

Entwicklung einer interaktiven, mobilen E-Learning Anwendung basierend auf .Net

Diplomarbeit

zur Erlangung des Grades eines Diplom-Ökonomen des
Fachbereichs Wirtschaftswissenschaften der
Universität Hannover

vorgelegt von:
Philipp Maske



Prüfer: Prof. Dr. Michael H. Breitner

Hannover, den 25.11.2004

Inhaltsverzeichnis

| | |
|--|------------|
| Abbildungsverzeichnis | V |
| Tabellenverzeichnis | XI |
| Verzeichnis der Programmlistings..... | XII |
| Abkürzungsverzeichnis | XIV |
| 1 Einleitung | 1 |
| 1.1 Motivation | 1 |
| 1.2 Zielsetzung und Vorgehensweise | 3 |
| 1.3 Softwareentwicklung mit CASE | 10 |
| 2 E-Learning | 12 |
| 2.1 Begriffliche Grundlagen | 12 |
| 2.2 Geschichte | 13 |
| 2.3 Formen..... | 15 |
| 2.4 Einsatzpotentiale und Beispiele..... | 18 |
| 2.4.1 E-Learning in der Hochschule..... | 18 |
| 2.4.2 E-Learning in der Schule..... | 21 |
| 2.4.3 E-Learning im Unternehmen..... | 23 |
| 2.5 Mobiles E-Learning als Herausforderung | 25 |
| 2.5.1 Vom E-Learning zum M-Learning..... | 25 |
| 2.5.2 Nutzenpotentiale des M-Learning | 26 |
| 2.5.3 Praxisbeispiele | 28 |
| 2.5.4 Drahtlose Internetverbindung als Problemfeld..... | 31 |
| 2.6 Zusammenfassung | 34 |
| 3 Systementwicklung | 36 |
| 3.1 Analysephase | 36 |
| 3.1.1 Istanalyse des UbiLearn Systems | 36 |
| 3.1.1.1 Datenmodell | 38 |
| 3.1.1.2 Fragetypen und das System der Punktevergabe | 42 |
| 3.1.1.3 Datenbankserver | 44 |
| 3.1.2 Istanalyse der Marktsituation mobiler Endgeräte..... | 48 |
| 3.1.2.1 Geräteklassen..... | 49 |
| 3.1.2.1.1 PDA | 50 |
| 3.1.2.1.2 Smartphone..... | 51 |
| 3.1.2.2 Hardwareplattformen und Hersteller | 52 |
| 3.1.2.2.1 Nokia | 52 |
| 3.1.2.2.2 PalmOne | 54 |
| 3.1.2.2.3 Hewlett-Packard | 55 |
| 3.1.2.2.4 Sony Ericsson | 56 |
| 3.1.2.2.5 Siemens..... | 57 |
| 3.1.2.2.6 Andere | 57 |
| 3.1.2.3 Betriebssysteme | 58 |

| | | |
|-----------|---|-----|
| 3.1.2.3.1 | Symbian OS | 59 |
| 3.1.2.3.2 | Linux | 60 |
| 3.1.2.3.3 | Palm OS | 61 |
| 3.1.2.3.4 | Windows CE | 62 |
| 3.1.2.3.5 | Alternative Betriebssysteme | 64 |
| 3.1.2.4 | Erkenntnisse der Marktanalyse | 65 |
| 3.1.2.5 | Erhebung zur Nutzung mobiler Kleinrechner | 68 |
| 3.1.2.5.1 | Vorgehensweise | 69 |
| 3.1.2.5.2 | Konzeption des Fragebogens und Durchführung | 71 |
| 3.1.2.5.3 | Ergebnisse | 74 |
| 3.1.3 | Grobkonzept einer mobilen E-Learning Anwendung | 77 |
| 3.1.3.1 | UbiLearn SQL-Inquirer | 79 |
| 3.1.3.2 | XML-Transportcontainer | 80 |
| 3.1.3.3 | UbiLearn mobile Anwendung | 80 |
| 3.1.4 | Auswahl einer geeigneten Zielplattform | 82 |
| 3.1.5 | Auswahl einer geeigneten Entwicklungsumgebung | 87 |
| 3.1.5.1 | Abgeleitete Anforderungen an die Entwicklungsumgebung | 87 |
| 3.1.5.2 | Java | 90 |
| 3.1.5.2.1 | Java 2 Standard Edition und Java 2 Enterprise Edition | 92 |
| 3.1.5.2.2 | Java 2 Micro Edition | 94 |
| 3.1.5.3 | Microsoft eMbedded Visual Tools 3.0 | 99 |
| 3.1.5.3.1 | eMbedded Visual Basic 3.0 | 100 |
| 3.1.5.3.2 | eMbedded Visual C++ 3.0 | 104 |
| 3.1.5.4 | Microsoft eMbedded Visual C++ 4.0 | 105 |
| 3.1.5.5 | Microsoft .NET | 106 |
| 3.1.5.5.1 | .NET Framework | 107 |
| 3.1.5.5.2 | .NET Compact Framework | 110 |
| 3.1.5.6 | Entscheidung für eine Entwicklungsumgebung | 112 |
| 3.1.5.7 | Auswahl einer geeigneten Programmiersprache | 115 |
| 3.2 | Entwurfsphase | 117 |
| 3.2.1 | Grundlagen der objektorientierten Systementwicklung | 118 |
| 3.2.2 | Grundlagen der Unified Modeling Language | 119 |
| 3.2.3 | Systementwurf | 122 |
| 3.2.4 | Pflichtenheft | 126 |
| 3.2.4.1 | Zielbestimmung | 126 |
| 3.2.4.2 | Einsatz | 128 |
| 3.2.4.3 | Umgebung | 128 |
| 3.2.4.4 | Funktionen | 128 |
| 3.2.4.4.1 | Laden eines neuen Lernprojekts | 129 |
| 3.2.4.4.2 | Starten eines neuen Lösungsversuchs | 129 |
| 3.2.4.4.3 | Beantwortung einer Frage | 129 |
| 3.2.4.4.4 | Lösen einer Aufgabe | 130 |
| 3.2.4.4.5 | Zurückstellen einer Aufgabe | 130 |
| 3.2.4.5 | Benutzeroberfläche | 131 |
| 3.2.4.6 | Entwicklungsumgebung | 131 |
| 3.2.5 | Programmentwurf | 133 |
| 3.2.5.1 | Schichtenarchitektur | 133 |
| 3.2.5.2 | Persistenzschicht | 136 |
| 3.2.5.3 | Anwendungsschicht | 137 |
| 3.2.5.4 | Präsentationsschicht | 140 |

| | | |
|------------|--|-----|
| 3.3 | Realisierungsphase | 142 |
| 3.3.1 | Programmierung der UbiLearn mobile Anwendung | 142 |
| 3.3.1.1 | Elemente der Präsentationsschicht | 144 |
| 3.3.1.1.1 | Klasse ubilearnGui | 145 |
| 3.3.1.1.2 | Klasse frmMain | 147 |
| 3.3.1.1.3 | Klasse frmAnswer | 153 |
| 3.3.1.1.4 | Klasse frmAttachment | 158 |
| 3.3.1.1.5 | Klasse frmBrowser | 161 |
| 3.3.1.1.6 | Klasse frmEvaluation | 162 |
| 3.3.1.1.7 | Klasse frmHelpAbout | 164 |
| 3.3.1.1.8 | Klasse frmOpenProject | 166 |
| 3.3.1.1.9 | Klasse frmPictureView | 168 |
| 3.3.1.1.10 | Klasse frmProjectNew | 168 |
| 3.3.1.1.11 | Klasse frmProjectMain | 172 |
| 3.3.1.1.12 | Klasse frmQuestionMultipleChoice | 174 |
| 3.3.1.1.13 | Klasse frmQuestionSingleChoice | 180 |
| 3.3.1.1.14 | Klasse frmQuestionText | 182 |
| 3.3.1.1.15 | Klasse frmSettings | 184 |
| 3.3.1.1.16 | Klasse frmSystemParameter | 188 |
| 3.3.1.1.17 | Klasse frmTrial | 189 |
| 3.3.1.2 | Elemente der Anwendungsschicht | 194 |
| 3.3.1.2.1 | Klasse ubilearnApplication | 195 |
| 3.3.1.2.2 | Klasse ubilearnProject | 196 |
| 3.3.1.2.3 | Klasse ubilearnTrial | 200 |
| 3.3.1.2.4 | Klasse ubilearnExercise | 205 |
| 3.3.1.2.5 | Klasse ubilearnExercisePart | 206 |
| 3.3.1.2.6 | Klasse ubilearnQuestion | 208 |
| 3.3.1.2.7 | Klasse ubilearnQuestionSingleChoice | 209 |
| 3.3.1.2.8 | Klasse ubilearnQuestionMultipleChoice | 210 |
| 3.3.1.2.9 | Klasse ubilearnQuestionText | 210 |
| 3.3.1.2.10 | Klasse ubilearnAnswer | 211 |
| 3.3.1.2.11 | Klasse ubilearnAnswerMultipleChoice | 211 |
| 3.3.1.2.12 | Klasse ubilearnAnswerSingleChoice | 212 |
| 3.3.1.2.13 | Klasse ubilearnAnswerText | 213 |
| 3.3.1.2.14 | Klasse ubilearnAnswerTextItem | 214 |
| 3.3.1.2.15 | Klasse ubilearnGivenAnswer | 214 |
| 3.3.1.2.16 | Klasse ubilearnGivenAnswerText | 214 |
| 3.3.1.2.17 | Klasse ubilearnGivenAnswerChoice | 215 |
| 3.3.1.2.18 | Klasse ubilearnSolvedExercise | 216 |
| 3.3.1.2.19 | Klasse ubilearnSolvedExercisePart | 216 |
| 3.3.1.2.20 | Klasse ubilearnAttachment | 217 |
| 3.3.1.3 | Elemente der Persistenzschicht | 218 |
| 3.3.1.3.1 | Klasse ubilearnFile | 218 |
| 3.3.1.3.2 | Klasse ubilearnProjectFile | 221 |
| 3.3.1.3.3 | Klasse ubilearnQuestionCatalogFile | 221 |
| 3.3.1.3.4 | Klasse ubilearnProjectIO | 221 |
| 3.3.1.3.5 | Klasse ubilearnNetwork | 224 |
| 3.3.1.4 | Querschnittselemente | 225 |
| 3.3.1.4.1 | Klasse settings | 225 |
| 3.3.1.4.2 | Klasse settingshardcoded | 228 |

| | | |
|-----------|--|------------|
| 3.3.1.4.3 | Klasse stringformatter | 228 |
| 3.3.1.4.4 | Klasse encryption | 229 |
| 3.3.1.4.5 | Klasse Xtea | 230 |
| 3.3.1.4.6 | Klasse ubilearnException | 231 |
| 3.3.1.4.7 | OpenNETCF Smart Device Framework..... | 231 |
| 3.3.1.5 | Die Konfigurationsdatei appSettings.xml..... | 235 |
| 3.3.2 | Aufbau des UbiLearn SQL-Inquirer | 236 |
| 3.3.2.1 | Datenbankzugriff..... | 237 |
| 3.3.2.2 | Auswahl eines Lernprojekts | 239 |
| 3.3.2.3 | Speichern eines Lernprojekts..... | 242 |
| 3.3.3 | Aufbau des XML-Transportcontainers..... | 244 |
| 3.3.3.1 | Datenstrukturbeschreibung mit XML-Schemas | 245 |
| 3.3.3.2 | Aspekte der Verschlüsselung | 246 |
| 3.3.3.3 | Struktur der XML-Nutzdaten | 247 |
| 3.3.4 | Programmtest..... | 250 |
| 3.3.4.1 | Festlegung des Testszenarios | 251 |
| 3.3.4.2 | UbiLearn SQL-Inquirer | 252 |
| 3.3.4.2.1 | Installation | 252 |
| 3.3.4.2.2 | Programmstart und Extraktion eines Lernprojekts..... | 253 |
| 3.3.4.3 | UbiLearn mobile Anwendung | 254 |
| 3.3.4.3.1 | Installation | 254 |
| 3.3.4.3.2 | Programmstart und Konfiguration..... | 256 |
| 3.3.4.3.3 | Laden und Bearbeiten eines Lernprojekts | 257 |
| 3.3.4.4 | Fazit des Programmtests und Feststellung der Problemfelder..... | 262 |
| 4 | Fazit und Ausblick..... | 267 |
| 4.1 | Beurteilung | 267 |
| 4.2 | Entwicklungsperspektiven..... | 269 |
| 4.3 | M-Learning Trends..... | 273 |
| | Literaturverzeichnis | 275 |
| | Anhang..... | 296 |
| A. | Struktogramme | 297 |
| B. | Datenmodell der UbiLearn mobile Anwendung..... | 302 |
| C. | Vollständiges XSD-Schema des Datenmodells | 305 |
| D. | Inhalt der CD zur Arbeit | 310 |
| E. | Empfehlung von Quellen | 313 |
| F. | Installation der UbiLearn mobile Anwendung auf einem Pocket PC Emulator .. | 315 |
| G. | Pocket PC PDA Kompatibilitätstabelle | 322 |
| H. | Palm OS PDA Kompatibilitätstabelle | 323 |
| I. | Symbian OS Kompatibilitätstabelle | 324 |
| J. | Lizenzbedingungen NSICOM CrEme | 325 |
| K. | Stellungnahme der Firma Esmertec zur Lizenzierung der Jeode JVM | 328 |
| L. | Ehrenwörtliche Erklärung | 329 |

1 Einleitung

1.1 Motivation

Über Jahrhunderte wurde die Gesellschaft in den nördlichen Ländern der Erde durch die Landwirtschaft geprägt, die sich Ende des 19. Jahrhunderts zur Industriegesellschaft wandelte. Ab diesem Zeitpunkt entwickelte sich die verarbeitende Industrie zum dominierenden Beschäftigungssektor in diesen Ländern¹.

Die Industriegesellschaft ist nach Gutenberg durch Produktionsbetriebe gekennzeichnet. Kern jeder Produktion ist die Kombination der drei Elementarfaktoren *menschliche Arbeitsleistung*, *Betriebsmittel* und *Werkstoffe*². Durch die fortschreitende Entwicklung der Informations- und Kommunikationssysteme gewinnen Informationen und Wissen einen immer höheren Stellenwert in unserer Gesellschaft³. Dies hat dazu geführt, dass heute von einer Informations- und Wissensgesellschaft gesprochen wird, mit Informationen und Wissen als den zentralen Produktionsfaktoren⁴.

Mit dem Entstehen der Wissensgesellschaft ergeben sich neue Anforderungen an das Aus- und Weiterbildungssystem. In der heutigen Gesellschaft wird es für die Gesellschaftsmitglieder immer wichtiger, Entscheidungen auf der Grundlage fundierten Wissens zu treffen und die Bereitschaft zu lebenslangem Lernen mitzubringen. Hinzu kommen der Umgang mit moderner Informationstechnologie, großen Informationsmengen, Kommunikationsfähigkeit und die Befähigung zum selbstgesteuerten Lernen. Das Aus- und Weiterbildungssystem seinerseits ist verantwortlich für die Schaffung der notwendigen Voraussetzungen, um diese Fähigkeiten zu erlangen.

Von Schulen und Hochschulen wird verlangt, die Fähigkeit der Gesellschaftsmitglieder zum Wissensmanagement zu fördern, damit diese Informationen nach Inhalt, Bedeutung und Nutzen selektieren, bewerten und daraus Wissen konstruieren können. Diese Fähigkeit wird umso wichtiger, da das verfügbare Wissen immer schneller wächst. Zurzeit wird von einer Verdopplung des verfügbaren Wissens alle fünf Jahre ausgegangen. Auf

¹ Vgl. GOTTWALD/SPRINKART 1998, S. 21.

² Vgl. GUTENBERG 1979, S. 3ff.

³ Vgl. SPINNER 2001, S. 319.

⁴ Um die Bedeutung von *Informationen* und *Wissen* als Produktionsfaktoren zu erklären, ist es notwendig, diese Begriffe abzugrenzen. Dazu können diese in eine Begriffshierarchie eingeordnet werden, auf deren unterster Ebene sich die *Zeichen* befinden. Wird den einzelnen Zeichen eine Bedeutung zugewiesen, so entstehen *Daten*. Sobald Daten zueinander in Beziehung gesetzt werden und der Zielerreichung dienen, handelt es sich um *Informationen*. Durch Bewertung, Einordnung in Zusammenhänge und Einbringung von Erfahrung wird aus *Informationen* *Wissen*. Vgl. TSICHRITZIS 1995, S. 106ff.

der Kehrseite dieses Prozesses steht der rasante Aktualitätsverlust des vorhandenen Wissens. Daher kommt dem Prozess des Wissenserwerbs durch Lehren und Lernen eine immer höhere Bedeutung zu⁵.

Parallel zur Bedeutungszunahme des Wissenserwerbs durch Lehren und Lernen hat sich im Laufe des letzten Jahrhunderts das Umfeld, in dem Hochschulen operieren, dramatisch verändert. Zu Beginn des 20. Jahrhunderts war die Hochschulausbildung das Privileg einer Minderheit und die Universitäten wurden stark durch den Staat beeinflusst. Seit einigen Jahren kommt es im Rahmen von umfassenden Reformvorhaben jedoch zu einem Rückzug des Staates. Die Hochschulen erhalten mehr Autonomie und es wird nach einer unternehmensorientierten Form der Hochschule verlangt, die ähnlich den amerikanischen Hochschulen einem freien Wettbewerb innerhalb eines belebten Marktes ausgesetzt sind. Trotz eingefrorener Budgets müssen sich Hochschulen immer mehr durch ihre Lehrleistung definieren⁶. Eine Abhilfe aus diesem Dilemma verspricht die Unterstützung des Lehrauftrags, indem durch den systematischen Einsatz moderner E-Learning Anwendungen das interaktive Selbstlernen ermöglicht wird. Interaktionen sind dabei ein qualitätsprägendes Merkmal des Lernprozesses, da diese den Lernenden mobilisieren und motivieren, das inhaltliche Verständnis fördern und Indikatoren für den Wissensstand beinhalten, der in Lernerfolgskontrollen überprüft werden kann⁷.

Die Gestalt von E-Learning Anwendungen unterliegt einem ständigen Weiterentwicklungsprozess. Eine erste Form dieser Anwendungen stellt das *Computer Based Training* (CBT) dar. In diesem Lernsystem werden dem Lernenden i. d. R. auf herkömmlichen PC-Systemen multimediale Lerninhalte vermittelt. Mit dem Aufkommen des *World Wide Webs* (WWW) erfolgte eine Weiterentwicklung zum *Web Based Training* (WBT). Diese Systeme werden nicht wie CBT-Systeme auf Datenträgern, sondern über das Internet bereitgestellt und meist durch herkömmliche Internetbrowser ausgeführt⁸.

Mit dem Aufkommen der „Mobilen Revolution“⁹, einer rasant steigenden Verbreitung von Mobiltelefonen und PDAs (Personal Digital Assistants) in der Bevölkerung und der stetigen Leistungssteigerung dieser Geräte gewinnt nun eine weitere Form des E-Learning eine dramatisch steigende Bedeutung: *Mobiles E-Learning* (*M-Learning*) ermöglicht dem Lernenden, ortsungebunden und zu jeder Zeit zu lernen. Voraussetzung

⁵ Vgl. ARNOLD 2004, S. 38-40.

⁶ Vgl. SIMON 2001, S. 19.

⁷ Vgl. GABELE/ZÜRN 1993, S. 227.

⁸ Vgl. SEUFERT/MAYR 2002, S. 25.

⁹ KEEGAN 2002.

hierfür ist eine an die spezifischen Anforderungen mobiler Endgeräte angepasste E-Learning Anwendung.

1.2 Zielsetzung und Vorgehensweise

Am Institut für Wirtschaftsinformatik der Universität Hannover beschäftigt sich seit Herbst 2002 in einem praktisch orientierten Forschungsbereich eine Projektgruppe unter der Leitung von Prof. Dr. M. H. Breitner mit der Entwicklung des *UbiLearn E-Learning Systems*. Ein wichtiges Ziel des Systems ist die Ubiquität (Allgegenwärtigkeit) des E-Learning, das durch eine Zusammenführung der bisher isolierten Anwendungsbereiche CBT, WBT und MBT¹⁰ erreicht werden soll¹¹. Im bisherigen Entwicklungsstatus besteht das UbiLearn System aus einem Java-basierten CBT-Modul und einem PHP-basierten WBT-Modul.

Zielsetzung dieser Arbeit ist, eine E-Learning Anwendung zu entwickeln, die zum einen auf mobilen Rechnern die Möglichkeit bietet, interaktiv und ortsungebunden zu lernen und sich zum anderen nahtlos als MBT-Modul in die Infrastruktur des vorhandenen UbiLearn Systems einfügt. Dabei kann das MBT-Modul entweder als mobiles WBT-Modul realisiert werden, das netzwerkgestützt und permanent online die Inhalte aus dem UbiLearn System abrufen und diese auf dem mobilen Endgerät darstellt, oder als mobiles CBT-Modul, welches ohne eine Netzwerkverbindung auskommt und die dafür notwendigen Daten lokal speichert. Heutzutage stellen Geräte, die mit einer drahtlosen Netzwerkschnittstelle oder einer Anschlusskomponente zum Mobilfunknetz ausgestattet sind, noch die Minderheit dar. Weitere Aspekte stellen die Verfügbarkeit und Kosten drahtloser Netzwerke dar. Mobilfunknetze bieten zwar eine gute Flächenabdeckung, sind in ihrer Nutzung aber bisher mit hohen Kosten verbunden. Die alternative Technologie Wireless-LAN ist in der Nutzung zwar kostengünstiger, bietet jedoch nur eine unzureichende Flächenabdeckung¹². Aus diesen Gründen wurde durch die Projektgruppe bereits im Vorfeld dieser Arbeit die Entscheidung getroffen, das MBT-Modul als mobiles CBT-Modul zu entwickeln.

Der eigentlichen Anwendungsentwicklung geht die Konzeption voraus. Der Begriff Konzeption wird in diesem Zusammenhang als Grobaufbau einer komponentenbasierten Anwendung sowie deren Beziehungen und jeweiligen Aufgaben untereinander verstanden. Im Rahmen der Entwicklung soll die konzipierte Anwendung auf einer Entwick-

¹⁰ Mobile Based Training

¹¹ Vgl. IWI 2004.

¹² Vgl. Abschnitt 2.5.4 – S. 31.

lungsumgebung unter Auswahl einer Programmiersprache realisiert und getestet werden. In Kapitel 2 wird zunächst eine Einführung in das E-Learning vorgenommen, um im Kapitel 3 die eigentliche Systementwicklung der Anwendung durchzuführen.

Da es sich bei der Systementwicklung der mobilen, interaktiven E-Learning Anwendung, die in dieser Arbeit **UbiLearn mobile** genannt wird, letztendlich um die Entwicklung eines Anwendungssystems handelt, wird die Systementwicklung von zwei Fragen bestimmt¹³:

- ▶ *Welche Aktivitäten sind in welcher Reihenfolge auszuführen?*
- ▶ *Wer führt wann welche Aktivitäten aus?*

Während für die Beantwortung der ersten Frage die Auswahl eines *Vorgehensmodells der Systementwicklung* notwendig ist, fällt die Beantwortung der zweiten Frage in den Aufgabenbereich des *Projektmanagements*.

Da sich fast alle Vorgehensmodelle der Systementwicklung nach der *Projektbegründung*, die sich aus der Zielsetzung dieser Arbeit ergibt, an den vier Phasen

- 1.) *Analyse*
- 2.) *Entwurf*
- 3.) *Realisierung*
- 4.) *Einführung*

orientieren, liegt die Einteilung des dritten Kapitels diesen Phasen zugrunde. Abbildung 1 stellt ein solches Vorgehensmodell grafisch dar.

¹³ Vgl. STAHLKNECHT/HASENKAMP 2002, S. 212-218.

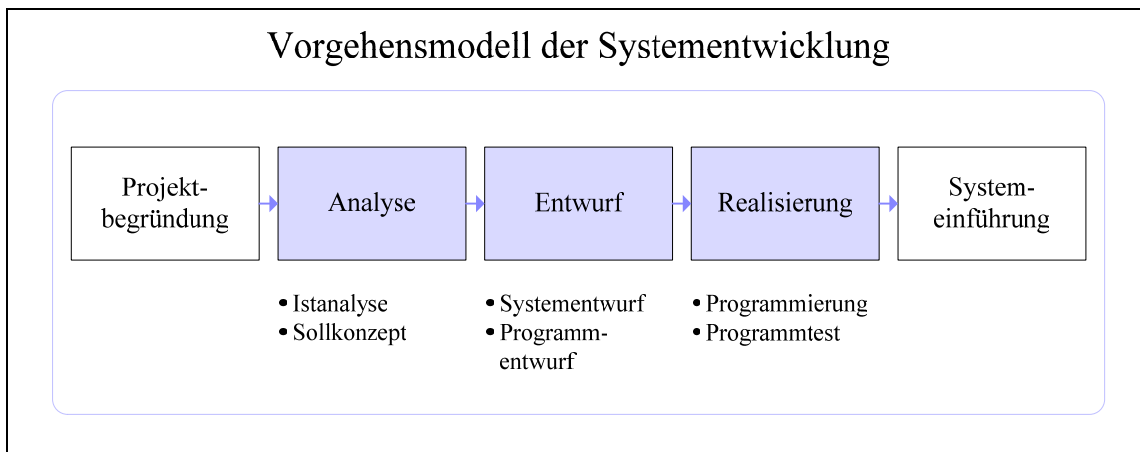


Abbildung 1: Vorgehensmodell der Systementwicklung
(In Anlehnung an: STAHLKNECHT/HASENKAMP 2002, S. 221).

In der Literatur werden eine Vielzahl modifizierter Modelle aufgeführt, die auf diesem Grundmodell aufbauen. Zwei Beispiele hierfür sind das *Wasserfallmodell* und das *Spiralmodell*.

Das Wasserfallmodell modifiziert das Grundmodell, indem die ursprünglichen vier Phasen weiter untergliedert werden und ein strikt sequentieller Phasenfortschritt vorgeschrieben wird, d. h. ein Rücksprung zur vorherigen Phase ist nur dann erlaubt, wenn sich deren Ergebnisse als fehlerhaft erweisen und zu korrigieren sind¹⁴. Die Feststellung, ob die gewünschten Ergebnisse richtig oder falsch sind (Validierung), erfolgt in diesem Modell möglichst experimentell anhand eines Prototypen der Anwendung. Generell eignet sich das Wasserfallmodell, das den Charakter eines *strengen Phasenmodells* hat, vor allem für „wohlstrukturierte“ Probleme, d. h. Anwendungssysteme mit eindeutiger Anforderungsdefinition¹⁵.

Das Spiralmodell ist eine Verfeinerung des Wasserfallmodells und berücksichtigt, dass es sich bei der Entwicklung von Informationssystemen um einen evolutionären Prozess handelt. Charakteristisches Merkmal dieses Modells ist, dass der Gesamtaufwand und der Projektfortschritt in Spiralzyklen dargestellt werden. Außerdem ist in diesem Modell vorgesehen, mehrere Alternativen zur Realisierung des Produkts bspw. in Form verschiedener Prototypen zu erstellen und diese Alternativen hinsichtlich der Projektziele und Nebenbedingungen zu beurteilen¹⁶. Das Spiralmodell ist ein Zyklenmodell und

¹⁴ Vgl. HANSEN/NEUMANN 2001, S. 207-209.

¹⁵ Vgl. STAHLKNECHT/HASENKAMP 2002, S. 223.

¹⁶ Vgl. HANSEN/NEUMANN 2001, S. 210-211.

eignet sich besonders für „schlecht strukturierte“ Probleme mit von vornherein uneindeutiger Anforderungsdefinition¹⁷.

Nach Ansicht des Autors eignet sich das Spiralmodell nicht als Vorgehensmodell für die Entwicklung der UbiLearn mobile Anwendung, da es sich in diesem Fall weniger um ein „schlecht strukturiertes“, sondern vielmehr um ein „wohlstrukturiertes“ Problem handelt. In die engere Betrachtung kommt das Wasserfallmodell. Nachteilig ist jedoch, dass einige Phasen des Modells (vor allem der Entwurf des Datenmodells) nicht benötigt werden. Die Begründung für das Verbot von Rückschritten über mehr als eine Projektphase liegt darin, dass solche Iterationen zu hohen Nachbearbeitungskosten führen. Nach Ansicht des Autors betrifft diese Begründung jedoch vor allem solche Projekte, an denen viele Projektmitglieder gleichzeitig beteiligt sind. Daher erfolgt aus diesen Gründen die Systementwicklung anhand des „klassischen“ Vorgehensmodells in den folgenden Phasen¹⁸:

1. Phase - Analyse: Die Analysephase beginnt zunächst mit einer *Istanalyse*, in der mit einer detaillierten Analyse des Umfeldes die Voraussetzungen für die Aufstellung eines *Sollkonzepts* geschaffen werden. Mit dem Sollkonzept soll in einem *Grobkonzept* festgelegt werden, *was* das Anwendungssystem leisten und *wie* das Anwendungssystem realisiert werden soll.

Im Falle der UbiLearn mobile Anwendung besteht das Umfeld aus mehreren Aspekten. Zunächst benötigt die Anwendung mobiles Endgerät als Hard- und Softwareplattform, auf dem sie ausgeführt werden kann. Des Weiteren soll die Anwendung Bestandteil des UbiLearn Systems sein, das bereits am Institut für Wirtschaftsinformatik in Betrieb ist.

Im Falle der Hardwareplattform stellt sich wiederum ein zweiteiliges Problem: Der Markt mobiler Endgeräte unterscheidet sich deutlich von der heutigen Situation bei Desktop PCs. Während aktuelle Desktop PCs bezüglich der Leistungsfähigkeit der Hardware sehr *homogen* sind, stellt sich die Situation bei den mobilen Endgeräten als eher *heterogen* dar. Daher ist es notwendig zu analysieren, in welchem Ausmaß die mo-

¹⁷ Vgl. STAHLKNECHT/HASENKAMP 2002, S. 223.

¹⁸ Als Erweiterung des „klassischen“ Vorgehensmodells wird jedoch während der Entwicklung ein Prototyp entwickelt, mit dem experimentiert wird. Nach STAHLKNECHT/HASENKAMP 2002 kann hinsichtlich der Verwendung zwischen zwei Arten von Prototypen unterschieden werden: Einem Wegwerfprototypen, der lediglich zur Sammlung von Erfahrungen benutzt wird, anhand derer das endgültige System neu erstellt wird (rapid prototyping) und einem wieder verwendbaren Prototypen, der schrittweise verbessert wird und in das endgültige Anwendungssystem einfließt (evolutionäres Prototyping). Im Falle dieser Arbeit wird das Konzept des evolutionären Prototyping vor allem in der Realisierungsphase angewandt.

bilen Endgeräte in Bezug auf die Leistungsfähigkeit der Hardware noch als heterogen bezeichnet werden können und welche spezifischen Eigenschaften der Hardware Einfluss auf den Prozess der Systementwicklung haben könnten.

Ebenfalls stellt sich das Problem der *Softwareplattform*, d. h. dem verwendeten Betriebssystem. Im Falle der Desktop PCs existiert heute eine „Monokultur“ von Microsoft Windows in verschiedenen Versionen. Ein Vorteil dieser Monokultur ist, dass die aktuellen Versionen von Windows untereinander fast durchgängig kompatibel sind und es bei der Softwareentwicklung unerheblich ist, welche Version von Windows auf dem Zielsystem vorhanden ist. Existiert als Anforderung der Systementwicklung die *Plattformunabhängigkeit*, d. h. soll das System auch auf weiteren Softwareplattformen lauffähig sein, so gibt es für Desktop PCs unterschiedliche Lösungsansätze, diese Plattformunabhängigkeit zu erreichen. Bspw. kann die Entwicklungsumgebung Java eingesetzt werden, die mit einer kostenlos verfügbaren Java Runtime Engine als Zwischenschicht das Ausführen der gleichen Anwendung auf unterschiedlichen Betriebssystemen ermöglicht.

Es ist notwendig herauszufinden, welche Betriebssysteme für mobile Endgeräte existieren und zu untersuchen, wie deren Marktanteile gewichtet sind.

Damit sich die UbiLearn mobile Anwendung in das UbiLearn System am Institut für Wirtschaftsinformatik integrieren lässt, sind die spezifischen Eigenschaften des vorhandenen UbiLearn Systems genau zu analysieren, da der nachfolgende Entwicklungsprozess auf den gegebenen Voraussetzungen aufbauen wird.

Die Aufstellung eines *Sollkonzepts* basiert auf den Ergebnissen der Istanalyse und hat das Ziel, die *Anforderungen* an das geplante System festzulegen. Insbesondere sollen die folgenden zwei Fragen beantwortet werden¹⁹:

- ▶ *Was die Anwendung leisten soll* und
- ▶ *Wie die Anwendung realisiert werden soll.*

Die Beantwortung der ersten Frage wird in STAHLKNECHT/HASENKAMP 2002 als *Fachentwurf* bezeichnet; die Beantwortung der zweiten Frage als *IV-Grobentwurf*. Es wird von anderen Autoren teilweise die Auffassung vertreten, dass die Aufstellung des Sollkonzepts sich ausschließlich auf den Fachentwurf, d. h. die Anforderungen an das geplante System beschränken soll. Da sich jedoch herausstellen wird, dass die zur Rea-

¹⁹ Vgl. STAHLKNECHT/HASENKAMP 2002, S. 247-251.

lisierung notwendige Auswahl einer Entwicklungsumgebung mit ihrer spezifischen Programmiersprache auch eine Kostenfrage sein wird, soll in dieser Arbeit der Auffassung von STAHLKNECHT/HASENKAMP 2002 gefolgt werden, dass der IV-Grobentwurf Bestandteil des Sollkonzepts ist.

In diesem Zusammenhang wird sich herausstellen, dass es keine „ideale“ Entwicklungsumgebung für die anschließende Realisierung gibt. Die Auswahl einer geeigneten Entwicklungsumgebung wird vielmehr ein Kompromiss innerhalb der Erfüllung der verschiedenen Anforderungen bleiben.

2. Phase - Entwurf: Die Entwurfsphase soll die Voraussetzungen für die nachfolgende Realisierungsphase schaffen, in der die eigentliche Programmierung erfolgt. Diese Phase verläuft in den folgenden Schritten²⁰:

- ▶ *Schritt 1:* Entwicklung eines Systementwurfs, wobei eine Entscheidung getroffen werden muss, ob dieser Systementwurf objektorientiert oder strukturiert erfolgen soll.
- ▶ *Schritt 2:* Zusammenstellung einer Programmspezifikation, d. h. detaillierte Vorgaben in Form eines Pflichtenhefts.
- ▶ *Schritt 3:* Erarbeitung eines systematischen und möglichst strukturierten Programmentwurfs anhand der Programmspezifikation.

Der Systementwurf der UbiLearn mobile Anwendung wird in dieser Arbeit aus mehreren Gründen objektorientiert erfolgen. Zum einen ist ein strukturierter Systementwurf mit einer Reihe von Problemen behaftet. Ein besonders schwerwiegender Grund ist die Trennung von Daten und Funktionen, die eine modellhafte Darstellung erschwert. Werden Daten bspw. durch ein Entity-Relationship Modell erfasst und die Funktionen durch Nassi-Shneider-Diagramme, bereitet die Abstimmung zwischen den beiden Modellen Schwierigkeiten²¹. Des weiteren unterstützen die in der Praxis verfügbaren Entwicklungsumgebungen quasi nur noch eine objektorientierte Programmierung, so dass die Festlegung auf einen objektorientierten Systementwurf Vorteile für die Umsetzung in der nachfolgenden Realisierungsphase bietet.

Im Rahmen der Zusammenstellung einer Programmspezifikation (Schritt 2) wird sich herausstellen, dass zum einen eine objektorientierte Modellierung des Umfeldes, d. h. dem gesamten UbiLearn System, und zum anderen eine objektorientierte Modellierung

²⁰ Vgl. STAHLKNECHT/HASENKAMP 2002, S. 258-260.

²¹ Vgl. STAHLKNECHT/HASENKAMP 2002, S. 277.

der eigentlichen UbiLearn mobile Anwendung erforderlich ist. Es wird sich weiterhin herausstellen, dass neben der Entwicklung der UbiLearn mobile Anwendung zusätzlich die Entwicklung eines Hilfsprogramms, das im folgenden **UbiLearn SQL-Inquirer** genannt wird, notwendig ist. Dieses Hilfsprogramm extrahiert aus der gemeinsamen Datenbasis des UbiLearn Systems die gewünschten Daten und „verpackt“ diese in XML-Dateien²², die von der UbiLearn mobile Anwendung verarbeitet werden können.

Der eigentliche Programmentwurf (Schritt 3) enthält die Vorgaben für die UbiLearn mobile Anwendung und das Hilfsprogramm UbiLearn SQL-Inquirer als Grundlage für die nachfolgende Realisierungsphase.

3. Phase – Realisierung: Die Realisierungsphase, die auch *Implementierung* genannt wird, beinhaltet die Vorgänge

- ▶ Programmierung der Anwendung, und
- ▶ einen anschließenden Programm- und Systemtest²³.

Die eigentliche Programmierung erfolgt auf der zuvor im Schritt *Analyse* ausgewählten Entwicklungsumgebung und deren Programmiersprache. Sowohl die UbiLearn mobile Anwendung als auch das Hilfsprogramm UbiLearn SQL-Inquirer enthalten verschiedene Klassen, die in diesem Abschnitt beschrieben werden. Damit die Programme besser verständlich und wartbar sind, fordert STAHLKNECHT/HASENKAMP 2002 eine ausführliche Selbstdokumentation (*Inlinedokumentation*). Diese besteht aus speziellen Kommentarzeilen innerhalb des Quelltextes, die beim Kompilieren ignoriert werden, aber die Lesbarkeit des Quelltextes verbessern. Daher wurde der Quelltext der Programme mit diesen speziellen Kommentarzeilen versehen. Aus der Inlinedokumentation lässt sich über spezielle Zusatzprogramme eine gesonderte Programmdokumentation generieren. Diese Programmdokumentation ist auf der Begleit-CD enthalten, die dieser Arbeit beiliegt.

Der anschließende Programmtest soll prüfen, ob die UbiLearn mobile Anwendung die gestellten Aufgaben zufrieden stellend löst und ob das Programm korrekt und stabil ausgeführt wird. Während der Erstellung dieser Arbeit stehen dem Autor als Testplattform ein PDA (Hewlett-Packard, iPAQ h4150) zur Verfügung. Zusätzlich wird die U-

²² eXtensible Markup Language ist eine Datenmodellierungssprache, die nicht nur maschinenlesbar ist, sondern auch von Menschen gelesen werden kann. Vgl. zur Vertiefung auch MÜNZ 2001 und Abschnitt 3.3.3 – S. 244.

²³ Vgl. STAHLKNECHT/HASENKAMP 2002, S. 288.

biLearn mobile Anwendung auf dem institutseigenen PDA am Institut für Wirtschaftsinformatik getestet (Hewlett-Packard, iPAQ h2210).

Als Testaufgaben werden die bereits im UbiLearn System vorhandenen Lernprojekte herangezogen.

4. Phase Einführung: Die UbiLearn mobile Anwendung ist eine Eigenentwicklung, daher erfolgt die Systemeinführung nach der *förmlichen Systemfreigabe* und erfolgreich abgeschlossenem *Abnahmetest*²⁴. Die Verantwortung dafür liegt beim Leiter des IT-Projekts, im Fall dieser Arbeit bei Herrn Prof. Dr. M. H. Breitner, und kann erst nach Abschluss dieser Diplomarbeit erfolgen. Aus diesem Grund wird diese Phase in der Ausarbeitung nicht weiter berücksichtigt.

Abschließende Beurteilung und Ausblick – Am Ende der Arbeit erfolgt eine Zusammenfassung der Ergebnisse des Programmtests, sowie der Erfahrungen während der Konzeption und Entwicklung der UbiLearn mobile Anwendung. Ferner wird ein Ausblick auf eine mögliche weitere Entwicklung der UbiLearn mobile Anwendung und des M-Learning unter Berücksichtigung dieser Ergebnisse vorgenommen.

1.3 Softwareentwicklung mit CASE

Die Bearbeitung der Phasen der Softwareentwicklung kann durch den Computer unterstützt werden. Solche Hilfsmittel werden in der Literatur allgemein unter den Begriffen Software-Werkzeuge oder *CASE-Tools* (computer aided software engineering tools) zusammengefasst.

CASE-Tools für die frühen Phasen der Softwareentwicklung, also von der Analyse bis zum Entwurf, werden genauer *Upper CASE-Tools* bezeichnet. Software-Werkzeuge für die späteren Phasen, d. h. Realisierung und Test, werden mit *Lower CASE-Tools* bezeichnet. CASE-Tools können eine Entwicklungsphase abdecken, es können aber auch mehrere oder alle Entwicklungsphasen von einem CASE-Tool abgedeckt werden. In diesem Fall bezeichnet man das Software-Werkzeug als integriertes CASE-Tool oder kurz *I-CASE*. Im Zusammenhang mit CASE-Tools werden häufig die Begriffe CARE-Tool und CAME-Tool gebraucht. Ein *CARE-Tool* (computer aided reengineering tool) wird vor allem für den Prozess der Softwarewartung benötigt, falls aus gegebenen Programmcode der zugehörige Entwurf oder die zugehörige Dokumentation abgeleitet werden muss. Ein *CAME-Tool* (computer-assisted measurement and evaluation tool)

²⁴ Vgl. STAHLKNECHT/HASENKAMP 2002, S. 321ff.

hilft bei der ingenieurmäßigen Softwareentwicklung als Analyse-, Mess- oder Bewertungstool.

Es existieren vielfältige Arten und Erscheinungsformen von CASE-Tools. Ebenfalls hängt es von der Art der Verwendung ab, ob eine Software ein CASE-Tool ist oder nicht. In der ersten *Analysephase* der Systementwicklung in Abschnitt 3.1 werden unterschiedliche Fakten und Abbildungen des Marktes mobiler Endgeräte gesammelt und in Skizzen bzw. Diagrammen zusammengeführt. In diesem Fall kann ein *Textverarbeitungssystem* wie bspw. Microsoft Word 2003 als CASE-Tool verwendet werden²⁵.

Innerhalb der *Entwurfsphase* in Abschnitt 3.2 wird eine grafische Modellierung des Systementwurfs in UML-Modellen der Typen Use-Case, Klassendiagramm und Sequenzdiagramm vorgenommen. Hierzu sind entsprechende grafische *Modellierungs-Tools* erforderlich²⁶. Im Rahmen dieser Arbeit wird dazu Microsoft Visio 2003 als CASE-Tool benutzt.

Im Rahmen der *Realisierungsphase* in Abschnitt 3.3 erfolgt die eigentliche Programmierung und die Erstellung einer Quelltextdokumentation. Für die Programmierung wird als grafische *Software-Entwicklungsumgebung* Microsoft Visual Studio .NET 2003 Professional und für die Erstellung der Quelltextdokumentation das Programm NDoc als CASE-Tool benutzt. Die grafische Software-Entwicklungsumgebung unterstützt zusätzlich teilweise den anschließenden Programmtest durch eine integrierte Debuggingfunktion.

²⁵ Vgl. DUMKE 2001, S. 131ff.

²⁶ Eine umfangreiche Übersicht aktueller Modellierungs-Tools bietet <http://www.jeckle.de/umltools.html>.

4 Fazit und Ausblick

4.1 Beurteilung

Ziel dieses Abschnitts ist, sowohl den Prozess der Systementwicklung als auch den aktuellen Stand der UbiLearn mobile Anwendung unter Berücksichtigung der Erkenntnisse aus Abschnitt 3.3.4.4 zusammenfassend zu beurteilen.

Für den Betrieb einer mobilen E-Learning Anwendung werden mobile Endgeräte benötigt. Auch wenn Notebooks prinzipiell auch transportfähig sind, bieten PDAs und Smartphones in Bezug auf Akkulaufleistung, Größe und Gewicht bessere Mobilitätsmöglichkeiten.

Der Prozess der Systementwicklung deutlich gemacht, dass die heterogenen Hardware- und Softwareumgebungen verschiedenartiger mobilen Endgeräte eine besondere Herausforderung für den Entwicklungsprozess darstellen und dass keine verfügbare Entwicklungsumgebung sämtliche Anforderungen, die in Abschnitt 3.1.5.1 formuliert wurden, optimal erfüllt.

Eine plattformunabhängige Realisierung, die uneingeschränkt auf sämtlichen verfügbaren mobilen Endgeräten lauffähig ist, kann wegen verschiedenartiger Betriebssysteme und Hardwarevoraussetzungen nicht vorgenommen werden. Stattdessen ist eine Entscheidung für eine bestimmte Zielplattform erforderlich, die wiederum die Wahl der Entwicklungsumgebung beeinflusst. Als Zielplattform kommen zunächst PDAs oder Smartphones in Betracht. Da PDAs derzeit die besseren Hardwarevoraussetzungen bieten und diese Geräte unter Studenten der wirtschaftswissenschaftlichen Fakultät der Universität Hannover weiter verbreitet sind, wurde im Rahmen dieser Arbeit die Geräteklasse der PDAs als Zielplattform ausgewählt. Innerhalb dieser Geräteklasse teilen sich einerseits die Pocket PCs und andererseits die Palm Handhelds den Markt. Da Pocket PCs bei den Neuverkäufen bereits heute den Markt dominieren und auch für die Zukunft eine bessere Wachstumsprognose als Palm Handhelds haben, wurden diese Geräte für den Prozess der Systementwicklung herangezogen.

Für diese Geräte stehen zwei mögliche Entwicklungsumgebungen zur Verfügung: Die Java 2 Micro Edition von Sun und das .NET Compact Framework von Microsoft. Obwohl der Titel der Arbeit bereits im Vorfeld die Entscheidung für .NET implizierte, hat die Analyse der alternativen Entwicklungsumgebungen gezeigt, dass für den Pocket PC das .NET Compact Framework derzeit als geeignetste Entwicklungsumgebung angese-

hen werden kann. Wegen dessen Bindung an Windows CE als Betriebssystem ist die Anwendung jedoch nicht auf Geräten mit Symbian OS oder PalmOS lauffähig. Da Palm OS jedoch für die Zukunft eine abnehmende Marktbedeutung prognostiziert wird, ist es wahrscheinlich, dass diese Einschränkung bald an Relevanz verlieren wird. Deutlich gravierender für eine spätere Portierung auf Smartphones ist jedoch der Umstand, dass die E-Learning Anwendung nicht mit Symbian OS lauffähig ist. Dieser Umstand gewinnt hinsichtlich von zwei Aspekten eine hohe Relevanz. Zum einen hat Symbian OS in der Geräteklasse der Smartphones bereits heute die Marktführerschaft. Das auf Windows CE .NET basierende Windows Mobile for Smartphones hat in dieser Geräteklasse nur einen unbedeutenden Marktanteil. Da Nokia als heutiger Weltmarktführer der Mobiltelefonproduktion maßgeblich am Symbian-Jointventure beteiligt ist, wird dieses Betriebssystem ständig weiterentwickelt mit dem Ziel, eine Marktdominanz von Windows CE zu verhindern. Jedoch hat Microsoft in der Vergangenheit auf anderen Märkten mehrfach ihre Fähigkeit unter Beweis gestellt hat, auch hohe Marktanteilsrückstände in kurzen Zeiträumen aufholen zu können. Daher ist es zum heutigen Zeitpunkt völlig ungewiss, welches Betriebssystem in Zukunft die Smartphones dominieren wird. Sehr wahrscheinlich ist jedoch, dass die Marktbedeutung des klassischen PDA in der Zukunft vom Smartphone abgelöst werden wird. Aus diesen Gründen werden Entwickler mobiler Softwareanwendungen noch für einen längeren Zeitraum mit verschiedenen, zueinander inkompatiblen Betriebssystemen konfrontiert werden.

Bezogen auf die unterstützte Zielplattform erfüllt die UbiLearn mobile Anwendung alle Musskriterien und viele derjenigen Wunschkriterien, die in Abschnitt 3.2.4.1 formuliert wurden. Im Gegensatz zu anderen offline M-Learning Anwendungen, die in Abschnitt 2.5.3 erläutert wurden, ermöglicht die Konzeption als eigenständig programmierte Anwendung eine Interaktion mit dem Lernenden. Im Rahmen der Beantwortung von Übungsaufgaben fordert die Anwendung den Lernenden zu Handlungen auf, deren Ergebnisse automatisch und individuell analysiert und ausgewertet werden. Daher ist die Anwendung in der Lage, aus Sicht des Lernenden die Rolle des Lehrers einzunehmen. Neben der reinen Wissensabfrage in Form von Übungsaufgaben ermöglicht die Einbindung multimedialer Attachments auch weitergehende Formen des Lernens. So ist es bspw. vorstellbar, komplette Vorlesungen als Video oder Tonaufzeichnung samt verbundener Schaubilder oder Folien als Attachment zu Beginn eines Lernprojekts abzuspielen. Die anschließenden Übungsaufgaben sollten sich dann auf die aufgezeichnete

Vorlesung beziehen. Somit rückt die Vision einer komplett virtuellen Lehrveranstaltung in greifbare Nähe.

Im derzeitigen Entwicklungsstand eignet sich die UbiLearn mobile Anwendung bereits für einen Systemeinsatz zu Probezwecken unter Studenten an der Universität Hannover im Fachbereich Wirtschaftswissenschaften. Im Rahmen des Programmtest konnte auf den beiden getesteten Geräten Hewlett-Packard iPAQ h4150 und h2210 bestätigt werden, dass die Anwendung wie vorgesehen funktioniert und stabil läuft. Durch die automatische Installationsroutine wird die Programminstallation zudem auch für technisch weniger versierte Benutzer ermöglicht. Es erscheint angemessen, während des Probebetriebs regelmäßige Feedbacks der Anwender einzuholen, um einerseits mögliche unerwartete technische Schwierigkeiten auf bisher nicht getesteten PDAs aufdecken zu können und andererseits anhand der Benutzerkommentare Kenntnisse über softwareergonomische Verbesserungspotentiale zu erhalten.

Wie gefordert werden die Lernprojekte zwar verschlüsselt, jedoch findet diese Verschlüsselung aus Gründen der Rechenzeit nur partiell auf die Textfragen statt. Daher können in der derzeitigen Entwicklungsversion weiterhin Fragen vom Typ Single-Choice und Multiple-Choice im Klartext durch Öffnen der Lernprojektdatei betrachtet werden. Weiterhin wird zu untersuchen sein, ob die Verarbeitungsdauer von Textaufgaben noch vom Benutzer akzeptiert wird und ob dieses Problem auch auf anderen PDAs zu beobachten ist. Da der Prozessor des iPAQ h4150 mit 400 MHz eine ausreichende Rechenleistung bereitstellt und somit die Prozessorleistung als Grund der langsamen Verarbeitung ausscheidet, stellt die Ursachenidentifikation eine Herausforderung für eine künftige Weiterentwicklung der Anwendung dar.

4.2 Entwicklungsperspektiven

Im Laufe der Systementwicklung konnten vielfältige Bereiche identifiziert werden, in denen sich Weiterentwicklungsperspektiven der UbiLearn mobile Anwendung ergeben. Zu den wichtigsten verbesserungswürdigen Bereichen gehören:

- ▶ Ein-/Ausgabefunktionen.
- ▶ Kommunikationsfunktionen mit anderen Nutzern.
- ▶ Kommunikationsfunktionen mit dem UbiLearn System.
- ▶ Didaktische Verbesserungen.
- ▶ Portierung auf andere Plattformen.

Ein-/Ausgabefunktionen: Die Analyse der mobilen Endgeräte hat gezeigt, dass die Mensch-Maschine Kommunikation, die für eine Interaktivität zwingend notwendig ist, aufgrund kleiner Displays und beschränkter Eingabemöglichkeiten Defizite im Vergleich zu Desktop PCs aufweist. Umso wichtiger ist es daher, einerseits die verfügbaren Ein-/Ausgabedevices optimal auszunutzen und andererseits den Einsatz von alternativen Ein-/Ausgabedevices zu prüfen. Neben der grafischen Ausgabe auf Displays unterstützen alle Windows Mobile kompatiblen PDAs die Ausgabe von Sounds über einen eingebauten Lautsprecher bzw. einen angeschlossenen Kopfhörer sowie die Aufzeichnung von Sounds über ein eingebautes Mikrofon. Mit synthetischer Sprachausgabe und automatischer Spracherkennung könnte sich der Lernende bei Bedarf die Lernmaterialien bzw. Übungsaufgaben vorlesen lassen und die Antworten per Sprache eingeben. Recherchen im Internet haben ergeben, dass es einige Anbieter gibt, die Sprachsynthese- und -erkennungsmodule zur Einbindung in selbsterstellte mobile Programme kostenpflichtig anbieten. In ersten Versuchen mit interaktiven Demonstrationen der Anbieter hat sich besonders das Produkt „Elan Sayso Pocket Speech“ der Firma Elanspeech³¹⁴ durch eine hohe Sprachausgabequalität ausgezeichnet.

Eine weitere interessante Option bildet die Kopplung der Anwendung mit Datenbrillen, mit denen Texte und Grafiken in das Blickfeld des Lernenden eingeblendet werden können. Hier sollte jedoch zunächst die Benutzerakzeptanz dieses Hilfsmittels in Zusammenhang mit M-Learning analysiert werden.

Kommunikationsfunktionen mit anderen Nutzern: In Abschnitt 2.5.2 wurde festgestellt, dass Kommunikationsfunktionen mit anderen M-Learning Nutzern vielfältige interessante Perspektiven zur Verbesserung des Lernerfolges bieten. Diese Kommunikationsfunktionen erfordern jedoch eine funktionierende Internetverbindung, weshalb die Anwendung in diesem Zusammenhang um die Fähigkeit, eine Onlineverbindung herstellen zu können, erweitert werden müsste. Im Rahmen der Konzeption wurde dafür bereits die Klasse `ubilearnNetwork` der Persistenzschicht vorgesehen, die zurzeit jedoch noch deaktiviert ist. Die Klasse `ubilearnBrowser` der Präsentationsschicht ermöglicht darüber hinausgehend bereits im aktuellen Entwicklungsstand die Darstellung von WWW-Seiten.

Kommunikationsfunktionen mit dem UbiLearn System: Im derzeitigen Entwicklungsstand unterstützt die Anwendung eine Kommunikation mit dem UbiLearn System

³¹⁴ <http://www.elanspeech.com>

nur in eine Richtung, d. h. vom UbiLearn System werden Lernprojekte extrahiert, die anschließend in die UbiLearn mobile Anwendung eingelesen werden können. Für eine vollständige Ubiquität ist es jedoch wünschenswert, dem Lernenden die Möglichkeit zu geben, je nach Bedarf zwischen den verschiedenen Trainingsmodulen wechseln zu können. In einem Beispiel ausgedrückt bedeutet dieses, dass der Anwender die Möglichkeit hat, ein Lernprojekt am internetfähigen Arbeitsplatz mit dem WBT-Modul zu beginnen und anschließend den aktuellen Zwischenstand zum Mobilgerät übertragen kann, um bspw. im Zug auf dem Heimweg weiterzulernen. Ist der Lernende schließlich zuhause angekommen, kann er den aktuellen Bearbeitungsstand wieder zum WBT-Modul übertragen und dort am komfortableren Heim PC weiterlernen. Technisch gesehen ist es möglich, diese Fähigkeit zu implementieren, da im Datenmodell des UbiLearn Systems bereits das Speichern von Zwischenständen unterstützt wird. Da das Datenmodell der UbiLearn mobile Anwendung das bestehende Datenmodell des UbiLearn System in seiner Struktur nahezu unverändert übernommen hat, besteht prinzipiell auch in der UbiLearn mobile Anwendung die Möglichkeit, Zwischenstände eines Lernprojekts zu speichern und anschließend wieder einzulesen. Ein weiterer Vorteil ergibt sich aus der Verwendung des ADO.NET Datenzugriffsmodells, da dieses Modell das Laden von Datensätzen aus unterschiedlichen Quellen erlaubt. Es macht daher für die UbiLearn mobile Anwendung keinen Unterschied, ob ein Lernprojekt aus einer XML-Datei, von einem XML-basierten Webservice oder über eine direkte TCP/IP Verbindung zu einem Datenbankserver geladen wird. Ähnlich wie das Übertragungsmedium für den Ladevorgang unerheblich ist, verhält es sich auch beim Speichern von Daten.

Auch wenn die UbiLearn mobile Anwendung aufgrund der in Abschnitt 2.5.4 geschilderten Gründe zumindest für die Kernfunktionalität, also das Bearbeiten von Übungsaufgaben, als offline M-Learning Anwendung konzipiert ist, bietet das ADO.NET Datenzugriffsmodell die Möglichkeit, die scharfe Trennung zwischen online und offline zu überwinden. Die offline M-Learning Anwendung könnte in eine hybride und intelligente M-Learning Anwendung weiterentwickelt werden, wobei unter Intelligenz in diesem Zusammenhang das automatische und möglichst vorausschauende Verhalten in Bezug auf die Verbindungszustände *online* und *offline* verstanden wird. Hierzu ist jedoch eine entsprechende Datenübertragungsstrategie erforderlich, die im MSDN Artikel „De-

signing ‚Sometimes Off-line‘ Microsoft .NET Compact Framework Applications“ an-satzweise beschrieben wird³¹⁵.

Im Zusammenhang mit der Erweiterung der Anwendung um Onlinefunktionen sollte in Betracht gezogen werden, dem Lernenden weitere Zusatznutzen durch die Bereitstellung von Diensten aus den Bereichen *Campus Service* und *Study Administration*³¹⁶ anzubieten. Diese Dienste setzen jedoch teilweise weitergehende organisatorische Veränderungen in der universitären Verwaltung sowie die programmtechnische Umsetzung durch einen entsprechenden Internetserver voraus.

Didaktische Verbesserungen: Der Bereich der didaktischen Verbesserungen umfasst vielfältige Aspekte. Hervorzuheben sind hierbei allgemeine Aspekte der Softwareergonomie (Benutzerführung, Farbgestaltung etc.) und die Fähigkeit der Anwendung, sich eigenständig und intelligent an das Lernverhalten und an die Lernerfordernisse des Lernenden anzupassen.

Portierung auf andere Plattformen: Im Rahmen der Portierung auf andere Plattformen ist zwischen zwei Szenarien zu unterscheiden. Einerseits die Portierung auf andere Windows CE .NET Plattformen und andererseits die Portierung auf Plattformen mit alternativen Betriebssystemen. Da .NET Programme grundsätzlich auf sämtlichen Betriebssystemplattformen mit installiertem .NET Framework lauffähig sind, erfordert die Portierung auf andere Windows CE .NET Plattformen (bspw. Smartphone mit Windows Mobile for Smartphone oder Tablet PC) lediglich eine Anpassung bzw. einen Austausch der Benutzeroberfläche, falls sich die Displaygröße der anzupassenden Plattform von der Ursprungplattform unterscheidet. Weiterhin könnte die UbiLearn mobile Anwendung auch auf einen Desktop PC mit installiertem Microsoft Windows portiert und als herkömmliches CBT-System betrieben werden; jedoch ist dazu der komplette Austausch aller Klassen der Präsentationsschicht erforderlich, weil sämtliche benutzten Formulklassen der Klassenhierarchie Microsoft.WindowsCE.Forms entstammen, die auf dem „großen“ .NET Framework nicht zur Verfügung stehen. Eine Portierung der Anwendung auf Plattformen mit alternativen Betriebssystemen (hier kommen vor allem Smartphones mit Symbian OS in Frage) erfordert jedoch, basierend auf dem Programm-entwurf in Abschnitt 3.2, eine vollständige Neuprogrammierung der Anwendung in Java

³¹⁵ Vgl. FORSBURG 2003.

³¹⁶ Vgl. Abschnitt 2.5.2 – S. 26.

und hat zur Folge, dass ab diesem Zeitpunkt zwei parallele Anwendungsstränge gewartet und entwickelt werden müssen.

4.3 M-Learning Trends

M-Learning kann als konsequente Weiterentwicklung des klassischen E-Learning betrachtet werden. Der wesentliche Unterschied scheint darin zu liegen, dass der Lernvorgang nicht auf stationären PCs, sondern auf mobilen Endgeräten erfolgt. In dieser Betrachtung liegt der wesentliche Vorteil des mobilen Lernens darin, dass Lernen endlich bspw. auch im Auto, im Zug oder auf der Parkwiese möglich ist. Diese Annahme ist zwar ein verbreitetes Verständnis, jedoch bieten sich mit M-Learning vielfältige weitergehende Perspektiven: M-Learning wird häufig auf das bloße Bereitstellen von Lernunterlagen beschränkt. Doch Lernunterlagen stellen streng genommen nicht selbst das *Lernen* dar, sondern sind bestenfalls Voraussetzung dafür. Gerade mobile Endgeräte in Form eines PDA oder Smartphone sind aufgrund ihrer geringen Displaygrößen zudem nur eingeschränkt für eine umfangreiche Inhaltsvermittlung geeignet. Die wirklichen Vorzüge von PDAs und Smartphones liegen woanders. Aufgrund des geringen Formats gehören diese Geräte zu den Accessoires eines Menschen, d. h. sie werden ähnlich einer Brille, dem Geldbeutel oder Schlüssel ständig am Körper mitgeführt. Die Geräte sind daher geeignet, ständig Kontextinformationen über den Besitzer zu liefern. Bspw. kann mithilfe eines GPS-Moduls der Aufenthaltsort bestimmt werden, es können Daten über persönliche Vorlieben und den individuellen Lerntyp ermittelt werden. Weiterhin kann das Gerät auch Statusinformationen über die aktuelle Verfügbarkeit des Benutzers bereithalten („am Arbeitsplatz“, „im Kino“, „bitte nicht stören“ etc.) oder seine Stimmungslage („gut drauf“, „schlecht gelaunt“)³¹⁷.

Diese Kontextinformationen bilden eine Voraussetzung, um personalisierte Lernangebote im Rahmen einer individuellen Lernumgebung zur Verfügung zu stellen. Des Weiteren könnte kollaboratives Lernen stark gefördert werden. Wenn bekannt ist, wer in einer Gemeinschaft mit der aktuellen Fragestellung beschäftigt ist und für eine Diskussion zur Verfügung steht, könnte gemeinsames Lernen per Diskussion endlich auch spontan stattfinden, ohne dass gemeinsame Termine gefunden werden müssen, ohne dass lange Anfahrtswege in Kauf genommen werden müssen und ohne umständlich Räume organisieren zu müssen.

³¹⁷ Vgl. FROHBERG 2003.

Auch wenn M-Learning eine wichtige Rolle vor allem beim Lernen in Distanz zu einer Präsenzveranstaltung zugerechnet wird, bieten sich interessante Perspektiven für den eingebetteten Einsatz innerhalb einer Lehrveranstaltung. Im Rahmen des Projektes „Notebook University“³¹⁸ an der TU Darmstadt kann in ausgewählten Lehrveranstaltungen über mobile Endgeräte interaktiv an der Vorlesung teilgenommen werden. Mit einer „Live Evaluierung“ können die Studenten während des Vortrages ihre Bewertungen zu Tempo und Relevanz des aktuellen Vorlesungsstoffes abgeben. Außerdem ist es den Studierenden möglich, Kommentare und Fragen an den Vortragenden zu senden, so dass dieser geeignete Fragen in den Fluss der Veranstaltung einbauen kann.

Insgesamt bieten mobile Endgeräte vielfältige interessante Perspektiven, um das reine M-Learning in Gestalt einer mobilen Version des klassischen E-Learning hin zu einem allgegenwärtigen persönlichen Lernen („P-Learning“) weiterzuentwickeln. Hierzu wird es jedoch erforderlich sein, die Einschränkungen und Besonderheiten mobiler Kleinrechner gegenüber den „großen“ Desktop PCs nicht mehr als Hindernis der Portierung einer herkömmlichen Lernsoftware auf einen PDA oder Smartphone aufzufassen. Stattdessen sollte das *Personal* im Wort Personal Digital Assistant als Chance gesehen werden, dem Lernenden vielfältige Zusatznutzen gegenüber herkömmlichen Lernlösungen zu bieten.

Um hier zu weiterführenden Erkenntnissen zu gelangen, liegt für die Zukunft noch ein erheblicher Forschungsbedarf vor. Die Schlagworte „Gemeinschaft“ und „Persönlichkeit“ machen allerdings deutlich, dass der wissenschaftliche Rahmen der Wirtschaftsinformatik möglicherweise um weitere Fachdisziplinen erweitern werden sollte, um zu adäquaten Ergebnissen zu kommen. Ein interdisziplinärer Forschungsansatz unter der zusätzlichen Einbeziehung pädagogischer, psychologischer, soziologischer und möglicherweise auch medizinischer Fachdisziplinen erscheint in diesem Zusammenhang Erfolg versprechend.

³¹⁸ Nähere Informationen zum Projekt „Notebook University“ sind unter <http://www.nu.tu-darmstadt.de/> online abrufbar.