

# Bachelorthesis

## Anbindung von „Microsoft SharePoint“ an „SAP Business One“

**Vorgelegt am:** 09.09.2013

**Von:** **Kästner, Marcel**  
Tauschaer Weg 13  
09322 Penig

**Studiengang:** Technische Informatik

**Studienrichtung:** Prozessinformatik

**Seminargruppe:** 1

**Matrikelnummer:** 4100053

**Praxispartner:** SIGMA Software und Consulting GmbH  
Am Erlenwald 13  
09128 Chemnitz

**Gutachter:** Herr Dipl.-Wirtschaftsinf. Jens Demut (SIGMA Software  
und Consulting GmbH)  
Herr Prof. Dr. rer. nat. Reinhardt Nindel (Staatliche  
Studienakademie Glauchau)

# Freigabeerklärung

Hiermit erklären wir uns einverstanden/nicht einverstanden\*), dass die Bachelorthesis des Studenten

**Name, Vorname: Kästner, Marcel SG: TI10/1**

zur öffentlichen Einsichtnahme durch den Dokumentenserver der Bibliothek der Staatlichen Studienakademie Glauchau bereitgestellt wird.

## **Thema der Arbeit:**

Anbindung von „Microsoft SharePoint“ an „SAP Business One“

.....  
Ort, Datum

.....  
Unterschrift Student

.....  
Stempel, Unterschrift Bildungsstätte

\*) Nichtzutreffendes bitte streichen



# Inhaltsverzeichnis

<b>Freigabeerklärung .....</b>	<b>II</b>
<b>Themenblatt Bachelorthesis.....</b>	<b>III</b>
<b>Inhaltsverzeichnis .....</b>	<b>IV</b>
<b>Abbildungsverzeichnis .....</b>	<b>VI</b>
<b>Tabellenverzeichnis .....</b>	<b>VII</b>
<b>Abkürzungsverzeichnis .....</b>	<b>VIII</b>
<b>1 Einleitung.....</b>	<b>1</b>
<b>2 Systemkomponenten.....</b>	<b>2</b>
2.1 Systemübersicht .....	2
2.2 SAP Business One .....	3
2.2.1 Übersicht.....	3
2.2.2 Funktionen .....	3
2.2.3 SAP Standard – Dokumente an Belegen.....	5
2.3 coresuite für SAP Business One.....	5
2.3.1 Übersicht.....	5
2.3.2 Funktionen .....	5
2.4 Dateisystem von Windows .....	8
2.5 SIGMA SharePoint Belegarchivierungsdienst.....	8
2.6 SharePoint .....	8
2.6.1 Übersicht.....	8
2.6.2 Funktionen .....	9
2.6.3 Vergleich „Microsoft SharePoint“/ Dokumenten Management System .....	10
<b>3 „Microsoft SharePoint“-Clientprogrammierung.....</b>	<b>12</b>
3.1 Einführung in das verwaltete Clientobjektmodell.....	12
3.2 .NET Framework.....	12
3.3 Funktionsweise .....	13
3.4 Erstellen einer „Microsoft SharePoint“-Clientanwendung.....	15
3.5 Erstellen einer Dokumentenbibliothek.....	16
3.6 Verwenden des Ausnahmebereiches.....	17
3.7 Hochladen von Dokumenten.....	18
3.8 Definieren von Eigenschaften .....	19
3.9 Authentifizierung .....	20
<b>4 Umsetzung.....</b>	<b>21</b>
4.1 Geplante Realisierung.....	21
4.1.1 Erweiterung des Add-ons „SIGMA Belegarchivierung“ .....	21
4.1.2 Auslösende Ereignisse.....	21

4.1.3	Sammeln der Daten .....	22
4.1.4	Einstellen des Dokumentes in „Microsoft SharePoint“ .....	24
4.1.5	Probleme bei der Umsetzung.....	24
4.2	Umgesetzte Realisierung.....	24
4.2.1	PDF-Generierung über coresuite .....	24
4.2.2	Dienst für das Hochladen in „Microsoft SharePoint“ .....	26
<b>5</b>	<b>Zusammenfassung und Ausblick.....</b>	<b>31</b>
	<b>Quellenverzeichnis.....</b>	<b>32</b>
	<b>Ehrenwörtliche Erklärung.....</b>	
	<b>Thesenpapier .....</b>	

## Abbildungsverzeichnis

Abbildung 1 Systemabbild „Microsoft SharePoint“-Belegarchivierung .....	2
Abbildung 2 Struktur coresuite country package .....	5
Abbildung 3 Basisinfrastruktur SharePoint .....	9
Abbildung 4 Intermediation-Konzept im .NET Framework .....	13
Abbildung 5 Das verwaltete Clientobjektmodell von „Microsoft SharePoint“ .....	14
Abbildung 6 Aufbau des Kontextes zur Website .....	15
Abbildung 7 Aufruf der ExecuteQuery-Methode .....	15
Abbildung 8 Erzeugen einer neuen Liste .....	16
Abbildung 9 Hinzufügen neuer Felder zu einer Liste .....	16
Abbildung 10 Aktivieren der Versionsverwaltung .....	17
Abbildung 11 Verwenden des Ausnahmebereiches .....	17
Abbildung 12 Hochladen eines Dokumentes .....	18
Abbildung 13 Laden eines Dokumentes .....	19
Abbildung 14 Hinzufügen von Metadaten .....	19
Abbildung 15 Caesar-Entschlüsselung .....	20
Abbildung 16 Durch die „SIGMA Belegarchivierung“ angelegte Aktivität .....	21
Abbildung 17 Druck, E-Mail, SMS und Fax Button im „SAP Business One“ .....	22
Abbildung 18 Ausgangsrechnung in „SAP Business One“ .....	22
Abbildung 19 Kundennummer aus einem Beleg abrufen .....	23
Abbildung 20 Datenbankabfrage über die „SAP DI API“ .....	23
Abbildung 21 PDF Button in „SAP Business One“ .....	24
Abbildung 22 Printdefinition coresuite .....	25
Abbildung 23 Generierung des Exportdateinamen .....	26
Abbildung 24 Start der Anwendung .....	27
Abbildung 25 Ersetzen der Hexadezimalen Zeichen durch Unicode Zeichen .....	28
Abbildung 26 Entfernen und Hinzufügen von Spalten in einer Ansicht .....	29
Abbildung 27 Aufruf zum Hochladen des Dokumentes .....	30

# Tabellenverzeichnis

Tabelle 1 Funktionen von „SAP Business One“ .....	4
Tabelle 2 Funktionen vom coresuite country package.....	8
Tabelle 3 Funktionen von „Microsoft SharePoint“ in den verschiedenen Versionen....	10
Tabelle 4 Clientseitige Klassen und ihre serverseitigen Entsprechungen .....	12

## Abkürzungsverzeichnis

API	Programmierschnittstelle (application programming interface)
ASCII	American Standard Code for Information Interchange
DI	Development Infrastructure
DMS	Dokumentenmanagement-System
ERP	Enterprise Resource Planning
HTTPS	Hypertext Transfer Protocol Secure
ID	Identifikationsnummer
JSON	JavaScript Object Notation
NTFS	New Technology File System
OCR	Optical Character Recognition
PDF	Portable Document Format
SAP	Systeme, Anwendungen, Produkte in der Datenverarbeitung
SQL	Structured Query Language
UDF	User Defined Field
UDT	User Defined Table
UI	User Interface
XML	Extensible Markup Language
ZIP	Dateiformat für komprimierte Dateien

# 1 Einleitung

In der Firma „SIGMA Software & Consulting GmbH“ wurde vor kurzem eine „Microsoft SharePoint“-Umgebung eingerichtet. Um das Interesse an „Microsoft SharePoint“ auch für Kunden der „SIGMA Software & Consulting GmbH“ zu wecken, wird eine Möglichkeit gesucht „Microsoft SharePoint“ in Verbindung mit „SAP Business One“ zu verwenden.

In dieser Thesis beschäftige ich mich mit der Anbindung der „Microsoft SharePoint Foundation 2010“ an „SAP Business One“. Das Ziel der Arbeit besteht darin, eine Möglichkeit zu schaffen, verschiedene Belege aus dem „SAP Business One“ auf einem Server mit „Microsoft SharePoint Foundation 2010“-Installation abzulegen. Die Schwerpunkte liegen zum einen in den theoretischen Grundlagen der einzelnen Systeme und zum anderen in der Programmierung eines Dienstes, um Dokumente in der „Microsoft SharePoint Foundation 2010“ abzulegen.

In unserer Firma ist der „Microsoft SharePoint Server 2010“ installiert. Dieser baut auf der „Microsoft SharePoint Foundation 2010“ auf. Da auf meinem „Microsoft SharePoint“-Testsystem nur die „Microsoft SharePoint Foundation 2010“ installiert ist, beziehe ich mich in der vorliegenden Arbeit ausschließlich auf diese. Sie enthält alle für die Thesis notwendigen Funktionalitäten.

## 2 Systemkomponenten

### 2.1 Systemübersicht

Für die Anbindung von „Microsoft SharePoint“ an „SAP Business One“ werden drei Systeme verwendet. Diese sind das „SAP Business One“, das Dateisystem von Windows sowie die „Microsoft SharePoint Foundation 2010“. Das „SAP Business One“ bietet dabei die Datengrundlage. Durch das Add-on „coresuite“ der Schweizer Firma „coresystems ag“ werden die Belege aus dem „SAP Business One“ im Dateisystem von Windows abgelegt. Diese werden anschließend über den Dienst „SIGMA SharePoint Belegarchivierung“ in der „Microsoft SharePoint Foundation 2010“ abgelegt. Der gesamte Prozess ist in der folgenden Abbildung 1 veranschaulicht.

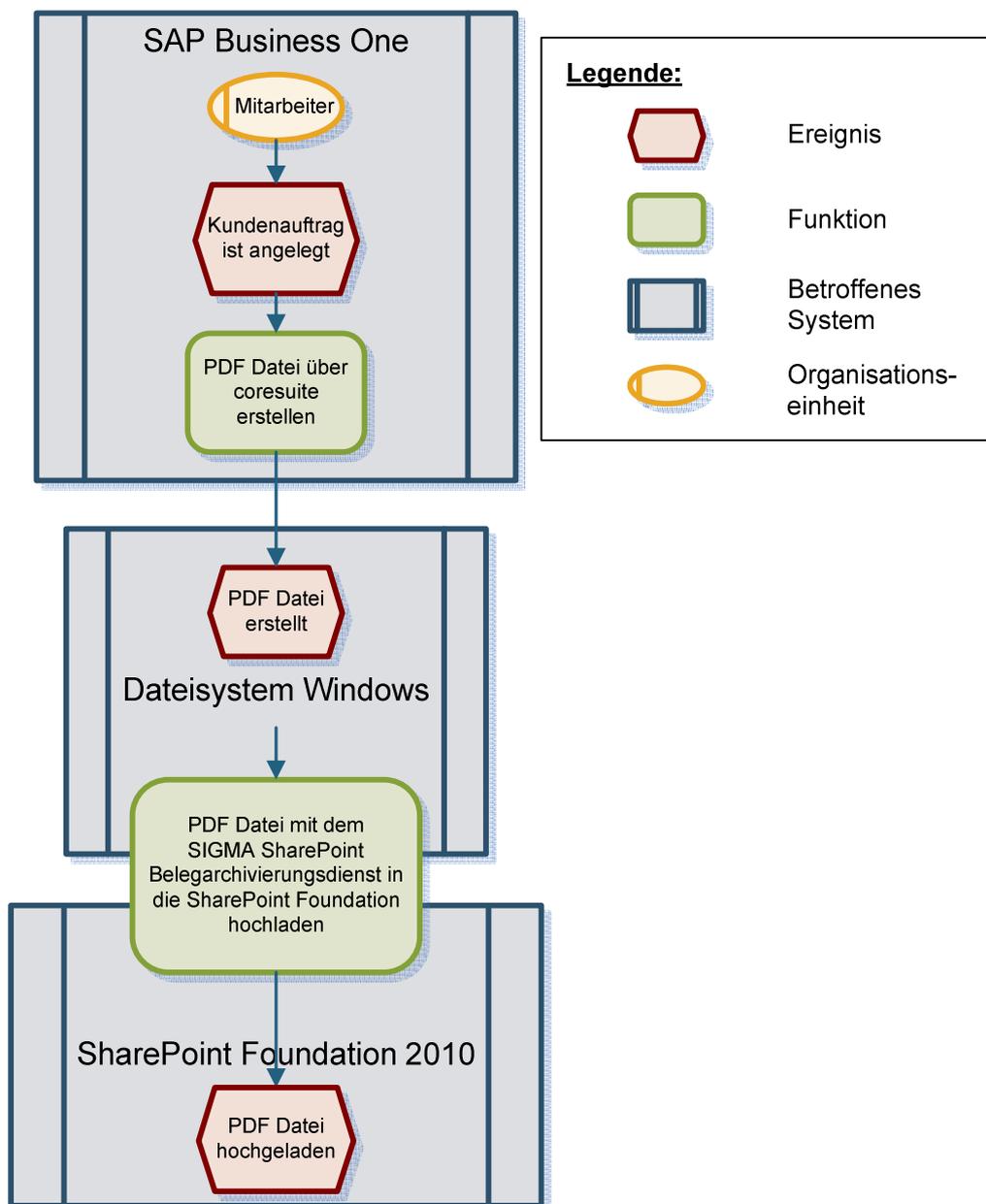


Abbildung 1 Systemabbild „Microsoft SharePoint“-Belegarchivierung

## **2.2 SAP Business One**

### **2.2.1 Übersicht**

Der Name SAP bedeutet ursprünglich „Systeme, Anwendungen, Produkte in der Datenverarbeitung“. Die „SAP AG“ wurde 1972 gegründet und ist heute einer der führenden Hersteller von ERP-Software. Der Sitz des Unternehmens befindet sich in Walldorf/Deutschland.<sup>1</sup>

ERP-Software (Enterprise Resource Planning) bedeutet übersetzt Planung der Unternehmensressourcen. Sie dient zur Optimierung von Geschäftsprozessen innerhalb eines Unternehmens. Hierzu bilden diese z. B. die Geschäftsprozesse im Bereich Materialwirtschaft, Finanz- und Rechnungswesen, Ein- und Verkauf sowie dem Marketing ab.<sup>2</sup> „Die Integration wird dabei von einer zentralen Datenbank unterstützt, wodurch Datenredundanz vermieden und integrierte Geschäftsprozesse ermöglicht werden.“<sup>3</sup>

„SAP Business One“ ist die Produktlösung für kleine und mittelständische Unternehmen mit 10-100 Mitarbeitern. Sie kommt in der Dienstleistungsbranche, dem Groß- und Einzelhandel sowie der Fertigungsindustrie zur Anwendung. Dabei setzt die „SAP AG“ auf eine duale Strategie. Zum einen wird das „SAP Business One“ in kleinen selbstständigen Unternehmen eingesetzt. Zum anderen bedient es auch Tochterunternehmen von multinationalen Konzernen. Diese Konzerne verwenden meist die „SAP Business Suite“, wodurch eine Kopplung der Systeme ermöglicht wird. Die Struktur von „SAP Business One“ zeichnet sich durch einen einfachen Aufbau aus. Dies ermöglicht ein schnelles Erlernen bzw. eine zügige Implementierung der Software im Rahmen der Standardfunktionen. Dadurch können die Kosten für die Installation und die Einrichtung des ERP-Systems niedrig gehalten werden. Die Anpassungsfähigkeit von „SAP Business One“ bildet eine der größten Stärken. So ist es möglich, alle denkbaren Geschäftsprozesse, welche nicht im Standard von „SAP Business One“ enthalten sind, mithilfe eines Add-ons hinzuzufügen. Daher kommt „SAP Business One“ in allen Branchen und Größenordnungen zum Einsatz. Die Anzahl der SAP-Arbeitsplätze kann dabei zwischen einem und mehr als hundert variieren.<sup>4</sup>

### **2.2.2 Funktionen**

Das „SAP Business One“ deckt einen breiten Funktionsumfang an Geschäftsprozessen ab. Alle wichtigen Bestandteile sind in der Tabelle 1 aufgeführt.

---

<sup>1</sup> vgl. MAYERHOFER, 2009, S. 25

<sup>2</sup> vgl. online: ERP-Software, 2013 (08.08.2013)

<sup>3</sup> MAYERHOFER, 2009, S. 25

<sup>4</sup> vgl. MAYERHOFER, 2009, S.28-29

Finanzwesen	Es besteht die Möglichkeit, Finanz- und Buchhaltungstransaktionen zu bearbeiten, Konten zu pflegen sowie Fremdwährungskorrekturen, Budgetierungen und Profitcenterrechnungen durchzuführen.
Bankenabwicklung	Man kann Ein- und Ausgangszahlungen, Kontoauszüge und Bankabstimmungen verwalten.
Verkauf/Einkauf	Ein weiterer wichtiger Bestandteil des „SAP Business One“ ist das Erstellen von Angeboten, Kundenaufträgen, Bestellungen und verschiedenen Rechnungen.
Kundenbeziehungsmanagement	Sämtliche Kunden- und Interessentenstammdaten können im „SAP Business One“ hinterlegt werden. Dazu zählen auch Adressen, Kontensalden und Analysen.
Lagerverwaltung	Die komplexe Lagerverwaltung bietet nicht nur die Möglichkeit, Lagerbestände von Artikeln zu verwalten, sondern auch Preislisten, Sonderpreisvereinbarungen, Serien- und Chargennummern.
Materialbedarfsplanung	Die einfache, aber leistungsstarke Planungsfunktion für Einkäufer und Produktionsplaner ermöglicht die Verwaltung der Beschaffung und Fertigung von Artikeln.
Berichtswesen	Viele verschiedene Standardberichte können über „SAP Business One“ abgerufen werden. Darunter zählen z. B. offene Rechnungen, Umsatzanalysen und Kundenkontaktübersichten. Weitere benutzerdefinierte Berichte können erstellt werden.
Servicemanagement	Das „SAP Business One“ bietet die Möglichkeit, verschiedene Servicetätigkeiten wie das Verwalten von Serviceverträgen oder eine Protokollierung von Kundeninteraktionen umzusetzen.
Personalverwaltung	Vorgesetzte können diverse Daten der Mitarbeiter einsehen und analysieren. Dazu gehören z. B. Mitarbeiterdetails oder Abwesenheitsberichte.
Ergänzende Branchenwendungen	Weitere Funktionalitäten, welche nicht im „SAP Business One“-Standard vorhanden sind, können durch Anpassungen in das System integriert werden.

**Tabelle 1** Funktionen von „SAP Business One“

(eigene Darstellung in Anlehnung an online: SAP Business One, 2008, S. 8 (08.08.2013))

### 2.2.3 SAP Standard – Dokumente an Belegen

Der „SAP Business One“-Standard ermöglicht es, PDF-Dateien aus Belegen zu generieren. Diese werden automatisiert an einem vorher definierten Speicherort abgelegt. Der Dateiname wird dabei folgendermaßen generiert:

<Dokumenttyp>\_<Datum>\_<Uhrzeit>.pdf.

Es besteht die Möglichkeit an den Ein- und Verkaufsbelegen, den Geschäftspartner Stammdaten, den Aktivitäten und an den Artikelstammdaten Dateianhänge zu hinterlegen.

## 2.3 coresuite für SAP Business One

### 2.3.1 Übersicht

Das „coresuite country package“ ist ein leistungsfähiges Add-on der Schweizer Firma „coresystems ag“. Es bietet Tools, um „SAP Business One“-Projekte schneller und einfacher umzusetzen. Die Besonderheit liegt darin, dass alle Module in einem Add-on zusammengefasst sind. Dies steigert die Performance des „SAP Business One“, da nur einmal Arbeitsspeicher reserviert wird. Alle Module, welche man selbst auf Basis des „coresuite framework“ erstellt, werden ebenfalls in dieses eine Add-on integriert. Die Abbildung 2 spiegelt die Struktur des „coresuite country package“ wider.<sup>5</sup>



**Abbildung 2** Struktur coresuite country package  
(online: coresystems, Präsentation coresuite country package, 2013, S. 8)

### 2.3.2 Funktionen

Das „coresuite country package“ enthält die Komponenten „coresuite framework“, „coresuite designer“, „coresuite enterprise search“, „coresuite accounting“, „coresuite widgets & dashboards“ und „coresuite customize“. Außerdem ist es möglich, eigene Erweiterungen auf Basis des „coresuite frameworks“ zu schreiben.<sup>6</sup>

<sup>5</sup> vgl. online: coresystems, 2013, S. 4 (09.08.2013)

<sup>6</sup> Ebenda, S. 6

Diese Funktionen sind in der Tabelle 2 näher erläutert.

„coresuite designer“	Der „coresuite designer“ ist ein leistungsfähiges Modul. Es ermöglicht die Erstellung von Drucklayouts. Die Besonderheit gegenüber den Standarddruckbelegen besteht darin, dass es ein Layout für alle Formen des „SAP Business One“, alle Belegtypen, alle Sprachen sowie für alle Länder gibt. Es gibt aber auch die Möglichkeit, für unterschiedliche Belege, andere Drucklayouts anzulegen oder auch mehrere Layouts für einen Beleg. So kann man z. B. zusammen mit einem Wareneingangsbeleg einen Barcode-Aufkleber drucken. Durch die Mehrsprachen-Unterstützung in allen Bereichen des „SAP Business One“ ist es möglich, einen Beleg automatisch in der richtigen Sprache zu drucken. <sup>7</sup>
„coresuite enterprise search“	Die „coresuite enterprise search“ bildet Google Funktionalitäten in „SAP Business One“ ab. Dies ermöglicht das Suchen in allen SAP Standardfeldern sowie den benutzerdefinierten Feldern. Es besteht die Möglichkeit, Tooltips für Detailinformationen der Suchergebnisse anzuzeigen. So kann man z. B. viele Daten eines Kundenauftrages sofort einsehen, ohne den Beleg öffnen zu müssen. <sup>8</sup>
„coresuite accounting“	„coresuite accounting“ ist ein weiteres Modul innerhalb des „coresuite country package“. Es enthält über dreißig vorgefertigte Finanzberichte. Diese dienen unter anderem der Gewinn- und Verlustrechnung, der Anzeige von Bilanzberichten und -vergleichen sowie der Generierung von Prognose-Berichten. Weiterhin bietet es eine Schnittstelle für die externe Buchhaltung. So können Excel-Dateien per Drag-and-Drop in die „SAP Business One“-Buchhaltung importiert werden. Die vorgefertigten Berichte ermöglichen einen einfachen und übersichtlichen Soll-Ist-Vergleich. <sup>9</sup>
„coresuite widgets & dashboards“	Mit „coresuite widgets & dashboards“ kann man Übersichtsdiagramme oder verschiedene Integrationen zu anderen Programmen, wie Twitter, YouTube oder Xing, direkt

<sup>7</sup> Ebenda, S. 13 ff.

<sup>8</sup> Ebenda, S. 8 f.

<sup>9</sup> Ebenda, S 20 ff.

	<p>im „SAP Business One“ anzeigen. Auch das Erstellen von eigenen Balken- oder Kuchendiagrammen ist möglich.<sup>10</sup></p>
„coresuite customize“	<p>Das „coresuite customize“-Modul bietet verschiedene Werkzeuge zur schnellen Anpassung von „SAP Business One“. Folgende Werkzeuge stehen dem Nutzer zur Verfügung: Ein Wizzard zum Aufzeichnen von Eventreihenfolgen. Ein Optimizer, um komplexe Regeln zu erstellen. Das Item Placement, um Systemfelder verschieben zu können. Das Function-Button-Werkzeug, um Rechtsklickfunktionen in das „SAP Business One“ einzubauen. New Items und New Menu, um eigene Felder und Menüpunkte zu erstellen. Easy Functions und Easy Row Functions, um Regeln ohne Programmierkenntnisse zu erstellen. Extended Code, um Globalfunktionen und Variablen zu erstellen. Ein Import-/Export-Werkzeug, um Anpassungen in andere Systeme importieren und exportieren zu können. Ein UDT-Handler, welcher das Erstellen von Eingabemasken ohne Programmierkenntnisse ermöglicht. Diese Werkzeuge ermöglichen es, unnötige Felder per Rechtsklick auszublenden, Pflichtfelder zu erstellen oder UDFs auf die Hauptform zu kopieren.<sup>11</sup></p>
„coresuite framework“	<p>Als Schnittstelle zwischen „SAP Business One“ und den „coresuite“-Modulen nutzt das „coresuite country package“ das „coresuite framework“. Dadurch wird eine Unabhängigkeit zur verwendeten „SAP Business One“-Version ermöglicht. Durch dieses Framework werden alle Module, welche auf Basis des „coresuite frameworks“ programmiert werden, zu einem Add-on zusammengefasst. Dies spart Arbeitsspeicher, da „SAP Business One“ für jedes Add-on ca. 100 MB Speicher reserviert. Außerdem können die einzelnen Module über gemeinsame Funktionen interagieren, was wiederum Programmieraufwand spart. Das „coresuite framework“ bietet einen Zugang zur „SAP DI“- und „SAP UI“-Komponente. Die „SAP DI“-Komponente (Development Infrastructure) ist die Schnittstelle zur Firmendatenbank von „SAP Business One“ und ermöglicht die Änderung von</p>

<sup>10</sup> Ebenda, S. 24 ff.

<sup>11</sup> Ebenda, S. 28 f.

	Daten. Die „SAP UI“-Komponente (User Interface) ist die Nutzerschnittstelle zur „SAP Business One“ Firmendatenbank. <sup>12</sup>
--	-----------------------------------------------------------------------------------------------------------------------------------

**Tabelle 2** Funktionen vom coresuite country package

## **2.4 Dateisystem von Windows**

Das Dateisystem von Windows dient bei der Archivierung von Belegen in der „Microsoft SharePoint Foundation 2010“ als Zwischenspeicher. In der Regel besitzen die Festplatten der Server die NTFS 3.1-Formatierung. Dieses Dateisystem wurde 2001 veröffentlicht und wird von Betriebssystemen ab Windows XP verwendet. Es zeichnet sich durch eine maximale Partitionsgröße von 256 TB und einer maximalen Dateigröße von 16 TB aus. Die maximale Länge des Dateinamens beträgt 255 Zeichen<sup>13</sup>. Da bei der Anbindung von „Microsoft SharePoint“ die Informationen zu den Belegen im Dateinamen abgespeichert werden, kann dieser eine Länge von über 150 Zeichen annehmen. Die maximale Zeichenanzahl reicht aber auf alle Fälle aus.

## **2.5 SIGMA SharePoint Belegarchivierungsdienst**

Mit Hilfe des „SIGMA SharePoint Belegarchivierungsdienstes“ können PDF-Dokumente in der „Microsoft SharePoint Foundation 2010“ abgelegt werden. Zu diesen Dateien werden zusätzlich diverse Metadaten hinterlegt. Bei der „SIGMA SharePoint Belegarchivierung“ wird der Belegtyp, die Belegnummer, die Kundennummer, der Name des Kunden, der Belegstatus, das Belegdatum, das Belegerstelldatum, der Nutzer, der den Beleg erstellt hat, das Aktualisierungsdatum des Beleges sowie der Nutzer, welcher den Beleg zuletzt aktualisiert hat, hinzugefügt. Eine detailliertere Beschreibung des Dienstes ist in 4.2.2 aufgeführt.

## **2.6 SharePoint**

### **2.6.1 Übersicht**

„Microsoft SharePoint“ ist ein Portal für die Zusammenarbeit innerhalb sowie außerhalb eines Unternehmens. In Firmen existieren oftmals viele Daten und Dokumente, welche verstreut in mehreren Speicherbereichen abgelegt sind. Informationen für Mitarbeiter, Kunden oder Lieferanten sind oft an verschiedenen Ablageorten gespeichert. Änderungen an diesen Daten müssen überall einzeln vorgenommen werden. Somit entstehen schnell inkonsistente Daten, wenn diese nicht an allen Stellen aktualisiert werden. Mit Hilfe von „Microsoft SharePoint“ werden diese Daten zentral an

<sup>12</sup> Ebenda, S. 34 f.

<sup>13</sup> vgl. online: NTFS und FAT: Alle Unterschiede im Überblick, 2011 (14.08.2013)

einem Ort gespeichert<sup>14</sup>. Durch eine intelligente Suche können die gewünschten Dokumente schnell gefunden und geändert werden. Die Abbildung 3 zeigt die Basisinfrastruktur von „Microsoft SharePoint“.



**Abbildung 3** Basisinfrastruktur SharePoint  
(online: POLJAKOV, 2012, S. 7 (13.08.2013))

Bei dieser Basisinfrastruktur wird ein Server mit dem Betriebssystem „Windows Server 2008 R2“ verwendet. Auf diesem Server wird der „Microsoft SQL Server 2008“ installiert. Dieser dient dem „Microsoft SharePoint“ als Datenspeicher. Das „Microsoft SharePoint“ an sich besteht aus der „Microsoft SharePoint Foundation 2010“ und dem „Microsoft SharePoint Server 2010“. Die „Microsoft SharePoint Foundation 2010“ ist ein kostenfreies Produkt von „Microsoft“ für die virtuelle Zusammenarbeit in Unternehmen<sup>15</sup>. Der „Microsoft SharePoint Server 2010“ basiert auf der „Microsoft SharePoint Foundation 2010“ und liefert verschiedene Dienstanwendungen und spezielle Funktionen. Ihn gibt es in einer Standard Edition und einer Enterprise Version. Letztere bietet erweiterte Funktionen zur Geschäftsdatenintegration sowie zur Microsoft Office Integration<sup>16</sup>. Dafür steht auch das Office Symbol in Abbildung 3.

## 2.6.2 Funktionen

„Microsoft SharePoint“ bietet die Möglichkeit, verschiedene Websites anzulegen. Somit kann eine klare Struktur innerhalb des Portals erzeugt werden. Auf diesen Webseiten können verschiedene Listen und Bibliotheken angelegt werden. Eine Liste kann z. B. eine Ansammlung von Hyperlinks, von Kontakten oder auch ein Kalender sein. Bibliotheken sind Listen, welche Dateien enthalten. Dabei kann eine Bibliothek

<sup>14</sup> vgl. online: POLJAKOV, 2012, S.5 f. (13.08.2013)

<sup>15</sup> vgl. online: Wikipedia, Microsoft SharePoint Foundation, 2013 (13.08.2013)

<sup>16</sup> vgl. online: Wikipedia, Microsoft SharePoint Server, 2013 (13.08.2013)

quasi jeden Dateityp enthalten, einschließlich Bilder, Audio- und Videodateien oder ZIP-Dateien. Verwaltet werden die Einträge einer Liste oder Bibliothek mithilfe von Metadaten<sup>17</sup>. Metadaten können verschiedene Typen annehmen. Diese können z. B. vom Typ Text, Nummern, Datum, Auswahl, Verweise oder Benutzer sein. Man kann einstellen, ob es sich bei einer Eigenschaft um ein Pflichtfeld oder einen optionalen Ausdruck handelt. Es besteht die Möglichkeit, eine Versionsverwaltung für Einträge einzustellen. So kann man ältere Datenbestände wiederherstellen, falls eine Änderung rückgängig gemacht werden soll<sup>18</sup>. Mithilfe der Tabelle 3 bekommt man einen Überblick über die Funktionalitäten der einzelnen „Microsoft SharePoint“-Versionen. Die Abkürzungen SPF, SPS-S und SPS-E stehen dabei für „SharePoint Foundation“, „SharePoint Server Standard“ und „SharePoint Server Enterprise“.

Bereich/ Funktion	SPF	SPS-S	SPS-E
Sites: Listen, Bibliotheken, Website-Vorlagen, Standard-Webparts, Workflows	X	X	X
Web Analytics, Secure Store Service		X	X
Communities: Wiki, Blogs, Diskussionsrunden	X	X	X
Meine Websites, Benutzerprofile, Unternehmenswiki		X	X
Content: Dokumentversionierung und -genehmigung	X	X	X
Verwaltete Metadaten, Dokumentmappen		X	X
Insights: Kennzahlen, Dashboards, Excel Services, Visio Services, PerformancePoint Services			X
Composites: Sandkastenlösungen	X	X	X
Workflowvorlagen		X	X
Access Services, Infopath Form Services, Geschäftsdatenintegration in den Office-Anwendungen			X

**Tabelle 3** Funktionen von „Microsoft SharePoint“ in den verschiedenen Versionen  
(eigene Darstellung in Anlehnung an online: POLJAKOV, 2012, S. 28 (13.08.2013))

### 2.6.3 Vergleich „Microsoft SharePoint“/ Dokumenten Management System

Bei der Betrachtung von „Microsoft SharePoint“ liegt es nahe, dieses als DMS-Lösung (Dokumentenmanagement System) zu nutzen. „Microsoft SharePoint“ besitzt aber nicht alle Funktionalitäten eines DMS. Im Folgenden sind einige Unterschiede zwischen einem klassischen DMS und „Microsoft SharePoint“ aufgeführt.

Das klassische DMS besitzt gegenüber „Microsoft SharePoint“ keine persönliche oder lokale Ablage von Dokumenten. Bei einem DMS erfolgt die Ablage der Daten geordnet. Dies ist für eine „Microsoft SharePoint“-Implementation untypisch. Das Anlegen von Blogs, Wikis, Diskussionsforen oder Teamkalendern ist ein Grundbaustein

<sup>17</sup> vgl. online: POLJAKOV, 2012, S.32 ff. (13.08.2013)

<sup>18</sup> vgl. online: ZARUBA, 2008, S.15 f. (14.08.2013)

von „Microsoft SharePoint“ für die Zusammenarbeit in Unternehmen. Dies ist in der Regel in einem klassischen DMS nicht möglich, aber auch nicht erforderlich. Automatische Erfassungskomponenten für Papiereingangspost, sprich das automatische Ablegen von gescannten Dokumenten, funktioniert im „Microsoft SharePoint“ nur durch den Einsatz von Anpassungen von Drittanbietern. Ein klassisches DMS-System besitzt so eine Komponente schon im Standard. Import- und Export-Werkzeuge für Dokumente und Metadaten sind in einem DMS vorhanden. In „Microsoft SharePoint“ sind diese Funktionalitäten nur spärlich ausgeprägt. Einen OCR-Filter sucht man im „Microsoft SharePoint“-Standard vergebens. Dieser kann wieder nur durch Drittanbieterkomponenten hinzugefügt werden. In vielen klassischen DMS wird er mit zur Verfügung gestellt. OCR-Filter dienen der Texterkennung aus Bildern. OCR steht dabei für Optical Character Recognition. Ein weiteres großes Manko an „Microsoft SharePoint“ als DMS ist die einfache Manipulation von Dokumenten. Um sicherzustellen, dass archivierte Dokumente nicht von Dritten geändert werden können, müssen die Websiteberechtigungen so eingestellt sein, dass die „Microsoft SharePoint“-Nutzer nur einen lesenden Zugriff auf abgelegte Dokumente erhalten. In einem klassischen DMS besteht die Möglichkeit, Dokumente direkt mit dem Hochladen zu archivieren. So wird eine unzulässige Manipulation ausgeschlossen. Die Einrichtung von Workflows ist zwar im „Microsoft SharePoint“ sowie auch auf klassischen DMS möglich, erfordert aber Programmierkenntnisse zur Einrichtung in „Microsoft SharePoint“.<sup>19</sup> „Ein Workflow ist das Modell eines durchzuführenden Prozesses, der aus einer Abfolge von Aktivitäten besteht.“<sup>20</sup>

Für die „SIGMA SharePoint Belegarchivierung“ bietet sich das „Microsoft SharePoint“ optimal an, da die Dokumente nicht dauerhaft archiviert werden müssen. Es wird ein „Microsoft SharePoint“-Belegarchivierungsnutzer angelegt. Dieser bekommt Schreibrechte auf die Belegarchivierungsbibliothek und wird nur im „Microsoft SharePoint“-Belegarchivierungsdienst verwendet. Andere Nutzer, welche die Dokumente einsehen können sollen, erhalten Leserechte auf die Bibliothek. Somit ist abgesichert, dass niemand die Dokumente unzulässig ändern oder löschen kann.

---

<sup>19</sup> online: RENTERGENT, 2010 (14.08.2013)

<sup>20</sup> online: POLJAKOV, 2012, S.38 (13.08.2013)

## 3 „Microsoft SharePoint“-Clientprogrammierung

### 3.1 Einführung in das verwaltete Clientobjektmodell

Mit dem verwalteten Clientobjektmodell der „Microsoft SharePoint Foundation 2010“ ist es möglich, auf „Microsoft SharePoint“-Inhalte zuzugreifen, ohne Programme auf dem Server mit „Microsoft SharePoint Foundation 2010“ installieren zu müssen. Anwendungen, welche das verwaltete Clientobjektmodell der „Microsoft SharePoint Foundation 2010“ nutzen, basieren auf .NET-verwalteten Codes und verwenden eine API, welche dem Objektmodell auf dem Server mit „Microsoft SharePoint Foundation 2010“ ähnelt. Dadurch kann man mithilfe des Clientobjektmodells auf Websitesammlungen, Websiteinformationen sowie Listen und Listenelementinformationen zugreifen. Eine API (application programming interface) ist eine Schnittstelle zur Anwendungsprogrammierung. Sie dient dem Zugriff auf Softwarebibliotheken, Datenbanken und Hardwarekomponenten. Die zwei Assemblys des verwalteten Clientobjektmodells enthalten fünf Namespace mit einer Vielzahl an Klassen, von denen viele nur innerhalb des Objektmodells benötigt werden. In der Tabelle 4 sind wichtige client- und serverseitige Äquivalente gegenübergestellt. Diese sind oftmals ausreichend für „Microsoft SharePoint“-Clientanwendungen.<sup>21</sup>

Client	Server
ClientContext	SPContext
Site	SPSite
Web	SPWeb
List	SPList
ListItem	SPListItem
Field	SPField

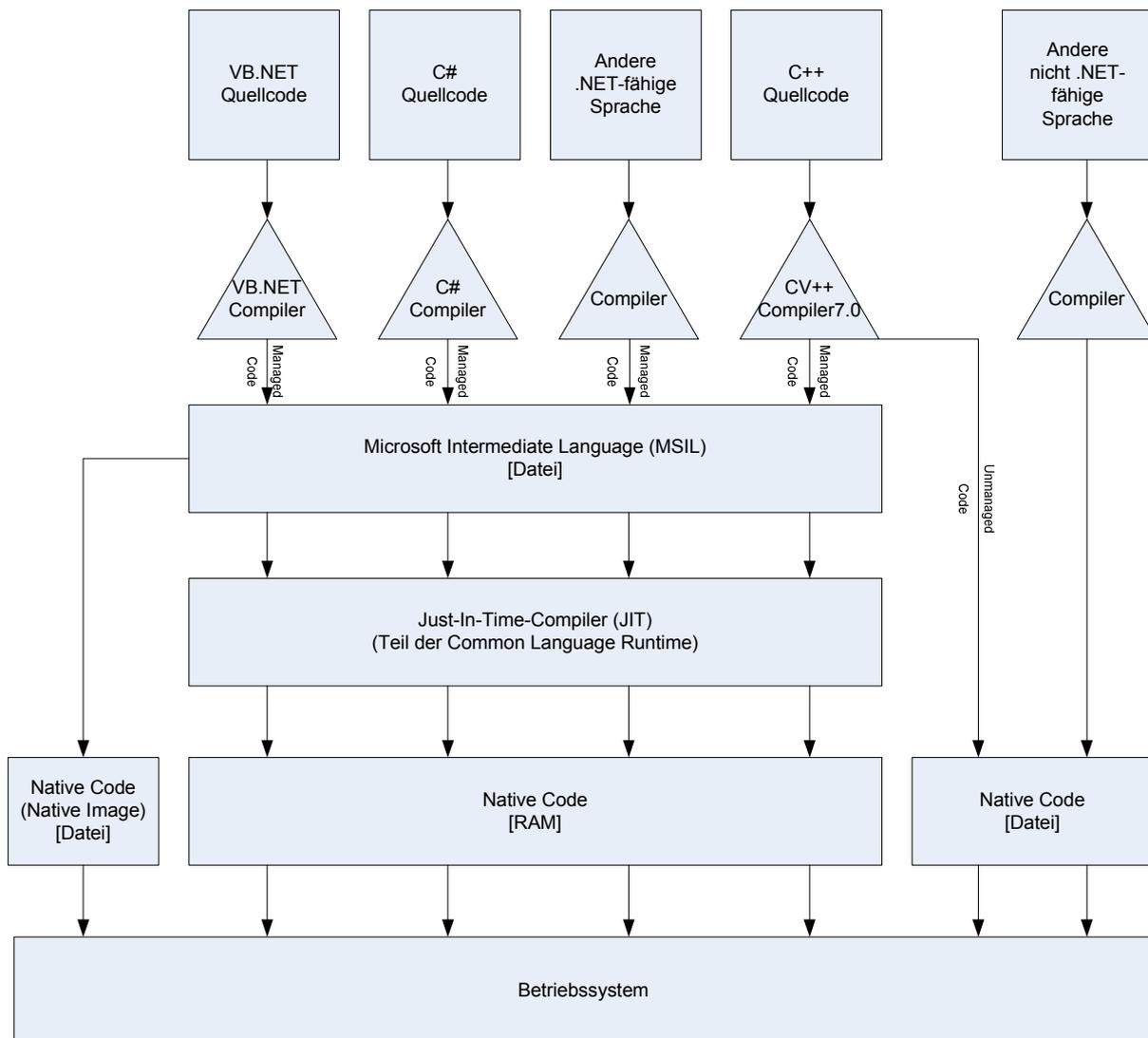
**Tabelle 4** Clientseitige Klassen und ihre serverseitigen Entsprechungen  
(eigene Darstellung in Anlehnung an online: WHITE, 2009)

### 3.2 .NET Framework

„Microsoft SharePoint“ baut grundlegend auf dem .NET Framework von „Microsoft“ auf. Das .NET Framework ist eine Programmierplattform, welche erstmals 2002 veröffentlicht wurde. Die aktuelle Version ist das .NET Framework 4.5. Diese ist 2012 erschienen. Entstanden ist es als Weiterentwicklung des Component Object Modell (COM). Bei der Kompilierung von .NET Quellcode wird eine plattformunabhängige Zwischensprache erzeugt und in einer sogenannten Assemblys abgespeichert. Dieser Zwischencode wird Microsoft Intermediate Language (MSIL) genannt. Zur Laufzeit wird der MSIL-Code durch einen Just-in-Time-Compiler in eine prozessorspezifische Maschinensprache umgewandelt und ausgeführt. Da der Code zur Ausführung Stück für Stück kompiliert und ausgeführt und nicht interpretiert wird, kann die Per-

<sup>21</sup> vgl. online: WHITE, 2009 (07.08.2013)

formance deutlich gesteigert werden. Der kompilierte Code, welcher vom System aufgeführt werden kann heißt Native Code. Die Abbildung 4 verdeutlicht das Konzept vom Microsoft .NET Framework.



**Abbildung 4** Intermediation-Konzept im .NET Framework  
(eigene Darstellung in Anlehnung an online: SCHWICHTENBERG o.J. (15.08.2013))

Um eine .NET-Anwendung ausführen zu können, benötigt man eine Laufzeitumgebung. Diese Common Language Runtime (CLR) stellt den Just-in-Time-Compiler sowie zahlreiche Basisdienste zur Verfügung. Ein solcher Dienst ist der Garbage-Collector, welcher sich um die Speicherverwaltung der Anwendung kümmert.<sup>22</sup>

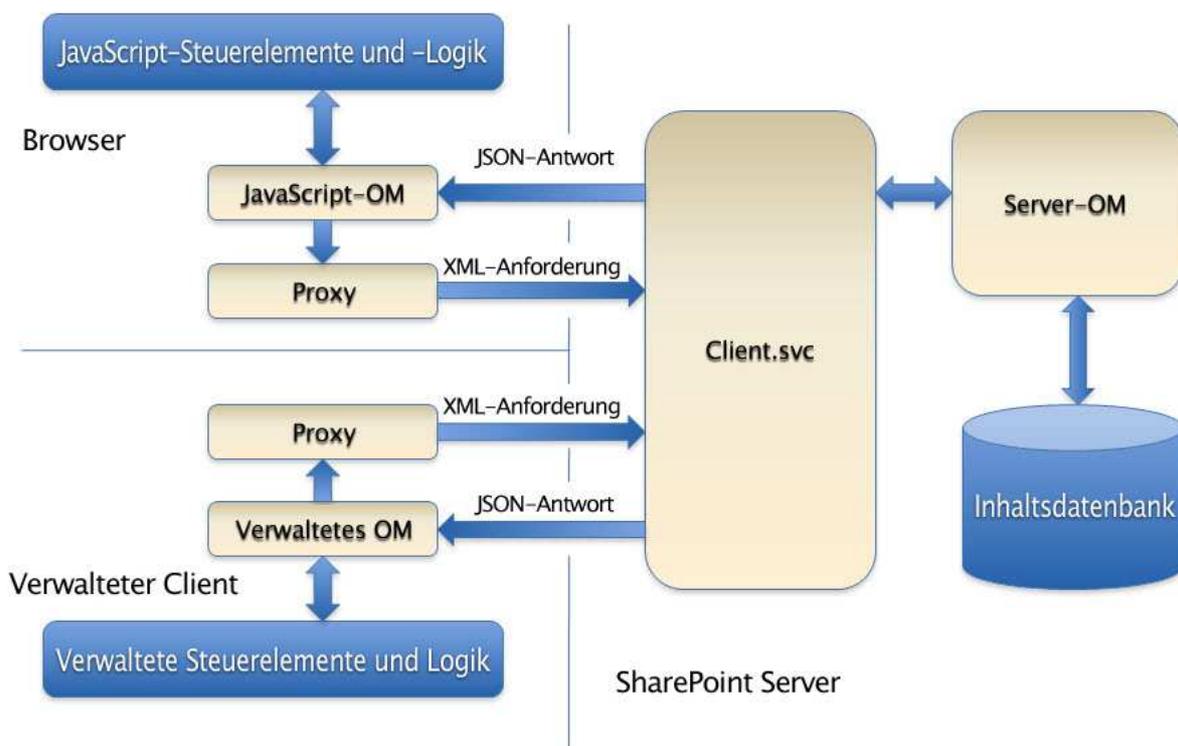
### 3.3 Funktionsweise

Eine Anwendung auf .NET Basis kann auf verschiedene Art und Weise mit der Client-API von „Microsoft SharePoint“ interagieren. Es können Methoden aufgerufen und deren Rückgabewerte abgefragt werden. Weiterhin ist es möglich, Collaborative Application Markup Language (CAML)-Abfragen zu übergeben und die Ergebnisse

<sup>22</sup> vgl. online: SCHWICHTENBERG o.J. (15.08.2013)

dieser abrufen. Eigenschaften von Elementen können ebenfalls abgefragt oder festgelegt werden. Alle Anfragen und Aufrufe an den Server mit der „Microsoft SharePoint Foundation 2010“ werden über das verwaltete Clientobjektmodell gebündelt und über eine XML-Struktur an den Server gesendet. Dieser führt die entsprechenden Aktionen aus, konvertiert das Ergebnis in die JavaScript Object Notation (JSON) und sendet den JSON-Code an das verwaltete Clientobjektmodell von „Microsoft SharePoint Foundation 2010“ zurück. Der Anwendung wird ein .NET Framework-Objekt übergeben, welches durch das Clientobjektmodell aus dem JSON-Code generiert wurde. Es ist möglich, eine „Microsoft SharePoint“-Client-Anwendung auch für einen Browser zu schreiben. In dem Fall erstellt das verwaltete Clientobjektmodell ein JavaScript-Objekt aus dem JSON-Code. Anhand der Abbildung 5 erkennt man den Ablauf einer solchen Anfrage an den Server mit der „Microsoft SharePoint Foundation 2010“.<sup>23</sup>

## Funktionsweise des Clientobjektmodells



**Abbildung 5** Das verwaltete Clientobjektmodell von „Microsoft SharePoint“  
(online: WHITE, 2009 (07.08.2013))

Durch die Bündelung der Methodenaufrufe an den Server wird eine bessere Netzwerkperformance erreicht. Würde jeder einzelne Methodenaufruf an den Server weitergeleitet werden, wäre das System durch den erhöhten Netzwerkverkehr stark be-

<sup>23</sup> vgl. online: WHITE, 2009 (07.08.2013)

einträchtig. Daher muss innerhalb der Programmierung explizit gesteuert werden, wann das verwaltete Clientobjektmodell eine Anforderung an den Server schickt. Außerdem kann vor dem Aufruf explizit festgelegt werden, welche Informationen benötigt werden. Somit schickt der Server z. B. nicht alle Informationen einer Liste an das verwaltete Clientobjektmodell zurück, sondern nur ein oder zwei gewünschte Felder.<sup>24</sup>

### **3.4 Erstellen einer „Microsoft SharePoint“-Clientanwendung**

Um eine Client-Anwendung für die „Microsoft SharePoint Foundation 2010“ zu programmieren, wird „Visual Studio 2008“ verwendet. Diese Software von „Microsoft“ bietet die Möglichkeit, Desktop- sowie Webanwendungen zu programmieren. Es werden unter anderem die Programmiersprachen Visual Basic, Visual C# und Visual C++ unterstützt. Beim „SIGMA SharePoint Belegarchivierungsdienst“ kommt die Programmiersprache C# zum Einsatz.

Nach dem Erstellen des „Visual Studio 2008“-Projektes müssen die zwei Assemblys „Microsoft.SharePoint.Client.dll“ und „Microsoft.SharePoint.Client.Runtime.dll“ für die Verwendung des verwalteten Clientobjektmodells in das Projekt eingebunden werden. Diese befinden sich auf dem Server mit der „Microsoft SharePoint Foundation 2010“-Installation.<sup>25</sup>

Um auf Inhalte des „Microsoft SharePoints“ zugreifen zu können, muss ein Kontext zur „Microsoft SharePoint“-Website aufgebaut werden. Dieser Kontext ist äquivalent zum Kontext des Serverobjektmodells von „Microsoft SharePoint Foundation 2010“. Der Aufbau der Verbindung wird in Abbildung 6 veranschaulicht.<sup>26</sup>

```
ClientContext clientContext = new ClientContext(Properties.Settings.Default.ClientContext);
```

**Abbildung 6** Aufbau des Kontextes zur Website

Der Aufruf der in Abbildung 7 abgebildeten Methode veranlasst das verwaltete Clientobjektmodell von „Microsoft SharePoint Foundation 2010“, die Daten an den Server zu senden. Bis zum Aufruf der ExecuteQuery-Methode entsteht kein Netzwerkverkehr. Somit wird die gewünschte Aktion erst nach dem Aufruf dieser ausgeführt.<sup>27</sup>

```
clientContext.ExecuteQuery();
```

**Abbildung 7** Aufruf der ExecuteQuery-Methode

---

<sup>24</sup> Ebenda.

<sup>25</sup> Ebenda.

<sup>26</sup> Ebenda.

<sup>27</sup> Ebenda.

### 3.5 Erstellen einer Dokumentenbibliothek

Beim Erstellen einer neuen Dokumentenbibliothek wird die Bibliothek zuerst vollständig definiert und anschließend über die ExecuteQuery-Methode im „Microsoft SharePoint“ angelegt. Zu Beginn wird ein ListCreationInfo-Objekt erstellt. Dieses Objekt erhält den Titel der neuen Liste und den TemplateType DocumentLibrary. Mithilfe dieser Informationen wird eine neue Dokumentenbibliothek angelegt. Abbildung 8 beinhaltet den dazugehörigen Quellcodeausschnitt.<sup>28</sup>

```
Web site = clientContext.Web;

// Erzeugen einer neuen Liste
ListCreationInformation listCreationInfo = new ListCreationInformation();
listCreationInfo.Title = Properties.Settings.Default.documentListName;
listCreationInfo.TemplateType = (int)ListTemplateType.DocumentLibrary;
List oNewList = site.Lists.Add(listCreationInfo);
```

**Abbildung 8** Erzeugen einer neuen Liste

Nachdem die Bibliothek definiert wurde, werden die einzelnen Felder hinzugefügt. Wie in Abbildung 9 visualisiert, können am Feld verschiedene Eigenschaften hinterlegt werden. In dem hier aufgeführten Beispiel handelt es sich um ein Auswahlfeld mit dem Titel Belegstatus und dem Format Dropdown. Man kann zwischen den Werten Offen, Geschlossen oder Storniert auswählen. Der folgende Parameter „true“ lässt die Spalte in der Standardansicht der Bibliothek anzeigen. Über die AddFieldOptions können verschiedene Eigenschaften für das Hinzufügen von Feldern eingestellt werden. Eine nähere Beschreibung dazu findet man in der MSDN Hilfe von „Microsoft“ unter <http://msdn.microsoft.com/de-de/library/microsoft.sharepoint.client.addfieldoptions.aspx> (16.08.2013).

```
Field oField5 = oNewList.Fields.AddFieldAsXml(@"<Field Type='Choice'
    DisplayName='Belegstatus'
    Format='Dropdown'>
    <CHOICES>
        <CHOICE>Offen</CHOICE>
        <CHOICE>Geschlossen</CHOICE>
        <CHOICE>Storniert</CHOICE>
    </CHOICES>
</Field>",
true, AddFieldOptions.DefaultValue);
```

**Abbildung 9** Hinzufügen neuer Felder zu einer Liste

Im Anschluss wird die Versionsverwaltung aktiviert und der Bibliothek der Befehl Update übergeben. Dadurch werden die Änderungen an der Liste übernommen. Dies wird durch die Quellcodeausschnitte in Abbildung 10 realisiert.<sup>29</sup>

---

<sup>28</sup> Ebenda.

<sup>29</sup> Ebenda.

```
oNewList.EnableVersioning = true;
oNewList.EnableMinorVersions = true;
oNewList.Update();
```

**Abbildung 10** Aktivieren der Versionsverwaltung

Erst nach dem darauf folgenden Aufruf der ExecuteQuery-Methode wird die vom verwalteten Clientobjektmodell von „Microsoft SharePoint Foundation 2010“ erzeugte XML-Struktur an den Server übergeben. Dieser führt dann die Befehle aus und legt die Bibliothek mit allen Feldern und Einstellungen an.<sup>30</sup>

### 3.6 Verwenden des Ausnahmebereiches

Die ExceptionHandlingScope-Klasse ermöglicht eine Ausnahmebehandlung für das verwaltete Clientobjektmodell von „Microsoft SharePont Foundation 2010“. So kann man z. B. überprüfen, ob eine Liste bereits vorhanden ist. Falls dies nicht der Fall ist, kann diese im Catch-Block erstellt werden. Wörtlich übersetzt bedeutet der Name ExceptionHandlingScope - Ausnahmebehandlungsbereich und definiert drei Bereiche. Der erste ist der Try-Bereich. Der Quellcode in diesem Bereich wird solange ausgeführt, bis ein Fehler auftritt oder alle Quellcodezeilen abgearbeitet sind. Falls bei der Abarbeitung Fehler auftreten, wird der Catch-Block aufgerufen. Er dient zur Behandlung des Fehlers. Beim dritten Bereich handelt es sich um den Finally-Block. Dieser ist optional und wird im Anschluss an den Try- oder Catch-Block ausgeführt.

Beim verwalteten Clientobjektmodell von „Microsoft SharePoint Foundation 2010“ wird ein ExceptionHandlingScope-Objekt erstellt. Dieses enthält die Try-, Catch- und Finally- Methoden.

```
ExceptionHandlingScope oScope = new ExceptionHandlingScope(clientContext);

using (oScope.StartScope())
{
    using (oScope.StartTry())
    {
        // Try-Quellcode
    }
    using (oScope.StartCatch())
    {
        // Catch-Quellcode
    }
    using (oScope.StartFinally())
    {
        // Finally-Quellcode
    }
}
```

**Abbildung 11** Verwenden des Ausnahmebereiches

---

<sup>30</sup> Ebenda.

Wie in Abbildung 11 ersichtlich, wird die using-Anweisung für die Behandlung des Ausnahmehereiches verwendet.<sup>31</sup> Das ExceptionHandlingScope-Objekt enthält eine Dispose-Methode. Diese dient zur Freigabe von nicht verwalteten Ressourcen, wenn ein Objekt nicht mehr benötigt wird. Durch die Kapselung des Quellcodes mithilfe der using-Anweisung wird diese Dispose-Methode nach Abarbeitung des Quellcodes ausgeführt. Sie wird auch ausgeführt, falls Ausnahmefehler auftreten.<sup>32</sup>

### 3.7 Hochladen von Dokumenten

Um Dokumente über das verwaltete Clientobjektmodell von „Microsoft SharePoint Foundation 2010“ hochzuladen, gibt es zwei Möglichkeiten. Die erste nutzt die Funktion SaveBinaryDirect des verwalteten Clientobjektmodells der „Microsoft SharePoint Foundation 2010“. Es werden folgende Variablen an die Funktion übergeben: der clientContext, der „Microsoft SharePoint“-Dateipfad, der Datenstrom der Datei und ein boolescher Wert, welcher aussagt, ob eine vorhandene Datei überschrieben oder beibehalten werden soll. Über diesen Aufruf kann ein Dokument direkt in „Microsoft SharePoint“ abgelegt werden. Bei der zweiten Möglichkeit wird das Objekt FileCreationInformation genutzt. Hier wird zuerst die Dokumentenbibliothek geladen. Anschließend werden dem FileCreationInformation-Objekt alle notwendigen Parameter zugewiesen. Diese sind der Dateiinhalt des Dokumentes, eine Eigenschaft, die ein Überschreiben eines vorhandenen Dokumentes erlaubt, sowie der Speicherort des Dokumentes im „Microsoft SharePoint“. Diese Daten werden einem Microsoft.SharePoint.Client.File-Objekt zugewiesen. Mithilfe der ExecuteQuery-Methode wird das Dokument im „Microsoft SharePoint“ abgelegt.<sup>33</sup> Die Abbildung 12 veranschaulicht diesen Prozess.

```
// Laden der Dokumentenbibliothek
List documentsList = clientContext.Web.Lists.GetByTitle(Properties.Settings.Default.documentListName);

// FileCreationInformations-Objekt erstellen
FileCreationInformation fileCreationInformation = new FileCreationInformation();

// Dateiinhalt in das Objekt kopieren
fileCreationInformation.Content = documentStream;

// Überschreiben eines vorhandenen Dokumentes erlauben
fileCreationInformation.Overwrite = true;

// Speicherort des Dokumentes festlegen
fileCreationInformation.Url = Properties.Settings.Default.ClientContext
    + Properties.Settings.Default.documentListURL + documentName;

// Objekt für das SharePoint Dokument erstellen
Microsoft.SharePoint.Client.File uploadFile = documentsList.RootFolder.Files.Add(fileCreationInformation);

// Abarbeitung starten/ Dokument in SharePoint hochladen
clientContext.ExecuteQuery();
```

**Abbildung 12** Hochladen eines Dokumentes

<sup>31</sup> vgl. online: Verwendung des Ausnahmehereichs, 2010 (07.08.2013)

<sup>32</sup> vgl. online: using-Anweisung, 2008 (19.08.2013)

<sup>33</sup> vgl. online: MAJEED, 2011 (19.08.2013)

### 3.8 Definieren von Eigenschaften

Das Hinzufügen von Metadaten zu einem Dokument unterscheidet sich danach, mit welcher Methode das Dokument in „Microsoft SharePoint“ eingestellt wurde. Bei der SaveBinaryDirect-Methode muss das Dokument aus der Bibliothek geladen werden. Dies wird durch die in Abbildung 13 abgebildeten Quellcodezeilen realisiert.<sup>34</sup>

```
// Laden des hochgeladenen Dokuments
Microsoft.SharePoint.Client.File newFile = clientContext.Web.GetFileByServerRelativeUrl(
    Properties.Settings.Default.documentListURL + documentName);
clientContext.Load(newFile);
clientContext.ExecuteQuery();
```

**Abbildung 13** Laden eines Dokumentes

Bevor dem Dokument Eigenschaften zugewiesen werden können, muss es über den Methodenaufruf newFile.CheckOut() im „Microsoft SharePoint“ abgemeldet werden. Mithilfe der folgenden Quellcodezeilen, in Abbildung 14, werden die Metadaten befüllt.<sup>35</sup>

```
// Setzen der Dokumenteneigenschaften
ListItem listItem = newFile.ListItemAllFields;
listItem[Properties.Settings.Default.Doctype] = documentType;
listItem[Properties.Settings.Default.Docnumber] = documentNumber;
listItem.Update();
```

**Abbildung 14** Hinzufügen von Metadaten

Nach dem Wiederanmelden im „Microsoft SharePoint“ mit newFile.CheckIn(string.Empty, CheckinType.OverwriteCheckIn) und dem Ausführen der Aktionen mithilfe der ExecuteQuery-Methode werden die Metadaten ins „Microsoft SharePoint“ übernommen. Anstelle des string.Empty Parameters, welcher eine leere Zeichenkette widerspiegelt, kann man einen Kommentar beim Wiederanmelden im „Microsoft SharePoint“ hinterlegen. Durch den CheckinType OverwriteCheckIn wird verhindert, dass die Versionsnummer des Dokumentes um eine Version angehoben wird.<sup>36</sup>

Beim Hochladen eines Dokuments über die FileCreationInformation können die Eigenschaften vor dem Ausführen der ExecuteQuery-Methode hinzugefügt werden. Dazu werden ebenfalls die Methoden CheckOut und CheckIn aufgerufen, so dass sich die Versionsnummer des Dokumentes beim Hochladen und Zuweisen der Eigenschaften nur um eine Nebenversion erhöht. Zwischen diesen beiden Methodenaufrufen werden die Eigenschaften zugewiesen. Dadurch, dass die ExecuteQuery-Methode bei dieser Variante des Hochladens eines Dokuments nur einmal aufgeru-

---

<sup>34</sup> vgl. online: Upload a document to a SharePoint list from Client Side Object Model, 2012 (07.08.2013)

<sup>35</sup> Ebenda.

<sup>36</sup> Ebenda.

fen wird, entsteht eine geringere Netzwerklast. Zum Vergleich: das Einstellen von Dokumenten über die SaveBinaryDirect-Methode verursacht an drei Stellen Netzwerkverkehr. Als erstes wird das Dokument hochgeladen. Als zweites wird dieses Dokument abgerufen und als drittes wird es im „Microsoft SharePoint“ aktualisiert.

### 3.9 Authentifizierung

Im verwalteten Clientobjektmodell von „Microsoft SharePoint Foundation 2010“ kann man Authentifizierungsdaten hinterlegen, um sich mit dem „Microsoft SharePoint“ zu verbinden. Falls man die Authentifizierung nicht verwendet, nutzt das verwaltete Clientobjektmodell die Windows-Anmeldeinformationen.<sup>37</sup> Da der „SIGMA SharePoint Belegarchivierungsdienst“ als Dienst auf einem Server installiert wird, ist dies nicht möglich. Um die Authentifizierungsmethode zu ändern, muss die Eigenschaft Credentials im ClientContext-Objekt zugewiesen werden. Diese Eigenschaft stellt ein NetworkCredential-Objekt dar. Bei der Erstellung des Objektes werden dem Konstruktor der Nutzername, das Kennwort und die Domain übergeben.<sup>38</sup>

Da die Nutzerdaten in der Konfigurationsdatei des Dienstes in Klartext abgespeichert sind, kann jeder, der Zugriff auf die Datei hat, diese auslesen. Um das zu verhindern, wird das Kennwort verschlüsselt abgespeichert. Dazu wird die Caesar-Verschiebung verwendet. Bei dieser Verschlüsselung wird eine ASCII-Code-Verschiebung um 5 Zeichen vorgenommen. Um das Kennwort, das dem NetworkCredential-Objekt im Klartext übergeben werden muss, zu entschlüsseln, wird die Funktion in Abbildung 15 verwendet.

```
// Caesar Entschlüsselung um 5 Stellen
private string decryptPassCode(string passCode)
{
    string passCodeDec = string.Empty;
    foreach (char c in passCode)
    {
        passCodeDec += (char)(c - 5);
    }
    return passCodeDec;
}
```

**Abbildung 15** Caesar-Entschlüsselung

Da der Kontoname und das Kennwort im ClientContext im Klartext übertragen werden, muss das „Microsoft SharePoint“ über das HTTPS-Protokoll aufgerufen werden.<sup>39</sup> Dieses sichere Hypertext-Übertragungsprotokoll ermöglicht es, die Daten verschlüsselt an den Server zu senden. Daher können die Daten während der Übertragung nicht von dritten abgegriffen, bzw. ausgelesen werden.

<sup>37</sup> vgl. online: Authentifizierung in den verwalteten Clientobjektmodellen, 2010 (07.08.2013)

<sup>38</sup> vgl. online: NetworkCredential-Klasse (21.08.2013)

<sup>39</sup> vgl. online: Authentifizierung in den verwalteten Clientobjektmodellen, 2010 (07.08.2013)

## 4 Umsetzung

### 4.1 Geplante Realisierung

#### 4.1.1 Erweiterung des Add-ons „SIGMA Belegarchivierung“

Ursprünglich war es geplant das Add-on „SIGMA Belegarchivierung“ für das „SAP Business One“ zu erweitern.

Die „SIGMA Belegarchivierung“ legt eine Aktivität im „SAP Business One“ an, wenn ein Druckauftrag gestartet oder ein Beleg per Mail oder Fax versendet wird. Dadurch kann man zurückverfolgen, wann ein Beleg gedruckt oder versandt wurde. Die Abbildung 16 veranschaulicht das Aussehen einer solchen Aktivität.

The screenshot shows a dialog box titled "Aktivität" with the following fields and values:

Aktivität	Sonstige	Nummer	1
Typ	Output	GP-Code	C20000
Thema	EMail	GP-Name	PC Welt GmbH & Co.KG
Zugeordnet zu	Benutzer Robert Leitner	Anspruchspartner	Christian Gander
Zugeordnet von	Robert Leitner	Telefon	30/56590-1

Additional fields and options:

- Persönlich
- Startzeit: 21.08.2013 14:12
- Endzeit: 21.08.2013 14:27
- Dauer: 15 Minuten
- Priorität: Normal
- Besprechungsort: [Empty]
- Erinnerung: 15 Minuten
- Inaktiv
- Abgeschl.
- Folgeaktivität (button)

**Abbildung 16** Durch die „SIGMA Belegarchivierung“ angelegte Aktivität

Bei der in Abbildung 16 gezeigten Aktivität erkennt man, dass es sich um einen Beleg handelt, welcher am 21.08.2013 um 14:12 Uhr per E-Mail durch Herrn Robert Leitner verschickt wurde. Im Reiter „Verknüpfter Beleg“ ist der dazugehörige Beleg hinterlegt.

Die „SIGMA Belegarchivierung“ soll dahingehend erweitert werden, dass beim Drucken oder Versenden von Belegen diese zusätzlich in der „Microsoft SharePoint Foundation 2010“ abgelegt werden.

#### 4.1.2 Auslösende Ereignisse

Um die Belege zu drucken oder zu versenden, drückt man im „SAP Business One“ auf einen der Buttons Druck, E-Mail oder Fax. Diese Buttons sind in Abbildung 17

dargestellt. Der transparente Button für die Versendung per SMS wird im „SAP Business One“-Standard nicht verwendet.



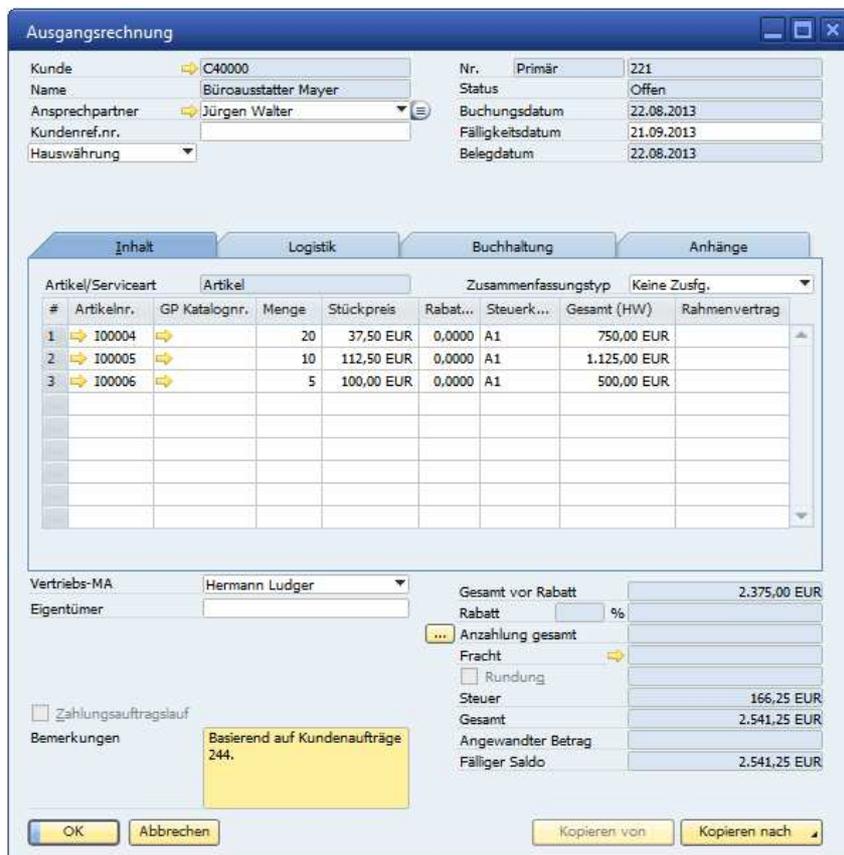
**Abbildung 17** Druck, E-Mail, SMS und Fax Button im „SAP Business One“

Jeder dieser Buttons besitzt eine eigene Menü-ID. Mithilfe dieser kann entschieden werden, ob der Beleg gedruckt oder per E-Mail oder Fax versendet werden soll.

#### 4.1.3 Sammeln der Daten

Nach der Betätigung eines Buttons werden verschiedene Metadaten des Beleges ausgelesen. Diese sind der Belegtyp, die Dokumentennummer, die Kundenreferenznummer, die Geschäftspartnernummer und dessen Name, eine Information über den Status des Belegs (Offen, Geschlossen, Storniert), das Belegdatum, das Datum, an dem der Beleg erstellt und zuletzt aktualisiert wurde, die SAP-Nutzer, die diesen Beleg erstellt bzw. zuletzt aktualisiert haben, und der Datenbankname der „SAP Business One“-Datenbank.

Alle Fenster für Belege im „SAP Business One“ sind identisch aufgebaut. Die Abbildung 18 zeigt dies am Beispiel einer Ausgangsrechnung.



**Abbildung 18** Ausgangsrechnung in „SAP Business One“

Jedes der hier abgebildeten Objekte hat eine interne Objekt-ID. So hat z. B. die Textbox mit der Kundennummer die ID 4, die Kundenreferenznummer die ID 14 und die Matrix mit den Rechnungspositionen die ID 38. Über diese Objekt-IDs kann man auf die einzelnen Objekte zugreifen, diese auslesen oder verändern. Dazu wird die API von „SAP Business One“ verwendet. Wie man z. B. die Kundennummer abfragt, ist in Abbildung 19 ersichtlich.

```
EditText oCardCodeEdit = m_oCurrentForm.Items.Item("4").Specific as EditText;
string sCardCode = oCardCodeEdit.Value;
```

**Abbildung 19** Kundennummer aus einem Beleg abrufen

Dieses Vorgehen ist mit allen Objekten auf der SAP-Form möglich. Für Informationen, welche nicht direkt auf der Form hinterlegt sind, wie z. B. das Datum, an dem der Beleg angelegt wurde, kann man über eine SQL-Abfrage direkt aus der Datenbank beziehen. Der dazu benötigte Quellcode ist in Abbildung 20 einzusehen.

```
SAPbobsCOM.Recordset oRecordset = null;
string strCreateTime = string.Empty;

try
{
    oRecordset = BiConnection.Instance.getRecordset();

    oRecordset.DoQuery(string.Format("SELECT T0.CreateDate FROM [OINV] AS T0 WHERE T0.DocNum = '{0}'", strDocNum));

    if (oRecordset.RecordCount > 0)
    {
        var oFields = oRecordset.Fields;
        var oItem = oFields.Item(0);
        var value = oItem.Value;
        strCreateTime = value.ToString();
        System.Runtime.InteropServices.Marshal.ReleaseComObject(oItem);
        System.Runtime.InteropServices.Marshal.ReleaseComObject(oFields);
    }
}
catch (Exception oEx)
{
    BiConnection.Instance.Application.StatusBar.SetText(
        string.Format("Fehler beim Ausführen von SELECT... Fehlernachricht: {0}", oEx.Message),
        SAPbouiCOM.BoMessageTime.bmt_Medium,
        SAPbouiCOM.BoStatusBarMessageType.smt_Error);
}
finally
{
    if (oRecordset != null) System.Runtime.InteropServices.Marshal.ReleaseComObject(oRecordset);
    oRecordset = null;
}
```

**Abbildung 20** Datenbankabfrage über die „SAP DI API“

Bei SQL-Abfragen über die „DI API“ von „SAP Business One“ wird ein Recordset erstellt. Ein Recordset ist eine Ansammlung von Datensätzen aus der Ergebnisrelation einer Datenbankabfrage. Die Variable oFields enthält den aktuellen, in diesem Fall ersten, Datensatz. In der Variable oItem befindet sich der Wert, welcher über oItem.Value abgerufen werden kann. Es ist zwar auch möglich, den Wert direkt über oRecordset.Fields.Item(0).Value.ToString() abzurufen. Dies ist aber speichertechnisch ungünstig, da die DI API den allokierten Speicher nicht automatisch wieder freigibt. Die Speicherfreigabe erfolgt daher über die Funktion System.Run-

time.InteropServices.Marshal.ReleaseComObject()). Um Fehler bei der Abfrage abzufangen, wird diese in einen Try-Catch-Block gesetzt.

#### 4.1.4 Einstellen des Dokumentes in „Microsoft SharePoint“

Um das Dokument auf den Server mit der „Microsoft SharePoint Foundation 2010“ hochzuladen, wird das verwaltete Clientobjektmodell, wie in Punkt 3.7 beschrieben, verwendet. Anschließend werden die gesammelten Metadaten am Beleg angefügt. (siehe Punkt 3.8)

#### 4.1.5 Probleme bei der Umsetzung

Während meiner Recherche zur vorliegenden Thematik hat sich herausgestellt, dass die geplante Umsetzung nicht auf diese Art realisiert werden kann. Es ist nicht möglich, beim Betätigen der Buttons Druck, E-Mail oder Fax auf das Dokument zuzugreifen, welches gedruckt oder versendet wird. Man erhält zwar die Metadaten aus dem geöffneten Beleg, das Dokument an sich ist aber nicht abrufbar.

## 4.2 Umgesetzte Realisierung

Da eine direkte Umsetzung über ein Add-on für „SAP Business One“ nicht möglich ist, habe ich einen Dienst entwickelt, welcher PDF-Dateien in der „Microsoft SharePoint Foundation 2010“ ablegen kann. Über coresuite ist es möglich, Belege als PDF Datei zu exportieren. Dies ist entweder automatisch direkt nach dem Anlegen des Beleges möglich oder beim Betätigen des PDF-Buttons im „SAP Business One“ (siehe Abbildung 21)



Abbildung 21 PDF Button in „SAP Business One“

Die aus dem coresuite heraus generierte PDF-Datei enthält alle notwendigen Metadaten für das „Microsoft SharePoint“ im Dateinamen. Sie werden durch den „SIGMA SharePoint Belegarchivierungsdienst“ auf den Server mit der „Microsoft SharePoint Foundation 2010“ hochgeladen.

#### 4.2.1 PDF-Generierung über coresuite

Für den Druck von Belegen bietet das coresuite zwei Definitionen. Die erste ist die Layoutdefinition. Mithilfe dieser können alle Einstellungen für das Layout des Beleges vorgenommen werden. Für die Funktionalität des „SIGMA SharePoint Belegarchivierungsdienstes“ können die Layoutdefinitionen der Belege im Standard belassen werden. Die zweite ist die Printdefinition. Sie verbindet eine oder mehrere Layoutdefinitionen mit dem Formtyp des Belegs.

Die notwendigen Einstellungen der Printdefinition werden im Folgenden anhand der Definition für die Kundenaufträge dargestellt. (siehe Abbildung 22)

The screenshot shows the 'Printdefinition' dialog box with the following details:

- Name:** Sales Order(DOC)
- Code:** T0000010
- PDF nach Speichern:** PDF (selected)
- Checkboxes:**  Druck,  Email,  PDF,  MS Word,  MS Excel
- Formtyp:** 139:Sales Order
- Sprache:** FromDocument
- Taste:** Keine Taste
- Reiter:** Dokumente (selected), Erweiterte Definition, Email/Fax, BP
- Status:** Aktiv
- Table:**

#	Layout	Kopien	Kopien M.	Freitext	Export Dateiname	Sprache
1	DOC:Document DIN - LD (DOC00008)	1	1		SELECT [:@T000000:	
- Buttons:** OK, Abbrechen, Funktionen, Designer öffnen

**Abbildung 22** Printdefinition coresuite

Im Kopf der Printdefinition kann man erkennen, dass es sich um die Printdefinition für Kundenaufträge (engl. Sales Order) handelt. Mithilfe der CheckBox PDF am rechten Rand der Form wird eingestellt, ob die Printdefinition beim Erstellen der PDF-Datei verwendet werden soll. Über die ComboBox links daneben stellt man „PDF nach Speichern“ ein. Dadurch wird automatisch nach der Anlage eines Beleges eine PDF-Datei erzeugt und im vorher definierten Ordner abgelegt.

Der Ordner wird im Reiter Email/Fax im Feld Exportpfad angegeben. Falls dieses Feld nicht gefüllt wird, wird die Datei im Ordner %temp%\swald\\exports abgelegt.

Im Reiter Dokumente ist die Layoutdefinition des Beleges hinterlegt. Für die Erstellung der PDF-Datei wird das hier hinterlegte Layout verwendet. In der Spalte „Export Dateiname“ kann man den Namen oder eine SQL-Abfrage, welche den Dateinamen generiert, angeben. Dieser wird dem Dokument zugewiesen. Da die Metadaten des Beleges über den Dateinamen übergeben werden, wird hier die in Abbildung 23 abgebildete SQL-Abfrage ausgeführt.

```

SELECT '[' + @T0000010 + ';' +
CAST(T1.DocNum AS NVARCHAR) + ';' +
CAST(CASE WHEN T1.NumAtCard IS NULL THEN '' ELSE T1.NumAtCard END AS NVARCHAR) + ';' +
CAST(T1.CardCode AS NVARCHAR) + ';' +
CAST(T1.CardName AS NVARCHAR) + ';' +
CAST(T1.CANCELED AS NVARCHAR) + ';' +
CAST(T1.DocStatus AS NVARCHAR) + ';' +
CAST(CONVERT(nvarchar(30),T1.TaxDate,120) AS NVARCHAR) + ';' +
CAST(CONVERT(nvarchar(30),T1.CreateDate,120) AS NVARCHAR) + ';' +
CAST((SELECT T2.USER_CODE FROM OUSR T2 WHERE T1.UserSign = T2.USERID) AS NVARCHAR) + ';' +
CAST(CONVERT(nvarchar(30),T1.UpdateDate,120) AS NVARCHAR) + ';' +
CAST((SELECT T2.USER_CODE FROM OUSR T2 WHERE T1.UserSign2 = T2.USERID) AS NVARCHAR) + ';' +
CAST(DB_NAME() AS NVARCHAR)
FROM ORDR T1
WHERE T1.DocEntry = [%DocEntry]

```

**Abbildung 23** Generierung des Exportdateinamen

Der erste String [@T0000010] ist ein Platzhalter. In coresuite gibt es verschiedene Platzhalter, um im Beleg verschiedene Sprachen auszudrucken. Je nachdem, welche Sprache im „SAP Business One“ ausgewählt ist, wird der zu diesem Platzhalter gehörende String eingefügt. Der Platzhalter [@T0000010] steht für Kundenauftrag. Als zweites wird die Belegnummer angegeben. Die Kundenreferenznummer (NumAtCard) ist das einzige Feld, welches leer sein kann. Deswegen wird ein leerer String zurückgegeben, wenn der Inhalt des Feldes NULL ist. Eine weitere Besonderheit weisen die Datumsfelder auf. Diese müssen erst in das Format „JJJJ-MM-TT HH\_MM\_SS“ umgewandelt werden. Als letzte Angabe des Selects wird der Datenbankname übergeben. Ein weiterer Platzhalter ist die Variable [%DocEntry] in der Whereklausel der SQL-Abfrage. Er wird in coresuite durch die interne Nummer des Beleges ersetzt. Auf diese Weise wird der String für den Dateinamen der PDF-Datei zusammengesetzt. So entsteht folgender Aufbau: <Belegtyp>; <Belegnummer>; <Kunden-/Lieferantenreferenznummer>; <Geschäftspartnernummer>; <Geschäftspartnername>; <Storniert>; <Status des Beleges>; <Belegdatum>; <Erstelldatum>; <Erstellt durch>; <Aktualisierungsdatum>; <Aktualisiert von>; <Datenbankname>.

In dem Reiter Erweiterte Definition muss der Haken von „Öffne Word/Excel/PDF“ entfernt werden, da sonst jedes Mal, wenn ein Beleg archiviert werden soll, die PDF-Datei geöffnet wird.

Auf diese Weise ist es nun möglich, die Belege als PDF-Dateien im Dateisystem von Windows abzulegen. Sie enthalten alle für das „Microsoft SharePoint“ notwendigen Metadaten.

#### 4.2.2 Dienst für das Hochladen in „Microsoft SharePoint“

Der „SIGMA Belegarchivierungsdienst“ ist in der Lage, PDF-Dateien in die „Microsoft SharePoint Foundation 2010“ hochzuladen. Um Metadaten im „Microsoft SharePoint“ am Dokument hinterlegen zu können, müssen diese im Dateinamen enthalten sein.

Das Programm ist mit „Microsoft Visual Studio 2008“ entwickelt. Als Programmiersprache wurde C# verwendet. Um verschiedene Einstellungen für den Dienst vornehmen zu können, wurden über die Eigenschaften des Projektes Anwendungseinstellungen hinzugefügt. Diese befinden sich in der Datei app.config und können auch nach der Installation des Dienstes geändert werden. Im Quellcode der Anwendung kann man über Properties.Settings.Default.<Einstellungsname> auf die Einstellungen zugreifen.

Beim Start der Anwendung wird die Klasse „main“ aufgerufen. Diese Klasse enthält den Haupteinstiegspunkt des Programms. Darin wird entschieden ob die „SIGMA SharePoint Belegarchivierung“ als Dienst oder als Windows Forms-Anwendung gestartet werden soll. Die Unterscheidung wird mit dem in Abbildung 24 ersichtlichen Quellcode realisiert.

```
#if SERVICE
    ServiceBase.Run(new SPBelegarchivierung());
#else
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new CMainForm());
#endif
```

**Abbildung 24** Start der Anwendung

Mithilfe der #if, #else und #endif Anweisungen können bedingte Direktiven festgelegt werden. Somit kann entschieden werden, ob die Anwendung im Debug-Modus oder im Release-Modus erstellt werden soll. In der Konfiguration für den Release-Modus wird das Symbol „SERVICE“ für die bedingte Kompilierung hinterlegt.

Bei der Ausführung der Anwendung wird der Dienst gestartet und ein Objekt der Klasse SPBelegarchivierung übergeben. Die Klasse SPBelegarchivierung erbt von ServiceBase. Dadurch stehen verschiedene Funktionen, wie OnStart oder OnStop, zur Verfügung. Diese zwei Funktionen werden beim Starten bzw. beim Beenden des Dienstes ausgeführt. Im Konstruktor der Klasse werden die Dienstinformationen „ServiceName“ und „EventLog“ gesetzt. Außerdem werden die Eigenschaften „CanPauseAndContinue“, „CanShutdown“ und „CanStop“ auf „true“ gesetzt. Die OnStart Methode der Basisklasse wird überschrieben. Bei der Ausführung dieser wird die Dokumentenbibliothek in der „Microsoft SharePoint Foundation 2010“ angelegt, falls diese noch nicht vorhanden ist. Weiterhin wird der WorkingThread gestartet. In diesem Thread werden die PDF-Dateien in die „Microsoft SharePoint Foundation 2010“ hochgeladen. Durch die OnStop-Methode kann der Dienst beendet werden. Die Methoden OnPause und OnContinue dienen dazu, dass der Dienst angehalten und wieder fortgesetzt werden kann. Beim Herunterfahren des Systems wird die Methode OnShutdown ausgeführt.

Um Fehler oder Informationen der „SIGMA SharePoint Belegarchivierung“ dokumentieren zu können, wird ein Log eingerichtet. Es wird täglich eine neue Log-Datei in einem vorher definierten Ordner abgelegt. In dieser Datei werden alle Meldungen abgelegt. Zusätzlich wird die Uhrzeit vermerkt, so dass man Fehler besser nachvollziehen kann. Das Öffnen, Bearbeiten und Schließen der Datei wird mit einem Lock-Objekt versehen. Dadurch wird abgesichert, dass nur einmal auf die Datei zugegriffen wird.

Um die Dokumentenbibliothek anzulegen bzw. zu prüfen, ob diese schon vorhanden ist, wird ein ClientContext zur „Microsoft SharePoint“-Website aufgebaut. Für die Authentifizierung wird das NetworkCredential-Objekt verwendet. (siehe Punkt 3.9). Nachdem dieses Objekt dem ClientContext zugewiesen wurde, beginnt die in Punkt 3.6 beschriebene Ausnahmebehandlung. Im Try-Block wird versucht, die Dokumentenbibliothek über den Titel zu laden. Falls dies fehlschlägt, wird die Dokumentenbibliothek im Catch-Block angelegt. Dazu wird der in Punkt 3.5 beschriebene Quellcode verwendet. Die Spaltennamen der Dokumentenbibliothek werden in der app.config hinterlegt. Dadurch ist es möglich, die Spalten mit einem anderen Namen anzulegen. Verwendet man Umlaute, Leerzeichen oder Bindestriche für die Spaltennamen, ersetzt das „Microsoft SharePoint“ diese durch den hexadezimalen HTML-Code. Deswegen müssen die Spaltennamen in der Konfiguration mit dem hexadezimalen Code geschrieben werden. Aus „Kunden-/ Lieferantenreferenznummer“ wird Kunden\_x002d\_x002f\_x0020\_Lieferantenreferenznummer. Bei der Anlage der Spalten muss der hexadezimale Code durch das entsprechende Unicode-Zeichen ersetzt werden. Dazu wird die in Abbildung 25 ersichtliche Funktion NormalizeString verwendet.

```
private string NormalizeString(string codedString)
{
    return codedString.Replace("_x0020_", " ").Replace("_x00e4_", "ä")
        .Replace("_x0020_", "ö").Replace("_x00e4_", "ü").Replace("_x0020_", "ß")
        .Replace("_x00e4_", "Ä").Replace("_x0020_", "Ö").Replace("_x00e4_", "Ü")
        .Replace("_x002d_", "-").Replace("_x002f_", "/");
}
```

**Abbildung 25** Ersetzen der Hexadezimalen Zeichen durch Unicode Zeichen

Wenn die Dokumentenbibliothek neu angelegt wurde, müssen die Spalten „Modified“ und „Editor“ von der Standardansicht entfernt werden. Diese werden für die „SIGMA SharePoint Belegarchivierung“ nicht benötigt. Da die Versionsnummer angezeigt werden soll, muss das Feld „\_UIVersionString“ zu der Ansicht hinzugefügt werden. Um die View bearbeiten zu können, muss zuerst die Website und anschließend die Standardansicht geladen werden. Die Abbildung 26 veranschaulicht das Entfernen und Hinzufügen von Spalten in einer Ansicht.

```

try
{
    if (oList.Title != Properties.Settings.Default.documentListName) ;
}
catch (Microsoft.SharePoint.Client.PropertyOrFieldNotInitializedException oEx)
{
    oList = clientContext.Web.Lists.GetByTitle(Properties.Settings.Default.documentListName);
    clientContext.Load(oList.Views);
    clientContext.ExecuteQuery();

    Microsoft.SharePoint.Client.View oView = oList.Views[0];
    clientContext.Load(oView.ViewFields);
    clientContext.ExecuteQuery();

    oView.ViewFields.Remove("Modified");
    oView.ViewFields.Remove("Editor");
    oView.ViewFields.Add("_UIVersionString");
    oView.Update();
    clientContext.ExecuteQuery();
}

```

### Abbildung 26 Entfernen und Hinzufügen von Spalten in einer Ansicht

Bei der Initialisierung des Threads `m_oWorkingThread` wird die Methode `SharePointBelegarchivierung` als eigenständiger Prozess ausgeführt. Dem Thread werden die Eigenschaften `IsBackground = true`, `Priority = ThreadPriority.Normal` und `Name = "SharePoint - Upload Document Thread"` zugewiesen. Anschließend wird er über den Methodenaufruf `m_oWorkingThread.Start()` gestartet.

In der Methode `SharePointBelegarchivierung` werden die PDF-Dateien abgerufen, die Metadaten aus dem Dateinamen ausgelesen und an eine Funktion für das Hochladen der Dateien übergeben. Dazu werden die Informationen des Speicherortes in einem `DirectoryInfo`-Objekt gespeichert. Über die Funktion `GetFiles(„*.pdf“)` des Objektes können alle PDF-Dateien abgerufen und in einer `foreach`-Schleife durchlaufen werden. Die `foreach`-Schleife enthält ein `FileInfo`-Objekt `oInfo` der aktuellen Datei. Die Metadaten werden aus dem Namen der Datei ausgelesen. Die Dateiendung wird mithilfe Funktion `LastIndexOf('.')` entfernt. Unter Verwendung der Funktion `Split(';')` werden die Metadaten zerlegt und in ein Array geschrieben. Die Informationen über den Status des Beleges befinden sich in zwei Einträgen des Arrays. Der fünfte Wert enthält die Information, ob der Beleg storniert wurde, der sechste entscheidet, ob der Beleg offen oder geschlossen ist. Aus diesen beiden Werten wird der Status des Beleges generiert. Der in Abbildung 27 gezeigte Methodenaufruf übergibt alle notwendigen Informationen an die Funktion für das Hochladen der Dateien.

```

// Hochladen des Dokumentes
UploadDocument(String.Format("{0}_{1}_{2}.pdf", FileProperties[12], FileProperties[0], FileProperties[1]), // Dateiname
FileProperties[0], // Belegtyp
FileProperties[1], // Belegnummer
FileProperties[2], // Kunden-/ Lieferantenreferenznummer
FileProperties[3], // Geschäftspartnernummer
FileProperties[4], // Name des Geschäftspartners
status, // Status
FileProperties[7].Replace('_', ':'), // Belegdatum
FileProperties[8].Replace('_', ':'), // Erstelldatum
FileProperties[9], // Beleg erstellt von
FileProperties[10].Replace('_', ':'), // Aktualisierungsdatum
FileProperties[11], // Beleg aktualisiert von
FileProperties[12], // Datenbankname
System.IO.File.ReadAllBytes(oInfo.FullName)); // Dateistream

```

### Abbildung 27 Aufruf zum Hochladen des Dokumentes

Diese Funktion befindet sich in einem Try-Block. Dadurch wird abgesichert, dass evtl. auftretende Fehler abgefangen werden. Wenn das Hochladen der Datei funktioniert hat, wird dies im Log erfasst. Anschließend wird die Datei im Windows Dateisystem gelöscht. Falls ein Fehler aufgetreten ist, wird dieser ebenfalls im Log dokumentiert. Um zu vermeiden, dass der Dienst immer wieder versucht, diese Datei in die „SharePoint Foundation 2010“ hochzuladen, wird sie in den Ordner „failure“ verschoben. Wenn beim Verschieben der Datei ein Fehler auftritt, wird dies auch im Log vermerkt. Mögliche Fehler können z. B. sein: Auf das Dokument kann nicht zugegriffen werden, da es noch von coeresuite erstellt wird. Das Hochladen ist fehlgeschlagen, da der Server nicht erreichbar ist. Nach dem Hochladen aller vorhandenen Dateien wartet der Thread über Thread.Sleep(10000) zehn Sekunden, ehe er wieder überprüft, ob neue Dateien vorhanden sind.

Beim Hochladen der Dateien über die Methode UploadDocument wird ein ClientContext erstellt. Dieser erhält die Authentifizierungsdaten über das NetworkCredential-Objekt. (siehe Punkt 3.9) Die Dateien werden in die „Microsoft SharePoint Foundation 2010“ eingestellt, indem die in Punkt 3.7 und Punkt 3.8 beschriebenen Abläufe Verwendung finden. , Um den Netzwerkverkehr gering halten zu können, wird ausschließlich die Möglichkeit über das FileCreationInformation-Objekt genutzt. Trotzdem habe ich beide Varianten im Quellcode der Anwendung (siehe beiliegende CD) umgesetzt. Das FileCreationInformation-Objekt wird in der Methode UploadDocument verwendet. Die zweite Variante über die SaveBinaryDirect-Methode befindet sich in der Funktion UploadDocument2, wird aber bei der Ausführung des Dienstes nicht verwendet.

## 5 Zusammenfassung und Ausblick

Das Ziel der Arbeit war die Anbindung der „Microsoft SharePoint Foundation 2010“ an „SAP Business One“. Diese Aufgabe konnte erfolgreich gelöst werden.

Nunmehr ist es möglich, Belege aus „SAP Business One“ in der „Microsoft SharePoint Foundation 2010“ abzulegen und mit Metadaten zu beschreiben. Die Anbindung konnte mithilfe des verwalteten Clientobjektmodells der „Microsoft SharePoint Foundation 2010“ realisiert werden.

Im Laufe der Entwicklung stellte sich heraus, dass eine Weiterentwicklung der „SIGMA Belegarchivierung“ nicht umsetzbar ist. Dies liegt daran, dass die Belege aus dem „SAP Business One“ nicht als PDF-Datei abgegriffen werden können. Die Nutzung von coesuite als Add-on für „SAP Business One“ und einem Dienst, der die generierten Dokumente in die „Microsoft SharePoint Foundation 2010“ einstellt und die dazugehörigen Metadaten pflegt, ermöglicht diese Anbindung.

Derzeit existiert noch keine Rückmeldung von der „Microsoft SharePoint Foundation 2010“ an das „SAP Business One“. Es ist vorstellbar, künftig die „SIGMA SharePoint Belegarchivierung“ so zu erweitern, dass man direkt aus dem „SAP Business One“ auf Belege in der „Microsoft SharePoint Foundation 2010“ zugreifen kann bzw. erkennt, ob ein Beleg schon in der „Microsoft SharePoint Foundation 2010“ archiviert wurde.

Die Metadaten spalten sind zum jetzigen Zeitpunkt fest im Quellcode hinterlegt. In einer Weiterentwicklung der „SIGMA SharePoint Belegarchivierung“ ist es denkbar, diese Zuweisung dynamisch über die Konfigurationsdatei des Dienstes zu gestalten. So könnte man die „SIGMA SharePoint Belegarchivierung“ auch dazu nutzen, andere Dokumente automatisch in die „Microsoft SharePoint Foundation 2010“ einzustellen.

## Quellenverzeichnis

Authentifizierung in den verwalteten Clientobjektmodellen. 2010, In:

<http://msdn.microsoft.com/de-de/library/ee535729%28v=office.14%29.aspx>

coresystems: coresuite country package – Effiziente Projekte mit SAP B1. 2013, In:

[http://coresystems.ch/media/medialibrary/2013/05/Presentation\\_coresuite\\_country\\_package\\_DE\\_web.pdf](http://coresystems.ch/media/medialibrary/2013/05/Presentation_coresuite_country_package_DE_web.pdf)

ERP-Software: ERP Software – Der Weg zum Erfolg. 2013, In: <http://www.erp-software.org/>

MAJEED, Shakir: Two ways to Upload file from Client Object Model. 2011, In:

<http://junooni.wordpress.com/2011/09/30/single-step-to-upload-file-from-client-object-model/>

MAYERHOFER, Robert: Einstieg in SAP Business One. Bonn, 2009

NetworkCredential-Klasse. In: [http://msdn.microsoft.com/de-](http://msdn.microsoft.com/de-de/library/system.net.networkcredential%28v=vs.80%29.aspx)

[de/library/system.net.networkcredential%28v=vs.80%29.aspx](http://msdn.microsoft.com/de-de/library/system.net.networkcredential%28v=vs.80%29.aspx)

NTFS und FAT: Alle Unterschiede im Überblick. 2011, In: [http://www.com-](http://www.com-magazin.de/img/1/9/6/2/2/f32ae1b3302e4c5b.png)

[magazin.de/img/1/9/6/2/2/f32ae1b3302e4c5b.png](http://www.com-magazin.de/img/1/9/6/2/2/f32ae1b3302e4c5b.png)

POLJAKOV, Helena: Präsentation SharePoint 2010. 2012, In: [http://www.profi-](http://www.profi-ag.de/wps/wcm/connect/34a757804b99b678ab45fb0b77f7885f/Praesentation-PROFIAG-SharePoint2010.pdf?MOD=AJPERES)

[ag.de/wps/wcm/connect/34a757804b99b678ab45fb0b77f7885f/Praesentation-PROFIAG-SharePoint2010.pdf?MOD=AJPERES](http://www.profi-ag.de/wps/wcm/connect/34a757804b99b678ab45fb0b77f7885f/Praesentation-PROFIAG-SharePoint2010.pdf?MOD=AJPERES)

RENTERGENT, Jürgen: SharePoint – einfaches DMS. 2010, In:

<http://www.computerwoche.de/a/sharepoint-ein-einfaches-dms,1929446,5>

SAP Business One: Die betriebswirtschaftliche Lösung für kleine und mittlere Unter-

nehmen. 2008, In: [http://www.sigmagmbh.de/pic/pdf/sap\\_business\\_one.pdf](http://www.sigmagmbh.de/pic/pdf/sap_business_one.pdf)

SCHWICHTENBERG, Holger: Das .NET Framework. In:

[http://www.dotnetframework.de/default2.aspx?start=http://www.dotnetframework.de/dotnet/DOTNET\\_Framework\\_Einfuehrung.asp](http://www.dotnetframework.de/default2.aspx?start=http://www.dotnetframework.de/dotnet/DOTNET_Framework_Einfuehrung.asp)

Upload a document to a SharePoint list from Client Side Object Model. 2012, In:

<http://stackoverflow.com/questions/9847935/upload-a-document-to-a-sharepoint-list-from-client-side-object-model>

Using-Anweisung: using-Anweisung (C#-Referenz). 2008, In:

<http://msdn.microsoft.com/de-de/library/yh598w02%28v=vs.90%29.aspx>

Verwenden des Ausnahmebereichs: Gewusst wie: Verwenden des Ausnahmebereichs. 2010, In: <http://msdn.microsoft.com/de-de/library/ee534976%28v=office.14%29.aspx>

WHITE, Eric: Verwenden des verwalteten Clientobjektmodells von SharePoint Foundation 2010. 2009, In: <http://msdn.microsoft.com/de-de/library/ee857094%28v=office.14%29.aspx>

Wikipedia: Microsoft SharePoint Foundation. 2013, In: [http://de.wikipedia.org/wiki/Microsoft\\_SharePoint\\_Foundation](http://de.wikipedia.org/wiki/Microsoft_SharePoint_Foundation)

Wikipedia: Microsoft SharePoint Server. 2013, In: [http://de.wikipedia.org/wiki/Microsoft\\_SharePoint\\_Server](http://de.wikipedia.org/wiki/Microsoft_SharePoint_Server)

ZARUBA, Stefan: Microsoft SharePoint Vortrag. 2008, In: [http://www.in-gmbh.de/uploads/media/Microsoft\\_SharePoint\\_Vortrag.pdf](http://www.in-gmbh.de/uploads/media/Microsoft_SharePoint_Vortrag.pdf)

# Ehrenwörtliche Erklärung

„Ich erkläre hiermit ehrenwörtlich“,

1. dass ich meine Bachelorthesis mit dem Thema

Anbindung von „Microsoft SharePoint“ an „SAP Business One“

ohne fremde Hilfe angefertigt habe,

2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe und

3. dass ich meine Bachelorthesis bei keiner anderen Prüfung vorgelegt habe.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

---

Ort, Datum

---

Unterschrift

## Thesepapier

Belege aus dem „SAP Business One“ können mithilfe der „SIGMA SharePoint Belegarchivierung“ in die „Microsoft SharePoint Foundation 2010“ hochgeladen werden.

Der Austausch von Daten zwischen der „SIGMA SharePoint Belegarchivierung“ und der „Microsoft SharePoint Foundation 2010“ wird durch das verwaltete Clientobjektmodell ermöglicht. Die PDF-Dateien werden durch das „coresuite“-Add-on der Firma „coresystems ag“ generiert.

Die Bezeichnung der Dokumentenbibliothek und der Spaltennamen in der „Microsoft SharePoint Foundation 2010“ kann in der Konfigurationsdatei der „SIGMA SharePoint Belegarchivierung“ initialisiert werden. Beim ersten Start des „SIGMA SharePoint Belegarchivierungsdienstes“ wird die Dokumentenbibliothek automatisch angelegt und konfiguriert.

Durch die vorliegende Arbeit ist es nun möglich, Belege aus dem „SAP Business One“ in der „Microsoft SharePoint Foundation 2010“ abzulegen und somit auf kosten- und ressourcenintensive Audrucke und deren aufwändige Archivierung zu verzichten.

In einem weiteren Schritt könnte der Dienst derart dynamisch gestaltet werden, dass er es ermöglicht, andere Dokumente automatisch in der „Microsoft SharePoint Foundation 2010“ abzulegen. Ein weiteres lohnendes Ziel besteht darin, eine automatisierte Rückmeldung der „Microsoft SharePoint Foundation 2010“ an des „SAP Business One“ zu realisieren.