleiteten, mit dieser Gruppe von Daten möglichen Operationen zusammengefasst. Hierzu zählen:

- Die Erstellung und das Abspeichern der CALL-Informationen in der Datenbank des SSMS.
- Das Durchsuchen des SSMS nach Informationen zu einem vorhandenen CALL.
- Das Öffnen und Bearbeiten von vorhandenen CALLs.
- Die Statusänderungen von CALLs.

Zwar kann das Abspeichern von zu einem spezifischen CALL gehörenden Applikationsdateien auch als Teil des CALL-Handlings betrachtet werden, dieser Vorgang ist jedoch im vorliegenden Zusammenhang nur von sekundärer Bedeutung und wird in Kapitel 5.4.5 im Rahmen der allgemeinen Datenhaltung von Applikationsdateien behandelt.

5.4.3.1. Erstellung und Abspeichern von CALL-Informationen in der Datenbank des SSMS

In der Prozessdefinition wurde dargelegt, dass die Erstellung von CALL-Informationen das Vorhandensein eines entsprechenden Kundendatensatzes mit dem Subdatensatz des Ansprechpartners beim Kunden voraussetzt (vgl. F31 in Abbildung 9.2-1 und in Abbildung 9.2-2). Im Falle eines CALLs wird also zunächst der entsprechende Datensatz des Ansprechpartners eines Kunden aufgerufen (vgl. Abbildung 5.4-4), damit der CALL diesem zugeordnet werden kann.

Innerhalb der Darstellung des Kundendatensatzes werden die Kategorisierungen und Beschreibungen von sämtlichen offenen Anfragen dargestellt, damit sich der Support-Mitarbeiter schnell einen Überblick über die derzeit in Bearbeitung befindlichen CALLs des Kunden verschaffen kann. Dies ist notwendig, um Mehrfachbearbeitungen von Anfragen durch unterschiedliche Supporttechniker zu vermeiden, da es dem Support-Mitarbeiter so möglich wird zu ermitteln, ob eine gleichartige Anfrage des Kunden bereits bearbeitet wird und wie der derzeitige Bearbeitungsstand ist.

Ist der CALL gegenwärtig noch nicht erfasst, so initiiert der Support-Mitarbeiter die CALL-Erfassung für den aktuellen Ansprechpartner. Das SSMS stellt daraufhin das CALL-Handling-Interface im Webbrowser des Clients dar (vgl. Abbildung 5.4-5 und F54.1 in Abbildung 9.2-3). Das SSMS generiert zudem eine eindeutige CALL-Nummer (CALL-ID) für die Anfrage. Nachdem der Support-Techniker den CALL entsprechend der Dringlichkeit des Problems priorisiert hat, wird innerhalb des Interfaces ein kaskadierendes Menu mit Kategorisierungslisten dargestellt. Der Support-Techniker kategorisiert nun mit Hilfe dieser Liste das Problem des Ansprechpartners nach Haupt- und Unterkategorie. Anschließend wird das Problem vom Anwender mit einer Kurzbeschreibung versehen und in einem weiteren Schritt mit einer stichwortartigen, sequentiellen Reproduktionsanleitung dokumentiert.

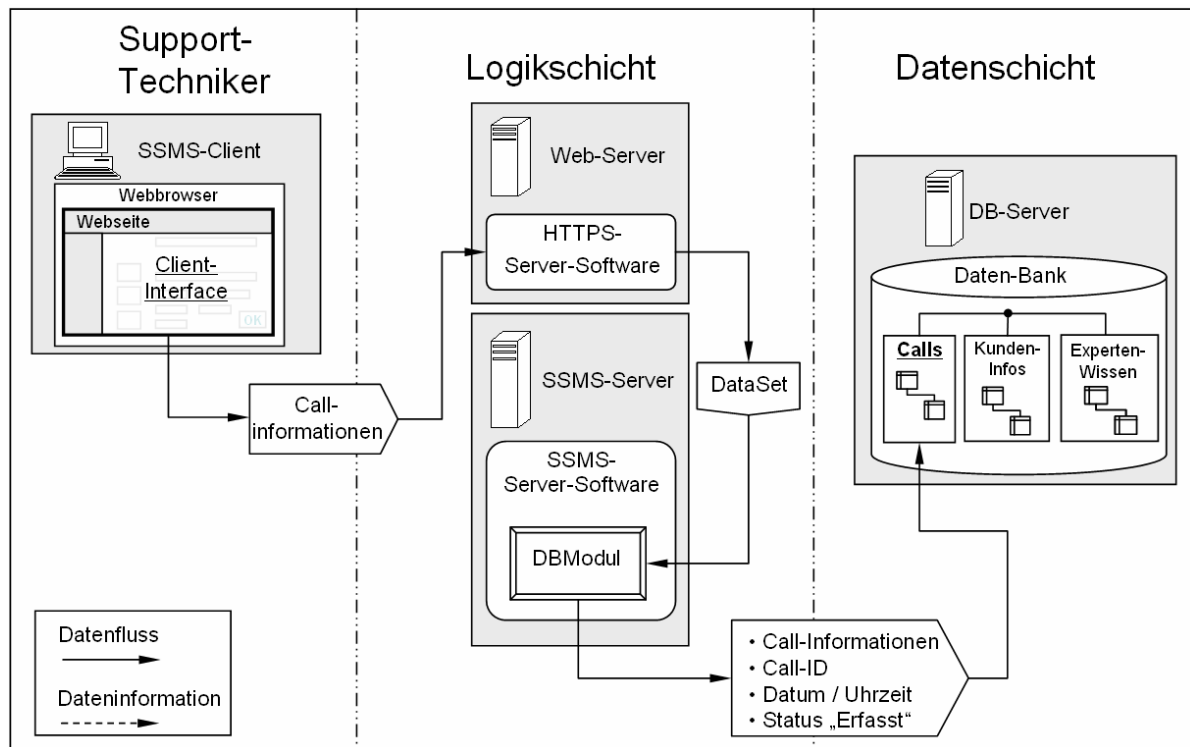


Abbildung 5.4-5: Datensicht des Erfassens und Speicherns von CALL-Informationen

Parallel zu diesem Vorgang stellt das Wissensmanagementsystem des SSMS die bereits dokumentierte Problemfälle und vorhandene Artikel der selben Haupt- und Unterkategorie im Client-Interface dar, so dass der Supporttechniker schon zu diesem frühen Zeitpunkt die Chance hat, nach dokumentierten Lösungen zum aktuellen Problemfall zu suchen. Ist eine Lösung zu dem Problem in der Wissensdatenbank dokumentiert, so kann diese umgehend an den Kunden übermittelt und der CALL unter Verweis auf den KB-Eintrag abgeschlossen werden. Die Datensicht dieses Vorgangs wird im Rahmen von Kapitel 5.4.4 im Zusammenhang mit der Datenhaltung von Expertenwissen noch näher erläutert.

Ist die Lösung für das geschilderte Problem unbekannt, so muss der Supporttechniker zum Zwecke der Reproduzierbarkeit eine Reihe von Informationen über die verwendete Systemkonfiguration vom Ansprechpartner sammeln und zum aktuellen CALL hinzufügen. Das SSMS unterstützt den Supporttechniker bei dieser Aufgabe mit Hilfe einer der gewählten Problemkategorie entsprechenden, spezifischen Eingabemaske mit den abzufragenden Informationen. Im Gegensatz zur Priorität und Kategorie des Problems sind diese Angaben jedoch keine Pflichtangaben und können auch nachträglich noch zu einem CALL hinzugefügt werden.

Ist die Problemaufnahme abgeschlossen, so initiiert der Support-Techniker den Speichervorgang für den aktuellen CALL. Die das Client-Interface bildende Webseite mit den Angaben zum CALL wird daraufhin an den Webserver übermittelt. Dieser unterzieht die enthaltenen Daten einer Plausibilitätsprüfung und übergibt wiederum die CALL-Informationen

in Form eines DataSets an das DB-Modul. Das Datenbank-Modul formuliert aus dem DataSet ein entsprechendes SQL-Statement und sendet dieses an die Datenbank. Dort werden die Informationen schließlich unter der CALL-Nummer (CALL-ID) gespeichert. Da es sich um einen neuen CALL handelt, wird dieser mit dem Status „Erfasst“ versehen.

5.4.3.2. Durchsuchen des SSMS nach Informationen zu einem vorhandenen CALL

Nachdem der CALL in der Datenbank gespeichert ist, beginnt für den Support-Mitarbeiter die eigentliche Arbeit der Reproduktion und Lösungsfindung. Sollte es sich bei dem Problem um einen Fehler im Quellcode der Applikation handeln, so ist es sinnvoll, dass bei der Übermittlung der Problembeschreibung an den Hersteller auch Bezug auf andere Anfragen zu demselben Problem genommen wird. Da innerhalb der Entwicklungsabteilungen von Softwareherstellern die Prioritäten bei der Lösung von Bugs stark durch die Häufigkeit beeinflusst werden, mit denen diese gemeldet werden, ist der zuvor beschriebene Bezug im Hinblick auf eine schnellstmögliche Lösung des Problems von erheblicher Bedeutung.

Da bei der Erstellung eines neuen CALLs die bereits abgeschlossenen, gleichartigen CALLs simultan vom Wissensmanagementsystem angezeigt werden, muss der Support-Mitarbeiter in einem solchen Fall nicht weiter suchen. Die Situation gestaltet sich jedoch anders, wenn ein gleichartiger CALL vom Hersteller noch nicht abschließend verifiziert wurde. In einem solchen Fall ist der entsprechende CALL auch noch nicht in die Wissensdatenbank übernommen worden und kann deshalb auch nicht automatisch angezeigt werden. Das SSMS muss aus diesem Grund dem Support-Techniker die Möglichkeit geben, ausgehend von einer aktuellen Anfrage nach offenen CALLs mit der selben Kategorisierung im SSMS suchen, bevor er den Fall an den Softwarehersteller übermittelt. Deshalb stellt das SSMS eine Suchfunktion bereit, welche diese Anforderung erfüllt.

Der Support-Techniker muss dazu lediglich aus einer CALL-Ansicht heraus über die Suchmaschine die Suche starten. Zusätzlich zu der Kategorisierung können Schlagwörter für die Durchsuchung der Kurzbeschreibungen offener CALLs ergänzt werden, um die Suche weiter einzuschränken. Die Syntax einer Suchabfrage auf die Kurzbeschreibungen folgt dabei der Logik von gängigen Internet-Suchmaschinen wie z.B. Google (z.B. Verknüpfung von zwei Schlagwörtern durch ein „+“, Ausschluß bestimmter Begriffe durch ein „-“ oder durch Verknüpfung von Suchabfragen mit Wildcards³² „*“).

Das Client-Interface übermittelt die Webseite mit der aktuellen Kategorisierung sowie die durch den Anwender eingegebenen, ergänzenden Identifikationsmerkmale an den

³² Wildcard = Platzhalter für andere Zeichen

Webserver. Dieser wandelt die Informationen aus der Client-Webseite wiederum in ein DataSet um und sendet dies an das Datenbank-Modul. Das DB-Modul formuliert aus den Suchmerkmalen ein SQL-Statement und sendet dieses an die Datenbank. Das Abfrageergebnis wird anschließend vom DBMS an das DB-Modul zurückgegeben und dort in Form eines weiteren DataSets an den Webserver übermittelt. Dort werden die Abfrageergebnisse wiederum in die Webseite des CALL-Handling-Interfaces integriert, an den Client-Computer übermittelt und dort im Browserfenster dargestellt. Der Ablauf einer Suchanfrage nach CALL-Informationen ist somit mit der dargestellten Suche nach Kundeninformationen in der Abbildung 5.4-3 vergleichbar. Im Unterschied hierzu wird jedoch ein wesentlicher Teil der Identifikationsmerkmale für die Suchabfrage direkt aus dem CALL-Handling-Interface in Form der Kategorisierung übernommen.

5.4.3.3. Öffnen und Bearbeiten von vorhandenen CALL-Informationen.

Die Bereitstellung von CALL-Informationen verläuft prinzipiell analog zu Abbildung 5.4-4, jedoch werden hier CALL- und nicht Kunden-Informationen von der Datenbank an den Client übermittelt. Die CALL-Informationen enthalten neben den geschilderten Angaben zum Sachverhalt des Problems auch Informationen über den Autor eines CALLs. Der Autor ist standardmäßig auch Bearbeiter und somit „Besitzer“ des CALLs, d.h. er hat beim Öffnen einer Anfrage auch automatisch die Schreibrechte für den CALL-Informations-Datensatz. Dies gilt natürlich auch für den Fall, dass die Bearbeitung im direkten Anschluss an die Erfassung eines CALLs erfolgt (vgl. Kapitel 5.4.3.1) und der zu bearbeitende CALL somit bereits geöffnet ist.

Für die Datensicht des Öffnens ergibt sich somit folgendes Bild: Die Ansicht eines Ansprechpartners oder eine explizite Suchabfrage durch einen Anwender führt stets zur Darstellung sämtlicher offener Anfragen nebst der assoziierten Kurzbeschreibung im Client-Interface. Der Anwender kann hier die gewünschte Anfrage selektieren und deren Öffnung über die dafür bereitgestellte Funktion der SSMS initiieren. Das Client-Interface prüft zunächst, ob der aktuelle Anwender gegenwärtig Besitzer des CALLs ist und öffnet den Datensatz anschließend mit den entsprechenden Berechtigungen. Ist der Anwender nicht Besitzer einer Anfrage, möchte diese jedoch bearbeiten, so muss er den CALL zunächst unter Verwendung der entsprechenden Funktion des Clients übernehmen. Das Client-Interface sendet hierauf eine Anfrage an die Datenbank und prüft, ob der entsprechende Datensatz gegenwärtig bearbeitet wird. Ist der CALL an einem anderen Client mit Schreibrechten geöffnet, so kann er von einem anderen Support-Mitarbeiter nicht in Bearbeitung genommen werden.

Der Anwender hat dann allerdings die Möglichkeit, den Bearbeiter des Datensatzes über das SSMS zu benachrichtigen, entweder um diesen zu veranlassen, den CALL zu schließen oder um ihm eventuell zu ergänzenden CALL-Informationen zu übermitteln. Ist der CALL nicht in Bearbeitung, so ändert das SSMS den Besitzer des CALLs in der Datenbank, sperrt den Schreibzugriff für alle übrigen Anwender und gibt die Anfrage schließlich am Client zur Bearbeitung frei. In jedem Falle wird der Status eines CALLs durch das SSMS in dem Moment zu „in Bearbeitung“ geändert, in welchem ein Anwender Schreibrechte für diesen erwirbt.

5.4.3.4. Statusänderung von CALLs

In der Prozessdefinition wurden bereits die unterschiedlichen Status eines CALL definiert. Während die Besitzrechte für eine Anfrage den Schreibzugriff steuern, so dienen die Status der Visualisierung des Bearbeitungszustandes einer Anfrage. Die Prozessdefinition liefert hierzu 3 verschiedene CALL-Zustände:

- erfasst
- in Bearbeitung
- geschlossen.

Die Änderung eines CALL-Status wird dabei auf unterschiedliche Art und Weise initiiert. Der Status „erfasst“ wird zunächst für sämtliche CALLs vergeben, die im Rahmen des Supports im SSMS gespeichert werden. Dabei ermöglicht diese spezielle Kennung die visuelle Abgrenzung zu sämtlichen, bereits aktiv bearbeiteten Anfragen. Während eine telefonische Anfrage direkt nach der Erfassung durch den Support-Mitarbeiter in den Bearbeitungsstatus übernommen wird, verschiebt der Disponent die ins SSMS überführten eMail-Anfragen in diesem lediglich erfassten Zustand in die Queue des zentrale Hotline-eMail-Accounts, wo sie gemäß der vergebenen Priorität in der Warteschlange positioniert werden. Die Differenzierung dieser unterschiedlichen Rollen geschieht mit Hilfe der Benutzersteuerung des SSMS. Die Anmeldung am System unter dem Benutzer „Disponent“ führt demnach zu einem anderen Workflow beim CALL-Handling als bei den Mitgliedern der Benutzergruppe „Support“.

Der Status „geschlossen“ muss in jedem Fall durch den aktuellen Bearbeiter mit Hilfe der entsprechenden Funktion des SSMS initiiert werden, da das System nicht automatisch erkennt, zu welchem Zeitpunkt eine Anfrage als abgeschlossen erachtet werden kann. Der Status ist, wie auch die Information über den Bearbeiter und den Autor, Teil des CALL-Informationen-Datensatzes und wird bei jedem Aufruf mit an das Client-Interface übergeben

und dort dargestellt. Der Ablauf dieses Vorgangs ist daher prinzipiell in den Beschreibungen der Kapitel 5.4.3.1, 5.4.3.2 und 5.4.2.3 enthalten.

Eine Besonderheit bildet jedoch das Erfassen und Abschließen eines CALLs, da hier zusätzlich zum Speichervorgang eine automatische eMail an den Ansprechpartner beim Kunden ausgelöst wird. Das SSMS verfügt zu diesem Zweck über ein eigenes Software-Modul, das Notifikations-Modul (kurz NoteModul). Das NoteModul verschickt, ausgelöst durch spezielle Schlüsselsignale, selbsttätig standardisierte eMails an unterschiedliche, in den Support-Prozess involvierte Personen. Zum Inhalt der eMail wird dabei stets nur wenige, variable Einträge durch das NoteModul ergänzt. So verschickt im vorliegenden Fall das NoteModul, initiiert durch die Änderung des CALL-Status in „erfasst“, eine eMail mit einem Standardtext, der Kurzbeschreibung und der CALL-ID an den Ansprechpartner beim Kunden. Dadurch hat dieser die Möglichkeit, bei künftigen Nachfragen direkten Bezug auf die CALL-ID zu nehmen. Die eMail dient außerdem als Bestätigung für den Ansprechpartner über die Bearbeitung seines Falls durch den Support. Beim Abschließen eines CALLs durch einen Support-Mitarbeiter erhält der Ansprechpartner, wiederum initiiert durch die Änderung des CALL-Status, eine weitere eMail durch das NoteModul. Der Status „geschlossen“ führt hierbei zu einer Generierung eines Inhalts, in welchem der Grund für das Schließen der Anfrage bzw. die Lösung des Problems zusammen mit der CALL-ID und der Kurzbeschreibung übermittelt wird.

Die Datensicht eines solchen Vorgangs stellt sich wie folgt dar (vgl. Abbildung 5.4-6):

Nach der Übergabe eines CALL-Informations-Datensatzes über den Webserver an das DB-Modul generiert dieses aus dem Datensatz neben der SQL-Abfrage ein weiteres Datenpaket und sendet dieses an das NoteModul. Dabei wird die Generierung und Übermittlung dieses Datenpakets allein über definierte Änderung des Status ausgelöst. Ein solcher Auslöser wird in der Datenbanktechnik als „Trigger“ (engl. Abzug) bezeichnet. Allgemein formuliert kann ein Trigger als ein Schlüsselereignis (i.d.R. die Änderung des Zustandes von Daten) bezeichnet werden, welches als Signal zur Initiierung weiterer Funktionen eines Programms dient. Auf das SSMS übertragen bedeutet dies, das im Falle einer Status-Änderung eines CALLs durch den Status-Trigger verschiedene, definierte Aktionen innerhalb der SSMS-Server-Software gestartet werden.

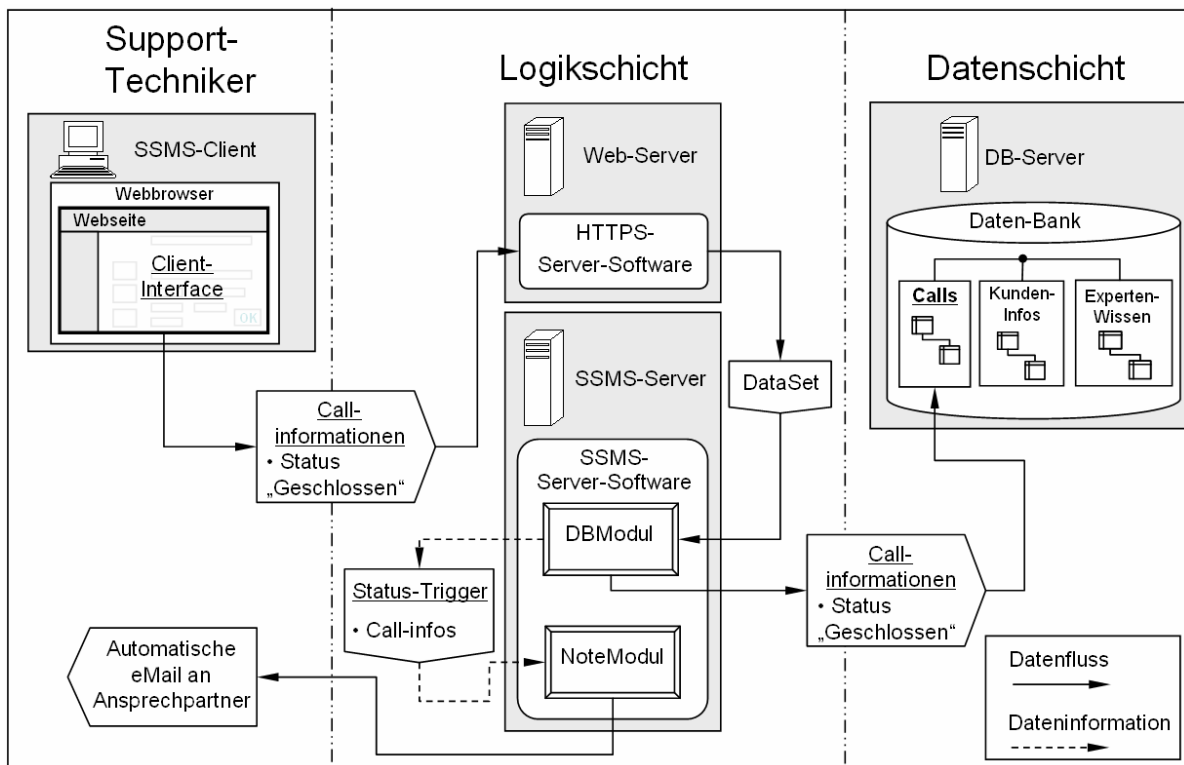


Abbildung 5.4-6: Datensicht der Versendung automatischer Notifikationen

Im betrachteten Fall dienen die Status-Änderungen „erfasst“ bzw. „geschlossen“ als Trigger und lösen die Übermittlung eines definierten Teils der CALL-Informationen vom DB-Modul an das NoteModul aus. Das NoteModul übernimmt aus dem Datenpaket die CALL-ID und die Kurzbeschreibung in die Betreffzeile einer neuen eMail und fügt einen statusspezifischen, standardisierten Text als Inhalt ein. Im Falle des Abschließens eines CALLs übernimmt das NoteModul zusätzlich noch die Problemlösung in den Text der eMail. Anschließend versendet das NoteModul die eMail an den Ansprechpartner des Kunden.

5.4.3.5. Formatierung von CALLs nach Herstelleranforderungen und Übergabe an das Clipboard

Die Bearbeitung von Kundenanfragen zu Problemen beim Einsatz einer komplexen Applikation wie einem 3D-CAD-Programm führt im Rahmen der Verifizierung durch den Support zu unterschiedlichen Schlussfolgerungen über die Ursache dieses Problems. Wird deutlich, dass das Problem durch einen Fehler im Quellcode der Applikation hervorgerufen wird, so muss der Softwarehersteller über diesen Umstand in Kenntnis gesetzt werden, damit dieser das Problem beheben kann. Für die Übermittlung von derartigen Problemen werden von den Herstellern unterschiedliche Wege und Dokumentations-Inhalte vorgeschrieben. Im Rahmen der Analyse wurde verdeutlicht, dass in jedem Falle eine komplette Dokumentation eines Applikations-Fehlers im Systemhaus vorliegen muss, da von hier aus der Dialog mit dem Kunden stattfindet. Auf der anderen Seite muss das SSMS Funktionen beinhalten die es einem Supportmitarbeiter ermöglichen, die für den Hersteller notwendigen Informationen möglichst effizient aus der ohnehin zu führenden CALL-Dokumentation im SSMS abzuleiten.

Diese Forderung wird durch die Integration eines sog. „Clipboard“ (engl. Zwischenablage) realisiert.

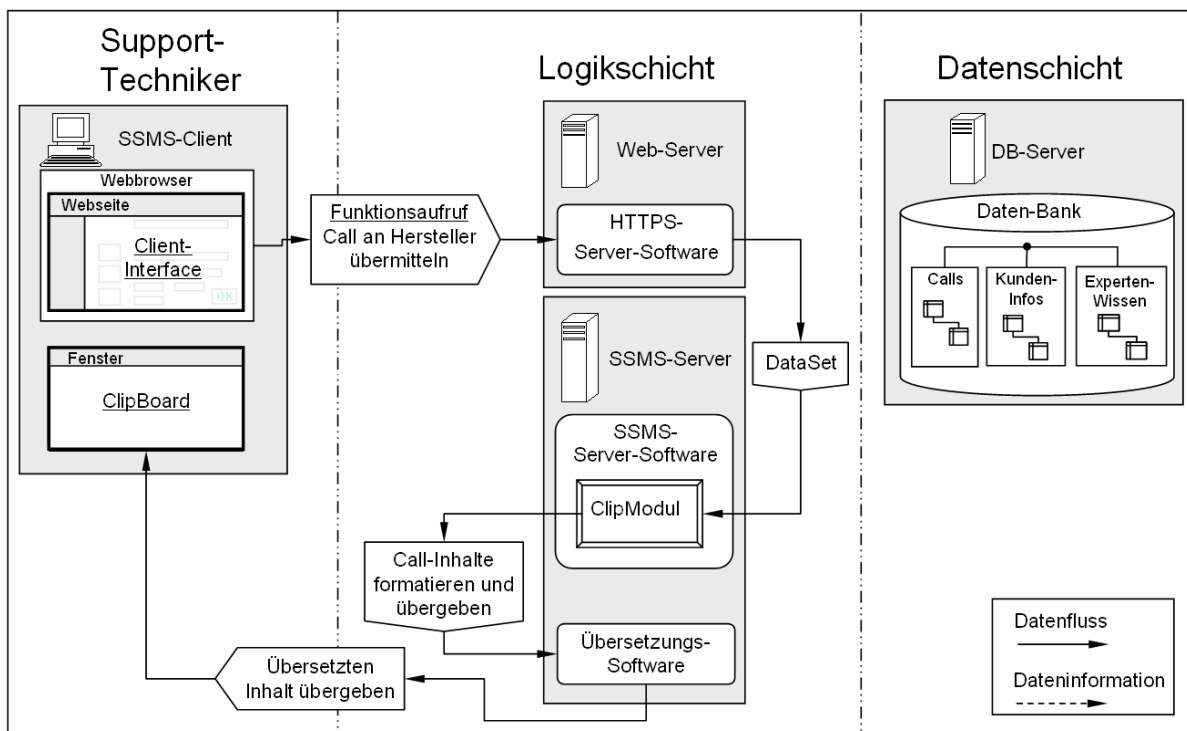


Abbildung 5.4-7: Datensicht der Formatierung und Übergabe von CALL-Informationen an das Clipboard

Das Clipboard dient dabei als Zwischenspeicher für sämtliche relevanten CALL-Informationen, welche vor der Übertragung nach den Anforderungen des Softwareherstellers formatiert werden. Zusätzlich verfügt das Clipboard über eine Verlinkung zu einer Übersetzungssoftware, da die Kommunikation mit dem Hersteller oftmals nicht in der Landessprache des Systemhauses sondern in einer Fremdsprache erfolgen muss. Das ClipBoard besteht prinzipiell aus einer Benutzerschnittstelle, dem ClipBoard-Interface und einem eigenen Software-Modul der SSMS-Server-Software, dem Clipboard-Modul (ClipModul, vgl. Abbildung 5.4-7).

Hat ein Support-Mitarbeiter ein Problem an den Hersteller zu übermitteln, so löst er die Übergabe der CALL-Informationen an das Clipboard manuell über die entsprechende Funktion des Client-Interfaces aus. Der Webserver übernimmt die CALL-Informationen und formuliert aus diesen ein DataSet, welches an das ClipModul übermittelt wird. Dieses übernimmt das DataSet und formatiert es nach vordefinierten, herstellerspezifische Schemata. Die Auswahl des Formatierungsschemas wird vom NoteModul automatisch anhand der Kategorisierung des CALLs vorgenommen. Zu den zu übergebenden CALL-Informationen zählen in der Regel Lizenzinformationen zum Softwareprodukt, die Kategorisierung, die Priorität, die Kurzbeschreibung des CALLs sowie die Reproduktionsbeschreibung. Das ClipModul fügt außerdem einen Link zu den lokalen Applikationsdateien in das Clipboard-Interface ein.

Über eine entsprechende Funktion kann der Mitarbeiter nun den Inhalt des Clipboards von der angebundenen Übersetzungs-Software in die gewünschte Sprache übersetzen lassen. Anschließend übermittelt der Anwender den CALL, z.B. per eMail oder über ein Service-Portal, an den Software-Hersteller. Da das Clipboard innerhalb eines eigenen Fensters innerhalb des SSMS-Clients dargestellt wird, kann der Anwender im CALL-Handling-Interface zwischenzeitlich auch neue CALLs aufnehmen oder in der Wissensdatenbank nach entsprechenden Einträgen suchen, ohne dass der zu bearbeitende CALL zwecks Übernahme der darin enthaltenen Informationen erneut aufgerufen werden muss.

5.4.4. Datenhaltung von Expertenwissen

Die Datenhaltung von Expertenwissen findet, ähnlich der Handhabung von CALL-Informationen, ausschließlich zwischen Client und Datenbank des SSMS statt. Im Rahmen der Prozessdefinition wurde der prinzipielle Ablauf bei der Erstellung von KB-Artikeln bereits dargelegt (vgl. Abbildung 5.3-3) In Abbildung 5.4-1 sind die in diesem Zusammenhang relevanten Operationen dargestellt, wobei sich diese formal mit den im Rahmen des CALL-Handlings behandelten Typen decken. Es folgt zunächst eine Auflistung der

unterschiedlichen Operationen:

- Erstellen und Speichern von KB-Einträgen
- Durchsuchen der Wissensdatenbank nach Artikeln und abgeschlossenen CALLs
- Öffnen und Bearbeiten von KB-Einträgen
- Status-Änderung von KB-Einträgen

Wissen, welches im Rahmen des Wissensmanagements innerhalb des SSMS dokumentiert, gepflegt und dargestellt wird, ist gemäß Kapitel 3.2.3 in drei unterschiedliche Bereiche entsprechend dem Ursprung der zu diesem Wissen führenden Erfahrungen und Erkenntnisse gegliedert:

- implementierungsspezifisches Wissen
- anwendungsspezifisches Wissen
- problemspezifisches Wissen

In dem nachfolgenden Kapitel soll nun die Datensicht dieser Abläufe näher beschrieben werden. Dabei wird auch dargestellt, inwiefern der Ursprung des Wissens Einfluss auf dessen Handhabung im SSMS nimmt.

5.4.4.1. Erstellen und Speichern von KB-Einträgen

Die Erstellung von Wissensdatenbank-Einträgen erfolgt gemäß des Ursprungs der Erkenntnis aus zwei unterschiedlichen Situationen heraus:

- Aus abgeschlossenen Anfragen im Kontext des CALL-Handlings (problemspezifisches Wissen)
- Aus Erfahrungen bei Implementierung und Einsatz von Applikationen (implementierungs- bzw. anwendungsspezifisches Wissen)

Im ersten Fall erfolgt die Übernahme von Erkenntnissen im direkten Anschluss an das Abschließen eines CALLs. Der Trigger für die Übernahme in die Wissensdatenbank ist dabei die Status-Änderung der Anfrage auf den Zustand „geschlossen“ (vgl. Abbildung 5.4-8). Bei der Betrachtung der Datenhaltung bezüglich dieses Vorgangs bewirkt die Status-Änderung in diesem Fall die Absendung des Triggers „Anfragestatus geschlossen“ durch das DB-Modul an das für die Verwaltung des Expertenwissens zuständige Software-Modul der SSMS-Server-Software - das KB-Modul. Dieses puffert zunächst den zusammen mit dem Trigger übermittelten CALL-Informations-Datensatz und schickt anschließend eine Abfrage an das Client-Interface, in welcher der Anwender gefragt wird, ob der aktuelle CALL in die Wissensdatenbank übernommen werden soll.

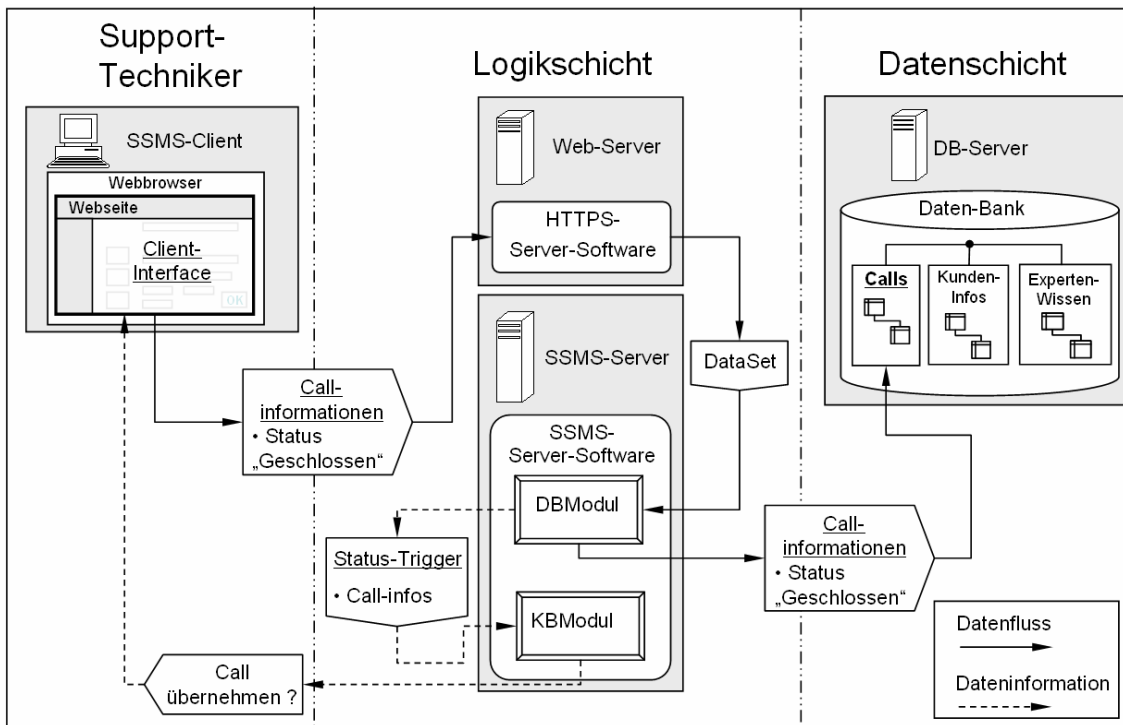


Abbildung 5.4-8: Datensicht der Initiierung der Übernahmen von CALL-Informationen in die KB

Bestätigt der Anwender die Übernahme des CALLs, so werden die entsprechenden CALL-Informationen in einem nachfolgenden Schritt in die Wissensdatenbank des SSMS übernommen (vgl. Abbildung 5.4-9).

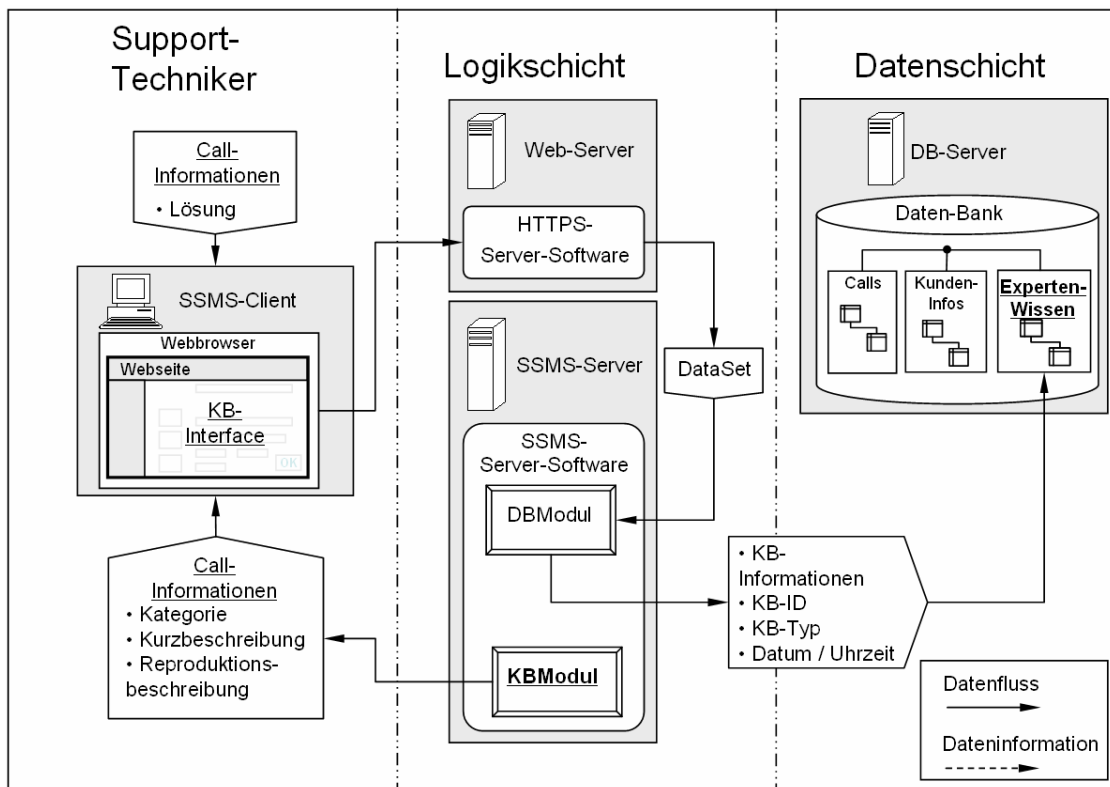


Abbildung 5.4-9: Datensicht der Übernahme eines abgeschlossenen CALLs in die KB

Hierzu übernimmt das KB-Modul zunächst aus den gepufferten CALL-Informationen die Kategorisierung, die SPR-Nummer (so vorhanden), die Kurzbeschreibung sowie die Reproduktionsbeschreibung in einen neuen KB-Eintrag und puffert diesen um anschließend den Anwender aufzufordern, diejenigen Abschnitte der CALL-Dokumentation zu markieren, welcher die Lösung des Problems beinhaltet. Hat der Anwender die entsprechenden Teile markiert, bestätigt er die Übernahme in einen neuen KB-Eintrag.

Der neue KB-Eintrag wird nun aus den gepufferten Informationen und den markierten Abschnitten als Lösung erstellt, und vom KB-Modul über den Webserver an das KB-Interface übermittelt. Der Anwender hat jetzt die Möglichkeit die Plausibilität des Eintrages zu prüfen und ggf. zu korrigieren oder zu ergänzen. Anschließend initiiert der User das Speichern des KB-Eintrags in der Wissensdatenbank, wo er unter einer eindeutigen Kennung (KB-ID) abgelegt wird. Mit Hilfe des Attributs „*KB-Typ*“ wird der Datensatz automatisch als problemspezifisches Wissen gekennzeichnet.

Eine andere Sicht ergibt sich bei der Darstellung der Erfassung von implementierungs- und anwendungsspezifischem Wissen. Hier werden durch die Support-Mitarbeiter Artikel zu verschiedenen Themengebieten des Einsatzes der betreuten Applikationen erfasst. Zwar wird sich hierbei ebenfalls der für das CALL-Handling verwendeten Kategorisierung bedient, die Erfassung erfolgt jedoch unabhängig vom CALL-Handling. Das Prinzip der Erfassung und des Ausbaus dieser Artikel lehnt sich hierbei an die sog. Wiki-Engines an. Diesem Prinzip folgend kann jeder Support-Mitarbeiter zunächst einen Artikel innerhalb einer vorgegebenen Kategorisierung verfassen und ist von diesem Zeitpunkt an auch „Besitzer“ dieses Artikels. Im Gegensatz zum CALL-Handling bedeutet dies jedoch nicht einen exklusiven Schreibzugriff des jeweiligen Support-Mitarbeiters auf den KB-Datensatz, sondern nur, dass dieser über jede Änderung oder Erweiterung des Artikels per eMail informiert wird. Somit „*moderiert*“ der ursprüngliche Autor eines Artikels denselben und stellt damit sicher, dass die Ergänzungen anderer Anwender nicht vom Thema abschweifen.

Für die Datensicht des Erstellens von derartigen Artikeln ergibt sich somit folgender Ablauf (vgl. Abbildung 5.4-10): Der Autor ruft zunächst aus dem Client-Interface mit Hilfe der dafür bereitgestellten Funktion das KB-Interface auf. Anschließend selektiert der Anwender die Kategorie und die Subkategorie des zu erstellenden Eintrags aus der durch das KB-Modul bereitgestellten Kategorien-Kaskade. Das Kategorien-Schema ist hierbei identisch mit jenem, welches auch bei der CALL-Erfassung zum Einsatz kommt. Das KB-Modul stellt hierauf die Kurzbeschreibungen aller vorhandenen Artikel mit derselben Kategorie im Clientinterface dar. Somit kann der Support-Mitarbeiter prüfen, ob bereits ein Artikel zu dem von ihm zu behandelnden Thema existiert und ggf. nur erweitert werden muss. Ist ein solcher Artikel noch nicht vorhanden, so bestätigt der Anwender die Erstellung eines neuen KB-Eintrags in der gewählten Kategorie.

Der Artikel wird sodann vom Support-Mitarbeiter mit einer Kurzbeschreibung versehen, gefolgt von der Eingabe des eigentlichen Artikel-Inhaltes. Zudem kann der Anwender ergänzende Bild- oder Applikationsdateien zu dem Artikel hinzufügen (vgl. Kapitel 5.4.5.3) oder Verweise auf andere Artikel ergänzen. Nach Abschluss der Bearbeitung initiiert der Anwender das Speichern des Artikels in der Wissensdatenbank.

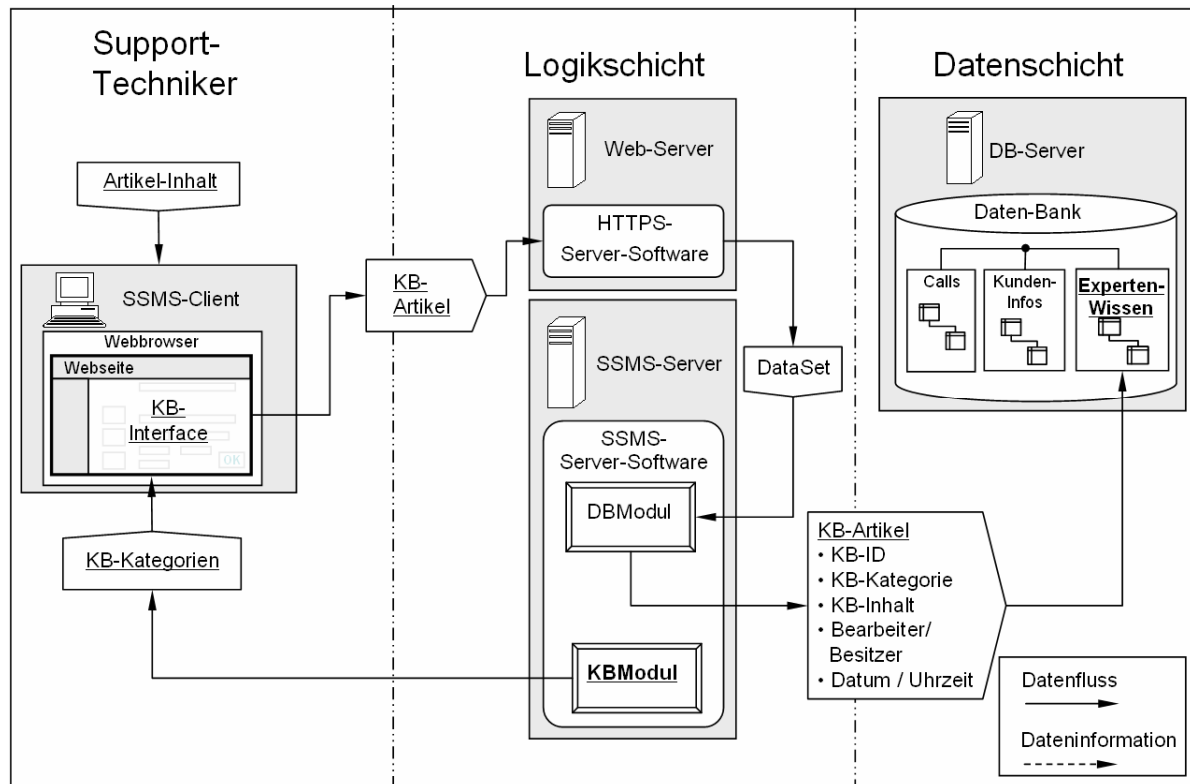


Abbildung 5.4-10: Erstellen und speichern von KB-Artikeln

Das KB-Interface übergibt hierauf den Artikel an den WebServer, welcher aus dem Inhalt ein DataSet formuliert und an das DB-Modul sendet. Das DB-Modul formt das DataSet in ein SQL-Statement um und übermittelt dieses an das DBMS, welches den gesamten Artikel unter einer eindeutigen Kennung in der Wissensdatenbank speichert. Das Attribut „KB-Typ“ wird in diesem Fall automatisch auf den Wert „implementierungs- / anwendungsspezifisches Wissen“ gesetzt.

5.4.4.2. Durchsuchen der Wissensdatenbank nach Artikeln und abgeschlossenen CALLS

Das Durchsuchen der Wissensdatenbank erfolgt innerhalb des SSMS auf zwei unterschiedliche Art und Weisen und zwar

- automatisch während der CALL-Erfassung entsprechend der gewählten Kategorien des aktuellen CALLs (vgl. F57 in Abbildung 9.2-3) oder

- bei der manuellen Erstellung neuer KB-Artikel mit Hilfe der Suchmaschine des SSMS

Im Fall der automatischen CALL-Erfassung übermittelt das Client-Interface dynamisch die gewählte Kategorie und Subkategorie eines neuen CALLs bzw. KB-Eintrags an den Webserver. Dieser generiert aus der Kategorisierung ein DataSet und übermittelt dieses an das DBModul. Das DB-Modul erzeugt wiederum aus dem DataSet eine Datenbankabfrage in Form eines SQL-Statements und übergibt diese an das DBMS. Das DBMS übergibt das Ergebnis dieser Abfrage wiederum an das Datenbank-Modul, wo ein weiteres DataSet aus dem Suchergebnis generiert und an den Webserver zurückgesendet wird. Der Webserver integriert nun diese Informationen in das KB-Interface des SSMS und stellt die Kurzbeschreibungen und KB-IDs der relevanten KB-Einträge parallel zum CALL-Handling-Interface im Client dar. Der Anwender kann nun anhand der Kurzbeschreibung relevante KB-Einträge identifizieren und selektieren oder das Suchergebnis mit Hilfe zusätzlicher Merkmale weiter eingrenzen. Erfolgt das Durchsuchen der Wissensdatenbank manuell durch den Support-Techniker, so kann dies nach sämtlichen hinterlegten Kategorien, SPR-Nummern sowie nach Schlagworten innerhalb der KB-Einträge über die Suchmaschine des SSMS erfolgen.

5.4.4.3. Öffnen und Bearbeiten von KB-Einträgen

Hat der Benutzer einen KB-Eintrag mit Hilfe der im vorigen Kapitel beschriebenen Vorgehensweise lokalisiert, so kann er diesen im nachfolgenden Schritt Öffnen und ggf. bearbeiten. Da nach dem Ausführen der Suchfunktion zunächst nur die KB-ID und die Kurzbeschreibung des KB-Artikels im KB-Interface dargestellt werden, muss der Anwender mit Hilfe der entsprechenden Funktion des SSMS anschließend die Darstellung des gesamten KB-Eintrages auslösen. Das Client-Modul übermittelt hierzu die KB-ID an den Webserver, welcher daraus wiederum eine Suchanfrage zu dem Datensatz mit Hilfe eines DataSets - analog zum bereits geschilderten Ablauf zur Auffindung der relevanten KB-Einträge – generiert. Das Abfrageergebnis wird vom DBMS über das DB-Modul an den Webserver übermittelt und von diesem in die Webseite des KB-Interfaces integriert, so dass der KB-Eintrag für den Anwender sichtbar am Client dargestellt wird.

KB-Artikel werden zunächst standardmäßig schreibgeschützt im KB-Interface geöffnet. Soll der jedoch Artikel bearbeitet werden, so muss dies durch den Anwender mit Hilfe der entsprechenden Funktion des KB-Interfaces explizit initiiert werden. Das KB-Interface sendet hierauf ein entsprechendes Signal an den Webserver, der wiederum ein DataSet aus diesem Signal formuliert und an das KB-Modul übergibt. Das KB-Modul sendet daraufhin eine entsprechende Anfrage an das DBMS, wo geprüft wird, ob der KB-Eintrag sich bereits in

Bearbeitung befindet. Ist der Schreibzugriff verfügbar, so erfolgt die Bearbeitungsfreigabe über KB-Modul und Webserver an das KB-Interface. Der Anwender kann nun den Eintrag in gewünschter Weise modifizieren oder ergänzen. Ist der KB-Eintrag jedoch bereits in Bearbeitung, so wird aus dem Abfrageergebnis des DBMS im KB-Modul eine Fehlermeldung generiert und über den Webserver an das KB-Interface übermittelt.

Im Rahmen der ergonomischen Betrachtungen der Software wird unter anderem auch zu klären sein, wie die unterschiedlichen Zugriffsrechte für den User optisch kenntlich gemacht werden können. Näheres hierzu ist in Kapitel 5.6.1.2 dargestellt. Nach Abschluss der Bearbeitung erfolgt wiederum ein durch den User auszulösender Speichervorgang (vgl. Kapitel 5.4.4.1). Dabei wird zunächst geprüft, ob der Bearbeiter mit dem „Ersteller“ des Eintrags übereinstimmt. Ist dies nicht der Fall, so wird an den „Ersteller“ eine automatische eMail gesendet, welche diesen über die Änderung des KB-Eintrags, die KB-ID und den Namen des „Bearbeiters“ in Kenntnis setzt. Diese Aufgabe übernimmt das bereits beschriebene NoteModul.

5.4.4.4. Revision von KB-Einträgen

In der Darstellung der Anforderungen an das SSMS wurde dargelegt, dass KB-Einträge einem Revisionsmanagement im Sinne eines Lebenszyklus unterliegen müssen. Der Status von problemspezifischen Anfragen ist dabei primär an das Applikationsrelease geknüpft, so nicht vom Hersteller spezifische Angaben zur Behebung eines Problems getroffen wurden. Dies bedeutet, dass spätestens zum Releasewechsel ein aus einem problemspezifischen CALL resultierender KB-Eintrag überprüft werden muss. Auch anwendungs- und implementierungsspezifische KB-Artikel sind einem Lebenszyklus unterworfen, der jedoch aus den genannten Gründen nicht pauschal an einen Releasewechsel geknüpft werden kann (vgl. Kapitel 3.3.6). So ist hierfür ein geeigneter Turnus festzulegen, wobei jeweils der Zeitraum seit der letzten Bearbeitung eines Artikels berücksichtigt werden muss.

Aus Datensicht maßgebend für den Lebenszyklus eines KB-Eintrags ist primär der Wert des Attributs „KB-Typ“. Als sekundäres Merkmal dient zusätzlich noch das Datum und die Uhrzeit der letzten Bearbeitung. Ist ein Revisionskriterium für einen KB-Eintrag erfüllt, so wird der Eintrag vom KB-Modul entsprechend optisch gekennzeichnet. Zusätzlich werden sämtliche zur Revision anstehenden Artikel mit KB-ID und Kurzbeschreibung in einer eigenen Sicht des KB-Interfaces dargestellt, so dass jeder Support-Mitarbeiter die entsprechenden Artikel überprüfen und für den nächsten Turnus freigeben oder aber Schließen und sperren kann.

5.4.5. Datenhaltung von Applikationsdateien

Sämtliche Applikationsdateien, die im Verlauf einer Kundenanfrage oder in Form von Beispieldateien bei der Verwaltung von Expertenwissen anfallen, müssen im Rahmen des SSMS gespeichert, verwaltet und gesichert werden. Da die anfallenden Daten sehr umfangreich sein können und somit viel Speicherplatz beanspruchen, vor allen Dingen da es sich um 3D-CAD-Daten handelt, ist es nicht zweckmäßig diese Daten in binärer Form direkt in der Datenbank des SSMS zu speichern. Sinnvollerweise werden diese deswegen innerhalb eines eigenständigen, redundanten Dateisystems in komprimierter Form abgelegt, wobei der SSMS-Server die Verwaltung dieses Dateisystems übernimmt. Ein solches Dateisystem wird auch als Vault (engl. „Tresor“) bezeichnet und häufig im Zusammenhang mit PLM-Produkten verwendet. Der Begriff des Daten-„Tresors“ verweist hierbei auf die im Kontext von PLM-Systemen notwendige, sichere, d.h. vor unbefugtem oder unbeabsichtigtem Zugriff geschützte und redundante Ablage von produktrelevanten Daten. Bei einem Vault werden die Applikationsdateien vom Client an die Serversoftware übermittelt und von dieser in einem hierarchischen Dateisystem unter einem eindeutigen Namen gespeichert.

Die Serversoftware speichert bei diesem Vorgang den Namen, die Position (Datei-Pfad) und gegebenenfalls den entsprechenden Versionsstand und das Speicherdatum der Applikationsdateien in einer Datenbank und kann diese somit dem Anwender bei einer Anfrage über die Client-Software jederzeit wieder zur Verfügung stellen. Die Verwaltung und der Zugriff auf diese Informationen innerhalb der Datenbank wird mit Hilfe des bereits mehrfach erwähnten Datenbank-Moduls (DB-Modul) der Serversoftware umgesetzt. Bei einer Anfrage des Clients zu einem bestimmten Datensatz kopiert die Server-Software diesen zunächst aus dem Vault in ein lokales Arbeitsverzeichnis auf dem Clientrechner. Wird der Datensatz modifiziert, so wird dieser bei der Rückführung in das Vault parallel zum ursprünglichen Datensatz gespeichert. Dadurch wird gewährleistet, dass sämtliche Bearbeitungsstände eines Datensatzes erhalten bleiben und jederzeit repliziert werden können.

Die genannten Eigenschaften und Funktionen eines Daten-Vaults werden auch im Rahmen des vorliegenden Konzeptes genutzt, um die anfallenden Daten aus dem CALL-Handling und dem Wissensmanagement zu speichern und zu verwalten. Im Unterschied zu PLM-Systemen müssen hier allerdings keine komplexen Referenzen, wie diese bei 3D-CAD-Daten vorkommen, verwaltet werden, da die Daten in jedem Fall in Form komprimierter Datenarchive abgelegt werden. CAD-fremde Datenformate werden zudem nur als einzelne, voneinander unabhängige Dateien im Vault gespeichert. Die Aufgabe der Datenorganisation innerhalb des Vaults sowie das Anlegen und die Pflege der dazugehörigen Informationen wird von einem eigenen Softwaremodul der SSMS-Serversoftware übernommen - dem

Vault-Modul.

Im Folgenden werden nun die unterschiedlichen Abläufe bei der Verwaltung von Applikationsdateien innerhalb des SSMS näher beschrieben werden. Im Rahmen dieser Datensicht wird fallspezifisch der Weg der Applikationsdateien über alle Instanzen des Systems hinweg, ausgehend vom Endkunden bis zum Vault, dargestellt. In Bezug auf die Applikationsdateien kann zwischen vier unterschiedlichen Vorgängen unterschieden werden, wie in Abbildung 5.4-1 bereits dargestellt wurde:

- Erstmaliges Speichern von CALL-spezifischen Applikationsdateien und deren Bearbeitungsstände im Vault.
- Bereitstellen von CALL-spezifischen Applikationsdateien aus dem Vault.
- Speichern und Revisionieren von mit Wissensdatenbank-Einträgen assoziierten Applikationsdateien im Vault.
- Bereitstellen von mit Wissensdatenbank-Einträgen assoziierten Applikationsdateien aus dem Vault.

5.4.5.1. Erstmaliges Speichern von CALL-spezifischen Applikationsdateien im Vault

Die Abbildung 5.4-11 zeigt das erstmalige Speichern von CALL-spezifischen Applikationsdateien im Vault. Die Speicherung dieser „*Originaldaten*“ ist notwendig, um jederzeit den ursprünglichen Zustand der zu einem spezifischen CALL vom Kunden gelieferten Applikationsdateien reproduzieren zu können. Dies ist schon allein deshalb unumgänglich, da jedem in einen CALL involvierten Support-Techniker einen Zugriff auf die zur Reproduktion einer Anfrage benötigten Daten ermöglicht werden muss. Ein weiterer Grund ist die Veränderung von Daten durch die unterschiedlichen Verfahren zur Lösungsfindung. Führt hierbei ein Lösungsansatz nicht zum Erfolg, so muss unter Verwendung der Originaldaten gegebenenfalls ein anderer verfolgt werden.

Die Übermittlung von Applikationsdateien geschieht stets anfragespezifisch unter Verwendung unterschiedlicher Medien wie eMail, bei größeren Datenvolumina auch mittels FTP-Server oder CD-ROM / DVD an den Support des Systemhauses. Der Supporttechniker muss zunächst einmal diese Daten in ein lokales Arbeitsverzeichnis kopieren, um mit der Fehleranalyse beginnen zu können. Dieses Arbeitsverzeichnis besteht aus einem übergeordneten Ordner mit dem Namen der Firma des Anfragestellers und einem oder mehreren Unterordnern, welche mit der CALL-Nummer (CALL-ID) und dem aktuellen Datum gekennzeichnet sind. Eine solche Struktur erlaubt dem Support-Techniker eine schnelle

Orientierung im Dateiverzeichnis, falls an einem Tag mehrere CALLs von ein- und demselben Kunden bei ihm eintreffen.

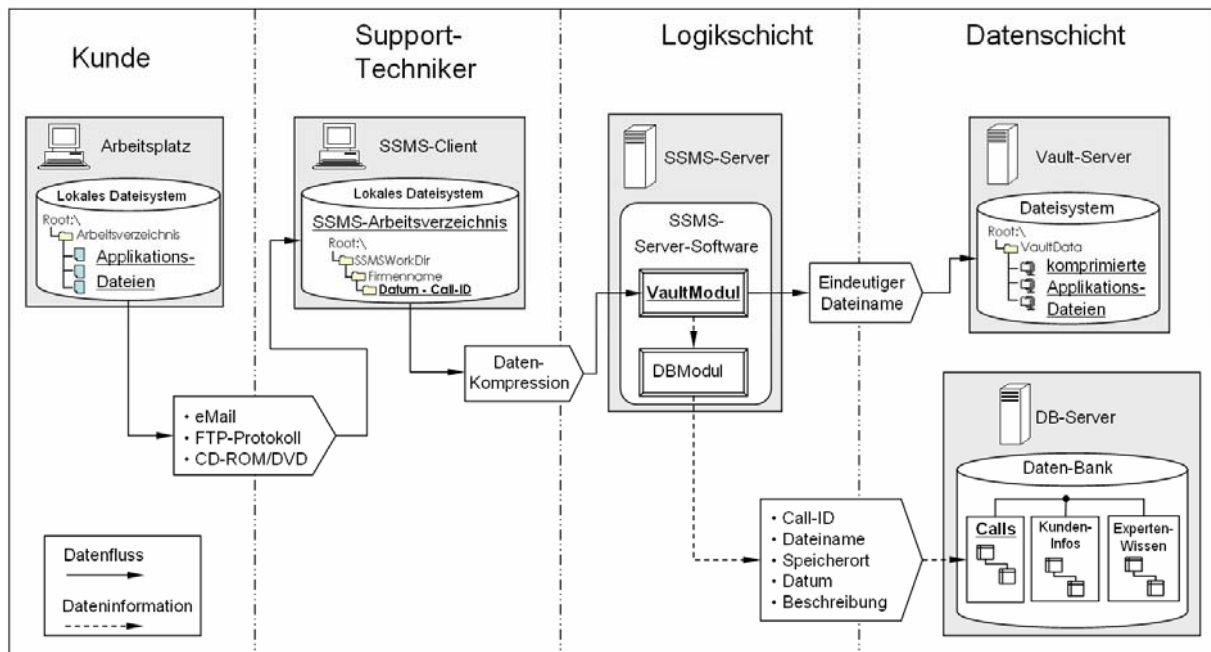


Abbildung 5.4-11: Datensicht des erstmaligen Speicherns von Applikationsdateien

Die Verzeichnisstruktur legt das SSMS innerhalb eines freigegebenen Hauptverzeichnisses, dem SSMSWorkDir, auf dem SSMS-Client-Rechner automatisch an, wenn der Support-Techniker bei der CALL-Erfassung die Zuordnung von Applikationsdateien mit Hilfe der entsprechenden Funktion initiiert. Nachdem das entsprechende Verzeichnis angelegt ist, kann der Support-Techniker die ihm vom Kunden zur Verfügung gestellten Daten in dieses Verzeichnis kopieren. Anschließend werden die Daten vom Vault-Modul auf dem Clientrechner komprimiert und unter Vergabe eines eindeutigen Namens in das Vault kopiert. Durch die Kompression vor dem Kopieren in das Vault werden die Belastung des Netzwerkes und der Speicherbedarf für das Vault auf ein möglichst geringes Maß reduziert.

Das Vault-Modul übergibt nach dem Kopiervorgang den Namen, den Speicherort, das Speicherdatum und eine benutzerdefinierte Beschreibung über ein DataSet an das Datenbank-Modul, welches diese Attribute mit Hilfe der CALL-ID zu den CALL-Informationen der entsprechenden Anfrage in der Datenbank des SSMS hinzufügt. Die benutzerdefinierte Beschreibung soll dabei dem User zu einem späteren Zeitpunkt die Identifizierung des Inhalts der zu Archiven komprimierten Applikationsdateien erleichtern. Nach dem Speichern der Applikationsdateien im Vault werden diese mit dem Namen, der zugeordneten Beschreibung und dem Speicherdatum in der CALL-Ansicht des SSMS-Clients dargestellt.

Beim Speichern von Bearbeitungsständen der Applikationsdateien läuft prinzipiell derselbe Vorgang ab. Das Speichern dieser Stände ist dann sinnvoll, wenn veränderte Daten zum

Zwecke der Problemlösung an den Kunden zurückgesendet werden, z.B. modifizierte CAD-Daten oder überarbeitete Postprozessoren, der ursprüngliche Zustand jedoch reproduzierbar bleiben muss. Beim Speichern eines Bearbeitungsstandes in das Vault wird vom Anwender wiederum eine Beschreibung der enthaltenen Dateien ergänzt. Die Beschreibung und das Speicherdatum eines CALLs ermöglichen es dem Support-Techniker somit auch zu einem späteren Zeitpunkt, die datenseitige Ausgangs- und Endsituation einer Anfrage nachzuvollziehen.

Für die Generierung des Beschreibungstextes wird im Rahmen der Prozesssteuerung neben einem Freitext auch eine Reihe von Standardbeschreibungen, z.B. „*Original Kundendaten*“ oder „*Bearbeitungsstand wie an Kunde übermittelt*“ im Workflow hinterlegt. Auch hier werden nach dem Speichern der Daten in das Vault Informationen wie Dateiname, zugeordnete Beschreibung und das Speicherdatum in der CALL-Ansicht des SSMS-Clients dargestellt.

5.4.5.2. Bereitstellen von CALL-spezifischen Applikationsdateien aus dem Vault

In der Abbildung 5.4-12 wird der Anwendungsfall Bereitstellung von Applikationsdateien aus dem Vault dargestellt. Dabei werden mit Hilfe des Vault-Moduls ausgewählte, einem CALL zugeordnete Dateien in einem lokalen Datenverzeichnis für den User zur Verfügung gestellt. Der Prozess der Bereitstellung wird durch den User über das Client-Interface initiiert. Da die Kurzbeschreibung der mit einem CALL assoziierten Daten bei dessen Aufruf im SSMS-Client-Interface dargestellt werden, kann der Techniker die gewünschten Datensätze anhand dieser Beschreibung und gegebenenfalls des Datums identifiziert. Anschließend selektiert der User die gewünschten Inhalte und initiiert die Bereitstellung. Das Client-Modul schickt hierauf eine Bereitstellungs-Anfrage mit dem Dateinamen an das Vault-Modul. Das Vault-Modul wiederum lokalisiert die entsprechende Datei im Vault und kopiert diese in das Arbeitsverzeichnis des Client-Rechners.

Dabei prüft das Vault-Modul zunächst, ob die entsprechende Ordner-Substruktur innerhalb des SSMSWorkDir bereits existiert. Ist dies nicht der Fall, werden zunächst die entsprechenden Ordner erzeugt. Erst im Anschluss daran kopiert das Vault-Modul die Daten in den mit CALL-ID und Speicherdatum benannten Ordner und dekomprimiert sie dort.

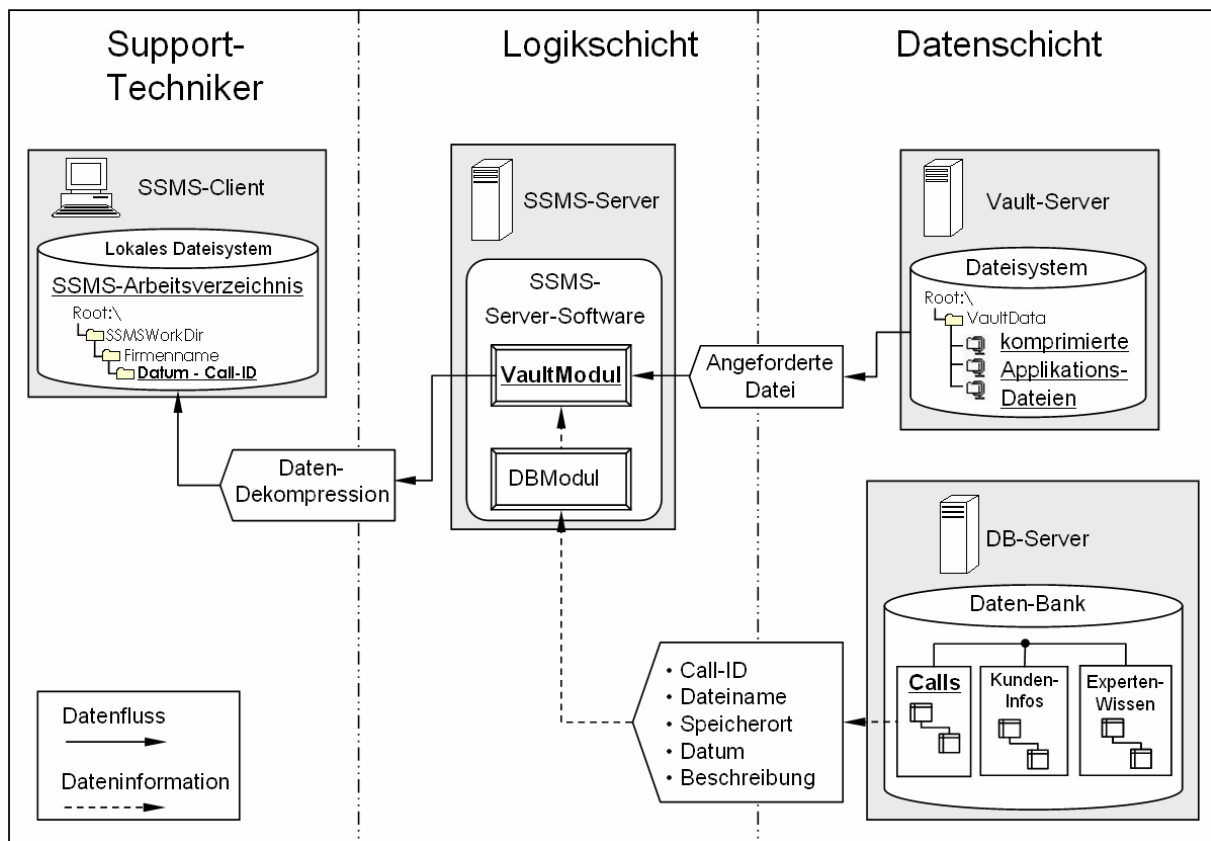


Abbildung 5.4-12: Datensicht der Bereitstellung von Applikationsdateien

5.4.5.3. Speichern von Applikationsdateien für KB-Einträge im Vault

Prinzipiell ähnelt der Vorgang beim Speichern und Revisionieren der Applikationsdateien, die Wissensdatenbank-Einträgen zugeordnet sind, dem in Kapitel 5.4.5.1 beschriebenen Vorgang. Da jedoch der Kontext ein anderer ist, soll dieser Ablauf aus der Datensicht im Folgenden näher beschrieben werden (vgl. Abbildung 5.4-13).

Grundsätzlich können der Zugang und das Verständnis zu einem softwaretechnischen Problem durch die Ergänzung praktischer Beispiele wesentlich vereinfacht werden. Zu diesem Zweck müssen Möglichkeiten geschaffen werden, diese Daten mit Einträgen in der Wissensdatenbank (**K**nowledge-**B**ase oder kurz KB) zu verknüpfen und somit dem Support-Techniker zugänglich zu machen. Wie in Kapitel 3.2.3 bereits dargestellt, kann ein Wissensdatenbank-Eintrag aus drei unterschiedlichen Rahmenbedingungen heraus entstehen:

- Aus Erkenntnissen, die aus der Bearbeitung von CALLs resultieren.
- Aus Erkenntnissen, die im Rahmen durchgeführter Implementierungen erworben werden.
- Aus Erfahrungen, die durch die Anwendung der Applikation in verschiedenen Einsatzgebieten erworben werden.

Für den ersten Fall kann eine Verknüpfung zu denselben Applikationsdateien verwendet werden, welche dem ursprünglichen CALL zugeordnet sind, aus welchem der KB-Eintrag abgeleitet wird. Hierbei muss lediglich eine zusätzliche Verlinkung zu bereits im Vault befindlichen Datensätzen erfolgen. In den übrigen Fällen müssen neue Datensätze im Vault gespeichert und mit den entsprechenden KB-Einträgen verknüpft werden. Im Gegensatz zu den in Kapitel 5.4.5.1 behandelten Kundendaten ist bei mit KB-Einträgen assoziierten Daten immer nur der aktuelle Stand erforderlich, da bei einer Aktualisierung eines Eintrags und der zugehörigen Daten der ursprüngliche Zustand nicht mehr relevant ist und überschrieben bzw. ersetzt werden kann. Im Rahmen des in Kapitel 5.4.4.4 behandelten Revisionsmanagements für KB-Artikel werden Funktionen implementiert, die den User im Falle der Aktualisierung eines KB-Eintrags ebenfalls zu Überprüfung der assoziierten Applikationsdateien auffordern.

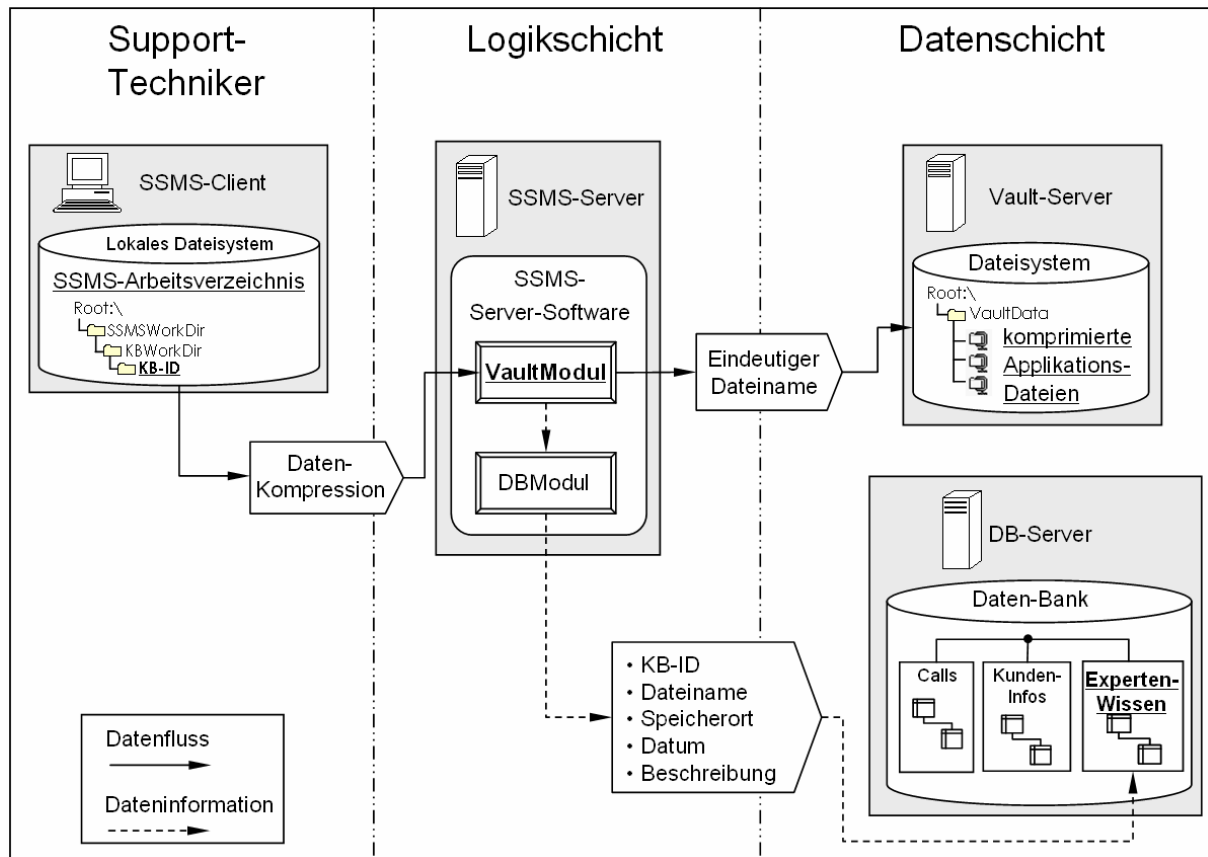


Abbildung 5.4-13: Datensicht des Speicherns von mit KB-Einträgen assoziierten Applikationsdateien

Wenn ein Support-Techniker Dateien zu einem KB-Eintrag hinzufügen will, so können diese zunächst in einem beliebigen Verzeichnis auf dem Client-Rechner abgelegt sein. Indem der User bei der Erstellung eines neuen KB-Eintrages die Zuordnung von Applikationsdateien zu dem aktuellen Eintrag über die entsprechende Funktion des SSMS-Clients initiiert, startet er das GUI für die Datenarchivierung. Hier kann nun den aktuellen Pfad zu den Daten angeben, oder einen Link zu einem spezifischen Datensatz eines CALLs erstellt werden.

Liegen die Daten lokal auf dem Client-Rechner, so komprimiert das Vault-Modul diese zunächst vor Ort und kopiert sie anschließend unter Vergabe eines eindeutigen Namens in das Datenverzeichnis des Vault-Servers. Anschließend übergibt das Vault-Modul die KB-Nummer (KB-ID), den Dateinamen, den Speicherort, das Speicherdatum und eine benutzerdefinierte Beschreibung des Datensatzes an das Datenbank-Modul, welches diese Informationen mittels der KB-ID dem entsprechenden Eintrag in der Wissensdatenbank zuordnet. Sind die Dateien Teil einer bestehenden CALL-Dokumentation, so kann der User über das GUI eine Verknüpfung zu diesem Datensatz erstellen. In einem solchen Fall übergibt das Vault-Modul lediglich die CALL-ID, dessen Namen, den Speicherort, das Speicherdatum und die Beschreibung des Datensatzes an das DB-Modul, welches diese Informationen dem KB-Eintrag hinzufügt. Nach Abschluss des gesamten Vorgangs werden

die assoziierten Daten mit Dateinamen, Datum und Beschreibung in der Ansicht des KB-Artikels innerhalb des Client-Interfaces angezeigt.

5.4.5.4. Bereitstellung und Revisionierung von Applikationsdateien für KB-Einträge

Das Anfordern und Revisionieren von einem KB-Eintrag zugeordneten Daten ähnelt wiederum dem entsprechenden Vorgang beim Handling der CALL-spezifischen Applikationsdateien (vgl. Abbildung 5.4-14). Die Notwendigkeit zu diesen Tätigkeiten ergibt sich zunächst aus der Notwendigkeit zum Nachvollziehen der in einem KB-Artikel dargelegten Sachverhalte sowie aus der Notwendigkeit zur regelmäßigen Aktualisierung der Artikel und somit auch der dazugehörigen Daten.

Die Bereitstellung der Daten erfolgt innerhalb einer Ordnersubstruktur im SSMSWorkDir, dem KBWorkDir, welche mit der Kennung des KB-Eintrags gekennzeichnet ist. Dies ermöglicht dem Supporttechniker eine schnelle Lokalisierung der angeforderten Daten. Wie bereits erläutert, wird im Gegensatz zu den in Kapitel 5.4.5.1 und den in Kapitel 5.4.5.2 beschriebenen Vorgängen für KB-Einträge in der Regel nur der aktueller Datensatz gehalten. Dies bedeutet, dass beim Speichern eines Datensatzes zu einem KB-Artikel der Anwender entscheiden muss, ob ein bereits vorhandener Datensatz ersetzt oder ob der neue Datensatz zusätzlich zu dem bereits vorhandenen mit dem KB-Eintrag verknüpft werden soll.

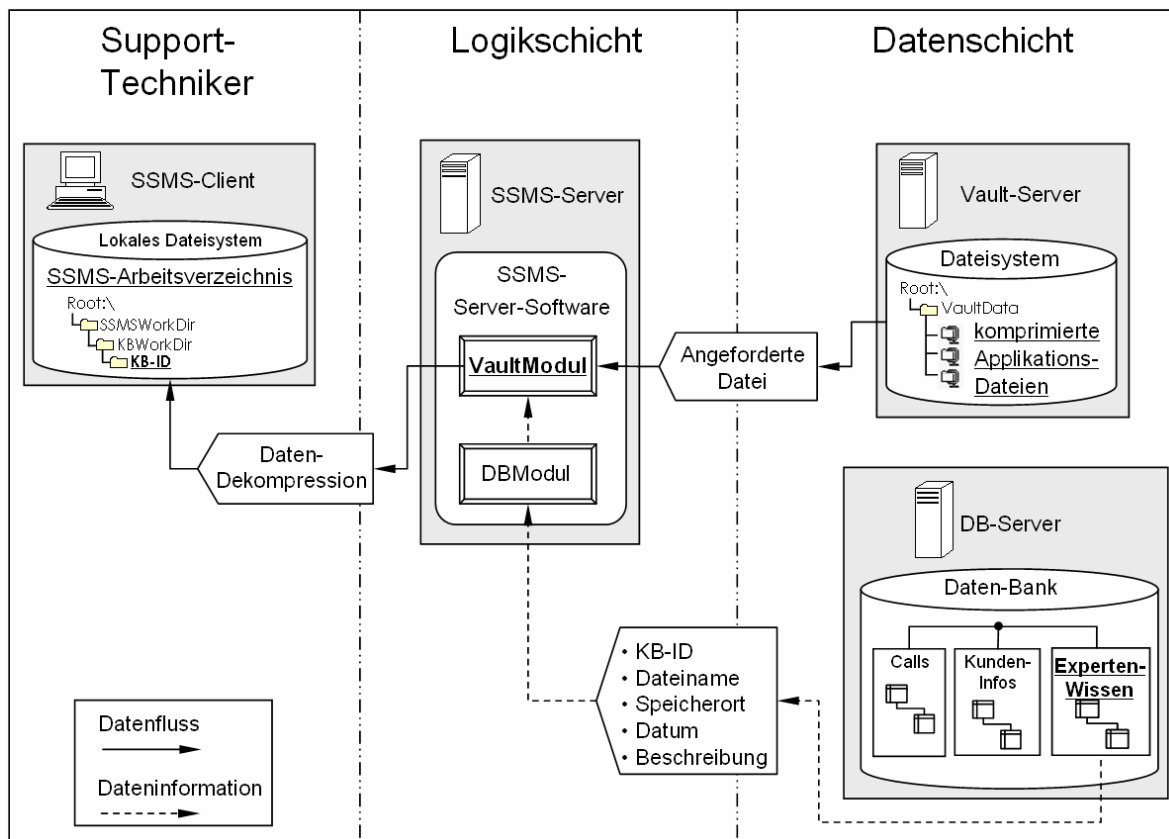


Abbildung 5.4-14: Datensicht des Bereitstellens von mit KB-Einträgen assoziierten Applikationsdateien

Die Bereitstellung von Daten wird im vorliegenden Fall vom User mit Hilfe einer entsprechenden Funktion innerhalb des SSMS-Clients initiiert. Wie bereits in Kapitel 5.4.5.1 beschrieben, wird auch im vorliegenden Fall beim Aufruf eines KB-Eintrags die Information über den Dateinamen, das Speicherdatum, den Speicherort und die Beschreibung aus der Datenbank gelesen und in der Ansicht des KB-Eintrags dargestellt. Der Support-Techniker identifiziert den benötigten Datensatz anhand der Beschreibung und startet anschließend den Bereitstellungsvorgang. Das Client-Modul schickt hierauf eine Bereitstellungs-Anfrage mit dem Dateinamen an das Vault-Modul. Das Vault-Modul wiederum lokalisiert die entsprechende Datei im Vault und kopiert sie in das Arbeitsverzeichnis des Client-Rechners. Dabei erzeugt das Vault-Modul zunächst die entsprechende Ordner-Substruktur innerhalb des Knowledge-Base-Arbeitsverzeichnisses (KBWorkDir). Erst im Anschluss daran kopiert das Vault-Modul die Daten in den mit der KB-ID benannten Ordner und dekomprimiert sie dort.

Bei der Revisionierung eines Datensatzes wird analog zur der Beschreibung in Kapitel 5.4.5.3. vorgegangen, das Client-Interface gibt dem User hierbei jedoch zusätzlich die Möglichkeit zu entscheiden, ob der Originaldatensatz überschrieben werden soll oder ob die revidierten Daten dem KB-Eintrag zusätzlich hinzugefügt werden soll. Handelt es sich

jedoch bei dem ursprünglichen Datensatz um einen zu einem CALL gehörenden Datensatz, welcher lediglich über eine Verlinkung dem KB-Eintrag zugeordnet ist, so wird der revidierte Datensatz in jedem Falle im Vault abgelegt und von diesem Zeitpunkt an ausschließlich dem KB-Eintrag gemäß der Abbildung 5.4-13 zugeordnet.

5.4.5.5. Zusammenfassung der Funktionen

Im Rahmen der Darstellungen der unterschiedlichen Datensichten wurden die Funktionen der verschiedenen Pseudo-Module der SSMS-Server-Software definiert. Diese sollen nun abschließend noch einmal stichwortartig zusammengefasst werden:

Datenbank-Modul (DB-Modul)

Funktion: Übergabe von Datensätzen (DataSets) von Webserver und Vault-Modul in Form von SQL-Statements über das DBMS an die Datenbank. Beinhaltet das Status- und Rechte-Management für CALLs und Kundeninformationen. Übermittlung von DataSets in Folge von Status-Triggern an NoteModul und KB-Modul. Formulierung von DataSets aus Abfrageergebnissen der Datenbank und Übermittlung dieser Datensätze an den Webserver und das Vault-Modul. Beinhaltet die Benutzerverwaltung und die Bedienlogik des KI- und CALL-Handling-Interfaces.

Wissensdatenbank-Modul (KB-Modul)

Funktion: Puffern von CALL-Informationen, Eingabeaufforderung bei Übernahme von CALL-Informationen in KB-Einträge über den Webserver an den Client übermitteln. Bereitstellung der Bedienlogik des KB-Interfaces und der Kategorienlisten. Beinhaltet das Statusmanagement für KB-Einträge.

Vault-Modul

Funktion: Verwaltung des Dateisystems, Komprimierung und Umbenennung von Dateiarchiven. Verwaltung des lokalen Dateisystems SSMSWorkDir und dessen Subordner. Übermittlung der Informationen über Speicherort und Dateiname von Applikationsdateien im Vault an das DB-Modul.

Notifikations-Modul (NoteModul)

Funktion: Automatisches Senden von standardisierten eMails ausgelöst durch Status-Trigger des DB-Moduls.

Clipboard-Modul (ClipModul)

Funktion: Zwischenspeicherung und Formatierung für an die Software-Hersteller zu

übermittelnde CALL-Information im Rahmen des Bugtrackings. Anbindung an die Übersetzungssoftware.

Übersetzungs-Software

Funktion: Übersetzung von CALL-Informationen innerhalb des Clipboards auf Anforderung durch den Support-Mitarbeiter. Zugekauftes Produkt, welches lediglich über das ClipModul angebunden wird.

5.5. Softwarearchitektur des SSMS

In Kapitel 5.1 wurde bereits der prinzipielle Aufbau des SSMS beschrieben. Im Folgenden wird nun, ausgehend von den Erkenntnissen aus den Betrachtungen von Prozessen und Datenhaltung, ein konkreteres Bild der Software-Architektur des SSMS gezeichnet (vgl. Abbildung 5.5-1).

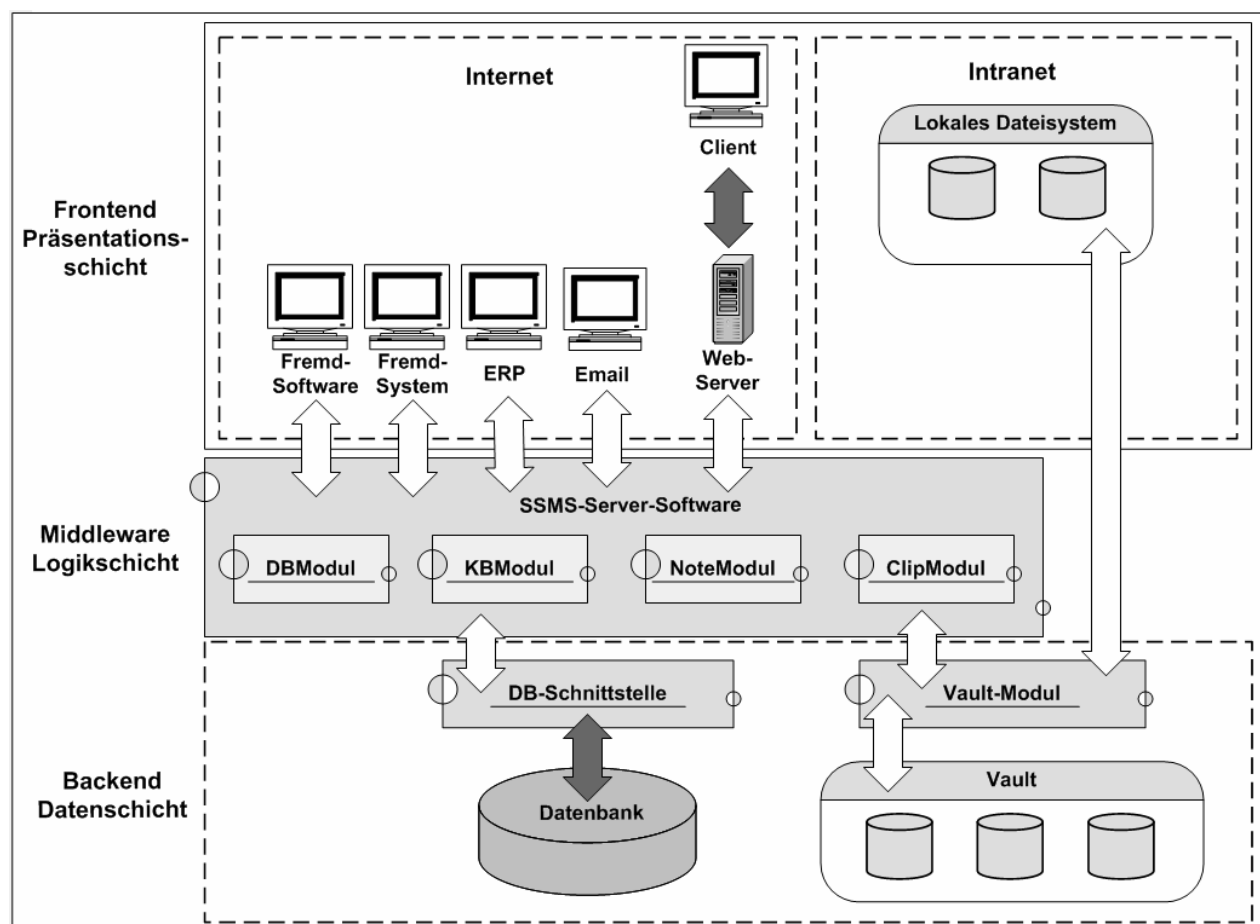


Abbildung 5.5-1: Software-Architektur des SSMS

Neben den in Kapitel 5.4.5.5 zusammengefassten Funktions-Modulen werden der Software-Architektur auch die internen und externen Schnittstellen des Systems zugerechnet. Diese

Schnittstellen sind zusätzlich noch zwischen Standardschnittstellen (grau dargestellt) und zu programmierenden Schnittstellen (weiß dargestellt) untergliedert.

Die internen Schnittstellen bezeichnen alle jene Schnittstellen des SSMS welche dem Datenaustausch und der Kommunikation zwischen den unterschiedlichen Softwaremodulen und Komponenten innerhalb des SSMS-Systems dienen. Der überwiegende Teil dieser Schnittstellen ist deshalb in der Abbildung nicht sichtbar, ein Beispiel findet sich jedoch bei der DB-Schnittstelle. Dieses Software-Modul kommuniziert die Daten der SSMS-Server-Software mit der Datenbank. Zwischen DB-Schnittstelle und Datenbank wird hierzu die Standardschnittstelle der Datenbank bzw. deren DBMS verwendet. Zwischen der SSMS-Server-Software und DB-Schnittstelle jedoch bedarf es einer noch zu konzipierenden, internen Schnittstelle zum Zwecke der Kommunikation mit den übrigen Modulen des SSMS.

Als externe Schnittstellen werden hingegen all jene Schnittstellen bezeichnet, die solche Software-Komponenten an das SSMS anbinden, welche nicht integraler Bestandteil der SSMS-Software sind. Hierzu zählen eMail-Client ggf. ein ERP-System, Fremdsysteme wie Herstellerportale oder das Clipboard, Fremdsoftware wie die Übersetzungssoftware oder das Softwaresystem der Telefonanlage sowie die Kommunikationsschnittstelle zum Webserver.

5.5.1. Interner Aufbau des DB-Moduls

In den vorausgegangenen Kapiteln wurden die einzelnen Funktionsbereiche des SSMS zusammengefasst und als Module bezeichnet, genau genommen handelt es sich dabei jedoch nicht um Software-Module im Sinne der Informatik sondern lediglich um eine Gruppierung von Funktionen in Form von Pseudo-Modulen. Die direkte Umsetzung eines solchen Pseudo-Moduls innerhalb eines Softwaremoduls würde durch den in einer Komponente vereinten Funktionsumfang viel zu wenig Flexibilität in Bezug auf die Anpassung an unterschiedliche Systemumgebungen und Aufgabenstellungen bieten. Aus diesem Grund und um der Forderung nach einem flexiblen Aufbau der Software zu entsprechen wird nun anhand des DB-Moduls der innere Aufbau eines Pseudomoduls aus einzelnen Softwaremodulen beschrieben. Ausgehend von Abbildung 5.5-1 stellen die nachfolgenden Abschnitte somit den Übergang vom schematischen Konzept zum konkreten System dar.

Zunächst werden die funktionalen Bestandteile des DB-Moduls einzelnen Softwaremodulen zugeordnet. Aus Kapitel 5.4.5.5 werden hierzu die einzelnen Funktionsblöcke übernommen:

1. Übergabe von Datensätzen (DataSets) vom Webserver und Vault-Modul in Form von SQL-Statements über das DBMS an die Datenbank. Formulierung von DataSets aus Abfrageergebnissen der Datenbank und Übermittlung dieser Datensätze an den

Webserver und das Vault-Modul.

2. Beinhaltet das Status- und Rechte-Management für CALL- und Kunden-Informationen.
3. Übermittlung von DataSets in Folge von Status-Triggern an NoteModul und KB-Modul.
4. Beinhaltet die Benutzerverwaltung und die Bedienlogik des KI- und CALL-Handling-Interfaces.

Punkt 1) führt zur Definition eines Softwaremoduls, welches die Informationen der unterschiedlichen Komponenten in eine Form übersetzt, welche die Nutzung der Standard-Schnittstelle eines DBMS ermöglicht – das Modul DB-Schnittstelle. Zur Anpassung des SSMS an ein beliebiges Datenbanksystem muss somit nur dieses Modul modifiziert werden.

Punkt 2) führt zur Definition eines Softwaremoduls für die Statusverwaltung von CALLs und Kundeninformationen (Status-Modul) sowie eines Software-Moduls für die Verwaltung von Zugriffsrechten (Rechte-Modul).

Punkt 3) führt zur Definition eines Software-Moduls für das Abfangen von Statustriggern (Trigger-Modul) und zur Übergabe von Daten-Objekten (DatenObjektModul) an KB- und NoteModul über eine Schnittstelle.

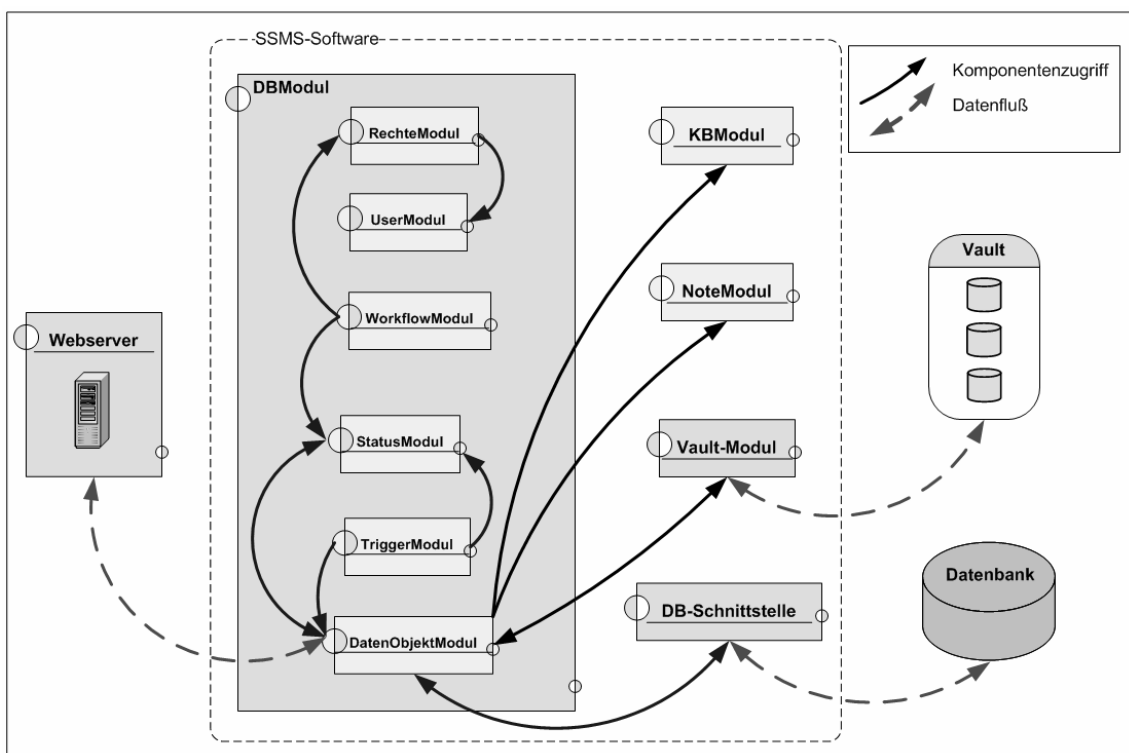


Abbildung 5.5-2: Interner Aufbau des DB-Moduls

Punkt 4) führt zur Definition eines Software-Moduls für die Benutzerverwaltung (UserModul)

sowie eines Moduls zur Steuerung des Arbeitsablaufs (WorkflowModul).

Somit ergeben sich aus den vier Funktionsblöcken sieben Software-Module, welche zuvor unter dem Pseudonym DB-Modul zusammengefasst wurden:

- DB-Schnittstelle
- Status-Modul
- Rechte-Modul
- Trigger-Modul
- DatenObjekt-Modul
- User-Modul
- Workflow-Modul

Diese Module sowie deren Beziehungen untereinander und zu den übrigen Systemkomponenten sind in Abbildung 5.5-2 dargestellt.

5.5.2. Klassen- und Objektstruktur des SSMS

In Klassendiagramme werden die für das SSMS notwendigen Klassen sowie deren Beziehungen untereinander dargestellt. Jede Klasse kann als ein einzelner Funktionsbaustein der Software verstanden werden, welcher Attribute (Variablen) und Methoden (Funktionsaufrufe) bereitstellt. Durch Vererbung können diese für weitere Klassen verfügbar gemacht werden. So enthält eine abgeleitete Klasse neben den eigenen Attributen und Methoden automatisch sämtliche Attribute und Methoden der Basisklasse. Um auch auf die Variablen und Funktionsaufrufe fremder Klassen zugreifen zu können, wird auf diese Klassen referenziert. Zur Laufzeit werden durch Instanzierung von den Klassen sog. Objekte abgeleitet, wobei deren Attribute dann mit konkreten Werten gefüllt sind. Im Folgenden wird nun auszugsweise die Klassenstruktur des SSMS dargestellt. Dabei wird ein Teil jener Klassen abgebildet, die für die Integration des CALL-Handling in das SSMS notwendig sind.

5.5.3. Auszug aus dem Klassendiagramm

Die Abbildung 5.5-3 zeigt die unterschiedlichen Klassen sowie deren Beziehungen durch Vererbung und Referenzierung. Die Notation für Attribute lautet dabei *Attribut*; während Methoden mittels einer Doppelklammer () gekennzeichnet werden. Vererbung wird mit gestrichelten, Referenzierung mit durchgehenden Pfeilen dargestellt.

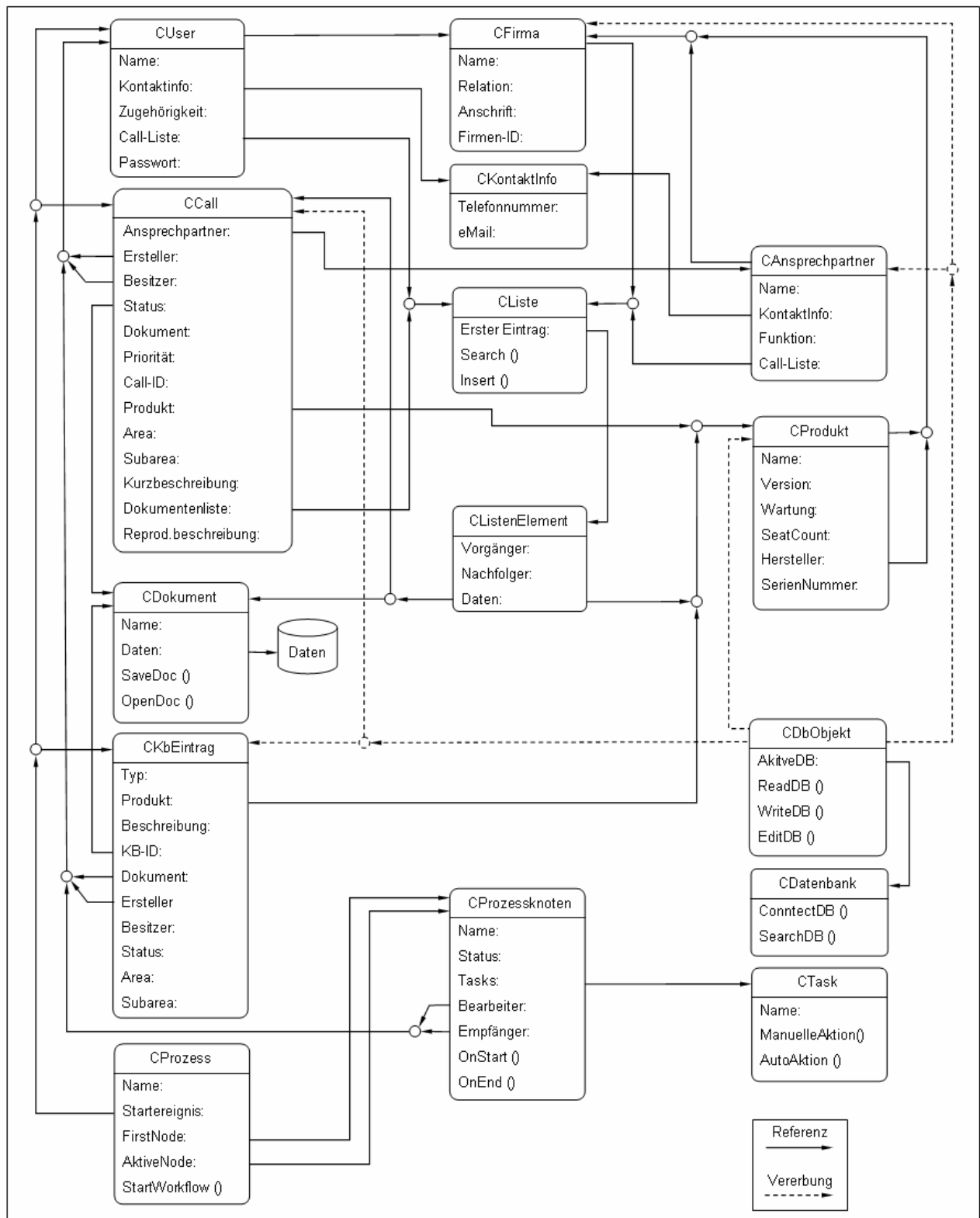


Abbildung 5.5-3: Auszug aus dem Klassendiagramm des SSMS

Die unterschiedlichen Klassen werden stets mit einem „C“ (für engl. Class = Klasse) als Prefix in der Benennung versehen. Da sich am System beispielweise unterschiedliche

Benutzer anmelden können müssen, stellt die Klasse *CUser* die notwendigen Attribute für die Anmeldung am System (*Name:* und *Passwort:*) sowie für die Rechteverwaltung (*Zugehörigkeit:*) bereit. Die *Kontaktinformationen:* eines Benutzers mit eMail-Adresse und Telefonnummer erhält das von der Klasse Instanzierte Objekt durch dessen Referenzierung auf die Klasse *CKontaktInfo*, ebenso werden die Attribute der Klasse *CFirma* und *CListe* von *CUser* referenziert.

Neben *CUser* referenzieren jene Klassen auf *CFirma*, deren Attribute einen Bezug zu einer Firmierung beinhalten müssen, also *CAnsprechpartner* und *CProdukt*. *CFirma* ist eine abgeleitete Klasse von der Basisklasse *CDbObjekt* und implementiert damit dessen Methoden und Attribute. Durch die Vererbung zwischen *CDbObjekt* und den Klassen *CProdukt*, *CAnsprechpartner*, *CCALL*, *CKbEintrag* sowie *CFirma* werden die für die Übergabe in die Datenbank notwendigen Softwarefunktionalitäten von *CDbObjekt* den übrigen Klassen zugänglich gemacht, so dass bei einem Austausch des zugrunde liegenden Datenbanksystems nur diese Klasse und die Klasse *CDatenbank* angepasst werden müssten.

Die Klasse *CListe* wird überall dort referenziert, wo eine Auflistung von Objekten abgebildet werden soll, etwa die offenen CALLs mit Bezug zu einem Ansprechpartner bei einem Kunden oder die Software-Lizenzen einer Firma, und enthält die Methoden zur Erstellung *Insert ()* und Durchsuchung *Search ()* einer Auflistung. *CListe* referenziert dabei lediglich durch das Attribut *Erster Eintrag:* auf das erste Element *CListenElement* einer solchen Auflistung, welches wiederum auf das nächste Listenelement (*Nachfolger:*) und auf den eigentlichen Inhalt, also die *Daten:* bzw. Dokumente (*CDokument*) der Auflistung referenziert.

Das Workflow-Management wird mit Hilfe der Klassen *CProzess*, *CProzessknoten* und *CTask* realisiert. Durch die Referenzierung des Attributes *Startereignis:* von *CProzess* auf *CKbEintrag*, *CCALL* und *CUser* kann ein Workflow von jedem Instanziereten Objekt dieser Klassen mit Hilfe der Methode *StartWorkflow()* initiiert werden. Die Klassen *CProzessknoten* enthält die Methoden *OnStart()* und *OnEnd ()* um damit die Prozessabfolge zu definieren und außerdem die Referenz auf die Klasse *CTask*, welche die im aktuellen Prozessschritt zu erledigende manuelle oder automatische auszuführende Aktion enthält.

5.5.4. Auszug aus dem Objektdiagramm

Aus den im vorhergehenden Kapitel skizzierten Klassen werden zur Laufzeit des SSMS konkrete Objekte instanziiert, deren Attribute jeweils Werte annehmen, die im Kontext mit dem aktuellen Prozess stehen. Die Abbildung 5.5-4 zeigt einen Auszug der Objekte und deren Relationen im Kontext zum Prozess CALL Erfassen (vgl. Abbildung 9.2-3). Dabei wird eine Situation abgebildet, in welcher der Disponent eine eMail-Anfrage erfasst und über die Queue an einen anderen Support-Mitarbeiter übergibt. Die Attribute sind hierbei zur Veranschaulichung mit fiktiven Werten gefüllt und werden im Text *kursiv* dargestellt.

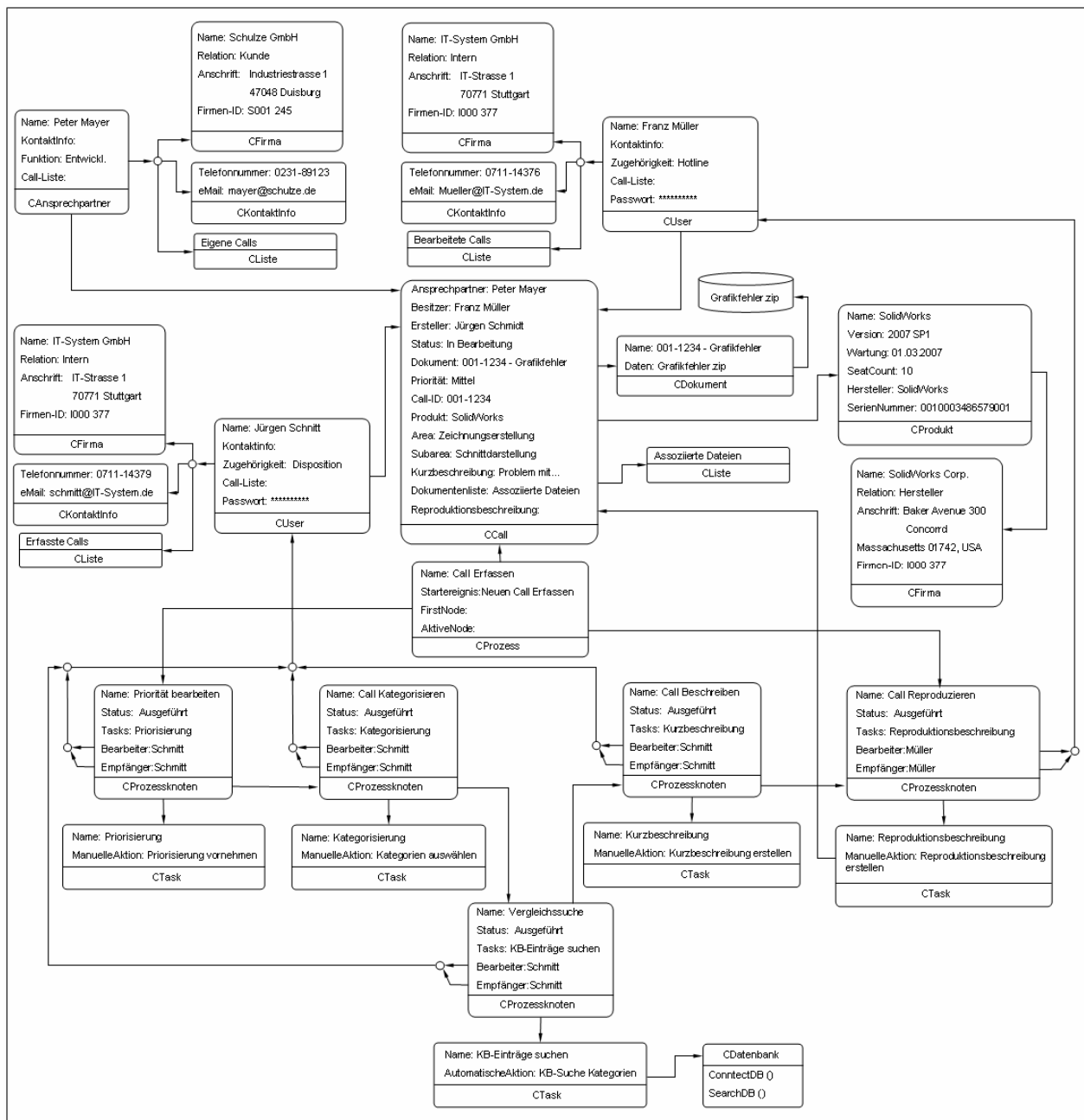


Abbildung 5.5-4: Auszug aus dem Objektdiagramm für den Prozess CALL-Erfassen

Das zentrale Objekt des CALL-Handling-Prozesses ist stets das CALL-Objekt, welches von der Klasse *CCALL* abgeleitet wird. Das CALL-Objekt referenziert seinerseits auf sowohl auf eine Reihe von Personen als auch auf eine Anzahl von Dateien und assoziierten Dokumenten. Im dargestellten Fall wird der Prozess *CALL Erfassen* ausgehend von dem *Ansprechpartner: Peter Mayer* des *Kunden: Schulze GmbH* durch den *Hotline-Mitarbeiter Jürgen Schnitt* des *VAR IT-System GmbH* gestartet. Dabei referenziert das Objekt *CALL Erfassen* stets auf das CALL-Objekt und übergibt diesem die Werte für die einzelnen Attribute.

Das Objekt *CALL erfassen* referenziert den ersten Prozessknoten *Priorität bearbeiten* als Startknoten (*FirstNode*) und *CALL Reproduzieren* als aktiver Knoten (*ActiveNode*).

Nach Abschluss des Tasks *Priorisierung* des ersten Prozessknotens wird über die Methode *OnEnd()* (vgl. Abbildung 5.5-3) der nächste Prozessknoten *CALL Kategorisieren* aktiviert.

Der Disponent hat hier die Kategorisierung des Problems vorzunehmen. Nach Abschluss der Kategorisierung wird wiederum der nächste Prozessknoten *Vergleichssuche* aktiviert. Dieser führt als Task eine *AutomatischeAktion: KB-Suche Kategorien* auf die Wissensdatenbank nach Einträgen derselben Kategorisierung aus und nutzt hierzu die Methoden *ConnectDB()* und *SearchDB()* des Objekts *CDatenbank*.

Nach Abschluss des Prozessschrittes *CALL Beschreiben* übergibt der Disponent den CALL an die Queue, von dort übernimmt der Hotline-Mitarbeiter Mayer und führt die Bearbeitung fort. Die Abbildung zeigt also genau jenen Moment, indem der Support-Mitarbeiter Müller den Prozessknoten *CALL Reproduzieren* für das durch den Ansprechpartner Herrn Mayer übermittelte Problem abgeschlossen hat.

5.6. Die Benutzerschnittstelle des SSMS-Client

Der SSMS-Client ist als Oberbegriff für die am Client-PC dargestellte, grafische Benutzerschnittstelle des SSMS zu verstehen. Diese wird in Form einer HTML-Seite innerhalb eines beliebigen Internetbrowsers bereitgestellt. Um den SSMS-Client zu starten, muss der Anwender deshalb zunächst seinen Internet-Browser öffnen und die Internetadresse des SSMS-Systems angeben. Der Webserver schickt anschließend ein Anmeldeformular an den Client, mit dessen Hilfe sich der Anwender mit einem Benutzernamen und einem Passwort authentifizieren muss, um Missbrauch und unbefugten Zugriff auf die teils sensiblen Daten des SSMS zu verhindern. Der Client besteht im Wesentlichen aus einer Startseite mit einer Suchfunktion und unterschiedlichen Links auf die verschiedenen Sub-Interfaces des Clients:

- Kundeninformations-Interface
- CALL-Handling-Interface mit Clipboard
- Wissensdatenbank-Interface

Die Suchfunktion innerhalb der Startseite ermöglicht dem Anwender einen schnellen Einstieg in die unterschiedlichen Bereiche des SSMS. So kann der Support-Mitarbeiter über jedes eindeutige Merkmal, z.B. Namen oder Telefonnummer eines Ansprechpartners, direkt in den dazugehörigen Datensatz springen, ohne zuvor das Kundeninformations-Interface aufrufen und dort nach dem Datensatz suchen zu müssen. Dies ist für eine effiziente Echtzeiterfassung von CALLs unerlässlich. Im Folgenden werden nun die unterschiedlichen Sub-Interfaces des SSMS-Clients sowohl in Bezug auf die enthaltenen Workflows als auch bezüglich ihrer grafischen Gestaltung dargestellt.

5.6.1.1. Startbildschirm und Anmeldeformular

Die Abbildung 5.6-1 zeigt die Startmaske der SSMS im Internet-Browser. Die Seite wird in diesem Beispiel über eine sichere Internetverbindung mit der Adresse *https://IT-System.SSMS* aufgerufen.

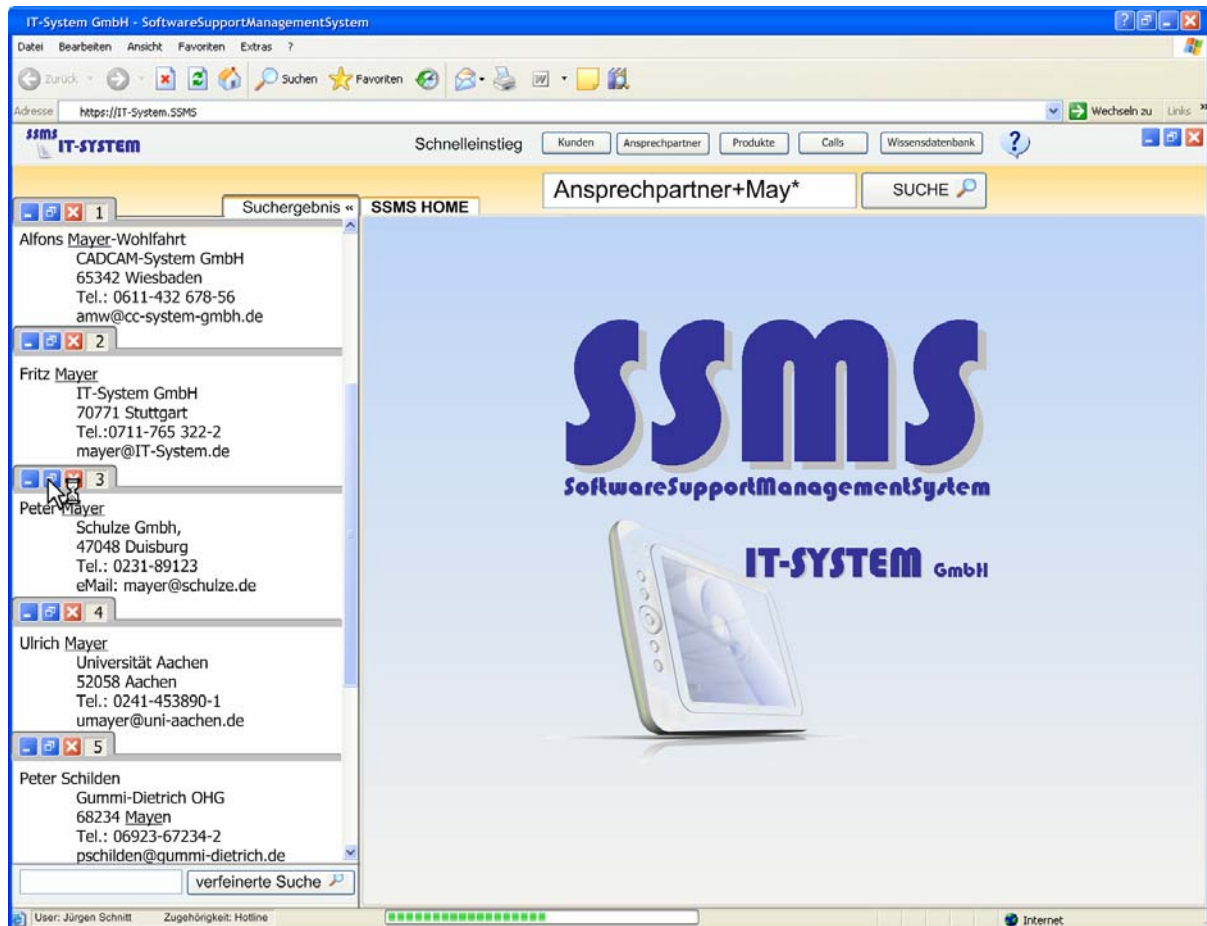


Abbildung 5.6-1: Startbildschirm mit Suchabfrage auf Ansprechpartner und Ergebnisliste

Dargestellt ist die Suchabfrage auf einen Ansprechpartner mit dem Namen Mayer bei einem Kunden. Da der Servicetechniker am Telefon den Nachnahmen nicht genau verstanden hat, gibt er anstatt der fehlenden Buchstaben eine Wildcard * ein. Die linke Spalte zeigt das Suchergebnis, wobei die Übereinstimmung im Datensatz unterstrichen ist. Nachdem der Servicetechniker die Firma erfragt hat, öffnet er den Datensatz von Peter Mayer von der Schulze GmbH.

5.6.1.2. Das Kundeninformations-Interface

Die Abbildung 5.6-2 zeigt das KI-Interface für Kunden und Ansprechpartner. Außerdem stellt das SSMS die sich in Bearbeitung befindlichen CALLs von Herrn Mayer in einem eigenen Fenster links neben dem Ansprechpartner dar. Der Techniker hat hier die Möglichkeit, die angezeigten CALLs noch weiter zu filtern, z.B. nach Stichworten oder der CALL-ID. Nachdem der gewünschte CALL gefunden wurde, wird dieser vom Anwender durch Anklicken geöffnet.

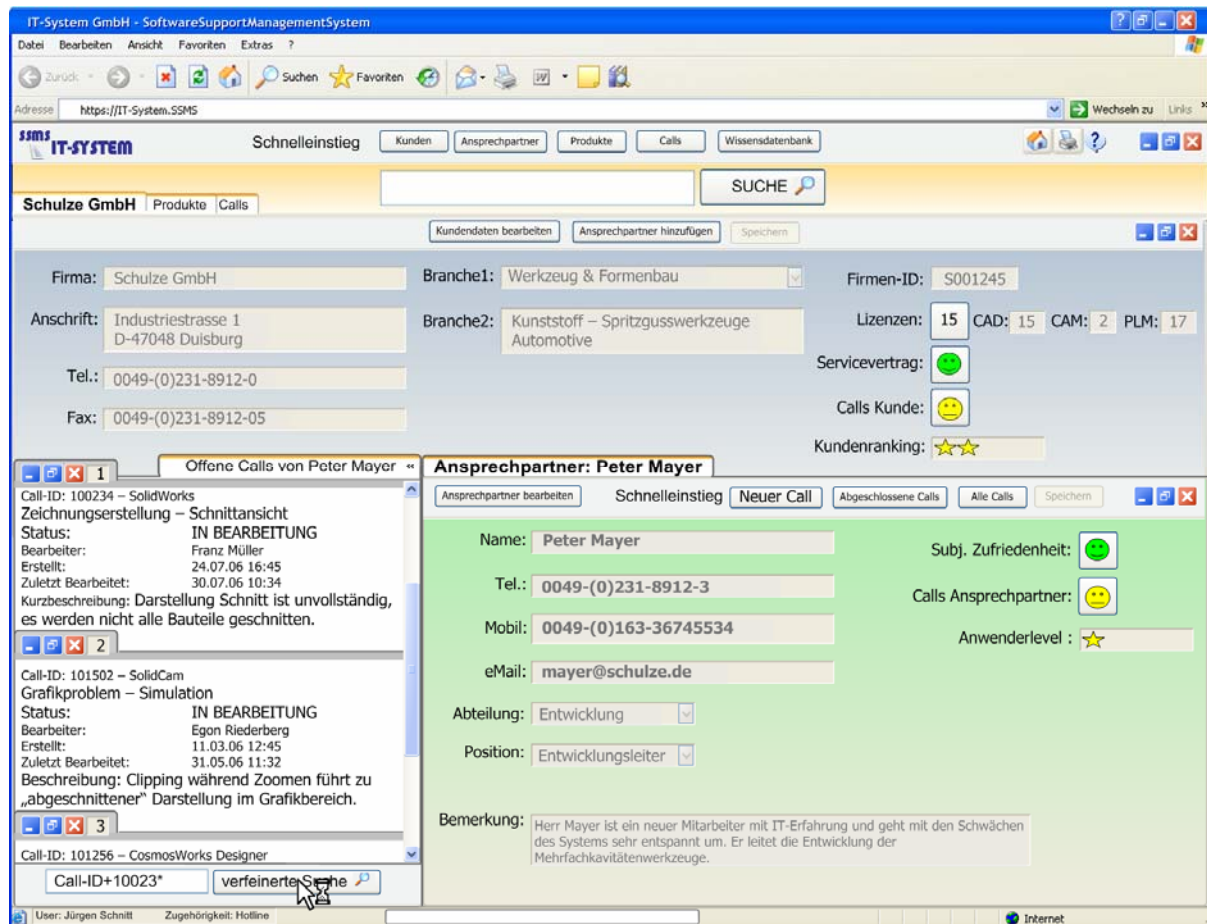


Abbildung 5.6-2: KI-Datensatz mit Absprung zum CALL-Handling

Der Bearbeitungsstatus der unterschiedlichen Bereiche wird durch graues Überblenden schreibgeschützter Felder gegenüber weißem Hintergrund für in Bearbeitung befindliche Felder visualisiert.

5.6.1.3. Das CALL-Handling-Interface und Wissensdatenbank-Interface

Die Abbildung 5.6-3 zeigt das CALL-Handling-Interface mit einer Ergebnisliste der Wissensdatenbank in der linken Spalte. Unter den KB-Artikeln werden sämtliche Datenbankeinträge zur gegenwärtig den gewählten Kategorien des CALL dargestellt. Der Servicetechniker kann zudem die Ergebnisliste nach zusätzlichen Stichwörtern durchsuchen und das Suchergebnis somit verfeinern. Das eigentliche CALL-Handling-Interface findet sich im rechten Bereich des GUI. Hier werden alle notwendigen, im Rahmen eines CALLs anfallenden Informationen gespeichert. Die grünen Haken verdeutlichen dem User, dass die jeweiligen Prozessschritte abgearbeitet wurden. Im obersten Bereich der CALL-Karteikarte befindet sich der Bereich mit den mit der Anfrage assoziierten System- und Lizenzinformationen sowie den Menüs für Status, Haupt- und Subkategorien.

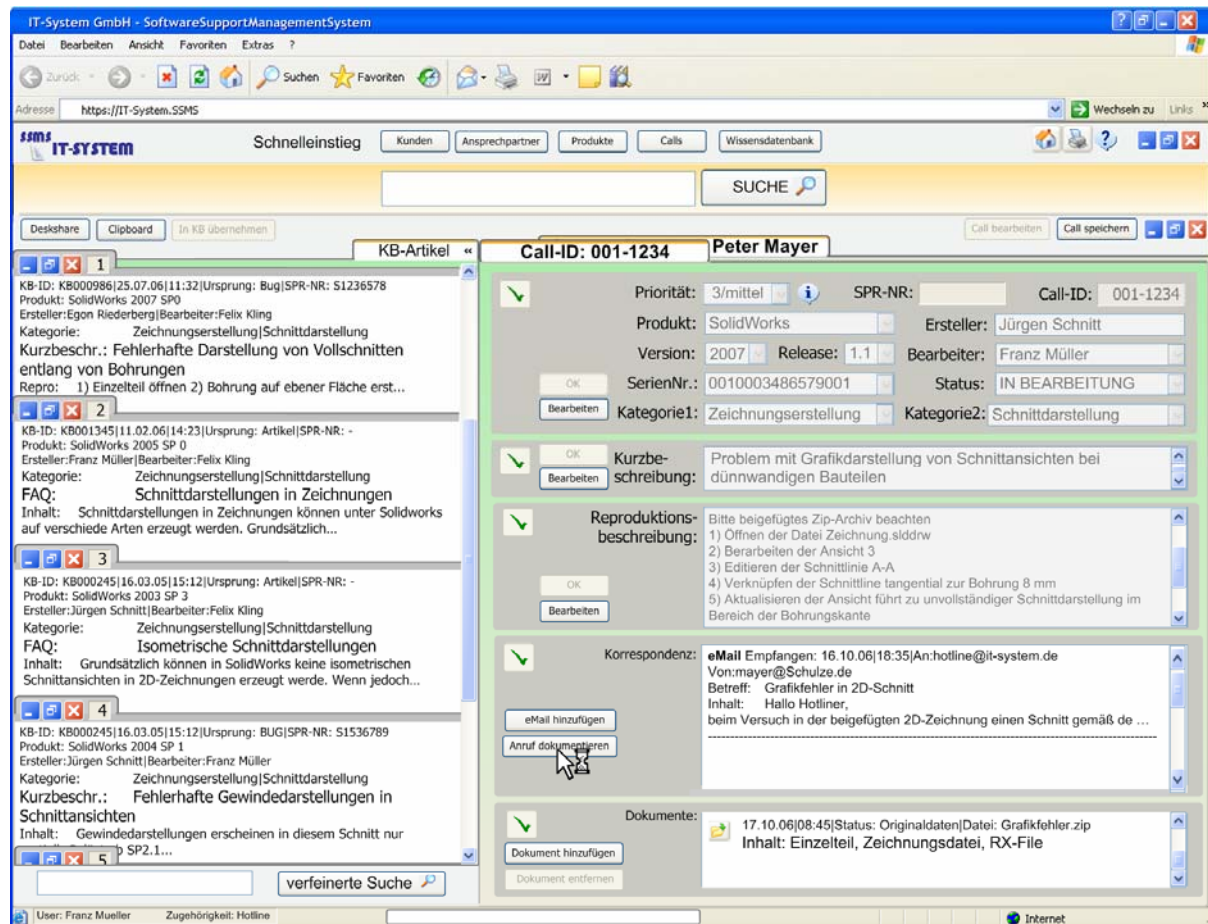


Abbildung 5.6-3: KB-Interface mit Ergebnisliste und CALL-Erfassung

Diese Pulldown-Menüs repräsentieren den Inhalt der im Anhang unter **Kapitel 9.3** befindlichen Kategorienlisten. Sind die notwendigen Selektionen getroffen bzw. die Felder ausgefüllt, so bestätigt der Anwender mit „OK“, worauf der nächste Prozessschritt aktiviert wird und der grüne Haken erscheint. Auf diese Art und Weise arbeitet sich der Servicetechniker von oben nach unten durch das CALL-Handling-Menü, wobei er stets durch die hinterlegten Workflows geleitet wird.

Über der Karteikarte des CALL befindet sich die Menüleiste mit den Befehlen zum Speichern des CALL oder zum aktivieren eines Fernwerkzeugstools und des Clipboards. Der Button „In KB übernehmen“ ist deaktiviert, da der CALL gegenwärtig noch im Status „in Bearbeitung“ ist. Alle Funktionsbuttons und Menüfelder sind Fallsensitiv, d.h. sie sind nur aktiv, wenn dies in der gegenwärtigen Situation Sinn ergibt.

Der Servicetechniker ist gerade im Begriff, den Anruf des Ansprechpartners unter „Korrespondenz“ zu dokumentieren. Ebenso können eMails über den Button „eMail hinzufügen“ aus dem eMail-Client importiert, und zur CALL-Dokumentation hinzugefügt werden.

6. Zusammenfassung

In der vorliegenden Arbeit wurden ausführlich die Schwierigkeiten und Defizite beim Support von komplexen Applikationen, wie sie 3D-CAD-Anwendungen darstellen, beschrieben. Im Rahmen der Bestandsanalyse wurden hierbei die Vorgehensweisen sowohl im Support von Eigenentwicklungen als auch von extern entwickelten Applikationen betrachtet.

Seit einiger Zeit sind im Umfeld von 3D-CAD-Systemen Entwicklungstrends von isolierten Insellösungen hin zu integrierten Entwicklungssystemen mit dem Ziel der Abdeckung des gesamten Produktlebenszyklus zu beobachten. Innerhalb solcher Systeme werden sämtliche, im Rahmen der produktrelevanten Geschäftsprozesse eingesetzten Applikationen (ERP, PLM, CAM, FEM etc.) miteinander verknüpft und deren Datenbasis über eine gemeinsame Zugriffsebene allen Involvierten zugänglich gemacht.

In gleichem Maße wie sich die Komplexität solcher Systeme durch diese Entwicklung erhöht, steigen auch die Anforderungen an den Softwaresupport für die integrierten Applikationen sowie das Gesamtsystem und gehen dabei weit über die ursprünglichen Aufgaben wie Softwarepflege und –Instandhaltung hinaus. Vielmehr nimmt der Software-Support im Kontext dieser Entwicklung eine Schlüsselposition ein, da er die Herstellung und Aufrechterhaltung von maximaler Produktivität durch den optimalen Einsatz dieser Applikationen erst ermöglicht.

Der mit dem Support betraute Applikations-Ingenieur sieht sich einer wachsenden Flut von Einsatzgebieten und Funktionalitäten der zu betreuenden Software-Produkte gegenüber. Durch die Tatsache, dass dadurch der Aufwand zur Erbringung der Supportdienstleistung ständig zunimmt, vom Kunden aber lediglich mit einem pauschalen Fixbetrag vergütet wird, erhöht sich gleichwohl der wirtschaftliche Druck zur Optimierung der Supportprozesse. Die gegenwärtig zu diesem Zweck eingesetzten Software-Tools mit redundanter Datenhaltung und dem Fehlen einer zentralen Zugriffsebene können diesen gestiegenen Anforderungen jedoch nicht mehr gerecht werden.

Die aus der Analyse abgeleiteten Defizite bezüglich der Arbeitsabläufe und der dabei eingesetzten IT-Werkzeuge führen zu der Erkenntnis, dass eine Optimierung des gesamten Supportprozesses durch Schaffung eines geeigneten Software-Support-Management-Systems eine unabdingbare Voraussetzung für die zukünftige erfolgreiche Behauptung aller Beteiligten am Markt ist. Diese Erkenntnis ergibt sich aus der wachsenden Komplexität der betreuten Applikationen sowie der parallel dazu steigenden Anforderungen zur Produktivitätssteigerung sowohl seitens der Systemhäuser als auch der Endkunden.

Das konzipierte SSMS wird dieser Anforderung im Wesentlichen dadurch gerecht, dass es sämtliche, innerhalb eines Systemhauses gesammelten, anfrage- oder applikations-

spezifischen Informationen verwaltet und jedem in den Supportprozess involvierten Mitarbeiter mit Hilfe einer einheitlichen Zugriffsebene zugänglich macht. Darüber hinaus werden die Supportprozesse durch deren Abbildung in unterschiedlichen Workflows wesentlich beschleunigt und vereinheitlicht. Anhand dieser Workflows wird der Supporttechniker durch die Standardprozesse einer Anfrage geführt. Das integrierte Wissensmanagement fungiert in diesem Kontext als Multiplikator für die ständig wachsenden Erkenntnisse im Umgang mit den betreuten Applikationen und schützt die Support-Organisation gleichzeitig vor Kompetenzlücken durch Personalfluktuaton. Das SSMS gewährleistet jedem Support-Ingenieur durch die implementierte Datenhaltung den Zugriff auf sämtliche im Rahmen einer Anfrage zu betrachtenden Applikationsdateien und sorgt durch die Statusverwaltung für Supportanfragen für Transparenz im Supportprozess. Abgerundet wird die Funktionalität des SSMS durch ein automatisches Benachrichtigungssystem, welches sämtliche involvierte Personen über Änderungen des Bearbeitungsstandes einer Anfrage benachrichtigt.

6.1. Ausblick

Die Entwicklung der Informationstechnologie ist heute mehr denn je dem stetigen Wandel unterworfen, wodurch sich ständig neue Wege eröffnen. Gegenwärtig entstehen so beispielsweise Möglichkeiten, innerhalb eines laufenden Betriebssystems ein weiteres, beliebiges Computer-System mit Hilfe einer sog. Virtual-Machine zu simulieren. Zwar ist diese Technologie gegenwärtig noch durch starke Einbußen bei der Performance gekennzeichnet, in Anbetracht der konstanten Leistungssteigerung von Desktop-PCs (insbesondere in Zusammenhang mit der Zunahme von Mehrkern-Prozessoren) ist es nach dem Mooreschen Gesetz jedoch nur eine Frage der Zeit, bis mit Hilfe einer Virtual-Machine eine hinreichende Leistungsfähigkeit erreichbar ist. Damit könnten die spezifische Systemkonfiguration mit sämtlichen Programmen eines beliebigen PCs gespeichert und auf einer Workstation simuliert werden, um bestimmte Konstellationen auf dieser Plattform nachzustellen. Für den Support eröffnen sich dadurch Möglichkeiten, die individuelle Konfiguration eines Kunden-PCs nachzustellen, dessen Probleme auf einer identischen Plattform zu reproduzieren und Modifikationen an diesem System durchzuführen, ohne zunächst das Originalsystem zu verändern.

Eine ähnliche Entwicklung ist bei Fernwartungstools bereits Realität. Durch die zunehmende Verbreitung von Hochgeschwindigkeits-Internetanschlüssen können sich in vielen Fällen bereits heute die Support-Techniker auf den Desktop eines Kunden einloggen, um dort ein Problem in Echtzeit nach verfolgen zu können oder Lösungswege direkt zu präsentieren.

Das vorgestellte Konzept eines SSMS ist in diesem Zusammenhang nur als erster Entwicklungsschritt zu einer neuen Generation von IT-Systemen zu sehen. Das Ende dieser Entwicklung, ebenso wie das des technologischen Fortschritts in der gesamten Hard- und Software-Welt ist jedoch noch nicht absehbar.

7. Literaturverzeichnis

1. N.N.: *Methodik zum Entwickeln technischer Systeme und Produkte*, VDI-Richtlinie 2221, VDI-Verlag, Düsseldorf 1993
2. Lobeck, F.: *Konzept für ein objektorientiertes, bereichsübergreifendes Dokumenten-informations- und -verwaltungssystems*, Shaker Verlag, 1. Auflage, 1999
3. Trozki, T.; Baulig, M.: *Von C++ nach COM - Was bringt Komponenten-Software?*, DotNetPro, 05/2003, S. 132 ff.
4. Kemper, A.; Eickler, A.: *Datenbanksysteme / Eine Einführung*, Oldenburg Verlag, ISBN 3-486-24136-2, 2. Auflage 1997
5. Schütten, U.: *Konzept eines modulbasierten Engineerings in der Anlagenautomatisierung*, Universität Duisburg-Essen, Diss. 2003
6. Programmer`s Reference Delphi, Borland GmbH, Langen
7. Kosch, A.: *Crashkurs .NET für Delphianer*, Software & Support Verlag GmbH, 2003
8. Stolpmann, J.: *Konzeption eines Software-Lifecycle-Managementsystems (SML) zur Unterstützung und Beschleunigung von Softwareentwicklungsprozessen*, Universität Essen, Diss., 2003
9. ISO/IEC 9579, Ausgabe: 2000-02 Informationstechnik – *Fernzugriff auf Datenbanken für SQL mit Sicherheitserweiterungen*
10. Balzert, H.: *Lehrbuch der Softwaretechnik/ Band 1/ Software-Entwicklung*, Spektrum Verlag, ISBN 3-8274-0042-2, 1998
11. Kutter, U.: *Konzept eines rechnergestützten Order-Entry-Systems unter der Berücksichtigung einer vorhandenen EDV-Umgebung und dessen Umsetzung*, Universität Duisburg-Essen, Diss. 2003
12. ISO/IEC 9075-1, Ausgabe: 1999-12 Informationstechnik – *Datenbanksprachen - SQL – Teil 1: Rahmenwerk (SQL/Rahmenwerk)*
13. Stracke, H.J.: *Betriebsdatenverarbeitung I / (C++)*, Seminarunterlage, Universität Essen, 2001
14. ANSI / IEEE Std. 729-1983, *IEEE Standard Glossary of Software Engineering Terminology*, New York: IEEE Inc., 1983
15. Rainer Dumke: *Software Engineering*, Vieweg Verlag, ISBN 3-528-25355-X, 2001
16. <http://www.wikipedia.de/> [02-2006]
17. Becker, Kugeler, Rosemann: *Prozessmanagement – Ein Leitfaden zur*

prozessorientierten Organisationsgestaltung, Springer-Verlag, ISBN 3-540-23493-4, 2005

18. George Shepherd: Microsoft ASP.NET – Schritt für Schritt, Microsoft Press ISBN 3-86063-553-0, 2006

8. Abbildungsverzeichnis

ABBILDUNG 2.3-1: CAD-ANWENDUNG (FALLBEISPIEL 1).....	19
ABBILDUNG 2.3-2: CAD-ANWENDUNG (FALLBEISPIEL 2).....	24
ABBILDUNG 2.3-3: CAD-ANWENDUNG (FALLBEISPIEL 3).....	29
ABBILDUNG 2.3-4: CAD-ANWENDUNG (FALLBEISPIEL 4).....	32
ABBILDUNG 5.1-1: N-TIER-ARCHITEKTUR DES SSMS	79
ABBILDUNG 5.2-1: NETZWERKSTRUKTUR DES SSMS	81
ABBILDUNG 5.3-1: EPK-LEGENDE	84
ABBILDUNG 5.3-2: PD FÜR DIE ERSTELLUNG UND BEARBEITUNG VON KUNDENINFORMATIONEN..	85
ABBILDUNG 5.3-3: PD FÜR DIE ERSTELLUNG UND BEARBEITUNG VON KB-ARTIKELN	87
ABBILDUNG 5.3-4: PD GESAMTPROZESS CALL-HANDLING.....	90
ABBILDUNG 5.4-1: KONTEXTDIAGRAMM DER ZU VERWALTENDEN DATEN	92
ABBILDUNG 5.4-2: DATENSICHT DES SPEICHERNS VON KUNDENINFORMATIONEN	96
ABBILDUNG 5.4-3: DATENSICHT DES DURCHSUCHENS DER DATENBANK NACH KUNDENINFORMATIONEN	98
ABBILDUNG 5.4-4: DATENSICHT DER DARSTELLUNG VON ABFRAGEERGEBNISSEN FÜR KUNDENINFORMATIONEN	99
ABBILDUNG 5.4-5: DATENSICHT DES ERFASSENS UND SPEICHERNS VON CALL-INFORMATIONEN	101
ABBILDUNG 5.4-6: DATENSICHT DER VERSENDUNG AUTOMATISCHER NOTIFIKATIONEN.....	106
ABBILDUNG 5.4-7: DATENSICHT DER FORMATIERUNG UND ÜBERGABE VON CALL- INFORMATIONEN AN DAS CLIPBOARD	107
ABBILDUNG 5.4-8: DATENSICHT DER INITIIERUNG DER ÜBERNAHMEN VON CALL- INFORMATIONEN IN DIE KB.....	110
ABBILDUNG 5.4-9: DATENSICHT DER ÜBERNAHME EINES ABGESCHLOSSENEN CALLS IN DIE KB	110
ABBILDUNG 5.4-10: ERSTELLEN UND SPEICHERN VON KB-ARTIKELN	112
ABBILDUNG 5.4-11: DATENSICHT DES ERSTMALIGEN SPEICHERNS VON APPLIKATIONSDATEIEN	117
ABBILDUNG 5.4-12: DATENSICHT DER BEREITSTELLUNG VON APPLIKATIONSDATEIEN.....	119
ABBILDUNG 5.4-13: DATENSICHT DES SPEICHERNS VON MIT KB-EINTRÄGEN ASSOZIIERTEN APPLIKATIONSDATEIEN	121
ABBILDUNG 5.4-14: DATENSICHT DES BEREITSTELLENS VON MIT KB-EINTRÄGEN ASSOZIIERTEN APPLIKATIONSDATEIEN	123
ABBILDUNG 5.5-1: SOFTWARE-ARCHITEKTUR DES SSMS.....	125
ABBILDUNG 5.5-2: INTERNER AUFBAU DES DB-MODULS.....	127
ABBILDUNG 5.5-3: AUSZUG AUS DEM KLASSENDIAGRAMM DES SSMS	129
ABBILDUNG 5.5-4: AUSZUG AUS DEM OBJEKTDIAGRAMM FÜR DEN PROZESS CALL-ERFASSEN.	131
ABBILDUNG 5.6-1: STARTBILDSCHIRM MIT SUCHABFRAGE AUF ANSPRECHPARTNER UND ERGEBNISLISTE	134
ABBILDUNG 5.6-2: KI-DATENSATZ MIT ABSPRUNG ZUM CALL-HANDLING	135
ABBILDUNG 5.6-3: KB-INTERFACE MIT ERGEBNISLISTE UND CALL-ERFASSUNG MIT KATEGORISIERUNG.....	136
ABBILDUNG 9.2-1: PD FÜR DIE ANNAHME EINES TELEFONISCHEN CALLS.....	145
ABBILDUNG 9.2-2: PD FÜR DIE DISPONIERUNG EINER EMAIL-ANFRAGE.....	146
ABBILDUNG 9.2-3: PD FÜR DIE ERFASSUNG EINES NEUEN CALLS	147
ABBILDUNG 9.2-4: PD FÜR DIE BEARBEITUNG EINES CALLS.....	148
ABBILDUNG 9.2-5: PD FÜR DIE VERIFIKATION EINES CALLS DURCH DEN 1.LEVEL-SUPPORT.....	149
ABBILDUNG 9.2-6: PD FÜR DIE VERFOLGUNG EINES CALLS DURCH DEN 2.LEVEL-SUPPORT.....	150
ABBILDUNG 9.2-7: PD FÜR DAS ABSCHLIEßEN EINES CALLS DURCH DEN 1.LEVEL-SUPPORT....	151

9. Anhang

9.1. Fragenkomplex der analysierten Praxisbeispiele

Unternehmen

- Um was für ein Unternehmen handelt es sich?
- Auf welchen Märkten betätigt sich das Unternehmen?
- Wie viele Mitarbeiter sind in den unterschiedlichen Bereichen Entwicklung, Support, Vertrieb tätig?

Software

- Welche Art(en) von Software werden entwickelt?
- Standalone oder im Zusammenhang mit einem speziellen Produkt?
- Ist die Software eigenständig oder ein Zusatzmodul zu einem anderen Produkt?
- Welches Leistungsspektrum weist die entwickelte Software auf?
- Auf welche Marktsegmente ist die Software zugeschnitten?

Aufgabenverteilung

- Wie sind die Zuständigkeiten für Applikationstests verteilt?
- Auf welchen Ebenen wird getestet?
- Welche Aufgabe übernimmt der Support hierbei?
- Welche Möglichkeiten haben die Kunden, Anfragen zu stellen?
- Wie werden die Anfragen auf die Supportmitarbeiter verteilt?
- Wie sind intern und extern die Zuständigkeiten geregelt?
- Welche externen und internen Kommunikationsmittel /-wege werden genutzt?
- Bei welchen Kommunikationsmitteln liegen die Schwerpunkte?
- Welche Werkzeuge kommen dabei zum Einsatz?

Prozessmanagement und Qualitätssicherung

- Wie werden die unterschiedlichen Prozesse bei der Problembearbeitung initiiert?
- Wie werden Anfragen / Kunden priorisiert?
- Nach welchen Regeln werden die Probleme kategorisiert?
- Nach welchen Regeln werden die Probleme dokumentiert?
- Nach welchen Regeln werden Probleme verifiziert?
- Welche anfragenbezogenen Softwaredaten werden gesammelt?
- Wie und wo werden Dokumentation und Softwaredaten gespeichert?
- Wie ist der Zugriff auf diese Informationen organisiert?
- Wie werden Verbesserungsvorschläge aufgenommen?
- Wie werden die Verbesserungsvorschläge an die Entwicklung übermittelt?
- Welche Mechanismen zur Kontrolle und Nachverfolgung von Anfragen werden genutzt?
- Welche Werkzeuge kommen dabei zum Einsatz?

Informationsmanagement

- Welche Informationen hat der Support bezüglich der Kunden?
- Wie sind diese Informationen abrufbar?
- Wo werden diese Informationen gespeichert?
- Welche Werkzeuge kommen dabei zum Einsatz?

Wissensmanagement

- Wie wird das Wissen innerhalb und außerhalb des Supports distribuiert?
- Wo wird das Wissen gespeichert?
- Wie sind Kunden und Support in den Aufbau von Expertenwissen eingebunden?
- Wie wird der Zugriff auf das Wissen organisiert?
- Welche Werkzeuge kommen dabei zum Einsatz?

Datenhaltung

- Wie werden die Informationen aus den Bereichen Aufgabenverteilung, Prozessmanagement, Informationsmanagement, Wissensmanagement gespeichert und abgeglichen?
- Wie kann auf die Daten zugegriffen werden?
- Gibt es ein einheitliches Portal für den Zugriff auf die Informationen aus allen supportrelevanten Bereichen?
- Wie werden die Zugriffsrechte vergeben, wer darf auf welche Daten zugreifen?

9.2. Prozessdiagramme

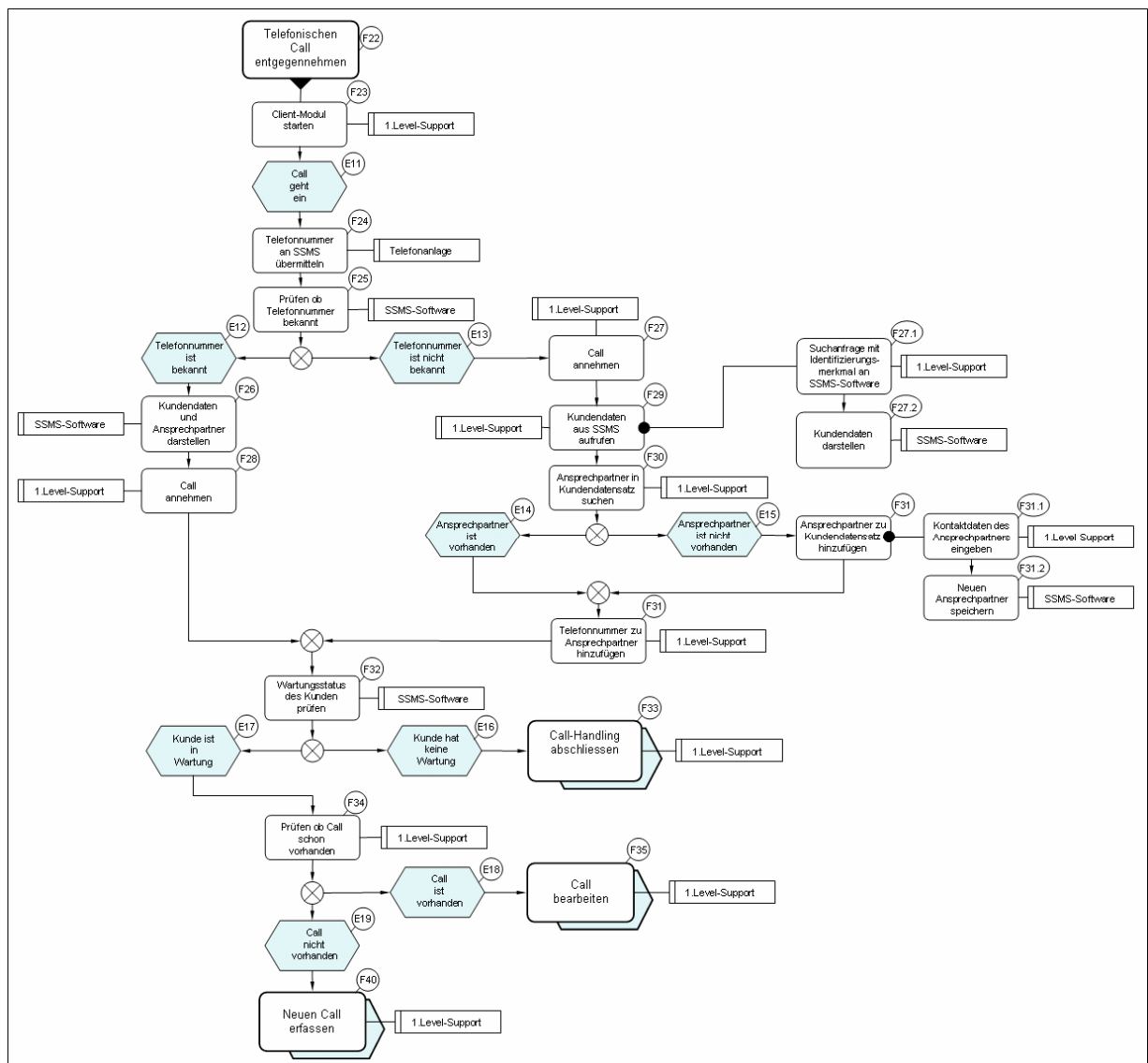


Abbildung 9.2-1: PD für die Annahme eines telefonischen CALLs

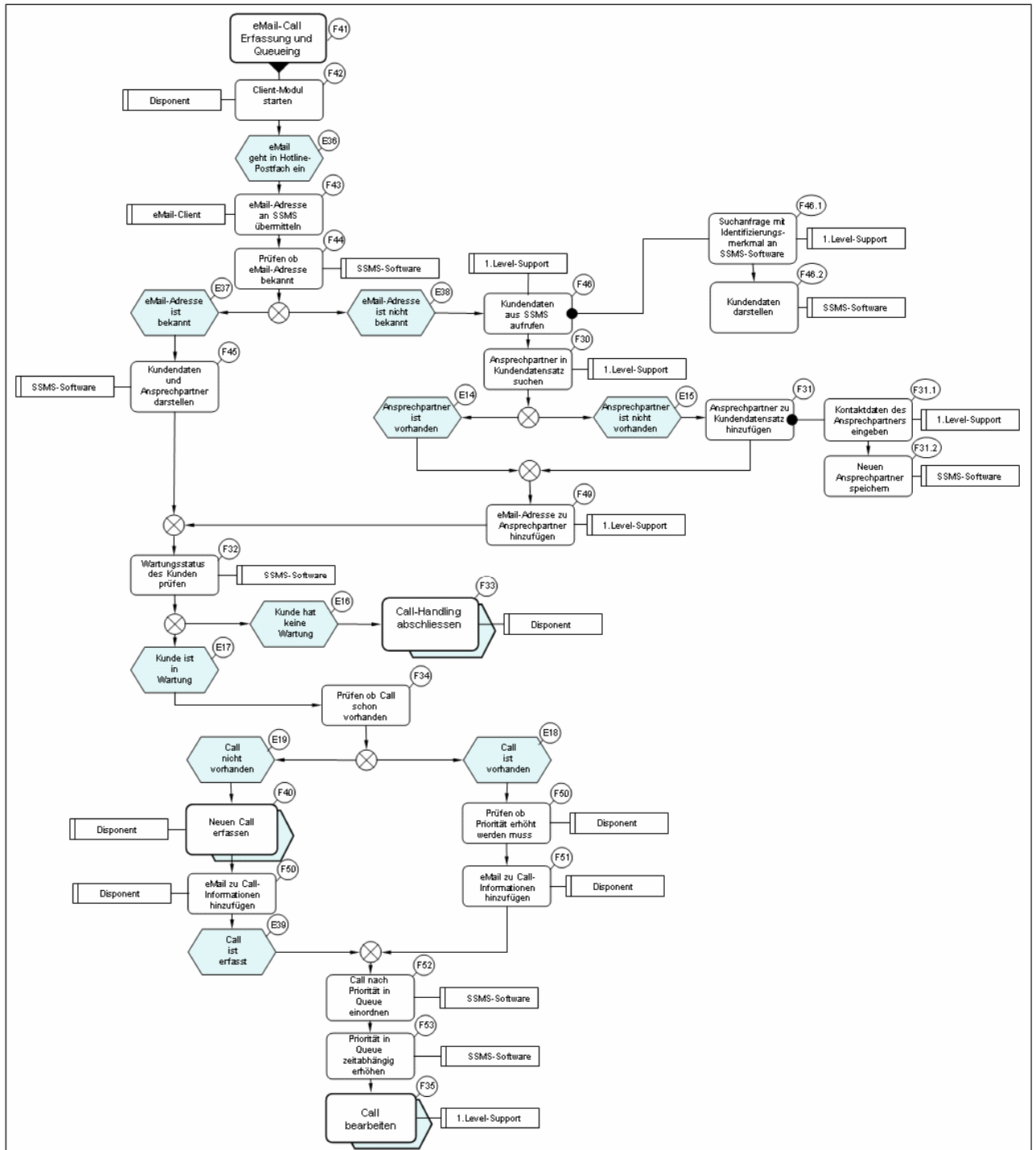


Abbildung 9.2-2: PD für die Disponierung einer eMail-Anfrage

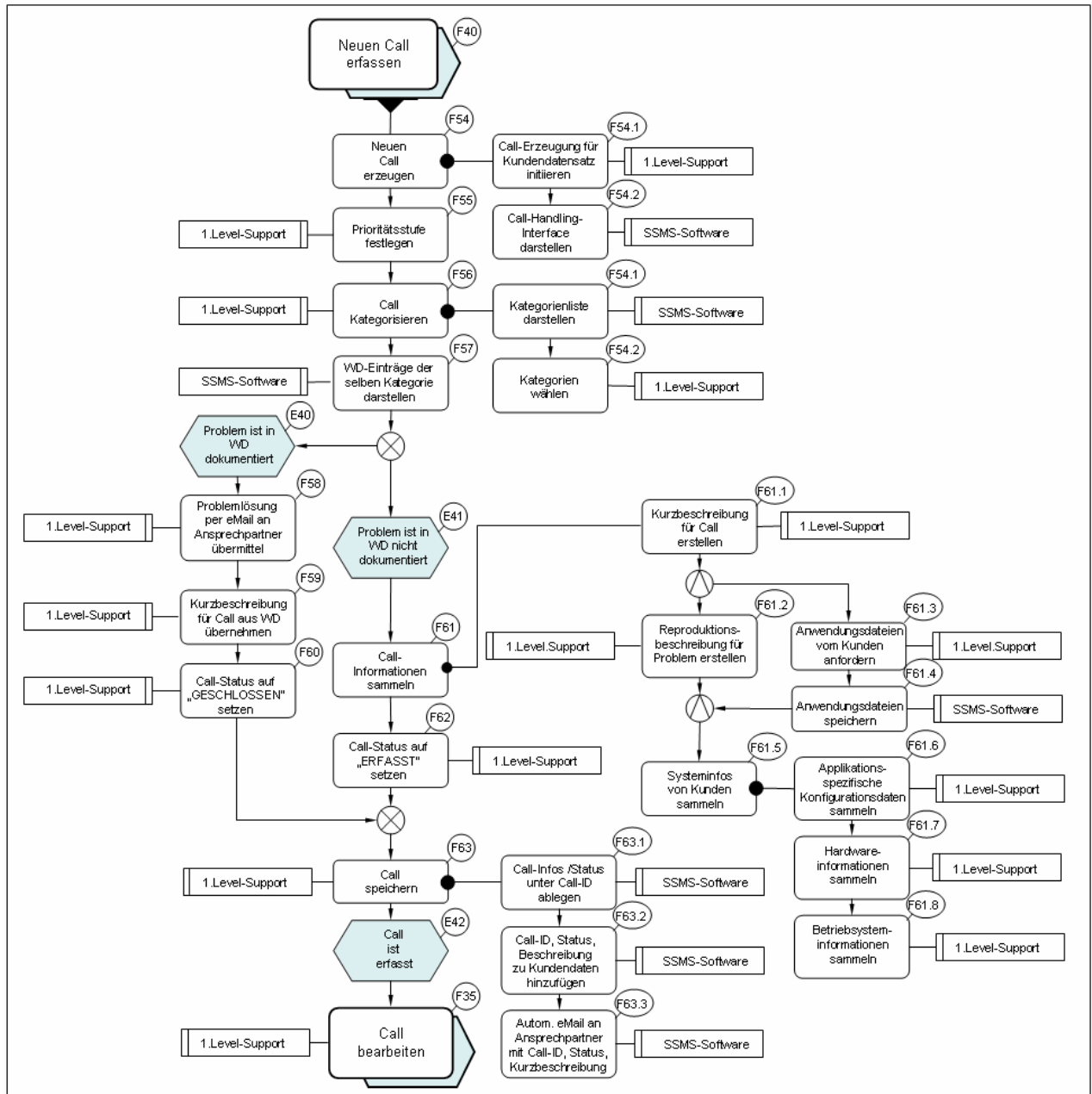


Abbildung 9.2-3: PD für die Erfassung eines neuen CALLS

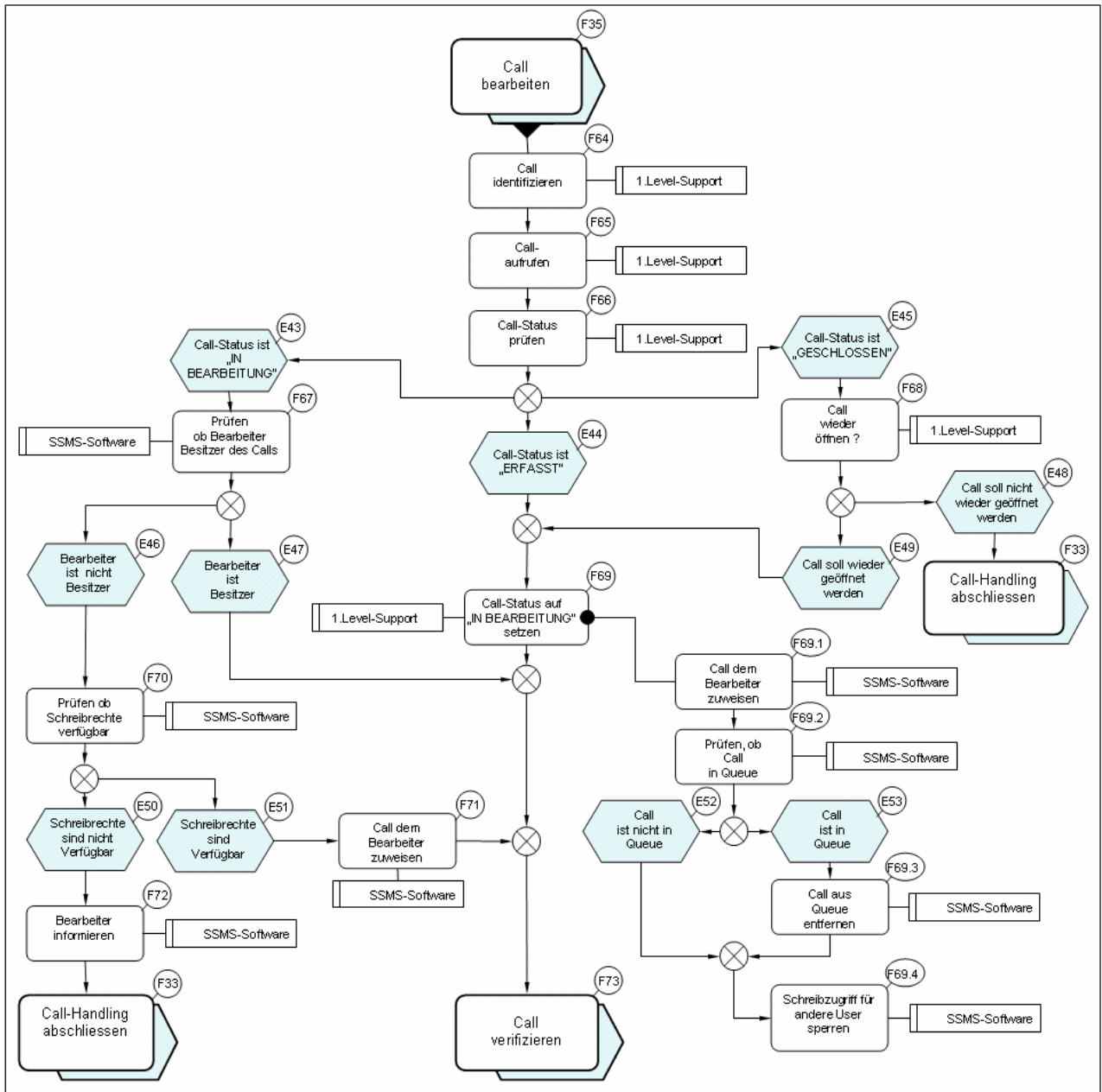


Abbildung 9.2-4: PD für die Bearbeitung eines CALLs

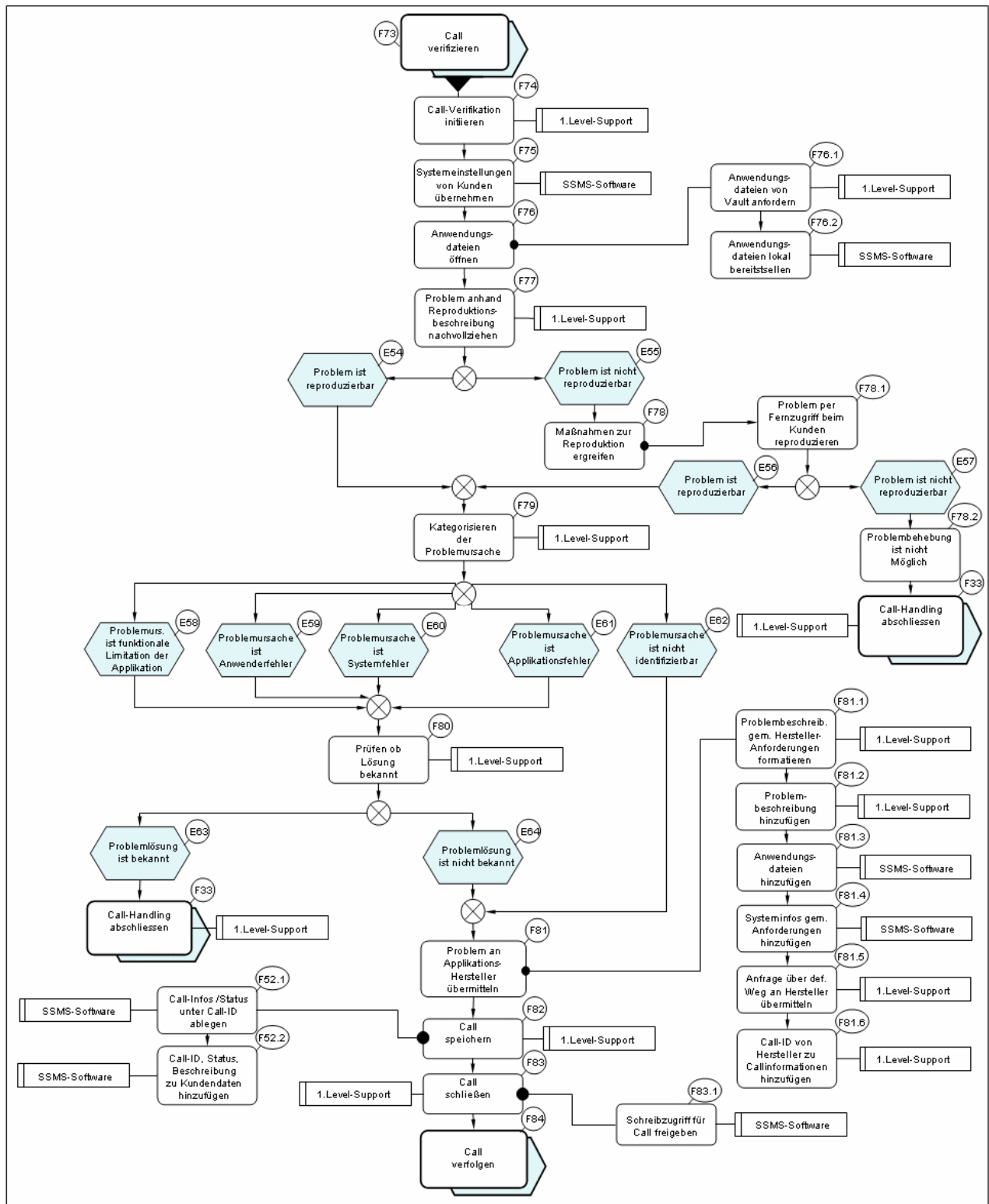


Abbildung 9.2-5: PD für die Verifikation eines CALLs durch den 1.Level-Support

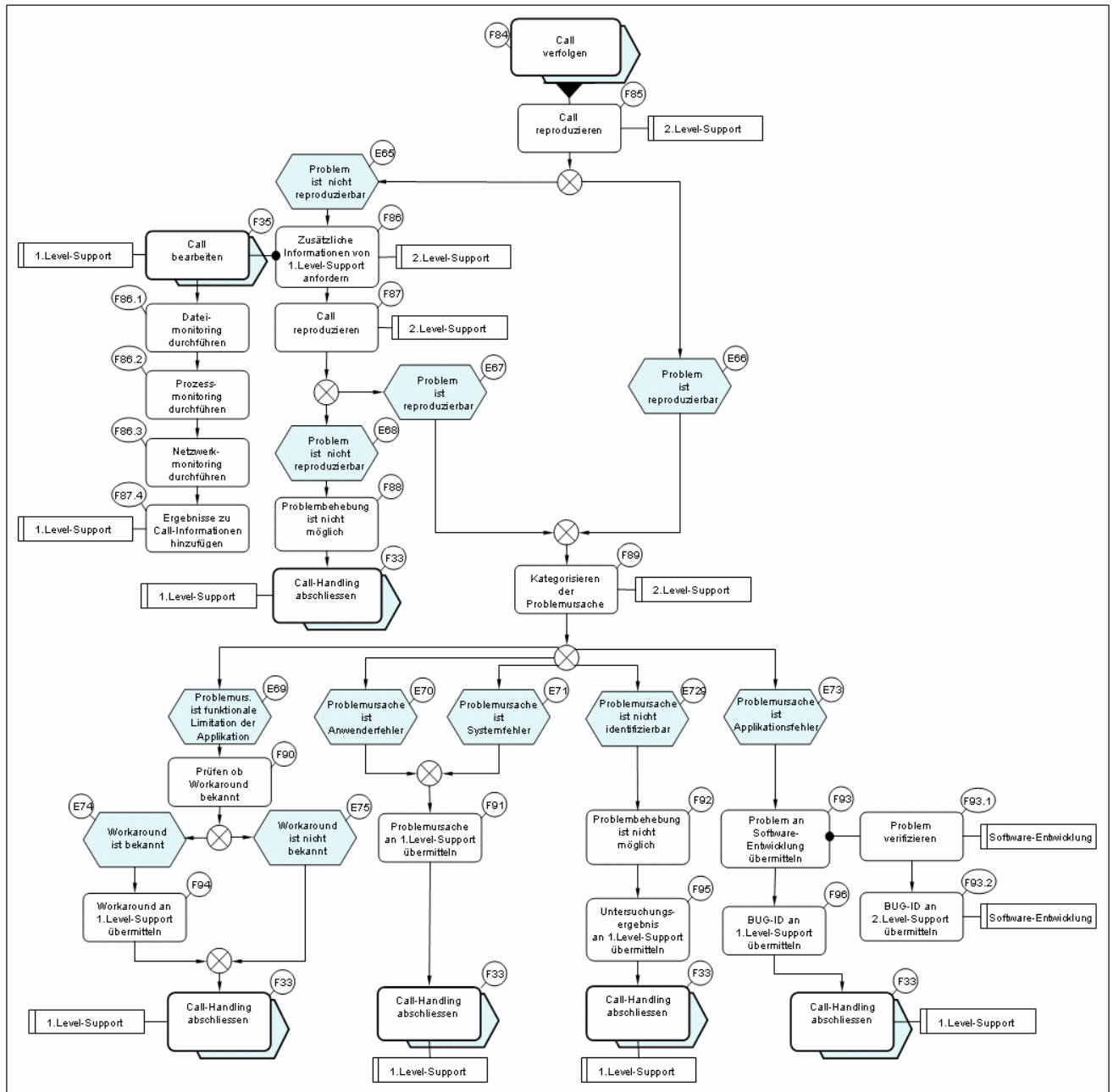


Abbildung 9.2-6: PD für die Verfolgung eines CALLs durch den 2.Level-Support

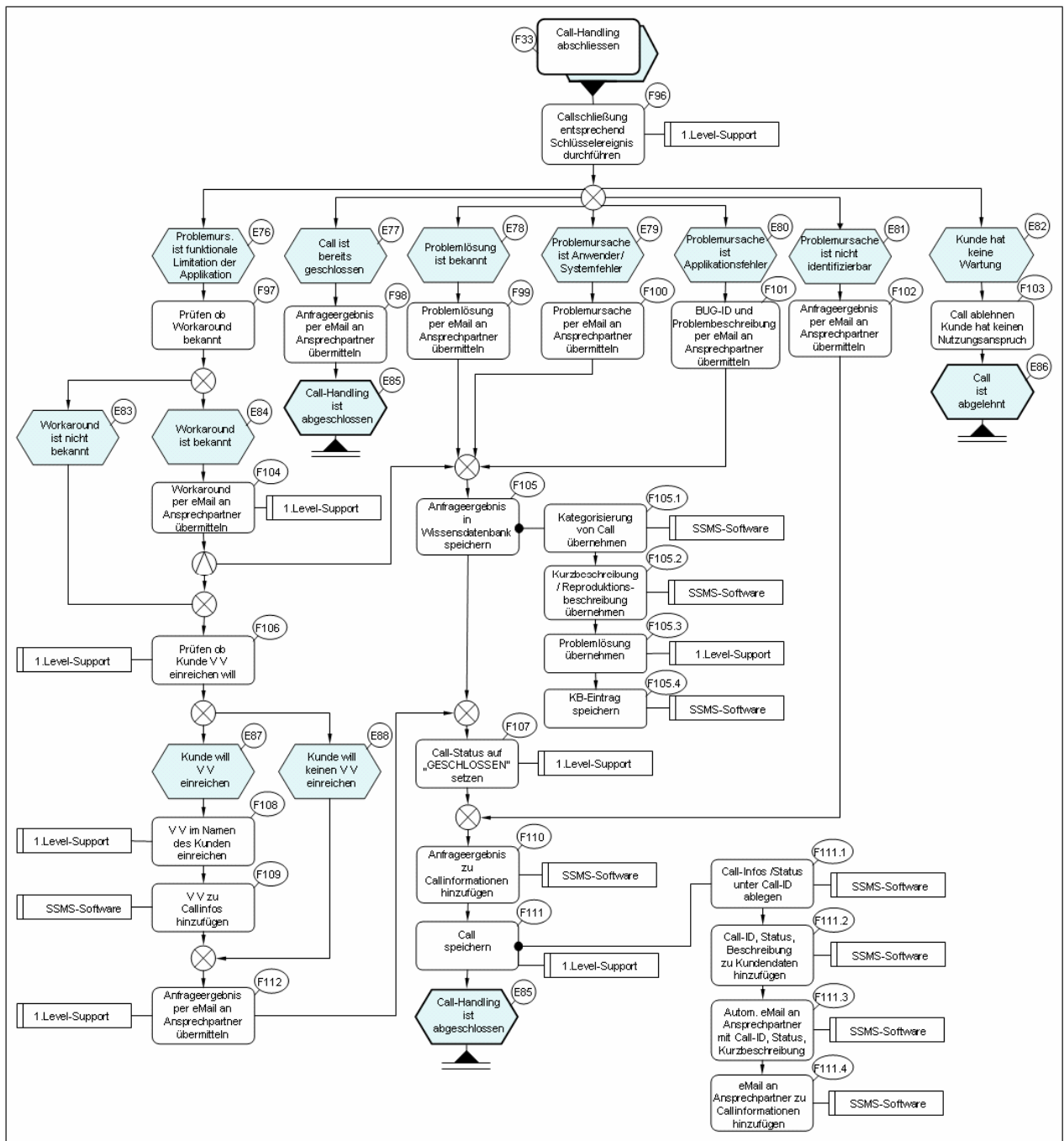


Abbildung 9.2-7: PD für das Abschließen eines CALLs durch den 1.Level-Support

9.3. Kategorienliste

<u>Area</u>	<u>Sub-Area</u>	<u>Area</u>	<u>Sub-Area</u>
Add-Ins	3D Content Central 3D InstantWebsite 3D Meeting Animator COSMOSWorks Designer Design Checker eDrawings/Pro FeatureWorks MoldBase PDMWorks PhotoWorks PhotoWorks2 Piping/Routing Toolbox ThirdParty Utilities	Data Translation	ACIS (SAT) CADKEY CATIA CGR DXF/DWG IGES Import Diagnosis Inventor MDT Parasolid PDF ProE Converter SolidEdge STEP AP203/214 STL UG VDAFS VRML
Annotations	Annotation Views Cosmetic Thread Datum GTOL Note Surface Finish Weld	Dimension	Alignment Equations Hide/Show Math Operator Ordinate Placement/Movement Tolerance Value
Analysis/Mass Props	CosmosXpress MassProps Measure Section Properties	Display	Curvature Fast HLR/HLV Hide/Show HLR / HLV Lighting/Camera Pan/Rotate/Zoom Perspective Real View Section Shaded Viewports Wireframe
Assembly	Cavity/Join Components Dynamic Motion Explode In-context Large Assembly Mode Lightweight Mates Move/Rotate Patterns/Mirror Simulation Smart Fasteners Weld	Drawings	Balloon Block BOM Callout CenterMark Dimension Drawing Views Font (Line/Note) Layers RapidDraft Sheet Formats
API/Macro	API Journal File Macro		
Configuration	Assembly Derived Configuration		

<u>Area</u>	<u>Sub-Area</u>	<u>Area</u>	<u>Sub-Area</u>
Free Products	DWG Editor eDrawings Viewer for Mac eDrawings Viewer Personal Edition Screensaver Student Edition SW Explorer Viewer XchangeWorks DWGgateway	Webpage	Manufacturing Network Knowledge Base Subscription Support VAR Portal 3rd Party Software Ethernet Card
General Utilities	2d Emulator ConversionWizard CopyOptionsWizard FeaturePalette Task Scheduler RX Viewer	Network	
Hardware / OS	AMD Processor CPU Network - General Network - Novell OS Performance Printer/Plotter RAM Video card Crash OS - Windows XP X64	Other	Other
		Part	Base/Mirror Part Draft Extrude Fastening Features Fillet/Chamfer Hole Library Feature Loft/Sweep Materials Multi-body Other Features ParentChild Pattern/Mirror Revolve Scale Sheet Metal Shelling Smart Components Split Part
Install	Activation CD Request Dongle Education InstallShield Registration SolidNetWork License Service Pack Windows Installer Admin Image Client Update Active Directory Admin Director Silent Installation	Reference Geometry	Axes Coordinate System Curves Planes Surfaces
		Regen/Rollback	Rebuild/Regen Rollback Undo
		Save/Open	Auto Recover Assembly Backup Files Drawing External References Part Startup
Installation	3rd Party Software CD Rom Disk Drive Operating System Memory Administration Navigation	Sketch	2d Sketch 3d Sketch Dimension Relations
Area	Sub-Area		

Transferred	API Support SolidWorks SolidWorks North America Technical Support Technical Support- Germany Customer Center
Upgrade	CD ROM Disk Drive Operating System Memory
Usage	End User Administrator Server Tasks Partner Apps
User Interface	Behavior Buttons Customization Custom Properties Document Templates Error Messages FeatureManager Foreign Language Layout Navigation OLE Online Help / Docs Print/Plot Search/Find/Query Terminology Web Template

10. Abkürzungsverzeichnis

ADO – Active Data Objects
ANSI - American National Standards Institute
Bug - Programmfehler einer Software
CAD - Computer Aided Design
CAE – Computer Aided Engineering
CAM - Computer Aided Manufacturing
CD-ROM - Compact Disc Read-Only Memory
CFD – Computational Fluid Dynamics
CRM - Customer Relationship Management
DVD - Digital Versatile Disc
eMail - elektronische Post
EPK – Ereignisgesteuerte Prozesskette
FAQ – Frequently asked Questions
FEM - Finite-Elemente-Methode
FTP - File Transfer Protocol
ID - Identifikationskennung
IEC - International Electrotechnical Commission
IEEE - Institute of Electrical and Electronics Engineers
IMAP4 – Internet Message Access Protocol Version 4
IT - Informations-Technologie
KB - Knowledge Base
KI - Kunden-Information
LAN – Local Area Network
NC - Numerical Control
ODBC - Open Database Connectivity
OLE – Object Linking and Embedding
OS - Operating System, Betriebssystem
PC - Personal Computer
PDA – Personal Digital Assistant
PDM - Product Data Management
PDF – Portable Document Format der Fa. Adobe
PLM - Product Lifecycle Management
POP3 – Post Office Protocol Version 3
PP - Postprozessor
PPS - Produktions- Planungs- und -Steuerungssystem
QM - Qualitätsmanagement
QS - Qualitätssicherung
RPC - Remote Procedure Call
SLM - Software Lifecycle Management
SPR - Software Performance Report
SQL - Structured Query Language
SR - Service Request
SSMS - Software-Support-Management-System
VAR - Value Added Reseller, Softwarehändler
VDI - Verband deutscher Ingenieure
VV – Verbesserungs-Vorschlag
WEB - Internet
XML - Extensible Markup Language
ZIP - Dateiformat für komprimierte Daten der Firma WinZip International