

# **Bachelorarbeit**

Jan Schwarzer

Evaluation von Nintex Workflow 2007 für die  
Erstellung von Workflow-Lösungen unter Microsoft  
SharePoint Technologien

**Jan Schwarzer**

Evaluation von Nintex Workflow 2007 für die  
Erstellung von Workflow-Lösungen unter Microsoft  
SharePoint Technologien

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung  
im Studiengang Angewandte Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Michael Schäfers  
Zweitgutachter : Prof. Dr. Bettina Buth

Abgegeben am 01. April 2009

## **Thema der Bachelorarbeit**

Evaluation von Nintex Workflow 2007 für die Erstellung von Workflow-Lösungen unter Microsoft SharePoint Technologien

## **Stichworte**

Nintex Workflow 2007, K2 blackpoint beta 2, SharePoint, XML, Web Services, Workflows, InfoPath

## **Kurzzusammenfassung**

Die Automatisierung und digitale Umsetzung von Geschäftsprozessen kann für Unternehmen einen großen wirtschaftlichen Mehrwert darstellen. Prozesse werden schneller und effizienter durchgeführt und die Kosten für den Gesamtprozess sinken. Aufwendige und nicht selten fehleranfällige Papierarbeit wird vermieden.

Um entsprechende Prozesse schnell und mit geringem Aufwand im Unternehmen unter Einbindung der Fachanwender, die das nötige interne Prozess-Know-how besitzen, einzuführen, bedarf es entsprechender Lösungen.

Diese Bachelorarbeit nimmt sich dieser Problematik an und analysiert eine 3rd-Party Lösung zur Umsetzung von Workflows auf Basis der Microsoft SharePoint Plattformen.

## **Title of the paper**

Evaluation of Nintex Workflow 2007 for creating workflow solutions with Microsoft SharePoint technologies

## **Keywords**

Nintex Workflow 2007, K2 blackpoint beta 2, SharePoint, XML, Web Services, Workflows, InfoPath

## **Abstract**

The automation and digitalization of business processes is a valuable goal for companies. Processes can be handled faster and much more efficient. The total cost of ownership decreases and the complex and often error prone paper work is not necessary anymore.

Therefore it is critical to choose the right software tools which allow to integrate the process specialists in the implementation of new processes.

This thesis discusses these problems and evaluates a 3rd-party-tool for developing workflows under the Microsoft SharePoint platform.

# Tagungsbeitrag

Im Rahmen der Aufgabenstellung von der Firma HanseVision wurden Teilergebnisse aus dieser Bachelorarbeit in Absprache mit dem Betreuer vorab veröffentlicht:

**Jan Schwarzer, Sven Maier;**

„Workflows mit SharePoint 3rd-Party-Lösungen: Nintex Workflow und k2.blackpoint – Ein Vergleich“; SharePoint Konferenz 2009, München, 12.Feb.2009;

<http://www.sharepointkonferenz.de/>

# Danksagung

Diese Bachelorarbeit markiert den Abschluss einer sehr schönen Zeit, wenn nicht sogar der schönsten Zeit meines bisherigen Lebens. Ohne einige Menschen wäre dieser Schritt allerdings niemals möglich gewesen und genau diesen möchte ich mit den folgenden Sätzen meinen Dank aussprechen.

Zu allererst möchte ich mich bei meinen Kommilitonen bedanken, mit denen ich diese schöne Zeit gemeinsam meistern konnte.

Im direkten Bezug auf diese Bachelorarbeit geht mein Dank an:

- Die Firma HanseVision GmbH
- Sven Maier von der Hansevision und meinem Erstprüfer Prof. Dr. Michael Schäfers, welche mich ausgezeichnet betreuten,
- Prof. Dr. Bettina Buth, meiner Zweitprüferin,
- Meinen Korrekturlesern: Ulf, Thomas und Martina

Zuletzt möchte ich mich bei meiner Familie bedanken, die mir jederzeit den Rücken freigehalten hat und mir mit Rat und Tat zur Seite stand.

# Inhaltsverzeichnis

1	Einleitung.....	10
1.1	Motivation.....	11
1.2	Zielsetzung.....	12
1.3	Zielgruppen.....	12
1.4	Gliederung.....	12
1.5	Persönliche Anmerkungen .....	13
1.5.1	Begründung für Forcierung auf Microsoft.....	13
1.5.2	Gewonnene Erkenntnisse.....	13
2	Technologien.....	14
2.1	Extensible Markup Language (XML).....	15
2.1.1	Bestandteile eines XML-Dokuments.....	16
2.1.1.1	<i>Document Information Item</i> .....	16
2.1.1.2	<i>Element Information Item</i> .....	16
2.1.1.3	<i>Attribute Information Item</i> .....	16
2.1.1.4	<i>Comment Information Item</i> .....	17
2.1.2	Gestaltung von XML-Dokumenten.....	17
2.1.3	XML-Schema-Definition (XSD).....	18
2.1.3.1	<i>Aufbau eines XSD-Dokuments</i> .....	18
2.1.3.2	<i>Einfache Datentypen</i> .....	19
2.1.3.3	<i>Komplexe Datentypen</i> .....	20
2.2	Web Services.....	21
2.2.1	Web Service Standards.....	22
2.2.2	Service-oriented Architecture (SOA).....	23
2.3	Microsoft Office InfoPath 2007.....	23
2.3.1	Formular vs. Formularvorlage.....	24
2.3.2	Bedarfsermittlung.....	26
2.3.2.1	<i>Effiziente Teamarbeit</i> .....	26
2.3.2.2	<i>Integration in Arbeitsprozesse</i> .....	26
2.3.3	Kollaboration.....	26

2.3.4	Besonderheiten.....	26
2.4	SharePoint Plattform .....	27
2.4.1	Anwendungsdomänen.....	28
2.4.2	Themenbereiche .....	28
2.4.3	Varianten.....	29
2.4.3.1	<i>Windows SharePoint Services 3.0 (WSS)</i> .....	30
2.4.3.2	<i>Microsoft Office SharePoint Server 2007 (MOSS)</i> .....	30
2.4.3.3	<i>Microsoft Office SharePoint Server 2007 for Search (MOSS)</i> .....	30
2.4.3.4	<i>Abgrenzung von MOSS 2007 zu WSS 3.0</i> .....	30
2.4.4	Gesamtzusammenhang .....	31
2.4.5	Leitfragen.....	32
2.4.6	Aufbau .....	32
2.4.6.1	<i>Inhaltstypen</i> .....	33
<b>3</b>	<b>Workflow .....</b>	<b>35</b>
3.1	Begriffe .....	36
3.1.1	Prozess.....	36
3.1.2	Geschäftsprozess.....	36
3.1.3	Workflow.....	36
3.1.3.1	<i>Sequentieller Workflow</i> .....	37
3.1.3.2	<i>Zustandsbasierter Workflow</i> .....	37
3.1.4	Workflow-Management .....	38
3.1.4.1	<i>Aufgaben</i> .....	38
3.1.4.2	<i>Ziele</i> .....	38
3.2	Workflow-Integration in SharePoint.....	39
3.2.1	Windows Workflow Foundation (WF) .....	39
3.3	Alternativen.....	39
3.4	Das Referenzbeispiel.....	39
3.4.1	Urlaubsantrag.....	40
3.4.1.1	<i>Anforderungen</i> .....	40
3.4.1.2	<i>Szenario</i> .....	40
<b>4</b>	<b>Nintex Workflow 2007 .....</b>	<b>42</b>
4.1	Allgemeine Informationen.....	43
4.1.1	Rollen.....	43
4.1.2	Data Connection Libraries.....	43
4.1.3	Forms Libraries .....	43
4.1.4	Cross Domaining .....	43
4.2	Das Tool.....	44
4.2.1	Lizenzen.....	44

4.2.2	Partner und Kunden .....	44
4.2.3	Kollaboration.....	44
4.2.4	Installation.....	45
4.2.4.1	Voraussetzungen.....	45
4.2.5	Integration .....	46
4.2.5.1	Zentraladministration.....	46
4.2.5.2	Webseitenadministration.....	46
4.2.5.3	Benutzer-Menü.....	47
4.2.5.4	Settings-Menü.....	48
4.2.6	Entwicklungstools .....	48
4.2.6.1	Web Designer.....	49
4.2.6.2	Software Development Kit (SDK) .....	50
4.2.7	Reporting.....	50
4.2.8	Workflow-Historie .....	51
4.2.9	Versionierung.....	52
4.2.10	LazyApproval.....	52
4.3	Beispiel: Urlaubsantrag.....	54
4.3.1	Start .....	55
4.3.2	Zustand: Mitarbeiter.....	56
4.3.3	Zustand: Vorgesetzter.....	57
4.3.4	Zustand: Personalabteilung.....	64
4.3.5	Zustand: Ende.....	65
4.3.6	Das Formular .....	66
4.3.7	Testfälle zur Prüfung der Richtigkeit.....	68
<b>5</b>	<b>Analyse und Evaluation.....</b>	<b>70</b>
5.1	Kategorien und Kriterien .....	71
5.1.1	Kriterienfindung.....	71
5.1.2	Kategorien.....	71
5.1.3	Kriterien.....	71
5.1.4	Gewichtungen der Kategorien.....	72
5.2	Bewertungssystem .....	74
5.2.1	Berechnungsformel.....	75
5.2.2	Gewichtung der Kriterien.....	75
5.2.3	Rechenbeispiel.....	75
5.3	Evaluation und Ergebnis .....	76
5.4	Auswertung des Ergebnisses.....	77
5.4.1	Workflow-Engine.....	77
5.4.2	Usability .....	79



5.4.3	Infrastruktur .....	80
5.4.4	Dokumentation .....	81
5.4.5	Allgemeine Produkteigenschaften.....	82
5.4.6	Support.....	83
5.5	Vor- und Nachteile .....	84
5.6	Fazit .....	84
<b>6</b>	<b>Alternative Workflow-Lösungen.....</b>	<b>86</b>
6.1	K2 blackpoint beta 2 .....	87
6.1.1	Problematiken und Begründung für den Ausschluss .....	87
6.1.2	Besonderheiten.....	87
6.1.3	Vor- und Nachteile.....	87
6.2	SAP Business Workflow .....	88
6.3	BlueSpring BPM .....	89
6.4	Citrix Workflow Studio.....	89
6.5	GEDYS IntraWare 7.....	90
6.6	IBM Lotus Workflow 7 .....	90
<b>7</b>	<b>Zusammenfassung .....</b>	<b>91</b>
<b>8</b>	<b>Ausblick .....</b>	<b>92</b>
	<b>Literaturverzeichnis .....</b>	<b>93</b>
<b>A</b>	<b>Verzeichnisse .....</b>	<b>94</b>
A.1	Abbildungsverzeichnis.....	94
A.2	Tabellenverzeichnis .....	95
A.3	Quellcodeverzeichnis.....	95
A.4	Formelverzeichnis .....	95
<b>B</b>	<b>Anhang.....</b>	<b>96</b>
B.1	Testfragen .....	96
B.2	MOSS- und WSS-Funktionalitäten im Vergleich.....	97
B.3	Nintex Workflow 2007 Aktionen .....	99
	<b>Erklärung.....</b>	<b>101</b>

# 1 Einleitung

Im Rahmen dieser Einleitung soll auf grundlegende Dinge eingegangen werden, die sich um die Beweggründe für die Evaluation von Nintex Workflow 2007 drehen.

Die Einleitung beginnt mit einer Motivation, warum es lohnend ist, sich mit diesem Thema auseinanderzusetzen. Gerade Fachanwender bzw. Entwickler können aus dieser Bachelorarbeit einen Mehrwert für die Arbeit mit *SharePoint*<sup>1</sup> bzw. Workflows unter SharePoint ziehen.

Die Zielsetzung nennt das eigentliche Ziel dieser Bachelorarbeit, woraufhin die Zielgruppen und die Verdeutlichung der Gliederung folgt. Hier soll vor allem der „*Rote Faden*“ durch die einzelnen Kapitel beschrieben werden. Im Rahmen dieser Beschreibung soll jedes Kapitel kurz erläutert und das aufeinander aufbauen der Themen gezeigt werden.

Abschließen wird die Einleitung mit einigen persönlichen Anmerkungen, wo es weniger um die sachlichen Werte geht, als vielmehr um Erfahrungen, die während des Erarbeitens gesammelt wurden.

Als letztes noch eine Anmerkung zum Thema an sich: Im Laufe der Evaluation zeigte sich, dass die Bachelorarbeit nicht in der ursprünglich angedachten Form umgesetzt werden konnte. Da das Thema zuerst anders geplant war, als es jetzt umgesetzt wurde, wird die Begründung über den Wegfall von *K2 blackpoint beta 2*<sup>2</sup> von der Firma K2 eine wichtige Rolle spielen. Das Tool sollte im gleichen Maße Bestandteil dieser Bachelorarbeit werden, wie es jetzt Nintex Workflow 2007 geworden ist. Die genaueren Gründe werden in [6.1 K2 blackpoint beta 2](#) weiter erläutert.

---

<sup>1</sup> SharePoint ist eine von Microsoft entwickelte Kommunikations- und Managementplattform und dient beispielsweise der gemeinsamen Zusammenarbeit an Inhalten in einem zentralen Firmenportal

<sup>2</sup> K2 blackpoint beta 2 ist ein Konkurrenzprodukt zu Nintex Workflow 2007

## 1.1 Motivation

In den letzten Monaten wurde eines immer wieder deutlich: Viele Kunden der Firma HanseVision möchten Kompetenzen bei der Workflow-Entwicklung im eigenen Hause aufbauen und so wurde der Ruf nach möglichen und anwenderfreundlichen Lösungen immer lauter. Auch zeigte sich, dass Geschäftsprozesse zunehmend automatisiert und digitalisiert werden. D.h. sie werden nicht mehr wie üblich mit „Stift und Papier“, sondern mittels Software-Lösungen abgewickelt.

Da viele dieser Kunden auf irgendeine Art Software setzen (i.d.R. SharePoint), welche auf ihre jeweils eigene Art und Weise Dokumente speichert, Daten verwaltet oder Geschäftsabläufe in digitaler Form abbildet, muss eine Workflow-Lösung her, die sich in die bereits vorhandene Unternehmens-Infrastruktur integrieren lässt. In dieser Hinsicht ist Nintex Workflow 2007 nicht die einzige, aber die hier untersuchte Software, welche sich direkt in Microsoft SharePoint einbettet (siehe [4.2.5 Integration](#)). Es werden jedoch noch Alternativen im weiteren Verlauf dieser Bachelorarbeit genannt (siehe Kapitel [6](#)).

Es gibt eine Vielzahl von Argumenten, die motivieren, sich mit dem Thema dieser Bachelorarbeit zu befassen. Folgend die wesentlichen Beweggründe:

- **Wachsender Markt**

„Das SharePoint Universum wird sich weiter ausdehnen“, schreibt Oliver Staubli (1). SharePoint erhält bei jeder neuen Version *neue Features*<sup>3</sup>, die mehr und mehr Kundenwünsche erfüllen und Geschäftsfelder abdecken. Daher werden hier immer wieder Fachleute gesucht, die sich mit SharePoint auskennen. Die Einsatzbereiche gehen über die einfache Entwicklung bis hin zum reinen Consulting.

- **Große Verbreitung**

Da SharePoint eine von Microsoft erstellte Software ist, hat diese entsprechende große Nachfragen - also viele Kunden, die eben diese nutzen. Das hat zur Folge, dass viele Kunden Dienstleistungen Dritter in Anspruch nehmen, damit diese dessen Wünsche umsetzen. In Deutschland werden die meisten SharePoint Projekte durch Firmen, wie die HanseVision, verwirklicht. Diese Firmen sind sehr spezialisiert und setzen ausschließlich auf Microsoft Produkte.

- **Weitgreifender Support**

Unter gewissen Voraussetzungen können die Support Möglichkeiten für das Arbeiten mit SharePoint und Nintex Workflow 2007 voll ausgeschöpft werden. Erstens gibt es den Microsoft Support, wenn es um SharePoint Fragen geht und zweitens bietet Nintex einen hervorragenden Support für die eigenen Produkte an. Beides sind wichtige Punkte, wenn es um die Einarbeitung in ein neues Tool und mit SharePoint in eine neue Umgebung geht.

- **Hohes Umsatzpotential**

Kunden, die sich mit SharePoint auseinander setzen bzw. es einsetzen, investieren i.d.R. auch entsprechende Budgets in Projekte. Daraus folgt auch für Projekte unter Nintex Workflow 2007 ein gewisses Potential, hier entsprechend Anwendung zu finden, da grundsätzlich alles in Verbindung mit SharePoint steht.

- **Breites Kundenspektrum**

Das Vorteilhafte an SharePoint ist, dass es eine sehr große Varianz an Kunden hat. Es gibt mittelständische, aber auch große Unternehmen, die SharePoint einsetzen. Deswegen ist hier

---

<sup>3</sup> <http://office.microsoft.com/de-de/getstarted/HA100738471031.aspx>

ein Einsatz von Nintex Workflow 2007 zu prüfen, ob sich nicht ähnliche Muster ableiten lassen, wie es bei der HanseVision zurzeit der Fall ist.

## 1.2 Zielsetzung

Das Ziel dieser Bachelorarbeit ist es, die Praxistauglichkeit von Nintex Workflow 2007 zu prüfen und anhand dieser Evaluation eine Auswertung zu erstellen, um dann letzten Endes zu einem aussagekräftigen Ergebnis zu kommen. Durch dieses Ergebnis kann die Praxistauglichkeit von Nintex Workflow 2007 abgeleitet werden.

Die Auswertung soll zusätzlich Unternehmen im Entscheidungsprozess zu einer Lösung unterstützen und ihnen genauestens die Vor- und Nachteile von Nintex Workflow 2007 aufzeigen.

## 1.3 Zielgruppen

Inhaltlich konzentriert sich diese Bachelorarbeit auf zwei Zielgruppen, welche im Unternehmen etwaige Prozesse mit diesem Tool umzusetzen haben. I.d.R. wird eine Entscheidung, für oder gegen Nintex Workflow 2007, durch diese Zielgruppen gefällt - welche wären:

- Die Entwickler
- Und die Fachanwender (ohne Programmierkenntnisse)

## 1.4 Gliederung

Diese Bachelorarbeit gliedert sich in insgesamt acht Kapitel. Folgend soll die Gliederung bzw. der „Rote Faden“ dieser verdeutlicht werden.

Die Einleitung in die Thematik wird von dem Kapitel 2 gefolgt. Das Kapitel 2 bildet eine sehr wichtige Grundlage für das Verständnis vieler eingesetzter Technologien, die sonst die meiste Zeit im Hintergrund verborgen bleiben. Als eine der grundlegendsten Technologien dient XML<sup>4</sup>. Daher sollen anfangs die nötigen Kenntnisse in XML vermittelt werden. Hierauf folgt die Thematik der Web Services<sup>5</sup>. Beide Themen gewinnen in der heutigen Zeit mehr und mehr an Bedeutung und werden im Rahmen dieser Bachelorarbeit vor allem in Verbindung mit Microsoft InfoPath<sup>6</sup> verwendet, welches im Anschluss verdeutlicht wird. Das Kapitel 2 schließt mit dem Themenkomplex der SharePoint Plattform ab. Das gesamte Kapitel ist gerade für einen Fachanwender interessant, da die wesentlichen Grundlagen geklärt werden.

Ähnlich verhält es sich mit dem Kapitel 3. Dieses schult das grundsätzliche Verständnis zur Thematik Workflows. Dabei geht es speziell um die damit verbundenen Fachbegriffe, als auch um die Integration von Workflows in SharePoint, etwaige alternative Lösungen und dem Referenzbeispiel, an welchem Nintex Workflow 2007 konkret vorgestellt wird. Dieses Kapitel ist für beide Zielgruppen gut geeignet.

Das Kapitel 4 soll dem Anwender das Tool an sich vorstellen, allgemeine Informationen zum Umgang mit Nintex Workflow 2007 vermitteln. Sowohl der Entwickler, als auch der Fachanwender finden hier den richtigen Einstieg in das untersuchte Tool.

Kapitel 5 beinhaltet die eigentliche Analyse und die Evaluation von Nintex Workflow 2007. Hier geht es ganz entscheidend darum, Nintex Workflow 2007 anhand von aufgestellten Kriterien zu beurteilen. Abschließen wird dieses Kapitel mit einer Auswertung der Ergebnisse und der Erklärung, warum das Tool an welcher Stelle wie benotet wurde. Ziel ist es hier, dem Leser einen

---

<sup>4</sup> XML steht für Extensible Markup Language und ist eine Metasprache

<sup>5</sup> Ein Web Service ist eine Internetanwendung, die es über international standardisierte Formalien ermöglicht, Informationen auszutauschen (via XML)

<sup>6</sup> Microsoft InfoPath erlaubt es, digitale und regelbasierte Formulare zu kreieren

soliden Eindruck zu vermitteln, inwieweit Nintex Workflow 2007 in der Praxis einsetzbar ist und wo dessen Stärken und Schwächen liegen.

Nintex Workflow 2007 ist der zentrale Bestandteil dieser Bachelorarbeit, jedoch gibt es sowohl für SharePoint, als auch für andere Plattformen, Alternativen. Auf genau diese Alternativen soll im Kapitel 6 eingegangen werden.

Die Bachelorarbeit endet mit den Kapiteln 7 und 8, wo es darum geht, das bereits Gelernte wieder aufzugreifen und ausblickend ein paar Denkanstöße zu geben, an welcher Stelle aufbauend weiter gearbeitet werden kann.

## 1.5 Persönliche Anmerkungen

Bevor das Kapitel 2 startet, sollen in diesem Abschnitt noch ein paar persönliche Anmerkungen zur Bachelorarbeit aufgeführt werden.

### 1.5.1 Begründung für Forcierung auf Microsoft

Der erste wichtige Punkt dreht sich generell um die Frage, warum in dieser Bachelorarbeit ausschließlich auf Microsoft Lösungen gesetzt wurde. Dem liegen zwei grundlegende Aspekte zu Grunde:

- Die Firma HanseVision setzt ausschließlich auf Microsoft Lösungen und ist zudem zertifizierter Microsoft Gold Partner
- Das hier untersuchte Produkt setzt ausschließlich auf Microsoft Produkte, hier speziell auf die SharePoint Plattform

Der Markt zeigt eine Vielzahl von Alternativen auf. Vor allem in Bezug auf die SharePoint Plattform, gibt es Lösungen anderer großer Hersteller, wie beispielsweise von IBM (siehe Kapitel 6). Ähnlich verhält sich dieses mit Nintex Workflow 2007. Auf etwaige Alternativen wird entsprechend hingewiesen.

### 1.5.2 Gewonnene Erkenntnisse

Der zweite wichtige Punkt sind die gewonnenen Erkenntnisse aus dem gesamten Bearbeitungsprozess dieser Bachelorarbeit, die an dieser Stelle festgehalten werden sollen.

- **Milestone Reports**  
Das Anlegen von wöchentlichen *Milestone Reports*<sup>7</sup>, um eigene Fortschritte im Laufe der letzten Monate gut verfolgen zu können, halfen enorm dabei, den Fortschrittsprozess dieser Bachelorarbeit immer wieder Richtung Ziel zu treiben. Auch halfen diese dabei, mit den betreuenden Professoren Wochenziele zu verifizieren, um so frühzeitig problemlösend einzugreifen.
- **Lerneffekt**  
Aus der ganzen Phase konnten wertvolle Erfahrungen gesammelt werden, auch wenn es an der einen oder anderen Stelle nicht weiterging. Trotzdem wird das Anlegen solcher wissenschaftlicher Arbeiten in Zukunft zum Einen einfacher und zum Anderem effizienter. So ergibt sich nicht nur eine fachliche Kompetenz aus dieser Bachelorarbeit, sondern auch die Erfahrung, sich an einer umfangreichen Arbeit „gemessen“ zu haben.

---

<sup>7</sup> Ein Milestone Report fasst in diesem Zusammenhang alle wichtigen Ereignisse einer Arbeitswoche zusammen und weißt zum Einen den zeitlichen Ablauf und zum Anderen die besonders guten bzw. besonders schlechten Ereignisse einer Arbeitswoche auf

## 2 Technologien

Das Thema dieser Bachelorarbeit ist eng verwoben mit vielen Lösungen aus der Microsoft Produktwelt. In diesem Rahmen ist es nicht möglich alle Facetten der verwendeten Lösungen in der vollen Tiefe zu beleuchten. Es wird jedoch ein Überblick über die wichtigsten Technologien im Kontext dieser Bachelorarbeit gegeben. Der Aufbau des vorliegenden Kapitels ist bewusst so gewählt, dass der Leser Schritt für Schritt an die Produkte herangetragen wird. Wichtig hierbei ist, einen Gesamtzusammenhang der hier beschriebenen Lösungen zu gewinnen und deren Einsatzmöglichkeiten kennenzulernen.

Da alle folgenden Inhalte dieses Kapitels, wie beispielsweise InfoPath oder die Web Services auf XML aufbauen, wird das Kapitel mit dem Thema XML eingeleitet. Geklärt werden soll hier in erster Linie, worum es sich bei XML handelt. Dazu werden sowohl der Aufbau eines XML-Dokuments besprochen, wie auch die Gestaltung und die Validierung von XML-Code.

Daraufhin wird auf den Themenbereich Web Services kurz eingegangen. Vor allem die Ursprünge und die technologiespezifischen Probleme von alternativen Lösungen sollen verstanden werden. Abschließend in diesem Abschnitt werden die aktuellen Standards besprochen und das Thema *Service-oriented Architecture (SOA)* angeschnitten (siehe [2.2.2 Service-oriented Architecture \(SOA\)](#)).

Als nächster Themenbereich wird InfoPath angesprochen. Nachdem geklärt wurde, um was es sich hierbei handelt, wird auf die grundlegenden Formularmodi eingegangen, da dies essentiell für die tägliche Arbeit mit InfoPath ist. Diesem Abschnitt folgt eine kurze Bedarfsermittlung, der Kollaboration (siehe [2.3.3 Kollaboration](#)) und abgeschlossen wird der Abschnitt mit einigen Hinweisen bzgl. der Besonderheiten von InfoPath.

Letztes Thema dieses Kapitels ist SharePoint, welches auch gleichzeitig der umfangreichste Abschnitt dieses Kapitels ist. SharePoint ist eine Software-Lösung, die mittlerweile in sehr vielen Bereichen der IT eingesetzt wird. Daher werden in diesem Abschnitt grundlegende Fragestellungen zu SharePoint geklärt und schwerpunktmäßig der Gesamtzusammenhang dieser Thematik, der Aufbau, die Einsatzgebiete und die Darstellung der verschiedenen Produktvarianten im Vergleich vorgestellt. Zudem werden Leitfragen besprochen, die bei der Entscheidungsfindung für eine SharePoint Lösung behilflich sein sollen.

Ziel dieses ganzen Kapitels ist es, dem Leser ein grundsätzliches Verständnis für die, in den darauf folgenden Kapiteln verwendeten, Technologien zu vermitteln. Gerade der Einstieg in die hier untersuchte Workflow-Lösung sollte sich nach diesem Kapitel, vor allem aber nach dem Abschnitt über SharePoint, einfacher gestalten. Auch sind die Code-Beispiele im XML-Abschnitt so gewählt, dass an dieser Stelle selber ausprobiert und nachprogrammiert werden kann.

## 2.1 Extensible Markup Language (XML)

XML wird zur Darstellung hierarchisch strukturierter Daten in Form von lesbaren Textdateien eingesetzt. Geschichtlich betrachtet stammt XML von der Beschreibungssprache *Standard Generalized Markup Language (SGML)* ab und enthält entsprechend eine Untermenge der Sprachmöglichkeiten von SGML (2).

SGML ist eine Metasprache. Das bedeutet, dass SGML in der Lage ist, eigene Sprachen zu definieren und man mit Hilfe dieser Sprache über andere Sprachen Aussagen machen kann. Ein Beispiel zur Verdeutlichung: Wenn beispielsweise in der deutschen Sprache über eine andere Fremdsprache gesprochen wird, dann ist die deutsche Sprache in diesem Falle die Metasprache und die, über die in diesem Moment gesprochene Fremdsprache, die Sprache, über welche Aussagen getroffen werden.

Was bedeutet dies im Hinblick auf XML? XML ermöglicht es, Grammatikstrukturen zu definieren (siehe [Quellcode 1](#)). Diese Grammatikstrukturen beschreiben eine Art eigene Sprache - die Metasprache, die wiederum mit natürlich sprachlichen Informationen gefüllt wird. Diese natürlich sprachlichen Informationen repräsentieren die eben genannte Fremdsprache.

Der große Vorteil von XML gegenüber anderen Sprachen, wie beispielsweise *HTML*<sup>8</sup> ist, dass sämtliche Elemente mit frei definierbaren Namen belegt werden können. In HTML gibt es nur vordefinierte Elemente wie `<table>`, `<div>`, `<br>`, `<p>`, `<i>`, usw. In XML hingegen, können Elemente wie `<Adresse>` oder `<Person>` ohne Probleme definiert werden. In [Quellcode 1](#) kann dazu ein einfaches Beispiel betrachtet werden. Bei der freien Namenswahl sollte allerdings der Zusammenhang zwischen dem ausgedachten Namen eines Elements und der wahren Bedeutung innerhalb einer Grammatik erkennbar sein.

### XML-Code

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Ein einfaches Beispiel -->
<Adresse>
  <Person Geschlecht="männlich">
    <Name>Max Mustermann</Name>
    <Ort>Musterstadt</Ort>
    <Plz>12345</Plz>
  </Person>
  <Person Geschlecht="weiblich">
    <Name>Maxi Musterfrau</Name>
    <Ort>Musterstadt</Ort>
    <Plz>12345</Plz>
  </Person>
</Adresse>
```

Quellcode 1 - Ein einfaches XML-Beispiel

Im Gegensatz zu HTML, ist XML Case-sensitive, d.h. es achtet auf die Groß- bzw. Kleinschreibung der einzelnen Elemente. Was jedoch beide gemein haben ist, dass beide Sprachen plattformunabhängig sind. Das ist natürlich dann interessant, wenn Systeme in Kommunikation treten sollen, die unterschiedliche Betriebssysteme nutzen. Ein anderes Beispiel wäre die Kommunikation von zwei Anwendungen, die in unterschiedlichen Sprachen entwickelt wurden. Auch hier ist die Anwendung von XML als Kommunikationsmedium einfach und praktikabel.

XML-Dokumente sind i.d.R. Dateitypen im XML-Format und tragen die Dateiendung `*.xml` während HTML-Dokumente die Dateiendung `*.html` haben. HTML und XML sind nicht zu

---

<sup>8</sup> HTML steht für Hypertext Markup Language, welches eine Beschreibungssprache ist und der Darstellung von Informationen in einem Browser dient

verwechseln. Beide besitzen deren Einsatzgebiet entsprechende Eigenschaften. Während es in HTML in erster Linie darum geht, das Aussehen von Dokumenten optimal und den Wünschen des Anwenders entsprechend zu formatieren, wird in XML hingegen nur die eigentliche Struktur eines Dokumentes festgelegt. Hier wird also ganz klar die Darstellung vom Inhalt abgekoppelt. Die eigentliche Formatierung findet mittels der *Extensible Stylesheet Language (XSL)* (siehe [2.1.2 Gestaltung von XML-Dokumenten](#)) oder *Cascading Style Sheets (CSS)*<sup>9</sup> statt. Auf die Cascading Stylesheets wird im Rahmen dieser Bachelorarbeit nicht weiter eingegangen.

### 2.1.1 Bestandteile eines XML-Dokuments

Ein XML-Dokument besteht aus genau einem *Information Set*. Ein Information Set beinhaltet alle Informationen eines Dokuments, also welche Elemente, Attribute, Kommentare usw. es enthält. Ein XML-Dokument ist somit ein Information Set mit n-beliebigen Einträgen, also konkret dessen verschiedenen Informationen.

In diesem Abschnitt sollen die wesentlichen Komponenten anhand von einfachen Beispielen erläutert werden.

#### 2.1.1.1 Document Information Item

##### XML-Code

```
<?xml version="1.0" encoding="UTF-8"?>
```

Quellcode 2 - Ein Document Information Item

In *Quellcode 2* ist ein Document Information Item definiert, der äußere Rahmen bzw. die Grundkonfiguration eines XML-Dokuments. Es definiert die Version *version="1.0"* sowie das Codierungsschema *encoding="UTF-8"*. Das Codierungsschema ist beliebig wählbar.

#### 2.1.1.2 Element Information Item

##### XML-Code

```
<Haus>
  <Zimmer>Wohnzimmer</Zimmer>
  <Zimmer>Schlafzimmer</Zimmer>
</Haus>
```

Quellcode 3 - Element Information Items

*Quellcode 3* zeigt Element Information Items, welche die Struktur eines XML-Dokuments definieren, indem diese ineinander verschachtelt werden. Diese bilden das Korsett für die anfangs erklärte Objektsprache. Wie bereits im vorhergegangenen Abschnitt erwähnt, sollten die Elemente aussagekräftige Namen tragen.

#### 2.1.1.3 Attribute Information Item

##### XML-Code

```
<Person Geschlecht="maennlich">
  <Name>Max Mustermann</Name>
</Person>
```

Quellcode 4 - Ein Attribute Information Item

In *Quellcode 4* ist ein Attribute Information Item zu sehen, welches im Gegensatz zu den Element Information Items keine weitere Unterstrukturierungen erlaubt. D.h. diese können nicht weiter verschachtelt werden. Die Hauptaufgabe ist es, einem Element Eigenschaften zuzuweisen.

<sup>9</sup> Cascading Style Sheets sind vor allem bekannt für die Gestaltung von HTML-Dokumenten



### 2.1.1.4 Comment Information Item

#### XML-Code

```
<!-- Ein Kommentar -->
```

Quellcode 5 - Ein Comment Information Item

Mit Hilfe des Comment Information Items werden Kommentare in XML definiert. Ein einfaches Beispiel ist in [Quellcode 5](#) zu sehen.

### 2.1.2 Gestaltung von XML-Dokumenten

Der Vollständigkeit halber ein kurzer Überblick über XSL bzw. *XSL Transformation (XSLT)*<sup>10</sup>: Mit Hilfe von XSL werden Layouts und Transformationen für XML-Dokumente definiert, weshalb diese häufig auch „*Stylesheets*“ genannt werden. Diese Definitionen können in ein XML-Dokument eingebunden werden. Wenn mit HTML bereits Erfahrungen gesammelt wurden, gestaltet sich XSL als nicht weiter schwierig. Lediglich das Einbinden der entsprechenden XML-Elemente in ein XSL-Dokument ist an dieser Stelle neu.

XSL-Dokumente werden im XML-Dokument mittels *Processing Instructions (PI)* eingebunden. Diese dienen dazu, Code aus anderen Dokumenten in einem Dokument zur Verfügung zu stellen, so dass dieses die entsprechenden Informationen verarbeiten kann. Umschlossen werden PIs immer von einem öffnenden `<?` und einem schließenden `?>`.

Ein in der Praxis sehr häufig angewandtes Beispiel einer PI ist eben das Einbinden eines XSL-Dokuments in ein XML-Dokument. Auch das Document Information Item, also der standardmäßig eingebundene Hinweis auf die XML-Version und das Codierungsschema ist eine PI. Hier ist vor allem die Definition der Codierungsschemata wichtig.

#### XSL- und XML-Code

##### XML-Dokument

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml:stylesheet type="text/xsl" href="beispiel.xsl"?>
<Person Geschlecht="maennlich">
  <Name>Max Mustermann</Name>
</Person>
```

##### XSL-Dokument

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:template xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <html>
    <body>
      <div style="color:red; font-weight: bold;">
        <xsl:value-of select="Person/Name"/>
      </div>
    </body>
  </html>
</xsl:template>
```

Quellcode 6 - Ein Beispiel für ein einfaches XSL-Dokument

An diesem Beispiel ist gut zu sehen, wie einfach sich die Programmierung von XML-Dokumenten gestaltet, die um XSL-Dokumente erweitert werden.

In [Quellcode 6](#) ist ein einfaches XML-Dokument zu sehen. Farblich hervorgehoben sind die PIs, die zuvor vorgestellt wurden. Die untere der beiden PIs enthält zwei Dinge: Zum Einen den Hinweis

<sup>10</sup> XSLT ist Teil von XSL. Mit Hilfe von XSLT ist es ferner möglich, ein Ursprungs-XML-Dokument in ein völlig neues Format zu überführen

darauf, dass es sich bei dem zu verarbeitenden Dokument um ein Stylesheet *xml:stylesheet* handelt und zum Anderen wird der Typ *text/xsl* und der Verweis zum XSL-Dokument *beispiel.xsl* definiert. Die eigentliche XML-Struktur ist bereits aus dem Abschnitt [2.1.1.2 Element Information Item](#) bekannt.

Im unteren Bereich steht der Code des XSL-Dokuments. Hier sind farblich der HTML-Code und die PI hervorgehoben. Es sei auf den kleinen Teil von *Inline-CSS-Code*<sup>11</sup> *color: red; font-weight: bold;* hingewiesen. Es ist leicht zu erkennen, wie einfach sich die Gestaltung von XML-Elementen mittels XSL in Verbindung mit HTML und CSS erweist. Mit Hilfe des Tags *xsl:value-of* wird konkret der Wert aus dem XML-Dokument gelesen: In diesem Fall der Wert aus *Person/Name*, was gleichzeitig der Suchhierarchie im XML-Dokument entspricht.

Das Ergebnis im Browser kann der [Abbildung 1](#) entnommen werden:

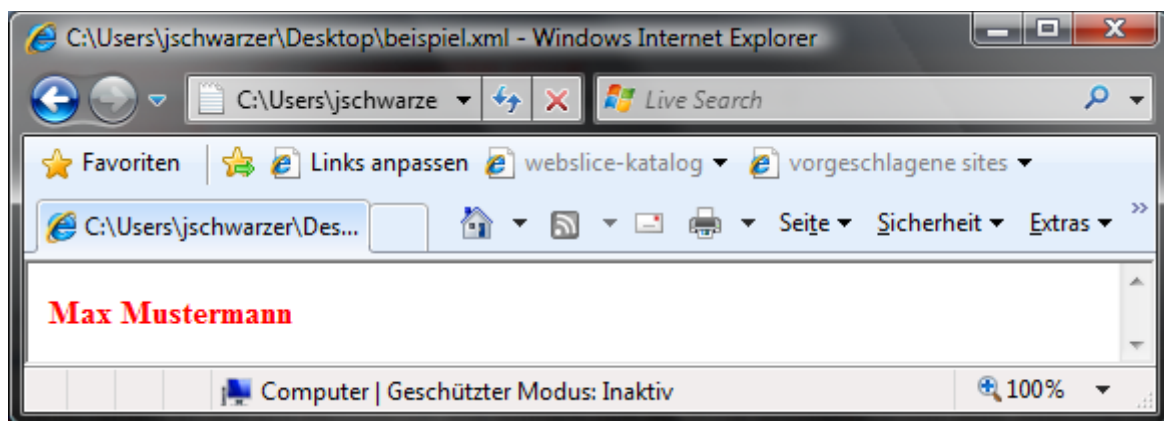


Abbildung 1 - Ein durch XSL formatiertes XML-Dokument im Browser

### 2.1.3 XML-Schema-Definition (XSD)

Ein wichtiger Bestandteil von XML sind die *Schema-Definitionen*. Mit Hilfe einer Schemasprache beschreibt XSD eigene Datentypen und Instanzen bzw. Gruppen von Instanzen, also die XML-Dokumente an sich, dieser Datentypen. XSD unterscheidet zwischen einfachen und komplexen Datentypen (siehe [2.1.3.2 Einfache Datentypen](#) bzw. [2.1.3.3 Komplexe Datentypen](#)). Wenn ein XML-Dokument eine Schema-Definition einbindet oder anders ausgedrückt, sich von einer Schema-Definition ableitet, muss es sich an die vordefinierten Regeln aus dem XSD-Dokument halten. XSD-Dokumente tragen die Dateierendung *\*.xsd*.

Schema-Definitionen sind immer dann ratsam, wenn es darum geht, den Informationsgehalt eines XML-Dokuments zu gewährleisten. Das hier aufgegriffene Problem ist, dass XML-Elemente frei benennbar sind und es so unter Umständen zu Kommunikationsschwierigkeiten zwischen zwei Parteien kommen kann. Ein klassischer Anwendungsfall wäre ein Web Service (siehe [2.2 Web Services](#)), bei welchem zwei Systeme über eben diesen in Kommunikation treten und Schnittstelleninformationen in einem XML-Dokument von einer Seite nicht eingehalten wurden - also ein Verstoß gegen die Konventionen der Schema-Definitionen vorliegt.

#### 2.1.3.1 Aufbau eines XSD-Dokuments

Ein XSD-Dokument enthält, wie auch ein normales XML-Dokument, ein Document Information Item (siehe [2.1.1.1 Document Information Item](#)). Neu hingegen ist das *<schema>* Element. Innerhalb dieses Elements wird zum Einen mittels der Attribute das XSD-Dokument konfiguriert und zum Anderen umschließt das Element alle weiteren Regeln oder Verschachtelungen von

<sup>11</sup> Ein Stück Quellcode, welches nicht in einer separaten Datei ausgelagert ist

weiteren Elementen. In [Quellcode 7](#) ist dieser Bereich durch `<!--Bereich für eigene Definitionen-->` markiert worden. Die zusätzlich farbig markierte Zuweisung, ist die Zuweisung des Standard Namensraums und wird mit der Anweisung `xmlns = „http://www.w3.org/2001/XMLSchema“` definiert. Diese Zuweisung muss in jedem XSD-Dokument definiert sein.

Mit Hilfe des Attributes `xmlns:xsd` wird auf den eben definierten Namensraum verwiesen. Oder anders ausgedrückt: So wird sichergestellt, dass nur Instanzen von einem Namensraum referenziert werden. Das hat den Hintergrund, dass in einem XML-Dokument mehrere Elemente mit identischen Namen auftreten können, dadurch aber eindeutig identifiziert werden. `xsd` ist in diesem Zusammenhang ein frei gewählter Bezeichner bzw. ein so genanntes „Präfix“, über welches alle in einem Namensraum definierten Elemente angesprochen werden können.

#### XSD-Code

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <!--Bereich für eigene Definitionen-->
</xsd:schema>
```

Quellcode 7 - Ein Beispiel für ein einfaches XSD-Dokument

Eine kurze Anmerkung noch zum Thema Validation von XSD: Zur Überprüfung von XSD-Code und dem daraus instanziierten XML-Code kann beispielsweise das einfache und hilfreiche Onlinetool *XML Schema Validator*<sup>12</sup> von der Firma DecisionSoft genutzt werden. Das Praktische daran ist, dass sogar hilfreiche Fehlermeldungen bei einem nicht validen Dokument ausgegeben werden, was sich während der Arbeit an diesem Themenabschnitt als besonders hilfreich erwies!

#### 2.1.3.2 Einfache Datentypen

Es gibt eine Vielzahl von einfachen Datentypen, wie in [Quellcode 8](#) zu sehen ist. Diese werden mit dem Element `simpleType` definiert. Je nachdem, welche Daten konkret für ein Element oder ein Attribut erwartet werden, kann auch explizit der Typ dazu angegeben werden. Das ist beispielsweise ein Vorteil gegenüber dem veralteten Standard *Documenttype Definition (DTD)*<sup>13</sup>. Die Schreibweise für einen Typen lautet immer: `Namensraum:Typ`. Im folgenden Beispiel wird der definierte Namensraum `xsd` wieder aufgegriffen und eine Sammlung der wichtigsten Datentypen in XSD gezeigt.

#### XSD-Code

```
xsd:string, xsd:decimal, xsd:integer, xsd:float, xsd:Boolean, xsd:date, xsd:time
```

Quellcode 8 - Eine Auflistung aller wichtigen einfachen Dateitypen in XSD

Wichtig ist: Ein einfacher Datentyp kann im Vergleich zu komplexen Datentypen keine eigenen Elemente oder Attribute beinhalten. Des Weiteren leitet sich ein einfacher Datentyp immer von einem Basistypen ab. Es gibt in XSD drei erlaubte Ableitungsformen: als *Beschränkung*, als *Liste* und als *Union*. Der Typ `double` ist beispielsweise eine beschränkte Ableitung des Typs `decimal`, d.h. dieser hat einen kleineren Gültigkeitsbereich. Im folgenden Beispiel wird das Prinzip eines einfachen Datentyps verdeutlicht. Dazu wird eine Liste als Ableitungsform des XSD-Dokuments verwendet.

<sup>12</sup> <http://tools.decisionsoft.com/schemaValidate/>

<sup>13</sup> Documenttype Definitions sind eine veraltete Technologie, die ähnlich wie XSD, Regeln für ein XML-Dokument definiert

XSD-und XML-Code
<b>XML-Dokument</b>
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;listItem&gt;a b c d e f &lt;/listItem&gt;</pre>
<b>XSD-Dokument</b>
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"&gt;   &lt;xsd:simpleType name="listOfStrings"&gt;     &lt;xsd:list itemType="xsd:string"/&gt;   &lt;/xsd:simpleType&gt;    &lt;xsd:element name="listItem" type="listOfStrings"/&gt; &lt;/xsd:schema&gt;</pre>

Quellcode 9 - Ein einfaches Beispiel für einen einfachen Datentypen

Sowohl im XSD-Code, als auch im XML-Code sind die wesentlichen Neuerungen bis hier farblich markiert worden.

Im oberen Bereich von [Quellcode 9](#) findet sich der zu validierende XML-Code, also eine mögliche Instanz des XSD-Codes. In diesem Fall ist der XML-Code eine valide Instanz, da dieser den definierten Regeln aus dem XSD-Dokument genügt.

Im unteren Bereich befindet sich der XSD-Code. Dieser macht nichts weiter, als ein Element *listItem* vom Typen *listOfStrings* und einen einfachen Datentypen *listOfStrings* zu definieren. Auf diesen einfachen Datentypen wird im Element verwiesen. Ohne ein Element kann ein einfacher Datentyp nicht dargestellt werden. Innerhalb des einfachen Datentyps wird eine Liste definiert, die sich aus *n* Strings zusammensetzen kann. Eine valide Instanz dieses XSD-Codes verhält sich also, wie der XML-Code weiter oben. Die Zeichenkette *a b c d e f* ist nicht aus Gründen der Übersicht mit *Leerzeichen* getrennt - das Leerzeichen ist ein fester Separator für Listeneinträge.

Es sei am Rande erwähnt, dass die Reihenfolge der Definitionen des Elements und des einfachen Datentypen keine Rolle spielen. Das Element *listItem* könnte genauso gut über dem einfachen Typ *listOfStrings* definiert werden.

### 2.1.3.3 Komplexe Datentypen

Ein komplexer Datentyp ist eine logische Gruppierung von Elementen und/oder Attributen. Dieser wird mit Hilfe des Elements *complexType* definiert. Das Vorteilhafte an den komplexen Datentypen ist, dass der Anwender dadurch in die Lage versetzt wird, Elemente ineinander zu verschachteln und so gewünschte Hierarchien im Dokument aufzubauen.

In [Quellcode 10](#) ist die Arbeitsweise eines komplexen Datentyps gut erkennbar. Der Code an sich bringt nicht viel Neues mit sich. Im Wesentlichen wurde hier das bekannte Beispiel aus [Quellcode 9](#) erweitert.

XSD-und XML-Code
<b>XML-Dokument</b> <pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;alphabet&gt;   &lt;abschnitt&gt;a bis f&lt;/abschnitt&gt;   &lt;listItem&gt;a b c d e f &lt;/listItem&gt;   &lt;abschnitt&gt;g bis l&lt;/abschnitt&gt;   &lt;listItem&gt;g h i j k l &lt;/listItem&gt; &lt;/alphabet&gt; </pre>
<b>XSD-Dokument</b> <pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"&gt;   &lt;xsd:simpleType name="listOfStrings"&gt;     &lt;xsd:list itemType="xsd:string"/&gt;   &lt;/xsd:simpleType&gt;    &lt;xsd:element name="alphabet"&gt;     &lt;xsd:complexType&gt;       &lt;xsd:sequence minOccurs="0" maxOccurs="unbounded"&gt;         &lt;xsd:element name="abschnitt" type="xsd:string"/&gt;         &lt;xsd:element name="listItem" type="listOfStrings"/&gt;       &lt;/xsd:sequence&gt;     &lt;/xsd:complexType&gt;   &lt;/xsd:element&gt; &lt;/xsd:schema&gt; </pre>

Quellcode 10 - Ein einfaches Beispiel für einen komplexen Datentypen

In [Quellcode 10](#) sind wieder die Neuerungen, die bis hier auftreten, farbig markiert worden. Der XML-Code ist auch hier eine valide Instanz des XSD-Codes.

Im XML-Code hat sich im Wesentlichen die Hierarchie geändert und eine Wiederholung von Elementen wurde eingeführt. Es gibt ein Alphabet und dieses wird in einzelne Abschnitte *abschnitt* und Einträge *listItem* gegliedert.

Der XSD-Code wurde um ein neues Element *alphabet* erweitert. In diesem Element befindet sich ein komplexer Datentyp *complexType*. Innerhalb des komplexen Datentyps ist eine Sequenz *sequence* definiert, welche durch die Attribute *minOccurs* und *maxOccurs* konfiguriert wird. Die Sequenz muss in diesem Fall mindestens *0-mal* und maximal *unendlich-mal* durchlaufen werden. In diesem Zusammenhang bedeutet *0-mal*, dass der XML-Code auch gültig wäre, wenn das Element *alphabet* keinerlei untergeordnete Elemente hätte. D.h. es können entweder n-beliebig viele oder keine Kombinationen aus *abschnitt* und *listItem* auftreten.

## 2.2 Web Services

Scott Short schreibt: „Die komponentenbasierte Entwicklung ist beliebter als je zuvor“ (3 S. 1) und mittlerweile werden in der Tat kaum noch Anwendungen geschrieben, die nicht mit Hilfe von irgendwelchen Komponenten in Kommunikation treten. Dabei sind diese im Zusammenhang mit den Web Services nicht nur als Komponenten in einem lokalen System oder Netzwerk zu verstehen, sondern auch als Komponenten eines Fremdsystems.

Es gibt bereits bekannte Lösungen, wie *CORBA*<sup>14</sup>, *Java RMI*<sup>15</sup>, *DCOM*<sup>16</sup> oder *.NET-Remoting*<sup>17</sup> auf dem Markt, warum also eine Neue? Dem liegen die folgenden mehr oder minder schwerwiegenden Probleme gegenüber:

- *CORBA*, *Java RMI*, *DCOM* und *.NET-Remoting* Komponenten sind, bezogen auf den Wirkungsgrad, immer auf ein lokales Firmennetzwerk beschränkt.
- Alle sind auf freie Ports angewiesen, was sich in den meisten Unternehmen mit einem Netzwerkadministrator als äußerst schwierig erweisen wird.

Werden beispielsweise in einer E-Commerce-Anwendung Kreditkarteninformationen eines Kunden benötigt, so muss die Bank bzw. der Server einer Bank davon überzeugt werden, die nötigen Informationen zu liefern. In einem solchen Fall stoßen *CORBA* oder *Java RMI* an Grenzen.

In solchen Anwendungsfällen kommen die, von einem ganzen Konsortium um Microsoft, Sun, Oracle uvm., entwickelten *Web Services* ins Spiel, welche später durch das *W3C-Konsortium*<sup>18</sup> standardisiert wurden. *Web Services* sind Anwendungen, die bestimmte Aktivitäten im Namen eines Clients aufrufen. D.h. der *Web Service* dient quasi als Schnittstelle zwischen dem Client und dem Server. Hier wird über gängige Webstandards, wie dem verbindungslosen *UDP-Protokoll*<sup>19</sup> und dem Port 80, auf den *Web Service* zugegriffen, um die damit verbundenen Probleme zu umgehen.

Warum ist dieser Themenbereich für diese Bachelorarbeit interessant? *Web Services* finden eine häufige Anwendung in Verbindung mit *InfoPath* (siehe [2.3 Microsoft Office InfoPath 2007](#)). *InfoPath* beschäftigt sich mit XML-basierten Formularen und Formularvorlagen. Da *Web Services* XML verstehen, nutzt *InfoPath* häufig die Funktionalitäten von *Web Services*. *InfoPath* wiederrum findet oft Anwendung in *Workflows* und hier schließt sich der Kreis.

### 2.2.1 Web Service Standards

Da *Web Services* auf den offenen *W3C*-Standards beruhen, ist es möglich beispielsweise *Java* Applikationen mit *ASP.NET*<sup>20</sup> Applikation über einen *Web Service* kommunizieren zu lassen. Neben *HTML* und *XML* gibt es drei weitere wichtige Standards in der *Web Service* Welt (siehe [Tabelle 1](#)):

Standard	Beschreibung
<b>WSDL (Web Service Description Language)</b>	WSDL ist eine XML-basierte Sprache, die dazu dient, die Schnittstellen eines <i>Web Services</i> zu veröffentlichen. Mit Hilfe von WSDL werden Parameter und Methoden eines <i>Web Services</i> übertragen und somit Informationen darüber, wie an welche Daten zu kommen ist.
<b>SOAP (Simple Object Access Protocol)</b>	„Wenn man XML als die Sprache von Webdiensten bezeichnet, dann ist SOAP die Grammatik.“ (4 S. 363). D.h. mit Hilfe von SOAP kann eine Anwendung gezielte Abfragen mittels SOAP-Messages an einem <i>Web Service</i> stellen.

<sup>14</sup> *CORBA* steht für Common Object Request Broker Architecture und wird von der Object Management Group (OMG) entwickelt. Es dient der Spezifikation plattformunabhängiger Protokolle und Dienste

<sup>15</sup> *Java RMI* steht für Java Remote Method Invocation und lässt verteilten *Java*-Anwendungen zu

<sup>16</sup> *DCOM* steht für Distributed Component Object Model und ist ein von Microsoft entwickelter Standard um Komponenten die Netzwerkkommunikation zu ermöglichen

<sup>17</sup> *.Net Remoting* ist ein von Microsoft entwickeltes Framework zur verteilten Nutzung von Objekten

<sup>18</sup> *W3C* steht für World Wide Web Consortium, welches Standards für das World Wide Web festgelegt

<sup>19</sup> Das *UDP-Protokoll* steht für User Datagram Protocol und definiert eine paketorientierte Kommunikation für das Internet

<sup>20</sup> *ASP.NET* ist eine Microsoft Technologie, um Webanwendungen zu entwickeln

<b>UDDI (Universal Description, Discovery and Integration)</b>	UDDI ist ein standardisierter Verzeichnisdienst der Anbieter IBM, Microsoft und Ariba. Dieser Verzeichnisdienst enthält Adress- und Produktdaten sowie Anwendungsschnittstellen der verschiedenen Web Services, um an eben diese Daten zu gelangen.
--	---

Tabelle 1 - Web Service Standards Übersicht

### 2.2.2 Service-oriented Architecture (SOA)

Unter einer Service-oriented Architecture wird eine Abstraktion von Diensten verschiedener Mitarbeiter oder Organisationen eines Unternehmens, um diese Dritten anbieten zu können, verstanden. Ein wichtiger Bestandteil dieser Abstraktion sind die eigentlichen Geschäftsprozesse eines Unternehmens, welche gleichzeitig die nötigen Arbeitsschritte widerspiegeln und abwickeln. Die „Höhe“ einer Abstraktion hängt hierbei von dem jeweiligen anzuwendenden Geschäftsprozess ab.

Folgendes Beispiel soll den Sachverhalt besser verdeutlichen: Der Geschäftsprozess *einen Neukunden aufnehmen* ist wesentlich abstrakter als der Geschäftsprozess *Kunde A überweist einen Betrag n auf ein Konto x*. Beim ersten Geschäftsprozess spielen mitunter viele Mitarbeiter, Datenbanken oder Bonitätsprüfungen eine große Rolle, wobei beim zweiten Geschäftsprozess lediglich eine Person einen Betrag gutschreibt.

Ziel von SOA ist es, durch *Orchestrierung*<sup>21</sup> von Services niedrigster Abstraktionsebenen, abstraktere Services unter der Prämisse der bestmöglichen Wiederverwendbarkeit zu schaffen. So wird die im Hintergrund stattfindende Komplexität sozusagen maskiert und damit verborgen.

Eine technische Umsetzung von SOA ist es, diese abstrahierten Dienste über Protokolle wie SOAP im Internet anzubieten, womit die Überleitung zu den Web Services vollzogen ist. Wichtig ist: „SOA ist eine Idee, keine Technik.“ (5 S. 1) und somit auch kein Web Service an sich. SOA kann im Zusammenhang mit Web Services so verstanden werden, dass diese einem Web Service gegenüber steht und als Schnittstelle zu einer unternehmensinternen Architektur bzw. Geschäftsprozesswelt dient, um so beispielsweise bei einer Bank Kreditkarteninformationen auslesen zu können.

## 2.3 Microsoft Office InfoPath 2007

Mit Hilfe von InfoPath ist es möglich, über elektronische Formulare Informationen zu sammeln (6 S. 223). Diese Informationen können beispielsweise ein Urlaubsantrag oder eine Reisekostenerstattung sein.

Neu in der Version 2007 sind die webfähigen Formulare. Das Praktische für den Anwender hierbei ist, dass keine Clientsoftware vorausgesetzt wird, sondern lediglich ein installierter Browser. Über diesen können dann die Formulare einfach bearbeitet und gespeichert werden.

Formulare in InfoPath werden in XML übertragen (siehe [2.3.1 Formular vs. Formularvorlage](#)). Gespeichert werden diese in SharePoint in sogenannten *Formularbibliotheken*, welche speziell für die webbasierte Bearbeitung von InfoPath-Formularen entwickelt wurden. Von hier aus können diese jederzeit zwecks Bearbeitung geöffnet und wieder gespeichert werden.

Auch sehr nützlich ist die Definition von Regeln, nach denen ein Formular auszufüllen ist. D.h. die Qualität von Formularen steigt enorm und der Kommunikationsaufwand bei Problemen wird gesenkt.

<sup>21</sup> Von Orchestrierung wird in diesem Fall dann gesprochen, wenn viele Services zusammengesetzt werden sollen

### 2.3.1 Formular vs. Formularvorlage

In InfoPath gibt es zwei Arten von Dokumenten. Die erste ist das eigentliche Formular ([Abbildung 3](#)), welches komplett auf XML basiert und auch als XML-Dokument abgespeichert wird. Dieses Dokument resultiert aus der zweiten Dokumentenart - der Formularvorlage ([Abbildung 2](#)). Die Formularvorlage wird im XSD-Format gespeichert und dient der Individualisierung des Entwurfs eines Formulars. In diesem Dokument werden *Felder*, *Buttons*, *Ansichten*, *Auswahllisten* uvm. definiert, die in das XML-Formular übernommen werden. Das XML-Formular wird dann wiederum dazu verwendet, diese Informationen über Web Services zur Verfügung zu stellen.

Von der Formularvorlage wird mittels *Vorschau* in den Formularmodus gewechselt. Mit Hilfe von *Vorschau schließen* wird vom Formularmodus wieder in die Formularvorlage gewechselt.

Die [Abbildung 2](#) zeigt einen einfachen Urlaubsantrag. Dieser enthält zunächst die Felder *Titel*, *Von*, *Bis*, *Vorgesetzter* und die beiden Schaltflächen *Schliessen* und *Einreichen*. Der Freigabedialog, der nur Benutzern mit einer zugewiesenen Aufgabe sichtbar ist, enthält die Felder *Von*, *Kommentar* und zusätzlich die zwei Schaltflächen *Ablehnen* und *Genehmigen*, die zur Steuerung eines Workflows dienen. Es soll an dieser Stelle nicht alles vorweggenommen werden, da auf dieses Beispiel in aller Ausführlichkeit in Kapitel 4 eingegangen wird. Es soll hier zur Veranschaulichung von Formularvorlagen dienen.

The screenshot shows a form titled "Urlaubsantrag" (Vacation Request) with "Version 1.0" below it. The top right corner has two buttons: "Schliessen" (Close) and "Einreichen" (Submit). The form fields are as follows:

- Titel:** A text input field.
- Von:** A date selection field with a calendar icon.
- Bis:** A date selection field with a calendar icon.
- Vorgesetzter:** A text input field containing "Hans Admin".

The bottom section, titled "Freigabedialog" (Release Dialog), has a red background and contains the following elements:

- A message: "Ihre Freigabe ist für diesen Prozess nötig!" (Your release is necessary for this process!).
- Von:** A text input field.
- Kommentar:** A large text area for comments.
- Two buttons at the bottom: "Ablehnen" (Reject) and "Genehmigen" (Approve).

At the very bottom, there is a small box labeled "Abschnitt" (Section).

Abbildung 2 - Eine InfoPath-Formularvorlage

Neben der Formularvorlage gibt es zusätzlich noch das Formular (siehe [Abbildung 3](#)), in welchem die möglichen Informationen vom Anwender bearbeitet und gespeichert werden können:



Abbildung 3 - Ein InfoPath-Formular

Ergänzt werden diese beiden Formarmodi durch das Arbeiten im *Online-* bzw. *Offlinemodus*. Dies ist ebenfalls ein großer Vorteil gegenüber webbasierten Formularen, bei denen eine Offlinebearbeitung i.d.R. nicht möglich ist (4 S. 45). Dem Anwender wird so gestattet, das auszufüllende Formular zu verschiedenen Zeitpunkten zu bearbeiten und abzuspeichern, ob dieser in diesem Moment Zugang zum Internet hat oder nicht.

Abgeschlossen werden soll dieser Abschnitt mit einem kleinen Rückblick zum XML-Abschnitt. Folgend der von InfoPath generierte XML-Code zu dem in [Abbildung 3](#) gezeigten Formular, ohne noch einmal im Detail auf das XML-Dokument einzugehen:

```

XML-Code
XML-Dokument
<?xml version="1.0" encoding="UTF-8"?>
<?mso-infoPathSolution solutionVersion="1.0.0.54" productVersion="12.0.0"
  PIVersion="1.0.0.0"
  href="file:///C:\Documents%20and%20Settings\Administrator\Local%20Settings
    \Application%20Data\Microsoft\InfoPath\Designer2\bc43d5c59b4041fc
    \manifest.xsf" ?>
<?mso-application progid="InfoPath.Document" versionProgId=
  "InfoPath.Document.2"?>
<my:Urlaubsantrag
  xmlns:my="http://schemas.microsoft.com/office/infopath/2003/myXSD
  /2008-11-21T10:06:07" xml:lang="en-us">
  <my:Titel>Sommerurlaub</my:Titel>
  <my:Von xmlns:xsi="http://www.w3.org/2001
    /XMLSchema-instance">2009-07-27</my:Von>
  <my:Bis xmlns:xsi="http://www.w3.org/2001
    /XMLSchema-instance">2009-08-07</my:Bis>
  <my:Vorgesetzter>Hans Admin</my:Vorgesetzter>
  <my:Logik>
    <my:XMLFileUrl</my:XMLFileUrl>
    <my:WorkflowStatus>Init</my:WorkflowStatus>
    <my:AufgabeFuerBenutzer>>false</my:AufgabeFuerBenutzer>
  </my:Logik>
  <my:VorgesetzterSicht>
    <my:Mitarbeiter</my:Mitarbeiter>
  </my:VorgesetzterSicht>
</my:Urlaubsantrag>

```

Quellcode 11 - Resultierendes XML-Dokument aus dem vorherigen InfoPath-Formular

In [Quellcode 11](#) sind die PIs zu sehen, die eingebunden und verarbeitet werden sollen. Also Informationen über die XML-Version sowie einen Hinweis auf eine InfoPath-Nutzung. Direkt unter den PIs wird der eigene Namensraum definiert, welcher für die folgenden Elemente verwendet wird.

Farblich hervorgehoben sind die Informationen, die zuvor durch einen Anwender in das Formular eingegeben und an dieser Stelle in der XML-Hierarchie gespeichert wurden.

Wird dieses XML-Dokument editiert, wird es in InfoPath geöffnet und die entsprechenden Informationen ausgelesen. Das Resultat entspricht dann wieder dem Formular in [Abbildung 3](#).

### 2.3.2 Bedarfsermittlung

Im folgenden Abschnitt soll anhand von verschiedenen Szenarien gezeigt werden, wann ein Einsatz von InfoPath sinnvoll ist. Das hier zu Grunde liegende Problem ist, dass sich Unternehmen in der heutigen Zeit einer sich immer ändernden Unternehmenslandschaft gegenüber sehen (4 S. 40). Geschäftsprozesse werden komplexer, die Anforderungen an die Teamarbeit steigen und zentral verwertbares Unternehmenswissen gewinnt mehr und mehr an Bedeutung.

#### 2.3.2.1 Effiziente Teamarbeit

Heutige Tätigkeiten umfassen häufig ein vielfältiges Aufgabengebiet. Neben den Verpflichtungen im Unternehmen, stehen zusätzlich oft Termine beim Kunden direkt vor Ort an oder Aufgaben werden von zu Hause aus erledigt. Die immer vielseitiger werdenden Arbeitsbereiche stellen immer neue Anforderungen an die Unternehmen. Hierzu erweist es sich als nützlich, Informationen - in diesem speziellen Fall die Dokumente - auch im Rahmen einer *Extranet-Seite*<sup>22</sup> zur Verfügung zu stellen, damit Mitarbeiter oder auch Kunden auf Informationen zugreifen können (4 S. 41).

#### 2.3.2.2 Integration in Arbeitsprozesse

Auch ist es möglich einen Kunden in laufende Projektarbeiten einzubeziehen. Es können beispielsweise Umfragedokumente veröffentlicht werden, welche die Mitarbeiter dazu veranlassen, deren Sicht bzgl. Anforderungen an eine neue Unternehmenssoftware zu schildern. Das vereinfacht es zum Einen, spätere Probleme in möglichen Abnahmetests zu vermeiden und zum Anderen vereinfacht es dem Entwickler, den Kunden besser zu „verstehen“.

### 2.3.3 Kollaboration

Ein weiterer und wesentlicher Vorteil von InfoPath ist die Verwendung von XML. Da Geschäftsprozesse häufig mit vielen Softwareanwendungen zusammenarbeiten, ist hier also Flexibilität gefragt: Genau das erreicht InfoPath durch die Verwendung von XML. Folgend einige ausgewählte Programme, die mit InfoPath zusammenarbeiten können:

- Microsoft Office Excel 2007
- Microsoft Office Outlook 2007
- Microsoft Access 2000 oder höher
- Microsoft SQL Server 2000 oder höher
- Microsoft Windows SharePoint Services
- *Forms Services*<sup>23</sup>
- Web Services
- uvm.

### 2.3.4 Besonderheiten

Abschließen soll der Abschnitt über InfoPath mit ein paar allgemeinen Hinweisen bzgl. der Besonderheiten. Festgestellt wurden zwei wesentliche Besonderheiten von InfoPath: Zum Einen die Verwendung von XML und zum Anderen das Verarbeiten von Regeln, um so Eingaben steuern zu können.

---

<sup>22</sup> Eine Extranet-Seite bietet alle von einem Unternehmen freigegebenen Informationen. Auch haben hier oftmals Kunden einen Zugang. Bei einer Internet-Seite hingegen hat i.d.R. jeder Zugriff

<sup>23</sup> Hinter dem Begriff der Forms Services versteht man die Veröffentlichung von webbasierten InfoPath-Formularen. Das Ausfüllen der Vorlage funktioniert via Browser

Es gibt aber eine Vielzahl von weiteren guten Gründen, warum InfoPath eingesetzt werden kann. An dieser Stelle soll auf das Buch von *Eckbert Jeschke, Helmut Reinke* und *Sven Maier* verwiesen sein, die noch einmal ausführlicher die Alleinstellungsmerkmale von InfoPath hervorheben (4 S. 44).

## 2.4 SharePoint Plattform

In diesem Abschnitt soll ein wesentlicher und zentraler Bestandteil dieser Bachelorarbeit vorgestellt werden - die SharePoint Plattform. Diese bildet die Basis für das hier untersuchte und zu analysierende Workflow-Tool und verfolgt über dessen verwendete Mechanismen das Ziel, eine Service-oriented Architecture zu schaffen (7 S. 31).

SharePoint dient in erster Linie zur zentralen Verwaltung von unternehmensweiten Informationen. Dies können Dokumente sein, beispielsweise in Form von Verträgen oder Reise- und Urlaubsanträgen, aber auch Informationen in Form von Wikis, Blogs oder Diskussionsforen spielen eine Rolle. In diesen Fällen wird von einer Kollaboration gesprochen. Ein anderes wichtiges Stichwort ist also die *zentrale Wissensverwaltung* in einem Unternehmen, dessen Speicherung und Wiederverwendbarkeit.

Zu unterscheiden sind zwei grundsätzliche Formen, welche im Abschnitt *2.4.3 Varianten* noch deutlicher beschrieben werden: Zum Einem die *Windows SharePoint Services (WSS)* und zum Anderen der *Microsoft Office SharePoint Server (MOSS)*. WSS sind kostenloser Bestandteil des *Windows Server 2003*, aber nicht des *Windows Server 2008*. In diesem Fall ist eine separate Installation notwendig. Jedoch: Bei beiden Varianten wird lediglich eine Lizenz für den Server an sich fällig. Für einen MOSS entstehen zusätzlich noch die Lizenzkosten für das Produkt.

Bevor auf alle technischen Fertigkeiten SharePoints eingegangen wird, hier ein optischer Eindruck vom dem, was im Folgenden noch detailliert beschrieben wird:



Abbildung 4 - SharePoint Webseite im Nintex Workflow 2007 Testcenter

SharePoint bietet Anwendern also eine Art Webseite an, über welche ein Informationsfluss laufen kann. Auf der in *Abbildung 4* gezeigten Startseite könnten sich konkrete Informationen rund um das Unternehmen befinden, aber auch eigens eingefügte Inhalte eines Mitarbeiters.

### 2.4.1 Anwendungsdomänen

Auch wenn sich SharePoint im Vordergrund als eine Webseite präsentiert, wäre es zu einfach SharePoint als einfache Intranet-Anwendung, die die in einem Unternehmen wichtigen Informationen den Mitarbeitern zur Verfügung stellt, zu bezeichnen. SharePoint deckt wesentlich mehr ab. Auch die Bereiche Extranet und Internet werden von SharePoint abgedeckt und bilden neben dem Intranet ein entscheidendes Standbein von SharePoint. Hierbei kommt dann ein umfangreiches Rollenmanagement ins Spiel, welches sich um die Zugriffsverwaltung kümmert. Die *Tabelle 2* zeigt die drei Varianten Intranet, Extranet und Internet in einer Übersicht und grenzt diese voneinander ab:

Domäne	Beschreibung
<b>Intranet</b>	Enthält sämtliches unternehmensweites Wissen, auf das ein Mitarbeiter der Rolle entsprechend zugreifen kann. Es gibt keine nicht authentifizierten Zugriffe.
<b>Extranet</b>	Das Extranet enthält nur einen Teil des unternehmensweiten Wissens. Genau so viel, wie mit einem Partner geteilt werden soll. Nicht authentifizierte Zugriffe gibt es hier ebenfalls nicht. Eine Extranet-Anwendung ist vor allem immer dann interessant, wenn mit Partnern oder Kunden ein reger Dokumentenaustausch stattfindet.
<b>Internet</b>	Das im Internet verfügbare Wissen ist eine Untermenge des im Extranet vorhandenen Wissens (8 S. 35), da neben dem anonymen Internetbenutzer auch die Extranet-Benutzer hier Wissen beziehen können.

Tabelle 2 - Abgrenzung der SharePoint Anwendungsgebiete

### 2.4.2 Themenbereiche

Mit welchen Themenbereichen in einem Unternehmen beschäftigt sich SharePoint ganz konkret?

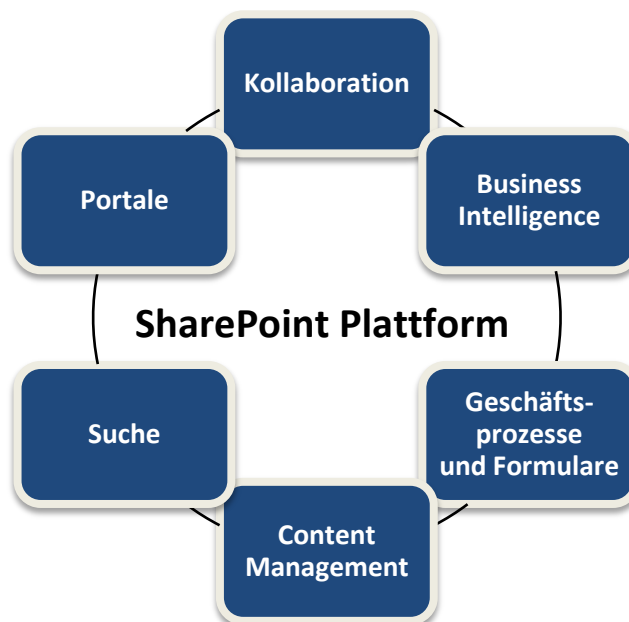


Abbildung 5 - Themenbereiche von SharePoint

Die in *Abbildung 5* gezeigten sechs Themenbereiche bilden das Grundgerüst von SharePoint. Aus diesem „Pool“ können die gewünschten Funktionalitäten ausgewählt und integriert werden. Um ein Gefühl für die einzelnen Bereiche zu entwickeln, hier einige kurze Definitionen:

- **Kollaboration**  
Unter der Kollaboration wird im Zusammenhang mit SharePoint beispielsweise die Zusammenarbeit an Dokumenten oder das gemeinsame Nutzen von Foren bzw. Blogs verstanden. Ziel ist es die Kommunikation über eine digitale Kommunikationsplattform zu steigern und Wissen dadurch zu zentralisieren.
- **Portal**  
Dem Bereich Portal werden die Integration von Office Programmen in SharePoint, wie Microsoft Excel, Microsoft Word oder Microsoft InfoPath, aber auch die benutzerspezifische Anzeige von Informationen im Portal selbst zugeordnet. Die Integration von Excel sowie InfoPath sind seit dem MOSS 2007 neu dabei.
- **Suche**  
Der Bereich Suche umfasst alle nötigen Maßnahmen für die effiziente Suche nach Inhalten oder Personen in einer SharePoint Infrastruktur. Das beinhaltet auch eine Bewertung der Suchergebnisse nach deren Relevanz. Durch die in der Enterprise Version enthaltene Suche, kann nach Inhalten in einer gesamten Unternehmensstruktur gezielt gesucht werden.
- **Content Management**  
Durch das Content Management werden Themengebiete wie das Archivieren von Inhalten nach vorgegebenen bestimmten Richtlinien, *inhaltsbezogene Workflows*<sup>24</sup> oder das Veröffentlichen von Inhalten abgedeckt.
- **Geschäftsprozesse und Formulare**  
Dieser Themenbereich beschäftigt sich mit dem Abbilden von Geschäftsprozessen in Workflows, dem Digitalisieren von Dokumenten und deren Verwendung in einem Browser.
- **Business Intelligence**  
Business Intelligence befasst sich mit dem Abbilden von Berichten, der Integration von Daten aus Datenquellen unterschiedlichster Hersteller und dem Verwenden von webbasierten Excel-Mappen mittels der *Excel Services*<sup>25</sup>.

### 2.4.3 Varianten

SharePoint liegt in der aktuellen Version 2007 vor und hat laut Boddenberg (8) insgesamt zwei Vorgängerprodukte:

- Die *Windows SharePoint Services 2* und den *SharePoint Portal Server 2003*<sup>26</sup>
- Den *Content Management Server 2002*<sup>27</sup>

Der Content Management Server 2002 ist in die aktuelle SharePoint Version mit eingeflossen, womit sich die folgenden aktuellen Produkte unterscheiden lassen:

---

<sup>24</sup> Inhaltsbezogene Workflows sind Workflows, die beispielsweise ein Genehmigungsverfahren für die Freigabe eines Dokuments abbilden

<sup>25</sup> Excel Services ermöglichen das Arbeiten an einer Excel-Mappe mittels eines Browsers

<sup>26</sup> Vorgänger vom Microsoft Office SharePoint Server 2007

<sup>27</sup> Durch die Wurzeln vom Content Management Server 2002 ist SharePoint in der Lage, flexible Veröffentlichung von Informationen im Intranet, Extranet und dem Internet zu tätigen

- Windows SharePoint Services 3.0
- Microsoft Office SharePoint Server 2007, Standard
- Microsoft Office SharePoint Server 2007, Enterprise
- Microsoft Office SharePoint Server 2007 for Search, Standard
- Microsoft Office SharePoint Server 2007 for Search, Enterprise

Es gibt also fünf grundlegende Modelle, wie SharePoint in welchem Umfang eingesetzt werden kann. Hier gilt es in einem Unternehmen den exakten Bedarf zu ermitteln, da z.B. mit einem vollwertigen MOSS die Kosten sehr in die Höhe „schießen“. Reicht also der Einsatz von Diskussionsforen, Blogs, Wikis oder das einfache Verwalten von Dokumenten aus, dann sind die WSS zu empfehlen. Gehen aber die Anforderungen darüber hinaus, beispielsweise sollen InfoPath-Formulare verwendet oder mit Workflows gearbeitet werden, so bleibt ein vollwertiger MOSS nicht selten die letzte Möglichkeit. Für weitere Informationen zu diesem Thema siehe [B.2 MOSS- und WSS-Funktionalitäten im Vergleich](#) bzw. [2.4.3.4 Abgrenzung von MOSS 2007 zu WSS 3.0](#).

#### 2.4.3.1 Windows SharePoint Services 3.0 (WSS)

WSS sind eine kostenlose Komponente des Microsoft Betriebssystems Windows Server 2003 bzw. *Microsoft Small Business Server 2003*. Mit deren Hilfe ist es möglich, auf ASP.NET basierende zentrale Webseiten zu erstellen. Die wesentlichen Fähigkeiten von WSS liegen im Bereich der Kollaboration, also der Zusammenarbeit an Inhalten. Das können Listen, Dokumente oder auch Blogs und firmeninterne Wikis sein. Bei allen Anforderungen darüber hinaus macht der Einsatz eines MOSS Sinn.

#### 2.4.3.2 Microsoft Office SharePoint Server 2007 (MOSS)

MOSS basiert auf den WSS und erweitert diese noch einmal erheblich. Dieser nutzt die WSS zur Darstellung der von ihm angebotenen Dienste. Gerade beim Anbinden von Fremdsystemen, einer umfangreichen Administration des SharePoint Webs, Zusammenarbeit mit anderen Office Produkten oder der individuellen Anpassung an die firmenspezifische *Corporate Identity*<sup>28</sup> fehlt es den WSS an Leistungsfähigkeit.

Bzgl. der Lizenzkosten sei an dieser Stelle auf die *Microsoft Lizenzbeschreibungen*<sup>29</sup> verwiesen, da diese nicht weiter in dieser Bachelorarbeit verdeutlicht werden sollen. Es sei jedoch kurz erwähnt, dass die Kosten eines MOSS gegenüber WSS deutlich höher sind - bei WSS fallen lediglich die Lizenzkosten für einen Windows Server an.

#### 2.4.3.3 Microsoft Office SharePoint Server 2007 for Search (MOSS)

Diese Version des MOSS besitzt die volle Suchfunktionalität, ist also in diesem Punkt dem voll ausgestatteten MOSS ebenbürtig. Dafür besitzt der MOSS for Search jedoch keine weiteren Eigenschaften, die der normale MOSS im Leistungsumfang mitbringt. Dieser erweitert also die WSS um eine erweiterte Suche. Nach einer genauen Betrachtung der gesteckten Anforderungen, kann evtl. so auf einen voll leistungsfähigen MOSS verzichtet werden.

#### 2.4.3.4 Abgrenzung von MOSS 2007 zu WSS 3.0

Die [Abbildung 6](#) zeigt noch einmal alle wesentlichen Unterschiede zwischen den möglichen Versionen, hier mit dem Schwerpunkt WSS, MOSS Standard und MOSS Enterprise.

Gut zu erkennen ist, dass die WSS die Basis für alle weiteren Versionen bilden. Diese umschließen die Bereiche *Projektmanagement* und *Team Kollaboration*. Wie schon erwähnt, decken diese also

---

<sup>28</sup> Corporate Identity: Definition des gesamten Erscheinungsbildes eines Unternehmens. Darunter fallen: Festlegen von Farben, Layouts, Logos und Schriftarten

<sup>29</sup> <http://office.microsoft.com/en-us/sharepointserver/FX102176831033.aspx>

speziell den Bereich der Zusammenarbeit an Inhalten in einem Unternehmen ab bzw. bilden für eben diese Anwendungsfälle die Basis.

Die Standard Version des MOSS bringt Features wie *Workflows*, *Dokumentenmanagement* und ein *Sicherheitsmodell* mit. Der vollwertige MOSS in der Enterprise Edition bringt zusätzlich Features wie eine *Suche*, *E-Forms (InfoPath)* und das *Reporting* mit.

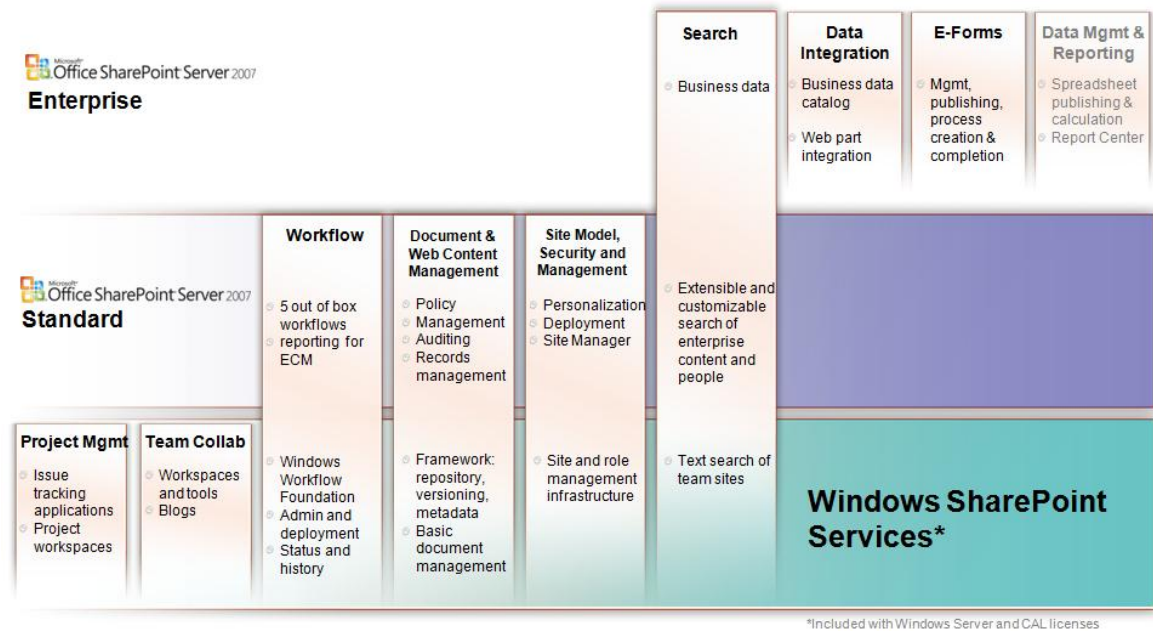


Abbildung 6 - Eine Übersicht der Unterschiede zwischen MOSS und WSS<sup>30</sup>

### 2.4.4 Gesamtzusammenhang

Nachdem die klaren Einsatzgebiete und Varianten von SharePoint beschrieben wurden, soll folgend SharePoint im Gesamtzusammenhang verdeutlicht werden.

Microsoft SharePoint ist nicht das Allheilmittel für sämtliche Unternehmensanforderungen. Es werden in Unternehmen zahlreiche andere Produkte eingesetzt, um entsprechend den Geschäftsabläufen zu handeln (8 S. 37). SAP wird beispielsweise zum Abwickeln von Geschäftsprozessen verwendet, Oracle um Datenbanken zu administrieren oder Lotus Notes um E-Mails zu verfassen. In diesem Kontext aber - der unternehmensweiten Wissensverwaltung - nutzt SharePoint eine zuvor nicht optimal gefüllte Lücke aus.

Die beiden folgenden Beispiele sollen diesen Aspekt untermauern:

Der *Microsoft Exchange Server* dient der Verwaltung von E-Mail-Nachrichten eines Unternehmens. Es erweist sich als schwierig, mit einem auf E-Mail-Kommunikation optimierten System Informationen beispielsweise in Form von Dokumenten zu verwalten. Es führt höchstens dazu, dass die Informationen das Postfach überfüllen oder nie wieder gefunden werden können, da ein Dokument umständlich beschriftet wurde.

Der *Microsoft SQL Server* dient der Speicherung von Informationen auf einer ganz anderen Granularitätsebene. Hier geht es darum, Informationen in Relation zueinander zu stellen, um so einen Informationsgewinn zu erhalten.

<sup>30</sup> Quelle: <http://blogs.technet.com/mikep/archive/2007/10/16/which-version-of-sharepoint-should-you-buy.aspx>

Um zu einer Entscheidungsfindung zu gelangen, ist es wichtig, sich gewisse Leitfragen zu stellen, um so konkrete und mögliche Anwendungsdomänen bzw. Themenbereiche analysieren zu können.

### 2.4.5 Leitfragen

In erster Linie gilt es die Prozesse im Unternehmen genauestens zu kennen und zu analysieren. Wann macht also ein Einsatz der Microsoft SharePoint Plattform Sinn? Um einem „Entscheider“ hier ein Stück weit zu unterstützen, wurden einige Leitfragen herausgearbeitet. Diese sollen dabei helfen, den Bedarf im Unternehmen zu erkennen und dazu anregen, mögliche neue Einsatzfelder kennenzulernen.

- Werden Dokumente vielfach dupliziert, sei es durch einen E-Mail-Versand oder durch das Kopieren auf Netzlaufwerken?
- Arbeiten Mitarbeiter häufig direkt vor Ort beim Kunden und brauchen einen Zugriff auf firmeninterne Informationen? (Mobilität heißt in diesem Zusammenhang mehr als nur einen E-Mail-Zugang oder ein Notebook zu besitzen.)
- Soll wertvolles firmeninternes Wissen dauerhaft digitalisiert werden?
- Werden Informationen handschriftlich notiert und gesammelt?
- Einsatz digitaler Formulare (InfoPath)?
- Sollen Geschäftsprozesse digitalisiert und automatisiert werden (Workflows)?
- Wird angestrebt die firmeninterne Kommunikation zu steigern, sei es durch Blogs, Wikis oder Diskussionsforen?
- Soll dieses Wissen indizierbar und durchsuchbar sein?
- Fremdsysteme sollen in die bisherige Firmeninfrastruktur integriert werden?
- Soll eine Office Software mit einem zentralen Dienst kommunizieren?
- Werden unternehmensweite Umfragen gewünscht, um so noch gezielter auf die Wünsche und Bedürfnisse der Mitarbeiter einzugehen?

Natürlich gibt es noch eine Menge weiterer interessanter Fragestellungen, doch der Kern der hier aufgegriffen Gedanken sollte erkennbar sein. Werden also mehrere dieser Fragen positiv beantwortet, sollte über einen Einsatz von SharePoint ganz klar nachgedacht werden. SharePoint bietet hier als zentraler Dienst eine Unmenge an Komfort und steigert bei einem angemessenen Gebrauch die Effizienz des Unternehmens. An dieser Stelle eignet sich ein Zitat des früheren Chief Executive Officer von Hewlett Packard *Lew Platt*: „Wenn HP wüsste, was HP weiß, dann wären wir dreimal so profitabel!“ (8 S. 33).

O'Connor und Buyens haben in deren Buch eine Vielzahl zusätzlicher Fragestellungen, die einem Kunden vor allem dabei behilflich sind, die genauen technischen Anforderungen im Unternehmen zu erkennen (6 S. 657).

### 2.4.6 Aufbau

Der letzte Abschnitt dieses Kapitels soll sich mit dem Aufbau einer SharePoint Plattform beschäftigen.

Eine SharePoint Website besteht aus drei wesentlichen Komponenten (siehe [Abbildung 7](#)). Die *Webanwendung* bildet den Grundriss bzw. das Korsett für eine SharePoint Anwendung. In diesem werden die *Webseitensammlungen* zusammengefasst und in diesen wieder die eigentlichen *Webseiten*.



Ohne eine Webseitensammlung macht ein *SharePoint Serverfarm*<sup>31</sup> für den Anwender wenig Sinn, da so keine Interaktion mit dem System stattfinden kann und keinerlei Informationen präsent sind. Im Fall der Fälle erhält der SharePoint Entwickler nach dem Anlegen der Webanwendung lediglich eine freundliche Fehlermeldung gezeigt, die darauf hinweist, dass eine entsprechende Webseitensammlung nicht angelegt wurde.

Auf den Webseiten befinden sich dann letzten Endes alle gewünschten Informationen, wie Wochenberichte, aktuelle Themen in den Diskussionsforen, Umfrageergebnisse oder Neuigkeiten. Eine SharePoint Serverfarm kann bis zu  $i$  Webanwendungen besitzen. Eine Webanwendung wiederum kann bis zu  $j$  Webseitensammlungen und diese wieder bis zu  $k$  Webseiten beinhalten.



Abbildung 7 - Architektur eines SharePoint Webs

Die beiden Bereiche Webseitensammlung und Webseite sind die Bereiche, die im Browser sichtbar sind. In der Webanwendung werden dagegen Konfigurationen getätigt. Diese Ansicht bleibt dem normalen Anwender verborgen.

#### 2.4.6.1 Inhaltstypen

Bevor folgend das Thema SharePoint verlassen wird, soll noch auf einen wichtigen Punkt eingegangen werden - den Inhaltstypen.

In Unternehmen gibt es eine Vielzahl von unterschiedlichen Dokumenttypen, seien es Verträge, Reisekostenabrechnungen, Tätigkeitsnachweise oder Urlaubsanträge. Die Dokumente werden im Allgemeinen immer wieder verwendet, da sich entsprechende Prozesse im Unternehmen immer wiederholen.

Ein Inhaltstyp ist eine Gruppe mit wiederverwendbaren Einstellungen, welche das Verhalten eines definierten Inhaltstypen bestimmen. Einen Inhaltstypen als *Template* zu bezeichnen ginge zu weit, aber das Verhalten ist an dieser Stelle sehr ähnlich!

Was bedeutet das? Ein Beispiel: Es soll ein Vertrag mit einem Kunden abgeschlossen werden. Dazu wird der fertige und fiktive Inhaltstyp *Vertrag* aus dem SharePoint System herangezogen. Es wird also dieser als Vorlage für den aktuellen Auftrag verwendet. Dieser Inhaltstyp definiert als Standard gewisse auszufüllende Felder, wie beispielsweise Kunde, Ansprechpartner, Projektleiter, Projektzeitraum, Umsatzvolumen uvm.

Was passieren kann ist, dass während des Verlaufs der Vertragsabwicklung der Punkt erreicht wird, an dem gewisse Sonderkonditionen im Vertrag mit aufgenommen werden müssen. Konkret bedeutet dies: Es müssen die Eingaben im digitalen Formular erweitert werden. Die Lösung in

<sup>31</sup> Eine SharePoint Serverfarm ist die Instanz, die eine Installation des entsprechenden SharePoint Produktes enthält und diese hostet: Es kann auch von einer SharePoint Infrastruktur gesprochen werden

diesem Fall ist, einen neuen Inhaltstypen zu definieren, der sich vom Inhaltstypen Vertrag ableitet und diesen um die gewünschten Felder erweitert.

Die [Abbildung 8](#) zeigt genau dieses Verfahren der Ableitung. In SharePoint ist der Inhaltstyp *System* der *Root-Inhaltstyp*<sup>32</sup> für alle ableitenden Inhaltstypen. Das wirkt sich auf die erbenden Inhaltstypen so aus, dass diese das *Titel*-Feld mit erben. In SharePoint kann dieses Feld nicht gelöscht, höchstens umbenannt und den Wünschen entsprechend angepasst werden (in diesem Fall ist es das Feld *Name*).

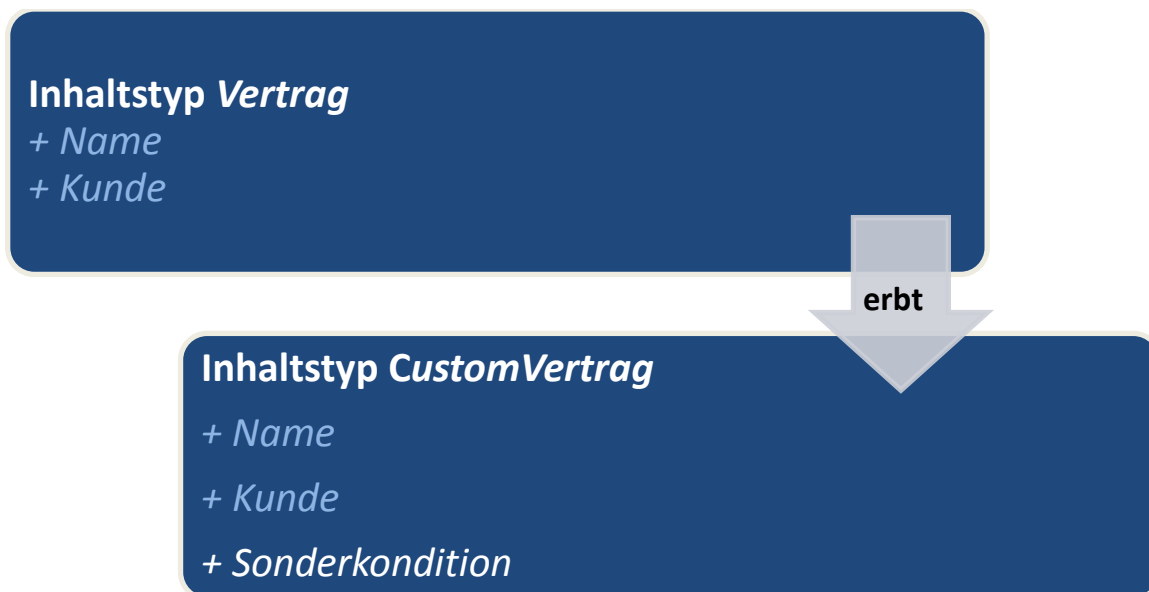


Abbildung 8 - Ein einfaches Vererbungsbeispiel von Inhaltstypen in SharePoint

<sup>32</sup> Der für alle ableitende Inhaltstypen als Basis dienende Inhaltstyp

# 3 Workflow

Im Rahmen von Nintex Workflow2007 fällt immer wieder der Begriff des *Workflows*. In dem folgenden Kapitel soll es darum gehen, dem Leser einen Eindruck davon zu vermitteln, was hinter diesem Begriff steckt.

Dazu wird das Kapitel mit grundlegendsten Begrifflichkeiten beginnen, welche klar voneinander unterschieden werden müssen. Gerade im Hinblick auf die Begriffe *Workflow*, *Prozess* und *Geschäftsprozess* ist es oft unklar, was konkret damit gemeint ist. Hier schwanken die Interpretationen, je nachdem, wie der Arbeitsprozess betrachtet wird. Es gibt zum Einen die Möglichkeit Arbeitsabläufe betriebswirtschaftlich und zum Anderen informationstechnisch zu betrachten.

Im zweiten Abschnitt dieses Kapitels, wird die Technologie besprochen, die SharePoint und Nintex Workflow 2007 als Basis zum Erstellen und Verwalten von Workflows nutzt. Im Rahmen von Workflow-Lösungen bei einer gleichzeitigen Nutzung von Microsoft Produkten, führt kein Weg an der hier beschriebenen Technologie vorbei.

Es gibt neben SharePoint auch andere Hersteller auf dem Markt, die auf Basis hauseigener Software eigene Workflow-Lösungen anbieten. Auf diesen Aspekt soll im vorletzten Abschnitt eingegangen werden.

Als letzter wichtiger Bestandteil dieses Kapitels geht es um das Szenario, anhand dessen im Kapitel 4 Nintex Workflow 2007 untersucht werden soll. Dazu wurden Anforderungen gestellt und der Ablauf des Arbeitsprozesses definiert.

Ziel dieses Kapitels ist es, dem Leser Kernbegriffe zu vermitteln, die SharePoint zu Grunde liegende Technologie zu verdeutlichen und das in der Bachelorarbeit fast durchgängig verwendete Beispiel näher zu bringen.

## 3.1 Begriffe

Im Zusammenhang mit dem Thema Workflows tauchen immer wieder einige wichtige Begriffe auf, welche geklärt und klar voneinander unterschieden werden müssen.

### 3.1.1 Prozess

Bei einem Prozess handelt es sich um eine Abfolge von unterschiedlichen Tätigkeiten, welche in irgendeiner Relation zueinander stehen. Kalenborn definiert den Begriff des Prozesses folgendermaßen: „*Ein Prozess ist eine inhaltlich abgeschlossene, räumlich und zeitlich angeordnete Menge von Aktivitäten, die in einem logischen Zusammenhang zueinander stehen.*“ (9 S. 51).

Ein Prozess kann materielle, als auch immaterielle Vorgänge beschreiben. Diese fassen in der Regel betriebswirtschaftlich relevante Objekte zusammen: dies können Objekte wie Rechnungen oder Aufträge sein. Generell beziehen sich diese auf geschlossene Erfüllungsgänge jeder Art.

### 3.1.2 Geschäftsprozess

Der Unterschied von einem Prozess zu einem Geschäftsprozess ist, dass ein Geschäftsprozess ökonomisch motiviert ist, d.h. konkret, dass ein Geschäftsprozess immer einen Input in einen Output transformiert. Dieser Output entspricht einer Leistung eines Unternehmens, wobei der Input die Anfrage eines Kunden darstellen kann.

Beide Begriffe bilden jedoch die Basis für jede Art der digitalen Erfassung der eben genannten Beispiele, welche dann im Zuge des Workflow-Managements (siehe [3.1.4 Workflow-Management](#)) stattfindet.

### 3.1.3 Workflow

Ein Workflow ist die digitale Umsetzung der zuvor beschriebenen Prozesse bzw. Geschäftsprozesse.

Dieser besitzt immer ein auslösendes Ereignis und ein definiertes Ende. Zwischen diesen beiden Zuständen ist ein Workflow aus Aktivitäten zusammengesetzt. Eine Aktivität kann beispielsweise das Einreichen eines bestimmten Formulars oder das Frei- bzw. nicht Freigeben einer Anfrage sein. Aktivitäten in einem Workflow können im Ganzen, in Teilen alternativ, parallel als auch sequentiell ausgeführt werden.

An dieser Stelle sei noch einmal der zentrale Unterschied von Workflows zu Geschäftsprozessen hervorgehoben. Geschäftsprozesse fixieren im Gegensatz zu einem Workflow die Erfüllung eines geschäftlichen Ziels eines Unternehmens (10 S. 8) - ein Workflow nicht immer zwingend, wie es auch bei einem einfachen Prozess der Fall ist.

In einigen Literaturquellen, werden die Begriffe *Workflow* und *Prozess* oft vertauscht. Dabei ist der Betrachtungskontext wichtig: Wird ein Ablauf aus betriebswirtschaftlicher Sicht betrachtet, wird eher von einem Prozess als von einem Workflow gesprochen. In der Informationstechnologie sähe das genau anders herum aus.

In diesem Abschnitt wurden bereits einige Ausführungen für Aktivitäten angedeutet. Im folgenden Abschnitt soll konkreter auf zwei wichtige Workflow-Arten eingegangen werden, die auch während des Erarbeitens dieser Bachelorarbeit eine wichtige Rolle spielten.

### 3.1.3.1 Sequentieller Workflow

Sequentielle Workflows zeichnen sich dadurch aus, dass die Aktivitäten sequentiell (also in einer festen Reihenfolge) durchlaufen werden. Ein ganz wesentlicher Nachteil der sequentiellen Workflows ist, dass ab einer bestimmten Komplexitätsstufe die Übersicht verloren geht. Es ist zwar generell möglich, viele Workflows sequentiell auszuführen, nur machen häufig die gleich folgenden zustandsbasierten Workflows mehr Sinn.

### 3.1.3.2 Zustandsbasierter Workflow

Zustandsbasierte Workflows zeichnen sich dadurch aus, dass diese nicht wie eine Sequenz aufgebaut sind, sondern gekapselt in einzelne Zustände. Das hat den großen Vorteil, dass in einem komplexen Workflow die Übersicht enorm gesteigert wird und die Logikanforderungen an die eigentliche Umsetzung sinken. Während des Bearbeitens der Beispielworkflows erwies sich so das Entwerfen von zustandsbasierten Workflows oft als große Hilfe.

Es muss aber fairerweise angemerkt werden, dass einfache Abnahmeworkflows nicht zwingend zustandsbasiert umzusetzen sind. Es macht vor allem dann Sinn, wenn beispielsweise viele Abnahmehierarchien eingebaut werden sollen.

Die [Abbildung 9](#) soll ein einfaches Beispiel zeigen, in dem der Sinn und Zweck eines zustandsbasierten Workflows klarer wird. Dieses Beispiel wird im Abschnitt [3.4 Das Referenzbeispiel](#) noch einmal aufgegriffen. Die dunkelblauen Felder in [Abbildung 9](#) markieren die einzelnen Zustände, die hellblauen die eigentlichen Aktionen und Bedingungen.



Abbildung 9 - Ein einfacher zustandsbasierter Workflow

Wird von einem komplexeren Workflow ausgegangen und das Beispiel aus [Abbildung 9](#) sequentiell gelöst, so ergeben sich viele redundante Aktionen und Abfragen, da ein Zustandswechsel in diesem Fall nicht möglich ist. Beispiel: Lehnt ein Vorgesetzter einen Antrag stets ab (Bedingung v), so muss dieser fortlaufend von einem Mitarbeiter überarbeitet werden. Nach jedem Überarbeitungsschritt muss ein Vorgesetzter diesen wieder genehmigen bzw. ablehnen (Bedingung v wird wiederholt). Dadurch kann eine Sequenz bis ins Unendliche anwachsen, weshalb sich ein solches Szenario sequentiell nicht darstellen lässt.

Das Praktische am zustandsbasierten Workflow ist zudem, dass klar einer Rolle entsprechend Funktionalitäten gekapselt werden können und somit die Übersicht und das Verständnis gesteigert werden.

### 3.1.4 Workflow-Management

In der Informationstechnologie ist ein Trend abzusehen, bei welchem zunehmend Funktionalitäten aus Anwendungen in ein zentrales Management System ausgelagert werden. Das wohl bekannteste Beispiel sind die *Datenbank Management Systeme (DBMS)*<sup>33</sup>.

Da im Zusammenhang mit der Ablaufsteuerung von Informationssystemen eine ähnliche Entwicklung wie beim DBMS zu beobachten ist, wird hier eine eigene Lösung notwendig. Diese Lösung ermöglicht es, auch in immer komplexer werdenden Informationssystemen, Konzepte und Technologien zur Verfügung zu stellen, auf die beim Arbeiten mit Workflows zurückgegriffen werden können. Das Workflow-Management stellt den äußeren Rahmen für eine Workflow-Anwendung zur Verfügung.

#### 3.1.4.1 Aufgaben

Dem Workflow-Management werden vier wesentliche Aufgaben zugeordnet:

- **Spezifikation von Workflow-Modellen**  
Sie findet unternehmensspezifisch statt, wobei ein Workflow-Modell die Bearbeitungsgrundlage (unternehmenstypische Regeln) für eine Menge von Vorgängen bieten muss.
- **Konfiguration einer Workflow-Anwendung**  
Die Konfiguration von Workflow-Anwendungen erfolgt auf Basis der Workflow-Modelle und konfiguriert einen Workflow für dessen Durchlauf mit entsprechenden Startparametern.
- **Steuerung der Vorgangsbearbeitung**  
Die Steuerung erfolgt durch eine Workflow-Anwendung. Diese dient im Wesentlichen dazu, entlang von Vorgängen die Bearbeitung der einzelnen Vorgangsschritte bzw. Funktionen anzustoßen und/oder den Datenfluss zwischen diesen Vorgangsschritten zu lenken.
- **Überwachung/Kontrolle der Vorgangsbearbeitung**  
Die Überwachung bzw. die Kontrolle geschieht auf der Grundlage von Kontrolldaten, welche im Zuge der Workflow-Steuerung ermittelt wurden. Diese Kontrolldaten machen eine Aussage über den Status eines Workflows, die Dauer der Workflow-Bearbeitung, wer eine Aufgabe bearbeitet bzw. noch nicht bearbeitet hat, sowie das Auftreten von Ausnahmefällen.

#### 3.1.4.2 Ziele

Dem Workflow-Management liegen folgende wesentliche Ziele zu Grunde:

- Steigerung der Qualität von Prozessen
- Reduzierung von Bearbeitungszeiten und die damit verbundenen Kosten
- Steigerung der Informationsverfügbarkeit
- Verbesserung der Flexibilität von Prozessen
- Erhöhung der Transparenz von Prozessen

---

<sup>33</sup> Ein Datenbank Management System verwaltet und regelt u.a. die Zugriffsrechte, Transaktionen oder die Konsistenz einer Datenbank

## 3.2 Workflow-Integration in SharePoint

In diesem Abschnitt soll explizit auf die von Microsoft umgesetzte Workflow-Lösung eingegangen werden, welche sich in die SharePoint Plattform integriert.

### 3.2.1 Windows Workflow Foundation (WF)

Mit der Windows Workflow Foundation bezeichnet Microsoft die hauseigene Lösung für Workflow-Prozesse und bildet somit die Basis für die digitale Abbildung von Geschäftsprozessen. Hierzu stellt die Workflow Foundation APIs, als auch Tools zu Verfügung, um mit diesen Workflows zu kreieren. Solche Tools können beispielsweise das Visual Studio sein, welches die Funktionalitäten der Workflow Foundation integriert.

Bei der Workflow Foundation handelt es sich um ein erweiterbares Framework, das speziell auf das Entwickeln von Workflow-Lösungen auf Windows-Plattformen spezialisiert wurde. Fertige Lösungen, die auf der Workflow Foundation basieren, bestehen aus verbundenen Komponenten und sind in .NET programmiert. Die Workflow Foundation unterstützt sowohl sequentielle als auch zustandsbasierte Workflows.

Die *Abbildung 10* zeigt die Bestandteile der Architektur der Workflow Foundation. Diese besteht aus ausführbaren Aktionen, dem Workflow-Designer, einer Rules Engine und einer Runtime.

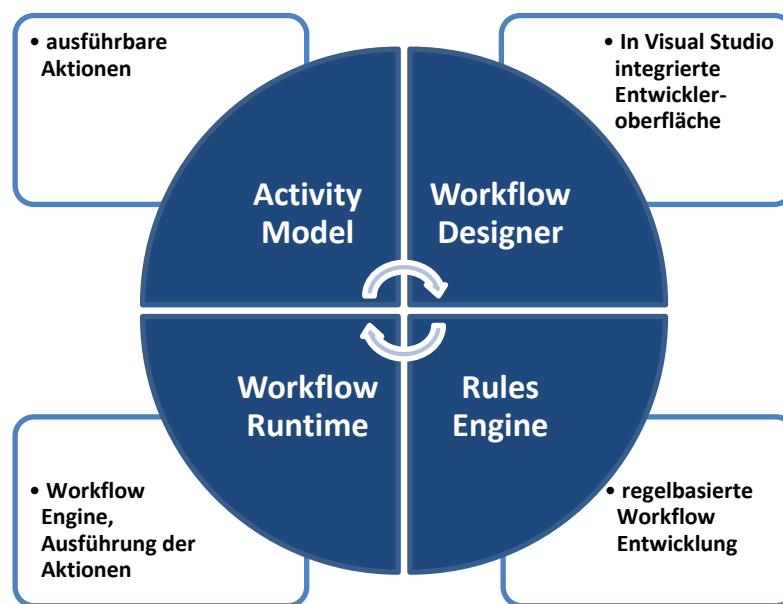


Abbildung 10 - Architektur der Windows Workflow Foundation

## 3.3 Alternativen

SharePoint ist zwar der zentrale Bestandteil dieser Bachelorarbeit, jedoch gibt es auf dem Markt ähnliche Lösungen, welche sich auch mit dem Thema *Workflow* auseinandersetzen.

Im Rahmen dieses Kapitels soll nicht explizit auf diese Alternativen eingegangen, jedoch auf deren Beschreibung in Kapitel 6 verwiesen werden.

## 3.4 Das Referenzbeispiel

In diesem letzten Abschnitt soll auf das Referenzbeispiel eingegangen werden, an welchem Nintex Workflow 2007 gemessen werden soll. Auch soll geklärt werden, warum der hier gewählte Workflow ein geeignetes Beispiel dafür ist, die Funktionsweise von Nintex Workflow 2007 zu verdeutlichen.

### 3.4.1 Urlaubsantrag

Im Rahmen dieser Bachelorarbeit wurde sich mit der HanseVision auf den Prozess eines Urlaubsantrags geeinigt. Warum ist ein Urlaubsantrags-Workflow ein geeignetes Beispiel, um die nötigen Informationen der Workflow-Lösung zu entnehmen und warum ist dieser nicht so simpel, wie es auf den ersten Blick erscheinen mag? Hierzu gibt es im Wesentlichen drei Punkte:

- Der Prozess eines Urlaubsantrags kommt in jedem Unternehmen vor
- Die Komplexität ist überschaubar, bietet aber eine Fülle an verschiedenen Arbeitsschritten, wie das Versenden einer E-Mail oder das Eintragen der Antragsdaten in eine Datenbank uvm.
- Da die Komplexität beliebig skalierbar ist, können relativ einfach die Fähigkeiten von Nintex Workflow 2007 geprüft werden

#### 3.4.1.1 Anforderungen

In Zusammenarbeit mit der HanseVision wurden folgende Anforderungen festgelegt:

- 2-stufiger Abnahmeprozess (Vorgesetzter und Personalabteilung)
- Verwendung von InfoPath zur Urlaubsdatenermittlung
- Archivierung der genehmigten Urlaubsdaten in einem *LOB-System*<sup>34</sup> über einen Web Service
- Eintragen der Urlaubsdaten in einen SharePoint Kalender
- Stellvertreterfunktion
- Überarbeitung von Anträgen muss möglich sein
- Archivierung von bereits genehmigten Anträgen
- Bearbeitungszeitfenster sollen verwendet werden

#### 3.4.1.2 Szenario

Generell gibt es drei wesentliche Rollen in dem Workflow: *Mitarbeiter*, *Vorgesetzter* und *Personalabteilung*.

Die auslösende Aktion ist das Stellen eines Antrags über ein InfoPath-Formular. In diesem Formular erfasst der Mitarbeiter alle Daten, die für einen Urlaub wichtig sind. Also von wann bis wann ist dieser im Urlaub, wer vertritt ihn in dieser Zeit usw.

Das Einreichen dieses Formulars in das SharePoint System eines Unternehmens, ist gleichzeitig der initiale Auslöser für den Workflow. Wurde das Formular eingereicht, erhält parallel ein Vorgesetzter einen Bescheid über das Vorhandensein eines neuen Urlaubsantrags, den dieser zu bearbeiten hat.

Damit ein Urlaubsantrag nicht verloren geht, wird dem Vorgesetzten ein Zeitfenster gesetzt, in welchem dieser einen Urlaubsantrag bearbeiten kann. Wird ein Vorgesetzter in dieser Zeit nicht aktiv, wird eine Benachrichtigung an die Personalabteilung geschickt, dass ein Antrag noch nicht bearbeitet wurde. Das System delegiert automatisch die Aufgabe an die Personalabteilung, welche sich dann dem Antrag widmen kann. Die Personalabteilung beendet den Workflow, in dem diese einen Urlaubsantrag genehmigt.

Natürlich sind beide Parteien nicht dazu verpflichtet, einen Antrag zu genehmigen bzw. abzulehnen, womit sich u.U. laufende Anträge im System häufen können. Im Szenario selbst ist aber nicht vorhergesehen, laufende Anträge nach einer gewissen Zeit  $t$  zu löschen oder in ein Archiv für nicht überarbeitete Anträge zu verschieben. Denn: Im Rahmen dieser Bachelorarbeit bieten solche Funktionalitäten wie auch das *Abbrechen* eines Workflows keinen Mehrwert, Nintex

---

<sup>34</sup> Ein LOB-System (line-of-business) kann eine Anwendung, wie beispielsweise SAP, sein



Workflow 2007 in neue Richtungen zu „stoßen“, um so noch mehr Erkenntnisse zu gewinnen. Im Regelfall wird also darauf vertraut, dass ein Vorgesetzter bzw. die Personalabteilung auf einen eingegangenen Antrag zeitgemäß reagieren.

Wird generell ein Antrag abgelehnt, wird dieser an den Mitarbeiter mit der Bitte zurückgewiesen, diesen zu überarbeiten. Wurde ein Urlaubsantrag ordnungsgemäß genehmigt, wird der Urlaub in einen SharePoint Kalender der SharePoint Anwendung übernommen, der Stellvertreter und der Mitarbeiter benachrichtigt, der Antrag archiviert und die Urlaubsdaten in ein LOB-System übernommen.

Die *Abbildung 11* verdeutlicht noch einmal den Vorgang des Genehmigens bzw. Ablehnens, wobei die grünen Pfeile den positiven Ausgang und die roten Pfeile den negativen Ausgang einer Bearbeitung darstellen sollen:

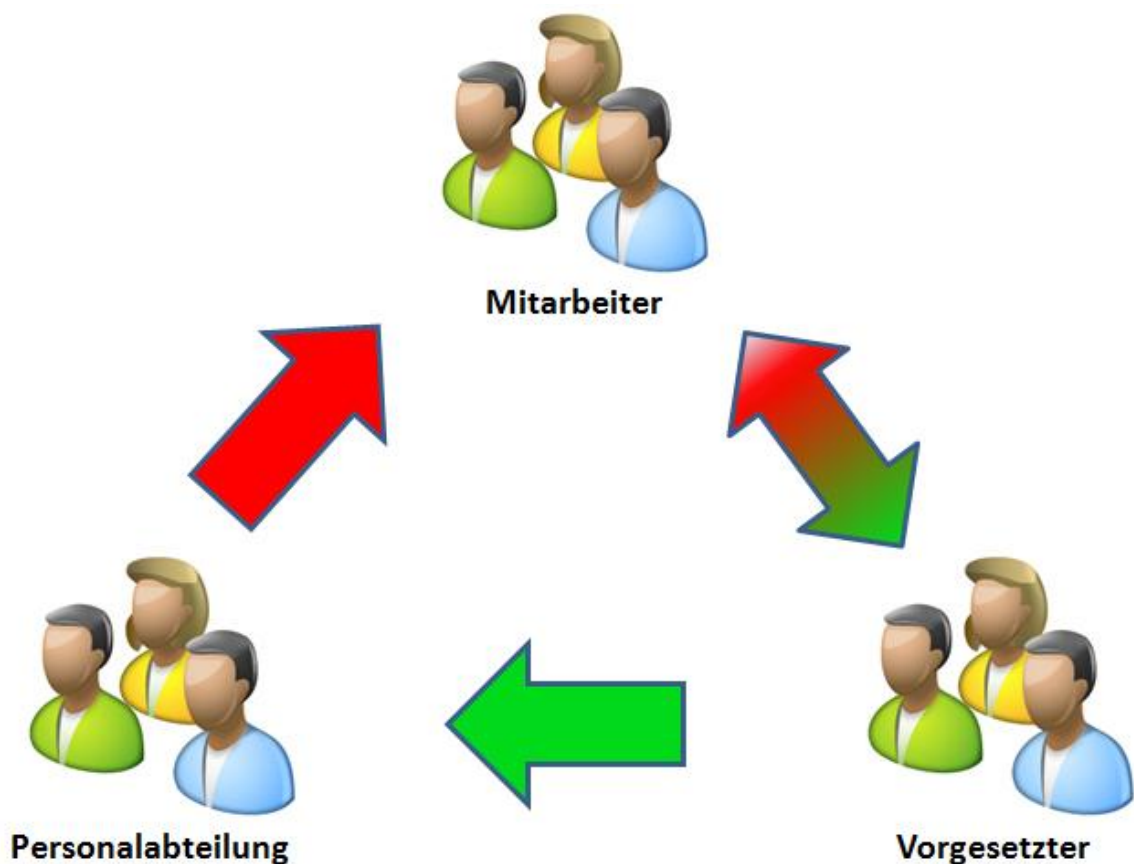


Abbildung 11 - Abnahmeprozess eines Urlaubsantrags

## 4 Nintex Workflow 2007

Dieses Kapitel widmet sich ausschließlich Nintex Workflow 2007 und greift die Inhalte aus dem vorhergegangenen Kapitel auf. Es soll hier im Schwerpunkt das untersuchte Produkt detailliert erläutert werden. Außerdem bildet es die Grundlage für das Kapitel 5, welches sich zu großen Teilen auf diese hier aufgeführten Informationen stützen wird.

In den folgenden Abschnitten sollen die grundlegenden Eckdaten und Informationen über Nintex Workflow 2007 präsentiert werden. Nicht betrachtet wird hingegen die umfangreiche Konfiguration der SharePoint Anwendung selbst, welche gleichzeitig die Basis stellt. Was hingegen explizit erwähnt werden soll, sind die Probleme, die während des Erarbeitens dieser Inhalte auffielen.

Das Ziel dieses Kapitels soll es sein, dass der Leser einen guten Eindruck davon bekommt, was mit Nintex Workflow 2007 möglich ist, gleichzeitig jedoch auch die Grenzen kennenlernt.

Zuletzt ein paar Worte zu der eingesetzten Testumgebung: Für das praktische Anwenden von Nintex Workflow 2007 ist es sinnvoll einen virtuellen PC zu verwenden. Hierzu kann beispielsweise das kostenlose Tool *Virtual PC*<sup>35</sup> verwendet werden. Gerade während der Einarbeitung half dies ungemein weiter, da so Konfigurationen und Einstellungen getestet werden konnten, ohne dabei in Gefahr zu laufen, das gesamte physikalische Hostsystem zu zerstören. Schlimmstenfalls wird ein neues virtuelles Laufwerk erstellt. Im Idealfall werden sauber installierte Systeme gesichert, welche je nach Belieben dann wieder eingebunden werden können.

---

<sup>35</sup> <http://www.microsoft.com/downloads/details.aspx?displaylang=de&familyid=04d26402-3199-48a3-afa2-2dc0b40a73b6>

## 4.1 Allgemeine Informationen

Es müssen einige grundlegende Dinge beachtet werden bzw. es helfen manche Einstellungen ungemein weiter, wenn diese von Anfang an berücksichtigt werden. In diesem Abschnitt soll auf diese „Stolpersteine“ hingewiesen werden. Dieser Abschnitt setzt eine erfolgreiche Installation des MOSS 2007 voraus. Um bei den später beschriebenen Beispielen auf weniger Probleme zu stoßen, sollten die folgenden Aspekte berücksichtigt werden:

### 4.1.1 Rollen

Da der Urlaubsantrag in dessen Logik immer einen Antragsteller und mindestens einen Vorgesetzten besitzt, ist es sinnvoll sich über Rollen im System Gedanken zu machen - also Rollen wie *Vorgesetzter* oder *Mitarbeiter* mit entsprechenden Rechten. Es empfiehlt sich zuvor im System eine vernünftige Unternehmensstruktur zu definieren. Dafür gibt es zwei Möglichkeiten: Das *Active Directory*<sup>36</sup> von Windows oder in SharePoint selbst. Es hat sich als praktisch erwiesen, eine einfache aber doch durchdachte Unternehmensstruktur im Active Directory des Testsystems angelegt zu haben.

### 4.1.2 Data Connection Libraries

Data Connection Libraries sind ein wichtiger Bestandteil von SharePoint. Diese sind immer genau dann nützlich, wenn Web Services zum Einsatz kommen sollen. Eine Data Connection ist immer ein XML-Dokument mit der Endung *\*.udcx* oder *\*.xml*.

Es sei folgendes Szenario gegeben: In einem Unternehmen sollen  $n$  neue Dokumente in das System integriert werden. Diese  $n$  Dokumente nutzen  $m$  Web Services. Jetzt liegt es auf der Hand, dass beide irgendwie miteinander kommunizieren müssen. Da aber die Infrastruktur im Unternehmen zu 99% anders aussehen wird, als in der Entwicklungsumgebung, müssten in allen Dokumenten physikalische Pfade zu den Web Services angepasst und geändert werden. Damit nicht alle Verbindungen und damit alle Dokumente geändert werden müssen, werden diese Datenverbindungen *konvertiert*. D.h. diese Verbindungen werden in den sogenannten *Connection Libraries* im SharePoint gespeichert. Es müssen also zukünftige Verbindungsänderungen lediglich in diesen einen zentralen Dokument getätigt werden.

Das Anlegen einer Data Connection Library ist also sehr empfehlenswert.

### 4.1.3 Forms Libraries

Die Forms Libraries dienen zur Verwaltung von InfoPath-Formularen. Dort können XML-Dokumente hinterlegt und mit Hilfe der *Forms Services*<sup>37</sup> im Browser editiert werden. Es hat sich als besonders hilfreich erwiesen, zumindest zwei verschiedene Forms Libraries anzulegen. Die eine beinhaltet die Formularvorlage und die andere die Formulare, welche nach und nach online gestellt werden.

### 4.1.4 Cross Domaining

In den SharePoint Einstellungen gibt es eine Option, die es InfoPath ermöglicht mit Web Services aus anderen Domänen zu arbeiten. Diese befindet sich in der Zentraladministration unter *Configure InfoPath Forms Services* und nennt sich *Allow cross-domain data access for user form templates that use connection settings in a data connection file*.

---

<sup>36</sup> Das Active Directory ist ein Verzeichnisdienst und verwaltet beispielsweise alle Benutzerkonten in einer Serverinfrastruktur

<sup>37</sup> Forms Services stellen das technische Gerüst, um SharePoint und InfoPath in Kommunikation treten zu lassen oder InfoPath-Formulare in Browser zu editieren

SharePoint Anwendungen laufen standardmäßig in einer eigenen Webanwendung. D.h. es gibt eine URL x, welche im Browser eingegeben werden kann und über die auf das SharePoint Web zugegriffen wird.

Das ist zunächst einmal nicht weiter schlimm. Soll aber ein InfoPath-Formular einen Web Service nutzen, der eine andere Webanwendung nutzt und eine eigene URL y besitzt, so erscheinen Authentifizierungsmeldungen im InfoPath-Formular, womit die angestrebte Lösung zunächst nicht umgesetzt werden kann.

Also: Laufen SharePoint Anwendungen und Web Services nicht in derselben Domäne, muss unbedingt diese Einstellung in der SharePoint Zentraladministration kontrolliert werden. Unter Umständen erspart dies eine stundenlange Recherche im Internet.

## 4.2 Das Tool

Dieser Abschnitt beschäftigt sich mit den einzelnen Facetten von Nintex Workflow 2007.

### 4.2.1 Lizenzen

Es gibt drei grundlegende Lizenzmodelle, zwischen denen gewählt werden kann. In [Tabelle 3](#) sind alle drei Modelle in der Übersicht mit den entsprechenden Kosten zu sehen:

Lizenz	Preis
Workgroup	1.900,-€ für maximal fünf Seitenaktivierungen
Standard	5.500,- € pro Server
Enterprise	11.000,- € pro Server

Tabelle 3 - Lizenzkosten von Nintex Workflow 2007

### 4.2.2 Partner und Kunden

Nintex baut auf ein breites Feld von Partnern und Kunden. Die Werte der [Tabelle 4](#) entstammen der Homepage von Nintex:

Art	Anzahl
Partner	60
Kunden	500

Tabelle 4 - Partner und Kunden von Nintex

### 4.2.3 Kollaboration

In diesem Unterabschnitt soll das Thema Kollaboration näher untersucht werden. Die [Abbildung 12](#) zeigt eine abstrakte Sicht der Integration von Nintex Workflow 2007:



Abbildung 12 - Kollaboration von Nintex Workflow 2007 mit anderen Programmen<sup>38</sup>

Besonders hervorgehoben wurde hier der BizTalk Server. Der BizTalk Server macht es möglich, mit *Microsoft-Fremd-Systemen* wie SAP in Kommunikation zu treten. Hier werden über genormte Standards Informationen ausgetauscht.

Warum ist das wichtig? In vielen Workflows werden oft Daten aus einem *Human Resources System (HR)*<sup>39</sup> benötigt. Solche Systeme sind häufig SAP-Systeme und sind somit in einer „fremden“ Technologie entwickelt worden und können nicht ohne Weiteres angebunden werden. In solchen Fällen ist es wichtig, flexibel zu bleiben und sich nicht auf eine Produktparte zu fixieren. Hier ist dann der Einsatz eines BizTalk Servers ratsam, welcher dem Kunden ein Höchstmaß an Flexibilität bietet und die Kommunikation von SharePoint und SAP gewährleisten kann.

Nicht explizit aufgenommen in der *Abbildung 12*, jedoch von der Marktstellung immer wichtiger ist: InfoPath. Dieses gliedert sich in den Bereich *Microsoft Office SharePoint Server 2007* in Form der Forms Services.

#### 4.2.4 Installation

Nintex Workflow 2007 lässt sich aus persönlicher Erfahrung einfach installieren und konfigurieren. An dieser Stelle sei auf die online zum *Download*<sup>40</sup> bereitgestellte Dokumentation verwiesen.

##### 4.2.4.1 Voraussetzungen

Folgende Komponenten müssen auf einem System installiert sein, um Nintex Workflow 2007 betreiben zu können:

- Windows 2003 Server
- Internet Explorer ≥ 6.x
- WSS 3.0 oder MOSS 2007
- Microsoft .NET Framework ≥ 3.x
- Microsoft SQL Server in den Versionen 2000 oder 2005

<sup>38</sup> Quelle: <http://www.nintex.com/en-US/Products/Pages/Workflow2007.aspx>

<sup>39</sup> Umschreibt Informationssysteme, die personenbezogene Daten beinhalten

<sup>40</sup> [http://nintexdownload.com/Nsupport/NW2007\\_Installation\\_guide.pdf](http://nintexdownload.com/Nsupport/NW2007_Installation_guide.pdf)

## 4.2.5 Integration

Nintex integriert dessen Funktionalitäten in verschiedene Bereiche der SharePoint Webseite. Folgend soll auf die einzelnen Bereiche eingegangen werden.

### 4.2.5.1 Zentraladministration

In der Zentraladministration werden zentrale und global gültige Einstellungen getätigt. Zu erreichen sind die Einstellungsmöglichkeiten über das Applikations-Management. *Abbildung 13* zeigt im unteren Bereich den Reiter *Nintex Workflow Management* in der SharePoint Zentraladministration:

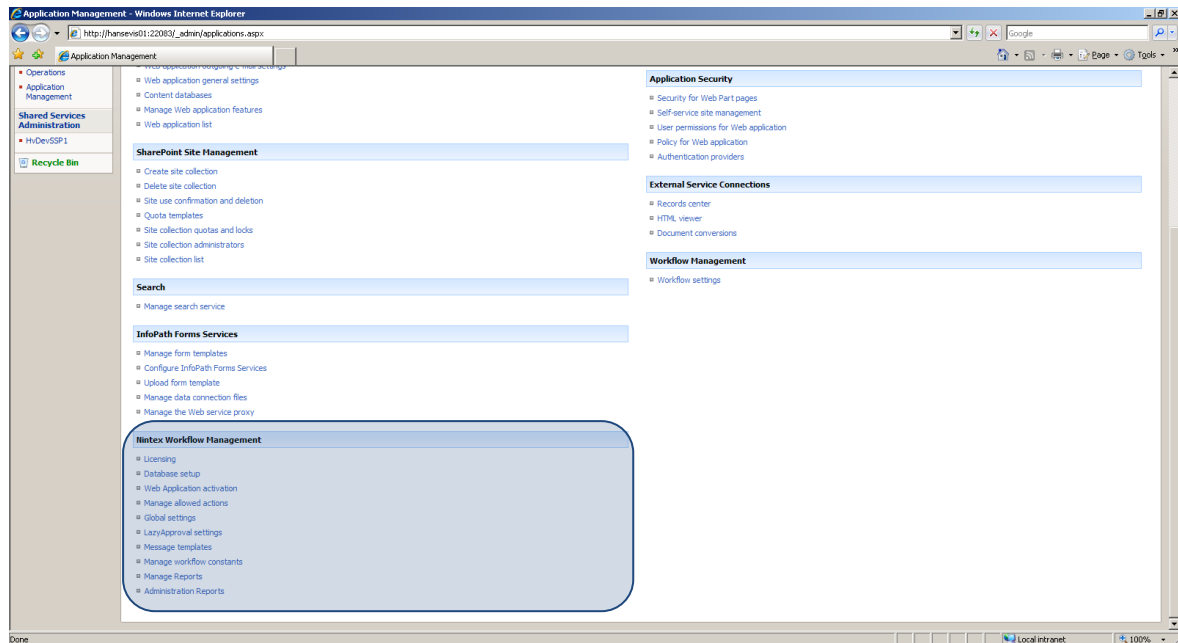


Abbildung 13 - Die Zentraladministration

Hier können folgende Einstellungen getätigt werden:

- Lizenzen
- Datenbank
- Aktivierung der Webanwendung
- Freigabe der erlaubten Aktionen
- Globale Einstellungen, wie beispielsweise E-Mail-Servereinstellungen
- LazyApproval Konfigurationen
- Nachrichten Templates
- Workflow-Konstanten
- Erlaubte Reports

### 4.2.5.2 Webseitenadministration

Die Webseitenadministration dient der Konfiguration von Nintex Workflow 2007 auf einzelnen Webseiten. Diese überschneiden sich mit den Zentraladministrations-Einstellungen zum Teil, nur sind diese dann im Rahmen genau einer Webseite gültig, nicht jedoch in der ganzen SharePoint Farm. Die *Abbildung 14* zeigt im unteren Bereich die Nintex-Einstellungsmöglichkeiten im Rahmen der Webseitenadministration:

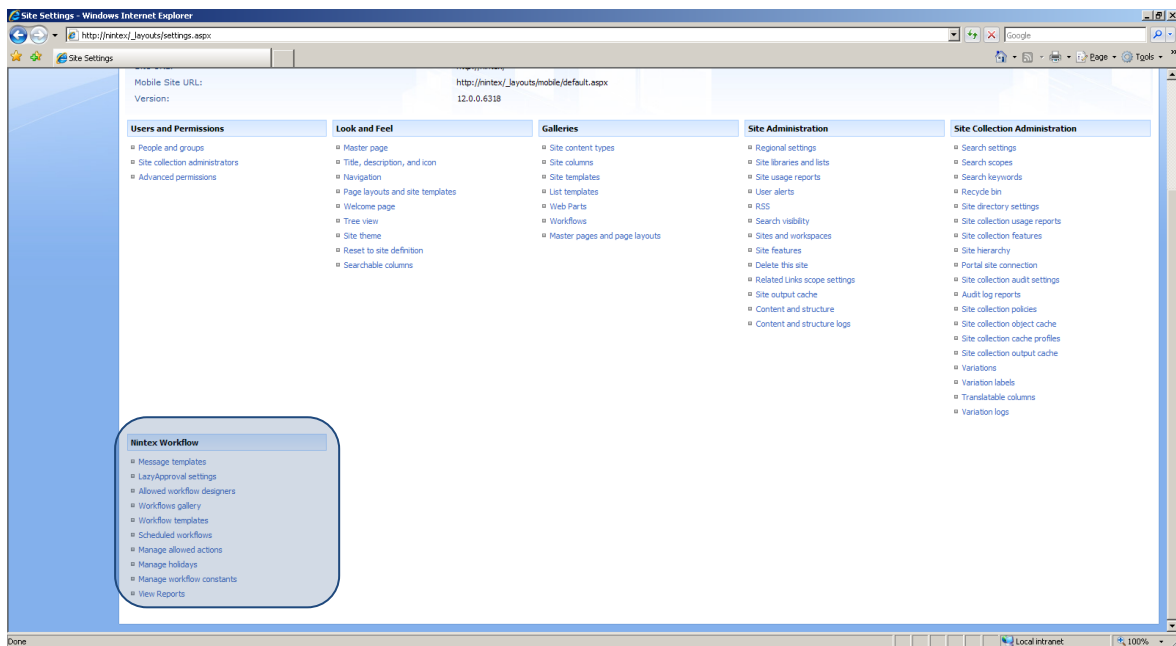


Abbildung 14 - Die Webseitenadministration

Hier können folgende Einstellungen gemacht werden:

- Nachrichten Templates
- LazyApproval Konfigurationen
- Erlaubte Workflow-Designer (Personen oder Gruppen)
- Workflow-Übersicht in Form von einer Workflow-Galerie
- Verwalten von Workflow-Templates
- Eine Übersicht von geplanten Workflows
- Freigabe der erlaubten Aktionen
- Urlaubseinstellungen zwecks Delegation
- Workflow-Konstanten
- Reports einsehen

#### 4.2.5.3 Benutzer-Menü

Über das Benutzer-Menü können, wie auch in *Abbildung 15* zu sehen ist, eingestellt werden:

- Der bevorzugte Weg der Benachrichtigung. I.d.R. wird die E-Mail-Benachrichtigung bevorzugt. Es sind aber auch SMS Benachrichtigungen möglich. Das muss hingegen die Serverstruktur hergeben und ist kein direktes Feature von Nintex.
- Es kann eingestellt werden, dass zu gewissen Zeitpunkten eine Aufgabe an einen Stellvertreter delegiert wird.

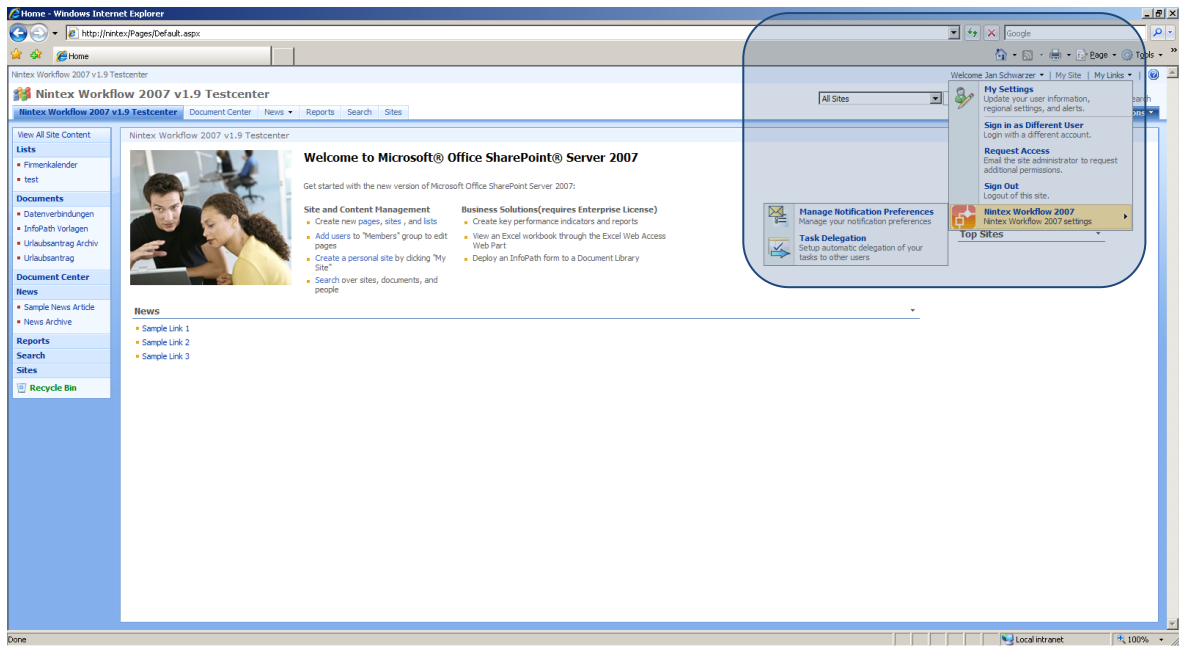


Abbildung 15 - Das Benutzer-Menü

#### 4.2.5.4 Settings-Menü

Über die Einstellungen in einem Settings-Menü in einer Liste oder einer Dokumentenbibliothek können Workflows hinzugefügt bzw. editiert werden. Zu erreichen ist dieses Menü wie in *Abbildung 16* zu sehen ist:

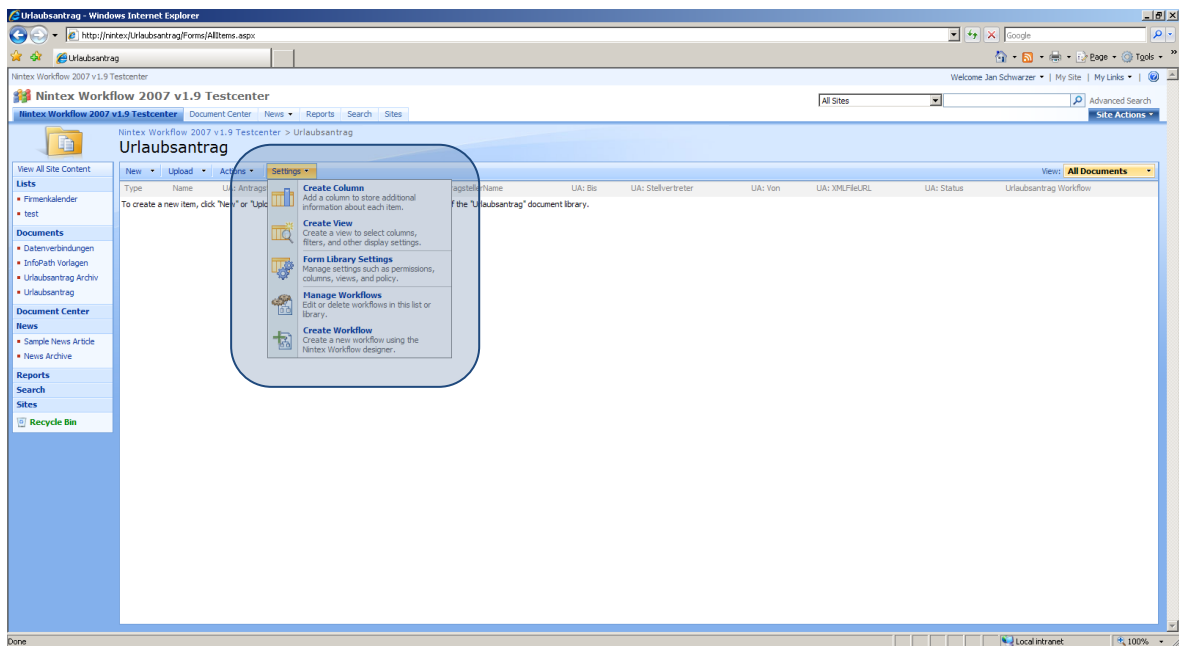


Abbildung 16 - Das Settings-Menü

#### 4.2.6 Entwicklungstools

In Nintex werden Workflows grundsätzlich über den integrierten Web Designer erstellt. Dieser soll folgend weiter beschrieben werden.



### 4.2.6.1 Web Designer

Workflows werden in Nintex Workflow 2007 ausschließlich im sogenannten *Web Designer* entwickelt. Aus persönlicher Erfahrung ist dieses Tool sehr intuitiv bedienbar und lässt den Anwender nach kürzester Zeit eigene Workflows bauen. Komplizierter wird es bei der Anbindung von anderen Diensten, wie der Integration von InfoPath-Formularen. Hier sind teilweise viele Einstellungen im Hintergrund zu beachten. Auf die wesentlichen Stolpersteine wird folgend noch weiter eingegangen.

Die *Abbildung 17* zeigt den Web Designer nach dem Erstellen eines leeren Workflows:

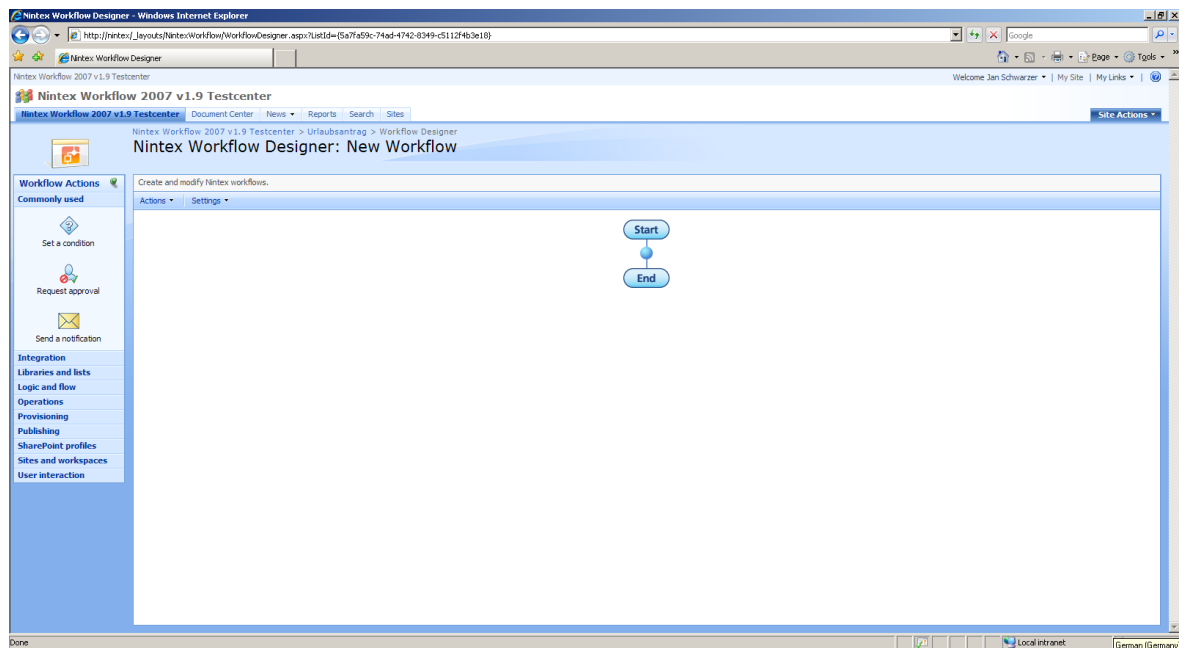


Abbildung 17 - Der Nintex Workflow 2007 Web Designer mit einem leeren Workflow

Für den Benutzer gibt es eigentlich nur zwei wichtige Dinge, die dieser beim Arbeiten mit dem Web Designer kennen sollte:

- Ganz links befinden sich die *Workflow Actions*, also die Aktionen, die einem zur Verfügung stehen. An dieser Stelle sei auf die komplette Aktionsliste im Anhang verwiesen (siehe *B.3 Nintex Workflow 2007 Aktionen*). Die *Abbildung 18* zeigt die Kategorie *Commonly used* als Beispiel:



Abbildung 18 - Das Aktions-Menü mit allen verfügbaren Aktionskategorien

- Direkt rechts der Workflow Actions, befinden sich die Menüs *Actions* und *Settings*, über welche grundlegende Konfigurationen gemacht werden oder der gesamte Workflow gespeichert bzw. veröffentlicht wird. Die *Abbildung 19* zeigt beide Menüs:

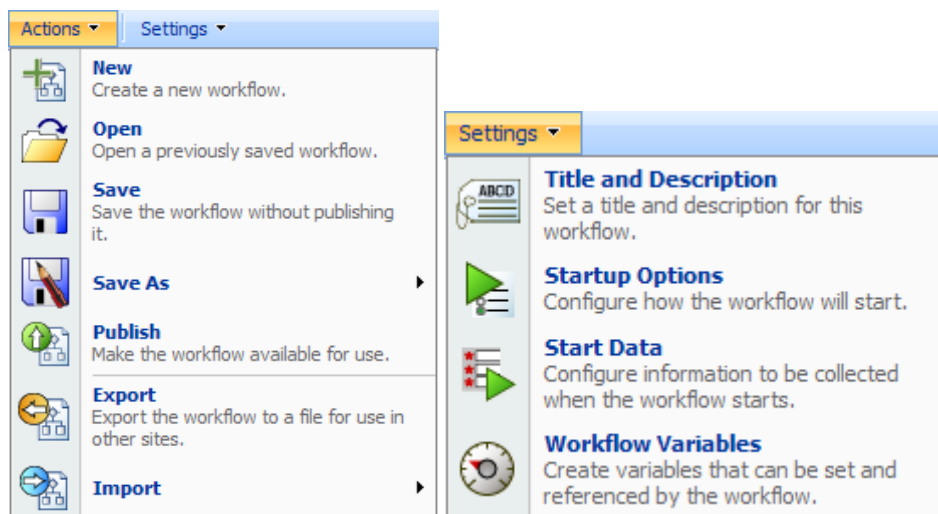


Abbildung 19 - Die Menüs Actions und Settings

Wirklich konfiguriert wird ausschließlich über das Settings-Menü. Hier können Startparameter oder -variablen definiert, sowie der Titel und die Workflow-Beschreibung gesetzt werden.

#### 4.2.6.2 Software Development Kit (SDK)

Nintex Workflow 2007 beherbergt zusätzlich die Möglichkeit, Aktionen in dem mitgelieferten SDK zu entwickeln. Es konnte aber bei den vielen Workflow-Anwendungsfällen, die im Laufe dieser Bachelorarbeit entstanden, kein explizites Beispiel ausgemacht werden, in dem nicht auch mit bereits vorhandenen Aktionen ausgekommen wurde. Dieser Aspekt muss Nintex zugestanden werden - es flossen viel Arbeit und Erfahrung in das Tool hinein, wodurch ein gleichermaßen für Standard- und Ausnahmefälle geeignetes Produkt entstanden ist. Trotzdem bleibt es dem Anwender offen, etwaige Problemstellungen in Eigeninitiative zu lösen.

#### 4.2.7 Reporting

Nintex ermöglicht es dem Anwender, Statistiken aufzustellen oder zugewiesene Aufgaben zu visualisieren. Die *Abbildung 20* zeigt verschiedene Reporting Möglichkeiten, die dem Anwender von Nintex Workflow 2007 *out-of-the-box* geboten werden. Das abgebildete Beispiel ist hier bewusst plakativ gewählt.

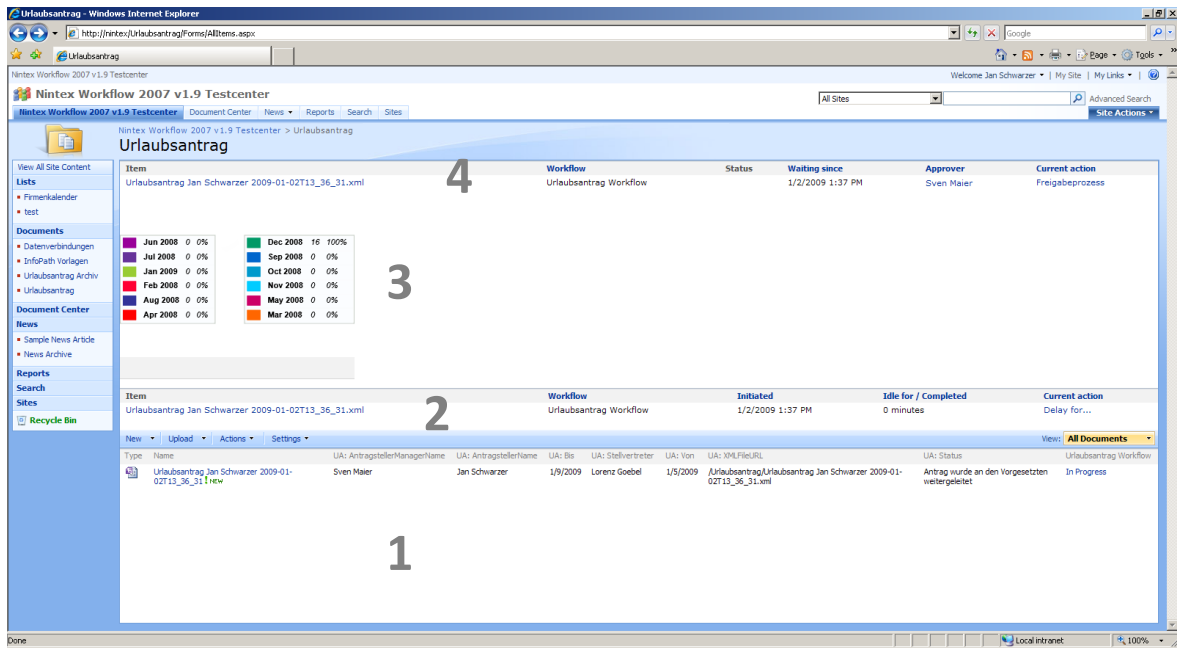


Abbildung 20 - Reporting Möglichkeiten mit Nintex Workflow 2007

Der **Abschnitt 1** in der **Abbildung 20** zeigt das eigentliche Dokument, welches eingereicht wurde und mit den nötigen Urlaubsinformationen gefüllt ist. Der **Abschnitt 2** zeigt alle Workflows an, die vom aktuell angemeldeten Benutzer gestartet wurden. Der **Abschnitt 3** visualisiert in diesem Beispiel alle Workflow-Aktivitäten der letzten 12 Monate. Der Zeitraum ist beliebig einstellbar. In **Abschnitt 4** sind alle zugewiesenen Aufgaben aufgelistet, welche in diesem Beispiel allen Benutzern zugewiesen wurden. Das macht natürlich nicht immer Sinn. Ein Mitarbeiter soll i.d.R. nicht wissen, was ein Vorgesetzter für Aufgaben besitzt. Der Einfachheit halber wurde diese Einstellung so gewählt, dass alle Aufgaben in einer SharePoint Farm, also von allen Anwendern, einsehbar sind.

Die Bereiche sind je nach Wunsch, ein- bzw. ausblendbar. Es hängt von der Rolle des Betrachters ab. Für Vorgesetzte sind Features wie dieses ein Mehrwert, für Mitarbeiter mag es hingegen eher nicht als notwendig erscheinen oder als ein „Kontrollfeature“ gehalten werden.

#### 4.2.8 Workflow-Historie

Ein sehr hilfreiches Feature von Nintex Workflow 2007 ist das Anzeigen einer Workflow-Historie. Diese Historie kann entweder tabellarisch oder graphisch betrachtet werden. Sie dient dazu, den aktuellen Stand eines Workflows anzuzeigen.

In Nintex Workflow 2007 gibt es in der graphischen Ansicht vier verschiedene Farbkennungen:

- Grün: Pfade die durchlaufen wurden
- Gelb: Aufgabe wurde noch nicht bearbeitet
- Rot: deutet auf einen Fehler hin oder im Regelfall auf das Ablehnen einer zugewiesenen Aufgabe (Aktion *Urlaubsantrag abgelehnt*)
- Grau: noch nicht durchlaufende Pfade

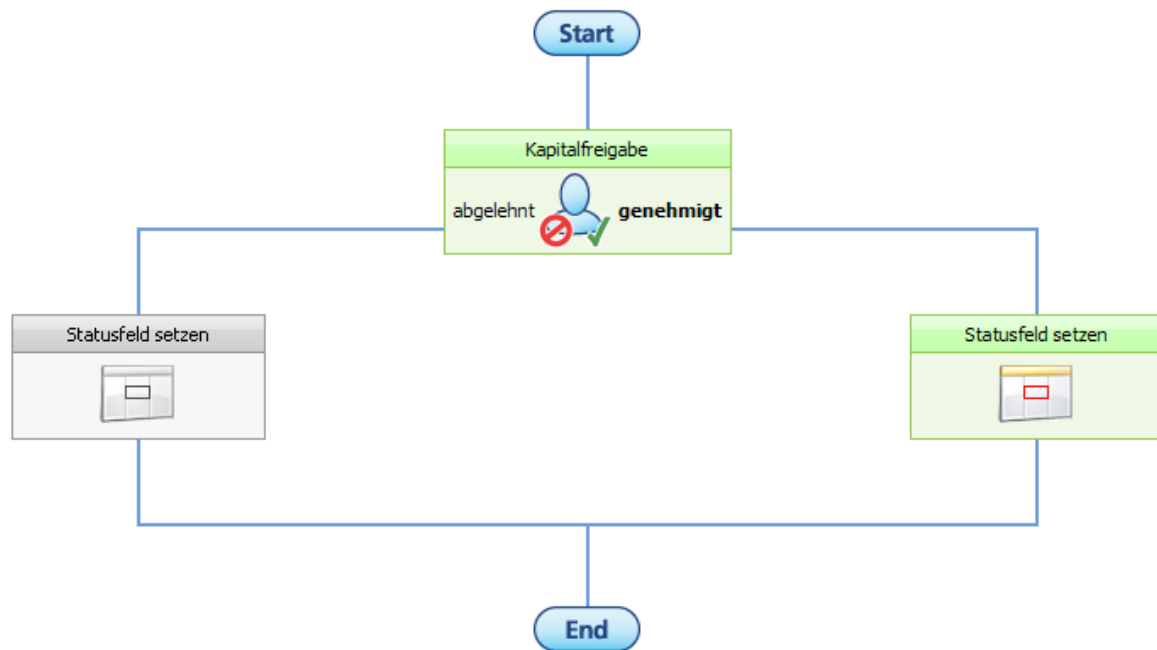


Abbildung 21 - Ein einfaches Historie Beispiel

Die [Abbildung 21](#) zeigt eine der Möglichkeiten, wie eine Historie aussehen kann. In diesem Fall wurde die Kapitalanfrage genehmigt, wodurch dieser Zweig des Workflows entsprechend durchlaufen wurde. Der Pfad über das Ablehnen des Antrags bleibt weiterhin grau, da dieser nicht durchlaufen wurde.

Im Fall einer Ablehnung würden sich die Aktion *Kapitalfreigabe* und der dazugehörige Pfad rot färben.

#### 4.2.9 Versionierung

Nintex Workflow 2007 bietet zusätzlich die Möglichkeit, Workflows zu versionieren. Das hat den Vorteil, dass Workflows die einmal erstellt wurden, nicht wieder verloren gehen. Die Workflows können also je nach Belieben wieder hergestellt werden und stehen dem Anwender stets zur Verfügung.

#### 4.2.10 LazyApproval

Ein besonderes Feature in Nintex Workflow 2007 ist das sogenannte *LazyApproval*. Deswegen wird diesem auch ein eigener Abschnitt gewidmet. Bevor weiter auf dieses Feature eingegangen wird, soll zuerst die [Abbildung 22](#) betrachtet werden:

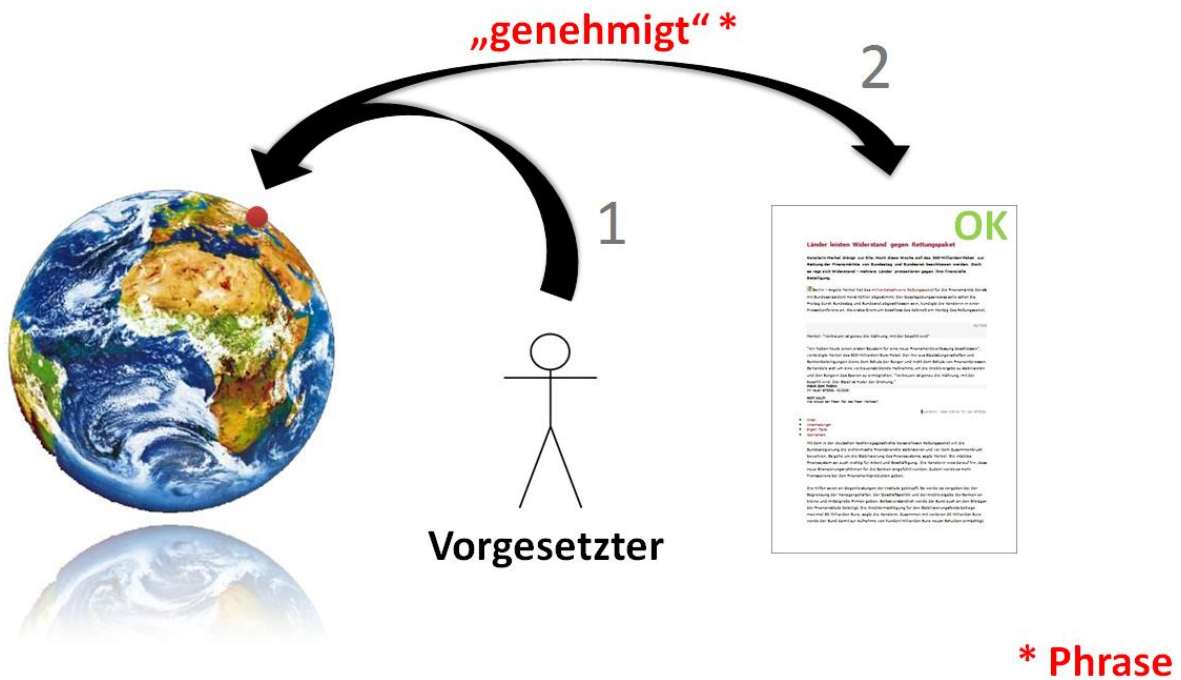


Abbildung 22 - Prinzip des LazyApprovals

Die **Abbildung 22** zeigt das Prinzip bzw. die Ablauffolge eines LazyApproval Vorgangs. Der Kern des Ganzen ist die Einflussnahme auf Workflows mit Hilfe von E-Mail-Nachrichten. Das funktioniert folgendermaßen:

Die **Eins** in der **Abbildung 22** symbolisiert den Anfang eines solchen Prozesses. Was diesem vorausgeht, ist die Zuweisung einer Aufgabe an einen Vorgesetzten aus einem Workflow heraus. Der Einfachheit halber soll der Vorgesetzte in diesem Fall einfach nur einem Kostenvoranschlag zustimmen. In der Regel würde der Vorgesetzte das SharePoint System aufrufen und dort auf die zugeteilte Aufgabe reagieren. Was ist aber, wenn dieser Vorgesetzte nicht im Inland ist oder gar keinen Zugang zu dem SharePoint System besitzt? In diesen Fällen würde ein Problem auftreten. Um dem vorzubeugen, entwickelte Nintex genau für solche Fälle eine Lösung - das LazyApproval.

Mit Hilfe dieses Features ist es dem Vorgesetzten möglich, mittels formatierten E-Mail-Nachrichten auf den Workflow Einfluss zu nehmen. In der **Abbildung 22** wird das durch die **Zwei** symbolisiert. Verwendet wird hierzu eine sogenannte **Phrase**. Phrasen sind in Nintex Workflow 2007 fest definierte Wörter die mit einem sogenannten **Outcome** assoziiert werden. Outcomes sind vom Nintex-System definierte Vokabeln, welches diese versteht und mit welchen es einen Prozess genehmigen bzw. ablehnen kann. Dieser Vorgang ist ebenfalls im 2. Schritt zu sehen. Die einzige Bedingung für die zu sendende E-Mail ist, dass in der ersten Zeile der Nachricht ausschließlich eine Phrase steht, die dann im System gesucht und deren Outcome ausgeführt wird.

Zur Verdeutlichung ein Beispiel:

Phrase	Funktion im System (Outcome)
Genehmigt	Approved
Abgelehnt	Declined

Tabelle 5 - Funktion einer Phrase und dem zugehörigen Outcome

In der [Tabelle 5](#) sind zwei Phrasen zu sehen. Wenn eine dieser beiden Phrasen in der ersten Zeile einer Antwort-Nachricht steht, wird der entsprechende Kostenvoranschlag genehmigt bzw. abgelehnt. Einer solchen Antwort-Nachricht geht eine E-Mail vom Nintex-System voraus, die einen Vorgesetzten drauf hinweist, dass dieser einem Prozess zuzustimmen hat.

Der Aufenthalt eines Vorgesetzten im Ausland oder das Fehlen eines Zutritts zum SharePoint System sind nicht der einzige Grund für das Verwenden von LazyApproval. Das ganze Prinzip funktioniert natürlich auch in einem internen Firmennetzwerk, in dem bereits ein Zugang zu einem SharePoint System vorhanden ist. Von hier aus kann, über die firmeninterne oder der privaten E-Mail-Adresse, Einfluss auf einen Prozess genommen werden.

Das Feature spart vor allem Zeit ein und vermeidet Logins oder umständliches Aufsuchen einer SharePoint Seite.

### 4.3 Beispiel: Urlaubsantrag

In diesem Abschnitt soll der Beispiel-Workflow aus [3.4 Das Referenzbeispiel](#) wieder aufgenommen und im Zusammenhang mit Nintex Workflow 2007 verdeutlicht werden.

Der Workflow ist recht umfangreich und kann auf einem Screenshot nur sehr undeutlich dargestellt werden. Es werden daher die einzelnen Teilabschnitte des Workflows visualisiert und im Gesamtzusammenhang verdeutlicht.

Als Basis für das Szenario dient ein zustandsbasierter Workflow, da dies die Übersicht enorm steigert und gleichzeitig die Verständlichkeit verbessert. Außerdem gäbe es in einer sequenziellen Lösung etliche Redundanzen, wie beispielsweise das *Beenden* des gesamten Prozesses.

Den Anforderungen entsprechend wurden vier Zustände implementiert, wobei der Zustand Vorgesetzter der initiiierende Zustand ist:

- Mitarbeiter
- Vorgesetzter
- Personalabteilung
- Ende

Auf die einzelnen Funktionalitäten bzw. Aktivitäten in den Zuständen, wird später noch eingegangen. Zuerst soll die nötige Vorarbeit im Workflow selbst erläutert werden.

Bevor die einzelnen Zustände besprochen werden, noch ein kurzer Hinweis auf eine immer wiederkehrende Aktion in den Zuständen - das Setzen eines Statusfeldes. Das Statusfeld hat zwei wesentliche Aufgaben:

- Es soll dem Anwender in der Standardansicht verdeutlichen, an welcher Stelle der eigene Workflow steht bzw. welchen Fortschritt es in dem Abnahmeprozess gibt. Hierzu kann natürlich auch die bereits beschriebene Historie gewählt werden, nur muss sich zu dieser erst einmal „*durchgeklickt*“ werden. Ein Feld in der Listen- bzw. Dokumentbibliothek ist in der Standardansicht sofort sichtbar und vermittelt dem Anwender einen direkten Eindruck.
- Der zweite und fast noch wichtigere Punkt ist, dass InfoPath mit Nintex Workflow 2007 nur über Listen- bzw. Dokumentbibliotheksfelder in Kommunikation treten kann. Das ist vor allem dann entscheidend, wenn in einem InfoPath-Formular Ansichten gewechselt, Bereiche ein- bzw. ausgeblendet oder Werte gesetzt werden sollen. Wie später beim Vorstellen des InfoPath-Formulars noch deutlicher wird, kann keine Interaktion beider Lösungen ohne diese besagten Felder zustande kommen.

Diese sind also Grundvoraussetzung! Das erklärt auch, warum in den Zuständen immer die Statusfelder entsprechend des Ablaufs gesetzt werden.

### 4.3.1 Start

Die *Abbildung 23* zeigt den Anfang des Workflow. Dem auslösenden Ereignis steht folgende Überlegung zu Grunde: Wird ein Dokument eingereicht, so wird der Workflow gestartet (Auslöser des gesamten Workflows). Der Workflow wartet auf Grund der Aktion *Auf das Einreichen eines Dokuments warten* auf eben genau diesen Vorgang. Gefolgt wird diese Aktion von einem sogenannten *Action Set*. Dieses Action Set beinhaltet folgende Aktionen:

- Setzen von Zugriffsrechten auf das eingereichte Dokument: Also Zugriffe Dritter sollen unterbunden werden.
- Zusätzlich wird noch ein Feld gesetzt, welches den physikalischen Pfad zu eben diesem XML-Dokument beinhaltet. Das hat den Hintergrund, dass erst zur Laufzeit der Pfad zum Dokument ermittelt werden kann und ein Workflow in diesem Fall an ein XML-Dokument gebunden ist.
- Als letztes werden noch Datumfelder gesetzt, um so zum Einen den beantragten Urlaub in einem SharePoint Kalender zu speichern und zum Anderen die Urlaubsinformationen in ein simuliertes SAP-System zu übertragen. Das funktioniert mittels eines Web Services, der diese Datumsinformationen als Übergabeparameter benötigt.



Abbildung 23 - Start des Workflows

### 4.3.2 Zustand: Mitarbeiter

Im Folgenden wird auf die Aktionen und Funktionalitäten des Mitarbeiterzustands eingegangen. Ein wichtiger Aspekt dieses Zustand ist es, dass dieser nur ausgeführt wird, wenn ein Urlaubsantrag überarbeitet werden muss, was auf Grund von Änderungswünschen der Personalabteilung bzw. des Vorgesetzten geschehen kann. Insofern gibt es im Zustand *Mitarbeiter* auch eine Aktion, die auf das Einreichen einer überarbeiteten Version des Dokuments wartet. Das Einreichen im Kontext des Mitarbeiter-Zustands ist aber nicht der Auslöser des gesamten Workflows (vgl. [4.3.1 Start](#)). Es dient lediglich dazu, den Abnahmeprozess wieder von vorne beginnen zu lassen, indem nach der Überarbeitung der Antrag wieder an den Vorgesetzten geleitet wird.

Aus Gründen der Übersicht wurden auch hier die Felder für das Setzen des Datums in einem Action Set zusammengefasst. Diese müssen wieder gesetzt werden, da sich die Urlaubstage durch das Überarbeiten evtl. geändert haben.

Wie in der [Abbildung 24](#) zu sehen ist, wird am Ende der Zustand wieder gewechselt. Vom logischen Aufbau her entspricht dies einem neuen Workflow-Durchlauf. Ein Vorgesetzter muss nach getätigten Änderungen den Antrag erneut abnehmen.

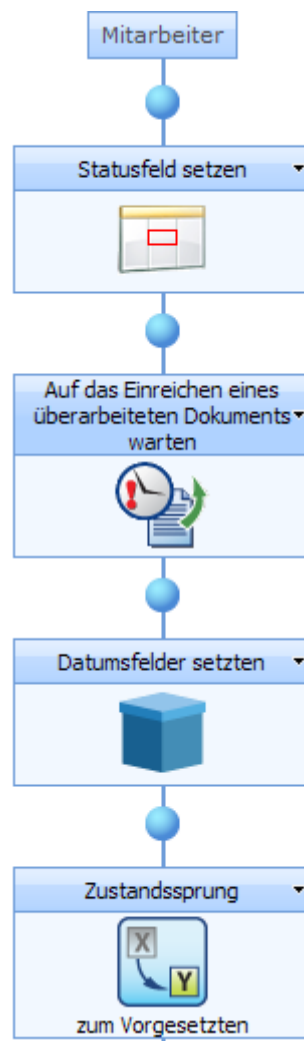


Abbildung 24 - Zustand Mitarbeiter



### 4.3.3 Zustand: Vorgesetzter

Im Folgenden soll der aufwendigste Zustand im gesamten Workflow besprochen werden. Bevor auf den Zustand eingegangen wird, soll zunächst die *Abbildung 25* betrachtet werden, welche eine Übersicht des kompletten Zustands zeigt, um so die folgenden Mechanismen besser zu verstehen:

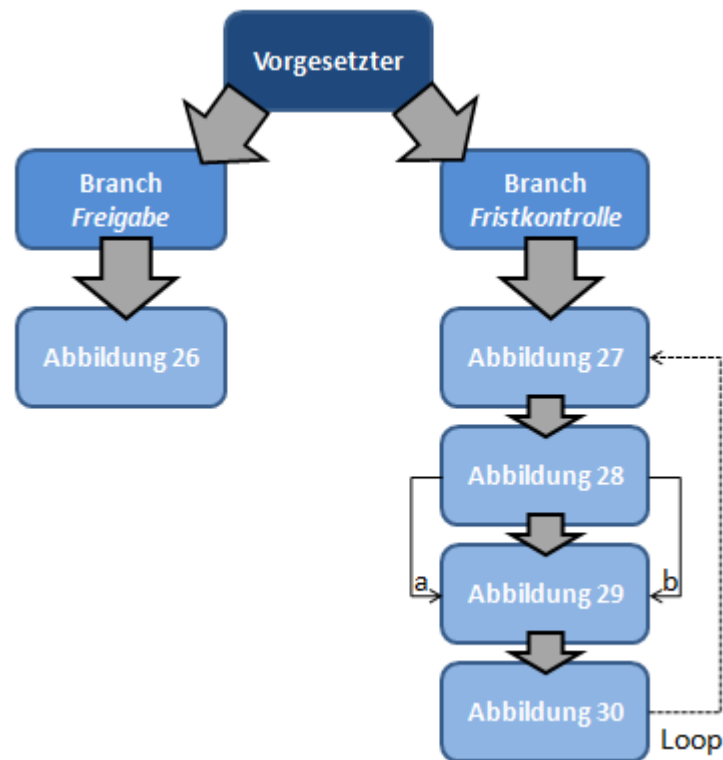


Abbildung 25 - Der Zustand Vorgesetzter in einer Übersicht

Ein ganz wesentlicher Kerngedanke beim Umsetzen des gesamten Zustands war, dass ein Vorgesetzter Vorgänge freizugeben bzw. nicht freizugeben hat, dabei aber immer berücksichtigt wird, ob ein Vorgesetzter in einem bestimmten Zeitraum auf eine ihm zugewiesene Aufgabe reagiert. Das hat den Hintergrund, das laufende Anträge auch definitiv bearbeitet werden und nicht verloren gehen.

Die Lösung für diese Anforderung war dann letztendlich das Parallelisieren beider Abschnitte. D.h. Freigabeprozess *Freigabe* bzw. Kontrollprozess *Fristkontrolle* laufen nebeneinander her (siehe *Abbildung 25*).

Zuerst soll der Abschnitt der Freigabe verdeutlicht werden, welcher die folgenden Aktionen enthält (Branch *Freigabe*):

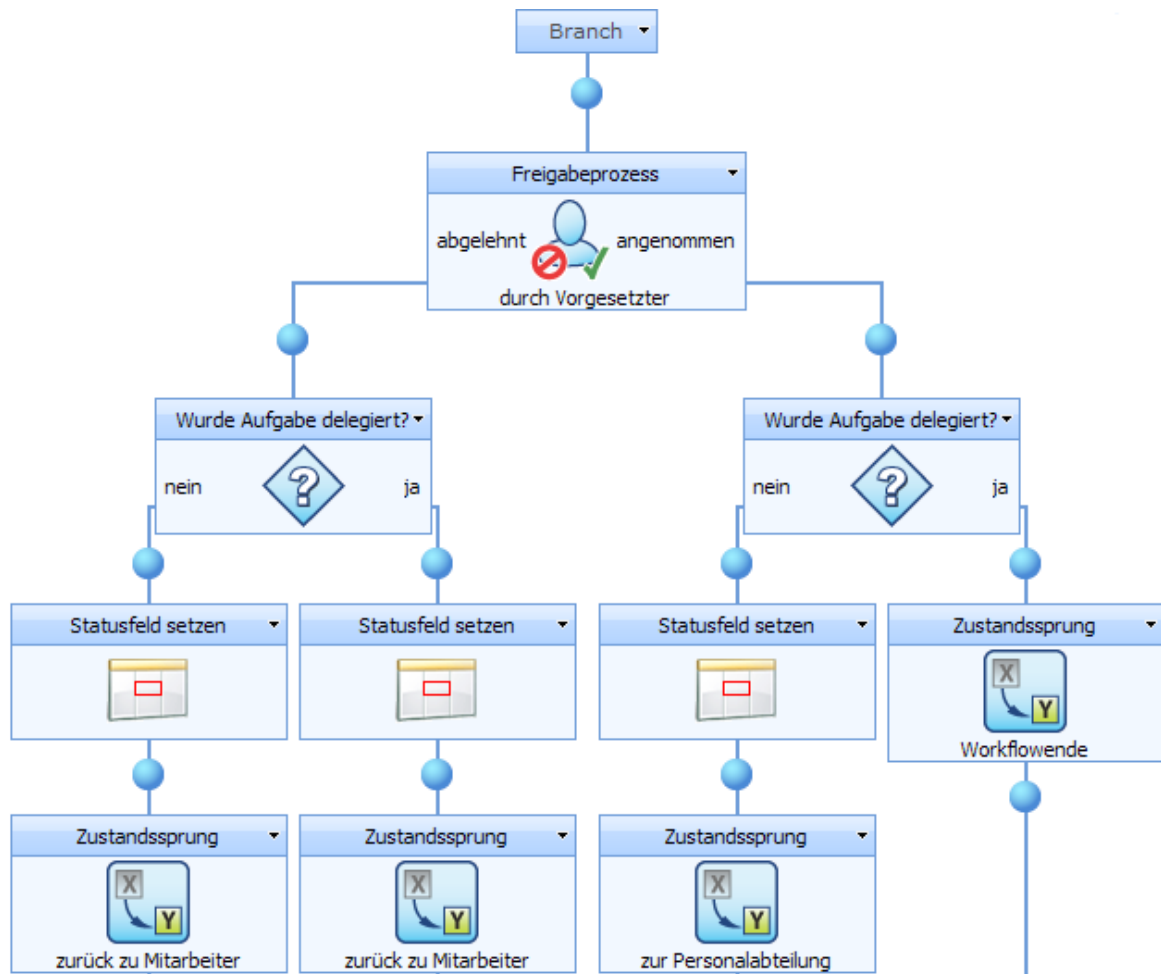


Abbildung 26 - Freigabeprozess des Vorgesetzten

Die [Abbildung 26](#) zeigt den Freigabeprozess des Vorgesetzten. Die Basisaktion ist in diesem Fall ein Freigabeprozess, der entweder angenommen bzw. abgelehnt wird. Auf beiden Seiten dieses Freigabeprozesses folgen zwei entscheidende Aktionen in der Logik dieses Workflows. Vorab: Hat ein Vorgesetzter nicht binnen eines gewissen Zeitraums auf die anstehenden Aufgaben reagiert, so wird die Aufgabe automatisch von Nintex Workflow 2007 an die Personalabteilung delegiert (siehe [Abbildung 30](#)). Daher muss hier noch einmal explizit zwischen einer Delegation bzw. einer nicht stattgefundenen Delegation differenziert werden.

Im linken Bildbereich (Freigabeprozess *abgelehnt*) scheinen beide Seiten der Aktion *Wurde Aufgabe delegiert?* identisch zu sein. Hier wird aber das Statusfeld anders gesetzt. Das Feld wird auf einen Wert gesetzt, der wiedergibt, ob ein Vorgesetzter bzw. die Personalabteilung (im Falle einer Delegation) einen Antrag abgelehnt hat. In beiden Fällen geht der Verlauf des Workflows wieder an den Mitarbeiter zurück, damit dieser den Antrag überarbeiten kann.

Im rechten Bildbereich (Freigabeprozess *angenommen*) befindet sich der Bereich, in dem ein Antrag genehmigt wurde. Wenn die Freigabe, durch eine Delegation an die Personalabteilung, genehmigt wurde, so wird der Workflow regulär beendet und wechselt in den Ende-Zustand (siehe [4.3.5 Zustand: Ende](#)). Wird der Antrag jedoch vom Vorgesetzten genehmigt, wird wiederum das Statusfeld gesetzt und zusätzlich ein Zustandssprung zur Personalabteilung durchgeführt, damit diese auf den Antrag eingehen kann.

Folgend soll der zweite Bereich im Zustand Vorgesetzter besprochen werden - die Kontrolle der Zeitspanne, in der eine Aktion vom Vorgesetzten erwartet wird (Branch *Fristkontrolle*). Da diese Aktion sehr komplex geworden ist, musste die Abbildung in mehrere Teile zerlegt und mit Hilfsgrößen versehen werden. Bevor also weiter auf den Workflow eingegangen wird, vorab ein paar Worte zu den Hilfsgrößen in den folgenden Abbildungen:

- Die zwei geschweiften Klammern } und { umschließen alle folgende Elemente in dem *Loop*
- ..., *a* und *b* sollen den Aufbau des Graphen wiedergeben

Folgend die Darstellung des Ablaufs:

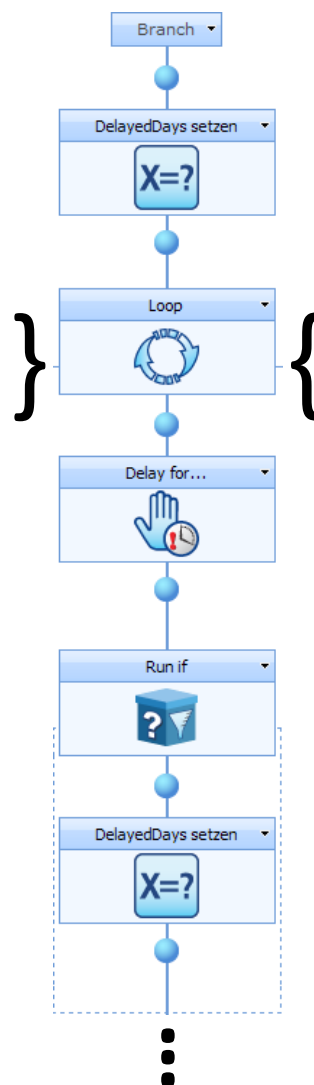


Abbildung 27 - Start des zweiten Branches

Abbildung 27 zeigt den Start in den zweiten Branch des Zustands *Vorgesetzter*. Hier wird die Variable *DelayedDays* auf einen Initialwert von 0 gesetzt. Das hat den Sinn und Zweck, beim ersten Durchlauf des Loops, das Aktivwerden innerhalb des ersten Tages zu überprüfen. Durch den Wert 0, wird das erste *Run if* im Loop ausgeführt und setzt *DelayedDays* auf den Wert 1.

Die Aufgabe des Loops ist, dass es die einzelnen Tage durchläuft, in denen ein Vorgesetzter in Aktion treten muss (in diesem Beispiel wurde die Anzahl der Tage auf *drei* begrenzt). Wie im weiteren Verlauf noch zu sehen ist, werden in dem Loop entsprechende Prüfungen gemacht, um

so beispielsweise die einzelnen Tage in der Variable DelayedDays zu speichern oder Erinnerungsmails an betroffene Personen zu versenden.

Das *Delay for...* dient dazu, den Workflow für ein Zeitspanne anzuhalten und nach Ablauf dieser Zeit, diesen wieder fortzuführen. Dadurch wird im Beispiel das Warten auf die Aktion eines Vorgesetzten simuliert. Diese Aktion wird immer bedingungslos ausgeführt.

Wenn innerhalb des ersten Tages ein Vorgesetzter aktiv geworden ist, wird in der *Abbildung 28* der rechte Pfad eingeschlagen (Antrag bereits bearbeitet? ja) und die Variable DelayedDays auf den Endwert des Loops gesetzt. In diesem Fall auf den Wert 4, wodurch das Loop beendet und der Ablauf mit *Abbildung 30* fortgesetzt wird.

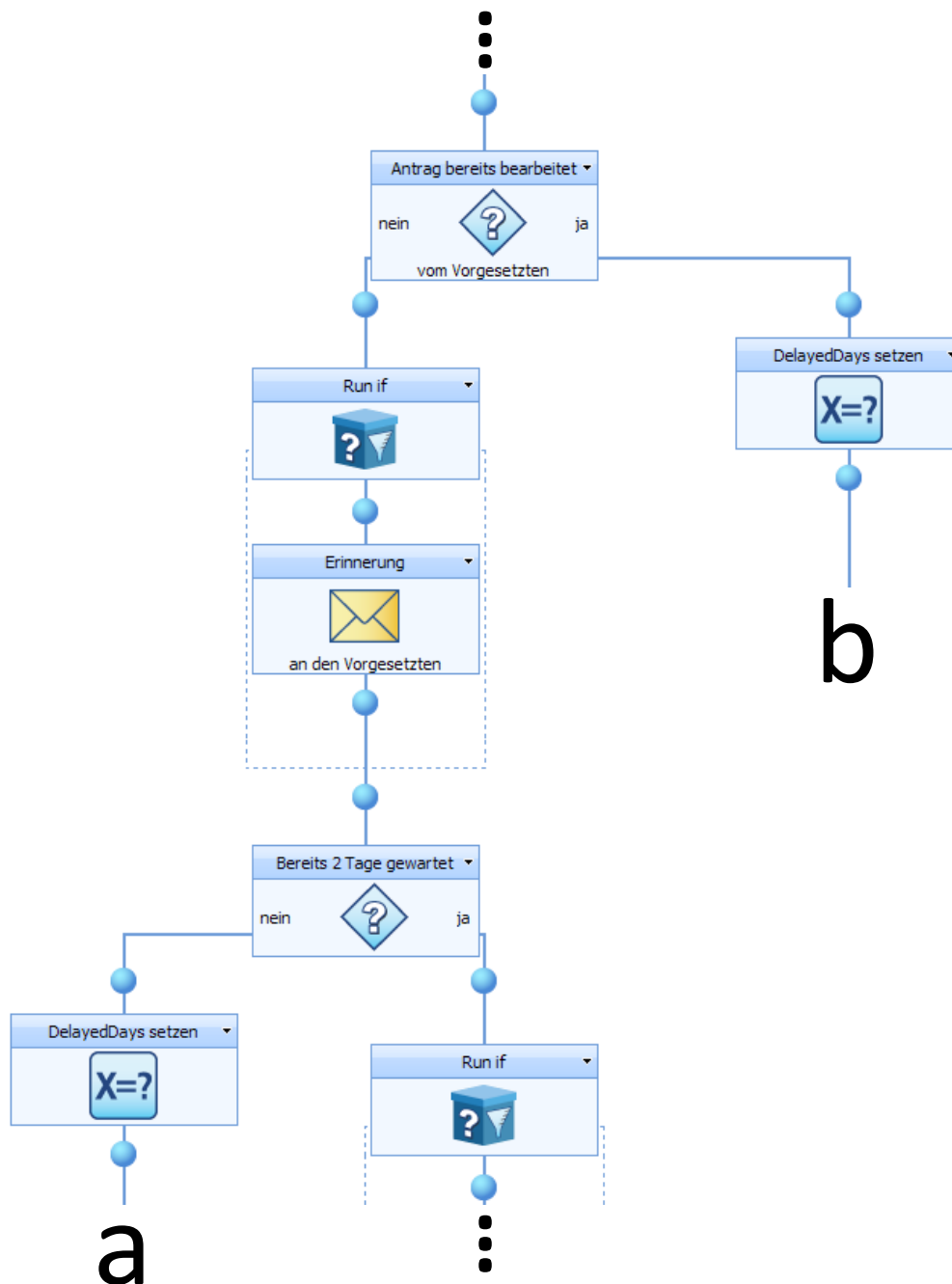


Abbildung 28 - Oberes Mittelstück

Wurde in *Abbildung 28* der linke Pfad betreten (Antrag bereits bearbeitet? *nein*), also es hat noch kein Vorgesetzter reagiert, so wird eine erste Erinnerung versendet. Der Wert der Variable DelayedDays steht in diesem Fall auf 1. Darauf folgend wird abgefragt, ob bereits zwei Tage gewartet wurde. Ist dies nicht der Fall, wird die Variable DelayedDays auf den Wert 2 gesetzt (Bereits 2 Tage gewartet? *nein*) und die Schleife beginnt wieder von vorne.

Für den Fall, dass bereits zwei Tage gewartet wurde, wird der rechte Zweig durchlaufen (Bereits 2 Tage gewartet? *ja*), womit der Workflow-Ablauf bei der *Abbildung 29* angekommen ist. An dieser Stelle wird dann eine zweite Erinnerung an den Vorgesetzten versendet.

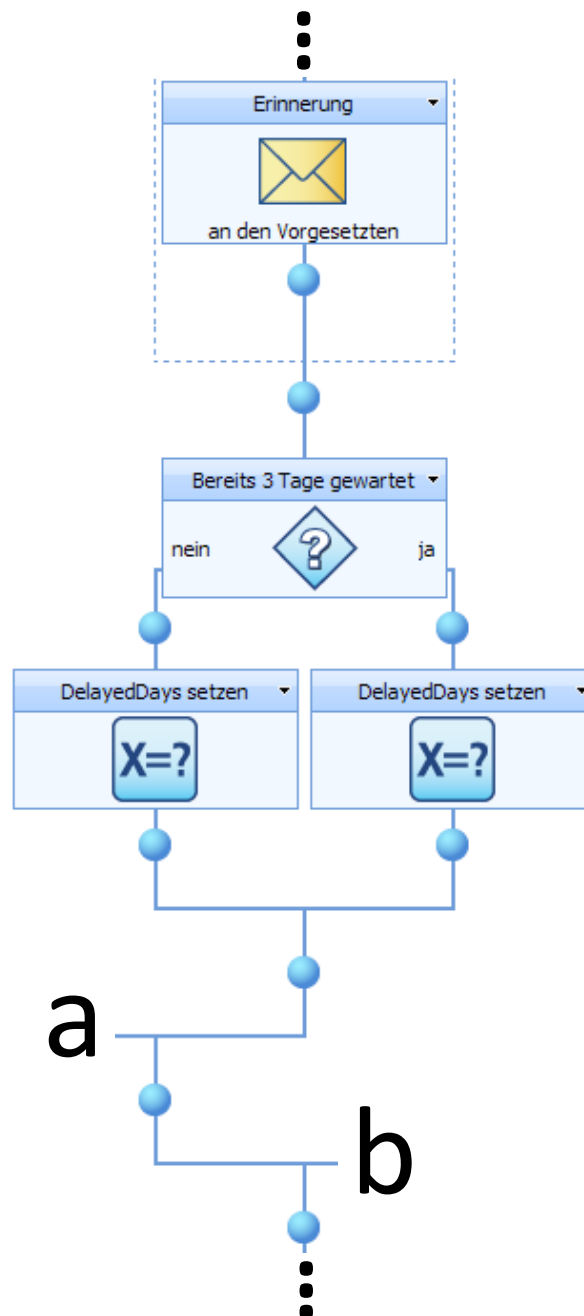


Abbildung 29 - Unteres Mittelstück

Abschließend wird abgefragt, ob bereits drei Tage gewartet wurde oder nicht. Ist dies nicht der Fall (Bereits 3 Tage gewartet? *nein*), wird die Variable DelayedDays auf den Wert 3 gesetzt und es wird ein letztes Mal das Loop durchlaufen. Wurde aber bereits drei Tage lang gewartet (Bereits 3

Tage gewartet? ja), so wird der Wert der Variablen DelayedDays auf 4 gesetzt. Das hat zur Folge, dass zum Einen das Loop nicht mehr durchlaufen wird und zum Anderen, dass der letzte Abschnitt dieses Branches angestoßen wird. Zu sehen ist dies in der *Abbildung 30*.

Der Unterschied des letzten Run ifs in der *Abbildung 30* zu den zuvor erwähnten Run ifs ist der, dass dieses die Variablen DelayedDays und Statusfeld gemeinsam abfragt. Warum beide Werte? Zuerst ist der Wert für DelayedDays entscheidend für den Abbruch des Loops und der zusätzliche Wert des Statusfeldes entscheidet letztendlich darüber, ob das Run if ausgeführt wird oder nicht.

Entspricht DelayedDays dem Wert 4 und Statusfeld *Antrag wurde an den Vorgesetzten weitergeleitet*, so tritt einer von zwei Fällen ein: Ein Vorgesetzter hat auf eine ihm zugeteilte Aufgabe nicht reagiert und die Aufgabe ist zu delegieren (das letzte Run if wird ausgeführt). Der zweite Fall tritt ein, wenn ein Vorgesetzter binnen der gesteckten Zeit reagiert. Trifft dies zu, wird, wie in *Abbildung 26* zu sehen ist, der Wert des Statusfeldes auf *Antrag wurde vom Vorgesetzten genehmigt* bzw. *Antrag wurde vom Vorgesetzten abgelehnt* gesetzt. Das hat den Effekt, dass das letzte Run if niemals ausgeführt werden kann und somit keine Delegation stattfindet, wie es in diesem Fall auch erwünscht ist.

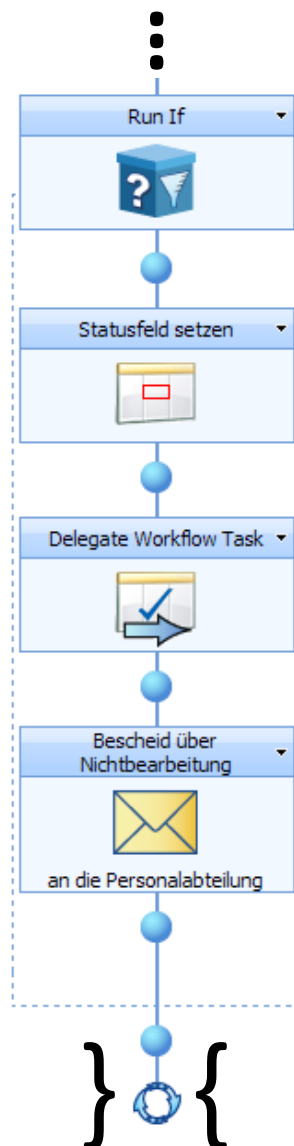


Abbildung 30 - Ende des Branches

Als letzte Anmerkung zum diesem Zustand, noch ein paar Worte zum Aufbau: Nintex bringt von Haus aus eine bereits fertige Aktion *TaskReminder* mit sich, welche sich aber nach dem Erreichen der eingestellten Bedingung nicht mehr zurücksetzt. Das Problem wird deutlich, wenn ein Vorgesetzter in einem Zeitraum nicht reagiert hat und die Aufgabe an eine Personalabteilung weitergeleitet wird. Lehnt diese einen Antrag ab, so wird dieser an den Mitarbeiter zurückgeschickt und anschließend überarbeitet. So ist es in den Anforderungen definiert (siehe [3.4.1.1 Anforderungen](#)).

Der gesamte Prozess sieht aber vor, dass der Antrag wieder von einem Vorgesetzten bearbeitet werden muss. Bei diesem erneuten Durchlauf bleibt jedoch der Zustand des TaskReminders auf dem alten stehen. Dieser wurde also beim ersten Mal durchlaufen und setzt sich leider nicht auf den Anfangszustand zurück, womit es unmöglich wird, diesen trotz richtiger (einmaliger) Funktionalität wieder zu verwenden.

Der offizielle Lösungsvorschlag seitens Nintex ist es, eine *Delay-Aktion* vor den TaskReminder zu setzen, welche etwa 5 Minuten warten soll. Der Kern des Problems ist, dass beim zweiten Durchlauf, der TaskReminder nicht rechtzeitig zurückgesetzt wurde und somit immer noch auf den vorhergegangenen Freigabeprozess verweist. Durch das künstliche Warten vor einem TaskReminder wird dieser vom System zurückgesetzt. Zurzeit lässt sich das aber nur über den Umweg einer Delay-Aktion realisieren.

Nintex räumte ein, dass dieses Verhalten nicht gewollt ist und das es in der nächsten Version dafür einen Hotfix geben wird. Für diese Bachelorarbeit wurde also der umständlichere Weg über die eigene Lösung gewählt, auch wenn dieser um Längen umfangreicher ist. Zudem ergab sich das Problem mit dem TaskReminder erst relativ spät, als der Workflow-Bau schon sehr weit fortgeschritten war.

Der Vollständigkeit halber: Die [Abbildung 31](#) zeigt einen TaskReminder, um einen besseren Eindruck davon zu bekommen, wie viel Arbeit hätte gespart werden können - sofern er dann auch richtig funktioniert. Dieser ersetzt die Abbildungen [27](#), [28](#) und [29](#), sofern der Workflow mittels eines TaskReminders umgesetzt worden wäre. Folgend die Variante, die Nintex als Lösung für das beschriebene Problem vorschlägt:

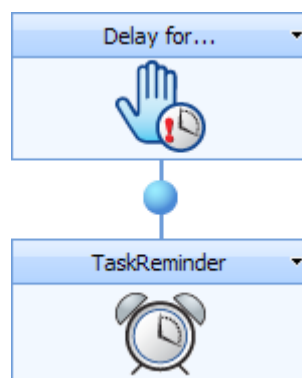


Abbildung 31 - Lösungsvorschlag seitens Nintex

Der Vorteil an einer Lösung mit dem TaskReminder ist, dass die Anwendung skalierbar bleibt. Die gewählte Lösung im Rahmen dieser Bachelorarbeit müsste umständlich angepasst werden, sobald beispielsweise ein Vorgesetzter *drei* oder *vier* Tage Zeit eingeräumt bekommt, um auf Prozesse einzugehen. Da ist der TaskReminder wesentlich flexibler.

### 4.3.4 Zustand: Personalabteilung

Der vorletzte Zustand der betrachtet werden soll, ist der der Personalabteilung (*Abbildung 32*). Die Personalabteilung ist die letzte Instanz im Workflow, die darüber entscheidet, ob ein Urlaubsantrag letzten Endes genehmigt bzw. abgelehnt wird.

Der Übersicht wegen wird in diesem Zustand zuerst das Statusfeld gesetzt, um kenntlich zu machen, dass der Personalabteilung der Urlaubsantrag vorliegt. Der nächste Schritt ist die Freigabe dieses Dokuments. Die Personalabteilung hat auch wie der Vorgesetzte zwei Möglichkeiten in den Prozess einzugreifen:

- Sie kann einen Urlaubsantrag genehmigen
- Bzw. einen Urlaubsantrag ablehnen

Wird ein Urlaubsantrag genehmigt, dann ist der Antrag an sich genehmigt. Dieses impliziert auch, dass der Vorgesetzte dem zustimmte. Der Workflow ist damit am Ende und wechselt in den Zustand *Ende*.

Wird der Urlaubsantrag hingegen abgelehnt, so wird der Antrag dem Antragsteller (Mitarbeiter) erneut mit der Bitte zugestellt, diesen zu überarbeiten (Zustandssprung zu Mitarbeiter). Nach der Überarbeitung des Urlaubsantrags, durchläuft dieser wieder den gesamten Abnahmeprozess, beginnend beim Vorgesetzten und abschließend durch die Personalabteilung.

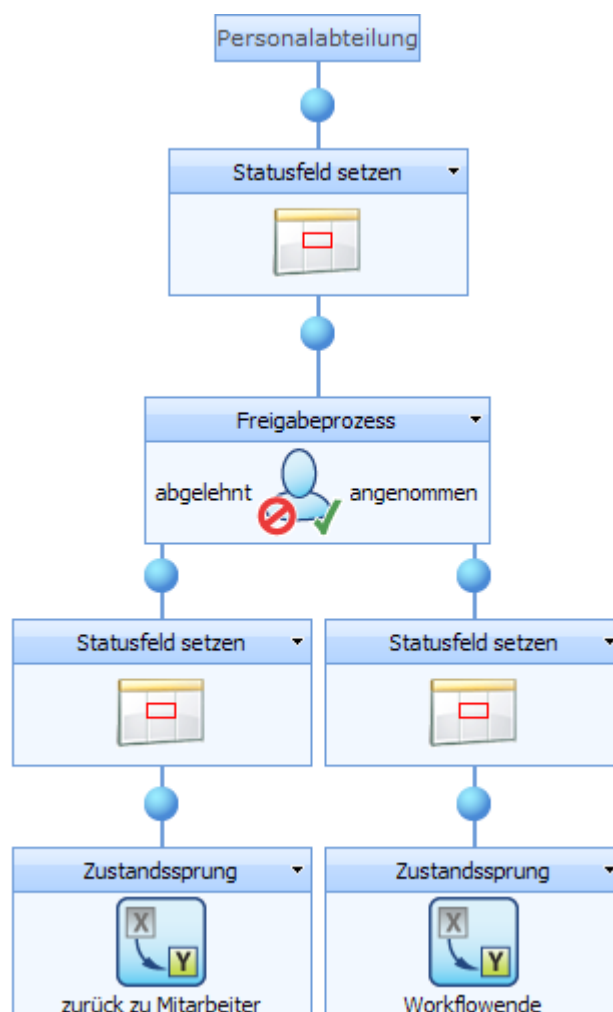


Abbildung 32 - Der Zustand Personalabteilung



### 4.3.5 Zustand: Ende

Der letzte Zustand entstand, weil an zwei Stellen im Workflow ein Sprung zum Workflow-Zustand Ende vorkommt und so redundante Aktionen vorhanden wären. Das hatte vor allem den Hintergrund, dass die Übersicht des Workflows dadurch unverständlich geworden wäre.

Was passiert hier? In diesem Zustand (*Abbildung 33*) werden die letzten Maßnahmen ergriffen, die mit dem Urlaubsantrag in Verbindung stehen. Die da wären:

- Die Benachrichtigung eines Stellvertreters über den Urlaubsantritt des Kollegen
- Das Senden der Daten an ein LOB-System. In diesem Fall diente zur Simulation ein SQL Server mit einer einfachen SQL-Datenbank
- Als nächstes soll der aktuelle Urlaubsantrag in ein Archiv übernommen werden, in welchem diese gesammelt und verwaltet werden. Das geschieht durch ein einfaches Verschieben des Urlaubsantrags in ein Dokumentenarchiv.
- Daraufhin werden die Urlaubstage in einen SharePoint Kalender übernommen
- Als vorletzte Aktion wird der Antrag aus der aktuellen Dokumentbibliothek gelöscht, da hier nur Urlaubsanträge erscheinen sollen, die auch aktuell zu bearbeiten sind. Dies soll die Übersicht im System steigern.
- Zu guter letzt wird der Workflow dann beendet

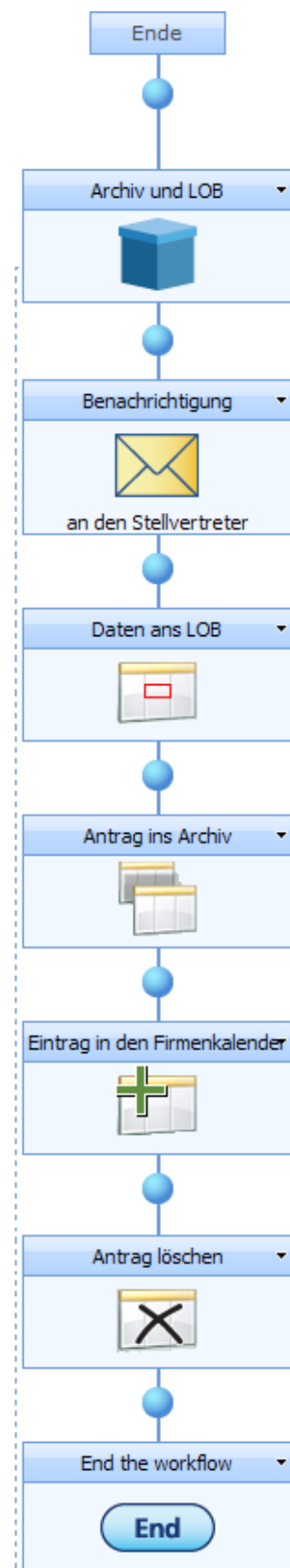


Abbildung 33 - Der Zustand Ende

### 4.3.6 Das Formular

Das InfoPath-Formular bildet die zweite wichtige Komponente des gesamten Prozesses des Urlaubsantrags. Das Formular bietet zwei wesentliche Ansichten:

- Die Ansicht für einen Antragsteller (siehe [Abbildung 34](#))
- und die Ansicht für den Vorgesetzten bzw. die Personalabteilung (siehe [Abbildung 35](#))

Das Formular soll nicht in allen Facetten besprochen, im Rahmen dieser Bachelorarbeit werden aber die wichtigsten Eckdaten erwähnt.

Das Formular gliedert sich grundsätzlich in vier Bereiche:

- Die Antragstellerdaten
- Die Urlaubsrückstände, sowohl aus dem aktuellen, als auch aus dem vorhergegangenen Jahr
- Die individuellen Angaben zum Urlaub selbst
- Und für den Antragsteller nicht zu sehen: der Freigabedialog für den Vorgesetzten und die Personalabteilung

**Urlaubsantrag BETA** Schliessen Einreichen  
Version 0.1

#### Antragsteller

**Name:** Jan Schwarzer  
**System Name:** jschwarzer  
**ID:** 1  
**Dateiname:** Urlaubsantrag Jan Schwarzer 2009-01-29T16:14:23  
**Titel:** Student  
**Mail:** jschwarzer@hvdev.hh  
**Department:** Portals & Workflows  
**Manager Name:** Sven Maier

#### Urlaubsrückstände

<b>Aktuelles Jahr:</b> 2009	<b>Menge:</b> 16
<b>Letztes Jahr:</b> 2008	<b>Menge:</b> 0

#### Ihre Angaben

**Stellvertreter:**   
**Von:**    
**Bis:**  **Anzahl der Tage:**   
**Urlaubsart:**    
**Kommentar:**

Abbildung 34 - Ansicht für einen Antragsteller

Urlaubsantrag *BETA*
Schliessen

Version 0.1

### Antragsteller

**Name:** Jan Schwarzer  
**Dateiname:** Urlaubsantrag Jan Schwarzer 2009-01-29T16:14:23  
**Titel:** Student  
**Mail:** jschwarzer@hvdev.hh  
**Department:** Portals & Workflows  
**Manager Name:** Sven Maier

### Urlaubsrückstände

<b>Aktuelles Jahr:</b> 2009	<b>Menge:</b> 16
<b>Letztes Jahr:</b> 2008	<b>Menge:</b> 0

### Ihre Angaben

**Stellvertreter:**

**Von:**

**Bis:**

**Anzahl der Tage:** **5**

**Urlaubsart:**

**Kommentar:**

### Freigabedialog

Bitte genehmigen bzw. nicht genehmigen Sie diesen Vorgang.

**Kommentar:**

Abbildung 35 - Ansicht für den Vorgesetzten bzw. der Personalabteilung

Die Antragstellerdaten bekommt das InfoPath-Formular von einem eigens entwickelten Active Directory Web Service *ADService*, der alle im Server vorhandenen Benutzerinformationen ausliest und diese InfoPath anbietet. Wie in diesem Fall zu sehen ist, sind die hier aufgeführten Informationen bei weitem nicht alle Informationen zu einem Benutzerprofil, jedoch sind diese hier die wesentlichen Daten, um einen Urlaubsantrag zu simulieren.

Die Urlaubsrückstände werden ebenfalls von einem Web Service *HR-WebService* ermittelt. Dieser Web Service nutzt mehrere Methoden um den Anforderungen gerecht zu werden. In diesem Fall selektiert dieser aus dem simulierten SAP-System die Anzahl der Resturlaubstage aus dem aktuellen sowie aus dem vorigen Jahr.

In dem vorletzten Abschnitt trägt der Antragsteller den Stellvertreter, die Urlaubsdaten, die Urlaubsart und einen Kommentar ein. Sowohl die Urlaubsart, als auch die Anzahl der Tage wird durch den HR-WebService ermittelt und dem InfoPath-Formular übermittelt.

In [Abbildung 35](#) ist noch der Freigabedialog hinzugekommen, welcher nur für einen Vorgesetzten oder der Personalabteilung sichtbar ist. Hiermit kann ein Antrag genehmigt bzw. abgelehnt werden. Das dahinterliegende Abnahmeverfahren wurde bereits verdeutlicht.

### 4.3.7 Testfälle zur Prüfung der Richtigkeit

In diesem Abschnitt sollen Testfälle aufgezeigt werden, die die Richtigkeit der Lösung darstellen sollen. Diese dienen als Basis für das Erstellen des Workflows. Voraussetzung für die Testfälle ist es, dass ein InfoPath-Formular ausgefüllt und korrekt eingereicht wurde. Die bei den Testfällen verwendete Variable  $x$  steht für den Zeitraum, in dem reagiert wurde. Insgesamt gibt es 12 Zustände (siehe *Tabelle 6*), aus welchem sich beliebig lange Ketten von Testfällen für den Workflow herleiten lassen. Der Startzustand *Urlaubsantrag einreichen* hat explizit den Index 0 erhalten, da dieser ansonsten keine Bedeutung mehr im weiteren Verlauf des Workflows besitzt.

Basiskonstrukt der Testfälle		
	Zustand	Resultat
0	Urlaubsantrag eingereicht	Start des Workflows und (Zustand 6, 7, 8 oder 9)
1	Personalabteilung genehmigt	Zustand 11
2	Personalabteilung lehnt ab	Zustand 5
3	Vorgesetzter genehmigt	Zustand 1 oder 2
4	Vorgesetzter lehnt ab	Zustand 5
5	Überarbeitung durch Mitarbeiter	Zustand 6, 7, 8 oder 9
6	Vorgesetzter reagiert ( $x < 1$ Tag)	Keine Benachrichtigung, Zustand 3 oder 4
7	Vorgesetzter reagiert ( $1 \leq x < 2$ Tage)	Erste Erinnerung erhalten, Zustand 3 oder 4
8	Vorgesetzter reagiert ( $2 \leq x < 3$ Tage)	Zweite Erinnerung erhalten, Zustand 3 oder 4
9	Vorgesetzter versäumt Genehmigung ( $x \geq 3$ Tage)	Zustand 10
10	Delegation an Personalabteilung	Zustand 1 oder 2
11	Urlaubsantrag genehmigt	Workflow-Ende

Tabelle 6 - Übersicht aller möglichen annehmbaren Zustände und deren Resultat

Aus dem hier vorgelegten Konstrukt für die Testfälle, lassen sich, wie bereits erwähnt, die verschiedensten Testfälle aufstellen. Dazu sollen ein paar Beispiele betrachtet werden, damit die Verwendung dieses Testfall-Baukonstrukts verständlicher wird.

Zuerst soll der reguläre Fall betrachtet werden, dass ein Urlaubsantrag ohne Widerworte den Workflow durchläuft. Dazu wird aus den möglichen Zuständen eine Folge von Zuständen zusammengestellt:

- $\{0\} \rightarrow \{3\} \rightarrow \{1\} \rightarrow \{11\}$   
**Also:** Urlaubsantrag eingereicht  $\rightarrow$  Vorgesetzter genehmigt  $\rightarrow$  Personalabteilung genehmigt  $\rightarrow$  Urlaubsantrag genehmigt

Auch ist es möglich, über Umwege das Ziel zu erreichen. Das folgende Beispiel zeigt gut, wie sich ein vermeintlich einfacher Abnahmeprozess in die Länge ziehen kann:

- {0} → {4} → {5} → {3} → {2} → {5} → {3} → {1} → {11}  
**Also:** Urlaubsantrag eingereicht → Vorgesetzter lehnt ab → Überarbeitung durch Mitarbeiter → Vorgesetzter genehmigt → Personalabteilung lehnt ab → Überarbeitung durch Mitarbeiter → Vorgesetzter genehmigt → Personalabteilung genehmigt → Urlaubsantrag genehmigt

Die grundlegendsten zu tätigen Testfälle, neben dem als erstes erwähnten Beispiel, sind die Folgenden:

- {0} → {3} → {2} → {5} → {3} → {1} → {11}  
**Also:** Urlaubsantrag eingereicht → Vorgesetzter genehmigt → Personalabteilung lehnt ab → Überarbeitung durch Mitarbeiter → Vorgesetzter genehmigt → Personalabteilung genehmigt → Urlaubsantrag genehmigt
- {0} → {4} → {5} → {3} → {1} → {11}  
**Also:** Urlaubsantrag eingereicht → Vorgesetzter lehnt ab → Überarbeitung durch Mitarbeiter → Vorgesetzter genehmigt → Personalabteilung genehmigt → Urlaubsantrag genehmigt
- {0} → {6} → {3} → {1} → {11}  
**Also:** Urlaubsantrag eingereicht → Vorgesetzter reagiert ( $x < 1$  Tag) → Vorgesetzter genehmigt → Personalabteilung genehmigt → Urlaubsantrag genehmigt
- {0} → {7} → {3} → {1} → {11}  
**Also:** Urlaubsantrag eingereicht → Vorgesetzter reagiert ( $1 \leq x < 2$  Tage) → Vorgesetzter genehmigt → Personalabteilung genehmigt → Urlaubsantrag genehmigt
- {0} → {8} → {3} → {1} → {11}  
**Also:** Urlaubsantrag eingereicht → Vorgesetzter reagiert ( $2 \leq x < 3$  Tage) → Vorgesetzter genehmigt → Personalabteilung genehmigt → Urlaubsantrag genehmigt
- {0} → {9} → {10} → {1} → {11}  
**Also:** Urlaubsantrag eingereicht → Vorgesetzter versäumt Genehmigung ( $x \geq 3$  Tage) → Delegation an Personalabteilung → Personalabteilung genehmigt → Urlaubsantrag genehmigt

Es gibt eine Vielzahl von Möglichkeiten, um das Ziel (also den Zustand 11) zu erreichen und noch viele andere mehr!

Als letzte Bemerkung zu den Testfällen, soll noch einmal auf das Abbruchkriterium des Workflows eingegangen werden. Es ist offensichtlich, dass das einzige Endkriterium eine Urlaubsantragsannahme ist und dass es hier durchaus auch Optionen für einen generellen Abbruch geben könnte. Nur erwies sich diese Art an Funktionalität nicht als sonderlich neu bzw. interessant, um weitere Facetten von Nintex Workflow 2007 zu untersuchen. Somit reichte dieses eine Endkriterium vollkommen aus (siehe [3.4.1.2 Szenario](#)).

Es ist zwar theoretisch möglich, in Zyklen hineinzulaufen, jedoch werden diese in der Praxis durch die Einflussnahme eines Vorgesetzten bzw. einer Personalabteilung (Genehmigung) aufgelöst. Ein Zyklus darf in diesem Fall nicht so verstanden werden, dass im Fall der Fälle das gesamte System zum Stillstand kommt, sondern dass der Workflow im Ablauf an einer Stelle auf eine Aktion wartet (z.B. Aktion *Überarbeitung durch Mitarbeiter*). Dies regelt die Workflow-Engine.

# 5 Analyse und Evaluation

In dem folgenden Kapitel geht es um die Analyse und die Evaluation von Nintex Workflow 2007. Dieses Kapitel soll die zentrale Schlussfolgerung dieser Bachelorarbeit herausstellen und verdeutlichen.

Um eine entsprechende Auswertung zu ermöglichen, wurde das Kapitel in verschiedene Abschnitte gegliedert.

Zuerst werden die einzelnen Kategorien, deren Kriterien und die Gewichtungen vorgestellt, anhand welcher Nintex Workflow 2007 hier beurteilt wird. Anschließend wird das Bewertungssystem vorgestellt, anhand dessen eine einfach zu verstehende Note ermittelt wird und die eigentliche Berechnung des Ergebnisses stattfinden soll.

Der zentrale Bestandteil dieses Kapitels ist die Evaluation von Nintex Workflow 2007 anhand der eben erwähnten Verfahren. Diese Evaluation enthält auch eine detaillierte Beschreibung dafür, warum an der einen oder anderen Stelle wie gewichtet wurde.

Bevor das Kapitel abschließt, werden die Vor- und Nachteile genannt, die während des Erarbeitens dieser Bachelorarbeit festgestellt wurden. Zu guter letzt wird das Fazit und damit die zentrale Schlussfolgerung dieser Bachelorarbeit besprochen.

Ziel ist es, den praktischen Einsatz von Nintex Workflow 2007 zu prüfen und ein aussagekräftiges Fazit zu ziehen. Es soll dem Leser einen möglichst objektiven Eindruck davon vermitteln, was und was nicht mit Nintex Workflow 2007 möglich ist.

## 5.1 Kategorien und Kriterien

Das wichtigste Standbein einer Bewertung sind die Kriterien, anhand welcher eine Bewertung durchgeführt und erst aussagekräftig wird. In diesem Zusammenhang war es wichtig, genauestens die Einsatzbereiche, das Einsatzverhalten und die Vielseitigkeit von Nintex Workflow 2007 zu kennen und zu analysieren. Das Resultat der Recherche war, dass sich die Bewertung in Kategorien und Kriterien zerlegen lässt. Auch wurde im Rahmen dieser Bachelorarbeit der Aspekt der flexiblen Gewichtung von Bewertungen berücksichtigt, um so noch mehr die resultierende Aussage des Ergebnisses den eigenen Wünschen entsprechend anzupassen.

### 5.1.1 Kriterienfindung

Die Kriterienfindung gestaltete sich anfangs durchaus als schwierig, da noch keine genaueren Kenntnisse im Bereich der Workflow-Entwicklung vorhanden waren. Jedoch konnte über die Monate ein gutes und solides Wissen aufgebaut werden, um so entsprechende Kriterien ausfindig zu machen und aufzustellen.

### 5.1.2 Kategorien

Wie bereits erwähnt, gliedert sich der Aufbau der Bewertung in Kategorien und Kriterien. Das dient vor allem dazu, gewisse Kategorien entsprechend gewichten zu können oder besonders hervorzuheben. Je nach Wunsch können die Gewichtungen anders spezifiziert werden. Konkret werden inhaltlich sechs verschiedene Kategorien unterschieden, welche der [Abbildung 36](#) entnommen werden können:

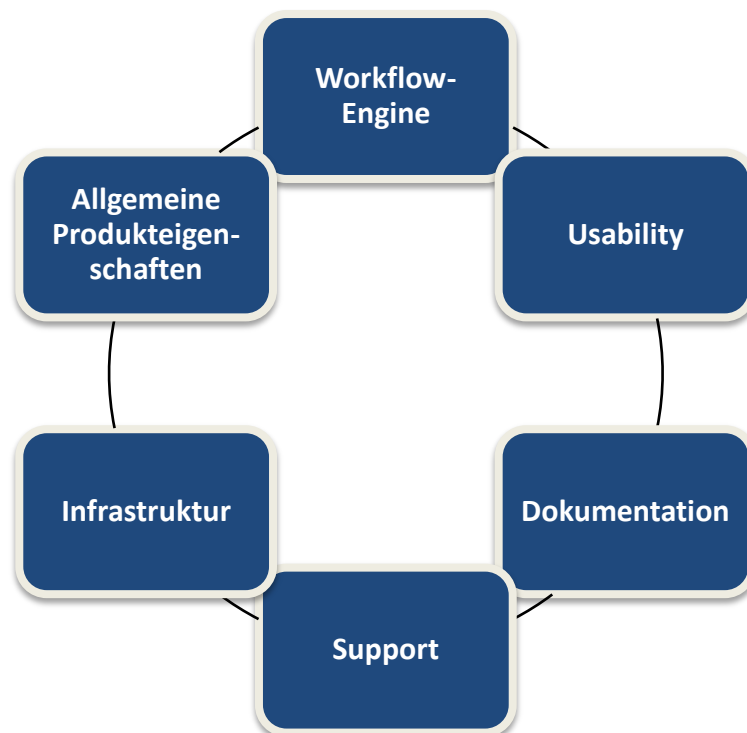


Abbildung 36 - Kategorien Übersicht

### 5.1.3 Kriterien

Natürlich machen die Kategorien erst Sinn, wenn diese entsprechend mit konkreten Merkmalen - also Kriterien - gefüllt sind. Der [Tabelle 7](#) können alle Kriterien entnommen werden:

<b>Workflow-Engine</b>
Mitgelieferte Workflow-Aktionen
State Machine Workflow
Import/Export von Workflows
Wiederverwendbarkeit
Schnittstellen
Workflow-Historie
Workflow-Status
Technologie
Workflow-Bindung
<b>Usability</b>
Erster Eindruck
User Interface
Bedienbarkeit
Test/Debugging
Zufriedenheit
<b>Infrastruktur</b>
Installationsvoraussetzungen
Installationsprozess
Deinstallation
Integration
Skalierbarkeit
<b>Dokumentation</b>
Verständlichkeit und Lesbarkeit
Richtigkeit
Sprachen
Umfang
Formate
<b>Allgemeine Produkteigenschaften</b>
Preis
Lokalisierung
Reporting
Anzahl Partner
<b>Support</b>
E-Mail
Community
Reaktionszeit

Tabelle 7 - Alle Kategorien und deren Kriterien im Überblick

#### 5.1.4 Gewichtungen der Kategorien

Jede Kategorie trägt unterschiedlich viel zu dem Ergebnis dieser Evaluation bei. Bevor auf die einzelnen Gewichtungen im Detail eingegangen wird, ein paar kurze Sätze zur Erklärung:

Die Kategorie Workflow-Engine ist die Kategorie die das meiste Gewicht in das zu ermittelnde Endresultat trägt. Das hat den Hintergrund, dass jedes Produkt allein durch die technischen Produktmerkmale oder anders gesagt, den Leistungsumfang, das Ergebnis besonders beeinflussen



kann und soll. Das ist besonders wichtig, um bei der Auswertung der Ergebnisse auch einen sehr sachlichen fundierten Anteil zu haben.

Eine weitere wichtige Kategorie ist die Usability. Das hat den Grund, dass die effizienteste Software einem nichts nutzt, wenn diese nicht auch in angemessener Weise anwendbar auf ein Problem ist. Die Devise von Nintex Workflow 2007 lautet „No Code“, also das Abwickeln von Geschäftsprozessen, ohne dabei eine Zeile Quellcode zu schreiben. Das sollte auch in der Anwendung berücksichtigt werden.

Als drittstärkste Kategorie kommt die Infrastruktur zum Tragen. Hier geht es vor allem darum, die Integration und die Skalierbarkeit des Produktes zu beurteilen aber auch den Installations- bzw. Deinstallationsprozess.

Zwei weitere wichtige Kategorien sind die Dokumentation und die Allgemeinen Produkteigenschaften. Ein Produkt kann eine relativ befriedigende Bewertung auch ohne diese zwei Kategorien erreichen, jedoch keine gute bis sehr gute. Zum Einen liegt es daran, dass eine vernünftige Dokumentation enorm die Usability steigert und zum Anderen den Einstieg in das Produkt viel effizienter gestaltet. Auch spielen Sachen wie der Preis oder die Lokalisierung eine wichtige Rolle.

Die letzte Kategorie ist nicht minder wichtig, trägt aber zum Ergebnis den geringsten Teil bei. Es ist die Kategorie Support. Da der Schwerpunkt dieser Bachelorarbeit, das Evaluieren einer Workflow-Lösung und somit auch das Erstellen von Workflows die zentrale Anwendung ist, fällt hier die Gewichtung geringer aus. Diese kann natürlich den Wünschen entsprechend angepasst werden.

Die [Abbildung 37](#) zeigt die Kategorien mit deren Gewichtungen im Überblick:

## Kategoriengewichtungen

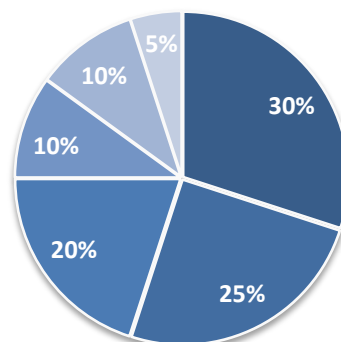


Abbildung 37 - Diagramm zu den Gewichtungen der einzelnen Kategorien

Neben der prozentualen Gewichtung einer jeden Kategorie, spielt auch die Gewichtung von Kategoriemengen im Gesamtzusammenhang eine wichtige Rolle. D.h. welchen Anteil am Gesamtergebnis haben verschiedene Kategoriemengen und was ist die daraus resultierende Aussage.

Warum ist diese Betrachtung wichtig? Zum Einen ist dies wichtig, um zu prüfen, ob eine bestimmte Bewertung einer Kategorie ein Produkt überbewertet. Das lässt sich immer gut im Gesamtzusammenhang ermitteln, um so etwaige Anpassungen zu tätigen. Ziel ist hier also die Justierung der prozentualen Anteile einer Kategorie am Gesamtergebnis. Zum Anderen zeigt diese Betrachtung sehr gut, wie sich Qualität bzw. eine gute Note im Zusammenhang mit den Zielen dieser Bachelorarbeit ermittelt.

## 5.2 Bewertungssystem

Für die Bewertung von Nintex Workflow 2007 wurde ein eigenes Verfahren entwickelt. Es ist möglich, Kriterien und deren Kategorien unterschiedlich zu gewichten und somit einen unterschiedlich großen Einfluss auf das Gesamtergebnis oder im Falle eines Kriteriums auf die Kategorie zu haben. Es gibt also ein zweistufiges Gewichtungssystem.

Ein Kriterium bekommt eine bestimmte Anzahl an Punkten, welche - je nach Güte - zwischen den Werten 0 und 5 variieren kann. Dabei markieren die 5 Punkte das Maximum (ein Kriterium wurde voll erfüllt) und die 0 Punkte das Minimum (ein Kriterium wurde nicht erfüllt). Den Kategorien wurden  $n$  Punkte zugewiesen, welche sich aus der Summe der jeweiligen untergeordneten Kriterien ergibt.

Um ein Höchstmaß an Flexibilität zu gewährleisten, wird zusätzlich einer Kategorie ein prozentualer Anteil am Gesamtergebnis zugewiesen. Dieser Anteil macht darüber eine Aussage, wie sich eine Kategorie prozentual auf ein Gesamtergebnis maximal auswirken darf. Das hat zum Einen den Hintergrund, dass so individuell die Gewichtungen der einzelnen Kategorien variiert werden können, um so diese den Bedürfnissen entsprechend anzupassen. Zum Anderen ist dies wichtig, da im Rahmen dieser Bachelorarbeit die Gewichtung der einzelnen Kategorien dem Thema entsprechend anzupassen waren.

Das Ergebnis der Evaluation zeigt eine einfache Note und den erreichten prozentualen Anteil am Gesamtergebnis.

Die Note steht weniger im Vordergrund, als vielmehr die erreichten Punkte, die Nintex Workflow 2007 erzielt hat. Jedoch unterstreicht diese zusätzliche Bewertung durch ein Notensystem das Gesamtbild eines Produkts. Bei einer guten Note ist auch der Produktiveinsatz gewährt. Bei einer schlechten Note verhält sich das Ganze dann genau anders herum.

Die [Tabelle 8](#) zeigt das besagte Notensystem (11):

Gesamtergebnis (in %)	Note
$x \geq 92$	sehr gut
$92 > x \geq 81$	gut
$81 > x \geq 67$	befriedigend
$67 > x \geq 50$	ausreichend
$50 > x \geq 30$	mangelhaft
$30 > x \geq 0$	ungenügend

Tabelle 8 - Berechnung und Staffelung der Noten

### 5.2.1 Berechnungsformel

Jedes Kriterium besitzt eine vorher festgelegte Anzahl an Punkten, welche bei diesem im Idealfall maximal erreichen kann.

Wie wird pro Kategorie ein Ergebnis ermittelt? Dazu wird die *Formel 1* bemüht:

$$Ergebnis = Anteil\ Kategorie\ (in\ \%) * \frac{erreichte\ Punkte\ in\ Kategorie}{erreichbare\ Punkte\ in\ Kategorie}$$

Formel 1 - Berechnung: Kategorieergebnis

Das Verhältnis von „erreichte Punkte in Kategorie“ zu „erreichbare Punkte in Kategorie“ gibt den Faktor an, um den das Resultat einer Kategorie geschwächt wird. Je mehr Punkte nicht erreicht wurden, desto niedriger fällt das Ergebnis der Gleichung aus und somit der Anteil dieser Kategorie am Gesamtergebnis der Evaluation.

### 5.2.2 Gewichtung der Kriterien

Soll ein Kriterium besonders hervorgehoben werden, so wird diesem das Doppelte an Punkten oder das Dreifache an Punkten zugewiesen (siehe *Tabelle 9*). Das Maximum an Punkten für ein einfach gewichtetes Kriterium beträgt 5 Punkte, für ein doppelt gewichtetes Kriterium 10 Punkte und für ein dreifach gewichtetes Kriterium 15 Punkte. *Formel 2* zeigt die Berechnung eines Kriteriums:

$$Anzahl\ Punkte\ (Kriterium) = erreichte\ Punkte * Gewichtungsfaktor$$

Formel 2 - Berechnung eines Kriteriums

### 5.2.3 Rechenbeispiel

Der Vollständigkeit halber noch ein einfaches Rechenbeispiel (siehe *Tabelle 9*). Losgelöst von den verwendeten Kategorien, wird in diesem Beispiel die Kategorie x mit insgesamt vier Kriterien verwendet. Die Kategorie x darf maximal 30% des Gesamtergebnisses beeinflussen:

Kriterium	Anzahl Punkte		Anzahl <u>erreichbarer</u> Punkte
	Berechnung	Ergebnis	
K1	(0 * 2)	0	10
K2	(5 * 1)	5	5
K3	(2 * 2)	4	10
K4	(5 * 3)	15	15
<b>Summe</b>		<b>24</b>	<b>40</b>

Tabelle 9 - Punktespiegel der Kategorie x

In die *Formel 3* werden als nächstes die ermittelten Werte eingesetzt:

$$Ergebnis = 30\% * \frac{24}{40} \approx 18\%$$

Formel 3 - Berechnung: Kategorieergebnis Beispiel

Wie an diesem Beispiel zu sehen ist, macht das Ergebnis eine tatsächliche Aussage darüber, wie hoch der wirkliche Anteil der Kategorie x am Gesamtergebnis ausfällt. Es wurden von den maximal erreichbaren 30% insgesamt 18% erreicht.

Die in diesem Fall erreichten 18%, gehen in die Berechnung des Gesamtergebnisses in *Formel 4* mit ein, welches sich wie folgt berechnet:

$$\text{Gesamtergebnis} = \text{Summe Kategorieergebnisse (in \%)}$$

Formel 4 - Berechnung: Gesamtergebnis

### 5.3 Evaluation und Ergebnis

Dieser Abschnitt beschäftigt sich mit der Ergebnisfindung und einer Aussage darüber, inwieweit Nintex Workflow 2007 praktikabel ist. Dazu wird zunächst auf das Ergebnis eingegangen und anschließend in *5.4 Auswertung des Ergebnisses* verdeutlicht, warum und wie dieses Resultat zustande kam.

Die *Tabelle 10* zeigt die erreichten Punkte, die prozentualen Anteile und das aus den Einzelwerten resultierende Ergebnis:

	Erreichte Punkte / Erreichter Prozentanteil		Erreichbare Punkte / Erreichbarer Prozentanteil
	Berechnung	Ergebnis	
<b>Workflow-Engine</b>			
Mitgelieferte Workflow-Aktionen	(5 * 2)	10	10
State Machine Workflow	(5 * 2)	10	10
Import/Export von Workflows	(4 * 2)	8	10
Wiederverwendbarkeit	(5 * 2)	10	10
Schnittstellen	(5 * 3)	15	15
Workflow-Historie	(5 * 3)	15	15
Workflow-Status	(5 * 3)	15	15
Technologie	(4 * 3)	12	15
Workflow-Bindung	(3 * 2)	6	10
<b>Summe</b>		<b>101 (≈ 27,55%)</b>	<b>110 (30%)</b>
<b>Usability</b>			
Erster Eindruck	(5 * 1)	5	5
User Interface	(4 * 3)	12	15
Bedienbarkeit	(4 * 3)	12	15
Test/Debugging	(0 * 3)	0	15
Zufriedenheit	(4 * 2)	8	10
<b>Summe</b>		<b>37 (≈ 15,42%)</b>	<b>60 (25%)</b>
<b>Infrastruktur</b>			
Installationsvoraussetzungen	(5 * 2)	10	10
Installationsprozess	(5 * 1)	5	5
Deinstallation	(5 * 1)	5	5
Integration	(4 * 3)	12	15
Skalierbarkeit	(4 * 3)	12	15
<b>Summe</b>		<b>44 (17,6%)</b>	<b>50 (20%)</b>

<b>Dokumentation</b>			
Verständlichkeit und Lesbarkeit	(5 * 3)	15	15
Richtigkeit	(3 * 3)	9	15
Sprachen	(2 * 2)	4	10
Umfang	(5 * 3)	15	15
Formate	(3 * 1)	3	5
<b>Summe</b>		<b>46 (≈ 7,67%)</b>	<b>60 (10%)</b>
<b>Allgemeine Produkteigenschaften</b>			
Preis	(4 * 2)	8	10
Lokalisierung	(5 * 2)	10	10
Reporting	(5 * 2)	10	10
Anzahl Partner	(5 * 1)	5	5
<b>Summe</b>		<b>33 (≈ 9,43%)</b>	<b>35 (10%)</b>
<b>Support</b>			
E-Mail	(5 * 2)	10	10
Community	(5 * 2)	10	10
Reaktionszeit	(5 * 2)	10	10
<b>Summe</b>		<b>30 (5%)</b>	<b>30 (5%)</b>

Tabelle 10 - Auswertung Nintex Workflow 2007

Aus diesen ermittelten Zahlen geht das folgende Ergebnis aus der *Formel 5* hervor:

$$\text{Gesamtergebnis} = 27,55\% + 15,42\% + 17,6\% + 7,67\% + 9,43\% + 5\% = 82,67\%$$

Formel 5 - Nintex Workflow 2007 Gesamtergebnis

Mit einem errechneten Ergebnis von 82,67% bekommt Nintex Workflow 2007 die Gesamtnote „gut“ (siehe *Tabelle 8*).

## 5.4 Auswertung des Ergebnisses

Nintex Workflow 2007 erreicht ein gutes Ergebnis. In diesem Abschnitt soll auf die einzelnen Kriterien eingegangen werden, warum diese wie viele Punkte erreicht haben.

### 5.4.1 Workflow-Engine

Zuerst wird die Kategorie Workflow-Engine betrachtet, welche zum größten Teil das Resultat beeinflusst, was bereits verdeutlicht wurde. Diese besteht zudem auch aus den meisten Kriterien.

#### ▪ Mitgelieferte Workflow-Aktionen

In der aktuellen Version gibt es insgesamt 58 fertige Aktionen. Vielen Geschäftsfällen stehen bereits äquivalente Aktionen gegenüber oder Aktionen, aus denen sich eben diese herleiten lassen. Alle Bedürfnisse wurden durch die vorhandenen Aktionen abgedeckt.

Tritt der Fall ein, dass individuell angepasste Aktionen nötig sind, so kann das von Nintex beigefügte SDK zur Entwicklung eigener Aktionen genutzt werden.

$$\rightarrow \text{Anzahl Punkte} = (5 * 2) = \underline{10}$$

- **State Machine Workflow**

Ein State Machine Workflow ermöglicht es, komplexere Workflows in Nintex Workflow 2007 zu kreieren. Diese sind wichtig, da viele Fälle mit einem sequentiellen Workflow nicht mehr realisierbar wären. In dieser Bachelorarbeit wurde konkret ein State Machine Workflow durch Nintex Workflow 2007 realisiert (siehe [4.3 Beispiel: Urlaubsantrag](#)). Dieser erwies sich als hilfreich und problemlösend.

→ Anzahl Punkte =  $(5 * 2) = \underline{10}$

- **Import/Export von Workflows**

Um eine möglichst große Flexibilität im Rahmen der Mobilität zu gewährleisten, gibt es ein Feature, mit welchem sich Workflows im- bzw. exportieren lassen. Das ist vor allem dann interessant, wenn auf anderen Systemen die erstellten Workflows wieder integriert werden sollen. Zur Abwertung vom sonst sehr guten Eindruck führten hier die Referenzen zweier importierter Workflows. Wenn beide durch eine Aktion (z.B. *Start Workflow*) einander verweisen, so war es nicht möglich, einen der beiden vernünftig - ohne einen Workaround - zu veröffentlichen.

→ Anzahl Punkte =  $(4 * 2) = \underline{8}$

- **Wiederverwendbarkeit**

Hier spielen Aspekte wie Vorlagen oder Aktionsschnipsel (auch „*Snippets*“ genannt) eine Rolle. Gerade das Arbeiten mit den Snippets erspart oft viel Zeit, da so schon fertige Komponenten wiederverwendet werden können. Ein klassisches Beispiel ist ein Freigabeprozess, der in vielen Szenarien vorkommt. Dieser wird mittels eines Snippets gespeichert und steht dadurch jederzeit zur Verfügung und braucht in der Regel nur noch entsprechend konfiguriert zu werden. In der praktischen Anwendung erwies sich das als sehr arbeitsfördernd.

→ Anzahl Punkte =  $(5 * 2) = \underline{10}$

- **Schnittstellen**

Von Schnittstellen wird in diesem Zusammenhang gesprochen, wenn es möglich ist, mit Technologien ganz anderer Natur in Kommunikation zu treten (siehe [Abbildung 12](#)). Dazu bietet Nintex Workflow 2007 beispielsweise fertige Aktionen, um einen Benutzer im Active Directory anzulegen, ihm ein E-Mail-Postfach zu erstellen oder über Web Services mit LOB-Systemen zu kommunizieren. Es konnten hier keine Einschränkungen festgestellt werden.

→ Anzahl Punkte =  $(5 * 3) = \underline{15}$

- **Workflow-Historie**

Wichtig ist auch oftmals, sich über den aktuellen Stand des laufenden Workflows zu erkundigen. Das kann zum Einen zu Testzwecken oder zum Anderen zu Überprüfungs Zwecken dienen. Dabei werden verschiedene Farben verwendet, um so noch mehr die Übersicht zu fördern (siehe [4.2.8 Workflow-Historie](#)). Grundsätzlich gibt es für eine Historie zwei Betrachtungsmöglichkeiten:

- Einsicht mittels einer visuellen Oberfläche
- Einsicht mittels Statistiken zu jeder einzelnen Aktion im Workflow

Die Workflow-Historie fiel besonders positiv auf.

→ Anzahl Punkte =  $(5 * 3) = \underline{15}$

▪ **Workflow-Status**

Neben einer aussagekräftigen Historie spielt auch der Status eine wichtige Rolle. Dieser sagt aus, an welchem Punkt der Workflow sich im Gesamtzusammenhang befindet. Der Status ist ebenso nützlich wie eine solide Workflow-Historie und hilft vor allem auch bei der Fehlersuche. Der Status ist im Gegensatz zur Historie textbasiert und ist als eine Art Protokoll zu verstehen. Das Arbeiten mit dem Workflow-Status erwies sich als sehr gut.

→ Anzahl Punkte =  $(5 * 3) = \underline{15}$

▪ **Technologie**

Das Kriterium Technologie macht darüber eine Aussage, inwieweit das Produkt in sich geschlossen einen Reifegrad aufzeigt. Dazu zählen unter anderem auch gefundene Probleme, die diesen Reifegrad beeinflussen. Nintex Workflow 2007 läuft stabil, jedoch wurden an der einen oder anderen Stelle Probleme ausfindig gemacht (siehe [4.3.3 Zustand: Vorgesetzter](#)). Einige Probleme hingen allerdings mit der neuen Browser-Version des *Internet Explorer* von Microsoft zusammen.

→ Anzahl Punkte =  $(4 * 3) = \underline{12}$

▪ **Workflow-Bindung**

Nintex Workflow 2007 unterstützt keine Bindung von Workflow an Inhaltstypen, jedoch an Listen und Dokumentbibliotheken. D.h. konkret: Werden Workflows in Nintex Workflow 2007 umgesetzt, sind diese immer an eine bestimmte Liste oder eine Dokumentbibliothek gebunden aber niemals an einen Inhaltstypen. Inhaltstypen bringen aber von Haus aus Vorteile mit sich (siehe [2.4.6.1 Inhaltstypen](#)).

→ Anzahl Punkte =  $(3 * 2) = \underline{6}$

## 5.4.2 Usability

Die Kategorie Usability geht als zweitgrößte Komponente in die Gewichtung des Ergebnisses mit ein.

▪ **Erster Eindruck**

Nintex Workflow 2007 macht auf den ersten Blick einen guten Eindruck. Es lassen sich relativ schnell Workflows erstellen, den Ansprüchen entsprechend anpassen und ggf. steht eine umfangreiche Hilfe zur Verfügung.

→ Anzahl Punkte =  $(5 * 1) = \underline{5}$

▪ **User Interface**

Das User Interface ist sehr übersichtlich und komfortabel umgesetzt worden. Die Details hierzu wurden bereits im Kapitel 4 beschrieben. Problematisch an dieser Stelle war, dass größere Workflows nicht mit der Auflösung des Bildschirms skalieren, wodurch u. U. die Übersicht leidet. Auch missfielen statische Eigenschafts-Fenster im Browser, welche sich nicht mittels einer Maus bewegen ließen.

→ Anzahl Punkte =  $(4 * 3) = \underline{12}$

- **Bedienbarkeit**

Während des Arbeitens an Nintex Workflow 2007 fiel die gute Bedienbarkeit auf. Aktionen konnten ohne Probleme in den Lauf des Workflows integriert werden und die wichtigen Menüpunkte waren klar und übersichtlich gestaltet. Zusätzlich konnten die nötigen Einstellungen nach kurzer Zeit gefunden werden. Einziges Manko an dieser Stelle ist: Eine Drag&Drop-Funktion wäre nützlich, um Aktionen so einfacher in den Workflow zu gliedern.

→ Anzahl Punkte =  $(4 * 3) = \underline{12}$

- **Test/Debugging**

Das Testen und Debugging sind wichtige Punkte in der Softwareentwicklung. Leider gibt es keine Möglichkeit, Tests in diesem Kontext zu automatisieren oder generell anzuwenden. Es bleibt lediglich die Möglichkeit des manuellen Testens und das Aufstellen von Testfällen (siehe [4.3.7 Testfälle zur Prüfung der Richtigkeit](#)). Das kostete teilweise sehr viel Zeit und fällt besonders unangenehm auf. Um an dieser Stelle effektiv zu bleiben, mussten alle möglichen Pfade durch einen Workflow analysiert werden, damit nicht unnötig länger getestet wurde.

→ Anzahl Punkte =  $(0 * 3) = \underline{0}$

- **Zufriedenheit**

Die gestellten Anforderungen konnten in einem vernünftigen Rahmen gelöst und erfüllt werden und seitens Nintex gibt es laufend Aktualisierungen und neue Features. Zu den Neuesten gehören beispielsweise ein deutsches Sprachpaket oder die Versionierung von Workflows. Aber: Vorhandene Fehler hielten teilweise bei der Arbeit auf.

→ Anzahl Punkte =  $(4 * 2) = \underline{8}$

### 5.4.3 Infrastruktur

Als nächste wichtige Kategorie soll die Infrastruktur betrachtet werden. Hier geht es darum, Aussagen über die Installation, Integration und die Skalierbarkeit zu machen.

- **Installationsvoraussetzungen**

Hier geht es darum festzustellen, wie weitreichend die Anforderungen für Nintex Workflow 2007 sind. Wie bereits in [4.2.4.1 Voraussetzungen](#) beschrieben, sehen die Anforderungen nicht anders aus, als bei einem MOSS oder den WSS. Es gibt hier also keine zusätzlichen Komponenten die installiert werden müssten.

→ Anzahl Punkte =  $(5 * 2) = \underline{10}$

- **Installationsprozess**

Eine wichtige Rolle spielt auch der Installationsprozess. Dieser kann sehr umfangreich werden und den Anwender mit Optionen überschütten. Hier beschränkt sich Nintex Workflow 2007 auf das Wesentlichste und führt einen Anwender Schritt für Schritt durch den Installationsprozess.

→ Anzahl Punkte =  $(5 * 1) = \underline{5}$



- **Deinstallation**

Was installiert wird, muss auch im Fall der Fälle vernünftig deinstalliert werden können. Dazu wurde eine Probendeinstallation durchgeführt, um genau dieses testen zu können. Es wurden alle Komponenten gelöscht. Dabei ist vor allem die Deinstallation der integrierten Funktionalitäten im SharePoint System interessant gewesen.

→ Anzahl Punkte =  $(5 * 1) = \underline{5}$

- **Integration**

Nintex Workflow 2007 integriert sich sehr tief in die SharePoint Infrastruktur. Es nutzt die Windows Workflow Foundation (siehe [3.2.1 Windows Workflow Foundation \(WF\)](#)) und es müssen keine eigenen Serverkomponenten installiert werden. Ein Nachteil ist allerdings, dass dadurch der Administrationsaufwand steigt. Nintex Workflow 2007 kann an verschiedenen Stellen im SharePoint System konfiguriert werden (Zentral- und Webseitenadministration), was zunächst zu Übersichtsproblemen führen kann. In beiden Bereichen überschneiden sich zum Teil die Einstellungen, je nachdem, welchen Wirkungsgrad eine Einstellung haben soll. Also: Gültigkeit in einer kompletten SharePoint Farm oder nur auf eine Webseite bezogen.

→ Anzahl Punkte =  $(4 * 3) = \underline{12}$

- **Skalierbarkeit**

Durch die starke Integration in die SharePoint Landschaft, hat Nintex Workflow 2007 an dieser Stelle einen Nachteil. Es skaliert abhängig von der gegebenen SharePoint Infrastruktur. Wird beispielsweise ein MOSS aufgesetzt, so bietet Nintex Workflow 2007 erheblich mehr Aktionen an, als es unter den WSS der Fall ist. Dieser Effekt ist u. U. nicht immer bewusst gewollt. An dieser Stelle sei auf den in [B.2 MOSS- und WSS-Funktionalitäten im Vergleich](#) beschriebenen Funktionsvergleich verwiesen.

→ Anzahl Punkte =  $(4 * 3) = \underline{12}$

#### 5.4.4 Dokumentation

Zwar fällt im Rahmen dieser Bachelorarbeit die Gewichtung der Dokumentation weniger ins Gewicht als andere Kriterien, jedoch ist und bleibt eine vernünftige Dokumentation ein wichtiges Muss.

- **Verständlichkeit und Lesbarkeit**

Die beigefügte Dokumentation ist auf englisch, aber im Gesamtzusammenhang eine sehr verständliche und lesbare Dokumentation. Diese bietet dem Anwender an verschiedensten Stellen Hilfestellungen an, sei es beim Erstellen oder Editieren eines Workflows. Auch gibt es zudem nützliche Tutorials, die den Anwender Schritt für Schritt durch die Beispiele führen und somit noch mehr das Verständnis schulen.

→ Anzahl Punkte =  $(5 * 3) = \underline{15}$

- **Richtigkeit**

Die Dokumentation ist, wie bereits erwähnt, sehr umfangreich. Jedoch wurden gerade bei den Tutorials manchmal Fehler gefunden. Das führte in der Regel dazu, dass am Ende z.B. eine InfoPath-Integration nicht funktionierte und dann die Suche nach den möglichen Fehlern begann. Bei der InfoPath-Integration lag der Fehler darin, eine neue Web Service Methode zu benutzen. Diese hieß wie die Vorgängermethode, nur wurde der aktuellen im Namen eine „2“ angefügt. In dem Tutorial hingegen, stand kein Hinweis, dass hier die neue Version zu

verwenden ist. Hier wurde die alte Methode gewählt. Das sind Fehler, welche seitens des Herstellers verhindert werden können.

→ Anzahl Punkte =  $(3 * 3) = \underline{9}$

- **Sprache**

Zurzeit gibt es nur das englische Sprachpaket. Neue Sprachpakete (auch in Deutsch) wurden angekündigt, konnten aber im Rahmen dieser Bachelorarbeit nicht mehr berücksichtigt werden.

→ Anzahl Punkte =  $(2 * 2) = \underline{4}$

- **Formate**

Es gibt eine Vielzahl von Formaten der Dokumentation bzw. Hilfestellungen. Dazu gehören u.a. Tutorials und Anleitungen in Form von PDF-Dateien oder eine eigene Hilfedatei (auch im Web Designer) mit einer eigenen Indexsuche.

→ Anzahl Punkte =  $(5 * 3) = \underline{15}$

- **Umfang**

Den einzigen Kritikpunkt zum Thema Umfang gibt es in der Hinsicht, dass manche Aktionen nicht ausreichend beschrieben werden. Ein gutes Beispiel in diesem Zusammenhang ist das Arbeiten mit Web Services in Verbindung mit sogenannten *Credentials*<sup>41</sup>. Hier wären an der einen oder anderen Stelle ausführlichere Informationen hilfreich gewesen - zumindest ein Hinweis darauf, wie diese im Zusammenhang mit Authentifikationsmechanismen arbeiten.

→ Anzahl Punkte =  $(3 * 1) = \underline{3}$

### 5.4.5 Allgemeine Produkteigenschaften

Folgend soll die Kategorie Allgemeine Produkteigenschaften betrachtet werden.

- **Preis**

Durch die Lizenzen für Nintex Workflow 2007 entstehen zusätzliche Kosten (siehe [4.2.1 Lizenzen](#)). Im Vergleich zu anderen Workflow-Lösungen bleibt es aber verhältnismäßig günstig (siehe [6.3 BlueSpring BPM](#)).

→ Anzahl Punkte =  $(4 * 2) = \underline{8}$

- **Lokalisierung**

Unter Lokalisierung werden Aspekte wie der Vorortsupport oder die Vertretung im eigenen Lande untersucht. Nintex unterhält ein großes und weltweites Netzwerk an Partnern und Kunden. Im Rahmen dieser Bachelorarbeit wurde explizit ein Ansprechpartner genannt, der sich um die deutschen Partner bzw. die deutschen Kunden kümmert, welcher dann auch Rede und Antwort bei offenen Fragestellungen stand.

→ Anzahl Punkte =  $(5 * 2) = \underline{10}$

---

<sup>41</sup> Credentials erlauben es, zentrale Berechtigungen auf Schnittstellen zu vergeben. Diese können beim Arbeiten mit einer SQL-Datenbank oder einem Web Service verwendet werden. Der Vorteil an Credentials ist es, dass Anmeldeinformationen - also Passwort und Benutzername - nicht für einen normalen Anwender sichtbar sind, sondern eben nur diese spezielle Credential

- **Reporting**

Eine gerade von Vorgesetzten geliebte Eigenschaft ist das Reporting. Mit Hilfe des Reportings können Aussagen über den Verlauf oder Status eines Workflows visualisiert werden. Auch ist das Aufstellen von angepassten Statistiken mittels Reporting möglich. Zwar gibt es seitens Nintex extra ein eigenes Produkt, jedoch sind die Möglichkeiten schon in Nintex Workflow 2007 umfangreich und für die Zwecke dieser Bachelorarbeit vollkommen ausreichend.

→ Anzahl Punkte =  $(5 * 2) = \underline{10}$

- **Anzahl Partner**

Nintex unterhält ein weites Netzwerk an Partnern. Es gibt zurzeit ca. 60 Partner und ca. 500 Kunden weltweit. Warum ist das interessant? Nintex Workflow 2007 ist eine relativ neue Workflow-Lösung. An dieser Stelle ist vor allem der Austausch zwischen den Partnern interessant, um so Projekterfahrungen oder Problemstellungen zu kommunizieren. In Deutschland gibt es bereits einige Partnerunternehmen die Nintex Workflow 2007 produktiv einsetzen. Die Anzahl der Partner ist zudem Zeugnis dafür, inwieweit das eigene Produkt eine gewisse Marktreife erreicht hat.

→ Anzahl Punkte =  $(5 * 1) = \underline{5}$

#### 5.4.6 Support

Als letzte Kategorie soll der Support besprochen werden.

- **E-Mail**

Sehr positiv fiel im Laufe der Recherchen nach Problemlösungen der E-Mail-Support auf. I.d.R. war einen Tag nach dem Einreichen einer Problemstellung auch schon eine Antwort im E-Mail-Postfach. Um ein gutes Beispiel zu nennen: In [4.3 Beispiel: Urlaubsantrag](#) wurde die Problematik mit dem TaskReminder angesprochen. Hierzu setzte sich nach dem Eingang der Nachricht das Entwicklerteam von Nintex zusammen, fand eine Lösung und bestätigte, dass die gestellte Problemstellung durchaus berechtigt wäre und dass das beschriebene Problem in Folgeversionen behoben werden soll.

→ Anzahl Punkte =  $(5 * 2) = \underline{10}$

- **Community**

Neben dem sehr guten E-Mail-Support gibt es zudem die Möglichkeit, Problemstellungen in der Nintex eigenen *Community*<sup>42</sup> zu stellen. Auch hier kann von schnellen Reaktionszeiten und tatsächlichen Problemlösungen ausgegangen werden.

→ Anzahl Punkte =  $(5 * 2) = \underline{10}$

- **Reaktionszeit**

Die Reaktionszeit auf Fragestellungen spielt im Support eine wesentliche Rolle, um so den Kunden oder Partner nicht abzuschrecken. Sowohl im Rahmen von E-Mail-Nachrichten oder dem Schreiben von Beiträgen im Community-Forum, fiel immer wieder die schnelle Bearbeitung eben dieser auf. Gerade in Sachen E-Mail-Support wurde die Reaktionszeit im Laufe der letzten Monate noch einmal verbessert.

→ Anzahl Punkte =  $(5 * 2) = \underline{10}$

---

<sup>42</sup> <http://connect.nintex.com/>

## 5.5 Vor- und Nachteile

Nach der Auswertung und dem Evaluieren des Ergebnisses soll in diesem Abschnitt explizit auf die Vor- und Nachteile von Nintex Workflow 2007 eingegangen werden (siehe [Tabelle 11](#)). Es sind vor allem grundsätzliche Punkte, die während der Arbeit an Nintex Workflow 2007 auffielen. Im Fazit werden die hier erwähnten Punkte noch einmal zur Ergebnisfindung dieser Bachelorarbeit aufgegriffen.

Bewertungstabelle	
<b>(++) sehr gut</b>	
▪	Schneller Einstieg in die Technologie
▪	Ausgereifte Technologie
<b>(+) gut</b>	
▪	Eigenes SDK
▪	LazyApproval
▪	ausführliche Hilfedatei und How-To's
▪	Versionierung von Workflows
▪	In der Enterprise Version gibt es viele nützliche Aktionen
<b>(-) schlecht</b>	
▪	Keine Workflow-Bindung an Inhaltstypen
▪	Verwaltung der Workflows ist umständlicher
▪	Für umfangreiches Reporting wird Nintex Reporting 2008 benötigt
▪	Bei komplexen Workflows „geht die Übersicht verloren“
<b>(- -) ungenügend</b>	
-	

Tabelle 11 - Übersicht der Vor- und Nachteile von Nintex Workflow 2007

## 5.6 Fazit

Zuerst soll im Rahmen dieses Fazits auf das errechnete Ergebnis eingegangen werden. Was sich aus dem Ergebnis von 82,67% (Note „gut“) ablesen lässt ist, dass Nintex Workflow 2007 eine Marktreife erlangt hat, an welcher Stelle dieses Produkt definitiv für den praktischen Einsatz geeignet ist. Im Laufe des letzten Jahres wurden also viel Zeit und Ressourcen investiert, um hier entsprechende Fortschritte zu machen. Nintex ist mit dessen Produkt definitiv über einen Beta-Status hinaus und kann sich in der aktuellen Form auf dem Markt behaupten.

Durch diesen Reifegrad ist es zudem möglich, interessierten Kunden dieses Produkt weiterzuempfehlen, damit diese hierdurch ihre eigene Prozesse selbstständig optimieren können. Dabei spielt es überhaupt keine Rolle, ob der Endanwender letztendlich ein Entwickler oder ein ganz normaler Anwender ohne Programmierkenntnisse ist. Die Arbeit mit diesem Tool erwies sich oft als sehr intuitiv, ohne dabei auf nötige Funktionalitäten verzichten zu müssen. Gestellte Anforderungen konnten schnell und effektiv gelöst werden, was vor allem durch die umfangreichen mitgelieferten Aktionen ermöglicht wurde. Durch die Einfachheit ist es deshalb auch möglich, etwaige Fachanwender daraufhin zu schulen, dieses Produkt in einem Unternehmen effektiv anzuwenden. Hier ist ein Schulungsaufwand von wenigen Tagen i.d.R. völlig ausreichend.

Folgend sollen noch ein paar konkrete Äußerungen zu den Vor- und Nachteilen von Nintex Workflow 2007 gemacht werden, welche der [Tabelle 11](#) zu entnehmen sind:

Die als sehr positiv aufgefallenen Eigenschaften sind, zum Einen der schnelle Einstieg in dieses Tool und zum Anderen der Reifegrad. Diese beiden Faktoren werden grundsätzlich durch eine solide Dokumentation gestützt, wobei der Reifegrad sich erst im Laufe der Arbeit beweisen musste und der schnelle Einstieg vom intuitiven Nintex Web Designer profitierte. Die Summe aller entwickelten Workflow-Beispiele machte einen soliden Eindruck und ließ sich entsprechend auf die Funktionalität prüfen und testen.

Zudem als positiv bewertet wurde, dass Nintex von Haus aus ein eigenes SDK mit sich bringt, womit der Variation von eigenen Aktionen keine Grenzen gesetzt sind. Auch schafft das SDK dann Abhilfe, wenn es um zu erstellende Aktionen geht, die standardmäßig nicht in Nintex Workflow 2007 enthalten bzw. mit den gebotenen Mitteln nicht umsetzbar sind. Ein zudem nützliches Feature ist das LazyApproval, was eine sehr effiziente und unmittelbare Interaktion mit Prozessen zulässt.

Wie bereits erwähnt, fallen vor allem die Dokumentation und die mitgelieferten How-To's von Nintex Workflow 2007 sehr positiv auf und helfen ungemein dabei, sich in die Nintex-Umgebung einzuarbeiten. Zusätzlich ist durch den Einzug einer Versionierung möglich geworden, eigene Workflows in verschiedenen Versionen zu verwalten.

Das einzige, was an Nintex Workflow 2007 wirklich stört ist, dass Workflows leider nicht an Inhaltstypen gebunden werden können. Das würde vor allem dann Zeit sparen, wenn von einem Inhaltstypen geerbt und somit auch auf den dahinter liegenden Workflow zugegriffen werden soll. An dieser Stelle muss sich mit Vorlagen aus der Nintex Vorlagenbibliothek oder aus dem Dateisystem begnügt werden. Alle anderen Eigenschaften, wie die umständlichere Verwaltung von Workflows, die zwar vorhandenen, aber nicht umfangreichen Reporting-Möglichkeiten oder die unübersichtliche Gestaltung von komplexeren Workflows, fallen hier nicht ins Gewicht, da diese an sich erfüllt werden, nur leider manchmal ein wenig umständlicher sind als es intuitiv angenommen wurde.

Abschließen soll das Fazit mit ein paar persönlichen Eindrücken:

Es ist gut an der Entwicklung an Nintex Workflow 2007 zu erkennen, was auf der einen Seite der Markt für solche Lösungen offeriert und wie intensiv ständig das Tool seitens des Herstellers weiterentwickelt wird. Das Interessante an dem Ergebnis dieser Bachelorarbeit ist zudem, dass anfangs noch als subjektiv aufgefasste Argumente, doch objektiv und damit völlig sachlich betrachtet werden können. Somit kann eine sehr gute Aussage über die Güte dieses Tools getroffen werden, hauptsächlich durch die Auswertung anhand der Kriterien gestützt. Ein nicht erwartetes und doch erfreuliches Ergebnis.

Teilergebnisse dieser Bachelorarbeit dienen nicht nur der Zusammenarbeit mit den Kunden der Firma HanseVision, sondern führten auch dazu, dass die HanseVision offizieller Nintex-Partner geworden ist. Zudem setzen die Kunden der HanseVision vermehrt auf Nintex Workflow 2007 und dessen Nachfrage nach Software-Lösungen in diesem Bereich steigt weiterhin.

Auch wurde während der Vorstellung der Teilergebnisse dieser Bachelorarbeit auf der deutschen SharePoint Konferenz 2009 (12) noch einmal deutlich, wie groß das Interesse nach Lösungen wie Nintex Workflow 2007 ist. Der Vortrag war sehr gut besucht und es wurden viele Fragen zu den verschiedensten Bereichen gestellt. Auch die Resonanz nach dem Vortrag war sehr positiv, hier waren besonders die Präsentationsfolien gefragt.

## 6 Alternative Workflow-Lösungen

Dieses Kapitel beschäftigt sich mit alternativen Lösungen zu Nintex Workflow 2007. Diese erstrecken sich von Anwendungen für SharePoint Plattformen bis hin zu Anwendungen für Plattformen anderer Hersteller. In diesem Kapitel sollen beide Varianten aufgegriffen werden.

Neben den beschriebenen Alternativen soll auch verdeutlicht werden, warum K2 blackpoint beta 2 nicht, wie ursprünglich geplant, fester Bestandteil dieser Bachelorarbeit wurde, wie es Nintex Workflow 2007 ist. Dazu werden die aufgetretenen Probleme beschrieben und beurteilt.

In diesem Kapitel werden die bis zuletzt erarbeiteten Kenntnisse über K2 blackpoint beta 2 aufgegriffen und in Form von Vor- und Nachteilen geschildert werden. Auch wird auf Besonderheiten speziell im Vergleich mit Nintex Workflow 2007 eingegangen.

## 6.1 K2 blackpoint beta 2

Wie bereits einleitend erwähnt wurde, zeigte sich während der Evaluation, dass die Bachelorarbeit nicht in der ursprünglich angedachten Form umgesetzt werden konnte. Trotzdem soll an dieser Stelle auf K2 blackpoint beta 2 besonders eingegangen werden, da auch hier viel Zeit investiert und eine Menge an wertvollen Erkenntnissen aus dieser Zeit gewonnen wurde. Zusätzlich wurden diese Erkenntnisse der Öffentlichkeit auf der deutschen *SharePoint Konferenz 2009* präsentiert (12).

K2 blackpoint beta 2 ist in jedem Fall ein ganz klarer Konkurrent zu Nintex Workflow 2007 und platziert sich entsprechend auf den Markt der Workflow-Entwicklung unter Sharepoint mit dem Slogan „*SharePoint workflow. More features. No code*“.

Im Schwerpunkt sollen an dieser Stelle vor allem die Abgrenzung zu Nintex Workflow 2007 und die Vor- und Nachteile besprochen werden. Diese stehen dann im direkten Vergleich zu denen von Nintex Workflow 2007, welche im letzten Abschnitt des Kapitels 5 beschrieben wurden. Bevor aber diese beiden Punkte weiter beleuchtet werden sollen, ein Wort zu der Motivation, K2 blackpoint beta 2 im Rahmen dieser Bachelorarbeit nicht mehr zu berücksichtigen.

### 6.1.1 Problematiken und Begründung für den Ausschluss

Es erwies sich insgesamt als nicht sonderlich einfach, einen ersten Schritt in K2 blackpoint beta 2 zu machen. Es gab zwei wesentliche Punkte, die eine tiefergreifendere Untersuchung von K2 blackpoint beta 2 im Rahmen dieser Bachelorarbeit verhinderten:

- Der Beta-Status: In Laufe der Arbeit an K2 blackpoint beta 2 wurden auf Anhieb *vier* von K2 bestätigte Bugs ausfindig gemacht, was die Arbeit mit der Anwendung nicht einfacher werden ließ
- Schlechte Tutorials: Zwar bietet K2 für dessen Produkt eine verständliche Hilfedatei an, jedoch sind die beigefügten Tutorials, erstens: nicht ausreichend und zweitens: für den Einstieg als „*Workflow-Neuling*“ nicht geeignet

Es verstrich also viel Zeit und es war noch nicht einmal im Ansatz ein etwas komplizierterer Workflow gebaut, um im Rahmen des Funktionsumfangs ein Fazit zwischen den beiden 3rd-Party-Tools zu ziehen. Es musste also eine Entscheidung gefällt werden und die fiel gegen K2 blackpoint beta 2 aus.

### 6.1.2 Besonderheiten

K2 blackpoint beta 2 löst folgende Aspekte grundsätzlich anders (um nur einige zu nennen):

- Installation eines eigenen Servers, um Prozesse (Workflows) zu verwalten
- Dienste sind zentralisierter verfügbar in einem sogenannten *Process Portal*
- Die Workflow-Entwicklung findet in einem Web Designer, als auch in einem einer Office-Anwendung sehr ähnlichem Tool statt - dem *K2 Studio*
- Die Upgrade-Möglichkeiten sehen anders aus. K2 blackpoint beta 2 bildet im Funktionsumfang eine Untermenge der Funktionen von *K2 blackpearl*, dem Produkt-Flaggschiff von K2, wieder

### 6.1.3 Vor- und Nachteile

In diesem Abschnitt sollen die Vor- und Nachteile von K2 blackpoint beta 2 besprochen werden (siehe *Tabelle 12*).

Bewertungstabelle	
<b>(++) sehr gut</b>	<ul style="list-style-type: none"> <li>klare Trennung von SharePoint und K2 blackpoint beta 2 dank eines eigens entwickelten Prozess Portals</li> <li>gezielte und granulare Rechtevergabe auf Workflows</li> </ul>
<b>(+) gut</b>	<ul style="list-style-type: none"> <li>Skalierung ist unabhängig von der SharePoint Infrastruktur</li> <li>Bindung von Workflows u.a. an Inhaltstypen</li> <li>Für jede Aktion gibt es einen hilfreichen Wizard</li> <li>ausführliche Hilfedatei</li> <li>detailliertere Reporting-Möglichkeiten</li> </ul>
<b>(-) schlecht</b>	<ul style="list-style-type: none"> <li>BizTalk Verwendung nur mit K2 blackpearl</li> <li>SAP Verwendung nur mittels K2 connect</li> <li>kein LazyApproval</li> <li>Beta Status - Fehler waren und sind noch vorhanden</li> </ul>
<b>(- -) ungenügend</b>	<ul style="list-style-type: none"> <li>K2 blackpoint beta 2 ist wesentlich aufwändiger in der Einarbeitung</li> </ul>

Tabelle 12 - Übersicht der Vor- und Nachteile von K2 blackpoint beta 2

## 6.2 SAP Business Workflow

Mit Hilfe von SAP Business Workflow können betriebswirtschaftliche Geschäftsprozesse in ein SAP-System integriert werden. Es ist diesem Tool möglich, sowohl einfache als auch komplexere Workflows zu erstellen.

Einen besonders effizienten Einsatz findet die Software dort, wo sich Geschäftsprozesse immer wiederholen oder Abnahmehierarchien eingehalten werden müssen. Dazu bietet SAP Business Workflow bereits eine Menge vorgefertigter Aktionen und Werkzeuge, wie es auch bei Nintex Workflow 2007 der Fall ist. Verwendet wird zum Erstellen bzw. Editieren von Workflows der sogenannte *Workflow Builder*. Dabei unterstützt der *Workflow Wizard Explorer* den Anwender beim Kreieren verschiedener Aktionen im Workflow selbst.

SAP Business Workflow integriert sich in das *SAP-R/3-System*<sup>43</sup> und verwendet dessen Transaktionen und Funktionen. Die bestehenden Funktionen können wieder verwendet werden, wobei die Steuerung vom *Workflow-System* übernommen wird. Die Wiederverwendung geschieht mittels des *Business Object Builders*, der alle Funktionen aus dem R/3-System dem Anwender dann über den *Business Workflow Explorer* zur Verfügung stellt.

Einsatzhemmende Faktoren für SAP Business Workflow sind u.a. die fehlende Kenntnis im Umgang mit der Technologie oder Zeitrestriktionen (13). Vorteilhaft hingegen ist der Einsatz in bereits vorhandenen SAP-Infrastrukturen, ohne dabei auf eine Software-Lösung Dritter zurückgreifen zu müssen. Außerdem soll, laut der unter (13) angegebenen Studie, die Bedeutung von SAP Business Workflow in Zukunft steigen.

Es wird generell empfohlen, vor einer Integration von SAP Business Workflow, alle Prozesse genauestens zu analysieren und mit einem SAP-Berater zu klären. Eine mögliche und unerwünschte Folge wäre, dass die Kosten unter Umständen erheblich in die Höhe steigen.

<sup>43</sup> Ein von SAP entwickeltes Unternehmens-Informationssystem



### 6.3 BlueSpring BPM

*BlueSpring BPM*<sup>44</sup> bietet ein Sammelsurium an Funktionalitäten, aus denen ein Kunde wählen kann. In diesem Kontext, können Workflow-Lösungen für Microsoft BizTalk, *Microsoft Dynamics*<sup>45</sup> oder Microsoft SharePoint gewählt werden.

Während der Recherche fiel vor allem eines auf: Es sammelten sich mehr und mehr Beschwerden. Auf diese soll im Folgenden kurz eingegangen werden:

- Sehr hohe Preise, welche sich zwischen einer hohen fünfstelligen und einer sechststelligen Zahl bewegen können
- Der Konfigurationsaufwand ist enorm und das trotz Live-Installationsanleitungen des Herstellers
- BlueSpring BPM bringt eine Menge an Komponenten mit sich, was die Übersicht mindert
- Es gibt kein webbasiertes Tool zum Erstellen bzw. Editieren von Workflows - nur eine Clientsoftware

Auch werden im Zusammenhang mit BlueSpring BPM oft die Alternativlösungen Nintex Workflow 2007 und K2 blackpoint beta 2 genannt. Aus den gesammelten Informationen scheint also der Einsatz nicht empfehlenswert zu sein. SAP Business Workflow hingegen, hat im Zusammenhang mit SAP-Lösungen weiterhin eine Daseinsberechtigung.

### 6.4 Citrix Workflow Studio

Das Citrix Workflow Studio benötigt zuallererst eine vorhandene Citrix-Infrastruktur, um darauf basierend Workflows zu erstellen. Es ermöglicht so die Zusammenstellung, Integration und Orchestrierung von Workflows. Dank der zu Grunde liegenden Infrastruktur, können die einzelnen Geschäftsprozesse miteinander kombiniert bzw. integriert werden. Auch ist es möglich, die Prozesse von zertifizierten Partnern zu implementieren und diese in der eigenen Anwendung zu nutzen.

Ähnlich wie bei Nintex Workflow 2007 und K2 blackpoint beta 2, gibt es auch für Citrix-Workflow-Lösungen eine grafische Oberfläche, um Workflows zu erzeugen. Dabei nutzt das Tool u.a. das volle Potenzial der Windows Workflow Foundation, so dass der Anwender Workflows „codefrei“ entwickeln kann.

Einen großen Vorteil bietet Citrix Workflow Studio durch dessen .NET-Umsetzung, wodurch die Anbindung an System-Management-Lösungen größerer Hersteller, wie HP oder IBM vereinfacht wird. Ein Nachteil ist, dass Citrix Workflow Studio auf die *Microsoft Windows PowerShell*<sup>46</sup> setzt und somit einen gewissen Grad an Kenntnissen im Umgang mit Konsolenprogrammen voraussetzt.

---

<sup>44</sup> BPM = Business Process Management

<sup>45</sup> Hinter Microsoft Dynamics verbergen sich alle Business-Lösungen von Microsoft

<sup>46</sup> Die Microsoft Windows PowerShell dient der Administration von Anwendungen durch ein Konsolenprogramm

## 6.5 GEDYS IntraWare 7

GEDYS IntraWare 7 setzt auf *IBM Lotus Notes*<sup>47</sup>. Es bietet, wie die anderen vorgestellten Lösungen auch, die klassischen Workflow-Lösungen für beispielsweise Anträge aller Art. Ziel ist es auch hier, Prozesse zu digitalisieren, um so den Aufwand für Arbeitsabläufe zu minimieren.

Es ermöglicht, jede IBM Lotus Datenbank workflowfähig zu machen, ohne dabei das gesamte Lotus Notes System zu beeinflussen. Grundsätzlich besteht das Workflow-System aus einer Konfigurationsdatenbank für Prozesse und Arbeitsschritte und einer Steuerungsdatenbank. Um auf Dokumente in der Anwendung zugreifen zu können, werden die entsprechenden Datenbanken in einer zentralen Datenbank vermerkt und für die Steuerung verwendet. Personeninformationen aus dem Lotus Notes Namens- und Adresssystem (NAB/Domino Directory) können jederzeit abgerufen werden. Dieses ähnelt stark der Infrastruktur des Active Directory unter Windows.

GEDYS IntraWare 7 bietet ebenfalls ein grafisches Tool, um eigene Prozesse zu modellieren. Einen großen Mehrwert bietet dieses Tool durch die Möglichkeit, Workflows testen zu lassen.

Ein weiterer Vorteil ist der modulare Aufbau von GEDYS IntraWare 7, wodurch bewusst nur einzelne, für die Unternehmung tatsächlich notwendige, Module zum Einsatz gebracht werden müssen und nicht das Gesamtsystem (z.B. Modul *Workflow*). Der einzige Nachteil liegt weniger am Produkt selbst, sondern viel mehr an der zu Grunde liegenden Software: IBM Lotus Notes erfordert eine aufwendige Einarbeitung.

Als letzte Info noch: Die GEDYS IntraWare GmbH erhielt von IBM den *IBM Lotus Award 2008*<sup>48</sup> unter dem Titel „*Best TOTAL Lotus Software Solution*“, was die Qualität der Anwendung unterstreicht.

## 6.6 IBM Lotus Workflow 7

IBM Lotus Workflow 7 ist ein Mitglied der IBM-Workflow-Familie. Es erweitert im Wesentlichen die Workflow-Funktionen der IBM-Domino-Software, welche gleichzeitig die Kollaborationsplattform von IBM darstellt und sich so auf dem Markt als Gegenstück zu Microsoft SharePoint positioniert.

Es ist ein eigenständiges Tool, welches sich in die bereits bestehende IBM-Infrastruktur integrieren lässt und einen eigenen Editor zur einfachen Erstellung von Workflow-Lösungen mitbringt. In erster Linie ermöglicht es die Automatisierung, Verwaltung und Überwachung von Geschäftsprozessen sowie die flexible Anpassung an Änderungen im Prozessablauf.

Im Zusammenhang mit IBM Lotus Workflow 7 werden als positiv die Möglichkeit zum Testen von Workflows oder die Wiederverwendbarkeit von Modulen genannt. Wie bereits bei GEDYS IntraWare 7 angedeutet, ist der Einarbeitungsaufwand in Lotus Notes selbst aufwendig.

---

<sup>47</sup> IBM Lotus Notes ist ein dokumentbasiertes und verteiltes Datenbanksystem mit einer sehr engen E-Mail-Anbindung

<sup>48</sup> <http://www.gedys-intraware.de/presse/pressemitteilungen/archiv/gedys-intraware-7-weltweit-beste-lotus-loesung.html>

# 7 Zusammenfassung

Dieses Kapitel soll alle wichtigen Aspekte zusammenfassen und die zentrale Botschaft der Bachelorarbeit aufgreifen und herausstellen.

In den Kapiteln 2 und 3 wurden die grundsätzlichen theoretischen Sachverhalte beschrieben. Als erstes wurde XML in allen Facetten begutachtet, also dessen Struktur, Gestaltung und dessen Anwendung. XML diente dann als Basis, um sich mit dem nächsten Thema auseinanderzusetzen - den Web Services. Diese setzen ausschließlich auf eine Kommunikation via XML-Dokumente und spielen in Bezug auf Nintex Workflow 2007 an verschiedenen Stellen eine wichtige Rolle. Als nächster großer Themenbereich wurde SharePoint erläutert, welches die Basis für alle Arbeiten mit Workflows auf Windows-Plattformen stellt. Hier war es vor allem wichtig, ein breites und zum Teil tiefes Verständnis zu entwickeln.

Im Kapitel 4 wurden dann alle nötigen Kenntnisse vermittelt, um mit Nintex Workflow 2007 arbeiten zu können. Dazu gehörten beispielsweise die Vor- und Nachteile, das Tool an sich, die Integration, Lizenzkosten, Beispiele uvm. Dieses Kapitel bildete die Grundlage für das Kapitel 5, um dort ein Fazit finden bzw. um grundsätzlich mit den verwendeten Kriterien arbeiten zu können. Kapitel 4 trägt zudem den Teil, in welchem Fachanwender und Entwickler konkret an das Tool herangeführt werden sollen, um so das Arbeiten damit zu erlernen.

Das Kapitel 5 beschäftigte sich mit der Beurteilung von Nintex Workflow 2007. Dazu wurden Kategorien und Kriterien aufgestellt, sortiert und entsprechend gewichtet. Anhand dieser Kriterien wurde ein Ergebnis evaluiert und eine Aussage darüber getroffen, inwieweit hier eine Praxistauglichkeit vorliegt.

Auch wurde in dieser Bachelorarbeit in Kapitel 6 noch einmal explizit auf die anfangs beschriebene Problematik mit K2 blackpoint beta 2 eingegangen, welche sich im Laufe der Arbeit ergeben hat. Es wurden auch hier entsprechende Vor- bzw. Nachteile genannt und die Problematik an sich noch einmal deutlicher beschrieben.

Nintex Workflow 2007 ist ein empfehlenswertes und einsetzbares Produkt, um unter SharePoint Workflows „codefrei“ umzusetzen. Die Vorteile überwogen und machten es so zu einer sehr interessanten Lösung. Es erreichte ein gutes Gesamtergebnis von 82,67% und damit die Gesamtnote „gut“. An Nintex Workflow 2007 wird auch in Zukunft weitergearbeitet bzw. es werden noch weitere Konkurrenzprodukte auf dem Markt folgen. In diesem Zusammenhang wurden derzeit erhältliche Alternativen zu SharePoint bzw. zu Nintex Workflow 2007 kurz vorgestellt, um so den „Blick über den Tellerrand“ zu schärfen (siehe Kapitel 6).

## 8 Ausblick

Das letzte Kapitel beschäftigt sich mit den Dingen, die in dieser Bachelorarbeit nicht weiter untersucht wurden oder aus zeitlichen Gründen nicht in diesen Rahmen passten.

Zuerst soll auf das beigefügte SDK verwiesen werden, welches Nintex standardmäßig dem Produkt beifügt. Hier sind vor allem Fragestellungen bzgl. des Umfangs, Testens, Leistungsfähigkeit und der Integration spannend. Aber auch Punkte, inwieweit das Programmieren von eigenen Aktionen für einen Fachanwender möglich ist oder ob das SDK einen wirklichen Mehrwert mit sich bringt. Ein SDK bleibt für einen Entwickler weiter interessant, um hier bereits vorhandenes Wissen aus der Programmierung einzusetzen.

K2 blackpoint beta 2 ist nicht im ursprünglich geplanten Umfang in diese Bachelorarbeit eingeflossen. Die Gründe dafür wurden ausgiebig erläutert. Interessant wäre aber an dieser Stelle, mit dem erlangten Wissen auch die gleichen Bewertungskriterien für K2 blackpoint beta 2 aufzustellen, um hier eine vergleichbare Aussage treffen zu können. Zusätzlich wäre auch eine Gegenüberstellung gleicher Workflows gut, um so die Unterschiede beider Tools zu analysieren, da sich die grundsätzliche Philosophie unterscheidet. Zudem würde eine solche Untersuchung die Güte einer Aussage zur Qualität der beiden Tools noch einmal deutlicher unterstreichen.

Das gesteckte Ziel dieser Bachelorarbeit ist die Evaluation von Nintex Workflow 2007 mit einer anschließenden Ergebnisfindung. Es sollte sich auf Basis dieses Wissens in Unternehmen entschieden werden können, ob ein Einsatz von Nintex Workflow 2007 für sinnvoll oder nicht sinnvoll gehalten wird.

Eine ganz andere Möglichkeit ergibt sich dadurch, das Erlernte aus der Microsoft Umgebung auf eine Umgebung anderer Hersteller zu spiegeln. Ein grundsätzliches Verständnis baut die Bachelorarbeit auf, hier sind dann ggf. Schnittstellen zu anderen Lösungen zu beachten. Gerade die Hersteller SAP und IBM bieten in diesem Marktsegment komplett eigene Lösungen, sowohl zur Erfassung von Workflows, als auch deren Digitalisierung.

Es wurde sehr umfangreich das Beispiel einer InfoPath-Anbindung an Nintex Workflows untersucht. Zusätzlich gäbe es die Möglichkeiten Excel-Formulare einzubinden, mit einem BizTalk Server oder einem SQL Server zusammenzuarbeiten, um an dieser Stelle nur mal ein paar Beispiele zu nennen. Nintex Workflow 2007 bietet eine Vielzahl von Möglichkeiten, welche dann aber ausblickend separat zu dieser Bachelorarbeit untersucht werden könnten.

Nintex Workflow 2007 deutet mit dem integrierten Reporting auf erste Varianten hin. Es gibt zusätzlich seitens Nintex noch ein weiteres Produkt, welches sich Nintex Reporting 2008 nennt. Mit diesem sind komplexere Reportings möglich. An anderer Stelle gilt es herauszufinden, wann ein Einsatz einer zusätzlichen Software sinnvoll ist.

# Literaturverzeichnis

1. **Staubli, Oliver.** <http://www.crealogix.com/>. [Online] Medienbeobachtung, Argus, 19. 09 2007. [Zitat vom: 25. 02 2009.]  
[http://www.crealogix.com/en/PortalData/1/Resources/content/holding/presse\\_ecke/pressreport/3782798.pdf](http://www.crealogix.com/en/PortalData/1/Resources/content/holding/presse_ecke/pressreport/3782798.pdf).
2. **Wikipedia.** Wikipedia. [Online] Wikimedia Foundation Inc. [Zitat vom: 23. Februar 2009.]  
<http://de.wikipedia.org/wiki/SGML>.
3. **Short, Scott.** *Webdienste mit dem .NET Framework entwickeln*. Redmond, Washington 98052-6399 : Microsoft Press, 2002.
4. **Egbert Jeschke, Helmut Reinke, Sven Maier.** *Microsoft Office InfoPath 2007*. Unterschleißheim : Microsoft Press, 2007.
5. **tecchannel.** In zehn Schritten zur SOA. [Online] IDG Business Media GmbH, 15. 12 2005. [Zitat vom: 13. 12 2008.]  
[http://www.tecchannel.de/webtechnik/soa/569662/in\\_zehn\\_schritten\\_zur\\_soa/](http://www.tecchannel.de/webtechnik/soa/569662/in_zehn_schritten_zur_soa/).
6. **Errin O'Connor, Jim Buyens.** *Microsoft Windows SharePoint Services 3.0 - Das Handbuch*. Redmond, Washington 98052-6399 : Microsoft Press, 2008.
7. **Dirk Grasekamp, Michael Greth, Bernhard Straub.** *Microsoft Office SharePoint Designer 2007 - Das Handbuch*. Unterschleißheim : Microsoft Press, 2008.
8. **Boddenberg, Ulrich B.** *Microsoft Office SharePoint Server 2007 & Windows SharePoint Services 3.0 - Das Praxisbuch für Administratoren und Entwickler*. Bonn : Galileo Press, 2008.
9. **Kalenborn, Axel.** *Prozessorganisation und Workflow-Management: Organisationstheoretisches Konzept und informationstechnische Umsetzung*. Aachen : Shaker Verlag GmbH, 2000.
10. **Galler, Jürgen.** *Vom Geschäftsprozessmodell zum Workflow-Modell*. Wiesbaden : Universität Saarbrücken, 1997.
11. **Wikipedia.** Wikipedia. [Online] Wikimedia Foundation Inc. [Zitat vom: 13. Dezember 2008.]  
<http://de.wikipedia.org/wiki/Schulnote>.
12. **Schwarzer, Jan und Maier, Sven.** *Workflows mit SharePoint 3rd-Party-Lösungen: Nintex Workflow und k2.blackpoint – Ein Vergleich*. München : ppedv AG,  
<http://www.sharepointkonferenz.de/>, 2009.
13. **Martin Meyer, Frank Wimmer.** Institut für Wirtschaftsinformatik der Universität Bern. [Online] 1998. [Zitat vom: 02. 03 2009.]  
<http://www.ie.iwi.unibe.ch/publikationen/berichte/resource/WP-108.pdf>.

# A Verzeichnisse

## A.1 Abbildungsverzeichnis

Abbildung 1 - Ein durch XSL formatiertes XML-Dokument im Browser.....	18
Abbildung 2 - Eine InfoPath-Formularvorlage.....	24
Abbildung 3 - Ein InfoPath-Formular.....	25
Abbildung 4 - SharePoint Webseite im Nintex Workflow 2007 Testcenter.....	27
Abbildung 5 - Themenbereiche von SharePoint .....	28
Abbildung 6 - Eine Übersicht der Unterschiede zwischen MOSS und WSS .....	31
Abbildung 7 - Architektur eines SharePoint Webs.....	33
Abbildung 8 - Ein einfaches Vererbungsbeispiel von Inhaltstypen in SharePoint .....	34
Abbildung 9 - Ein einfacher zustandsbasierter Workflow.....	37
Abbildung 10 - Architektur der Windows Workflow Foundation .....	39
Abbildung 11 - Abnahmeprozess eines Urlaubsantrags .....	41
Abbildung 12 - Kollaboration von Nintex Workflow 2007 mit anderen Programmen .....	45
Abbildung 13 - Die Zentraladministration.....	46
Abbildung 14 - Die Webseitenadministration.....	47
Abbildung 15 - Das Benutzer-Menü .....	48
Abbildung 16 - Das Settings-Menü.....	48
Abbildung 17 - Der Nintex Workflow 2007 Web Designer mit einem leeren Workflow .....	49
Abbildung 18 - Das Aktions-Menü mit allen verfügbaren Aktionskategorien .....	49
Abbildung 19 - Die Menüs Actions und Settings.....	50
Abbildung 20 - Reporting Möglichkeiten mit Nintex Workflow 2007.....	51
Abbildung 21 - Ein einfaches Historie Beispiel .....	52
Abbildung 22 - Prinzip des LazyApprovals.....	53
Abbildung 23 - Start des Workflows.....	55
Abbildung 24 - Zustand Mitarbeiter.....	56
Abbildung 25 - Der Zustand Vorgesetzter in einer Übersicht .....	57
Abbildung 26 - Freigabeprozess des Vorgesetzten .....	58
Abbildung 27 - Start des zweiten Branches.....	59
Abbildung 28 - Oberes Mittelstück .....	60
Abbildung 29 - Unteres Mittelstück .....	61
Abbildung 30 - Ende des Branches.....	62
Abbildung 31 - Lösungsvorschlag seitens Nintex.....	63
Abbildung 32 - Der Zustand Personalabteilung .....	64
Abbildung 33 - Der Zustand Ende.....	65
Abbildung 34 - Ansicht für einen Antragsteller.....	66
Abbildung 35 - Ansicht für den Vorgesetzten bzw. der Personalabteilung .....	67
Abbildung 36 - Kategorien Übersicht .....	71
Abbildung 37 - Diagramm zu den Gewichtungen der einzelnen Kategorien.....	73

## A.2 Tabellenverzeichnis

Tabelle 1 - Web Service Standards Übersicht .....	23
Tabelle 2 - Abgrenzung der SharePoint Anwendungsgebiete.....	28
Tabelle 3 - Lizenzkosten von Nintex Workflow 2007 .....	44
Tabelle 4 - Partner und Kunden von Nintex.....	44
Tabelle 5 - Funktion einer Phrase und dem zugehörigen Outcome .....	53
Tabelle 6 - Übersicht aller möglichen annehmbaren Zustände und deren Resultat .....	68
Tabelle 7 - Alle Kategorien und deren Kriterien im Überblick .....	72
Tabelle 8 - Berechnung und Staffelung der Noten.....	74
Tabelle 9 - Punktespiegel der Kategorie x.....	75
Tabelle 10 - Auswertung Nintex Workflow 2007 .....	77
Tabelle 11 - Übersicht der Vor- und Nachteile von Nintex Workflow 2007 .....	84
Tabelle 12 - Übersicht der Vor- und Nachteile von K2 blackpoint beta 2.....	88

## A.3 Quellcodeverzeichnis

Quellcode 1 - Ein einfaches XML-Beispiel .....	15
Quellcode 2 - Ein Document Information Item.....	16
Quellcode 3 - Element Information Items.....	16
Quellcode 4 - Ein Attribute Information Item .....	16
Quellcode 5 - Ein Comment Information Item.....	17
Quellcode 6 - Ein Beispiel für ein einfaches XSL-Dokument .....	17
Quellcode 7 - Ein Beispiel für ein einfaches XSD-Dokument.....	19
Quellcode 8 - Eine Auflistung aller wichtigen einfachen Datentypen in XSD.....	19
Quellcode 9 - Ein einfaches Beispiel für einen einfachen Datentypen .....	20
Quellcode 10 - Ein einfaches Beispiel für einen komplexen Datentypen .....	21
Quellcode 11 - Resultierendes XML-Dokument aus dem vorherigen InfoPath-Formular .....	25

## A.4 Formelverzeichnis

Formel 1 - Berechnung: Kategorieergebnis .....	75
Formel 2 - Berechnung eines Kriteriums.....	75
Formel 3 - Berechnung: Kategorieergebnis Beispiel .....	75
Formel 4 - Berechnung: Gesamtergebnis.....	76
Formel 5 - Nintex Workflow 2007 Gesamtergebnis.....	77

# B Anhang

## B.1 Testfragen

Diese Testfragen enthalten die zentralen und wichtigen Fakten dieser Bachelorarbeit. Anhand dieser Fragen kann der eigene Wissensstand überprüft werden.

- ✓ Was wird unter XML verstanden und wie wird es eingesetzt?
- ✓ Wozu werden XSD und XSL verwendet und worin unterscheiden sich beide?
- ✓ Was sind Web Services und wozu dienen sie?
- ✓ Was ist und leistet SharePoint?
- ✓ Wie wird ein Workflow definiert?
- ✓ Was steckt im Kern hinter Nintex Workflow 2007?
- ✓ Womit werden in Nintex Workflow 2007 Workflows gebaut?
- ✓ Womit wird untersucht, wie ein Workflow-Ablauf ausgesehen hat?
- ✓ Was sind die Stärken und Schwächen von Nintex Workflow 2007?
- ✓ Wie integriert sich Nintex Workflow 2007 in die SharePoint Landschaft?
- ✓ Wie verhält sich die Skalierbarkeit von Nintex Workflow 2007?



## B.2 MOSS- und WSS-Funktionalitäten im Vergleich

Feature	WSS 3.0	MOSS 2007 Standard	MOSS 2007 Enterprise	MOSS 2007 Search
<b>Collaboration</b>				
Real Time Präsenz	x	x	x	x
<i>Social Networking Webparts</i> <sup>49</sup>		x	x	x
Standard Websitevorlagen	x	x	x	x
Wikis	x	x	x	x
Blogs	x	x	x	x
Personen- und Gruppenlisten	x	x	x	x
Koordination von Aufgaben	x	x	x	x
Umfragen	x	x	x	x
Dokumentbasierte Zusammenarbeit	x	x	x	x
<b>Portal</b>				
Inhalte für Benutzergruppen zuweisen		x	x	
Site Manager		x	x	
Unterstützung für mobile Geräte	x	x	x	
Portal-Vorlagen		x	x	
Webparts für soziale Netzwerke		x	x	
Integration mit Access, Excel, PowerPoint, Word 2007	x	x	x	x
Integration mit Outlook 2007	x	x	x	x
Integration mit SharePoint Designer 2007	x	x	x	x
<b>Suche</b>				
Unternehmensweite Inhaltssuche		x	x	x
Bewertung der Suchergebnisse nach Relevanz		x	x	x
Administration und Management		x	x	x
Suche nach Personen		x	x	
Suche in via <i>Business Data Catalog</i> <sup>50</sup> (BDC) bereitgestellter Daten			x	
<b>Content Management</b>				
Workflow für Geschäftsdokumente		x	x	
Websitevorlagen für Dokument-Management		x	x	
Integration mit Microsoft <i>Information Rights Management (IRM)</i> <sup>51</sup>		x	x	
Einheitliches Branding für Websites		x	x	
Navigationselemente		x	x	

<sup>49</sup> Ein Social Networking Webpart dient der Kommunikation, beispielsweise mittels eines integrierten Chats. Das Webpart an sich bildet einen Bereich auf einer SharePoint Webseite.

<sup>50</sup> Der Business Data Catalog definiert alle Daten, die aus Fremdsystemen zur Integration gesammelt wurden

<sup>51</sup> IRM definiert Regeln, wie wer auf Dokumente zuzugreifen hat

Diverse Websitevorlagen für Content Management		x	x	
Seitenlayouts		x	x	
Websitevariationen		x	x	
Bibliotheken für Folien		x	x	
<b>Geschäftsprozesse und Formulare</b>				
Browserbasierte Formulare aus InfoPath-Formularen			x	
Zentralisiertes Formularmanagement			x	
Assistent für den Import von Formularen			x	
<i>Deployment-Modell für No Code-Formulare</i> <sup>52</sup>			x	
Kompatibilitätsprüfer			x	
<b>Business Intelligence (BI)</b>				
Integrierte Veröffentlichung von Excel 2007-Mappen			x	
Verwaltung von Excel 2007 Mappen			x	
Business Data Catalog			x	
<i>Business Data Webparts</i> <sup>53</sup>			x	
Business Data-Aktionen			x	
<i>Integriertes BI-Dashbord</i> <sup>54</sup>			x	
<i>Report Center</i> <sup>55</sup>			x	
<i>Key-Performance-Indikatoren (PKI)</i> <sup>56</sup>			x	

<sup>52</sup> No Code Formulare wurden in einem Editor ohne jede Art der Programmierung erstellt, Beispiel InfoPath 2007

<sup>53</sup> Ein Webpart wird verwendet um die gewünschten Informationen in einem bestimmten Bereich des Browsers darzustellen

<sup>54</sup> Ein Dashboard sind letztendlich Seiten, die Reports (Informationen aus Excel-Tabellen) enthalten

<sup>55</sup> Der Report Center dient der Darstellung komplexer Informationen

<sup>56</sup> Key-Performance-Indikatoren werden zur Analyse von Sachverhalten verwendet, um diesen eine Semantik zuzuordnen. Z.B. *ein Projekt lief gut* oder *das Umsatzziel einen Jahres wurde erreicht*

## B.3 Nintex Workflow 2007 Aktionen

Nintex Workflow 2007 bringt standardmäßig die folgenden 58 Aktionen in der aktuellen Version 1.9 mit:

Aktion
<b>Integration</b>
Call web service
Execute SQL
Query BDC
Query Excel Services
Query LDAP
Query XML
Search query
Send / Receive BizTalk
Start Workflow
Stop a Workflow
<b>Libraries and lists</b>
Cancel Checkout
Check in item
Check out item
Copy an item
Create an item
Create List
Delete an item
Log in the history list
Query list
Set approval status
Set field value
Set Item Permissions
Update an item
<b>Logic and flow</b>
Action set
Change State
Filter
For each
Loop
Run If
Run parallel actions
Set a condition
State Machine
<b>Operations</b>
Build dynamic string
Calculate a date
Collection operation

Delay for...
Delay until...
End the workflow
Math operation
Regular expression
Set a variable
Wait for an item update
<b>Provisioning</b>
Add User to AD Group
Create AD Account
Create AD Group
Decommission AD Account
Delete AD Group
Enable LCS/OCS
Provision User on Exchange
Remove User from AD Group
Update AD Account
<b>Publishing</b>
Copy to fileshare
Copy to SharePoint
Submit record
<b>SharePoint profiles</b>
Compile audience
Create audience
Delete audience
Query user profile
Update user profile
<b>Sites and workspaces</b>
Create a site
Delete a site
<b>User interaction</b>
Complete workflow task
Delegate Workflow Task
Request approval
Request data
Request review
Send a notification
Task reminder

# Erklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit im Sinne der Prüfungsordnung nach §22(4) bzw. §24(4) bzw. §25(4) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 01. April 2009

---

Ort, Datum, Unterschrift