

Interaktionsdesign und sprachliches Handeln

Vortrag Colloquium Hoffmann, 28. Juni 2005

User Interface Design

Folie 1: Der Rahmen unserer Arbeit

Computer sind Artefakte, d. h. irgend jemand hat **Designentscheidungen** getroffen, mit denen der Benutzer und die Benutzerin konfrontiert werden, wenn sie eine Software dazu verwenden, einen bestimmten Zweck zu erreichen.

Software wird zum größten Teil dazu eingesetzt, Lohnarbeit zu erledigen, die Benutzer haben häufig keine direkte Wahl. Designentscheidungen führen bei **Investitionsgütern** in unterschiedlichem Masse zur Zufriedenheit des Benutzers, sie tragen zur Effizienz bei etc.

Der private Gebrauch von Software ist vermutlich am ehesten im Kontext Webdesign zu reflektieren, wo Erfolg als **Konversionseffekt** gemessen wird.

Ein großer Softwarehersteller spricht ganz allgemein von seiner Zielgruppe als ‚Information Worker‘, und UI Design hat die Aufgabe, **Informationsverarbeitung** (besser: -bearbeitung) durch die Gestaltung geeigneter Werkzeuge zu unterstützen.

Zur Rolle des ‚Entwicklers‘

Von außen betrachtet scheint es so, als würden diejenigen, die die Erscheinung der Software bestimmen, auch dieselben sein, die diese entwickeln. Das ist das Modell der 70er Jahre, und es stimmt schon längst nicht mehr, zumindest nicht für große Firmen, Abteilungen oder Produkte/ Projekte. In der praxis-fremden Literatur findet man gelegentlich Aussagen wie: „Die **Entwickler** haben nicht an den Benutzer gedacht“, das verunklart die Situation. Entweder werden die verschiedenen Rollen in der Softwareentwicklung nicht differenziert, d. h. die unterschiedlichen Verantwortlichkeiten, Ausbildungen etc. nicht reflektiert, oder der Prozessschritt UI Design wird unterschlagen. Daraus ergibt sich ein recht märchenhaftes Bild im Kopf des Zuhörers, nämlich so, als würden böse Feen, die nächtelang rätselhaft Programmcodes einhacken, dieselben sein, die am Ende über Farben und Formen auf den Screens entscheiden. So etwas gibt es nicht mehr, User Interface Design als Disziplin ist über 30 Jahre alt, und sobald mehr als 3 Leute sich eine Aufgabe teilen, tritt auch hier eine Differenzierung der Verantwortlichkeit zu Tage.

Entwickler dafür verantwortlich zu machen, dass man eine Taste nicht gleich findet, klingt so ähnlich, als würde man sich über die Maurer beschweren, wenn man in der Uni das Klo nicht finden kann.

Wer macht UI Design?

UI ist schlecht, obwohl sich sogenannte **UI Experten** damit befassen. Oder gar WEIL sie es tun. Es gibt in Deutschland keine etablierte Ausbildung zum UI Designer, in der Regel handelt es sich, aus den USA, aber auch Skandinavien übernommen, um eine Spezialisierung von Psychologen, zuweilen Pädagogen, wobei das Berufsinteresse aus dem europäischen Raum eher dahin ging, Arbeitsbedingungen für Menschen zu verbessern, während die amerikanische Tradition eher dadurch motiviert ist, Computer aufzurüsten, um Handlungsziele von den Begrenztheiten des menschlichen Gehirns unabhängig zu machen. Beiden Ansätzen gemeinsam ist die Annahme, dass der Computer **Schwächen des Menschen** auszugleichen hätte.

In einem vielleicht vertrauteren Modell ausgedrückt: Werbung in seiner Kleinstadtmittelstandsvariante bemüht sich, einen uninformierten Verbraucher über die Vorzüge eines Produktes zu informieren, in der Hoffnung, dass dieser sozusagen in Kopfrechnung einen Nettonutzen erkennt und das Produkt erwirbt. Bis zu einem gewissen Grade ist das auch erfolgreich, und es ist nicht einfach. (Ein ganz vergleichbares Modell findet man übrigens auf dem Feld Gebrauchsanweisungen.) Richtig teure Werbung aber tut etwas ganz anderes: sie nimmt das Menschliche am Menschen ernst und kommentiert oder konfrontiert und erreicht Handlungsziele unter Inanspruchnahme von kognitiven und emotionalen Leistungen.

De facto gibt es übrigens sehr wenig Crossover von der kreativen Werbung zum UIDesign, was möglicherweise hauptsächlich mit wirtschaftlichen Gegebenheiten erklärt werden kann.

Fehler! Es ist nicht möglich, durch die Bearbeitung von Feldfunktionen Objekte zu erstellen.

Folie 2: Die Profession des UI Designers

Wer sollte UI Design machen?

Damit komme ich zur **Linguistik**. Ich brauche das hier nicht näher auszuführen, aber es sei daran erinnert, dass Kommunikation dann funktioniert, wenn jeder Gesprächspartner die Fähigkeit seines Gegenübers, angesichts gemeinsamen Wissens und ausgetauschter Informationen Handlungspläne zu entwickeln, wahr und ernst nimmt.

An dieser Stelle ist aber auch darauf hinzuweisen, dass Linguisten professionell notwendige, aber nicht hinreichende Fähigkeiten zum Designprozess von User Interfaces beitragen können: die Reflexion dessen, was gestaltet werden muss. Die Befähigung zur kreativen Lösung ist hier vermutlich nicht erlernbar.

Meine Vision für das Thema Linguistik und UI Design ist, dass Linguisten in einem Feld, das ganz eindeutig als multi-disziplinär anerkannt ist, eine Rolle, oder eine größere Rolle spielen. Dabei geht es selbstverständlich nicht um ein Ad-hoc-Einbringen von Wissens-elementen (auch wenn das Reizmaterial unten es so erscheinen lässt), sondern darum, das Thema UI Design aus der Sicht von Kommunikationsmodellen zu begreifen.

Ich selber habe bei Jochen Rehbein in Hamburg den Magister in Linguistik gemacht und arbeite seit 10 Jahren als User Interface Designer bei Siemens, in wechselnden Rollen. Ich merke in der alltäglichen Gestaltungsarbeit, dass ich meine Interaktionsmodelle aus der Funktionalen Pragmatik beziehe, aber eine aktive Zusammenführung von Linguistik und UI Design habe ich selber kaum betrieben, noch findet man sie in der Literatur. Ich wünsche mir eine engere Zusammenarbeit mit Sprachwissenschaftlern, und unsere Beteiligung am Colloquium ist ein Versuch, Schritte für die Zusammenarbeit auszuloten.

Psychologen tragen in vielfältiger Weise zum Thema bei, es wird von ihnen nahezu dominiert, das wird im Methodeninstrumentarium deutlich, bei den Herangehensweisen und Themenschwerpunkten. Ich will das hier nicht vertiefen, aber die deutlichsten Spuren hat die Psychologie im Bereich des Testens hinterlassen (**Usability Test**), sie bemüht sich seit Jahren, Bedienoberflächen zusammen zu **basteln**, und sie erkennt neuerdings Themen wie **‚Emotionen‘**, **‚Akzeptanz‘**, **‚Workload‘** und so weiter.

Wie wir UI Design machen

Ein Wort zur **Kreativität**. Auch wenn es m. E. im UI Design ohne Kreativität nicht geht, will ich mich heute nicht darüber äußern, wie sie entsteht oder gar im Prozess verankert wird. Wir sind uns ja hier auch klar, dass Philologien eben gerade nicht kreativ, sondern reflektierend mit ihren Gegenständen umgehen.

Mein Hauptanliegen, das diesen Vortrag durchziehen wird, ist eher Sensibilität und Offenheit. Zwei Gestaltungslösungen sind eben funktional nicht gleich, auch wenn sie beide dieselben Fehler nicht aufweisen oder anhand einer endlichen Liste von Kriterien als gleichwertig zu bezeichnen sind – **Sensibilität**. Wenn sich keine Unterschiede mehr finden lassen, muss man den Rahmen der funktionalen Erklärungen erweitern – **Offenheit**.

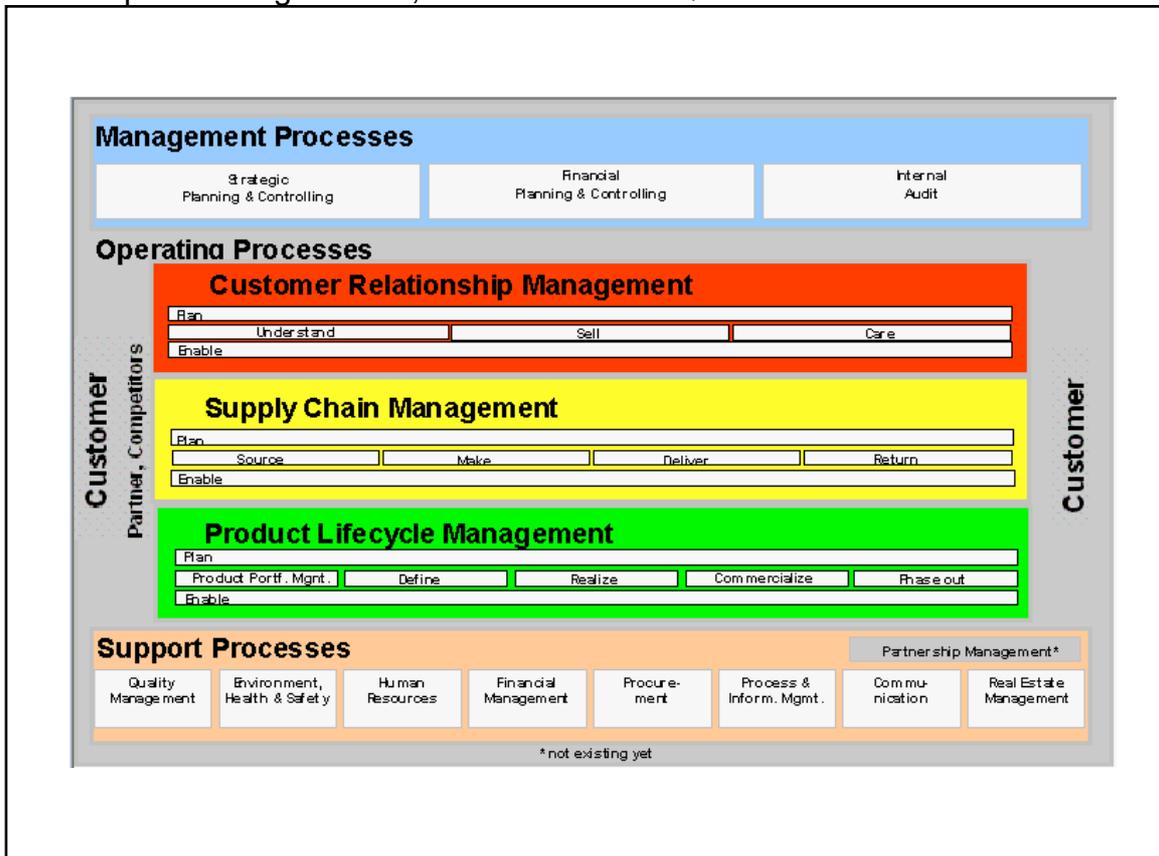
Programmatisch ausgedrückt und um es in einer Analogie zur Kommunikation und in einem Beispiel deutlich zu machen: Wenn jemand an der Supermarkttheke Käse kauft, so lässt sich aus Sicht der Interaktionspartner feststellen, dass sie, wenn beide ihr Handlungsziel erreichen, nicht mehr über die eingesetzten Mittel nachdenken. Wenn wir als Sprachwissenschaftler diese Interaktion verstehen wollen, werden wir da reflektierend herangehen müssen. Dabei ist uns auch klar, dass wir das Geschehen vermutlich mit dem Duden in der linken und einer Mittelstufengrammatik in der rechten Hand nicht entschlüsseln werden, d. h. wir benötigen Analyseverfahren. Analysen dieser Art sind auch im UI Design notwendig, um produzierte Gestaltungen zu verstehen.

Der große Unterschied liegt selbstverständlich darin, dass es zu unserer Profession gehört, diese Gestaltungslösungen selber zu produzieren und ihre Qualität danach zu analysieren, um weiter zu iterieren.

Stakeholder

Anhand des Siemens-Prozesshauses will ich kurz erläutern, dass sich niemand einfach was ausdenken kann, wenn es um SW Entwicklung geht, sondern Input und Output sowie der Binnenprozess geregelt ablaufen.

Was ich zeige, ist Siemens, aber man kann sagen, dass es durchaus dem Standard in der Industrie weltweit entspricht. Das hat auch damit zu tun, dass SW entweder Auftragsentwicklung oder längerfristige Kooperation darstellt, und damit die Auftragnehmer (Kunden) ihr Vertrauen nicht aus dem Funktionieren des Endproduktes gewinnen, sondern aus der Qualität des Prozesses.



Folie 3: Komplexer Gesamtprozess mit verteilten Verantwortlichkeiten

Für uns wichtig zu verstehen ist, dass sich die Produktentwicklung links Erwartungen von Kunden abholt und diese rechts mit einem funktionierenden Produkt beantwortet. Als Kulturmodell ist das sehr viel moderner, als man es vermuten würde, das graue Feld im Hintergrund, auf dem alle Aktionen liegen, repräsentiert ein Gefüge von vorhersehbaren Normen und Werten, und die Handlungsschritte der Aktanten im SW Design fügen sich dort ein.

Die Ausrichtung von Entwicklungsprozessen auf den Endkunden entspricht durchaus der Vorstellung von kultureller Eingebundenheit auch des sprachlichen Handelns, es kommt insoweit der funktionalen Pragmatik sehr viel näher als etwa aus der Black Box der behavioristischen Psychologie bzw. einer kognitions-wissenschaftlichen Herangehensweise abgeleiteten Erwartung an Sprache.

SW-Design übrigens befindet sich im Define- und z. T. im Realize-Kästchen des Product Lifecycle Management.

Softwareentwicklung ist nicht das Metier von genialen Chaoten, die in einer leer stehenden Garage an irgendwas herumfummeln!

Fehler! Es ist nicht möglich, durch die Bearbeitung von Feldfunktionen Objekte zu erstellen.

Folie 4: Prozessbeteiligte UI Design

Prozessmodell UI Design

Im folgende werde ich mich mit der Binnenstruktur unserer Aufgaben auseinander setzen und kurz darstellen, wie der **UI Designprozess** sich gestaltet.

Fehler! Es ist nicht möglich, durch die Bearbeitung von Feldfunktionen Objekte zu erstellen.

Folie 5: UI Design im Prozess

Anforderungen

Die Liste der **Anforderungen** ist tendenziell unbegrenzt, jeder Mitarbeiter und Kunde kann Anforderungen einlasten und wir sind in den meisten Bereichen verpflichtet, diese zu listen und zu beurteilen. Diese Leute, die Anforderungen einbringen, übernehmen aber keine Produktverantwortung.

Produktverantwortung heisst, Anforderungen zu bewerten, denn da gibt es oft Duplikate und Widersprüche, Unsinniges und Unerlaubtes, schon Vorhandenes und technisch Nicht-Machbares. Ziel ist, die Anforderungen nach vorher festgelegten Kriterien zu beurteilen und das Resultat zum Produktvorschlag zu verdichten.

Wichtig zu verstehen ist, dass Anforderungen lösungsfrei sein müssen. Werden sie als Lösungen formuliert, sind dies nur Bilder der Kommunikation, aber in diesem Prozessschritt hat niemand das Recht eine Lösung vorzudefinieren.

Interessant ist dies wiederum aus dem Kommunikationshandeln. Es ist sehr wohl möglich, jemanden zu einer kommunikativen Handlung zu verpflichten („Entschuldigen Sie sich!“ oder auch „Wie spät ist es?“, „Würden Sie sich bitte kurz vorstellen!“), das wäre eine Anforderung. Niemand käme aber auf die Idee, dem Interaktionspartner die genaue Formulierung oder die Tonhöhe oder den Dialekt vorschreiben zu wollen. Variationseinschränkungen in der Auswahl der Formulierung etwa sind strikt auf den

juristischen Diskurs beschränkt und beziehen sich immer auf die Proposition. Alles andere ist verhandelbar. Dies lässt sich mit mentalen Begrenzungen erklären, vor allem aber mit der notwendigen Situationsanpassung von – Lösungen.

Die Liste der Anforderungen, die für das Produkt gelten sollen, gehen in unseren Prozess ein. Dies ist allerdings ein Handshake, kein Handover: wir können sowohl die akzeptierten als auch die nicht-akzeptierten Anforderungen verhandeln. Und es ist auch allen klar, dass UI Design zuvor nicht formulierte Anforderungen einbringen kann.

Modellierung

Bisher haben wir von Listen gesprochen, die kann erstens kein Mensch verarbeiten und zweitens muss so schnell wie möglich die Perspektive einer Gesamtheit zurückgeholt werden, und Modellieren tut genau dies auf viele verschiedene Weisen:

Use Cases – das ist die Fassung von Anforderungen als Handlungsabläufe. Damit wird aus der Liste eine Geschichte.

Mental Models – das ist der Versuch, das Verhältnis von Anforderungen, wie es der Benutzer im Sinne hat, zu modellieren. Wichtigkeit, Häufigkeit, Schwierigkeit, Nützlichkeit; Erwartungen, Erfahrungen, Befürchtungen; Tauglichkeit, Attraktivität, Überraschung; Nutzen und Emotionen – unter all diesen Aspekten werden die Anforderungen bedacht und gewägt. Treibender Input bei Investitionsgütern ist die Frage, wie gut Menschen mit dem Produkt ihre Arbeit erledigen können. (Siemens verdient Geld damit, dass unsere Kunden mit unseren Produkten Geld verdienen.)

Hier möchte ich daran erinnern, welches Menschenbild der Organisation zugrunde liegt. Es gibt bei Siemens Menschen die sagen, man muss nicht auf die Kunden hören, denn jeder will was anderes. Es scheint, als würde auch die Psychologie sich dem Individuum bei komplexen Interaktionen so nähern. In der HCI-Gemeinde ist Cooper dem auf eine recht fragwürdige Weise begegnet, indem er die sogenannten *personas* einführte. Es gibt in der Sprachwissenschaft eine lange Tradition des Streites darüber, wie viel Sprachwissen dem einzelnen Sprecher zugetraut werden kann, zu meiner Zeit in der Linguistik musste man sich noch immer mit Chomsky und seiner Idee von der angeblich so sehr defizitären Performanz auseinandersetzen. Die funktionale Pragmatik erkennt den authentischen Sprachgebrauch als wesentliche Wissensquelle an, wenn man etwas über Sprache wissen will. Im

Produktdesign ist der Endbenutzer der Experte, wenn es um die Bedingungen und Notwendigkeiten für seine Werkzeuge geht: die Software.

Ingenuity

UI Design, wie einer unserer Kollegen mal gesagt hat, lässt sich nicht ausrechnen. Anzunehmen, dass sich Modelle finden ließen, die sich quasi-mathematisch ableitend in Lösungen überführen ließen, ist Unfug. Wäre dem anders, wäre sowohl der Terminus **Design** (haben wir nicht gelernt, dass Zeichen, signs, arbitrary sind?) falsch, als auch der des **Engineering**. Um einen Ausflug in die Volksethymologie zu machen, ohne Genius kein Ingenieur, zumindest dem Anspruch nach.

Damit ist übrigens keine Kreativität um der Kreativität willen gemeint, sondern in etwa dieselbe Gestaltungsfreiheit, die wir grundsätzlich in unserem Kommunikationshandeln besitzen: welche Kleider wir tragen, wie wir von einem Ort zum anderen gelangen, wie wir uns sprachlich ausdrücken ist einerseits durch Regeln geleitet (das sind die Modelle, von denen ich oben sprach), andererseits haben wir aber einen Grad an Freiheit zur Performanz, den man niemandem abspricht, aus dem aber in gleichem Maße Qualität entspringt. Ohne Performanz wird halt die Kompetenz nicht sichtbar, ich bin sicher auch dies schließt an etliche Diskussionen in der gegenwärtigen Linguistik an.

Solution incl. Graphic Design

Die in genialer Weise beantworteten Anforderungen drücken sich also in Lösungen aus. Aus Lösungen leitet sich unsere Aufgabe als UI Designer ab, indem wir Lösungen reflektieren und herausfinden, welche dem Zweck des Produktes und der engeren Gebrauchsumgebung, aber auch dem Bild der Fa. Siemens oder des Geschäftsbereichs, der Domäne der Verwendung, der Zeit und dem Ort der Anwendung etc. angemessener ist.

Im nächsten Teil des Vortrages werde ich darauf genauer eingehen und auch über das Verhältnis von Grafikdesign zu Lösungskonzept sprechen. Hier nur soviel, dass die Darstellung in Farben und Formen das Vehikel darstellt, Lösungen zu kommunizieren.

Grafikdesign ist im Designprozess das letzte, was passiert, aber für den Benutzer das erste, was er sieht.

Ich will die Analogien zur Sprache nicht übersimplifizieren, man könnte aber in etwa folgende Parallelität ausmachen: die gestalterische Darstellung (letztlich beliebige Farben und Formen) entsprechen der Stimme, dem Dialekt und der Lautstärke. Die nicht beliebigen Farben und Formen, letztlich Entscheidungen über Fokussierungen, Gruppierungen, Gemeinsamkeiten und Unterschiede der Form/Farbe entsprechen den Suprasegmentalfunktionen der

Stimmführung. Die dargestellten Elemente sind Propositionen und als Verweise illokutive Anteile.

Prototype

Selbst UI Designexperten haben große Mühe, sich das Erfahren einer Software anhand einzelner statischer Bilder oder gar Modelle vorzustellen, daher werden fast immer Prototypen erstellt. Damit werden dann zwar keine Apparate gesteuert, aber wenn man eine Bediengeschichte (Story Board) erzählt, werden die geplanten Interaktionen erfahrbar.

Prototypen helfen beim Interieren der Designprozesse, sie sind mehr oder weniger ähnlich wie die tatsächliche Software (low fidelity, high fidelity).

Viele Artikel der empirischen Sprachforschung analysieren einzelne Passagen von authentischen Sprechhandlungen, präsentieren aber das Gesamttranskript im Anhang. Der kommunikative Zweck ist derselbe wie beim Erstellen von Prototypen im Softwaredesign: Ganzheitlichkeit herzustellen. So wenig wie jemand in Einzelwörtern kommuniziert, so wenig wird Software in Einzelfunktionen bedient.

Validation

Dies sind Verfahren der Ethnologie oder Soziologie, manchmal auch des alltäglichen Kommunikationshandelns, und das Ziel ist, die Annahmen, die in gestalteten Lösungen „sedimentiert“ (K. Ehlich) sind, auf ihre Gültigkeit hin abzuklopfen. Dies ist ein Rückkopplungsverfahren und soll ebenfalls dazu beitragen, Qualität sicherzustellen. Ein Nebeneffekt ist, dass die Gestalter nicht nur etwas über das Produkt lernen, sondern auch über ihre Modellierungs- und Problemlösungskompetenzen. Prototypen werden hierbei als Vehikel eingesetzt.

Wichtig ist, dass authentische, erfahrene Benutzer als Validierungspartner quasi-fertige Lösungen validieren. Derjenige, der den Diskurs steuert, verweist die Diskutanten (meist eine Gruppe) darauf, die Lösungen an ihren Anforderungen zu spiegeln. Die Diskutanten handeln die Angemessenheit der Lösung für eine ihnen gemeinsame Aufgabe aus.

Für die Sprachwissenschaft interessant wäre erstens die Analyse des Diskurses Validierung. Was geschieht genau, wenn verschiedene Benutzer das Mapping von Bedienoberflächen auf ihre Arbeitsweise aushandeln?

Eine andere Nähe ergibt sich daraus, dass in recht artifiziellen Validierungssitzungen etwas in den Prozess hineingeholt werden muss, was man in der alltäglichen Kommunikation ad hoc und in situ erfährt, nämlich die Reaktion des Hörers auf eine Sprechhandlung, die es dem Sprecher ermöglicht, seinen

Handlungsplan anzupassen. Dergleichen ist im SW-Gestaltungsprozess ausgeschlossen und muss sozusagen simulierend adressiert werden. (Microsoft versucht dergleichen zaghaf, indem etwa Hilfe über das Internet angeboten und Feedback auf diesem Wege eingeholt wird.)

Das erinnert an die asynchrone Kommunikation der Schriftlichkeit, bereichert um die Komplexität einer eins-zu-vielen Situation.

Ein dritter Anknüpfungspunkt zur Linguistik ergibt sich aus sehr ähnlichen Bedingungen für Gebrauchsliteratur und Medien.

Ein vierter zur Literaturwissenschaft. Zu den inzwischen sehr ausgefeilten Prozessen der Entstehung von Literatur gehört, dass manche Autoren (erst Recht in der Filmindustrie) ihre Werke mit einem Zielpublikum testen, bevor es als Buch, Film oder Show freigegeben wird.

Testing

Das Usability Testing ist als Verfahren der Qualitätssicherung fest etabliert und wird m. E. leider gewaltig überschätzt. Dieses Verfahren hatte die Psychologie im Gepäck, als es in den 80 Jahren über die Ergonomie in die Qualitätssicherung von SW Design gelangte. 1993 errechnete der HCI (Human Computer Interaction) Guru Jacob Nielsen, dass 5 Benutzer 90% aller Softwareprobleme finden. Damit hatte ein komplexer, nicht fassbarer, nicht beherrschbarer und im Kern beängstigender Prozess eine Art Sicherungsleine gefunden, und UI Design lässt sich leider seither über Usability und Usability Testing verkaufen. Ich hatte eingangs erwähnt, dass UI Design von manchen als Mangelverwaltung betrieben wird. Usability Testing ist das sichtbarste Symbol dessen.

Das Usability Testing hat seine Daseinsberechtigung, wenn es als Maßband beim Design eingesetzt wird. Über die Qualität einer Lösung sagt es nichts aus. Manchmal, und das ist Literaturwissenschaftlern sehr vertraut, ist gerade derjenige erfolgreich, der die Regeln durchbricht.

Fehler! Es ist nicht möglich, durch die Bearbeitung von Feldfunktionen Objekte zu erstellen.

Folie 6: Details und Beispiele zu den Stufen des Designprozess

Fehler! Es ist nicht möglich, durch die Bearbeitung von Feldfunktionen Objekte zu erstellen.

Folie 7: Projektgeschäft und Forschung

Ohne Ideen ist ein UI Prozess sinnlos, ohne UI Prozess gibt es keinen Platz für Ideen!

6 Ebenen des Designs

Designen ist, wie sprachliches Handeln auch, Optimieren, und beim Einschätzen der Qualität wird es immer wieder darum gehen, verschiedenartige Regelverstöße in ihren Auswirkungen gegeneinander abzuwägen.

Das Folgende ist der Versuch, die Parameter, die bei der Gestaltung von Bedienoberflächen einfließen, zu gruppieren und ihre sich überschneidenden Sinnzusammenhänge (Funktionen) zu erläutern. Anders ausgedrückt, trifft UI Design Einzelentscheidungen im Hinblick auf jede der 6 angegebenen Schichten.

Fehler! Es ist nicht möglich, durch die Bearbeitung von Feldfunktionen Objekte zu erstellen.

Folie 8: Entscheidungen auf 6 Ebenen

Information architecture

Wenn eine Eingabemaske für Patientendaten gestaltet werden muss, obliegt es nicht dem UI Designer, die notwendigen Felder festzulegen, dieser Aspekt der Gestaltung 'gehört' den Systemanalysten. Ob Postleitzahlen 4 oder 5-stellig eingegeben werden können oder das Format der Kreditkartennummer sind keine wirklichen Designentscheidungen. Dass ein Fahrkartenautomat irgendwie den Preis der Fahrkarte anzeigen muss, damit der Kunde einen ausreichenden Betrag einwerfen kann, ist nicht verhandelbar.

Die Informationsarchitektur stellt also dem UI Design ein Gerüst an Daten und Regeln bereit, meist ist dies auch der Startpunkt beim Designvorgang.

Die Interaktion zwischen Benutzer und System wird dabei allerdings noch nicht berücksichtigt.

Vielleicht könnte man sagen, dass die Informationsarchitektur der Proposition der Designäußerung entspricht.

Es gibt einen nur sehr geringen Gestaltungsspielraum der UI Designer im Hinblick auf die Informationsarchitektur, allerdings sei an den oben erwähnten Handshake im Prozessschritt von Anforderung an Lösung erinnert. Je mehr Daten zur Erfüllung einer Aufgabe darzustellen sind, desto geringer wird der Gestaltungsspielraum des UI Designs. Ich selber bin der Meinung, dass sich das Notwendige näher am Sinnvollen befindet, als das Mögliche; ich glaube auch, dass der Versuch, das Mögliche komplett abzubilden, aus dem Fehlen des Verständnisses der Arbeitsabläufe von Benutzern herrührt oder der Angst, Entscheidungen zu treffen oder genauer: dabei Fehler zu machen.

In diesem Zusammenhang ist an die Griceschen Konversationsmaxiomen zu erinnern, das Relevanzprinzip gilt im UI Design genauso wie in der verbalen Kommunikation, Benutzer werden immer versuchen, den präsentierten Informationen einen

Sinn in ihrem Bedienhandeln zuzuordnen und sind dann von der Fülle schnell verwirrt. s

Graphic design

Damit kommen wir zu der Schicht, die nicht nur als erste wahrgenommen wird, sondern häufig mit dem Aspekt „Design“ vollständig assoziiert wird: Farben, Formen etc. In einem allzu sehr banalisierenden Modell von UI Design wären Fachexperten für die Systemarchitektur zuständig und ein Designer würde dem ganzen Farbe verleihen, vielleicht noch ein paar schmückende Ideen einbringen, und das sei dann die Bedienoberfläche, wie der Benutzer sie erlebt. Dies entspricht nicht dem Prozess im UI Design von Software, wie wir ihn hier postulieren, es wäre auch ganz und gar unangemessen, Bedienoberflächen werden gebraucht, sie sind Teil eines größeren Handlungszusammenhangs, sie reflektieren Vorwissen und Erwartungen der Benutzer und tragen die Vielschichtigkeit von Zwecken und Zielen in sich – alle diese Aspekte sind in jeder Bedienoberfläche enthalten, und ich möchte postulieren, dass gestalterische Qualität davon abhängt, wie gut alle beteiligten Ebenen repräsentiert werden, nicht nur die präanalytisch offensichtlichen.

Die Kombination aus Informationsarchitektur und graphischer Gestaltung als vollständig anzusehen erinnert an den Dadaismus. Auch hier wurde Kommunikation für den Handlungsgebrauch unfähig gemacht, indem an sich Sinn tragende Elemente in einer Weise miteinander verknüpft wurden, die das emotionale Empfinden ansprachen oder auf die Gestaltung von Schrift, Laut und Semantik verwiesen, aber eben nicht mehr als Werkzeug im eigentlichen oder unmittelbaren Sinne dienen.

Der Gestaltungsspielraum von Grafik Design ergibt sich aus Auffassungen von und Anforderungen an Ästhetik, aber auch daran, Designentscheidungen der anderen Ebenen (insbesondere ab Task-Ebene) deutlich und angemessen zu kommunizieren. Wir werden im 2. Vortrag noch etwas darüber hören.

Ähnlich wie in der Prosodie oder bei den Valeurs der menschlichen Stimme sind die Gestaltungsfreiräume im Hinblick auf die Ästhetik am größten, aber auch am schwierigsten zu kontrollieren. Qualitätsansprüche sind hier tendenziell unbegrenzt.

Zugleich sind die Ausprägungen von Designentscheidungen hinsichtlich der grafischen Repräsentanz zumindest für Ingenieure, aber auch für die Fraktion der Fehlervermeider schlecht zu fassen.

Viele Stakeholder haben Mühe, die Bedeutung von Farb- und Formgestaltung im Rahmen ihrer Disziplin zu akzeptieren. Einen letzten Beweis über deren Funktion bleibt das Grafikdesign vermutlich immer

schuldig. Dies scheint vergleichbar mit der Problematik, die Verwendung eines ungewöhnlichen oder vielleicht sogar unangemessenen Registers bei Gericht juristisch greifbar zu machen.

Um einem Missverständnis vorzubeugen, Graphic Design ist nach diesem Verständnis der beliebige, interaktiv neutrale Gebrauch von Farben und Formen. Wir setzen grafische Mittel selbstverständlich auch dazu ein, Gestaltungsziele der anderen Ebenen zu kommunizieren, darauf wird in dem nächsten Vortrag einzugehen sein.

Tools

Mit den Tools begeben wir uns auf die erste Ebene des Interaktionsdesigns. Interaktionsdesign bedeutet, dass die Software in ihrem (kommunikativen) Gebrauch betrachtet wird.

Hier spreche ich von Werkzeugen im engeren Sinne. Diese Einschränkung erscheint mir notwendig, da man in übertragener Weise zurecht behaupten kann, Software sei ein Werkzeug, mit dem Benutzer ein komplexes, auch geteiltes Handlungsziel erreichen.

Tools werden dem Benutzer einer Software bereitgestellt, um *lokale* Handlungsziele effizient, effektiv etc. zu erreichen. Der kleine Kalenderblock, der beim Buchen einer Bahnfahrkarte über das Internet erscheint ist ein solches Tool, aber auch der Dialog, den Word anbietet, wenn man ein Dokument abspeichern will oder die Schrift verändern möchte, sind solche Tools.

Tools sind in ihrer Handlungswirkung begrenzt, es gibt bei jeder Aufgabe das Zusammenspiel von mehreren Tools. Ein Tool wird in der Software in etwa so herangezogen, wie ein Elektriker einen Schraubenzieher zur Hand nimmt und wieder weglegt, oder ein Chirurg das Skalpell. Türenschlüssel sind solche Tools, Mixer oder eine TV Fernbedienung. Tools stehen jederzeit zur Verfügung, die Abfolge ihrer Verwendung ist schlecht, die Auswahl der Tools pro Aufgabe gut vorhersehbar, und Benutzer legen, wenn sie unterbrochen werden, das Tool normalerweise nicht aus der Hand, bevor sie die damit zu erledigende Teilaufgabe nicht abgeschlossen haben. Tools und ihre Verwendung werden im normalen Handlungszusammenhang kaum wahrgenommen, sie sind in gewisser Weise eine Verlängerung und Spezialisierung der menschlichen Hand.

In der sprachlichen Kommunikation sind Tools möglicherweise auf der Ebene der sog. Satzbaupläne angesiedelt, aber auch Prozeduren nach Ehlich gehören hier her. Nicht alle Sprecher einer Sprachgemeinschaft mögen immer in gleichem Masse über die Fähigkeit verfügen, komplexe Sprachspiele zu beherrschen, s. u. Ich wage aber die These, dass beim Einsatz von sprachlichen Mitteln der Tool-Ebene nur sehr geringe Unterschiede zwischen einzelnen Sprechern bestehen. Menschen

sind mehr oder weniger gut fähig, Ereignisse zu schildern oder komplexe Sachverhalte zu erläutern; vergleichsweise große Qualitätsunterschiede bei der Anwendung deiktischer Prozeduren erwarten wir eigentlich nicht.

Bei der Softwaregestaltung erkennt man den elementaren und universellen Charakter von Tools, bei denen man kaum Abweichungen von Software zu Software kennt. Viele Tools wurden von Apple, Xerox PARC und anderen Pionieren in den 70er und tw. 80 Jahren entwickelt und von Microsoft als Standard durchgesetzt. Bei der Gestaltung unsere Software gibt es nur selten Gründe, diese Tools neu zu erfinden, da sie meist recht ausgereift sind und Benutzer sich an sie gewöhnt haben.

Der Entscheidungsfreiraum von Softwaregestaltung liegt nicht so sehr in der Ausgestaltung der Tools, sondern darin, die Zugriffsmöglichkeiten zu definieren, siehe das folgende Kapitel. Allerdings haben wir auch die Qualität der Bedienoberflächen durch eine geeignete Gestaltung von Tools optimieren können.

Im Hinblick auf die Sprachwissenschaft möchte ich hierbei an das Thema und die Leistungen der klassischen Rhetorik erinnern. Große Redner setzen ihre Mittel (Tools) so ein, dass sie ihr Ziel erreichen, und zwar indem sie Widerstände (oder auch nur Trägheit) des Publikums so bearbeiten, dass sich eine gemeinsame Handlungsfähigkeit herstellt. Gute Software sollte dasselbe erreichen.

Tasks

Es besteht Einigkeit darüber, dass Informationsarchitektur, Grafikdesign und Tools nicht ausreichend sind, um eine brauchbare Software zu gestalten, die nachfolgenden Ebenen sind dem gegenüber Massnahmen, mit denen sich die Qualität verbessern und Vorteile im Wettbewerb erzielen lassen.

Tasks, also Aufgaben, sind das Kernstück beim Interaktionsdesign. Das Reden von Benutzern beinhaltet das Designen für Aufgaben. Im Prinzip wird auch erst dann von Lösungen gesprochen, wenn mit einer Software (oder einem Produkt) Aufgaben gelöst werden können.

In der HCI Literatur wird das als User Centered Design bezeichnet, es hat sich zur Zentralperspektive des Designs entwickelt, zu der sich jeder UI Designer verhalten muss, auch wenn die Formulierung des Anspruchs selbstverständlich noch nicht zur der notwendigen Beherrschung der Technik führt.

Tasks, Aufgaben, sind aus Handlungszielen abgeleitet, es geht darum, die Veränderung eines Zustandes in der Welt herbei zu führen. Aufgaben einer Software sind, wie anderen Handlungsziele auch, verschachtelt und verschränkt. Um zu überleben muss man satt sein, um satt zu werden Nahrung beschaffen,

dazu Obst und Brot einkaufen, das Geld dafür verdienen etc. Um jemanden zu überzeugen muss ich ihm einen Sachverhalt darstellen, um diesen Sachverhalt darstellen zu können die beteiligten Objekte beschreiben, die Zuhörerbereitschaft des Hörers aufrecht erhalten, seine Emotionen gewinnen etc. Um einen Tumor auf dem Röntgenbild dreidimensional sehen zu können, muss ich Röntgenbilder anfertigen, diese aufreihen, den Tumor finden, ihn umschreiben, das umliegende Gewebe herausfiltern, die Übertragung von der Zweidimensionalität zur Dreidimensionalität erreichen etc.

Häufig wird versucht, die gegenwärtigen Handlungsabläufe als Anforderungen aufzufassen, daher sprach ich anfangs von Use Cases als einer Methode der Modellierung. Man fertigt hierarchisch-sequenzielle Handlungsanalysen an und ermittelt den sog. Punkt der Handlung. Es werden die Bedingungen und erwarteten Folgen sowie Varianten ermittelt. Man denkt auch darüber nach, ob der Handelnde wenig Entscheidungsspielraum hat (Supermarktkassen, Bankautomaten, Maskenbedienung, Bahnkarten) – das heisst man kann dem Benutzer eher mit einem Wizzard helfen, der ihn von Schritt zu Schritt führt, oder ob der Benutzer Herr des Handlungsstranges bleiben muss (Bilderauswertung, Anordnen und Verschreiben im Krankenhaus; Schreibprogramme, Excel, Power Point; Parametrierung von Abfüllanlagen, Planung von Schaltkreisen eines Hochhauses, Anpassen einer Software an die Arbeitsbeginnungen eines bestimmten Newsroom einer Fernsehgesellschaft etc.).

Hier ist es, wo die Qualität des Design sich entscheidet, hier werden die meisten Fehler gemacht, hier ist das Risiko am größten. Hier sind Entscheidungen des Designers von besonderem Gewicht. Dies ist tatsächlich der Umschlag von der Anforderung an die Lösung, s. o.

Der Deutschunterricht trainiert die Anfertigung von Bericht, Beschreibung, Erzählung etc. – diese Ebene ist der Task vergleichbar. In den USA kann man das Schreiben von Gedichten und Romanen studieren, jeder Jurist lernt das Argumentieren vor Gericht, Ärzte das Aufklären der Patienten. Diese Ebene ist gemeint, wenn wir von Tasks sprechen.

Inwieweit die Unterstützung bei Aufgaben tatsächlich bereits zur Erwartung der Benutzer gehören, lässt sich diskutieren. Mit einem anderen Bild: Messer und Ofen gehören zur Grundausstattung der Küche, sie sind anerkannte Tools, und niemand wird versuchen, sie als elektive Werkzeuge beim Kochen und Backen zu charakterisieren. Ohne Messer kann man eine Zwiebel nicht zerkleinern, und ohne einen Ofen keinen Kuchen backen. Ob eine Brotschneidemaschine oder gar Brotbackmaschine bereits zum Standard gehört, lässt sich argumentieren, die Grenzen verschieben sich. Notwendig im engeren Sinne sind beide jedenfalls nicht, eine Küchenausstattung, die elektive Werkzeugen integriert, wird vermutlich als attraktiver wahrgenommen.

System

Dieses Qualitätskriterium wird bislang noch wenig beachtet, es behandelt das Gefüge von potenziellen Handlungsalternativen. Wir haben oben durchwegs einen ökonomischen, d. h. minimalistischen Gestaltungsansatz, häufig auch verkürzt und schlagkräftig formuliert als „Form follows function“ oder „Less is more“, vertreten. Benutzer wollen in der Regel beim Handeln nicht die Gesamtheit aller möglichen Handlungswege, sondern das Angebot der hilfreichen, nützlichen oder gar notwendigen Schritte verfügbar haben.

Trotzdem möchten Anwender gerne das Gesamt der Software verstehen. Sie wollen etwa wissen, ob sie 10 oder 90% der möglichen Funktionen kennen gelernt haben. Sie müssen Hypothesen über die Gestaltung wiederkehrender Elemente ausbilden und die zum Teil vielfältigen Entscheidungen im Optimierungsgeschäft nachvollziehen etc. Aus der Gesamtsicht wird häufig klar, warum bestimmte Aufgaben nur suboptimal unterstützt werden können oder Tools nicht immer dann zur Verfügung stehen, wenn sie gebraucht werden. Die Reflektion der Gesamtsicht trägt auch dazu bei, unterschiedliche Anwendungen nicht zu verwechseln.

Zur Verdeutlichung ein Beispiel aus der Architektur. Während Entscheidungen hinsichtlich der Lage der Badezimmer zu den Gästeräumen eines Hotels oder der Küche zum Restaurant lokal und im Hinblick auf „Aufgaben“ getroffen werden, wird der Architekt auch überlegen, ob die Ausgestaltung des Gebäudes als Ganzes stimmig ist. Bewohner, die etwa in einer modernen Klinik herumirren, beklagen, dass sie nach kurzer Zeit keine Vorstellung mehr davon haben, wie die verschiedenen Abteilungen, die sie besucht haben, zueinander angeordnet sind. Dies ist schon verwirrend und tw. ärgerlich in öffentlichen Räumen – die Virtualität von Software verschlimmert das Phänomen des Kontrollverlusts ihrer Benutzer.

Die Sprache stellt eine Fülle an Mitteln, Prozeduren, bereit, das es dem Hörer erlaubt, neues Wissen an bestehendes in qualifizierter Weise anzuknüpfen. Konjunktionen und Präpositionen gehören ebenso hierher wie etwa Einleitungen, Zusammenfassungen und andere Mittel, um Einschübe, Vorgriffe etc. deutlich zu machen.

Alleine die Tatsache, dass ein Vortrag zeitlich begrenzt ist, erlaubt es den Zuhörern, sich eine Vorstellung vom Gesamt zu machen und ihre jeweiligen lokalen Erfahrungen während des Vortrages zu lokalisieren. Wer etwa Edmund Stoiber das erste Mal bei einer Rede zuhört, wird sehr viel schneller ungeduldig als jemand der weiß, dass der Redner Konventionen zu Dauer und Gestaltung regelmäßig frustriert. Es mag dem politischen Geschehen mehr oder weniger zuträglich sein, wenn

die Mehrheit der Menschen ihre Kontrollbedürfnisse delegiert. In der Softwaregestaltung sind wir der Meinung, sollten Benutzer jederzeit wissen, wo sie sich befinden.

Warum wird dies bei der Gestaltung von Software heute noch so wenig beachtet? Ich hatte anfangs erwähnt, wie viele Rollen an der Gestaltung von Software beteiligt sind, darüber hinaus wird selten eine große Software auf einmal geplant und gestaltet. Die Firmensoziologie der Hersteller verhindert oft, dass es einen Stakeholder für die Gesamtqualität gibt.

Tool- und Tasklevel der Softwaregestaltung gehen eindeutig auf Handlungsziele der Benutzer zurück, eine Entsprechung für die Systemqualität zu finden, ist deutlich schwieriger. Man könnte an Gesamtzufriedenheit denken, das ist aber nicht befriedigend. Insofern stellt die Systemsicht einen Bruch des Designprozesses dar, der ansonsten Anforderungen der Benutzer beantwortet. Mit anderen Worten, die Gesamtsystemsicht gehört als Anforderung eher dem Gestalter einer Software, der erreichen möchte, dass Benutzer ein Bild vom Gesamten der Software entwickeln können, erst dann sind sie in der Lage, sich gut zu orientieren und zielgerichtet zu navigieren. Wir erwarten überdies, dass Benutzer, die sich im Gesamt besser orientieren können, toleranter auf suboptimale Gestaltungen von Tasks reagieren, siehe etwa die Clickzählstudie von Jared Spool, der 2001 herausfand, dass Benutzer die Website von Amazon.com als deutlich weniger ‚clicky‘ beurteilten als die von GAP, auch wenn tatsächlich mehr Mausklicks bis zum Abschließen einer Bestellung notwendig waren. Spool vermutet als Grund, dass Amazon.com die momentane Position im Handlungsprozess besser kommuniziert.

Werden wir nicht auch einen kurzen, langweiligen Film als länger empfinden, weil er unsere Erwartungen an die Gesamtgestaltung frustriert, als einen langen, der immer an der richtigen Stelle Spannung produziert?

Hier tragen etwa die sogenannten Bread crumbs im Web design bei, Site Maps sind (zur Zeit leider noch recht schlechte) Lösungsansätze und auch das Bestreben, die einzelnen Applikationen einer Software, also Tasks zwar ähnlich aber doch gegeneinander profilierend zu gestalten, hat hier seinen Raum. Man wird darauf achten, dass ähnliche Aufgaben über die Gesamtsoftware hinweg ähnlich gestaltet sind, dass die jeweilige Navigationstiefe in gleicher Weise aus den verschiedenen Aufgaben abgeleitet wurde etc.

Experience/ Learning

Dieser letzte und komplexeste Layer der Softwaregestaltung nimmt die diachrone Erfahrung des Interaktionsdesigns auf. Benutzer haben eine Geschichte mit dem System. Sie sind am Anfang Novizen, die ein wenig Unterstützung von Mitbenutzern bekommen werden, dann werden sie zu Leuten, die sich gut auskennen und entwickeln möglicherweise das Bedürfnis oder

reagieren auf die Notwendigkeit, sich zum Experten ausbilden zu lassen. Zu Beginn einer jeden Phase muss die Softwaregestaltung sich mit „Acceptance“ auseinandersetzen, danach erweist sich das, was man als „Perceived Usability“ bezeichnet, also ein Vorurteil im Hinblick auf die Gestaltung, und später rundet sich dann die tatsächliche Erlebnisqualität, die „User Experience“.

Eine wirklich gelungene Software beinhaltet also Anteile des Software-Erlebens, die auf die Nutzungsgeschichte des einzelnen Verwenders verweist. Das geht über das Erlernen und die Professionalisierung hinaus.

Aus der Einkaufsforschung weiß man, dass Menschen, nachdem sie einen Gegenstand mit „Selbstpotential“ gekauft haben, diesen auch für dasselbe Geld nicht wieder weggeben würden: Kleider, Handy, Computer. Die Psychologie diskutiert dieses Phänomen als „Verlustaversion“, es gibt aber darüber hinaus eine zeitliche Komponente. Das heißt, die Haltung gegenüber einem Produkt, ändert sich über die Zeit, aber auch vermittelt durch Identifikationsfaktoren. Dasselbe gilt für Software.

Der Wert, von dem ich hier spreche, ist etwa dem vergleichbar, der sich aus der Gestaltung von „Stadtarchitektur“ oder „Erlebnisarchitektur“ (bei Ferienanlagen) ergibt. Ebenso damit vergleichbar ist das „Programm“ eines Buchverlages oder Theaters, sofern das Ansprechen bestimmter Zielgruppen und zugleich auch die Profilierung der Institution angestrebt wird. Sehr ähnlich auch die Entwicklung eines Seminarprogramms über das Semester hinweg, bei der der Professor den Lernzuwachs und das Verhältnis von intrinsischer und extrinsischer Motivation im Auge behalten wird.

Gestaltungsentscheidungen auf dieser Ebene gehören noch ganz und gar dem UI Design, es gibt derzeit keine Anforderungen im Sinne der Prozessdefinition an die verschiedenen Phasen der Interaktion zwischen Benutzer und Software.

Was die Übertragbarkeit zur Linguistik betrifft, so haben wir hier vielleicht ihren Kernbereich verlassen und befinden uns in der Mitte der Literaturwissenschaft, so dass die Software eher einem Roman verglichen werden muss, oder der Rezeptionsgeschichte eines Autors.

Zusammenfassend sei daran erinnert, dass jedes Element einer gestalteten Software sich auf jeder der 6 Ebenen qualitativ verorten lässt.

Ideen

Mehrfach wurde darauf hingewiesen, wo sich eine Nähe zwischen User Interface Design und Sprachwissenschaft zeigen könnte. Hier noch weitere Hinweise.

Warum wird die Forderung, Bedienoberflächen hätten **intuitiv** oder **benutzerfreundlich** zu sein, von Experten des HCI heute nicht mehr ernst genommen? Was ist denn falsch an Intuition und

Nutzerfreundlichkeit? Die Antwort der Psychologen lautet da meist, beide Termine seien paternalistisch und nähmen den Benutzer nicht ernst. Intuition nehme ihn nicht als denkendes Wesen ernst, Freundlichkeit nicht als professionellen Arbeiter. Wie absurd beide Ausdrücke als Qualitätskriterien sind, zeigt sich unmittelbar dann, wenn man etwa fordert, ein Zeitungsartikel habe intuitiv verständlich zu sein oder der Wetterbericht freundlich formuliert. Nicht diese Anforderungen als solche sind falsch, sondern die Beschreibungsebene, auf der sie agieren. Besser wäre es, sich auf eine detaillierte Funktionsanalyse zu verlegen und weit konkretere Qualitätsziele für eine bestimmter Software zu formulieren.

Deutlich oberflächlichere Themen, die ich in meinem Vortrag weiter nicht verorten wollte, da sich ihre Bedeutung für das User Interface Design nahezu von selber ergeben, sind die **Ikonsierung** von Zeichen, der Zusammenhang zwischen Software und **Bedienanleitungen** und etwa das Zusammenspiel aus Softwarebedienung und Mensch-Mensch-Kommunikation beim **kooperativen Bedienen**, die Technik des **Lauten Denkens** beim Usability Testing und die vielfältigen Formen der **Dokumentation** (Spezifikationen, Style Guides), die jeden Entwicklungsprozess begleitet. Hier gibt es eine unendliche Fülle von Aktionsfeldern, auf denen Linguisten sich tummeln könnten, nein: sollten!

Offen

Sprache ist immer konsekutiv, und viele Mittel sind daraus abgeleitet, dies zu bearbeiten. Software, vor allem grafische Software, hat eine simultan-statische und eine konsekutive Komponente. Insofern ist nicht nur die Linguistik, sondern auch die Kunstgeschichte eine Disziplin, die uns dabei hilft, die Qualität und Funktion eingesetzter Mittel zu verstehen. Dazu wird Axel Platz nun noch mehr sagen.

Literatur

Barnum, C., Bevan, N., Cockton, G., Nielsen, J., Spool, J., and Wixon, D., The "Magic Number 5": Is It Enough for Web Testing?" in *CHI 2003 Extended Abstracts, ACM, 698-699, 2003*. [Zur Empirie im UI Design](#)

Carroll, J.M. (ed.), Scenario Based Design: Envisioning Work and Technology in Systems Development, Wiley, 1995. [Berühmter Band, machte die Handlungsperspektive salonfähig, ohne diese allerdings wirklich Ernst zu nehmen](#)

Cockton, G., Value-Centred HCI. NordiCHI '04, October 23-27, 2004 Tampere, Finland, 2004. Überblicksartikel zur Geschichte der HCI, mit programmatischem Ausblick, dabei zu wenig kreationsbetont. Guter Themeneinstieg

Cooper, A. and Reimann, R.M, *About Face 2.0: The Essentials of Interaction Design*, Wiley 2003. Handlungsperspektive aufs Neue.

Go, K., Carroll, J.M., The Blind Men and the Elephant: Views of Scenario-Based System Design. Interactions Nov 2004. Überblick zum Thema Stakeholders

Hassenzahl, M., The Interplay of Beauty, Goodness, and Usability in Interactive Products. HUMAN-COMPUTER INTERACTION, 2004, Volume 19, pp. 319–349, 2004, Lawrence Erlbaum Associates, Inc.

sowie

Hassenzahl, M. Beautiful objects as an extension of the Self. A reply. Human Computer Interaction, 2004, 19(4), 377-386. Praxis und Forschung, Gebrauchstauglichkeit und Attraktivität, Kognition und Emotion.

Nielsen, J. Einer der lautesten Stimmen zum Thema, allerdings nicht mehr ganz auf der Höhe der Zeit. Bitte im Internet suchen. Arbeitet jetzt mit Norman in der nng (nielsen-norman-group zusammen)

Norman, D. A. Emotional Design: Why We Love (or Hate) Everyday Things. New York: Basic Books. 2004 Norman ist einer der HCI-Päpste. Er entdeckt den Common Sense unter den Bergen der psychologischen Literatur. Vorsicht bei Teekannen.

Storrer, A., Text-Bild-Bezüge und Nutzermetaphern im World Wide Web. 2004 Mitteilungen des deutschen Germanistenbunds. Themenheft zu „Sprache und Bild“ (Hg.: Werner Holly & Ulrich Schmitz). Bezieht sich auf den Systemlayer des Interaktionsmodells, nicht so sehr Task oder Tool. Ausserdem interessant zum Thema Metaphern im UI Design.

Wagner, J.: Mensch-Maschine-Kommunikation. Sprachwissenschaftliche Aspekte. Frankfurt/M. etc. 2002. Ermittelt Anknüpfungspunkte von Sprachwissenschaft zu User Interface Design, aber ohne praktische Erfahrung im UID