

# Realisierung einer offline E-Learning Anwendung basierend auf .Net

## Diplomarbeit

zur Erlangung des Grades eines Diplom-Ökonomen der  
Wirtschaftswissenschaftlichen Fakultät der Leibniz Universität Hannover

Vorgelegt von

**Robert Brandt**



Erstprüfer: Prof. Dr. Michael H. Breitner

Hannover, den 11.06.2007

---

## Inhaltsverzeichnis

<b>Abbildungsverzeichnis .....</b>	<b>V</b>
<b>Verzeichnis der Programmlistings.....</b>	<b>VII</b>
<b>Abkürzungsverzeichnis .....</b>	<b>VIII</b>
<b>1 Einleitung.....</b>	<b>1</b>
1.1 Motivation .....	1
1.2 Zielsetzung und Vorgehensweise.....	2
<b>2 Grundlagen.....</b>	<b>5</b>
2.1 Ubiquitous Learning System .....	5
2.1.1 Grundlagen.....	5
2.1.2 Administrationsebene .....	7
2.1.3 Benutzerebene .....	8
2.1.4 UbiMotion .....	12
2.2 Vom E-Learning zum Blended Learning.....	12
2.2.1 Definition.....	12
2.2.2 Varianten und Einsatzmöglichkeiten.....	15
2.2.3 Anwendungsbeispiele.....	17
2.3 Zwischenergebnis .....	18
<b>3 Anwendungsentwicklung.....</b>	<b>19</b>
3.1 Softwareentwicklung .....	19
3.1.1 Vorgehensmodelle.....	19
3.1.2 Software-Qualität.....	24
3.1.3 Auswahl Entwicklungsmodell.....	25
3.2 Analysephase.....	26
3.2.1 Ist-Analyse des UbiLearn Systems.....	26
3.2.1.1 Datenmodell.....	26
3.2.1.2 SQL-Inquirer und XML-Transportcontainer.....	29
3.2.1.3 Punktevergabe.....	30
3.2.2 Ist-Analyse der Marktsituation von Betriebssystemen .....	32

---

3.2.2.1	Microsoft Betriebssysteme.....	32
3.2.2.2	Alternative Betriebssysteme .....	33
3.2.3	Grobkonzept einer offline E-Learning Anwendung .....	34
3.2.4	Auswahl der Entwicklungsumgebung .....	35
3.2.4.1	Microsoft .NET .....	35
3.2.4.2	Mono Project.....	40
3.2.4.3	Entscheidung über Entwicklungsumgebung .....	42
3.2.4.4	Auswahl der Programmiersprache.....	43
3.3	Entwurfsphase.....	44
3.3.1	Systementwurf.....	45
3.3.2	Pflichtenheft.....	48
3.3.2.1	Zielbestimmung .....	48
3.3.2.2	Umgebung .....	49
3.3.2.3	Funktionen .....	49
3.3.2.4	Benutzeroberfläche.....	50
3.3.2.5	Entwicklungsumgebung.....	50
3.3.3	Programmmentwurf .....	51
3.4	Realisierungsphase.....	52
3.4.1	Programmierung der UbiLearn CBT Anwendung .....	52
3.4.1.1	Elemente der Präsentationsschicht .....	53
3.4.1.1.1	Klasse Form_Main .....	53
3.4.1.1.2	Klasse Form_User_Login.....	56
3.4.1.1.3	Klasse Form_User_Add .....	57
3.4.1.1.4	Klasse Form_Project_Add .....	60
3.4.1.1.5	Klasse Form_Project.....	61
3.4.1.1.6	Klasse Form_Question.....	63
3.4.1.1.7	Klasse Form_Answer .....	68
3.4.1.1.8	Klasse Form_Image.....	69
3.4.1.1.9	Klasse Form_Attachment.....	70
3.4.1.1.10	Klasse Form_About.....	72
3.4.1.2	Elemente der Anwendungsschicht.....	73
3.4.1.2.1	Klasse UbiLearnCBT.....	73
3.4.1.2.2	Klasse UbiLearnLearnObject .....	73
3.4.1.2.3	Klasse UbiLearnEncryption.....	76

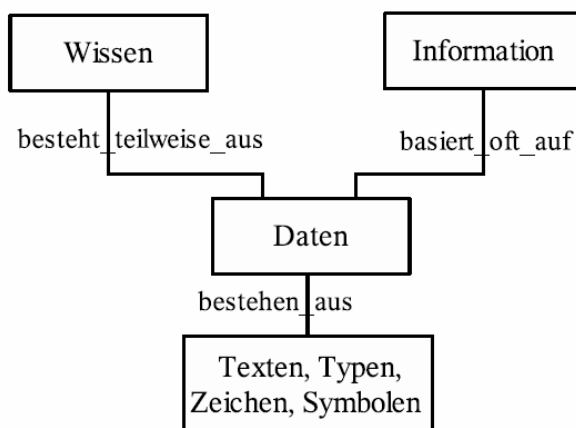
---

3.4.1.2.4	Klasse DBConnection .....	77
3.4.1.2.5	Klasse UbiLearnAttachment.....	78
3.4.1.2.6	Klasse UbiLearnAttachmentLoad.....	78
3.4.1.3	Querschnittselemente .....	79
3.4.1.3.1	Klasse stringformatter .....	79
3.4.1.3.2	Klasse Xtea.....	79
3.4.1.3.3	Datenbank ubilearn.s3db .....	80
3.4.1.4	Assemblies .....	80
3.4.2	Anpassung an Mono.....	81
3.4.2.1	MoMA .....	81
3.4.2.2	Assamblies .....	83
3.4.2.3	Installationstroutine.....	83
3.4.3	Programmtest.....	84
3.4.3.1	Windows XP .....	85
3.4.3.1.1	Testumgebung .....	85
3.4.3.1.2	Installation und Programmstart .....	86
3.4.3.1.3	Nutzerverwaltung und Hinzufügen eines Lernobjektes .....	86
3.4.3.1.4	Bearbeitung eines Lernobjektes.....	87
3.4.3.2	Windows Vista .....	90
3.4.3.2.1	Testumgebung .....	90
3.4.3.2.2	Installation und Programmstart .....	91
3.4.3.2.3	Nutzerverwaltung und Hinzufügen eines Lernobjektes .....	91
3.4.3.2.4	Bearbeitung eines Lernobjektes.....	92
3.4.3.3	Ubuntu Linux.....	94
3.4.3.3.1	Testumgebung .....	94
3.4.3.3.2	Programmstart .....	95
3.4.3.3.3	Nutzerverwaltung und Hinzufügen eines Lernobjektes .....	95
3.4.3.4	Ergebnis des Programmtests und Problemanalyse .....	96
<b>4</b>	<b>Schlussbetrachtung und Ausblick .....</b>	<b>98</b>
4.1	Fazit .....	98
4.2	Entwicklungsperspektiven .....	99
	<b>Literaturverzeichnis.....</b>	<b>102</b>

## 1 Einleitung

### 1.1 Motivation

im 21. Jahrhundert, dem Übergang vom Industrie- ins Informationszeitalter, wird die Ressource Wissen und die Effizienz der nationalen Bildungssysteme für den globalen Wettbewerb zum Erfolgsfaktor. Lernen wird dadurch von einem isolierten Ereignis in einen dauerhaften Prozess mit wachsender Eigenverantwortung des Lernenden gewandelt.<sup>1</sup>



**Abbildung 1:** Abgrenzung Wissen und Informationen  
(Quelle: Spitta 2006, S. 50).

Anforderungen, die an das Bildungssystem von heute gestellt werden ergeben sich in erster Linie aus der Art wie die Ressource Wissen verarbeitet werden kann. Eine Herausforderung ist, die immer stärker steigenden Wissensmengen weiterzugeben. Wie kann eine Weitergabe von immer mehr Informationen bei gleichbleibenden Zeitressourcen ermöglicht werden. Statistischen Untersuchungen zufolge verdoppelt sich das vorhandene Wissen jeweils in einem Zeitraum von gerade mal 5 Jahren. Die zweite Herausforderung ist, der immer geringer werdenden Halbwertszeit von Wissen Herr zu werden. Fügt man nun die beiden Herausforderungen zusammen, das wachsende Wissen und die zugleich sinkende Geltungsdauer, entsteht der Eindruck, dass sich unsere Gesellschaft schrittweise in eine Informations-Sackgasse bewegt. Wird an einem Ende der Gesellschaft kontinuierlich neues Wissen geschaffen, versucht der Großteil der Betroffenen die Innovationen zu erlernen. Dabei muss nicht nur versucht werden, möglichst viel von

<sup>1</sup> Vgl. Schwuchow 2002, S. 29.

dem entstehenden Wissen zu erlernen, sondern auch seinen bestehenden Wissensstand zu aktualisieren. Dass herkömmliche Vermittlungsformen an diesen Herausforderungen scheitern liegt auf der Hand.<sup>2</sup>

Der Einsatz von E-Learning an Hochschulen kann als eine Lösung der oben genannten Probleme gesehen werden. Ein Vorteil des E-Learning Einsatzes ist, der Faktor Kosten, die sich im Vergleich zum herkömmlichen Lernen erheblich senken lassen. Zudem wird Lernen unabhängig von Zeit und Ort, wodurch zum einen der Lernerfolg verbessert und die Reaktionsgeschwindigkeit der Lerninhalte vermittelnden Institution gesteigert wird.<sup>3</sup> Als eine Mischform von herkömmlichen und E-Learning, bietet der Einsatz von Blended Learning die größten Potentiale für den Hochschuleinsatz. Neben den üblichen Präsenzveranstaltungen wird eine E-Learning Plattform bereitgestellt, die es Lernenden ermöglicht das erlangte Wissen zu überprüfen und sich gegebenenfalls vertiefendes Wissen anzueignen.<sup>4</sup>

Eine den Lernenden unterstützende Methode des E-Learning ist eine, auf dem eigenen Computer installier- und ausführbare E-Learning Anwendung, für die Bearbeitung der vorlesungsbegleitenden Lerninhalte.

## 1.2 Zielsetzung und Vorgehensweise

Zielsetzung dieser Arbeit ist, die Entwicklung einer stationären offline E-Learning Anwendung auf der Basis der .NET Technologie, zur nahtlosen Eingliederung als stationäres CBT-Modul in das bestehende UbiLearn System. Die zu entwickelnde Anwendung soll sich dabei in Funktionalität und Bedienung an den vorhandenen Modulen für web basiertes (UbiLearn WBT) und mobiles (UbiLearn mobile) Lernen orientieren und auf Daten bereits vorhandener Technologien zugreifen (SQL-Inquirer). Einsatzgebiet sollen Desktop Computer und Laptops sein, die nicht notwendigerweise über einen Internetzugang verfügen. Der Name der Anwendung ist UbiLearn CBT.

**Grundlagen:** Für eine vernünftige Integrierung der Anwendung, wird in Kapitel 2 zunächst eine Beschreibung des bestehenden UbiLearn Systems vorgenommen und eine Einführung in die Begrifflichkeiten des E-Learning ge-

---

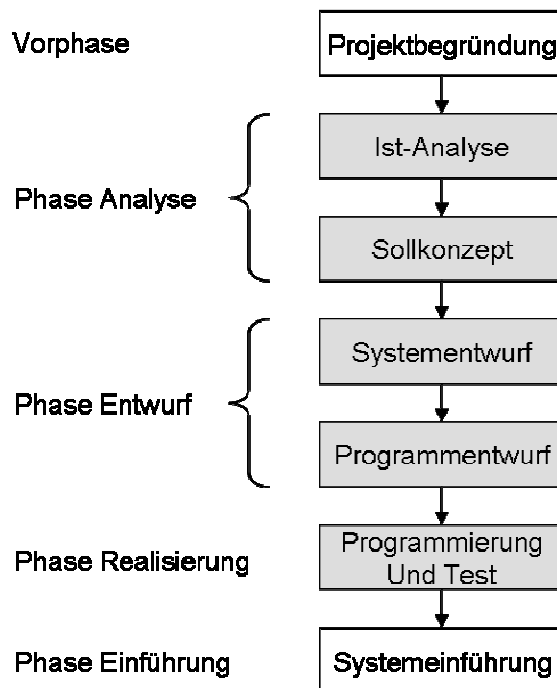
<sup>2</sup> Vgl. Bauer / Philippi 2001, S.135 - 140.

<sup>3</sup> Vgl. Neumann / Reichert 2002, S. 199.

<sup>4</sup> Vgl. Reinmann-Rothmeier 2003, S. 28 - 29.

geben. Die eigentliche Systementwicklung der Anwendung findet danach in Kapitel 3 statt.

Gerade im Bereich der Systementwicklung ist es wichtig wie der Verlauf der Entwicklung aussehen soll nach welchen Qualitätsmerkmalen die Anwendung entwickelt werden soll. Der Verlauf der Anwendungsentwicklung wird in einem Vorgehensmodell beschrieben, das den gesamten Entwicklungsprozess in verschiedene Phasen einteilt. Auch die Softwarequalität wird durch die Wahl eines Vorgehensmodells entscheidend beeinflusst und stellt einen Teil der Anwendungsentwicklung dar. Fast alle in Abschnitt 3.1.1 beschriebenen Vorgehensmodelle beinhalten ähnliche Phasen, weswegen an dieser Stelle die Unterteilung der Systementwicklung vorgenommen wird, ohne eine konkrete Auswahl eines Vorgehensmodells vorweg zu nehmen.



**Abbildung 2:** Phasen der Systementwicklung  
(Quelle: Eigen Darstellung in Anlehnung an Stahlknecht / Hasenkamp 2005, S. 218).

**Analysephase:** Die Analysephase befasst sich zunächst mit einer Ist-Analyse des Entwicklungsumfeldes. Zielsetzung der Analysephase ist die Konstruktion eines Sollkonzepts, welches Grundlage für ein aufzustellendes Grobkonzept ist.

Die Umfeldanalyse befasst sich zuerst mit der Analyse des bestehenden Ubi-Learn Systems. Dabei werden zunächst die Module untersucht, welche für die Realisierung der E-Learning Anwendung relevant sind. Im Speziellen sind

dies das Datenmodell des UbiLearn Systems, der SQL-Inquirer und die vom UbiLearn System verwendete Punktevergabe. Der zweite Schritt der Analysephase befasst sich mit einer Marktanalyse aktueller, auf den Zielplattformen eingesetzter Betriebssysteme gefolgt von der Aufstellung eines Sollkonzeptes aus den ermittelten Ergebnissen. Das Sollkonzept zeigt auf was die Anwendung leisten soll und wie die Anwendung realisiert werden soll. Aus der geplanten Vorgehensweise der Anwendung wird ein Grobkonzept entwickelt und im weiteren Verlauf die Werkzeuge für die Entwicklung näher betrachtet. Als erstes soll die Frage nach einer Entwicklungsumgebung beantwortet werden gefolgt von der Auswahl einer unterstützten Programmiersprache. Bei der Wahl der Werkzeuge soll insbesondere auf die Vorgaben des UbiLearn Systems nach Plattformunabhängigkeit, Qualität und Wiederverwendbarkeit von Code und Inhalten geachtet werden. Alle zusammengefassten Ergebnisse bilden die Grundlage für die folgende Phase.

**Entwurfphase:** Die Entwurfphase hat die Aufgabe, aus dem in der Analysephase aufgestelltem Sollkonzept den eigentlichen Systementwurf abzuleiten, den Systementwurf in Vorgaben für ein Pflichtenheft zu gliedern und auf Basis des Pflichtenheftes einen Programmentwurf zu formulieren.<sup>5</sup>

Der Systementwurf wird objektorientiert formuliert. Objektorientiert deshalb, da moderne Entwicklungsumgebungen fast ausschließlich objektorientierte Programmierung unterstützen und somit eine gute Trennung zwischen Daten und Funktion ermöglicht. Der zweite Schritt der Entwurfphase ist das Erstellen eines Pflichtenheftes, welches eine ausführliche Beschreibung der fachlichen Anforderungen an die Anwendung enthält. Neben den Zielen, die die Anwendung erfüllen soll beinhaltet das Pflichtenheft die Anforderungen an die Zielplattform, die zu erfüllenden Funktionen und eine Beschreibung der verwendeten Entwicklungsumgebung. Abgeschlossen wird die Entwurfphase durch den Programmentwurf, der die theoretische Konzeption der UbiLearn CBT Anwendung enthält und Basis für die Realisierungsphase ist.

**Realisierungsphase:** In Rahmen der Realisierungsphase findet die eigentliche Programmierung der UbiLearn CBT Anwendung statt gefolgt von einem Programmtest.<sup>6</sup>

---

<sup>5</sup> Vgl. Stahlknecht / Hasenkamp 2005, S. 211.

<sup>6</sup> Vgl. Stahlknecht / Hasenkamp 2005, S. 282.



Programmiert wird die Anwendung auf der zuvor ausgewählten Entwicklungsumgebung und der implementierten Programmiersprache. Dabei wird die Anwendung in verschiedene Klassen eingeteilt, die jeweils eine Funktion der Anwendung übernehmen und somit in verschiedene Schichten eingeteilt werden können. Die Realisierungsphase wird abgeschlossen durch einen ausführlichen Programmtest auf verschiedenen Computern und unterschiedlichen Betriebssystemen. Der Programmtest überprüft ob die Anwendung alle vorher formulierten Anforderungen erfüllt und ob das Programm stabil und korrekt ausgeführt wird. Die dabei verwendeten Lernprojekte werden alle aus denen im UbiLearn System vorhandenen gewählt.

**Schlussbetrachtung und Ausblick:** Abschließend befasst sich die Arbeit mit einer Zusammenfassung der Ergebnisse der vorangegangenen Abschnitte. Weiterhin werden Erfahrungen mit der Konzeption und der Entwicklung aufgezeigt gefolgt von Entwicklungsperspektiven der UbiLearn CBT Anwendung und eine mögliche Integration in die Infrastruktur der wirtschaftswissenschaftlichen Fakultät der Leibniz Universität Hannover.

## 2 Grundlagen

### 2.1 Ubiquitous Learning System

#### 2.1.1 Grundlagen

Das Ubiquitous Learning System (UbiLearn) ist eine kundenorientierte Individualsoftwarelösung aus dem Bereiche E-Learning. Entwickelt wurde es am Institut für Wirtschaftsinformatik der Leibniz Universität Hannover im Rahmen der Diplomarbeit von A. Otmakhov.<sup>7</sup>

Den Aufbau des UbiLearn Systems verdeutlicht Abbildung 3.

---

<sup>7</sup> Vgl. Breitner 2006.

festgelegten Größe angezeigt und die darin enthaltenen Elemente wurden falsch angeordnet. Der Datenbankzugriff funktionierte, was dadurch zu belegen ist, dass sowohl ein neuer Benutzer angelegt wurde als auch eine Anmeldung reibungslos funktionierte. Bei dem Versuch ein Lernprojekt hinzuzufügen brach die Anwendung mit einem nicht zu behebbenden Fehler ab. Als Überprüfung der Ursache des Fehlers wurde die für Mono angepasste Version mit dem .NET Framework unter Windows ausgeführt. Hierbei traten zwar auch kleine grafische Unzulänglichkeiten auf, jedoch funktionierte die Anwendung vollständig. Als zweiter Test wurde die Anwendung über das Mono-Framework unter Windows aufgerufen, optisch ähnelte sie der vorhergegangenen Version jedoch verursachte auch sie einen Fehler bei dem Versuch einen neuen Lerninhalt hinzuzufügen. Als Ergebnis dieser Tests kann also bestätigt werden, dass es sich bei dem Problem um ein Mono spezifisches Problem handelt, welches möglicherweise in einer neueren Version behoben wird.

## **4 Schlussbetrachtung und Ausblick**

### **4.1 Fazit**

Dieser Abschnitt soll abschließend den Prozess der Anwendungsentwicklung und den Entwicklungsstand der UbiLearn CBT Anwendung unter Berücksichtigung der Ergebnisse des Abschnitts 3.4.3.4 zusammenfassen und beurteilen.

Der Prozess der Anwendungsentwicklung hat verdeutlicht, dass sowohl die Auswahl eines Vorgehensmodells und auch die Bewertung von Software-Qualität eine große Herausforderung darstellen. Die in der Literatur vorgestellten Vorgehensmodelle enthalten zwar annähernd die gleichen Phasen, ein vollständiger Durchlauf aller Phasen wird in dieser Entwicklung jedoch nicht vorgenommen. Vielmehr wird ein einfaches Phasenmodell verwendet, welches für die benötigten Anforderungen modifiziert wurde. Ähnlich verhält es sich mit der, in Abschnitt 3.1.2 behandelten Softwarequalität. Die Bestimmungen und Vorgaben nach ISO 9001 sind für diese Entwicklung zu stark, weshalb auf ein Qualitätsmodell nach DIN 66 272 zurückgegriffen wird. Die

darin enthaltenen Qualitätsmerkmale reichen für die Qualitätsbestimmung aus und werden von der Anwendung erfüllt.

Der Wunsch nach Plattformunabhängigkeit wurde durch die Wahl der Entwicklungsumgebung und der Programmiersprache versucht zu realisieren, er erwies sich jedoch als schwierig. Aus diesem Grund wurde die Entscheidung für eine spezielle Zielplattform getroffen, auf der die Anwendung vollständig lauffähig ist und alle Anforderungen des UbiLearn Systems erfüllt. Während der Entwicklung wurde gleichzeitig darauf geachtet, die Anwendung so zu entwickeln, dass eine Portierung auf alternative Betriebssysteme ermöglicht werden soll. Hauptaugenmerk wurde auf die aktuellen Betriebssysteme von Microsoft und die .NET Plattform gelegt. Unter Zuhilfenahme des Mono-Frameworks sollte die Anwendung auch auf UNIX-basierten Systemen laufen, jedoch scheiterte der Versuch wie in Abschnitt 3.4.3.3 beschrieben an einem nicht zu behebbenden Fehler, der nach weiterer Überprüfung der Inkompatibilität des Mono Frameworks zum .NET Framework zugeschrieben werden kann. Der Test der UbiLearn CBT Anwendung auf den Microsoft-Systemen Windows XP und Windows Vista konnte erfolgreich abgeschlossen werden und zeigte, dass alle geforderten Ziele und Funktionen des in Abschnitt 3.3.2 aufgestellten Pflichtsystems erfüllt werden. Lediglich der Wunsch nach online herunterladbaren Lernprojekten konnte nicht erfüllt werden. Die Anwendung erlaubt es dem Anwender sich selbständig mit den Lerninhalten zu befassen. Nicht nur die reine Wissensabfrage, parallel zu den Vorlesungen wird unterstützt, sondern ist es auch vorstellbar, komplette Vorlesungen über Audio und Video in die Lernprojekte einzubinden, so dass eine vollständig virtuelle Lernveranstaltung denkbar wäre.

## 4.2 Entwicklungsperspektiven

In Bezug auf die Weiterentwicklung der UbiLearn CBT Anwendung sollte genauesten beobachtet werden, in wie weit sich die Kompatibilität des Mono-Framework im Vergleich zur Microsoft-Plattform entwickelt, gerade in Bezug auf optische Darstellung und den in Abschnitt 3.3.2 aufgetauchten Fehler. Wenn die vollständige und identische Implementierung der Windows-Forms abgeschlossen ist und so ein identisches Aussehen und Bedienbarkeit unter UNIX basierten Betriebssystem gewährleistet werden kann, sollte eine Mög-

lichkeit gefunden werden den Fehler beim Einlesen der XML-Transportcontainer zu umgehen. Ein möglicher Ansatzpunkt könnte die Erweiterung der SQL-Inquirer Anwendung um verschiedene Ausgabeformate der Lerninhalte. So wäre es z. B. möglich, die in der MySQL-Datenbank enthaltenen Lernprojekte als SQL-Struktur auszulesen und diese über die UbiLearn CBT Anwendung in die anwendungseigene Datenbank einzubinden.

Eine weitere mögliche Erweiterung der UbiLearn Anwendung ist eine Verbindung zur WBT und der mobilen Variante des UbiLearn Systems. Da die verwendbaren Lernprojekte in allen Anwendungen identisch sind, könnte eine Verknüpfung der verschiedenen Module ein Bearbeitung der Lernprojekte auf allen unterstützten Plattformen ermöglichen. So kann ein Lernprojekt im WBT begonnen werden, die erreichten Ergebnisse auf die CBT Anwendung übertragen werden und dort fortgeführt werden. Die Plattform für die kumulierten Ergebnisse sollte dabei das UbiLearn WBT bilden, da diese auf den Server mit Lernprojekten und weiteren Daten zugreifen kann. In diesem Rahmen könnte auch ein virtueller Klassenraum aufgebaut werden, mit online Veranstaltungen, Diskussionsforen und online Chat. Dieses würde jedoch bedeuten, dass die Anwendung eine Verbindung zum Internet benötigt um den vollen Funktionsumfang nutzen zu können. Die UbiLearn CBT würde sich also von einer reinen offline Anwendung in eine hybride Anwendung wandeln. Voraussetzung für die oben genannten Entwicklungsperspektiven ist die Unterstützung und Implementierbarkeit durch die gewählte Entwicklungsumgebung. Da die .NET Plattform im speziellen genau für die Vernetzung von Informationssystem entwickelt wurde ist die Grundlage für ein komplett vernetztes UbiLearn System gegeben um allgegenwärtiges Lernen zu ermöglichen.

Die Einführung der Anwendung an der Leibniz Universität Hannover könnte möglicherweise über die Online-Prüfungsanmeldung geschehen. So wäre es z.B. möglich auf dieser Plattform die XML-Transportcontainer für die angemeldeten Prüfungen den Lernenden anzubieten. Diese Vorgehensweise könnte gerade für die Idee von vollständig virtuellen Lernveranstaltungen verwendet werden um den Zugang nur berechtigten Benutzergruppen zu ermöglichen.

Abschließend kann gesagt werden, dass das UbiLearn System und die darin enthaltenen UbiLearn CBT Anwendung dem Lernenden die Möglichkeit bietet das universitäre Lernen entscheidend zu erweitern und verbessern. Dem Anwender steht die Wahl der Lernmethode offen, so dass er die für seine Lerngewohnheiten beste Methode wählen kann und so flexibel in Bezug auf Ort und Zeit ist. Ein allgegenwärtiges Lernen kann somit ermöglicht werden.