

**Konzipierung und prototypische Entwicklung einer E-Mail Integration
auf Basis der Microsoft Exchange Web Services im Service Desk der
CAFM Software GEBman 10**

Bachelorarbeit
zur Erlangung des Abschlusses
Bachelor of Engineering
im Studiengang Informationstechnik

eingereicht von:
Zilewitsch, Paul

1. Gutachter: Dr. rer. nat. Dipl.-Chem. Hansi Schilling
2. Gutachter: B.Sc. Sebastian Schulze

Tag der Themenübergabe: 22.04.2016

Tag der Einreichung: 18.07.2016

Autorenreferat

ZILEWITSCH, Paul: Konzipierung und prototypische Entwicklung einer E-mail Integration auf Basis der Microsoft Exchange Web Services im Service Desk der CAFM Software GEBman 10, Berufsakademie Sachsen, Staatliche Studienakademie Dresden, Studiengang Informationstechnik, Bachelorarbeit, 2016.

41 Seiten, 42 Literaturquellen , 4 Anlagen.

Die KMS Computer GmbH entwickelt seit 2011 die Webanwendung GEBman 10. Diese Webanwendung besitzt über 40 Module. Eines davon ist das Modul Service Desk. Ziel der vorliegenden Arbeit ist es, Verbesserungsmöglichkeiten dieses Service Desk-Moduls festzuhalten. Dadurch könnte es als Service Desk-Softwarelösung des Supports der KMS Computer GmbH dienen und auf eine externe Lösung verzichtet werden. Hierfür werden verschiedene, auf dem Markt befindliche, Service Desk-Softwarelösungen analysiert und ausgewertet, um anschließend Verbesserungsmöglichkeiten für das Service Desk-Modul identifizieren zu können.

Außerdem soll die E-Mail Integration des Service Desk-Moduls erweitert werden. Dabei wird auf die bereits vorhandenen Funktionalitäten in GEBman 10 zurückgegriffen, die auf den Microsoft Exchange Web Services basieren.

Inhaltsverzeichnis

| | |
|--|------|
| Inhaltsverzeichnis | IV |
| Abbildungsverzeichnis | VI |
| Abkürzungsverzeichnis | VII |
| Tabellenverzeichnis | VIII |
| 1 Einleitung | 1 |
| 1.1 Die Software GEBman 10 von der KMS Computer GmbH | 1 |
| 1.2 Das Modul Service Desk | 2 |
| 1.3 Motivation der Arbeit | 2 |
| 1.4 Vorgehen bei der Arbeit | 2 |
| 2 Service Desk nach ITIL v3 | 4 |
| 2.1 Begriffsabgrenzung | 4 |
| 2.2 Unterschied zu Help Desk | 6 |
| 2.3 Aufgaben eines Service Desk | 7 |
| 2.4 Nachteile ohne Service Desk | 8 |
| 3 Analyse verschiedener Service Desk-Lösungen | 10 |
| 3.1 Schwerpunkte der Analyse und Ausgewählte Service Desk-Lösungen | 10 |
| 3.2 Freshdesk | 11 |
| 3.3 Desk.com | 13 |
| 3.4 Zendesk | 13 |
| 3.5 SysAid | 14 |
| 3.6 Fazit der Analyse | 16 |
| 4 Der Service Desk in GEBman 10 | 18 |
| 4.1 Aktuelle Umsetzung | 18 |
| 4.2 Verbesserungsmöglichkeiten des Service Desk Moduls | 21 |
| 4.3 Anforderungen der Erweiterung | 22 |
| 5 Microsoft Exchange Server | 23 |
| 5.1 Grundlagen | 23 |
| 5.2 Exchange Web Services | 25 |
| 5.2.1 Funktionsweise | 25 |
| 5.2.2 Derzeitige Verwendung in GEBman 10 | 26 |
| 6 Konzipierung | 27 |
| 6.1 Vorbetrachtung | 27 |

| | | |
|-----|--|----|
| 6.2 | Zielsetzung und Grundidee Module Service | 28 |
| 6.3 | UML - Modellierung | 29 |
| 6.4 | Sicherheitsaspekte | 33 |
| 7 | Umsetzung | 35 |
| 7.1 | Erweiterung des bestehenden Service Desk Moduls | 35 |
| 7.2 | Erläuterung einzelner Methoden | 36 |
| 7.3 | Testfälle | 37 |
| 7.4 | Fehlschläge/Erfahrungen | 38 |
| 7.5 | Erweiterungsmöglichkeiten der E-Mail Integration | 39 |
| 8 | Fazit | 40 |
| | Anhangsverzeichnis | 42 |
| | Anhang | 43 |
| | Literaturverzeichnis | 48 |

Abbildungsverzeichnis

| | |
|--|----|
| Abbildung 1: ITIL Service Lifecycle. | 5 |
| Abbildung 2: Single Point of Contact | 6 |
| Abbildung 3: GEBman 10 Service Desk Dashboard. | 19 |
| Abbildung 4: Aufbau einer klassischen RPC Verbindung. | 24 |
| Abbildung 5: RPC over HTTP Prinzip | 24 |
| Abbildung 6: EWS Funktionsumfang | 26 |
| Abbildung 7: Aktivitätsdiagramm der Erweiterung | 30 |
| Abbildung 8: Zustandsdiagramm der Erweiterung | 31 |
| Abbildung 9: Klassendiagramm der Erweiterung | 32 |
| Abbildung 10: Sicherheitsprobleme der Erweiterung | 33 |
| Abbildung 11: Verweise der EWS Managed API | 35 |
| Abbildung 12: Entwurfsmuster Fabrikmethode. | 36 |
| Abbildung 13: Aufruf des Module Service | 36 |

Abkürzungsverzeichnis

| | |
|----------|---|
| API | Application Programming Interface |
| CAFM | Computer-Aided Facility Management |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| ITIL | Information Technology Infrastructure Library |
| REST API | Representational State Transfer Application Programming Interface |
| SOAP | Simple Object Access Protocol |
| SPoC | Single Point of Contact |
| UML | Unified Modeling Language |
| XML | Extensible Markup Language |
| XSS | Cross-Site-Scripting |

Tabellenverzeichnis

| | |
|--|----|
| Tabelle 1: Bewertung der Service Desk-Lösungen | 16 |
| Tabelle 2: Übersicht der Status im Service Desk..... | 20 |
| Tabelle 3: Einige Eigenschaften für eine Meldung | 27 |
| Tabelle 4: UML-Diagrammtypen | 29 |

1 Einleitung

1.1 Die Software GEBman 10 von der KMS Computer GmbH

Die IT ist ein ständiger Begleiter im Arbeitsalltag eines jeden Unternehmens. Sie erleichtert nicht nur die Arbeit mit großen Datenmengen, sondern unterstützt auch wichtige Geschäftsprozesse in -und außerhalb des Unternehmens. Um einen möglichst effizienten Einsatz der IT zu ermöglichen, halten sich viele Unternehmen an bestimmte Best Practice Frameworks. Diese Leitfäden beinhalten unter anderem den Einsatz der IT im Bereich Kundenservice. Dabei übernimmt ein sogenannter Service Desk eine wichtige Funktion. Diese zentrale Anlaufstelle dient als Kommunikationsschnittstelle zwischen dem Kunden und der IT-Organisation. Viele Unternehmen setzen auf eine externe Service Desk-Softwarelösung, die dem Support eine bessere Kommunikation mit dem Kunden und ein effizienteres Abarbeiten von Vorfällen ermöglicht.

Die KMS Computer GmbH bietet ebenfalls eine Lösung eines Service Desk an, die jedoch auf den Bereich Facility Management spezialisiert ist. Das Unternehmen wurde 1990 gegründet und konzentriert sich heute vorrangig auf den Vertrieb von entwickelter Software im Bereich Computer-Aided Facility Management (CAFM). Computer-Aided bedeutet so viel wie „computergestützt“. Den Begriff Facility Management beschreibt der Spezialist für Gebäude- und Immobilienwirtschaft Nävy präzise „als strategische Management-Disziplin, die die Analyse, Dokumentation und Optimierung aller kostenrelevanten Vorgänge rund um Gebäude und ihre Anlagen und Einrichtungen (Facilities) unter besonderer Berücksichtigung von Arbeitsplatz und Umfeld der Nutzer umfaßt.“¹

Seit 2011 entwickelt und vertreibt die KMS Computer GmbH die webbasierte Software GEBman 10. Es handelt sich bei GEBman 10 um eine CAFM-Software für Kommunen, Industrie und Gebäudeverwalter. Übersichten geografischer Informationen oder die Analyse von Sachdaten können individuell auf die Kunden abgestimmt werden. Außerdem ist es ein Werkzeug zur Verwaltung und ein Arbeitsmittel zur Unternehmensführung oder der finanziellen Planung. Die Anwendung kann als Desktop-Installation, als interne Weblösung oder als Cloud-Lösung betrieben werden. Aber auch mobile Lösungen einzelner Module sind bereits in Verwendung und werden stetig weiterentwickelt. Es wird schon jetzt deutlich, dass GEBman 10 mannigfaltig ist und mit über 40 Modulen auch in vielen Branchen zum Einsatz kommt. Gerade in sehr speziellen Bereichen, wie beispielsweise Außenbeleuchtung oder Baumverwaltung, kann es zu den unterschiedlichsten und ungewöhnlichsten Problemen kommen. Eine Grundvoraussetzung für die effiziente Lösung von Problemen, ist das Festhalten der genauen Vorkommnisse. Hierbei kann das Modul Service Desk in GEBman 10 durchaus hilfreich sein und alle erforderlichen Prozesse zur Problemlösung unterstützen.

¹ Nävy (2006), S. VII

1.2 Das Modul Service Desk

Der Service Desk in GEBman 10 ist stark an die anderen Module gebunden und auf den Bereich Facility Management ausgelegt. Im Modul Service Desk ist es möglich, Meldungen für verschiedenste Objektarten aufzugeben. Ist beispielsweise eine Außenbeleuchtung eines Gebäudes ausgefallen, kann ein Benutzer eine Störmeldung bezüglich der Außenbeleuchtung aufgeben. Dabei wählt er das entsprechende Gebäude aus und trägt die genaue Problemstellung in die Meldung ein. So hat ein Techniker zum Beispiel die Möglichkeit, in die Störmeldung Einsicht zu nehmen und die defekte Außenbeleuchtung zu reparieren. Oder aber er fragt nach den genauen Ursachen und antwortet somit auf die Störmeldung. Probleme und Vorfälle können durch den Service Desk genau spezifiziert und archiviert werden. Dadurch ist eine effizientere Lösung des Problems bei einem erneuten Auftreten möglich.

1.3 Motivation der Arbeit

Die KMS Computer GmbH nutzt derzeit eine externe Service Desk-Software, um sämtliche Vorfälle in der Support-Abteilung zu bearbeiten und zu protokollieren. Ziel der Bachelorarbeit ist es, das Service Desk Modul in GEBman10 so anzupassen, dass auf eine externe Service Desk-Lösung verzichtet und das hauseigene Produkt eingesetzt werden kann. Dadurch könnten nicht nur die Kosten für die Lizenzierung der externen Lösung eingespart, sondern ein effektiveres Arbeiten mit der eigenen Service Desk-Lösung ermöglicht werden. Hierfür muss das Modul Service Desk um eine E-Mail Integration erweitert werden. Mit dieser Erweiterung soll es möglich sein, über den E-Mail Verkehr auf Meldungen im Service Desk zu antworten oder neue Meldungen zu erstellen. Für die E-Mail Integration muss ein Konzept erstellt werden, um anschließend eine gute Implementierung zu erreichen.

Des Weiteren sollen andere Service Desk-Lösungen analysiert werden, um Verbesserungsmöglichkeiten für das Service Desk Modul in GEBman10 zu identifizieren. Dadurch soll das Service Desk Modul vom Facility Management Bereich gelöst werden, wodurch dem Support ein komfortableres Arbeiten ermöglicht werden soll.

1.4 Vorgehen bei der Arbeit

Zu Beginn wird im Punkt 2 der Begriff Service Desk in einen Kontext gebracht und allgemeine Anforderungen bestimmt. Durch eine Analyse verschiedener Service Desk - Softwarelösungen im Punkt 3 können Anregungen für mögliche Verbesserungen des Service Desk-Moduls gesammelt werden. Im Anschluss in Punkt 4 wird der gegenwärtige Service Desk in GEBman 10 beleuchtet. Dabei werden die Erkenntnisse aus Punkt 3 dazu beitragen, Verbesserungsmöglichkeiten des Moduls zu benennen. Es werden außerdem die endgültigen Anforderungen an

die E-Mail Integration beschrieben. Der Punkt 5 konzentriert sich auf den Microsoft Exchange Server, der einen wesentlichen Bestandteil der E-Mail Integration bildet. Auch wird an dieser Stelle die derzeitige Verwendung der Exchange Web Services in GEBman 10 näher betrachtet.

Durch eine Analyse der bis dato gewonnenen Erkenntnisse, kann im Punkt 6 eine Konzipierung erfolgen. Hierzu werden verschiedene Modellierungen vorgenommen und auf Sicherheitsaspekte eingegangen. Bei der Umsetzung im Punkt 7 steht die implementierte Erweiterung des Moduls Service Desk im Vordergrund. Wichtige Methoden, Erfahrungen und Fehlschläge während der Umsetzung sollen hier erörtert werden. Abschließend kann dann im Punkt 8 ein Fazit gezogen werden. Ein Ausblick auf die Verbesserungen des Service Desk-Moduls in GEBman 10 wird diese Arbeit abrunden.

2 Service Desk nach ITIL v3

2.1 Begriffsabgrenzung

Für die Klärung des Begriffs „Service Desk“ ist es sinnvoll, sich auf die Information Technology Infrastructure Library - kurz ITIL - zu beziehen. ITIL ist zwar keine Norm, die in der IT-Branche eingehalten werden muss, dennoch bezieht man sich im IT-Service Management vorrangig auf ITIL. Bereits 1989 wurde die Central Computer and Telecommunication Agency (CCTA) von der britischen Regierung beauftragt, Geschäftsprozesse und ihre Abhängigkeiten zu beschreiben und schriftlich festzuhalten.² Ziel war es, Abläufe in der Unternehmenswelt darzustellen und dadurch die IT-Betriebskosten zu reduzieren. Im Laufe der Jahre wurden die ersten Ausarbeitungen überarbeitet und ergänzt. Die ITIL Edition 2011 ist die derzeitig neuste Fassung und stellt ein Update der 2007 veröffentlichten Version ITIL v3 dar.³ Auch bestimmte Normen leiten sich aus dem ITIL-Rahmenkonzept ab. Der internationale Standard ISO/IEC 20000 beispielsweise basiert auf der Version ITIL v2 und definiert die Minimalanforderungen des IT-Service-Managements für Organisationen.⁴ ITIL kann deshalb als "Quasi-Standard" gesehen werden. Es ist ein Best-Practice Leitfaden, der beschreibt "Was zu tun ist, aber nicht wie". Das macht deutlich, dass ITIL durchaus Handlungs -und Interpretationsspielraum zulässt, aber dennoch in einem vorher definierten Rahmen greifbar sein muss. Beschrieben wird ITIL v3 in fünf Büchern, auf die später noch eingegangen wird:

- Service Strategy
- Service Design
- Service Transition
- Service Operation
- Continual Service Improvement

ITIL kann als Framework gesehen werden, mit dem Abläufe im Bereich IT Service beschrieben werden können. Genauer gesagt, spricht man von IT Services Management, kurz ITSM. Hier werden alle Methoden erläutert, die nötig sind, um die bestmögliche Unterstützung von Geschäftsprozessen durch die IT-Organisation zu erreichen.⁵ In ITIL ist der Prozess definiert als ein „Satz von in Wechselbeziehung oder Wechselwirkung stehenden Tätigkeiten (und Mitteln), die Eingaben in Ergebnisse umwandelt.“⁶ Zu den Mitteln können Personal, Einrichtungen und Anlagen, Technologie und Methodologie gehören. Eingaben für einen Prozess sind üblicherweise Ergebnisse anderer Prozesse.

² Vgl. Olbrich (2008), S. 1.

³ Website: Andenmatten (abgerufen am: 10.05.2016)

⁴ Vgl. Buchsein, R. et al. (2007), S. 5ff.

⁵ Vgl. Ebel (2008), S. 27ff.

⁶ Vgl. DIN EN ISO 9000 (2005), S. 23.

Jeder IT Service Management-Prozess hat eine charakteristische Zielrichtung und wird durch Funktionen unterstützt. Eine Funktion besteht aus einer Gruppe von Personen und deren Werkzeugen, die dafür verwendet werden, ein oder mehrere Prozesse oder Aktivitäten zu stützen.⁷ Außerdem bewirkt das Zusammenspiel verschiedener IT Service Management-Prozesse, dass dem Kunden die notwendigen IT Services zur wirkungsvollen Unterstützung seiner Geschäftsprozesse geliefert werden. Der Service Lifecycle in Abbildung 1 veranschaulicht genau diese Kernprozesse und Kernfunktionen, die ein IT-Prozess während seiner gesamten Lebensdauer besitzt. Die einzelnen Teilbereiche decken sich mit den zuvor aufgeführten Büchern von ITIL v3.⁸

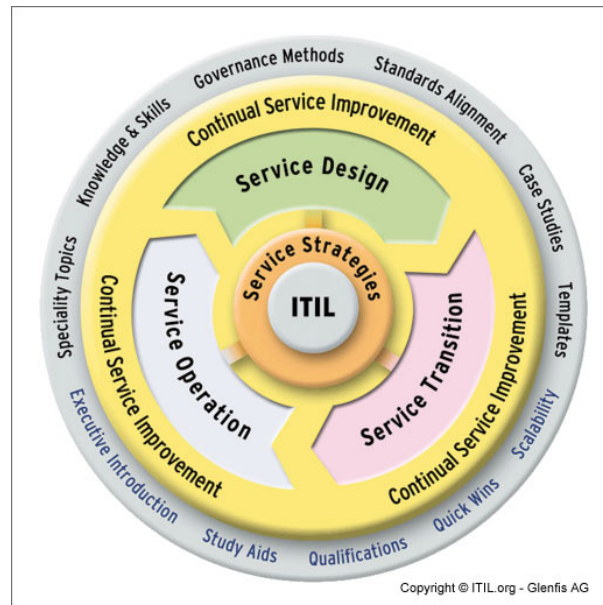


Abbildung 1: ITIL Service Lifecycle, Quelle: http://os.itil.org/osMedia/site/t1media/JPEG/01_ital_imap.jpg

Es ist nicht notwendig, auf alle Teilbereiche einzugehen. Der Service Desk ist nämlich Bestandteil der Service Operation und somit die richtige Anlaufstelle für die Begriffsklärung.

„Service Operation beschreibt den Abschnitt des Lebenszyklus, der von den Kunden primär wahrgenommen wird.“⁹ In der Service Operations-Phase werden die Prozesse und Funktionen beschrieben, die einen stabilen und bestmöglichen IT Service garantieren sollen. Bei dieser Verbindung von IT Organisation und Kunde wird besonders auf den Kunden eingegangen. Der Service Desk ist hierbei „die zentrale Anlaufstelle, der Single Point of Contact (SPoC) zwischen Anwender und der IT-Organisation“.¹⁰ Wie der Name schon verrät, kommt der Anwender nur über diese Schnittstelle in Kontakt mit der IT. Hier werden Meldungen der Anwender üblicherweise erfasst, kategorisiert und eingetragen. Der Service Desk ist nicht nur eine Kommunikationsunterstützung, sondern bietet gleichzeitig eine Auskunft für bereits bekannte Probleme. Dadurch kann bei häufig auftretenden Service-Unterbrechungen schneller gehandelt werden. Auch Supportanfragen, Beschwerden, Verbesserungsvorschläge oder Änderungswünsche können in den Service Desk eingetragen werden. Der Service Desk fungiert dadurch als

⁷ Vgl. Cannon, D. / Wheeldon, D. (2007), S. 233.

⁸ Vgl. Ebel (2008), S. 27ff.

⁹ Ebel (2008), S. 439.

¹⁰ Ebel (2008), S. 439.

sogenannter „Single Point of Contact“. Die nachfolgende Abbildung macht deutlich, dass der Kunde (Anwender) eine einzige Anlaufstelle hat, um seine Anliegen dem Support mitzuteilen.

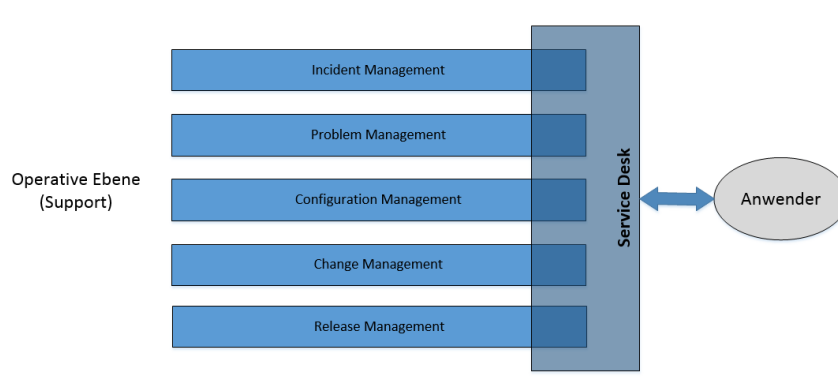


Abbildung 2: Single Point of Contact, Quelle: <http://edoc.hu-berlin.de/conferences/dfn2006/fischlin-roger-105/PDF/fischlin.pdf>

Der Support kann die Meldungen der Anwender nun den entsprechenden Managementbereichen zuordnen. Eine Störmeldung von einem technischen Gerät würde beispielsweise zum Incident Management gehören. Ein Feature-Wunsch ist eher dem Teilbereich Change Management einzuordnen. Der Single Point of Contact erleichtert so die Kommunikation zwischen dem Anwender und dem Support.¹¹

2.2 Unterschied zu Help Desk

Bei der Begriffsabgrenzung zwischen Help Desk und Service Desk ist Vorsicht geboten. In mehreren Quellen ist zu finden, dass Help Desk (auch User-Help-Desk) lediglich ein veralteter Begriff für den Service Desk sei.^{12,13} Im Internet heißt es beispielsweise auf Wikipedia: „Die Artikel Helpdesk und Servicedesk überschneiden sich thematisch.“¹⁴ In anderen Literaturquellen tritt der Begriff Help Desk erst gar nicht auf oder wird dem Service Desk gleichgesetzt.¹⁵ Im Zweifelsfall sollte man sich direkt auf das Buch ITIL Service Operation beziehen. In dem heißt es übersetzt unter dem Stichwort Help Desk: „Eine Anlaufstelle für Anwender, um Incidents zu erfassen. Ein Help Desk ist in der Regel eher technisch orientiert als ein Service Desk und stellt keinen Single Point of Contact für die gesamte Interaktion bereit. Der Begriff 'Help Desk' wird häufig auch als Synonym für Service Desk verwendet.“¹⁶

In den weiteren Ausführungen gilt deshalb der Help Desk als Synonym für den Service Desk.

¹¹ Vgl. Ebel (2008), S. 65 ff.

¹² Vgl. Buchsein, R. et al. (2007), S. 26.

¹³ Vgl. Meier, A. / Myrach, T. (2004), S. 26.

¹⁴ Website: Wikipedia (abgerufen am: 24.05.2016)

¹⁵ Vgl. Olbrich (2008), S. 19.

¹⁶ Cannon, D. / Wheeldon, D. (2007), S. 233. Übersetzung entnommen aus: Ebel, N. (2008) S. 699.

2.3 Aufgaben eines Service Desk

Im Folgenden sollen die wichtigsten Aufgaben eines Service Desk aus Sicht der ITIL v3 betrachtet werden. Hierfür wird zunächst stichpunktartig die Kernaussage festgehalten, um sie anschließend zu erläutern. Dabei beziehen sich die Kernaussagen auf die Ausarbeitung Olbrichs aus „ITIL Kompakt und verständlich“¹⁷ und sind eine leicht abgewandelte Form vom ITIL v3 Band Service Operation.¹⁸

- „Einheitlichen, zentralen Kommunikationsschnittstelle (SPoC) mit konkreten Ansprechpartner“

Der Kunde hat mehrere Möglichkeiten, den Support zu kontaktieren. Schreibt der Kunde eine E-Mail an den Support, könnte diese ausgewertet und im Service Desk eingetragen werden. Ebenso könnte er selbst einen Eintrag über ein Web-Frontend erstellen. Oder aber der Support erstellt einen solchen Eintrag im Service Desk, wenn der Kunde zum Telefon greift. Es ist jedoch Voraussetzung, dass eine einheitliche und zentrale Kommunikationsschnittstelle bereitgestellt wird.

- „Aufnahme, Dokumentation und Auswertung aller Vorfälle“

Wenn alle Vorfälle ordnungsgemäß aufgenommen und dokumentiert wurden, kann schneller reagiert werden, wenn sich ein Problem wiederholt. Dass alle Vorfälle auch ausgewertet werden sollten, ist verständlich und kann eventuell dazu beitragen, Folgeprobleme frühzeitig zu erkennen.

- „Überwachung, Nachverfolgung und Eskalation von laufenden Supportvorgängen. Frühzeitiges Erkennen von Bedürfnissen und Problemsituationen“

Wie im Punkt zuvor erwähnt, können Probleme frühzeitig erkannt werden, indem Vorfälle genauestens ausgewertet werden. Das ist aber längst nicht die einzige Möglichkeit, Bedürfnisse der Kunden zu erkennen. Gute Mittel für vorausschauende Handlungen sind bspw. Monitoring-Systeme oder Log-Files. Sie liefern technische Informationen, die - nach einer Auswertung - Aufschluss über die aktuelle Lage des Kunden geben und in den Service Desk integriert werden könnten. Nach ITIL v3 gibt es zwei Arten von Eskalation. Die funktionale Eskalation, bei der ein besonders schweres Problem oder die vergangene Zeit ein Auslöser wie die Eskalation eines Incidents ist. Im Gegensatz dazu ist bei hierarchische Eskalation eine Weiterleitung des Problems an die nächst höhere Eskalationshierarchie vorgesehen. Sollte beispielsweise ein Problem im Service Desk längere Zeit nicht bearbeitet und somit kein Fortschritt erzielt werden, eskaliert das Problem und wird entsprechend weitergeleitet^{19,20}

¹⁷ Olbrich (2008), S. 19f.

¹⁸ Vgl. Cannon, D. / Wheeldon, D. (2007), S. 110.

¹⁹ Vgl. Steinbaerg, R. A. / Orr, A. T. (2005), S. 80f.

²⁰ Vgl. Victor, F. / Günther, H. (2005), S. 26ff.

- „Überprüfung der Einhaltung des Dienstleistungsgegenstands anhand von Service-Level-Agreements“

Mithilfe des Service Desks kann kontrolliert werden, ob die vereinbarten Leistungen zwischen Auftraggeber und Beauftragten eingehalten wurden, wenn alle Vorfälle dokumentiert wurden.

- „Reporting – Beauskunftung gegenüber den Usern (Kunden) und dem Management. Informationen über den aktuellen Status von Vorgängen, geplanten Änderungen und verschiedenen Nutzungsmöglichkeiten“

Der Service Desk dient außerdem dazu, stets mit dem Kunden im Kontakt zu stehen. So können Informationen an den Kunden weitergeleitet oder auf der anderen Seite aktuelle Vorgänge, Status etc. des Kunden verfolgt und analysiert werden.

- „Überprüfen der Kundenzufriedenheit, Stärkung der Kundenbeziehung. Kontaktpflege. Aufspüren neuer Geschäftschancen“

Nicht zuletzt kann der Service Desk auch als Instrument für einen ständigen Kontakt zum Kunden eingesetzt werden. Der Kunde hat dadurch den Eindruck, permanent mit der Support und somit der Firma verbunden zu sein. Das kann das Verhältnis zum Kunden stärken oder gar neue Geschäftsmöglichkeiten eröffnen.

2.4 Nachteile ohne Service Desk

Die vielseitigen Aufgaben eines Service Desks machen deutlich, welche Nachteile ein nicht vorhandener Service Desk für ein Unternehmen haben kann. Benutzer wissen bei Problemfällen nicht sofort, wer für die Bearbeitung des Vorfalls zuständig ist. Auch ist der Melder beispielsweise für eine Störmeldung nicht sofort identifizierbar. Rückfragen sind daher schwer möglich und die gesamte Bearbeitung des Problems verzögert sich stark.

Bei der Erfassung von Vorfällen sind im Service Desk bestimmte Informationen nötig, die normalerweise über ein Formular o.ä. erfasst werden. Wenn der Service Desk - und damit auch ein Formular zur Erfassung - komplett fehlt, kann es zu unvollständigen Vorfallbeschreibungen kommen. Auch hier wird der Prozess für die Problembehandlung verlangsamt.

Probleme, die häufig auftreten, werden nur teilweise bis gar nicht dokumentiert. Dadurch werden immer wieder neue Lösungsansätze gesucht, ohne auf bereits vorhandene und bewährte Lösungen zurückzugreifen. Eine Wissensdatenbank ist daher unmöglich.

Das Anschaffen einer Service Desk-Lösung sowie die zugehörigen Schulungsausgaben sind natürlich mit Kosten verbunden, doch ein effektives Vorgehen beim Lösen eines Vorfalls mittels des Service Desks kann die Gesamtkosten reduzieren. Demzufolge würde ein gut implementierter Service Desk dazu beitragen, die Kosten insgesamt zu senken.²¹

Unter bestimmten Voraussetzungen ist es möglich, dass ein Kunde direkt mit dem Fachpersonal - beispielsweise aus der Entwicklungsabteilung - kommuniziert, anstatt mit einem Supportmitarbeiter. Das geschieht, wenn keine Filterung der eingetroffenen Probleme vorgenommen wird. Ein Entwickler könnte somit einem fragenden Kunden gegenüberstehen, dessen Problem eher in den Bereich Release Management einzuordnen ist. Ein Supportmitarbeiter wäre in diesem Fall der bessere Gesprächspartner für den Kunden.

Der Kunde hat beim Fehlen eines Service Desk keinen Single Point of Contact mehr und könnte möglicherweise nicht wissen, an wen er sich bei Problemen zu wenden hat. Es fehlt ein wichtiger Kommunikationsweg, der nicht ohne weiteres ersetzt werden kann. Dadurch könnte auch die Beziehung zwischen dem Unternehmen und dem Kunden leiden.

In diesem Punkt wurden die begrifflichen Grundlagen für das weitere Vorgehen gelegt. Es wurde deutlich gemacht, welche Aufgaben ein Service Desk hat und somit auch, wie wichtig er für die Kommunikationsstruktur eines Unternehmens sein kann. Nicht nur der Kontakt zum Kunden kann durch den Service Desk verbessert werden, auch das Protokollieren und Lösen von Problemen innerhalb einer Firma ist mit dem Service Desk besser zu bewältigen.

²¹ Vgl. Olbrich (2008), S. 26f.

3 Analyse verschiedener Service Desk-Lösungen

3.1 Schwerpunkte der Analyse und Ausgewählte Service Desk-Lösungen

Nach diesem sehr theoretischen Ansatz, die Einsatzmöglichkeiten und Aufgaben eines Service Desks zu klären, werden nun Praxisbeispiele analysiert. Wichtig sind dabei nicht Kriterien wie das äußere Erscheinungsbild oder die Kosten. Ziel soll es sein, durch einen Vergleich gängiger Softwarelösungen Verbesserungsmöglichkeiten der eigenen Service Desk-Funktionalitäten in GEBman 10 zu ermitteln. Dabei wird auf die drei folgenden Punkte Wert gelegt:

- **Funktionalität:**

Die Funktionalität ist das fundamentalste Kriterium. Hier ist entscheidend, welche Möglichkeiten dem Benutzer gegeben werden, bspw. Meldungen/Tickets anzulegen, zuzuweisen, zu suchen oder zu filtern. Wichtig ist aber auch, welche Informationen in welcher Darstellungsform enthalten sind (Diagramme etc.) und welche Daten erfasst werden müssen bzw. können.

- **Bedienbarkeit:**

Aus diesem Blickwinkel ist es entscheidend, welche Bedienelemente vorhanden und wie diese angeordnet sind. Eine Bewertung nach intuitiver Bedienbarkeit ist schwierig vorzunehmen, da das immer eine subjektive Ansicht enthält.

- **Anpassbarkeit:**

Inwieweit kann bspw. die grafische Oberfläche vom Benutzer geändert und auf die eigenen Bedürfnisse angepasst werden.

Nach den Recherchen auf mehreren Review und Ranking Websites zum Thema Service Desk / Help Desk, sind drei Softwarelösungen wiederholt erwähnt und gut bis sehr gut bewertet worden.^{22,23,24} Diese drei webbasierten Anwendungen werden nun vorgestellt und anschließend ihre Stärken bzw. Besonderheiten dargelegt. Im Anhang auf Seite 45 befindet sich jeweils ein Screenshot von jeder Softwarelösung.

- **Freshdesk:**

Girish Mathrubootham beschloss 2010 nach dem Lesen eines Nachrichtenartikels, die Firma Freshdesk ins Leben zu rufen. Das gleichnamige Produkt wird laut eigenen Angaben von rund 70.000 Kunden aller Unternehmensgrößen genutzt.²⁵

²² Website: Bilton (abgerufen am: 20.06.2016)

²³ Website: Thornton (abgerufen am: 20.06.2016)

²⁴ Website: Haslam (abgerufen am: 20.06.2016)

²⁵ Website: Freshdesk (abgerufen am: 15.05.2016)

- Desk.com:

Das Unternehmen Salesforce.com legt laut eigenen Angaben großen Wert auf mobile Endgeräte und soziale Netzwerke.²⁶ Die Service Desk -Lösung des Unternehmens nennt sich Desk.com und zu ihren Kunden zählen unter anderem die Firma FlixBus und die Commerzbank.

- Zendesk:

Zendesk beschreibt sich selbst als „Kundenservice-Plattform“. Die gleichnamige Firma hat nach eigenen Angaben mehr als 75 000 Unternehmen, die diese Plattform nutzen. Entstanden ist das Unternehmen 2007 aus einer Idee von drei Freunden aus Kopenhagen.²⁷

Aktuell benutzt der Support von der KMS Computer GmbH die Software SysAid für den Service Desk. Durch eine Vielzahl von Einträgen und Erfahrungen der Mitarbeiter im Support ist es sinnvoll, auch diese Anwendung mit in den Vergleich einfließen zu lassen.

- SysAid:

Die Help Desk-Software SysAid wird nach eigenen Angaben in über 10 000 Unternehmen in 140 Ländern eingesetzt. Die Firma SysAid Technologies wurde 2002 gegründet und zählt somit zu den erfahrenen Unternehmen dieser Branche.²⁸

3.2 Freshdesk

Alle im Freshdesk angemeldeten Support Mitarbeiter werden als Agenten bezeichnet. Der Freshdesk besticht mit seinem sehr simplen Dashboard, was der Bedienbarkeit zugutekommt. Zur Erklärung: ein Dashboard ist üblicherweise eine kompakte, meist grafisch aufbereitete Ansicht von Informationen. Der Agent erhält hier nur die wichtigsten bzw. neuesten Informationen. Auf den ersten Blick kann der Agent sehen, welche Tickets offen, nicht zugewiesen oder überfällig sind. Zusätzlich erhält der Agent die Möglichkeit, individuelle Aufgaben zu notieren und wie eine Checkliste abzuarbeiten. Dies ist ein sehr nützliches Feature für kleinere Notizen bzw. Probleme.

Erst im zweiten Menüpunkt kann der Agent Tickets filtern. Hierfür gibt es gängige auswählbare Filter oder die Option, einen Filter selbst zu konfigurieren und zu speichern. Wählt der Agent hier nun ein oder mehrere Tickets aus, gelangt er in eine Detailansicht eines Tickets und kann dann zwischen den ausgewählten wechseln. In der Detailansicht kann er nun den gesamten

²⁶ Website: Salesforce (abgerufen am: 16.05.2016)

²⁷ Website: Zendesk (abgerufen am: 17.05.2016)

²⁸ Website: SysAid (abgerufen am: 17.05.2016)

Verlauf betrachten, auf das Ticket antworten, das Ticket weiterleiten etc. . Ein Seitenmenü ermöglicht die Einsicht auf die wichtigsten Informationen des Ticketerstellers. Außerdem können die Ticket-Eigenschaften in diesem Seitenmenü geändert und somit die Priorität und der Status festgelegt werden. Zusätzlich gibt es eine Typisierung des Tickets. Im Punkt 2.1 wurden die verschiedenen IT Service Managementbereiche dargelegt. Durch die Typisierung kann festgelegt werden, welchem Managementbereich das Ticket entspricht. Handelt es sich beispielsweise um einen Notfall, wird der Typ als Incident bestimmt und kann so bei der Suche schneller gefunden werden.

Ein weiterer Menüpunkt nennt sich „Soziales“ und hat eine spezielle Funktionalität. Freshdesk bieten nämlich die Möglichkeit, die Kunden über einen „sozialen Support“ zu betreuen. Laut eigenen Angaben begründet Freshdesk den Kontakt mit Kunden über soziale Kanäle wie folgt: „Tatsächlich erwarten 32 % der Kunden in sozialen Netzwerken eine Antwort auf Ihre Anfragen innerhalb von 30 Minuten.“ Weiterhin heißt es, bei schnellen und effektiven Antworten würden 71% der Kunden den Support weiterempfehlen. Wie Freshdesk zu diesen Zahlen kommt, bleibt zunächst unklar. Recherchiert man ein wenig im Netz, so kann man diese Zahlen in einem Bericht von t3n wiederfinden, der auf einer Studie von Bain & Company beruht.²⁹ Wichtiger als die Zahlen ist jedoch die Idee, über soziale Netzwerke mit dem Kunden zu interagieren und somit eine teils freundschaftliche Beziehung aufzubauen. Erlaubt man Freshdesk im Administrator-Bereich, sich über ein Twitter-Konto einzuloggen, so kann direkt im Service Desk auf Twitter zugegriffen werden. Auch eine Facebook-Seite kann in den Service Desk von Freshdesk integriert werden. Diese Funktionalität wirkt sehr modern und bietet neue Ansatzpunkte in der Kundenbetreuung.

Im dritten Menüpunkt erhält der Agent die Option, eine Knowledge Base anzulegen. Diese Wissensdatenbank kann direkt vom Kunden aufgerufen werden, um eine erste Hilfestellung bei bekannten Problemen zu erhalten. Mit Kategorien wie Frequently Asked Questions (FAQs) kann die Wissensdatenbank unterteilt werden und bietet dem Kunden so eine gute Übersicht. Auch ein Forum kann mit dem Freshdesk gepflegt werden. Im Unterpunkt „Foren“ können mehrere Foren verwaltet werden und mit Themen wie „Tips und Tricks“ oder „Wie erstelle ich ein Ticket?“ dem Kunden ein optimales Handbuch oder Nachschlagewerk liefern.

Des Weiteren kann im Menüpunkt „Berichte“ auf umfassende Analysen Einsicht genommen werden. Nicht nur wie lange ein Ticket durchschnittlich bearbeitet wurde ist in Diagrammen dargestellt, sondern auch wie viele Tickets die Kunden aufgegeben haben oder wie viele Tickets ein Agent schon bearbeitet hat. Trotz der großen Informationswiedergabe bleibt Freshdesk übersichtlich und gut strukturiert. Die Berichte kann sich der Agent auch per Mail in Form einer PDF - oder CSV-Datei zuschicken lassen.

Der bereits erwähnte Administrator-Bereich kann gut genutzt werden, um den Service Desk anzupassen. Von allgemeinen Einstellungen wie den Feldern, die bei der Ticketerstellung ausgefüllt werden müssen, bis zu dem Import von Daten aus anderen Service Desk-Lösungen kann der Freshdesk gut auf die Bedürfnisse der Agents oder Gruppen von Agents angepasst werden.

²⁹ Website: Rixecker (abgerufen am: 19.05.2016)

Egal in welchem Menüpunkt der Agent sich bewegt, es steht ihm immer ein Button für das Anlegen eines neuen Tickets und einer Suche in der oberhalb liegenden Menüleiste zur Verfügung. Der Freshdesk ist somit ein in sich schlüssiges System mit vielen modernen und anschaulichen Extras.

3.3 Desk.com

Desk.com ist eine Service Desk-Lösung, die besonderen Wert auf den mobilen Einsatz des Supports legt. Das spiegelt sich auch in der Desktop Webanwendung wieder. Die Oberfläche erinnert stark an eine App auf einem mobilen Endgerät. Die Benutzer werden in Desk.com ebenfalls Agents genannt. Eine statische Menüleiste ist auch hier oberhalb der Ansicht zu finden. Hier befinden sich große Buttons für das Anlegen eines neuen Tickets, eine Suche, ein Button für weitere Menüs und eine Art Tab-Ansicht der neusten Tickets. Das Dashboard besteht aus einer Auflistung aller Tickets, die nur die nötigsten Informationen liefern. Spalten können sich aber noch zusätzlich einblenden lassen. Filtern lassen sich die Tickets an dieser Stelle mit einer Auswahl auf der linken Seite. Jedoch ist diese Auswahl sehr eingeschränkt auf alle Tickets oder Tickets, die dem Agent zugewiesen wurden. Eine Hoverbox ermöglicht die Einsicht der Beschreibung des Tickets, ohne dieses öffnen zu müssen. Hierfür muss der Agent lediglich den Mauszeiger über das Ticket halten und nach wenigen Sekunden erscheint ein kleines Fenster. Die Service Desk-Lösung von Desk.com hat ebenfalls einen eigenen Menüpunkt für die Einsicht von Berichten - nicht ganz so umfangreich wie beim Freshdesk, aber dennoch anschaulich in Diagrammen dargestellt. Die Funktionalität einer Wissensdatenbank bietet der Desk.com ebenso. Allerdings kann diese nur im Administrator-Bereich verwaltet und erweitert werden. Insgesamt wirkt der Desk.com maßgeschneidert für mobile Endgeräte, und genau hier liegen auch die Stärken der Service Desk-Lösung. Durch die großen Bedienelemente und stark vereinfachten Ansichten wird das Arbeiten auf Tablet o.ä. deutlich erleichtert. Für eine Desktop Variante ist dies nicht die beste Wahl, auch weil sich der Service Desk - wenn überhaupt - nur sehr umständlich auf die individuellen Bedürfnisse anpassen lässt. Für Mitarbeiter, die ständig unterwegs sind, ist das System durchaus attraktiv.

3.4 Zendesk

Der Zendesk hat seine Stärken in den umfassenden Hilfestellungen für die Agents. Schon im Dashboard erhält ein Agent eine knappe aber präzise Erklärung der einzelnen Teilbereiche des Service Desk. Zunächst muss der Agent die einzelnen Kanäle wie E-Mail oder Telefon einrichten. Auch im Zendesk sind Twitter und Facebook als Kommunikationswege denkbar. In fast allen Einrichtungsschritten wird der Agent mit Anweisungen unterstützt und kann sich direkt

im Zendesk ein Video-Tutorial anschauen.

Eine statische Leiste befindet sich im oberen Teilbereich der Webanwendung mit einer Suchfunktion und der Möglichkeit neue Tickets zu erstellen. Ein ebenfalls festes Menü am Seitenrand hat nur die wichtigsten Unterpunkte: Dashboard, Tickets, Berichte und Einstellungen. Eine Anordnung, die bereits aus den anderen Service Desk-Lösungen bekannt ist. Das Menü kann angepasst werden, wenn alle Kanäle eingerichtet sind.

Neben den bereits bekannten Features von den anderen Softwarelösungen bietet Zendesk weitere Funktionalitäten. Durch eine große Auswahl an Apps kann der Service Desk ganz nach eigenen Vorstellungen des Agents angepasst werden. Ein einfaches Beispiel hierfür ist das Anzeigen von Kontaktinformationen direkt neben einem Ticket. Die App trägt den schlichten Namen „Benutzerdaten“. Wurde diese Erweiterung erfolgreich installiert, kann der Agent über einen Button die Funktionalitäten der Apps bei der Ticketübersicht nutzen, um weitere Informationen über den Kunden zu gewinnen. Dabei besteht die Möglichkeit, Notizen oder Details über den Kunden in der App einzutragen.

Über ein Web Widget ist es möglich, auf Komponenten von Zendesk wie die Wissensdatenbank oder Live-Chat zuzugreifen. Als Widget bezeichnet man einfache kleine clientseitige Programme, die durch minimalen Eingabeaufwand zusätzliche Funktionen oder Informationen bereitstellen.^{30,31} Dieses Web Widget kann in Webseiten eingebettet werden, indem in den Einstellungen das Widget aktiviert und auf der Webseite der Source-Code eingebunden wird. Eine Funktionalität, die vor allem bei Unternehmenswebseiten eingebaut werden könnte.

Um festzustellen, wie sehr die Kunden mit dem Service Desk zufrieden sind, kann mit dem Zendesk eine Kundenumfrage gestartet werden. Die Fragen hierfür lassen sich allerdings nicht konfigurieren. Der Kunde kann (bei aktivierter Option) auch die Tickets bewerten und somit ebenfalls Feedback für den Agent geben.

Zendesk bietet eine Vielzahl von Funktionalitäten, die sowohl neue Kommunikationsmöglichkeiten mit dem Kunden garantieren, als auch die Bedienung des Service Desk für die Agents erleichtern. Es ist aber anzumerken, dass es durchaus eine gewisse Zeit in Anspruch nimmt, alle Features einzurichten und richtig zu bedienen. Einsteiger Agents sollten deshalb gut geschult werden.

3.5 SysAid

Da diese Arbeit in einem knappen Zeitrahmen fertiggestellt werden muss, konnten keine ausführlichen Betrachtungen der Service Desk-Lösungen durchgeführt werden, wenn diese mit vielen Daten gefüllt sind. Hierzu wären mehrere Kontaktinformationen und Ticket-Erstellungen nötig. Deshalb wird nun der firmeneigene Support nach Besonderheiten in der Softwarelösung SysAid befragt. Bei täglichem Gebrauch kommen unvorhersehbare Situationen zustande, die

³⁰ Website: Mattscheck (abgerufen am: 22.05.2016)

³¹ Website: Recommendation (abgerufen am: 22.05.2016)

kaum in der Vorbetrachtung zu erahnen sind. Daher ist die Betrachtung vom SysAid der KMS Computer GmbH eine sinnvolle Vorgehensweise bei der Ermittlung von Verbesserungsmöglichkeiten im Bereich Service Desk-Anwendungen.

Die Startseite von SysAid wirkt nicht besonders aussagekräftig. Dem Benutzer werden mehrere Fenster angezeigt, die nur sehr wenige und auch nur allgemeine Information über aktuelle Tickets enthalten. Um zu der Übersicht der Tickets zu gelangen, muss erst in den Menüpunkt Service Desk gewechselt werden. Hier sind nun alle Tickets mit einer ID gelistet und wirken sehr strukturiert.

SysAid verfügt auch über eine Wissensdatenbank, die vom Support angepasst werden kann. Die Anzeigefenster sind allerdings sehr klein gehalten und lassen sich in ihrer Größe nicht anpassen. Für einen Kunden wirkt die Wissensdatenbank deshalb nicht sehr ansprechend.

Als eine Stärke von SysAid ist die Erinnerung an eskalierte Tickets zu nennen. Wenn ein Ticket über ein einstellbares Intervall nicht bearbeitet wurde oder schon seit längerer Zeit im Status „in Bearbeitung“ verharrt, eskaliert das Ticket.

Durch diese Funktion wird ein Support-Mitarbeiter daran erinnert, wenn sich bei einem Ticket keine Veränderung zeigt. Hier sollte dann mit Nachdruck auf die Lösung hingearbeitet werden. Eine weitere Stärke von SysAid ist die Möglichkeit, das Problem eines Tickets direkt in die Wissensdatenbank aufzunehmen. Dadurch wird das Erstellen eines Eintrags erleichtert.

Das System erkennt gleiche Absender nicht und alle Tickets kommen daher in ein einziges Sammelbecken. Das macht eine Zuordnung schwierig, ist aber mit einer Eintragung übergeordneter ID's manuell möglich.

Beim Erstellen eines Tickets ist es wie bei Zendesk und Desk.com nicht möglich, Bilder direkt in den Text einzufügen. Durch das Anhängen von Screenshots o.ä. geht nach dem Support der direkte Bezug zwischen Beschreibung des Problems und der Verdeutlichung mit Hilfe eines Bilds verloren.

Der Support hat angegeben, dass die Wissensdatenbank von Kunden kaum genutzt wird. Das könnte daran liegen, dass sich die Kunden in SysAid nicht einloggen. Es werden auch wenige Tickets direkt über SysAid erstellt, sondern über die E-Mail Integration.

Der Administrationsbereich von SysAid bietet die Möglichkeit, die Benutzeroberfläche umfassend anzupassen. Dadurch können nicht gebrauchte Informationen ausgeblendet werden und die Benutzeroberfläche für den Support-Mitarbeiter auf das Nötigste reduzieren. Des Weiteren können die Betriebszeiten des Help Desks eingestellt werden, was sinnvoll für die Eskalation von Meldungen ist.

Es fällt auf, dass die Alarmer in SysAid nur begrenzt konfigurierbar sind. Es sind lediglich vier Alarmstufen bzw. Status, die man damit abdecken kann, was von den Mitarbeitern des Supports bemängelt wird. Auch eine grafische Auswertung fehlt in der Version, die der Support von der KMS Computer GmbH zur Verfügung hat. Es gibt aber die Option, Reports in Form von Excel- oder PDF-Dateien erstellen zu lassen.

3.6 Fazit der Analyse

Die Service Desk-Lösungen werden nun nach den in Punkt 3.1 genannten Kriterien bewertet. Die Tabelle 1 stellt das Ergebnis dar. Jedes der Kriterien kann mit maximal 10 Punkten bewertet werden. Die Funktionalität hat die höchste Gewichtung, gefolgt von der Bedienbarkeit und der Anpassbarkeit. Die Gewichtungen stehen hinter den Kriterien in Klammern. Die Gesamtpunktzahl findet sich in der letzten Tabellenzeile wieder.

| | Freshdesk | Desk.com | Zendesk | SysAid |
|--------------------|-----------|----------|---------|--------|
| Funktionalität (5) | 9 | 6 | 8,5 | 8 |
| Bedienbarkeit (3) | 9 | 8 | 9 | 6,5 |
| Anpassbarkeit (2) | 7,5 | 5 | 7,5 | 6 |
| Gesamtpunktzahl | 87 | 64 | 84,5 | 71,5 |

Tabelle 1: Bewertung der Service Desk-Lösungen, Quelle: eigene Darstellung

Die Lösung von Freshdesk und Zendesk überzeugten am meisten. Aufgrund des großen Funktionalitätsumfangs und guter Bedienbarkeit erhielten sie fast die höchste Punktzahl in diesen Bereichen. Nur bei der Anpassbarkeit - gerade in der Gestaltung der Bedienoberfläche - mussten kleine Abstriche gemacht werden.

Desk.com stach weder bei den Funktionalitäten, noch in der Anpassbarkeit heraus und konnte sich deshalb auch in der Punktetabelle nicht durchsetzen. Lediglich bei der schlicht gehaltenen Bedienung gab es Punkte.

SysAid befindet sich im Mittelfeld der Punktetabelle, da die Funktionalitäten gut sind, die Softwarelösung aber in der Bedienbarkeit und Anpassbarkeit nicht überzeugen konnte.

Alle Service Desk-Systeme, die betrachtet wurden, waren auf ihre Weise individuell. Die Schwerpunkte waren unterschiedlich gelegt und einen „Sieger“ der Analyse zu bestimmen wäre daher nicht sinnvoll. Wichtiger ist vielmehr, wie die Stärken der Systeme möglicherweise Anreize für Verbesserungen der Service Desk-Lösungen in GEBman 10 bieten.

Der Freshdesk hat seine Stärken klar in der grafischen Aufbereitung der erfassten Daten und in der Einbindung von sozialen Medien. Auch wirkte diese Service Desk-Lösung sehr übersichtlich in allen Teilbereichen. Das lag nicht zuletzt an der Beschränkung auf die wichtigsten Informationen im jeweiligen Menüpunkt. Bilder konnten direkt in der Ticket-Beschreibung eingefügt werden, um somit einen direkten Bezug zu der Vorfallbeschreibung herstellen zu können. Desk.com konnte besonders im mobilen Bereich überzeugen. Die Bedienelemente wurden eher schlicht gehalten und das System bietet Funktionalitäten im richtigen Maße für mobile Einsätze. Das könnte für die KMS Computer GmbH interessant werden, da der mobile Ausbau von GEBman10 in vollem Gange ist und ein Service Desk Modul noch nicht umgesetzt wurde.

Als Stärke des Zendesk sind die guten und ausführlichen Hilfestellungen für die Benutzer (Agents) zu nennen. Besonders für Einsteiger dürfte das eine große Erleichterung in der Einarbeitung sein. Zudem kann der Zendesk sehr einfach erweitert und somit auf individuelle Einsatzbereiche angepasst werden.

Bei der Service Desk-Lösung SysAid wurde die Eskalation von Tickets und die damit einhergehende Erinnerung an den Support-Mitarbeiter als vorteilhaft im Arbeitsalltag benannt. Sollte doch einmal eine Meldung in Vergessenheit geraten, kann dem durch diese Sicherheitsmaßnahme entgegengewirkt werden.

Es lassen sich aber auch Gemeinsamkeiten benennen, die sich in allen Systemen in leicht abgewandelter Form wiederfinden und die somit einen besonderen Stellenwert besitzen:

- Ein Menüleiste ermöglichte es dem Agent/Benutzer jederzeit zu suchen und ein neues Ticket anzulegen.
- Durch farbliche Kennzeichnung waren die Status der Tickets sofort einsehbar.
- Die Kontaktdaten vom Kunden befanden sich in der Detailansicht eines Tickets.
- Filtereinstellungen für Tickets waren konfigurierbar und speicherbar.
- Alle Berichte/grafischen Auswertungen befanden sich in einem Extramenüpunkt.
- Es bestand die Möglichkeit eine Wissensdatenbank zu erstellen und zu verwalten.
- Eine E-Mail Integration bot eine Kommunikationsschnittstelle, die über neu angelegte Tickets per E-Mail informierte und das Erstellen neuer Meldungen ermöglichte.

Besonders dem letzten Punkt soll hohe Beachtung geschenkt werden. In GEBman 10 ist die E-Mail Integration nur teilweise vorhanden und wird im nächsten Punkt genauer erläutert. Nach der Betrachtung der aktuellen Umsetzung des Service Desk Moduls in GEBman 10 kann dieses Fazit der Analyse der verschiedenen Vergleichssysteme dazu genutzt werden, Verbesserungsmöglichkeiten im Service Desk Modul zu bestimmen. Gleichzeitig dient es als Anregung für die Konzipierung der Erweiterung der E-Mail Integration.

4 Der Service Desk in GEBman 10

4.1 Aktuelle Umsetzung

Zunächst ist anzumerken, dass sich das Service Desk Modul stark an den anderen Modulen in GEBman 10 orientiert und an sie gebunden ist. Das unterscheidet sich deshalb sehr stark von den zuvor betrachteten Service Desk-Lösungen und wurde bisher nur im Bereich Facility Management eingesetzt. Aus diesem Grund ist es nur bis zu einem gewissen Grad möglich, diese Lösung in GEBman 10 mit anderen Service Desk-Softwarelösungen zu vergleichen. Ihre Aufgabenfelder und Schwerpunkte schlicht zu unterschiedlich sind. Dennoch ist es möglich, Verbesserungsmöglichkeiten zu identifiziert, welche das Service Desk Modul auch außerhalb des Facility Management Bereiches einsetzbar machen können. Hierzu wird zunächst die aktuelle Umsetzung von dem Service Desk in GEBman10 festgehalten.

Der Service Desk ist ein eigenständiges Modul, welches standardmäßig in jeder Version von GEBman 10 enthalten ist. Es dient in erster Linie dazu, unvorhergesehene Störungen für Geräte, Gebäude, Inventare, Fahrzeuge, Bäume, Grünflächen und Beleuchtungseinheiten erfassen zu können. Aus diesem Grund ist das Modul stark an andere Module gebunden und legt die Schwerpunkte auf den Bereich Facility Management. Das Modul kann aber auch für eine Service Desk-Lösung für das Unternehmen verwendet werden. Hierzu müssen lediglich alle Mitarbeiter einen Account anlegen, um so als Melder agieren oder als Bearbeiter reagieren zu können.

Die Bedienoberfläche ist in einzelne Sektionen unterteilt. Im Dashboard dieses Moduls werden alle Meldungen in einer Sektion angezeigt, die erstellt wurden. Alle Meldungen haben eine eindeutige Nummer (Identifikator). Rechts daneben wird der Standort des Objektes angezeigt, für das die Meldung aufgegeben wurde. Es werden auch bereits abgeschlossene bzw. erledigte Meldungen angezeigt. Der Benutzer erhält mittels eines Berichtes in Diagrammform in einer weiteren Sektion direkt einen Einblick auf die Meldungen innerhalb einer Woche. Unterteilt wird hierbei in eingegangene Meldungen, Meldungen, die in Bearbeitung sind, unbearbeitete Meldungen und fertige Meldungen. Ein weiterer Bericht veranschaulicht die Verweildauer einer Meldung, in dem die Zeit zwischen dem unbearbeiteten Zustand und dem der Bearbeitung protokolliert wird. Unter den beiden Diagrammen finden sich alle Fakten noch einmal in Form von Zahlen wieder. Es ist derzeit noch nicht Möglich, diese Berichte/Diagramme individuell anzupassen. Der Aufbau wird in der Abbildung 3 deutlicher.

Der Benutzer hat im Dashboard die Möglichkeit für eine direkte Detailansicht einer oder mehrerer Meldungen, eine neue Meldung anzulegen, eine Filterung der Meldung vorzunehmen oder die Einstellungen anzupassen. Bei den Einstellungen kann entschieden werden, an welchen Positionen die Sektionen der Bedienoberfläche verankert werden sollen.

Die Speicherung der Filter ist in der Dashboard-Ansicht des Service Desk Moduls nicht mög-

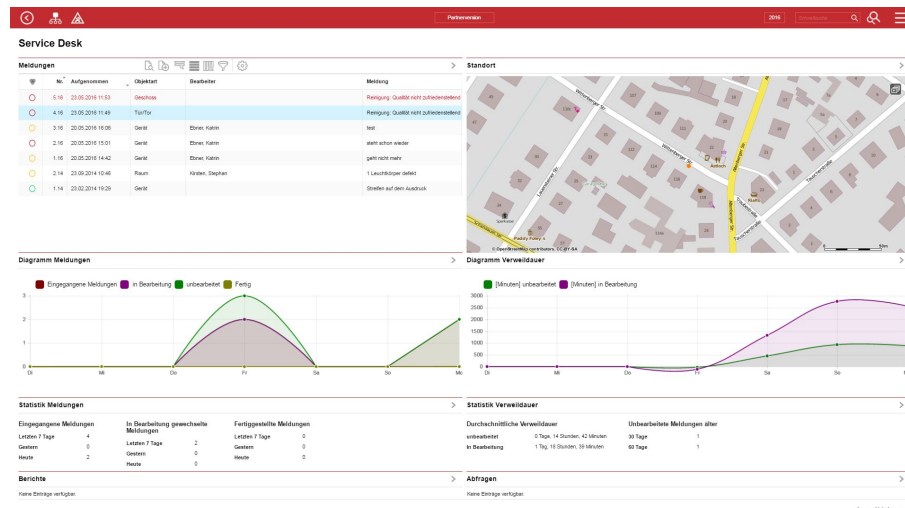


Abbildung 3: GEBman 10 Service Desk Dashboard, Quelle: GEBman 10

lich. Hierfür muss der Benutzer in der Suchliste eine Abfrage definieren und diese abspeichern. Außerdem ermöglicht die Suchliste eine detailliertere Filterung, einen Excel Export, das Löschen von Meldungen und die Generierung von Maßnahmen. Das Generieren von Maßnahmen ist eine zentrale Funktionalität im Service Desk. Manche Vorfälle lassen sich nicht ohne Fachpersonal bewältigen. Hierfür müssen unter Umständen spezielle Mitarbeiter oder eine andere Firma angefordert werden. Mit der Maßnahmengenerierung kann ein Auftrag für den Mitarbeiter bzw. die Firma erstellt werden. Wenn sich eine andere Firma um die Maßnahme für den Vorfall kümmert, wird das am Objekt in der Sektion Fremdvergabe angezeigt.

Bevor man den Service Desk nutzt, ist es sinnvoll, die drei Standardkataloge für den Service Desk zu bearbeiten. Mit Katalogen können in GEBman 10 vorgefertigte Auswahlmöglichkeiten erstellt werden, die später im entsprechenden Modul zur Anwendung kommen. Dem Service Desk stehen folgende Kataloge zur Verfügung:

- **Art:**
Mit der Art ist in diesem Fall die Meldungsart gemeint. Dem Benutzer steht frei, welche Meldungsarten er hier definiert, um auch die ausgefallensten Vorfälle möglichst präzise beschreiben zu können.
- **Meldungsvorlage:**
Nicht selten ähneln sich Meldungen, da sich Vorfälle wiederholen können. Ein Beispiel hierfür wäre ein Kraftfahrzeug in der Fuhrparkverwaltung, welches neue Reifen benötigt. Um nun nicht jedes Mal die gleiche Meldung ausformulieren zu müssen, gibt es die Möglichkeit eine Meldungsvorlage zu erstellen.

- Schnellantwort:

Es können ebenfalls Schnellantworten vordefiniert werden, die eine zügige Antwort auf eine Meldung ermöglichen.

Wenn ein Mitarbeiter beim Erstellen einer Meldung informiert werden soll, so muss zunächst im Modul Verwaltung der entsprechende Exchange Server dafür konfiguriert werden. Hierfür muss der Benutzer unter Einstellungen und dann die Rubrik Groupware wählen. Die Einrichtung des Exchange Servers gestaltet sich wie üblich mit den Kontodaten und der verwendeten Exchange Server Version.

Interessanter gestaltet sich die Konfiguration der Events. Hier können Events erstellt werden, bei denen eine Mail von an dem zuvor erstellten Server an eine beliebige Person geschickt werden kann. Ein Beispielszenario für das Modul Service Desk:

Bei dem Service Desk bietet es sich an, die Meldungen für einen möglichen Eventauslöser zu bestimmen. Es kann entschieden werden, ob beim Erstellen, Bearbeiten, Löschen oder bei einer Wertänderung dieser Meldung eine Mail verschickt werden soll. Dafür müssen noch ein Betreff und eine Beschreibung der Mail eingetragen werden. GEBman 10 besitzt das Feature, das es ermöglicht, Parameter in der Mail mit zu übergeben. Somit kann beispielsweise mit der Mail in Erfahrung gebracht werden, zu welchem Gebäude die Meldung gehört. Dadurch hat der Empfänger der Mail einen direkten Einblick auf die Daten, ohne direkt in den Service Desk schauen zu müssen. Für das Verständnis der Status im Service Desk ist die nachfolgende Tabelle hilfreich.







| Status | Kurzform | Bedeutung |
|---|------------------|---|
|  Rot + roter Text | Offen | Meldung wurde noch nicht gelesen. |
|  Rot | Offen | Die Meldung wurde aufgegeben und gelesen, jedoch noch nicht bearbeitet. |
|  Gelb | In Arbeit | Die Meldung befindet sich in Bearbeitung, ist aber noch nicht fertiggestellt. |
|  Grün | Technisch fertig | Die Meldung ist erledigt. |
|  Haken | Fertig | Die Meldung ist erledigt und abgeschlossen. |
|  Blau | Wiedereröffnet | Die Meldung wurde wiedereröffnet. |

Tabelle 2: Übersicht der Status im Service Desk, Quelle: in Anlehnung an Handbuch GEBman 10

4.2 Verbesserungsmöglichkeiten des Service Desk Moduls

Das Service Desk Modul wurde kurz vorgestellt und mit Hilfe der Analyse aus Punkt 3 lassen sich einige Verbesserungsmöglichkeiten identifizieren. Dabei ist nochmals klar anzumerken, dass eine vollständige Entkopplung des Service Desk Moduls vom restlichen System und damit auch dem Facility Management Bereich nicht möglich ist.

Zunächst soll auf die Bedienoberfläche eingegangen werden. Als erstes fällt auf, dass sich direkt auf der Startseite des Service Desk Moduls mehrere Diagramme und Statistiken befinden. In allen Lösungen, die zum Vergleich betrachtet wurden, sind diese Auswertungen in Form von Berichten in einem extra Menüpunkt. Deshalb könnte über eine Auslagerung der Diagramme und Statistiken nachgedacht werden.

In allen Service Desk-Lösungen hat der Benutzer/Agent jederzeit die Möglichkeit, eine neue Meldung aufzugeben, egal in welchem Menüpunkt er sich befindet. In einigen Fällen ist ein Button dafür sogar farblich hervorgehoben. In GEBman 10 gibt es einen solchen Button auch, dieser hebt sich aber nicht von den anderen Buttons (wie einer Detailansicht) ab. Zudem ist nicht auf jeder Bedienebene eine Möglichkeit für das Anlegen einer neuen Meldung gegeben. Befindet sich der Benutzer im Service Desk Modul beispielsweise in der Detailansicht einer Meldung, muss er erst in eine höhere Bedienebene zurückkehren, um eine Meldung anlegen zu können. In GEBman 10 gibt es in jedem Modul eine kleine Übersichtsleiste, die auch eine Suchfunktion beinhaltet. Es ist also jederzeit möglich zu suchen. Es könnte darüber nachgedacht werden, im Service Desk Modul einen Button o.ä. auch in diese Übersichtsleiste zu integrieren. Somit kann in jeder Bedienebene eine neue Meldung aufgegeben werden. Außerdem könnte über eine farbliche Vorhebung oder zumindest über eine Vergrößerung des Buttons nachgedacht werden, der das Anlegen einer neuen Meldung ermöglicht.

In der Suche des Service Desk Moduls befinden sich 20 Felder, nach denen gesucht werden kann. Darunter kann nach Objekten oder Standorten gefiltert werden. Um die enge Bindung an den Facility Management Bereich zu lockern, wäre es sinnvoll, diese Felder zu unterteilen. Damit ist gemeint, dass eine Sektion speziell für die Felder des Facility Management Bereiches gebildet wird und eine Sektion für allgemeine Felder. Somit könnten die Suchfelder wie Standort oder Objekte ausgeblendet werden und Felder wie Status, Art oder Meldung könnten überschaubarer wirken. Durch diese Veränderung würde sich möglicherweise die Übersicht stark verbessern.

Die Funktionalitäten des Service Desk Moduls bietet ebenfalls Verbesserungsmöglichkeiten. Die Betrachtung der Lösungen Freshdesk und Zendesk hat ergeben, dass zunehmend auf die Integration von sozialen Netzwerken wie Twitter oder Facebook gesetzt wird. Dieser interessante Ansatz, eine bessere Kommunikation mit dem Kunden zu ermöglichen, bedarf jedoch eines hohen Aufwandes in der Umsetzung. Eine gute Planung müsste hierfür als Grundlage dienen. Im Rahmen dieser Bachelorarbeit kann aus zeitlichen Gründen nicht weiter darauf eingegangen werden.

Im Freshdesk ist es möglich, Bilder direkt in die Beschreibung der Meldung einzufügen. Dadurch kann ein besserer Bezug zwischen Text und Bild hergestellt werden und somit das Problem auf den ersten Blick verdeutlichen. Dieses Feature wurde über einen Button realisiert, der das Einfügen von Bildern erlaubt. Es wäre also möglich, auch eine Prüfung vorzunehmen, ob es sich wirklich um ein Bild handelt oder womöglich versucht wird, andere Dateitypen oder Schadcode einzufügen.

Aufgefallen ist außerdem, dass im Service Desk Modul keine Wissensdatenbank vorhanden ist. Durch das Einführen einer Wissensdatenbank könnten häufig auftauchende Meldungen archiviert werden und somit zu einer schnelleren Problemlösung beitragen. Es wäre zu überlegen, eine solche Wissensdatenbank auch für andere Module bereitzustellen.

Neben diesen bereits genannten Verbesserungsmöglichkeiten steht in erster Linie die Erweiterung der E-Mail Integration des Service Desk Moduls im Vordergrund. Alle betrachteten Service Desk-Lösungen hatten eine E-Mail Kommunikationsschnittstelle, mit der Tickets/Meldungen im Service Desk angelegt werden konnten. Diese Funktionalität soll auch in GEBman 10 implementiert werden. Dafür müssen zunächst die Anforderungen der Erweiterung bestimmt werden.

4.3 Anforderungen der Erweiterung

Qualität ist ein Maß für das Erfüllen von Anforderungen. Die Qualität des Service Desk kann deshalb nur gesichert werden, wenn die Anforderungen möglichst genau definiert werden. Neben den zu erfüllenden Anforderungen sollte aber noch festgehalten werden, welche Anforderungen nicht erfüllt werden sollen. Letzteres wird häufig nicht beachtet, ist jedoch ein wesentlicher Schritt für das Sicherstellen der Anforderungen.

Ziel der Erweiterung des Service Desks ist es, über den E-Mail Kommunikationsweg Meldung zu erstellen, oder auf eingegangene Meldungen zu antworten. Um E-Mail Meldungen eindeutig zuordnen zu können, muss in der Betreffzeile der E-Mail ein Identifikator (kurz ID) enthalten sein. Diese ID muss ausgewertet und mit den ID's der Meldungen verglichen werden. Sollte es keine Meldung mit der ID geben, die in der Betreffzeile der E-Mail steht, wird eine neue Meldung im Service Desk Modul angelegt und erhält vom System eine eindeutige ID. Bei einer erfolgreichen Erstellung einer Meldung soll der Benutzer eine Bestätigungs-E-Mail erhalten, die dann gleichzeitig die eindeutige ID im Betreff beinhaltet. Sollte die Erstellung einer Meldung oder einer Antwort scheitern, muss der Benutzer informiert werden. Hierzu muss eine Ausnahmebehandlung im Programmcode und die damit verbundene Benachrichtigung an den Benutzer geplant werden. Außerdem müssen die Anhänge in GEBman 10 ebenfalls abgespeichert werden, wenn die E-Mail einen solchen besitzt.

Es soll nicht möglich sein, über eine E-Mail Maßnahmen einer Meldung zu generieren, bestehende Meldungen zu löschen oder zu bearbeiten. Auch eine Veränderung des Status einer Meldung ist über die E-Mail Integration nicht vorgesehen.

5 Microsoft Exchange Server

5.1 Grundlagen

Der Microsoft Exchange Server ist eine serverseitige Anwendung, die den Nachrichtenaustausch und die Zusammenarbeit im Unternehmen erleichtern soll.³² Im Juni 1996 wurde die erste Version von Microsoft Exchange veröffentlicht. Sie löste das Mailsystem MS Mail ab, da dieses für einen Gebrauch mit über 500 Postfächern nicht ausgelegt und somit für größere Unternehmen nicht mehr sinnvoll war. Dieser Wechsel der Software wurde passend durch den Namen Exchange beschrieben, da es so viel heißt wie „Austausch“. Ziel des Microsoft Exchange Servers ist es, Nachrichten zu verarbeiten und zu verwalten. Obwohl Exchange ein Mailserver ist, können neben E-Mails auch Termine angelegt oder Aufgaben vergeben werden. Somit wäre es denkbar, Exchange als zentrale Anlaufstelle der Unternehmenskommunikation einzusetzen.

Microsoft Exchange ist klar in den Bereich Groupware einzuordnen. Groupware-Software wird vor allem zur Unterstützung der internen als auch der externen Unternehmenskommunikation genutzt. Ellis, Gibbs und Rein beschreiben die bekannteste Form von Groupware als ein „computer-based message system, which supports the asynchronous exchange of textual messages between groups of users“³³, also als ein Computersystem für den asynchronen Austausch von Textnachrichten innerhalb einer Gruppe. Meistens sind dies Arbeitsgruppen im Unternehmensumfeld. Microsoft Exchange erfüllt dieses Kriterium, vorausgesetzt die Benutzer verfügen über eine Client-Software.

Um auf die Inhalte des Microsoft Exchange Servers zugreifen zu können, benötigt jeder Benutzer eine Client-Software. Verwaltung des Postfaches, Zugriff auf öffentliche Ordner und natürlich auch Empfangen und Senden von E-Mails sind die Hauptziele einer Client-Software. Im Anhang auf Seite 43 findet man eine Abbildung, die eine umfassende Übersicht verschiedener Clients darstellt. Diese Übersicht ist nicht vollständig, bildet aber dennoch die am häufigsten verwendeten Client-Systeme ab.

Am bekanntesten ist sicher der Microsoft Outlook-Client. Aus diesem Grund wird die Verbindung eines Clients mit dem Exchange Server am Beispiel des Outlook-Clients erläutert. Outlook kommuniziert nach dem RPC-Prinzip mit dem Exchange Server. Die Variante RPC über TCP/IP gilt zwar seit Exchange 2013 als veraltet, ist aber für das nähere Verständnis durchaus wichtig, da alle Weiterentwicklungen RPC im Hintergrund weiter nutzen. RPC bedeutet „Remote Procedure Call“ und wird verwendet, um eine Verbindung zu einem Dienst eines Servers herzustellen. Als Übertragungsprotokoll fungiert TCP/IP (Transfer Control Protocol/Internet Protocol).³⁴ Um die Kommunikation von RPC über TCP/IP zu verstehen, ist die nachfolgende Abbildung hilfreich.

³² Vgl. Joos (2014), S. 26.

³³ Ellis, C. A. / Gibbs, S. J. / Rein, G. L. (1991), S. 38 ff.

³⁴ Website: Carius (abgerufen am: 13.06.2016)

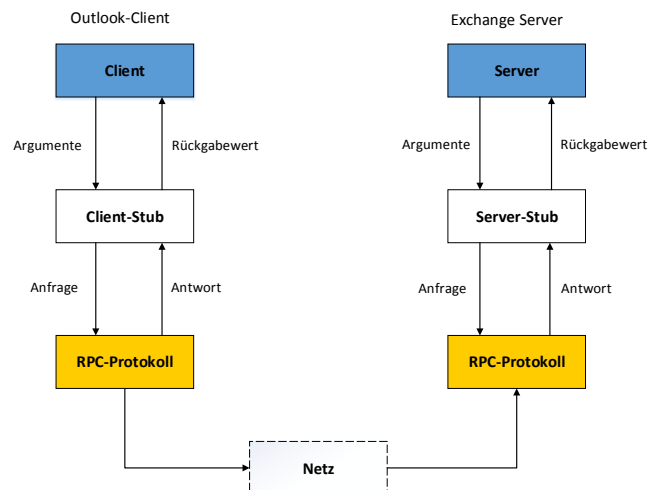


Abbildung 4: Aufbaueiner klassischen RPC Verbindung, in Anlehnung an:
<https://technet.microsoft.com/de-de/library/e8feb37e-f3a9-4f26-bed0-6583d8a110ed>

Der Client ruft eine Prozedur mit spezifischen Argumenten (Eingabeparameter) auf. Der Client-Stub aktiviert eine gleichnamige Prozedur und wandelt die Argumente in ein plattformunabhängiges Datenformat um. Die Daten werden dann über das Netz mithilfe des TCP/IP-Protokolls an den Server geschickt. Dort erhält der Server-Stub die Anfrage und wandelt die Argumente in das lokale Format des Servers um. Nun ruft der Server die gewünschte Prozedur mit den Eingabeparametern auf und der Rückgabewert kehrt in den Server-Stub zurück. Nach einer erneuten plattformunabhängigen Umwandlung der Datenformate wird die Antwort an den Client-Stub geschickt. Im letzten Schritt erhält der Client die in das lokale Format umgewandelte Antwort (Rückgabewerte) auf seine Anfrage.³⁵

Ab Exchange 20013 wird der Datenaustausch zwischen dem Outlook-Client und dem Exchange Server standardmäßig über RPC/HTTP (auch Outlook Anywhere genannt) geregelt. Wie der Name schon vermuten lässt, läuft die gesamte Kommunikation bei dieser Variante über HTTP bzw. HTTPS. Somit kann über das Internet auf den Exchange Server zugegriffen werden. Outlook Anywhere ist besonders für Mitarbeiter geeignet, die von zu Hause auf ihr Exchange-Postfach zugreifen möchten, da sie sich nicht im Unternehmensnetzwerk befinden müssen.³⁶

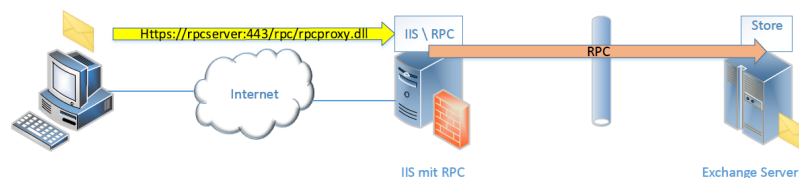


Abbildung 5: RPC over HTTP Prinzip, in Anlehnung an:[http:// www.msxfaq.de/clients/oagrundlagen.htm](http://www.msxfaq.de/clients/oagrundlagen.htm)

In der Abbildung 5 sieht man deutlich, dass der Outlook-Client über HTTPS mit dem IIS kommuniziert, auf dem ein virtuelles RPC-Verzeichnis existiert. Der IIS ist eine Microsoft-

³⁵ Vgl. Schneider (2012), S.406 f.

³⁶ Vgl. Joos (2014), S. 33, S. 254.

Webserverplattform, die Webanwendungen und Dienste bereitstellen und verwalten kann.³⁷ Der IIS wiederum baut über einen RPC-Proxy eine Verbindung mit dem Exchange Server auf. Wichtig ist hierbei, dass die Kommunikation des Clients sich auf HTTP oder HTTPS beschränkt.³⁸

5.2 Exchange Web Services

Die Grundlagen von Microsoft Exchange Server sind geklärt. Doch wie können nun Programmierer über Quellcode auf den Exchange Server zugreifen? Hierfür hat Microsoft im Laufe der Jahre eine Reihe von Programmierschnittstellen bereitgestellt. Eine Schnittstelle zur Anwendungsprogrammierung wird als API (englisch: application programming interface) bezeichnet. Es ist nicht nötig, alle Programmierschnittstellen zu erläutern, da eine Vielzahl der APIs keine Verwendung mehr findet oder nicht mehr unterstützt wird. Zu Informationszwecken befindet sich jedoch im Anhang auf Seite 44 eine Übersicht einiger Programmierschnittstellen ab dem Jahr 1992. Ab Exchange 2007 setzt Microsoft immer mehr auf die Exchange Web Services (kurz EWS) als Programmierschnittstelle und baut diese seitdem weiter aus.³⁹ Redmond schreibt im Handbuch Exchange 2010: „Abgesehen von Windows PowerShell liegt der Schwerpunkt für die meisten Entwickler jetzt auf der API EWS (Exchange-Webdienste), die in Exchange Server 2007 eingeführt wurde.“⁴⁰ Er macht deutlich, dass EWS die erste Anlaufstelle für Entwickler ist. Eine weitere Programmierschnittstelle namens REST API erschien 2015 mit Office 365. Der REST API wird aber keine weitere Beachtung geschenkt, da sie lediglich in der Office 365-Umgebung Anwendung findet.⁴¹

5.2.1 Funktionsweise

Standardmäßig bildet das Simple Object Access Protocol (kurz SOAP) eine Grundlage für Web-Services. Schneider beschreibt SOAP als „ein RPC-Mechanismus, bei dem die übertragenen Daten im XML-Format codiert werden.“⁴² Schill und Springer schreiben genauer: „Das objektorientierte Kommunikationsprotokoll SOAP ermöglicht die Kommunikation zwischen heterogenen Diensten unter interner Nutzung des Hypertext Transfer Protocol (HTTP) und mit Kodierung der Parameter in der eXtensible Markup Language (XML).“⁴³ Auch die Exchange Web Services funktionieren nach diesem Prinzip.⁴⁴

Mit Exchange 2007 wurden erstmals die Exchange Web Services bereitgestellt, die sich seitdem stetig weiterentwickelt haben. Die nachfolgende Abbildung macht diese Entwicklung bis zu Exchange 2010 deutlich.

³⁷ Vgl. Volodarsky, M. et al. 2009, S.3.

³⁸ Website: Carius (abgerufen am: 13.06.2016)

³⁹ Website: Carius (abgerufen am: 14.06.2016)

⁴⁰ Redmond 2011, S.634.

⁴¹ Website: Microsoft (abgerufen am: 15.06.2016)

⁴² Schneider 2012, S.407.

⁴³ Schill, A. / Springer, T. 2012, S.69.

⁴⁴ Website: Microsoft (abgerufen am: 13.06.2016)

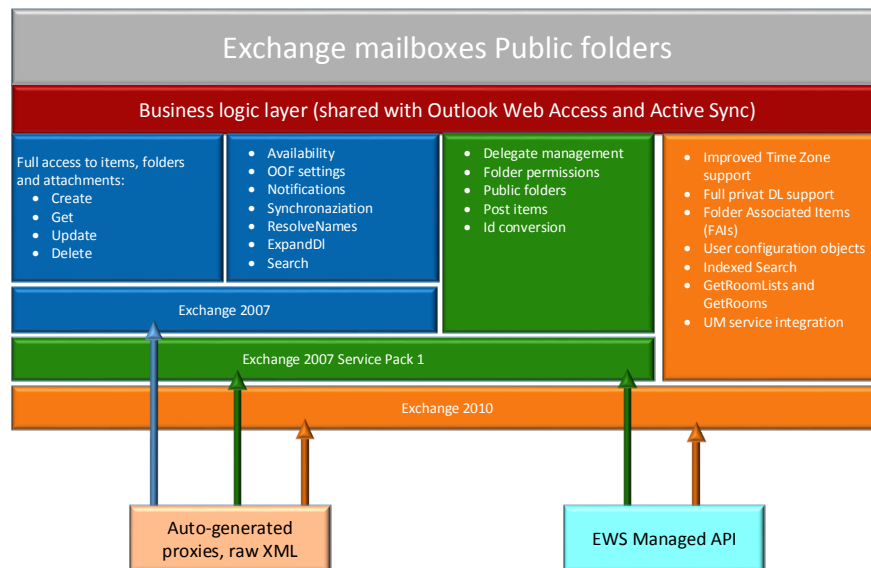


Abbildung 6: EWS Funktionsumfang, Quelle: <http://www.msxfaq.de/code/ews.htm>

EWS bietet einen großen Funktionsumfang und ist somit eine hervorragende Möglichkeit, auf die Daten des Exchange Servers zuzugreifen. Aus diesem Grund haben sich die Entwickler der KMS Computer GmbH für diese Programmierschnittstelle entschieden.

5.2.2 Derzeitige Verwendung in GEBman 10

Um bei der Implementierung den Exchange Web Service nutzen zu können, musste die Microsoft Exchange Web Services Managed API 2.2. dem Projekt als .NET-Assembly hinzugefügt werden. Wie bereits im Punkt 4 erwähnt, kann im Adminbereich von GEBman10 in der Rubrik GroupWare die Konfiguration des Exchange Servers durchgeführt werden. Wichtig ist hierbei zunächst, welche Version vom Exchange Server verlangt wird, da das entscheidend für die Microsoft Web Services Managed API ist. GEBman10 unterstützt folgende Exchange Server-Versionen:

- Exchange 2007 inkl. SP1
- Exchange 2010 inkl. SP1
- Exchange 2013

Nachdem die Version bestimmt und die Benutzerdaten eingetragen wurden, kann direkt im Code mittels der API direkt auf den Exchange Server zugegriffen werden. Derzeit ist das Versenden von E-Mails über den Code implementiert, sodass eine Grundlage für die Erweiterung bereits vorhanden ist. Auch das Auswerten von E-Mails ist über diese API möglich und eine Schnittstelle muss nicht mehr implementiert werden. Diese vorhandenen Funktionalitäten werden als Grundlage für die folgende Konzipierung und anschließende Implementierung genutzt.

6 Konzipierung

6.1 Vorbetrachtung

Für die Erstellung einer Meldung sind mehrere Informationen bzw. Eigenschaften nötig. Dabei sind einige Informationsangaben Pflichtfelder in der Benutzeroberfläche von GEBman 10. Füllt der Benutzer diese nicht aus, kann keine neue Meldung erstellt werden. Die Tabelle 3 zeigt nun einige Eigenschaften, die für die spätere Erstellung einer Meldung im Programmcode wichtig sein werden.

| Bezeichnung | Erläuterung |
|------------------------|---|
| * Art | Hier muss die Art der Meldung angegeben werden. Ist es beispielsweise eine Störmeldung oder eine Servicemeldung. |
| * Status | Dieses Pflichtfeld ist standardmäßig auf „unbearbeitet“ gestellt. |
| * Standort oder Objekt | Eines der beiden Felder muss ausgefüllt sein, um die Meldung einem Standort oder einem Objekt zuweisen zu können. |
| * Meldung | Das ist das wichtigste Feld, denn hier wird kurz beschrieben, welcher Vorfall sich ereignet hat. |
| Beschreibung | Für eine genauere Beschreibung des Vorfalls ist dieses Feld vorgesehen und bietet hierfür ausreichend Platz. |

Tabelle 3: Einige Eigenschaften für eine Meldung, Quelle: eigene Darstellung

Die Eigenschaften mit dem Stern (*) als Kennzeichnung sind die erwähnten Pflichtfelder. Bei diesen erforderlichen Eigenschaften ergibt sich ein Problem für die Erstellung einer Meldung über die E-Mail Integration. Derjenige, der eine E-Mail an den Exchange Server von GEBman 10 schreibt, müsste eine genaue Angabe über den Standort und/oder das Objekt der Meldung machen. Das ist in der Praxis nicht bedienerfreundlich und könnte zu Schreibfehlern führen. Auf der Seite der Implementierung hieße das, dass ein möglicher Schreibfehler abgefangen werden muss, da dieser Standort oder diese Objekte nicht in der Datenbank existieren. Außerdem hatten bereits mehrere Kunden von der KMS Computer GmbH dieses Pflichtfeld als nicht hilfreich bezeichnet. Will der Kunde einfach nur Meldungen in dem Service Desk Modul aufgeben, ist diese Eigenschaft unnütz. Aus diesen Gründen wurde entschieden, die Felder Standort und Objekt nicht mehr als notwendige Eigenschaften einer Meldung zu definieren. Dadurch werden diese beiden Eigenschaften in der Implementierung auch weiter beachtet und der Service Desk wird weiter vom Facility Management Bereich gelöst.

Die Informationsangabe Beschreibung (ohne den Stern) ist bei der Erstellung einer Meldung nicht unbedingt nötig, aber durchaus sinnvoll. Es gibt noch weitere Felder, die der Benutzer in der Benutzeroberfläche ausfüllen kann, die aber nicht verpflichtend sind. Diese zusätzlichen Informationen sind für die nachfolgenden Ausführungen nicht relevant.

6.2 Zielsetzung und Grundidee Module Service

Ausgangspunkt ist ein Benutzer, der Mails an den Exchange Server schreiben kann, um somit neue Meldungen im Service Desk Modul anzulegen. Alle E-Mails, die an den Exchange Server gesendet werden, landen im Posteingang. Ziel ist es, durch das Auswerten dieser im Posteingang befindlichen E-Mails, Meldungen zu erstellen, ohne dass sich ein Benutzer in GEBman 10 anmelden muss. Der Benutzer soll somit eine neue Meldung erstellen oder auf eine vorhandene Meldung antworten können. Zentrales Hilfsmittel hierfür wird die ID der Meldung sein. Will der Benutzer eine neue Meldung erstellen, beschreibt er in der Betreffzeile der E-Mail das konkrete Probleme (hier zunächst ohne ID) und schickt die E-Mail an den Exchange Server, der in GEBman 10 konfiguriert wurde. Der Module Service wird in regelmäßigen Abständen alle E-Mails auswerten und die Betreffzeile überprüfen. Sollte eine E-Mail keine ID aufweisen, handelt es sich um eine neue Meldung. Diese wird im Service Desk erstellt und bekommt eine eindeutige ID. Wenn die Meldung erfolgreich erstellt wurde, soll eine Bestätigungsmail an den Ersteller geschickt werden, die in der Betreffzeile die ID der neu angelegten Meldung enthalten soll.

Wenn der Benutzer einer Meldung eine Notiz o.ä. hinzufügen möchte, so muss in der Betreffzeile der E-Mail die entsprechende ID der Meldung vorhanden sein. Der Module Service soll diese ID erkennen und eine Antwort für die entsprechende Meldung generieren. Demzufolge ist der Betreff einer E-Mail entscheidend für das weitere Vorgehen im Programmcode.

Außerdem soll es möglich sein, eventuell vorhandene E-Mail Anhänge direkt an die Meldung zu binden. In GEBman 10 hat jede Meldung bereits einen Bereich für Dokumente. Über Programmcode kann dieser Bereich erweitert werden. Hierfür wird der E-Mail Anhang als Dokument an die Meldung gebunden.

In GEBman 10 ist die Klasse *ModuleService* implementiert, die es ermöglicht, einen Service asynchron vom restlichen System für das entsprechende Modul laufen zu lassen. Demnach muss in GEBman 10 ein neuer Service erstellt werden, der in bestimmten Intervallen seine Funktion ausführt. Das geschieht im Code im *Package Common*. In diesem zentralen Package werden Funktionalitäten implementiert, die für alle Module allgemein gültig sind. Dadurch kann die Erweiterung später auch für andere Module verwendet werden und nicht nur im Service Desk Modul.

Nach dem Auswerten der E-Mails im *Package Common* muss im Modul Service Desk eine Meldung erstellt oder bearbeitet werden. Das heißt, dass diese Funktionalität im *Package ServiceDesk* implementiert werden sollte. Hier befinden sich alle Komponenten, die auf die Meldungen in GEBman 10 zugreifen.

Um diese Grundidee besser erläutern und umsetzen zu können, ist eine Darstellung der einzelnen Schritten in verschiedenen UML-Diagrammen sinnvoll.

6.3 UML - Modellierung

Grundlage der Modellierung bildet die grafische Notation Unified Modeling Language (UML) in der Version 2.3. UML hat sich in den letzten Jahren bei der Erstellung objektorientierter Modelle bewährt und ermöglicht somit einheitliche Diagrammdarstellungen und Begriffsabgrenzungen. Deshalb kann UML als Standard für die Modellierung objektorientierter Software gesehen werden.^{45,46}

Grundsätzlich werden zwei Sichtweisen in der UML unterschieden. Die nachfolgende Tabelle erläutert die wichtigsten Unterschiede:

| Diagrammtyp | Verhaltensdiagramm | Strukturdiagramm |
|--------------|--|---|
| Sichtweise | dynamisch | statisch |
| Beispiele | Aktivitätsdiagramm, Zustandsdiagramm, Sequenzdiagramm | Klassendiagramm, Objektdiagramm, Paketdiagramm |
| Beschreibung | Es werden die Komponenten des Systems erläutert, die sich während der Laufzeit verändern. Dabei werden die Abläufe des Systems ersichtlich und auch inwiefern der Benutzer diese beeinflusst | Aus dieser Sicht werden die Komponenten des Systems betrachtet, die unabhängig von der Laufzeit sind. Ihre Ein- und Ausgabedaten können sich verändern, aber die Beziehungen zwischen den Komponenten bleiben bestehen. |

Tabelle 4: UML-Diagrammtypen, Quelle: in Anlehnung an Schneider (2012), S. 234.

Insgesamt gibt es sieben Verhaltens- und sieben Strukturdiagramme. Es können nicht alle vierzehn Diagramme in dieser Arbeit Platz finden und bei einem solchen Vorgehen würde auch der Fokus auf die wichtigsten Fragestellungen verloren gehen. Aus diesem Grund wurden zwei Verhaltensdiagramme und ein Strukturdiagramm ausgewählt, die die Abläufe bei der Erweiterung des Service Desk von GEBman 10 ersichtlich machen und die Implementierung erleichtern sollen.

Das erste Verhaltensdiagramm soll die Frage klären, welche Aktionen der Benutzer mit dem Versenden einer E-Mail ausführen kann und wie das System darauf reagiert. Im Punkt 4.2 wurde auf die Anforderungen der Erweiterung eingegangen. Diese werden nun mit dem Aktivitätsdiagramm in der Abbildung 7 erläutert.

Der Benutzer muss sich zu Beginn entscheiden, ob er eine neue Meldung erstellen oder auf eine bestehende Meldung antworten möchte. Sollte er ein neues Problem erkannt haben, schickt er eine E-Mail an GEBman 10. Im Betreff muss die Meldung eingetragen werden und im Textkörper der E-Mail kann der Benutzer die ausführliche Beschreibung des Problems angeben.

⁴⁵ Vgl. Schneider (2012), S. 233.

⁴⁶ Vgl. Balzert (2010), S. V.

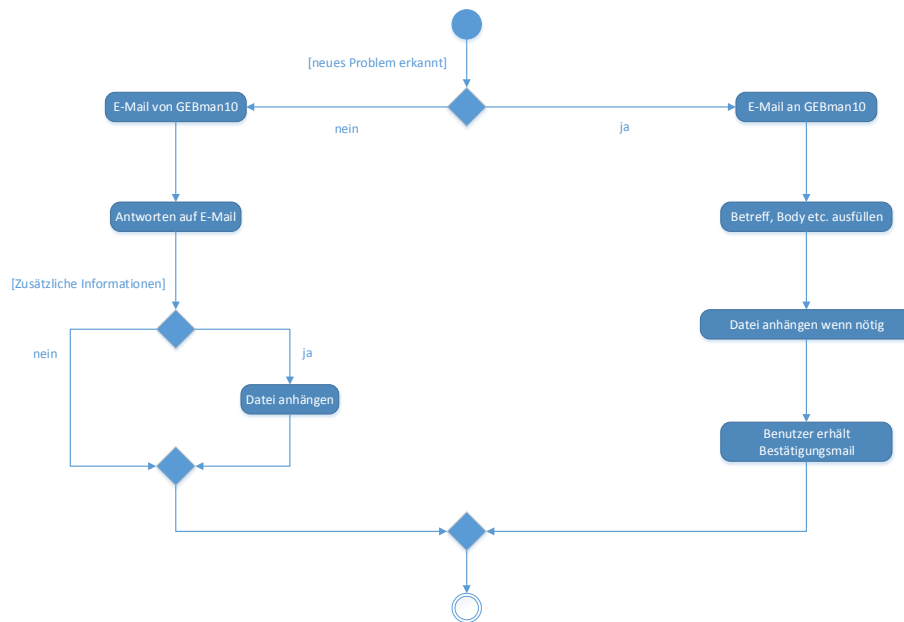


Abbildung 7: Aktivitätsdiagramm der Erweiterung, Quelle: eigene Darstellung

Der ModuleService in GEBman 10 wertet diese E-Mail aus und legt eine neue Meldung im Service Desk an. Diese Meldung erhält eine eindeutige ID. Der Benutzer erhält anschließend eine Bestätigungsmail mit der ID der Meldung, wenn die Erstellung erfolgreich war.

Möchte der Benutzer allerdings Rückmeldung auf bereits bestehende Meldungen geben, so kann er direkt auf eine E-Mail von GEBman 10 antworten. Der Betreff wird automatisch übernommen und somit ist die ID schon eingetragen. Sollte der Benutzer zusätzliche Informationen wie eine Excel-Tabelle oder ein Bild an die Meldung binden wollen, kann er der E-Mail einen Anhang hinzufügen.

Anders als bei dem Aktivitätsdiagramm wird in der nachfolgenden Abbildung ein Zustandsdiagramm dargestellt. Die beiden Diagramme ähneln sich von ihrer Notation sehr. Das Zustandsdiagramm legt den Fokus jedoch auf die Zustände des Systems, die es während der Laufzeit annehmen kann. Deshalb ist es auch das zweite Verhaltensdiagramm. Hierbei ist es wichtig, dass immer ein Ereignis eintreffen muss, damit das System in einen anderen Zustand wechseln kann.⁴⁷

Sobald der Webserver gestartet ist, befindet sich das System im Zustand aktiver Module Service. In regelmäßigen Zeitabständen, werden über diesen Service die neuesten Mails von dem Exchange Server geholt. Dann werden die in der Betreffzeile der Mail befindlichen ID's ausgewertet. Ist keine ID vorhanden, geht der Module Service in den Zustand einer neuen Meldungserstellung über. Dabei werden alle nötigen Informationen der E-Mail entnommen. Anschließend wird die Bestätigungsmail versendet, wenn die Erstellung erfolgreich war.

⁴⁷ Vgl. Balzert (2010), S. 40.

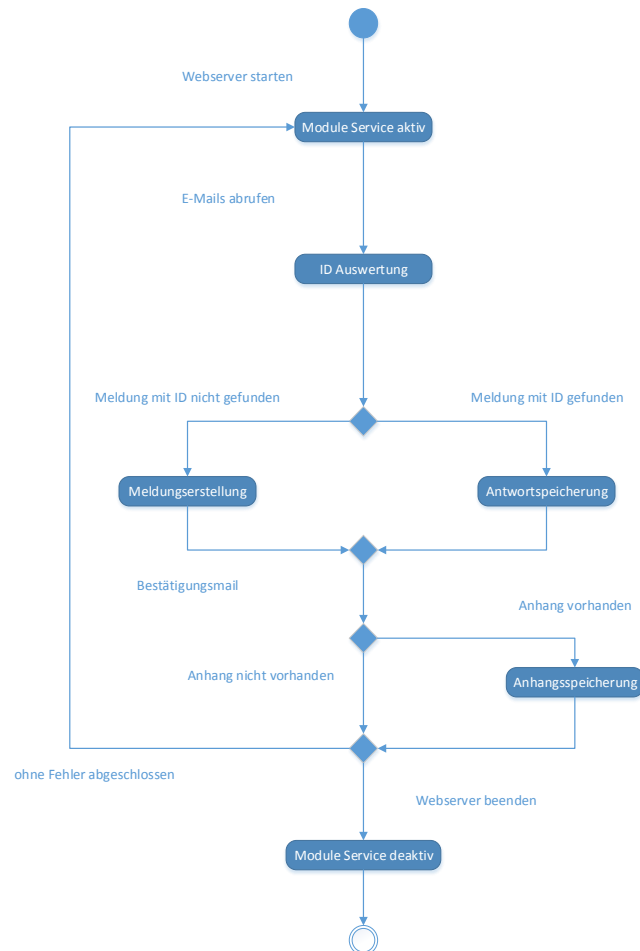


Abbildung 8: Zustandsdiagramm der Erweiterung, Quelle: eigene Darstellung

Wurde bei der ID Auswertung allerdings festgestellt, dass eine Meldung mit identischer ID vorhanden ist, wechselt das System in den Zustand der Antworterstellung. Hier wird der Textkörper der E-Mail als Antwort der Meldung hinzugefügt.

Nachdem eines der beiden Szenarien abgearbeitet wurde, kann das System noch in den Zustand der Anhangsspeicherung gelangen. In diesem Zustand wird dann der Anhang der E-Mail an die Meldung gebunden. Wurden alle Prozesse ohne Fehler abgeschlossen, kehrt das System wieder zu dem Anfangszustand des aktiven Webservices zurück. Der Module Service wird dann im nächsten Intervall die Prozedur wiederholen. Nur wenn der Webserver beendet wird, ist logischerweise auch der Module Service deaktiviert. Ansonsten wird der Service permanent laufen.

Die Verhaltensdiagramme aus Benutzer- und Systemperspektive sind somit abgeschlossen. Das Klassendiagramm in Abbildung 9 ist ein Strukturdiagramm, welches einen groben Überblick über den zu implementierenden Module Service und seiner Beziehung zu dem restlichen System geben soll.

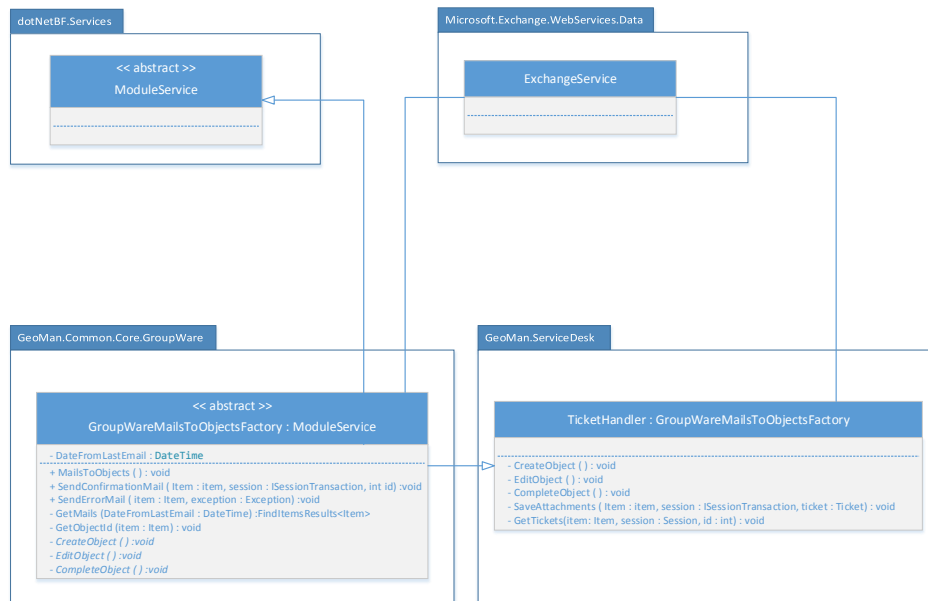


Abbildung 9: Klassendiagramm der Erweiterung, Quelle: eigene Darstellung

Die Funktionalitäten für das Versenden von E-Mails in GEBman 10 befinden sich im Package *GeoMan.Common.Core.GroupWare*. Die abstrakte Klasse für den Module Service wird *GroupWareMailToObjectFactory* heißen und sich ebenfalls in diesem Package befinden. Sie erbt von der ebenfalls abstrakten Klasse *ModuleService*, welche sich in einem Framework befindet, das ein Partnerunternehmen der KMS Computer GmbH entwickelt.

GroupWareMailToObjectFactory wird nur das öffentliche Attribut *DateFromLastEmail* besitzen. Diese Variable speichert das Ankunftsdatum von der neuesten E-Mail im Exchange Postfach und hat deshalb den Datentypen *DateTime*. Somit müssen nicht alle E-Mails abgerufen werden, sondern nur die, die nach dem Ankunftsdatum der letzten E-Mail eingetroffen sind.

Diese Klasse *GroupWareMailToObjectFactory* wird auf die Exchange Webservices zugreifen und sich Methoden der Klasse Exchange Service bedienen, um die E-Mails von Exchange Server abzufragen. Das übernimmt die Methode *GetMails()*, die den Zeitpunkt der zuletzt abgerufenen E-Mail als Argument übergeben bekommt. Der Rückgabewert ist eine ItemView vom Exchange Webservice. Deshalb wird eine Referenz auf die *Microsoft.Exchange.WebServices.Data* benötigt.

Die Methode *CheckIdFromMails()* wird die ID's überprüfen, die in den Betreff der Mails eingetragen wurde. Auch ist sie dafür zuständig, die entsprechenden Methoden für das weitere Vorgehen aufzurufen. Die Methoden *SendConfirmationMail()* und *SendErrorMail()* werden dafür zuständig sein, eine Bestätigungsmail oder eine Fehlermail zu versenden, wenn der entsprechende Fall eintritt. Es wird drei abstrakte Methoden geben, die eine Meldung erstellen (*CreateObject*), eine Meldung bearbeiten (*EditObject*) oder den Status einer Meldung ändern (*CompleteObject*). Ihnen wird jeweils die entsprechende Mail als Attribut mitgegeben.

Die abstrakten Methoden werden dann in der Klasse *TicketHandler* überschrieben. Erst in dieser Klasse wird es zu der eigentlichen Erstellung, Bearbeitung oder Änderung des Status kommen. Dafür muss die Klasse *TicketHandler* von der abstrakten Klasse *GroupWareMailToObjectFactory* erben. Die Microsoft Web Services werden ebenfalls in der Klasse *TicketHandler* benötigt, da erst hier die Parameter wie E-Mail-Body oder Recipient ausgewertet werden.

6.4 Sicherheitsaspekte

Immer wieder vernachlässigen Entwickler die Sicherheit ihrer Implementierungen.^{48,49} Das liegt häufig an mangelnder Zeit, da Releases einen festen Zeitplan verfolgen, den es einzuhalten gilt. Es kann aber auch sein, dass die Implementierung nicht aus dem Blickwinkel der Sicherheit betrachtet wird und der Fokus eher auf Bereiche wie Benutzerfreundlichkeit gelegt wird. Natürlich hat das wenig mit Sicherheit zu tun. Dabei können es Entwickler mit wenig Aufwand Angreifern deutlich schwerer machen. Deswegen werden im Nachfolgenden zwei Sicherheitsprobleme für die Umsetzung des Konzepts in GEBman 10 besprochen. Die Abbildung 10 zeigt zwei kritische Bereiche, die genauer erläutert werden müssen.

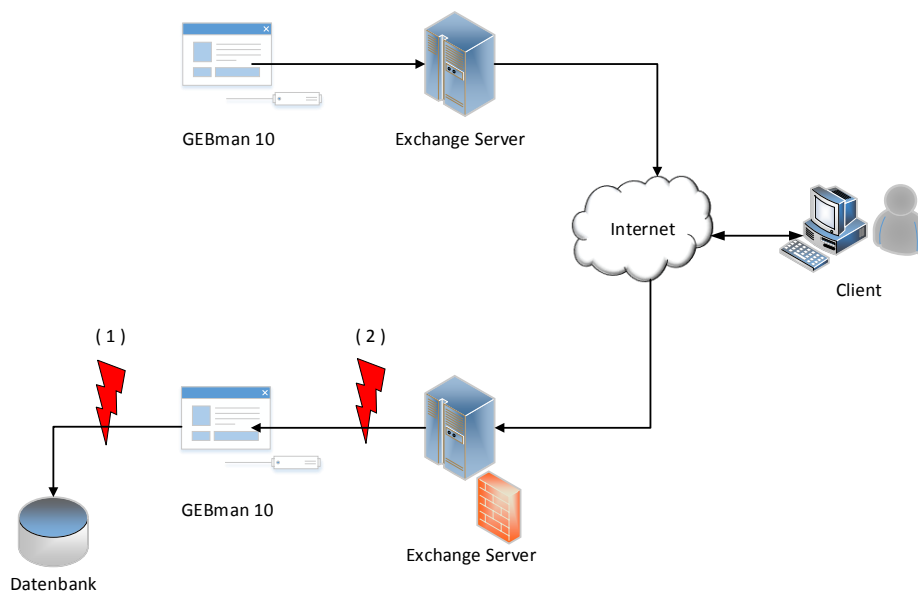


Abbildung 10: Sicherheitsprobleme der Erweiterung, Quelle: eigene Darstellung

Der erste kritische Bereich - gekennzeichnet mit der (1) - symbolisiert das erste Problem. Sollte der Exchange Server mit E-Mails überhäuft werden, so wird für jede E-Mail eine neue Meldung in GEBman 10 angelegt. Das bedeutet, dass wenn pro Minute 100 E-Mails den Exchange

⁴⁸ Vgl. Ziegler (2014), S. 53.

⁴⁹ Website: Gaßner (abgerufen am: 23.06.2016)

Server erreichen, auch 100 Meldungen angelegt werden. Das würde das System deutlich verlangsamen, wenn nicht sogar zum Absturz bringen. Eine Möglichkeit, dieser Gefahrenquelle entgegenzutreten, ist, die Sender der E-Mails auszuschließen, sobald diese eine ungewöhnlich hohe Anzahl an E-Mails versenden. Dadurch würden dann die E-Mails von dem Benutzer nicht ausgelesen werden, alle anderen E-Mails jedoch schon.

Bei dem zweiten kritischen Bereich der Abbildung 10 - gekennzeichnet mit der (2) - muss auf die Speicherung der Daten in der Datenbank geachtet werden. GEBman 10 synchronisiert in regelmäßigen Abständen die Nachrichten vom hinterlegten Exchange Server. Entsprechend ihrer ID werden die Nachrichten in der Datenbank von GEBman 10 gespeichert. Nun könnte ein Angreifer beispielsweise versuchen, in den Textkörper der E-Mail JavaScript-Code oder HTML Befehle einzubetten. Diese Angriffsmethode nennt sich Cross-Site-Scripting (XSS). Gelangt nun ein Benutzer auf die Seite, auf der dieser XSS-Code injiziert wurde, könnten Informationen preisgegeben und versendet werden, auf die es der Angreifer abgesehen hat.⁵⁰

Es wäre möglich, den Textkörper und den Betreff einer E-Mail zu prüfen, bevor diese in der Datenbank gespeichert wird. Jedoch ist das nicht nötig, da GEBman 10 über spezielle Abwehrmechanismen verfügt, die das Injizieren von XSS-Code nutzlos macht und somit keine Gefahr für einen Benutzer darstellt.

⁵⁰ Vgl. Rohr (2015), S. 90ff.

7 Umsetzung

7.1 Erweiterung des bestehenden Service Desk Moduls

Grundlage für die Erweiterung ist die bereits erwähnte Microsoft Exchange Web Services Managed API 2.2, die kostenlos zum Download von Microsoft angeboten wird. Voraussetzung für diese API ist ein Betriebssystem von Windows (mindestens Windows 7) und das .NET Framework 3.5 oder höher.⁵¹

Da diese API bereits in GEBman10 für das Versenden von E-Mails integriert wurde, konnte direkt ohne zusätzlichen Aufwand auf die Funktionalitäten zugegriffen werden. Wichtig war lediglich das Einfügen des Assemblerverweises in die entsprechenden Projektmappenordner. Um nicht den vollständigen Namespace jeder Klasse ausschreiben zu müssen, wurde außerdem die Using-Direktive in den Klassen *MailsToObjectsFactory* und *TicketHandler* hinzugefügt.

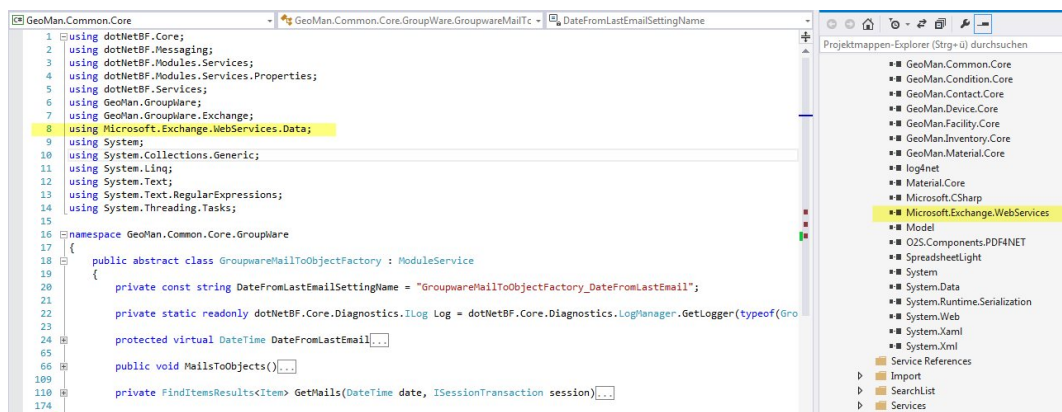


Abbildung 11: Verweise der EWS Managed API, Quelle: eigene Darstellung

Bei der Implementierung wurde sich stark an das Fabrikmethoden-Entwurfsmuster gehalten. Bevor hierauf näher eingegangen wird, sollte das Entwurfsmuster näher erläutert werden.

Die Fabrikmethode (Factory Method) ist ein Erzeugungsmuster, bei dem die Objekterstellung von der Objektverarbeitung getrennt wird. Hierbei wird die Unterklasse durch eine abstrakte Methode der Oberklasse erzeugt.⁵² Dieses Entwurfsmuster wurde auch bei der Implementierung in GEBman10 verwendet.

Die Fabrikmethode (Abbildung 12) macht deutlich, dass die abstrakte Klasse *GroupwareMailToFactory* der Erzeuger und *TicketHandler* der konkrete Erzeuger ist. Der konkrete Erzeuger wiederum erstellt dann ein Objekt über die abstrakte Methode *CreateObject*(). Das Objekt ist in diesem Falle ein Ticket (Meldung). Vorteil dieses Entwurfsmusters ist die klare Kapselung des Erzeugers. Somit kann die Klasse *GroupwareMailToFactory* auch für andere Module eingesetzt werden, die ebenfalls neue Objekte aus E-Mails erzeugen sollen.

⁵¹ Website: Microsoft (abgerufen am: 21.06.2016)

⁵² Vgl. Eilebrecht, K. / Starke, G. 2013, S.34ff.

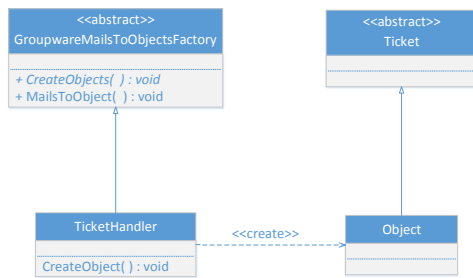


Abbildung 12: Entwurfsmuster Fabrikmethode, Quelle: in Anlehnung an Eilebrecht, Starke (2013) S.35

```
<SchedulerService>
<InMemoryTimerService />
<ScheduleSubscription Name="GeoMan.ServiceDesk.TicketHandler" Schedule="T00:01:00">
  var service : GeoMan.ServiceDesk.TicketHandler =
    Module.GetService(typeof(GeoMan.ServiceDesk.TicketHandler));
  service.MailsToObjects();
</ScheduleSubscription>
</SchedulerService>
<GeoMan.ServiceDesk.TicketHandler />
```

Abbildung 13: Aufruf des Module Service, Quelle: eigene Darstellung

Die beiden Experten für Entwurfsmuster Eilebrecht und Starke sehen die Anwendung der Fabrikmethode als sinnvoll, wenn „eine Klasse die von ihr zu erzeugenden Objekte nicht im Voraus kennt“.⁵³ Genau dieser Anwendungsfall trifft hier zu und ist deshalb auch gerechtfertigt.

Bei jedem Intervall wird über den *SchedulerService* (Abbildung 13) der Module Service *TicketHandler* instanziiert und die Methode *MailsToObjects()* der abstrakten Oberklasse aufgerufen. Das Intervall wurde hier mit „T00:01:00“ bestimmt, was einen Zeitabstand von einer Minute entspricht.

7.2 Erläuterung einzelner Methoden

Die Methode *GetMails()* in der Klasse *GroupwareMailsToObjectFactory* ist die entscheidende Methode der Erweiterung bei dem Abrufen der E-Mails eines Exchange Servers. Hier werden nicht nur alle Mails abgerufen, sondern auch verschiedene Attribute der E-Mails geladen. Im Anhang auf Seite 47 befindet sich ein Codeausschnitt, in dem diese Schritte in der Programmiersprache C#/.NET implementiert sind. Zu sehen ist hier, wie zunächst alle Exchange Server aus der Datenbank in einer Liste gespeichert werden. Daraufgehend wird der erste Server aus der Liste dazu genutzt, einen Exchange Service zu erzeugen. Für den Exchange Service sind Benutzername, Passwort, die Domain und eine Autodiscover URL notwendig, um auf alle Funktionalitäten zugreifen zu können. Nun können auch schon alle E-Mails aus dem Postfach des Exchange Servers abgefragt werden. Jedoch müssen einige Eigenschaften (Properties) der E-Mails (Items) erst geladen werden, bevor auf sie zugegriffen werden kann. Alle für den weiteren Verlauf benötigten Eigenschaften sind in der Variablen *additionalProperties* definiert und werden mit der Methode *LoadPropertiesForItems()* geladen. Der Exchange Service speichert alle E-Mails (Items), die er gefunden hat, in die *ItemCollection result*, die am Ende der Methode *GetMails()* den Rückgabewert bildet.

Das Erstellen einer Meldung (bzw. eines Objektes) erfolgt in der Methode *CreateObject()*. Diese Methode ist in der *GroupwareMailsToObjectFactory* als abstrakte Methode definiert, wird

⁵³ Eilebrecht, K. / Starke, G. 2013, S.34ff.

der Klasse *TicketHandler* vererbt und hier auch explizit aufgerufen. Wichtig bei diese Methode ist die Übergabe einer Session. In GEBman10 benötigt jede Datenbankbearbeitung eine neue *SessionTransaction*.⁵⁴ Mithilfe der Session lässt sich somit auf die Create-Methode der Ticket Klasse zugreifen. Anschließend werden die Eigenschaften des E-Mail-Items den zuvor ange-deuteten nötigen Eigenschaften eines Tickets zugewiesen. Nach diesem Prozess wird die Sessi-on *committed*, um die neu erstellte Meldung in der Datenbank zu speichern. Erst jetzt erhält die Meldung eine eindeutige ID. Diese ID kann dazu genutzt werden, um im nächsten Schritt einen möglichen Anhang der E-Mail zu speichern. Das übernimmt die Methode *SaveAttachment()*. In dieser Methode wird zunächst ein Dokument mit demselben Inhalt des Anhangs der E-Mail erstellt. Dann wird dieses Dokument dem entsprechenden Ticket mit Hilfe der übergebenen ID zugeordnet. Wurden diese Anweisungen erfolgreich durchgeführt, wird eine Bestätigungsmail mit der Methode *SendConfirmationMail()* gesendet.

Bei jedem Versuch, die Datenbank zu verändern, kann es zu Fehlern kommen, die dem Be-nutzer auch mitgeteilt werden müssen. Der Benutzer muss schließlich darüber informiert wer-den, wenn eine Meldung nicht korrekt angelegt wurde. Aus diesem Grund wurden try-catch-Anweisungen bei jedem Datenbankzugriff verwendet. Durch diese try-catch-Anweisungen wer-den Ausnahmen beim Ausführen des Codes abgefangen. Dadurch ist es möglich, eine Ausnah-mebehandlung zu vollziehen.⁵⁵

7.3 Testfälle

In den letzten Jahren hat sich das Testen von Software zunehmend etabliert und ist ein fester Bestandteil von Softwareprojekten für die Qualitätssicherung.⁵⁶ Deshalb werden auch bei der Erweiterung des Service Desk Moduls einige Testfälle die wichtigsten Funktionalitäten über-prüfen. Hierfür muss zunächst ein Testablauf erarbeitet und die zu erwartenden Werte des Test-resultats festgehalten werden.

Bei den Testfällen in GEBman 10 wird im ersten Schritt der Webserver gestartet. Dadurch wird auch der implementierte Module Service *TicketHandler* initialisiert. Nun werden alle Modul-tests ausgeführt. Für den Test des Module Service wird ein Exchange Server benötigt. Die-ser wird mittels einer Methode in der *TestHelper*-Klasse angelegt. An diesen Exchange Ser-vice wird eine E-Mail gesendet und anschließend der Module Service aufgerufen. Theoretisch könnte der Modultest auch bis zum nächsten Intervall des Module Service warten, bis dieser die neuesten Mails vom Exchange Server abfragt. Doch ein Modultest sollte so wenig Zeit wie möglich beanspruchen. Aus diesem Grund wird die Methode *MailsToObjects()* vom Module Service manuell aufgerufen. Eine neue Meldung wird aus der zuvor gesendeten E-Mail erstellt.

⁵⁴ Vgl. Fowler 2003, S.103ff.

⁵⁵ Website: Microsoft. (abgerufen am: 27.06.2016)

⁵⁶ Vgl. Vivenzio, A. / Vivenzio, D. 2012, S.1f.

Erst jetzt beginnt der eigentliche Test. Es wird eine Datenbankabfrage erstellt, die nach der Meldung sucht, die der Module Service angelegt haben müsste. Sollte die Abfrage erfolglos sein, wird der Modultest mit einem Fehler beendet. Sollte die Meldung gefunden werden, wird im nächsten Schritt geprüft, ob der Anhang der E-Mail als Dokument der Meldung angelegt wurde. Wurde kein Dokument gefunden, das der zuvor erstellten Meldung zugeordnet ist, schlägt der Modultest fehl und eine entsprechende Meldung wird ausgegeben.

Die Testfälle können noch weiter ausgebaut werden, indem weitere Methoden implementiert werden, die eine erfolgreiche Antworterstellung oder das Versenden von Bestätigungsmails prüfen. Abschließend ist festzuhalten, dass niemals alle möglichen Szenarien in Testfällen berücksichtigt werden können, aber auch gar nicht müssen. Es geht lediglich darum, die Szenarien zu testen, die am häufigsten in der Praxis auftreten.⁵⁷

7.4 Fehlschläge/Erfahrungen

Bei der Implementierung in GEBman10 gab es zwei nicht vorhersehbare Probleme. Beide haben ihren Ursprung in der Managed API von Microsoft. In der Methode *GetMails()* werden alle Mails aus dem Posteingang des Exchange Servers abgefragt. Hierfür stellt die API eine Methode namens *FindItems()* zur Verfügung, die eine *Collection* aller Items liefert. Nun muss aber für diese Collection eine *ItemView* übergeben werden, die einen bestimmten Integerzahlenwert als festdefinierte Größe benötigt. Es ist also in der Theorie durchaus möglich, dass sich mehr Mails im Posteingang des Exchange Servers befinden, als die Menge an Elementen der *ItemView*. Das hat zur Folge, dass alle Mails außerhalb der *ItemView* nicht in der *Collection* von der Methode *FindItems()* enthalten sind und auch nicht ausgewertet werden können. Um dieses Problem kurzerhand zu beseitigen, wurde die *ItemView* Größe auf den maximalen Zahlenwert des Integerwertebereiches gesetzt. Diesen Wert zu überschreiten ist in der Praxis so gut wie unmöglich, da er sich in einem 10-stelligen Bereich bewegt.

Das zweite Problem bestand bei der Abfrage der neuesten Mails vom Exchange Server, ebenfalls in der *GetMails()*-Methode beschrieben. Bei jedem Abfrage-Intervall wird der Zeitpunkt der zuletzt eingetroffenen Mail gespeichert. Beim nächsten Intervall-Durchlauf werden dann nur alle Mails abgefragt, die ein aktuelles Datum besitzen. Hierfür bietet die Managed API einen *SearchFilter* an, der die Mails nach definierten Eigenschaften suchen kann. In diesem Fall ist die Filter-Methode *IsGreaterThan()* optimal für dieses Vorhaben. Theoretisch hätten alle Mails abgefragt werden müssen, die ein „größeres“ und damit aktuelleres Datum als die zuletzt eingetroffene Mail vom vorherigen Intervalldurchlauf hatten. Tatsächlich hat sich beim Debuggen gezeigt, dass der Filter zusätzlich ebenfalls die Mails liefert, die das gleiche Datum besitzen. Dadurch würde eine Mail in jedem Intervall zweimal gefiltert werden, was eine wo-

⁵⁷ Vgl. Witte 2015, S.11f.

möglich redundante Objekterzeugung zur Folge hätte. Wodurch dieses Problem entstand, bleibt unklar. Umgangen wurde es, indem dem Datum der zuletzt eingetroffenen Mail nur eine Sekunde hinzugefügt wurde. Nach dieser minimalen Veränderung wurden die Mails korrekt gefiltert.

7.5 Erweiterungsmöglichkeiten der E-Mail Integration

Die E-Mail Integration könnte dahingehend erweitert werden, dass der Benutzer mit einer E-Mail mehr Möglichkeiten zum Verändern vorhandener Meldungen hat. Sollte ein Benutzer beispielsweise den Status einer Meldung auf „Technisch fertig“ setzen wollen, könnte er das über eine E-Mail bewerkstelligen, ohne dabei in GEBman 10 eingeloggt sein zu müssen. Denkbar wäre es, in den Betreff der E-Mail einen Parameter wie „#fertig“ einzusetzen. Beim Auslesen der E-Mails durch den Module Service könnte das erkannt werden und dem Absender der E-Mail könnte ein Link geschickt werden. Mit diesem Link könnte der Benutzer dann zu einem temporären Webfrontend gelangen, auf dem dann die Fertigstellung der Meldung noch einmal bestätigt wird.

In GEBman 10 können mehrere Benutzer mit ihren E-Mail-Adresse hinterlegt werden. Der Module Service könnte abgleichen, ob ein Absender einer E-Mail im Exchange Postfach eine identische E-Mail Adresse wie ein registrierter Benutzer in GEBman 10 hat. Ist die E-Mail Adressen eines hinterlegten Benutzers identisch mit der E-Mail-Adresse eines Absenders für eine neue Meldung, kann in der Meldung im Service Desk-Modul direkt der registrierte Benutzer als Melder eingetragen werden. Die Eigenschaft Melder ist zwar nicht unbedingt notwendig, dadurch kann eine Meldung aber besser gefiltert werden und man hat direkte Einsicht darauf, an wen man sich bei Fragen wenden kann.

Des Weiteren wird in der prototypischen Implementierung die Abfrage der E-Mails nur für einen Exchange Server aus der Datenbank durchgeführt. Es ist in GEBman 10 aber durchaus möglich, mehrere Exchange Server zu konfigurieren. Diese zusätzlichen Server werden derzeit noch nicht berücksichtigt und demzufolge werden auch keine E-Mails abgerufen. Da alle Exchange Server im Programmcode in einer Liste gespeichert werden, sollte dies jedoch keine große Hürde darstellen.

8 Fazit

In dieser Bachelorarbeit wurde das Service Desk Modul von GEBman 10 auf Verbesserungsmöglichkeiten untersucht und eine Erweiterung der E-Mail Integration auf Basis von den Microsoft Exchange Web Services implementiert. Hierfür wurde zunächst der Begriff Service Desk erläutert, in einen Kontext gebracht und allgemeine Aufgaben festgehalten. Daraus hat sich ergeben, dass der Service Desk den Grundstein für eine gute Kommunikation zwischen Unternehmen und Kunden legt. Demnach ist das Arbeiten mit einer Service Desk-Softwarelösung Bestandteil des Arbeitsalltags und muss daher speziellen Anforderungen gerecht werden.

Anschließend wurden im Punkt 3 verschiedene Service Desk-Softwarelösungen getestet und aus bestimmten Blickwinkeln analysiert. Diese Lösungen hatten sowohl Unterschiede als auch Gemeinsamkeiten in Hinblick auf die Funktionalitäten. Das lag nicht zuletzt an den unterschiedlichen Einsatzgebieten und gesetzten Schwerpunkten der einzelnen Lösungen.

Die Betrachtung der aktuellen Umsetzung des Service Desk Moduls im Punkt 4 bot einen Einblick in die Funktionalitäten eines speziell auf den Bereich Facility Management abgestimmten Service Desk. Mit Hilfe der Analyse aus Punkt 3 war es möglich, Verbesserungsmöglichkeiten für das Service Desk-Modul von GEBman 10 zu identifizieren. Außerdem wurden die Anforderung für die Erweiterung der E-Mail Integration in diesem Punkt festgehalten.

Für die Umsetzung der Erweiterung mussten zuvor die Exchange Web Services näher betrachtet werden. Hierfür wurden die Grundlagen und die Funktionsweise von Microsoft Exchange kurz erläutert, um anschließend die Vorteile der zur Verfügung stehenden Web Services erklären zu können. Es stellte sich heraus, dass durch eine API der Web Services der Zugriff auf wichtige Elemente des Exchange Servers über Programmcode möglich war. Bevor jedoch eine direkte Implementierung in GEBman 10 vorgenommen wurde, musste eine Vorbetrachtung eine umfassende Modellierung vorgenommen werden. Dabei wurde auch auf Sicherheitsaspekte der Erweiterung eingegangen. Diese Modellierung im Punkt 6 erleichterte die anschließende Implementierung.

Die Umsetzung wurde nicht nur auf die reine Implementierung des Programmcodes beschränkt. Es wurde aufgezeigt, wie die Managed API von Microsoft in das System eingebunden und genutzt wurde. Testfälle wurden entwickelt, um die wichtigsten Funktionalitäten sicherzustellen. Außerdem wurde auf Probleme während der Implementierung eingegangen und Verbesserungsmöglichkeiten dargelegt.

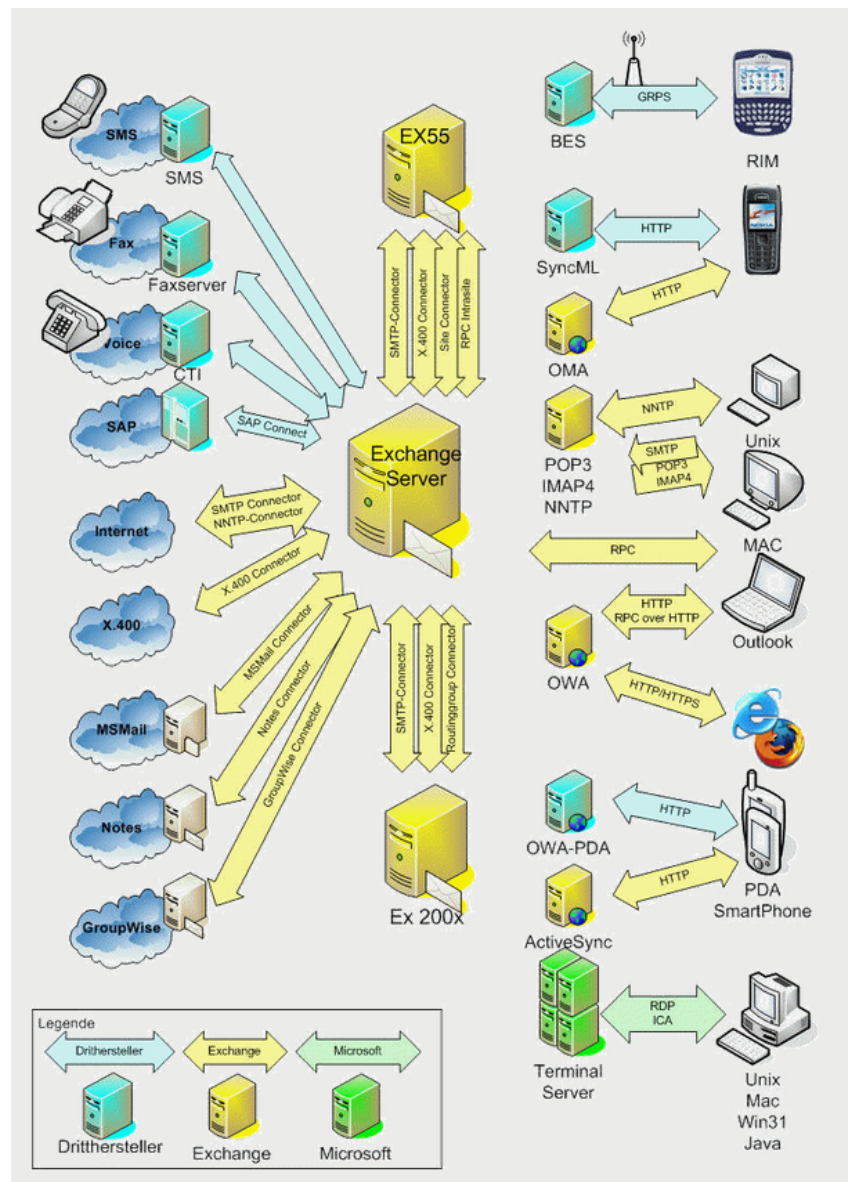
Diese Bachelorarbeit soll nun mit einem Ausblick auf die Verbesserungen des Service Desk-Moduls und auf den Einsatz der Erweiterung abgeschlossen werden. Das Service Desk-Modul von GEBman 10 könnte mit einigen Verbesserungen die aktuelle externe Softwarelösung SysAid ablösen und somit für die alltägliche Arbeit im Support eingesetzt werden. Dadurch könnten nicht nur Kosten reduziert werden, sondern es wäre auch möglich, das Service Desk-Modul

weiter an die Bedürfnisse des Supports anzupassen. Erst nach einiger Zeit der Nutzung können sich Aspekte für die Verfeinerung herausstellen, die in dieser Arbeit nicht betrachtet werden konnten.

Die E-Mail Integration könnte um die erwähnten Erweiterungsmöglichkeiten in Punkt 7.5 ergänzt und anschließend den Kunden präsentiert werden. Da es sich um eine prototypische Entwicklung handelt, ist darauf hinzuweisen, dass weitere Tests mit größeren Datenmengen ratsam wären. Es wurde mit dieser prototypischen Erweiterung dennoch der Grundstein für einen Einsatz als eigenständige Service Desk-Softwarelösung gelegt, die nicht mehr so stark an den Facility Management Bereich gebunden ist.

Anhangsverzeichnis

| | | |
|-------------------|--------------------------------------|----|
| Anhang 1. Blatt 1 | Exchange Verbindungen | 43 |
| Anhang 2. Blatt 1 | Geschichte der APIs | 44 |
| Anhang 3. Blatt 1 | Screenshot Freshdesk Dashboard | 45 |
| Anhang 3. Blatt 1 | Screenshot Desk.com Dashboard | 45 |
| Anhang 3. Blatt 2 | Screenshot Zendesk Dashboard | 46 |
| Anhang 3. Blatt 2 | Screenshot SysAid Dashboard | 46 |
| Anhang 4. Blatt 1 | Codeausschnitt GetMails() | 47 |



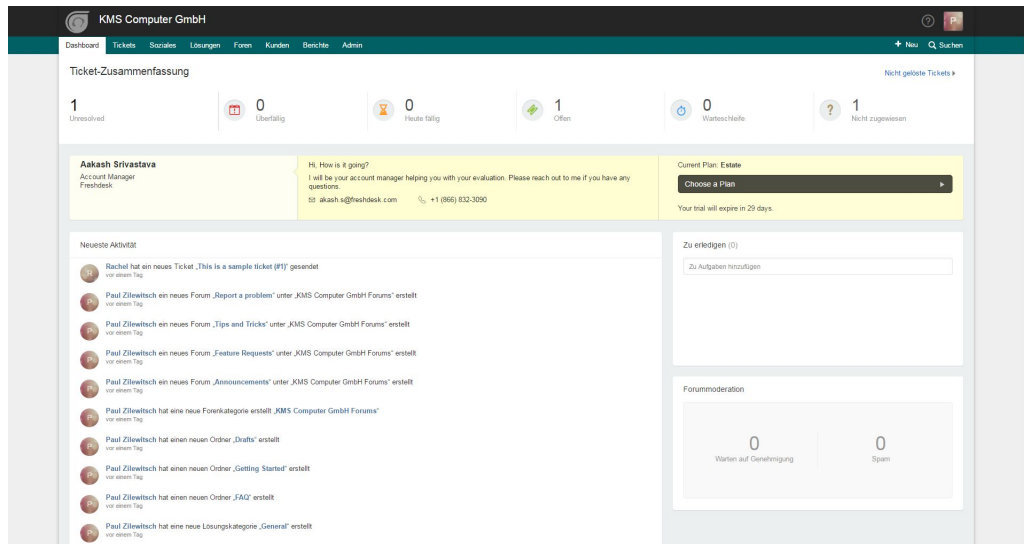
Exchange Verbindungen, Quelle: <http://www.msxfaq.de/basics/excomm.htm>

| Jahr | APIs Quelle | Beschreibung |
|------|---|--|
| 1992 | ■ SimpleMAPI 1.0 MS-Mail 3.0 | Diese Schnittstelle wurde mit Microsoft Mail 3.0 erstmals veröffentlicht. Das war der Startschuss von MAPI und es war sicher nicht abzusehen, dass es so ein Langläufer wird. Damals gab es noch andere MAPI-Schnittstellen wie VIM (Vendor Independent Messaging von Lotus Notes, und CMC Common Messaging C??? von Novell/WordPerfect) |
| 1995 | ■ Extended MAPI Windows95 | Durch Windows 95 wurde auch der neue "Exchange Client" verfügbar, welcher nun der Nachfolger von MS-Mail sein sollte. Dieser Client wurde auch mit Exchange Server genutzt, ehe Outlook 97 dann den Wechsel bedeutet (Nicht mit dem Exchange Server) zu verwechseln. |
| 1996 | ■ Directory API ■ Electronic Forms Designer ■ Exchange Developer Kit Gateway ■ OLE Messaging 1.0 (CDO 1.0) ■ Common Messaging Calls 1.0 Exchange 4.0 | Exchange 4.0 hat nun auch im Backend die alte Microsoft Mail Datenbank (MAILDATA) ersetzt und entsprechend sowohl für den Server als auch auf dem Client neue APIs mitgebracht |
| 1997 | ■ Active Messaging SDK 1.1 (CDO 1.1) Exchange 5.0 | |
| 1997 | ■ CDO 1.2 ■ CDO Rendering (CDOHTML) ■ Event Service Agents Exchange 5.5 | Exchange 5.5 hat Exchange weiter für das Internet geöffnet. Erstmals konnte per "OWA" auf das Postfach zugegriffen werden. Zudem gab es nun auch den "Event Agent", welcher auf dem Server eingestellte Ordner überwachte und Skripte beim Eintreffen neuer Nachrichten ausführt. |
| 1997 | ■ CDONTS Windows NT4 Option Pack | Durch das Windows NT4 Option pack war es nun mit dem darin enthaltenen IIS4 und dem SMTP-Server möglich, auch Mails mit Windows zu verarbeiten. CDONTS war dazu die erste API, um Nachrichten auf dem Server zu senden |
| 1998 | ■ CDO 1.2.1 ■ Routing Objects ■ HTML Form Converter Exchange 5.5 SP1 | |
| 2000 | ■ Antivirus API (VSAPI) Exchange 5.5 | In der Zeit hat sich Sybari den Grundstock für ihr Unternehmen gelegt, indem Sie einen Weg gefunden haben, Viren in Exchange zu finden und zu stoppen, ehe diese in der Datenbank landen. Erst durch die VSAPI, die Microsoft nachgereicht hat, konnten alle Hersteller dann über eine offizielle API gehen. |
| 2000 | ■ CDO für Exchange 2000 (CDOEx) ■ CDO für Exchange Management (CDOExM) ■ CDO für Workflow ■ Exchange Installable File System ■ Exchange OLEDB provider (ExOLEDB) ■ WSS Forms ■ FrontPage Extensions ■ OWA URL commands ■ Backup and Restore API (a.k.a. ESEdbcli2) ■ Queue Viewer API ■ Store Events ■ WebDAV ■ WMI classes für Exchange management ■ Workflow Designer Exchange 2000 | Das Jahr 2000 war auch der Start von Exchange 2000 und einer ganz neuen Plattform für Nachrichten und Zusammenarbeit. Entsprechend gibt es einen ganzen Satz neuer APIs die heute noch genutzt werden. Einige hingegen werden eher selten eingesetzt, z.B.: die Queue Viewer API |
| 2000 | ■ CDO für Windows 2000 (CDOSYS) ■ SMTP Transport Events Windows 2000 | Durch die Installation von Windows 2000 und dem darin enthaltenen SMTP-Server kommen ebenfalls einige APIs mit, die auch in Exchange vorhanden sind. |
| 2001 | ■ Antivirus API 2.0 (VSAPI 2.0) Exchange 2000 | |
| 2003 | ■ Antivirus API 2.0 (VSAPI 2.3) Exchange 2003 | Im Hinblick auf APIs hat Exchange 2003 nicht viel Neuerungen mitgebracht. Es ist ja auch eher eine kleine aber wichtige Weiterentwicklung von Exchange 2000. |
| 2007 | ■ AutoDiscover ■ EWS ■ OAB per HTTP | Mit Exchange 2007 wurden erstmals nicht nur Autodiscover eingeführt sondern neben der "alten" WebDAV-Schnittstelle auch ein richtiger Webservice. |
| 2015 | ■ RestAPI | Aktuell ist nur in Office 365 eine Vorabversion einer Rest-API auf die Inhalte eines Exchange Store verfügbar. |

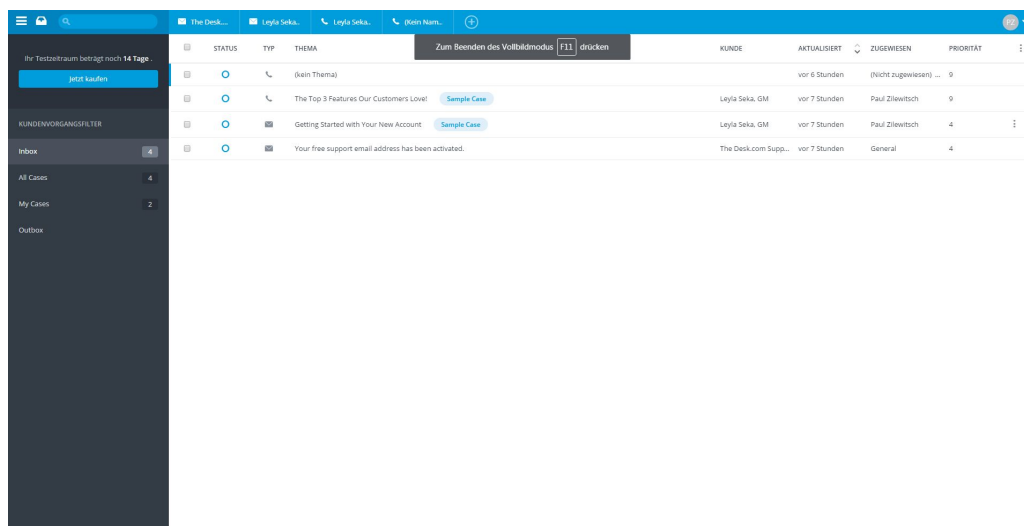
Geschichte der APIs, Quelle: <http://www.msxfaq.de/code/wege.htm>

Anhang 3

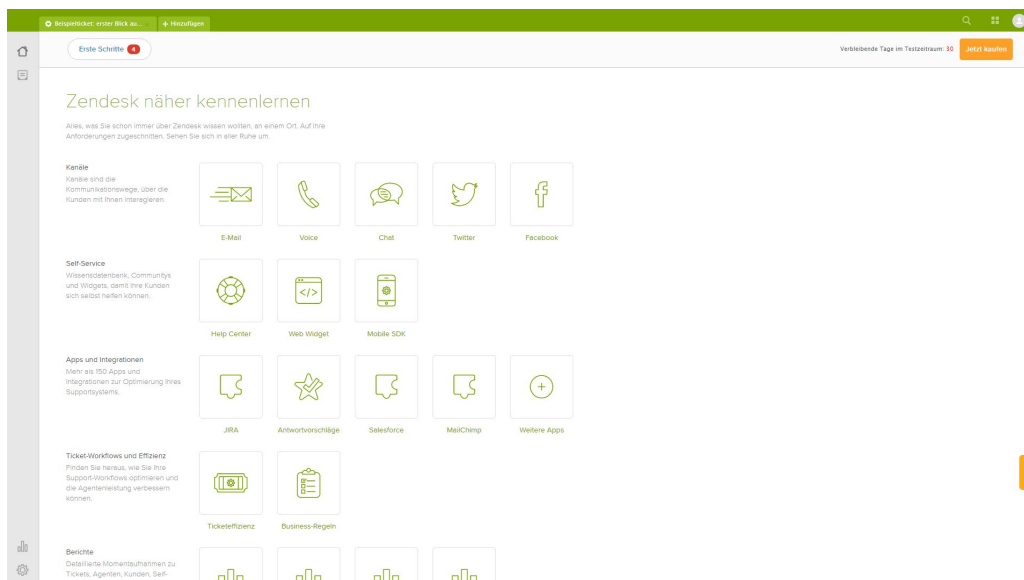
Blatt 1



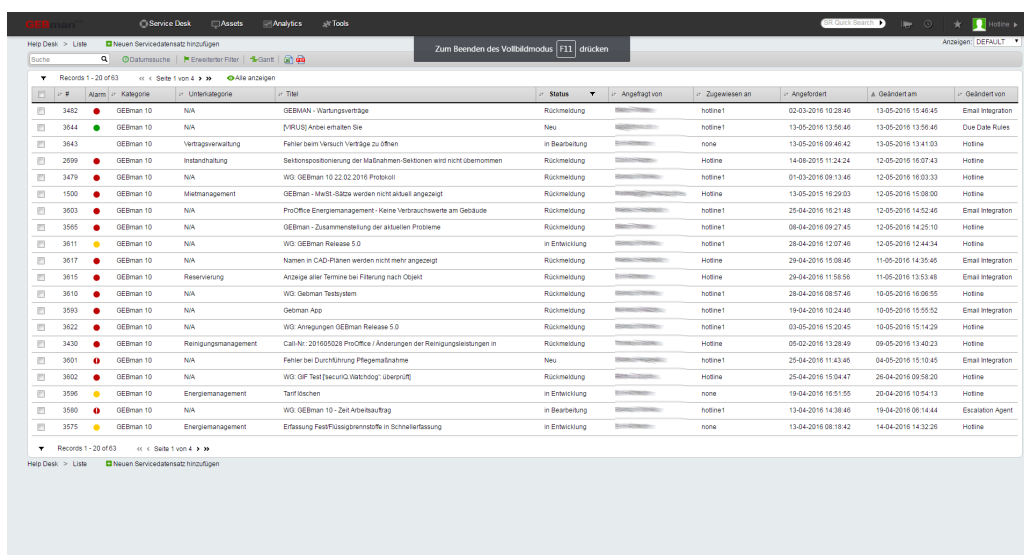
Screenshot Freshdesk Dashboard,
Quelle: <https://kmscomputergmbh.freshdesk.com/helpdesk/dashboard>



Screenshot Desk.com Dashboard,
Quelle: <https://kmscomputergmbh.desk.com/web/agent>



Screenshot Zendesk Dashboard,
Quelle: <https://kmscomputergmbh.zendesk.com/agent/discovery>



Screenshot SysAid Dashboard,
Quelle: <http://gebmanhelp.gebman.com:8080/HelpDesk.jsp?helpdeskfrm&fromId=List>

```
private FindItemsResults<Item> GetMails(DateTime date, ISessionTransaction session)
{
    //Server aus Datenbank in Liste speichern
    List<Server> serverList = Server.QueryExtent(session.Broker).ToList();
    var server = serverList.First();

    //ExchangeService erzeugen
    ExchangeService service = new ExchangeService((ExchangeVersion)server.Version);
    service.Credentials = new WebCredentials(server.UserName, Crypto.DecryptString(server.UserPwd), server.Domain);
    service.AutodiscoverUrl(server.UserEmail);

    FindItemsResults<Item> result = null;

    if (ObjectBroker.IsNullValue(date) || date == DateTime.MinValue)
    {
        //maximaler Wert des Integerwertebereiches für die ItemView
        result = service.FindItems(WellKnownFolderName.Inbox, new ItemView(int.MaxValue));

        //einige properties müssen geladen werden, da sonst kein Zugriff
        //auf https://msdn.microsoft.com/en-us/library/bb508824%28EXCHG.80%29.aspx steht, welche das genau sind
        var additionalProperties = new PropertyDefinitionBase[]
        {
            ItemSchema.DateTimeReceived,
            ItemSchema.TextBody,
            ItemSchema.Subject,
            ItemSchema.Attachments,
            ItemSchema.MimeContent,
            ItemSchema.HasAttachments,
            EmailMessageSchema.From,
            EmailMessageSchema.ToRecipients,
            EmailMessageSchema.MimeContent,
            EmailMessageSchema.Attachments
        };
        service.LoadPropertiesForItems(result, new PropertySet(additionalProperties));
    }
    else
    {
        return result;
    }
}
```

Codeausschnitt *GetMails*(), Quelle: eigene Darstellung

Literaturverzeichnis

- [1] Martin Andenmatten. Itil® 2011 edition – mehr als nur ein update. <http://blog.iti.org/2011/06/iti/iti%C2%AE-2011-edition-mehr-als-nur-ein-update>, 28.06.2011. abgerufen am: 10.05.2016.
- [2] H. Balzert. *UML 2 kompakt: mit Checklisten*. Spektrum Akademischer Verlag, 3 edition, 2010.
- [3] Ricardo Bilton. 10 powerful, cloud-based help-desk services to help you win at customer service. <http://venturebeat.com/2014/01/22/help-desk-software-heres-some-of-the-best-and-most-interesting/>, 22.01.2014. abgerufen am: 20.06.2016.
- [4] Buchsein, R. et al. *IT-Management mit ITIL® V3: Strategien, Kennzahlen, Umsetzung*. Vieweg+Teubner Verlag, 2 edition, 2007.
- [5] Cannon, D. / Wheeldon, D. *Service Operation*. The Stationery Office (TSO), 2007.
- [6] Frank Carius. <http://www.msxfaq.de/code/ews.htm>, . abgerufen am: 14.06.2016.
- [7] Frank Carius. <http://www.msxfaq.de/clients/oagrundlagen.htm>, . abgerufen am: 14.06.2016.
- [8] Frank Carius. <http://www.msxfaq.de/windows/rpc.htm>, . abgerufen am: 13.06.2016.
- [9] DIN EN ISO 9000. *DIN EN ISO 9000 (2005) : Qualitätsmanagementsysteme - Grundlagen und Begriffe*. Berlin, 2005.
- [10] N. Ebel. *ITIL V3 Basis-Zertifizierung: Grundlagenwissen und Zertifizierungsvorbereitung für die ITIL-Foundation-Prüfung*. Pearson Deutschland, 2008.
- [11] Eilebrecht, K. / Starke, G. *Patterns Kompakt - Entwurfsmuster für effektive Software-Entwicklung*. Springer-Verlag Berlin Heidelberg, 4 edition, 2013.
- [12] Ellis, C. A. / Gibbs, S. J. / Rein, G. L. *Groupware - Some Issues and Experiences. Communications of the ACM*. ACM New York, NY, USA, 1991.
- [13] M. Fowler. *Patterns für Enterprise-Application-Architekturen*. mitp-Verlag, 2003.
- [14] Freshdesk. <http://freshdesk.de/kunden/>. abgerufen am: 15.05.2016.
- [15] S. Gaßner. Studie: Webentwickler vernachlässigen die sicherheit. <http://www.zdnet.de/88123948/studie-webentwickler-vernachlassigen-sicherheit/>, 18.09.2012. abgerufen am: 23.06.2016.

- [16] Oliver Haslam. The best help desk software of 2016. <http://uk.pcmag.com/software/70330/guide/the-best-help-desk-software-of-2016>, 22.03.2016. abgerufen am: 20.06.2016.
- [17] T. Joos. *Microsoft Exchange Server 2013*. O'Reilly, Deutschland, 2014.
- [18] Markus Mattscheck. <http://www.onlinemarketing-praxis.de/glossar/widget>. abgerufen am: 22.05.2016.
- [19] Meier, A. / Myrach, T. *IT-Servicemanagement in deutschen Unternehmen - Ergebnisse einer empirischen Studie zu ITIL*. dpunkt.verlag.Heidelberg, 2004.
- [20] Microsoft. Microsoft exchange web services managed api 2.2. <https://www.microsoft.com/en-us/download/details.aspx?id=42951>, . abgerufen am: 21.06.2016.
- [21] Microsoft. Office 365-rest-apis für e-mail, kalender und kontakte. <https://msdn.microsoft.com/de-de/library/office/dn776319%28v=exchg.150%29.aspx>, . abgerufen am: 15.06.2016.
- [22] Microsoft. Exchange server 2010 web services sdk. <https://msdn.microsoft.com/en-us/library/office/dd877045%28v=exchg.140%29.aspx>, . abgerufen am: 15.06.2016.
- [23] Microsoft. try-catch (c-sharp-referenz). <https://msdn.microsoft.com/de-de/library/0yd65esw.aspx>. abgerufen am: 27.06.2016.
- [24] J. Nävy. *Facility Management: Grundlagen, Computerunterstützung, Systemeinführung, Anwendungsbeispiele*. Springer-Verlag Berlin Heidelberg, 4 edition, 2006.
- [25] A. Olbrich. *ITIL kompakt und verständlich: effizientes IT-Service-Management - den Standard für IT-Prozesse kennenlernen, verstehen und erfolgreich in der Praxis umsetzen*. Vieweg+Teubner Verlag, 4 edition, 2008.
- [26] W3C Recommendation. <https://www.w3.org/TR/widgets/#Introduction>, 27.12.2012. abgerufen am: 22.05.2016.
- [27] T. Redmond. *Microsoft Exchange Server 2010 - Das Handbuch*. O'Reilly, Deutschland, 2011.
- [28] Kim Rixecker. <http://t3n.de/news/zufriedene-kunden-geht-support-549532/>, 05.06.2014. abgerufen am: 19.05.2016.
- [29] M. Rohr. *Sicherheit von Webanwendungen in der Praxis: Wie sich Unternehmen schützen können – Hintergründe, Maßnahmen, Prüfverfahren und Prozesse*. Springer Fachmedien Wiesbaden, 2015.
- [30] Salesforce. <http://www.salesforce.com/de/company/newspress/press-releases/2012/02/120201.jsp>. abgerufen am: 16.05.2016.

- [31] Schill, A. / Springer, T. *Verteilte Systeme: Grundlagen und Basistechnologien*. Springer Berlin Heidelberg, 2012.
- [32] U. Schneider. *Taschenbuch der Informatik: mit 108 Tabellen*. Fachbuchverl. Leipzig im Carl-Hanser-Verlag, 7 edition, 2012.
- [33] Steinbaerg, R. A. / Orr, A. T. *ITIL Service Operation 2011*. The Stationery Office Ltd, 2011 ed. edition, 2011.
- [34] SysAid. <https://www.sysaid.com/de/customers>. abgerufen am: 17.05.2016.
- [35] James Thornton. 10 great online support and help desk apps. <http://web.appstorm.net/roundups/communication-roundups/10-online-support-and-help-desk-apps/>, 20.05.2014. abgerufen am: 20.06.2016.
- [36] Victor, F. / Günther, H. *Optimiertes IT-Management mit ITIL*. Springer Fachmedien Wiesbaden, 2 edition, 2005.
- [37] Vivenzio, A. / Vivenzio, D. *Testmanagement bei SAP-Projekten: Erfolgreich Planen , Steuern, Reporten bei der Einführung von SAP-Banking*. Springer Fachmedien Wiesbaden, 2012.
- [38] Volodarsky, M. et al. *Microsoft Internetinformationsdienste (IIS) 7.0 – Die technische Referenz*. Microsoft Press Deutschland, 2009.
- [39] Wikipedia. Servicedesk. <https://de.wikipedia.org/wiki/Servicedesk>, 09.05.2016. abgerufen am: 24.05.2016.
- [40] F. Witte. *Testmanagement und Softwaretest: Theoretische Grundlagen und praktische Umsetzung*. Springer Fachmedien Wiesbaden, 2015.
- [41] Zendesk. <https://www.zendesk.de/about/>. abgerufen am: 17.05.2016.
- [42] M. Ziegler. *Web Hacking: Sicherheitslücken in Webanwendungen – Lösungswege für Entwickler. Mit Playground im Internet*. Carl Hanser Verlag GmbH & Company KG, 2014.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne unerlaubte fremde Hilfe angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Stellen sind als solche kenntlich gemacht. Die Zustimmung des Partnerunternehmens in der Praxis zur Verwendung betrieblicher Unterlagen habe ich eingeholt. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Ort, Abgabetermin

Unterschrift des Verfassers

Dresden, 18.07.2016