

**Universität Hannover
Fachgebiet Software Engineering
Institut für Informationssysteme
Fachbereich Informatik**

**Konzept und Realisierung eines Werkzeugs zur
Erhebung, Verwaltung und Bewertung
von Erfahrungen**

Bachelorarbeit

im Studiengang Angewandte Informatik

von

Thomas Liro

**Prüfer: Prof. Dr. Kurt Schneider
Zweitprüfer: Prof. Dr. Rainer Parchmann**

Langenhagen, 31. August 2004

Zusammenfassung

Die Qualitätsverbesserung von Software spielt eine immer größere Rolle, weil die Software-Produkte stets mächtiger und komplexer werden. Dabei gewinnt die erfahrungsbasierte Qualitätsverbesserung stets an Bedeutung. Damit aber Erfahrungen bei ihrer Wiederverwendung ein adäquates Hilfsmittel darstellen, müssen sie in einer geeigneten Form behandelt und schließlich zur erneuten Nutzung angeboten werden.

In dieser Arbeit wird ein Werkzeug entwickelt, das den Umgang mit Erfahrungen – von ihrer Erhebung über die Speicherung und Aufbereitung bis hin zur Wiederverwendung – ermöglicht und so den Software-Entwicklungsprozess unterstützt.

Ich danke Herrn Prof. Dr. Kurt Schneider und Herrn Dipl.-Wirt.-Inf. Daniel Lübke für die stets hervorragende Betreuung meiner Bachelorarbeit.

Inhaltsübersicht

1	Einleitung	2
1.1	Problemstellung	2
1.2	Zielsetzung	2
1.3	Gliederung	2
2	Nutzung von Erfahrungen in der Software-Technik	4
2.1	Warum Erfahrungen?	4
2.2	Erfahrungskreislauf	5
2.2.1	Erfahrungserfassung	6
2.2.2	Erfahrungsaufbereitung	7
2.2.3	Erfahrungsaufbewahrung	7
2.2.4	Erfahrungsnutzung	8
2.3	Konzept der Experience Factory	8
3	Bewertung von Erfahrungen	10
4	Realisierung einer Erfahrungsdatenbank	13
4.1	Vorüberlegungen und Konzept	13
4.2	Wahl des Entwicklungsmodells	14
4.3	Entwicklungsprozess	15
4.3.1	Analyse	16
4.3.2	Entwurf	20
4.3.3	Verteilung und Inbetriebnahme	31
4.4	Anwendungsszenario	33
4.5	Möglichkeiten der Erweiterung	33
5	Einordnung der Erfahrungsdatenbank in übergeordnete Konzepte	35
6	Zusammenfassung (und Ausblick)	37
A)	Abbildungsverzeichnis	39
B)	Tabellenverzeichnis	40
C)	Literaturverzeichnis	41

1. Einleitung

1.1 Problemstellung

Im Zuge von immer komplexer und mächtiger werdenden Softwaresystemen sind Qualität und monoton steigende Kosten zu einem Problem geworden. Vor allem leidet die Qualität, weil sich Entwickler einem stetig größer werdenden Kostendruck ausgesetzt fühlen. Um Kosten zu sparen wird oft das Softwareprodukt un- oder nur oberflächlich getestet ausgeliefert, weil für eine angemessene Qualitätssicherung kaum Zeit bleibt. Dieser vermeintlichen Zeit- und somit Kostenersparnis folgt dann oftmals das Gegenteil [MUE04]: Die Kosten zur Fehlerbeseitigung steigen umso höher, je später ein Fehler entdeckt wird. Es sind also präventive Maßnahmen erforderlich, um die Fehlerrate im fertigen Softwaresystem zu minimieren.

Eine Methode zur Qualitätsverbesserung von Software während der Entwicklungszeit ist Nutzung von schon gemachten Erfahrungen. Ein erfahrener Entwickler wird einen gemachten Fehler mit großer Wahrscheinlichkeit nicht wiederholen; positive Erfahrungen wird er hingegen wieder verwenden. Offensichtlich minimiert Erfahrung das Fehlerrisiko. Damit von Erfahrungen profitiert werden kann, müssen diese auch anderen Mitarbeitern angeboten werden. Insbesondere entwickelte V. Basili mit der Experience Factory ein Konzept [BCR94] zum Umgang mit Erfahrungen, das in jüngerer Zeit immer häufiger in dieser und in daran orientierten Varianten, so z.B. in [SS01], bei der Software-Entwicklung zum Einsatz kommt.

Die sinnvolle Nutzung von Erfahrungen benötigt wesentliche vorbereitende Schritte: Erfahrungen müssen in einer geeigneten Form angeboten werden, um ihre erfolgreiche Wiederverwendung zu ermöglichen. Hier knüpft das Ziel dieser Arbeit an: *Konzept und Realisierung eines Werkzeugs zur Erhebung, Verwaltung und Bewertung von Erfahrungen.*

1.2 Zielsetzung

Ziel dieser Arbeit ist es, anhand bekannter Konzepte zur Erfahrungsnutzung sowie einem geeigneten Bewertungssystem ein Werkzeug zu entwickeln, das durch Erhebung, Verwaltung und Bewertung von software-technischen Erfahrungen ein Hilfsmittel bei der Software-Entwicklung darstellt.

1.3 Gliederung

Die nachfolgenden Kapitel dieser Arbeit gliedern sich wie folgt:

- Kapitel 2 beschäftigt sich einleitend mit dem Begriff „Erfahrung“ und seiner Bedeutung für die Software-Technik. Anschließend stellt es das bekannte Konzept der Experience Factory vor.
- In Kapitel 3 wird anhand bekannter Beispiele eine Möglichkeit zur Bewertung von Erfahrungen ausgearbeitet.

- Kapitel 4 stellt die Realisierung der gewünschten Erfahrungsdatenbank vor. Es wird der gesamte Entwicklungsprozess mit seinen Entwicklungsphasen beschrieben. Anschließend werden ein mögliches Anwendungsszenario vorgestellt und Möglichkeiten der Erweiterung der Erfahrungsdatenbank diskutiert.
- Kapitel 5 stellt die im vorherigen Kapitel realisierte Erfahrungsdatenbank im Zusammenhang mit zwei software-technischen Konzepten vor.
- Abschließend finden sich in Kapitel 6 eine Zusammenfassung und ein Ausblick.

2 Nutzung von Erfahrungen in der Software-Technik

2.1 Warum Erfahrungen?

Die Qualitätsverbesserung von Software ist eine u.a. auf Erfahrungen beruhende Tätigkeit. Oft werden dafür erfahrene Qualitätsexperten angestellt [SCH01]. Das zeigt, dass Erfahrungen eine nicht zu unterschätzende Wissensquelle sind. Viele Probleme der Software-Technik bieten ad hoc keine adäquate Lösung; sie sind entweder nicht berechenbar oder auch nur schwer abzuschätzen. So bleibt am Scheidungsweg eines solchen Problems oft nur die Quelle der Erfahrungen als einziges Kriterium zu seiner Bewältigung.

Die Nutzung bzw. Wiederverwendung von in vergangenen Projekten gewonnenen Erfahrungen hilft, zukünftige Projekterfolge zu wiederholen und begangene Fehler zukünftig zu vermeiden. Es kann sogar Zeit eingespart werden, wenn durch Erfahrungen eine spätere Problemanalyse oder Wartung vermieden werden kann. Die Wiederverwendung bezieht sich auf alle möglichen Formen von Erfahrungen, die in Verbindung mit verschiedenen Aufgaben der Software-Technik stehen. In [SCH01] wird die Notwendigkeit für die Erfahrungsnutzung zusammenfassend wie folgt konkretisiert:

Erfahrungsunterstützung wird benötigt, wenn

- keine konkreten Erfolgskriterien vorliegen oder
- technisches Expertenwissen mit menschlichem Beurteilungsvermögen und Kompromissfähigkeiten kombiniert werden muss.

Erfahrungen bieten also ein Rezept zur Bewältigung beider oben genannter Situationen.

Nach dem „warum“ bleibt nun zu klären, „was“ Erfahrungen eigentlich sind. In der Literatur existieren viele unterschiedliche Definitionen zum Begriff „Erfahrung“. Man betrachte die zwei folgenden zufällig gewählten Definitionen:

„Die Gesamtheit der gelernten Kenntnisse, Verhaltensweisen und Fertigkeiten.“

aus Microsoft Encarta Professional 2002 Enzyklopädie

„Ein Erlebnis, aus dem man lernt; [...] das durch eigenes Erleben erworbene Wissen“

aus Wahrig Deutsches Wörterbuch

Diese beiden Definitionen sind zwar nicht einheitlich, weisen jedoch – wie auch alle übrigen – eine wesentliche Gemeinsamkeit auf: Erfahrungen basieren auf durchgeführten Tätigkeiten, also auf Praktischem. Folglich sind Erfahrungen keine Sammlungen von theoretischem Lehrbuchwissen, sondern aus der unmittelbaren Praxis gewonnenes Wissen. Im Zusammenhang mit der Software-Technik beziehen sich Erfahrungen beispielsweise auf die Anwendung von Analyse- und Entwurfsmetho-

den, den Umgang mit Entwicklungsumgebungen oder die Durchführung von Testprozessen.

Da der Transfer von Erfahrungen im Allgemeinen einen Kommunikationsprozess darstellt, lässt sich die Erfahrungsweitergabe in eine schriftliche und eine mündliche Form unterscheiden. Die schriftliche Form kann beispielsweise mittels Datenbanken und Intranet geschehen, die mündliche findet statt bei Meetings, Video-Konferenzen usw. Beide Formen haben ihre Vor- und Nachteile. Sicherlich ist das Ausfragen von erfahrenen Mitarbeitern einfach und schnell an das jeweilige Problem anpassbar, während die zweite Form kostspielig und komplizierter ist. Jedoch verbleiben schriftlich festgehaltene Erfahrungen auch beim Ausscheiden eines Experten im Unternehmen. Wie die schriftliche Erfahrungsweitergabe funktioniert und welche Vorgänge bis zur Wiederverwendung einer Erfahrung ablaufen, klärt der folgende Abschnitt.

2.2 Erfahrungskreislauf

Der Umgang mit Erfahrungen in der Software-Technik spiegelt einen Kreislauf wieder [SCH00], den Erfahrungen durchlaufen. Der Erfahrungskreislauf besteht aus vier wesentlichen Aktivitäten. Die Abbildung 2-1 zeigt den nach [SCH00] sog. Experience Activity Cycle.

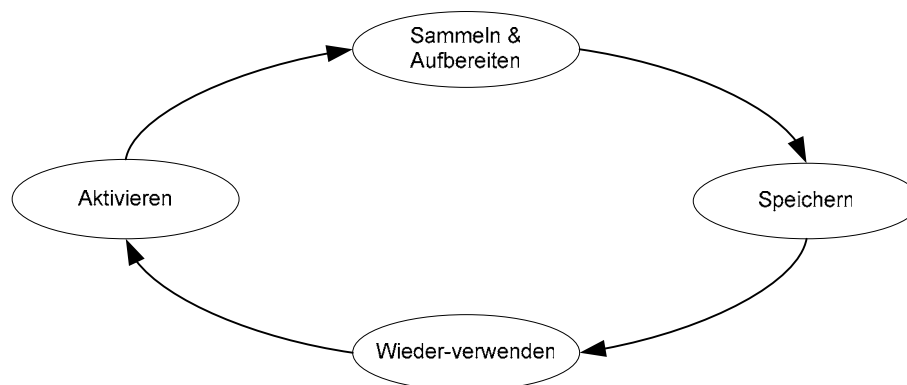


Abbildung 2-1: Erfahrungskreislauf, deutsche Übersetzung

Die Pfeile deuten die Richtung und die Abhängigkeiten zwischen den Aktivitäten an.

Aktivieren deutet an, dass der Vorgang der Erhebung von Erfahrungen ein aktiver ist. Die Erfahrung muss in einem aktiven Prozess von ihrem Erfahrungsträger, das kann sein eine Person, eine Situation oder ein Dokument, gelöst werden, um später unabhängig genutzt werden zu können. Die Lösung der Erfahrung insbesondere von einer Person muss durch eine Aufforderung geschehen, z.B. durch Fragebögen, in denen diejenige Person regelrecht ausgefragt wird.

Sammeln heißt, die Erfahrungen zu identifizieren und *Aufbereiten* bedeutet, die Erfahrungen entsprechend zu verallgemeinern, anzupassen und zu formalisieren, um sie so überhaupt erst wieder verwendbar zu machen. Der Schritt der Aufbereitung wird als Experience Engineering bezeichnet.

Speichern bezieht sich auf die technische, persistente Ablegung der Erfahrungen z.B. im Internet, in Datenbanken oder in den dafür vorgesehenen Programmen. Dadurch werden Erfahrungen in einer Form angeboten, die es erleichtert, sie wieder zu verwenden.

Mit *Wiederverwenden* ist der Vorgang der (erneuten) Nutzung der Erfahrungen gemeint. Das Wort Wiederverwenden ist im Erfahrungskreislauf durch einen Bindestrich getrennt, um den Kreislauf nochmals zu betonen: Der Kreislauf einer Erfahrung endet an dieser Stelle nicht, sondern wird weiter fortgesetzt. Dieses stellt sicher, dass fehlerhafte Erfahrungen durch nochmalige Aufbereitung verbessert und erneut zur Wiederverwendung freigegeben werden. So, wie der Kreislauf keinen Endpunkt hat, besitzt er natürlich auch keinen vorbestimmten Startpunkt.

Die vier wesentlichen Aktivitäten – im Folgenden Erfahrungserfassung, -aufbereitung, -speicherung und -nutzung genannt – werden in den nächsten Abschnitten näher betrachtet.

2.2.1 Erfahrungserfassung

Erfahrungen stellen meistens Wissen von einzelnen Personen dar und sind der Allgemeinheit nicht frei zugänglich. Folglich erfolgt ihre Nutzung auch nur ausschließlich durch die betroffene Person selbst. Um Erfahrungen auch anderen verfügbar zu machen, ist zu aller erst die Erfassung dieser nötig. Damit ist die Überführung der Erfahrung in eine schriftliche Form gemeint. Jedoch funktioniert die Erfassung nicht von selbst, sondern muss von einem aktiven Vorgang begleitet werden. D.h. dass der Erfahrungsträger erst dazu animiert werden muss, sein in der Praxis erworbenes Wissen preiszugeben.

In [SCH01] werden die folgenden beiden Methoden zur schriftlichen Erfassung von Erfahrungen vorgestellt.

a) Observation Packages

Eine Methode, Erfahrungen in eine schriftliche Form zu überführen, ist durch die Verwendung von Beobachtungsnotizen, den sog. Observation Packages, gegeben [SCH01]. Das sind in der Regel einseitige Formulare, die einem Erfahrungsbesitzer die Erfahrung „herauslocken“. So ein Formular beinhaltet neben dem Namen des Beobachters und des Datums folgende Fragen (nach K. Schneider):

- Kontext: Wobei wurde die Beobachtung gemacht?
- Beobachtung: Was ist aufgefallen? Welche Erfahrung hat man gemacht?
- Emotion und Bedeutung: Wieso und in wiefern ist dies bemerkenswert?
- Hypothese / Schlussfolgerung: Wie kam es dazu und was folgt daraus?
- Lösungsidee, Bemerkung, Sonstiges

Diese Beobachtungsnotiz filtert nicht nur die reine Tätigkeit, die die Erfahrung beschreibt, sondern stellt sie auch im größeren Zusammenhang dar. Der Beobachter wird animiert, den Wert und Nutzen dieser Erfahrung zu beschreiben. Dies ist bei einer späteren Wiederverwendung oder bei der Speicherung in eine Datenbank wichtig.

b) Light-Weight Documents

Eine weitere Methode, Erfahrungen in schriftlicher Form zu Erfassen, wird durch die sog. Light-Weight Documents (LIDs) beschrieben. LIDs sind optimiert für die Aufnahme von Erfahrungen einer Mitarbeitergruppe, die zusammen für idealerweise drei Monate an einer gewöhnlichen Projektphase gearbeitet haben. Das endgültige LID, das eine Erfahrung trägt, besitzt ein vordefiniertes Inhaltsverzeichnis, das die Gruppe bei der Erfahrungsabgabe „führen“ soll.

2.2.2 Erfahrungsaufbereitung

Eine erfasste Erfahrung liegt zunächst in „roher“ Form vor. Um sie wieder verwenden zu können muss ihr Wertigkeit hinzugefügt werden. Dieses ist die Aufgabe des Experience Engineering.

Vor ihrer Speicherung sollten die Erfahrungen analysiert werden. Man kann sie miteinander kombinieren und bewerten [SCH01]. Im großen und ganzen ist aus der rohen Erfahrung, die in einem speziellen Umfeld gewonnen wird, dasjenige herauszufiltern, was auch in ähnlichen Zusammenhängen, aber nicht zwingenderweise den gleichen, genutzt werden kann. Mit anderen Worten: Es gilt, die abstrakten Ideen aus dem Konkreten zu filtern.

2.2.3 Erfahrungsaufbewahrung

Die Erfahrungsaufbewahrung spielt in der erfahrungsbasierten Qualitätsverbesserung von Software eine entscheidende Rolle. Sehr häufig werden Erfahrungsspeicher als Erfahrungsdatenbanken (engl. experience databases) bezeichnet, was zwar richtig ist, aber oft missverstanden wird. Denn Erfahrungen werden nicht bloß in reinen relationalen Datenbanken gespeichert, sondern können auf verschiedene andere Weisen strukturiert werden [SH01]. Man muss sich bei der Benutzung des Begriffs Erfahrungsdatenbank im Klaren darüber sein, dass es sich dabei um ein Software-System handelt. Der Aufbau des Systems Erfahrungsdatenbank kann auf unterschiedliche Weise geschehen. In [SH01] werden fünf Qualitätsaspekte vorgestellt, mit denen sich der Entwickler einer Erfahrungsdatenbank befasst. Eine Auseinandersetzung mit den Aspekten Benutzerführung, Bedienbarkeit, Prozesskonformität, Rückmeldungsfunktionen und Wartbarkeit ist nötig, bevor der Entwurf einer Erfahrungsdatenbank ins technische Detail geht.

Die Qualitätsaspekte werden in Kapitel 4 bei der Entwicklung eines solchen Erfahrungsdatenbanksystems berücksichtigt und in Kapitel 5 näher beurteilt.

2.2.4. Erfahrungsnutzung

Die erfolgreiche Erfahrungsnutzung ist das Ziel der gesamten Arbeit mit Erfahrungen. Dabei fällt einer Person die Wiederverwendung von eigenen Erfahrungen leichter als die von Fremden und aus Datenbanken stammenden Erfahrungen. Das, was Biologen als Trial-And-Error- bzw. Versuch-und-Irrtum-Methode [ENC02] bezeichnen, findet auch in einer Person statt: Was sich schon einmal als richtig erwiesen hat, wird wiederholt; alles andere wird zumindest skeptisch hinterfragt. Diese Skepsis ist wichtig für den Erfahrungskreislauf, denn nur so werden Erfahrungen „verbessert“ und optimiert in einen weiteren Durchlauf im Erfahrungskreislauf geschickt.

2.3 Konzept der Experience Factory

Der wohl bekannteste Ansatz zur systematischen Nutzung von Erfahrungen in der Software-Technik ist das Konzept der Experience Factory, entwickelt u.a. von V. Basili [BCR94]. Als Teil des Quality Improvement Paradigm, einem Ansatz zur kontinuierlichen Verbesserung der Softwarequalität, ist das Konzept der Experience Factory eine viel versprechende Möglichkeit, durch Erfahrungsnutzung qualitativ hochwertige Software zu geringeren Kosten innerhalb von akzeptablen Zeitspannen zu entwickeln. Dabei handelt es sich um eine logische und/oder physische Organisationseinheit, die von der Projektorganisation der Software-Entwicklung getrennt ist. Zumindest die logische Trennung zwischen Projektorganisation und Experience Factory ist durch den Konflikt zwischen den lokalen projektbezogenen Zielen und dem globalen Ziel der Wiederverwendung begründet. Denn die lokalen projektbezogenen Ziele benötigen konkrete und auf das spezielle Ziel hin ausgerichtete Informationen, während das Ziel der Wiederverwendung ein globales ist und somit eine abstraktere Form von Erfahrungsinformationen erfordert (vgl. Abschnitt 2.2.2). Während die Projektorganisation die Software-Entwicklung zum Ziel hat, spielen in der Experience Factory die Projektunterstützung, die kontinuierliche Akkumulation von aufbereiteten Erfahrungen in die Erfahrungsdatenbank und die Analyse die wesentlichen Rollen. Die Projektunterstützung vereinfacht die Interaktion zwischen Entwickler und Analytiker, indem die ausgetauschten Informationen gepflegt und gespeichert werden; in der Analyse werden die Informationen in eine Form gebracht, in der sie leicht an bestimmte Projektanforderungen angepasst werden können.

Die Experience Factory stellt eine schriftliche Form des Erfahrungstransfers dar. Aus durchgeführten Projekten wird die Experience Factory mit Informationen versorgt, d.h. Erfahrungen werden aufbereitet und schließlich in der Erfahrungsdatenbank, dem zentralen Bestandteil der Experience Factory, in Form von gruppierten Erfahrungspaketen gespeichert. Diese bilden das Hauptprodukt einer Experience Factory. Die Struktur und der Inhalt eines solchen Pakets ist stark davon abhängig, welche Art von Erfahrung es enthält [BCR94].

An dieser Stelle sei nochmals an den Erfahrungskreislauf aus dem Abschnitt 2.2 erinnert, der hier mit seinen vier Stationen Aktivieren, Aufbereiten, Speichern und Wiederverwenden wieder zu finden ist.

In der Literatur existieren mittlerweile verschiedene Varianten dieses Konzeptes. Die Abbildung 2-2 zeigt das Konzept nach [BCR94] grafisch.

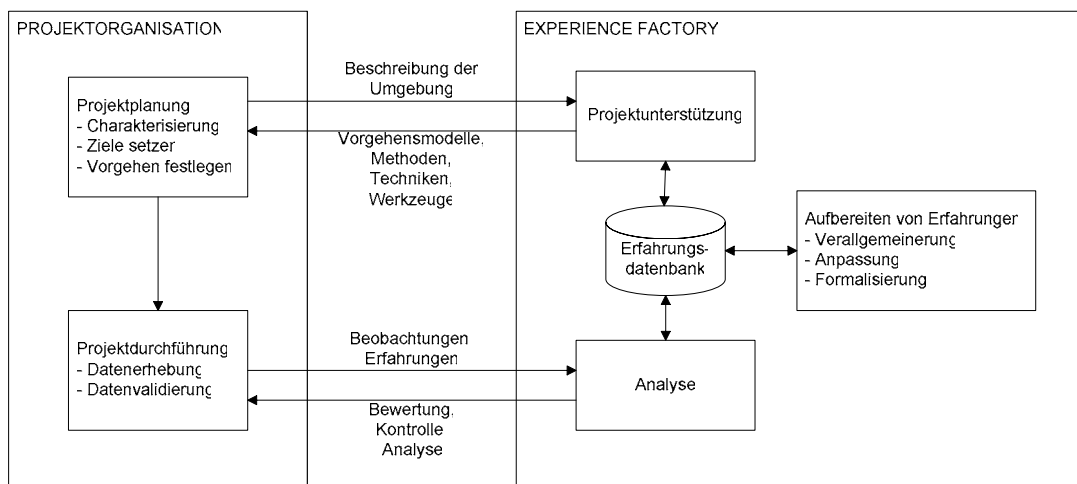


Abbildung 2-2: Konzept der Experience Factory [BCR94], deutsche Übersetzung

Viele der Berichte zum Thema Experience Factory sind junge universitäre Veröffentlichungen. Diese Tatsache belegt die Aktualität der Thematik um die Qualitätssteigerung durch Erfahrungsnutzung in der Software-Technik. Die vorhandenen Berichte aus der industriellen Praxis sind stets positiv geprägt [HS], verdeutlichen aber auch den großen zeitlichen Aufwand beim Aufbau einer eigenen Experience Factory mit einem genau zu kalkulierenden Kosten-Nutzen-Verhältnis.

3 Bewertung von Erfahrungen

Eine Besonderheit der in Kapitel 4 zu realisierenden Erfahrungsdatenbank ist es, Erfahrungen bewerten zu können. Die Bewertung von Erfahrungen erfüllt dabei die Funktion, die Erfahrungen in einer gewissen Weise zu sortieren. Dadurch wird es dem Nutzer vereinfacht, hilfreiche von nicht hilfreichen Erfahrungen zu unterscheiden.

Bei Überlegungen, wie eine Bewertungsfunktion zu realisieren ist, stellen sich die folgenden fundamentalen Fragen:

- Was wird bewertet?
- Nach welcher Bewertungsskala wird bewertet?
- Wird eine Begründung der Bewertung verlangt?
- Wie werden Bewertungen anderen Nutzern sichtbar gemacht?

Um Antworten auf diese vier Fragen zu finden, ist es hilfreich zu untersuchen, welche Lösungen woanders benutzt werden. Im Internet findet man zahlreiche Beispiele, wo Bewertungen eine Rolle spielen. Die beiden bekanntesten von allen sind wohl die das Internet-Auktionshaus eBay (www.ebay.de) und der Internet-Shop Amazon (www.amazon.de). Anhand der folgenden Analyse dieser beiden Beispiele wird ein Bewertungssystem für eine Erfahrungsdatenbank erarbeitet. Dabei soll im Folgenden ein Bewertungssystem ein System sein, das Antworten auf die obigen Fragen liefert.

Beim Vergleich der beiden Lösungen von eBay und Amazon stellt man fest, dass zwei unterschiedliche Bewertungssysteme benutzt werden. Während bei eBay nur zwischen positiven und negativen Bewertungen unterschieden wird, darf der Benutzer von Amazon seine Bewertung aus der Menge {1,2,3,4,5} wählen, bei der 5 die beste Bewertung darstellt. Den Grund für diese unterschiedlichen Bewertungsskalen erhält man, wenn man erforscht, was überhaupt bewertet wird. Bei eBay bewertet man kein Produkt, sondern die an der Auktion teilnehmende Person. eBay basiert vorrangig auf der Zuverlässigkeit der Auktionäre. Dafür ist in der Tat eine zweistellige Notenskala die beste Lösung: Entweder die Auktion ist erfolgreich beendet, dann wird positiv bewertet, oder einer der beiden Auktionspartner ist nicht zufrieden und bewertet den Anderen negativ. Hätte man bei eBay eine Bewertungsskala mit mehr als zwei Elementen gewählt, so würde es dem Bewertenden sicherlich schwer fallen, eine passende Entscheidung zu fällen. Schließlich hängt der Erfolg einer Auktion davon ab, ob der Käufer das Produkt bekommt (positive Bewertung) und zwar so, wie in der Produktbeschreibung beschrieben, oder nicht (negative Bewertung). Bei Amazon dagegen wird das angebotene Produkt bewertet. Hätte man nur zwei Noten, z.B. gut und schlecht, zur Auswahl, so wäre eine Bewertung zu undifferenziert. Der Kunde könnte bei einer zweistelligen Skala beispielsweise kleine Unzufriedenheiten nicht zum Ausdruck bringen, wie es ihm die fünfstellige Notenskala erlaubt.

Auch bei der Betrachtung der Begründungen für die Bewertungen erhält man zwei unterschiedliche Bilder. Zusätzlich zur Bewertung wird der Benutzer von eBay auf-

gefordert, diese kurz, d.h. mit maximal 80 Zeichen, zu begründen. Längere Bewertungen sind in der Tat nicht nötig, da man schon mit der Bewertung zum Ausdruck bringt, ob man den Händler weiterempfiehlt oder nicht. So fällt die Begründung in der Praxis oft sehr kurz und mager aus (z.B. „Alles ok.“). Bei Amazon wird der Benutzer aufgefordert einen Titel für seine Bewertung abgeben und diese dann ausführlich zu begründen. Für die Begründung stehen ihm maximal 1000 Wörter zur Verfügung, was auch sinnvoll ist, da diese für andere Interessenten eine eventuelle Kaufentscheidung darstellt. Hier wird das Produkt beschrieben, dessen Stärken und Schwächen man konkret angeben kann.

Die Darstellung der Bewertung findet bei eBay nicht graphisch statt. Es wird der Prozentsatz der positiven Bewertungen angezeigt. Bei Amazon wird die Durchschnittsnote aller Bewertungen in Form von Sternen angezeigt. Diese graphische Darstellung hat die Genauigkeit von einem halben Stern. Zusätzlich erhält man die auf eine Nachkommastelle gerundete Durchschnittsnote durch das Zeigen auf die Sterne mit der Maus.

Anhand der Bewertungssysteme von eBay und Amazon, kann man relativ leicht ein Bewertungssystem für Erfahrungen einer Erfahrungsdatenbank erstellen. Da die Erfahrung, also das Produkt der Erfahrungsdatenbank, bewertet wird, empfiehlt sich ein Bewertungssystem, das ähnlich zu dem von Amazon ist. Die Bewertungsskala wird auf die vierelementige Menge $\{1,2,3,4\}$ reduziert, wobei vier Sterne für eine sehr hilfreiche Erfahrung stehen. Der Vorteil gegenüber von fünf Noten ist, dass bei vier Noten keine in der Mitte liegt. Das zwingt auch den unentschiedenen Benutzer dazu, sich eher positiv (3) oder eher negativ (2) zu entscheiden und somit seiner Bewertung mehr Aussagekraft zu verleihen. Zusätzlich zur Bewertung soll dem Nutzer auch die Möglichkeit einer Begründung und Äußerung (Feedback) zu einer Erfahrung gegeben werden. Um bei vielen Feedbacks eine bessere Übersicht zu gewähren, soll wie bei Amazon jedes Feedback einen Titel erhalten. Das Feedback selbst soll in seiner Länge unbegrenzt sein. Die Darstellung der Bewertung erfolgt wie bei Amazon durch Sterne. Dabei stehen gelb gefüllte Sterne für die Durchschnittsnote aller Bewertungen einer Erfahrung. Die Genauigkeit der Darstellung ist auf einen halben gefüllten Stern begrenzt, eine Verfeinerung dieser Genauigkeit, würde zu optischen Missverständnissen führen. Schließlich muss man noch festlegen wie ungerade Durchschnittsbewertungen dargestellt werden, also die Grenzen für halbe Sterne festlegen. Entweder ein halber Stern beginnt bei 0,5 und endet bei 1 (entsprechend für 1,5 usw.). Das würde aber die Ergebnisse z.B. bei 0,49 zu stark verfälschen. Aus diesem Grund ist es besser, die Grenzen von 0,25 bis kleiner als 0,75 festzulegen. Die Abbildung 3-1 enthält für die ganze Bewertungsskala eine Darstellungsübersicht.



Abbildung: 3-1 Bewertung von Erfahrungen durch Sterne

Doch neben der Bewertung an sich, spielt offensichtlich auch die Anzahl der Bewertungen eine große Rolle. Denn überall wo Bewertungssysteme zu finden sind, wird auch die Anzahl der Bewertung angegeben. Eine Erfahrung die eine gute Durchschnittsbewertung bei erst wenigen Bewertungen hat, ist weniger aussagekräftig als eine mit vielen. Folglich wird auch in der Erfahrungsdatenbank nicht auf die Anzeige der Anzahl der Bewertungen verzichtet.

Zusammenfassend stellt die Tabelle 3-1 die drei vorgestellten Bewertungssysteme nochmals übersichtlich dar.



	eBay	Amazon	Erfahrungsdatenbank
Was wird bewertet?	Käufer, Verkäufer	Produkt	Erfahrung
Bewertungsskala	{positiv, negativ}	{1,2,3,4,5}	{1,2,3,4}
Bewertungsbegründung	max. 80 Zeichen	max. 1000 Wörter, mit Titel	unbegrenzt, mit Titel
Aussage der Bewertung	Zuverlässigkeit des Käufers	Kaufempfehlung	Erfahrung hilfreich?
Darstellung	in Prozent positiver Bewertungen	durch Sternchen, optische Genauigkeit: halber Stern	durch Sternchen, optische Genauigkeit: halber Stern
Beispiel	Positive Bewertungen: 100%		

Tabelle 3-1: Bewertungssysteme im Überblick

4 Realisierung einer Erfahrungsdatenbank

4.1 Vorüberlegungen und Konzept

Die zu realisierende Erfahrungsdatenbank – im folgenden Experience Base genannt – soll ein Aufbewahrungsort für software-technische Erfahrungen sein, im großen und ganzen also ein Werkzeug, das projektunterstützend beim Software Engineering hilft. Über eine graphische Benutzeroberfläche sollen die Benutzer ihre Erfahrungen abgeben sowie abrufen können. Des Weiteren sollen Möglichkeiten der Bewertung der gespeicherten Erfahrungen und der Abgabe von kritischen Kommentaren, positiven wie negativen, nachfolgend Feedbacks genannt, vorhanden sein. Zudem wäre eine Benutzerunterscheidung zwischen normalen Benutzern und dem Administrator sinnvoll; der Administrator soll neben den üblichen Funktionen auch Löschfunktionen erhalten. Folglich ist eine Anmeldefunktion mit einer Name-Passwort-Kombination nötig, um die Benutzer unterscheiden zu können und jede Erfahrung und jedes Feedback einem Benutzer eindeutig zuordnen zu können. Diese zunächst sehr groben Anforderungen sollen einen ersten Überblick über das gewünschte Software-Produkt verschaffen. Eine konkrete Spezifikation wird in der Phase der Analyse (Abschnitt 4.3.1) erarbeitet, wo alle Anforderungen genau dokumentiert werden.

Bei dem Projekt der Experience Base handelt es sich um ein kleines Projekt, da nur eine Person an der Entwicklung für einen Kunden beteiligt ist [ZBGK01]. Zu einem kleinen Projekt gehören neben der Auslieferung der gewünschten Software noch weitere Leistungen und Produkte, die der Entwickler zu erstellen hat. Das sind Dokumente der Anforderungsanalyse, des Entwurfs sowie eine Entwickler- und Benutzerdokumentation. Diese Produkte werden in den folgenden Abschnitten erarbeitet und erklärt. Das Vorgehen bei der Entwicklung der Experience Base wird sich an aus der Praxis bekannte Konzepte der Software-Technik halten. Dazu gehört neben dem Einsatz eines passenden Entwicklungsmodells, auf das im nächsten Abschnitt eingegangen wird, die Nutzung der standardisierten UML-Notation (Unified Modelling Language) [UML]. Um eventuell einen späteren Austausch der Software mit außerdeutschen Interessenten zu ermöglichen, wird die Benutzerfunktionalität der Experience Base in englischer Sprache gehalten.

Aus programmieretechnischer Sicht ist die Experience Base ein objekt-orientiertes System. Durch den Einsatz der objekt-orientierten Programmiersprache Java wird einerseits Plattformunabhängigkeit gewährleistet, was später andererseits auch einen möglichen Betrieb der Experience Base als Applet im Intra- oder Internet ermöglicht. Die Umwandlung der fertigen Applikation in ein Applet wird in dieser Arbeit nicht berücksichtigt, sie ist jedoch mit keinem großen Aufwand verbunden. Als Entwicklungsumgebung dient Eclipse 3.0 [ECL] mit dem Java-SDK in der Version 1.4.2. Eclipse bietet eine große Mächtigkeit an Funktionalität, zudem existieren unzählige Plug-Ins für diese Plattform, insbesondere solche, die die Funktionalität um die Möglichkeit der Erstellung von UML-Diagrammen erweitern. Zudem ist Eclipse als sog. Open Source kostenlos verfügbar. Da über die Experience Base Informationen gespeichert werden, ist der Einsatz eines Datenbank-Management-Systems notwendig. Es wird das ebenfalls frei verfügbare und für verteilte Anwendungen weit verbreitete MySQL verwendet [SQL].

4.2 Wahl des Entwicklungsmodells

Bei diesem Software-Projekt soll nicht auf eine bedachte Wahl eines systematischen Vorgehensmodells verzichtet werden. Denn auch bei kleineren Projekten ist eine strukturierte Vorgehensweise wichtig, auch im Hinblick auf spätere Erweiterungen. Die meisten Vorgehensmodelle basieren auf den grundsätzlichen Arbeitsschritten Analyse, Entwurf und Implementierung mit anschließender Inbetriebnahme und Wartung. In [ZBGK01] werden die in der Praxis etabliertesten Vorgehensmodelle vorgestellt. Wichtig bei der Wahl eines Vorgehensmodells ist, dass das Modell an das Projekt angepasst ist, nicht umgekehrt. Denn nicht jedes noch so gute Modell ist für jedes Projekt sinnvoll. Meistens entscheidet die Größe des Projektes über die Wahl des Modells. Natürlich sind auch Kombinationen verschiedener Modelle in Betracht zu ziehen.

Bei dem Projekt der Experience Base fiel die Wahl auf das sog. inkrementelle Modell, das die Abbildung 4-1 zeigt.

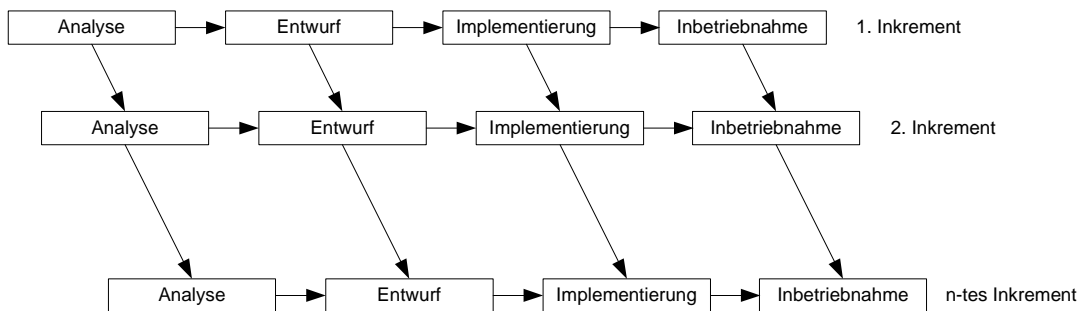


Abbildung 4-1: Inkrementelles Modell

Der Grund für die Wahl dieses Modells lag am Fehlen einer konkreten Spezifikation. D.h. dass vor und selbst während der Entwicklung der Experience Base die Anforderungen nicht komplett waren, sondern erst mit der Entwicklung gewachsen sind. Das inkrementelle Modell löst das Problem der fehlenden Anforderungen, weil das System schrittweise aufgebaut wird. Sobald neue Anforderungen hinzukommen, wird im nächsten Inkrement ein weiterer Zyklus durchlaufen. So kann auf Änderungen und Hinzukommen von Anforderungen besser und schneller reagiert werden.

Die Experience Base wurde mit insgesamt zwei Inkrementen entwickelt. Auf Grund des kurzen Projektzeitraums waren mehr als zwei Inkremente nicht zu realisieren. Um den Gesamtüberblick über den Entwurf nicht zu erschweren, wurden gleiche Phasen beider Inkremente jeweils gleichzeitig behandelt; damit eine Zuordnung trotzdem möglich ist, wird im folgenden immer wieder darauf hingewiesen, um welches Inkrement es sich handelt; in Abbildungen werden die beiden Inkremente farblich unterschieden.

4.3 Entwicklungsprozess

Dieser Abschnitt beschreibt die einzelnen Phasen des Entwicklungsprozesses der Experience Base, wie sie im inkrementellen Modell vorgestellt wurden.

4.3.1 Analyse

In der Phase der Analyse werden Anforderungen an das gewünschte Software-System ermittelt. Die Anforderungen werden nicht in einem Schritt, sondern in mehreren Iterationen dokumentiert und angepasst. Anforderungen können auf verschiedene Weisen festgehalten werden. Die UML-Notation [UML] bietet dafür sog. Anwendungsfall- bzw. Use-Case-Diagramme zur Übersicht; genau beschrieben werden die Anforderungen dann in den Anwendungsfallbeschreibungen. Das Anwendungsfalldiagramm zusammen mit der Beschreibung der Anwendungsfälle bildet das Dokument der Analyse.

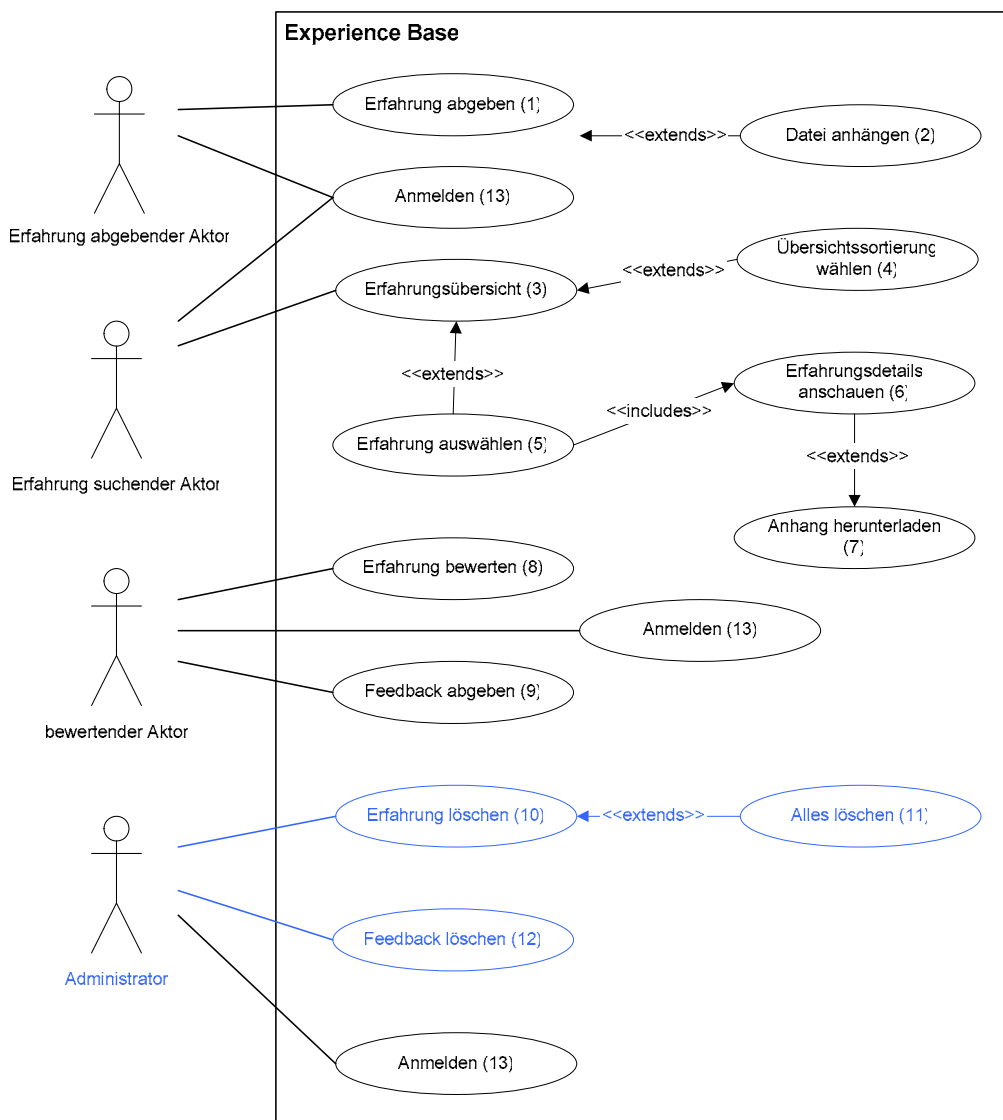


Abbildung 4-2: Anwendungsfalldiagramm

Die Abbildung 4-2 zeigt das Anwendungsfalldiagramm für die Experience Base. Diejenigen Anforderungen, die zum zweiten Inkrement gehören, das sind die nur den Administrator betreffenden Anforderungen, sind im Diagramm blau hervorgehoben.

Die Experience Base unterscheidet zwischen zwei Benutzergruppen: dem „normalen“ Benutzer und dem Administrator. Der normale Benutzer der Experience Base kann je nach Intention drei verschiedene Rollen annehmen: Er kann Erfahrungen abgeben (Erfahrung abgebender Akteur), er kann nach Erfahrungen suchen (Erfahrung suchender Akteur), und er kann Erfahrungen bewerten sowie diese durch Feedbacks kommentieren (bewertender Akteur). Dem Administrator stehen dieselben Funktionen zur Verfügung. Zusätzlich besitzt er Löschfunktionen, um Erfahrungen und Feedbacks löschen zu können.

Da durch das Anwendungsfalldiagramm alleine keine vollständige Klarheit über die Anforderungen gegeben ist, enthält die Tabelle 4-1 zu jedem Anwendungsfall eine genaue Beschreibung und den Ablauf, den dieser Anwendungsfall symbolisiert. Die Anwendungsfälle zehn bis zwölf sind die administrativen Anwendungsfälle. Die übrigen Anwendungsfälle betreffen alle Benutzer. Zur besseren Übersicht stimmen die Nummern im Anwendungsfalldiagramm überein mit denen in der folgenden Tabelle.

Nr.	1
Titel	Erfahrung abgeben
Vorbedingungen	Der Benutzer ist angemeldet.
Beschreibung	Der Benutzer gibt seine Erfahrung mit Titel und Kategorie-Zuordnung ab.
Ablauf	<p>Ereignis: Der Benutzer wählt das Fenster zur Eingabe einer neuen Erfahrung aus.</p> <p>Aktion: Das Programm wartet auf die Eingaben Titel, Kategorie und Erfahrungsbeschreibung.</p> <p>Ereignis: Der Benutzer gibt die geforderten Daten ein und beendet Eingabe.</p> <p>Aktion: Das Programm speichert die eingegebenen Daten in der Datenbank inklusive dem Benutzernamen und des Datums der Eingabe. Anschließend kehrt das Programm zum Hauptfenster zurück.</p>
Auswirkungen	Neue Erfahrung erfasst.

Nr.	2
Titel	Datei anhängen
Vorbedingungen	Der Benutzer gibt eine neue Erfahrung ein.
Beschreibung	Ereignis: Der Benutzer wählt eine Datei aus, die er der Erfahrung anhängen möchte.
Ablauf	Die Datei wird kopiert und als Kopie auf dem Server gespeichert.
Auswirkungen	Dateianhang wird erfasst.

Nr.	3
Titel	Erfahrungsübersicht
Vorbedingungen	Der Benutzer ist angemeldet.
Beschreibung	Der Benutzer sieht eine tabellarische Übersicht aller Erfahrungen mit Kategorie, Titel, Datum, Autor, Anzahl der Stimmen und Durchschnittsbewertung.
Ablauf	kein
Auswirkungen	keine

Nr.	4
Titel	Übersichtssortierung wählen
Vorbedingungen	Der Benutzer wählt eine Sortierung der Übersicht aus. Sortiert werden können die Erfahrungen nach Datum (absteigend), Anzahl der Stimmen (absteigend), Durchschnittsbewertung (absteigend) und Kategorie (alphabetisch aufsteigend).
Beschreibung	Der Benutzer wählt eine Sortierung der Übersicht aus. Sortiert werden können die Erfahrungen nach Datum (absteigend), Anzahl der Stimmen (absteigend), Durchschnittsbewertung (absteigend) und Kategorie (alphabetisch aufsteigend).
Ablauf	Ereignis: Der Benutzer wählt eine der Sortierungsmöglichkeiten. Aktion: Das Hauptfenster wird mit den nach dem ausgewählten Kriterium sortierten Erfahrungen neu geladen.
Auswirkungen	keine

Nr.	5
Titel	Erfahrung auswählen
Vorbedingungen	Der Benutzer ist angemeldet.
Beschreibung	Der Benutzer wählt eine Erfahrung aus.
Ablauf	Ereignis: Der Benutzer wählt eine Erfahrung aus der Übersicht. Aktion: Start von Anwendungsfall Nr.6
Auswirkungen	Die ausgewählte Erfahrung wird gemerkt.

Nr.	6
Titel	Erfahrungsdetails anschauen
Vorbedingungen	Der Benutzer hat eine Erfahrung ausgewählt.
Beschreibung	Der Benutzer sieht alle Details, die der ausgewählten Erfahrung gespeichert sind.
Ablauf	kein
Auswirkungen	keine

Nr.	7
Titel	Anhang herunterladen
Vorbedingungen	Der Benutzer hat eine Erfahrung ausgewählt.
Beschreibung	Der Benutzer lädt den Anhang herunter.
Ablauf	Ereignis: Der Benutzer wählt den Speicherort für den Speicherpfad. Aktion: Die Datei wird in das angegebene Verzeichnis kopiert.
Auswirkungen	Der Anhang ist auf dem durch den Benutzer angegebenen Speicherpfad verfügbar.

Nr.	8
Titel	Erfahrung bewerten
Vorbedingungen	Der Benutzer hat eine Erfahrung ausgewählt.
Beschreibung	Für die Erfahrung kann der Benutzer die Noten 4 Sterne (sehr hilfreich) bis 1 Stern (nicht hilfreich) vergeben - vorausgesetzt, er hat dies noch nicht getan.
Ablauf	Ereignis: Der Benutzer wählt die Anzahl der Sterne aus, mit der er die Erfahrung benoten möchte. Aktion: Die Anzahl der Stimmen wird um eins inkrementiert und eine neue Durchschnittsnote berechnet. Daraufhin wird das Fenster mit den aktualisierten Erfahrungsdetails neu geladen und angezeigt.
Auswirkungen	Der Benutzer wird für die nochmalige Bewertung dieser Erfahrung gesperrt.

Nr.	9
Titel	Feedback abgeben
Vorbedingungen	Der Benutzer hat eine Erfahrung ausgewählt.
Beschreibung	Der Benutzer gibt ein Feedback (kritischer Kommentar) ab.
Ablauf	Ereignis: Der Benutzer wählt das Fenster zur Eingabe eines neuen Feedbacks aus. Aktion: Das Programm wartet auf die Eingaben Titel und Text des Feedbacks. Ereignis: Der Benutzer gibt die geforderten Daten ein und beendet die Eingabe. Aktion: Das Programm speichert die eingegebenen Daten in der Datenbank inklusive dem Benutzernamen und des Datums der Eingabe. Anschließend kehrt das Programm zur aktualisierten Detailansicht zurück.
Auswirkungen	Der Benutzer wird für die nochmalige Abgabe eines Feedbacks für diese Erfahrung gesperrt.

Nr.	10
Titel	Erfahrung löschen
Vorbedingungen	Der Administrator hat eine Erfahrung ausgewählt.
Beschreibung	Der Administrator löscht eine Erfahrung.
Ablauf	Ereignis: Der Benutzer wählt die zu löschende Erfahrung aus der Übersicht aus. Aktion: Die Erfahrung wird gelöscht und das Hauptfenster aktualisiert.
Auswirkungen	Erfahrung ist gelöscht.

Nr.	11
Titel	Alles löschen
Vorbedingungen	Der Benutzer ist der Administrator.
Beschreibung	Der Administrator löscht alle Inhalte der Experience Base. Die Experience Base erhält den Zustand wie nach der ersten Inbetriebnahme.
Ablauf	kein
Auswirkungen	Alle Datenbanktabellen sind leer.

Nr.	12
Titel	Feedback löschen
Vorbedingungen	Der Administrator hat eine Erfahrung ausgewählt.
Beschreibung	Der Benutzer löscht ein Feedback.
Ablauf	Ereignis: Der Benutzer wählt das zu löschende Feedback aus. Aktion: Das Feedback wird gelöscht und die Detail-Ansicht aktualisiert.
Auswirkungen	Feedback ist gelöscht.

Nr.	13
Titel	Anmelden
Vorbedingungen	Der Benutzer startet die Experience Base.
Beschreibung	Der Benutzer meldet sich an.
Ablauf	Ereignis: Der Benutzer gibt in das Anmeldefenster seinen Benutzernamen, sein Passwort, den Server und den Datenbanknamen ein. Aktion: Sind die Daten korrekt, so erscheint das Fenster mit der Erfahrungsübersicht (Anwendungsfall 3). Ansonsten erscheint eine Fehlermeldung.
Auswirkungen	Benutzer ist angemeldet.

Tabelle 4-1: Anwendungsfallbeschreibungen

4.3.2 Entwurf

Im Arbeitsschritt Entwurf werden alle Aspekte des Software-Systems so definiert, dass später eine dem Entwurf entsprechende Implementierung möglich ist. Wie in der Phase der Analyse entstehen als Ergebnis des Entwurfs auch Produkte, die eine exakte technische Beschreibung des Systems d.h. der Architektur, der Anwenderschnittstelle, der Klassen, des Datenbankmodells, der Schnittstellen und der verwendeten Algorithmen und Datenstrukturen sowie Entwurfsmuster, bieten. Diese Produkte des Entwurfs stellen für die Implementierung so etwas wie die Anleitung dar.

a) Architektur

Die Architektur eines Software-Systems beschreibt, aus welchen Elementen das System zusammengesetzt ist. Auf oberster Architekturebene unterscheidet man zwischen verteilten und Stand-Alone-Systemen. Die Experience Base ist ein verteiltes System, basierend auf einer Client-Server-Architektur. Die Wahl dieser Architektur ist dadurch begründet, dass die Applikationen der Benutzerrechner, also die Clients, auf eine gemeinsame Datenbank zugreifen müssen, in der die Erfahrungen gespeichert sind; der Zugriff auf die Datenbank erfolgt eventuell durch mehrere Benutzer gleichzeitig. Dabei steht die Bezeichnung Datenbank für jede mögliche Speicherart, in diesem Fall also sowohl für die Relationen, in denen die Erfahrungen mit allen ihren Details gespeichert werden, als auch für Dateien, die Erfahrungen angehängt werden können. Das System ist also in zwei Teile aufgespalten. Die Abbildung 4-3 zeigt den Aufbau der vorliegenden Architektur.

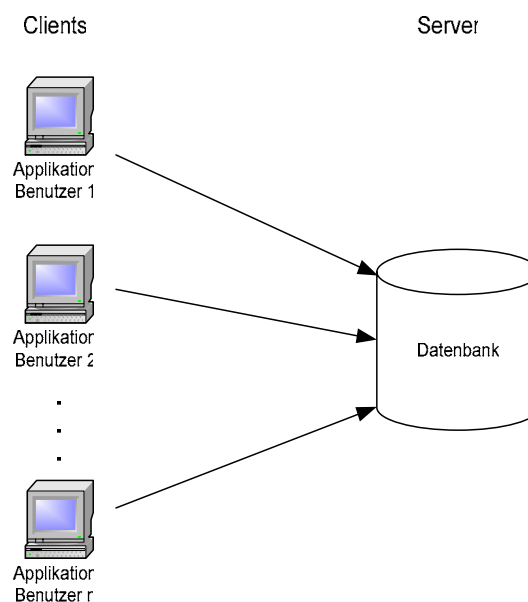


Abbildung 4-3: System-Architektur der Experience Base

b) Entwurf der Anwenderschnittstelle

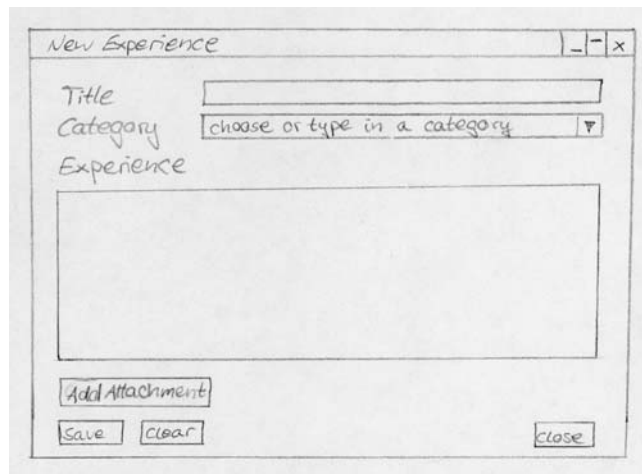
Die Anwenderschnittstelle ist ein wichtiger Teil des Systems, der mitentscheidend für den Erfolg einer Software ist. Denn über die Anwenderschnittstelle kommuniziert der Benutzer mit dem System, und sie ist der einzige Teil des Systems, das dem Benutzer zur Verfügung steht. Anhand dieser bewertet der Benutzer dann auch die gesamte Software. Beim Entwurf der Anwenderschnittstelle sollte auf eine effektive sowie intuitive Bedienbarkeit geachtet werden. Vor allem ist es wichtig bereits standardisierte Elemente zu verwenden, die der Benutzer schon von anderen ähnlichen Systemen her kennt, um ihm den Einstieg in die Nutzung der neuen Software zu erleichtern und die Benutzung so einfach wie möglich zu gestalten. Es sollte eine zum möglichst hohen Grade selbsterklärende Wirkung geschaffen werden. Diese wird in zahlreichen Fällen durch die Verwendung von Metaphern geschaffen; der Papierkorb im Betriebssystem Windows ist ein Beispiel dafür. Außerdem sollte auf eine möglichst ansprechende Gestaltung geachtet werden.

Auf alle diese Aspekte wurde auch bei der Erstellung der Anwenderschnittstelle der Experience Base geachtet. Die Anwenderschnittstelle ist eine graphische – wie man sie von Windows-Programmen her kennt – und wird auch als Graphical User Interface, kurz GUI, bezeichnet. Diese Anwenderschnittstelle wurde mit Hilfe von Swing erstellt, einer Bibliothek für die Gestaltung graphischer Anwenderschnittstellen, die fester Bestandteil des Java Development Kit ist [KRU00]. Die GUI der Experience Base besteht aus einem Hauptfenster mit einer Übersicht aller gespeicherten Erfahrungen und einigen weiteren Fenstern, die im Folgenden vorgestellt werden. Allen Fenstern ist gemeinsam, dass sie in ihrer rechten oberen Ecke die gewohnten Elemente Maximieren, Minimieren und Schließen zur Fenstersteuerung besitzen. Ferner besitzt jedes Fenster in seinem unteren Bereich eine Schaltflächenleiste. Auf dieser ist jeweils die Schließen- bzw. Beenden-Schaltfläche rechts angeordnet, andere Schaltflächen sind linksbündig in dieser Leiste angeordnet. Durch diese Gemeinsamkeiten bekommt der Benutzer in jedem Fenster der GUI eine ähnliche Bedienbarkeit, die ihm die Navigation erleichtert.

Weiterhin ist zwischen Administratorfenstern und Fenstern für normale Benutzer zu unterscheiden. Wie schon oben erwähnt wurde, besitzt der Administrator dieselben Funktionen wie jeder weitere Benutzer auch und zusätzlich nur ihm zur Verfügung stehende Funktionen. Aus diesem Grund unterscheiden sich die Fenster des Administrators nur dadurch, dass die Funktionen, die nur dem Administrator zur Verfügung stehen, in der GUI rot hervorgehoben sind. Der normale Benutzer sieht diese Funktionen nicht. Das Fenster der Klasse DeleteFeedbackGUI ist nur dem Administrator vorbehalten. Je nachdem, ob ein normaler Benutzer oder ein Administrator eingeloggt ist, wird das entsprechende Fenster geöffnet. Die Tabelle 4-2 auf den folgenden Seiten beschreibt die Klassen der GUI und zeigt jeweils eine Entwurfsskizze und ein fertiges Beispielfenster einer jeden Klasse.

Klassenname: NewExperienceGUI

Entwurfsskizze:



Funktion: Eingabe von neuen Erfahrungen.

Initialisierung: Die Category-ComboBox zeigt "Choose or type in a category.". Alle übrigen Eingabefelder sind leer.

Verwendete Objekte: -

Dynamisches Verhalten: Minimierung, Maximierung sind möglich.

Aktionen: -

Category-ComboBox: Zeigt alle bisher vorhandenen Kategorien. Diese ComboBox ist editierbar, um neue Kategorien einzugeben.

AddAttachment: Es wird ein Auswahldialog zum Auswählen einer Datei angezeigt. Die ausgewählte Datei wird im Feld Attachment mit ihrem Pfadnamen angezeigt.

Free Text: Eingabe der Erfahrung durch frei erstellten Text

Observation Package: Eingabe der Erfahrung in ein vordefiniertes Formular.

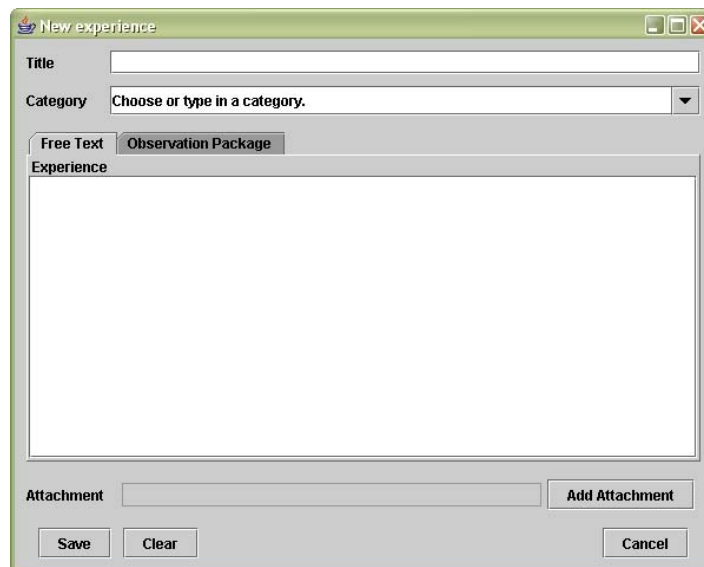
Save: Das Fenster wird geschlossen und die Eingaben als neue Erfahrung gespeichert.

Clear: Alle Eingabefelder werden geleert.

Cancel: Das Fenster wird geschlossen und alle Eingaben verworfen.

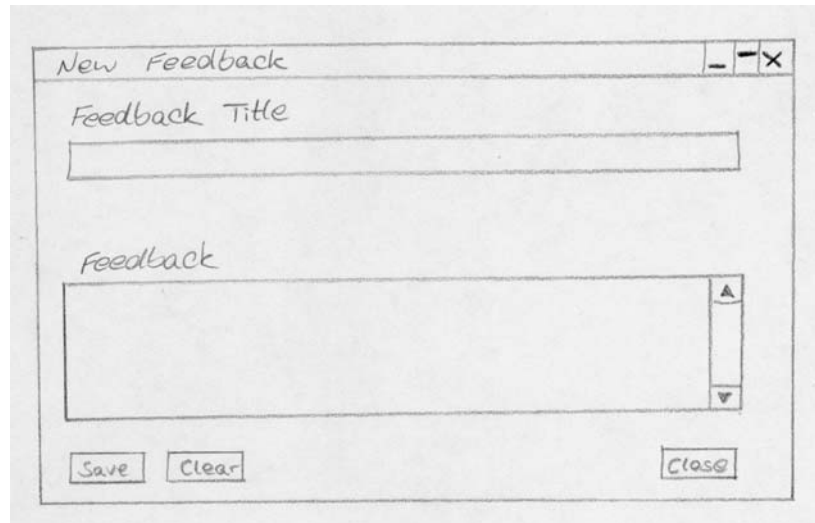
Systemmeldungen: Wird eines der Felder Title, Category oder Experience nicht ausgefüllt und versucht die Eingaben zu speichern (Save), so wird die Fehlermeldung "Not all fields have been filled out." angezeigt.

GUI-Fenster:



Klassenname: NewFeedbackGUI

Entwurfsskizze:



Funktion: Eingabe von neuen Feedbacks.

Initialisierung: Alle Eingabefelder sind leer.

Verwendete Objekte: -

Dynamisches Verhalten: Minimierung, Maximierung sind möglich.

Aktionen:

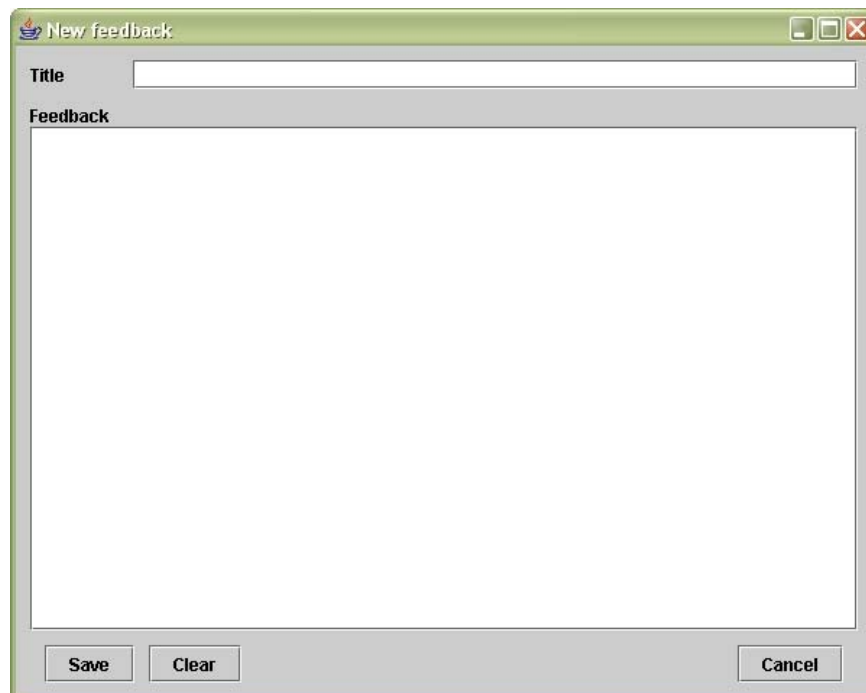
Save: Das Fenster wird geschlossen und die Eingaben als neues Feedback gespeichert.

Clear: Alle Eingabefelder werden geleert.

Cancel: Das Fenster wird geschlossen und alle Eingaben verworfen.

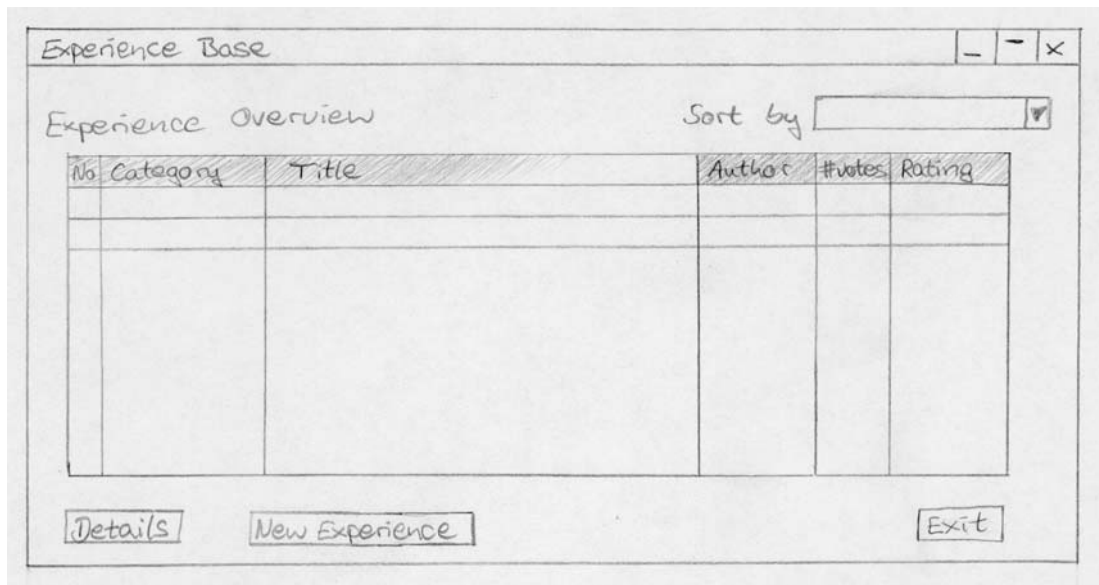
Systemmeldungen: Wird eines der Felder Title oder Feedback nicht ausgefüllt und versucht die Eingaben zu speichern (Save), so wird die Fehlermeldung "Not all fields have been filled out." angezeigt.

GUI-Fenster:



Klassenname: ExperienceBaseGUI

Entwurfsskizze:



Funktion: Startfenster nach dem Einloggen zeigt eine Erfahrungsübersicht an.

Initialisierung: Die Tabelle Experience Overview zeigt zeilenweise alle verfügbaren Erfahrungen an. Die Tabelle enthält die Spalten: Author, Date, Category, Title, average Rating und Number of Votes.

Verwendete Objekte: In der Tabelle Experience Overview werden alle vorhandenen Objekte der Klasse "Experience" zeilenweise angezeigt.

Dynamisches Verhalten: Minimierung, Maximierung sind möglich. Durch klicken auf eine Zelle der Tabelle wird die jeweilige Zeile markiert.

Aktionen:

Sort by - ComboBox: Es kann zwischen den Sortierungsmöglichkeiten nach Date, Number of Votes, Average Rating, Category gewählt werden. Bei der Auswahl wird die Tabelle Experience Overview mit der entsprechenden Sortierung neu geladen.

New Experience: Öffnet das Fenster "NewExperienceGUI".

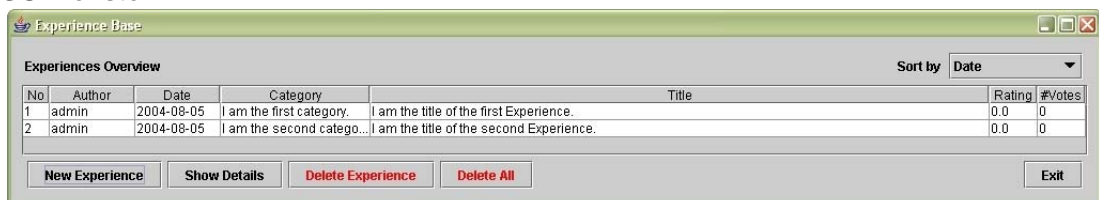
Show Details: Öffnet das Fenster "ExperienceDeatialsGUI" für die ausgewählte Erfahrung.

Exit: Schließt das Fenster und beendet das Programm.

Doppelklick auf eine Tabellenzelle: Öffnet das Fenster der Klasse "ExperienceDeatialsGUI" für die angeklickte Erfahrung.

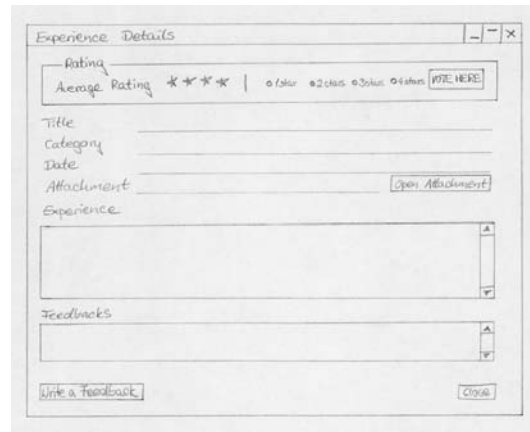
Systemmeldungen: Wird auf "Show Details" geklickt, obwohl vorher keine Erfahrung in der Tabelle markiert wurde, so wird die Meldung "No Experience has been selected." angezeigt. Wird auf "Exit" geklickt, so erhält der Benutzer den JA-NEIN-Auswahldialog "Log out and exit the program now?".

GUI-Fenster:



Klassenname: ExperienceDetailsGUI

Entwurfsskizze:



Funktion: Anzeige von allen Erfahrungsdetails inklusive Feedbacks.

Initialisierung: Zeigt Detailinformationen der im Fenster "ExperienceBaseGUI" ausgewählten Erfahrung an. Im Feld "Average Rating" wird die Durchschnittsbewertung in Sternen (1-4) angezeigt. Im Feld Voting kann der Benutzer seine eigene Bewertung (1-4 Sterne) abgeben, es ist keine Vorauswahl getroffen. Die Felder Title, Category, Datum, Attachment zeigen die entsprechenden Informationen dieser Erfahrung an. Das Experience-Feld zeigt den Text der Erfahrung. Das Feedbacks-Feld zeigt alle abgegebenen Feedback zu dieser Erfahrung an.

Verwendete Objekte: Objekte der Klassen Experience und Feedback.

Dynamisches Verhalten: Minimierung, Maximierung sind möglich.

Aktionen:

Vote: Die gewählte Anzahl an Sternen wird in die Durchschnittswertung dieser Erfahrung ausgenommen und im Feld "Average Rating" aktualisiert.

Download Attachment: Öffnet den Speichern-Dialog zum speichern des Dateianhangs. Nur aktiviert, falls eine Datei dieser Erfahrung anhängt.

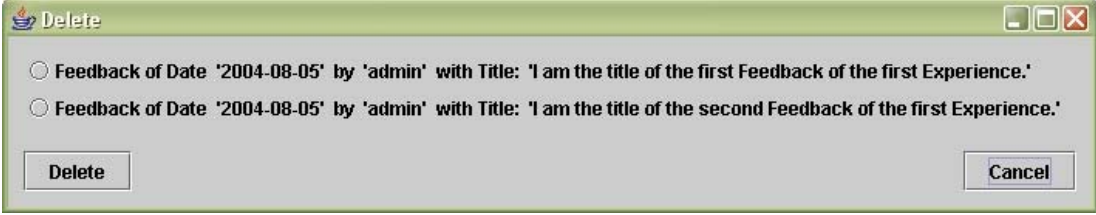
New Feedback: Öffnet das Fenster der Klasse "NewFeedbackGUI", um ein neues Feedback abzugeben.

Close: Schließt das Fenster.

Systemmeldungen: Wird auf "Vote" geklickt, obwohl vorher keine Sternenwertung ausgewählt wurde, so wird die Meldung "No stars have been selected to vote for this Experience." angezeigt.

GUI-Fenster:



Klassenname: DeleteFeedbackGUI
Fuktion: Fenster zum Löschen von Feedbacks.
Initialisierung: Fenster wird mit Datum, Autor und Titel aller vorhandenen Objekte der Klasse "Feedback", die zu dieser Erfahrung gehören, gefüllt.
Verwendete Objekte: Objekte der Klasse "Feedback".
Dynamisches Verhalten: Minimierung, Maximierung sind möglich.
Aktionen: Delete: Löscht das ausgewählte Feedback, schließt das Fenster und aktualisiert das Fenster mit den Erfahrungsdetails. Cancel: Schließt das Fenster ohne Löschung.
Systemmeldungen: Wird auf "Delete" geklickt, ohne dass vorher ein Feedback ausgewählt wurde, so wird die Meldung "No Feedback has been selected." angezeigt.
GUI-Fenster: 

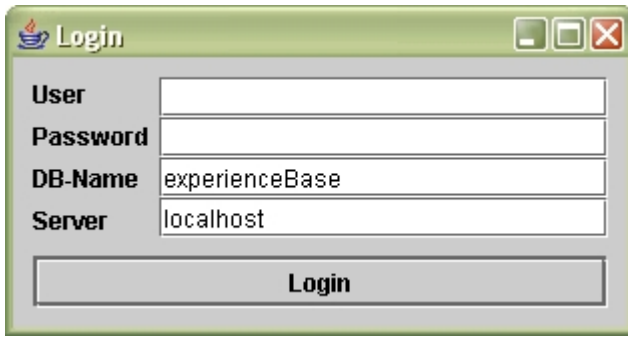
Klassenname: LoginGUI
Funktion: Benutzeranmeldung
Initialisierung: Alle Eingabefelder sind leer.
Verwendete Objekte: -
Dynamisches Verhalten: Minimierung, Maximierung sind möglich.
Aktionen: - Login: Das Fenster wird geschlossen, der Benutzer angemeldet und die Experience Base gestartet. Systemmeldungen: Sind die Benutzername (User) und / oder Benutzerpassword (Password) nicht korrekt, so wird die Fehlermeldung "Wrong User/Password combination." angezeigt.
GUI-Fenster: 

Tabelle 4-2: Klassen der graphischen Benutzeroberfläche

c) Packages

Bevor man sich dem konkreten Entwurf der Klassen widmen kann, ist es sinnvoll, Pakete (Packages) zu definieren. In einem Paket werden Klassen, die einen ähnli-

chen Zweck verfolgen, zusammengefasst. Für die Experience Base sind vier Pakete definiert. Das Paket

- Das Paket *gui* enthält Klassen, die die graphische Benutzeroberfläche (GUI), also die Anwenderschnittstelle, definieren;
- Das Paket *logic* enthält Klassen, die die logische Steuerung der Interaktionen zwischen dem Benutzer und dem System definieren;
- Das Paket *database* enthält die Klasse, die Verbindungen zur Datenbank schafft;
- Das Paket *model* enthält Klassen, die das Kernmodell der Experience Base, also die Erfahrungen mit allen Details, implementieren.

Um welche Klassen es sich genau handelt klärt der folgende Abschnitt d).

c) Entwurf der Klassen

Die Tabelle 4-3 stellt eine Klassenübersicht aller Klassen der Experience Base dar.

Paket	model
Klassen	Constants Experience Feedback
Paket	database
Klassen	DBManager
Paket	gui
Klassen	ErrorGUI ExperienceBaseGUI ExperienceDetailsGUI FeedbackDeleteGUI LoginGUI NewExperienceGUI NewFeedbackGUI YesNoGUI
Paket	logic
Klassen	ExperienceBaseLogic ExperienceDetailsLogic FeedbackDeleteLogic LoginLogic NewExperienceLogic NewFeedbackLogic

Tabelle 4-3: Klassenübersicht

Die Klasse *DBManager* ist ein sog. Singleton [GHJ97]. Von einem Singleton kann nur ein einziges Objekt erzeugt werden. Auf dieses Objekt wird eine globale Zugriffsmöglichkeit zur Verfügung gestellt, d.h. es kann nur über die Methode `getInstance()` auf die einzige Instanz zugegriffen werden. Der Konstruktor dieser Klasse ist privat, um zu verhindern, dass andere Klassen durch Anwendung des `new`-Operators eine weitere Instanz erzeugen. Die Klasse *DBManager* stellt Funktionen zur Kommunikation mit der MySQL-Datenbank zur Verfügung. Um die Datenbank nicht unnötig zu belasten, ist hier nur eine Instanz pro Benutzer sinnvoll.

Die Klasse *Experience* ist die zentrale Klasse der Experience Base. Ein Objekt dieser Klasse stellt eine Erfahrung dar. Sie wird benötigt, wenn Erfahrungen in die Datenbank eingetragen oder aus der Datenbank gelesen werden.

Objekte der Klasse *Feedback* stellen die Feedbacks für Erfahrungen dar. Der Konstruktor der Klasse *Feedback* hat u.a. ein Objekt der Klasse *Experience* als Übergabeparameter, so dass jedes *Feedback* einer Erfahrung eindeutig zugeordnet werden kann.

Die Klassen mit dem Suffix *GUI* in ihrem Namen bilden die Darstellung der graphische Anwenderschnittstelle. Die Klassen mit dem Suffix *Logic* sind für die Interaktionen zwischen dem Benutzer und der Anwenderschnittstelle zuständig.

d) Datenbankmodell

Um das Datenbankmodell zu entwerfen, muss man sich klar machen, welche Daten in der Datenbank gespeichert werden sollen. Das Meiste kann man aus dem Anwendungsfalldiagramm ableiten. Mit einigen Überlegungen kommt man auf die folgende Auflistung:

- Anwendungsfall 1 (Erfahrung abgeben):
Jede Erfahrung hat einen Autor, das Datum der Erstellung, einen Titel und den eigentlichen Text. Es ist auch sinnvoll zu jeder Erfahrung eine Kategorie anzugeben, um die Suche für die Benutzer zu beschleunigen und eine bessere Übersicht zu schaffen.
- Anwendungsfall 8 (Erfahrung bewerten):
Jeder Erfahrung muss eine Bewertung zugeordnet werden, also die Durchschnittsnote. Es muss auch gespeichert werden, welcher Autor welche Erfahrung schon bewertet hat, damit er nicht zweimal wählen darf. Außerdem kann auch die Anzahl der Bewertungen gespeichert werden, um oft frequentierte Erfahrungen hervorzuheben.
- Anwendungsfall 2 (Datei anhängen):
Jeder Erfahrung kann eine Datei angehängt werden. Zunächst auf eine Datei beschränkt.

- Anwendungsfall 9 (Feedback abgeben):
Jedes Feedback enthält einen Autor, das Datum der Erstellung, einen Titel und den eigentlichen Text.

Diese Aufzählung führt schließlich zu dem Entity-Relationship-Diagramm in Abbildung 4-4.

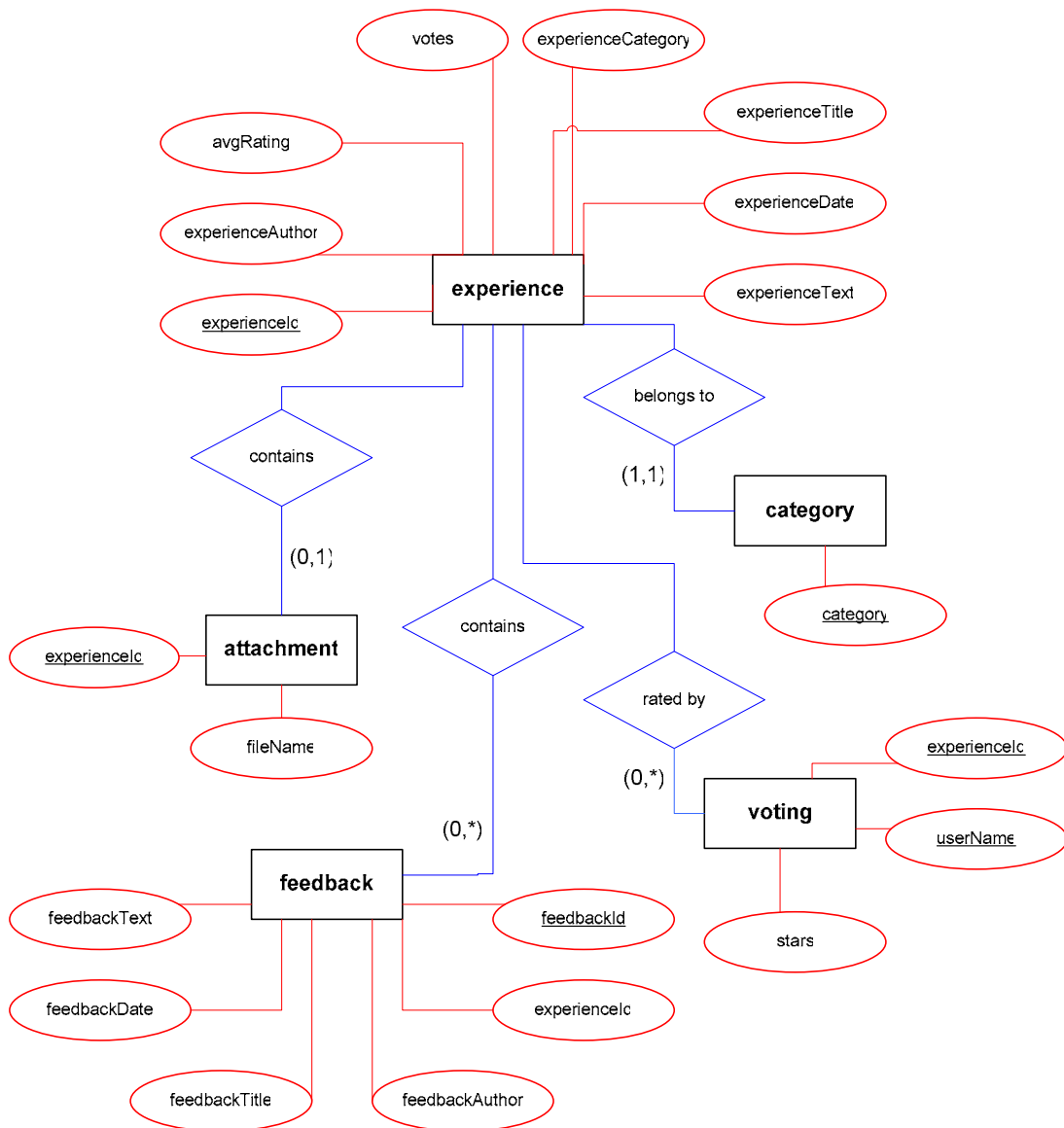


Abbildung 4-4: Entity-Relationship-Diagramm

Jede Entität wird in der Datenbank durch eine Tabelle modelliert. Die Attribute der Entitäten entsprechen den Feldern der Relationen (rot). Alle Attribute einer Entität, die eindeutig sein müssen (unterstrichen), bilden den Primärschlüssel der Tabelle. Der Primärschlüssel ist eindeutig und identifiziert somit einen einzigen Datensatz.

Die Relationen im Diagramm (blau) spiegeln zusammen mit den Kardinalitäten den Zusammenhang zwischen den Entitäten wieder. Die Kardinalitäten sind wie folgt zu interpretieren (von oben nach unten):

- Jede Erfahrung gehört genau einer Kategorie an.
- Jede Erfahrung besitzt höchstens einen Dateianhang.
- Jede Erfahrung wird durch beliebig viele Bewertungen bewertet.
- Jede Erfahrung enthält beliebig viele Feedbacks

Insgesamt lassen sich aus den oberen Überlegungen und dem Entity-Relationship-Diagramm die fünf in Tabelle 4-4 dargestellten Tabellen der Datenbank ableiten.

experience	Attribute	Datentyp	Schlüssel	Extra
	experienceId	int(11)	primary key	auto increment
	experienceAuthor	varchar(50)		
	experienceDate	date		
	experienceCategory	varchar(50)		
	experienceTitle	text		
	experienceText	text		
	votes	int(11)		
	avgRating	double		
feedback	Attribute	Datentyp	Schlüssel	Extra
	feedbackId	int(11)	primary key	auto increment
	experienceId	int(11)	foreign key	
	feedbackDate	date		
	feedbackTitle	text		
	feedbackText	text		
	feedbackAuthor	varchar(50)		
attachment	Attribute	Datentyp	Schlüssel	Extra
	experienceId	int(11)	foreign key	
	fileName	text		
voting	Attribute	Datentyp	Schlüssel	Extra
	experienceId	int(11)	primary key	
	userName	varchar(50)	primary key	
	stars	int(11)		
category	Attribute	Datentyp	Schlüssel	Extra
	category	varchar(100)	primary key	

Tabelle 4-4: Relationen der Datenbank

Bei der Erstellung der Tabellen in der Datenbank ist zu beachten, dass das Datenbanksystem MySQL – beispielsweise im Gegensatz zu Oracle – keine Fremdschlüssel unterstützt. Deshalb muss die Programmlogik die fehlende Fremdschlüsselfunktion auffangen. Das SQL-Skript, das alle benötigten Tabellen der Experience Base erzeugt, sieht folgendermaßen aus:

```

CREATE TABLE experience
    (experienceId INT NOT NULL AUTO_INCREMENT,
    experienceAuthor VARCHAR(50),
    experienceDate DATE,
    experienceCategory VARCHAR(100),
    experienceTitle TEXT,
    experienceText TEXT,
    votes INT DEFAULT '0',
    avgRating DOUBLE DEFAULT '0',
    PRIMARY KEY (experienceId)
);

CREATE TABLE feedback
    (feedbackId INT NOT NULL AUTO_INCREMENT,
    experienceId INT,
    feedbackDate DATE,
    feedbackTitle TEXT,
    feedbackText TEXT,
    feedbackAuthor VARCHAR(50),
    PRIMARY KEY (feedbackId)
);

CREATE TABLE attachment
    (experienceId INT NOT NULL,
    fileName TEXT NOT NULL,
    PRIMARY KEY (experienceId)
);

CREATE TABLE voting
    (experienceId INT NOT NULL,
    userName VARCHAR(50),
    stars INT NOT NULL,
    PRIMARY KEY (experienceId, userName)
);

CREATE TABLE category
    (category VARCHAR(100) NOT NULL,
    PRIMARY KEY (category)
);

```

4.3.3 Verteilung und Inbetriebnahme

Das Experience Base wird in einem sog. JAR-Archiv [SUN] ausgeliefert. Dieses fasst alle zur Applikation gehörenden Dateien, also .class- und Bilddateien, zusammen. Ein JAR-Archiv ist beispielsweise unter Windows per Mausclick startbar, vorausgesetzt, das Java Runtime Environment (JRE) ist installiert, das zum Ablufen von Java-Programmen benötigt wird.

Bevor die Experience Base gestartet werden kann, muss eine MySQL-Datenbank angelegt werden. Der Name der Datenbank ist frei wählbar. Außerdem müssen Benutzer angelegt werden. Der Administrator muss den Benutzernamen „admin“ besitzen. Nur dann werden die Administratorfunktionen zur Verfügung gestellt. Beim Starten des Programms erscheint zunächst das Anmeldefenster, das Abbildung 4-5 zeigt.

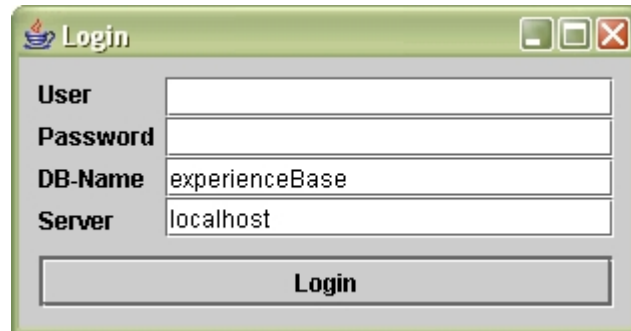


Abbildung 4-5: Anmeldefenster

- Das Feld *User* erhält den Namen des Benutzers.
- Das Feld *Password* erhält das zum Benutzernamen zugehörige Passwort.
- Das Feld *DB-Name* erhält den Name der Datenbank, z.B. *experienceBase*.
- Das Feld *Server* erhält den Namen oder die IP-Adresse des Rechners, auf dem das Datenbank-Management-System MySQL läuft. *localhost* steht für den Namen des eigenen Rechners.

Nach Eingabe aller Daten wird durch einen Klick auf die Login-Schaltfläche zuerst überprüft, ob die Benutzer-Passwort-Kombination richtig ist und ob die benötigten Relationen in der angegebenen Datenbank auf dem angegebenen Server vorhanden sind. Falls nicht werden sie neu erstellt. Daraufhin wird Experience Base mit der Erfahrungsübersicht gestartet.

Eine schriftliche Dokumentation ist für die Benutzung der Experience Base nicht erforderlich. Es wurde bei der Entwicklung der Erfahrungsdatenbank darauf geachtet, dass die Bedienung der GUI selbsterklärend ist. Die meisten Schaltflächen und Eingabefelder besitzen zudem einen sog. Tooltip-Text. Dieser wird immer dann angezeigt, wenn die Maus über dem jeweiligen Element gehalten wird. Die Tooltip-Texte besitzen zwei Funktionen: Entweder liefern sie dem Benutzer zusätzliche Informationen oder sie haben erklärende Wirkung. Ein Beispiel für einen Tooltip-Text zeigt die Abbildung 4-6; dieser erscheint, wenn die Maus über den Sternen gehalten wird.



Abbildung 4-6: Tooltip-Text

4.4 Anwendungsszenario

Im Folgenden wird ein mögliches Anwendungsszenario für die entwickelte Experience Base vorgestellt. Es bezieht sich auf das jährlich stattfindende Software-Projekt im Studiengang Informatik an der Universität Hannover. Im Rahmen dieses Projektes erstellen Studierende in kleinen Gruppen ein Softwareprodukt in einem zeitlich festgesteckten Rahmen.

Zu Ende jeder Phase des Projektes können die Studenten dazu verpflichtet werden, ihre in der jeweiligen Phase gemachten Erfahrungen in die Experience Base eintragen. Dadurch wird sichergestellt, dass die Datenbank sich allmählich füllt. Im Gegenzug bekommen sie die Möglichkeit, die Experience Base zu jeder Zeit nach Erfahrungen zu durchsuchen. Diese Möglichkeit der Hilfe wird in Situationen in Anspruch genommen, in denen wichtige Projektschritte zu erfolgen sind, die die weitere Entwicklung der Software prägend beeinflussen: Das ist der Fall beispielsweise bei der Wahl der Entwicklungsumgebung oder des Entwicklungsmodells. Zudem wird bei organisatorischen Zweifelsfällen die Experience Base hilfreich sein.

Neben der Speicherung und Nutzung ist auch die Bewertung und Kommentierung der Erfahrungen durch die Studenten wichtig. So können als hilfreich bewertete und oft frequentierte Erfahrungen hervorgehoben werden, um nachfolgenden Projektgruppen Warnungen oder besonders hilfreiche Ratschläge zu sein (vgl. Kapitel 3).

4.5 Möglichkeiten der Erweiterung

Natürlich ist das Projekt der Experience Base noch ausbaufähig. Im Folgenden werden mögliche Erweiterungen aufgelistet.

- *Flexible Auswertungswahl*
Der Erfahrung abgebende Benutzer kann eine Möglichkeit bekommen, selbst zu bestimmen, nach welchen Kriterien seine Erfahrung bewertet werden soll. Damit aber die Erfahrungsbewertungen einzelner Erfahrungen immer noch miteinander vergleichbar bleiben, muss die beste Bewertung durch vier Sterne und die schlechteste durch einen Stern dargestellt werden.

- *Administrator-Auswertung*
 Der Administrator kann nur Erfahrungen und Feedbacks löschen, um die Datenbank von falschen oder unsinnigen Beiträgen zu befreien. Zurzeit spielt er in der Experience Base keine auswertende Rolle. Zusätzlich sind folgende Auswertungsfunktionen für den Administrator denkbar.
 - Erfahrungen können durch den Administrator zusätzlich als hilfreich oder nicht hilfreich ohne Notenabgabe bewertet werden. Dadurch bekommen die Benutzer einen ersten neutralen Eindruck und es besteht eine gute Vergleichbarkeit von Erfahrungen, auch wenn die flexible Auswertungswahl benutzt wird.
 - Erfahrungen können einzelnen Projektphasen eines Software-Projektes zugeordnet werden, um eine bessere Zuordnung für die Benutzer zu gewährleisten.

- *Suchfunktion*
 Es kann eine Suchfunktion nach Suchwörtern implementiert werden. Dies ist vor allem bei zahlreichen vorhandenen Erfahrungen sehr hilfreich, damit der Benutzer gezielt suchen kann.

- *LIDs*
 Bei der Erfahrungsabgabe kann der Benutzer zwischen eigens frei erstelltem Text und dem Formular des Observation Package wählen. Hier kann eine Erweiterung um die in Abschnitt 2.2.1 vorgestellten LIDs erfolgen, so dass die Erfahrung noch gezielter von ihrem Träger selektiert werden kann.

5 Einordnung der Erfahrungsdatenbank in übergeordnete Konzepte

Damit eine Erfahrungsdatenbank ihren Zweck, Erfahrungen zu speichern und den Benutzern adäquat anzubieten, erfüllt, ist die Beachtung einiger Qualitätsaspekte wesentlich, die vor dem technischen Entwurf erörtert werden müssen. Im Folgenden werden fünf solche Qualitätsaspekte nach [SH01] vorgestellt. Gleichzeitig wird gezeigt, wie die Experience Base diese Aspekte umsetzt oder wie ihre Umsetzung erfolgen kann.

- *Benutzerführung (user guidance)*
Der Benutzer möchte Erfahrungen sofort und ohne vorherige Analyse nutzen können. Deshalb muss eine Erfahrungsdatenbank den Zweck verfolgen, Erfahrungen schon aufbereitet anzubieten. Dieses Ziel ist schwer zu erreichen, weil die Wiederverwendung eine Anpassung an die jeweilige Situation erfordert. Die Experience Base nähert sich dem Ziel durch die Verwendung der Observation Packages.
- *Bedienbarkeit (usability)*
Bei der Entwicklung einer Erfahrungsdatenbank muss darauf geachtet werden, dass der Benutzer die Bedienbarkeit so versteht, wie sie beabsichtigt wurde. Dies wurde bei der Experience Base dadurch erlangt, dass nur wesentliche Funktionen implementiert wurden. Auf Spielereien wurde bewusst verzichtet, um das Ziel der Erfahrungsnutzung nicht in den Hintergrund zu stellen.
- *Prozesskonformität (process conformance)*
Ein System sollte möglichst gut an Prozesse angepasst sein, die es unterstützt. Die Experience Base unterstützt den Prozess der Software-Entwicklung. Die Prozesskonformität ist in der Experience Base dadurch umgesetzt, dass jeder Erfahrung eine Kategorie zugeordnet ist und nach diesen Kategorien sortiert werden kann. Um eine bessere Zuordnung von Erfahrungen zu den Projektphasen eines Software-Projekts zu schaffen, kann eine Prozesszuordnung z. B. durch den Administrator erfolgen (vgl. Abschnitt 4.5).
- *Rückmeldungsfunktionen (feedback mechanism)*
Erfahrungen befinden sich in einem stetigen Entwicklungsprozess (Erfahrungskreislauf, vgl. Abschnitt 2.2). Sie bedürfen einer ständigen Aktualisierung und Verbesserung. Deshalb sind Rückmeldungen und Kommentare von anderen Personen wichtig. Die Experience Base unterstützt zurzeit eine mögliche Rückmeldungsfunktion, nämlich das Kommentieren von Erfahrungen durch Feedbacks. Erweiterungen der Rückmeldungsfunktionen um weitere Kanäle verstärkt die Erfahrungsentwicklung. Hier ist vor allem eine Erweiterung um persönliche Kontaktadressen wie E-Mail-Adressen oder Telefonnummern zu nennen, um eine direkte Kommunikation zwischen dem Erfahrungsnutzer und dem, der die Erfahrung abgegeben hat, zu ermöglichen.

- *Wartbarkeit (maintainability)*

6 Zusammenfassung und Ausblick

Ziel dieser Arbeit war es, ein Konzept für ein Werkzeug zur Erhebung, Verwaltung und Bewertung von Erfahrungen zu erstellen und dieses zu realisieren.

Hierzu war zunächst die eingehende Beschäftigung mit dem Begriff „Erfahrung“ erforderlich. So wurden in Kapitel 2 der Begriff aus software-technischer Sicht betrachtet, um die Wichtigkeit der Erfahrungsnutzung für die Entwicklung qualitativ hochwertiger Software zu betonen, und bekannte Konzepte zur Erfahrungsnutzung vorgestellt.

In Kapitel 3 wurde ein Konzept zur Bewertung von Erfahrungen entwickelt. Dabei wurde anhand von zwei anderen Beispielen deutlich, dass nicht jedes Bewertungskonzept aus der Praxis auch für die Bewertung von Erfahrungen geeignet ist.

In Kapitel 4 wurde das in Kapitel 3 entwickelte Bewertungskonzept bei der Realisierung des gewünschten Werkzeugs, der Experience Base, verwendet. Es wurde der Entwicklungsprozess der Experience Base aus software-technischer Sicht erklärt; dazu gehörten die begründete Wahl eines Entwicklungsmodells und die Vorstellung dessen einzelner Entwicklungsschritte. Anschließend wurden ein typisches Anwendungsszenario beschrieben, in dem die Experience Base zum Einsatz kommen kann, und Möglichkeiten der Erweiterung genannt.

In Kapitel 5 wurde gezeigt, welche Qualitätsaspekte eine Erfahrungsdatenbank erfüllen sollte und wie diese Qualitätsaspekte bei der Entwicklung des Konzeptes der Experience Base befolgt wurden.

Abschließend lässt sich sagen, dass mit der Entwicklung der Experience Base das Ziel dieser Arbeit erreicht wurde. Es liegt damit ein Software-Produkt vor, mit dem Erfahrungen gezielt gespeichert, verwaltet und bewertet werden können.

Bei kritischer Betrachtung lässt sich der Nutzen der Experience Base bei den Benutzern schlecht voraussagen. Um einen Erfolg mit diesem Werkzeug zu erzielen, ist eine häufige Nutzung durch viele Personen nötig, weil nur dann das Bewertungskonzept funktioniert. Deshalb ist es ratsam, die Projektmitarbeiter in jeder Projektphase eines Projektes zur Anwendung der Experience Base aufzufordern, um einerseits gemachte Erfahrungen Anderen verfügbar zu machen und andererseits schon gespeicherte Erfahrungen wieder zu verwenden.

Anknüpfend an diese Arbeit können folgende weitere Forschungsbereiche näher betrachtet werden:

- Um den Einsatz des vorgestellten Werkzeugs und somit die Erfahrungsnutzung effektiv zu gestalten, ist zu untersuchen, wie die Experience Base in die Arbeit eines Software-Entwicklers bestmöglich zu integrieren ist. Insbesondere kann eine mögliche Integrierbarkeit in Eclipse oder andere Entwicklungsumgebungen geprüft werden.

- In Kapitel 3 wurde eine Methode zur Bewertung von Erfahrungen entwickelt. Um eine flexible Bewertungswahl zu ermöglichen, sind weitere Bewertungsverfahren zu entwickeln und es ist zu untersuchen, in welchen Situationen welche Bewertungsverfahren vorteilhaft sind.
- Die Auswertung von Erfahrungen spielt für die Wiederverwendung eine wesentliche Rolle. Nach welchem möglichst automatisierten Schema können Erfahrungen ausgewertet werden, um ihre Wiederverwendung möglichst effektiv zu gestalten?

Diese drei Aufgabenstellungen können insbesondere durch die Anwendung oder Erweiterung des vorliegenden Werkzeugs gut erforscht werden.

A) Abbildungsverzeichnis

Abbildung 2-1	Erfahrungskreislauf	4
Abbildung 2-2	Konzept der Experience Factory	8
Abbildung 3-1	Bewertung von Erfahrungen durch Sterne	10
Abbildung 4-1	Inkrementelles Modell	13
Abbildung 4-2	Anwendungsfalldiagramm	14
Abbildung 4-3	System-Architektur der Experience Base	19
Abbildung 4-4	Entity-Relationship-Diagramm	28
Abbildung 4-5	Anmeldefenster	31
Abbildung 4-6	Tooltip-Text	32
Abbildung 5-1	Experience Base im Erfahrungskreislauf	35

B) Tabellenverzeichnis

Tabelle 3-1	Bewertungssysteme im Überblick	11
Tabelle 4-1	Anwendungsfallbeschreibungen	15
Tabelle 4-2	Klassen der graphischen Benutzeroberfläche	21
Tabelle 4-3	Klassenübersicht	26

C) Quellenverzeichnis

- [MUE04] V. Mücke. *Softwarequalität nicht dem Zufall überlassen*, Computerwoche Online, 13.08.2004
- [SCH01] K. Schneider. *Experience Magnets – Attracting Experiences, not Just Storing Them*, Profes-Conference, Kaiserslautern, Deutschland, September 2001
- [SCH00] K. Schneider. *LIDs: A Light-Weight Approach to Experience Elicitation and Reuse*, Profes-Conference, Oulu, Finnland, Juni 2000
- [SH01] K. Schneider, J. von Hunnius. *Effective Experience Repositories for Software Engineering*, 2001
- [BCR94] V. R. Basili, G. Caldiera, H.D. Rombach. *The Experience Factory*, 1994
- [HS] F. Houdek, K. Schneider. *Software Experience Center: The Evolution of the Experience Factory Concept*
- [ZBGK01] W. Zuser, S. Biffel, T. Grechnig, M. Köhle. *Software Engineering mit UML und dem Unified Process*, Pearson Studium, 2001
- [SQL] <http://www.mysql.org>, zuletzt besucht: 20.08.2004
- [ECL] <http://www.eclipse.org>, zuletzt besucht: 22.07.2004
- [ENC02] Microsoft Encarta Enzyklopädie 2002
- [UML] Introduction to OMG's Unified Modeling Language (UML), http://www.omg.org/gettingstarted/what_is_uml.htm, zuletzt besucht: 25.08.2004
- [KRU00] Guido Krüger. *GoTo Java 2 – Handbuch der Java-Programmierung*, Addison Wesley Verlag, 2. Auflage, 2000
- [GHJ97] Erich Gamma, Richard Helm, Ralph E. Johnson. *Design Patterns*, Addison-Wesley Professional, Juli 1997
- [SUN] <http://java.sun.com/docs/books/tutorial/jar/TOC.html>, zuletzt besucht: 23.08.2004

Erklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und ohne fremde Hilfe verfasst habe und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe.

Langenhagen, August 2004

Thomas Liro