

Masterarbeit

Entwicklung eines Management Tools zur Optimierung der Unternehmensstrategie

Christian Maier

Technische Universität Graz
Institut für Wissensmanagement
Institutsvorstand: Univ.-Prof. Dr. rer. nat. Klaus Tochtermann



Betreuer: Ass.Prof. Dipl.-Ing. Dr.techn. Univ.-Doz. Denis Helic

Graz, Oktober 2010

Kurzfassung

Das Ziel dieser Masterarbeit ist es, in einem Unternehmen eine unterstützende Software einzusetzen, die als Management Cockpit dient. Das Management Cockpit implementiert dabei das Konzept der Balanced Scorecard zur Überwachung, Dokumentation und Steuerung der Aktivitäten einer Organisation im Hinblick auf seine Vision und Strategie. Nach einer Evaluierung bestehender Softwareprodukte für Balanced Scorecard, wurde eine GWT-basierende Webanwendung entwickelt. Diese Softwarelösung ermöglicht die Verwaltung von Kennzahlen zur Kontrolle und Steuerung der Unternehmensstrategie. Im Vordergrund steht dabei die Visualisierung der Kennzahlen mittels verschiedener Auswertungen und Trendanalysen. Weichen Kennzahlen von den geplanten Vorgaben zu stark ab, können Maßnahmen über die Softwarelösung eingeleitet werden. Somit ist die Umsetzung der Unternehmensstrategie an zentraler Stelle dokumentiert und für das Management besser kontrollierbar.

Schlüsselwörter: Webanwendung, GWT, Datenbank, Balanced Scorecard, Kennzahlen

Abstract

The goal of this Master's Thesis is the development of a web application that serves as a management cockpit. The management cockpit implements the concept of balanced scorecards to monitor, document and control the activities of an organization towards its vision and strategy. After an evaluation of existing software products for the Balanced Scorecard, a GWT-based web application was developed. This software solution helps to manage key performance indicators for monitoring and controlling the corporate strategy. The visualization of key performance indicators with various reports and trend analysis is focused by the software. If these key performance indicators differ too much from the planned values, actions can be assigned with the software. Furthermore, the control of corporate strategy is documented in a central location to improve the control opportunities by the management.

Keywords: web application, GWT, database, balanced scorecard, key performance indicators

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am

.....
(Unterschrift)

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

.....
date

.....
(signature)

Danksagung

Ich möchte mich herzlich bei allen, die mich bei dieser Arbeit und dem vorausgehenden Studium so tatkräftig unterstützt haben, bedanken. Ohne deren Hilfe hätte diese Arbeit nicht entstehen können.

Im Besonderen gilt dies für meine Eltern, die mir die Ausbildung an der Technischen Universität Graz ermöglicht haben.

Des Weiteren gilt mein Dank dem Initiator dieser Diplomarbeit, Dr. Wolfgang Sattler, der mich in allen Dingen der Ausführung der Arbeit unterstützt hat.

Abschließend möchte ich mich bei Dipl.-Ing. Dr.techn. Univ.-Doz. Denis Helic für die Betreuung dieser Arbeit und seine fachliche und organisatorische Unterstützung recht herzlich bedanken.

Graz, Oktober 2010

Christian Maier

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	2
1.2	Disposition	2
2	Web	3
2.1	Web 1.0	4
2.2	Web 2.0	4
3	Webanwendungen	7
3.1	Entwicklung der Webprogrammierung	7
3.1.1	Webserver	8
3.1.2	Webclient	8
3.1.3	Protokoll	8
3.1.4	HTML	8
3.2	Rich Internet Applications	9
3.2.1	Einsatzbereiche und Technologien	9
3.2.2	Architektur von RIAs	11
3.2.3	Vor- und Nachteile von RIAs	11
3.3	Clientseitige Webentwicklung	13
3.3.1	Ajax	13
3.3.2	Google Web Toolkit	15
3.3.3	Flash	17
3.3.4	Microsoft Silverlight	19
3.3.5	Java Applets	20
3.4	Serverseitige Webentwicklung	22
3.4.1	Java Server Pages	22
3.4.2	ASP.NET	23
3.4.3	PHP	24
3.5	Webserver	26
3.5.1	Apache HTTP Server	26
3.5.2	Internet Information Services	27
3.6	Datenbanken	29
3.6.1	Bestandteile einer Datenbank	29

3.6.2	Datenbankmodelle	30
3.6.3	MySQL	31
3.6.4	Oracle	32
3.6.5	MS SQL Server	33
3.7	Drei Schichten Architektur	33
3.8	Open Source vs. Closed Source	34
4	Balanced Scorecard	36
4.1	Geschichte	37
4.2	Bestandteile der Balanced Scorecard	38
4.2.1	Vision	38
4.2.2	Mission	38
4.2.3	Strategie	39
4.2.4	Perspektiven der Balanced Scorecard	39
4.2.5	Kennzahlen/Indikatoren	44
4.3	Ursachen-Wirkungsketten	45
4.4	Entwicklung einer Balanced Scorecard	46
4.4.1	Strategie klären	46
4.4.2	Strategische Ziele ableiten	47
4.4.3	Ursache-Wirkungs-Beziehungen erkennen	47
4.4.4	Kennzahlen festlegen	47
4.4.5	Zielwerte definieren	48
4.4.6	Strategische Aktionen	48
4.4.7	Ableitung nachfolgender Balanced Scorecards	48
4.5	Balanced Scorecard als Managementinstrument	49
4.5.1	Formulieren und Umsetzen von Vision und Strategie	49
4.5.2	Kommunizieren und organisatorische Ausrichtung	50
4.5.3	Ausführen der Strategie	51
4.5.4	Strategisches Feedback und Lernen	51
5	Umsetzung des Management Tools	53
5.1	IT-Unterstützung für die Balanced Scorecard	53
5.2	Anforderungsanalyse	55
5.2.1	Ziel- und Projektdefinition	55
5.2.2	Use Cases	57
5.2.3	Kriterienkatalog	64
5.3	Implementierung der Applikation	68
5.3.1	Systemumfeld	68
5.3.2	Programmstruktur	69
5.3.3	GWT „Best Practice“ Implementierung	70
5.3.4	Benutzersicht	74

6	Softwaretests und Resultate	82
6.1	Fehler und Verbesserungen	82
6.1.1	Mandantenfähigkeit	82
6.1.2	Berechtigungsmatrix	83
6.1.3	Berechnung und Darstellung der KPI Baumstruktur	83
6.1.4	Verbesserungen in der Darstellung	84
6.1.5	Logout Button	85
6.1.6	Zusätzliche Benachrichtigungen	85
6.2	Evaluierung der Software	86
6.2.1	Definition Evaluierung	86
6.2.2	Warum sollte evaluiert werden?	87
6.2.3	Durchführung einer Software-Evaluierung	88
6.3	Planung der Evaluierung	89
7	Zusammenfassung und Ausblick	90
A	Anhang	92
A.1	Source Code	92
A.1.1	Kennzahlen-Hierarchie Berechnung	92
	Literaturverzeichnis	98

Abbildungsverzeichnis

3.1	Einsatzbereiche für RIA Technologien	10
3.2	Technologie-Optionen für RIAs	10
3.3	Architektur-Optionen für RIAs	12
3.4	Klassische Kommunikation	14
3.5	Ajax Kommunikation	15
3.6	RPC Mechanismus mit Klassen und Interfaces	18
3.7	Flash Entwicklungsumgebung	19
3.8	Silverlight Architektur	20
3.9	Softwarearchitektur im Web mit einem Applet	21
3.10	JSP Modell 1 Architektur	23
3.11	JSP Modell 2 Architektur	24
3.12	ASP.NET Architektur	25
3.13	Client Server Kommunikation	25
3.14	Apache Architektur	27
3.15	IIS Architektur	28
3.16	Schalenmodell eines Datenbanksystems	31
3.17	Komponenten von MySQL	32
3.18	Beispiel einer Drei-Schichten-Architektur	34
3.19	Verteilungsalternativen beim Client/Server Konzept	35
4.1	Erste nicht finanzielle Scorecard von Analog Devices Inc	37
4.2	Perspektiven der Balanced Scorecard	40
4.3	Das generische Wertkettenmodell	42
4.4	Ursachen-Wirkungskette in der BSC	45
4.5	Ableitung der Balanced Scorecard	49
4.6	BSC als strategischer Handlungsrahmen	50
4.7	Arten des Lernens	52
5.1	Kategorien von Softwarelösungen für die Balanced Scorecard	54
5.2	Kategorien von Softwarelösungen für die Balanced Scorecard im deutschsprachigen Raum	54
5.3	Beispiel der Baumstruktur einer Kennzahl	60
5.4	Systemumfeld der Applikation	69
5.5	Struktur der Applikation	70

5.6	Systemübersicht	71
5.7	GWT Remote Procedure Call	72
5.8	Registerkarten der Hauptseite	74
5.9	Visualisierung der Strategiemappe	75
5.10	Kennzahlen-Übersicht	77
5.11	Kennzahlen-Details	78
5.12	Kennzahlen erstellen/bearbeiten	79
5.13	Planwerte erstellen/bearbeiten	80
5.14	Maßnahmen-Übersicht	81
6.1	Erfassungsdatum der Istwerte	85
6.2	Spaltenbreite und Scrollbalken	85
6.3	Zusätzliche Benachrichtigungen	86
6.4	Formative und summative Evaluierung während des „Software-Lifecycle“	88

Tabellenverzeichnis

2.1	Web 1.0 vs. Web 2.0 Anwendungen nach O'Reilly	6
4.1	Beispiel für die Finanzperspektive einer BSC	41
4.2	Beispiel der Kundenperspektive einer BSC	42
4.3	Beispiel der internen Prozessperspektive einer BSC	43
5.2	Funktionale Kriterien der Applikation	67
5.1	Technische Kriterien der Applikation	68
6.1	Berechtigungsmatrix der Applikation	83

Abkürzungsverzeichnis

Ajax	Asynchronous JavaScript and XML
ASP	Active Server Pages
BSC	Balanced Scorecard
CLR	Common Language Runtime
CSS	Cascading Style Sheets
GWT	Google Web Toolkit
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JSP	Java Server Pages
JVM	Java Virtual Machine
KPI	Key Performance Indicator
PHP	Hypertext Preprocessor
RIA	Rich Internet Application
XAML	Extensible Application Markup Language
XML	Extensible Markup Language

Kapitel 1

Einleitung

Durch die schnell wechselnden Umgebungsbedingungen und sich ändernden Kundenanforderungen stehen Unternehmen ständig im Wettstreit mit ihren Mitbewerbern. Um dabei konkurrenzfähig zu bleiben, ist die Strategie eines Unternehmens von großer Bedeutung. Für die Umsetzung der Unternehmensstrategie gibt es mittlerweile einige Konzepte. Eines der bekanntesten Modelle ist die Balanced Scorecard von Kaplan/Norton. Dadurch ist es möglich, die Erfolgsfaktoren und Leistungstreiber darzustellen, zu messen und entsprechende Maßnahmen einzuleiten. Durch die Messbarkeit der Strategie und die Darstellung der Ziele mittels visualisierter Kennzahlen, können diese operativ gesteuert werden.

Um beste Ergebnisse erzielen zu können, muss die Balanced Scorecard im gesamten Unternehmen implementiert und von den Mitarbeitern regelmäßig benutzt werden. Hierfür spielen Software-Applikationen eine besondere Rolle. Software-Applikationen sollen besonders die Umsetzung, Kontrolle und Anpassung der Unternehmensstrategie im Management unterstützen. Balanced Scorecard Anwendungen sammeln dabei verteilte Daten, überwachen die Kennzahlen und analysieren die gesammelten Daten des Unternehmens. Dadurch haben alle Mitarbeiter Zugang zu der Strategie der Organisation und werden durch Feedbacks der Anwendung bei ihren Aufgaben unterstützt.

1.1 Motivation

Bisher wurde im Unternehmen *Microsoft Excel* zur Abbildung der Balanced Scorecards verwendet. Zwar kann damit ein Teil der Funktionalität bereit gestellt werden, jedoch haben Tabellenkalkulationsprogramme deutliche Nachteile hinsichtlich Skalierbarkeit, Dateneingabe, Kommunikation bzw. Kollaboration, sowie erschwerte Analysefähigkeiten bei großen Datenmengen. In den letzten Jahren wurden viele BSC-Software Produkte entwickelt, die aber größtenteils nur Standard-Funktionalität bieten oder viel zu umfangreiche und teure „Business-Intelligence-Systeme“ sind. Die speziellen Anforderungen des Unternehmens können nur teilweise durch am Markt vorhandener Software abgedeckt werden, wodurch eine Individuallösung notwendig ist.

1.2 Disposition

In **Kapitel 2** wird die Entwicklung des Web beschrieben. Der Fokus wird hier auf die Unterschiede und Möglichkeiten von Web 1.0 und Web 2.0 gelegt.

Kapitel 3 befasst sich mit der Entwicklung von Webanwendungen. Dabei werden die notwendigen Komponenten einer Webanwendung beschrieben. Des Weiteren werden unterschiedliche Webtechnologien erörtert und Konzepte der Softwareentwicklung erläutert.

In **Kapitel 4** wird der theoretische Hintergrund zur Balanced Scorecard genauer betrachtet. Ausgehend von den Bestandteilen einer BSC wird die Entwicklung einer BSC in einer Organisation erklärt.

In **Kapitel 5** wird die Umsetzung der Software beschrieben. Nach der Anforderungsanalyse der Anwendung folgt eine genaue Beschreibung der Struktur und Implementierung der Webanwendung.

Abschließend wird in **Kapitel 6** die Arbeit zusammengefasst und ein kurzer Ausblick in die Zukunft gegeben.

Kapitel 2

Web

Die Geburt des *Internets* wird auf das Jahr 1969 datiert und ist somit etwa 20 Jahre vor dem *World Wide Web* entstanden. Paul Baran und Donald Watts Davies lieferten die ersten theoretischen Arbeiten für ein „Netz“ in den sechziger Jahren. Sie hatten die Idee eines paketbasierten, selbstorganisierten Netzwerkes. Grund dafür dürfte die Notwendigkeit von ausfallsicheren Netzen zur Zeit des Kalten Krieges gewesen sein. 1969 wurden in den Vereinigten Staaten vier Universitäten durch ein WAN (Wide Area Network) miteinander verbunden. Vier Jahre später bildeten bereits 40 Universitäten das sogenannte ARPANET. In Deutschland wurde 1984 der „Verein zur Förderung eines deutschen Forschungsnetzes e.V.“ (DFN) gegründet, der bis heute die deutschen Forschungseinrichtungen und Hochschulen versorgt. Das Internet bietet eine große Anzahl an verschiedenen Diensten wie telnet oder FTP, der wichtigste ist jedoch „das Web“ (oder WWW). Das WWW hat gegenüber anderen Diensten folgende Vorteile [1]:

- Hyperlinks um auf Dokumente zu verweisen
- Multimediale Komponenten für Texte, Grafiken, Audio und Video

Die Idee für das WWW hatte der in London geborene Physiker Sir Timothy J. Bernes-Lee, der im Jahre 1989 den Informationsdienst Gopher an der europäischen Forschungseinrichtung CERN ablösen wollte. Somit war die erste Webadresse überhaupt: <http://info.cern.ch>. [1]

Sein Konzept des World Wide Web basiert auf der Zusammenarbeit von mehreren Komponenten [1]:

- Webserver
- Webclient
- Vermittlungsprotokoll (HTTP)
- Formatierungssprache (HTML)

In der Kurzfassung seines Vorschlags zum *World Wide Web* schreibt Tim Berners-Lee folgendes:

„HyperText is a way to link and access information of various kinds as a web of nodes in which the user can browse at will. Potentially, HyperText provides a single user-interface to many large classes of stored information such as reports, notes, data-bases, computer documentation and on-line systems help. We propose the implementation of a simple scheme to incorporate several different servers of machine-stored information already available at CERN, including an analysis of the requirements for information access needs by experiments.“

(Tim Berners-Lee, 12. November 1990 [2])

2.1 Web 1.0

Erst nachdem der Begriff Web 2.0, das sogenannte „Mitmach-Internet“ definiert wurde, ergab sich die äquivalente Kurzbezeichnung Web 1.0 für das ursprüngliche World Wide Web. Das World Wide Web (WWW) ist ein Hypertext-System, das die Darstellung verschiedener Medien wie Bilder und Text ermöglicht. Diese multimediale Visualisierung wurde erst mit dem Browser Mosaic möglich. Web 1.0 basiert fast ausschließlich auf (X)HTML-Dokumenten, die in einer Baumstruktur hierarchisch organisiert sind. Web 1.0 wurde nur von wenigen Benutzern mit Inhalten versorgt und hauptsächlich zum Empfangen von Informationen und Daten genutzt. [3]

2.2 Web 2.0

Der Begriff Web 2.0 wurde vom Verleger Tim O’Reilly im Jahr 2004 geprägt. In einer Brainstorming-Session ging es darum, Prinzipien von Firmen zu identifizieren,

die den Crash der New Economy erfolgreich bewältigt hatten. Dabei wurden die Veränderungen, die das Web durchlebt hat, gar nicht so genau betrachtet. Viel mehr beschreibt O'Reilly die Kernkompetenzen, von denen erfolgreiche Firmen mindestens eine besitzen [4] :

- Nutzung des Webs als Plattform
- Einbeziehung der kollektiven Intelligenz der Nutzer (zB: durch Blogs)
- Zugang zu schwer erreichbaren Daten
- Entwicklung von Software mit Einbeziehung von Usern
- Software, die auf mehreren Geräten einsetzbar ist

O'Reilly ordnet typische Web 2.0 Dienste klassischen Internet Anwendungen zu, die denselben Zweck erfüllen. Als Beispiel werden *Double Click* und *Google AdSense* genannt. Beide Applikationen dienen der Vermarktung von Internet Werbung. Jedoch verwendet *Double Click* klassische Werbeflächen wie Banner usw. ohne den Besucher der Webseite zu berücksichtigen. *Google AdSense* reagiert auf die Suchbegriffe der Nutzer und ermöglicht somit kontextabhängiges Marketing und eine höhere Erfolgswahrscheinlichkeit. Dasselbe Prinzip der Interaktion mit dem Benutzer findet sich auch bei Wikis wie *Wikipedia* wieder. Solche Systeme bieten vielen verschiedenen Autoren die Möglichkeit gemeinsam ein Dokument zu erstellen bzw. zu bearbeiten. *Tagging* ist ein weiteres Anwendungsgebiet der Miteinbeziehung von Usern. Man versteht darunter die Kennzeichnung von beliebigen Inhalten mit Metadaten durch den Benutzer. Ein weiteres Merkmal von Web 2.0 ist die Verteilung bzw. Dezentralität von Daten. *Mp3.com* beinhaltete ein zentrales Musikangebot, bei dem sich die Benutzer die Musik von einer Webseite herunterladen konnten. *Napster* jedoch verfolgte die Strategie eines Peer-to-Peer Netzwerkes, bei dem jeder Nutzer gleichzeitig Anbieter und Kunde von Musikdateien war. Dies war vor allem bei großen Dateien von Vorteil, da sich Inhalte dezentral kostensparender bereitstellen lassen als über zentrale Dienste. Social Networks wie *Xing* und *MySpace* erfüllen in vieler Hinsicht ähnliche Zwecke wie klassische Homepages und haben diese mittlerweile teilweise ersetzt. *Blogs* erleichtern durch Standardbausteine die Gestaltung von Seiten und Veröffentlichung von Inhalten. Durch Querverweise ist es möglich verschiedene *Blogs* miteinander zu vernetzen. Umgekehrt können über Webservices andere Anwendungen wie Wetterdienste oder Nachrichten in der eigenen Webseite integriert werden.

Web 1.0	Web 2.0
DoubleClick	Google AdSense
Ofoto	Flickr
Akamai	Bittorrent
mp3.com	Napster
Britannica Online	Wikipedia
personal websites	blogging
evite	upcoming.org
domain name speculation	search engine optimization
page views	cost per click
screen scraping	web services
publishing	participation
content management systems	wikis
directories (taxonomy)	tagging(“folksnomy“)
stickiness	syndication

Tabelle 2.1: Web 1.0 vs. Web 2.0 Anwendungen nach O’Reilly [6]

Diese Interoperabilität mit gemeinsamen Standards ermöglichen sogenannte Mash-Ups. Mash-Ups sind Kombinationen zweier Dienste zu einem neuen Angebot. Ein Beispiel wäre die Foto-Community *Flickr* mit Geodaten zu versehen, die den Ort der Aufnahme beinhalten. Somit könnte man über eine Schnittstelle die Bilder von Flickr mit der interaktiven Weltkarte *Google Earth* verknüpfen.

Web 2.0 umfasst Webanwendungen und Plattformen, die den Nutzer durch seine eigenen Inhalte, Kommentare und Tags in die Wertschöpfung integrieren. Die wichtigsten Merkmale der Wertschöpfung sind bei Web 2.0 Interaktivität, Dezentralität und Dynamik. Über gemeinsame Standards wird sichergestellt, dass die Zusammenarbeit räumlich und zeitlich verteilter Benutzer möglich ist. [5]

Kapitel 3

Webanwendungen

Eine Webanwendung ist nichts anderes als eine Applikation, die im Web lauffähig ist. Der Begriff Webanwendung stammt aus einer Zeit, in der das Internet noch nicht für komplexe Anwendungen verwendet wurde. Klassische Client-Server Anwendungen wurden ausschließlich auf mehreren Clients, sowie einem oder mehreren zentralen Servern installiert. Erst die Bereitstellung primitiver Anwendungen im Internet, also HTML Oberflächen, die ohne zusätzliche Installation mit einem Browser auf einem PC aufrufbar waren, wurde der Begriff Webanwendung geprägt. Eine Webanwendung ist also Software, die über eine Web-(HTML-) Oberfläche mithilfe eines Browsers ausgeführt werden kann. [7]

3.1 Entwicklung der Webprogrammierung

Die ursprüngliche von Berners-Lee entwickelte World Wide Web Technik kommt mit nur wenigen Komponenten aus. Diese Komponenten sind im Wesentlichen: ein Serverprogramm zur Auslieferung von angeforderten Webseiten, ein Clientprogramm zur Darstellung der Informationen, ein Protokoll für die Definition der Kommunikation zwischen Server und Client und eine Auszeichnungssprache zur Codierung der Informationen. Mit diesen simplen Komponenten war es zunächst nur möglich einfache statische Informationen anzuzeigen. Nachstehende Abschnitte erklären diese Kernelemente im Detail. [1]

3.1.1 Webserver

Das zentrale Element im World Wide Web ist die Bereitstellung eines Webserver. Der Webserver ist eine Anwendung, die auf Anfragen (Request) im HTTP-Protokoll reagiert und entsprechende Antworten (Response) erstellt. Der erste Webserver war der CERN-Webserver, welcher später zum Apache weiterentwickelt wurde (siehe Kapitel 3.5). [1]

3.1.2 Webclient

Die beiden wichtigsten Aufgaben eines Webclients sind die Formulierung von Anfragen an den Webserver und die Darstellung dessen Antwort. Der erste Webclient war der Mosaic-Browser, der von Marc Andreessen entwickelt wurde. Andreessen gründete die Firma Netscape und der Mosaic-Browser wurde zum Netscape-Browser weiter entwickelt. [1]

3.1.3 Protokoll

Damit die Kommunikation zwischen Webclient und Server funktioniert, muss eine Sprachebene definiert werden. Dies beginnt bei den netzwerktechnischen Bestandteilen bis hin zu den gültigen Befehlen. Diese Befehle umfassen zum Beispiel die Anforderung eines Dokuments, sowie die möglichen Antwortarten des Servers. Die physische Ebene der Kommunikation wird durch TCP/IP festgelegt, die eigentliche Kommunikation basiert auf dem HTTP-Protokoll. Zusätzlich kommt noch das Adressierungsschema der URL respektive URI hinzu. [1]

3.1.4 HTML

Für die Formatierung und Darstellung der Informationen am Client ist eine Auszeichnungssprache notwendig. HTML und die Möglichkeit Grafiken und Hyperlinks zu visualisieren ist ein zentraler Grund für den Erfolg des World Wide Web. Mit HTML war es jedoch nur möglich statische Informationen abzubilden. Deshalb war es notwendig eine erweiterte Programmierung bereitzustellen. Auf diese unterschiedlichen Technologien wird in den folgenden Abschnitten eingegangen. [1]

3.2 Rich Internet Applications

Rich Internet Applications sind Webapplikation, die eine vielfältige Menge an Interaktionsmöglichkeiten mit der Benutzeroberfläche bieten. RIA sind vor allem durch die Internetdienste „Flickr“, „Google Maps“ oder „MySpace“ bekannt. Diese neue Entwicklungsstufe der Webtechnologie erweitert den Anwendungsbereich für Webapplikationen innerhalb von Unternehmen massiv. Geschäftsprozesse lassen sich erheblich einfacher automatisieren oder alte Mainframe-basierte Informationsdienste können abgelöst werden. Nachdem vor allem das World Wide Web als globale Informationsplattform dient, wurden in den letzten zehn Jahren auch immer mehr Unternehmensapplikationen auf Basis von Webtechnologien eingesetzt. Webapplikation bieten viele betriebliche Vorteile, wie das zentrale Management von Applikationen. Vor allem der von Jesse James Garrett geprägte Begriff „Ajax“ machte JavaScript basierte Internettechnologie salonfähig und das Thema „Rich Internet Applications“ wurde in der breiten Öffentlichkeit bekannt. Der RIA-Trend wurde noch zusätzlich von der fast gleichzeitig entstehenden „Web 2.0“ Welle unterstützt. Da der User bei Web 2.0 nicht nur Inhalte konsumiert sondern auch Web-Content beisteuern kann, werden RIAs benötigt. [8]

3.2.1 Einsatzbereiche und Technologien

Die Implementierung von Formularen, Tabellen und Reitern ist mit HTML relativ einfach. Sobald aber Baumstrukturen oder eine mehrstufige Navigation benötigt werden, können befriedigende Lösungen nur mit erheblichem Aufwand erstellt werden. Tabellen, die sehr viele Daten beinhalten, müssen eine Logik bereitstellen, damit nur bestimmte Elemente an den Client übertragen werden („lazy loading“). Dasselbe Problem gibt es auch bei Seiten, die partiell aktualisiert werden. Hier ist ebenfalls ein asynchrones Laden der Daten notwendig. Diese Techniken können nur mit RIA-Technologien implementiert werden.

Abbildung 3.1 veranschaulicht die verschiedenen Anforderungen an Webanwendungen und welche Technologien für die Implementierung eingesetzt werden können.[8] Können die Anforderungen an das User Interface mit einer einfachen HTML-Lösung nicht abgedeckt werden, muss eine geeignete Technologie zur Umsetzung einer Rich Internet Application ausgewählt werden. Grundsätzlich erfolgt die Einteilung von RIA Anwendungen in drei Klassen [8]:

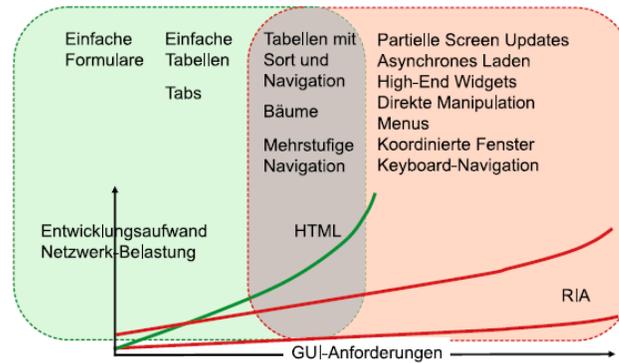


Abbildung 3.1: Einsatzbereiche für RIA Technologien [8]

- Ajax-Anwendungen mit JavaScript
- Reine Java Anwendungen (zB: Applet)
- Adobe Flash Applikationen

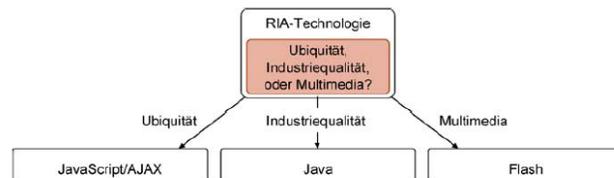


Abbildung 3.2: Technologie-Optionen für RIAs [8]

Die Auswahl der Technologie sollte ausschließlich aufgrund der konkreten Anforderungen und den Randbedingungen erfolgen (siehe Abbildung 3.2).

Ubiquität: *Ist die Anwendung beliebigen Benutzern des Internets zugänglich oder wird ein bestimmter Benutzerkreis definiert?*

Soll die Anwendung für jedermann zugänglich sein, sollten zusätzliche Installationen am Client vermieden werden. Hier bieten sich reine Ajax/JavaScript-Lösungen an.

Industriequalität: *Welche Anforderungen werden bezüglich Wartbarkeit, Stabilität, Verfügbarkeit, Performance und Sicherheit an die Anwendung gestellt?*

Wird die Anwendung nur einem bestimmten Benutzerkreis zur Verfügung gestellt,

stellt die zusätzliche Installation einer Laufzeitumgebung kein Problem dar. In diesem Fall bieten sich Java-basierte Lösungen an, die eine objektorientierte Technologie und ausgereifte GUI Bibliotheken mit sich bringen.

Multimedia: *Ist ein besonderes Design notwendig? Werden in die Anwendung Animationen oder andere multimediale Komponenten integriert?*

Für Anwendungen mit vielen multimedialen Elementen werden Flash-basierende Webanwendungen empfohlen. Hier ergibt sich jedoch der Nachteil, dass ein Plugin für den Browser notwendig ist. Des Weiteren ist die Applikationsentwicklung um einiges komplexer als bei reinen Java-Anwendungen, da hier unterschiedliche Technologien (Java, ActionScript, XML, SWF u.v.m.) verwendet werden.[8]

3.2.2 Architektur von RIAs

Die vielen unterschiedlichen Technologievarianten von RIAs bringen auch verschiedene RIA-Architekturen mit sich. Diese reichen von *Rich Fat Clients* bis hin zu *Rich Thin Clients* (siehe Abbildung 3.3). Klassische, rein HTML-basierte Webanwendungen sind ganz rechts in Abbildung 3.3 veranschaulicht. Bei diesem Architekturtyp wird die gesamte Präsentations- und Geschäftslogik am Server implementiert. Am Client ist somit nur eine „HTML-Rendering-Engine“ (Browser) notwendig, um die Anwendung anzeigen zu können. Bei der *Rich Thin Client* Applikation ist ebenfalls die gesamte Applikation serverseitig implementiert, jedoch wird am Client eine mächtigere „Rich User Interface Rendering Engine“ benötigt. Diese ist entweder separat auf dem Client oder als Plugin für den Internetbrowser zu installieren. Alle anderen Architekturvarianten unterscheiden sich vom *Rich Thin Client* dadurch, dass immer mehr Logik vom Server auf dem Client ausgelagert wird. Im extremsten Fall, dem *Rich Fat Client*, befindet sich sämtliche Applikationslogik auf dem Client und es werden nur noch die Daten auf einem Server gehalten.[8]

3.2.3 Vor- und Nachteile von RIAs

Mit RIA Technologien ist es möglich, das Applikationsportfolio zu konsolidieren. So gab es in der Vergangenheit in vielen Unternehmen sowohl *Rich Fat Client*, als auch Webapplikationen um die unterschiedlichen Anforderungen abdecken zu können. Mit Rich Internet Applications können diese Anforderungen mit verschiedenen Benutzerkreisen auf einer Plattform erfüllt werden. RIA-Technologie ermöglicht die

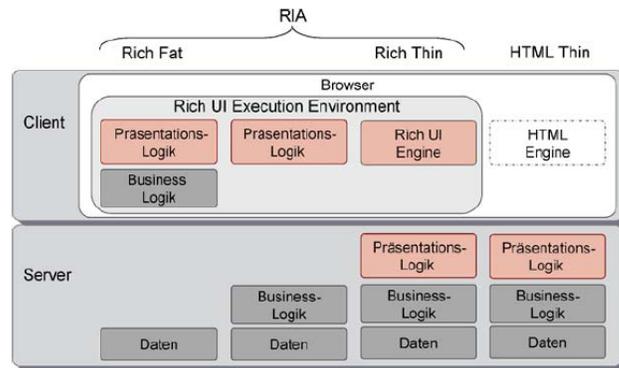


Abbildung 3.3: Architektur-Optionen für RIAs [8]

sogenannte „Software aus der Steckdose“. Gemeint ist damit „Software as a Service“, bei dem reichhaltige Benutzerschnittstellen mit Webtechnologie kombiniert und serverzentrierte Lösungen erstellt werden. Somit könnte die Software „vermietet“ werden, wodurch neue Geschäftsmodelle für den Vertrieb von Applikationen entstehen. Des Weiteren besteht die Möglichkeit „Rich Graphical User Interfaces“ zu erstellen. Diese Rich GUIs könnten in Kombination mit einer serviceorientierten Architektur (SOA) das Frontend von Legacy-Systemen ablösen. Schon seit vielen Jahren versuchen Unternehmen die Frontends ihrer Legacy-Applikationen mit Webtechnologien zu ersetzen. Jedoch mussten hier aufgrund der Unzulänglichkeiten von HTML oftmals Kompromisse hinsichtlich der Benutzbarkeit der Anwendungen eingegangen werden. Einige RIA Technologien erlauben den Offlinebetrieb von Webanwendungen, d.h. sie müssen nicht unbedingt zu jedem Zeitpunkt mit dem Webserver verbunden sein.

Werden RIA Technologien im Internet eingesetzt, können sich eventuell lange Download- bzw. Ladezeiten für die Anwendung ergeben. Durch die große Datenmenge der RIAs, entsteht auch eine höhere Ressourcenbelastung für den Client-Rechner. Einige RIA Technologien wie *Flash* oder *Microsoft Silverlight* benötigen ein Plugin im Browser. Somit ergeben sich hier wiederum Sicherheitslücken, die von Hackern ausgenutzt werden könnten. [8], [9]

3.3 Clientseitige Webentwicklung

Webanwendungen können sowohl client- als auch serverseitig erfolgen. Meistens ist eine Kombination aus beiden Ansätzen notwendig. Die clientseitige Webentwicklung beinhaltet die direkte Interaktion von Benutzern mit aktiven Programmen im Browser. Ein Beispiel hierfür wäre die Reaktion auf einen Mausklick oder direkte Überprüfung von Eingabedaten vor der Übertragung zum Server. Durch die asynchrone Kommunikation ist es möglich ein ähnliches Verhalten wie bei einer Desktop-Applikation im Browser anzubieten. Die wichtigsten Kerntechnologien in diesem Bereich sind [1]:

- (X)HTML und CSS
- GWT und JavaScript
- Flash-Techniken und ActionScript
- Ajax
- Java Applets
- Microsoft Silverlight

Mit diesen Technologien, insbesondere mit Java Applets, kann eine direkte Kommunikation mit einem Datenbanksystem hergestellt werden.. Dies wird jedoch aus Sicherheitsgründen heute nicht mehr angewendet. [1]

3.3.1 Ajax

Ajax, also „Asynchronous JavaScript and XML“, wurde nicht neu entwickelt, sondern ist eigentlich eine Kombination aus bereits bekannten Technologien. Diese Technologien sind XHTML, CSS, DOM, XML, XMLHttpRequest und Javascript. Mittels Ajax können Daten zwischen Client und Server dynamisch ausgetauscht werden ohne, dass lange Wartezeiten für den User entstehen. Die immer schnelleren Desktop-Rechner und die hohen Übertragungsraten im Internet bilden die technische Basis für Ajax. Das Konzept von Ajax beinhaltet eine clientseitige Webtechnologie, die Desktop-Applikationen ähnelt und dabei keine zusätzlichen Plugins für den Browser benötigt. AJAX Anwendungen sollen die Vorteile von Web- und Desktop-Applikationen verbinden. Webanwendungen können auf Services und Informationen

des Internets zugreifen, was bei Desktop-Anwendungen nicht machbar ist. Für jeden Datenabruf muss immer eine Anfrage (Request) getätigt werden. Nachdem eine Verbindung zum Webserver aufgebaut wurde, arbeitet dieser die empfangene Anfrage ab und schickt eine Antwort (Response) zum Client zurück. Dieser Vorgang kann unter Umständen mit langen Wartezeiten verbunden sein, welche mit Ajax vermieden werden können. Ajax Applikationen sollen ein ähnliches Laufzeitverhalten wie Desktop-Anwendungen erreichen, indem immer nur bestimmte Teilbereiche des User Interfaces aktualisiert werden.[10]

3.3.1.1 Klassische Kommunikation

Die klassische Kommunikation zwischen Webserver und Client (Browser) erfolgt über Anfragen (Requests) und Antworten (Responses). Über diese beiden Objekte wird der komplette Datenaustausch geregelt. [10]

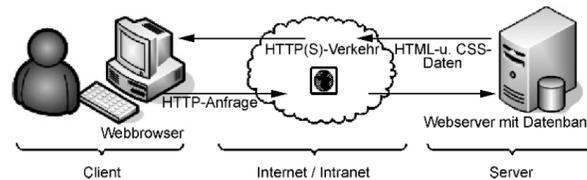


Abbildung 3.4: Klassische Kommunikation [10]

Der Kommunikationsverlauf zwischen Client und Webserver wird in Abbildung 3.4 dargestellt. Der Benutzer löst durch eine Interaktion mit dem User Interface des Webbrowsers eine Anfrage an den Server aus. Die Kommunikation zwischen Server u Client ist im HTTP-Protokoll geregelt. Nachdem der Server den Request erhalten hat, werden die gewünschten Daten über einen Response an dem Client gesendet. Der Browser interpretiert die Daten und stellt diese für den Benutzer dar. Nach jedem Erhalt eines Response vom Server muss der Webclient die komplette Seite neu aufbauen, was eventuell sehr zeitintensiv ist. [10]

3.3.1.2 Ajax Kommunikation (XMLHttpRequest)

Den Kern einer Ajax-Anwendung bildet der sogenannte „XMLHttpRequest“. Dieses Objekt ist eine asynchrone Anfrage an den Server und ermöglicht das dynamische

Nachladen von Daten. Wird ein XMLHttpRequest von einer JavaScript Anwendung an den Server gesendet, kann der Client den Antwortstatus überwachen. Bei HTTP handelt es sich um ein zustandsloses Protokoll und jede Verbindung zum Server wird nach der Kommunikation beendet. Deshalb ist es schwierig eine Sitzung zu verfolgen. Ajax bietet Methoden um dieses Problem umgehen zu können und das Nachladen von Daten vereinfachen. Bei einer Ajax-Anwendung wird eine spezielle „Ajax-Engine“ eingefügt, welche die Kommunikation zwischen Client und Server beschleunigt (siehe Abbildung 3.5). [10], [11], [12]

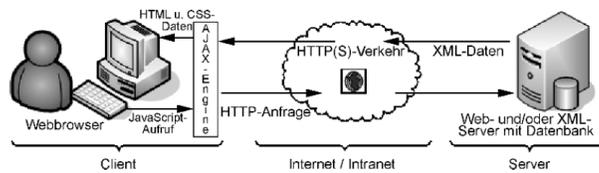


Abbildung 3.5: Ajax Kommunikation [10]

Sobald ein Client einen HttpRequest an den Server versendet, wird zuerst die Ajax-Engine kontaktiert. Bei Anfragen, die nicht zwingend eine Verbindung mit dem Server benötigen, kann die Ajax Engine die Antwort für den Client generieren. Ein Beispiel dafür wäre die Gültigkeitsüberprüfung von Eingabefeldern. Muss der Client mit dem Server kommunizieren, so wird ein XMLHttpRequest-Objekt an den Webserver gesendet. Nachdem der Webserver den Request abgearbeitet hat, muss der Response in einem Ajax-Engine verständlichen Format zurück gesendet werden. Die Ajax-Engine ist nichts anderes als ein JavaScript Objekt, welches die Daten vom Server empfängt, analysiert und diese dynamisch auf der bestehenden Seite im Browser darstellt. Diesen Prozess bezeichnet man als „Parsen“ und wird im Hintergrund, ohne dass es der Benutzer bemerkt, durchgeführt. Bei der Ajax-Kommunikation werden immer nur Teile einer Webseite ausgetauscht, was eine schnelle und flexible Datenaktualisierung ermöglicht. [10], [1]

3.3.2 Google Web Toolkit

Google Web Toolkit (GWT) ist eine RIA-Technologie, die auf einer HTML- und JavaScript-Architektur basiert. Bei GWT wird die Anwendung selbst in Java geschrieben und es folgt eine Konvertierung während der Laufzeit zu JavaScript. Die

erste Version von GWT wurde im Mai 2006 von Google veröffentlicht. Die GWT Bibliothek wurde entwickelt, um die Programmierung von Ajax Anwendungen zu vereinfachen respektive zu beschleunigen. Die wichtigsten Features von GWT sind [13], [14]:

- Volle Unterstützung von Java 5
- Ein „Java-to-JavaScript“ Compiler mit „hosted mode“ Simulation für das Debuggen der Applikation
- Browserunabhängigkeit
- Basis-Widget Support (zB: Button, Formulare, einfache Tabellen, usw.)
- Viele Möglichkeiten für die Kommunikation mit dem Server (RPC, XML, JSON)
- JUnit Tests
- Internationale Sprachunterstützung

Die Entwicklung von Ajax Anwendungen mit traditionellen Technologien ist sehr schwierig und fehleranfällig. Viele Frameworks versuchen eine möglichst hohe Abstraktion anzubieten, jedoch muss in vielen Fällen weiterhin in JavaScript implementiert werden. Das Problem bei der Programmierung in JavaScript liegt darin, dass sich der Code in den einzelnen Browsern unterschiedlich verhalten kann. Somit entsteht ein erheblicher Mehraufwand, wenn man die Browserunterschiede per Code berücksichtigen muss.

Genau hier liegt der große Vorteil von GWT, denn durch die Abstraktionsschicht bei GWT wird keine Implementierung in JavaScript benötigt und die Browserunterschiede werden automatisch berücksichtigt. Sollte man sich entscheiden eine RIA-Anwendung mit GWT zu entwickeln, so müssen folgende Punkte genauer betrachtet werden [15]:

- **Keine Indizierungen durch Suchmaschinen:** Die Indizierung von Ajax bzw. GWT Applikationen funktioniert derzeit noch nicht richtig. Aktuelle Suchmaschinen und deren Crawler können JavaScript Code noch nicht auswerten. Momentan werden nur die Startseiten von Ajax-Anwendungen durchsucht und die dynamischen Elemente der Webseite werden ignoriert.

- **Reine JavaScript Applikation:** Sobald JavaScript im Browser deaktiviert ist, kann die Ajax/GWT Anwendung nicht mehr ausgeführt werden. Hier kann der User nur darüber informiert werden, dass dieser JavaScript im Browser aktivieren muss, um die Anwendung starten zu können. Es ist nicht möglich, eine weniger benutzerfreundliche, beispielsweise reine HTML Version der Applikation bereitzustellen.
- **Keine klare Trennung zwischen Code und Layout:** Viele Frameworks bieten gute Möglichkeiten um das Aussehen der Anwendung („Styling“) und den eigentlichen Code sauber zu trennen. Diese Trennung ist in GWT nicht möglich. Bei GWT Anwendungen kann das Design und die reine Implementierung nicht aufgeteilt werden. Zwar werden Cascading Style Sheets (CSS) unterstützt, diese müssen aber direkt im Java Code angegeben werden.

3.3.2.1 Architektur

Für die Kommunikation zwischen Client und dem Server stellt das Google Web Toolkit mehrere Wege zur Verfügung. Jedoch kann der Datenaustausch am Besten über Remote Procedure Calls durchgeführt werden. Hierbei wird das XMLHttpRequest Objekt benötigt um einen asynchronen Methodenaufruf ausführen zu können. Dieser asynchrone RPC hat zur Folge, dass eine weitere Methode, die sogenannte Callback Methode, mit dem Ergebnis am Client aufgerufen wird. Das Ergebnis eines RPC kann unterschiedliche Formate wie XML, HTML, reinen Text oder JSON beinhalten. Damit diese Art von Kommunikation funktionieren kann, müssen mehrere Schnittstellen definiert werden (siehe Abbildung 3.6). [16], [17]

GWT erstellt auf der Clientseite eine Proxy-Klasse, die das Interface *ServiceDefTarget* und *UnserServiceAsync* implementiert. *UnserServiceAsync* ist dabei die Beschreibung der asynchronen Schnittstelle der Anwendung. Diese asynchrone Schnittstelle benötigt ein synchrones Interface (*UnserService*), welches von *RemoteService* abgeleitet ist. Dieses synchrone Interface gibt dabei die Implementierung des Servers vor. [17]

3.3.3 Flash

Flash ist eine clientbasierte Technologie, die besonders den Bereich der Animationen im Web fokussiert. Dies können grafische Effekte auf einer Webseite sein oder

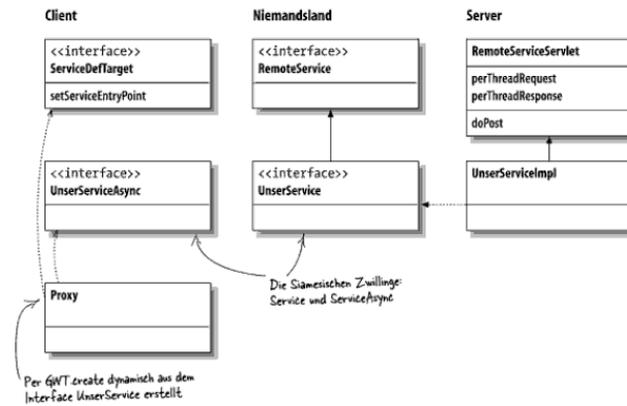


Abbildung 3.6: RPC Mechanismus mit Klassen und Interfaces [17]

ein Intro, also eine kurze Filmsequenz nach dem Laden der Seite. Flash wurde ursprünglich konzipiert um animierte GIF Bilder abzulösen. Seit der Version 4 können einfache clientbasierte Anwendungen mittels *ActionScript* entwickelt werden. Flashanwendungen basieren auf dem „Small Web Format“ (swf), einem vektorbasierten Grafikformat. Deshalb wird zum Abspielen der Flash-Animationen der Flash-Player benötigt. Der Flash-Player ist ein Browser-Plugin, der mittlerweile für alle Plattformen frei verfügbar ist. Für die Programmierung von Flash-Applikationen wird eine spezielle Entwicklungsumgebung benötigt. Diese ist ähnlich wie Filmbearbeitungssoftware aufgebaut. In dem zentralen Filmfenster, der sogenannten Flash-Bühne, werden alle für die Animation notwendigen Objekte abgelegt und über eine Zeitleiste gesteuert (siehe Abbildung 3.7). Die fertige Applikation wird im swf-Format exportiert und über HTML in die Webanwendung eingebunden. [1]

3.3.3.1 ActionScript

Durch die Verwendung von ActionScript können Flash Anwendungen wesentlich umfangreicher gestaltet werden, da Animationen mit einer Programmierung erweitert werden können. Dadurch können zum Beispiel Flash-Animationen durch Eingaben vom Benutzer gesteuert werden. ActionScript ist eine objektorientierte Sprache, dessen Scripts ereignisgesteuert arbeiten. Einzelne Flash-Elemente sind immer Instanzen entsprechender Klassen. Mit ActionScript können komplexe Softwarelösungen realisiert werden, die auch Datenbankanbindungen verwenden. [1]

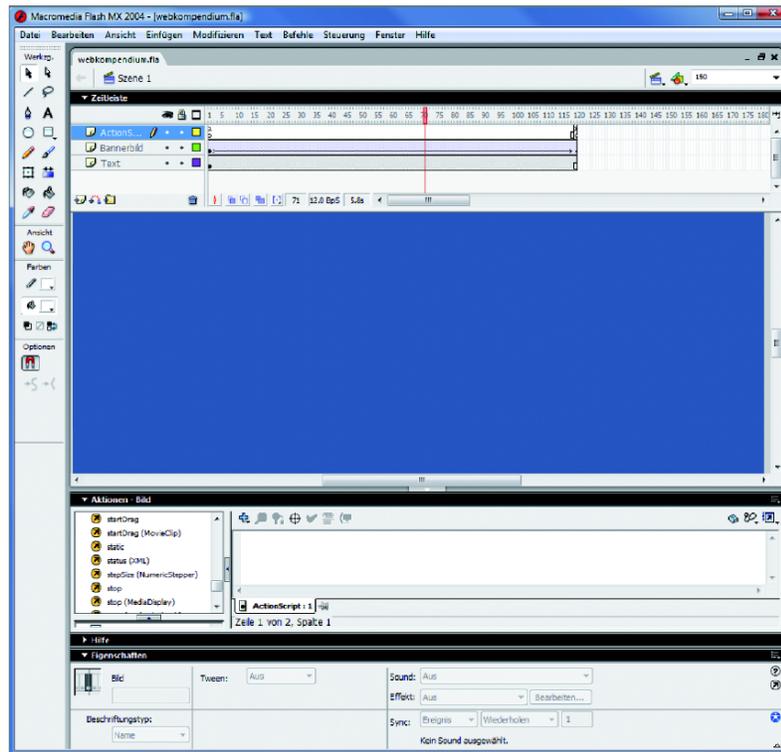


Abbildung 3.7: Flash Entwicklungsumgebung [1]

3.3.4 Microsoft Silverlight

Microsoft Silverlight ist ein weiteres Framework zum Erstellen von clientseitigen Webanwendungen. Zur Darstellung der Applikation am Client wird wie bei Flash ein spezielles Browser Plugin benötigt, das zusätzlich installiert werden muss. Jedoch können durch dieses Plugin weit umfangreichere Anwendungen erstellt werden, als es mit HTML und JavaScript Code möglich ist. Silverlight Webseiten unterstützen interaktive Grafiken, Vektoranimationen, Video- und Audiodateien. Die Idee von Silverlight ist es, die mächtigen Programmiersprachen der .NET Familie mit der Plattformunabhängigkeit von Flash zu kombinieren. Dadurch kann server- und clientseitig dieselbe Programmiersprache verwendet werden (C# oder VB). [18]

3.3.4.1 Architektur

Die Silverlight Plattform besteht im Prinzip aus drei Schichten. Wie in Abbildung 3.8 dargestellt, gibt es für Silverlight ein .NET Framework, das viele Komponenten

und Bibliotheken beinhaltet. Dazu gehören Tools für die Datenanbindung, Controls für die Benutzerschnittstelle und die Common Language Runtime (CLR). Alle Komponenten und Services für die Darstellung des User Interface werden dem sogenannten Präsentationskern zugeordnet. Dazu gehören unter anderem vektorbasierte Animationen, Bilder, Eingaben und das Abspielen diverser Medien.

Die Auszeichnungssprache XAML (eXtensible Application Markup Language) bildet dabei die Schnittstelle zwischen dem .NET Framework und der Präsentationsschicht.

[19]

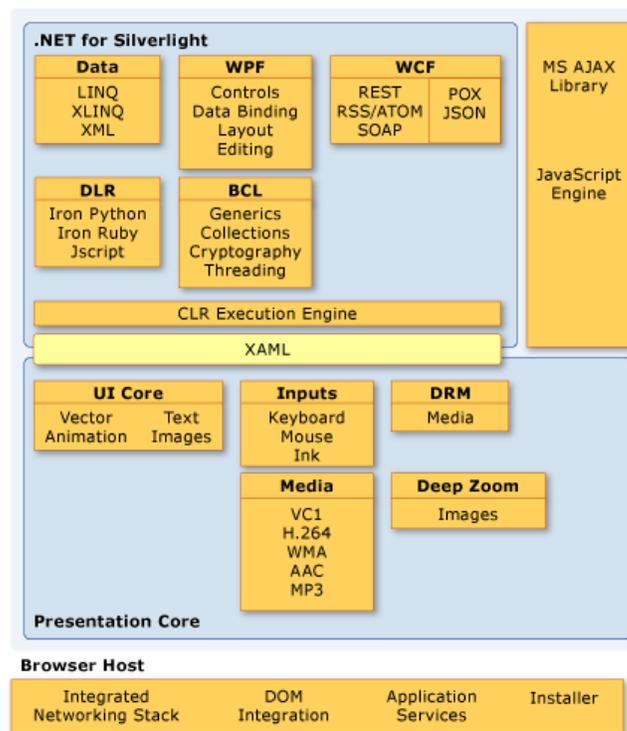


Abbildung 3.8: Silverlight Architektur [19]

3.3.5 Java Applets

Applets sind kleine in Java entwickelte Applikationen, die nicht als eigenständige Programme laufen, sondern in einer HTML Seite eingebettet sind. Ein Applet wird nach dem Aufrufen der Webseite von einem Webserver geladen. Damit das Applet überhaupt ausgeführt werden kann, muss eine *Java Virtual Machine* (JVM) inner-

halb des Browsers gestartet werden. Dieses Laden der JVM geschieht jedoch automatisch ohne, dass der Benutzer eine weitere Aktion durchführt. Damit eine JVM gestartet und das Applet ausgeführt werden kann, muss eine Java-Laufzeitumgebung installiert sein. [20], [21]

Applets basieren auf einer grundlegend anderen Softwarearchitektur als die meisten gegenwärtigen Webtechnologien. Nachdem das Applet auf dem Client heruntergeladen wurde, ist die Kommunikation mit dem Webserver nicht mehr nötig (siehe Abbildung 3.9). Ist eine Verbindung mit einer Datenbank notwendig, so kommunizieren Webclient und Datenbankserver direkt miteinander. Dadurch ergibt sich ein wesentliches Problem, denn durch diese direkte Kommunikation muss der Datenbankserver weltweit erreichbar sein und ist somit angreifbar. [1]

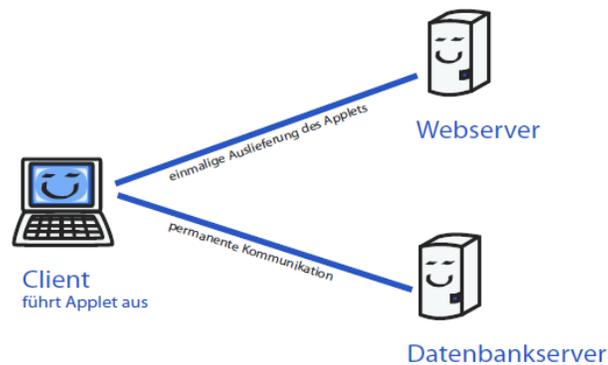


Abbildung 3.9: Softwarearchitektur im Web mit einem Applet [1]

Applets bieten viele Möglichkeiten, jedoch ergeben sich auch gravierende Nachteile [1]:

- Wie schon vorher erwähnt, wird zum Ausführen eines Applets eine *Java Virtual Machine* benötigt. Diese muss, wenn sie nicht im Browser vorinstalliert ist, nachgeladen werden.
- Ist eine JVM bereits am Client vorhanden, kann es trotzdem passieren, dass einige Applets spezielle oder ältere Java Versionen benötigen, die vielleicht nicht mehr verfügbar sind.
- Suchmaschinen können Applets nicht indizieren

- Wie bei JavaScript muss die Ausführung von Applets explizit im Client erlaubt sein.

Dies sind nur einige Gründe, warum Applets heute nicht mehr verbreitet sind. [1]

3.4 Serverseitige Webentwicklung

Serverseitig sind wesentlich komplexere Programmierungen möglich, insbesondere, weil hier direkte Datenbankverbindungen genutzt werden können. Serverseitig ist auch die Anzahl verschiedener Technologien sehr viel höher [1]:

- CGI
- PHP
- Ruby on Rails
- Java Servlets
- Java Server Pages
- ASP.net
- u.v.m.

3.4.1 Java Server Pages

JSP (Java Server Pages) ist eine von Sun Microsystems entwickelte Webtechnologie, mit der dynamisch (X)HTML Dokumente generiert werden können. Diese (X)HTML Dokumente beinhalten dabei statische Elemente, die unabhängig von der konkreten Anfrage definiert werden und dynamische Anteile, die zur Anfragezeit in Abhängigkeit von Parametern berechnet werden. Dabei wird der statische Inhalt direkt als HTML Code formuliert und der dynamische Teil mittels JSP-Anweisungen integriert. Die JSP-Anweisungen sind intern mit der Programmiersprache Java verbunden und werden zur Anfragezeit ausgeführt. Das Ergebnis sind die dynamischen Informationen für das Enddokument, welches schlussendlich aus reinem (X)HTML besteht. [22], [23] Um eine Java Server Page ausführen zu können, ist eine JSP-Engine notwendig. Viele dieser Engines sind frei verfügbar, wie etwa Apache Tomcat [24]. Während der Laufzeit generiert die Engine aus einer JSP ein Servlet. Dieses Servlet stellt den HTML-Code der JSP dar. Die JSP-Engine führt dies in der

Übersetzungsphase (Translation) beim ersten Start und bei jeder Veränderung der Seite durch. Dabei übersetzt ein Webserver die JSP in ein Servlet, welches darauf instanziiert wird. Der vom Servlet generierte HTML Code wird an den Browser geschickt und von diesem angezeigt. [25]

3.4.1.1 Architektur

In der Modell 1 Architektur ist einzig und allein die JSP für die Behandlung von Request und Response verantwortlich. Der Datenzugriff erfolgt, getrennt von der JSP, über Beans (Model). Modell 1 ist ideal für kleinere Anwendungen (siehe Abbildung 3.10). [22]

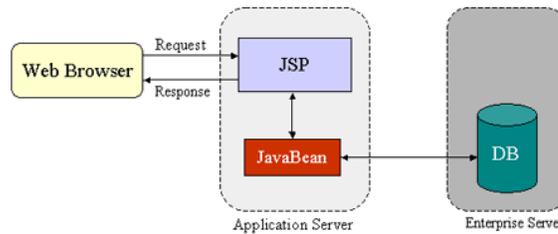


Abbildung 3.10: JSP Modell 1 Architektur [26]

Wie in Abbildung 3.11 dargestellt, werden bei der Modell 2 Architektur *Model*, *View* und *Controller* strikt voneinander getrennt. Der *Controller* (üblicherweise ein Servlet) erhält einen Request vom Browser, instanziiert die notwendigen Beans und ruft die *View* auf. Das *Model* (JavaBean) realisiert, wie bei Modell 1, den Datenzugriff. Die *View* (JSP) holt sich nun nur noch die Daten von den Beans und stellt diese dar. Eine solche Trennung kommt bei größeren Projekten zum Einsatz. Programmierer sind für die Entwicklung der Java-Logik verantwortlich und Designer für die Benutzeroberfläche. [22]

3.4.2 ASP.NET

Microsoft stellte im Jahr 2001 eine neue Technologie vor, die ein modernes Entwickeln von Webanwendungen ermöglichen sollte. ASP.NET ist Teil des .NET Frameworks und unterstützt somit alle objektorientierten Programmiersprachen der .NET Familie. Bei ASP.NET können Programm-Code-Elemente zwar innerhalb von

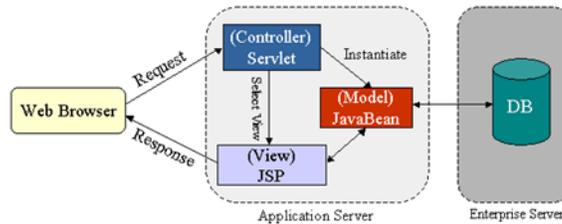


Abbildung 3.11: JSP Modell 2 Architektur [26]

HTML verschachtelt werden, jedoch wird bei ASP.NET ein Code-Behind-Konzept verfolgt. Durch diesen Ansatz können Programm-Code und HTML-Layout vollständig voneinander getrennt werden. Da die ASP.NET Seiten und Komponenten vor der Ausführung kompiliert werden, ergibt sich hier ein Performancevorteil gegenüber den klassischen Technologien. [27], [28]

3.4.2.1 Architektur

Alle Anfragen von Besuchern einer ASP.NET Seite werden an den Webserver gesendet und von diesem entgegen genommen. Dieser leitet den Request an die sogenannte ISAPI-Erweiterung weiter, welche überprüft, ob der von der Webseite getrennte ASP.NET Code bereits durch einen vorangegangenen Aufruf vorkompiliert wurde. Wird die Seite zum ersten Mal aufgerufen, muss das Kompilieren natürlich durchgeführt werden. Durch das Kompilieren kann die Seite mit ihren „Controls“ in der *Common Language Runtime* (CLR) „gerendert“ werden. Das Ergebnis wird über den Webserver zurück an den Client (Browser) gesendet. Dieser Ablauf wird in Abbildung 3.12 veranschaulicht. [29]

3.4.3 PHP

PHP wurde im Jahre 1995 von Rasmus Lerdorf entwickelt. Lerdorf benötigte damals ein Script, welches die Besucherzahlen seiner Homepage protokollierte. Zu diesem Zeitpunkt war das WWW noch sehr jung und Tools wie diese gab es eigentlich nicht. Daraufhin entwickelte Lerdorf eine Sammlung von Perl-Skripten namens „Personal Home Page“ (PHP). Heute bedeutet PHP jedoch *Hypertext Preprocessor* und ist eine serverseitige Skriptsprache zum Erstellen von dynamischen Webseiten. [30]

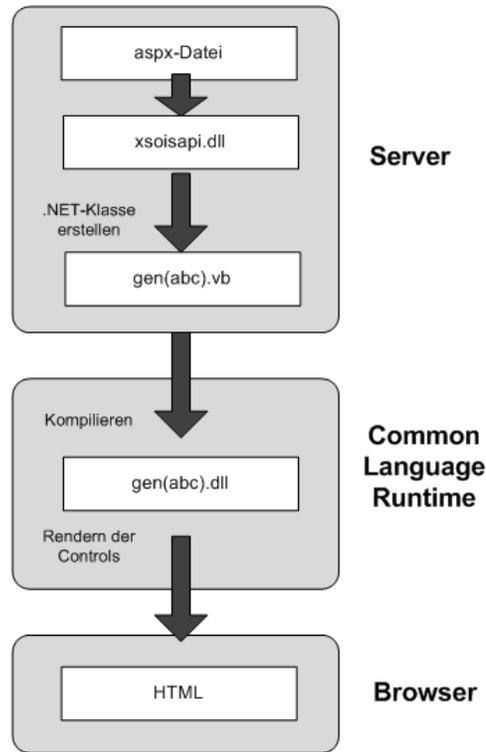


Abbildung 3.12: ASP.NET Architektur [29]

3.4.3.1 Architektur

Die Funktionsweise von PHP Programmen wird in Abbildung 3.13 illustriert.

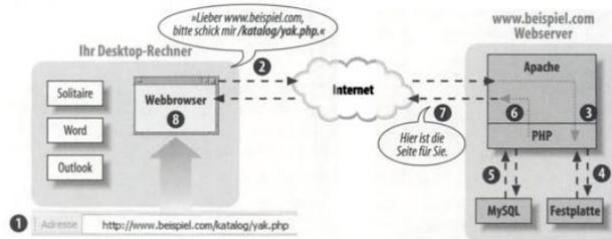


Abbildung 3.13: Client Server Kommunikation [26]

Nachdem der Benutzer eine Webseite aufruft, wird ein Request an den Webserver übermittelt (1,2). Der Webserver lädt das entsprechende PHP Programm von der Festplatte (3,4). Die Anweisungen im Programm werden vom PHP Interpre-

ter ausgeführt und eventuell Daten von einer Datenbank geladen. (5). Anschließend generiert der PHP Interpreter eine Antwort, welche wieder zurück an den Client übermittelt wird (7). Die Antwort ist im Normalfall HTML und JavaScript Code, der im Browser dargestellt wird (8). [31]

3.5 Webserver

Ein Webserver ist ein Serverprogramm, welches das HTTP-Protokoll anwendet. Das heißt, dass der Server einen TCP-Port belegt und alle ein- bzw. ausgehenden Anfragen nach den Regeln des HTTP-Protokolls beantwortet. [1]

Die beiden wichtigsten Webserver sind *Apache HTTP Server* und *Microsoft Internet Information Services*, die in den beiden folgenden Abschnitten näher erläutert werden.

3.5.1 Apache HTTP Server

Der Apache ist mit einem Marktanteil von 57 Prozent der meistbenutzte HTTP (Web-)Server im Internet (Stand Oktober 2010). [32] Entstanden ist der Apache im Jahre 1995. Der damals populärste Webserver NCSA-HTTPD wurde von einer kleinen Entwicklergruppe mit Erweiterungen versehen bzw. wurden durch sogenannte Patches Fehler behoben. Daraus entstand ein „gepatchter“ Server, woraus sich der Name Apache („a patchy server“) ergab. Am 1. Dezember 1995 wurde die Version 1.0 des Apache Servers veröffentlicht. Innerhalb von einem Jahr überholte der Apache den HTTPd Webserver und ist seit dem die am meisten eingesetzte Software für Webserver. [33]

3.5.1.1 Architektur

Der Apache HTTP ist ein stabiler Webserver mit einer großen Anzahl an Funktionen und Modulen, was auch der Grund für seine Popularität sein dürfte.

Apache läuft mit einem sogenannten „Preforking“-Modell. „Preforking“ ist eine Methode, bei der ein Serverprozess kurz vor dem Senden eines Requests gestartet wird. Diese Serverprozesse warten anschließend auf das Beantworten der Requests. Somit versucht der Webserver einige freie Prozesse bereitzustellen, um eintreffende Anfragen möglichst schnell beantworten zu können.

Apache hat einen modularen Aufbau, wodurch neue Funktionen relativ einfach eingebunden werden können (siehe Abbildung 3.14). Der Kern bildet den Mittelpunkt des Apache Servers und beinhaltet alle grundlegenden Funktionalitäten. Module können Apache-Funktionalitäten erweitern, verändern oder überschreiben. Dadurch ist der Apache Server nahezu unbegrenzt skalierbar, da alle Module mit einer Schnittstelle mit dem Kern interagieren. Der Informationsaustausch zwischen Modulen kann nur über den Kern erfolgen. [34]

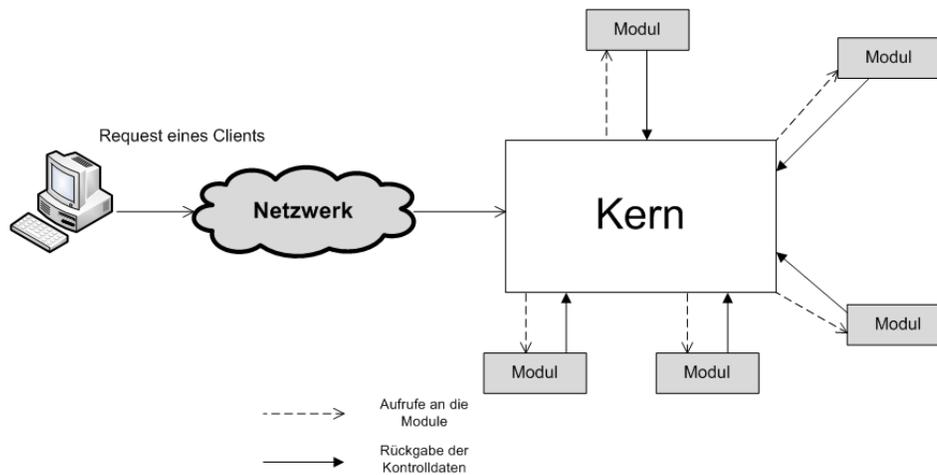


Abbildung 3.14: Apache Architektur [34]

3.5.2 Internet Information Services

Internet Information Services (IIS) wurde von Microsoft entwickelt und ist eine Plattform für mehrere Dienste. Diese umfassen einen Webserver, FTP-Server, NNTP-Server und andere Kommunikationsprotokolle wie beispielsweise SMTP oder POP3. [35]

Ab der Version 5 änderte Microsoft den Namen „Server“ auf „Services“ um auf die Vielfältigkeit des Webserver hinzuweisen. Internet Information Services können nur auf Microsoft Betriebssystemen installiert werden. Über IIS können ASP.NET Anwendungen ausgeführt werden, mit zusätzlichen ISAPI-Filtern werden auch PHP und JSP unterstützt. [36]

3.5.2.1 Architektur

Ab der Version 6 hat sich die Architektur von IIS grundlegend geändert und verbessert. In den Vorgängerversionen gab es immer wieder Probleme, weil alle Webseiten und deren dazugehörige Applikationen in einem einzigen Prozess liefen. Stürzte eine Applikation ab, waren auch alle anderen Anwendungen davon betroffen. Ab der Version 6 wird der sogenannte „Worker Process Isolation Mode“ unterstützt. IIS separiert die Webseiten und Applikationen in speziellen „Application Pools“. Jeder *Worker Process* arbeitet somit in seinem eigenen Speicherbereich. Im Falle eines Absturzes sind also keine weiteren *Worker Processes* betroffen. [34]

Wie die Abarbeitung einer Anfrage mittels *Worker Process* erfolgt wird in Abbildung 3.15 dargestellt.

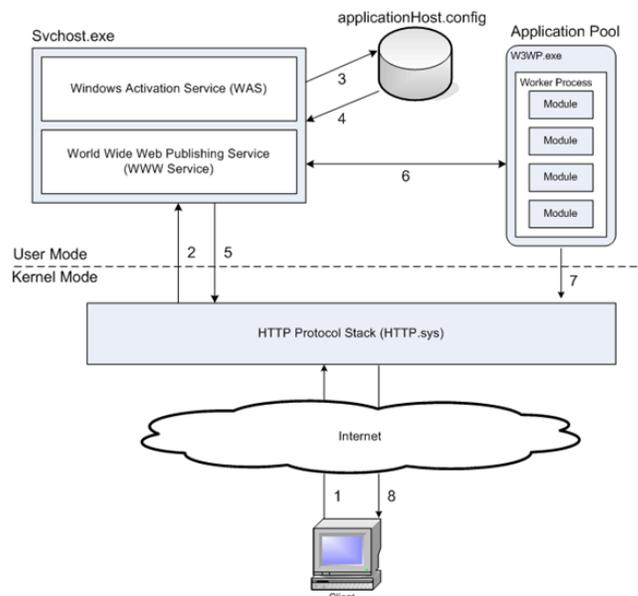


Abbildung 3.15: IIS Architektur [37]

Alle empfangenen Anfragen werden vom „HTTP Protocol Stack“ entgegen genommen (1). Anschließend wird der „Windows Activation Service“ (WAS) kontaktiert. Dieser lädt die Konfiguration für die angeforderte Anwendung vom „Application Host“ (3, 4). Anschließend übergibt der WWW Service diese Informationen dem „HTTP Protocol Stack“ (5). Im nächsten Schritt startet der WAS einen neuen „Worker Process“ im entsprechenden „Application Pool“ (6). Der „Worker Process“ arbeitet die Anfrage ab und übergibt die Antwort dem „Protocol Stack“ (7), welcher

die Antwort zurück an den Client übermittelt (8). [37]

3.6 Datenbanken

Eine Datenbank muss beliebige Daten verwalten können, Informationen aus diesen Daten liefern und nur berechtigten Personen den Zugriff gewähren. Unter dem Verwalten von Daten wird dabei das Einfügen von neuen Daten, das Löschen von alten Daten und das Aktualisieren von bestehenden Daten verstanden. [38]

Rene Steiner definiert den Begriff Datenbanken in seinem Buch „Grundkurs Relationale Datenbanken“ folgendermaßen:

„Eine Datenbank ist eine selbständige und auf Dauer ausgelegte Datenorganisation, welche einen Datenbestand sicher und flexibel verwalten kann.“

(Rene Steiner, [38])

Eine Datenbank hat folgende Aufgaben:

- Der Benutzer soll Zugriff auf gespeicherte Daten haben ohne, dass dieser wissen muss, wie die Daten im System organisiert sind.
- Eine Datenbank muss die Zugriffe von Benutzern kontrollieren. Das heißt, ein Benutzer der keine Zugriffsberechtigung hat, darf Daten weder sichten, noch manipulieren.
- Die interne Datenorganisation muss geändert werden können, ohne dass Applikationen, die auf Daten zugreifen, angepasst werden müssen.

3.6.1 Bestandteile einer Datenbank

Damit oben genannte Aufgaben erfüllt werden können, muss eine Datenbank bestimmte Werkzeuge und Komponenten bereitstellen. Die Kernkomponente ist dabei das Datenbankverwaltungssystem (DBMS). Diese beinhaltet alle notwendigen Systemroutinen für die gesamte Datenverwaltung wie Suchen, Lesen und Schreiben. Alle anderen Programme können nur über Schnittstellen des DBMS auf Daten zugreifen. Damit ein Benutzer mit dem DBMS kommunizieren kann, wird eine Datenbanksprache benötigt. Bei allen relationalen Datenbanken sind dies Abwandlungen

der Sprache SQL (Structured Query Language).

SQL umfasst folgende vier Aufgabenbereiche:

- In der *Datendefinition* (Data Definition Language) wird die Datenstruktur aufgebaut (Tabellen, Felder, usw.)
- Die *Datenmanipulation* ermöglicht es Daten einzufügen, zu löschen und zu verändern.
- Durch die *Datenabfrage* können Daten nach frei wählbaren Kriterien angezeigt/gefiltert werden.
- Der *Datenschutz* hat die Aufgabe, Daten vor unberechtigtem Zugriff zu schützen.

Eine weitere wichtige Komponente der Datenbank ist der Maskengenerator. Dieser erlaubt das Erstellen von Eingabemasken für den Benutzer. Durch diese Eingabeformulare kann der Benutzer Daten relativ einfach manipulieren. Zusätzlich können Daten vor dem Speichern überprüft werden oder komplizierte Berechnungen im Hintergrund durchgeführt werden. Der Reportgenerator ist für die Auswertung und Darstellung der gespeicherten Daten verantwortlich. Dieses Modul wird vor allem für die Programmierung von komplexen Abfragen und die gefilterte bzw. formatierte Darstellung von Daten benötigt. Sobald eine Datenbankanwendung mehrere Eingabeformulare besitzt oder eine Liste mit Datenbanktransaktionen erstellt wird, kommt ein Menügenerator zum Einsatz. Dem Benutzer steht dann ein Menü zur Verfügung, in dem er die entsprechenden Programme und Masken aktivieren kann. Zusätzlich zu den soeben beschriebenen Modulen, gibt es noch weitere Komponenten wie Netzwerkprogramme, Programmierspracheninterfaces, Abfragewerkzeuge, usw. Die wichtigsten Datenbankwerkzeuge sind in Abbildung 3.16 dargestellt.

3.6.2 Datenbankmodelle

Prinzipiell gibt es drei Arten von Datenbanken:

- Hierarchische Datenbanken
- Relationale und objektrelationale Datenbanken
- Objektorientierte Datenbanken

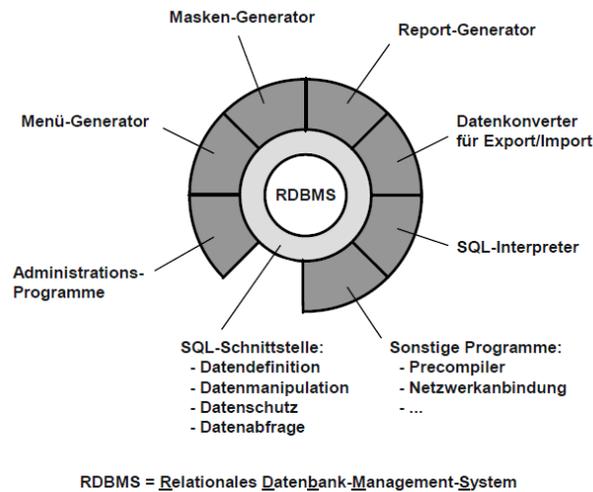


Abbildung 3.16: Schalenmodell eines Datenbanksystems [38]

Eine hierarchische Datenbank speichert die Daten in Form einer einzelnen sequentiellen Datei. Deshalb sind hierarchische Daten sehr unflexibel bezüglich Strukturänderungen. Relationale Datenbanken speichern die Daten geordnet nach Themenkreisen (Entitäten) in Tabellen. Eine objektrelationale Datenbank baut auf dem Modell der relationalen Datenbanken auf und wurde mit Techniken der objektorientierten Programmierung erweitert. Bei objektrelationalen Datenbanken können zusätzlich beliebige benutzerdefinierte Datentypen verwendet werden, während relationale Datenbanken nur eine begrenzte Anzahl von Standarddatentypen erlauben. Objektorientierte Datenbanken sind durch die objektorientierte Programmierung entstanden. Hier steht nicht die Tabelle, sondern das Objekt selbst im Zentrum der Betrachtungsweise. Das Objekt umfasst die Daten und die Methoden mit denen diese Daten manipuliert werden können.

3.6.3 MySQL

Es gibt mittlerweile eine Reihe an relationalen Datenbank-Management-System, jedoch ist MySQL für nichtkommerzielle Anwendungen derzeit die erste Wahl. Die Software steht kostenlos zur Verfügung und kann sowohl auf Linux als auch auf Windows installiert werden. MySQL entstand im Jahre 1994 als SQL-basierter Server für Webanwendungen. Zwei Jahre später (1996) wurde MySQL als Binärdistribution

für Linux und Solaris frei verfügbar gemacht. Mittlerweile ist auch eine Portierung auf Windows erfolgt.[39]

MySQL verwendet eine Client/Server-Architektur, dessen Serverprogramm (mysqld) von einem oder mehreren Client-Programmen genutzt werden kann (Abbildung 3.17).

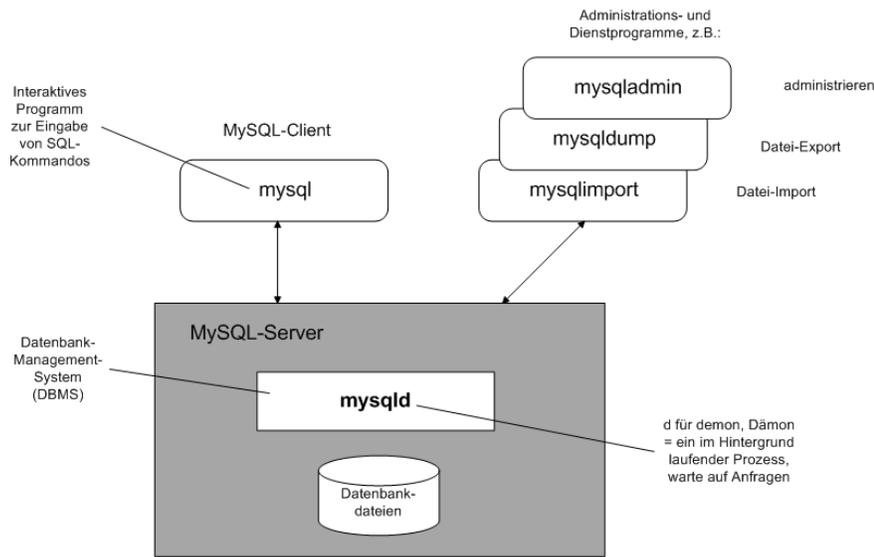


Abbildung 3.17: Komponenten von MySQL [39]

Das wichtigste Clientprogramm ist mysql. Es dient zur interaktiven Eingabe von SQL-Kommandos und kann zur Administration verwendet werden. Weitere Client-Programme für Administrationsaufgaben sind: mysqladmin, mysqldump und mysqlimport. [39]

3.6.4 Oracle

Oracle RDBMS ist, wie der Name schon sagt, ein relationales Datenbank-Management-System der Oracle Corporation. Larry Ellison startete im Jahre 1977 mit der Entwicklung der Oracle Datenbank. 1979 wurde die erste Version am Markt veröffentlicht. Mittlerweile steht Oracle bei der Version 11g (Oktober 2010) und kann auf allen gängigen Plattformen wie Windows und Linux installiert werden. [40]

3.6.5 MS SQL Server

Der SQL Server ist ein von Microsoft entwickeltes relationales Datenbank-Management-System und wurde im Jahre 1989 erstmals veröffentlicht. [41] Es gibt mehrere Editionen des SQL-Servers, die sich vor allem in ihren Funktionen und Hardwareeinschränkungen unterscheiden. Der SQL Server wird hauptsächlich als Data Warehouse eingesetzt. Außerdem bietet der SQL Server eine Vielfalt an verschiedenen Services und Tools, wie Analysis Services, Reporting Services, Integration Services und Sync Services. [42]

3.7 Drei Schichten Architektur

Bei einer Schichtenarchitektur erfolgt eine Zuordnung einzelner Aspekte einer Software zu einer Schicht. Diese Schichten sind Benutzeroberfläche (User Interface), Fachkonzept (Geschäftslogik) und Datenhaltung. Von einer oberen Schicht kann immer nur auf die jeweils direkt darunter liegenden Schicht zugegriffen werden. Innerhalb einer Schicht können Module beliebig untereinander kommunizieren. Diese strikte Ordnung sollte stets eingehalten werden um eine Verteilung auf Clients und Server zu ermöglichen. [43]

Ein grundlegendes Konzept der Softwareentwicklung besteht darin, User Interface und die Geschäftslogik voneinander zu trennen. Somit kann die Benutzeroberfläche jederzeit ausgetauscht werden, ohne dass die Geschäftslogik angepasst werden muss. Durch die Trennung zwischen Daten und Geschäftslogik ergibt sich eine weitere Schicht: die Datenhaltung. Dadurch erhält man ein Drei-Schichten-Modell wie in Abbildung 3.18 dargestellt. [43]

Die Komponenten innerhalb der Schichten können beliebig aufeinander zugreifen, jedoch gelten zwischen den Schichten strengere Zugriffsregeln [43]:

- **Schichten mit linearer Ordnung**

Bei einer Schicht mit linearer Ordnung darf immer nur auf die nächstniedrigere Schicht zugegriffen werden. Der Pfeil **c** in der Abbildung 3.18 ist somit nicht erlaubt. Ein Beispiel für ein Schichtenmodell mit linearer Ordnung wäre die TCP/IP-4 Schichtenarchitektur.

- **Schichten mit strikter Ordnung**

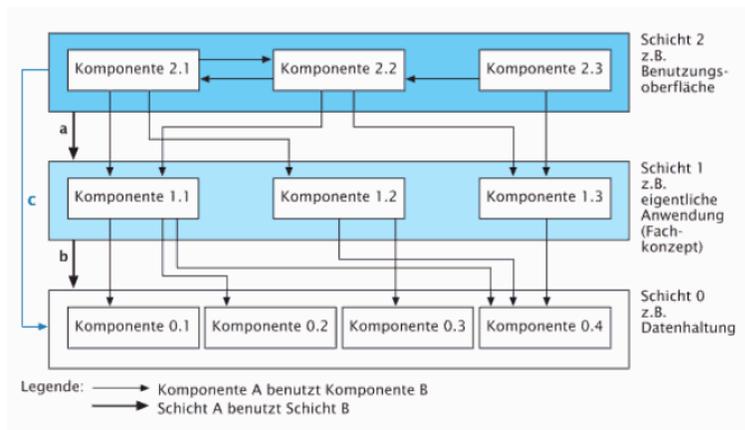


Abbildung 3.18: Beispiel einer Drei-Schichten-Architektur [43]

Die Schichten werden ihrem Abstraktionsniveau entsprechend angeordnet. Beim Modell der strikten Ordnung darf eine höher liegende Schicht auf alle tiefer liegenden Schichten zugreifen, aber nicht umgekehrt. Dies wird durch die Pfeile **a**, **b** und **c** in Abbildung 3.18 veranschaulicht.

Eine Schichtenarchitektur sollte angewendet werden, wenn

- sich die Dienstleistungen einer Schicht auf demselben Abstraktionslevel befinden und
- die Schichten entsprechend ihrem Abstraktionslevel geordnet sind, sodass höher liegende Schichten nur auf tiefer liegenden Schichten zugreifen müssen.

Durch diese Architektur ist es möglich, die einzelnen Schichten je nach Bedarf auf Client und Server zu verteilen. Bei der Client/Server Aufteilung muss manchmal die Geschäftslogik ebenfalls aufgeteilt werden (siehe Abbildung 3.19). [43]

3.8 Open Source vs. Closed Source

Die Diskussion über die Vor- und Nachteile von Open Source Software wird schon jahrelang geführt. Es gibt jedoch ein paar Punkte worin sich Open und Closed Source klar voneinander abgrenzen lassen.

Da Open Source Projekte für jedermann zugänglich sind, kann auch der Quellcode von vielen Personen kontrolliert werden. Deshalb werden Open Source Lösungen oft

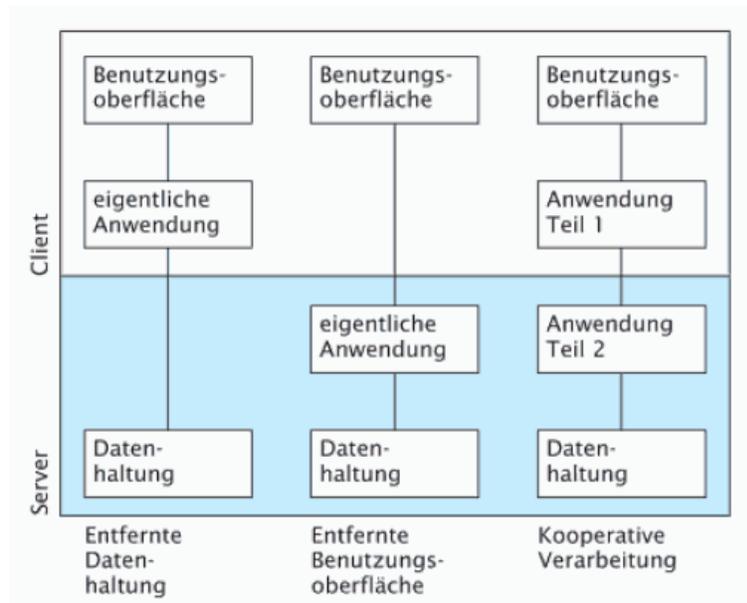


Abbildung 3.19: Verteilungsalternativen beim Client/Server Konzept [43]

schon alleine deshalb als fehlerfreier und sicherer eingestuft. Jedoch nur, weil die Möglichkeit besteht, dass viele Menschen einen Quellcode einsehen könnten, heißt dies noch lange nicht, dass es auch wirklich passiert.

Wenn ein Fehler von einem Reviewer gefunden wird, ist es fraglich, ob dieser den Fehler meldet oder für potentielle Angriffe ausnutzen möchte. Sicherheitslücken werden natürlich auch im Closed Source Umfeld nicht zwingend an den Hersteller gemeldet, sondern werden oft auf dem Schwarzmarkt gehandelt.

Open Source kann von jedermann weiterentwickelt und für seine Zwecke angepasst werden. Jedoch finanzieren mittlerweile viele große Unternehmen Open Source Projekte und geben auch klar die Richtung für die Software vor.

Ein weiterer Vorteil von Open Source Projekten wäre die Ersetzbarkeit von Entwicklern. Fällt ein Programmierer aus, kann die Software von anderen Personen weiter implementiert werden.

Bei kommerzieller Software ist der Quellcode nicht frei einsehbar. Aus diesem Grund ist es für potentielle Angreifer schwieriger Fehler zu finden, was die Software insgesamt sicherer macht.

Auch aus Gründen der Haftung werden oftmals kommerzielle Komponenten bevorzugt. Im Schadensfall kann der Hersteller haftbar gemacht werden. [44]

Kapitel 4

Balanced Scorecard

Unternehmen stehen heute mehr denn je im Wettstreit mit ihren Mitbewerbern. Die Gründe dafür sind enorm rasch wechselnden Umgebungsbedingungen und die sich ändernden Kundenanforderungen. Zusätzlich werden die Unternehmen von der Globalisierung des Warenverkehrs und des Finanzmarktes, sowie der weltweiten Vernetzung von Konzernen beeinflusst. Änderungen der Informationstechnologie und der Produktion, neue Werkstoffe und Vertriebswege und nicht zu Letzt Anforderungen der Lieferanten, Abnehmer, Mitarbeiter und des Umweltschutzes stellen die Unternehmen immer wieder vor neue Herausforderungen. Daraus resultiert das Hauptverbesserungspotential in Unternehmen:

Die Umsetzung der Strategien!

Das Modell der Balanced Scorecard setzt auf die eindeutig definierten Ziele, welche nicht alleinstehend sind, sondern mit anderen, auch nicht monetären Zielen verbunden sind. Bei bestehenden Umsetzungskonzepten und Leistungsmesssystemen entsteht daraus ein Konflikt, welcher immer zu Gunsten lediglich eines der vielen Ziele entschieden wird. Dabei werden Synergien innerhalb der Zieldefinitionen erkannt, jedoch aufgrund bestehender Hierarchiestufen und unklarer Zuständigkeiten meist nicht umgesetzt. Erst der Verbund dieser Synergieeffekte bringt den entscheidenden Wettbewerbsvorteil und sichert unter Einbindung aller Mitarbeiter den nachhaltigen Erfolg.

Hier setzt der Kerngedanke der Balanced Scorecard an. Diese ermöglicht es, die Erfolgsfaktoren und Leistungstreiber gleichgewichtet in verschiedenen Perspektiven darzustellen, zu messen und entsprechende Maßnahmen einzuleiten. Durch die Mess-

barkeit der Strategie und die Darstellung der Ziele mittels visualisierter Kennzahlen, können diese operativ gesteuert werden. [45]

4.1 Geschichte

Das Konzept der Balanced Scorecard, welches auch als „ausbalancierte Scorecard“ oder „ausgeglichene Anzeigetafel“ übersetzt wird, wurde von Kaplan/Norton entwickelt mit dem Ziel Unternehmensstrategien besser umzusetzen. Die Balanced Scorecard wurde ausgehend von einem bestehenden, jedoch nicht zukunftsorientierten Kennzahlen- und Managementsystem entworfen. Die erste Scorecard wurde 1987 bei *Analog Devices* unter Beteiligung von A.M. Schneidemann ausgearbeitet und von Ray Sata, dem CEO von *Analog Devices*, in den 5 Jahresplan (1987-1992) der nicht-finanziellen Scorecard aufgenommen (siehe Abbildung 4.1). [46]

Non-Financial Scorecard as Published in Sloan Management Review in 1988

Analog Devices: Quality Improvement Program Goals			
Measurement	1987	Half-Life (in months)	1992
External			
On-Time Delivery	85%	9	>99.8%
Outgoing Defect Level	500 ppm	9	<10 ppm
Lead Time	10 weeks	9	<3 weeks
Internal			
Manufacturing Cycle Time	15 weeks	9	4-5 weeks
Process Defect Level	5000 ppm	6	<10 ppm
Yield	20%	9	>50%
Time to Market	36 months	24	6 months

Abbildung 4.1: Erste nicht finanzielle Scorecard von Analog Devices Inc [46]

Durch die Veröffentlichung eines Artikels im *Harvard Business Review* im Jahre 1992 wird das Konzept der Balanced Scorecard weltweit bekannt. Robert S. Kaplan und David P. Norton arbeiten in den 90er Jahren an einem Forschungsprojekt mit 12 amerikanischen Unternehmen und entwickeln dabei das Verfahren zur Balanced Scorecard. [46]

4.2 Bestandteile der Balanced Scorecard

Eine Balanced Scorecard sollte folgende Elemente enthalten, wenn sie in einem Unternehmen eingeführt wird [47]:

- Vision
- Mission
- Strategien
- Perspektiven
- Kennzahlen/Indikatoren

4.2.1 Vision

Die Vision einer Organisation beinhaltet Wunschvorstellungen, die die obersten Ziele eines Unternehmens anstreben. Die Vision darf natürlich nicht völlig von der Realität abweichen, da es sonst nicht möglich ist, sie umzusetzen. Die Formulierung einer Vision erweist sich meistens als schwierig, jedoch gilt der Grundsatz: Je kürzer eine Vision in Worte gefasst ist, desto einprägsamer ist sie und desto leichter kann sie in Ziele und Strategien überführt werden. Die Vision darf nicht nur im Kopf des Managements vorhanden sein, sondern Mitarbeiter müssen von Anfang an in den Entwicklungs- und Formulierungsprozess mit einbezogen werden. [47] Es wird immer aus der Vision die Gesamtstrategie abgeleitet und ohne Visionen ist kein echter Fortschritt in Unternehmen möglich. [47], [48]

4.2.2 Mission

Im Gegensatz zur Vision, die sich nur auf das eigene Unternehmen konzentriert, versucht die Mission eine Wirkung nach außen zu erreichen. Die Mission gibt Auskunft darüber, wie das Unternehmen von der Umwelt und vor allem vom Kunden wahrgenommen werden soll. Die Mission muss sehr genau und treffend formuliert werden, da sonst das Gegenteil des Beabsichtigten eintreffen kann. [47]

4.2.3 Strategie

Die Strategie stellt den Ausgangspunkt bei der Erstellung der Balanced Scorecard dar. Aus diesem Grund ist die Wahl der Strategie von großer Bedeutung. [47] Eine Strategie ist ein Gesamtkonzept um ein oder mehrere Ziele erreichen zu können. Ziele werden immer auf längere Zeit ausgelegt und beinhalten aggregierte Größen. Unternehmensstrategien umfassen entweder Vorgaben für Handlungsweisen oder enthalten Planwerte, die in Handlungen umgesetzt werden sollen. Eine Gesamtstrategie setzt sich immer aus einer Menge von Einzelstrategien zusammen. [48]

4.2.4 Perspektiven der Balanced Scorecard

Kaplan und Norton schlagen in ihrem Grundkonzept zur Balanced Scorecard vier Perspektiven vor:

- Finanzwirtschaftliche Perspektive
- Kundenperspektive
- Interne Prozessperspektive
- Lern- und Entwicklungsperspektive

Diese Einteilung der BSC in den Perspektiven soll einen Rahmen für die Vermittlung von Vision und Strategie schaffen (siehe Abbildung 4.2). [45]

4.2.4.1 Finanzperspektive

Die Finanzperspektive ist ausschlaggebend für den Erfolg bzw. Misserfolg einer Strategie. Diese Perspektive enthält Ziele und Messgrößen, die den finanziellen Erfolg der Strategieumsetzung messen. Außerdem dokumentiert diese, ob ein wirtschaftlicher Erfolg realisiert werden konnte. [49]

Aus diesem Grund nimmt die Finanzperspektive eine Sonderstellung in der Balanced Scorecard ein. Zwar sollen alle Perspektiven gleichwertig behandelt werden, jedoch ist das Hauptziel einer Unternehmung immer monetärer Art. Die Kennzahlen der Finanzperspektive definieren die finanzielle Leistung, die von der Strategie erwartet wird. Zusätzlich fungieren diese aber auch als Endziele für die Kennzahlen

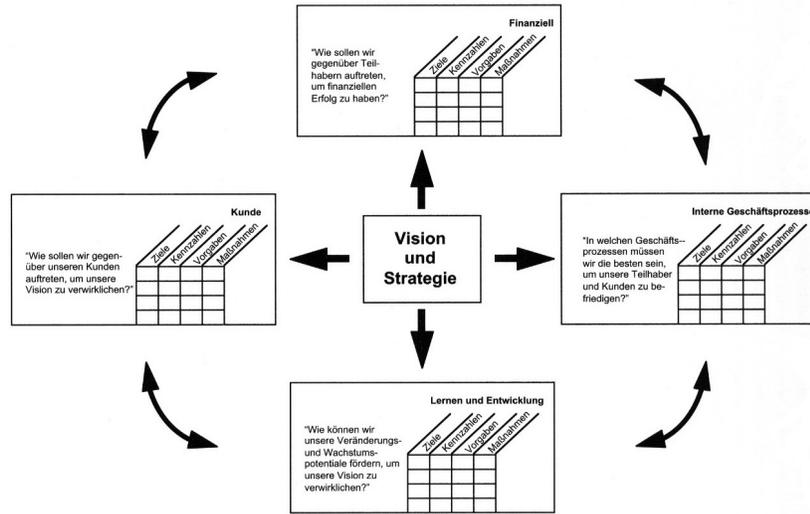


Abbildung 4.2: Perspektiven der Balanced Scorecard [45]

aller Perspektiven. [50] Es sollen Ziele festgelegt werden, die die Existenzberechtigung des Unternehmens für die Zukunft nachweisen und mit Alternativen vergleichen. Dies können Rentabilitäts- und Umsatzkennzahlen, sowie liquiditätsbezogene Größen sein. [51]

Wesentlich für die BSC sind die finanzwirtschaftlichen Ziele, die abhängig von der Lebenszyklusphase des Unternehmens unterschiedlich ausgeprägt sein können. Kaplan/Norton beschreiben hierfür drei Phasen mit unterschiedlichen strategischen Optionen: *Wachstum*, *Reife* und *Ernte*. In der Wachstumsphase steht das Umsatzwachstum im Mittelpunkt. Bei der Reife- und Erntephase hingegen wird der Schwerpunkt auf Deckungsbeiträge oder Kapitalrenditen gelegt. Zusätzlich zu den ertragswirtschaftlichen Zielen gibt es noch Kostensenkungsziele und Investitionsziele. [45]

Tabelle 4.1 soll mögliche Ziel- und Messgrößen, sowie Vorgaben und Maßnahmen der Finanzperspektive einer BSC veranschaulichen.

4.2.4.2 Kundenperspektive

Alle Ziele, die den Marktauftritt und die Marktpositionierung betreffen, gehören zu der Kundenperspektive. Dabei muss das Unternehmen klären, auf welche Kunden

„Wie kann ein finanzieller Erfolg erreicht werden?“			
Strategisches Ziel	Kennzahl	Vorgaben (operativ)	Maßnahmen
Finanzkraft stärken	Eigenkapitalsquote	Bis Ende 2009 auf 35% erhöhen	Kapitalserhöhung
Aktienkurs steigern	Aktienkurs	Aktienkurs verdoppeln	Aktienrückkauf-Programm

Tabelle 4.1: Beispiel für die Finanzperspektive einer BSC (Eigene Darstellung) [52]

man sich vorrangig konzentrieren möchte und welcher Nutzen geboten werden soll. [49]

In der Kundenperspektive werden alle Kunden- und Marktsegmente identifiziert, in denen der Betrieb konkurrenzfähig sein soll. Gleichzeitig sind diese Segmente für den Erfolg der finanzwirtschaftlichen Ziele ausschlaggebend. [45]

Dabei greift die Kausalkette der Kundenzufriedenheitsforschung: *Kundenorientierung* → *Kundennutzen* → *Kundenzufriedenheit* → *Kundenbindung* → *ökonomischer Erfolg*. [53] Alle daraus abgeleiteten Kennzahlen stellen die Kernkennzahlen vieler Unternehmen dar. Diese Kennzahlen umfassen Marktanteil, Kundentreue, Kundenakquisition, Kundenrentabilität und Kundenzufriedenheit. Außerdem gibt es in der Kundenperspektive Kennzahlen für den spezifischen Leistungstreiber der Kundenergebnisse (z. B. Qualität, Funktionalität, Preis, Services, usw.). Diese Kennzahlen geben Auskunft darüber, was ein Unternehmen seinen Kunden bieten muss, um einen hohen Grad an Zufriedenheit und in weiterer Folge einen hohen Marktanteil zu erreichen. [54]

Die Kundenperspektive ist besonders für Unternehmen mit intensiver Kundenorientierung die wichtigste Ebene in der Balanced Scorecard. Durch die zunehmende Übersättigung der Märkte zeigt sich ein starker Trend zur Kundenorientierung, wodurch die Kundenperspektive immer mehr in den Vordergrund rückt. Tabelle 4.2 zeigt eine exemplarische Darstellung der Kundenperspektive.

„Wie kann die Vision verwirklicht werden?“			
Strategisches Ziel	Kennzahl	Vorgaben (operativ)	Maßnahmen
Kundenbindung verbessern	Stammkundenanteil	Stammkundenanteil $\geq 60\%$	Einführung eines Treuerabatts
Innovator-Image aufbauen	Anteil neuer Produkte	Zwei neue Produkte p.a.	Forschungsbudget erhöhen

Tabelle 4.2: Beispiel der Kundenperspektive einer BSC (Eigene Darstellung) [52]

4.2.4.3 Interne Prozessperspektive

Die Prozessperspektive gibt die notwendigen Ausgaben und Leistungsergebnisse von Prozessen vor, um die Kunden- bzw. Finanzziele zu erreichen. Hier soll keine Auflistung aller Prozesse erfolgen. Es sollen nur jene Prozesse fokussiert werden, die eine wichtige Rolle bei der Strategieumsetzung spielen. [49] Es müssen also die Prozesse analysiert werden, die für das Erreichen von Kunden- und Anteilseignerzufriedenheit problematisch sind. [53]

Des Weiteren muss bei der Balanced Scorecard die komplette interne Wertschöpfungskette betrachtet werden. Diese Wertschöpfungskette umfasst dabei alle Prozesse, vom Innovationsprozess bis hin zu den Kundenprozessen (siehe Abbildung 4.3). Daher werden aus den Zielen und Kennzahlen der Kundenperspektive die strategischen Ziele für den gesamten Wertschöpfungsprozess abgeleitet. Der interne Prozess beginnt dabei stets mit dem Kundenwunsch. [55]



Abbildung 4.3: Das generische Wertkettenmodell [45]

Das Unternehmen erforscht im Innovationsprozess die Kundenwünsche, um diese dann in den Produkten oder Dienstleistungen entsprechend umzusetzen. Im Betriebsprozess werden die Produkte und Dienstleistungen produziert und an den Kunden geliefert. In der letzten Stufe werden Serviceleistungen nach dem Verkauf des Pro-

„Mit welchen Geschäftsprozessen kann die beste Kundenzufriedenheit erreicht werden?“			
Strategisches Ziel	Kennzahl	Vorgaben (operativ)	Maßnahmen
Kürzere Durchlaufzeiten	Durchlaufzeit	Senkung um 15%	Workflow-Systeme entwickeln
Technologischen Vorsprung gegenüber Konkurrenten	Anzahl der Prozessverbesserungen	Steigerung um 4%	MA-Verbesserungszirkel errichten

Tabelle 4.3: Beispiel der internen Prozessperspektive einer BSC (Eigene Darstellung) [52]

dukts durchgeführt. [54]

In Tabelle 4.3 sind Beispiele für Ziel- und Messgrößen, sowie Vorgaben und Maßnahmen der Prozessperspektive ersichtlich.

4.2.4.4 Lern- und Entwicklungsperspektive

Die vierte Perspektive der BSC identifiziert Kennzahlen, die den mittel- bis langfristigen Erfolg und das Wachstum der Organisation sichern sollen. Deshalb wird hier speziell die Entwicklung, Förderung und der Ausbau der Infrastruktur fokussiert um die Ziele der anderen Perspektiven erreichen zu können. [51]

Damit ein langfristiges Wachstum garantiert werden kann, muss ein Unternehmen in Personal, Systeme und Prozesse investieren.[45] Unternehmen befinden sich demzufolge stets in einem Wandlungsprozess, um sich den aktuellen Marktverhältnissen anpassen zu können. [49] Kaplan/Norton haben drei Hauptkategorien für die Lern- und Entwicklungsperspektive festgelegt:

- **Mitarbeiterpotenziale**

Mitarbeiter werden nicht als Produktionsfaktor betrachtet, sondern stellen vielmehr eine Ressource dar. Eine Sicherung der Unternehmensziele ist somit nur möglich, wenn ständig in das Potenzial von Mitarbeitern investiert wird. Daraus ergeben sich die Kennzahlen Mitarbeiterzufriedenheit, Personaltreue oder Mitarbeiterproduktivität. [45]

- **Potenzial von Informationssystemen**

Da Mitarbeiter ständig über Kunden, unternehmensinterne Prozesse und finanzielle Konsequenzen ihrer Entscheidungen informiert sein müssen, sind effektive Informationssysteme notwendig. Diese müssen die notwendigen Daten ausführlich und zielgerichtet zur Verfügung stellen. Mögliche Kennzahlen dieser Kategorie wären Zugriffsgeschwindigkeit oder Anzahl der elektronisch vernetzten Mitarbeiter. [45]

- **Motivation**

Nur wenn Mitarbeiter aktiv mitwirken, kann ein Nutzen aus Weiterbildungen und schnellen Informationssystemen erzeugt werden. Voraussetzung dafür ist ein positives Klima für Mitarbeitermotivation und Mitarbeiterinitiative innerhalb des Unternehmens. Beispielsweise wäre die Anzahl der eingereichten Verbesserungsvorschläge eine Kennzahl für diese Kategorie. [56]

4.2.5 Kennzahlen/Indikatoren

Ausgehend von der Vision, der festgesetzten Strategie und der Zuteilung zu der jeweiligen Perspektive werden Kennzahlen als erfolgsprüfende Indikatoren festgelegt. Kennzahlen (engl. „Key Performance Indicators“) sind eine komprimierte Form der objektiven Einzeldarstellung von Sachverhalten in und außerhalb eines Unternehmens. Damit wird eine einfache Geschäftskontrolle ermöglicht und ein Hilfsmittel für die Planung und Steuerung einer Organisation zur Verfügung gestellt. Die Verkettung der objektiven Einzelkennzahlen zu Kennzahlensystemen und Perspektiven erlaubt eine übergreifende Betrachtung und Beziehungserstellung. [45]

Kennzahlen besitzen einen absoluten bzw. relativen Informationsgehalt in Hinblick auf:

- Finanz- und Betriebswirtschaft
- Kundenstruktur und -wünsche
- Prozessfähigkeit, Technologie und Leistungsbereitschaft
- Mitarbeiter, Forschung und Entwicklung
- Ökologie

Bei den gewählten Kennzahlen sollen die Anteile der vergangenheitsbezogenen, ergebnisorientierten „lagging indicators“ und den zukunftsbezogenen, leistungstreibenden „leading indicators“ ausgewogen sein. Die ergebnisorientierten Kennzahlen messen die Reaktion auf Handlungen und erlauben somit die Überprüfung, ob sich das Geschäft verbessert oder nicht. Leistungstreiber hingegen geben den Weg vor, wie man zu den Ergebnissen kommt. [45]

4.3 Ursachen-Wirkungsketten

In einer BSC sollte die Strategie in einem Ursache-Wirkungsketten-Diagramm formuliert werden. In diesem Diagramm werden die geplanten Ziele mit ihren Ursachen in den verschiedenen Perspektiven miteinander verbunden. Somit entsteht eine sogenannte „Strategy Map“, die den Umformungsprozess von immateriellen Werten bis hin zu materiellen Werten wiedergibt (siehe Abbildung 4.4). [57]

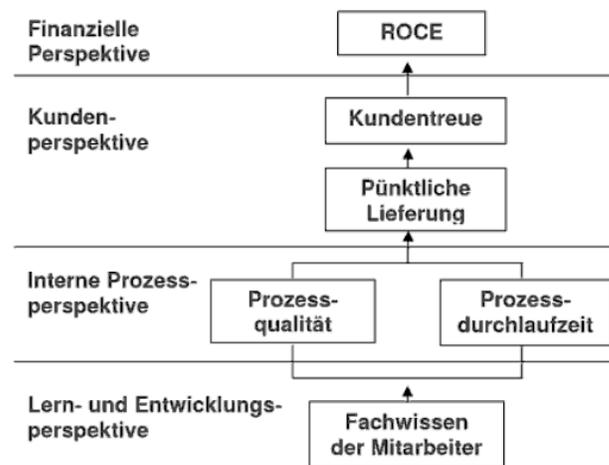


Abbildung 4.4: Ursachen-Wirkungskette in der BSC [45]

Diese Ursachen-Wirkungsbeziehungen sind das Hypothesensystem der Strategie und können innerhalb einer Perspektive, aber auch zwischen den einzelnen Ebenen einer Balanced Scorecard existieren. Ursachen-Wirkungsketten veranschaulichen die Zusammenhänge der strategischen Ziele und die Abhängigkeiten zur Zielerreichung. [58]

Die Ursachen-Wirkungsbeziehungen können in einer Reihe von Wenn-Dann-Aussagen

ausgedrückt werden, wie es am folgenden Beispiel zu erkennen ist:

*„**Wenn** die Produktkenntnis der Mitarbeiter gefördert wird, **dann** sind sie mit der gesamten Produktpalette, die sie verkaufen können, besser vertraut.*

***Wenn** die Mitarbeiter mehr über die Produkte wissen, **dann** wird sich ihre Verkaufseffektivität verbessern.*

***Wenn** ihre Verkaufseffektivität besser wird, **dann** werden sich die durchschnittlichen Gewinnspannen für die Produkte erhöhen.“*

(Kaplan, Norton 1997, [45])

4.4 Entwicklung einer Balanced Scorecard

Die Entwicklung einer Balanced Scorecard erfolgt in mehreren Teilprozessen:

- Strategie klären
- Strategische Ziele ableiten
- Ursache-Wirkungs-Beziehungen erkennen
- Kennzahlen festlegen
- Zielwerte definieren
- Strategische Aktionen

Diese Teilprozesse werden nun in den folgenden Abschnitten näher erläutert.

4.4.1 Strategie klären

Jede Balanced Scorecard baut auf eine abgeklärte und adaptierte Strategie auf. Die Strategie selbst wird über die Analyse und Bewertung der Ausgangslage von Umwelt und Unternehmen, sowie Visionen und Leitbild des Unternehmens entwickelt. Meistens wird eine Adaption notwendig, wenn die ersten Detailfragen an die Manager herangetragen werden. Jedoch bleibt der Fixpunkt die exakte Definition der Strategie: „Wo will das Unternehmen hin und wie umfangreich sollen die Veränderungen sein?“. Es werden strategische und langfristige Stoßrichtungen aus

den Visionen, Meinungen, Geschäftsplänen, Produkt-/Kundenkombinationen, Lebenszyklusphasen, Wertschöpfungstiefen und Kernkompetenzen festgelegt. [59] Das Modell der Balanced Scorecard dient zur Umsetzung von vorhandenen Strategien, prüft jedoch nicht die Korrektheit und ersetzt auch nicht die Aufgaben des Managements. [49]

4.4.2 Strategische Ziele ableiten

Die strategischen Stoßrichtungen werden in konkrete Ziele überführt. Diese Ziele sind der Kern der Balanced Scorecard und müssen innerhalb des Unternehmens abgestimmt werden. Prinzipiell werden vier bis sieben strategische Ziele pro Perspektive ermittelt und beeinflussen unmittelbar den Erfolg des Unternehmens. Wichtig ist es, eine Dokumentation über die strategischen Ziele zu erstellen. Diese sollte folgende Inhalte aufweisen [59]:

- Zielbenennung und Perspektivenbezug
- Zieldefinition und -erläuterung
- Zielbegründung
- Umsetzungstreiber, Prozesstreiber

4.4.3 Ursache-Wirkungs-Beziehungen erkennen

Die strategischen Ziele, sowie deren Kennzahlen und die strategischen Aktionen werden über die Ursachen-Wirkungs-Beziehungen miteinander verknüpft. Ausgehend von den Kennzahlen der Perspektiven müssen die strategischen Stoßrichtungen erkannt werden. Außerdem müssen auch bei Betrachtung der Strategie diese Stoßrichtungen in den Perspektiven wieder gefunden werden. Können diese Beziehungen nicht erkannt werden, deutet dies auf eine Dissonanz der strategischen Ziele hin. Deshalb gilt die Grundfrage „Was war die strategische Absicht für dieses strategische Ziel?“. [59]

4.4.4 Kennzahlen festlegen

Grundsätzlich reichen vier bis sieben Kennzahlen je Perspektive um ein strategisches Ziel klar ableiten zu können. Mit Hilfe der Kennzahlen können strategische

Ziele unmissverständlich ausgedrückt werden und das Mitarbeiterverhalten in die gewünschte Richtung gelenkt werden. Dabei können harte, finanzielle und weiche, soziale Kennzahlen festgelegt werden. Wichtig ist jedoch, dass eine überschaubare Anzahl von Kennzahlen (15-25) verwendet wird. [59]

4.4.5 Zielwerte definieren

Für jede Kennzahl müssen Zielwerte mit Toleranzen, Grenzen und zeitliche Einreichung festgelegt werden, damit die strategischen Ziele vollständig beschrieben werden können. Diese Werte werden meist ausgehend von der finanziellen Perspektive Top Down vorgegeben und mit den anderen drei Perspektiven abgestimmt. Als Vergleichsbasis wird der Zeitvergleich, Soll-Ist-Vergleich und der Betriebsvergleich herangezogen. [59]

4.4.6 Strategische Aktionen

Für die Umsetzung der strategischen Ziele werden nun Aktionen, Projekte und Initiativen benötigt. Zusätzlich kommt die Reservierung der Ressourcen von laufenden Projekten und deren Bewertung hinzu. Die einzelnen Aktionen werden zu Maßnahmen zusammengefasst und budgetiert. Des Weiteren werden Prioritäten durch die Abschätzung des Aufwandes vergeben. Die Realisierung und Einhaltung der Zielsetzungen soll durch das Verbinden der strategischen Aktionen mit dem Maßnahmen-, Durchführungs- und Prämissencontrolling garantiert werden. [59]

4.4.7 Ableitung nachfolgender Balanced Scorecards

Ausgehend von der Balanced Scorecard des Gesamtunternehmens werden alle nachfolgenden, tiefer gestuften BSCs abgeleitet. Dies erfolgt horizontal auf gleicher Ebene und vertikal auf unterschiedlicher Ebene (siehe Abbildung 4.5). Die Kennzahlen der einzelnen Perspektiven in einer tiefer gestuften Ebene müssen nicht den Kennzahlen einer höheren Ebene entsprechen. Es muss zwar eine sachlogische Verkettung vorhanden sein, eine mathematische Abhängigkeit auf Grund der unterschiedlichen Strategien der Ebenen wird als nicht zielführend erachtet.[49]

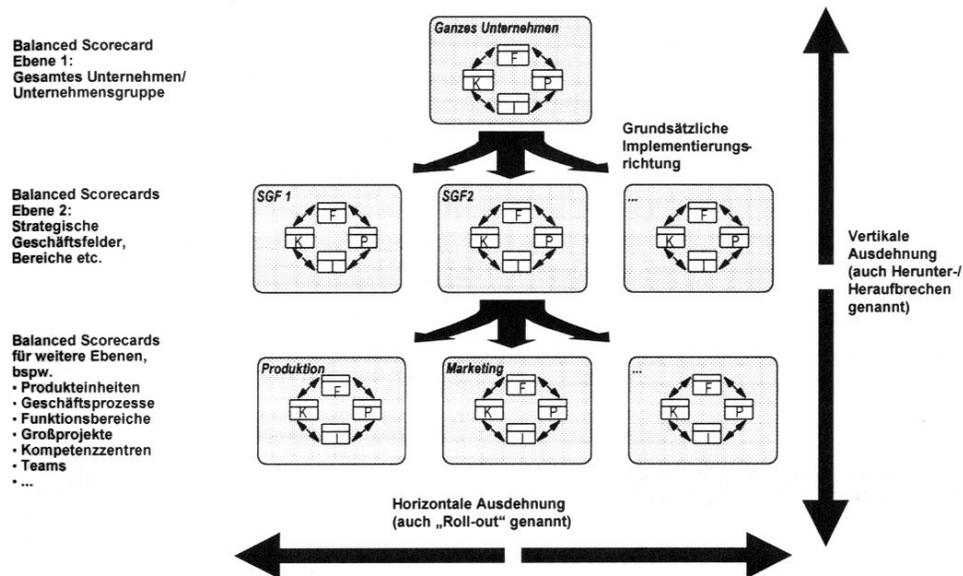


Abbildung 4.5: Ableitung der Balanced Scorecard [49]

4.5 Balanced Scorecard als Managementinstrument

Die Balanced Scorecard ist ein Instrument, das Prozesse der Strategieentwicklung, Strategieverwirklichung und Strategieüberwachung integriert. Es werden dabei typische Aufgaben des strategischen Controllings mit Aufgaben des strategischen Managements kombiniert. Die BSC bietet hierfür einen Handlungsrahmen, der in vier Stufen unterteilt ist (siehe Abbildung 4.6). [48]

4.5.1 Formulieren und Umsetzen von Vision und Strategie

Die Erarbeitung von Balanced Scorecards erfolgt im Zuge von Projekten. Dabei setzt sich das Projektteam großteils aus Führungskräften des Unternehmens zusammen. Jedoch könnte der Entwicklungsprozess durch externe Berater offener gestaltet werden, weil diese unabhängiger von den politischen Hintergründen des Unternehmens arbeiten können.

Die strategischen Ziele für die verschiedenen Geschäftseinheiten werden aus der Unternehmensstrategie abgeleitet. Höchste Priorität hat die Herstellung eines Konsenses über die Inhalte von Strategie und Unternehmenszielen zwischen den Führungskräften. Dadurch soll ein einheitlicher Verständnisrahmen geschaffen werden, der der BSC

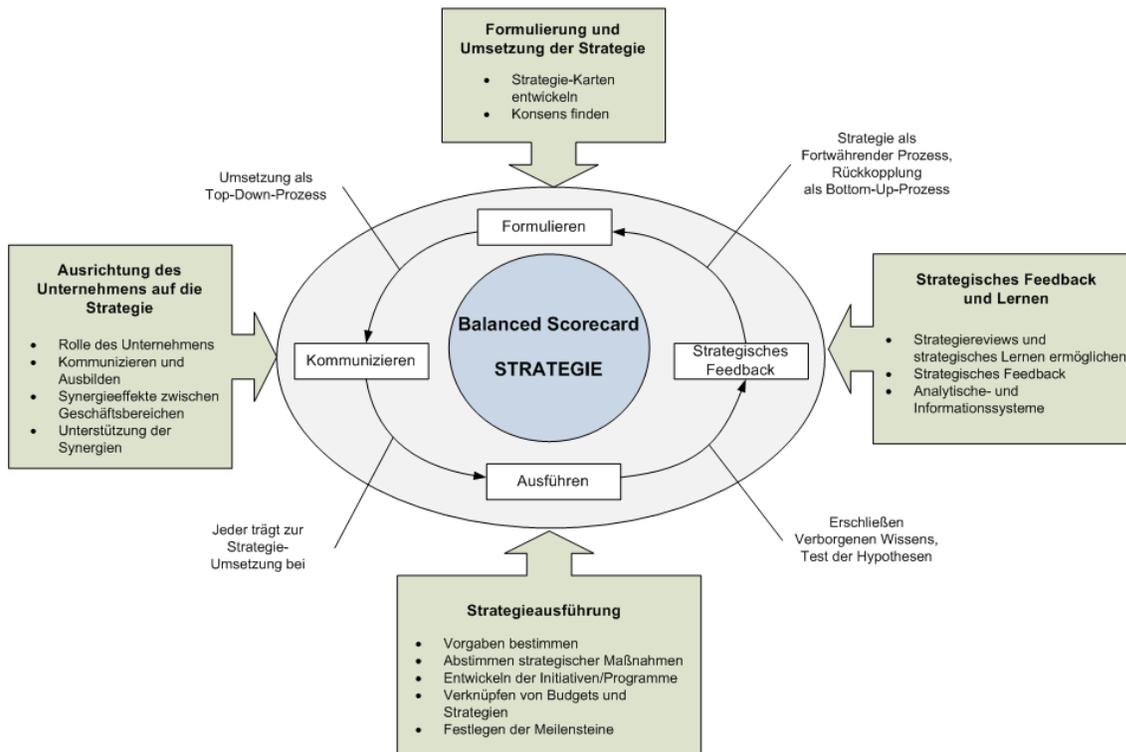


Abbildung 4.6: BSC als strategischer Handlungsrahmen [48]

zu Grunde liegt. [48]

4.5.2 Kommunizieren und organisatorische Ausrichtung

Sobald Strategie und Ziele klar formuliert sind, müssen diese in alle Unternehmensbereiche übertragen werden, um die gewünschte strategische Ausrichtung erreichen zu können. Dabei entstehen Maßnahmen, die durch die strategischen Geschäftseinheiten des Unternehmens ausgeführt werden müssen. Jede Geschäftseinheit hat ihre eigenen, individuellen Prioritäten und erstellt ihre eigene BSC. Das Ziel dieser Stufe ist es, die Strategien der Geschäftseinheiten mit der Unternehmensstrategie abzustimmen und Synergien zwischen den Geschäftseinheiten zu schaffen. Dies kann nur durch die Kommunikation der strategischen Ziele innerhalb der Organisation erreicht werden. Besonders wichtig ist dabei, dass die Kommunikation bidirektional erfolgt, um die aktive Auseinandersetzung aller Beteiligten mit den definierten Zielen zu fördern. [48]

4.5.3 Ausführen der Strategie

In dieser Stufe sollen die langfristigen Ziele bezüglich Kunden, interne Geschäftsprozesse, sowie Lern- und Wachstumsprogramme aufeinander abgestimmt werden. Die Zielgrößen der Kennzahlen einer Perspektive werden dabei für einen Zeitraum von drei bis fünf Jahren definiert und anschließend zur Umsetzung in kürzeren Perioden übergeben. Damit diese Zielgrößen erreicht werden können, ist die Entwicklung von Initiativen und Programmen notwendig. Anschließend werden für diese Programme und Initiativen die benötigten Ressourcen ermittelt. Des Weiteren muss das Budget bereitgestellt und die Dauer und die Verantwortlichen bestimmt werden. Zusätzlich werden Meilensteine gesetzt, die zur Überprüfung des Zielerreichungsgrades dienen. [48]

4.5.4 Strategisches Feedback und Lernen

Die letzte und wichtigste Stufe des Handlungsrahmens ist das strategische Feedback, da die BSC in einen strategischen Lernprozess eingebunden wird. Eine Strategie umfasst im Grunde immer ein Bündel von Hypothesen über Ursachen und Wirkungen, die kontinuierlich getestet werden. Hierfür wird ein strategisches Feedback-System benötigt, das laufend Rückmeldungen über die Ursache-Wirkungsprinzipien liefert, auf die sich die Strategie stützt. Das Feedback-System muss vier wesentliche Funktionen erfüllen [48]:

- Rückmeldungen über die verfolgte Strategie
- Überprüfung der Hypothesen, auf die sich die Strategie stützt
- Überwachung der Umsetzungsaktivitäten der Strategie
- Anpassung oder grundlegende Änderungen

Das Ergebnis des Lernens sollte immer eine Veränderung des Wissens sein. Es gibt drei unterschiedliche Möglichkeiten, um das Wissen in einer Organisation zu verändern (siehe Abbildung 4.7). [48]

Die ersten drei Stufen des strategischen Handlungsrahmens stellen einen Rückkopplungsprozess dar, der als Anpassungslernen („Single-Loop-Learning“) bezeichnet wird. Es handelt sich dabei um einen linearen Prozess, der sich ausgehend von der Strategieformulierung, ihrer Kommunikation und Verknüpfung mit der Organisation,

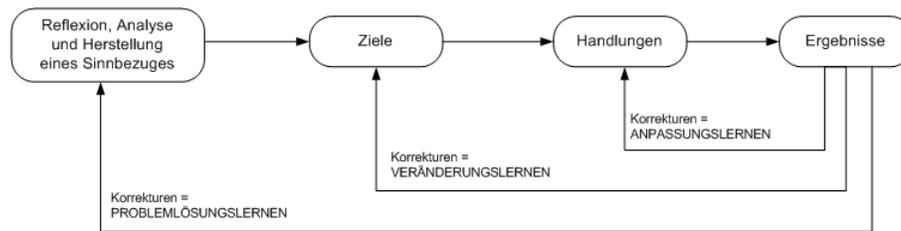


Abbildung 4.7: Arten des Lernens [48]

bis hin zur Abstimmung von Initiativen und Programmen mit langfristigen Zielen erstreckt. Die vierte Stufe ermöglicht dem Unternehmen nun ein Veränderungs- und Problemlösungslernen („Double-Loop-Learning“). Double-Loop-Effekte treten auf, sobald Manager Voraussetzungen in Frage stellen und sich überlegen, ob die bisherigen Handlungen mit den jetzigen Voraussetzungen und Erfahrungen noch übereinstimmen. [48]

Damit dies möglich ist, müssen Unternehmen die richtigen Informationen sammeln und diese zum Auswerten von bisherigen Vorgehensweisen verwenden. Des Weiteren soll mit diesen Informationen überprüft werden, inwieweit die Voraussetzungen der verfolgten Strategie noch gültig sind. Ein solches Frühaufklärungssystem muss ein permanenter Prozess sein, um die Vorgehensweisen überprüfen zu können und neue Chancen für Geschäftsfelder aufzudecken. [48]

Kapitel 5

Umsetzung des Management Tools

In diesem Abschnitt wird die Entwicklung der Softwarelösung beschrieben. Nach einer kurzen Evaluierung über die mögliche IT-Unterstützung der Balanced Scorecard, wird die Anforderungsanalyse und die eigentliche Implementierung der Individuelllösung beschrieben.

5.1 IT-Unterstützung für die Balanced Scorecard

Software-Anwendungen im Management sollen die Umsetzung, Kontrolle und Anpassung der Unternehmensstrategie unterstützen. In einer Studie über die Nutzung und Verbreitung von Software für Performance Management wurde festgestellt, dass die Hälfte aller Unternehmen immer noch Tabellenkalkulationsprogramme wie Excel benutzen. Nur zirka 30 Prozent setzen spezielle Software für die Umsetzung der Balanced Scorecard ein (siehe Abbildung 5.1).

Eine weitere Studie im deutschsprachigen Raum bestätigt dieses Ergebnis. Bei einer Untersuchung von 100 BSC-Applikationen lag der Anteil von Excel oder Access Anwendungen bei 71 Prozent (Abbildung 5.2).

Ein Großteil der Anforderungen von BSC-Anwendungen kann zwar von Tabellenkalkulationsprogrammen abgedeckt werden, jedoch weisen sie auch folgende Nachteile auf:

- **Skalierbarkeit**

Die Kapazitätsgrenzen von Tabellen sind schnell erreicht. Dadurch werden die Daten unübersichtlich und können nur schwierig bearbeitet werden.

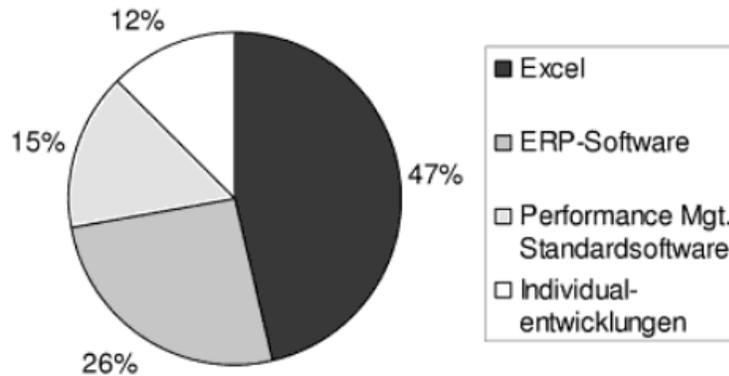


Abbildung 5.1: Kategorien von Softwarelösungen für die Balanced Scorecard

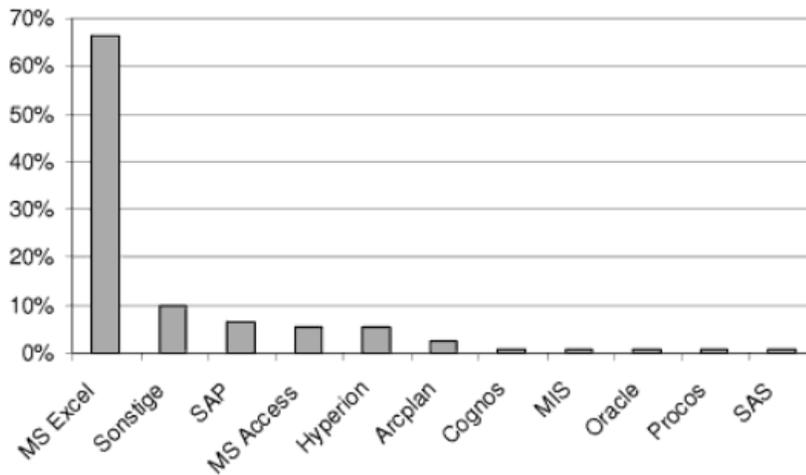


Abbildung 5.2: Kategorien von Softwarelösungen für die Balanced Scorecard im deutschsprachigen Raum

- **Dateneingabe**

Die manuelle Dateneingabe bei Tabellen ist sehr zeitaufwändig. Des Weiteren ergibt sich durch die manuelle Datenpflege eine erhöhte Fehlergefahr.

- **Kommunikation**

Die Daten werden lokal bearbeitet und können sich deshalb auf unterschiedlichen Desktops befinden.

- **Analyse**

Sobald mehrere Tabellen zusammengefügt werden müssen, kann die Analyse der Daten zu einem zeitaufwändigen und mühsamen Prozess werden.

Die Untersuchung im deutschsprachigen Raum ergab ebenfalls, dass nur zehn Prozent der Anwender mit den Excel oder Access-Applikationen „sehr zufrieden“ sind. Tabellenkalkulationsprogramme eignen sich nur für eine kleine Gruppe von Anwendern, die Zugriff auf ein gemeinsames Laufwerk haben und nur wenige Daten benötigen.

BSC-Anwendungen sollen den Unternehmen helfen die Strategie zu implementieren, indem sie verteilte Daten sammeln, Kommunikation zu den Kennzahlen ermöglichen und die gesammelten Daten des Unternehmens zu analysieren. Dadurch haben alle Mitarbeiter Zugang zu der Strategie der Organisation und werden durch Feedbacks der Anwendung bei ihren Aufgaben unterstützt.

5.2 Anforderungsanalyse

Als Basis für den Software-Auswahlprozess dient eine fachliche und technische Anforderungsanalyse. Alle notwendigen und gewünschten Funktionen der Anwendung sollen identifiziert werden und werden in einem Kriterienkatalog festgehalten.

5.2.1 Ziel- und Projektdefinition

Innerhalb des Unternehmens kommen Balanced Scorecards zum Einsatz, um die Durchführbarkeit der ausgelegten Unternehmensstrategie kontrollieren zu können bzw. der Führungsebene einen Überblick über die Leistungsfähigkeit der Organisation bieten zu können.

Bisher wurden alle Balanced Scorecards mittels Microsoft Excel erfasst bzw. berechnet. Da die Erfassung von mehreren Personen durchgeführt wird, ist die Verwendung von Exceltabellen natürlich fehleranfällig. Es können hier leicht Werte von anderen Kennzahlen überschrieben werden und Formeln für Berechnungen unabsichtlich gelöscht werden. Ein weiterer Nachteil der Exceltabellen liegt in der Darstellung von Ergebnissen. Zwar ist es möglich ein Ampelsystem mittels Hintergrundfarben darzustellen, jedoch können Trends (Pfeile) oder Fortschritte (Tachometer) nicht

grafisch visualisiert werden.

Zur besseren Erfassung und Kontrolle der Balanced Scorecards wird eine webbasierte Softwarelösung benötigt. Dadurch können alle Benutzer an zentraler Stelle kontrolliert auf die Daten zugreifen und diese bearbeiten. Die Webapplikation soll auf einem Linux System installiert werden und ihre Daten in einer MySQL Datenbank speichern.

Einstiegspunkt der Anwendung soll eine Übersicht der Ursache-Wirkungsbeziehungen der Unternehmensstrategie sein. Diese Ursache-Wirkungsbeziehungen sollen mit einer Strategiemappe (Strategy Map) visualisiert werden. Des Weiteren wird eine komplette Übersicht der Kennzahlen der Unternehmensziele benötigt. Eine Kennzahl (KPI) soll die Zielerreichung mit Zahlenwerten ausdrücken. Dazu werden jährlich Planwerte zu den einzelnen KPIs vorgegeben. Für jede Kennzahl ist eine Person im Unternehmen verantwortlich. Diese Person hat die Aufgabe monatlich die Istwerte des jeweiligen KPIs zu erfassen. Durch ein Ampelsystem in der Übersichtstabelle soll der Zielerreichungsgrad leicht ersichtlich sein. Zusätzlich soll der Trend einer Kennzahl durch Pfeile dargestellt werden. In einer Detailansicht sollen die Istwerte eines gesamten Jahres eingetragen werden. Ein Tachometer veranschaulicht den momentanen Status der Kennzahl im Vergleich zum Planwert. Ein Balkendiagramm soll den kompletten Verlauf der Kennzahl innerhalb eines Jahres visualisieren.

Weichen die Ist-Zahlen von den Planwerten für eine Kennzahl zu stark ab, sollen Maßnahmen über die Software eingeleitet werden. Für jede Maßnahme muss ein Verantwortlicher festgelegt werden. Dieser hat die Aufgabe genau zu dokumentieren, warum die Abweichung zustande gekommen ist bzw. welche Initiativen gesetzt werden müssen um das Ergebnis zu verbessern.

Die Anwendung soll drei verschiedene Usergruppen beinhalten. Es wird eine Administratorgruppe benötigt, die für die Stammdatenpflege verantwortlich ist. Für eine weitere Abstrahierung der Berechtigung sind zwei weitere Gruppen notwendig. Die Gruppe der Superuser ist für Abteilungsleiter des Unternehmens vorgesehen. Diese dürfen auf alle Kennzahlen und Maßnahmen der Organisation lesend zugreifen. Des Weiteren sollen sie in speziellen Übersichten ihre KPIs und Maßnahmen bearbeiten können. Die letzte Gruppe wird für die Standarduser benötigt. Diese ha-

ben einen sehr eingeschränkten Zugriff auf die Anwendung. Ein Standarduser darf nur seine Kennzahlen und Maßnahmen bearbeiten.

5.2.2 Use Cases

Die funktionalen Anforderungen werden in den folgenden Abschnitten mittels Use Case Beschreibungen genau erläutert.

5.2.2.1 Use Case: Perspektive erstellen/bearbeiten/löschen

Laut Definition von Balanced Scorecards gibt es prinzipiell vier Perspektiven. Diese sind:

- Finanzperspektive: Kennzahlen zum Erreichen der finanziellen Ziele
- Kundenperspektive: Kennzahlen zum Erreichen der Kundenziele
- Interne bzw. Prozessperspektive: Kennzahlen zum Erreichen der internen Prozess- und Produktionsziele
- Mitarbeiter-, Ressourcenperspektive: Kennzahlen zum Erreichen der (langfristigen) Überlebensziele der Organisation

In einer Tabelle sollen alle vorhandenen Perspektiven angezeigt werden. Der User kann einen Datensatz in der Tabelle auswählen und erhält alle Details in einer Bearbeitungsübersicht. Nach dem Auswählen einer Perspektive kann der User diese in einem Textfeld bearbeiten und anschließend speichern. Des Weiteren besteht die Möglichkeit neue Perspektiven anzulegen. Wird eine Perspektive gelöscht, werden alle von der Perspektive abhängigen Daten auch gelöscht. Dazu gehören Ziele, KPIs, sowie alle Plan- und Istwerte.

Daten einer Perspektive:

- Titel

Akteur: Administrator

Fehlerfälle:

- doppelte Einträge

5.2.2.2 Use Case: Firma erstellen/bearbeiten/löschen

Die Anwendung soll unternehmensübergreifend eingesetzt werden und muss die Strategiemappe und Balanced Scorecard von mehreren Unternehmen abbilden können. Daher können beliebig viele Firmen in der Anwendung angelegt werden. Alle bereits vorhandenen Firmen sollen wieder tabellarisch angezeigt werden. Nach dem Auswählen eines Datensatzes kann dieser in einer eigenen Maske bearbeitet werden. Eine Firma benötigt folgende Daten:

- Name
- Beschreibung

Wird eine Firma gelöscht, so werden alle abhängigen Daten wie Kennzahlen, Plan- und Istwerte ebenfalls gelöscht.

Akteur: Administrator

Fehlerfälle:

- doppelte Einträge

5.2.2.3 Use Case: Ziel erstellen/bearbeiten/löschen

Precondition: Perspektive muss existieren.

Ziele werden auf Grund der festgelegten Unternehmensstrategie definiert. Zu einem Ziel gehören somit bestimmte Kennzahlen (Key Performance Indicators). Ein Ziel ist also ein Sammelbegriff für mehrere KPIs. Das Erstellen/Bearbeiten und Löschen funktioniert analog zum Use Case *Perspektive erstellen/bearbeiten/lösen*. Ein Ziel wird zur Darstellung der Strategy Map benötigt und einer bestimmten Perspektive zugeordnet. Es bestehen Abhängigkeiten zwischen Zielen. So kann beispielsweise ein Ziel auf Prozessebene ein Ziel der Kundenperspektive beeinflussen.

Eine Ziel benötigt folgende Daten:

- Titel
- Beschreibung
- Zuordnung zu einer Perspektive
- Zuordnung zu einer Firma

Akteur: Administrator

Fehlerfälle:

- Doppelte Einträge
- Perspektive existiert nicht

5.2.2.4 Use Case: Kennzahl(KPI) erstellen/bearbeiten/löschen

Precondition: übergeordnetes Ziel und Perspektive muss existieren.

Ein Key Performance Indicator stellt eine Messgröße dar, die die Durchführbarkeit der Unternehmensstrategie veranschaulicht. Ein KPI braucht daher folgende Werte:

- Titel
- Beschreibung
- Verantwortlichen
- Einheit (absolut, relativ)
- Zuordnung zu einer Perspektive (finanzielle Sicht, Kundenperspektive, usw.)
- Zuordnung zu einem übergeordneten Ziel
- Zuordnung zu einem übergeordnete KPI
- Zuordnung zu einer Firma

Für das Eintragen der Ist-Werte muss ein Verantwortlicher des Key Performance Indicators definiert werden. Des Weiteren wird dem KPI eine Einheit zugeordnet. Der KPI kann entweder relativ (prozentual) oder absolut (Zahlenwert) erfasst werden. Ein KPI muss einer Perspektive zugeordnet werden (Finanzperspektive, Kundenperspektive, Prozessperspektive, Ressourcenperspektive). Zusätzlich wird ein KPI einem übergeordneten Ziel unterstellt. Dieses übergeordnete Ziel wird in der Strategiemappe definiert. Außerdem muss es möglich sein, dass ein Key Performance Indicator aus der Summe tiefer gestellter KPIs berechnet wird. Dadurch können Abhängigkeiten von Kennzahlen in einer Baumstruktur dargestellt werden. Die Tiefe eines Baumes muss beliebig wählbar sein. Alle Knoten und Blätter der Kennzahlen-Baumstruktur müssen dieselbe Einheit haben, da sonst eine automatische Berechnung der KPIs nicht möglich ist. Deshalb wird die Einheit eines KPIs automatisch

vom „Root“-Knoten übernommen. Des Weiteren sollen die Plan- und Istwerte nur für Blätter der Baumstruktur erfasst werden. Alle Werte der dazwischen liegenden Knoten bis zum Wurzelknoten werden aus der Summe der Blätter berechnet. Für die Ergebnisbildung muss die Einheit berücksichtigt werden:

- Relativ: Mittel aller SubKPIs
- Absolut: Summe aller SubKPIs

Abbildung 5.3 soll die Baumstruktur von Kennzahlen nochmals veranschaulichen.

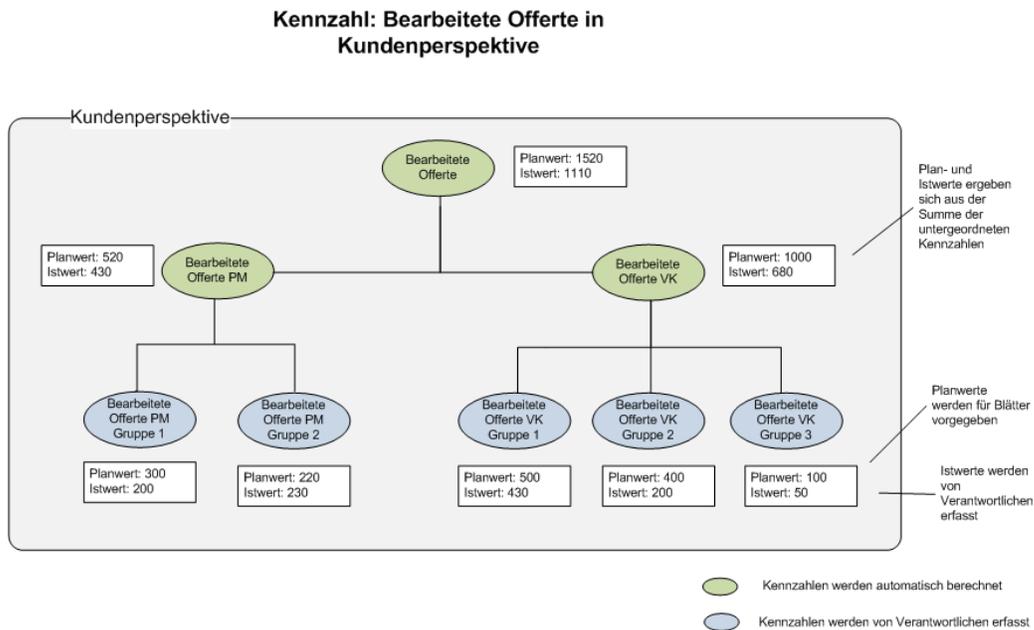


Abbildung 5.3: Beispiel der Baumstruktur einer Kennzahl

Akteur: Administrator

Fehlerfälle:

- übergeordnetes Ziel oder Perspektive existieren nicht
- übergeordnete Perspektive existiert nicht

5.2.2.5 Use Case: Kennzahl-Planung erstellen/bearbeiten/löschen

Precondition: Kennzahl existiert.

Am Beginn des Jahres muss für alle KPIs der Zielwert festgelegt werden. Dieser

Jahreswert wird automatisch auf alle durchzuführenden Erfassungen aufgeteilt. Hier muss ebenfalls die Einheit einer Kennzahl berücksichtigt werden:

- **Absolut:** Der Planwert muss auf die Anzahl der Erfassung gleichmäßig aufgeteilt werden.
- **Relativ:** Die Planwerte sind gleich hoch.

Für einen Planwert kann eine Abstufung (Toleranzen) festgelegt werden. Diese Toleranzen dienen zur Darstellung des Ampelsystems beim Soll-Ist-Vergleich der Kennzahlen.

Zu einem Planwert sollen nachstehende Werte gespeichert werden:

- Jahr für den Planwert
- Wert
- Schwellwert - Ampeldarstellung rot
- Schwellwert - Ampeldarstellung gelb
- Zuordnung zu einem übergeordnete KPI

Akteur: Administrator

Fehlerfälle:

- KPI existiert nicht

5.2.2.6 Use Case: Kennzahl Ist-Wert erfassen

Precondition: KPI-Planung existiert

Zu jeder Kennzahl müssen Ist-Werte zu den Planwerten erfasst werden. Die Erfassung der Ist-Werte erfolgt monatlich und wird von den festgelegten Verantwortlichen durchgeführt. Zu jedem Istwert kann ein Kommentar verfasst werden, um den derzeitigen Status der Kennzahl zu argumentieren. Zur Überprüfung, ob die Eingaben der Istwerte pünktlich durchgeführt wurden, soll zusätzlich ein Zeitstempel abgespeichert werden.

Ein Istwert benötigt daher folgende Felder:

- Wert

- Kommentar
- Zeitstempel
- Zuordnung zu einem Planwert

Akteur: User

Fehlerfälle:

- Keine KPI-Planung vorhanden

5.2.2.7 Use Case: Maßnahme erfassen

Precondition: KPI existiert.

Werden Planvorgaben für einen KPI nicht eingehalten, können Maßnahmen/Initiativen gesetzt werden. Eine Maßnahme dient zur Verbesserung bzw. Korrektur der Kennzahlen. Für einen Key Performance Indicator können beliebig viele Initiativen eingeleitet werden. Standardmäßig ist der Verantwortliche des KPI auch gleichzeitig der Verantwortliche für die Maßnahme, jedoch kann die Initiative auch anderen Personen zugeordnet werden. Maßnahmen sollen in einem bestimmten Zeitraum von dem Verantwortlichen abgearbeitet werden. Der Verantwortliche muss seine Tätigkeiten laufend dokumentieren und den Fortschritt (0 - 100%) angeben. Die Tätigkeiten werden in sogenannte „ToDo's“ beschrieben (siehe Use Case: *Maßnahmen-Status bearbeiten*) Eine Maßnahme/Initiative muss folgende Werte enthalten:

- Zuordnung zum KPI
- Titel
- Beschreibung
- Durchführungszeitraum (Start, Ende)
- Fortschritt
- Verantwortlicher

Akteur: Administrator

Fehlerfälle:

- KPI nicht vorhanden

5.2.2.8 Use Case: Userspezifische Daten anzeigen

Precondition: KPIs für den User existieren.

Dem Benutzer werden alle ihm zugeordneten Key Performance Indicators auf einer Übersichtsseite tabellarisch angezeigt. Hier werden für das aktuelle Jahr alle Plan- und Ist-Werte für die KPIs aufgelistet. Des Weiteren soll mittels Pfeil der aktuelle Trend eines KPIs in der Tabelle ersichtlich sein. Mit einem Link gelangt der User direkt in die Erfassung der Ist-Werte eines KPI. Zusätzlich muss auch eine Maßnahmenübersicht vorhanden sein. Die Maßnahmen sollen ebenfalls in einer Tabelle übersichtlich dargestellt werden. Initiativen, deren vorgegebener Zeitraum bereits abgelaufen ist, sollen markiert werden und für den Benutzer sofort ersichtlich sein. Für die Anwendung sind drei Benutzerrollen vorgesehen:

- **Administrator**

Der Administrator hat vollen Zugriff auf die Applikation. Er hat die Aufgabe die Stammdaten zu pflegen, sprich die Perspektiven, Ziele, Kennzahlen und Maßnahmen der BSC zu erfassen bzw. zu verwalten.

- **Superuser**

Ein Superuser kann in alle Kennzahlen und Maßnahmen einsehen, jedoch nur seine KPIs und Maßnahmen bearbeiten. Er hat keinen Zugriff auf die Stammdaten. Diese Rolle ist für Abteilungsleiter im Unternehmen vorgesehen.

- **User**

Der User hat nur Zugriff auf seine Kennzahlen und Maßnahmen. Seine Aufgabe ist es die Istwerte der Kennzahlen zu erfassen und seine Maßnahmen abzuarbeiten. Auf die gesamte Kennzahlen- und Maßnahmenübersicht soll der User keinen Zugriff haben. Des Weiteren dürfen von ihm auch keine Stammdaten gepflegt werden.

Akteur: User

5.2.2.9 Use Case: Maßnahmen/Status bearbeiten

Precondition: Maßnahme existiert.

Dem User werden die vordefinierten Details zur Maßnahme angezeigt (Titel, Beschreibung, Zeitraum, Verantwortlicher). Der Verantwortliche bearbeitet den Fortschritt der Maßnahme. Der Status wird prozentual (mittels Schieberegler oder Ähnlichem)

laufend aktualisiert. Des Weiteren können die geplanten ToDo's für die Maßnahme in einer Tabelle eingetragen werden. Ist ein ToDo's erledigt, wird dieser in der Tabelle der bisherigen ToDo's dargestellt. Zusätzlich besteht die Möglichkeit Kommentare des Verantwortlichen zur Maßnahme zu speichern.

Akteur: User

Fehlerfälle:

- Maßnahme nicht vorhanden

5.2.2.10 Use Case: Darstellung der Strategy Map

Die Strategy Map setzt sich aus den Ursache-Wirkungsbeziehungen der Ziele zusammen. Da diese Darstellung in den einzelnen Perspektiven sehr schwierig ist, können Bilder der Strategiemappe in die Applikation eingebunden werden. Das Bild der Strategiemappe darf nur von einem bestimmten Benutzerkreis (Administrator) verändert werden, muss aber in der Startansicht der Applikation für alle User ersichtlich sein.

Akteur: User

5.2.3 Kriterienkatalog

Ausgehend von der Projektdefinition und der Use Case Analyse wurde ein Kriterienkatalog erstellt. Dieser enthält die wichtigsten Vorgaben hinsichtlich technischer und funktionaler Kriterien für die Entwicklung der Anwendung. Der detaillierte Kriterienkatalog ist in folgenden Tabellen dargestellt.

Funktionale Kriterien	
<i>Integration</i>	
Anbindung operativer Systeme	<ul style="list-style-type: none"> • Integration in das Intranet
Scorecarding	

Unterstützung beim Aufbau	<ul style="list-style-type: none"> • Anlage von Unternehmen, Perspektiven, Zielen, Kennzahlen, Maßnahmen • Assistenten • Hilfe
Einbettung in Zielvereinbarung	<ul style="list-style-type: none"> • Übernahme von Zielvorgaben • Berechnung der Zielerreichung
Personalisierung	<ul style="list-style-type: none"> • Aufbau von persönlichen Scorecards • Zuweisung von Scorecardteilen (Kennzahlen/Maßnahmen) an Verantwortlichen • Personalisierte Sichten/Einstiegspunkte (Abhängig von Berechtigung)
Strategy Maps	<ul style="list-style-type: none"> • Darstellung der Strategy Map für jede BSC • einfacher Austausch der Strategy Map
Kennzahlenbeschreibung	<ul style="list-style-type: none"> • Verwaltung wichtiger Attribute: Beschreibung, Berechnung (Einheit), Planwerte, Istwerte, Kommentare • Trendberechnung • Unterstützung von hierarchischen Kennzahlen

Maßnahmen	<ul style="list-style-type: none"> • Planung von Maßnahmen • Verwaltung wichtiger Attribute: Beschreibung, Zeitraum, Verantwortlicher, Fortschritt • Dokumentation von Teilschritten der Maßnahmen (ToDo's)
Dokumentation/ Kommentierung	<ul style="list-style-type: none"> • Zeitliche Zuordnung • Abgabe von Kommentaren zu Kennzahlen und Maßnahmen • Abschließen für Scorecard Perioden
Benutzer- /Rechteverwaltung	<ul style="list-style-type: none"> • Anlegen von Benutzern • 3 Berechtigungsgruppen: Admin, Superuser, User • Benutzerzuordnung zu Unternehmen
<i>Präsentation und Analyse</i>	
Visualisierung/Oberfläche	<ul style="list-style-type: none"> • Ampelfunktion für farbliche Kennzeichnung der Kennzahlen • Tachometerfunktion für Kennzahlen • Balkendiagramm für Kennzahlen
Trends/Entwicklung	<ul style="list-style-type: none"> • Statusanzeige • Grafische Darstellungsform (Pfeile)

Navigation	<ul style="list-style-type: none">• einfache und intuitive Bedienung• Navigation zwischen Scorecard, Kennzahlen, Zielen, Planung, Maßnahmen• Filtermöglichkeiten
Berichte	<ul style="list-style-type: none">• Übersicht der Scorecards eines Unternehmens• Details zu Kennzahlen• Details zu Maßnahmen

Tabelle 5.2: Funktionale Kriterien der Applikation

Technische Kriterien	
Architektur	<ul style="list-style-type: none"> • Client/Server Architektur
Applikationstyp	<ul style="list-style-type: none"> • Webanwendung
Betriebssysteme	<ul style="list-style-type: none"> • Linux Server • Windows Clients
Webserver	<ul style="list-style-type: none"> • Apache Tomcat
Datenbankserver	<ul style="list-style-type: none"> • MySQL

Tabelle 5.1: Technische Kriterien der Applikation

5.3 Implementierung der Applikation

Da keine der am Markt vorhandenen Softwarelösungen die Funktionalität einer Kennzahlen-Hierarchie zur Verfügung stellt, wurde eine Individuallösung mittels *Google Web Toolkit* und *MySQL* erstellt. Als Entwicklungsumgebung diente *Eclipse* mit dem *Google Web Toolkit Plugin*. Für das Erstellen des Datenmodells wurde *MySQL Workbench CE 5.2* verwendet. Die Anwendung wurde in den Browsern *Mozilla Firefox*, *Google Chrome* und *Microsoft Internet Explorer* getestet.

5.3.1 Systemumfeld

Abbildung 5.4 zeigt die Systemarchitektur der Applikation. Der Benutzer interagiert über den Browser mit der GWT Webanwendung, die auf einem Apache Tomcat Webserver veröffentlicht wurde. Für die Persistenzschicht wurde eine MySQL-Datenbank gewählt, in welcher sämtliche Daten der Applikation gespeichert werden. Sowohl Webserver, als auch Datenbank-Management-System, wurden auf einer virtuellen Linux Ubuntu Server Maschine installiert. Die Anwendung verschickt Informations-Mails über einen Microsoft Exchange Mailserver. Alle benötigten Reports werden direkt in der GWT-Anwendung generiert und basieren auf reinem HTML.

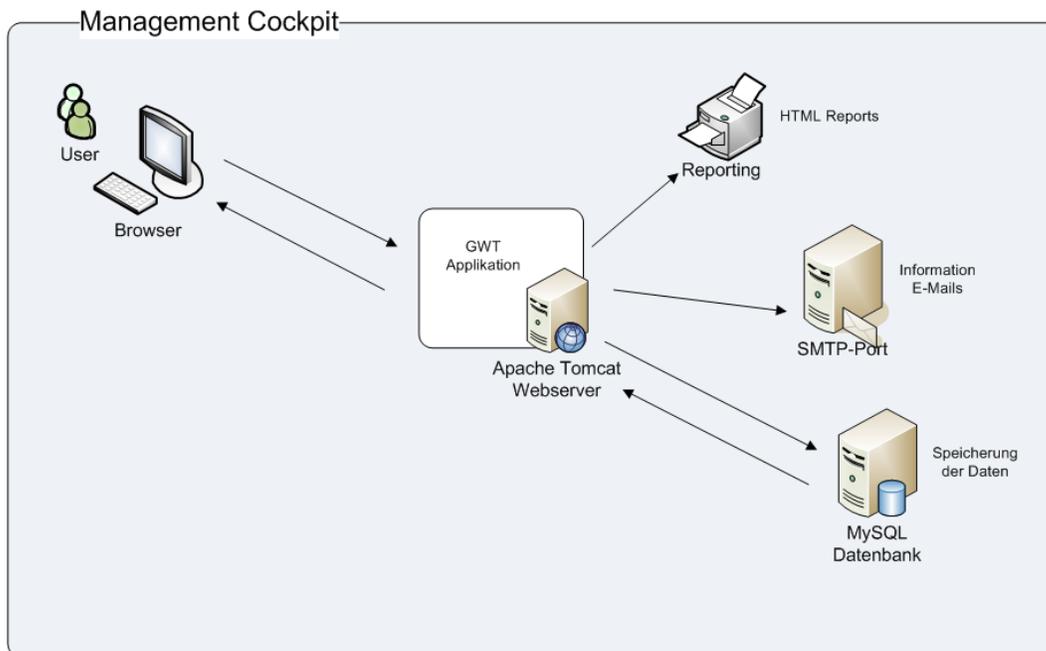


Abbildung 5.4: Systemumfeld der Applikation

5.3.2 Programmstruktur

Ein Balanced Scorecard gliedert sich in mehrere Schichten. Die höchste Ebene einer Balanced Scorecard ist eine Perspektive. Einer Perspektive können beliebig viele Ziele untergeordnet werden. Die Ziele sind geplante Vorgaben der Unternehmensstrategie. Zur Messbarkeit der Ziele werden Kennzahlen („Key Performance Indicators“) benötigt. Insofern können Ziele wiederum beliebig viele Kennzahlen enthalten. Des Weiteren soll die Möglichkeit bestehen, dass sich Kennzahlen aus anderen Kennzahlen zusammensetzen. Diese Baumstruktur soll beliebig tief wählbar sein. Jedoch dürfen nur zu Kennzahlen auf unterster Ebene Planwerte und Istwerte erfasst werden. Zu jedem Key Performance Indicator können Maßnahmen eingeleitet werden. Die Abarbeitung der Maßnahmen wird über Statuseinträge durchgeführt. Das Erfassen und Bearbeiten von Perspektiven, Zielen, Kennzahlen, Planwerten und Maßnahmen darf nur von einem User mit Administrator-Rechten durchgeführt werden. Ein Standardbenutzer darf nur Istwerte erfassen und Statuseinträge zu Maßnahmen eintragen. Abbildung 5.5 stellt die eben beschriebene Struktur der Anwendung dar.

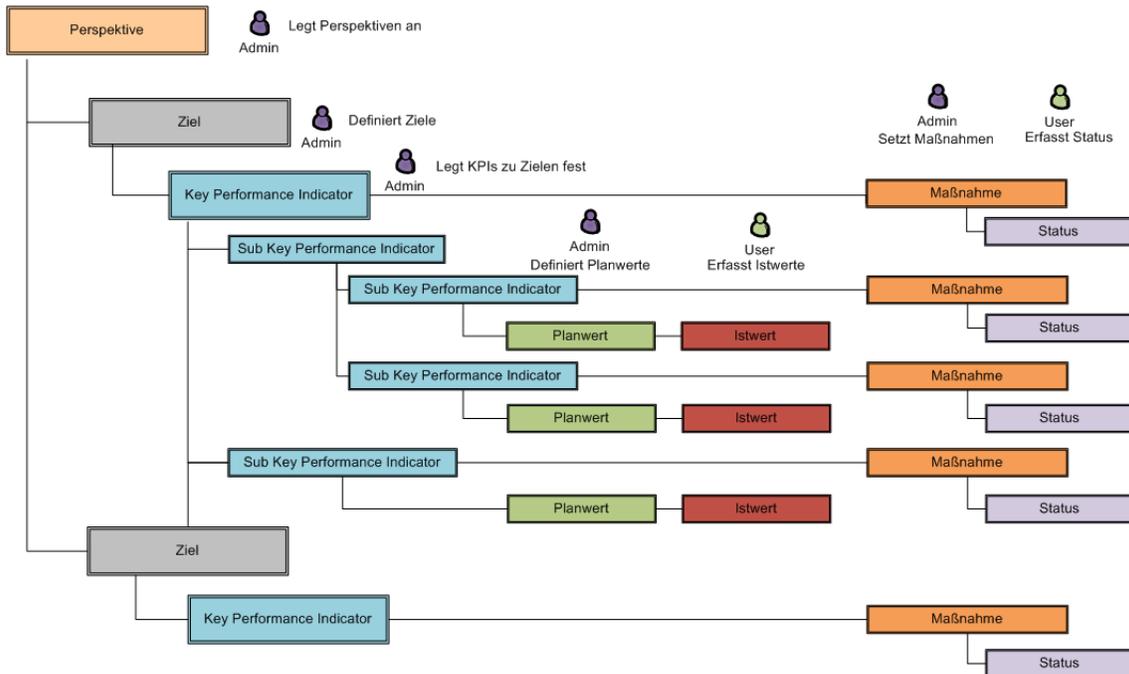


Abbildung 5.5: Struktur der Applikation

5.3.3 GWT „Best Practice“ Implementierung

Für die Implementierung der Webanwendung wurden die „GWT Application Architecture Best Practices“ von Ray Ryan verwendet. [60] Diese beinhalten unter anderem die Umsetzung des „Model-View-Presenter Pattern“, sowie die Verwendung eines zentralen „Event Bus“.

Abbildung 5.6 zeigt in einer Systemübersicht die Umsetzung der Komponenten mit den „Best Practices“.

Am Client kontrolliert der ApplicationController den EventBus und startet auch die Applikation. Die Eventhandler erstellen auf Grund von erhaltenen Events das Model, welches an Presenter/View weitergereicht wird. Der Presenter wirft seinerseits wieder Events auf den EventBus um zum Beispiel neue Daten anzufordern. Die gesamte Kommunikation zwischen Client und Server erfolgt über Remote Procedure Calls (RPC). Alle Daten, die vom Client über einen RPC angefordert werden, müssen über das RPC-Service am Server abgearbeitet werden. Dieses RPC Service lädt die Daten über die Datenbankschnittstelle (DBAccess) oder versendet E-Mails über das

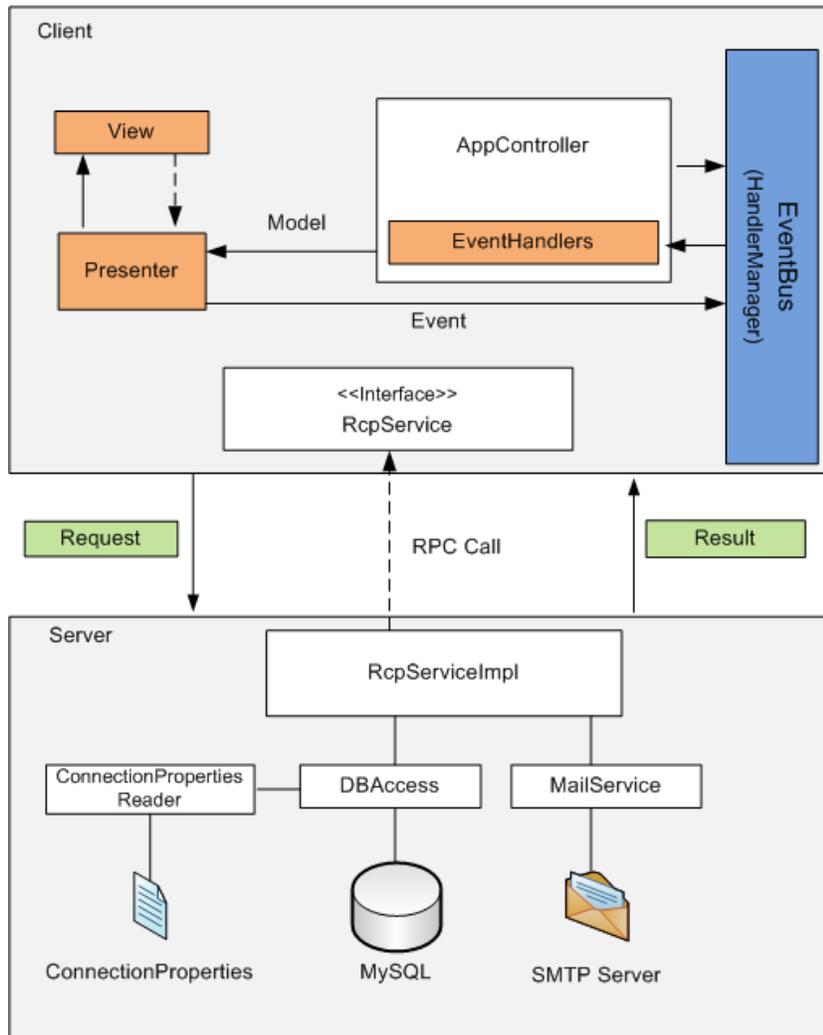


Abbildung 5.6: Systemübersicht

MailService. Die Daten werden als Result asynchron via RPC Callback zurück an den AppController gesendet, der dann entweder den Presenter damit versorgt, oder eine neue Presenter/View Variante erstellt.

5.3.3.1 Client-Server Kommunikation mit GWT RPC

Die Client-Server Kommunikation wird in GWT mittels GWT RPC (Remote Procedure Calls) durchgeführt. Dazu müssen drei Elemente beachtet werden. Das „Server-Sided Service“ (die Methode am Server, die aufgerufen wird), der Code am Client (der dieses Service aufruft) und die Datenobjekte, die zwischen Server und Client

hin- und hergereicht werden. Das folgende, aus der GWT Dokumentation entnommene Schaubild (Abbildung 5.7), stellt die benötigten Elemente für einen GWT RPC dar. [61]

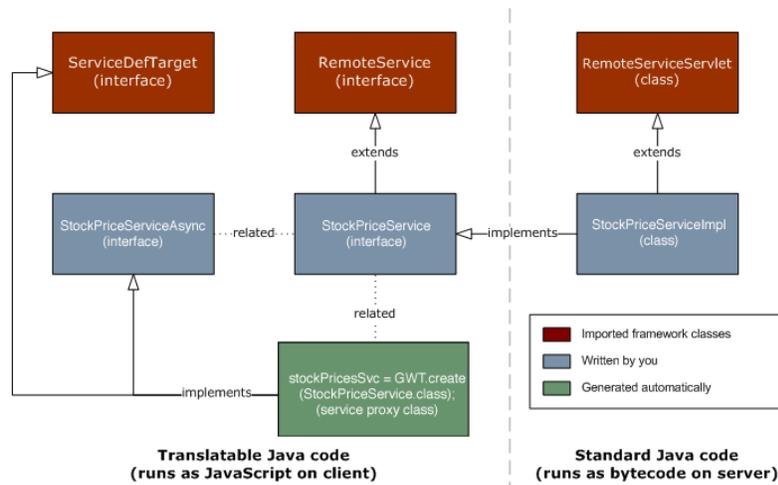


Abbildung 5.7: GWT Remote Procedure Call [61]

Für die Definition einer RPC Schnittstelle werden folgende Elemente benötigt [61]:

1. Ein Interface (**StockPriceService**) für das Service, das von **RemoteService** abgeleitet ist und alle RPC Methoden implementiert.
2. Eine serverseitige Klasse (**StockPriceServiceImpl**), welche von **RemoteServiceServlet** abgeleitet ist, das in 1. erwähnte Interface implementiert und die Serverimplementierung der RPC Kommunikation darstellt.
3. Schlussendlich muss noch ein asynchrones Interface (**StockPriceServiceAsync**) zum Service implementiert werden, welches vom Client aufgerufen werden kann.

5.3.3.2 Model-View-Presenter Pattern

Wie schon vorher erwähnt, sollte bei einer GWT Anwendung das Model-View-Presenter Pattern implementiert werden, um das Modell (Model) und die Ansicht (View) komplett voneinander zu trennen. Dabei steht neben einer deutlich verbesserten Testbarkeit auch die strenge Trennung der einzelnen Komponenten im Vor-

dergrund. [60]

Model

Das Modell enthält die Logik der Ansicht, kann aber auch die Geschäftslogik beinhalten. Sämtliche Funktionalität muss über das Modell erreichbar sein, um die Ansicht mit Daten versorgen zu können. Gesteuert wird das Modell über den Präsentator (Presenter). Das Modell kennt keine anderen Komponenten, sprich Ansicht oder Präsentator. [60]

View

In der Ansicht ist keine steuernde Logik enthalten. Sie ist allein für die Darstellung und die Ein- bzw. Ausgaben verantwortlich. Auch die Ansicht hat keinen Zugriff auf den Präsentator und das Modell. Die Steuerung der Ansicht erfolgt ebenfalls über den Präsentator. [60]

Presenter

Der Präsentator verbindet das Modell und die Ansicht. Alle logischen Abläufe zwischen den beiden anderen Komponenten werden vom Präsentator gesteuert.[60]

Des Weiteren kommen beim MVP-Pattern für Model und View Interfaces zum Einsatz. In den jeweiligen Interfaces wird der Aufbau von Model und View genau festgelegt. Über den Presenter und den Schnittstellen werden die beiden Schichten miteinander verbunden. Somit ist völlige Austauschbarkeit und Wiederverwendbarkeit der Schichten gewährleistet. [60]

5.3.3.3 Event Bus

Zusätzlich zum Model-View-Presenter Prinzip wird ein Eventbus verwendet. Dieser soll vor allem „Spaghetti-Code“ vermeiden und die asynchrone Kommunikation erleichtern. Der Eventbus wird dabei mittels GWT Handlermanager realisiert. Die Kommunikation zwischen Client und Server erfolgt über den Eventhandler. Damit dies möglich ist, müssen sich alle Eventhandler entsprechend am Eventbus, sprich GWT Handlermanger, registrieren. Sobald ein Event beispielsweise durch einen Buttonklick gefeuert wird, nimmt dies der entsprechende Eventhandler wahr und arbeitet das Event ab. Müssen Daten vom Client an den Server übertragen werden,

werden diese in einen Request verpackt. Die Antwort vom Server wird anschließend in einem Result-Paket wieder an den Client geschickt. [60]

5.3.4 Benutzersicht

Im Folgenden werden anhand einiger Masken die wichtigsten Funktionalitäten der Applikation vorgestellt.

Die Einstiegsseite ist zugleich die wichtigste Maske der Applikation. Alle für den Benutzer notwendigen Informationen werden innerhalb der sechs Registerkarten der Hauptseite dargestellt (siehe Abbildung 5.8) Diese Registerkarten beinhalten:

- **Strategiemappe:** Darstellung der Strategy Map eines Unternehmens
- **KPI Übersicht:** Darstellung aller Kennzahlen eines Unternehmens
- **Maßnahmen Übersicht:** Darstellung aller Maßnahmen eines Unternehmens
- **Meine KPIs:** Darstellung der KPIs, die dem eingeloggtem User zugeordnet sind
- **Meine Maßnahmen:** Darstellung der Maßnahmen, die dem eingeloggtem User zugeordnet sind
- **Info:** Zentrale Darstellung aller relevanten Zusatzinformationen für die Applikation und Balanced Scorecards

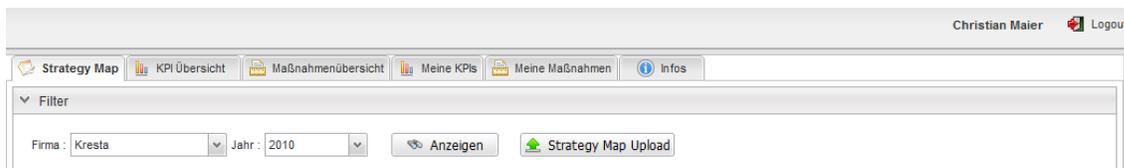


Abbildung 5.8: Registerkarten der Hauptseite

5.3.4.1 Strategy Map

Abbildung 5.9 veranschaulicht die Darstellung der Strategy Map. Da die Visualisierung der Strategy Map innerhalb einer Webapplikation sehr schwierig umzusetzen ist, werden die Strategiemappen nur als Bilddateien innerhalb der Anwendung hochgeladen und anschließend dargestellt.

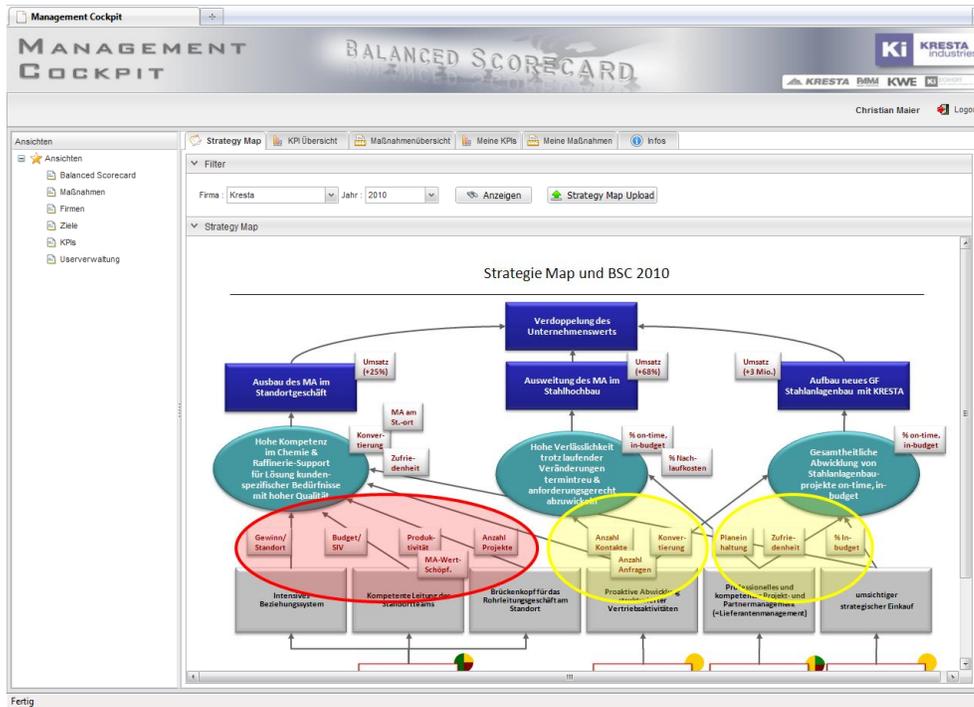


Abbildung 5.9: Visualisierung der Strategiemappe

5.3.4.2 Kennzahlen Übersicht

Besonders wichtig ist die Ampeldarstellung und die Berechnung der Trends der Kennzahlen (siehe Abbildung 5.10):

- **Leere, weiße Felder:** Es wurden noch keine Ist-Werte erfasst
- **Rote Felder:** Die Ist-Werte der Kennzahl liegen weit unter dem vorgegeben Planwert
- **Gelbe Felder:** Die Ist-Werte liegen unter dem Planwert
- **Grüne Felder:** Die Ist-Werte erreichen die vorgegebenen Planwerte

Der Trend einer Kennzahl wird ebenfalls mittels Ampelfarben dargestellt. Dabei wird der zuletzt erfasste Ist-Wert mit dem Jahresdurchschnitt der Ist-Werte verglichen.

Des Weiteren werden die Kennzahlen auf die einzelnen Perspektiven aufgeteilt. Innerhalb der *Kundenperspektive* ist eine Kennzahlen-Hierarchie am Beispiel „Registrierte Offerte“ ersichtlich. Die Soll- und Ist-Werte und somit die Ampeldarstellungen bzw. Trends der Knoten ergeben sich aus der Summe der Blätter des „Kennzahlen-Baums“.

The screenshot shows a software interface for KPI Overview. At the top, there are navigation tabs: Strategy Map, KPI Übersicht (selected), Maßnahmenübersicht, Meine KPIs, Meine Maßnahmen, and Infos. Below the tabs is a filter section with 'Firma: Kresta', 'Jahr: 2010', and 'Ansicht: Quartalsansicht'. There are also 'Anzeigen' and 'Drucken' buttons.

The main content is divided into two sections: 'Finanzen' and 'Kunden'. Each section contains a table of KPIs with columns for Soll (Target) and Ist (Actual) values for 2009 and 2010, and quarterly data for 2010 (Q1-Q4). Trend indicators (up/down arrows) are shown for each quarter.

Finanzen Table:

KPI	Verantwortlicher	Soll 2009	Ist 2009	Soll 2010	Ist 2010	1		2		3		4		Trend	Details
						Soll 1	Ist 1	Soll 2	Ist 2	Soll 3	Ist 3	Soll 4	Ist 4		
Umsatz (in TEUR)	Sattler Wolfgang	0	0	60.000	31.700	15.000	13.000	15.000	9.500	15.000	8.000	15.000	1.200	↓	
EGT (in TEUR)	Sattler Wolfgang	0	0	3.600	1.890	900	670	900	590	900	630	900		↑	
DB-Quote (in %)	test test	0 %	0 %	26 %	9,42 %	26 %	17,67 %	26 %	15,33 %	26 %	4,67 %	26 %		↓	
Umsatz bei Bestandskunden (in	Sattler Wolfgang	0	0	20.000	13.540	5.000	4.390	5.000	4.200	5.000	4.950	5.000		↑	

Kunden Table:

KPI	Verantwortlicher	Soll 2009	Ist 2009	Soll 2010	Ist 2010	1		2		3		4		Trend	Details
						Soll 1	Ist 1	Soll 2	Ist 2	Soll 3	Ist 3	Soll 4	Ist 4		
Zusatzauftragsquote (in %)	Sattler Wolfgang	0 %	0 %	25,12 %	0 %	25,12 %		25,12 %		25,12 %		25,12 %		↔	
Registriert Offerte	Sattler Wolfgang	0	0	850	467	212,5	146	212,5	157	212,5	95	212,5	69	↓	
Registrierte Offerte Verkauf	Lanz Thomas	0	0	600	316	150	116	150	115	150	44	150	41	↓	
Registrierte Offerte Tea	Maier Christian	0	0	400	193	100	68	100	85	100	15	100	25	↑	
Registrierte Offerte Tea	Maier Christian	0	0	200	123	50	48	50	30	50	29	50	16	↑	
Registrierte Offerte PM	Maier Christian	0	0	250	151	62,5	30	62,5	42	62,5	51	62,5	28	↓	
Registrierte Offerte Tea	Maier Christian	0	0	100	76	25	24	25	16	25	23	25	13	↑	
Registrierte Offerte Tea	Maier Christian	0	0	150	75	37,5	6	37,5	26	37,5	28	37,5	15	↓	

At the bottom, there are expandable sections for 'Prozesse' and 'Ressourcen'.

Abbildung 5.10: Kennzahlen-Übersicht

5.3.4.3 Kennzahlen Details

Abbildung 5.11 zeigt die Darstellung der Kennzahlendetails. Neben der genauen Beschreibung der Kennzahl wird der Verantwortliche in einem „mailto“ Link angezeigt. Der Vergleich des Ist-Wertes zum Plan-Wert der Kennzahl wird in einem Tachometer visualisiert. Zusätzlich werden die Grenzen der Schwellwerte für die Ampeldarstellung (rot, gelb, grün) im Tachometer angezeigt. In einem Diagramm wird der Verlauf der Ist-Werte für das gesamte Jahr abgebildet. Außerdem werden die Ist-Werte für das gesamte Jahr in einer Tabelle dargestellt.

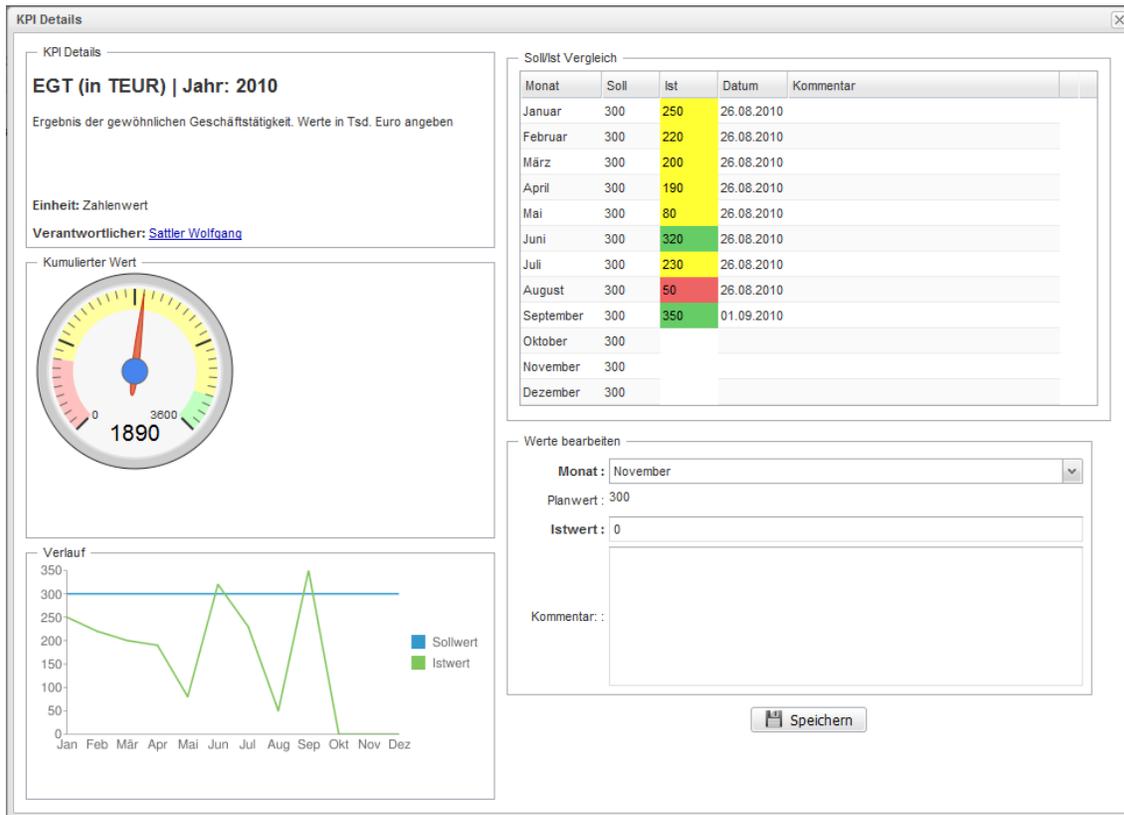


Abbildung 5.11: Kennzahlen-Details

5.3.4.4 Kennzahlen erstellen/bearbeiten

Sämtliche KPIs werden in einer Übersichtstabelle dargestellt. Für eine einfache Suche kann jede Spalte einzeln gefiltert werden. Der Verantwortliche einer Kennzahl kann über einen „mailto-Link“ sofort per EMail kontaktiert werden. In der Detailansicht erfolgt die Zuordnung der Kennzahlen zu einer Baumstruktur. In einer „Combobox“ werden alle möglichen übergeordneten KPIs aufgelistet. (siehe Abbildung 5.12)

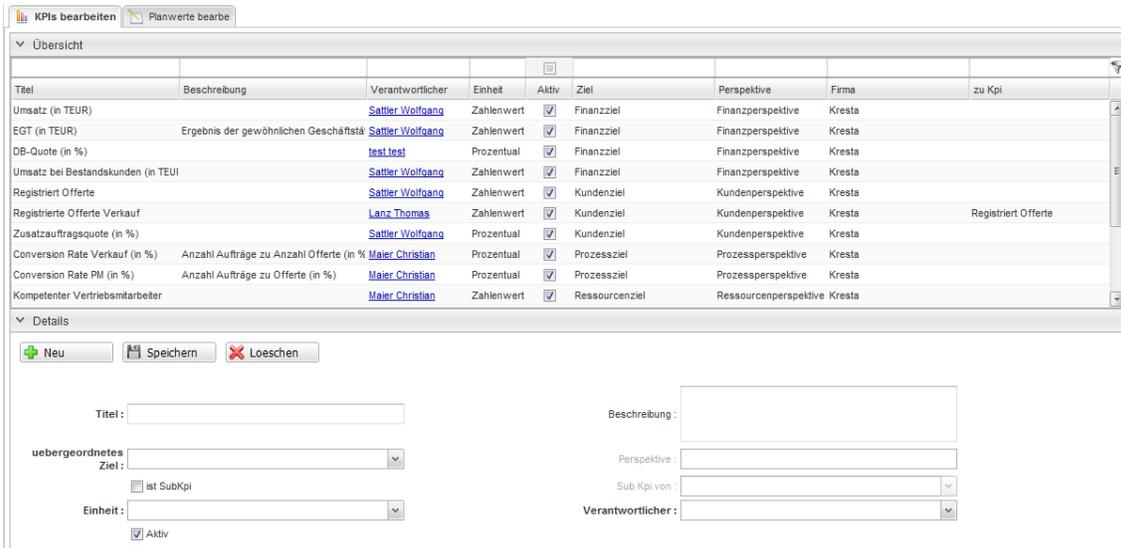


Abbildung 5.12: Kennzahlen erstellen/bearbeiten

5.3.4.5 Planwerte erstellen/bearbeiten

Zu jeder Kennzahl kann jährlich ein Planwert erfasst werden. Wie in Abbildung 5.13 veranschaulicht, werden zu einer Kennzahl die Planwerte aus bereits vergangenen Jahren ebenfalls in der Tabelle dargestellt. Dadurch ist sofort ersichtlich wie sich der Key Performance Indicator über die Jahre entwickelt bzw. verändert hat. Am Ende eines Jahres werden die Planwerte für das nächste Jahr erfasst. Deshalb wird ausgehend vom aktuellen Datum noch eine Spalte für den Planwert des nächsten Jahres in der Tabelle eingefügt. In der Detailansicht werden zu jedem Planwert die Schwellwerte für die Ampeldarstellung eines KPI abgespeichert.

The screenshot shows a software interface with two tabs: 'KPIs bearbeiten' and 'Planwerte bearbeiten'. The 'Planwerte bearbeiten' tab is active, displaying a table with the following data:

KPI	Verantwortlicher	Jahr 2006	Jahr 2007	Jahr 2008	Jahr 2009	Jahr 2010	Jahr 2011	Ziel	Firma
Umsatz (in TEUR)	Sattler Wolfgang					60 000		Finanzziel	Kresta
EGT (in TEUR)	Sattler Wolfgang					3.600		Finanzziel	Kresta
DB-Quote (in %)	test test					26 %		Finanzziel	Kresta
Umsatz bei Bestandskunden (in TEUR)	Sattler Wolfgang					20 000		Finanzziel	Kresta
Zusatzauftragsquote (in %)	Sattler Wolfgang					25.12 %		Kundenziel	Kresta
Conversion Rate Verkauf (in %)	Maier Christian					38 %	40 %	Prozessziel	Kresta
Conversion Rate PM (in %)	Maier Christian					37,85 %		Prozessziel	Kresta
Kompetenter Vertriebsmitarbeiter	Maier Christian					1		Ressourcenziel	Kresta
Vertriebsorientierter Pm	Lanz Thomas					21		Ressourcenziel	Kresta
Registrierte Offerte Team 1	Maier Christian					400		Kundenziel	Kresta
Registrierte Offerte Team 2	Maier Christian					200		Kundenziel	Kresta
Registrierte Offerte Team 3	Maier Christian					100		Kundenziel	Kresta
Registrierte Offerte Team 4	Maier Christian					150		Kundenziel	Kresta

Below the table, there is a 'Details' section with the following controls:

- Buttons: **Speichern** (Save) and **Loeschen** (Delete)
- Key Performance Indicator:
- Jahr:
- rot bis: %
- Einheit:
- Sollwert:
- gelb bis: %

Abbildung 5.13: Planwerte erstellen/bearbeiten

5.3.4.6 Maßnahmen

Alle Maßnahmen werden in einer Tabelle mit Filtermöglichkeit sämtlicher Spalten angezeigt (siehe Abbildung 5.14). Der Fortschritt der Maßnahme kann durch einen Schieberegler prozentual angegeben werden. Der zeitliche Fortschritt ist ebenfalls durch einen Schieberegler dargestellt. Zu einer Maßnahme können einzelne „ToDo's“ erfasst werden. Diese sind auf zwei Tabellen aufgeteilt (offene ToDo's und abgeschlossene ToDo's). Die Aufteilung der Maßnahme in ToDo's soll eine schrittweise Abarbeitung ermöglichen.

The screenshot displays a web-based management tool interface. At the top, there are navigation tabs: 'Strategy Map', 'KPI Übersicht', 'Maßnahmenübersicht', 'Meine KPIs', 'Meine Maßnahmen', and 'Infos'. The main content area is titled 'Meine Maßnahmen' and contains a table with the following data:

Titel	Beschreibung	KPI	Start	Ende	Zeit (%)	Fortschritt (%)	Verantwortlicher	Abgeschlossen	Firma
Umsatz steigern		Umsatz (in TEUR)	10/9/2010	31/12/2010	47,32 %	15 %	Maier Christian	<input type="checkbox"/>	Kresta

Below the table, the 'Details' section is expanded, showing:

- Buttons: **Speichern** (Save) and **Abschliessen** (Close).
- Maßnahme Details**:
 - Umsatz steigern | KPI: Umsatz (in TEUR)**
 - Start: 2010-09-10 Ende: 2010-12-31
 - Zeit [%] slider: 0 to 100, current value 47.
 - Kommentar/Fortschritt:
 - Kommentar: neuer Auftrag bei Firma XYZ
 - Fortschritt [%] slider: 0 to 100, current value 15.
- abgeschlossene ToDo's**:
 - ToDo: neuen Auftrag fixiert
- offene ToDo's**:
 - ToDo: Auftrag für ABC (with 'Abschliessen' and 'Loeschen' buttons)
- neuer ToDo**:
 - Field: neuer ToDo
 - Button: **Hinzufügen** (Add)

Abbildung 5.14: Maßnahmen-Übersicht

Kapitel 6

Softwaretests und Resultate

Während der Entwicklung der Software wurden immer wieder Tests von einigen Benutzern im Unternehmen durchgeführt. Die Resultate werden in folgenden Abschnitten kurz erläutert.

6.1 Fehler und Verbesserungen

Die entwickelte Webanwendung wurde im Unternehmen von mehreren Benutzern ausführlich getestet. Dabei wurden einige Fehler in der Programmierung gefunden, sowie Verbesserungsvorschläge für die Usability mit eingebracht. Im Folgenden werden die wichtigsten Punkte diskutiert.

6.1.1 Mandantenfähigkeit

Ursprünglich war die Software nur für einen Mandanten ausgelegt. Pro Jahr konnte nur eine BSC für ein Unternehmen erstellt werden. Die Firma hatte sich jedoch in der Zwischenzeit vergrößert und ein weiteres Unternehmen dazugekauft. Aus diesem Grund muss eine Möglichkeit für mehrere Mandanten geschaffen werden. Die Software und das Datenmodell wurden entsprechend erweitert um mehrere Firmen und Scorecards erstellen bzw. bearbeiten zu können. Zusätzlich wurde der Use Case Firma erstellen/bearbeiten/löschen in die Anforderungsanalyse aufgenommen.

Berechtigung	Admin	Superuser	User
Navigation	Ja	Nein	Nein
Stammdaten einsehen/bearbeiten	Ja	Nein	Nein
Strategy Map anzeigen	Ja	Ja	Ja
Strategy Map bearbeiten	Ja	Nein	Nein
KPI Übersicht anzeigen	Ja	Ja	Nein
KPI Übersicht bearbeiten	Ja	Nein	Nein
Maßnahmen Übersicht anzeigen	Ja	Ja	Nein
Maßnahmen Übersicht bearbeiten	Ja	Nein	Nein
Maßnahmen Übersicht - ToDo's bearbeiten	Ja	Nein	Nein
Meine KPIs anzeigen	Ja	Ja	Ja
Meine KPIs bearbeiten	Ja	Ja	Ja
Meine Maßnahmen anzeigen	Ja	Ja	Ja
Meine Maßnahmen bearbeiten	Ja	Ja	Ja
Info anzeigen	Ja	Ja	Ja
Info bearbeiten	Ja	Nein	Nein

Tabelle 6.1: Berechtigungsmatrix der Applikation

6.1.2 Berechtigungsmatrix

Die einzelnen Berechtigungen für die Usergruppen wurden erst in der Testphase durch die Benutzer bestimmt. Die Mitarbeiter des Unternehmens konnten sich zuvor nicht einigen welche Berechtigungen für die Anwender der Software festgelegt werden sollten. Schlussendlich wurde die in der Tabelle 6.1 dargestellte Berechtigungsmatrix erstellt.

6.1.3 Berechnung und Darstellung der KPI Baumstruktur

Ein komplexerer Teil der Applikation ist die Berechnung und Darstellung der Kennzahlen-Hierarchie. Für Kennzahlen einer Baumstruktur gelten mehrere Einschränkungen:

- Es dürfen in der KPI Übersicht nur Kennzahlen mit zugewiesenen Planwerten angezeigt werden
- Planwerte dürfen nur für Blätter der KPI-Baumstruktur festgelegt werden
- Die Planwerte der übergeordneten Kennzahlen werden aus den Planwerten der untergeordneten KPIs der Hierarchie berechnet

Der Source Code der Berechnung ist im Anhang zu finden.

6.1.4 Verbesserungen in der Darstellung

Durch die Softwaretests wurden auch einige Verbesserungsvorschläge in der Visualisierung der Daten eingebracht.

6.1.4.1 Strategy Map

Die Darstellung der Strategy Map über die Webanwendung war für die Benutzer nicht zufriedenstellend. Die Software generierte ein einfaches Bild der definierten Unternehmensziele mit den zugeordneten Kennzahlen. Eine farbliche Darstellung, sowie das Anzeigen von Abhängigkeiten war nicht möglich. Deshalb wurde die Darstellung der Strategiemappe umgebaut und es wird nur noch ein Datei-Upload bereitgestellt. Die Strategiemappen können nun als Bild hoch geladen werden und werden von der Anwendung auf der Startseite dargestellt.

6.1.4.2 Erfassungsdatum der Istwerte

Die Istwerte der Kennzahlen können jederzeit erfasst werden. Es kann beispielsweise der Istwert für den Monat Januar im August eingetragen werden. Ursprünglich wurde kein Datum der Erfassung innerhalb der Anwendung gespeichert. Eine Überprüfung, ob die Eintragungen der Benutzer fristgerecht erfolgen, war nicht möglich. Deshalb wird nun zu jeder Erfassung eines Istwertes ein Zeitstempel abgespeichert. In der KPI-Detailansicht wird der Zeitstempel der Erfassung in der Tabelle des Soll-Ist-Vergleich angezeigt (siehe Abbildung 6.1).

6.1.4.3 Spaltenbreiten der Übersichtsseiten

Standardmäßig werden alle Spalten des „List Grid“ Controls von Smart GWT gleich breit dargestellt. Vor allem bei der Monatsansicht der Kennzahlen-Übersicht waren alle Spalten sehr zusammengerafft und die Daten nicht sofort ersichtlich. Der Benutzer musste immer händisch die gewünschte Spaltenbreite nachstellen. Die Software wurde dahingehend angepasst, dass beim Erstellen des „List Grid“ Controls für alle Spalten eine bestimmte Breite vordefiniert und falls notwendig ein Scrollbalken eingefügt wird (siehe Abbildung 6.2).

Monat	Soll	Ist	Datum	Kommentar
Januar	300	250	26.08.2010	
Februar	300	220	26.08.2010	
März	300	200	26.08.2010	
April	300	190	26.08.2010	
Mai	300	80	26.08.2010	
Juni	300	320	26.08.2010	
Juli	300	230	26.08.2010	
August	300	50	26.08.2010	
September	300	350	01.09.2010	
Oktober	300			
November	300			
Dezember	300			

Abbildung 6.1: Erfassungsdatum der Istwerte

KPI	Verantwortlicher	Soll 2009	Ist 2009	Soll 2010	Ist 2010	1	2	3	4				
						S 1	I 1	S 2	I 2	S 3	I 3	S 4	I 4
Umsatz (in TEUR)	Sattler Wolfgang	0	0	60.000	31.700	5.000	6.000	5.000	5.000	5.000	2.000	5.000	7
EGT (in TEUR)	Sattler Wolfgang	0	0	3.600	1.890	300	250	300	220	300	200	300	
DB-Quote (in %)	test test	0 %	0 %	26 %	9,42 %	26 %	12 %	26 %	27 %	26 %	14 %	26 %	
Umsatz bei Bestandskunden (in	Sattler Wolfgang	0	0	20.000	13.540	1.666,7	1.590	1.666,7	1.600	1.666,7	1.200	1.666,7	1

Abbildung 6.2: Spaltenbreite und Scrollbalken

6.1.5 Logout Button

Innerhalb der Anwendung gab es zunächst keinen Logout-Button. Sobald der Browser vom Benutzer geschlossen wurde, war die Sitzung somit beendet. Erst beim nächsten Start erscheint wieder die Login-Maske und der Benutzer musste sich wieder neu anmelden. Dies sorgte bei den Testern jedoch für Verwirrung. Viele User sind es beispielsweise von Webmail-Anwendungen gewohnt sich wieder von der Applikation abzumelden. Deshalb wurde nachträglich ein Logout-Button eingebaut, der den Benutzer wieder auf die Login-Maske weiterleitet.

6.1.6 Zusätzliche Benachrichtigungen

Wurden Daten in der Anwendung gespeichert oder gelöscht, erhielt der Benutzer keine weiteren Informationen. Die Tester waren sich deshalb nicht sicher, ob die durchgeführten Aktionen auch wirklich fehlerfrei funktioniert hatten. Deshalb wur-

den zusätzliche Benachrichtigungen eingefügt, die den User nach jeder Aktion einblendet werden. Ein Beispiel hierfür ist in Abbildung 6.3 ersichtlich.

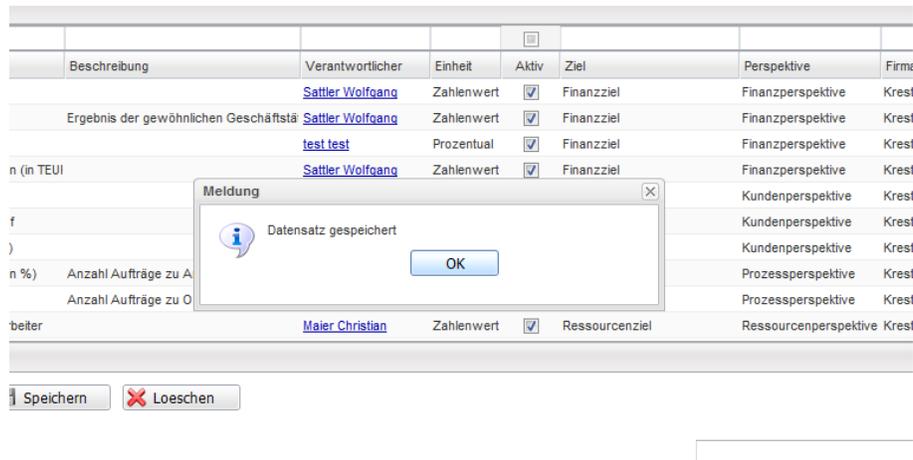


Abbildung 6.3: Zusätzliche Benachrichtigungen

6.2 Evaluierung der Software

Eine Evaluierung der Software mit allen Benutzern war im Zuge der Diplomarbeit nicht mehr möglich. Die Applikation wird erst mit Beginn des Jahres 2011 im Unternehmen eingesetzt. Balanced Scorecards müssen immer für ein gesamtes Jahr erfasst werden. Der Echtbetrieb der Software im Jahr 2010 ist deshalb nicht mehr sinnvoll. Eine komplette Erfassung sämtlicher Kennzahlen für das Jahr 2010 ist auf Grund mangelnder Daten nicht mehr möglich. Durch die fehlende Evaluierung entsteht eine gewisse Problematik in der Softwareentwicklung der Applikation.

6.2.1 Definition Evaluierung

Eine Evaluierung umfasst das Sammeln und Kombinieren von Daten mit einem gewichtetem Satz von Skalen mit denen entweder vergleichende oder numerische Beurteilungen möglich sind. [62] Es gibt drei wesentliche Bestimmungselemente für Evaluationen [63]:

- **systematisch:** Grundlage der Evaluierung ist eine systematisch gewonnene Datenbasis über Voraussetzungen, Prozesse und Wirkungen.

- **methodisch:** Die Daten werden mit Hilfe einer bestimmten Methode erfasst und gewichtet.
- **prozessorientiert:** Eine Evaluierung bezieht sich nicht primär auf die Bewertung des Verhaltens, sondern ist fester Bestandteil der Entwicklung, Realisierung und Kontrolle.

Dix definiert für die Software-Evaluierung drei Hauptziele [64]:

- Ermittlung von Problemen der Benutzer im Umgang mit der Software
- Bewertung der Funktionalität der Software
- Ermittlung der Effekte einer Software auf die Benutzer

Prinzipiell gibt es zwei Arten von Evaluierungen:

- **formativ:** Es erfolgt eine Evaluierung während der Software-Entwicklung
- **summativ:** Es wird eine bestehende Software bewertet

Formative und summative Evaluierungen können bezogen auf den „Software-Lifecycle“ wie in Abbildung 6.4 dargestellt werden.

Je nachdem welches Ziel mit einer Evaluierung verfolgt wird, ergeben sich verschiedene Kriterien, die in die Evaluierung einfließen. Diese Kriterien legen fest, was genau und wie umfangreich etwas evaluiert werden soll. Beispiele hierfür sind Attribute wie Akzeptanz, Erlernbarkeit, Fehlertoleranz, Persönlichkeitsförderlichkeit, Handhabbarkeit, Wartbarkeit, Motivation usw. Selten ist es möglich alle Kriterien gleich gut mit einer Software zu erfüllen, da sehr oft Abstriche gemacht werden müssen. [66]

6.2.2 Warum sollte evaluiert werden?

Eine Evaluierung dient vor allem der Qualitätssicherung. Durch eine ausführliche Evaluierung kann sichergestellt werden, ob sich die Anwendung wirklich so verhält, wie vom Entwickler erwartet. Des Weiteren kann mit Hilfe der Evaluierung festgestellt werden, ob die Applikation überhaupt den Bedürfnissen der Benutzer entspricht. Die Stärken und Schwächen der Software können gefunden und eventuelle Verbesserungen diskutiert werden. Durch die Bewertung kann der Nutzen der Software im Unternehmen bestimmt werden. [67]

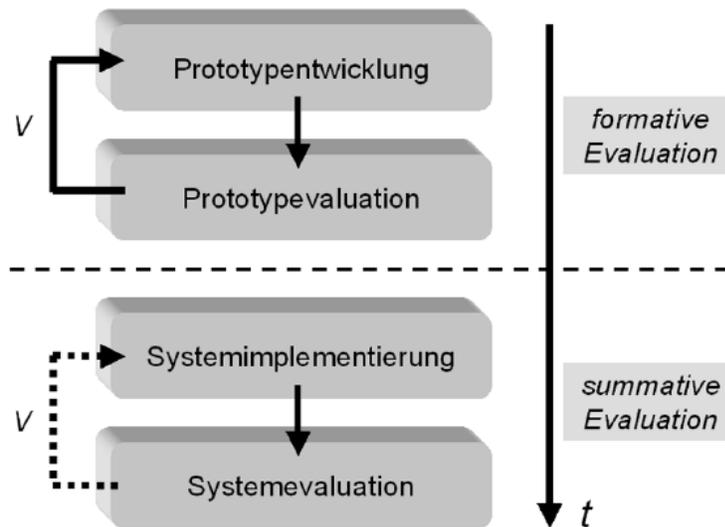


Abbildung 6.4: Formative und summative Evaluierung während des „Software-Lifecycle“ [65]

6.2.3 Durchführung einer Software-Evaluierung

Es gibt mehrere Möglichkeiten eine Software-Evaluierung durchzuführen. Entscheidend sind dabei die verschiedenen Evaluierungsmittel, mit denen die einzelnen Kriterien einer Evaluierung beurteilt werden können. Prinzipiell werden die Evaluierungsmittel nach dem Grad, inwieweit Menschen bei der Beurteilung der Evaluierungskriterien beteiligt sind, unterschieden. Nach Oppermann können die Evaluierungsmittel folgendermaßen klassifiziert werden [66]:

1. Subjektive Evaluierungsmittel

Die Evaluierungskriterien werden von den Menschen beurteilt, die das System auch benutzen. Hier unterscheidet man zwischen drei Möglichkeiten:

- mündliche Befragung: Den Benutzern werden Fragen gestellt, die zur Beurteilung der gewählten Evaluationskriterien dienen
- schriftliche Befragung: Hier wird ein Fragebogen erstellt und von den Benutzern ausgefüllt.
- lautes Denken: Die Benutzer sollen während einer Aufgabenbewältigung ihre Überlegungen und Probleme laut vor sich her sagen.

2. Objektive Evaluierungsmittel

Bei den objektiven Evaluierungsmitteln sollen möglichst alle subjektiven Einflüsse der Benutzer ausgeschlossen werden. Man unterscheidet folgende Beobachtungen:

- anwesende Beobachtung: Der Benutzer wird von einer weiteren Person beobachtet und die Evaluierungskriterien werden anhand beobachteter Handlungen beurteilt.
- abwesende Beobachtung: Der Benutzer wird indirekt durch eine Videoaufzeichnung beobachtet, die anschließend ausgewertet wird.

3. Leitfadenorientierte Evaluierungsmittel

Bei den leitfadenorientierten Evaluierungsmitteln wird die Software entlang eines Prüfleitfadens beurteilt. Bei diesem Beurteilungstyp ist kein Benutzer notwendig.

6.3 Planung der Evaluierung

Die Software sollte nach der Einführung im Unternehmen im Jänner 2011 evaluiert werden. Da es in einem Unternehmen eher schwierig ist, umfangreiche Evaluierungen mit mehreren Personen durchzuführen, muss diese möglichst einfach und verständlich gestaltet werden. Für die Evaluierung sollten zuvor Test-Szenarien festgelegt werden, die dann von den verschiedenen Benutzern durchgeführt werden. Dadurch wird geprüft, an welchen Stellen Schwierigkeiten bei der Benutzung auftreten. Zusätzlich sollte ein Fragebogen zusammengestellt werden. Die Benutzer können mit Hilfe des Fragebogens die Test-Szenarien bewerten. Des Weiteren wäre es von Vorteil die Benutzer während der Evaluierung mündlich zu befragen, da für einige User wichtige Bewertungskriterien im Fragebogen eventuell nicht aufgelistet sind. Es sollte also ein *subjektives Evaluierungsmittel mit schriftlicher und mündlicher Befragung* eingesetzt werden. Die Auswertung des Fragebogens und die Interviews mit den Benutzern sollten die Schwachstellen der Software aufdecken und Verbesserungsvorschläge für die Benutzerfreundlichkeit analysieren.

Kapitel 7

Zusammenfassung und Ausblick

Diese Arbeit erläutert die Entwicklung und Umsetzung einer Softwarelösung für Balanced Scorecards in einer Organisation. Viele Unternehmen verwenden Tabellenkalkulationsprogramme für das Performance Management. Jedoch ist die Fehleranfälligkeit bei Tabellenkalkulationsprogrammen sehr hoch und die Daten befinden sich verteilt auf einzelnen Desktops. Durch die Entwicklung einer Webanwendung können die Daten zentral auf einen Datenbankserver gespeichert und ein kontrollierter Zugriff mit Berechtigungssteuerung auf die Daten sichergestellt werden. Die Daten werden automatisch in einer Tabelle mit Ampelsystem dargestellt, wodurch der Fortschritt der Strategieumsetzung sofort ersichtlich ist. Des Weiteren können Trends der Kennzahlen durch verschiedene Diagramme (Tachometer, Balkendiagramm, Pfeildarstellung) schnell erkannt werden. Das integrierte Maßnahmenwesen soll eingeleitete Schritte zur Verbesserung der Kennzahlen und somit die Strategiesteuerung der Organisation unterstützen. Ein weiterer zentraler Punkt der Applikation liegt in der genauen Erfassung und Dokumentation der Kennzahlen und Maßnahmen. Somit ist eine genaue Überwachung über die Eingaben der Werte, sowie eine Kontrolle über den Fortschritt von Maßnahmen möglich.

Mit der Implementierung der Balanced Scorecard wurde nur ein Teil des Performance Managements in einer Softwarelösung umgesetzt. Das Unternehmen plant die Anwendung auszubauen und noch weitere Module in die Webanwendung aufzunehmen. Durch die Webanwendung erhält die Management Ebene einen genauen Überblick über die Umsetzung der Unternehmensstrategie. Die Leistungen von einzelnen Abteilungen und Mitarbeitern werden messbar. Außerdem können Fehlentwicklungen

innerhalb der Organisation frühzeitig erkannt und eingeleitete Gegenmaßnahmen zentral dokumentiert werden.

Anhang A

Anhang

A.1 Source Code

Folgende Abschnitte beinhalten wichtige Codeausschnitte der Anwendung.

A.1.1 Kennzahlen-Hierarchie Berechnung

Zunächst müssen alle „Blätter“ der Kennzahlen Hierarchien geladen werden.

Listing A.1: Laden der Kennzahlen Blätter

```
public ArrayList<BscSurveyGridItem> getBscSurveyGridItems(int
    companyId, int perspectiveId, int year) {
    // all leaves
    ArrayList<Kpi> kpiList = this.getKpis(companyId, perspectiveId,
        year);

    LinkedHashMap<Integer, BscSurveyGridItem> map = new
        LinkedHashMap<Integer, BscSurveyGridItem>();

    BscSurveyGridItem bsc;
    BscSurveyGridItem parent;
    User u;
    PlanValue p;

    for (Kpi k: kpiList) {
        bsc = new BscSurveyGridItem(k);
```

```

    // get Person in Charge
    u = getUser(bsc.getPersonInChargeId());
    bsc.setPersonInCharge("<a href=\"mailto:\" + u.getEmail()
        +\"\\\">\" + u.getLastName() + \" \" + u.getFirstName() +
        \"</a>");
    // get Values
    p = getPlanValue(bsc.getId(), year-1);
    if (p != null) {
        bsc.setPlanValueLastYear(p.getValue());
        bsc.setActualValueLastYear(getSumActualValue(p.getId(),
            bsc.getUnit()));
    }

p = getPlanValue(bsc.getId(), year);
if (p != null) {
    bsc.setPlanValueThisYear(p.getValue());
        bsc.setThresholdRed(p.getThresholdRed());
        bsc.setThresholdYellow(p.getThresholdYellow());
    bsc.setActualValueThisYear(getSumActualValue(p.getId(), bsc.
        getUnit()));
    ArrayList<ActualValue> vals = getActualValues(p.getId());

    double sum = 0;
    double lastVal = 0;

    if (vals.size() > 0) {
        for (ActualValue v : vals) {
            bsc.setActualValue(v.getMonth()-1, v.getValue());
            sum += v.getValue();
        }

        lastVal = vals.get(vals.size()-1).getValue();
        bsc.setTrend(getTrend(sum, lastVal, vals.size()));
    }
    else
        bsc.setTrend(0);
}

```

```

map.put(bsc.getId(), bsc);

if (bsc.getParentId() != -1) {
    if (map.containsKey(bsc.getParentId())) {
        parent = map.get(bsc.getParentId());
        parent.put(bsc.getId(), bsc);
        findParent(parent, map);
    }
    else {
        parent = new BscSurveyGridItem(getKpi(bsc.getParentId()
            ()));
        parent.put(bsc.getId(), bsc);
        // get Person in Charge
        u = getUser(parent.getPersonInChargeId());
        parent.setPersonInCharge("<a href=\"mailto:\" + u
            .getEmail() + \">\" + u.getLastName() + \" \" +
            u.getFirstName() + \"</a>");
        map.put(parent.getId(), parent);
        findParent(parent, map);
    }
}
}
}

return new ArrayList<BscSurveyGridItem>(map.values());
}

```

Anschließend werden ausgehend von den Blättern alle übergeordneten Knoten in der Baumstruktur gesucht.

Listing A.2: Laden der Knoten der Baumstruktur

```

public void findParent(BscSurveyGridItem bsc, LinkedHashMap<
    Integer, BscSurveyGridItem> map) {
    BscSurveyGridItem parent;
    User u;

    if (bsc.getParentId() != -1) {
        if (map.containsKey(bsc.getParentId())) {

```

```

        parent = map.get(bsc.getParentId());
        parent.put(bsc.getId(), bsc);
        map.put(parent.getId(), parent);
        findParent(parent, map);
    }
    else {
        parent = new BscSurveyGridItem(getKpi(bsc.getParentId()
            ));
        parent.put(bsc.getId(), bsc);
        // get Person in Charge
        u = getUser(parent.getPersonInChargeId());
        parent.setPersonInCharge("<a href=\"mailto:\" + u.
            getEmail() + \">\" + u.getLastName() + \" \" + u.
            getFirstName() + \"</a>");
        map.put(parent.getId(), parent);

        findParent(parent, map);
    }
}
else {
    bsc.calculateValues();
}
}

```

Zuletzt werden alle Werte der Knoten in einer Rekursion ausgehend von den Wurzelknoten berechnet. Die Baumstruktur muss also zwei mal durchlaufen werden. Zuerst wird von unten nach oben die Kennzahlen-Hierarchie aufgebaut. Anschließend werden von oben nach unten in einer Rekursion alle Werte der Knoten berechnet.

Listing A.3: Rekursion zur Berechnung der Werte der Knoten

```

public void calculateValues() {
    double sum = 0;
    double lastVal = 0;
    if (children.size() > 0) {
        clearValues();
        Collection<BscSurveyGridItem> c = children.values();
        Iterator<BscSurveyGridItem> itr = c.iterator();
    }
}

```

```
while (itr.hasNext()) {
    BscSurveyGridItem b = itr.next();
    b.calculateValues();

    // planned value last year
    planValueLastYear += b.getPlanValueLastYear();

    // actual value last year
    actualValueLastYear += b.getActualValueLastYear();

    // planned value this year
    planValueThisYear += b.getPlanValueThisYear();

    // actual value this year
    actualValueThisYear += b.getActualValueThisYear();

    // thresholds for traffic lights
    thresholdRed += b.getThresholdRed();
    thresholdYellow += b.getThresholdYellow();

for (int i = 0; i < 12; i++) {
    actualValues[i] += b.getActualValue(i);
    sum += actualValues[i];

    if (actualValues[i] > 0)
        lastVal = actualValues[i];

    if (lastVal > sum/12)
        trend = 1;
    else if (lastVal < sum/12)
        trend = -1;
    else
        trend = 0;
}
}

thresholdRed /= children.size();
```

```
thresholdYellow /= children.size();

if (unit.equals("Prozentual")) {
    planValueLastYear /= children.size();
    actualValueLastYear /= children.size();

    planValueLastYear /= children.size();
    planValueThisYear /= children.size();

    for (int i = 0; i < 12; i++)
        actualValues[i] /= children.size();
}
}
```

Literaturverzeichnis

- [1] Thomas Walter. *Kompendium der Web-Programmierung*. X.media.press. Springer Berlin Heidelberg, 2008.
- [2] Tim Berners-Lee. Worldwideweb: Proposal for a hypertext project. <http://www.w3.org/Proposal.html>, 1990. Zugriff am 05.10.2010.
- [3] Torsten Stapelkamp. Web 2.0 – das mitmach-internet. In *Web X.0*, X.media.press, pages 398–475. Springer Berlin Heidelberg, 2010.
- [4] Tom Alby. *Web 2.0. Konzepte, Anwendungen, Technologien*. Hanser Verlag, Februar 2008.
- [5] *Web 2.0: Neue Perspektiven für Marketing und Medien*. Springer, November 2007.
- [6] Tim O’Reilly. What is web 2.0. <http://oreilly.com/web2/archive/what-is-web-20.html>, 2005. Zugriff am 29.09.2010.
- [7] Andy Bosch. *Java Server Faces - das Standard-Framework zum Aufbau webbasierter Anwendungen*. Addison-Wesley, 2004.
- [8] Hans-Dirk Walter. Rich internet applications – eine perfekte kombination benutzerfreundlicher schnittstellen mit webtechnologie. *Informatik-Spektrum*, 31:333–343, 2008.
- [9] Sonja Bettel. Warum web 2.0? oder: Was vom web 2.0 wirklich bleiben wird. In Andreas Blumauer and Tassilo Pellegrini, editors, *Social Semantic Web*, X.media.press, pages 23–41. Springer Berlin Heidelberg.
- [10] Hassan Moussaoui and Klaus Zeppenfeld. Ajax-grundkenntnisse. In *AJAX, Informatik im Fokus*, pages 25–44. Springer Berlin Heidelberg, 2008.

- [11] Hassan Moussaoui and Klaus Zeppenfeld. Ajax-technologien. In *AJAX*, Informatik im Fokus, pages 45–110. Springer Berlin Heidelberg, 2008.
- [12] Deepak Vohra. Ajax with java-gwt. In *Ajax in Oracle JDeveloper*, pages 61–87. Springer Berlin Heidelberg, 2008.
- [13] Jeff Dwyer. Why gwt? In *Pro Web 2.0 Application Development with GWT*, pages 3–15. Apress, 2008.
- [14] Grant Slender. Overview of ext gwt and gwt. In *Developing with Ext GWT*, pages 1–7. Apress, 2009.
- [15] Bram Smeets, Uri Boness, and Roald Bankras. Introducing google web toolkit (gwt). In *Beginning Google Web Toolkit*, pages 21–44. Apress, 2008.
- [16] V. Gupta. *Accelerated GWT: Building Enterprise Google Web Toolkit Applications*. Apress Series. Apress, 2008.
- [17] M. Seemann. *Das Google Web Toolkit: GWT*. O’Reilly, 2007.
- [18] M. MacDonald. *Pro Silverlight 4 in C#*. Pro to Expert Series. Apress, 2010.
- [19] Silverlight architecture. <http://msdn.microsoft.com/en-us/library/bb404713%28VS.95%29.aspx>, 2010. Zugriff am 13.10.2010.
- [20] Cornelia Heinisch, Frank Müller-Hofmann, and Joachim Goll. Applets. In *Java als erste Programmiersprache*, pages 771–800. Teubner, 2007.
- [21] Gregor Fischer and Jürgen Gudenberg. Applets. In *Programmieren in Java 1.5*, Xpert.press, pages 263–266. Springer Berlin Heidelberg, 2005.
- [22] D. Wißmann. *JavaServer Pages*. W3L GmbH, 2009.
- [23] Javaserer pages overview. <http://java.sun.com/products/jsp/overview.html>, 2010. Zugriff am 13.10.2010.
- [24] Apache tomcat. <http://tomcat.apache.org/>, 2010. Zugriff am 13.10.2010.
- [25] Introduction to jsp. <http://www.visualbuilder.com/jsp/tutorial/>, 2010. Zugriff am 13.10.2010.

- [26] Qusay H. Mahmoud. Servlets and jsp pages best practices. <http://www.oracle.com/technetwork/articles/javase/servlets-jsp-140445.html>, 2003. Zugriff am 13.10.2010.
- [27] Matthew MacDonald and Mario Szpuszta. Introducing asp.net. In *Pro ASP.NET 2.0 in C# 2005*, pages 3–22. Apress, 2005.
- [28] Lee Wei-Meng. What is asp.net. <http://ondotnet.com/pub/a/dotnet/2005/09/19/what-is-asp-net.html>, 2005. Zugriff am 14.10.2010.
- [29] C. Wenz, J. Kotz, and K. Samaschke. *ASP.net 3.5 mit Visual Basic 2008: Leistungsfähige Webapplikationen programmieren*. Programmer's Choice. Addison Wesley Verlag, 2008.
- [30] J. Lengstorf. *PHP for Absolute Beginners*. For Absolute Beginners. Apress, 2009.
- [31] D. Sklar. *Einführung in PHP 5*. Safari Books Online. O'Reilly, 2004.
- [32] Netcraft. October 2010 web server survey. <http://news.netcraft.com/archives/category/web-server-survey/>, 2010. Zugriff am 15.10.2010.
- [33] S. Wolfgarten. *Apache Webserver 2*. Addison-Wesley, 2004.
- [34] V. Hockmann and H.D. Knöll. *Profikurs Sicherheit von Web-Servern: Ein praxisorientiertes Handbuch- Grundlagen, Aufbau und Architektur- Schwachstellen und Hintertüren- Konkrete Praxisbeispiele realisiert unter Windows und Unix/-Linux*. Vieweg+Teubner Verlag, 2008.
- [35] C. Adams, C. Agramont, and G. Whitley. *How to Cheat at IIS 7 Server Administration*. How to Cheat. Syngress, 2007.
- [36] Joachim Rossberg and Rickard Redler. Internet information services. In *Pro Scalable .NET 2.0 Application Designs*, pages 307–343. Apress, 2006.
- [37] Reagan Templin. Introduction to iis 7 architecture. <http://learn.iis.net/page.aspx/101/introduction-to-iis-7-architecture/>, 2010. Zugriff am 15.10.2010.

- [38] René Steiner. *Grundkurs relationale Datenbanken: Einführung in die Praxis der Datenbankentwicklung für Ausbildung, Studium und IT-beruf*. Vieweg+Teubner Verlag, 2009.
- [39] M. Pollakowski. *Grundkurs MySQL und PHP: So entwickeln sie Datenbanken mit Open Source Software*. Vieweg, 2005.
- [40] R. Greenwald, R. Stackowiak, and J. Stern. *Oracle Essentials: Oracle Database 11g*. Essentials Series. O'Reilly, 2007.
- [41] Euan Garden. Sql server is really a sybase product not a microsoft one. <http://blogs.msdn.com/b/euanga/archive/2006/01/19/514479.aspx>, 2006. Zugriff am 21.10.2010.
- [42] K. Konopasek. *SQL Server 2008- der schnelle Einstieg: Abfragen, Transact-SQL, Entwicklung und Verwaltung*. Addison-Wesley, 2008.
- [43] H. Balzert. *Java: objektorientiert programmieren: Vom objektorientierten Analysemodell bis zum objektorientierten Programm*. W3L GmbH, 2010.
- [44] Christoph Wegener. Open-source vs. closed-source. *Datenschutz und Datensicherheit - DuD*, 33:189–189, 2009.
- [45] R.S. Kaplan and D.P. Norton. *Balanced Scorecard: Strategien erfolgreich umsetzen*. Handelsblatt-Bücher. Schäffer-Poeschel Verlag, 1997.
- [46] Arthur M. Schneiderman. Why balanced scorecards fail. *Journal of Strategic Performance Measurement*, 1999.
- [47] H. Ehrmann. *Unternehmensplanung*. Kompendium der praktischen Betriebswirtschaft. Kiehl, 2002.
- [48] T. Pietsch and T. Memmler. *Balanced Scorecard erstellen: Kennzahlenermittlung mit Data Mining*. Schmidt Erich Verlag, 2003.
- [49] Horvath und Partner GmbH. *Balanced Scorecard umsetzen*. Schäffer-Poeschel, 2007.
- [50] H. Jung. *Allgemeine Betriebswirtschaftslehre*. Oldenbourg Wissensch.Vlg, 2004.

- [51] H.R. Friedag and W. Schmidt. *Balanced Scorecard.: Mehr als ein Kennzahlen-system*. Haufe Mediengruppe, 2002.
- [52] A. Müller. *Strategisches Management mit der Balanced Scorecard*. Kohlhammer W., 2005.
- [53] M.K. Welge and A. Al-Laham. *Strategisches Management: Grundlagen - Prozess - Implementierung*. Gabler, 2008.
- [54] J. Weber and U. Schäffer. *Balanced Scorecard & Controlling: Implementierung - Nutzen für Manager und Controller - Erfahrungen in deutschen Unternehmen*. Gabler, 2000.
- [55] Burkhard Rüth. *Balanced Scorecard. Von der Strategie zum Tagesgeschäft*. Weka, 2004.
- [56] B. Morganski. *Balanced Scorecard*. Vahlen, 2003.
- [57] R.S. Kaplan, D.P. Norton, P. Horvath, and D. Kralj. *Die strategiefokussierte Organisation: Führen mit der Balanced Scorecard*. Schäffer-Poeschel, 2001.
- [58] P. Horvath. *Controlling und finance excellence: Herausforderungen und Best-Practice-Lösungsansätze*. Schäffer-Poeschel, 2006.
- [59] Thomas Wunder. Wie konkret muss eine balanced scorecard sein. *Controller Magazin*, 2:133–139, 2001.
- [60] Ray Ryan. Google web toolkit architecture: Best practices for architecting your gwt app. <http://www.google.com/intl/de-DE/events/io/2009/sessions/GoogleWebToolkitBestPractices.html>, 2009. Zugriff am 29.10.2010.
- [61] Gwt rpc mechanism. <http://code.google.com/webtoolkit/doc/latest/tutorial/RPC.html>, 2010. Zugriff am 28.10.2010.
- [62] H. Wottawa and H. Thierau. *Lehrbuch Evaluation*. Huber, 2003.
- [63] R. Fricke. *Evaluation von Multimedia*. Arbeiten aus dem Institut für Empirische Pädagogik und Instruktionspsychologie // Technische Universität Braunschweig.
- [64] A. Dix. *Mensch, Maschine, Methodik*. Prentice Hall, 1995.

- [65] A. Holzinger. *Basiswissen Multimedia*. Number Bd. 2 in Vogel-Fachbuch. Vogel, 2001.
- [66] R. Oppermann, B. Murchner, M. Koch, and H. Reiterer. *Software-Ergonomische Evaluation: Der Leitfaden Evadis II*. Mensch-Computer-Kommunikation, Band 5. Walter De Gruyter Incorporated, 1992.
- [67] United States. Dept. of Health and Human Services. *The program manager's guide to evaluation: an evaluation handbook series from the Administration on Children, Youth and Families*. U.S. Dept. of Health and Human Services, Administration for Children, and Families, Administration on Children, Youth and Families, 1998.