

# **Konzeption und Entwicklung eines Online-Datenbankmoduls zur Unterstützung der Fremd- und Selbstbewertung von Kompetenzen im handlungsorientierten Unterricht**

## **Examensarbeit**

im Unterrichtsfach Informatik  
Studiengang Lehramt an der Oberstufe  
- Berufliche Schulen -  
der  
Universität Hamburg  
und der  
Technischen Universität Hamburg-Harburg

**Henning Walter Klaffke**

Erstprüfer: Prof. Dr. Sönke Knutzen  
Zweitprüfer: Dr.-Ing. Erich Boeck

Bearbeitungszeitraum: 13. April 2004 bis 13. Juli 2004

Hamburg, 9 Juli 2004

## Kurzfassung

Gegenstand der hier vorgestellten Arbeit sind die Konzeption und die Entwicklung einer Web-Applikation, die zur Unterstützung der Fremd- und Selbstbewertung von Kompetenzen in handlungsorientierten Unterrichten eingesetzt werden soll.

Im handlungs- und projektorientierten Unterricht gestalten sich Leistungsbeurteilungen und die darauf basierenden Bewertungen aufgrund der geforderten Fremd- und Selbsteinschätzungen in der Unterrichtspraxis schwierig. Durch Schaffung einer Web-Applikation soll als Produkt ein Hilfsmittel für diese Form der Bewertung entstehen.

Die Konzeption und die Entwicklung dieser Web-Applikation dienen als Beispiel für eine Modellierung bei der Software-Entwicklung. Die einzelnen Phasen der Entwicklung, die eingesetzten IT-Technologien und die Entwürfe dieser Applikation bilden den Schwerpunkt dieser Arbeit.

**Schlagwörter:** Leistungsbeurteilung, Web-Applikation, Datenbank, Modellierung, Software-Entwicklung.

# Inhaltsverzeichnis

|  |           |
|--|-----------|
| <b>Kurzfassung</b> .....                                       | <b>2</b>  |
| <b>Inhaltsverzeichnis</b> .....                                | <b>3</b>  |
| <b>Abbildungsverzeichnis</b> .....                             | <b>5</b>  |
| <b>Einleitung</b> .....  | <b>6</b>  |
| <b>1 Ziel und Aufbau der Arbeit</b> .....                      | <b>7</b>  |
| 1.1 Ziel der Arbeit.....                                       | 7         |
| 1.2 Aufbau der Arbeit .....                                    | 7         |
| <b>2 Modelle der Software-Entwicklung</b> .....                | <b>9</b>  |
| 2.1 Die Wasserfall-Methode .....                               | 9         |
| 2.2 Das Spiral-Modell .....                                    | 10        |
| 2.3 Das Phasen-Modell mit Prototyping .....                    | 12        |
| <b>3 Konzeption und Entwicklung der Web-Applikation</b> .....  | <b>13</b> |
| 3.1 Die Planungsphase .....                                    | 13        |
| 3.1.1 Projektbezogene Anforderungen an die Software.....       | 13        |
| 3.1.2 Das Lastenheft der Web-Applikation .....                 | 16        |
| 3.1.3 Ist-Analyse .....  | 19        |
| 3.1.4 Abschluss der Planungsphase .....                        | 21        |
| 3.2 Die Definitionsphase .....                                 | 21        |
| 3.2.1 Das Pflichtenheft der Web-Applikation .....              | 21        |
| 3.2.2 Technische Produktumgebung .....                         | 27        |
| 3.2.3 Abschluss der Definitionsphase.....                      | 29        |
| 3.3 Die Entwurfsphase .....                                    | 30        |
| 3.3.1 Darstellung der Systemarchitektur.....                   | 30        |
| 3.3.2 Modellierung der Datenbank der Web-Applikation.....      | 37        |
| 3.3.3 Benutzermanagement .....                                 | 38        |
| 3.4 Die Implementierungsphase.....                             | 38        |
| 3.4.1 Vollständiger Aufbau des Backend.....                    | 39        |
| 3.4.2 Realisierter Prototyp.....                               | 40        |
| 3.4.3 Teilergebnisse der Implementierungsphase .....           | 48        |
| <b>4 Fazit</b> .....   | <b>49</b> |
| <b>5 Glossar:</b> .....  | <b>50</b> |
| <b>Literaturverzeichnis</b> .....                              | <b>53</b> |
| <b>Internethinweise:</b> .....                                 | <b>54</b> |
| <b>Anhang A: Anwendungsfall- und Aktivitätsdiagramme</b> ..... | <b>55</b> |

---

|  |           |
|--|-----------|
| <b>Anhang B: Lizenzbestimmung.....</b> | <b>63</b> |
| <b>Erklärung.....</b>                  | <b>68</b> |

## Abbildungsverzeichnis

|  |    |
|--|----|
| Abbildung 1: Das Wasserfallmodell nach Boehm (vgl. BOEHM 1981: 30) .....   | 9  |
| Abbildung 2: Das Spiral-Modell (vgl. SOMMERVILLE 2001: Kap.1) .....  | 10 |
| Abbildung 3: Das Phasen-Modell mit Prototyping als Entwicklungskonzept<br>der Web-Applikation (vgl. BOEHM 1981: 38)..... | 12 |
| Abbildung 4: Mind Map der Web-Applikation zur Prozessanalyse der<br>Bewertung .....                                      | 18 |
| Abbildung 5: Kompetenzabfrage des Projektes FÖRAK (vgl. BERBEN et al.<br>2003: 49) .....                                 | 20 |
| Abbildung 6: Anwendungsfalldiagramm der Web-Applikation .....  | 22 |
| Abbildung 7: Aktivitätsdiagramm des Anwendungsfalls „Klassensatz<br>anlegen“ .....                                       | 23 |
| Abbildung 8: Das interpretierte Login Skript der Web-Applikation .....   | 25 |
| Abbildung 9: Projektablauf der Web-Applikation zur Fremd- und<br>Selbstbewertung von Kompetenzen .....                   | 26 |
| Abbildung 10: Stammbaumarchitektur der Web-Applikation (Anhang B).....   | 26 |
| Abbildung 11: Aufruf einer statischen Webseite (vgl. LORENZ 2002: 42) .....  | 31 |
| Abbildung 12: Aufruf einer dynamischen Webseite (vgl. LORENZ 2002: 43) .....   | 31 |
| Abbildung 13: Beispiel eines Testskriptes für PHP (vgl. KOFLER 2003: 79f.) .....   | 34 |
| Abbildung 14: Struktur eines Datenbanksystems (vgl. BALZERT 2000: 720).....  | 35 |
| Abbildung 15: Die Datenbank der Web-Applikation im ER-Modell .....   | 37 |
| Abbildung 16: Login für Schüler der Web-Applikation .....  | 46 |
| Abbildung 17: Entitäten, Relationen und Attribute der Kompetenz-Datenbank.....   | 47 |
| Abbildung 18: MySQL Oberfläche des phpMyAdmin-Tools.....   | 47 |

## Einleitung

Durch einen stetigen Wandel der beruflichen Facharbeit, der sich aus neuen Arbeitsorganisations- und Produktionsformen sowie immer kürzer werdenden Technologiezyklen begründet, ist die berufliche Bildung in besonderer Weise herausgefordert. Statt auf die vielfach getrennte Vermittlung von Fertigkeiten und Kenntnissen zu setzen, streben zeitgemäße Konzepte der beruflichen Bildung die ganzheitliche Förderung beruflicher Handlungskompetenz an.

Handlungskompetenz schließt als Kern fachliche, methodische, soziale und individuelle Kompetenzen ein. Dieses Ziel soll durch eine handlungs- bzw. arbeitsprozessorientierte Lehr-Lerngestaltung erreicht werden. An dieser didaktischen Ausrichtung soll sich mit der Einführung der lernfeldorientierten Rahmenlehrpläne seit 1997 auch der Unterricht in der Berufsschule orientieren (vgl. KMK 2000). Die Förderung von Kompetenzen zieht jedoch eine veränderte Form der Leistungsbeurteilung nach sich. Der Bewertungsschwerpunkt teilt sich nicht nur in fachliche und soziale Normen auf, vielmehr werden individuelle Leistungen im Sinne einer Handlungskompetenz ganzheitlich berücksichtigt.

Bewertungskriterien werden in den Konzepten der ganzheitlichen Leistungsbeurteilung für handlungsorientierten Unterricht gemeinsam von Schülern und Lehrern erarbeitet und dienen dann als Grundlage für die prozessbegleitende Fremd- und Selbstbewertung in der Lerngruppe. Eine mehrfach durchzuführende Bewertung innerhalb einer Unterrichtssequenz erfordert einen hohen Koordinationsaufwand und produziert große Datenmengen, die in der Unterrichtspraxis von Hand nur schwer zu erfassen und auszuwerten sind.

In dem BLK-Projekt FÖRAK „Förderung arbeitsprozessbezogener Kompetenzen“ (vgl. BERBEN et al. 2003: 47 ff.) ist daher eine Software zur Unterstützung der Fremd- und Selbstbewertung von Kompetenzen in handlungsorientierten Unterrichten entwickelt worden, die den Aufwand deutlich reduziert. Diese speichert die anfallende Datenmenge, verwaltet die Dokumentation der Beurteilung, wertet die Daten aus und bereitet die Ergebnisse einer Bewertung für den Einsatz im Unterricht auf. Die Software dient als Hilfsmittel für die Bewertung, weist aber erhebliche Mängel in der Unabhängigkeit der Daten und im Unterrichtseinsatz auf. Als Verbesserung wurde von den Partnern des Projekts FÖRAK daher die Entwicklung einer Online-Datenbank zur Unterstützung der Fremd- und Selbstbewertung von Kompetenzen angeregt. Die Anforderungen zur Verbesserung der Software haben allerdings gezeigt, dass eine Neuentwicklung der Datenbank notwendig ist und somit die gesamte Software neu zu konzipieren ist.

In dieser Arbeit werden daher die Konzeption und die Entwicklung einer Web-Applikation zur Unterstützung von Fremd- und Selbstbewertung nach Grundlagen der Software-Technik vorgestellt.

# 1 Ziel und Aufbau der Arbeit

## 1.1 Ziel der Arbeit

Die Konzeption und die Entwicklung einer Web-Applikation sind Ziele der hier vorgestellten Arbeit, die zur Unterstützung der Fremd- und Selbstbewertung von Kompetenzen in handlungsorientierten Unterrichten eingesetzt wird. Als Web-Applikation wird ein in sich abgeschlossenes Produkt bezeichnet, das unterschiedliche Techniken einsetzt, um Anwendungen an das Internet anzubinden. Dazu gehört neben der Bereitstellung von Internetseiten über einen *Webserver*<sup>1</sup> auch die Anbindung an eine *Datenbank* zur Speicherung und Verwaltung anfallender Datenmengen.

Die Web-Applikation wird im Rahmen der Arbeit nicht fertig gestellt sondern analytisch nach Kriterien der Software-Technik entwickelt. Dazu gehören Verfahren der Prozessmodellierung und der Dokumentation bei Software-Entwicklung, sowie die Einteilung des Projektes in Phasen.

## 1.2 Aufbau der Arbeit

In dem ersten Teil der Arbeit werden zwei klassische Prozessmodelle der Software-Entwicklung vorgestellt. Unterschiedliche Prozesse stellen unterschiedliche Anforderungen an die Entwicklung einer Web-Applikation, sowohl aus der Sicht des Entwicklers als auch des Auftraggebers. Ein Vergleich der Prozessmodelle zeigt auf, welche Vor- und Nachteile bei der Entwicklung auftreten. Die Kombination beider Prozessmodelle wird für die Entwicklung der Web-Applikation angewandt und wird im Anschluss an den Vergleich vorgestellt - das Phasen-Modell mit *Prototyping*.

Der zweite Teil der Arbeit orientiert sich an Planungs-, Definitions-, Entwurfs- und Implementierungsphasen des Phasen-Modells mit *Prototyping*. Die Web-Applikation zur Unterstützung der Fremd- und Selbstbewertung wird anhand dieser Phasen beschrieben und es wird im Einzelnen auf die Konzeption, die Entwicklung, die Anforderungen und auf die Struktur des Produkts eingegangen:

- Planungsphase:  
Bevor mit der Entwicklung der Web-Applikation begonnen wird, ist es wichtig, eine Anforderungsdefinition zu erstellen und die Relevanz der Web-Applikation für ganzheitliche Leistungsbeurteilung nachzuweisen.

---

<sup>1</sup> Die in dieser Arbeit verwendeten Abkürzungen, Akronyme oder IT- Begriffe werden „*kursiv*“ dargestellt und anhand eines Glossars (vgl. Kapitel 6) beschrieben.

Das Potenzial, die Innovation, wichtige Eckdaten, der Entwicklungsrahmen und die Erfolgsfaktoren der zu entwickelten Software sind einzuschätzen. In dieser Phase wird ein Lastenheft angelegt, das im späteren Verlauf zum Pflichtenheft als Grundlage zur Umsetzung der Software wird.

- **Definitionsphase:**  
In der Definitionsphase wird das Grundgerüst für die Entwicklung gelegt. Hier werden die Anforderungen, die sich aus dem Lastenheft und den Analysen ergeben, in einem Produktmodell konkretisiert. Die Anforderungen werden ermittelt, beschrieben, als fachliche Lösung modelliert und analysiert. Die Ergebnisse aus der Definitionsphase bestehen meist aus einem Pflichtenheft, einem Produktmodell und einem Oberflächen-Prototypen.
- **Entwurfsphase:**  
In dieser Phase wird das so genannte Design oder die Software-Architektur entwickelt. Es wird dargestellt, mit welchen Techniken gearbeitet wird und wie diese eingesetzt werden. Entwürfe für Benutzeroberflächen, die eigentliche Anwendung und die Datenhaltung werden erstellt. Die Modellierung der *Datenbank* geschieht auch in der Entwurfsphase, hier wird in Modellen das Design der *Datenbank* erfasst. Die Entwurfsphase wird auch als „Programmieren im Groben“ bezeichnet.
- **Implementierungsphase:**  
In der Implementierungsphase werden die Programmieraktivitäten durchgeführt. Da die Entwicklung der Web-Applikation nicht ganz abgeschlossen ist, werden in dieser Phase die ersten Skripte der Implementierung und die Benutzerführung des Prototypsvorgestellt.

Das letzte Kapitel zeigt einen Ausblick auf die abschließende Testphase der Entwicklung, die im Rahmen dieser Arbeit noch nicht behandelt werden kann. Nach Fertigstellung der Implementierungsphase wird die Web-Applikation in einer Test- und Überprüfungsphase auf ihre Einsatzfähigkeit evaluiert und abschließend dokumentiert. Als Abschluss der Arbeit wird der Prototyp der Web-Applikation auf einem Web-Server unter folgender Adresse zur Verfügung stehen: <http://www.kompetenzbewertung.de>

## 2 Modelle der Software-Entwicklung

In diesem Abschnitt werden zwei klassische Prozessmodelle der Software-Entwicklung, die Elemente und Phasen des Arbeitsablaufs bei Entwicklungsverfahren darstellen, kurz vorgestellt. Durch Gegenüberstellung der Prozessmodelle ergeben sich Vor- und Nachteile bei der Entwicklung. Daher wird abschließend das Phasen-Modell, nach dem die Web-Applikation zur Unterstützung der Fremd- und Selbstbewertung entwickelt wird, als Kombination beider Modelle vorgestellt.

### 2.1 Die Wasserfall-Methode

Modelle der Software-Entwicklung zeigen den Ablauf und die Konzeption ganzer Projekte, es werden allgemeine und formale Prozesse der Software mit Hilfe von Diagrammen abstrahierend dargestellt.

Die traditionelle Methode, Software-Anwendungen zu entwickeln, ist eine Entwicklung in Stufen. Die Wasserfall-Methode erhält ihren Namen von einer visuellen Interpretation des Prozesses. Ein Wasserfall beginnt an einer Stelle, fällt auf eine zweite Stufe, dann auf eine dritte, vierte usw., kommt aber nie zur Anfangsstufe zurück. Diese Methode wird als Wasserfall-Methode bezeichnet, weil ein Entwicklungsprojekt in einer Reihe aufeinander folgender Entwicklungsphasen unterteilt wird. Die üblichen Phasen sind dabei Planung, Definition, Entwurf, Implementierung, Testen und Einsatz. Eine neue Phase zu beginnen, ist in der Wasserfall-Methode nur erlaubt, wenn die vorausgehende Phase erfolgreich abgearbeitet worden ist.

Auf dem Weg zu einem besseren Entwicklungsprozess wurde 1970 von Royce ein Vorläufer der Wasserfall-Methode (stagewise model) vorgestellt. Verbreitung und Namensgebung erfuhr die Wasserfall-Methode dann 1976 durch Boehm (vgl. BOEHM 1983).

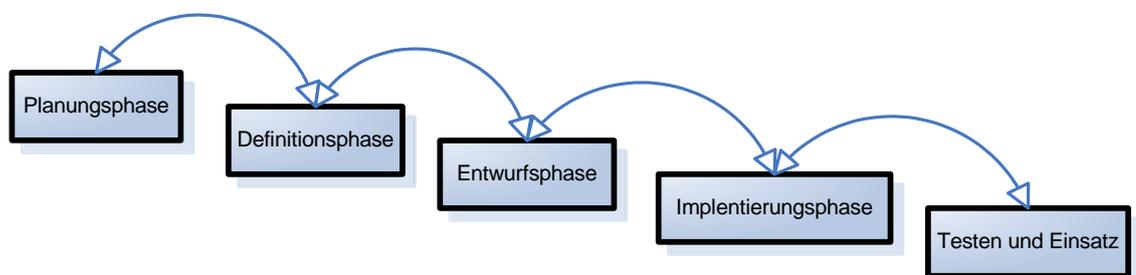


Abbildung 1: Das Wasserfallmodell nach Boehm (vgl. BOEHM 1981: 30)

Die Bedeutung der Wasserfall-Methode liegt darin, dass der Entwicklungsprozess in logisch aufeinander aufbauende Phasen strukturiert wurde. Jede Phase muss vollständig abgeschlossen sein, bevor die nächste begonnen werden kann (Abwärtspeile). Durch eine Ergebnissicherung in Form von unterschiedlichen

Dokumenten wird die Phase protokolliert. Die Kette der Aufwärtspfeile beschreibt die Überprüfung der Phasen, so dass das geschaffene System die gewünschten Anforderungen erfüllt.

Mit dem Erscheinen neuer, objektorientierter Programmiersprachen wie z.B. "Smalltalk" und "Java" wurde die Wasserfall-Methode von vielen Anwendungsentwicklern zugunsten eines *iterativen* Programmentwicklungsverfahrens weiterentwickelt. Diese iterativen Modelle werden häufig als "Spiralmodell" und "Fontänenmodell" bezeichnet. Bei diesen Verfahren wird mit einem ersten Entwurf in Form eines Prototyps für ein Programmsystem begonnen. Dieser Entwurf wird dann diskutiert und in mehreren Iterationsschritten verfeinert, und zwar solange, bis er allgemein akzeptiert wird.

## 2.2 Das Spiral-Modell

Die Vorgehensweise nach dem Spiral-Modell folgt einer risikoorientierten Strategie. An einer festen Position stehen im Spiral-Modell Aktivitäten, die den Umgang mit Risiken zum Inhalt haben. Die Reihenfolge der eigentlichen Entwicklungsschritte ist von äußeren Risiken bestimmt. Die *Iterationen* des Spiral-Modells haben das Ziel, Risiken zu minimieren und die Entwicklung dabei einen Schritt voran zu bringen. Der Radius der Spirale ist ein Maß für die aufgelaufenen Kosten und der Winkel gibt den Fortschritt innerhalb der vier Sektoren an.

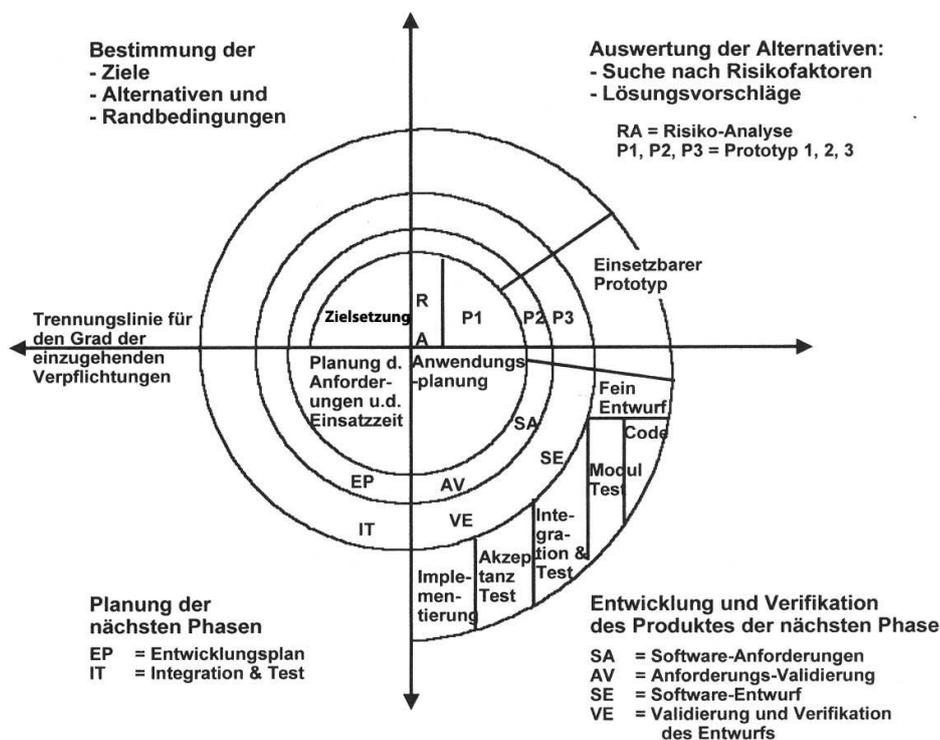


Abbildung 2: Das Spiral-Modell (vgl. SOMMERVILLE 2001: Kap.1)

Eine Iteration beginnt im linken oberen Sektor mit der Zielsetzung für den folgenden Umlauf sowie mit der Betrachtung von Lösungsalternativen und Einschränkungen.

kungen. Daran schließt sich eine Risikoanalyse der erwogenen Lösungsalternativen an, bei der unter Umständen ein Prototyp entwickelt wird, um letzte Unsicherheiten auszuschließen. Nun folgt die Entwicklung des geplanten Teilprodukts unter dem Vorzeichen eines minimierten Risikos. Die Auswertung des erzielten Ergebnisses und die Planung weiterer Phasen schließen eine Iteration ab.

Für den Einsatz dieser Entwicklungsmodelle werden folgende Vor- und Nachteile gesehen (vgl. SOMMERVILLE 2001: Kap.1):

|           | Wasserfall-Methode  | Spiral-Modell   |
|-----------|---|---|
| Vorteile  |   |   |
|           | Die Festschreibung der Anforderungen zu Beginn des Projekts ermöglicht beim Auftraggeber-Auftragnehmer-Verhältnis eine klare Aufgabendefinition.  | Die Entwicklung eines Prototyps gibt eine Übersicht über das Produkt. Der Auftraggeber erhält dadurch einen frühen Einblick in die Entwicklung des Produktes. Änderungswünsche können eher berücksichtigt werden. |
|           | Die einzelnen Phasen werden nach Abschluss dokumentiert.  | Nach jeder Iteration wird erneut eine Analyse durchgeführt.   |
|           | Mit Hilfe dieses Modells kann eine so genannte „Meilenstein-Planung“ gut durchgeführt werden.   | Kosten für die Entwicklung werden durch frühe Änderungen gering gehalten.   |
| Nachteile |   |   |
|           | Die entstehenden Dokumente sind ihrer Natur nach passiv. Sobald sie fertig gestellt sind, nehmen sie auf den weiteren Projektverlauf kaum noch steuernd Einfluss. Sie beantworten nicht - wie z. B. ein Prototyp oder eine Risikoanalyse - die Frage "Was sollen wir als nächstes tun?" | Die Zeitplanung ist schwierig, da Projekte dazu tendieren gesetzte Entwicklungspläne signifikant zu überschreiten. Es entsteht ein hoher Managementaufwand, der sich für kleinere Projekte nicht lohnt.           |
|           | Das frühe Festschreiben der Anforderungen ist kritisch, da Änderungen umso teurer werden, je weiter sie sich auf die früheren Phasen auswirken.   | Der Kunde erhält frühzeitig einen Prototyp. Das Projekt könnte vorzeitig beendet werden, indem der Kunde zu anderen Entwicklern wechselt.   |
|           | Die Einführung des Systems erfolgt im Entwicklungszyklus sehr spät.   | Es scheint einen schweren Konflikt zwischen wünschenswertem Zeitpunkt der Markteinführung und Produktakzeptanz zu geben.  |

Tabelle 1: Vor- und Nachteile der Wasserfall-Methode und des Spiral-Modells

Der abgestufte Entwicklungsansatz der Wasserfall-Methode kann in der Regel nicht so schnell auf Anforderungsänderungen reagieren, die die Produktakzeptanz

tanz berühren, wie das eine rein iterative Entwicklung könnte. Daher muss ein Kompromiss zwischen gewünschtem Markteinführungszeitpunkt, Produkterscheinung und Produktfunktionalität gefunden werden.

### 2.3 Das Phasen-Modell mit Prototyping

Ein sinnvolles Entwicklungsmodell wäre daher eine Kombination von abgestufter Entwicklung gestützt durch eine "sinnvolle Wasserfall-Methode" mit einer teilweise iterativen Entwicklung. Für die Entwicklung der Web-Applikation zur Unterstützung der Fremd- und Selbstbewertung wird daher in Anlehnung an die Wasserfall-Methode ein Phasen-Modell mit Prototyping angewendet, das den Entwurf eines Prototyps beinhaltet. Dieser Prototyp ist ein Oberflächen-Prototyp und dient der besseren Erkennung der Anforderungen, wird aber für den Projekt-Entwurf nicht weiter benutzt.

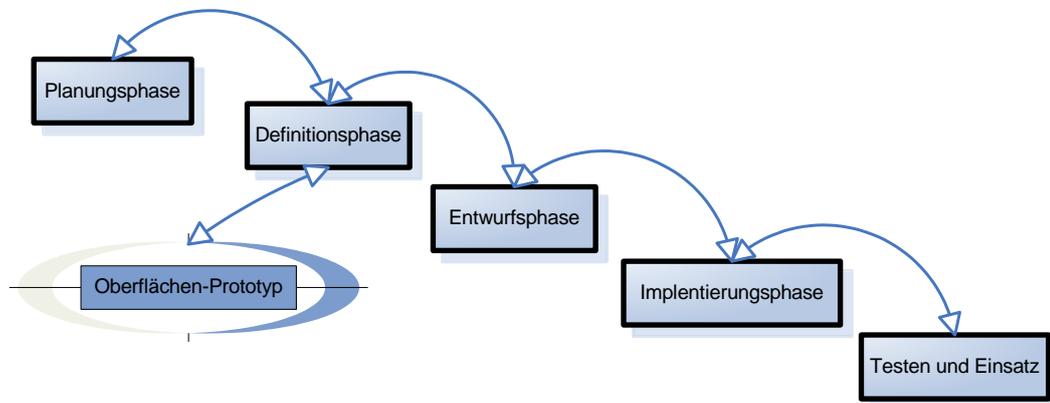


Abbildung 3: Das Phasen-Modell mit Prototyping als Entwicklungskonzept der Web-Applikation (vgl. BOEHM 1981: 38)

Durch die Einführung eines Prototyps in den strukturellen Ablauf des Phasen-Modells ergeben sich folgende Vor- und Nachteile:

|          | Phasen-Modell mit Prototyping   |
|----------|---|
| Vorteile |   |
|          | Reduzierung des Entwicklungsrisikos                                     |
|          | Prototypen können mit Hilfe einfacher Werkzeuge schnell erstellt werden |
|          | Ein Prototyp verbessert die Planung von Software-Entwicklungen          |
| Nachteil |   |
|          | höherer Entwicklungsaufwand   |

Tabelle 2: Vor- und Nachteile des Phasen-Modells

## **3 Konzeption und Entwicklung der Web-Applikation**

### **3.1 Die Planungsphase**

Das für die Entwicklung der Web-Applikation zur Fremd- und Selbstbewertung von Kompetenzen eingesetzte Verfahren ist das in Anlehnung an die Wasserfall-Methode entstandene Phasen-Modell. Diese Phasen beginnen grundsätzlich mit der Planung. Folgende Ergebnisse sind für eine Planungsphase charakteristisch (vgl. BALZERT 2000: 58-96):

- Durchführbarkeitsstudie,
- Lastenheft,
- Wirtschaftlichkeitsprüfung.

Bei einer Durchführbarkeitsstudie werden ökonomische, fachliche sowie personelle Faktoren berücksichtigt. Das Risiko einer Entwicklung wird festgestellt, und Alternativen, z. B. die Anpassung von Standardsoftware, werden diskutiert.

Das Lastenheft dokumentiert Adressaten, Inhalt und Wünsche der Anforderungen des Projektes sowie den Rahmen der Web-Applikation. Es dient als Grundlage für das Pflichtenheft der Definitionsphase und wird daher als grobes Pflichtenheft bezeichnet.

Eine Wirtschaftlichkeitsprüfung ist aufgrund des Projektumfangs zu vernachlässigen.

Die Durchführbarkeitsstudie für die Web-Applikation zur Fremd- und Selbstbewertung beschreibt die fachliche Relevanz eines Softwareeinsatzes. Das Lastenheft dokumentiert den Rahmen der Arbeit und stellt die Anforderungen des Auftraggebers dar.

#### **3.1.1 Projektbezogene Anforderungen an die Software**

##### **3.1.1.1 Kompetenzdimensionen im handlungsorientierten Unterricht**

In der beruflichen Bildung spielen technische Innovationen eine immer größere Rolle. Die Innovationszyklen verkürzen sich in rasantem Tempo. In diesem Prozess der permanenten Erneuerung von Produkten, Fachinhalten und Arbeitstechniken sind Informationen und Wissen die entscheidenden Wertschöpfungspotentiale.

Informationen und Wissen haben in der beruflichen Bildung in ihrem Bezug zur praktischen Tätigkeit einen hohen Stellenwert, wobei Wissen stets an Personen

gebunden ist. Die Ausprägung der Fähigkeiten zur Aufnahme, zur Verarbeitung und zur Nutzung von Informationen und Wissen gibt den Ausschlag für die dadurch generierbaren beruflichen Lernerfolge. Diese Fähigkeiten und persönlichen Qualifikationen herauszubilden, ist eine Kernaufgabe der beruflichen Bildung (vgl. BADER 1989: 73-77).

In Anbetracht der zunehmenden Dynamik des heutigen Arbeitslebens ist jedoch die Betrachtung der Qualifikation im klassischen Sinne allein nicht mehr ausreichend. Ohne die wichtige Rolle der Basisqualifikation durch berufliche Aus- und Weiterbildung in Frage zu stellen, ist es erforderlich geworden, die individuellen Voraussetzungen für prozessbezogene Tätigkeiten auch aus einer anderen Perspektive zu betrachten. Dabei steht der Begriff Handlungskompetenz im Mittelpunkt.

Laut Handreichung der Kultusministerkonferenz (vgl. KMK 2000) wird Handlungskompetenz als die Bereitschaft und Fähigkeit des Einzelnen, sich in beruflichen, gesellschaftlichen und privaten Situationen sachgerecht durchdacht sowie individuell und sozial verantwortlich zu verhalten, verstanden.

Durch die Einteilung in Kompetenzen wird das konkrete Handeln in den Mittelpunkt der Betrachtung gestellt und geht somit weit über bildungsrelevante Persönlichkeitsmerkmale hinaus. Ausgehend von den unterschiedlichen Handlungsarten werden Kompetenzen wie folgt klassifiziert (vgl. BADER 1989: 73-77):

- Fachkompetenzen werden mit den selbst organisierten Problemlösungs-, Denk- und Wertungsprozessen verbundenen "fachlichen Handlungen" zugeordnet,
- Methodenkompetenzen beziehen sich auf die instrumentellen Handlungen im Zusammenhang mit manuellen Verrichtungen,
- Sozialkompetenzen entsprechen den kommunikativen und kooperativen Handlungen,
- Individualkompetenzen werden den reflexiven Handlungen (Selbstreflexion auf der Grundlage von Werten und Motivationen) zugeordnet.

Handlungskompetenz steht für die Integration der vorgenannten Kompetenzen und bezieht sich auf das gesamte Spektrum des selbst organisierten Lernens und Handelns. Im Vordergrund der Beurteilung von Kompetenzen stehen nicht die unmittelbaren Lernresultate einer Qualifizierung oder eines Lernprozesses, sondern die Fähigkeiten zur Übertragung der Lernresultate auf den Arbeitsprozess.

### 3.1.1.2 Phasen einer ganzheitlichen Leistungsbeurteilung

Um die Ausprägung von Kompetenzen von Berufsschülern einschätzen zu können, muss überprüft werden, welche Befähigungen die Schüler besitzen, Probleme zielorientiert und kooperativ innerhalb variierender Rahmenbedingungen

bestmöglich zu lösen. Damit wird weitgehend den tatsächlichen Bedingungen der praktischen Tätigkeit in beruflicher Realität entsprochen, in der Entscheiden und Handeln in nicht vorhersehbaren Situationen häufig vorkommen.

Mit der Betrachtung von Kompetenzen ist das Ziel verbunden, die Arbeitsprozesse über das klassische Verständnis von Qualifikation hinausgehend als Lernprozesse kommunizierender und kooperierender Schüler und Gruppen zu verstehen, in denen sich Fähigkeiten zur Selbstorganisation, -steuerung und -optimierung herausbilden.

In der ganzheitlichen Leistungsbeurteilung sind deshalb die Formen der Lernzielkontrolle von engen behavioristischen Lernvorstellungen abzulösen, denn übergeordnete Zielstellungen lassen sich im Lernprozess kaum mit standardisierten Tests überprüfen. Der Ablauf einer Beurteilung von Kompetenzen bildet charakteristische Phasen, die innerhalb der Gestaltung von Unterrichtssequenzen zu berücksichtigen sind. Schüler sollen demnach (vgl. OTT 2000: 219-223)

- zu Lernbeginn (in einem Lernkontrakt) präzise über die intendierten (übergeordneten) operationalen Lernziele und Beurteilungskriterien informiert werden, respektive diese Ziele selbst planen, und so die Beurteilungskriterien mitentscheiden,
- im Lernprozess gezielt, d.h. nach vorstrukturierten und mitbestimmten Kriterien beobachtet werden (Fremdbeurteilung) bzw. diese Verlaufsbeobachtung selbst mit durchführen (Selbstbeurteilung),
- in der Rückkoppelung ihren Arbeits-, Lern- und Gruppenprozess gezielt reflektieren (Gruppenbeurteilung) und in einem Fördergespräch selbst (mit-) bewerten.

Dies ist eine Ablaufstruktur, die sich in vielen ganzheitlichen Bewertungskonzepten wieder findet. Die Handreichung des Modellversuchs SELUBA (vgl. RICHTER 2002: 20-38) sowie das Bewertungskonzept von FÖRAK (vgl. BERBEN et al. 2003: 47 ff.) weisen auf einen Einsatz ganzheitlicher Leistungsbeurteilung in der beruflichen Bildung.

### 3.1.1.3 Fremd- und Selbstbewertung von Kompetenzen

Bei ganzheitlicher Leistungsbeurteilung steht die eigene Einschätzung des Lernhandelns vom Schüler im Vordergrund. Neben der „objektiven“ Fremdbewertung durch die Lehrenden ist es somit angeraten, die „subjektive“ Selbsteinschätzung der Lernenden mit in die Leistungsbeurteilung einzubeziehen. Neben dem zusätzlichen Informationsgewinn wird mit der Einbeziehung der Selbsteinschätzung der Lernenden eine grundlegende pädagogische Forderung an den handlungsorientierten Unterricht erfüllt: Eine offene und transparente Kommunikation zwischen Lernenden und Lehrenden (vgl. RICHTER 2002: 38).

Fremd- und Selbstbewertung bedeutet in diesem Zusammenhang, dass die Bewertung von Schülern und Lehrern gemeinsam vollzogen wird. Die Kriterien für die Bewertung werden aus den Kompetenzdimensionen entwickelt bzw. diesen zugeordnet. Die Schüler haben nun die Möglichkeit anhand dieser Bewertungskriterien sich „selbst“ zu bewerten und im Gegenzug ihre Partner- bzw. Gruppenmitglieder „fremd“ zu bewerten. Der Lehrer bewertet die Schüler aus seiner Sicht, also ebenfalls „fremd“. Durch Zusammenfassung dieser Bewertungsergebnisse können so objektivere Resultate gefunden werden.

#### 3.1.1.4 Relevanz einer Bewertungssoftware

Ein Bewertungsablauf stellt folgende Anforderungen an das Bewertungsmedium:

- Große Datenmengen müssen erfasst und ausgewertet werden.
- Die Daten müssen gespeichert und verwaltet werden.
- Die Pflege und Eingabe der Daten muss einfach, aber trotzdem sicher konzipiert werden.
- Die Daten müssen flexibel zur Verfügung stehen, es muss von den Teilnehmern gleichzeitig und von verschiedenen Orten zugegriffen werden.

Diese Faktoren sprechen für den Einsatz einer internetbasierten Lösung als Bewertungsmedium. Im Lastenheft werden noch weitere Anforderungen an die Entwicklung der Web-Applikation zur Unterstützung der Fremd- und Selbstbewertung von Kompetenzen formuliert.

### 3.1.2 Das Lastenheft der Web-Applikation

Das Lastenheft dient als Kommunikationsgrundlage zwischen dem Auftraggeber und dem Entwickler. Es beschreibt die Anforderungen, die der Auftraggeber an die Web-Applikation stellt. Diese Anforderungen werden in der Definitionsphase aufgenommen und in einem Pflichtenheft weiter verarbeitet. Die Fremd- und Selbstbewertung von Kompetenzen läuft nach vordefinierten Prozessschritten ab. Diese müssen zunächst erfasst werden und der entsprechende Rahmen der Web-Applikation muss herausgearbeitet werden.

#### 3.1.2.1 Zielbestimmung

Der Paradigmenwechsel in der beruflichen Bildung hin zu handlungs- und projektorientiertem Unterricht stellt neue Anforderungen an das Bewertungssystem. Es ist eine Web-Applikation zu erstellen, die eine Fremd- und Selbstbewertung von Kompetenzen unterstützt.

#### 3.1.2.2 Überblick über die Ziele

Der Auftraggeber verfolgt mit der Realisierung der Web-Anbindung die folgenden Ziele:

- Unabhängiger und flexibler Zugang zum Bewertungssystem,
- Speicherung der Bewertungsdaten auf einem Server,
- Einrichtung von Formularen, die die Informationsentnahme erleichtern,
- Einführung von Sicherheitsregeln für den Gebrauch der Datenbank.

### 3.1.2.3 Kurzdarstellung der Projektpartner

Auftraggeber für die Entwicklung des Bewertungsmediums ist die Universität Hamburg vertreten durch Herrn Prof. Dr. Thomas Vollmer, der das Projekt „FÖRAK- Förderung arbeitsprozessbezogener Kompetenzen zwischen den Lernorten Schule und Betrieb“ wissenschaftlich begleitet hat:

Prof. Dr. Thomas Vollmer  
Universität Hamburg - FB Erziehungswissenschaft  
Institut für Berufs- und Wirtschaftspädagogik  
Sedanstraße 19  
20146 Hamburg  
  
Tel.: 040-428 38 3740  
Fax: 040-428 38 6787  
E-mail: vollmer.thomas@erzwiss.uni-hamburg.de

Entwicklung:

Henning Klaffke  
Lisztstraße 20c  
22763 Hamburg  
  
Tel.: 040-411 629 25  
E-mail: h.klaffke@tu-harburg.de

### 3.1.2.4 Produktfunktionen

Die Software greift typische Arbeitsabläufe einer Bewertung auf und unterstützt deren Realisierung im Unterricht. Zur Erfassung dieser Arbeitsabläufe wurde von den Projektpartnern eine Mind-Map erstellt:

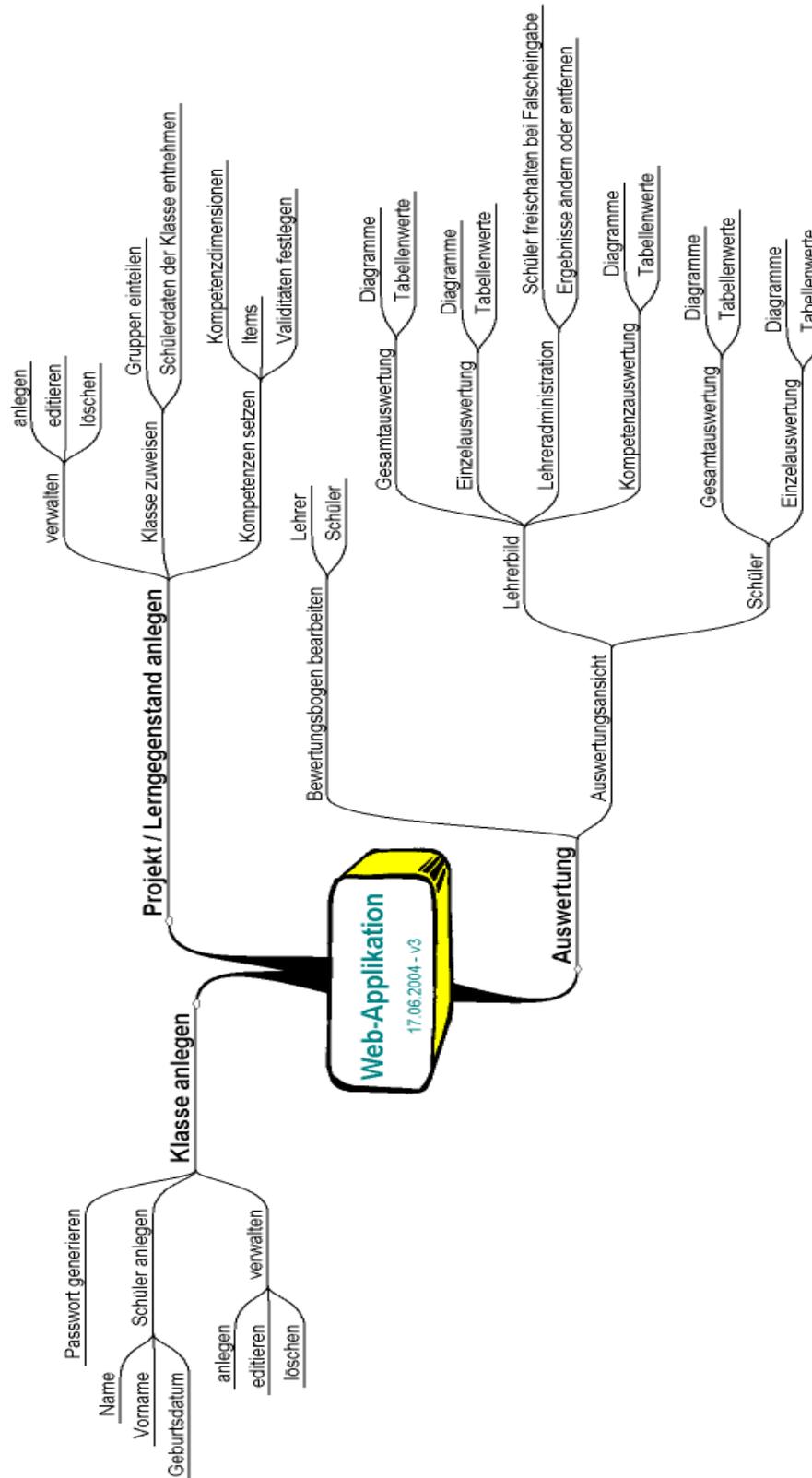


Abbildung 4: Mind Map der Web-Applikation zur Prozessanalyse der Bewertung

Die hier gesammelten Informationen sind noch nicht strukturiert, die Reihenfolge der Arbeitsabläufe ist nicht erkennbar. Deswegen werden so genannte Lastenheft-Funktionen (/LF/) generiert. Die Lastenheft-Funktionen werden in Zehnerschritten zur einfacheren Einpflege weiterer Ergänzungen angegeben (vgl. BALZERT 2000: 62):

- /LF10/ Ersterfassung, ändern und löschen von Klassensätzen,
- /LF20/ Ersterfassung, ändern und löschen von Schülerangaben,
- /LF30/ Dem Klassensatz Projekte zuordnen, editieren oder neu erstellen,
- /LF40/ Den Projekten Schülergruppen zuordnen,
- /LF50/ Die Kompetenzdimensionen festlegen oder editieren,
- /LF60/ Bewertungsformulare ausfüllen,
- /LF70/ Anzeigen von Auswertungsdaten.

Zu diesem frühen Zeitpunkt der Entwicklung ist noch nicht abzusehen, ob diese Arbeitsabläufe vollständig durch Software realisiert werden können. Das Lastenheft dient dem Pflichtenheft in der Definitionsphase als Grundlage für die zu modellierenden Arbeitsabläufe.

#### 3.1.2.5 Anwendungsbereich

Die Web-Applikation zur Fremd- und Selbstbewertung von Kompetenzen wird vorwiegend in der beruflichen Bildung eingesetzt. Die Kriterien zur Kompetenzbewertung werden von Lehrern und Schülern gemeinsam im handlungs- und projektförmigen Berufsschulunterricht erarbeitet. Die Web-Applikation unterstützt die Eingabe der Kriterien und sorgt für eine Verarbeitung der Bewertungsinformationen. Alle Ergebnisse der Bewertung werden in diesem Medium erfasst, gespeichert und ausgewertet.

#### 3.1.2.6 Zielgruppen

Die Web-Applikation wird einerseits für Lehrkräfte in der beruflichen Bildung entwickelt, um die Bewertung in projekt- und handlungsorientierten Unterrichten zu unterstützen, andererseits richtet sich dieses Medium an Berufsschüler, die damit ihre Unterrichtserfahrungen dokumentieren und ihre Leistungen selbst und gegenseitig beurteilen können.

### 3.1.3 Ist-Analyse

Die Ist-Analyse skizziert den derzeitigen Stand der bereits bestehenden Software.

In dem BLK-Projekt FÖRAK „Förderung arbeitsprozessbezogener Kompetenzen“ (vgl. BERBEN et al. 2003: 47 ff.) ist eine Software zur Fremd- und Selbstbewertung von Kompetenzen in handlungsorientierten Unterrichten entwickelt worden, die die anfallende Datenmenge abspeichert, die Dokumentation der Beurteilung verwaltet und die Ergebnisse einer Bewertung tabellarisch ausgibt. Die Software dient als Hilfsmittel für die Bewertung, weist aber erhebliche Mängel in der Unabhängigkeit der Daten und im Unterrichtseinsatz auf. Sie besteht aus einer lokalen

Microsoft-Access-Datenbank, die mit Hilfe von Formularen und Abfragen die Bewertung speichert und ausgibt. In der folgenden Abbildung ist ein Formular zu sehen, das in ähnlicher Form in der Web-Applikation umzusetzen ist:

The screenshot shows a web application interface for a competency assessment. The title bar reads 'Schüler'. Below it are four tabs: 'Fachliche Qualifikation', 'Methodenkompetenz', 'Lernkompetenz', and 'Sozialkompetenz'. The main content area is divided into three sections, each with a heading and a table of student responses.

**Selbständigkeit**  
 ... habe/hat selbstständig geplant und die Maßnahmen durchführt

|        |        | stimmt                   |                          |                                     |
|--------|--------|--------------------------|--------------------------|-------------------------------------|
|        |        | überhaupt nicht          |                          | vollkommen                          |
| Fabian | Frisch | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Maria  | Meyer  | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Ich    |        | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Toni   | Turek  | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |

**Alternativen**  
 ... habe/hat Alternativmaßnahmen selbst erarbeitet und bewertet

|        |        | stimmt                   |                          |                                     |
|--------|--------|--------------------------|--------------------------|-------------------------------------|
|        |        | überhaupt nicht          |                          | vollkommen                          |
| Fabian | Frisch | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Maria  | Meyer  | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| ich    |        | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Toni   | Turek  | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |

**Thementreue**  
 ... weiche/weicht nicht vom eigentlichen Thema ab

|        |        | stimmt                   |                          |                                     |
|--------|--------|--------------------------|--------------------------|-------------------------------------|
|        |        | überhaupt nicht          |                          | vollkommen                          |
| Fabian | Frisch | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Maria  | Meyer  | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| ich    |        | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Toni   | Turek  | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |

On the right side of the form, there is a logo for 'Förak' featuring a gear and a lightbulb.

Abbildung 5: Kompetenzabfrage des Projektes FÖRAK (vgl. BERBEN et al. 2003: 49)

Die Datenbank lässt sich aufgrund ihres strukturellen Aufbaus nicht für die Konzeption der Web-Applikation verwenden. Für die Erweiterung der Software kann jedoch der inhaltliche Ablauf einer Bewertung als Orientierungsvorlage dienen.

### 3.1.4 Abschluss der Planungsphase

Den Abschluss der Planungsphase bildet die Zusammenfassung der erreichten Ergebnisse in dem Lastenheft (vgl. Kapitel 4.1.2 – 4.1.3):

- Die projektbezogenen Anforderungen einer Bewertung wurden erarbeitet,
- Ziel, Zielgruppen und Einsatz der Web-Applikation wurden genannt,
- Die Lastenheft-Funktionen wurden dargestellt.

Diese Ergebnisse bilden die Vorlage für die folgende Definitionsphase.

## 3.2 Die Definitionsphase

Zu Beginn der Definitionsphase sind die Eingaben aufgrund der vorwiegend ungenau erfassten Formulierungen des Lastenhefts und der gestellten Anforderungen vage, unvollständig und inkonsistent beschrieben. Ziel in der Definitionsphase ist es, als Ergebnis vollständige, erfüllbare und eindeutige Anforderungen zu erhalten. Diese Anforderungen werden durch folgende Dokumente gesichert (vgl. BALZERT 2000: 97-150):

- Pflichtenheft
- Produkt-Modell
- Produkt-Layout

### 3.2.1 Das Pflichtenheft der Web-Applikation

*„Ein Pflichtenheft bestimmt sich nicht durch den Umfang oder eine starre Struktur, sondern es übernimmt die Aufgabe, in Form einer Dokumentation, präzise die Lösungswege und Konzepte zu beschreiben, die zur Erreichung der in einem Lastenheft definierten Projektziele notwendig sind. Hierbei spielt die Erläuterung der Definition über Ziel- und Umsetzung die bedeutende Rolle. Ein Pflichtenheft muss gleichermaßen für fachlich korrekte Informationen sorgen, aber auch für den Auftraggeber verständlich sein.“*  
(RÜTTGER 2003: 376)

Das Pflichtenheft für die Web-Applikation zur Unterstützung der Fremd- und Selbstbewertung wird in den folgenden Unterkapiteln dargestellt. Die Aspekte hinsichtlich der Ziele, Zielgruppen und Auftraggeber sowie der grobe inhaltliche Ablauf der Bewertung sind aus dem Lastenheft bekannt und werden hier nicht mehr aufgegriffen. Das Pflichtenheft dokumentiert u. a. folgende Anforderungen:

### 3.2.1.1 Produkt-Modell

Der aus dem Lastenheft geforderte Bewertungsablauf wurde durch die Lastenheft-Funktionen dokumentiert. Diese Funktionen sind im Pflichtenheft aufgegriffen und in Form eines Produkt-Modells dargestellt. Das Produkt-Modell dient dazu, den Arbeitsablauf der Software zu skizzieren. Dabei können unterschiedliche Modellierungsverfahren eingesetzt werden. Der Arbeitsablauf der Web-Applikation zur Unterstützung der Fremd- und Selbstbewertung wird anhand eines Anwendungsfalldiagramms modelliert:

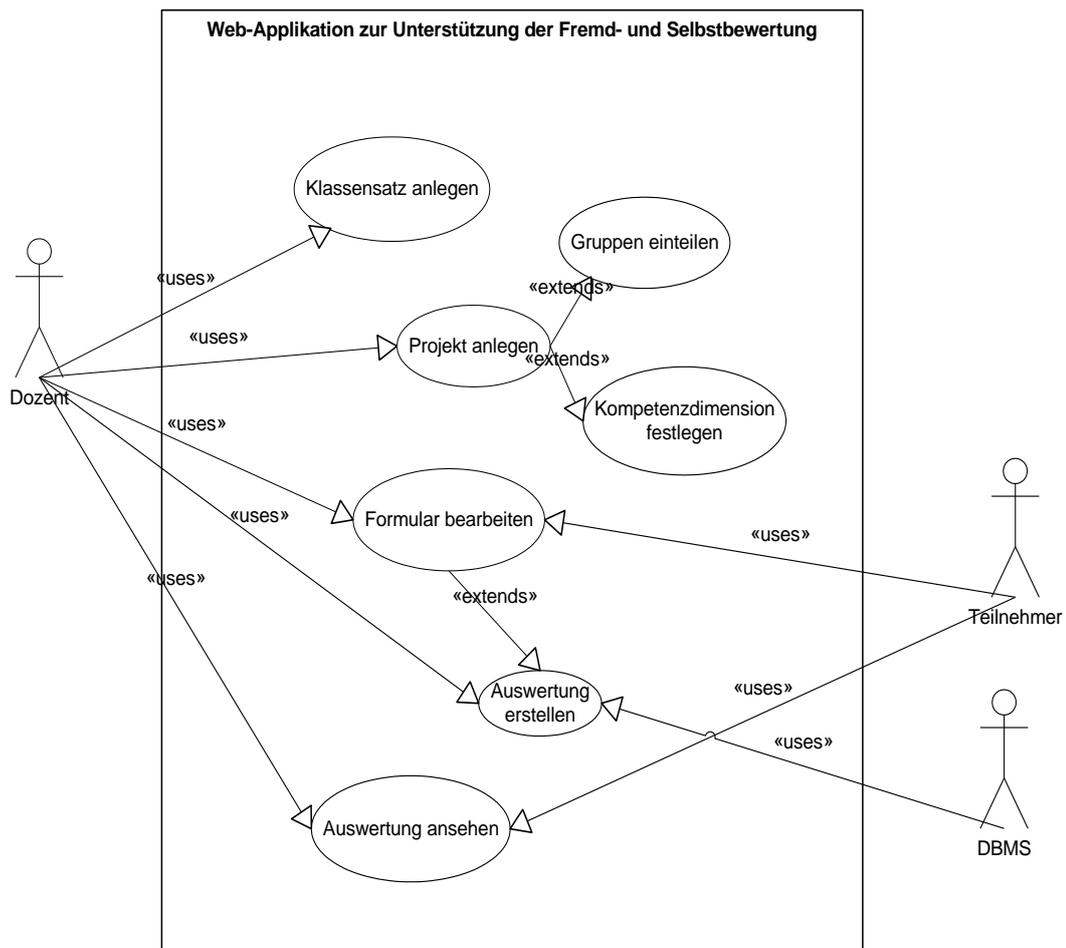


Abbildung 6: Anwendungsfalldiagramm der Web-Applikation

Ein Anwendungsfall wird in dem Diagramm durch eine Ellipse dargestellt, die den Namen des Anwendungsfalls trägt, und beschreibt eine typische Interaktion des Anwenders mit einem Computersystem. Durch das Diagramm wird das gewünschte externe Systemverhalten aus Sicht des Anwenders beschrieben und stellt somit Anforderungen auf, die das System erfüllen muss.

Anwendungsfälle sind ein Hilfsmittel zur Anforderungsanalyse im Rahmen eines objektorientierten Entwicklungsprozesses. Einzelne Schritte, Transaktionen oder Operationen werden aus einem Anwendungsfalldiagramm nicht deutlich. Ein Anwendungsfall ist die Beschreibung dessen, was ein System leisten muss, aber nicht, wie es dies leisten soll (vgl. OESTERREICH 2002: 5-14).

Ein Anwendungsfalldiagramm zeigt die Beziehungen zwischen den Anwendungsfällen und ihren Akteuren, in diesem Fall den Dozenten, den Teilnehmern, dem Datenbankmanagementsystem (DBMS), an. Auf das DBMS wird bei der Beschreibung der Entwicklungsumgebung (vgl. Kap. 6.1.2) eingegangen. Das System dient bei diesem Anwendungsfall ebenfalls als Akteur.

Das rechteckige Symbol steht für das Softwaresystem, in dem die Prozesse ablaufen. Wird der Verlauf der Prozesse betrachtet, werden die im Lastenheft bezeichneten Funktionen wieder aufgegriffen. Zusätzlich zu den Anwendungsfällen werden jedoch weitere Dokumentationen angefertigt.

### 3.2.1.2 Produkt-Funktion

Hinter jedem Anwendungsfall steht bei der Entwicklung der Web-Applikation ein Aktivitätsdiagramm und im Regelfall eine Dokumentation:

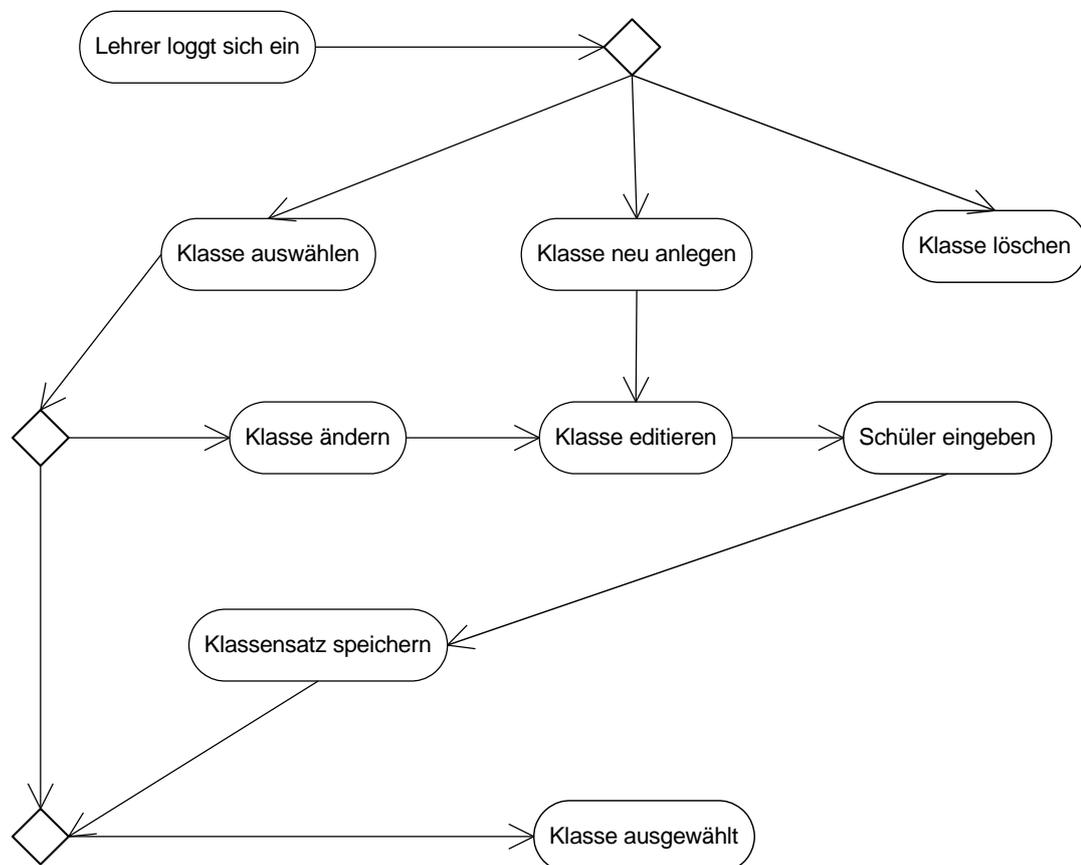


Abbildung 7: Aktivitätsdiagramm des Anwendungsfalles „Klassensatz anlegen“

Das dargestellte Aktivitätsdiagramm beschreibt die Aktivitäten hinter dem Anwendungsfall „Klassensatz anlegen“. In der Hierarchie untersteht das Aktivitätsdiagramm dem Anwendungsfalldiagramm und erläutert den genauen Ablauf dieser Sequenz. Anhand einer Dokumentationsvorlage werden die Aktivitäten genauer beschrieben. Diese Vorlage umfasst folgende Punkte, die je nach Einsatz variiert werden:

|                        |   |
|------------------------|---|
| Akteure:               | Lehrer  |
| Kurzbeschreibung:      | Der Lehrer erstellt einen Klassensatz. Er hat die Möglichkeit, alle eingegeben Daten bezüglich der Klassen- und Schülerdaten zu editieren.  |
| Auslöser/Vorbedingung: | Der Lehrer gibt die Daten über ein Eingabeformular in die Datenbank. Das Eingabeformular und die nötige Datenbank sind bereit zur Eingabe.  |
| Ergebnis:              | Der Lehrer hat einen oder mehrere Klassensätze erstellt.  |
| Essentieller Ablauf:   | <ol style="list-style-type: none"><li>1. Lehrer loggt sich auf der Seite ein.</li><li>2. Neuerstellung der Klassen.</li><li>3. Speicherung des Klassensatzes.</li><li>4. Klassensatz ist erzeugt.</li></ol> |

Alle zu den Anwendungsfällen aufgeführten Aktivitätsdiagramme und die Dokumentationen befinden sich im Anhang dieser Arbeit (Anhang A).

### 3.2.1.3 Das Produkt-Layout

Das Produkt-Layout soll die Anforderung an eine zielgruppengerechte Gestaltung digitaler Informationen erfüllen. Bei Administration und Ausgabe der Bewertung hat das Design für klare Strukturen zu sorgen und darf nicht überlastend wirken. Die informationshaltigen Seiten und Eingabemasken sind daher neutral gehalten und auf die Verwendung von vier Farben beschränkt.

Die Entwicklung des Produkt-Layouts bezieht sich auf das *Frontend* dieser Applikation. Mit der Entwicklung des Layouts werden die folgenden Ziele verfolgt (vgl. BALZERT 2000: 620 f.):

- Durchgängige Gestaltung als Wiedererkennungsmerkmal,
- Schaffung eines ausgewogenen Verhältnisses von Grafik und Text,
- Geringe Ladezeiten der Seiten durch professionelles Bearbeiten der Gestaltung,
- Unterschiedliche Farbhintergründe bei Bearbeitung durch Schüler und Lehrkräfte.

Die Umsetzung dieser Ziele wird durch folgendes Gestaltungskonzept realisiert:

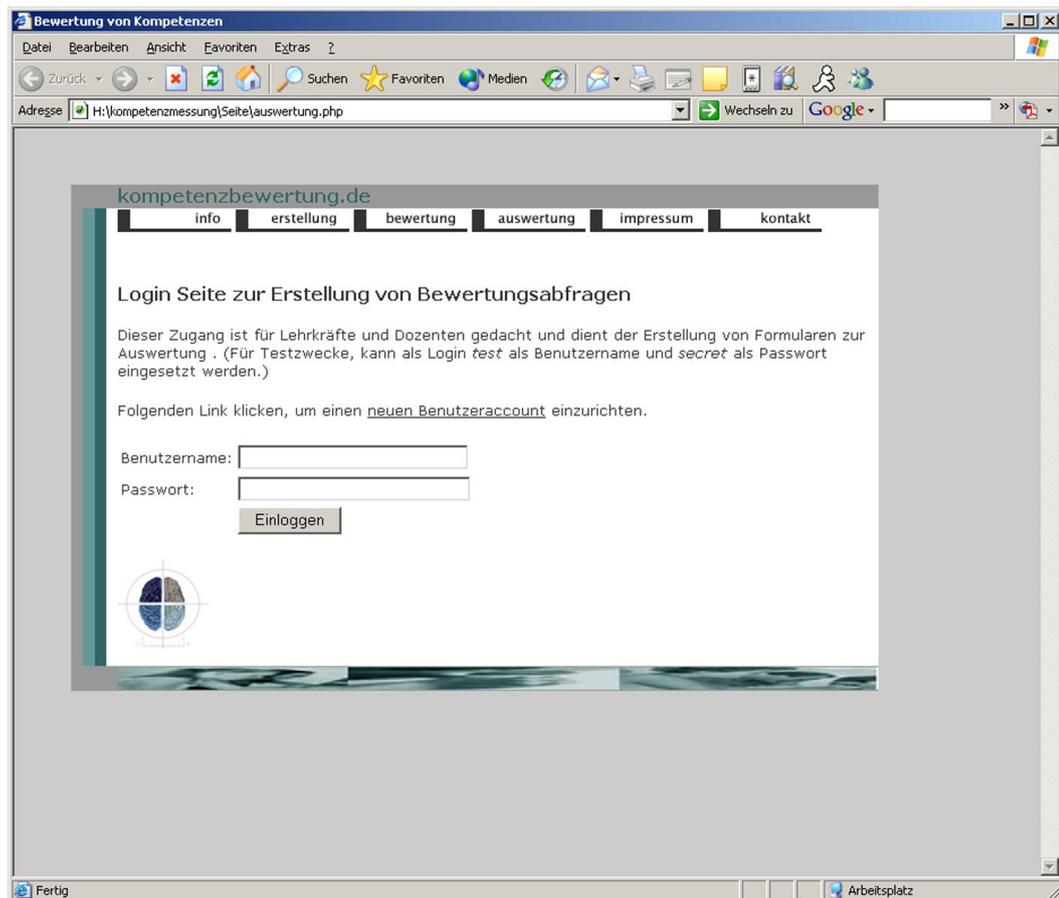


Abbildung 8: Das interpretierte Login Skript der Web-Applikation

Durch die Kombination der weißen Hintergrundfläche des Schriftfelds mit dem dunkleren Seitenhintergrund erhält die Website Räumlichkeit und Tiefe.

Die Farbkombination der Homepage wird auch auf den Folgeseiten angewendet. Im Vordergrund stehen die Navigationselemente. Die sachlich aufgebaute Informations- und Navigationsarchitektur wird durch das Layout-Design hervorgehoben.

Als Typografie der grafischen Elemente wird die Standardschrift „Verdana“ eingesetzt, um zu gewährleisten, dass jeder Benutzer diese Schrift im Browser sehen kann. Andere serifenlose Schriften wie z.B. Futura werden nicht auf jedem Rechner dargestellt und werden deshalb nicht verwendet. Weitere Layoutbeispiele sind im Anhang aufgeführt (Anhang B).

#### 3.2.1.4 Projekt-Ablauf

Die Web-Applikation kann bei fristgerechter Datenlieferung, die in den nachfolgenden Abschnitten definiert wird, am 05.07.2004 in einem Testbereich implementiert und am 28.07.2004 online geschaltet werden. Bei Bekanntgabe eines positiven Testergebnisses bis spätestens zum 13.08.2004 wird die Entwicklung in der ersten Version abgeschlossen sein.

Die folgende Übersicht dient allen Projektbeteiligten als Orientierungshilfe, die den zeitlichen Ablauf und die Aufgabenverteilung darstellt:

| K<br>e<br>n<br>n<br>u<br>n<br>g | Aufgabenname              | Anfang     | Abschluss  | Dauer | Jun <i>jjjj</i> |  |  | Jul <i>jjjj</i> |  |  | Aug <i>jjjj</i> |
|---------------------------------|---------------------------|------------|------------|-------|-----------------|--|--|-----------------|--|--|-----------------|
|                                 |                           |            |            |       |                 |  |  |                 |  |  |                 |
| 1                               | Planungsphase             | 01.06.2004 | 17.06.2004 | 2w 3t | █               |  |  |                 |  |  |                 |
| 2                               | Definitionsphase          | 14.06.2004 | 24.06.2004 | 1w 4t | █               |  |  |                 |  |  |                 |
| 3                               | Entwurfphase              | 21.06.2004 | 01.07.2004 | 1w 4t |                 |  |  | █               |  |  |                 |
| 4                               | Implementierungsp<br>hase | 05.07.2004 | 28.07.2004 | 3w 3t |                 |  |  | █               |  |  |                 |
| 5                               | Ausblick                  | 28.07.2004 | 16.08.2004 | 2w 4t |                 |  |  |                 |  |  | █               |

Abbildung 9: Projektablauf der Web-Applikation zur Fremd- und Selbstbewertung von Kompetenzen

Bei Projektabschluss wird eine Anwenderdokumentation mit Beschreibungen der *Templates*, technischen Besonderheiten und der Dateistruktur an den Auftraggeber übergeben. Die Dateihierarchie der *HTML*-Dateien wird durch einen Strukturbaum dargestellt.

### 3.2.1.5 Produkt-Daten

Die Produkt-Daten beschreiben das *Backend* der Web-Applikation zur Fremd- und Selbstbewertung. Das Produkt beinhaltet die Anbindung an eine Datenbank. Die Ordnungshierarchien der Webseiten und Skripte sind an der folgenden Stammbaumarchitektur zu erkennen:

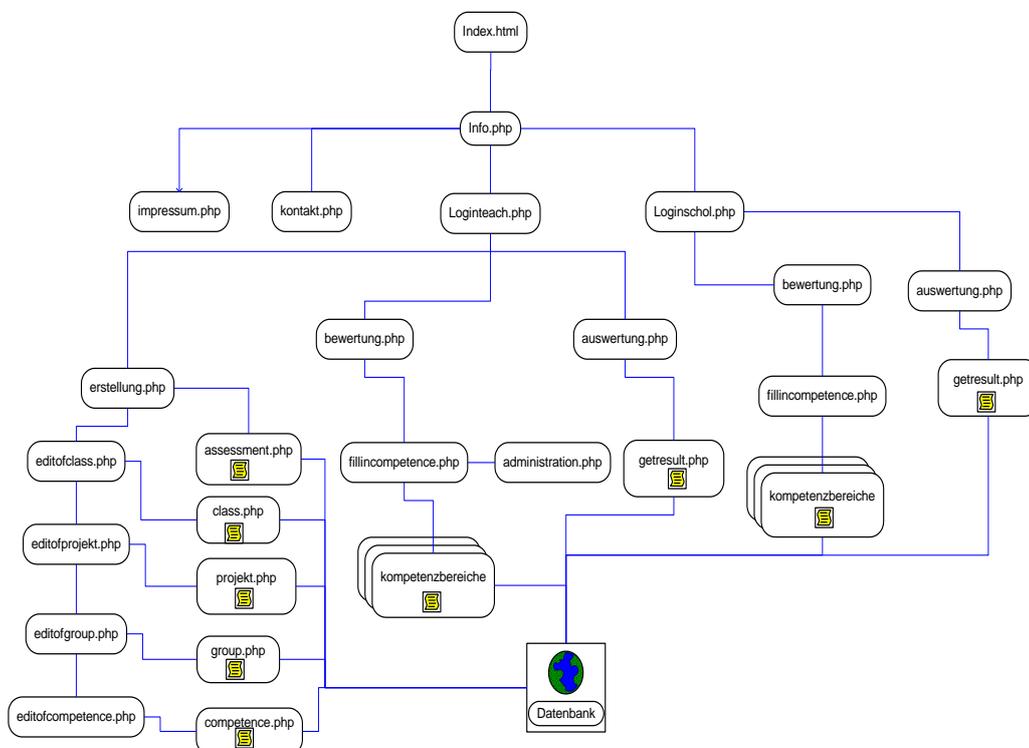


Abbildung 10: Stammbaumarchitektur der Web-Applikation (Anhang B)

Die blauen Verbindungslinien deuten auf einen Informationsaustausch zwischen den einzelnen Seiten oder einer Datenbank. Eine genauere Spezifikation der

Informationen wird an dieser Stelle nicht gegeben. Die Speicherung der eingegebenen Daten erfolgt auf dem Webserver, genauer in der Datenbank.

### 3.2.2 Technische Produktumgebung

Die Web-Applikation zur Fremd- und Selbstbewertung von Kompetenzen besteht aus einer Verbindung von Webserver, Interpreter und Datenbank. Zur Entwicklung einer Web-Applikation können unterschiedliche Technologien eingesetzt werden. Diese Technologien besitzen alle einen ähnlich strukturierten Ablauf, der in der folgenden Entwurfphase bei Vorstellung der System-Architektur genauer beschrieben wird (vgl. Kap. 6.2).

In den folgenden Kapiteln wird auf die Entwicklungsumgebung und auf die eingesetzten Tools bzw. Anwendungsprogramme zur Programmierung und Verwaltung eingegangen.

#### 3.2.2.1 Entwicklungsumgebung

Als Plattform für die Realisierung der Web-Applikation zur Fremd- und Selbstbewertung wird ein so genanntes *LAMP*-System gewählt. *LAMP* steht für *Linux*, *Apache*, *MySQL* und *PHP*, eine Kombination aus Betriebssystem, Webserver, Datenbanksystem und Skriptsprache. Ein *LAMP*-System bietet hohe Leistung zu einem niedrigen Preis. Die gesamte für ein *LAMP*-System benötigte Software besteht aus *Open-Source* Projekten und kann aus dem Internet herunter geladen werden [Linux, Apache, MySQL, PHP]. *Open-Source* bedeutet, dass die Projekte für nicht kommerzielle Zwecke frei in ihrem Quellcode zur Verfügung stehen (vgl. SNOOPY 2000). Ein Auszug der Lizenzbestimmungen für die Nutzung der Technik befindet sich im Anhang C.

Linux erhält als Betriebssystem für Webanwendungen immer größere Bedeutung. Für die Entwicklung wird die Distribution von Suse Linux in der Version 9.0 verwendet. Das Betriebssystem Linux lässt sich problemlos neben einem bestehenden Windowssystem betreiben. Diese Kombination hat gewisse Vorteile, da einige spezielle Anwendungsprogramme nicht unter Linux und umgekehrt funktionieren (vgl. KOFLER 2002: Kap.1).

Der für die Entwicklung eingesetzte *Apache* Webserver ist eine Weiterentwicklung des „*http-Dämon*“ des National Center for Supercomputing Applications (NCSA). Die Versionsnummer des eingesetzten Webservers lautet 2.0.47-63. Für die Entwicklung der Web-Applikation läuft der *Apache* Webserver als Dienst auf dem Betriebssystem Linux. Der Serverdienst läuft aber auch unter anderen Betriebssystemen wie Windows oder Unix (vgl. ROßBACH 1998: 11-15).

*MySQL* ist eine relationale Datenbank, die sich besonders für Webanwendungen eignet. *MySQL* kann mit der Datenbanksprache SQL verwaltet werden, es existieren aber auch viele Bedienungsoberflächen, die die Administration der Datenbank erleichtern. Zur Entwicklung von *MySQL*-Anwendungen gibt es eine ganz

Reihe von APIs (Application Programming Interfaces) bzw. Bibliotheken. Sie können zur *Client*-Programmierung daher u. a. die Sprachen C, C++, Java, Perl, *PHP*, Python und Tcl einsetzen. MySQL ist plattformunabhängig und gilt als vergleichsweise schnell (vgl. KOFLENER 2003).

Zwei Gründe führten zur Entscheidung für diese Entwicklungsplattform:

Durch die freie Lizenznutzung für den Schulbereich sind Open-Source Techniken erheblich kostengünstiger. Viele Schulen besitzen häufig einen eigenen Webserver, der das Lesen von Skripten ermöglicht. PHP-Skripte werden dabei von fast allen marktüblichen Webservern gelesen, was bei Microsoft-Produkten nicht der Fall ist.

Ein zweiter Grund für den Einsatz von *Open-Source* Techniken sind die auftretenden Sicherheitslücken bei Microsoft-Produkten. Es wird zwar durch umfangreiche Updates seitens Microsoft für ein verhältnismäßig stabiles System gesorgt, der Wartungsaufwand ist dadurch allerdings recht hoch (vgl. SCHMIDT 2001).

### 3.2.2.2 Tools für die Entwicklung

Zur Einrichtung eines LAMP Systems stehen heutzutage viele Möglichkeiten zur Verfügung. Das eingesetzte Betriebssystem Suse Linux 9.0 bietet bei seiner Installation die Möglichkeit, die notwendigen Pakete für den Webserver und die Datenbank gleich mit zu installieren. Somit ist die erste Installation nicht besonders schwierig. Aufwändiger ist jedoch die Einrichtung der Sicherheitseinstellungen für den Pop-Interpreter und die MySQL-Datenbank, da diese per Editor in die Konfigurationsdateien eingegeben werden müssen. Der Webserver ist ebenso gegen unbefugtes Eindringen zu sichern (vgl. Kapitel 3.4.2).

Folgende Tools und Anwendungen werden zur Entwicklung der Webapplikation eingesetzt:

- phpMyAdmin Version 2.5.6:  
Dies ist eine in PHP geschriebene Webanwendung, die eine grafische Oberfläche zur Pflege von MySQL Datenbanken bereitstellt. Mit diesem Tool ist es möglich, Datenbanken zu erstellen und zu löschen. Einzelne Tabellen und Datensätze (Tupels) können editiert werden. SQL Befehle können direkt über die Eingabe importiert und ausgeführt werden.
- MySQL Control Center Version 0.94  
Das MySQL Control Center, oft auch kurz mysqlcc genannt, ist ein Programm, dass von der Firma MySQL zur Verfügung gestellt wird. Über eine Eingabemaske werden die Datenbanken ohne Benutzung des Webservers bequem verwaltet.

- Microsoft Visio 2002

Microsoft Visio 2002 ist ein Programm zur Erstellung von Diagrammen und Zeichnungen der unterschiedlichsten Art. Es können Geschäftsprozesse definiert werden und das Programm umfasst die Erstellung von Organigrammen, Flussdiagrammen, Zeit- und Projektplanungen, Raumplänen sowie Standard-Netzwerkdiagrammen.

- Dreamweaver MX

Dreamweaver MX ist ein professionelles Tool zum Erstellen von Webseiten und Internetanwendungen, das sich durch eine leistungsstarke Kombination aus Werkzeugen zur Grafikbearbeitung, Funktionen zur Anwendungsentwicklung und Optionen für die Skriptbearbeitung auszeichnet.

- The GIMP

The GIMP (GNU Image Manipulation Program) ist ein Bildbearbeitungsprogramm, das unter der freien GNU General Public Licence (Anhang C) veröffentlicht wird. Der Leistungsumfang für die reine Bildbearbeitung ist vergleichbar mit dem professioneller, kommerzieller Programme wie beispielsweise Photoshop.

### 3.2.2.3 Serverpflege und Aktualisierungen

Die Serverpflege und Aktualisierung erfolgt ab Anbindung an das Internet durch den Auftraggeber. Während des Testbetriebs werden vom Auftraggeber sämtliche Inhalte auf Korrektheit überprüft. Korrekturwünsche werden einmalig vom Auftragnehmer umgesetzt. Der Auftragnehmer kümmert sich um einen geeigneten Provider und um die Domainreservierung. Folgende Parameter werden bei der Auswahl berücksichtigt:

- Domainname: kompetenzbewertung.de
- Speicherplatz max. 200 MB
- Ablagemöglichkeit eigener PHP Skript
- Installationsmöglichkeit einer MySQL-Datenbank

Zur Zeit der Entwicklung ist noch unklar, auf welchen Server die Web-Applikation später eingesetzt wird.

### 3.2.3 Abschluss der Definitionsphase

Den Abschluss der Definitionsphase bildet die Zusammenfassung der erreichten Ergebnisse:

- Erstellung eines Pflichtenheftes (Anhang B),
- Beschreibung der Arbeitsabläufe anhand eines Modells,

- Erstellung eines Oberflächen Prototypen,
- Vorstellen der technischen Produktumgebung.

Diese Ergebnisse bilden Anforderungen zur softwaretechnischen Umsetzung für die Entwurfsphase.

### 3.3 Die Entwurfsphase

Der Übergang von der Definitionsphase zur Entwurfsphase verläuft fließend. In der Definitionsphase wird das zu entwickelnde Produkt aus der Sicht des Anwenders beschrieben. In der Entwurfsphase werden die erstellten Anforderungen aus software-technischer Sicht gesehen. Die Entwürfe dieser Phase bilden den Abschluss der Modellierung. Folgende Ergebnisse sind für die Entwurfsphase bei der Entwicklung der Web-Applikation zu erwarten:

- Darstellung der Systemarchitektur,
- Modellierung der Datenbank,
- Vorstellung der Benutzeroberfläche.

#### 3.3.1 Darstellung der Systemarchitektur

Die Basisplattform für jede Web-Anwendung besteht aus einem so genannten Webserver. Der *Apache* Webserver ist lizenzfrei und führt mit fast 64% Marktanteil den Weltmarkt an. In Kombination mit dem Betriebssystem Linux gilt er als stabil und sicher. Der Internet-Information-Server (IIS) von Microsoft rangiert auf Platz zwei mit knapp 24% Marktanteil[vgl. NETCRAFT 2004].

Die Funktionsweisen von Webserverdiensten sind nahezu identisch. Um den Ablauf einer statischen Internetanfrage darzustellen dient die folgende Grafik:

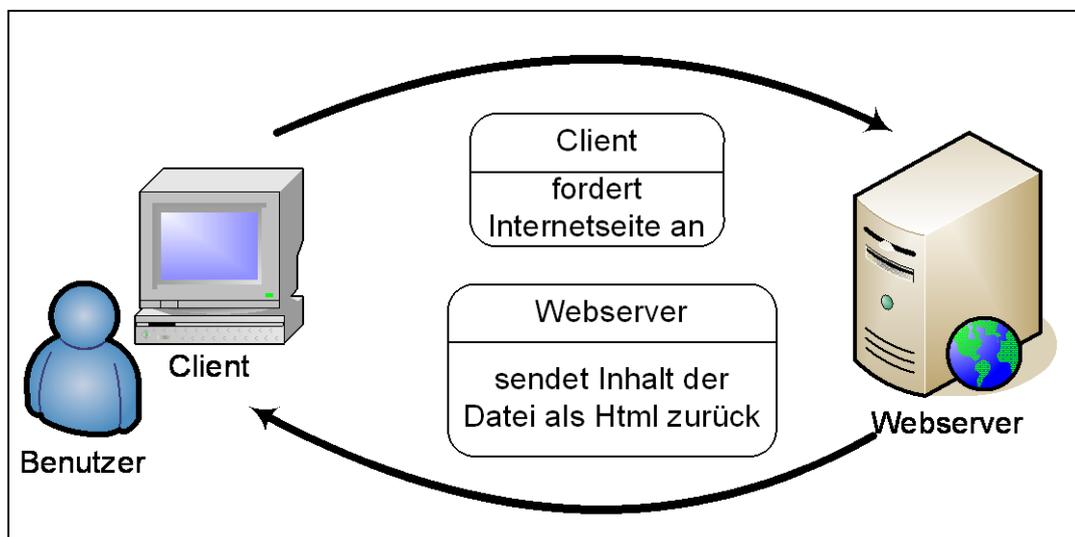


Abbildung 11: Aufruf einer statischen Webseite (vgl. LORENZ 2002: 42)

Auf dem hier dargestellten Webserver ist zusätzlich zum Betriebssystem ein Webserverdienst aktiviert und eingerichtet. Dieser Dienst sorgt u. a. dafür, dass *HTML*-Dokumente gespeichert und versendet werden. Das Hypertext-Transfer-Protokoll (*http*) ist das notwendige Protokoll, mit dessen Hilfe die Informationen zwischen Servern und *Clients* über das Internet übertragen werden. Der Benutzer fordert über einen *Client* Rechner, d.h. einen PC mit Internet-Zugang, eine Datei auf dem Webserver an, indem er z.B. einen *URL* in seinem *Browser* eingibt. Der Webserver nimmt diese Anfrage auf und kontrolliert, ob diese Datei physikalisch auf dem Webserver vorhanden ist. Wenn das der Fall ist, sendet der Webserver eine *Html*-Seite zum *Client* zurück. Dieser erhält nun ein Abbild der Datei, die typischerweise eine *HTML*-Anweisung enthält. Der *Browser* des *Clients* kann die Anweisung interpretieren und darstellen. Befindet sich eine angeforderte Datei nicht auf dem Webserver, gibt dieser automatisch eine Fehlermeldung aus.

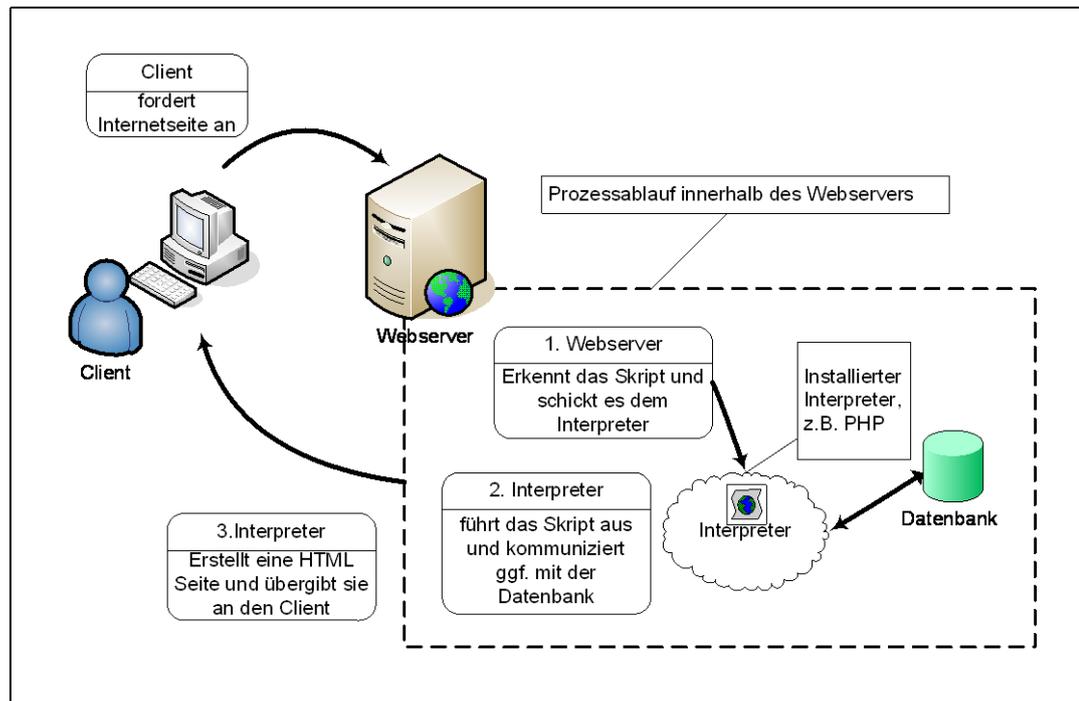


Abbildung 12: Aufruf einer dynamischen Webseite (vgl. LORENZ 2002: 43)

Der Aufruf von dynamischen Internetseiten verläuft prinzipiell ähnlich. Die *Client*-Anfrage wird vom Server aufgenommen. Da zusätzlich zu den *HTML*-Anweisungen noch ein weiteres Skript, z.B. ein *PHP*-Skript in der Datei steht, leitet der Webserver die Anfrage intern an ein Interpreterprogramm weiter (1). Es wird von dem Interpreterprogramm überprüft, ob die Datei vorhanden ist und das zusätzliche Skript befiehlt z. B. den Verbindungsaufbau zur Datenbank (2). Internetdatenbanken dienen als Datenspeicher und können unterschiedlich an die Webserver-Umgebung angeschlossen sein. Diese Datenbank liefert die vom Skript geforderten Informationen und übergibt sie dem Interpreter. Der Interpreter erstellt aus den neuen Informationen eine Internetseite und leitet sie als *HTML*-

Format an den Browser des *Clients* weiter (3). Durch diese Informationsverarbeitung erhält der *Client* eine für ihn interpretierte Seite, wie es z.B. bei einem Shopsystem vorkommt.

Für die Entwicklung der Web-Applikation werden also Kenntnisse im Bereich der Interpretersprache und der Datenbanksprache benötigt. Diese Sprachen sind je nach eingesetztem System unterschiedlich, die Abläufe jedoch weitgehend identisch.

### 3.3.1.1 Interpretersprachen

Das Interpreterprogramm ist für die Bearbeitung der Skripte zuständig. Es liest den Skriptcode zeilenweise in dem Moment, in dem das Skript übergeben wird, stellt gegebenenfalls eine Verbindung zur Datenbank her und übergibt eine fertige Webseite an den *Client*. Beispiele für Interpretersprachen sind: ASP mit integriertem Visual Basic for Applications (VBA), Java-Skript, Perl und PHP. Interpreterskripte sind für die Flexibilität und Dynamik von Webseiten von entscheidendem Vorteil. Die Verarbeitung geschieht serverseitig und ist somit unabhängig vom *Client*, egal ob der Benutzer als Betriebssystem Apple Macintosh OS, Unix oder Windows benutzt, welchen Browser er einsetzt und welche Sicherheitseinstellungen er clientseitig besitzt.

Bei der Entwicklung der Web-Applikation wird die Interpretersprache PHP eingesetzt. PHP ist ein rekursives Akronym für „Hypertext **P**reprozessor“. Dabei handelt es sich um eine Skript-Programmiersprache, die in HTML-Seiten eingebettet ist. Um die Struktur von PHP darzustellen, wird in den nächsten Kapiteln, die Dokumentensprache HTML und die Einbettung der Interpretersprache kurz vorgestellt.

#### 3.3.1.1.1 HTML – Hypertext Markup Language

HTML ist die Dokumentenbeschreibungssprache im Internet und wurde am 18.2.1998 als Sprachstandard von der „Centre Européen de Recherches Nucleaires“ (CERN) verabschiedet. Die Gründe für die Entwicklung dieses Standards liegen in der hohen Akzeptanz dieser Sprache und deren Kompatibilität zu allen Betriebssystemen. Die offizielle Spezifikation gibt es auf den Seiten des W3-Konsortiums [W3C]. HTML 4.0 segnet neben Frames auch die Einbindung von kaskadierenden Style Sheets (CSS) und von Skriptsprachen in HTML ab. Damit sind fast alle heute weit verbreiteten Ergänzungstechnologien für HTML offizieller Standard oder zumindest im Standard vorgesehen.

HTML kann in vielen Editoren geschrieben werden, unter Linux mit KWrite oder Kate, in Windows unter dem Editor oder Notepad. Viel komfortabler sind allerdings so genannte WYSIWYG (What you see, is what you get) Editoren, wie z.B. MS-Frontpage, Golive, Dreamweaver, Homepage oder Netscape Composer.

### 3.3.1.1.2 PHP – Hypertext Preprozessor

Das in die HTML-Dateien eingebettete PHP-Skript wird vom Server ausgeführt. PHP ist eine Skriptsprache zur Erstellung dynamischer Webseiten. Die erste Version wurde 1994 entwickelt mit der Abkürzung „Personal Home Page“. Später entwickelte die *Open Source*-Gemeinde die rekursive Version "PHP Hypertext Preprozessor". Seit dieser Zeit entwickelte sich PHP aufgrund der freien Verfügbarkeit rasant weiter. Bei dem Konkurrenzprodukt von Microsoft, den „Active Server Pages“ (ASP), ergibt sich ein Nachteil durch die Einschränkung auf Microsoft-Plattformen.

PHP besticht durch einen großen Funktions- und Supportumfang. Die Foren im Internet besitzen gute Dokumentationen und geben Anwendungsbeispiele, die zum Teil frei zur Verfügung stehen. Besonders deutlich wird dies bei den Datenbankfunktionen. PHP unterstützt neben MySQL auch andere Datenbanksysteme.

Die Dateinamenserweiterung von PHP ist entweder .php3, .php4, .php oder .phtml statt .html. So registriert der Webserver sofort, an welchen Interpreter diese Datei weiterzuleiten ist. Die statischen Anteile einer Webseite werden weiterhin durch die bekannten HTML-Befehle erzeugt, die variablen Anteile werden durch die Sprachelemente der PHP-Skriptsprache erzeugt. Im HTML-Skript werden die PHP-Skriptelemente durch besondere *Tags* gekennzeichnet.

Im folgenden Beispiel wird die Mischung aus Html und PHP deutlich:

```
<!-------Einfacher HTML Tag--->
<html>
<body>
  <!-------Beginn PHP Skript--->
  <?
  echo "Hallo Welt!";
  ?>
  <!-------Ende PHP Skript--->
</body>
</html>
  <!-------Ende des Skripts--->
```

Diese Datei als test.php abgespeichert, liefert in dem Browser folgendes Ergebnis:

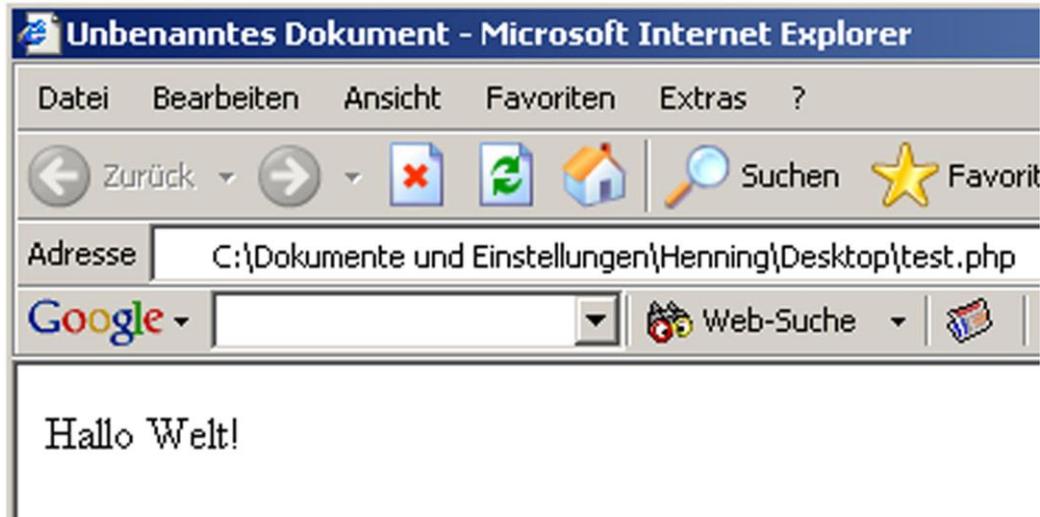


Abbildung 13: Beispiel eines Testskriptes für PHP (vgl. KOFLER 2003: 79f.)

Diese Seite kann einfach in HTML geschrieben werden, sie steht aber hier als Beispiel für eine einfache Einbettung von PHP-Skripten in HTML. Durch die serverseitige Ausführung des Skriptes ergibt sich ein wesentlicher Vorteil: Als Produkt bekommt der *Client* eine fertig interpretierte Webseite. Wenn keine clientseitigen Skripte eingesetzt werden, minimieren sich die Inkompatibilitäten mit anderen Browsern

Dass PHP-Skripte noch viel mächtiger sind, zeigt die Webserver-Struktur in Abbildung 3. Eine Anbindung an Datenbanken eröffnet neben einfacher Skriptausführung weitere Optionen.

### 3.3.1.2 Die Struktur des Datenbanksystems

Beim Schreiben und Nutzen von Programmen entstehen Daten. Diese gehen bei Programmende verloren, wenn sie nicht auf Speichermedien als Dateien abgelegt werden. Die Dateien und deren Inhalte sind dauerhaft (persistent) gespeichert, bis sie verändert oder gelöscht werden. Verschiedene Programme benötigen teilweise die gleichen Daten. Es kann daher zu redundanter Speicherung kommen und die Programme überschreiben sich gegenseitig. Daher ist es sinnvoll, Daten eines Anwendungssystems integriert zu verwalten. Eine solche Verwaltung wird von einem Datenbanksystem koordiniert, das aus folgenden Komponenten besteht:

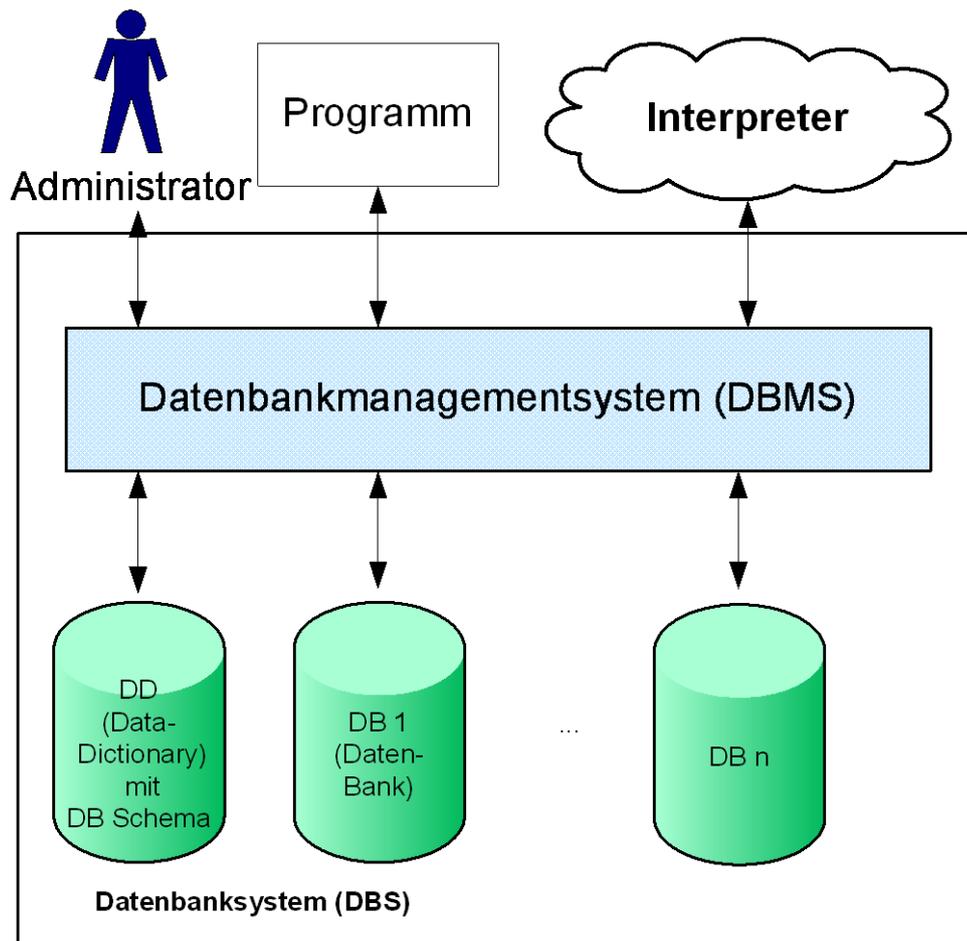


Abbildung 14: Struktur eines Datenbanksystems (vgl. BALZERT 2000: 720)

Ist eine bestimmte Anzahl von Dateien logisch miteinander verknüpft, so wird diese Sammlung allgemein als *Datenbank (DB)* bezeichnet.

In dem *Data Dictionary (DD)* wird das Datenbankschema gespeichert. Es beinhaltet die Beschreibung des Datenbankaufbaus und seine Verknüpfungen.

Das *Datenbankmanagementsystem (DBMS)* sorgt allgemein für die Verwaltung und Koordination der Datenbank(en) unter Berücksichtigung des Schemas der Datenbank im *DD*. Moderne Datenbanksysteme koordinieren hierüber auch den Multi-User Zugriff, damit nicht unkontrolliert Daten überschrieben, gelöscht oder bearbeitet werden.

Folgende Leistungen sprechen für einen Einsatz eines Datenbanksystems (vgl. SCHULZ/SIERING 2003):

- Zuverlässige Verwaltung durch das Datenbankmanagementsystem,  
Die Daten werden verwaltet und im Falle eines Systemabsturzes kann eine Wiederherstellung der Datenbank erfolgen (recovery).
- Unabhängige Verwaltung,

Die Datenbanken und die Programme sind voneinander getrennt. Unabhängig von den Programmzugriffen können die Daten im DD schematisiert werden.

- Komfortable Bedienung,

Durch die Datenunabhängigkeit können Schnittstellen eingesetzt werden, die sich um Speicherungsdetails kümmern.

- Flexiblen Einsatz,

Administratoren können schnell mit selbst programmierten Skripten oder Befehlen die Daten auslesen und anpassen.

- Die Sicherheit der Daten,

Durch geschützte Bereiche kann nicht jedes Programm oder jeder Benutzer die Daten einsehen oder verändern.

- Mehrfachbenutzbare Datenbank,

Programme und Anwender können gleichzeitig auf den Datenbestand zugreifen. Ein Datenbanksystem koordiniert diese Zugriffe.

- Umfangreiche Datenaufnahme.

Die Daten müssen nicht vollständig im Arbeitsspeicher gehalten werden.

- Integrität der Datenbank.

Daten werden geordnet und redundanzarm gespeichert. Teilweise benötigen Programme nicht den gesamten Datensatz. Sichten auf den Datensatz sorgen für eine effiziente Datenverwaltung.

Diese Leistungen kann ein Datenbanksystem allerdings nur dann erbringen, wenn die Inhalte der Datenbank durch Angabe ihrer Bedeutung in einem Datenmodell beschrieben werden. Anders als ein Datenbankschema, das für die Beschreibung einer Datenbank dient, legt das Datenmodell Eigenschaften, Struktur und Konsistenzbedingungen für in die Datenbank zu speichernde Datenelemente einschließlich der erlaubten Erzeuge-, Manipulations- und Löschoptionen fest. Es legt weiterhin fest, welche Operationen zum Speichern, Auffinden, Ändern und Löschen von Datenmengen erlaubt sind.

Das mit 98% am meisten verwendete Datenmodell ist das relationale Datenmodell (vgl. BALZERT 2000: 722). Die Daten werden in Tabellenform gespeichert und werden dann in Beziehungen (Relationen) gesetzt. Andere Datenmodelle sind z.B. die objektorientierten Datenmodelle, in denen die Daten nicht tabellarisch sondern in Objekten erfasst werden. Das für die Web-Applikation zugrunde liegende Datenmodell ist relational.

### 3.3.2 Modellierung der Datenbank der Web-Applikation

Die Datenbank der Web-Applikation zur Unterstützung der Fremd- und Selbstbewertung wird durch ein Entity/Relationship-Modell (ER-Modell) beschrieben.

Das ER-Modell wurde 1976 von Peter Chen zur Modellierung von Datenbanken vorgeschlagen und hat bis heute eine hohe Bedeutung sowohl im Datenbankentwurf als auch in anderen Teilbereichen der Informatik, in denen Modellierung eine wichtige Rolle spielt (vgl. VOSSSEN 2000). Das ER-Modell beschreibt eine Abstraktion der realen Welt aus Anwendersicht und stellt physisch oder gedanklich existierende Gegenstände als Entitäten (*engl.* entities) und ihre Beziehungen (*engl.* relationships) untereinander dar. Es ist unabhängig von einem bestimmten Datenbanksystem, so dass bei der Implementierung des Modells nicht mit Systemeinschränkungen gerechnet werden muss (vg. BALZERT 2000: 740).

Entitäten sind reale oder begrifflich existierende Gegenstände mit einer festen und bekannten Menge an Eigenschaften (Attributen). Zusammengefasst werden diese Entitäten zu einem Entitäts-Typ. In dem nachfolgenden Diagramm wäre ein Entitäts-Typ also die Zusammenfassung der Teilnehmer (Entitäten) mit ihren bestimmten Attributen zu dem Entitäts-Typ (Teilnehmer). In dem ER-Modell werden Entitäts-Typen als Rechtecke, Beziehungen als Karos und Attribute als Ovale gekennzeichnet

Folgendes ER-Modell wurde für die Webapplikation erstellt:

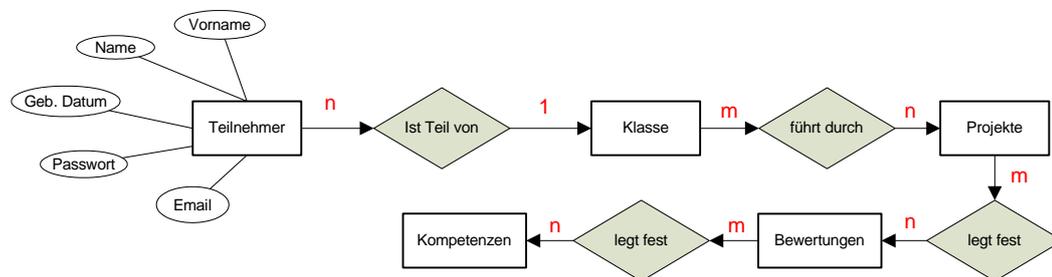


Abbildung 15: Die Datenbank der Web-Applikation im ER-Modell

Aus Gründen der Übersichtlichkeit werden in diesem ER-Modell nur die Attribute des Teilnehmers dargestellt. Die roten Zahlen und Buchstaben bezeichnen Beziehungstypen, so genannte Kardinalitäten. Es gibt drei Typen von Beziehungen zwischen den Entitäten (vgl. KOFLER 2003: 175):

1:1 – Beziehungen: Jeder Datensatz des einen Entitäts-Typen entspricht genau einem Datensatz des anderen Entitäts-Typen. Die Beziehungen sind eher selten, weil die Informationen ebenso in einem einzigen Entitätstypen gespeichert werden könnten.

1:n – Beziehungen: Ein Datensatz des Entitäts-Typen kann in mehreren Datensätzen des zweiten Entitäts-Typen auftreten. Bei der Web-Applikation können mehrere Schüler (Teilnehmer) einer Klasse angehören. Umgekehrt ist keine Mehrdeutigkeit vorhanden. Ein Schüler befindet sich in einer Klasse.

n:m – Beziehungen: Ein Datensatz des Entitäts-Typen kann in mehreren Datensätzen des zweiten Entitäts-Typen auftreten und umgekehrt. Eine Klasse führt mehrere Projekte durch. Diese Projekte können aber auch in anderen Klassen durchgeführt werden.

Ziel dieser Beschreibung ist es, ein übersichtliches Datenbank-Design zu erhalten, das als Grundlage für die Implementierung zu sehen ist. In der Implementierungsphase wird das Er-Modell wieder aufgegriffen und tabellarisch dargestellt.

### 3.3.3 Benutzermanagement

Da die Möglichkeit, Daten in der Web-Applikation abzulegen, auf bestimmte, identifizierbare Personen begrenzt werden muss, ist es notwendig, ein Benutzermanagement anzulegen. Hier bieten sich zwei Lösungsmöglichkeiten an:

Zum einen kann die webserverinterne Benutzerverwaltung verwendet werden. Dies bedeutet, dass Benutzer nur auf bestimmte Verzeichnisse Zugriff haben. Durch eine entsprechende Vergabe der Dateirechte wird ein Zugriff auf die Skriptdateien durch unbefugte Benutzer unterbunden. Der Administrator besitzt einen geschützten Bereich, in dem er die Berechtigung hat, neue Datenbanken zu erzeugen und anzulegen (vgl. KOFLEER 2003: 74f.).

Die andere Lösung besteht darin, die Benutzerdaten der Web-Applikation in einer eigenen Tabelle datenbankintern zu speichern und zu verwalten. Dies hat den entscheidenden Vorteil, dass beliebige Zusatzinformationen (Attribute) abgelegt werden können. Weiterhin sind die Zugangsdaten wie Passwort und Benutzername völlig unabhängig von den systemintern verwendeten Daten, was einen weiteren Vorteil aus Sicht der Datenbankadministration darstellt.

## 3.4 Die Implementierungsphase

Der Übergang von der Entwurfsphase zur Implementierungsphase verläuft ebenfalls fließend. In der Implementierungsphase wird die Web-Applikation vollständig programmiert. Die entwickelten Modelle, das Lasten- sowie das Pflichtenheft dienen als Grundlage für die Programmierung. Der Oberflächen-Prototyp dient als Vorlage und wird nun durch HTML- und PHP-Skripte neu entwickelt. Folgende Ergebnisse sind in der Implementierungsphase bei der Entwicklung der Web-Applikation zu erwarten:

- Vollständiger Aufbau des *Backend*
- Programmieren der Eingabe- und Ausgabemasken,

### 3.4.1 Vollständiger Aufbau des Backend

Das *Backend* der Web-Applikation baut auf die in den vorherigen Kapiteln vorgestellten Techniken auf. In der Implementierungsphase werden nun diese Techniken auf einem Test-System eingerichtet und installiert.

#### 3.4.1.1 Typische Einrichtungs-Probleme

Im Laufe dieser Examensarbeit traten bei der Einrichtung einige typische Probleme zu Tage, die hier kurz skizziert werden:

- Voreinstellungen des Webservers

Die Konfiguration eines Webservers sollte vor Inbetriebnahme hinsichtlich der Sicherheitseinstellungen immer überprüft werden. Beispielsweise müssen die eingesetzten Module/Bibliotheken korrekt installiert und gestartet sein (PHP, MySQL, SSL). Die Pfade zu den Konfigurationsdateien müssen stimmen und der Server muss auch nach einem Absturz wieder einwandfrei laufen.

- Benutzerrechte der Datenbank

Die Benutzer-Zugriffsrechte auf einer Datenbank sollten immer eingeschränkt sein. Bei der Installation von MySQL ist beispielsweise darauf zu achten, dass der privilegierte Benutzer ein Kennwort erhält und anschließend der Datenbank-Dienst neu gestartet wird.

- Vertrauen in hidden-Fields

PHP Skripte werden in HTML-Seiten als versteckte Felder (hidden-Fields) angezeigt. Die versteckten Felder in HTML-Formularen sind für einen versierten Benutzer natürlich nicht unsichtbar, daher sollten hier grundsätzlich keine sicherheitsrelevanten Informationen (beispielsweise Klartext-Kennworte) angegeben werden.

- Dateirechte auf dem Server

Sollte der Webserver auch anderen Personen Zugang bieten, so muss genau auf die Dateirechte geachtet werden. Beispielsweise sollten Dateien, die das Kennwort zur Datenbank enthalten, nur für den Webserver und Administrator der Applikation lesbar sein.

Um alle diese Einstellungen zu tätigen, wurde folgende Dokumentation und Fachliteratur zur Hilfe genommen:

- Für die Einrichtung und Verwaltung des Webservers sind Sicherheitsmaßnahmen besonders wichtig, einerseits sind die Daten der Web-Applikation zu schützen, andererseits dürfen keine ungewünschten Zugriffe den Server in seiner Funktion beeinträchtigen (vgl. ROßBACH 1998: 95-126).

- Für die Datensicherheit und den beschränkten Zugriff zur Datenbank sind die Sicherheitseinstellungen beim MySQL-Dienst einzuhalten (vgl. KOF-LER 20003: 320-265).

### 3.4.2 Realisierter Prototyp

Der von mir entwickelte Prototyp erfüllt noch nicht alle zuvor gestellten Anforderungen, ist teilweise funktionsfähig und kann ohne tief greifende Veränderungen weiterentwickelt werden. Die PHP-Skripte besitzen momentan noch keine Anbindung an die Datenbank, außer der Login-Verwaltung. Als Beispiel für die Funktion des Prototyps werden anhand des folgenden Skripts die Verbindung zur Datenbank und das Aussehen des Frontends beschrieben.

#### 3.4.2.1 Login-Skript für Schüler

Das Skript wird durch die gräulich gekennzeichneten Kommentare unterbrochen. Das interpretierte Ergebnis des Skripts wird im Anschluss an dieses Kapitel dargestellt.

```
// der dateiname des php skripts lautet: loginschool.php
// login für die schüler
// formular variablen: formUser, formPassword, formSubmit
// include Befehl schließt die verbindung zur datenbank ein
// die datei mysqlconnect.inc.php befindet sich in einem
// geschützten verzeichnis auf dem Server. sie enthält die
// zugangsdaten zur datenbank

<?php
include("mysqlconnect.inc.php");

$formSubmit = array_item($_POST, 'formSubmit');
$formSchool = array_item($_POST, 'formSchool');
$formPassword = array_item($_POST, 'formPassword');
$formUser = array_item($_POST, 'formUser');

// magic quotes verhindert das senden von 0-byte Dateien
// hier wird in das leere feld ein schrägstrich zugeführt,
// damit die datenbank nicht eine fehlermeldung ausgibt und
// sich nicht aufhängt

if(!get_magic_quotes_gpc()) {
    $formSchool = addslashes($formSchool);
    $formPassword = addslashes($formPassword);
    $formUser = addslashes($formUser); }
```

```
// verwaltet das session-management, d.h es erlaubt
// rücksprünge mit dem zurück-button
// deswegen soll der vorher eingegebene inhalt
// zwischengespeichert werden. Ab der mysql version 4.0
// erleichtert dieser befehl das management...

session_start();

header("Cache-control: private");

$_SESSION['sesUserID'] = '';
$_SESSION['sesUserName'] = '';

// sid1/-2 falls cookies clientseitig abgestellt wurde.

if(SID) {
    $sid1 = "?" . SID;
    $sid2 = "&" . SID; }
else {
    $sid1=""; $sid2=""; }

// verbindung zur kompetenz-datenbank
$connID = connect_to_kompetenz();

// test zum überprüfen der daten

if($formSubmit) {
    $result =

// sql abfrage: select datentabelle, eintrag user, user und
// password überprüfung

    mysql_query("SELECT IDadresse, userName FROM tbl_adresse " .
                "WHERE VarchName = '$formUser' " .
                " AND Varchpassword = PASSWORD('$formPassword')");

// benutzer gefunden

if(mysql_num_rows($result)==1) {
    $_SESSION['sesUserID'] = mysql_result($result, 0, 0);
    $_SESSION['sesUserName'] = mysql_result($result, 0, 1);

// wechsel zur bewertung

    header("Location: bewertung.php$sid1");

    exit;
}
}

?>

// html skript startet hier, das bis hier beschriebene php-
// skript läuft im hintergrund im (backend) ab.

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
```

```
<html>
<head>
<title>Bewertung von Kompetenzen</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

// java skript, zum rollover der buttons. wenn mit der maus
// über die buttons der seite gegangen wird, verändert sich
// die farbe.

<script language="JavaScript" type="text/JavaScript">

<!--

function MM_swapImgRestore() { //v3.0

    var i,x,a=document.MM_sr; for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++)
    x.src=x.oSrc;

}

function MM_preloadImages() { //v3.0

    var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();

    var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
    if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}

}

function MM_findObj(n, d) { //v4.01

    var p,i,x;  if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
        d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}

    if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++)
x=d.forms[i][n];

    for(i=0;!x&&d.layers&&i<d.layers.length;i++)
x=MM_findObj(n,d.layers[i].document);

    if(!x && d.getElementById) x=d.getElementById(n); return x;

}

function MM_swapImage() { //v3.0

    var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array;
for(i=0;i<(a.length-2);i+=3)

    if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x; if(!x.oSrc)
x.oSrc=x.src; x.src=a[i+2];}

}

</script>

// ende des java skripts

// hier befindet sich der pfad zum cascading style sheet
// (css)

<link href="sheet/nav.css" rel="stylesheet" type="text/css">

</head>

// hier beginnt die darstellung im html skript
```

```

<body bgcolor="#CCCCCC" text="#333333" link="#333333" vlink="#333333"
alink="#333333" leftmargin="50" topmargin="50"
onLoad="MM_preloadImages('pics/info2.gif','pics/auswertung2.gif','pics/impressum2.
gif','pics/kontakt2.gif')">
<table width="711" height="449" border="0" cellpadding="0" cellspacing="0">
  <tr bgcolor="#999999">
    <td width="10">&nbsp;</td>
    <td height="21">&nbsp;</td>
    <td width="10">&nbsp;</td>
    <td width="14">&nbsp;</td>
    <td width="747"><font color="#336666" size="3" face="Verdana, Arial,
Helvetica, sans-serif"><strong>kompetenzbewertung.de
      </strong></font></td>
  </tr>
  <tr>
    <td bgcolor="#999999">&nbsp;</td>
    <td bgcolor="#669999">&nbsp;</td>
    <td bgcolor="#336666">&nbsp;</td>
    <td bgcolor="#FFFFFF">&nbsp;</td>
    <td bgcolor="#FFFFFF"><a href="index.php" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('info','','pics/info2.gif',1)"></a>
      <a href="auswertung.php" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('auswertung','','pics/auswertung2.gif',1)">
        </a><a href="impressum.php" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('Impressum','','pics/impressum2.gif',1)"></a>
        <a href="kontakt.php" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('kontakt','','pics/kontakt2.gif',1)"></a>
      </td>
  </tr>
  <tr>
    <td bgcolor="#999999"></td>
    <td width="20" bgcolor="#669999">
      <p><a href="index.php" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('info','','pics/info2.gif',1)">
        </a><font size="1" face="Verdana, Arial, Helvetica, sans-
serif"></font><font face="Verdana, Arial, Helvetica, sans-serif"></font></p>
      </td>
    <td bgcolor="#336666"></td>
    <td bgcolor="#FFFFFF"></td>

```

```

    <td bgcolor="#FFFFFF"> <h2><font size="1" face="Verdana, Arial, Helvetica,
sans-serif"></font><br>

        <font size="3" face="Verdana, Arial, Helvetica, sans-serif">Login Seite
        f&uuml;r Sch&uuml;ler</font></h2>

        <font face="Verdana, Arial, Helvetica, sans-serif">

// php skript: gibt eine fehlermeldung aus, wenn das
// password falsch eingegeben wurde

        <?php
if($formSubmit and empty($userID))
    echo '<p><font color="#ff0000">Nutzername oder Passwort ',
        "sind ung&uuml;ltig. Bitte erneut versuchen.</font></p>\n";
?>

// php skript: ende

        <p><font size="2">Dieser Zugang ist f&uuml;r Sch&uuml;ler gedacht. Hier
        werden Bewertungen durchgef&uuml;hrt. (F&uuml;r Testzwecke, kann als Login
        <i>test</i> als Benutzername und <i>secret</i> als Passwort eingesetzt
        werden.)</font></p>

        </font>

// beginn html formular

        <form method="post" action="bewertung.php">

        <p>

        <table>

        <tr>

            <td><font size="2" face="Verdana, Arial, Helvetica, sans-
            serif">Schule:</font></td>

            <td><font face="Verdana, Arial, Helvetica, sans-serif">

                <input type="school" name="formSchool" size="30" maxlength="30">

            </font></td>

        <tr>

            <td><font size="2" face="Verdana, Arial, Helvetica, sans-
            serif">Benutzername:</font></td>

            <td><font face="Verdana, Arial, Helvetica, sans-serif">

                <input name="formUser" size="30"
                maxlength="30" />

            </font></td>

        <tr>

            <td><font size="2" face="Verdana, Arial, Helvetica, sans-
            serif">Passwort:</font></td>

            <td><font face="Verdana, Arial, Helvetica, sans-serif">

                <input type="password" name="formPassword"

```

```
        size="30" maxlength="30" />
      </font></td>
    <tr>
      <td></td>
      <td><font face="Verdana, Arial, Helvetica, sans-serif">
        <input type="submit" value="Einloggen"
          name="formSubmit" />
      </font></td>
    </tr>
  </table>
  <font face="Verdana, Arial, Helvetica, sans-serif"></p> </font>
</form>

// ende des formulars

  <font face="Verdana, Arial, Helvetica, sans-serif"> <strong></strong></font></td>
</tr>
<tr bgcolor="#999999">
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td><div align="right"><font face="Verdana, Arial, Helvetica, sans-
serif"><strong></strong></font></div></td>
</tr>
</table>
</body>
</html>

// ende des skripts
```

### 3.4.2.2 Erscheinungsbild des Login-Skripts für Schüler

Das im vorigen Kapitel beschriebene Skript zeigt interpretiert folgendes Eingabeformular:

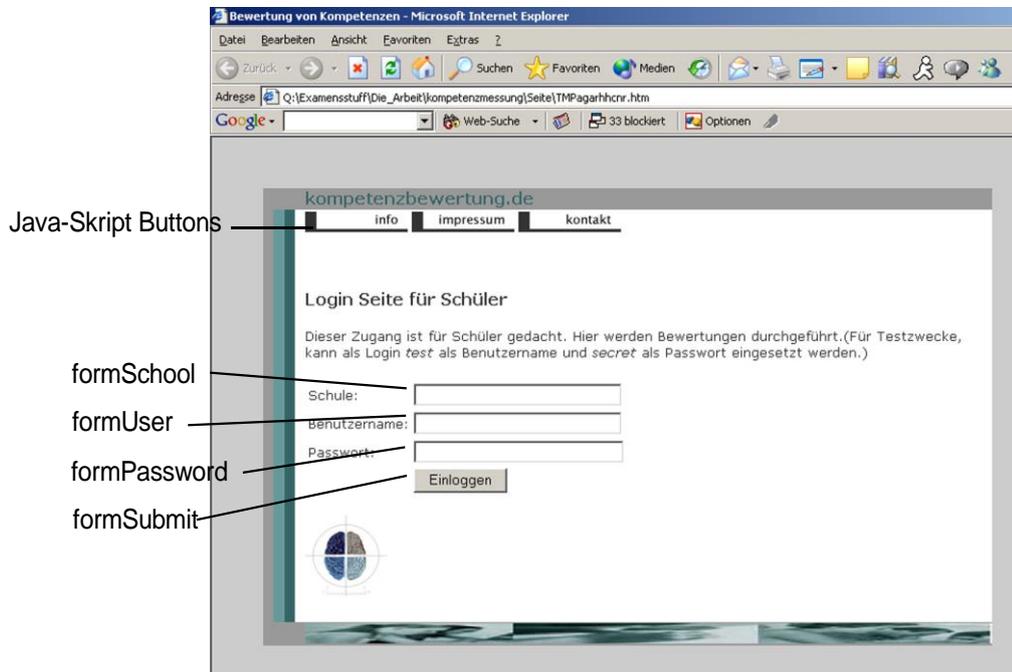


Abbildung 16: Login für Schüler der Web-Applikation

Das PHP-Skript wird von dem Interpreter des Webservers bearbeitet und die Verbindung zur Datenbank hergestellt. Sobald der Button „Einloggen“ gedrückt wird, werden die Inhalte der Formularfelder „formSchool, formUser, formPassword“ durch die SQL-Abfrage mit der Datenbank verglichen. Stimmen die übertragenden Daten nicht mit dem Ergebnis der SQL-Abfrage überein oder werden keine Daten übertragen, wird ein Fehler ausgegeben. Stimmen die Daten überein, gelangt der Schüler zum nächsten Skript, in dem er dann zwischen Bewertung und Auswertung wählen kann.

### 3.4.2.3 Datenbankansicht der Web-Applikation im ER-Modell

Die Administration der Datenbank kann über die in Linux integrierte Shell absolviert werden. Die Shell ist ein Befehlsfenster, über das das gesamte Betriebssystem gesteuert werden kann. Die Eingaben für die Datenbank und deren Tabellen sind jedoch mit großem Aufwand verbunden. Daher erfolgt die Implementierung der Datenbank über das Tool „php-MyAdmin“ (vgl. Kapitel 3.2.2.2), mit dessen Hilfe die Datenbank sowie die einzelnen Tabellen angelegt werden können. Folgende Datenbankstruktur ist entwickelt worden:

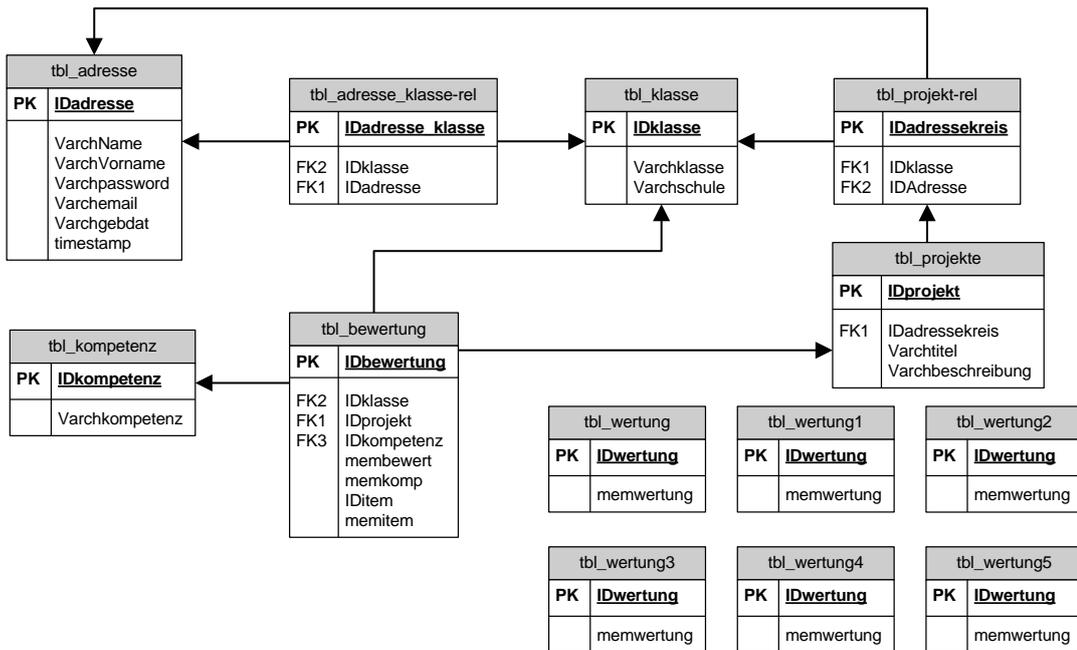


Abbildung 17: Entitäten, Relationen und Attribute der Kompetenz-Datenbank

Diese Abbildung zeigt die Weiterentwicklung des ER-Modells und beschreibt nicht nur die einzelnen Entitäten bzw. „Tabellen“ der Datenbank sondern schon die Primär- und Fremdschlüssel und damit die Relationen der einzelnen Tabellen zueinander. Die Implementierung der Datenbank in MySQL ist über das Tool „phpMyAdmin“ in folgender Abbildung zu erkennen:

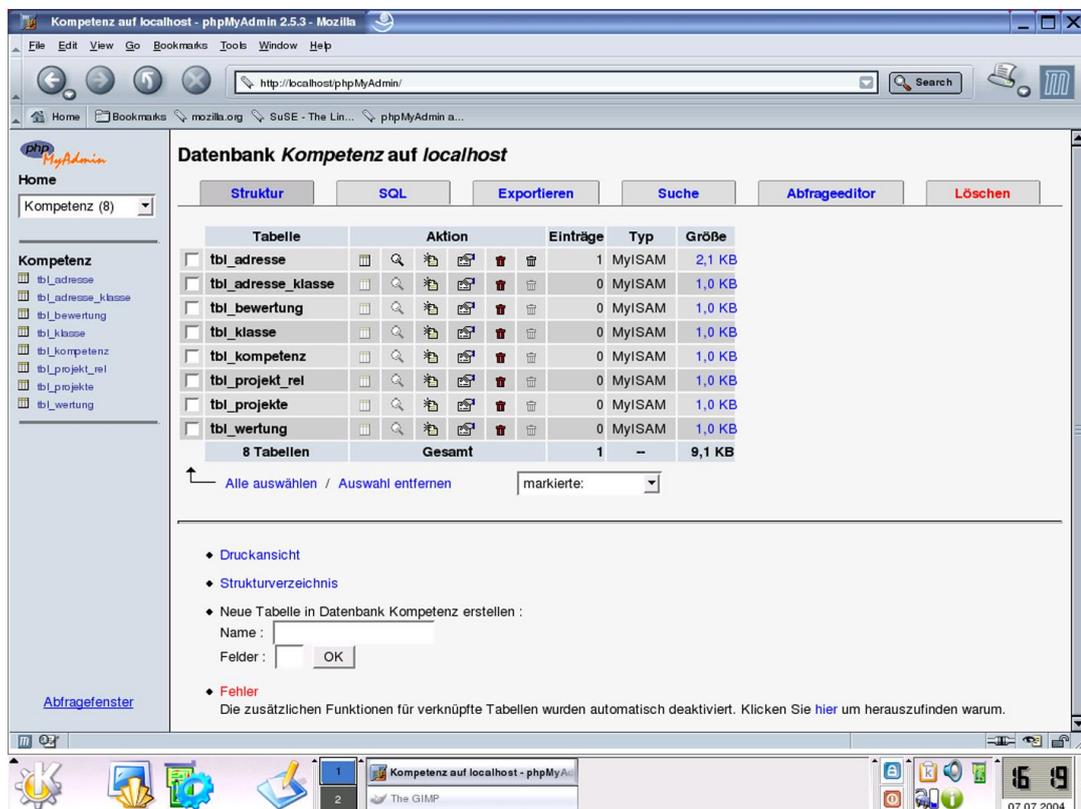


Abbildung 18: MySQL Oberfläche des phpMyAdmin-Tools

In der ersten Zeile steht die aufgerufene Datenbank „Kompetenz“ der Testumgebung. Durch die darunter liegenden Karteireiter können die angelegten Tabellen nach der gewünschten Auswahl bearbeitet werden. Die Tabellen sind in dem ersten Karteireiter „Struktur“ zeilenweise aufgeführt und können hier gelöscht und neu angelegt werden. Hinter den Tabellen können div. Editier-Aktionen durchgeführt werden, die Anzahl der vorhandenen Einträge ist aufgelistet und der Tabellen-Typ und -Größe werden genannt.

Als Tabellen-Typ ist für die Web-Applikation der MyISAM-Dynamic-Typ ausgewählt. Dieser Typ wird standardmäßig in MySQL-Datenbanken eingesetzt und zeichnet sich durch besondere Effizienz, hohe Datensicherheit und geringen Platzbedarf aus. Andere Tabellen-Typen sind (vgl. KOFLEER 2003: 199 ff.) z. B. InnoDB-Tabellen, Gemini Merge, BDB usw. Auf die Eigenschaften aller Tabellen-typen wird hier nicht weiter eingegangen.

### **3.4.3 Teilergebnisse der Implementierungsphase**

Folgende Teilergebnisse der Implementierungsphase sind bereits erreicht:

- Einrichtung des Webservers,
- Anlegung von Sicherheitsmechanismen,
- Programmierung des Login-Skripts,
- Anlegung der Datenbank nach dem ER-Modell.

Die komplette Programmierung der Skripte sowie die weiteren Implementierungsschritte erfolgen nach dem in Kapitel 3.2.1.4 beschriebenen Ablaufdiagramm. Die Testphase beginnt demnach am 28.07.2004 und endet am 13.08.2004. Für die Testphase ist es wichtig, dass das System nicht nur in der Testumgebung funktioniert sondern auch an das Internet angebunden wird.

## 4 Fazit

Gegenstände dieser Arbeit waren die Konzeption und die Entwicklung der Web-Applikation zur Unterstützung der Fremd- und Selbstbewertung. In dieser Arbeit dienten daher Modelle der Software-Entwicklung als Grundlage für die Beschreibung der Web-Applikation. In der Praxis der Software-Entwicklung haben sowohl die Modelle als auch die Instrumentarien der Lasten- und Pflichtenhefte ihren festen Stand. Die Beschreibung von Techniken der Software-Entwicklung anhand der Web-Applikation besitzt nicht den Anspruch der Vollständigkeit. Der Einsatz der Modellierungsverfahren zeigt aber einen großen praktischen Nutzen, der auch für Projekte im Informatikunterricht anwendbar ist.

Die Konzeption der Web-Applikation befindet sich zunächst in der Implementierungsphase. Die abschließende Test- und Überprüfungsphase beginnt laut Projektplan am 28.07.2004.

Die Realisierung des Prototyps erforderte einen umfangreichen Erwerb von Kenntnissen. Angefangen im Grundlagenbereich in der Behandlung von Webservern und Datenbanken bis hin zur PHP- und SQL-Programmierung. Wobei die Nutzung von PHP nicht ohne die Kenntnisse von HTML und der objektorientierten Programmierung möglich ist.

Durch das Schreiben der Arbeit musste ich die Entwicklung „einfrieren“. Daher ist es fraglich, ob die Testphase planmäßig beginnen kann. Eine *Domain* ist für die Web-Applikation bereits bei einem Internet-Provider gesichert. Um die Funktionsfähigkeit der Web-Applikation nachzuweisen, wird bereits zur Abgabe der Arbeit die fertige Version des Prototyps mit Nutzung des Login-Bereichs zunächst an das Internet angebunden sein und über folgende Adresse erreichbar sein:

<http://www.kompetenzbewertung.de>

Der spätere Einsatz der fertigen Web-Applikation wird dann durch Herrn Prof. Dr. Thomas Vollmer koordiniert.

## 5 Glossar:

|           |  |
|-----------|--|
| Apache    | Ein Webserverdienst, der auf Open Source beruht.   |
| Backend   | Gegenstück zum Frontend. Technik, die im Hintergrund läuft und die vom Benutzer über das Frontend eingegebenen Daten verarbeitet. Die Techniken sind z.B Datenbanken und Interpreter, die auf dem Webserver laufen und dann das Ergebnis an das Frontend schickt.  |
| Browser   | Ein <i>Client</i> -Programm für das „Stöbern“ (englisch: to browse) im Web. Versteht HTML, meist auch Java-Skript und kann Java-Applets ausführen. Es gibt verschiedene Hersteller, z.B. Netscape, Microsoft und Opera.  |
| Client    | Ein Computer oder Programm, das die Dienste eines anderen Rechners (>Servers) in Anspruch nimmt.   |
| Cookies   | Dateien, die beim Besuch mancher Websites auf der Festplatte des Besuchers abgelegt werden. Cookies sind Text-Dateien und dienen der (Zwischen-)Speicherung von Daten. Cookies dürfen eine maximale Größe von 4 KB nicht überschreiten. Sie beinhalten oft Informationen über Besuchsdauer, besuchte Seiten oder Eingaben des Besuchers. |
| CSS       | Cascading style sheets ermöglichen die Trennung von Inhalt und Layout bei der Gestaltung von Webseiten. Layouteigenschaften wie z.B. Layer, Schriftarten und Schriftgrößen werden über CSS ähnlich wie Formatvorlagen deklariert. CSS können auch aus HTML-Dateien ausgelagert werden.   |
| Dämon     | Abkürzung für „Disk And Execution Monitor“. Ein Server-Programm, das nur auf Anfrage eines Clients aktiviert wird.   |
| Datenbank | Eine Datenbank ist ein elektronisches System zum Speichern, Verwalten und Auslesen von Daten. Man unterscheidet Datenbanken sehr häufig nach ihrer technologischen Ausrichtung, relational oder objektorientiert.  |
| DB        | Abkürzung für Datenbank.   |

---

|          |   |
|----------|---|
| DBMS     | Das <b>Datenbankmanagementsystem</b> verwaltet und kontrolliert zentral unter Berücksichtigung des Datenbankschemas die Datenbestände der Datenbank.  |
| DD       | Abkürzung für das „Data Dictionary“, in dem das Schema der Datenbank gespeichert ist.   |
| Domain   | Eine Domain ist eine eindeutige textuelle Beschreibung einer Adresse im Internet. Die Domainnamen sind hierarchisch von rechts nach links gegliedert. Der ganz rechte Teil nach dem Punkt heißt Top Level Domain, der davor Second Level Domain oder einfach Domain. Alle weiteren Namensteile links davon sind jeweils Sub-Domains.                    |
| Frontend | Der Teil des Programms, mit dem der Endbenutzer in Berührung kommt. Im Fall der Web-Applikation ist es das Formular, mit dem der Benutzer die Eingaben tätigt.  |
| HTML     | Abkürzung für Hypertext Markup Language, die Seitenbeschreibungssprache des Web. Über HTML wird die logische Struktur eines Dokumentes festgelegt, nicht jedoch das genaue Layout. Für das Layout ist der Browser zuständig. HTML basiert auf SMGL (Standard Generalized Markup Language), einer ISO-Norm zur Definition von strukturierten Datentypen. |
| HTTP     | Abkürzung für Hypertext Transfer Protokoll. HTTP regelt die Kommunikation zwischen Browser und Webserver und wird im Web als Übertragungsprotokoll verwendet.   |
| iterativ | Eine wiederholende Tätigkeit, die Entwicklung schrittweise, aber zielgerichtet durchzuführen.   |
| LAMP     | Abkürzung für Linux/Apache/MySQL/PHP. LAMP beschreibt eine beliebte Serverkonfiguration, die vollständig aus Open Source-Komponenten besteht.   |
| Linux    | Ein Betriebssystem, entwickelt von Linus Torvald, das als UNIX Derivat gilt.  |
| MySQL    | Relationales Datenbanksystem, das vorwiegend für den Interneteinsatz gebraucht wird. Open Source Lizenz.  |

---

|                    |  |
|--------------------|--|
| Open Source        | Steht für Programme, deren Quellcode veröffentlicht worden ist. Die Lizenz zum Betreiben und Verändern der Programme steht in den Lizenzbestimmungen (Anhang C) (vgl. SNOOPY 2000)   |
| PHP                | Abkürzung für Hypertext Preprozessor. PHP ist als Interpretersprache in HTML eingebettet und führt serverseitige Aktionen durch.   |
| redundant          | Einträge sind doppelt oder mehrfach in einer Datenbank vorhanden. Es kann zu Fehlern oder Abstürzen von Programmen kommen.   |
| Session Management | Das Sessionmanagement stellt eine zentrale Funktion dar, die entscheidend für die komfortable Benutzung der Web-Applikation ist. Durch das Sessionmanagement stehen dem Benutzer nach einmaliger Anmeldung die nachfolgenden Seiten zur Verfügung. Ein Logout erfolgt erst nach einem manuellen Logout oder nach einem vordefinierten Zeitabschnitt. |
| SQL                | Abkürzung für „structured query language“. SQL ist eine Verwaltungssprache für Datenbanken, keine Programmiersprache.  |
| Tags               | Tags sind Sonderzeichen, die einen Kommentar oder eine Programmiersprache einleiten.   |
| Template           | Die englische Bezeichnung für Schablone. Die Seiten der Web-Applikation werden so benannt, da sie eine Schablone als Vorbild haben.  |
| URL/URI            | Abkürzung für Uniform Resource Locator bzw. Identifier. Dabei handelt es sich um Zeichenketten, die eine Quelle im Web identifizieren, zum Beispiel Dokumente, Dateien, Dienste, Email oder anderes.   |
| Webserver          | Ein Server, der Webseiten und andere Dienste bereitstellt.   |

## Literaturverzeichnis

**BADER, R (1989):** Berufliche Handlungskompetenz. In: Die berufsbildende Schule 41.

**BALZERT, H (2000):** Lehrbuch der Software Technik Bd.1. Software-Entwicklung. Berlin: Spektrum, Akad. Verl.

**BERBEN, T. / JIRITSCHKA, M. / STAMMER, R. / VOLLMER, T. (2003):** Abschlussbericht zum BLK-Modellversuch „Förderung arbeitsprozessbezogener Kompetenzen zwischen den Lernorten Schule und Betrieb (FÖRAK)“. Hamburg.

**BOEHM, B. (1981):** Wirtschaftliche Software-Produktion. Wiesbaden: Forkel.

**KMK (2000):** Handreichung für die Erarbeitung von Rahmenlehrplänen der Kultusministerkonferenz (KMK) für den berufsbezogenen Unterricht in der Berufsschule und ihre Abstimmung mit Ausbildungsordnungen des Bundes für anerkannte Ausbildungsberufe (Stand: 15.09.2000). Hrsg.: Sekretariat der Ständigen Konferenz der Kultusminister der Länder in der Bundesrepublik Deutschland. In: <http://www.kultusministerkonferenz.de/beruf/home.htm> (01.04.2001).

**KOFLER, M. (2002):** Linux Installation, Konfiguration, Anwendung. München: Addison-Wesley.

**KOFLER, M. (2003):** MySQL Einführung, Programmierung, Referenz. München: Addison-Wesley.

**LORENZ; P. (2002):** ASP.NET Grundlagen und Profiwissen Webserverprogrammierung und XML Web Services im .NET-Framework. München: Hanser, 40-50.

**OTT, B. (2000):** Grundlagen des beruflichen Lernens und Lehrens, Berlin: Cornelsen, 219 – 223.

**ROßBACH, S. (1998):** Der Apache Webserver. Einrichtung und Verwaltung. München: Addison-Wesley

**RÜTTGER, M. (2003):** Das Webdatenbank Pflichtenheft. Bonn: mitp.

**SNOOPY, M. (2000):** Open-Source kurz & gut. Köln: O'Reilly.

**SOMMERVILLE, I. (2001):** Software Engineering, 6. Auflage. München: Pearson Studium.

**SCHMIDT, J. (2001):** Sicherheitsrisiko Microsoft. Die Kehrseite des Windows Komforts. In: c't Magazin für Computer Technik (2001/21). Hannover: Heise, 121-125

**SCHULZ, H. / SIERING, P. (2003):** Open-Source Datenbanken im Vergleich. In: c't Magazin für Computer Technik (2003/05). Hannover: Heise, 142 – 148

**VOSSEN, G. (2000):** Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme. München: Oldenbourg, 80

**Internethinweise:**

NETCRAFT: [http://news.netcraft.com/archives/web\\_server\\_survey.html](http://news.netcraft.com/archives/web_server_survey.html)

W3C: <http://www.w3c.org>

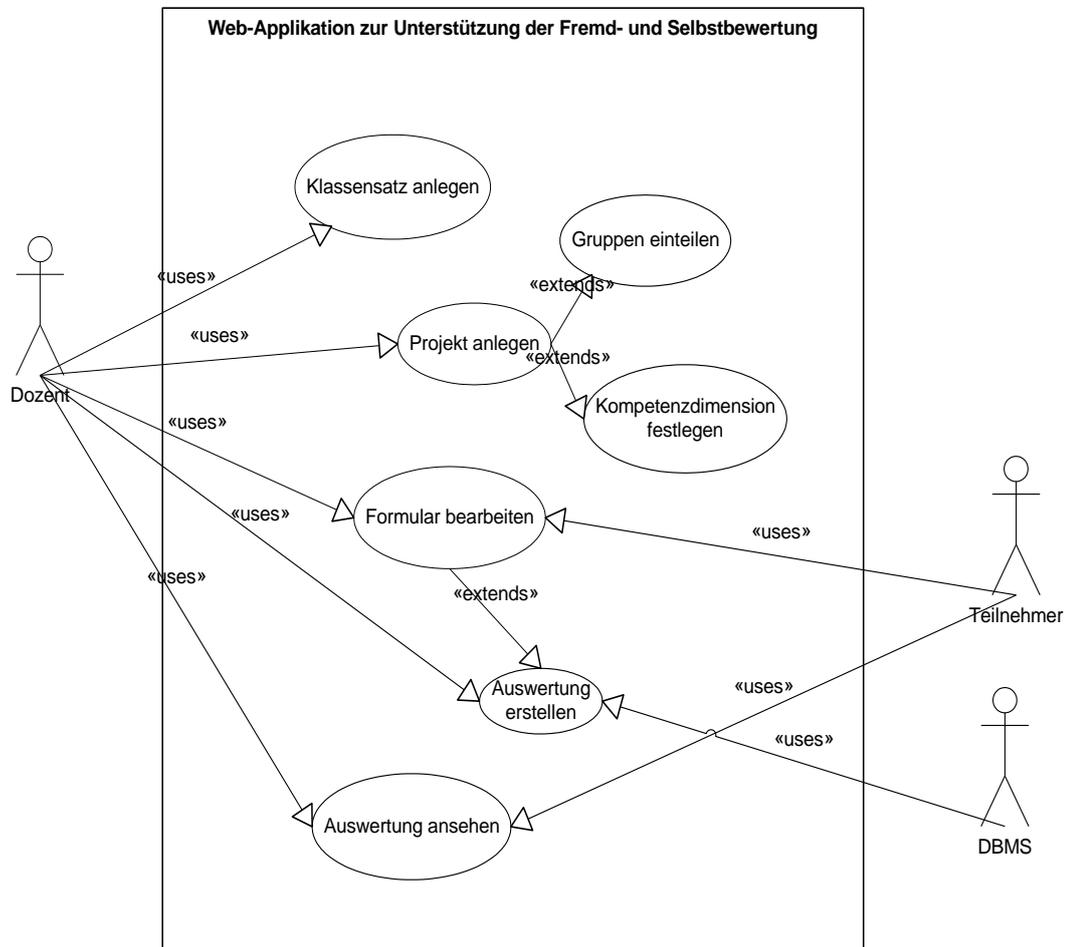
Linux <http://www.linuxiso.org>

Apache <http://www.apache.org/>

MySQL <http://www.mysql.com>

PHP <http://www.php.net>

## Anhang A: Anwendungsfall- und Aktivitätsdiagramme



Schriftliche Ergänzung für das Anwendungsfalldiagramm

Kurzbeschreibung:

In diesem Diagramm werden die Anwendungsfälle des zu erstellenden Datenbankmoduls erläutert.

Dieser Modul soll für Bereiche der Ergebnissicherung eingesetzt werden. Ziel dieser Anwendung ist es, mit Hilfe einer Datenbank eine Bewertung von Kompetenzen zu ermöglichen. Da Kompetenzbegriffe unterschiedlich verwendet werden, ist eine Bewertung bezüglich des Projektes- und Lerngegenstandes individuell zu gestalten.

AF-Nr.1: Klassensatz anlegen

Akteure: Lehrer

Kurzbeschreibung: Der Lehrer erstellt einen Klassensatz. Er hat die Möglichkeit, alle eingegebenen Daten bezüglich der Klassen- und Schülerdaten zu editieren.

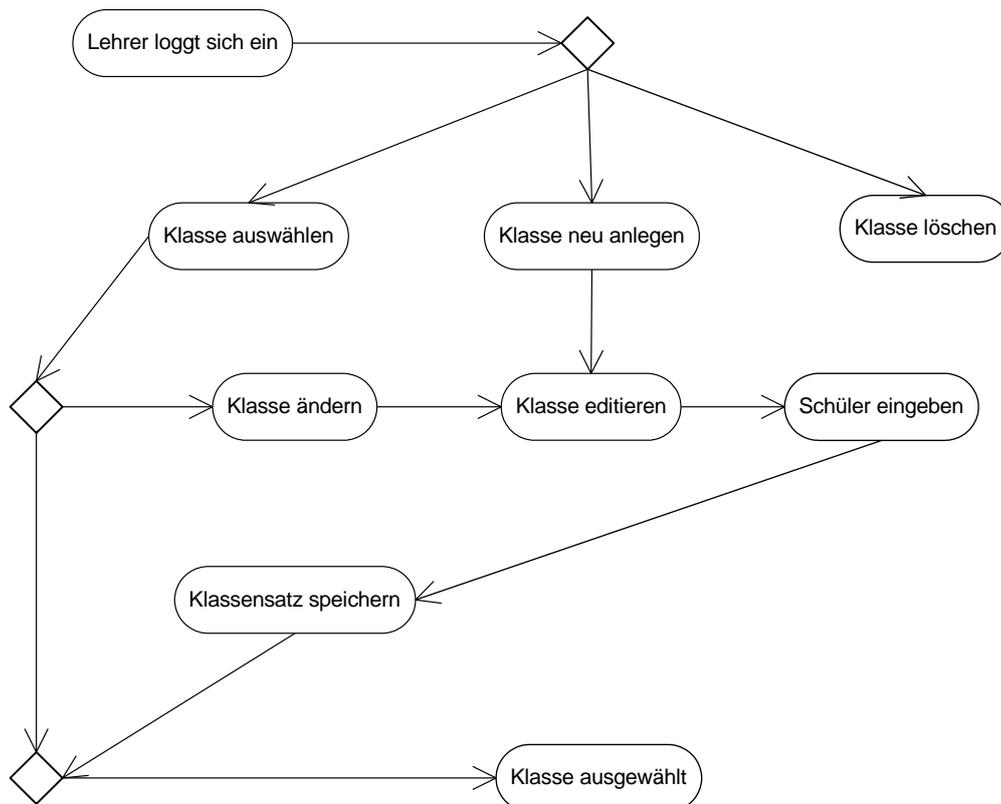
Auslöser/Vorbedingung: Der Lehrer gibt die Daten über ein Eingabeformular in die Datenbank. Das Eingabeformular und die nötige Datenbank sind bereit zur Eingabe.

Ergebnis: Der Lehrer hat einen oder mehrere Klassensätze erstellt.

Essentieller Ablauf:

1. Lehrer loggt sich auf der Seite ein.
2. Neuerstellung der Klassen.
3. Speicherung des Klassensatzes.
4. Klassensatz ist erzeugt.

Diagramme: Aktivitätsdiagramme siehe Anlage 1



AF-Nr.2: Projekte anlegen

Akteure: Lehrer

Auslöser/

Vorbedingung: Der Lehrer gibt die Daten über ein Eingabeformular in die Datenbank. Das Eingabeformular und die nötige Datenbank sind bereit zur Eingabe.

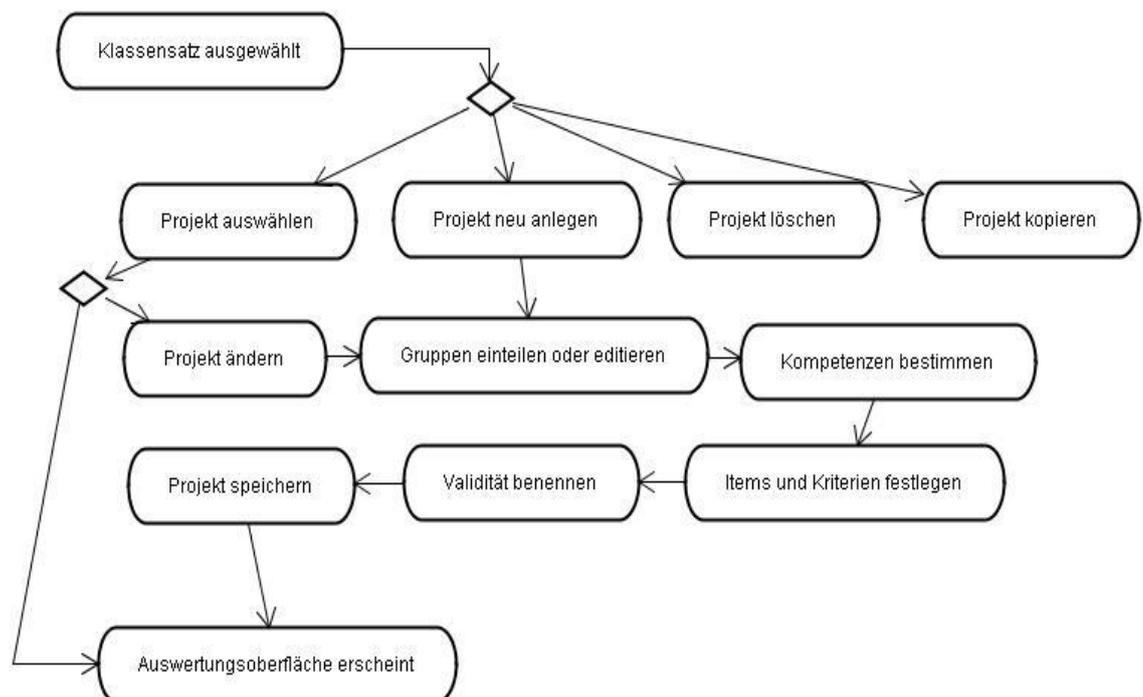
Ergebnis: Der Lehrer hat einen oder mehrere Datensätze erstellt. Die Projekte sind nun den Klassen und den Schülern zugeordnet. Über diese Projekte werden nun die Kompetenzdimensionen bestimmt.

Essentieller

Ablauf:

1. Der Lehrer wählt den zuvor erstellen Klassensatz aus.
2. Der Lehrer legt den Projektgegenstand an.
3. Über das Projekt greift er auf den zuvor angelegten Klassensatz zu und legt die Gruppeneinteilung der Schüler fest.
4. Nach Erstellen der Gruppen bestimmt der Lehrer die zu bewertenden Kompetenzen.

Diagramme: Aktivitätsdiagramme siehe Anlage 2



AF-Nr.3: Gruppen einteilen(include)

Akteure: Lehrer

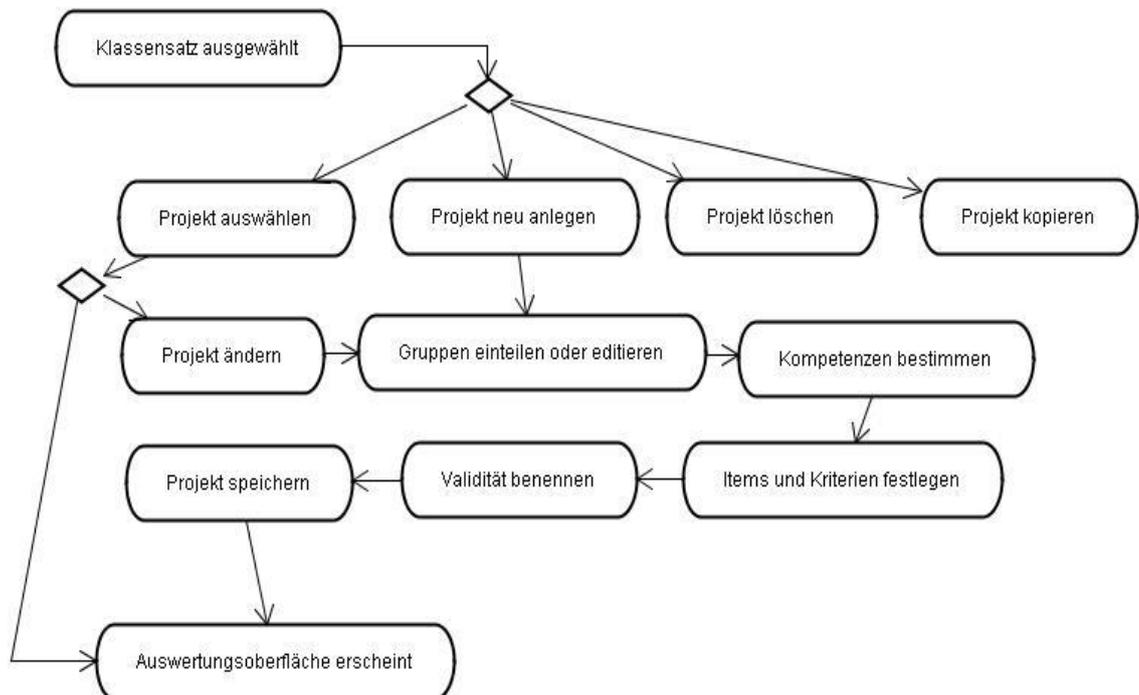
Auslöser/

Vorbedingung: Der Lehrer hat nach dem Erstellen des Klassensatzes zu der jeweiligen Klasse die Projekte zugeordnet und teilt nun die Gruppen-, Partner- oder Einzelarbeiten den Schülern der Klasse zu. Das Eingabeformular ist bereit zum Speichern der Daten

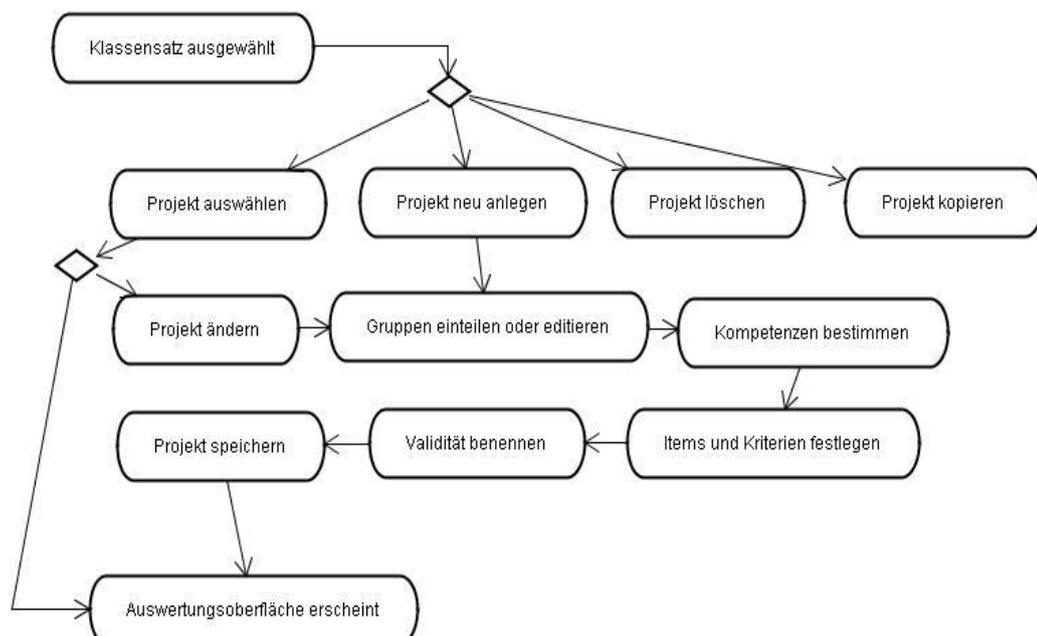
Ergebnis: Der Lehrer hat zu dem Projekt die Gruppen eingeteilt. Der Schülersatz ist mit der Zugehörigkeit zur Gruppen-, Partner oder Einzelarbeit erweitert worden. Als Ergebnis erhält der Lehrer eine Auflistung der Gruppen.

Essentieller

- Ablauf:
1. Der Lehrer hat den Klassensatz angelegt und das Projekt bzw. den Lerngegenstand festgelegt.
  2. Zu dem Projekt teilt der Lehrer die von den Schülern zu erbringenden Arbeiten nach Gruppen-, Partner oder Einzelarbeit ein.
  3. Als Ergebnis hat der Lehrer nun die Schüler eingeteilt und das Projekt mit der Klasse verbunden.



- AF-Nr.4: Kompetenzdimensionen festlegen (include)
- Akteure: Lehrer
- Auslöser/
- Vorbedingung: Der Lehrer hat den Klassensatz angelegt, das Projekt mit dem Klassensatz verbunden und somit die Schülern in ihre Arbeitsformen eingeteilt.
- Ergebnis: Der Lehrer hat für die Schüler individuell die zu bewertenden Kompetenzen festgelegt. Er hat Kompetenzdimensionen bestimmt und für das Projekt angepasste Bewertungskriterien aufgestellt. Die Begriffe für die Validitätsskalen hat der Lehrer ebenfalls festgelegt.
- Essentieller
- Ablauf:
1. Der Lehrer bestimmt die Kompetenzbegriffe und legt Dimensionen an.
  2. Der Lehrer stellt zu den Dimensionen Bewertungskriterien auf.
  3. Der Lehrer Legt die Bezeichnung an den Validitätsskalen fest.
  4. Er speichert den Datensatz ab.
  5. Er kann in dieser Oberfläche die angelegten Kompetenzen und Kriterien verwalten.
  6. Der Lehrer speichert den Datensatz ab und beendet somit die Arbeitsoberfläche.



AF-Nr.5: Formular bearbeiten

Akteure: Schüler, Lehrer

Auslöser/

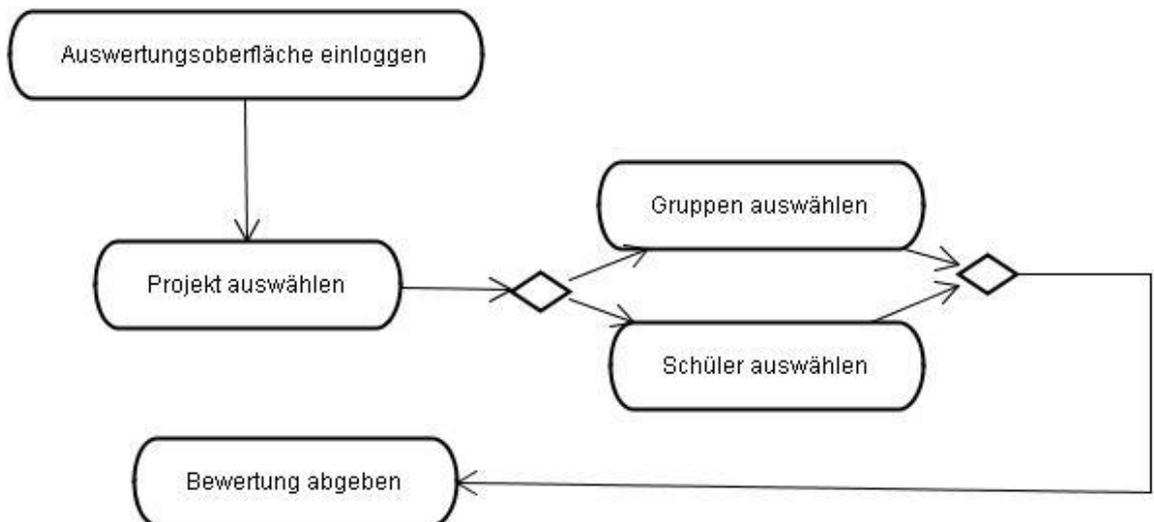
Vorbedingung: Der Lehrer hat auf der Arbeitsoberfläche die Inhalte für die Formulare und Abfragen eingegeben.

Ergebnis: Die Akteure wählen nach ihrem Login den zu bewertenden Projekt- oder Lerngegenstand aus, Durchlaufen die Abfragen und Formulare und erzeugen somit einen Datensatz, der für die Auswertung als Basis dient.

Essentieller

- Ablauf:
1. Akteure loggen sich ein
  2. Sie wählen Klasse und Projekt aus
  3. Durchlaufen die zuvor erstellten Abfragen und Formulare
  4. Die Akteure speichern die Datensätze ab.

Diagramm:



AF-Nr.6: Auswertung erstellen

Akteure: Lehrer, Interpret

Auslöser/

Vorbedingung: Lehrer und Schüler haben AF- Nr.5 durchlaufen und der Lehrer hat sich auf der Auswertungsoberfläche eingeloggt.

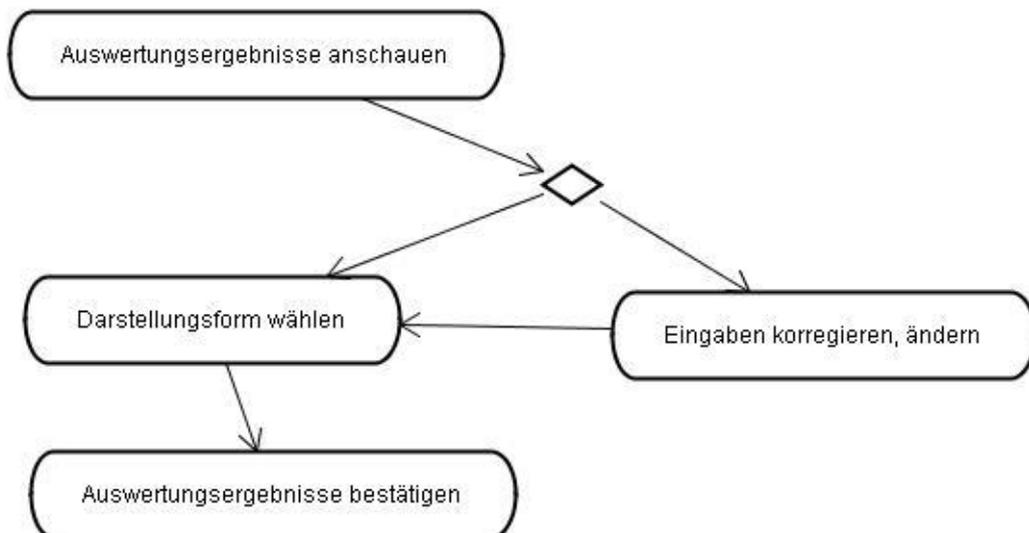
Ergebnis: Der Lehrer kann sich anschauen, je nach Klasse und Projekt, welche Auswertungen stattgefunden haben. Er kann Schüler freischalten, die eine Falscheingabe getätigt haben. Er kann die Auswertung beeinflussen.

Essentieller

Ablauf:

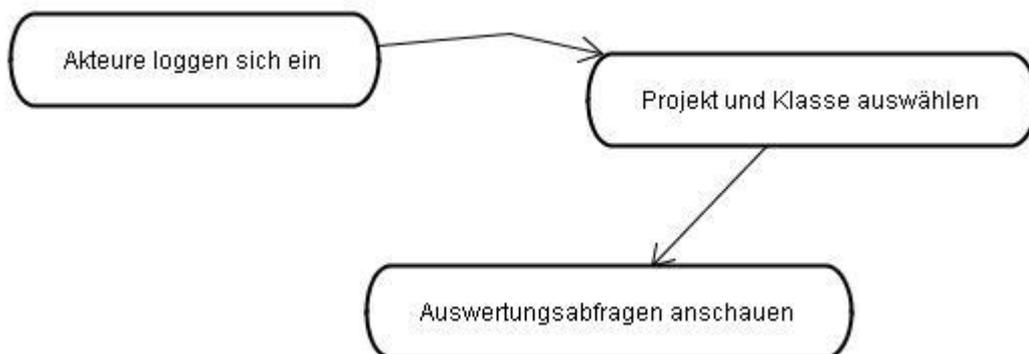
1. Der Lehrer loggt sich auf der Auswertungsoberfläche ein
2. Er wählt Klasse und Projekt aus
3. Der Lehrer editiert ggf. die Auswertungen
4. Der Lehrer übergibt den Datensatz und speichert die Auswertung.
5. Die Auswertung ist fertiggestellt

Diagramme:



|               |  |
|---------------|--|
| AF-Nr.7:      | Auswertung ansehen   |
| Akteure:      | Lehrer, Schüler  |
| Auslöser/     |  |
| Vorbedingung: | Der Lehrer hat AF- Nr.6 durchlaufen und hat die Auswertung erstellt.   |
| Ergebnis:     | Die Akteure können sich in unterschiedlichen Darstellungen, je nach Klasse und Projekt anschauen, wie die Auswertungen aussehen.   |
| Essentieller  |  |
| Ablauf:       | <ol style="list-style-type: none"><li>1. Die Akteure loggen sich auf der Auswertungsoberfläche ein.</li><li>2. Sie wählen Klasse und Projekt aus</li><li>3. Sie betrachten die Auswertungsergebnisse und können sie ausdrucken.</li><li>4. Die Kompetenzbewertung ist beendet. Der Datensatz bleibt so lange gespeichert, bis der Klassensatz oder das Projekt gelöscht wurde.</li></ol> |

Diagramme:



## Anhang B: Lizenzbestimmung

### The PHP License

Redistribution and use in source and binary forms, with or without modification, is permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name "PHP" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact [group@php.net](mailto:group@php.net).
4. Products derived from this software may not be called "PHP", nor may "PHP" appear in their name, without prior written permission from [group@php.net](mailto:group@php.net). You may indicate that your software works in conjunction with PHP by saying "Foo for PHP" instead of calling it "PHP Foo" or "phpfoo".
5. The PHP Group may publish revised and/or new versions of the license from time to time. Each version will be given a distinguishing version number. Once covered code has been published under a particular version of the license, you may always continue to use it under the terms of that version. You may also choose to use such covered code under the terms of any subsequent version of the license published by the PHP Group. No one other than the PHP Group has the right to modify the terms applicable to covered code created under this License.
6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes PHP, freely available from [<http://www.php.net/>](http://www.php.net)".

This software consists of voluntary contributions made by many individuals on behalf of the PHP Group.

The PHP Group can be contacted via Email at [group@php.net](mailto:group@php.net).

7. For more information on the PHP Group and the PHP project, please see [<http://www.php.net>](http://www.php.net)

## Open Software License v. 2.1

This Open Software License (the "License") applies to any original work of authorship (the "Original Work") whose owner (the "Licensor") has placed the following notice immediately following the copyright notice for the Original Work:

Licensed under the Open Software License version 2.1

1) Grant of Copyright License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license to do the following:

- \* to reproduce the Original Work in copies;
- \* to prepare derivative works ("Derivative Works") based upon the Original Work;
- \* to distribute copies of the Original Work and Derivative Works to the public, with the proviso that copies of Original Work or Derivative Works that You distribute shall be licensed under the Open Software License;
- \* to perform the Original Work publicly; and
- \* to display the Original Work publicly.

2) Grant of Patent License. Licensor hereby grants You a world-wide, royalty-free, non-exclusive, perpetual, sublicenseable license, under patent claims owned or controlled by the Licensor that are embodied in the Original Work as furnished by the Licensor, to make, use, sell and offer for sale the Original Work and Derivative Works.

3) Grant of Source Code License. The term "Source Code" means the preferred form of the Original Work for making modifications to it and all available documentation describing how to modify the Original Work. Licensor hereby agrees to provide a machine-readable copy of the Source Code of the Original Work along with each copy of the Original Work that Licensor distributes. Licensor reserves the right to satisfy this obligation by placing a machine-readable copy of the Source Code in an information repository reasonably calculated to permit inexpensive and convenient access by You for as long as Licensor continues to distribute the Original Work, and by publishing the address of that information repos-

itory in a notice immediately following the copyright notice that applies to the Original Work.

4) Exclusions From License Grant. Neither the names of Licensor, nor the names of any contributors to the Original Work, nor any of their trademarks or service marks, may be used to endorse or promote products derived from this Original Work without express prior written permission of the Licensor. Nothing in this License shall be deemed to grant any rights to trademarks, copyrights, patents, trade secrets or any other intellectual property of Licensor except as expressly stated herein. No patent license is granted to make, use, sell or offer to sell embodiments of any patent claims other than the licensed claims defined in Section 2. No right is granted to the trademarks of Licensor even if such marks are included in the Original Work. Nothing in this License shall be interpreted to prohibit Licensor from licensing under different terms from this License any Original Work that Licensor otherwise would have a right to license.

5) External Deployment. The term "External Deployment" means the use or distribution of the Original Work or Derivative Works in any way such that the Original Work or Derivative Works may be used by anyone other than You, whether the Original Work or Derivative Works are distributed to those persons or made available as an application intended for use over a computer network. As an express condition for the grants of license hereunder, You agree that any External Deployment by You of a Derivative Work shall be deemed a distribution and shall be licensed to all under the terms of this License, as prescribed in section 1(c) herein.

6) Attribution Rights. You must retain, in the Source Code of any Derivative Works that You create, all copyright, patent or trademark notices from the Source Code of the Original Work, as well as any notices of licensing and any descriptive text identified therein as an "Attribution Notice." You must cause the Source Code for any Derivative Works that You create to carry a prominent Attribution Notice reasonably calculated to inform recipients that You have modified the Original Work.

7) Warranty of Provenance and Disclaimer of Warranty. Licensor warrants that the copyright in and to the Original Work and the patent rights granted herein by Licensor are owned by the Licensor or are sublicensed to You under the terms of this License with the permission of the contributor(s) of those copyrights and patent rights. Except as expressly stated in the immediately preceding sentence, the Original Work is provided under this License on an "AS IS" BASIS and

WITHOUT WARRANTY, either express or implied, including, without limitation, the warranties of NON-INFRINGEMENT, MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY OF THE ORIGINAL WORK IS WITH YOU. This DISCLAIMER OF WARRANTY constitutes an essential part of this License. No license to Original Work is granted hereunder except under this disclaimer.

8) Limitation of Liability. Under no circumstances and under no legal theory, whether in tort (including negligence), contract, or otherwise, shall the Licensor be liable to any person for any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or the use of the Original Work including, without limitation, damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses. This limitation of liability shall not apply to liability for death or personal injury resulting from Licensor's negligence to the extent applicable law prohibits such limitation. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this exclusion and limitation may not apply to You.

9) Acceptance and Termination. If You distribute copies of the Original Work or a Derivative Work, You must make a reasonable effort under the circumstances to obtain the express assent of recipients to the terms of this License. Nothing else but this License (or another written agreement between Licensor and You) grants You permission to create Derivative Works based upon the Original Work or to exercise any of the rights granted in Section 1 herein, and any attempt to do so except under the terms of this License (or another written agreement between Licensor and You) is expressly prohibited by U.S. copyright law, the equivalent laws of other countries, and by international treaty. Therefore, by exercising any of the rights granted to You in Section 1 herein, You indicate Your acceptance of this License and all of its terms and conditions. This License shall terminate immediately and you may no longer exercise any of the rights granted to You by this License upon Your failure to honor the proviso in Section 1(c) herein.

10) Termination for Patent Action. This License shall terminate automatically and You may no longer exercise any of the rights granted to You by this License as of the date You commence an action, including a cross-claim or counterclaim, against Licensor or any licensee alleging that the Original Work infringes a patent. This termination provision shall not apply for an action alleging patent infringement by combinations of the Original Work with other software or hardware.

11) Jurisdiction, Venue and Governing Law. Any action or suit relating to this License may be brought only in the courts of a jurisdiction wherein the Licensor resides or in which Licensor conducts its primary business, and under the laws of that jurisdiction excluding its conflict-of-law provisions. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any use of the Original Work outside the scope of this License or after its termination shall be subject to the requirements and penalties of the U.S. Copyright Act, 17 U.S.C. Â§ 101 et seq., the equivalent laws of other countries, and international treaty. This section shall survive the termination of this License.

12) Attorneys Fees. In any action to enforce the terms of this License or seeking damages relating thereto, the prevailing party shall be entitled to recover its costs and expenses, including, without limitation, reasonable attorneys' fees and costs incurred in connection with such action, including any appeal of such action. This section shall survive the termination of this License.

13) Miscellaneous. This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

14) Definition of "You" in This License. "You" throughout this License, whether in upper or lower case, means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License. For legal entities, "You" includes any entity that controls, is controlled by, or is under common control with you. For purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

15) Right to Use. You may use the Original Work in all ways not otherwise restricted or conditioned by this License or by law, and Licensor promises not to interfere with or be responsible for such uses by you.

This license is Copyright (C) 2003-2004 Lawrence E. Rosen. All rights reserved. Permission is hereby granted to copy and distribute this license without modification. This license may not be modified without the express written permission of its copyright owner.

## Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe verfasst und mich dabei anderer als der angegebenen Hilfsmittel nicht bedient habe.

Ich bin mit einer späteren Ausleihe der Arbeit nicht einverstanden.

---

Ort, Datum

---

Unterschrift