

# Usability von Anwendungssystemen – didaktische Aspekte

Prof. Dr. Jörg Raasch

HAW Hamburg  
Department Informatik  
Berliner Tor 7  
20099 Hamburg  
raasch@informatik.haw-hamburg.de

## **Abstract:**

Zur menschengerechten Softwareentwicklung gibt es einige konstruktive Hilfen, aber bisher ist eine ergänzende, nachgehende, analytische Evaluierung entwickelter Systeme für eine Beurteilung der Gebrauchstauglichkeit unabdingbar.

Daher haben wir für die Informatik an der HAW Hamburg ein Usability-Labor eingerichtet, um in der Lehre und Forschung das Thema der Usability-Evaluation stärker ins Bewußtsein der Studierenden zu rücken. Konzeption und Nutzung dieses Labors in Lehre und Forschung ist Gegenstand dieses Beitrags.

## **1 Motivation**

Unter den Qualitätszielen, die bei jeglicher Bereitstellung von Anwendungssystemen zu beachten sind (DIN 55350), spielt die Benutzbarkeit eine wichtige Rolle, weil hier die Sichtweise des Benutzers zum Maßstab gemacht wird. Dabei ist zu unterscheiden zwischen Ergonomie (DIN EN ISO 9241-1) und Gebrauchstauglichkeit (Usability, DIN EN ISO 9241-11). Ergonomie umfasst alle Aspekte des Arbeitsplatzes und der Arbeitsplatzgestaltung und fordert, diese an die „... Eigenschaften, Fähigkeiten und Grenzen möglicher Benutzer ...“ anzupassen (Ergonomiedefinition nach DIN EN ISO 9241-1). Gebrauchstauglichkeit bezeichnet die Nützlichkeit und Verwendbarkeit der Softwarewerkzeuge im Arbeitsprozeß. Die grundsätzlichen Faktoren der Gebrauchstauglichkeit der DIN EN ISO 9241-11 sind:

- *Effektivität* (Wird ein erforderliches Arbeitsziel mit dem Softwareprodukt überhaupt erreicht?)
- *Effizienz* (Wie aufwändig wird das erforderliche Arbeitsziel mit dem Softwareprodukt erreicht?)
- *Zufriedenstellung* (Wie gross ist die subjektive Beeinträchtigung aus Sicht der Benutzer hierbei?)

Die Gebrauchstauglichkeit erstreckt sich auf die funktionale Ebene, also das Leistungsangebot des Softwaresystems und dann auch auf die Gestaltungsebene, also die Leichtigkeit, mit der das System zu nutzen ist. Dabei gilt stets der Grundsatz: „form follows function“. Die Gestaltungsebene muss die Nutzung der implementierten Funktionalität erlauben und sie darf sich nur auf diese beziehen. Diese Betrachtungsweise wird durch Architekturnotwendigkeiten ergänzt: die funktionale Ebene ist zu implementieren als Funktionskomponente, die Gestaltungsebene als Interaktionskomponente. Die Trennung von Interaktion und Funktionalität ist ein gut etablierter Grundsatz der Softwarearchitektur [Züllighoven 2004] und bereits in den grundlegenden Entwurfsmustern Model-View-Controller [Krasner 1988] bzw. Beobachter [Gamma 1994] zum Architekturprinzip erhoben worden. Insofern gibt es eine Entsprechung zwischen Gebrauchstauglichkeit und Architektur, die wir näher untersuchen und die unsere Aufmerksamkeit in der Informatik verlangt. Eine Konsequenz ist auch, dass sich die Untersuchung der Gebrauchstauglichkeit nicht auf Oberflächenelemente beschränken darf, sondern stets auf das in Benutzung befindliche Gesamtsystem und seine Architektur ausgerichtet ist.

Gebrauchstauglichkeit adressiert damit ein Qualifikationsmerkmal, das innerhalb der Informatikausbildung eine beherrschende Rolle einnehmen sollte, denn die Zufriedenheit des Benutzers mit den von Informatikern bereitgestellten Werkzeugen ist entscheidendes Abnahme- und Nutzungskriterium.

Der Benutzer arbeitet bislang in einem Interaktionsmodell, das von einer Dialogschleife [Norman 1988] ausgeht: Er artikuliert seine Bearbeitungswünsche als Eingaben an das Anwendungssystem, erhält als Reaktion von diesem eine Antwort, die seine nächste Bearbeitungsvorgabe prägen kann. Die Benutzung von Softwarewerkzeugen erfolgt nicht zum Selbstzweck, sondern aufgrund einer Aufgabenstellung, die nicht primär die Rechnernutzung zum Ziel hat, sondern ein Problem des Benutzers lösen soll. Die Aufgabe muss in ein Format übersetzt werden, das für das Softwarewerkzeug mit seinen Eingaberegeln verständlich ist (s. Abb. 1).

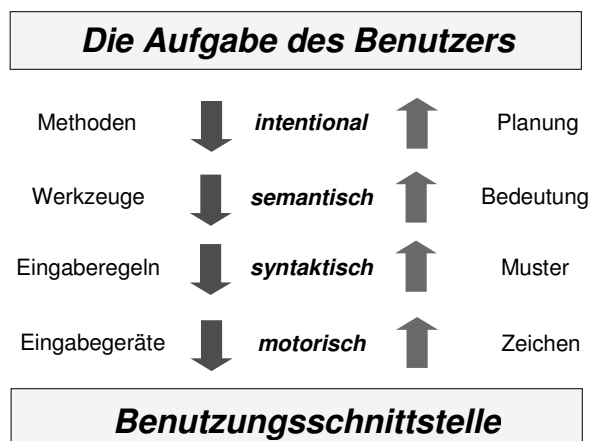


Abbildung 1: Übersetzung, Indirektheit, Distanzen (vgl. [Herczeg 1994])

Der Benutzer muss die für seine Aufgabe geeignete und seinen Intentionen entsprechende Methode und zu dieser das passende Werkzeug herausuchen. Seine Aufgabe muss er schließlich zu Interaktionen mit den Elementen der Anwendung konkretisieren. Dabei muss er auch syntaktische Regeln befolgen, geeignete Zeichen benutzen und sie über die bekannten Endgeräte eingeben. Jeder Schritt auf die nächste Ebene ist mit aufwändigen Übersetzungsvorgängen verbunden. Ausgaben des Computers müssen vom Benutzer wahrgenommen, interpretiert und schließlich bewertet werden. Auch sind Übersetzungsprozesse nur für die Rechnernutzung notwendig, die nichts mit der Aufgabe des Anwenders zu tun haben.

Insgesamt entstehen auf intentionaler, semantischer, syntaktischer und motorischer Ebene Übersetzungsvorgänge [Herczeg 1994], die zusammen eine Ablenkung des Benutzers von seiner eigentlichen Aufgabe bedeuten. Die Folge ist eine Indirektheit der Bearbeitung, weil die fachlichen Aufgaben auf einer anderen, nicht fachlich motivierten und damit im Anwendungskontext nicht unmittelbar verständlichen, technischen Ebene bearbeitet werden. Die Folge sind Distanzen, der Benutzer entfernt sich mehr und mehr von seiner eigentlichen Aufgabe, weil er den Umgang mit dem Rechner und seinen Anwendungen zu seinem Arbeitsinhalt machen muss.

Für die Anwendungsentwicklung und damit auch für die Didaktik der Informatik erhebt sich die Frage, wie Anwendungssysteme mit guter Gebrauchstauglichkeit bereitgestellt werden können.

## **2 Konstruktion und Evaluation, Benutzungsmodelle**

In der Softwareentwicklung unterscheidet man konstruktive und analytische Maßnahmen zur Qualitätssicherung. Die konstruktiven Maßnahmen sind besonders wichtig, weil sie die Vorgehensweise des Entwicklers anleiten und unterstützen und damit die Entstehung von Qualität bereits im Konstruktionsprozess fördern.

Im Hinblick auf das Qualitätsziel der Gebrauchstauglichkeit gibt es einige konstruktive Maßnahmen, die auch unter dem Begriff „Benutzerzentrierte Anwendungsentwicklung“ (user centered design [Norman 1988]) angesprochen werden. Maßgeblich ist, dass der Benutzer durch einen Autor-Kritiker-Zyklus in einen möglichst permanenten Abstimmungsprozess eingebunden wird. Unterstützt wird dies durch eine Sichtweise der Anwendungsentwicklung, die von der Anwendungsorientierung, der Kundenorientierung und damit von den Arbeitszusammenhängen des Benutzers ausgeht und seinen Arbeitsplatz und seine Arbeitsprozesse zum hauptsächlichen Untersuchungsgegenstand macht. Der intensive Gebrauch von Leitbildern und Metaphern hilft dem Benutzer, sein Arbeitswissen direkt und sofort in der Anwendungsnutzung anzuwenden. Der WAM-Ansatz [Züllighoven 2004] liefert eine Konstruktionslehre zur menschengerechten Softwareentwicklung. Softwarewerkzeuge werden so entworfen, dass sie direkt die Arbeitsabläufe des Anwenders unterstützen. Konkretisiert wird diese Unterstützung durch die Entwicklung eines Benutzungsmodells. Darin werden alle Informationen über die Systemnutzung zusammengestellt. Geschäftsprozesse bilden einen wichtigen Ausgangspunkt, aber auch die Klassifikation der Benutzer und damit die Differenzierung der Systemleistungen im Hinblick auf fachliche und technische Erfahrungshintergründe sowie die Gewohnheiten des Benutzers sind zu beachten.

Als konstruktive Hilfe der benutzergerechten Anwendungsentwicklung wird auch oft die Nutzung von Styleguides genannt. Diese können aber die Geschäftsprozesse des Benutzers nicht berücksichtigen. Styleguides konzentrieren sich stattdessen auf die Gestaltung der Fenster und bilden damit nur wenig Unterstützung hinsichtlich fachlich optimierter Dialogabläufe.

Es gibt also durchaus konstruktive Hilfsmittel für die Softwareentwicklung im Hinblick auf Benutzbarkeit. Allerdings ist nicht sicherzustellen, dass allein durch Beachtung der Regeln zwangsläufig ein benutzungsfreundliches System entsteht. Deswegen sind neben den konstruktiven auch unterstützende analytische Maßnahmen erforderlich.

### **3 Das Usability-Labor**

Im Jahre 2003 wurde an der HAW-Hochschule für angewandte Wissenschaften in Hamburg ein Usability-Labor speziell für die Informatik eingerichtet, das einen breiten Einsatz in Lehre, Projektarbeit im Rahmen von Firmenkooperationen, Forschung und Entwicklung ermöglicht.

### 3.1 Architektur des Labors

Das Labor besteht aus einem Testraum (Abbildung 2) und einem Regieraum (Abbildung 3). Im Testraum bearbeiten eine oder zwei Testpersonen gemeinsam die vom Testleiter gestellten Aufgaben. Dabei werden sie von vier Kameras beobachtet. Das Gespräch zwischen den Testpersonen oder auch ein sog. Laut-Denken-Protokoll werden über zwei Mikrofone aufgenommen. Außerdem wird die Ausgabe auf dem Testrechner als weiterer Filmclip aufgezeichnet. Schließlich werden über eine Software (Eigenentwicklung) einige Metriken aufgezeichnet, wie etwa die Anzahl der Mausklicks, der Tastatureingaben, die auf dem Bildschirm mit der Maus zurückgelegte Strecke etc.. Diese werden sowohl auf einem Filmclip aufgezeichnet als auch für die spätere Auswertung alle halbe Sekunde kumulativ in eine Datei geschrieben.



Abbildung 2: Der Testraum

Der Testleiter verfolgt den Test in der Regel im Regieraum. Dazu hat er für jede Kamera sowie für die Bildschirmausgabe des Testrechners jeweils einen Monitor. Über die Mikrofone hört er das aktuell Gesprochene und kann aufgrund dieser Informationen gegebenenfalls über eine Gegensprechanlage in den Testablauf eingreifen und Hilfestellungen geben oder zur Bearbeitung der nächsten Aufgabe auffordern.



Abbildung 3: Der Regieraum

Nach Abschluß der Aufzeichnung wird der Film gerendert. Dabei wird eine Bild-in-Bild Einbettung aller Filmclips in einem Film vorgenommen, der zudem erheblich komprimiert wird und dann die Basis für eine Tiefenanalyse bildet. Abbildung 4 zeigt eine Momentaufnahme aus einem solchen Film, in der die vier Kamerasichten, der verkleinerte Bildschirminhalt des Testrechners und das Fenster mit den Metrikwerten erkennbar sind.

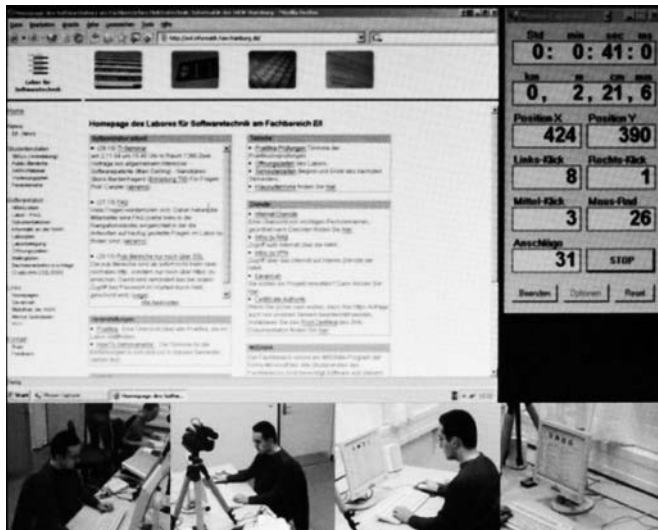


Abbildung 4: Momentaufnahme aus einem Film

Der technische Aufbau des Labors ist in Abbildung 5 wiedergegeben..

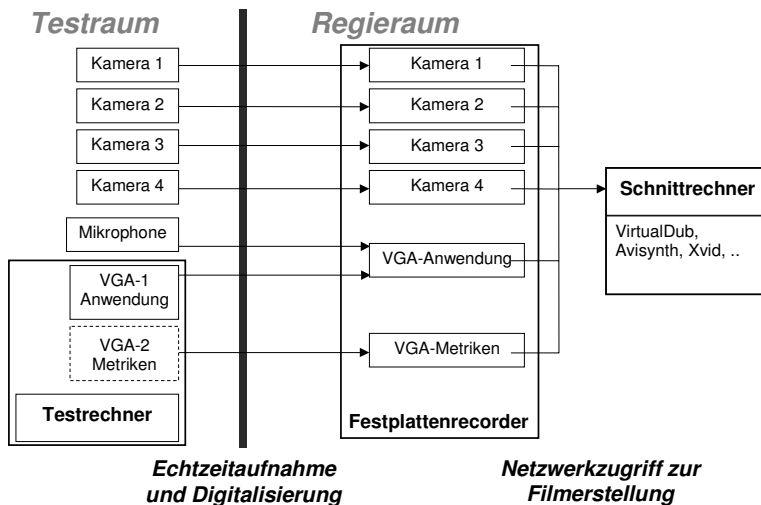


Abbildung 5: Architektur des Labors

Die Aufzeichnung der Kameras und Mikrophone erfolgt analog und parallel, damit über den Sitzungsablauf mehrere Filmclips entstehen, die das Geschehen aus unterschiedlichen Perspektiven zeigen. Vor der Speicherung auf den parallel arbeitenden Festplattenrecordern wird die Digitalisierung vorgenommen. Die auf diese Weise im Rahmen einer Aufzeichnung entstandenen Filmclips werden auf dem Schnittrechner zu einem Film zusammengeschnitten. Der Schnittrechner greift auf die digitalisierten Filmclips auf den Festplattenrecordern über ein Netzwerk zu.

### 3.2 Experimente im Labor

Eine Sitzung im Labor dauert in der Regel kaum länger als zwanzig Minuten. Es folgt eine Nachbereitung, in deren Rahmen der Testleiter in den Testraum geht, um mit den Testpersonen ein Gespräch auf Metaebene zu führen. Die Nachbereitung wird filmisch aufgezeichnet. Im Gespräch wird besonders die intentionale Ebene ins Gedächtnis gerufen, Gründe für Entscheidungen sowie direkt artikulierbare Verbesserungsvorschläge werden erfragt. Die Testpersonen haben in jedem Falle das Schlußwort.

### 3.3 Usability unter kontrollierten Bedingungen

Die Umgebung im Labor unterscheidet sich gravierend von der späteren Arbeitsplatzumgebung und damit von der Einsatzumgebung der Anwendungen. Im Labor werden sämtliche Störeinflüsse (zum Beispiel Telefon, priorisierte Anfragen von Kunden, Kollegen etc.) ausgeschaltet. Stattdessen wird der Ablauf der Sitzung detailliert beobachtet. Das Labor entspricht aber in seinem Ambiente den anderen üblichen Laborräumen im Hause. Auch ist sicher, dass Testpersonen sehr schnell in einen Zustand hoher Konzentration kommen („Eintauchen“, Immersion) und dann die Beobachtungsumgebung vergessen oder zumindest verdrängen.

Die Testpersonen können sich im Labor sehr beobachtet fühlen, kleine Schwächen werden sichtbar, sogar filmisch dokumentiert. Es ist daher in jedem Falle besonders hervorzuheben, dass niemals die Expertise der Testperson untersucht wird. Es geht vielmehr darum, die Anwendungssoftware zu testen und deren Defizite unabweisbar aufzuzeigen. Dabei spielt die Testperson die Rolle des aktiven Teils des Testsystems. Auch sind die Aufgaben zumeist so angelegt, dass höchstwahrscheinlich im Test Probleme auftreten werden, die damit nicht als Prestigeverlust für die Testpersonen gewertet werden können.

Die Wahl der Testpersonen verlangt erhebliche Aufmerksamkeit. Selbstredend kommt als entscheidende Instanz nur der spätere Systemnutzer in Frage. Die Testpersonen sollen dem Alter, Geschlecht und dem Erfahrungsschatz der späteren Nutzer entsprechen [Nielsen 1994], [Rubin 1994]. Er oder sie muss letztlich mit dem System arbeiten und ist daher letzte Instanz in der Beurteilung von Defiziten der Benutzbarkeit. Allerdings haben wir in vielen Evaluierungsprojekten eine erhebliche Kluft zwischen dem späteren Nutzer und erfahrenen Testpersonen festgestellt. Fehler zu finden erfordert Erfahrung. Deswegen benutzen wir zumeist in einer Pilotphase erfahrene Tester, die als Aussenstehende noch nicht die mögliche Betriebsblindheit der späteren Systembenutzer entwickelt haben und daher unbelastet testen können. Unsere Studierenden sind für diese Aufgabe gut geeignet. Damit wird vor den eigentlichen Tests die Verständlichkeit der Testaufgaben abgesichert, es werden aber auch bereits Defizite aufgedeckt, die den Testpersonen aus einem betrieblichen Umfeld eventuell nicht auffallen würden. Für die Einbettung von Usability-Tests in eine Anwendungsentwicklung halten wir es für erforderlich, mit den Tests zu beginnen, sobald erste Prototypen verfügbar sind. Die ersten Tests mit erfahrenen Testern beziehen sich auf die Kriterien der benutzungsfreundlichen Gestaltung von Dialogsystemen (Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit, Steuerbarkeit, Erwartungskonformität, Fehlertoleranz, Individualisierbarkeit, Lernförderlichkeit; DIN EN ISO 9241-110). Wenn Defizite der Prototypen im Hinblick auf diese Kriterien erkannt und möglichst auch beseitigt sind, sollte man zusammen mit den späteren Nutzern die Defizite des Anwendungssystems im Hinblick auf Einbettung in die Geschäftsprozesse und Arbeitszusammenhänge des Nutzers herausarbeiten.



Weiterhin beziehen sich Beiträge der Informatik auf vertiefende Experimente zur Immersion, zur Natürlichkeit des Handelns, zur nicht-gefühlten Beobachtung. Um diese Aspekte genauer untersuchen zu können, haben wir auch eine mobile Variante des Labors (mit eingeschränktem Leistungsumfang) angeschafft. Mit dieser können wir überprüfen, ob bzw. unter welchen Bedingungen eine Untersuchung der Gebrauchstauglichkeit direkt am Arbeitsplatz Sinn macht. In der mobilen Version des Labors können wir eine Kameranicht (Videokamera) mit einem Filmclip über den Bildschirminhalt und mit Metriken zusammenzuführen. Die Speicherung der unterschiedlichen Quellen erfolgt auf einem Notebook.

Die meisten Untersuchungsobjekte sind in ihrer Benutzungsschnittstelle so umfangreich, dass ein Test aller Funktionen des Systems wegen des damit verbundenen Aufwands unmöglich wird. Deswegen legen wir vor dem Test die Zielrichtung fest: Meistens geht es darum, die im Rahmen der Vorbereitung vermuteten Defizite unabweisbar zu dokumentieren. Ein Videofilm mit Dokumenten des Scheiterns berührt viel stärker, macht betreffender und ist damit wesentlich schwerer zu verdrängen als eine inhaltlich gleichgestellte Argumentation darüber. Verantwortlichen Entscheidern kann der Handlungs- und Entscheidungsbedarf bewiesen werden, mit Entwicklern kann konkreter über die Defizite und ihre Beseitigung kommuniziert werden. Daneben gibt es aber auch explorative Fragestellungen, bei denen dem Testleiter vor den Untersuchungen im Labor nicht klar ist, ob es Defizite gibt und worin diese bestehen.

Besonderer Wert muss in den Filmen auf ein Laut-Denken-Protokoll gelegt werden, ohne das die Auswertung schwierig bis unmöglich wird. Den meisten Testpersonen fällt es aber recht schwer, im Moment der Problemlösung die eigenen Gedanken zu schildern, was aber für eine Erfassung der intentionalen Ebene und für die möglichen Schwierigkeiten auf der semantischen Ebene (die richtigen Werkzeuge und deren Funktionen finden) erforderlich wäre. Allerdings benötigt man im Laut-Denken-Protokoll nicht etwa einen Diskurs über die Begründungen, weshalb bestimmte Wege eingeschlagen wurden, sondern nur eine Protokollierung, dass diese Wege gewählt wurden. Begründungen sind dann unter anderem Inhalt des nachbereitenden Gesprächs zwischen Testleiter und Testpersonen. Ein Laut-Denken-Protokoll entsteht auf natürliche Weise, wenn man statt einer Testperson zwei Personen gemeinsam die Aufgaben bearbeiten lässt. Die beim Problemlösen auf natürliche Weise auftretende Kommunikation zwischen den Testpersonen ist als Laut-Denken-Protokoll meistens gut geeignet.

## **4 Nutzung des Labors**

### **4.1 Lehre**

Das wichtigste Ziel des Labors ist es, möglichst jedem Studierenden im Laufe seines Studiums mehrfach vor Augen zu führen, welche Probleme ein Nutzer seiner Systeme tatsächlich hat. Diese werden meistens unterschätzt. Ein Entwickler hat oft keine Vorstellung von den Fragen und Problemen, die der Nutzer für eine erfolgreiche Arbeit mit dem System lösen muss. Auch hinsichtlich der Zeit, die für Arbeitsschritte erforderlich wird, haben Entwickler gar zu oft unrealistische Vorstellungen.

Also wird zum Beispiel im Rahmen der vorlesungsbegleitenden Praktika zum Software-Engineering die Qualitätssicherung durch eine Evaluierung der entwickelten Systeme im Usability-Labor thematisiert. Dabei werden die Aufgaben jeweils von einem anderen Team getestet, also nicht vom Entwicklerteam selber. Damit entsteht der willkommene Effekt, dass die Systeme auch in Abwesenheit der Entwickler funktionieren müssen.

Im Projektstudium haben wir seit Bestehen des Labors stets ein Angebot aus dem Usability-Labor. Darin enthalten sind Evaluierungsaufgaben, aber auch Weiterentwicklung der Technik des Labors. Zum Beispiel wurde das Werkzeug zur Erhebung der Metriken (s.o.) im Rahmen des Projektstudiums entwickelt.

Sehr viele Evaluierungsprojekte fallen im Rahmen von Abschlußarbeiten an (Bachelor, Master). Wenn eine Systementwicklung Teil der Arbeit ist, erhebt sich in jedem Falle die Frage nach der Evaluierung des Systems im Hinblick auf die speziellen Entwicklungsziele, aber auch im Hinblick auf die übergreifenden, stets gegenwärtigen Usability-Ziele, die von einer Anwendung nicht missachtet werden dürfen.

### **4.2 Firmenkooperationen**

Die Praxisorientierung des Studiums führt zu Aufgabenstellungen, die für unsere Firmenpartner von direktem Interesse sind. Zu den Abschlußveranstaltungen von Projekten laden wir auch stets Firmenvertreter ein. Dabei entsteht oft der Plan, Anwendungssysteme von Firmen einer vertiefenden Evaluierung der Benutzungsschnittstelle zu unterziehen.

Seit Bestehen des Labors wurden stets Kooperationen mit Firmen der Region durchgeführt. Die Wahrnehmung der Systems durch Kunden (z.B. Otto-Versand), für Mitarbeiter (z.B. Volksfürsorge, intersoft AG) sowie die Analyse technischer Konzepte (EADS) wurden in den Untersuchungen thematisiert. Die Tests wurden von Studierenden geleitet, die bereits in intensiver Projektarbeit im Rahmen der Lehrveranstaltungen die erforderlichen Kenntnisse und Erfahrungen gesammelt hatten.

Derartige Kooperationen gehen wir sehr gerne ein, denn damit werden dringende Praxisfragen in die Hochschule und in unsere Lehrveranstaltungen getragen. Dabei können und wollen wir natürlich Unternehmen, die Usability-Evaluationen professionell anbieten, keine Konkurrenz machen.

### **4.3 Forschung und Entwicklung**

Das Usability-Labor erlaubt auch Forschungs- und Entwicklungsaufgaben in sehr unterschiedlichen Richtungen.

Zunächst ist die vordringliche Frage nach der Aussagekraft der Untersuchungen und nach der zielgerichteten Interpretation und Auswertung aller erhobenen Informationen zu nennen. Zum Beispiel entstehen durch das Werkzeug für Erhebung von Metriken umfangreiche Datenbestände, deren Interpretation geleistet werden muss. Wie sind in diesen Daten Phasen der Eingabe, der Mausaktivität und des Nachdenkens möglichst sicher maschinell erkennbar und damit abgrenzbar [Wolf 2006]?

In den letzten Jahren rückte das Konzept der Barrierefreiheit durch das BGG-Behindertengleichstellungsgesetz in den Fokus des öffentlichen Interesses an Anwendungssystemen. In §11 Barrierefreie Informationstechnik heißt es, „IT-Systeme sollen so gestaltet sein, dass sie von behinderten Menschen grundsätzlich uneingeschränkt genutzt werden können.“. Im Labor können wir einerseits Benutzungsschnittstellen im Hinblick auf barrierefreie Gestaltung evaluieren, andererseits Werkzeuge zur Unterstützung der Barrierefreiheit (z.B. Screenreader) untersuchen.

Bisher erfordert eine Versuchsauswertung eine Filminterpretation mit Mitteln, die auch bei Filmen im Unterhaltungsbereich angewandt werden und die daher den Auswertenden sofort vertraut sind. Doch erhebt sich die Frage, ob man jenseits einer derartigen sprachlichen und psychologischen Inhaltsinterpretation ohne messende Methoden weitere, ggf. objektivere Informationsquellen erschließen kann. Hierzu gehört die Blickmessung mit Eye-Trackern.

Neben die Ebene der Filminterpretation treten weitere Interpretationsansätze. Hier ist zunächst der Trace der Anwendung zu nennen. Durch Benutzereingaben entstandene programminterne Vorgänge werden intern aufgezeichnet und dann (white-box) in die Interpretation einbezogen. Unsere Ansätze nutzen die aspektorientierte Programmierung [Elrad 2001], die eine Instrumentierung des Untersuchungsobjektes ermöglicht, ohne dieses gravierend zu verändern [Bruns 2005].

Genauso wesentlich für eine ganzheitliche Interpretation ist eine Protokollierung der sich im Testablauf eventuell verändernden Befindlichkeit der Testperson. Empfindet diese Stress, Wut, Begeisterung? Welche Auswirkungen haben unverständliche Fehlermeldungen? Welche Auswirkungen haben Störungen? In welchen physiologischen Faktoren sind diese Gefühle erkennbar, bzw. welche Größen können als Indikatoren herangezogen werden? Wie lassen sich diese Größen messen und in die Auswertungssystematik einbeziehen?

Das Labor versteht sich auch als Dienstleister für andere Entwicklungsthemen in der Informatik. Zum Beispiel entwickeln wir eine Untersuchungstechnik und –methodik für die Evaluierung im Rahmen des Ubiquitous Computing. Allgegenwärtige Geräte und Anwendungen zu evaluieren erfordert auch, dass die Evaluierungstechnik in ihrer Wahrnehmung durch den Einsatzkontext zurücktritt.

Die bisher im Labor eingerichtete Grundausrüstung erlaubt kontrollierte Experimente zur Gebrauchstauglichkeit. Ein weiterer Ausbau ist geplant in Richtung auf psychologische Untersuchungen, ähnlich zu den Arbeiten an der Universität Osnabrück [Gediga 2002], [Gediga 2002b]. Analog zu Debatten in der Psychologie experimentieren wir mit unterschiedlichen Arten des Lauten Denkens sowohl für einzelne Personen als auch für Teams aus zwei Personen.

Schließlich ist das Hauptziel zu nennen: Welche im Labor gesammelten Erkenntnisse kann man so abstrahieren, dass daraus Regeln für eine um Usability-Aspekte angeereicherte Konstruktionslehre für Anwendungssysteme entstehen? Damit würden aus der analytischen Qualitätssicherung konstruktive Hilfestellungen abgeleitet werden.

## **5 Fazit**

Das Usability-Labor verfolgt in der Hauptsache das Ziel, den Studierenden der Informatik Sichtweisen der Gebrauchstauglichkeit zu vermitteln. Es soll klar werden, wie schwierig und differenziert die Gestaltung von Benutzungsschnittstellen sein kann. Studierende sollen für Fragen der Gebrauchstauglichkeit sensibilisiert werden. Dadurch soll auch eine gewisse Bescheidenheit entstehen, die eine Erkennung der eigenen Grenzen ermöglicht. Bestimmte Regeln kann man leicht befolgen und damit entstehen Systeme, die zunächst als akzeptabel gelten können. Weiterführendes erfordert auch besondere Maßnahmen, interdisziplinäre Einbindung spezieller Qualifikationen etc..

In Kooperationen mit Firmen beleben wir ständig den Praxisbezug unserer Ansätze. Wir helfen dabei unseren Firmenpartnern in einer Untersuchungsrichtung, die bisher weithin zu wenig Beachtung fand und geben unseren Studierenden eine Möglichkeit, nach der entsprechenden Lehrveranstaltung vertiefende echte Praxiserfahrung zu sammeln und in diesem Zusammenhang auch durchaus wichtige eigene Kontakte zu Firmen zu knüpfen.

Wir sind nun auch in der Lage, in Kooperation mit Psychologen und Gestaltern Werkzeuge zu entwickeln, die es ermöglichen, Konzepte aus diesen Bereichen zu untersuchen und zu belegen. Informatik kann damit in ihren Grenzbereichen im Rahmen interdisziplinärer Kooperation Hilfestellung für Gestalter und Psychologen liefern.

Wir werden natürlich den Arbeitsschwerpunkt Usability in diesem Sinne weiter intensiv ausbauen, die Analysemethodik verfeinern, die Technik des Labors weiterentwickeln.

## Literaturverzeichnis

- [Bruns 2005] Andreas Bruns: Integration von Usability-Methoden in den agilen Entwicklungsprozess am Beispiel der Erfassung und Auswertung von Tracingdaten. Diplomarbeit, HAW Hamburg, 2005.
- [Elrad 2001] Elrad, T.; Filman, R.E.; Bader, A.: Aspect-Oriented Programming. In: CACM, 44(2001), Nr.10, S.33-38
- [Gamma 1994] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: Design Patterns – Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994
- [Gediga 2002] Gediga, G., Hamborg, K.-C. & Düntsch, I. (2002). Evaluation of Software Systems. In: A. Kent (Ed.), Encyclopedia of Library and Information Science (S. 166-192), Volume 72. New York: Marcel Dekker, Inc.
- [Gediga 2002b] Gediga G. & Hamborg, K.-C. (2002). Ergonomische Evaluation von Software: Methoden und Modelle im Software-Entwicklungsprozess. Zeitschrift für Psychologie, 210 (1), 40 – 57.
- [Herczeg 1994] Herczeg, M.: Software-Ergonomie. Grundlagen der Mensch-Computer-Kommunikation. Addison-Wesley, 1994.
- [Krasner 1988] Krasner, G.E.; Pope, S.T.: A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80. JOOP, 1(3), 1988, 26-49.
- [Nielsen 1994] Nielsen, J.: Usability Engineering. Academic Press, 1994.
- [Norman 1988] Donald A. Norman: The Psychology of Everyday things. Harper-Collins, 1988.
- [Rubin 1994] Rubin, J.: Handbook of Usability Testing. 1994.
- [Wolf 2006] Wolf, D.R.: Bereitstellung effizienter Auswertungsmethoden für Usability-Tests sowie prototypische Entwicklung eines Werkzeuges zur Analyse. Bachelorarbeit, HAW Hamburg, 2006 (in Vorbereitung).
- [Züllighoven 2004] Züllighoven, H. et.al.: Object-Oriented Construction Handbook. dpunkt; Morgan-Kaufmann, 2004.

