

# Scribblor

## Ein Interface zur Unterstützung kollaborativer Interaktionsformen für die Arbeit auf gemeinsam genutzten Displays

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Medieninformatik**

eingereicht von

**Clemens Sagmeister**

Matrikelnummer 0525566

**Thomas Nägele**

Matrikelnummer 0303683

an der  
Fakultät für Informatik der Technischen Universität Wien

Betreuung: ao.Univ.Prof. Dr. Peter Purgathofer

Wien, 24.03.2011

\_\_\_\_\_  
(Unterschrift Verfasser)

\_\_\_\_\_  
(Unterschrift Betreuung)



# SCRIBBLER

CLEMENS SAGMEISTER & THOMAS NÄGELE



Ein Interface zur Unterstützung kollaborativer Interaktionsformen für die Arbeit  
auf gemeinsam genutzten Displays

März 2011

Clemens Sagmeister & Thomas Nägele: *Scribbler*, Ein Interface zur Unterstützung kollaborativer Interaktionsformen für die Arbeit auf gemeinsam genutzten Displays, © März 2011

*Zusammenkommen ist ein Beginn, Zusammenbleiben ein Fortschritt,  
Zusammenarbeiten ein Erfolg.*

— Henry Ford

An alle Designer.



## ERKLÄRUNG ZUR VERFASSUNG DER ARBEIT

---

Clemens Sagmeister  
Kanitzgasse 35/2/4,  
1230 Wien

Thomas Nägele  
Lustkandlgasse 39/30,  
1090 Wien

Hiermit erklären wir, dass wir diese Arbeit selbständig verfasst haben, dass wir die verwendeten Quellen und Hilfsmittel vollständig angegeben haben und dass wir die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht haben.

---

Clemens Sagmeister

---

Thomas Nägele

---

(Ort, Datum)



## ABSTRACT

---

When working together on a project, people share their documents, materials and tools. Real life objects allow users to communicate thoughts, ideas and concepts very easily in short time. There has been some effort to recreate a similar experience when using digital media for collaborative work. Unfortunately, existing solutions still do not offer an efficient and transparent interface between real-life and digital artifacts which is the reason why people still prefer conventional methods for collaboration.

The following paper describes possible implementations to use traditional design methods on digital artefacts in theory and in practice. *Scribbler* is one representative of a collaborative system which uses *sketching*, the most important design method, to find this missing link.

## KURZFASSUNG

---

Kollaboratives Arbeiten bedeutet das Teilen von Materialien, Dokumenten und Werkzeugen. Herkömmliche, greifbare Medien schaffen eine Kommunikationsebene zwischen den Personen, auf der es möglich ist, Gedanken, Ideen und Konzepte rasch und zugänglich aufzubereiten. In der Vergangenheit sind schon einige Versuche unternommen worden, diese Art des Arbeitens auf digitalen Medien umzusetzen. So gibt es bereits Ansätze, die die Zusammenarbeit auf einem gemeinsamen großen Display ermöglichen sollen. Jedoch stößt man bei der Arbeit mit diesen kollaborativen Systemen immer wieder auf Barrieren, da es bisher noch nicht gelungen ist, eine effiziente und transparente Schnittstelle zwischen analogen und digitalen Objekten zu schaffen.

Anhand theoretischer und praktischer Ansätze beschreibt die folgende Arbeit die Unterstützung traditioneller Arbeitsweisen von Designern auf digitalen Objekten. Das kollaborative System *Scribbler* setzt auf die wohl wichtigste Designmethode, das *Skizzieren*, und versucht eine barrierefreie Schnittstelle zwischen realen und digitalen Objekten zu schaffen.



*We have seen that computer programming is an art,  
because it applies accumulated knowledge to the world,  
because it requires skill and ingenuity, and especially  
because it produces objects of beauty.*

— Donald E. Knuth (Knuth, 1974)

## DANKSAGUNG

---

Wir möchten uns an dieser Stelle sehr herzlich bei allen bedanken, die uns bei unserem Projekt, den Auswertungen und dem Schreiben der vorliegenden Arbeit unterstützt haben.

Besonderer Dank gilt dabei jenen Personen, die sich bereit erklärten *Scribbler* zu testen. Durch sie fanden wir nicht nur Bestätigung in unserer Arbeit, sondern gewannen auch wichtige Erkenntnisse und Anregungen zur Weiterentwicklung.

Tom und Tom, wir bedanken uns bei euch für eure Unterstützung und Beratung und Peter, vielen vielen Dank für diese tolle Illustration und die Zeit, die du dir für uns genommen hast.

Ein großes Dankeschön gebührt auch unserem Betreuer, Ao. Univ.-Prof. Peter Purgathofer für die Inspiration, Motivation und Unterstützung, die er in dieser Zeit für uns gewesen ist.

— *Danke.*

## SPEZIELLER DANK

---

THOMAS

Mein größter Dank gebührt meiner geliebten Familie, die mir immer Rückenwind gegeben und mich stets motiviert hat. Mum, Dad, Gerry — ohne euch und eure Unterstützung wäre all dies nicht möglich gewesen. Ihr habt mir immer vertraut und mir Selbstbewusstsein geschenkt. Dafür bin ich euch ewig dankbar.

Anna — danke, dass du da bist. Danke, dass du mich begleitest. Danke dass du immer Geduld und Verständnis zeigst und danke, dass du diese Zeit so unglaublich schön für mich machst.

CLEMENS

Ich möchte mich besonders bei meinen Eltern bedanken, die mich nicht nur finanziell unterstützten und mir mein Studium erst ermöglicht haben, sondern mich auch über all die Jahre hinweg motivierten und stärkten. Ihr wart meinen Plänen und Wünschen gegenüber immer offen. — Ich danke euch von ganzem Herzen.

Außerdem möchte ich allen Freunden und Bekannten, mit denen ich das letzte halbe Jahre auf viel gemeinsame Zeit verzichten musste, für ihre moralische Unterstützung danken. — Ihr seid die Besten, vielen Dank an euch alle.

© verfasst von Clemens Sagmeister  
© verfasst von Thomas Nägele  
© verfasst von Clemens Sagmeister & Thomas Nägele

# INHALTSVERZEICHNIS

---

EINLEITUNG	1	
<b>I THEORIE</b>	<b>3</b>	
<b>1 BESTEHENDE SYSTEME</b>	<b>5</b>	(cs) (tn)
1.1 Arbeitsweisen von Designteams	5	
1.2 Skizziertools in kooperativen Settings	8	
1.2.1 XSketch	8	
1.2.2 We-Met	11	
1.2.3 Group Drawing Tools	15	
1.2.4 Screencrayons	20	
1.2.5 Ein durch Tablets unterstütztes Meetingtool	26	
1.2.6 Team Storm	31	
1.3 CSCW und Groupware Tools	34	
1.3.1 Roomware	35	
1.3.2 Weitere Konzepte zur Kooperation im selben Raum	37	
<b>2 DESIGNTHEORIE</b>	<b>43</b>	(cs)
2.1 Was ist Design?	43	
2.1.1 Design als Prozess	44	
2.1.2 Design als Repräsentation	46	
2.1.3 Designdisziplinen	46	
2.2 Designmethoden	49	
2.2.1 Artefakte	50	
2.2.2 Skizzieren	52	
2.2.3 Prototyping	60	
2.2.4 (User) Testing	67	
2.2.5 Personas & Szenarien	71	
2.3 Spezielle Eigenschaften	74	
2.3.1 Designvokabular	74	
2.3.2 Der kollaborative Charakter	76	
<b>3 KOLLABORATIVES DESIGN</b>	<b>79</b>	(tn)
3.1 Soziale Aspekte & Probleme	80	
3.2 Analyse eines kollaborativen Designmeetings	86	
3.2.1 Rollen und Beziehungen	86	
3.2.2 Planung und Tätigkeiten	89	
3.2.3 Sammeln und Teilen von Informationen	94	
3.2.4 Analyse und Verständnis des Problems	97	
3.2.5 Generierung & Adaptierung von Konzepten	98	
3.2.6 Vermeidung und Lösung von Konflikten	100	
3.2.7 Schlussfolgerungen aus der Studie	102	
<b>4 DESIGN VS COMPUTER</b>	<b>105</b>	(cs)
4.1 Skizzieren auf Papier	105	

	4.2	Skizzieren am Computer . . . . .	108
	4.2.1	Electronic Ink . . . . .	108
	4.2.2	Der Unterschied zwischen Eingabestift und Maus . . . . .	109
	4.2.3	Interaktionstechniken für stiftbasierte Skizziersysteme . . . . .	111
	4.2.4	Das Modusproblem . . . . .	116
	4.3	Die Bedeutung von Gestik . . . . .	119
	4.4	Case Study - Digital and Paper Media . . . . .	124
	4.4.1	Kognitive Merkmale . . . . .	126
	4.4.2	Soziale Merkmale . . . . .	127
(tn)	5	CSCW & GROUPWARE . . . . .	129
	5.1	Klassifikation von CSCW . . . . .	132
	5.1.1	Asynchrone Systeme . . . . .	134
	5.1.2	Synchrone Systeme . . . . .	135
	5.1.3	Multisynchrone Systeme . . . . .	135
	5.2	Design von Groupware . . . . .	136
	5.3	Problemfelder . . . . .	143
	II	PRAXIS . . . . .	153
(cs)(tn)	6	SCRIBBLER . . . . .	155
	6.1	Ausgangssituation & Rahmenbedingungen . . . . .	155
	6.2	Anforderungen an kollaborative Skizziersysteme . . . . .	156
	6.3	Vorgehensweise . . . . .	159
	6.3.1	Recherche . . . . .	160
	6.3.2	Skizzieren . . . . .	161
	6.3.3	Prototyping . . . . .	161
	6.3.4	(User) Testing . . . . .	163
	6.4	Scribbler at a Glance . . . . .	165
	6.4.1	Hardware . . . . .	166
	6.4.2	Grafisches User Interface . . . . .	168
	6.4.3	Programmlogik . . . . .	170
	6.5	User Review . . . . .	175
	6.5.1	Positives Feedback . . . . .	176
	6.5.2	Kritisches Feedback . . . . .	180
	6.6	Achievements . . . . .	188
		EPILOG . . . . .	193
	III	APPENDIX . . . . .	195
(cs)(tn)	A	GESPRÄCHE MIT DESIGNERN . . . . .	197
	A.1	Interview mit einem Produktdesigner . . . . .	197
	A.2	Interview mit zwei User Experience Designern . . . . .	207
		LITERATURVERZEICHNIS . . . . .	219
		INDEX . . . . .	235

## ABBILDUNGSVERZEICHNIS

---

Abbildung 1.1	Framework zur Untersuchung von Designaktivitäten (Tang and Leifer, 1988) . . . . .	7
Abbildung 1.2	Auszug des Transkripts inklusive dazugehöriges angefertigtes Artefakt der ersten Designsession (Tang and Leifer, 1988) . . . . .	7
Abbildung 1.3	Statistische Ergebnisse der ersten Designsession (Tang and Leifer, 1988) . . . . .	7
Abbildung 1.4	GroupSketch (Greenberg et al., 1992b) . . . . .	17
Abbildung 1.5	GroupDraw (Greenberg et al., 1992b) . . . . .	19
Abbildung 1.6	Crayon Highlighting (Olsen et al., 2004) . . . . .	25
Abbildung 1.7	Finding Regions for Circles/Scribbles (Olsen et al., 2004) . . . . .	25
Abbildung 1.8	Drei verschiedene Repräsentationen eines Annotationsbeispiels (Olsen et al., 2004) . . . . .	25
Abbildung 1.9	Das kollaborative Skizziersystem (Bastéa-Forte and Yen, 2007) . . . . .	26
Abbildung 1.10	Die unterschiedlichen Arbeitszustände im Vergleich (Bastéa-Forte and Yen, 2007) . . . . .	28
Abbildung 1.11	Logfile Analyse (Bastéa-Forte and Yen, 2007) . . . . .	28
Abbildung 1.12	Repräsentationen der Strichstatistik (Bastéa-Forte and Yen, 2007) . . . . .	30
Abbildung 1.13	Gemeinsame und private Arbeitsfläche in <i>Team Storm</i> (Hailpern et al., 2007) . . . . .	33
Abbildung 1.14	Ausarbeitung von Konzepten (Hailpern et al., 2007) . . . . .	34
Abbildung 1.15	Designkonzepte für Systeme zur kreativen Zusammenarbeit (Wang and Blevis, 2004) . . . . .	39
Abbildung 2.1	Der Produktentwicklungsprozess (Buxton, 2007) . . . . .	45
Abbildung 2.2	Visualisierung des User Interfaces (Sagmeister, 2008) . . . . .	48

Abbildung 2.3	Verwandte Disziplinen von Interface- und Interaction Design (Saffer, 2007) . . . . .	48
Abbildung 2.4	Die Skizzierwechselbeziehung. (Buxton, 2007) . . . . .	53
Abbildung 2.5	Physikalische Modelle. (Vyas et al., 2008) . . . . .	57
Abbildung 2.6	Skizzen und Prototypen im Designprozess. (Sagmeister, 2008) . . . . .	58
Abbildung 2.7	Übergang von Skizzen zu Prototypen. (Buxton, 2007) . . . . .	59
Abbildung 2.8	Getting the Right Design vs. Getting the Design Right. (Greenberg and Buxton, 2008) . . . . .	59
Abbildung 2.9	Digitale Prototypen. (Saffer, 2007) . . . . .	63
Abbildung 2.10	Physikalische Prototypen. (Saffer, 2007) . . . . .	63
Abbildung 2.11	Hintergrundmasken (Snyder, 2003) . . . . .	66
Abbildung 2.12	Paper Prototyping (Sagmeister, 2008) . . . . .	66
Abbildung 2.13	(User) Testing (Saffer, 2007) . . . . .	68
Abbildung 2.14	Wizard of Oz (Buxton, 2007) . . . . .	69
Abbildung 2.15	Persona (Saffer, 2007) . . . . .	72
Abbildung 3.1	Tree Swing Sketch (Kilker, 1999) . . . . .	81
Abbildung 4.1	Darstellungsformen (Johnson et al., 2009) . . . . .	106
Abbildung 4.2	Kräfte bei Buttonklicks. (Johnson et al., 2009) . . . . .	110
Abbildung 4.3	Tortenmenüs (Johnson et al., 2009) . . . . .	112
Abbildung 4.4	Gedric (Geißler, 1995) . . . . .	113
Abbildung 4.5	CrossY (Johnson et al., 2009) . . . . .	114
Abbildung 4.6	I Love Sketch (Johnson et al., 2009) . . . . .	115
Abbildung 4.7	Sketchpad (Johnson et al., 2009) . . . . .	117

Abbildung 4.8	GRAIL Funktionalität (Johnson et al., 2009) . . . . .	117
Abbildung 4.9	Modiwechsel (Johnson et al., 2009) . . . . .	119
Abbildung 4.10	Beispiel von Gestik in einem Designmee- ting. (Le Dantec, 2009) . . . . .	123
Abbildung 4.11	VideoDraw (Tang and Minneman, 1991) . . . . .	123
Abbildung 4.12	Designstudie über die Benutzung von No- tizbüchern (Hinckley et al., 2010) . . . . .	125
Abbildung 5.1	CSCW-Kategorien (Rama and Bishop, 2006) . . . . .	133
Abbildung 5.2	»Payoff« durch Nutzung zweier Textver- arbeitungsmaschinen (Markus and Connolly, 1990) . . . . .	146
Abbildung 5.3	»Payoff« durch Nutzung eines automati- schen Meeting Planers (Markus and Connolly, 1990) . . . . .	148
Abbildung 5.4	Die kritische Masse (Markus and Connolly, 1990) . . . . .	150
Abbildung 6.1	Anforderungen eines kollaborativen Skiz- zersystems . . . . .	157
Abbildung 6.2	Aufzeichnung zur frühesten Designfrage .	160
Abbildung 6.3	Skizzierte Szenarien in Scribbler . . . . .	162
Abbildung 6.4	Detailskizze eines technischen Ablaufs . .	162
Abbildung 6.5	Mockups in Scribbler . . . . .	164
Abbildung 6.6	Scribbler testing in einer Designsession . .	165
Abbildung 6.7	Tablet Tastenbelegung . . . . .	167
Abbildung 6.8	Tabletstiftfarben . . . . .	169
Abbildung 6.9	Proximity Event . . . . .	169
Abbildung 6.10	Scribbler Menu Icon . . . . .	171
Abbildung 6.11	Skizzieren in Scribbler . . . . .	171
Abbildung 6.12	Accessibility Daten zur aktuellen Maus- position . . . . .	174
Abbildung 6.13	Anforderungsauswertung für Scribbler . .	189

## TABELLENVERZEICHNIS

---

Tabelle 3.1	Polarisierende Designideale (Kilker, 1999) . . . . .	83
-------------	---	----

Tabelle 4.1	Elektronische Dokumente und Papierdokumente (Wellner, 1993) . . . . .	106
Tabelle 5.1	Fragen der Awareness (Gutwin and Greenberg, 1999) . . . . .	139
Tabelle 5.2	Tangs Designkriterien (Tang, 1989) . . . . .	141

## LISTINGS

---

Listing 6.1	Global and Local Event Monitoring . . . . .	172
Listing 6.2	Key Window Handling . . . . .	175
Listing 6.3	Laden von Accessibility Daten . . . . .	176

## AKRONYME

---

CSCW	Computer Supported Cooperative Work
HCI	Human-Computer Interaction
PDF	Portable Document Format
HTML	Hypertext Markup Language
WIMP	Window-Icon-Menu-Pointing Device
GUI	Graphical User Interface
SDG	Single Display Groupware
USB	Universal Serial Bus
CAD	Computer-Aided Design

## AUSZÜGE

---

Auszug 3.1	Vorschlag von Kerry . . . . .	86
Auszug 3.2	John will die Problematik klären . . . . .	87
Auszug 3.3	Ein Hinweis . . . . .	87
Auszug 3.4	Das Anliegen wird ignoriert . . . . .	87

Auszug 3.5	Kerry wird erhört . . . . .	87
Auszug 3.6	Der zeitliche Ablauf . . . . .	88
Auszug 3.7	Bestätigung für Ivan . . . . .	88
Auszug 3.8	Planung zuerst . . . . .	89
Auszug 3.9	Es ist bereits spät . . . . .	90
Auszug 3.10	Zeitdruck . . . . .	90
Auszug 3.11	Sollen wir bereits voranschreiten? . . . . .	91
Auszug 3.12	Nochmals alles abklären . . . . .	91
Auszug 3.13	Der Prototyp . . . . .	91
Auszug 3.14	Kerry teilt ihre Vorstellung mit . . . . .	92
Auszug 3.15	Maß nehmen . . . . .	92
Auszug 3.16	Die Rechtfertigung . . . . .	92
Auszug 3.17	Unterbrechung . . . . .	93
Auszug 3.18	Wieder vereint . . . . .	93
Auszug 3.19	Inspektion . . . . .	94
Auszug 3.20	Wieder zurück . . . . .	94
Auszug 3.21	Der Zielpreis . . . . .	95
Auszug 3.22	Der Report . . . . .	95
Auszug 3.23	Missverständnisse . . . . .	96
Auszug 3.24	Bereits vergessen . . . . .	97
Auszug 3.25	Was erwartet der Benutzer? . . . . .	98
Auszug 3.26	Ausformulieren . . . . .	99
Auszug 3.27	Eine Idee . . . . .	99
Auszug 3.28	Beharren . . . . .	99
Auszug 3.29	Emotionale Bindung . . . . .	100
Auszug 3.30	Auflockerung . . . . .	100
Auszug 3.31	Drängen . . . . .	100
Auszug 3.32	Uneinigkeit . . . . .	101
Auszug 3.33	Zwei gegen Einen . . . . .	102
Auszug 4.1	Gestik in Designmeetings. . . . .	122
Auszug 4.2	Arbeiten an digitalen Tabletop-Systemen. . . . .	127
Auszug 6.1	Traumsituation . . . . .	177
Auszug 6.2	Einsatz im Unterricht . . . . .	177
Auszug 6.3	Farben . . . . .	178
Auszug 6.4	Zugehörigkeit . . . . .	179
Auszug 6.5	Kommunikation . . . . .	179
Auszug 6.6	Graphic-Recording . . . . .	180
Auszug 6.7	Fehlende Funktionalität . . . . .	182
Auszug 6.8	Transport und Kosten . . . . .	183
Auszug 6.9	Nutzen . . . . .	184
Auszug 6.10	Nutzbar machen . . . . .	184
Auszug 6.11	Radieren . . . . .	185
Auszug 6.12	Verschieben . . . . .	186
Auszug 6.13	Trackpad . . . . .	186
Auszug 6.14	Inaktive Fenster . . . . .	187



## EINLEITUNG

---

In den letzten Jahrzehnten wurde viel Forschung betrieben, um die Unterstützung kollaborativer Arbeitssituationen durch technologische Mittel voran zu treiben. Vor Allem im Web stieg die Zahl der Tools zur entfernten Zusammenarbeit. Gleichzeitig musste man aber feststellen, dass die bequeme Onlinekollaboration auch Nachteile mit sich brachte. Simple, jedoch wichtige Kommunikationsformen, wie Augenkontakt, Gestik und Mimik fehlten. Aus diesem Grund können kollaborative Meetings bis dato immer noch nicht vollständig ersetzt werden. Besonders Designer verbringen viel Zeit in Meetings, um zusammen mit anderen Ideen zu erarbeiten. Ihr wichtigstes Werkzeug sind dabei Stift und Papier, mit deren Hilfe sie Notizen und einfache Skizzen anfertigen.

Da aber auch hier die Technisierung nicht halt machte, und Designer vermehrt mit digitalen Medien zu tun haben, kommt es oft zu ungewolltem Mehraufwand. Designs die nur elektronisch existieren, können zwar meist in einer geeigneten Form (z.B. mittels Projektor) präsentiert, jedoch Anmerkungen, Notizen oder neue Ideen nie direkt auf das digitale Design geschrieben oder gezeichnet werden.

Es existieren zwar bereits Systeme, die das kollaborative Skizzieren auf einem gemeinsamen Display ermöglichen, jedoch werden Skizzen an das jeweilige System gebunden, wodurch es unmöglich ist, auf beliebigen Inhalten zu arbeiten.

Die vorliegende Arbeit erläutert die Problemfelder dieser Situation und zeigt anhand des Projekts *Scribbler* einen möglichen Lösungsansatz. In den folgenden Kapiteln werden zunächst in [Kapitel 1](#) bestehende Systeme beschrieben und im Zuge dessen, wichtige Anforderungen an ein solches System gesammelt. [Kapitel 2](#) und [Kapitel 3](#) erklären Arbeitsweisen von Designteams mittels Methoden und Verhaltensweisen in kollaborativen Settings. In [Kapitel 4](#) wird der Unterschied zwischen dem Arbeiten mit traditionellen und digitalen Medien spezifiziert und das anschließende Kapitel [Kapitel 5](#), beschäftigt sich mit der Definition von Computer Supported Cooperative Work (CSCW) und Groupware und betrachtet Eigenheiten und Implikationen. Die so gewonnenen Erkenntnisse werden dann im zweiten Teil der Arbeit, [Kapitel 6](#) veranschaulicht und ein Bezug zur Praxis hergestellt.

*Anmerkung: Über die gesamte Arbeit werden Anmerkungen, seitlich durch das ♣-Symbol gekennzeichnet. Kommentare werden ebenfalls seitlich platziert.*



*Kommentar*



Teil I

THEORIE



## BESTEHENDE SYSTEME

---

In der Arbeitswelt kommt es immer wieder zu Situationen, in der sich mehrere Personen treffen, um zusammen Probleme zu diskutieren. Statistiken besagen, dass Büroangestellte sogar bis zu 70 Prozent ihrer Arbeitszeit in Meetings verbringen (Panko, 1993). Die meisten Meetings werden mit Hilfsmitteln wie Whiteboards, Flipcharts oder Notizblöcken abgehalten, da meistens Notierungsbedarf bei neuen Ideen, Problemlösungen etc. besteht. Computer können dabei sehr hilfreich sein, wäre da nicht die Hürde der Dateneingabe die zu bewältigen ist. Herkömmliche Eingaben via Maus und Tastatur verlangsamten den natürlichen, explorativen Charakter eines Meetings. Elektronische Skizzierwerkzeuge, inklusive gemeinsam verwendbarer Zeichenoberflächen, können hier Abhilfe für kollaboratives Arbeiten schaffen.

Das folgende Kapitel soll anhand von Beispielen und Projekten aus der Literatur zeigen, durch welche Methoden Meetings bzw. im speziellen Designsessions unterstützt werden können. Besonderes Augenmerk soll dabei auf das Skizzieren als Designmethode und kooperatives Arbeiten in CSCW Umgebungen liegen.

### 1.1 ARBEITSWEISEN VON DESIGNTEAMS

Um herauszufinden, mit welchen Methoden oder Werkzeugen man kollaborative Meetings und im Zuge dessen den Designprozess unterstützen kann, muss man zuerst wissen wie Designer zusammen arbeiten. John Tang und Larry Leifer versuchten in (Tang and Leifer, 1988) als Teil einer Studie mittels empirischer Daten, das Verhalten von Designern in Meetings zu untersuchen. Dazu bildeten sie ein Framework und konzentrierten sich auf deren Arbeitsaktivitäten wie *Auflisten*, *Zeichnen* und *Gestikulieren*, welche sie dem zugehörigen Kontext zuordneten. Um diesen einzuschränken, bildeten sie vier Funktionen, die den Hauptzweck der Aktivitäten beschreiben: *Festhalten von Informationen*, *Vermitteln von Ideen*, *Darstellen von Ideen* und *Erlangen von Aufmerksamkeit*. In einem Framework (siehe [Abbildung 1.1](#)) stellten sie diese gegenüber und verwendeten die Daten zur Auswertung.

Obwohl sich Tang und Leifer mit ihrer Arbeit gezielt auf den Designprozess zubewegen, erwarten sie dadurch auch relevante Ergebnisse für kollaboratives Arbeiten im Allgemeinen.

Kollaboratives Arbeiten ist für Designer eine gängige Art Probleme anzugehen und kann von den sonstigen individuellen Aktivitäten deutlich abgegrenzt werden, wie auch frühere empirische Designstudien bestätigen können: (Akin, 1978; Lera, 1983; Ullman et al., 1987; Ballay, 1987).

In ihrer Studie führten sie 8 Sessions mit Teams zu je 3-4 Personen durch, welche jeweils eineinhalb Stunden dauerten. Die Gruppen trafen sich dafür in Konferenzräumen und saßen um einen Tisch mit einem großen Bogen Papier als Schreibunterlage. Jede Gruppe arbeitete zwar an einer anderen Designaufgabe, jedoch waren alle Teil des Designs eines Mensch-Maschinen Interfaces für ein »smartes« (Mikroprozessor gesteuertes) Gerät. Die Analyse der Tests wurde durch das *NoteCards Hypertext Environment* (Halasz et al., 1987) erheblich vereinfacht. Das System half, die Daten zu strukturieren und aufzubereiten. [Abbildung 1.2](#) zeigt einen Auszug des Transkripts der ersten Session mit Anmerkungen zu einzelnen Aktivitäten und einen Teil der Schreibunterlage, welche die dazu erstellte Skizze beinhaltet. Jede Aktivität wurde anhand der Funktionen des Frameworks kategorisiert und abgezählt, um in Folge dessen die Daten statistisch auszuwerten. [Abbildung 1.3](#) veranschaulicht die Ergebnisse.

Wie die Abbildung zeigt, sind knapp die Hälfte (46%) aller Aktivitäten von illustrativem Charakter, was die Wichtigkeit von Skizzen belegt. Zusätzlich kommt der Vermittlung und Darstellung von Ideen (zusammen 43%) auch große Bedeutung zu.

- ♣ *Anmerkung: An diesem Punkt der Arbeit soll noch nicht auf die gesamten Erkenntnisse der Analyse eingegangen werden, sondern lediglich auf die der Ideenentwicklung. Weitere Ergebnisse werden im [Abschnitt 4.3](#) des Kapitels **DESIGN VS COMPUTER** behandelt.*

Die Schreibunterlage spielt im Prozess der Ideenentwicklung neben der des Festhaltens von Informationen und der Vermittlung von Ideen eine aktive Rolle. Die Unterstützung dieses Prozesses (z.B. durch elektronische Hilfsmittel) ist aber alles andere als unproblematisch, wie beispielsweise die Frustrationen von Designern beweisen, die versuchen, konventionelle Computer-Aided Design (CAD) Tools für die Weiterentwicklung von Ideen - von konzeptionellen Skizzen zu formellen Spezifikationen - zu verwenden. Dieser Aspekt zeigt die Verwendung der Schreibunterlage als Teil eines interaktiven Prozesses, anstatt die Reduzierung dessen auf ein Text-Grafik Artefakt.

Die Studie beobachtete zwei Schlüsselmerkmale, auf die Designer bei der Entwicklung von Ideen zurückgriffen: a) In der Lage sein ohne weiteres Darstellungen von Ideen auf einer Schreibunterlage auszuprobieren, und b) diese Darstellungen stufenweise in genaue Artefakte weiterzuverarbeiten - oft durch die Zusammenarbeit mit anderen Partizipanten. (Tang and Leifer, 1988)

		TALK		
Function \ Action		LIST	DRAW	GESTURE
Store information				
Convey ideas				
Represent ideas				
Engage attention				

Abbildung 1.1: Das Framework zur Untersuchung von Designaktivitäten. In ihm werden die Aktivitäten den zweckmäßigen Funktionen gegenübergestellt. Es dient als empirische Grundlage.

(a) Transkriptauszug

(b) Artefakt

Abbildung 1.2: Ein Auszug des Transkripts, erstellt mit Hilfe von NoteCards (Halasz et al., 1987), inklusive dazugehörigen Artefakt der ersten Designsession. Es zeigt wie jeder Teilnehmer in kürzester Zeit einen eigenen Gedanken äußert und diesen im Artefakt manifestiert.

Function \ Action	LIST	DRAW	GESTURE	Total
Store information	40	19	1	60 (27%)
Convey ideas	0	22 (36%)	24	46 (20%)
Represent ideas	2	41	9	52 (23%)
Engage attention	0	21	46	67 (30%)
Total	42 (19%)	103 (46%)	80 (35%)	225

Abbildung 1.3: Statistische Ergebnisse der ersten Designsession. Alle Aktivitäten wurden dafür im Framework anhand der Funktionen kategorisiert und anschließend abgezählt.

Gemeinsam verwendbare Zeichenoberflächen spielen auch laut Sara Bly (Bly, 1988) eine besonders wichtige Rolle in Designsessions mit mehreren Teilnehmern. Bly beobachtete das Skizzierverhalten zweier Personen in Face-to-Face Designsessions und zeichnete diese für spätere Analysen auf Video auf. Bei der Auswertung stellte sie fest, dass nahezu die Hälfte aller Aktivitäten, in denen Gebrauch von Schreibwerkzeugen gemacht wurde, Skizzieren war. Noch interessanter war aber die Tatsache, dass sich im Laufe der Session verschiedene Zeichenbereiche herauskristallisierten. So entstanden einzelne Bereiche, in denen nur ein Designer gezeichnet hatte und Bereiche, die zusammen benutzt wurden. 78% aller Aktivitäten fanden jedoch im gemeinsamen Bereich statt und 25% aller Aktivitäten eines Designers entstanden im Anschluss zu einer Zeichnung eines anderen Designers. Die Analyse zeigt also die Wichtigkeit von gemeinsamen Bereichen in Designsessions mit mehreren Teilnehmern. Zusätzlich spielten Gesten und Betonungen in Erläuterungen eine wichtige Rolle, welche die Notwendigkeit der physischen Anwesenheit nahelegt.

## 1.2 SKIZZIERTOOLS IN KOOPERATIVEN SETTINGS

Tools und Umgebungen herkömmlicher Brainstorming-Meetings besitzen zwar physikalische Qualitäten, unterstützen jedoch nur unzureichend gemeinsames Zeichnen mehrerer Personen. Gleichzeitiges Arbeiten an einer Skizze, z.B. am Whiteboard, ist heikel, da Personen nahe beisammen stehen müssen und sich gegenseitig oft im Weg sind. Es ist physikalisch unmöglich, an der exakt selben Stelle der Tafel zu arbeiten. Tang spricht hier von einem eindeutigen Bedarf an einem gleichzeitigen Zugriff auf die Arbeitsfläche in kollaborativen Gruppenmeetings (Tang, 1991).

Über die Jahre wurden einige Versuche gestartet, diesem Wunsch mit Hilfe von elektronischen Mitteln nachzukommen. Im folgenden werden nun einige davon beschreiben.

### 1.2.1 XSketch

Bereits 1990 nahm sich das Department of Computer Science an der Universität in Toronto Tangs und Blys Arbeiten als Ansporn, bzw. Grundlage für weitere Arbeiten und entwickelte *XSketch*: ein multi-user Skizzierwerkzeug für X11<sup>1</sup>, das Jeffrey J. Lee in (Lee, 1990) vorstellt.

Die Grundidee von *XSketch* ist, ein simples Werkzeug für multi-user Design- und Brainstormingsessions bereit zu stellen. Da es aber auch für einzelne Benutzer geeignet sein soll, hat der Multi-

<sup>1</sup> X11 ist ein Softwaresystem für UNIX Plattformen, das ein Graphical User Interface (GUI) zur Verfügung stellt - auch bekannt als X-Windows.

user-Faktor nur wenig Einfluss auf das User Interface. Es setzt auf eine Server-Client Architektur, in der alle Nachrichten und Entscheidungen zentral gesteuert werden und Benutzer via TCP/IP Protokoll kommunizieren. Das Zeichenmodell kann man mit einem Notizblock oder Flipchart vergleichen, in dem Benutzer auf leeren Seiten zeichnen und zwischen mehreren Seiten hin und her »blättern« können. Zeichnungen basieren auf einem objektorientierten Ansatz, sodass man leicht ganze Objekte ausschneiden und kopieren kann. Den Benutzern stehen Polylinien<sup>2</sup>, Rechtecke und simpler Text als Objekte zur Verfügung. Die Interaktion basiert dabei hauptsächlich auf Mauseingaben, die Tastatur wird lediglich zur Eingabe und Änderung von Texten und Dateinamen verwendet. Die meisten Editieroperationen sind ebenfalls an die Mausbuttons gebunden. Mit dem Vorsatz, das Interface so schlicht wie möglich zu halten, existieren relativ wenige Modi und keine Möglichkeit, Attribute wie Linienart, -stärke, Pfeilspitzen etc. zu ändern. Dafür ist es möglich, einen Telepointer durch Druck auf eine bestimmte Maustaste einzublenden, um so das Gezeichnete für die anderen Benutzer hervorheben zu können. Diese Funktion soll als Ersatz für Gesten dienen. Die einzige Multi-user-Funktion ist der »Invite Button«, der dazu benutzt werden kann, anderen Benutzern Nachrichten zu schicken und sie zu einer Session einzuladen. Nehmen diese teil, können sie alle Objekte ändern, nicht nur die eigenen. Ansonsten unterscheidet sich eine multi-user Session kaum von einer single-user Session.

Obwohl der Prototyp von *XSketch* allen anfänglichen Zielen und Anforderungen der Projektgruppe entspricht, gibt es nach Lees Erfahrungen in einigen Fällen Verbesserungspotential. So ist z.B. das Zeichnen von Skizzen unbefriedigend. Ein möglicher Lösungsansatz für dieses Problem wäre, neue grafische Objekte wie Ellipsen oder Splines zur Objektauswahl hinzuzufügen und das Einfügen von Bitmapbildern zu erlauben, um so das Repertoire an Tools aufzustocken. Im Gegensatz dazu könnte man das Repertoire aber auch verringern und durch und durch auf reine Freihandzeichnungen setzen. Welcher der bessere Ansatz wäre, müsse ausprobiert werden.

Ein weiterer verbesserungsbedürftiger Punkt ist das strikte Cut&Paste Modell. Obwohl sich das Entwicklungsteam anfangs eigentlich bewusst dafür entschieden hat, um Kollisionen im Datenmanagement vorzubeugen, steht es dadurch vor Usabilityproblemen. Durch die Tatsache, dass man existierende Objekte nicht nachträglich ändern kann, entsteht Frust bei den Benutzern.

*Ein gutes multi-user Skizzierprogramm benötigt eine große Auswahl an Tools oder sollte sich durch und durch auf Freihandzeichnungen beschränken.*

<sup>2</sup> Polylinien sind eine Aneinanderreihung von Linien. In der Computergrafik werden sie zur Annäherung an Kurven benutzt.

*Nachträgliches  
Ändern von Skizzen  
und eine  
dazugehörige Undo-  
Funktion sind  
wünschenswert.*

*Alle Teilnehmer  
müssen stets über  
die Tätigkeiten der  
anderen im Bilde  
sein.*

*Ein Grafiktablet in  
Verbindung mit  
einem Flatscreen  
wäre ein ideales  
Setting für ein  
multi-user  
Skizziertool.*

Ein optimistisches Sperrmodell<sup>3</sup> könnte, im Gegensatz zu konventionellen zentralisierten (pessimistischen) Sperrmodellen, das Selektieren und nachträgliche Ändern von Objekten ohne lästige Verzögerung ermöglichen. Erstrebenswert wäre diesbezüglich auch eine dazugehörige Undo Funktion, welche die derzeitige Pseudo-Undo Funktion mittels Löschen ersetzt.

Wie auch schon Bly feststellte, ist in einer Designsession mit mehreren Teilnehmern der Prozess des Zeichnens für den Designprozess genau so wichtig wie die Zeichnungen selbst (Bly, 1988). Aus diesem Grund ist es wichtig, dass alle Teilnehmer sehen können, was die anderen machen. Leider können Benutzer von *XSketch* nicht sehen, wenn andere Benutzer Objekte erstellen oder selektieren, da der Vorgang bis zu seinem Abschluss rein lokal abläuft. Diese Entscheidung ist laut Lee schlichtweg ein Fehler und muss unbedingt ausgebessert werden. Ebenso das Ausbleiben einer Möglichkeit, untereinander kommunizieren zu können, sieht er als Fehlgriff, da es im geplanten Setting oft vorkommt, dass viele User in einem großen Raum verteilt sitzen und kein Telefon bei sich haben, um erreichbar zu sein. Hier wäre eine integrierte Nachrichtenfunktion im Programm denkbar, wie sie z.B. auch in *MUSK* (Crampton, 1987) umgesetzt wurde. Das würde den Benutzern zusätzlich helfen, eine Übersicht über alle Teilnehmer zu bekommen.

Schließlich, resümiert Lee, war es rückblickend auch falsch anzunehmen, dass sich die Anforderungen von multi-user Systemen von single-user Systemen nicht unterscheiden. Der Prototyp wurde in simplen Designsessions mit einem, zwei und drei Benutzer(n) getestet und zeigte zwar über eine 19,2Kbaud Verbindung eine gute Performance, jedoch beschränken die oben genannten Probleme den Einsatz noch auf Institutsexperimente. Unglücklicherweise zeigten die Tests auch, dass ein mausbasiertes System für interaktive Skizziersessions nicht ideal ist. Ein ideales Interface für diese Art von Programm wäre laut Lee ein gut integriertes Grafiktablet in Verbindung mit einem Flatscreen. Das wäre die bestmögliche Annäherung zu einer Technologie, die Stift und Papier ersetzen könnte.

<sup>3</sup> Sperrmodelle in der elektronischen Datenverarbeitung, sind Mechanismen die Datenbankressourcen sperren, um zu verhindern, dass mehrere Datenzugriffe gleichzeitig stattfinden können. Man unterscheidet zwischen optimistischen und pessimistischen Sperrstrategien. Eine pessimistische Sperrstrategie schützt die Integrität der Datenbank, indem Datenbankressourcen während der gesamten Dauer einer Transaktion von Anfang bis Ende gesperrt werden. Eine optimistische Sperrstrategie hingegen senkt die Isolationsstufe, sodass weniger Sperren auf die Datenbankressourcen angewandt werden. Auf diese Weise können mehr Anwendungen gleichzeitig auf eine Datenbank zugreifen, was den Durchsatz potenziell erhöhen kann. (IBM Corporation, 1996)

### 1.2.2 *We-Met*

Dem Gedanken treu geblieben, entwickelten Wolf et al. zwei Jahre später mit *We-Met* (Wolf et al., 1997) einen Prototypen eines Meeting Support Tools. Ihr System umfasste nun Grafiktablets anstatt Mäuse für jeden Teilnehmer. Das stiftbasierte Interface sollte bei Face-To-Face Meetings als auch bei geographisch entfernten Meetings zusammen mit einer Telefonkonferenz eingesetzt werden und durch digitales Skizzieren die Kommunikation zwischen Meetingteilnehmern fördern und die nachträgliche Auswertung der angefertigten Zeichnungen erleichtern. Jeder Teilnehmer benötigt dafür neben dem stiftbasierten Eingabetablet einen Computer mit einem Bildschirm, welcher über ein lokales LAN Netzwerk mit den anderen verbunden wird. Das Interface bietet mehrere gemeinsame Bereiche für Skizzen und Zeichnungen, die nach jedem vollendeten Strich eines Benutzers aktualisiert werden. Somit können alle Teilnehmer die Zeichnungen der anderen nahezu in Echtzeit betrachten.

Alle Meetings werden aufgezeichnet und können während dessen oder nach Abschluss der Sitzung aufgearbeitet werden. Jeder Zeichenstrich wird mit einem Zeitstempel versehen, sodass der Zeichenprozess Schritt für Schritt vor- und zurückgespult werden kann. Außerdem erlaubt *We-Met* bestimmte zeitliche und räumliche Zustände der Skizzen mit Tags zu versehen, sodass sie leichter zwischen diesen hin und her springen können. Die Form der Tags ist an Markierungen angelehnt, die Personen auch in ihren echten Notizbüchern verwenden, denn sie können nicht nur aus Textelementen, sondern auch aus handgezeichneten Schnörkeln, Kügelchen oder Sternchen bestehen, wodurch das Anlegen von Tags einfacher und intuitiver wird.

Schon während der Entwicklung von Prototypen ist es wichtig, das Konzept so früh wie möglich zu testen und zu evaluieren. *We-Met* wurde daher schon sehr bald in einer Studie mit potentiellen Nutzern getestet. Die Entwickler zogen drei Gruppen mit je drei Teilnehmern und eine Gruppe mit zwei Teilnehmern heran. Die Testpersonen erhielten die Aufgabe, unter Gebrauch von *We-Met* einen Haushaltsroboter zu konzipieren, der Müll aufsammelt und in einen dafür vorgesehenen Behälter wirft. Viel mehr als ein finales Design zu schaffen, ging es darum, möglichst viele Ideen zu generieren und zu skizzieren. Im folgenden Absatz werden die Resultate dieser Studie beschrieben.

Alle Teilnehmer empfanden das stiftbasierte Interface einfach zu benutzen. Es fiel ihnen nicht schwer, während des Schreibens und Kritzelns der Diskussion zuzuhören und sich aktiv daran zu beteiligen. Das Arbeiten mit einer Tastatur hingegen erfordert bei vielen Personen einen zu hohen kognitiven Aufwand, um einer

Diskussion noch mit ausreichender Aufmerksamkeit beiwohnen zu können. Aufgrund dieser Tatsache hat das stiftbasierte Interface das Potenzial, die Produktivität eines solchen Meetings zu erhöhen, denn es erlaubt den Teilnehmern die parallele Durchführung mehrerer Aktivitäten.

Die Testpersonen sprachen, zeichneten, schrieben, gestikulierten eifrig und hielten viel Augenkontakt während der Diskussion, ähnlich wie in herkömmlichen Meetings ohne Unterstützung von Computern. Das stiftbasierte Interface ermöglichte dabei sehr rasche und flüssige Übergänge beim Wechsel zwischen diesen Kommunikationskanälen.

Eine der Testgruppen arbeitete in einer höchst kollaborativen Art und Weise zusammen. Häufig definierte eine Person Anforderungen an den Haushaltsroboter und hielt diese handschriftlich fest, während eine andere Person die Anforderungen verfeinerte und Skizzen dazu anfertigte. Interessanterweise wählten diese Form der Interaktion genau jene Teilnehmer, die sich vorher nicht kannten. Alle Gruppen befanden einstimmig, dass es einfacher sei, sich in die Diskussion einzubringen, als bei herkömmlichen Meetings, in denen Whiteboards eingesetzt werden. Oft bedeutet in jenen Sitzungen, etwas beizutragen: aufzustehen, zur Tafel zu gehen, und jemand anderem den Stift zu geben. Die natürliche Hemmschwelle, die dadurch entsteht, entfällt bei *We-Met*, da jeder über einen Computer und Eingabestift verfügt. Der kreative Prozess der Ideenfindung kann so optimiert werden.

Die zweite Gruppe an Testpersonen wählte eine andere Form der Interaktion. Nach einer anfänglichen gemeinsamen Diskussion zeichnete und skizzierte jeder Teilnehmer für sich. Nachdem alle fertig waren, wurden die Ergebnisse hergezeigt und noch einmal diskutiert. Die Personen arbeiteten also getrennt parallel. Im weiteren Verlauf der Sitzung kam es auch vor, dass zwei Teilnehmer miteinander diskutierten, während ein Dritter für sich skizzierte. Nachdem die anderen fertig diskutiert hatten, brachte der Dritte seine neuen Ideen ein und zeigte den anderen die angefertigten Skizzen. Durch diese Art der getrennten Parallelität, die *We-Met* ermöglicht, können potenziell mehr Ideen gefunden werden und das Ergebnis des Meetings somit verbessert werden.

Anders als bei den vorhergehenden ergab sich in der dritten Testgruppe eine moderierte Form der Diskussion. In den ersten fünfzehn Minuten sprachen alle drei Teilnehmer miteinander, aber nur einer schrieb Notizen mit. Diese Person kontrollierte auch die Diskussion. Man kann hier das typische Modell eines Meetings mit Whiteboard erkennen.

Auf die Frage »Was gefällt Ihnen an *We-Met*?«, antworteten Teilnehmer aus allen drei Gruppen, sie würden es gut finden,

dass es die Möglichkeit biete, ein gemeinsames Produkt hervorzubringen. Im Vergleich zu herkömmlichen Meetings gäbe es ein besseres Allgemeinverständnis in der Gruppe, wodurch Missverständnisse reduziert würden.

Der getestete Prototyp von *We-Met* bot den Teilnehmern keinen privaten Platz für Skizzen und Zeichnungen. Einige der Testpersonen wünschten sich eine private Arbeitsfläche zur Aufzeichnung diverser Notizen. Sie wollten Skizzen auch zuerst fertigstellen, bevor sie bereit waren, sie den anderen Teilnehmern zu zeigen. Daher kam es vor, dass manche sich fernab vom eigentlichen Geschehen auf dem Canvas einen eigenen Platz für ihre Zeichnungen suchten. Andere Gruppen teilten die Arbeitsfläche auf die Teilnehmer auf, sodass jeder seinen eigenen Platz zum Skizzieren fand. Das Fehlen der privaten Arbeitsfläche könnte dazu führen, dass Ideen nicht mitgeteilt werden, da man sie bereits unfertig herzeigen müsste und ebenso könnten Ideen verloren gehen, die man sich für später notiert hätte.

In keinen der Testgruppen gab es Probleme bei der Koordination und es kam nie vor, dass Teilnehmer aus Versehen versuchten, auf dem selben Bereich der Arbeitsfläche zu zeichnen und sich dadurch in die Quere gekommen wären. Da jeder die Zeichenschritte der anderen Teilnehmer Strich für Strich verfolgen konnte, war jedem immer bewusst, wer gerade was zeichnete. Schwierigkeiten der Koordination gab es nur bei zwei Aktionen: beim Wechseln der Szene und Scrollen der Arbeitsfläche.

Um zu einer neuen Szene zu wechseln, drückt einer der Teilnehmer auf den »Neu« Button, wodurch die neue Szene allen Teilnehmern angezeigt wird. Zum Wechseln zu einer bereits vorhandenen Szene drückt man auf den »Szenen« Button und wählt die gewünschte Szene aus der Liste aller zuvor angelegten Szenen aus. Die Liste der Szenen wird nur der Person angezeigt, die auch den »Szenen« Button gedrückt hat.

Es gibt drei Aspekte dieses Szenarios, die das Potenzial haben, die Teilnehmer zu verwirren: das Entscheiden, ob die Szene gewechselt werden soll, das Wechseln der Szene und das Erkennen, dass die Szene soeben gewechselt wurde. Die Entscheidung zum Wechsel wurde erwartungsgemäß meist problemlos durchgeführt: Eine Person teilte den anderen Teilnehmern ihre Intention zum Wechsel mit und wartete auf die Zustimmung der anderen.

Die zweite Aktion, das tatsächliche Wechseln der Szene, führte gelegentlich zu Konfusion. Es kam vor, dass nach dem Einverständnis, die Szene zu wechseln, zwei Benutzer gleichzeitig eine neue Szene anlegten, ohne die Aktion des anderen dabei zu bemerken. Dadurch wurde eine Szene mehr angelegt als die Gruppe im Sinn hatte. Folglich geschah es, dass die Teilnehmer

verwirrt waren, als sie zu einem späteren Zeitpunkt versuchten, zur entsprechenden Szene zurück zu wechseln. In einem konkreten Fall wechselte einer der Teilnehmer fünf Mal die Szene beim Versuch, die gewünschte zu finden. Daraufhin versuchte ein anderer Benutzer, die korrekte Szene zu finden und die damit zusammenhängenden Wechsel führten zu leichtem Frust bei einem dritten Teilnehmer.

Gelegentlich realisierten Benutzer nicht, dass eine Szene gewechselt wurde. Das *We-Met* Konzept sieht zwar vor, für jede Szene einen eindeutigen Namen bzw. eine eindeutige ID anzuzeigen, aber der Prototyp war zum Testzeitpunkt noch nicht so weit entwickelt, weshalb den Personen weniger Informationen als notwendig angezeigt wurden und Fehlerquellen offen blieben.

Einigen Teilnehmern war nicht ganz klar, dass das Scrolling sich nur auf den eigenen Viewport und nicht den der anderen bezieht. Sie erwarteten sich ein ähnliches Verhalten wie beim Szenenwechsel, welcher von einem Teilnehmer für die gesamte Gruppe durchgeführt wurde. Deshalb gab es oft Probleme, die Bereiche zu finden, in denen gerade ein anderer Benutzer zeichnete. Den Testpersonen war es auch nicht einfach möglich, auf den Bildschirm eines anderen zu schauen, um sich bei der Suche nach dem gewünschten Bereich zu behelfen. Die Gruppe versuchte zwar, sich gegenseitig weiterzuhelfen, hatte jedoch Schwierigkeiten dabei. Das Konzept des Scrollens bereitet oft schon Probleme, wenn es um single-user Software geht, und im multi-user Softwarebereich scheint sich diese Problematik noch zu verschärfen.

Diese verlorene Zeit beim Koordinieren der Gruppe ist definitiv ein Rückschritt im Ideenfindungsprozess, aber die gesteigerte Effizienz im kreativen Teil der Arbeit ist ein Schritt in die richtige Richtung.

Zusammenfassend kann man sagen, dass die Studie, durchgeführt mit dem Prototypen des *We-Met* stiftbasierten Interfacekonzepts, deutliche Verbesserungen im Kommunikationsprozess während der kreativen Phase der Ideenfindung zeigt. *We-Met* erlaubt verschiedene Formen der Interaktion und ist dadurch sehr flexibel. Neben diesen Vorteilen sind auch sehr konkrete Schwachpunkte des Systems zum Vorschein gekommen. Ein unendlich großer Arbeitsbereich, der auf LCD Monitoren nur stark eingeschränkt dargestellt werden kann und die Tatsache, dass jeder Benutzer unabhängig scrollen kann, bringt einige Schwierigkeiten in der Gruppenkoordination mit sich.

### 1.2.3 Group Drawing Tools

Tangs und Blys Observationen animierten auch Greenberg et al. im selben Jahr (1992) *Group Drawing Tools* zu erstellen. In (Greenberg et al., 1992a) und (Greenberg et al., 1992b) beschreiben sie die Probleme und Erfahrungen, die sie mit dem Design von insgesamt vier Systemen gemacht haben, welche realtime Online Gruppenarbeit ermöglichen: *GroupSketch* & *XGroupSketch* - beides multi-user Skizziertools, *GroupDraw* - ein objektbasiertes multi-user Zeichenpaket und *GroupKit* - ein *Groupware Toolkit*.

Alle Systeme bauen auf sechs Designkriterien auf, die Tang im Anschluss an seine Observationen in (Tang, 1989) festlegte. Sie geben vor, was gemeinsam genutzte Arbeitsflächentools unterstützen sollten. Er hebt dabei die seines Erachtens hohe Bedeutung hervor, allen die Möglichkeit zu bieten, über die Arbeitsfläche gestikulieren zu können und betont in diesem Zusammenhang, dass der Prozess des Zeichnens an sich auch eine Geste ist, die allen Teilnehmern durch kontinuierliches feingradiges Feedback gezeigt werden muss. Ein weiterer wichtiger Punkt ist, dass das Tool nicht nur gleichzeitige Aktivitäten unterstützt, aber sie dennoch fördert. Sie bilden das Fundament von *GroupSketch*, *XGroupSketch*, *GroupDraw* und *GroupKit*.

Anmerkung: Tangs Designkriterien finden sich im Kapitel CSCW & GROUPWARE. Tabelle 5.2 zeigt die sechs Kriterien und eine Zusammenfassung der Gründe, warum sie gewählt wurden. ♣

#### 1.2.3.1 GroupSketch

*GroupSketch* ist ein simples, gruppenorientiertes Zeichentool, das einer beliebigen Anzahl an Teilnehmern ermöglicht, auf einem virtuellen Blatt Papier (dem Bildschirm) zu zeichnen (Greenberg and Bohnet, 1991). Jeder Benutzer hat einen eigenen Computer, der mit den anderen Computern über ein Netzwerk verbunden ist, daher hat jeder eigene Ein- und Ausgabegeräte zur Verfügung. Zu seinen Hauptmerkmalen zählen:

- ein What-You-See-Is-What-I-See (WYSIWIS) Editor, wodurch garantiert wird, dass jeder Benutzer zu jedem Zeitpunkt dasselbe auf seinem Monitor sieht wie alle anderen
- multiple, aktive Cursor, die alle Teilnehmer identifizieren und stets sichtbar sind
- unterstützte gleichzeitige Interaktion, sodass jeder Teilnehmer zu jeder Zeit arbeiten kann
- Benutzeraktionen (Cursorbewegung oder Zeichnung), egal wie kurz, sind sofort auf allen Screens sichtbar

- Zeichnen, Gestikulieren und Auflisten sind so einfach wie möglich umgesetzt (Benutzer müssen keine eigenen Modi auswählen)

**Abbildung 1.4** zeigt einen typischen *GroupSketch* Screen mit vier Teilnehmern einer Designsession. Auf der linken Seite befindet sich die gemeinsam benutzbare Zeichenfläche, in der die Benutzer zeichnen, schreiben oder zeigen können. Jede Person hat einen eigenen Cursor, der mit ihrem Namen versehen ist. Vier Aktionsmodi werden dabei unterstützt: mit dem Cursor zeigen, zeichnen, bzw. löschen/radieren und mit der Tastatur schreiben. Der Cursor spiegelt immer den aktuellen Modus wieder. Wenn keine Maus- oder Tastaturtasten gedrückt werden, nimmt er die Form einer zeigenden Hand an (Sandys Hand). Durch das Bewegen der Maus, kann ein Benutzer auf Dinge zeigen und simple Gesten durchführen. Zum Zeichnen verwendet man die linke Maustaste und zum Schreiben kann man einfach los tippen und *GroupSketch* platziert die Schrift an der aktuellen Position der Maus. Während des Zeichnens und Schreibens verwandelt sich der Cursor von einer zeigenden Hand in einen Schreibstift (Sauls Cursor). Durch Drücken der mittleren Maustaste wechselt das Bild des Cursors zu einem großen Pfeil, der dazu dient, die Aufmerksamkeit der anderen Benutzer zu erlangen (Irenes Cursor). Die rechte Maustaste ermöglicht das Löschen von Zeichnungen und Notizen, was wiederum den Cursor zu einem Radierer ändert (Wilfs Cursor).

Das Menü auf der rechten Seite in **Abbildung 1.4** erlaubt es einem Teilnehmer, privat ein Bild zu speichern, gespeicherte Bilder am Gruppenscreen zu zeigen, die öffentliche Arbeitsfläche zu leeren oder von einer Kollaboration auszusteigen. Mausbewegungen außerhalb der Arbeitsfläche bzw. im Menübereich sind privat und werden somit nicht an die anderen Teilnehmer gesendet. Laden eines Bildes oder Leeren der Arbeitsfläche wirkt sich auf alle Teilnehmer aus.

Greenberg & Bohnet testeten *GroupSketch* in (Greenberg and Bohnet, 1991) unter rein informellen Konditionen, ohne aufwendige Usabilitystudien durchzuführen. Dennoch konnten sie ein paar wichtige Beobachtungen machen: In einem typischen *GroupSketch* Szenario agieren die Teilnehmer auf die selbe Weise, wie sie es auch bei einem gemeinsam verwendeten Blatt Papier tun würden. Trotzdem ähnelte es nur einem Face-To-Face Meeting. Die Teilnehmer tendierten dazu, sich eifrig auf die gemeinsame Arbeitsfläche zu konzentrieren, da - dadurch dass sie sich einander nicht sahen und die natürliche Gestik fehlte - eine limitierte Wahrnehmung der Tele-Präsenz auftrat. Sie steckten ihre volle Aufmerksamkeit in Objekte, indem sie auf sie zeigten oder sie mit dem Cursor umkreisten. Zeichnen und Auflisten sind gleichzeitig unabhängig

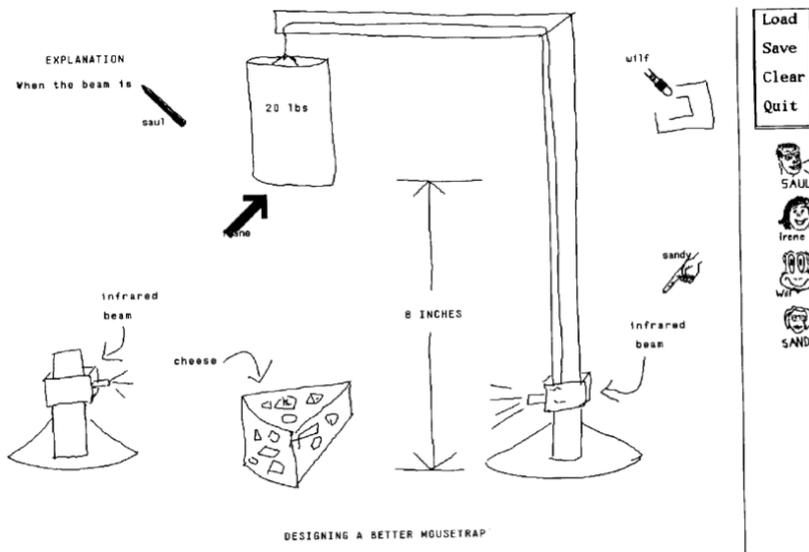


Abbildung 1.4: Eine typische *GroupSketch* Session mit 4 Teilnehmer. Sie zeigt die Grundfunktionen..

(eine Person ist verantwortlich für eine Zeichnung) und kooperativ (mehrere Personen arbeiten zusammen an einer Zeichnung). Personen arbeiten gleichzeitig, egal an welcher Stelle am Screen, und jeder kann aktiv eine Zeichnung erstellen, bearbeiten oder darauf zeigen.

### 1.2.3.2 *XGroupSketch*

*XGroupSketch* verfolgt ein ähnliches Konzept wie *GroupSketch*, unterscheidet sich aber deutlich in einigen wichtigen Punkten. Während *GroupSketch* immer im Vollbildmodus läuft, wird *XGroupSketch* in einem Fenster mit horizontalen und vertikalen Scrollbalken dargestellt. Dadurch handelt es sich streng genommen nicht mehr um einen WYSIWIS-Editor, da nicht mehr gewährleistet werden kann, dass jeder Benutzer zu jedem Zeitpunkt dasselbe sieht. *XGroupSketch* bietet gegenüber *GroupSketch* eine größere Palette an Funktionen. Über mehrere Menüs können die Benutzer unterschiedliche Linienstärken, Farben und Schriften wählen, jedoch bedeutet dies, dass die Applikation weitaus komplizierter in ihrer Bedienung ist, als *GroupSketch*.

### 1.2.3.3 *GroupDraw*

Im Gegensatz zu den vorhergehenden zwei rasterbasierten Zeichenprogrammen ist *GroupDraw* eine objektorientierte Applikation. Da das Design auf den Erfahrungen von *GroupSketch* basiert, wurde das Konzept mit multiplem Cursor und gleichzeitigem Arbeiten beibehalten. Zusätzlich wurden Funktionen wie in strukturierten Zeichen- und Skizzierpaketen (wie z.B. in *Claris MacDraw*)

hinzugefügt. Benutzer können nun Zeichenobjekte, wie Rechtecke, Linien, Kreise, Texte, etc. erstellen und diese selektieren und positionieren. Auch bei Groupware wurde das strenge WYSIWIS-Konzept gelockert. Genauso wie in *XGroupSketch* wurde der WYSIWIS Screen durch eine scrollbare Arbeitsfläche ersetzt und bietet den Benutzern zusätzlichen Platz, um an privaten Zeichnungen zu arbeiten. [Abbildung 1.5](#) zeigt einen *GroupDraw* Screen mit drei Benutzern, die an einem Design arbeiten. Das kleinere Anmeldefenster listet alle Teilnehmer einer Session auf, deren Standort, Telefonnummer und eine Schnellwahltaste, um diese telefonisch zu erreichen.

Das objektorientierte Konzept und die dadurch gewonnenen Möglichkeiten bringen eine Reihe an Herausforderungen an das Interface Design bzw. somit auch eine erhebliche Anzahl an möglichen Konflikten mit sich. Bereits die Tatsache, dass Objekte von Benutzern selektiert und verändert werden können, erfordert Maßnahmen zur Tilgung von Konflikten, die auftreten können. So muss überlegt werden, was passieren soll, wenn zwei Benutzer gleichzeitig versuchen, das Objekt zu verändern. Die einfachste Methode ist jene, das Objekt sofort zu sperren wenn ein Benutzer Zugriff erhält. Dazu muss auch eine sinnvolle Art der Visualisierung der Sperre gegenüber anderen Benutzern gefunden werden. Es wäre denkbar, dass das Objekt für jene Benutzer, die gerade keinen Zugriff darauf haben, halbtransparent dargestellt wird. Ebenso muss der Unterschied zwischen dem Selektieren und dem eigentlichen Zugriff auf ein Objekt für die Benutzer verdeutlicht werden. Dazu könnte eine zusätzliche Zeitverzögerung zwischen den beiden Stati helfen. Im vorgestellten Prototyp können Benutzer sofort auf ein Objekt zugreifen und es manipulieren, da das System annimmt, dass keine Konflikte auftreten werden. Falls doch ein Konflikt auftritt und der Zugriff gesperrt wird, schnappt das Objekt zurück zu seiner ursprünglichen Position. In der Praxis erscheint dies jedoch wie ein Flackern, da die Systemreaktion fast unverzüglich stattfindet.

Problematisch erweist sich auch der nahtlose Übergang von Aktionen und Funktionen, wie es von Tang ([Tang, 1989](#)) empfohlen wird (vgl. [Abbildung 1.1](#)). Die Tatsache, dass Benutzer nun aus einer Reihe an Objekttypen auswählen und zusätzlich einen Zeichenmodus auswählen müssen, erschweren Tangs Anforderung. Die Selektion eines Objektes von einer Palette oder einem Menü kann die Kontinuität der Gruppenaktivität stören. Während es in single-user Applikationen grundsätzlich nicht problematisch ist, ein komplexeres Interface zu haben, kann es in multi-user Software eine erhebliche Einschränkung im Workflow der gesamten Gruppe bedeuten.

*Die Auswahl von  
Objekten aus  
Paletten oder Menüs,  
stören den flüssigen  
Ablauf von  
Gruppenaktivitäten.*

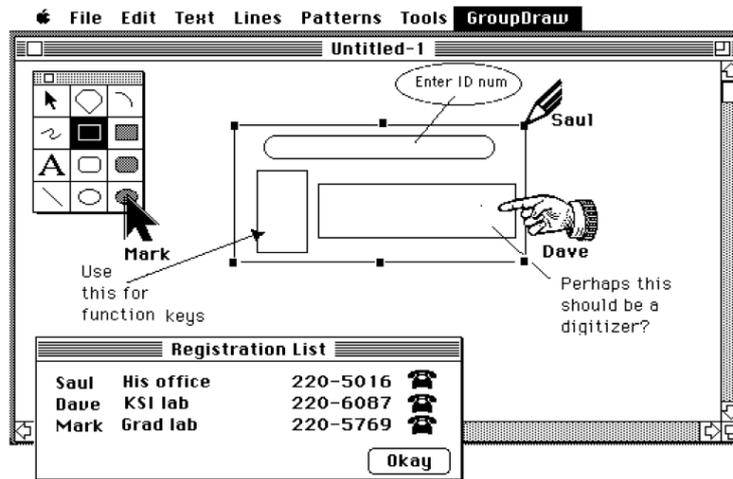


Abbildung 1.5: Beispiel einer Groupware Session. Sie zeigt die Arbeitsfläche und das Anmeldefenster

Ein weiteres Problem sind private Zeichnungen, welche laut Tangs Observationen gute Hilfsmittel darstellen. An ihnen wird oft gearbeitet und der Gruppe erst später präsentiert. *GroupDraw* verfügt über zwei Mechanismen, die dem Schutz der Privatsphäre innerhalb der Gruppe dienen. Einerseits können sich Benutzer durch Scrollen einen eigenen Bereich zum Skizzieren suchen und diesen erst nach persönlicher Evaluierung der Zeichnungen den anderen Teilnehmern zeigen. Zum anderen kann jedem Objekt ein »Verknüpfungstatus« zugewiesen werden. (Dewan and Choudhard, 1991)

#### 1.2.3.4 GroupKit

*GroupKit* ist ein Toolkit zur Entwicklung von verteilter Echtzeit-Groupware. Obwohl Applikationen, die mit Hilfe von *GroupKit* entwickelt werden, auch für andere Szenarios als das bloße Zeichnen geeignet sind, waren die Erkenntnisse der zuvor vorgestellten Systeme sehr wichtig für das Design von *GroupKit*.

Die Praxiserfahrungen mit *GroupSketch*, *XGroupSketch* und *GroupDraw* zeigen, dass neben den von Tang definierten Designkriterien (Tang, 1989) noch folgende Faktoren bei der Entwicklung von Groupware Zeichenapplikationen von Bedeutung sind:

- Die Notwendigkeit einer höheren Telepräsenz
- Die Möglichkeit nahtloser Übergänge zwischen lokalen, privaten Daten und gemeinsam erstellten Arbeiten zu bieten
- Berücksichtigung der Anzahl der Benutzer in der Gruppe

Die Recherchen (Greenberg et al., 1992a) haben gezeigt, dass *GroupSketch* und *GroupDraw* während der Zusammenarbeit von

räumlich distanzierten Gruppen eine ausreichend hohe Telepräsenz gewährleisten, solange Zeichnen und Skizzieren im Mittelpunkt der Konzeption stehen. Saß die Gruppe im selben Raum, so wandte sich die Aufmerksamkeit vom Computer ab und die Teilnehmer gingen über zu face-to-face Kommunikation, sobald die Skizzen nicht mehr zentrales Thema der Diskussion waren.

Ferner weisen die Autoren (Greenberg et al., 1992a) darauf hin, dass es zur Zeit der Untersuchungen (1992) noch zu hohe Barrieren zwischen virtuellen und realen Artefakten gibt. Im Zuge eines Meetings ist es nicht ohne weiteres möglich, echte Artefakte zu digitalisieren und mit anderen zu teilen.

Letztlich ist die Anzahl der Benutzer ein kritischer Faktor für die Art und Weise, wie eine gemeinsame Zeichenfläche genutzt wird. Die vorgestellten Systeme *GroupSketch* und *GroupDraw* funktionieren gut im Einsatz mit Gruppen, die aus 2-4 Personen bestehen. Sehr häufig arbeiten aber weitaus mehr Menschen zusammen und es bleibt die Frage im Raum, ob die Systeme dementsprechend skalieren. Ein Test mit acht Teilnehmern hat bereits einige Problemstellen entlarvt. Das Konzept der multiplen Cursor kann sehr schnell dazu führen, dass der Bildschirm überfüllt wirkt, und die vielen Bewegungen mögen etwas irritierend sein. Um diesem Phänomen entgegen zu wirken, verkleinern sich die Cursor entsprechend ihrer Anzahl. Wenn Benutzer bestimmte Aktionen durchführen, vergrößert sich der jeweilige Cursor während dessen wieder auf normale Größe, damit Interaktionen gut erkennbar bleiben.

#### 1.2.4 *ScreenCrayons*

Anders als bei den bis jetzt vorgestellten Systemen, die sich mit kollaborativem Zeichnen befassten, konzentrierten sich Olsen, Jr et al. mit *ScreenCrayons* 2004 - über ein Jahrzehnt später - auf den praktischen Aspekt von Zeichentools, nämlich auf barrierefreies Zeichnen bzw. Annotieren über die Grenzen eines Programmfensters hinaus. Der Stein, der ihr Projekt ins Rollen brachte, war eine Studie von Adler et al. (Adler et al., 1998), in der gezeigt wurde, dass Probanden beim Arbeiten an Dokumenten 70% mit Lesen verbringen und einen maßgeblichen Teil dieser Zeit mit Schreiben vermischen. Beim Erstellen und Bearbeiten von Dokumenten wird laut Studie zwar nur 18% während des Lesens geschrieben, jedoch benötigt das Anmerken und Schreiben von Notizen davon 48% der Zeit. Schilit beschreibt dies als »aktives Lesen« (Schilit et al., 1998), bei dem die Person die gelesenen Informationen verbessert, aussortiert, hervorhebt, zusammenfasst und organisiert. Hier haben Olsen, Jr et al. (Olsen et al., 2004) ein und versuchen eine möglichst vielseitig einsetzbare computerunterstützte Software zu entwickeln, die diese Aktivität unterstützt.

Aber warum softwarebasiert? Die grundlegende Metapher für moderne Computerworkstations ist Papier. Seit den Anfängen des *Xerox Stars*<sup>4</sup>, wurden Programmfenster so modelliert, dass sie herkömmlichen Papierseiten ähneln. Die meist benutzten Textverarbeitungs-, Zeichen- und Tabellenkalkulationsprogramme benutzen diese Metapher. In der Zeit vor Computern gab es zwar Schreibmaschinen zum Verfassen von Dokumenten, jedoch waren diese nicht besonders einfach zu handhaben, wenn es darum ging, Fehler auszubessern. Papierdokumente abzuändern oder in Umlauf zu bringen wurde dadurch ebenfalls erschwert. Das einzige, was einfach war, war auf den Dokumenten Markierungen und Anmerkungen anzufertigen, da dies handschriftlich geschah. Durch diese Ungleichheit wurde der Fokus der meisten Bürosoftwaretools auf das »schwere« Erstellen und Veröffentlichenden bzw. Verbreiten von Dokumenten gesetzt und das Annotieren außen vor gelassen. Aber gerade in heutiger Zeit, wo der Speicher immer billiger wird, über das Internet kommuniziert wird und der Computeraufschwung bzw. die Einführung von Standardformaten wie Portable Document Format ([PDF](#)) oder Hypertext Markup Language ([HTML](#)) eine Verlagerung der Benutzung von Dokumenten hervorruft, ist der Großteil der Leseaktivität digital geworden. Ein digitales Annotierungstool ist somit eine naheliegende Erweiterung.

Anhand eines ausführlichen Szenarios überlegten sich Olsen, Jr et al. im Vorfeld folgende Punkte, die das Anfertigen von Notizen mit sich bringt:

1. Notizen entstehen spontan während der Arbeit und müssen nicht zwangsweise mit der derzeitigen Aufgabe zusammenhängen.
2. Notizen entstehen im Zusammenhang mit vielen Applikationen. Im Szenario wurden z.B. ein Email-Client, ein Textverarbeitungsprogramm, ein Tabellenkalkulationsprogramm, ein Webbrowser, ein [PDF](#) Reader und eine eigens geschaffene Spezialsoftware verwendet.
3. Notizen sind häufig Zusammenfassungen oder hervorgehobene Auszüge vom Gelesenen. Sie dienen dazu, die Aufmerksamkeit an sich zu ziehen, damit man in Zukunft nicht erneut das gesamte Dokument lesen muss.
4. Notizen sind oft der Ausgangspunkt für späteres Schreiben (z.B. für eine Zusammenfassung).

---

<sup>4</sup> Der Xerox Star (Xerox 8010) wurde 1981 von der Xerox Systems Development Division entwickelt. Er war der erste kommerzielle PC mit einer anwenderfreundlichen grafischen Benutzeroberfläche, bestehend aus einem per Maus bedienbaren Desktop mit Menüs und Fenstern.

5. Notizen können verschiedene Informationsquellen miteinander verknüpfen, in denen Zusammenhänge gefunden wurden. Z.B. kann ein Zitat mit einem anderen Artikelabschnitt verknüpft werden.

Die meisten Annotierungssysteme unterstützen lediglich das Erstellen von Anmerkungen auf spezifischen Artefakt- bzw. Dokumententypen. Das Erlernen von solchen Notizsystemen für jede einzelne Anwendung ist dabei nicht nur unangenehm sondern auch mühsam. Das Ziel von *ScreenCrayons* ist es, eine einfache und universelle Software zur Verfügung zu stellen, die unabhängig von Programmimplementierungen oder Dateitypen arbeitet und alle oben genannten Bedürfnisse erfüllt, ohne andere Arbeiten damit zu beeinflussen. Sie soll über die gesamte Arbeit des Benutzers wirken.

Eine universelle Annotierungsmöglichkeit zu schaffen ist zwangsweise eine Herausforderung. In Hinblick auf die Softwarearchitektur wäre der einfachste Ansatz für alle Artefakte, an denen Notizen vorgenommen wurden, ein »Special Purpose Model<sup>5</sup>« zu erstellen. Alle Notizen, die in den einzelnen Programmen erstellt werden, müssten in dieses Modell »übersetzt« werden.

Der Vorteil dieses Ansatzes wäre, dass die Notizen in die Modellrepräsentation eingebettet werden können und somit eine Vielzahl an Visualisierungen (der Notizen in Verbindung mit dem zugehörigen Inhalt) designed und implementiert werden können. Der Nachteil, den das Artefaktmodelldesign mit sich bringen würde wäre, dass dokumentorientierte oder Electronic-ink Ansätze einen Übersetzer erfordern. Falls für eine Applikation kein Übersetzer existiert, gibt es auch keine Annotierungsmöglichkeit.

Die zweite Möglichkeit wäre, ein eigenes Protokoll zu entwickeln, das alle Applikationen verwenden. So ein Protokoll würde Signale wie »hier ist ein Electronic-Ink Strich, liefere mir eine Notizreferenz«, »hier ist eine Notizreferenz, zeig sie in der Applikation« und »hier ist eine Notizreferenz, liefere mir ihre *Bounding Box*<sup>6</sup> am Bildschirm«, umfassen. Zusätzlich würde es allen Applikationen, die das Protokoll verwenden, erlauben, Notizen hinzuzufügen. Die große Aufgabe wäre, alle interessanten Applikationen mit dem Protokoll auszustatten.

5 Model bzw. dt. Modell, ist ein Bestandteil des 1979 ins Leben gerufene MVC (Model-View-Controller) Softwarearchitekturmusters. Ziel des Musters ist eine spätere Änderung oder Erweiterung zu erleichtern und eine Wiederverwendbarkeit der einzelnen Komponenten zu ermöglichen. Die Modellkomponente enthält die darzustellenden Daten und ist von Präsentation und Steuerung unabhängig.

6 Eine Bounding Box oder auch minimal umgebendes Rechteck, ist das kleinstmögliche achsenparallele Rechteck, das ein oder mehrere geometrische Objekt(e) umgibt.

Mit *ScreenCrayons* entschieden sich Olsen, Jr et al. zu einem dritten Ansatz, der hingegen vollkommen im Bildraum arbeitet. Dadurch müssen zwar alle Applikationen ihre Informationen als Bilder bereitstellen, jedoch ist dies nicht weiters problematisch, da alle größeren fensterbasierenden (Betrieb-)Systeme die Möglichkeit bieten, Abbildungen des Bildschirms (sog. *Screenshots*) zu erstellen. Somit besteht ein universelles Medium, um Notizen auf jede Art von Information zu erstellen, ohne direkt mit der Applikation »kooperieren« zu müssen. Das ist ein wichtiger Schritt, um Annotierungen universell einsetzbar zu machen. Der Nachteil dieses Ansatzes ist jedoch, dass die Notizen - ohne Zugang zu einem Modell - keine strukturellen Vorteile mit sich bringen und sich somit total unintelligent verhalten. *ScreenCrayons* versucht daher eigenständig, aus dem Bildmaterial eine simple Struktur zu entnehmen. Ein Nachteil, dem *ScreenCrayons* jedoch nicht entgegenwirken kann, entsteht z.B. durch Scrolling<sup>7</sup> eines Dokuments, da ein Teil des nicht sichtbaren Kontextes verloren geht.

Olsen, Jr et al. fassen somit die für sie wichtigsten Anforderungen an ein Annotierungstool wie folgt zusammen:

1. Notizen müssen im Kontext, passend zu der jeweiligen Arbeit, erstellt werden - nicht in einem separaten Programm.
2. Das Erstellen von Notizen muss eine leicht anwendbare Tätigkeit sein, die keinen großen Aufwand mit sich bringt.
3. Eine Notiz muss den visuellen Kontext, in dem sie erstellt wurde, erhalten.
4. Die entstehenden Notizen müssen in kompakter, zusammengefasster Form verfügbar gemacht werden, damit sie nachträglich leicht durchstöbert und geändert werden können.

In *ScreenCrayons* besteht eine Notiz aus einem Namen, einem Screenshot, dem Gezeichneten und keinem oder mehreren Kommentaren. Um etwas zu annotieren, muss der Benutzer also 1) kennzeichnen, an welchem digitalen Artefakt eine Notiz angebracht werden soll, 2) dem Artefakt ein Thema zuweisen, um die Notiz zu kategorisieren und 3) optional Kommentare hinzufügen, um zu zeigen, was an der Notiz wichtig ist.

Die zentralen Instrumente in *ScreenCrayons* sind, wie der Name schon verrät, die sog. Crayons. Sie sind Electronic-Ink Spender, welche in einer »Crayon Box« aufbewahrt werden. Findet der Benutzer eine interessante Stelle, die er gerne kommentieren möchte, schaltet er in den »Crayon Modus«, schnappt sich einen Crayon

*Ein universelles Annotierungsprogramm benötigt ein universelles Medium um Notizen auf beliebigen Content zu erstellen.*

*Scrolling befördert einen Teil des Annotierungskontextes aus dem Blickfeld.*

---

<sup>7</sup> Scrolling: dt. Bildlauf

*Eine »auf dem  
Bildschirm kritzeln«  
Metapher ist einfach  
und universell  
einsetzbar.*

aus der Box und benutzt ihn, um Stellen am Screen hervorzuheben. Diese »auf dem Bildschirm kritzeln« Metapher ist einfach zu erlernen und komplett unabhängig von anderen Applikationen. Das Erfassen von Notizen geschieht mit Hilfe von Screenshots. Dabei reicht es aber nicht, die Screenshots einfach abzuspeichern, da die Ergebnisse unbrauchbar groß wären. Statt dessen können die Benutzer von *ScreenCrayons* interessante Stellen annotieren, und das Programm errechnet aus dieser Information den Bereich, der erfasst werden soll. Erkannt werden Kreise, die Text umgeben und Striche, die Text entweder unterstreichen, durchstreichen oder seitlich markieren. Die errechnete Boundingbox zieht sich bis an die Textgrenzen, wodurch der Kontext erhalten bleibt. (siehe [Abbildung 1.7](#))

*Modi sollten visuell,  
z.B. durch farbliches  
Einrahmen eines  
Fensters, erkennbar  
gemacht werden.*

Beim Schalten in den Crayon Modus wird zuerst ein Screenshot angefertigt und über den gesamten Screen bzw. alle Fenster gelegt. Der Benutzer sieht zwar keine Änderung, jedoch sind dadurch alle Applikationen scheinbar stillgelegt. Mauseingaben werden zu Zeichnungen auf dem Bildschirm, anstatt zu den jeweiligen Applikationen weitergeleitet zu werden. Um dem Benutzer zu zeigen, in welchen Modus er sich befindet, wird das aktive Fenster mit einem halbtransparenten Rahmen versehen, in der selben Farbe wie der ausgewählte Crayon. Alle Hintergrundapplikationen hingegen werden mit einer hellen Farbe überblendet, um die Blicke davon abzulenken. Die »Crayon Box« erscheint ebenfalls am Screen und bietet verschiedene Optionen, wie die Auswahl anderer Crayons, das Speichern von Notizen, das Abbrechen vom Notiz-Erstellvorgang oder das Ändern von Crayons. (siehe [Abbildung 1.6](#))

Die gesammelten Annotationen, bestehend aus Screenshots, den Highlightings und Kommentaren, können für den Benutzer auf verschiedene Arten visualisiert werden. [Abbildung 1.8](#) zeigt die drei verschiedenen Annotationsrepräsentationen mit ihren unterschiedlichen Kontextdetailgraden.

*Manuelles  
Umschalten von  
Modi ist mühsam.*

Das manuelle Umschalten vom Benutzer zum »Crayon Modus« ist zwar mühsam, aber notwendig. Ohne Kontrolle auf die laufenden Applikationen gibt es keinen anderen Weg, um zwischen den Annotierungseingaben und den Applikationseingaben zu unterscheiden. Jede Annotierungseingabe via Stift oder Maus könnte genauso gut als Eingabe der darunterliegenden Applikation aufgefasst werden. Das Überlagern vom gesamten Bildschirm mit dem Annotierungsscreenshot bewahrt den Kontext jeder Applikation und zeigt, dass Eingaben nun Annotierungen sind - keine Applikationsoperationen.



*Anmerkung: Die Problematik der Modiumscheidung wird im Kapitel [DESIGN VS COMPUTER](#) unter dem Abschnitt [Das Modusproblem](#) näher beschrieben.*

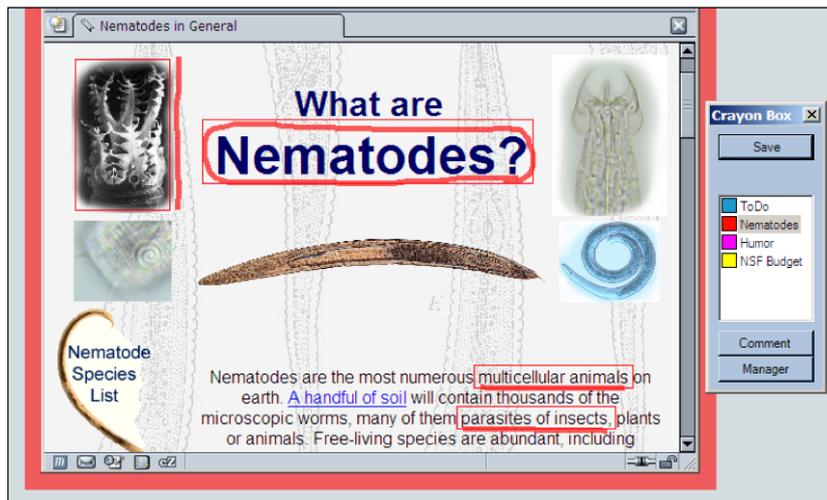


Abbildung 1.6: Der Zeichenmodus. Charakterisiert durch die Hervorhebung des aktuellen Fensters mit einem Rahmen in der aktuellen Crayonfarbe und der Crayon Box. Alle Hintergrundapplikationen werden überblendet.

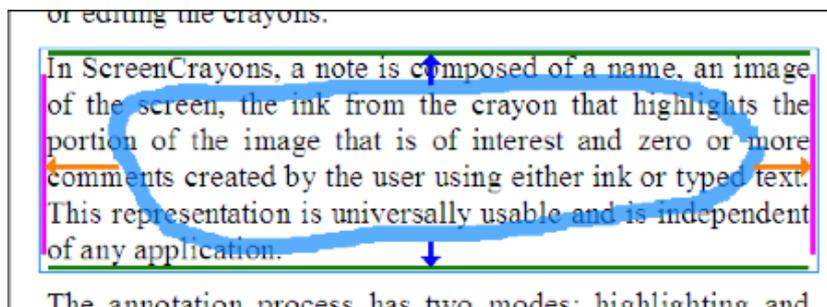
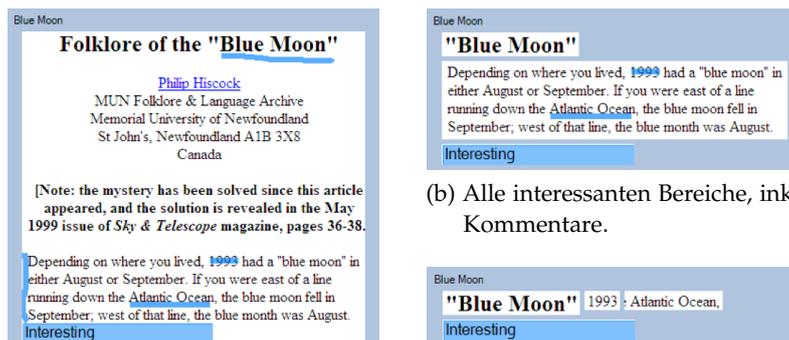


Abbildung 1.7: Visualisierung der Bereichsberechnung. Kreise und Geritzel bilden die Grundlage zur Berechnung der Boundingbox. Der Kontext bleibt bewahrt.



(a) Alle Highlights und Kommentare.

(b) Alle interessanten Bereiche, inkl. Kommentare.

(c) Direkt markierte Texte und Kommentare.

Abbildung 1.8: Drei verschiedene Repräsentationen eines Annotationsbeispiels. (a) zeigt alle Annotierungen im original Dokument mit zusätzlichem Kommentar. (b) zeigt die Annotierungen reduziert auf die interessanten Bereiche und zusätzlichem Kommentar und (c) zeigt die davon direkt angezeichneten Texte und den Kommentar.

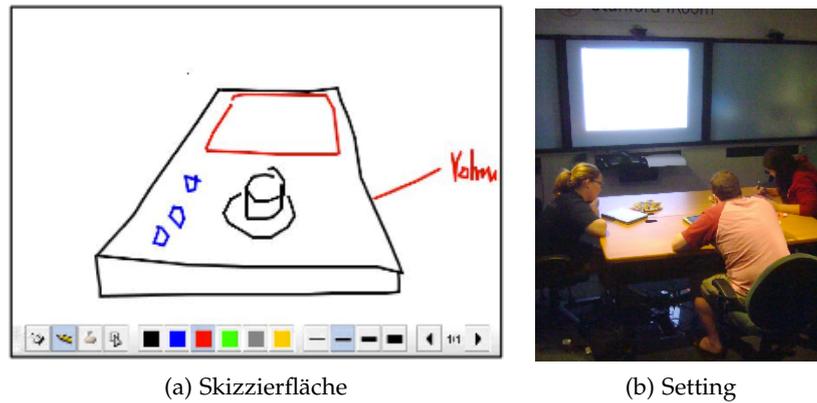


Abbildung 1.9: Das kollaborative Skizziersystem ermöglicht gleichzeitige Sicht und Eingaben auf eine gemeinsam genutzte Skizzierfläche (a) durch vernetzte Tablet PCs und einem digitalen Whiteboard. (b)

### 1.2.5 Ein durch Tablets unterstütztes Meetingtool

2007 entwickelten Bastéa-Forte und Yen ein kollaboratives Skizziertool, das mehreren Benutzern erlaubt, gleichzeitig auf einer gemeinsamen Arbeitsfläche zu arbeiten. Ihr System umfasst Tablet PCs mit integrierten Screens für jeden Teilnehmer und ein digitales Whiteboard für die Gruppe. Auf den Geräten wird eine vernetzte Skizzierapplikation ausgeführt, welche jedem Gruppenteilnehmer erlaubt, die gemeinsame Skizzierfläche zu sehen und auf ihr zu zeichnen (siehe [Abbildung 1.9](#)). Sie sind der Auffassung, dass die Zugangsgleichstellung vom Betrachten und Zeichnen die Hemmschwelle zum Mitzeichnen senkt und dass das Arbeiten im selben Raum die Kollaboration fördert.

Das kollaborative Skizziersystem ist ein Zeichenprogramm, welches in Java 5 realisiert wurde und auf den Tablet PCs bzw. dem digitalen Whiteboard läuft. Die Skizzierfläche wird auf allen Geräten »gespiegelt«, sodass jede auf dem Tablet kreierte Zeichnung sofort auf den anderen Tablets und dem Whiteboard sichtbar wird. Dadurch wird interaktives kollaboratives Skizzieren ermöglicht. So kann beispielsweise beim Designen eines Fernsehgerätes ein Teilnehmer die Grundrisse zeichnen und ein weiterer gleichzeitig beginnen, die Knöpfe hinzuzufügen. Alle Benutzer können mit Leichtigkeit gleichzeitig Skizzen sehen und neue hinzufügen.

Das Programm bietet Tools zum Freihandzeichnen und Löschen verschiedener Linienfarben & -stärken und eine Freiformselektion um Striche innerhalb einer Seite zu bewegen. Jeder Cursor ist mit dem Namen des jeweiligen Teilnehmers gekennzeichnet und erscheint am Whiteboard beim Zeichnen, Selektieren oder Löschen. Den Benutzern steht auch ein *Gesture Tool* zu Verfügung,

das es ihnen erlaubt, bestimmte Regionen auf der Arbeitsfläche hervorzuheben. Die Applikation unterstützt mehrere Seiten. Die Navigation zwischen den Seiten betrifft alle Displays und hält die Gruppe »auf der selben Seite«.

Bastéa-Forte und Yen gingen davon aus, dass das vorgestellte Setting die Hemmschwelle zum Mitzeichnen in Gruppenmeetings verringern wird, die Gesamtanzahl an Skizzen steigen wird und die Teilnehmer gleich stark mitarbeiten werden. Um diese These zu überprüfen, führten sie Untersuchungen mit Studentengruppen aus den Bereichen Human-Computer Interaction (HCI)<sup>8</sup> und Maschinenbau der Universität in Stanford durch. Fünf Gruppen zu je drei Studenten wurden mit Tablet PCs und einem digitalen Whiteboard ausgestattet, die miteinander vernetzt waren und eine gemeinsame Zeichenfläche zeigten. Im »Kontrollzustand« waren die Geräte von einander isoliert und zeigten von einander unabhängige Zeichenflächen, um Tools aus konventionellen Gruppenmeetings nachzuahmen. Im »Experimentalzustand« waren die Geräte miteinander vernetzt. [Abbildung 1.10](#) zeigt die beiden Zustände gegenübergestellt.

Am Anfang der Untersuchung, wurden den Teilnehmern die digitalen Tools und das Programm erklärt. Nach einer 10-minütigen Eingewöhnungsphase wurde der Gruppe 30 Minuten Zeit gegeben, um ein Design für eine Multifunktionsfernbedienung (für einen Herd, einen Backofen und eine Stereoanlage) zu erarbeiten. Anschließend hatten sie eine Minute Zeit, um ihr finales Design zu präsentieren. Die Aufgabe war eine vereinfachte Version jener, wie sie in ([Tang, 1991](#)) an Kleingruppen gestellt wurde. Nachträglich mussten die Teilnehmer einen Fragebogen mit je 20 Likert-Skala<sup>9</sup> Fragen und sechs offenen Fragen über Gruppendynamik, Gruppendynamik, Brainstormingstrategien und Brauchbarkeit des Systems in Bezug auf Kommunikation und Aufgabenlösung, beantworten. Zusätzlich gab es eine Nachbesprechung, in der die Teilnehmer über ihre Erfahrungen mit dem System befragt wurden. Die Server Applikation sammelte während der gesamten Zeit über Daten durch Aufzeichnen der Zeichen- und Zeigeeingaben in einem Logfile.

---

8 Human Computer Interaction (dt. Mensch-Computer-Interaktion) ist ein Teilgebiet der Informatik und beschäftigt sich mit der benutzergerechten Gestaltung von Systemen.

9 Die Likert-Skala ist ein Skalierungsverfahren zur Messung persönlicher Einstellungen, die durch sog. *Items* abgefragt werden. Die Antwortmöglichkeiten für ein Item, bilden den Grad der Zustimmung der befragten Person. Z.B.: Trifft zu - Trifft eher zu - Weder noch - Trifft eher nicht zu - Trifft nicht zu. Verwendung finden Likert-Skalen in Fragebogenerhebungen, insbesondere in der empirischen Sozial-, Markt- und Wahlforschung und der Psychologie.

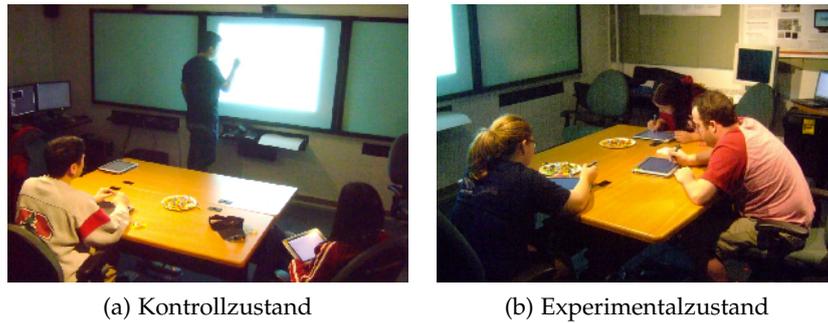
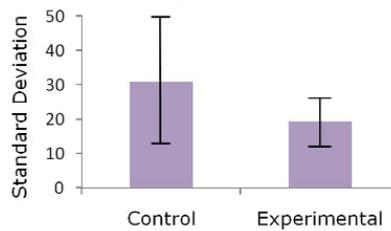
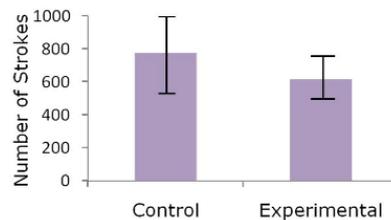


Abbildung 1.10: Die unterschiedlichen Arbeitszustände im Vergleich. Im Kontrollzustand arbeiten die Teilnehmer an einer Brainstormingaufgabe mit von einander unabhängigen Geräten - Tablet PCs und digitalem Whiteboard (a) - und im Experimentalzustand mit vernetzten Geräten (b)



(a) Mittlere Standardabweichung vom prozentuellen Skizzierbeitrag jedes Teilnehmers



(b) Mittelwert der gesamten Skizziereingaben aller Gruppen

Abbildung 1.11: Statistische Analyse des Logfiles. Sie zeigt das schlechte untereinander aufgeteilte Skizzierverhalten (a) und die dafür größere Anzahl an Gesamtskizzen (b) im Kontrollzustand im Gegensatz zum Experimentalzustand.

Die gesammelten Zeichendaten deuten darauf hin, dass die Skizzierverteilung im Experimentalzustand innerhalb der Gruppe eher ausgeglichen ist. Die Standardabweichung des Mittelwerts über den prozentuellen Anteil an gezeichneten Strichen pro Benutzer war ebenso wie angenommen kleiner als im Kontrollzustand ( $p = 0.17$ , siehe [Abbildung 1.11](#)). Der relative Skizzierbeitrag eines Benutzers ergibt sich aus dem Prozentsatz an Strichen, die ein Benutzer einer Gruppe gezeichnet hat. Dessen Standardabweichung zeigt die Schwankungen innerhalb der Gruppe. Im Gegensatz zu der anfänglichen Annahme zeigen die Zeichendaten, dass der durchschnittliche Zeicheninput der Kontrollgruppen größer als der der Experimentalgruppen ist ( $p = 0.21$ ). Die meisten Kontrollgruppen folgten einem typischen Meetingablauf, gezeichnet durch einen selbsterkorenen Schreiber am Whiteboard und den sitzenden Gruppenteilnehmern, die gelegentlich die Tablet PCs für private Skizzen nutzten und diese manchmal später der Gruppe zeigten. Im Gegensatz dazu näherte sich die Experimentalgruppe kaum dem Whiteboard. Durchschnittlich 57% aller gezeichneten Striche der Kontrollgruppen waren auf dem Whiteboard zu finden - in der Experimentalgruppe lediglich 4.4% ( $p = 0.01$ ). Die Antworten der Fragebögen bzw. die informellen Interviews der Nachbesprechung weisen darauf hin, dass das digitale Whiteboard im Experimentalzustand hauptsächlich zum Betrachten der Zeichnungen benutzt wurde und den Fokus der Gruppe erhielt. Trotz der gespiegelten Sicht auf die Tablet PCs, meinten Experimentalteilnehmer, dass das Whiteboard wichtig war um zu sehen was jeder arbeitet und alle Zusammenhänge zu erkennen. Weiters mochten sie die Tatsache, nicht aufstehen zu müssen um das Whiteboard zu benutzen. Eine Gruppe sagte sogar, dass durch die einzelnen Geräte das Whiteboard scheinbar mehr Macht erhielt, was sie daran hinderte es direkt zu benutzen, wie sie es vielleicht ohne die Geräte getan hätten.

Teilnehmer merkten weiters an, dass das System im Gegensatz zu traditionellen Tools von Anfang an klar stellt, wer welche Zeichnung beisteuert. Zusätzlich nehmen sie im Experimentalsystem bewusster wahr, wieviel jede Person zeichnet und halten sich deswegen sogar in manchen Fällen mit dem Zeichnen zurück. Eine Gruppe klammerte sich an diesen Aspekt und entschied, Farben zur Kennzeichnung der Skizzen zu verwenden. Darüber hinaus sagten einige Teilnehmer, dass sie unsicherer über die Qualität der Ideen waren, die sie auf das Whiteboard legten. So fühlten sie z.B. eine »bedingungslose Verantwortung« für ihre Zeichnungen, mit der Angst, keine »echten Beiträge« zu leisten. Einige Teilnehmer berichteten, dass sie über Ideen länger nachdachten, um sie ausgereifter der Gruppe mittels Skizzen zu präsentieren. Das ausgeprägtere Bewusstsein über die Qualität und Quantität der Beiträge erklärt möglicherweise, warum weni-

*Ein gemeinsam  
genutztes  
Whiteboard ist ein  
wichtiger Bestandteil  
kollaborativen  
Arbeitens.*

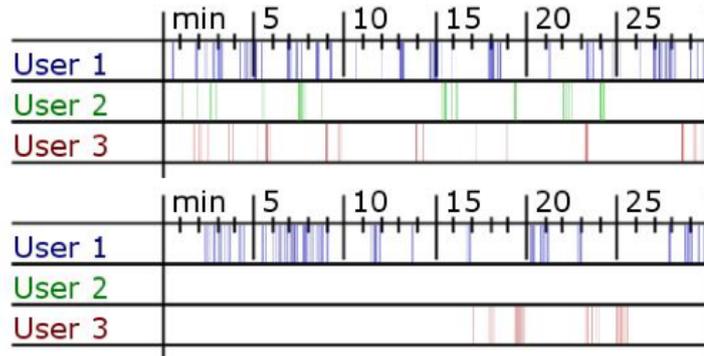


Abbildung 1.12: Darstellung der Strichhäufigkeit zweier Gruppen. Die Ergebnisse der Experimentalgruppe (oben) zeigen deren Interaktivität beim Skizzieren. Im Gegensatz die Kontrollgruppe (unten), mit nur einzelnen Wechsel zwischen den Teilnehmern.

ger Zeichnungen im Experimentalzustand gemacht wurden. Die geringere Beitragsrate widerspricht jedoch dem »go for quantity« Leitsatz von Brainstormings. Bastéa-Forte und Yen vermuten, dass dieses Bewusstsein aufgrund einer expliziten Skizzenidentifikation entsteht oder durch die Tatsache, dass Skizzen mehr ausdrücken als Worte.

Im Experimentalzustand wurde öfter interaktives kollaboratives Zeichnen beobachtet (siehe [Abbildung 1.12](#)). Gruppenmitglieder tendierten dazu, an andere Zeichnungen anzuknüpfen, indem sie Notizen zu existierenden Skizzen hinzuschrieben oder eine alternative Idee daneben aufzeichneten. In einer Gruppe bildeten alle drei Teilnehmer Listen von Ideen, in denen jeder Punkte hinzufügte. In anderen Gruppen wiederum wurde ein Schreiber ernannt. Den Teilnehmern fiel auf, dass ihnen die Möglichkeit, während ein Kollege spricht oder zeichnet gleichzeitig skizzieren zu können, sehr gefällt. Bei konventionellen Tools würde das sich Nähern einer weiteren Person zum Whiteboard die Aufmerksamkeit der Gruppe an sich ziehen. Das vorgestellte System erlaubt hingegen den Teilnehmern, Ideen ohne Unterbrechung »zu Papier zu bringen«. Eine genannte Konsequenz des Systems, war die Reduktion der Face-To-Face Kommunikation. Die Teilnehmer waren zu sehr beschäftigt, auf die Skizzen ihres Tablet PCs zu achten, dass sie sich untereinander weniger ansahen. Andererseits besteht das Problem auch bei konventionellen Whiteboardpraktiken, da die Gruppenmitglieder dazu neigen, zum Whiteboard zu schauen, anstatt zu den Kollegen. Trotz des unbegrenzten virtuellen Zeichenplatzes, hat das System nur einen limitierten Platz am Tabletdisplay. Teilnehmer erwähnten, dass die Handhabung manchmal schwer war, da sie öfters mehrere Skizzen gleichzeitig im Blick haben wollten.

Alles in allem zeigte die Studie die Auswirkungen von gemeinsamem Zeichnen auf Brainstormings. Der Zugriff auf eine gemeinsam genutzte Skizzierfläche mittels vernetzter Tablet PCs und ein digitales Whiteboard, schafft eine gleichmäßige Zeichenbeteiligung der einzelnen Gruppenmitglieder, da die eigenen Geräte die Beteiligungshemmschwelle heruntersetzt. Völlig unerwartet zeigen die Ergebnisse aber auch, dass das System die Gesamtanzahl an Skizzen beeinflusst. Es werden zwar weniger Skizzen gezeichnet, jedoch wird das Bewusstsein für die Anzahl an Beiträgen pro Teilnehmer gestärkt und somit die Qualität der Beiträge angehoben.

Als nächsten Schritt sehen Bastéa-Forte und Yen ein Hybridsystem, in dem eine private Zeichenfläche zum bestehenden System mit der geteilten öffentlichen Zeichenfläche, hinzugefügt wird. Sie glauben, dass eine Fläche zum privaten Entwickeln von Ideen - mit der öffentlichen Fläche in Sichtweite - helfen würde, den Hemmungen (von denen ein paar Teilnehmer berichteten) entgegenzuwirken. Durch Beisammenhalten der privaten und öffentlichen Zeichenfläche können Benutzer Skizzen anfertigen, ohne den Anschluss an die Gruppendiskussion zu verlieren und Ideen einfach zwischen den beiden Flächen übertragen. Weiters sind sie daran interessiert, die Vorteile des digitalen Whiteboards, beispielsweise durch das Erhöhen der Anzahl an sichtbaren Skizzen oder durch deren Einsatz als zentralen Blickfänger, auszureizen. Eine Möglichkeit dies zu bewerkstelligen wäre, das Whiteboard zu benutzen, um mehrere Seiten gleichzeitig anzuzeigen bzw. über größere Flächen navigieren zu können. (Bastéa-Forte and Yen, 2007)

*Eine zusätzliche private Zeichenfläche könnte mögliche Zeichenhemmungen nehmen.*

#### 1.2.6 Team Storm

*Team Storm* (Hailpern et al., 2007) ist ein Groupware System, das die Möglichkeit bietet, parallel an mehreren Ideen zu arbeiten. Es kommt bei Meetings in frühen Konzeptionsphasen zum Einsatz und fördert Kreativität innerhalb der Gruppe. Es konzentriert sich auf das Skizzieren von Ideen und bietet private und gemeinsame Arbeitsflächen, auf denen die Designer arbeiten können.

Das System besteht aus drei Hauptkomponenten: den sogenannten Canvases und den privaten sowie gemeinsamen Arbeitsbereichen. Eine Skizze, bzw. ein Design repräsentiert einen Canvas. Die Benutzer können eine beliebige Anzahl an Skizzen erstellen, dabei verwenden sie entweder Tablet-PCs oder andere stiftbasierte Eingabegeräte.

Zeichnungen werden durch Icons dargestellt, die beliebig auf der privaten Arbeitsfläche (Abbildung 1.13, unteres Fenster) positioniert und skaliert werden können. Diese Darstellung bietet deutliche Vorteile: Designer können Relevanz und Fortschritt

selbst definieren und die Skizzen dementsprechend anordnen bzw. skalieren. Durch diese Freiheit können ebenso semantisch verknüpfte Zeichnungen in Gruppen geordnet werden. Der Nachteil dieser Methode ist der hohe Platzbedarf auf dem Monitor oder dem Tablet-PC.

Auf der gemeinsamen Arbeitsfläche ([Abbildung 1.13](#), oberes Fenster), können Designer ihre Canvases mit den anderen Sitzungsteilnehmern teilen, diskutieren, überarbeiten und organisieren. Dieser Bereich, der für jeden vollständig sichtbar ist, bietet allen Benutzern die selben Möglichkeiten wie der private Arbeitsbereich. [Abbildung 1.13](#) zeigt im oberen Programmfenster eine exemplarische Anordnung mehrerer Skizzen, die von verschiedenen Designern angefertigt und mit den anderen geteilt wurden.

Während ein Designer eine Skizze innerhalb des gemeinsamen Arbeitsbereichs überarbeitet oder ergänzt, sehen alle anderen Teilnehmer unmittelbar seine Änderungen. Die Gruppe kann auch gleichzeitig an unterschiedlichen Skizzen im gemeinsamen Bereich arbeiten.

Der gemeinsame Bereich wird nicht nur auf den Tablet-PCs der Designer, sondern auch auf einem großen, für alle sichtbaren Monitor dargestellt. Ähnlich wie ein Whiteboard lädt diese Form der Darstellung dazu ein, sich davor hin zu stellen und Konzepte mit Hilfe zusätzlicher Kommunikationsformen, wie Gestik und Mimik, zu artikulieren. [Abbildung 1.14](#) zeigt, wie einer der Designer sich vor den Monitor stellt, um eine seiner Ideen zu erläutern.

Skizzen, die von einem Teilnehmer aus seinem privaten Arbeitsbereich in den gemeinsamen Bereich gezogen werden, befinden sich standardmäßig im Ausstellungsmodus. Das bedeutet, dass alle Designer die Zeichnung sehen, sie positionieren und skalieren, jedoch nicht ändern können. Sollte der Urheber der Skizze nicht wünschen, dass Größe und Position von seinen Kollegen verändert werden dürfen, so kann er den Zugriff sperren. Er kann die Zeichnung jedoch auch ganz freigeben, sodass nicht nur Skalierung und Position für die Kollegen veränderbar sind. Um eine Zeichnung vom gemeinsamen Bereich zu entfernen, zieht der Designer sie wieder zurück in seinen privaten Arbeitsbereich.

Designer können freigegebene Skizzen direkt auf der gemeinsamen Arbeitsfläche modifizieren, damit die anderen Teilnehmer die Änderungen live mitverfolgen können. Zusätzlich gibt es die Möglichkeit, sich eine private Kopie zu erstellen, die ohne Einsicht der anderen Teilnehmer editiert werden kann. Dadurch können Designs in mehreren Iterationen editiert und verfeinert werden. Die unterschiedlichen Versionen nebeneinander angeordnet zeigen die Evolution einer bestimmten Idee.

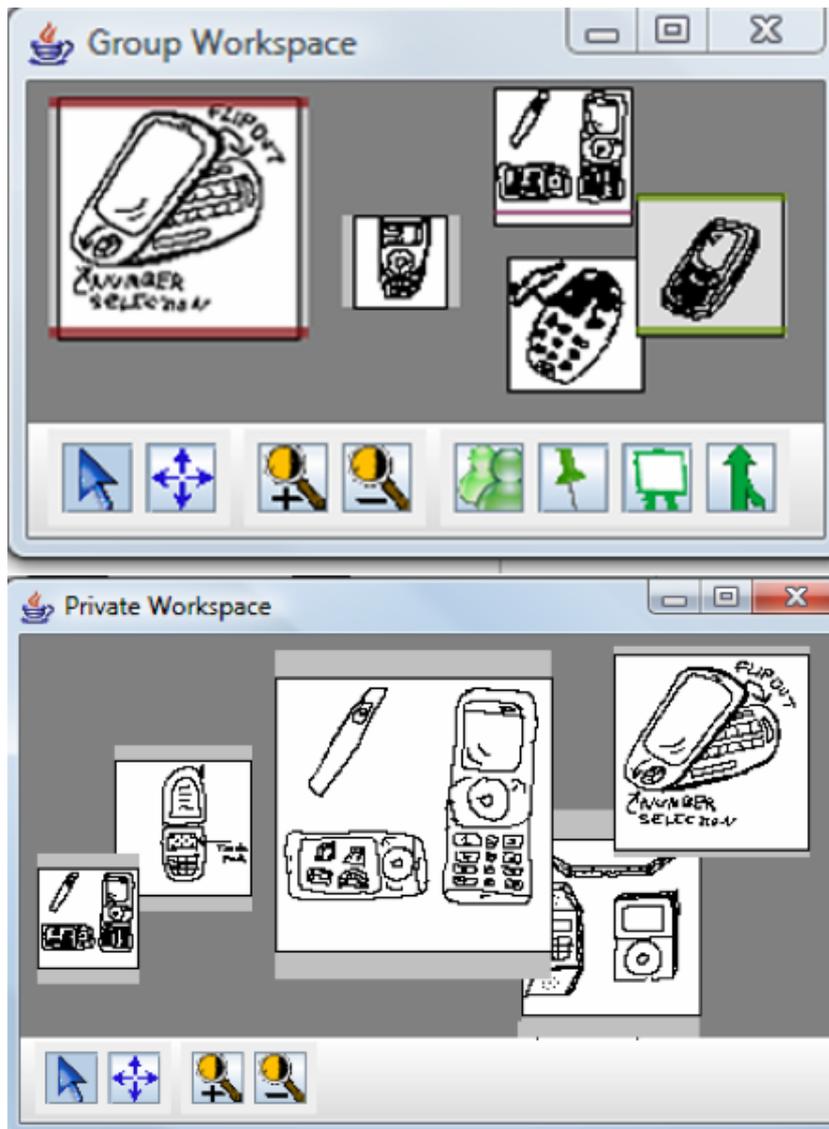


Abbildung 1.13: Gemeinsame und private Arbeitsbereiche in *Team Storm*. Designer können ihre Skizzen räumlich anordnen und die Größe anpassen.

Die an *Team Storm* durchgeführte Studie (Hailpern et al., 2007) zeigt, dass Designer, die das erste Mal mit dem System arbeiteten, einen sehr einfachen und direkten Zugang zu dieser Groupware fanden. Es wurden sehr viele Ideen generiert und die Teilnehmer nutzten sehr stark die Features zur Organisation der Konzepte und Designs. Unterschiedliche Charaktere brachten unterschiedliche Arbeitsweisen zum Vorschein: einige Designer teilten offen jeden ihrer gezeichneten Striche mit, während andere es bevorzugten, Skizzen erst im privaten Bereich anzufertigen, um sie selbst zu evaluieren, bevor sie sie herzeigten.

Die Gruppen nutzten das System auf sehr individuelle Art und Weise. Während manche hauptsächlich gemeinsam an einzelnen

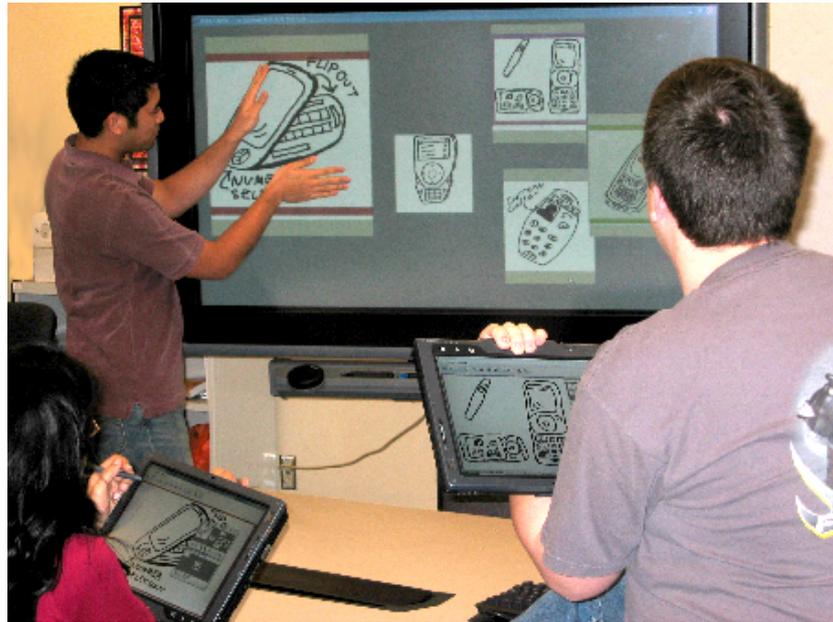


Abbildung 1.14: Designer bei der Ausarbeitung von Konzepten mit *Team Storm*.

Designs arbeiteten, zogen andere es vor, parallel an verschiedenen Konzepten zu arbeiten, die dann gemeinsam evaluiert wurden.

Neben diesen positiven Eindrücken kristallisierten sich auch einige mögliche Optimierungen des Systems heraus. Die Designer wünschten sich, Konzepte in Gruppen zu organisieren, die sie dann als Einheit positioniert, skaliert und hergezeigt hätten. Die Navigation wurde von vielen bei steigender Anzahl von Skizzen als ineffizient empfunden. Das System sah nur Scrolling und Zooming vor, die Teilnehmer wünschten sich hier weitere Möglichkeiten. Zusätzlich kam das Bedürfnis auf, andere digitale Artefakte (z. B. Bilder und Webseiten) einzubinden und dadurch Skizzen anzureichern.

### 1.3 CSCW UND GROUPWARE TOOLS

»Computer Supported Cooperative Work« ist jene Wissenschaft, die sich der Erforschung von kooperativer Zusammenarbeit mehrerer Personen widmet, um folglich Applikationen und Systeme zur Verbesserung selbiger zu konzipieren (Bannon and Schmidt, 1990). Der Fokus liegt darin, die Natur der kooperativen Arbeit besser zu verstehen und daraus Designkriterien abzuleiten, die es ermöglichen, effizientere Systeme zur Unterstützung dieses komplexen Prozesses zu entwickeln.

Dem gegenüber steht der Begriff Groupware, auch bekannt als »kollaborative Software« (Bannon and Schmidt, 1990), die mehreren Personen ermöglicht, gemeinsam am selben Projekt zu arbeiten. Groupware ist die technisch fokussierte Anwendung

der Erkenntnisse, welche die Erforschung von CSCW hervorgebracht hat. Diese bestimmte Art von Software zielt darauf ab, gruppensdynamische Prozesse zu vereinfachen und die Kommunikation innerhalb kooperierender Gruppen zu ermoglichen, beziehungsweise zu fordern. Der zusatzliche Koordinationsaufwand soll minimiert und das Losen von Problemstellungen in Teams optimiert werden (Rama and Bishop, 2006). Gerlicher und Ansgar (Gerlicher, 2007) unterteilen Groupware in synchrone, asynchrone und multisynchrone Systeme.

Im folgenden werden kurz ein paar Systeme und Konzepte vorgestellt, die CSCW unterstutzen.

*Anmerkung: Eine ausfuhrlichere Definition und Begriffserklarung folgt im Kapitel CSCW & GROUPWARE.*



### 1.3.1 Roomware

Mit dem Roomware Konzept versuchen Streit et al. (Streitz et al., 2001, 2002; Tandler et al., 2002), die komplette Einrichtung eines Sitzungsraums zu elektronische Geraten umzufunktionieren, um Teamarbeit moglichst effizient zu gestalten. Das Setting der ersten Generation von Roomware (Streitz et al., 2001) umfasst die DynaWall, die CommChairs, den InteracTable und den ConnecTable, welche im folgenden genauer beschrieben werden.

#### 1.3.1.1 DynaWall

DynaWall ist eine interaktive, elektronische Wand, die aus drei Segmenten besteht, welche alle uber ein Ruckprojektionssystem zur Darstellung von digitalen Inhalten und einer beruhrungsempfindlichen Oberflache verfugen. So kommt eine DisplaygroBe von 4,50m x 1,10m zustande. Jedes dieser Segmente ist mit einem eigenen Computer verbunden, jedoch vereint die zugehorige Software BEACH (Tandler, 2000) diese nahtlos und bietet dadurch eine homogene, ganzheitliche Arbeitsflache. Es konnen zwei oder mehrere Personen gemeinsam oder parallel uber die DynaWall interagieren.

Die auBerordentliche GroBe dieses Gerates bringt nicht nur Vorteile, sondern auch groBe Herausforderungen an die darunter liegende Software mit sich. Beispielsweise ware es auBerst umstandlich, ein digitales Objekt mit einem Finger von einem Ende des Bildschirms an das andere, 4,50m entfernte Ende zu ziehen. Daher haben die Entwickler das sogenannte »take and put« (nehmen und ablegen) Konzept implementiert, das ahnlich funktioniert wie das Ausschneiden und wieder Einfugen auf herkommlichen Desktopcomputern. Zusatzlich ist es moglich, Objekte herumschubsen. Ein Benutzer kann so ein beliebiges Objekt zu einem anderen Benutzer hin schubsen. Das Erstellen,

Bewegen und Löschen von Objekten und ihren Inhalten basiert auf Gestenerkennung und erfordert daher keine Auswahl eines bestimmten Modus (vgl. [Das Modusproblem](#)).

#### 1.3.1.2 *CommChairs*

Bei den *CommChairs* handelt es sich um mobile Stühle mit eingebautem Tablet-PC. Sie ermöglichen die Kommunikation und den Austausch von Informationen mit anderen Benutzern, die ebenfalls in einem *CommChair* sitzen, oder andere Roomware, beispielsweise die *DynaWall* verwenden. Man kann am Tablet-PC persönliche Notizen auf einem eigenen privaten digitalen Bereich machen, aber auch direkt auf die *DynaWall* schreiben und dadurch mit digitalen Objekten auf verschiedenen Geräten interagieren. Die Applikation *BEACH* ([Tandler, 2000](#)), die ebenso auf den *CommChairs* läuft, ermöglicht diesen nahtlosen Übergang zwischen den Roomwarekomponenten. Die Kommunikation läuft über ein drahtloses Netzwerk und jeder Stuhl ist mit einer eigenen Stromversorgung ausgestattet, um volle Mobilität zu gewährleisten.

#### 1.3.1.3 *InteracTable*

Für bis zu sechs Personen geeignet, ist der *InteracTable* ein gutes Hilfsmittel bei der Konzeption, Evaluierung und Ausarbeitung von kreativen Ideen. Es handelt sich dabei um einen Tisch mit einem in die Oberfläche integrierten Display, der Benutzereingaben von einem speziellen Stift, den Fingern oder einer kabellosen Tastatur entgegennimmt ([Streitz et al., 2001](#)). Da die Benutzer in einem Kreis um den *InteracTable* stehen, gibt es — anders als bei herkömmlichen vertikalen Displays — weder Unten und Oben, noch Rechts oder Links. *BEACH* unterstützt spezielle Gesten, die dazu dienen, Objekte am Display beliebig auszurichten. Somit können Benutzer aus jeder Perspektive eine gute Sicht auf die Objekte erlangen. Zusätzlich können für jedes Objekt mehrere »Views« (Ansichten) angelegt und jede zu einem beliebigen Benutzer hin ausgerichtet werden.

#### 1.3.1.4 *ConnecTable*

Der *ConnecTable* kam erst in der zweiten Generation von *Roomware* zur Ausstattung hinzu ([Streitz et al., 2002](#)). Dieser Tisch, ebenfalls mit eingebautem Display, wurde so konzipiert, dass man individuell oder zusammen mit anderen daran arbeiten kann. Seine Höhe ist stufenlos verstellbar und kann daher im Stehen als auch im Sitzen verwendet werden. Die Neigung des Displays kann darüber hinaus ebenso verstellt werden. Mehrere *ConnecTables* zusammengestellt erkennen sich automatisch gegenseitig, und ihre Displays fungieren als ein großer Darstellungsbereich.

Die Interaktion mit dem Tisch funktioniert in gleicher Weise wie die mit dem *InteracTable*, und durch eine drahtlose Netzwerkverbindung und eigene Stromversorgung sind die *ConnecTables* genauso mobil wie die *CommChairs*.

### 1.3.2 Weitere Konzepte zur Kooperation im selben Raum

Auf der Suche nach Konzepten zur Förderung der kreativen Zusammenarbeit von Industriedesignern in gemeinsamen Meetings halten Wang und Blevis (Wang and Blevis, 2004) sich an die Methode des »human centered design«<sup>10</sup> und führen zuerst eine ethnographische Studie an der Zielgruppe durch. Ausgehend von den so erlangten Erkenntnissen entwickeln sie vier unterschiedliche Konzepte, die speziell darauf abzielen, die konkrete Arbeitsweise der Industriedesigner zu unterstützen. Diese Konzepte sind aus technischer Sicht nicht unbedingt innovativ, dafür aber für ihre Zielgruppe besonders gut geeignet.

In ihrer Studie konzentrieren sich Wang und Blevis (Wang and Blevis, 2004) darauf, die Interaktionen zwischen den Designern zu beobachten und nicht die Interaktion mit speziellen interaktiven Geräten. Ferner observieren die Forscher die verschiedenen nicht technologischen Artefakte und Geräte sowie deren Orientierung und Gebrauchsweise, die bei der Zusammenarbeit zum Einsatz kommen. Davon ausgehend versuchen sie, möglichst ganzheitliche Konzepte zu entwickeln. Einen weiteren Fokus der Studie bildet das Identifizieren der Unterschiede zwischen privaten und gemeinsamen Arbeitsbereichen im kollaborativen Prozess.

#### 1.3.2.1 Designkriterien

Die folgenden Punkte haben sich aus dieser Studie als relevant herauskristallisiert und fließen als Designkriterien in die danach vorgestellten Konzepte mit ein.

- *Sitzordnung*  
Augenkontakt ist eine der wichtigsten Formen der Kommunikation bei der Zusammenarbeit in Gruppen. Daher bevorzugen die Teilnehmer, um einen Tisch herum zu sitzen, damit sie alle direkt ansehen können.
- *Reichweite*  
Es ist besonders wichtig, dass alle Objekte für jeden Teilnehmer in Reich- und Sichtweite liegen. Durch die Interaktion

<sup>10</sup> Beim human centered design steht der potenzielle Benutzer eines Produktes im Mittelpunkt des Entwicklungsprozesses. Seine Aufgaben, Ziele, Fähigkeiten und Eigenschaften werden analysiert und gelten als Maßstab bei der Gestaltung des Produkts. Human centered design zielt darauf ab, möglichst benutzerfreundliche Produkte zu schaffen.

mit den Objekten können Teilnehmer sich in den Fokus der Aufmerksamkeit stellen und besser kommunizieren. Rechteckige Tische bieten Nachteile für jene Personen, die an den Ecken sitzen, weshalb runde Tische bevorzugt werden.

- *Simultanität*  
Häufig interagieren mehrere Benutzer gleichzeitig mit den Objekten im Arbeitsbereich und fast immer sind viele verschiedene Artefakte präsent. Ein gutes Konzept muss daher diese Gleichzeitigkeit gewährleisten können, um die Effizienz durch paralleles Arbeiten nicht zu gefährden.
- *Gebrauch physischer Objekte*  
In der direkten face-to-face Kommunikation bevorzugen Personen physische Objekte, da sie im direkten Gespräch effizienter genutzt werden können als digitale Objekte.
- *Große Arbeitsflächen, viele Blätter, Wiederfindung und Vergleich*  
Designer benötigen sehr große Arbeitsflächen im Zuge der kreativen Ideenfindung. Sie verteilen die verschiedenen Konzepte darauf und Anordnung und Gruppierung spielen eine wichtige Rolle. Um einzelne Ideen leichter wieder zu finden und untereinander zu vergleichen, zeichnen sie üblicherweise nur eine davon auf ein gesamtes Blatt Papier.
- *Privatsphäre*  
Die Teilnehmer benötigen private Arbeitsbereiche, in denen sie Konzepte ausarbeiten und selbst evaluieren können, und sie brauchen gemeinsame Arbeitsbereiche, auf denen sie alle zusammen die Entwicklung vorantreiben.
- *Ausrichtung von Objekten*  
Die Ausrichtung von Objekten ist Teil der Privatsphäre. Sind Objekte zu einer bestimmten Person hin ausgerichtet, so gelten diese als privat, wenn sie hingegen zur Gruppe hin gerichtet sind, finden alle Teilnehmer einen Zugang.

Auf diesen Erkenntnissen aufbauend, haben Wang und Blevis (Wang and Blevis, 2004) vier verschiedene Konzepte entwickelt, die kollaborative Meetings von Industriedesignern unterstützen.

#### 1.3.2.2 Multi-user ClearBoard mit MiniNavigator

Das Multi-user ClearBoard mit MiniNavigator (Abbildung 1.15) ist eine Weiterentwicklung des von Hiroshi Ishii et al. konzipierten ClearBoard (Ishii et al., 1994). Alle Benutzer sitzen an einem Tisch, und vor ihnen wird ein großes Bild auf die Wand projiziert. Auf diesem werden die digitale Arbeitsfläche und ein Videobild der Teilnehmer übereinander gelegt, sodass jeder die Gestiken und Mimiken der anderen Kollegen als auch die digitalen Artefakte

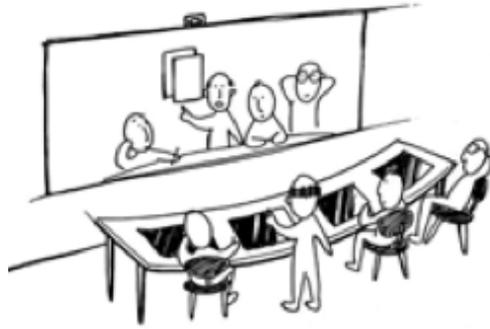
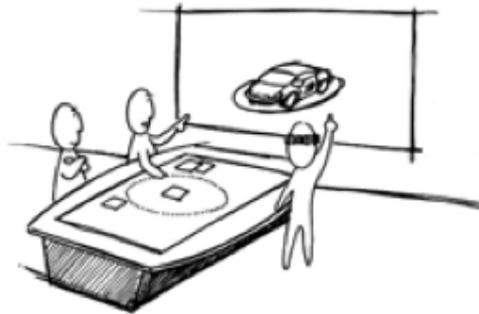
(a) *Multi-user ClearBoard mit MiniNavigator*(b) *3D-Navigator*(c) *ConverTable*(d) *More is more*

Abbildung 1.15: Vier verschiedene Konzepte wurden von Wang und Blevins entwickelt. Sie berücksichtigen die Designkriterien, die zuvor in einer Studie mit Industriedesignern ermittelt wurden.

sehen kann. Zusätzlich verfügt jeder Benutzer über ein eigenes Touch-Display. Auf diesem wird ebenfalls dasselbe Bild wie an der Wand angezeigt und dient zur Interaktion mit den digitalen Elementen auf der Arbeitsfläche.

### 1.3.2.3 3-D Navigator

Das 3-D Navigator Konzept ermöglicht den Designern, dreidimensionale Modelle aus verschiedenen Perspektiven zu betrachten, indem sie natürliche Gesten verwenden. Das System besteht aus einem großen, berührungsempfindlichen Tabletop-Display<sup>11</sup> und einem Projektor, der ein großes Bild auf eine Wand projiziert. Das Tabletop-Display dient als Schnittstelle und zeigt alle verschiedenen dreidimensionalen Modelle in Handflächengröße an. Diese können von den Benutzern mit dem Finger verschoben werden. Damit ein Modell vom Projektor auf die Wand projiziert wird, muss es auf dem Tabletop-Display in den in der Mitte dargestellten Kreis geschoben werden.

### 1.3.2.4 More is more

*More is more* berücksichtigt das Bedürfnis der Designer nach einer großen, gemeinsamen Arbeitsfläche, als auch jenes nach privaten Bereichen zur eigenständigen Entwicklung von Konzepten und Ideen. [Abbildung 1.15](#) zeigt die geschickte Anordnung von sechs Displays auf einem Tisch, die einzeln in eine vertikale oder horizontale Position gebracht werden können. Im aufgerichteten Zustand stellt ein Display einen privaten Arbeitsbereich dar. Designer können so für sich selbst arbeiten. Indem sie dann das Display hinunter klappen, wechseln sie in einen halb privaten Modus, denn Kollegen, die neben oder vor ihnen sitzen, können nun das Display sehen und mitarbeiten. Andere Personen, die nicht so nah sitzen haben jedoch noch keine Einsicht. Damit jeder teilhaben kann, gibt es im mittleren Bereich des Tisches ein gemeinsames, für alle sichtbares Display. Die Designer können ihre Modelle und Konzepte dort hin ziehen und für alle zugänglich machen.

### 1.3.2.5 ConverTable

Mit dem *ConverTable* schlagen Wang und Blevis ein weiteres Konzept vor, das es sehr vereinfacht, zwischen privatem und gemeinsamem Arbeitsmodus zu wechseln. Designer können *ConverTable* als einfachen Tisch nutzen und mit physischen Objekten und Artefakten darauf arbeiten. Bei Bedarf bringen sie das integrierte

<sup>11</sup> Tabletop-Display bezeichnet einen großen Flachbildschirm, der in eine Tischplatte integriert ist. In bestimmten Fällen ist dieser auch berührungsempfindlich und dient nicht nur der Ausgabe, sondern auch der Eingabe.

Display in eine vertikale Position und schaffen sich somit Zugang zu virtuellen Modellen. Ausgestattet mit Rädern, ist der *Convertible* ein höchst mobiles und elektronisch raffiniertes Möbelstück, das auch für spontanes Zusammenarbeiten sehr gut geeignet ist.

#### ZUSAMMENFASSUNG

Es gibt verschiedene ältere und jüngere Systeme, die Personen bei ihrer Zusammenarbeit innerhalb einer Gruppe unterstützen. Darunter einige relativ simple Anwendungen, beispielsweise *XSketch* und *ScreenCrayons*, die auf normalen Rechnern laufen, als auch sehr anspruchsvolle Systeme wie *Roomware*. Dieses vereint mehrere hardware- und softwarebasierte Lösungen und schafft so einen kompletten elektronischen Meetingraum. Der Überblick über den aktuellen Stand der Wissenschaft ist Grundlage für die weitere Fortführung der Arbeit, als auch für die Konzeption und Umsetzung der kollaborativen Annotationssoftware *Scribbler*.



*»Schaut man sich die gegenwärtige Lage des Designs an,  
fällt der eklatante Widerspruch zwischen der Publizität  
des Designbegriffs - nach dem Motto "Alles ist Design"  
- und der Theorielosigkeit des Designs auf.«*

— Gui Bonsiepe (Bonsiepe, 1992)

Anders als in den meisten Professionen kann die Disziplin *Design* nicht verallgemeinert werden. Design unterscheidet sich je nach Anwendungsgebiet in der Theorie als auch in der Praxis. Viele Designer arbeiten ungebunden von jeglicher Theorie, da die Theorie meistens nur versucht, passende Modelle zu finden, um die Praxis zu beschreiben. Wer also nun eine allgemeine Theorie über Design erwartet, *»muss zur Kenntnis nehmen, dass eine solche Theorie nicht das intelligible Produkt eines Einzelnen sein kann, sondern allenfalls das Ergebnis einer Designdisziplin sein könnte, welche ihre Praxis reflektiert«* (Schneider et al., 2008).

Im folgenden Kapitel wird daher vorerst versucht, allgemeine Eigenschaften und Erklärungen von Design zu erläutern. Im Anschluss wird die Designpraxis, insbesondere von HCI und Interaction Design, anhand modellzeigender Designmethoden erfasst.

## 2.1 WAS IST DESIGN?

*»Fragt man Fachleute verschiedener Abteilungen oder Berufsgruppen nach einer Definition von Design, so wird jeder eine andere Antwort finden«* (Sagmeister, 2008). Ein Bauingenieur würde beispielsweise den Vorgang zur Berechnung der Stahlträgerabmessungen für ein Gebäude als Design bezeichnen. Ein Modedesigner hingegen wäre verwundert über die Benutzung des Wortes *Design* in diesem Kontext, da damit ein relativ präziser, systematischer und mechanischer Vorgang beschrieben wird und nicht etwas Phantasievolles, Unberechenbares und Spontanes. Das Wort *Design* ist also das erste Problem, dem man sich stellen muss. Es wird tagtäglich verwendet, trägt aber für verschiedene Gruppen von Personen eine unterschiedliche Bedeutung. (Lawson, 1997)

John Heskett schrieb einst: *»Design is to design a design to produce design«* (Heskett, 2005). Neben dem Beweis, dass das Wort *Design* an sich bereits Verwirrung stiftet, zeigt er ein wenig überspitzt, die ebenfalls existierende grammatikalische Mehrdeutigkeit. Das erste ist ein Nomen und beschreibt Design als generellen

Überbegriff, wie z.B. in: »Design ist wichtig für die Wirtschaft.« Das zweite ist ein Verb und steht für eine Tätigkeit bzw. Prozess, wie z.B. in: »Sie hat den Auftrag einen neuen Mixer zu designen.« Das dritte ist wiederum ein Nomen, welches diesmal für ein Konzept bzw. einen Vorschlag steht, wie in: »Das Design wurde dem Auftraggeber zur Freigabe vorgelegt.« und das letzte ist wieder ein Nomen, das für ein fertiges Produkt steht, welches durch das Konzept umgesetzt wurde. Wie z.B.: »Der neue VW Beetle lässt klassisches Design neu aufleben.« (Heskett, 2005) Durch die Zerlegung kann man nicht nur erkennen, dass »design« ein Nomen und ein Verb sein kann, sondern auch, dass es zwei große Teilbereiche beschreibt. Zum einen die ursprüngliche Philosophie von Design - eine Tätigkeit bzw. ein Prozess um ein Produkt zu entwickeln - und zum anderen die Repräsentation eines Produkts. Webster beschreibt diesen Zusammenhang beider Bereiche wie folgt:

*»A design is an information base that describes aspects of this object, and the design process can be viewed as successive elaborations of representations, such as adding more information or even backtracking and exploring alternatives.«*

(Webster, 1988)

### 2.1.1 Design als Prozess

Webster beschreibt den Designprozess als schrittweise Ausarbeitung von Repräsentationen. Wann findet aber diese Ausarbeitung innerhalb eines Projektes statt?

Im klassischen Projektmanagement spricht man dabei von der Designphase. »Es gibt zwar Ansichten, dass die Designphase gleich der Entwicklungsphase ist (vgl. Zitat von Nokes), jedoch sind meist das die Projekte, die am Ende scheitern oder Schwierigkeiten haben, das Projekt in angegebener Zeit mit vordefiniertem Budget zu beenden.« (Sagmeister, 2008)

*»... on software projects, for example, the design and build phase are synonymous.«*

(Nokes, 2003)

Zur Entwicklung eines Produkts sollte man sich im Vorhinein ausführliche Gedanken über mögliche Probleme und Umsetzungsmöglichkeiten zur Bewältigung dieser machen. Viele Projektmodelle wie z.B. das Wasserfallmodell berücksichtigen bereits diesen Schritt. Jedoch gehen all diese Modelle von standardisierten Entwicklungsschritten aus, die zeitlich sequentiell

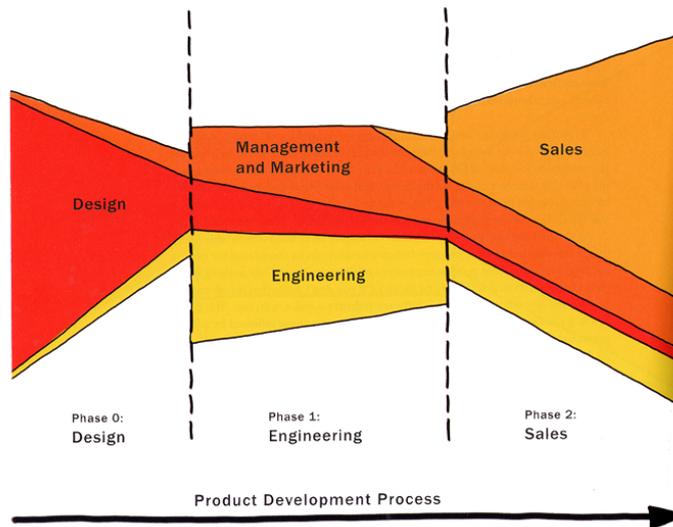


Abbildung 2.1: Der Produktentwicklungsprozess unterteilt in seine Verantwortungsbereiche. Von zentraler Bedeutung ist der Begriff Design, der das Design der Business- und Entwicklungspläne, sowie des Produkts selbst einschließt.

durchlaufen werden sollen. Die Ergebnisse der Analyse- bzw. Forschungsphasen werden dabei in Dokumenten festgelegt um die weitere Entwicklung zu leiten und zu unterstützen. Das Zurückgreifen auf vergangene Phasen ist hier nicht erlaubt, jedoch in der Praxis die Regel und resultiert auf sog. »Entwicklungsfehlern«. Genau hier sollte das Umdenken beginnen. Die zwei (leider) gebräuchlichsten Mythen in der Industrie sind:

1. »That we know what we want at the start of a project«, und
2. »That we know enough to start building it«.

(Buxton, 2007)

Die Realität sieht aber anders aus. Vor der Entwicklung stehen naturgemäß nicht alle Probleme fest, welche im Projektverlauf auftreten können. Es ist im Vorhinein also nicht möglich, eine vollständige Analyse durchzuführen ohne nicht auch Entscheidungen zu treffen, welche Analysen beeinflussen. Aus diesem Grund sollte der Designprozess durchgehend im Projektablauf verankert sein. [Abbildung 2.1](#) zeigt ein mögliches Modell eines Produktentwicklungsprozesses. Der Designprozess (in rot gehalten) erstreckt sich hierbei über alle Phasen im Projekt bis zum Verkauf und beinhaltet das Design der Business- und Entwicklungspläne, sowie das Design des Produkts an sich. Plump gesprochen ist die Hauptaufgabe des Designprozesses das Auffinden und Lösen von Problemen. Je nach Aufgabenstellung und

Art des Projektes variieren naturgemäß die Probleme und somit auch die Aufgaben von Design als Tätigkeit. (Buxton, 2007; Sagmeister, 2008)

### 2.1.2 *Design als Repräsentation*

Das Entwickeln verwendbarer Repräsentationen eines Produkts ist essenziell im Designalltag. Repräsentationen können formal oder formlos, exakt oder vage sein und dienen verschiedenen Zwecken innerhalb des Designablaufs. Designer sollten zum einen die Fähigkeit besitzen, eine passende Darstellung bzw. Repräsentation für ihre Aufgaben zu wählen, und zum anderen in der Lage sein, diese auch richtig einzusetzen. Im Laufe eines Produktdesigns können verschiedene Repräsentationen bzw. Modelle innerhalb des Designprozesses von Nöten sein. Ein gutes Modell ist präzise genug, um die Eigenschaften des Systems wiederzuspiegeln, und einfach genug, um Verwirrung zu vermeiden. Es verwendet eine Art der Repräsentation, welche zum Zweck passt. Die Wahl passender Modelle und das Konstruieren dieser ist ein schwieriger aber wichtiger Teil der Arbeit von Designern. Sie müssen dabei stets beachten, wie das Modell verwendet werden soll und wer es benutzen wird. Deswegen sollte auch die Modellierungstechnik auf den Abstraktionsgrad und den »Empfänger« (der mit dem Modell arbeiten wird) angepasst werden. (Preece et al., 1994)

Wie im Produktdesign gibt es auch im Softwaredesign eine große Menge an Repräsentationsarten bzw. Techniken, welche die Aufmerksamkeit auf verschiedene Aspekte des Designs lenken können. Diese Techniken bzw. Methoden, werden in Abschnitt [2.2 Designmethoden](#) näher erläutert.

### 2.1.3 *Designdisziplinen*

Design umfasst eine große Anzahl an verschiedenen Disziplinen. Sei es Produkt- bzw. Industriedesign, Kommunikationsdesign, Softwaredesign oder Modedesign, sie alle beinhalten auf ihrem Gebiet fachmännische Fähigkeiten und dienen zur Unterscheidung von Kompetenzen professioneller Designer. Über die Jahre kristallisierten sich ebenfalls die unterschiedlichsten Arbeitsfelder heraus. Das für Softwaredesign bedeutungsvollste der letzten Jahre ist (User) Interface Design bzw. Interaction Design, auf welche sich der folgende Text bezieht.

Um Interface Design beschreiben zu können, ist es vorerst nötig den Begriff des User Interfaces zu definieren. Das User Interface eines Systems besteht aus dem System selbst, dem Benutzer des Systems und der Weise, in der sie aufeinander einwirken. Es

enthält somit Elemente die Teil des Systems sind, Elemente die ein Teil des Benutzers sind und Kommunikationsmethoden um Informationen von einem Ort zum anderen zu bringen.

Wie [Abbildung 2.2](#) zeigt, gibt es eine Grenze zwischen den Elementen des Systems, die Teile des User Interfaces sind und denen, die für die internen Funktionen des Systems stehen. Das Finden der passenden Grenze ist Aufgabe des Designers und fällt in den Aufgabenbereich von User Interface Design ([Barfield, 1993](#)). Es befasst sich also mit der Entwicklung passender Benutzerschnittstellen zwischen Mensch und Maschine.

Interaction Design beschäftigt sich ähnlich wie Interface Design mit dem Verhalten von Mensch und Maschine. Jedoch spezialisiert es sich auf die Wechselwirkungen bzw. Interaktionen zwischen Menschen, welche durch Verbindungen mit maschinellen Produkten entstehen. Der Zusammenhang zwischen User Interface Design/Engineering bzw. Interaction Design ist in [Abbildung 2.3](#) ersichtlich. Sie zeigt ebenfalls verwandte Disziplinen, wie *Information architecture*, *Communication design*, *Usability engineering* oder *Human-computer interaction*, welche sie nicht nur beeinflussen, sondern auch Teil von ihnen sind. Man merkt somit wie schwierig es ist, Disziplinen, die zwar separat existieren aber sich mit vielen anderen Disziplinen überschneiden, abzugrenzen und ihre Funktionen zu beschreiben. Nicht jede Organisation benötigt einen Spezialisten für jede Disziplin. Eine Person, möge er sich Informationsarchitekt oder Interface Designer nennen, kann und wird höchstwahrscheinlich in mehreren Disziplinen, je nach Bedarf tätig sein. ([Saffer, 2007](#))

Zu diesen zählen ebenfalls andere akademische Disziplinen, wie z.B. im sozialen, technologischen oder organisatorischen Sektor. Natürlich ist es nicht nötig, dass Designer alle Fähigkeiten zur Erstellung von Designs besitzen - vor allem da das Designen von Interaktiven Systemen meist Aufgabe eines ganzen Teams ist - jedoch müssen sie in der Lage sein, Techniken anderer Disziplinen anwenden zu können bzw. wenn nötig die Mittel zur Erforschung dieser besitzen. ([Benyon et al., 2005](#))

Im Allgemeinen lässt sich aber User Interface- bzw. Interaction Design wie folgt beschreiben:

*»The design of the subjective and qualitative aspects of everything that is both digital and interactive.«*

oder allgemeiner:

*»The design of everything that is both digital and interactive.«*

([Moggridge, 2007](#))

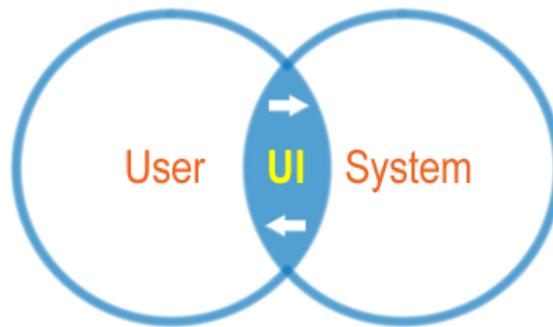


Abbildung 2.2: Visualisierung des User Interfaces. Das User Interface, welches aus dem System und dem Benutzer geformt wird, stellt Methoden zu Verfügung um die Kommunikation beider Seiten bestmöglich zu unterstützen. Das Finden einer passenden Grenze von Funktionen, die zum Interface oder intern zum System gehören, ist Aufgabe von User Interface..

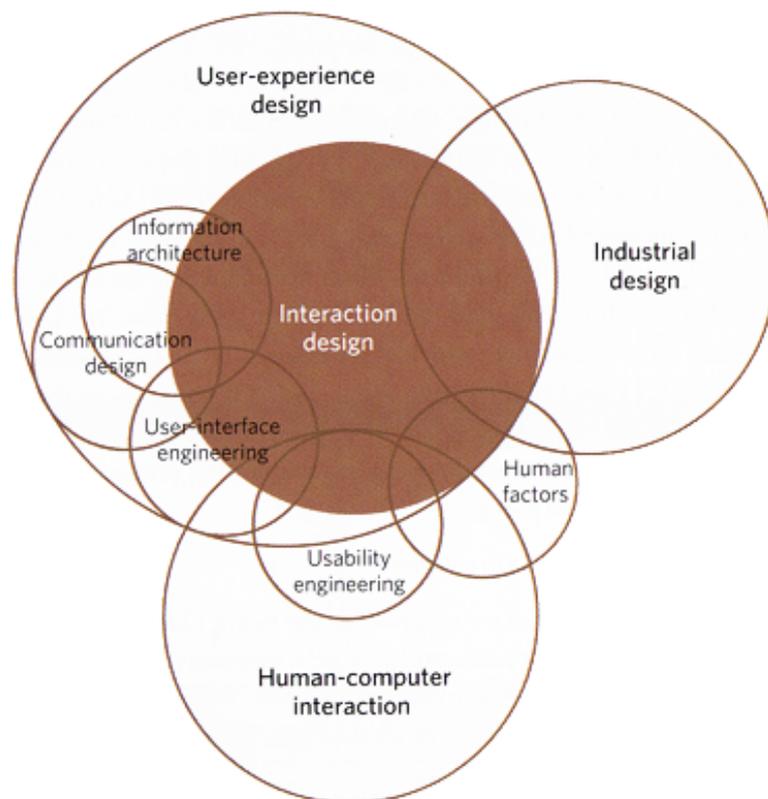


Abbildung 2.3: Verwandte Disziplinen von Interface- und Interaction Design. Sie beeinflussen nicht nur, sondern sind auch Teil von Interface bzw. Interaction Design. Die Schwierigkeit Disziplinen abzugrenzen, zeigt den interdisziplinären Charakter, den Designer an den Tag legen.

## 2.2 DESIGNMETHODEN

Vorstellungskraft ist grundlegend für effektive Designarbeiten. Sie hilft, Dinge aus einem anderen Blickwinkel zu sehen und Designkonzepte bzw. Ideen mit anderen zu erforschen. Verschiedene Repräsentationen von Designideen können in verschiedenen Etappen für verschiedene Menschen nützlich sein und ihre Vorstellungen beeinflussen. Sie helfen beim Finden, Austauschen und Auswerten von Ideen.

Es gibt viele Techniken bzw. Methoden, die benutzt werden können um Designprobleme zu verstehen und mögliche Lösungen zu entwickeln. Keine dieser Methoden führt höchstwahrscheinlich zum perfekten Design, aber zu einer Art Dokument oder Darstellung, die benutzt werden kann, um sich besser mit Auftraggebern, Benutzern und Kollegen zu verständigen. Verständigung ist es auch, wodurch Lösungen entstehen, ausgewertet, und eventuell zum Endprodukt umgewandelt werden (Benyon et al., 2005).

Es werden im Folgenden nun wichtige Methoden und Hilfsmittel für Designer beschrieben und Einblicke über deren Grundfunktionsweise geboten. Welche Techniken aber in einem Projekt tatsächlich Anwendung finden, hängt von mehreren Faktoren ab: die Herangehensweise an die Arbeit des Entwicklungsteams, die Art des Projekts, die zur Verfügung stehenden Hilfsmittel etc. Das Auswählen von passenden Methoden bzw. Repräsentationen ist Aufgabe des Designers und erfordert Geschick, um sie auch gut einsetzen zu können. Dabei sollte aber nicht die Repräsentation an sich, sondern die Vorstellungskraft und das Verständnis im Vordergrund stehen, geistig und vor allem erfahrungsmäßig. (Sagmeister, 2008)

*»Experience is a very dynamic, complex and subjective phenomenon. It depends upon the perception of multiple sensory qualities of a design, interpreted through filters relating to contextual factors. For example, what is the experience of a run down a mountain on a snowboard? It depends upon the weight and material qualities of the board, the bindings and your boots, the snow conditions, the weather, the terrain, the temperature of air in your hair, your skill level, your current state of mind, the mood and expression of your companions. The experience of even simple artefacts does not exist in a vacuum but, rather, in dynamic relationship with other people, places and objects.«*

(Buxton, 2007)

### 2.2.1 Artefakte

Harrison und Minneman beobachteten in (Harrison and Minneman, 1996), dass Objekte wesentliche Bestandteile von Designkommunikation sind, da sie einen Teil des *Repräsentationspools* von Designern ausmachen. Diese Objekte oder auch Artefakte genannt, können praktisch alles sein; z.B. ein Stift, ein Sessel, eine Zeichnung oder ein einfaches Blatt Papier. Im Konzeptdesign sind diese besonders wichtig, da in diesem Stadium des Produktentwicklungsprozesses noch kein »gemeinsames« Designobjekt existiert (Tuikka and Kuutti, 2001). Aus diesem Grund muss ein gemeinsames Artefakt erst durch Kollaboration konstruiert werden, um Diskussionen über Designoptionen und Ideen zu ermöglichen. (Larsson, 2003)

Artefakte sind ein bewährtes Mittel in HCI und CSCW. Verschiedene Studien zeigen, dass sie eine große Rolle in kooperativen Arbeiten spielen (vgl. Bardram and Bossen, 2005; Heath and Luff, 1992; Hutchins, 1995; Robinson, 1993; Schmidt and Wagner, 2002; Sellen and Harper, 2003; Shapiro et al., 1994; Vyas et al., 2008). Die Forschung und Literatur (z.B. Randall et al., 2007) beschäftigt sich dabei vorwiegend mit drei Hauptaspekten in Bezug auf Artefakte bei der Arbeit: dem *ökologischen*, dem *koordinativen* und dem *organisatorischen*.

*Ökologisch.* Die Umwelt von Artefakten kann einiges über die Arbeitspraktiken von Personen aussagen. Der Ort, die Position, der Aufbau und die Ausrichtung der Artefakte erlaubt uns zu verstehen was und wie gearbeitet wird. Verschiedene CSCW Studien (Heath and Luff, 1992; Sellen and Harper, 2003) zeigten, dass die Organisation des Arbeitsbereiches Auswirkungen auf die Arbeit hat. Kidd (Kidd, 1994) zeigte, dass die ökologische Struktur von Artefakten eine »primitive Sprache« bildet, welche Arbeit erklärt. Er vertritt den Standpunkt, dass die persönliche Anordnung von Artefakten (wie z.B. Papier), Personen befähigt ein besseres Verständnis aufzubauen, da sie so ein Gesamtbild ihrer Arbeit bekommen. Ebenso werden Aufbewahrungsmöglichkeiten, wie z.B. Papierstapel oder -mappen zu externer Repräsentation, die zusätzliches Verständnis vermitteln und bewahren. Der physische Kontext und die Positionierung geben Artefakten oder ihrer Aufbewahrung eine »Bedeutung«. Somit spiegelt der ökologische Aspekt eines Artefakts einige Vorgänge wieder, die zum Verständnis von Arbeitspraktiken beisteuern können.

*Koordinativ.* Verschiedenste Variationen von Artefakten, welche im Arbeitsbereich, zu Hause oder sonst wo benutzt werden, können als Wissensvermittler dienen. Flugstreifen<sup>1</sup> (vgl. Shapiro et al., 1994), Pinnwände (vgl. Bardram and Bossen, 2005) oder Papierdokumente (vgl. Sellen and Harper, 2003), zeigten beispielsweise schon in der Vergangenheit, dass wichtige Arbeitsschritte durch Artefakte koordiniert werden. Ein Papierartefakt kann eine bleibende Informationsform sein oder auch ein Medium, welches von einem Arbeitsbereich zum nächsten wandert, um kollaboratives Arbeiten unter Mitarbeitern zu unterstützen. Sellen und Harper untersuchten in (Sellen and Harper, 2003) die unterschiedlichen Merkmale von Papier und zeigten, dass dessen physikalische Eigenschaften verschiedene menschliche Handlungen - wie greifen, herumtragen, falten oder darauf schreiben - ermöglichen. Sie fungieren durch ihre weitgehende Verfügbarkeit (Heath and Luff, 1992) und Verteilung (Robinson, 1993), als Koordinations-tool mehrerer Personen, die an gemeinsamen Projekten arbeiten (Vyas et al., 2008). Gemeinsamkeiten können somit durch physikalische Objekte einfacher erarbeitet werden als durch verbale Kommunikation alleine (Larsson, 2003).

*Organisatorisch.* Wie lange ein Artefakt an einem Arbeitsbereich benutzt wird, sagt viel darüber aus, wie die Arbeit organisiert ist. In einem Betrieb wandert Information durch verschiedene Darstellungsformen. Ein Artefakt, im speziellen dessen raumzeitlicher Aspekt, kann somit Auskunft über wichtige Prozesse, Protokolle oder Konventionen eines Arbeitsvorgangs geben. Schmidt und Wagner zeigen in (Schmidt and Wagner, 2002), dass CAD Zeichnungen als mehrlagiges Artefakt die Koordination und Organisation mehrerer verschiedener Aktivitäten erleichtern kann. Eine CAD Zeichnung mit einer Mischung aus Chiffren für Funktionen und Materialien könnten so beispielsweise auch Details über Verantwortungsbereiche einer Arbeit veranschaulichen. Kidd zeigt in (Kidd, 1994), dass durch die fühlbaren raumzeitlichen Aspekte der Artefakte (wie z.B. Papierstöße), Arbeitsfortschritte gemessen werden können. (Vyas et al., 2008)

Die tatsächlichen Eigenschaften von Artefakten müssen zwangsweise nicht optimal oder vollkommen geeignet sein, um das Denken von Designern zu unterstützen. In einer Studie über die Benutzung von Artefakten von Designstudenten erkannte Breton (Breton and McGarry, 2000), dass es im Vorhinein nicht möglich ist, das »richtige« Artefakt für eine Designsituation zu bestimmen:

---

<sup>1</sup> Flugstreifen, engl. *Flight (Progress) Strips*, werden in der Luftraumüberwachung eingesetzt. Die noch üblicherweise in Papier gehaltenen Streifen beinhalten wichtige Informationen zu allen derzeitigen und zukünftigen Flügen. (Bentley et al., 1992)

*»The hardware was simply conveniently available and had some attribute that meant students found it helpful to gesture and think with.«*

(Brereton and McGarry, 2000)

#### 2.2.1.1 *Conversational Props*

Der beabsichtigte Zweck von Artefakten, das Verständnis essentiell zu fördern, bedeutet, dass Designer Objekte suchen, die ihnen helfen, ihre eigenen Gedanken zu formen und diese den anderen Teilnehmern eines Designteams zu vermitteln. Diese Objekte werden auch *Conversational Props* (Brinck and Gomez, 1992) genannt, da mit ihnen ein Element der Realität zu einer Konversation hinzugefügt wird. (Larsson, 2003)

#### 2.2.1.2 *Boundary Objects*

Artefakte können auch als Mediatoren zwischen verschiedenen Personen oder Gruppen dienen, indem sie das Terrain werden »on which conflicts and collaboration occur« (Perry and Sanderson, 1998). Bei Gruppenmitgliedern verschiedener Disziplinen mit unterschiedlichen Interessen und Zielen werden Artefakte als sog. *Boundary Objects* (Star and Griesemer, 1989) angesehen und benutzt. Sie sind nützliche Repräsentationen für alle Gruppenmitglieder, haben aber möglicherweise noch zusätzliche Bedeutungen für jeden einzelnen. (Larsson, 2003)

*Boundary Objects »have different meanings in different social worlds but their structure is common enough to more than one world to make them recognizable, a means of translation.«*

(Star and Griesemer, 1989)

Laut Bucciarelli (Bucciarelli, 2002) sind Boundary Objects oder gemeinsame Artefakte auch für Teilnehmer gleicher Disziplinen essentiell, da die analytische Natur einer Sprache meist nicht ausreicht.

*»[Language] hardly allows . . . the kind of experimentation and innovative thinking that designing requires.«*

(Bucciarelli, 2002)

#### 2.2.2 *Skizzieren*

Die Kunst des Skizzierens ist eine Fähigkeit, die jeder Designer beherrschen sollte. Ideen und Gedanken können schnell veranschaulicht und erforscht werden - entweder für sich selbst oder für andere. (Sagmeister, 2008)

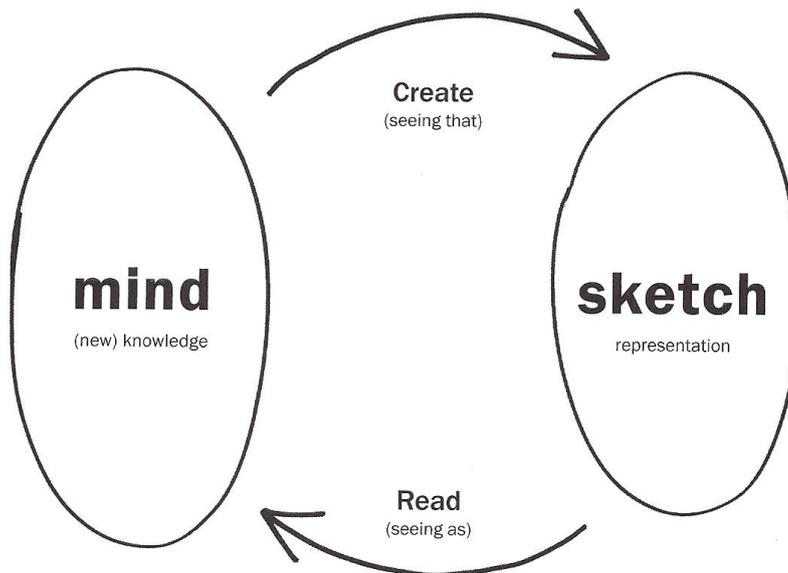


Abbildung 2.4: Die Konversation bzw. Wechselbeziehung zwischen der Skizze (rechte Blase) und dem Verstand des Designers (linke Blase). Eine Skizze wird aus dem aktuellen Wissen erstellt (oberer Pfeil). Das Lesen bzw. Interpretieren der gewonnenen Repräsentation (unterer Pfeil, erzeugt neues Wissen.)

Donald Schöns Beobachtungen zeigten, dass Designer beim Skizzieren eine Wechselbeziehung mit ihren Zeichnungen eingehen. Er nennt dies »*conversation with the materials of the situation*« (Schön, 1983), womit er den Vorgang beschreiben will, in der Designer etwas zeichnen, dann ihre Skizze interpretieren und wieder weiter zeichnen. Den selben Prozess beobachtete auch Gedenryd, in dem Designer durch wie er es nennt »*stepwise reasoning-by-drawing*« nachdenken und neue Ideen generieren (Gedenryd, 1998). [Abbildung 2.4](#) veranschaulicht diese Wechselbeziehung.

#### 2.2.2.1 Warum skizzieren?

Skizzen sind ein Weg, um Ideen offen zu legen, eigene Gedanken zu veröffentlichen oder flüchtige Gedanken zu fixieren. Wörter können zwar das selbe, aber Skizzen haben den Vorteil, bildliche Gedanken direkt zu vermitteln. Benutzte Elemente und räumliche Relationen auf Papier verdeutlichen Elemente und räumliche Relationen in der Realität. Das könnte auch ihre Allgegenwart erklären; Landkarten und architektonische Pläne wurden in verschiedenen Kulturen überall auf der Welt in Stein geritzt, in Leder gebrannt, in Ton geprägt und auf Papier gezeichnet. Skizzen können ebenfalls abstrakte Ideen metaphorisch beschreiben, wo Elemente und räumliche Relationen auf Papier, abstrakte

Elemente und Relationen ausdrücken. Das Äußern von Ideen in einem bildhaften Medium macht Verständnis und Folgerung einfacher als in einem abstrakten Medium wie Sprache. Die Klarheit von Skizzen und gleichartigen kognitiven Werkzeugen treibt das Erinnerungsvermögen an, in dem sie etwas liefern, das nicht nur auf das unzuverlässige Gedächtnis des Menschen vertraut. Sie bieten ebenso ein Souvenir an frühere Gedanken, in dem sie gleichzeitig den Inhalt und dessen Bedienung vermitteln. Die öffentliche Natur von Skizzen erlaubt einer Gemeinschaft, Ideen zu beobachten, zu kommentieren oder zu ändern - und dies in der Repräsentation festzuhalten.

So wie Sprache (gesprochen oder geschrieben), sind Skizzen eine Form von Kommunikation. Aber anders als Sprechen dienen Skizzen auch zur Kommunikation mit sich selbst. Eine Aufgabe von Skizzen ist, die Vollständigkeit und interne Konsistenz einer Idee - im Besonderen räumliche Ideen - zu prüfen. Eine Skizze ist ein schriftliches Modell einer Idee, eine Existenzüberprüfung. Eine weitere Aufgabe ist es, neue Relationen und Formen für sich selbst zu entdecken, welche zu neuen Ideen führen können. Skizzen werden mit einer bestimmten Ausgangsidee und einem Ziel begonnen, enden aber meist durch Zufall in neuen Objekten und Konfigurationen. Dies führt zu unbeabsichtigten Entdeckungen und kann eine fruchtbare Quelle für neue Designideen sein. (Tversky, 2002)

Goel geht einen Schritt weiter und versucht konkrete Eigenschaften für Skizzen zu formulieren:

1. Die dichte Anordnung von einzelnen Skizzen, die sich vielleicht auch nur gering unterscheiden, hilft eine Skizze gegebenenfalls in eine andere umzuwandeln und das Ausschließen von Möglichkeiten im Vorhinein zu unterbinden.
2. Die Mehrdeutigkeiten von Skizzen versichern, dass die Darstellungen bzw. auch Inhalte während der frühen Phasen des Designs unbestimmt bleiben. Das ist wichtig, um die Entwicklung des Designs durch Konkretisieren von Ideen nicht zu früh zu blockieren.
3. Die dichte Anordnung von inhaltlich ähnlichen Skizzen, oder auch Skizzen der gleichen Bezugsquelle versichern, dass Möglichkeiten nicht ausgeschlossen werden, um Ideen in andere umzuwandeln.

(Goel, 1995)

#### 2.2.2.2 Was beinhalten Skizzen?

*Schematische Strukturen.* Das beste und weitverbreiteste Beispiel dafür sind Landkarten und Wegbeschreibungen. Sie beinhalten

wichtige Informationen über ihren Zweck und eliminieren die unwichtigen. Darüber hinaus vereinfachen und verzerren sie sogar die Informationen, um einer bestimmten Struktur zu entsprechen. Die Struktur, die mit den Skizzen eingefangen wird, ist nicht die Struktur der Umwelt, sondern eine konzeptuelle Struktur der Information. Wie Sprache, bestehen Skizzen aus Elementen. Diese können vereinfachte Figuren, Linien, Kurven und tropfenartige Gebilde sein (Tversky et al., 2000). Diese Elemente können in verschiedenen Weisen kombiniert werden um verschiedene, sprachähnliche, Bedeutungen zu erlangen (Goodman, 1968).

*Hierarchische Strukturen.* Skizzen von Regionen zeigen z.B. andere Charakteristiken. Die Reihenfolge in der Personen Landkarten zeichnen, spiegelt die konzeptuelle Struktur der Karten wieder (Taylor and Tversky, 1992). Versuchen Teilnehmer eine gesehene Karte zu reproduzieren, bleibt die Reihenfolge der Elemente gleich. Das zeigt, dass Personen hierarchische Strukturen aufbauen um die Umwelt zu strukturieren (Tversky, 2002).

Skizzen sind keine Präsentationen der Realität, sondern Repräsentationen der Realität (Tversky and Lee, 1999). Sie unterscheiden sich von der Realität auf folgenden Arten: sie lassen bestimmte Informationen weg, fügen Informationen hinzu und verzerren Informationen. Somit sind sie keine Bilder, zumindest nicht im »klassischem« Sinne (Kosslyn, 1980). Skizzen können vielmehr Abbildungen von Ideen sein. Als solche können sie Ideen - besonders bildhafte - effektiv vermitteln.

### 2.2.2.3 Was gewinnt man aus Skizzen?

Man kann nicht garantieren, dass Personen Skizzen so verstehen, wie sie der Macher beabsichtigte. Häufig sind sie selbst für denjenigen, der sie angefertigt hat ein Rätsel, wenn er sie nach langer Zeit erneut betrachtet. In zahlreichen Bereichen, mitunter bei Karten und Gerätedarstellungen, wird versucht, eine Struktur zu vermitteln. Die Struktur einer Route kann durch eine gut gezeichnete Wegbeschreibung, genauso wie die Struktur eines Gerätes durch eine gute schematische Darstellung entnommen werden (Heiser and Tversky, 2002).

Skizzen bieten Designern, wie bereits erwähnt, eine Quelle für neue Ideen. Skizzen und Diagramme können auch nützlich sein, um mehr als nur eine Struktur zu vermitteln. Sie sind auch ein effektives Mittel, um bestimmte Abstraktionen zu verdeutlichen, welche von der Struktur gefolgert werden und keinen direkten Zusammenhang mit der Skizze an sich haben. Gute Beispiele hierfür wären eine Fahrradpumpe, die Bremsen in einem Auto oder ein Flaschenzug. All diese Dinge sind in Bewegung bzw. bewegen sich, um einen bestimmten Zustand zu erreichen. Wie sie sich bewegen und was ihr Ziel ist, wird üblicherweise nicht

direkt in den Skizzen verdeutlicht. Das Hinzufügen eines Pfeiles hingegen ändert ihre Interpretation.

*»When asked to write descriptions of what is portrayed in the diagram, participants viewing simple diagrams of a car brake, pulley system or bike pump write structural descriptions. When arrows were added to the diagrams, participants write functional descriptions of the devices, explaining what they do, step-by-step.«*

(Tversky, 2002)

Pfeile vermitteln zeitliche Abläufe und erlauben den Betrachtern ein Gerät gedanklich zu animieren (Hegarty, 1992).

#### 2.2.2.4 Sketching for Experience

Egal ob Designer zeichnen oder schreiben, die besten Werkzeuge dafür bleiben Stift und Papier. Kein digitales Medium war bis jetzt in der Lage, die Flexibilität, die Geschwindigkeit und die Mühelosigkeit des Skizzierens auf Papier oder Whiteboard zu überbieten (Sagmeister, 2008). Warum das so ist, soll in [Kapitel 4](#) erörtert werden.

Eine weitere bewährte Form des Skizzierens ist Modellierung. Modelle können durch eine Vielzahl an Materialien, von Ton bis Styropor, gebildet werden und bieten somit dreidimensionale Repräsentationen (siehe [Abbildung 2.5](#)). Modelle wie Skizzen können in einem kurzen Zeitraum schnell zusammengefügt werden, um grobe Annäherungen von Gegenständen und Umgebungen zu erstellen. Die Unbestimmtheit dieser, die auch schon Goel erwähnte, führt dazu, dass die Betrachter sich in ihrer Kommentarfunktion uneingeschränkt fühlen und stets neue Ideen und Anregungen einbringen.

Bill Buxton spricht bei dieser Art von Skizzen (dies inkludiert auch Modellierung) von *Sketches for Experience*:

*»One thing that we know is that sketches for experience and interaction design will likely differ from conventional sketching since they have to deal with time, phrasing, and feel - all attributes of the overall user experience.«*

(Buxton, 2007)

Auch wenn sich *Sketches for Experience* von herkömmlichen Skizzen unterscheiden, unterliegen sie trotzdem denselben Eigenschaften.



Abbildung 2.5: Einige physikalische Modelle, erstellt in einem Designstudio.

Skizzen sollten:

- schnell, reichlich und in kurzer Zeit herstellbar, billig und verworfbar sein,
- eindeutige Gestik, minimale Detailstufe und angemessenen Grad an Feinheit haben,
- und durch Mehrdeutigkeit zum Erforschen und Vorschlagen anregen.

(Buxton, 2007)

#### 2.2.2.5 *Low-fidelity Prototypen*

Skizzen werden auch öfters fälschlicherweise als Prototypen angesehen. Da im nächsten Punkt näher auf Prototypen eingegangen wird, soll an dieser Stelle lediglich auf die Eigenschaften von *Low-fidelity Prototypen*, welche Skizzen am ähnlichsten sind, Bezug genommen und der Unterschied zu Skizzen verdeutlicht werden.

*Low-fidelity Prototypen* haben wie Skizzen wenig Ähnlichkeit mit fertigen Produkten. Es werden Materialien, wie z.B. Papier oder Karton verwendet, die sich von der gewollten fertigen Version unterscheiden. Sie sind simpel, billig in der Herstellung und leicht zu verändern. *Low-fidelity Prototypen* werden nie mit dem Hintergedanken produziert sie zu behalten und in das Endprodukt zu integrieren (Sharp et al., 2002).

Trotz der Ähnlichkeit zu Skizzen besteht ein feiner Unterschied. Beide beschreiben zwar Designkonzepte, dienen aber unterschiedlichen Zwecken und finden folglich in verschiedenen Stadien des Designprozesses Einsatz. Skizzen werden wie schon erwähnt in den anfänglichen Phasen der Ideenfindung verwendet, wobei

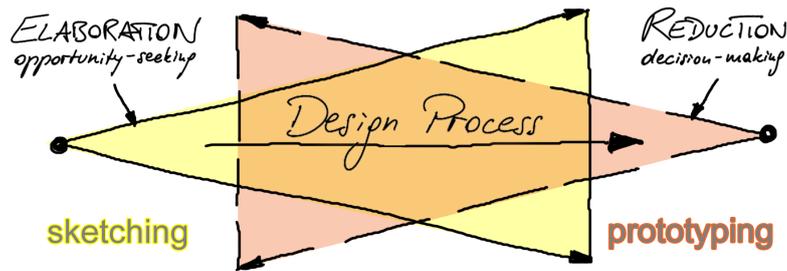


Abbildung 2.6: Skizzen und Prototypen im Designprozess. Die Reduktion, die durch Entscheidungsfindungen bei Prototypen entsteht, wird durch Ausarbeitung von neuen Ideen mit Skizzen ausgeglichen. Dies verdeutlicht die Bedeutungen beider Methoden und deren Einsatzzweck.

Prototypen im Allgemeinen erst in späteren Phasen eingesetzt werden. Wie [Abbildung 2.6](#) zeigt, können *Sketching* und *Prototyping* als Trichter im Design-Prozess verstanden werden. Am Anfang des Prozesses steht die Ausarbeitung verschiedenster Ideen und Möglichkeiten – am Ende die Reduzierung dieser bzw. die Entscheidungsfindung. Ebenfalls unterscheidet sich der Aufwand beider Methoden, da Prototypen im Gegensatz zu Skizzen mehr Zeit benötigen bis sie fertig gestellt sind. Sie besitzen somit auch nicht den Wegwerfcharakter einer Skizze.

Der Unterschied liegt also nicht in der Form der Methoden, sondern eher im Nutzen bzw. in der Bedeutung dieser, welcher durch fließende Übergänge beider Methoden in [Abbildung 2.7](#) anschaulich dargestellt wird ([Sagmeister, 2008](#)).

Per Definition ist eine Skizze ein grobes, ungefähres Design. Es ist lückenhaft, mangelhaft und hat unausgereifte Eigenschaften. Im Gegensatz dazu hilft ein Prototyp, Ideen auszuwerten - entweder durch Absprache mit dem Klienten, die im Zuge dessen den Prototypen ausprobieren, oder durch Usabilitytests. Nicht überraschend, können vorzeitige Usabilityauswertungen von Skizzen bzw. Prototypen signifikante Probleme aufzeigen, die das Design frühzeitig vernichten - vor allem wenn neue Designs mit konservativeren verglichen werden. Das zieht Folgen für die Entwickler und Forscher mit sich ([Greenberg and Buxton, 2008](#)).

#### 2.2.2.6 *Getting the Right Design vs. Getting the Design Right*

Die Kehrseite frühzeitiger Usability Auswertungen von Skizzen animieren Entwickler, jedes Problem in iterativen Schritten zu lösen. Das führt zu »local hill climbing«, in dem viel Aufwand betrieben wird, um das anfängliche Design richtig hinzubekommen (*Getting the Design Right*) (siehe [Abbildung 2.8a](#)).

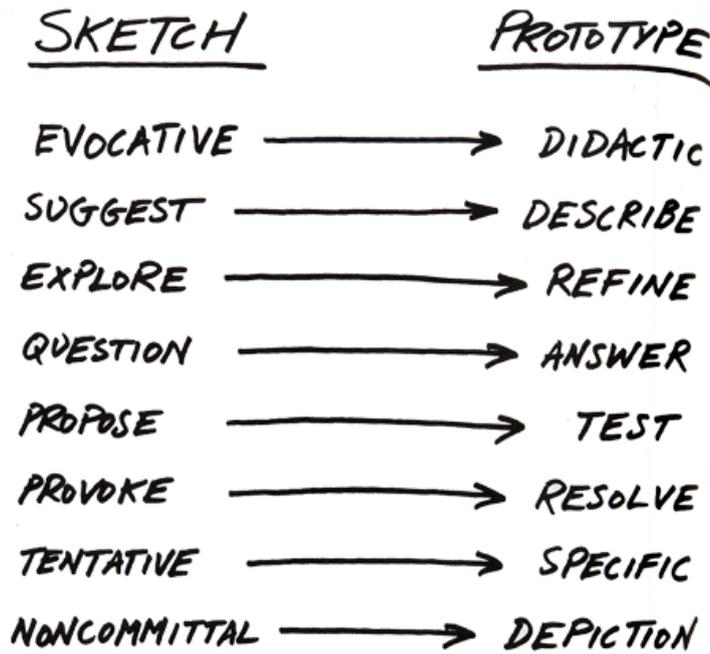


Abbildung 2.7: Übergang von Skizzen zu Prototypen. Der Unterschied von Skizzen und Prototypen liegt nicht in der Form der Methoden sondern im Nutzen bzw. in der Bedeutung. Die Pfeile zeigen, dass es sich hierbei um einen fließenden Übergang handelt.

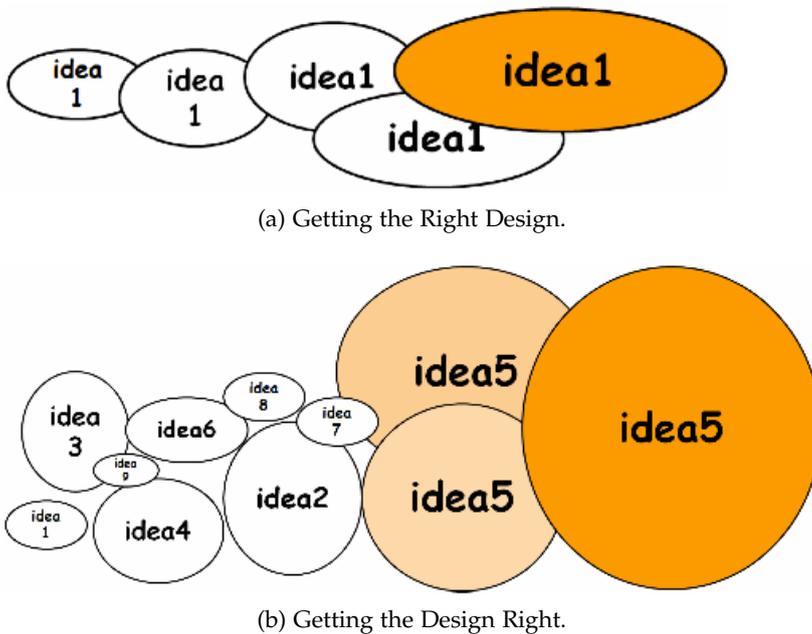


Abbildung 2.8: Getting the Right Design vs. Getting the Design Right. Die richtige Herangehensweise ist das Erstellen von Skizzen, danach das iterative Design, gefolgt von Auswertungen.

Unglücklicherweise blockieren Auswertungen früherer Skizzen Designer oft darin, andere, vielleicht bessere Ideen zu entwickeln. (Greenberg and Buxton, 2008)

Eine Skizze bildet typischerweise nur eines der möglichen Designs und Variationen zum Abwägen ab. Frühes Design benötigt jedoch viele Ideenskizzen, einen Vergleich vieler konkurrierender Ideen und eine Auswahl des aussichtsvollsten Designs (siehe [Abbildung 2.8b](#)). Die aussichtsvolle Idee wird dann weiter abgestimmt und entwickelt, bis man sie als Prototyp verwenden kann. In anderen Worten: Skizzieren dreht sich um das Finden des richtigen Designs (*Getting the Right Design*) (Tohidi et al., 2006; Buxton, 2007; Greenberg and Buxton, 2008).

### 2.2.3 Prototyping

Der abschließende und wichtigste Schritt bevor ein Produkt oder Service eingeführt oder idealerweise mit Benutzern getestet wird, ist die Erstellung eines Prototyps - oder besser mehrerer Prototypen.

*»Prototyping is where, finally, all pieces of the design come together in a holistic unit.«*

(Saffer, 2007)

Viele Auftraggeber haben Schwierigkeiten ein Design zu verstehen, bevor sie nicht einen Prototypen gesehen und benutzt haben. Prototypen sind ein Kommunikationswerkzeug. Sie übermitteln die Botschaft »So könnte das Ganze aussehen und funktionieren«.

Industrial Designer z.B. erstellen in dieser Phase ein glaubwürdiges Modell eines Produktgehäuses, einschließlich Knöpfen, Drehschaltern oder anderer Bauteile, welches so real aussieht, dass es sogar die Auftraggeber täuschen kann. Zusätzlich werden oft Screen Designs, Animationen oder interaktive Demonstrationen erstellt, um Benutzern zu zeigen, wie das Produkt reagieren soll (Vertelney and Booker, 1990).

Welche Form aber nun Prototypen im Allgemeinen annehmen, hängt von den Mitteln ab, die Designern zu Verfügung stehen bzw. von der Art des Produkts um das es sich handelt. Ein Designer mit den richtigen Ressourcen kann *high-fidelity Prototypen* erstellen, welche so aussehen und reagieren als wären sie das fertige Produkt.

Der Zweck von Prototypen ist aber das Erforschen der Eigenschaften des Endprodukts. Ein Prototyp ist vielleicht effizient und ausdruckslos und ein anderer ist wiederum skurril und zugänglich. Der eine ist Menü-basiert und der andere benutzt direkte

Manipulationen. Designer benutzen Prototypen um herauszufinden, welche Funktionen oder Eigenschaften funktionieren, und zwar für sich selbst, die Auftraggeber und die Benutzer (Saffer, 2007). Da Designer nicht unfehlbar sind und somit das erste Design nicht perfekt sein wird, werden die Erkenntnisse, welche aus diesen sog. »*formative Evaluations*« gewonnen werden, zur Erstellung verbesserter Prototypen benutzt. Dieser iterative Prozess aus Verbesserungen läuft so lange, bis klarerweise keine Verbesserungen mehr nötig sind. Oder mit anderen Worten:

»This is exactly how iterative prototyping works: you start somewhere, evaluate it to see how to make it better, change it to make it better and then keep on doing this until it can't get any better.«

(Dix et al., 2004)

Eine Gefahr, die Prototyping mit sich bringen kann, ist die Wahl eines falschen Anfangsdesigns. Wenn man mit einem schlechten Design-Konzept beginnt Prototypen zu erstellen, kann es sein, dass man am Ende nur eine »verschönerte« Version der schlechten Idee bekommt (vgl. 2.2.2.6 [Getting the Right Design vs. Getting the Design Right](#)).

Die Voraussetzungen, um Prototyping Methoden richtig einsetzen zu können sind also:

1. Verständnis über die Elemente, die neu designed gehören und wie man sie verbessern kann und
2. Ein gutes Anfangsdesign als Startprodukt.

(Dix et al., 2004)

Gute Designer schaffen es vielleicht auch durch Erfahrung und Urteilungsvermögen ein passendes Anfangsdesign zu finden, jedoch wird auch ihnen dies durch die Komplexität der Probleme, gerade in Interaction Design, erschwert. Das Ausarbeiten von Skizzen und Modellen, um Probleme aus verschiedenen Blickwinkeln zu beleuchten ist deswegen gerade vor dem Erstellen von Prototypen essenziell. Die meisten Designer arbeiten mit drei Arten von Prototypen: Digitalen Prototypen, physikalischen Prototypen und Prototypen aus Papier. Es soll nun kurz näher auf diese eingegangen und im Besonderen *Paper Prototyping* diskutiert werden.

### 2.2.3.1 Digitale Prototypen

Digitale Prototypen (siehe [Abbildung 2.9](#)) können verschiedenste Formen annehmen, von statischen Bildern bis hin zu komplexen

3D Umgebungen. Sie können in ihrer Funktionalität sehr eingeschränkt sein – z.B. indem sich Benutzer durch eine Reihe an Bildern »durchklicken« - oder im hohen Grade funktionell, so dass User in der Lage sind mit dem System zu arbeiten als wäre es das Endprodukt. Entscheidet man sich für die komplexere Variante, sollten reichhaltig Funktionen vorhanden sein, um das problemlose Interagieren des Users gewährleisten zu können. Es ist schwer für Benutzer sich z.B. ein Werkzeug um Rechtecke zu zeichnen vorzustellen, ohne es dabei wirklich benutzen zu können. Die Gefahr dabei ist jedoch, dass Benutzer und Auftraggeber denken könnten, dass der Prototyp bereits das finale Produkt ist. Fügt man einmal ein visuelles Design hinzu, richtet man seine Aufmerksamkeit auf dieses anstatt auf die Funktionalität (Saffer, 2007).

Seit den 80er Jahren gibt es für beide Formen digitaler Prototypen eine Vielfalt an Programmen zur Erstellung dieser. Seien es Programme wie *MacDraw*, *Protoscreens*, oder *Prototyper* damals, oder Tools wie *Photoshop*, *Shockwave* oder *Visual Basic* heutzutage; sie alle verfügen über die Funktionalität, Prototypen (mit ihrem beschränkten Grad an Nutzen) zu erstellen (Miller-Jacobs, 1991).

Ein Vorteil der daraus entsteht, ist dass digitale Prototypen leicht veröffentlicht werden können. Designer können sie einfach aufs Web stellen oder auf Disks verbreiten, um Benutzern die Möglichkeit zu bieten, die Prototypen egal wo sie sind auszuprobieren.

#### 2.2.3.2 *Physikalische Prototypen*

Physikalische Prototypen (siehe [Abbildung 2.10](#)) können simple Teile eines Designs (wie zum Beispiel einzelne Schalter oder Wahltasten) bis hin zu kompletten physikalischen Umgebungen schaffen. Wie aber auch bei anderen Prototypen kann sich dabei die Funktionalität und Beschaffenheit unterscheiden. Ein Gerät beispielsweise kann aus exakt denselben Materialien wie das Endprodukt erstellt werden (z.B. Plastik oder Metall), oder durch Materialien wie Holz oder Ton angeglichen werden.

#### 2.2.3.3 *Papier Prototypen*

Papier Prototypen sind meistens die schnellste Art um die Funktionalität eines Produkts oder Services zu demonstrieren. Vor allem im Interaction Design kann ein Designer auf Papier leicht einen gesamten Systemdurchlauf erstellen, wobei jedes Blatt Papier einen *Moment* des Designs repräsentiert. Diese Momente könnten eine Webseite, ein Screenshot oder Teil eines Services sein. Zuseher können so Schritt für Schritt den Prototyp durchlaufen indem sie sich in einer gewissen Reihenfolge zwischen

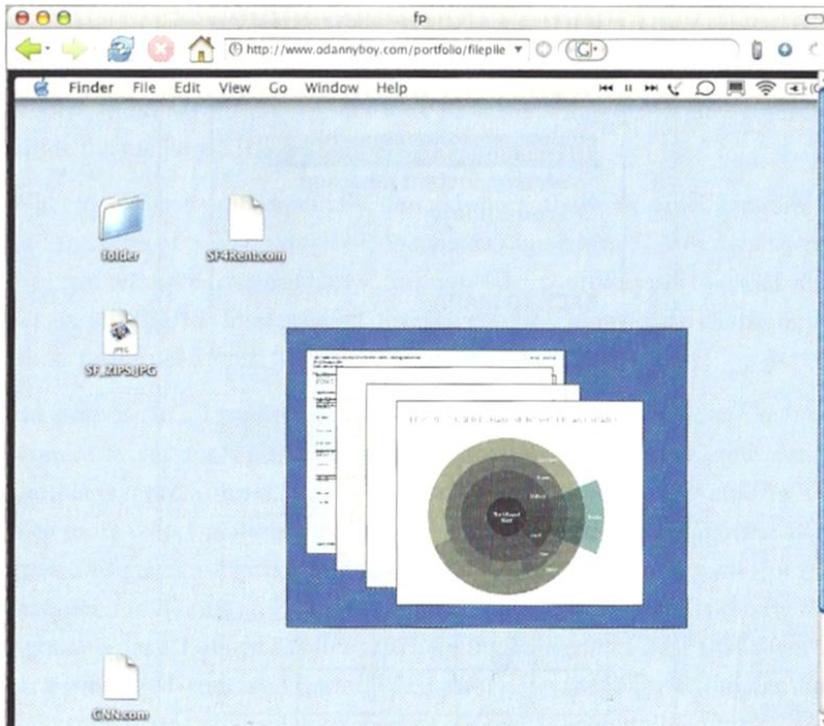


Abbildung 2.9: Digitale Prototypen können Formen in verschiedener Detailstufen annehmen und leicht über das Internet verbreitet werden.

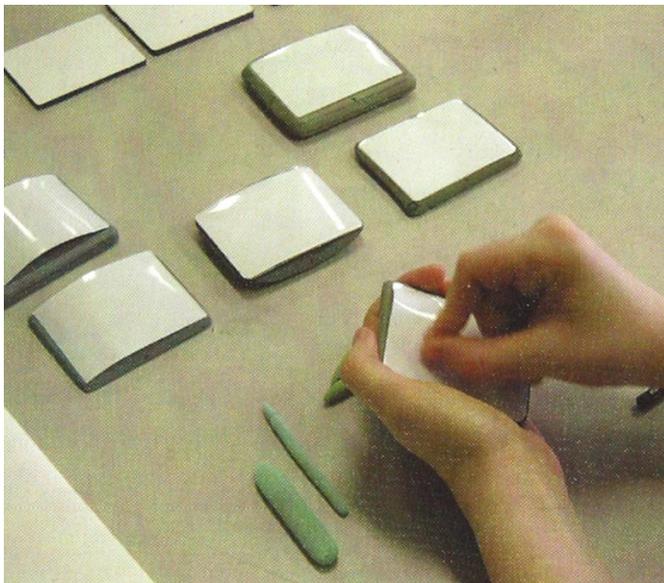


Abbildung 2.10: Physikalische Prototypen können so klein wie ein Schalter sein, oder so groß wie ein ganzer Raum.

den Seiten bewegen. Wie diese Bewegung abläuft, kann auf mehrere Arten erfolgen. Dan Saffer z.B. schlägt vor, die Seiten zu nummerieren und dem User klare Instruktionen zu geben, wie er vorzugehen hat (»Wenn Sie diesen Button drücken, gehen Sie auf Seite 9«). Während des Testens können bei diesem Verfahren Beteiligte, also Benutzer und Designer, Kommentare und Notizen direkt auf dem Prototyp schreiben (Saffer, 2007).

Carolyn Snyder hingegen vertritt eine andere Auffassung von *Paper Prototyping*:

*»Paper prototyping is a variation of usability testing where representative users perform realistic tasks by interacting with a paper version of the interface that is manipulated by a person "playing computer", who doesn't explain how the interface is intended to work.«*

(Snyder, 2003)

Laut ihr hat mindestens eine Person neben dem User zu sitzen und agiert (vollkommen still) als Computer und übernimmt dessen Aufgaben. Kommentare und Notizen machen andere Personen, die ebenfalls still als Beobachter im Raum sitzen. Zusätzlich kann noch ein Moderator hinzu gezogen werden, welcher dem User das Umfeld erklärt bzw. ihm Aufgabenstellungen erteilt.

Im Allgemeinen signalisieren Papier Prototypen, egal nach welcher Methode erstellt, den Benutzern, dass es sich hierbei noch nicht um das Endprodukt handelt und gewähren ihnen somit unterbewusst mehr Freiheiten, Eigenschaften und Funktionsweisen zu kommentieren. Snyder fasst weiters folgende Vorteile zusammen:

Papier Prototypen:

- liefern substanzielles User Feedback in einem frühem Stadium des Entwicklungsprozesses (bevor Bemühungen in die Implementierung gesteckt wurden),
- fördern schnelle iterative Entwicklung, durch Experimente mit mehreren Ideen, anstatt nur mit einer,
- erleichtern die Kommunikation innerhalb des Entwicklungsteams sowie zwischen dem Team und den Benutzern,
- erfordern keine technischen Fähigkeiten, womit eine multidisziplinäre Mannschaft zusammenarbeiten kann und
- regen Kreativität im Produktentwicklungsprozess an.

(Snyder, 2003)

Paper Prototyping folgt somit dem »Maximum Feedback for Minimum Effort« - Prinzip. Wie sieht aber nun ein Papier Prototyp aus und wie wird er erstellt?

Materialien wie einfaches weißes Papier, unlinierte Karteikarten, Marker, Stifte, Scheren, durchsichtiges entfernbares Klebeband, transparente Folien, Permanentmarker usw. dienen als Werkzeuge in Paper Prototyping.

Durch sie werden anfangs Hintergründe bzw. Hintergrundmasken kreiert. Diese können z.B. gezeichnete Repräsentationen eines Browserfensters sein, wenn der Browser mit seinen herkömmlichen Funktionen über den Testverlauf hinweg unverändert bleibt, oder andere statische Gegebenheiten. Ein Hintergrund ist zwar nicht immer notwendig, zeigt dem Benutzer jedoch, dass er es mit einer Repräsentation eines Computerbildschirms oder anderen elektronischen Gerät zu tun hat und sich somit auf diese konzentrieren soll. Es kann manchmal auch helfen - besonders bei kleinen Geräten (z.B. bei PDAs oder Handys) - durch Masken den nutzbaren Platz zu begrenzen, da gerade hier oft jeder Pixel zählt. [Abbildung 2.11](#) zeigt dazu ein Beispiel.

Nachdem die Hintergründe bzw. Masken erstellt wurden, ist es nötig alle Interface Widges, wie *buttons*, *checkboxes*, *dialog boxes*, *text fields*, *lists*, *cursors* etc. zu erstellen. Interface Widges helfen später im User Testing die Interaktivität zu simulieren ([Snyder, 2003](#)).

Sie sollten deswegen so gut wie möglich vorbereitet werden, um lange Wartezeiten im Testing zu vermeiden. Im Allgemeinen ist es natürlich auch erlaubt, Screenshots von Elementen (inklusive Hintergrund) zu machen, jedoch muss man sich dabei immer fragen wie viel man vielleicht nachträglich ändern will. Meistens ist es schneller, Elemente selbst zu zeichnen. [Abbildung 2.12](#) zeigt Vorbereitungselemente aus den verschiedensten Materialien und deren möglichen Einsatzzweck ([Snyder, 2003](#)).

Nachdem ein Prototyp fertig gestellt wurde, sollte er logischerweise auch Einsatz finden. Das richtige Publikum dafür sind die User, die das Endprodukt benutzen werden. Man nennt diesen Prozess auch User Testing, der im folgenden Punkt beschrieben wird.

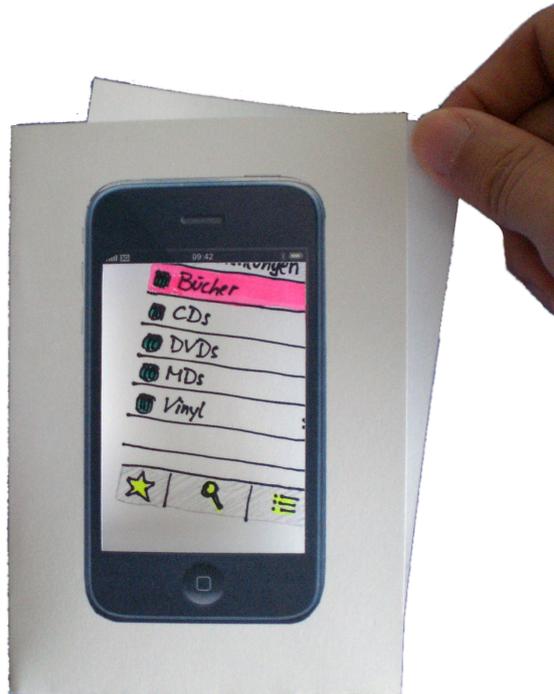
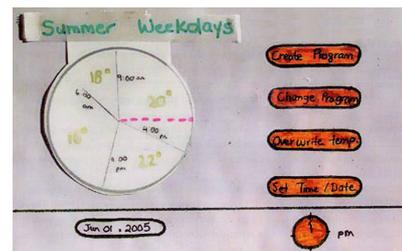


Abbildung 2.11: Hintergrundmasken werden im ersten Schritt von Paper Prototyping erstellt. Besonders bei kleinen Geräten kann es helfen, den nutzbaren Bereich einzugrenzen, da hier meist jeder Pixel zählt.



(a) Papierelemente



(b) Papier Prototyp

Abbildung 2.12: Papierelemente (a) für Interface Widgets können durch einfache Materialien wie Papier und Stifte erstellt werden. Sie sollten vor dem User Testing vorbereitet werden, um lange Wartezeiten zu vermeiden. Ein Papier Prototyp, wie in (b), imitiert zwar nur das Endprodukt, bietet dem User aber die volle Interaktivität. Der Benutzer interagiert dabei direkt mit dem Prototypen als wäre es ein Touch Screen.

### 2.2.4 (User) Testing

Der oft verwendete Begriff User Testing ist eigentlich eine Fehlbezeichnung. Getestet werden natürlich nicht die User sondern das Produkt bzw. das Service. Das Testen mit potentiellen Benutzern hilft, falsche Annahmen, die in der Vorbereitung bzw. Erstellung des Designs gemacht wurden, zu korrigieren. Am besten werden diese durch Beobachtungen und Rücksprache mit den Testpersonen aufgedeckt. Dabei ist es oft am besten, wenn Designer anderen Teammitgliedern oder *Usability Spezialisten* gestatten, die Tests zu leiten, um selbst eine Beobachterrolle einzunehmen. Designer neigen meist dazu, ihr Gegenüber zu belehren (»Schau doch mal zu den Button rechts oben«), weil sie ihr Design besser kennen. Es sollte zusätzlich vermieden werden, dass sich Designer als solche zu erkennen geben, da Testpersonen oft ihr Feedback danach richten bzw. gegebenenfalls ihre Kritik abschwächen. Auch wenn für Designer nichts demütigender ist, als anzusehen wie Testpersonen durch ihr Design stolpern, müssen sie diese Bürde auf sich nehmen, die Testdurchläufe ernst nehmen und nach Verhaltensmustern Ausschau halten. Diese bewegen sie dann dazu, Prototypen zu ändern und sie erneut zu testen. Erfahrene Designer wissen eins mit Sicherheit: Das erste Design ist selten das richtige.

Testings werden am besten direkt im geplanten Umfeld der Endnutzer durchgeführt, es sei denn es handelt sich um ein Service, das eine bestimmte Prototypumgebung erfordert. Digitale und physikalische Prototypen sollten somit auf ihren geplanten Computern und Umfeldern Anwendung finden (vgl. [Abbildung 2.13](#)). Papier Prototypen erfordern eine bestimmte Prototypumgebung, die es erlaubt die Testings auch in Testlaboren o.ä. durchzuführen. Testlaboren haben den Vorteil, dass Designer viele Tests an einem einzigen Tag durchführen können, ohne dabei den Ort wechseln zu müssen ([Saffer, 2007](#)).

Je nach Qualität bzw. Aufwand der für einen Prototypen betrieben wurde, variiert auch der Ablauf eines User Testings. Oft reicht es, wie z.B. bei *high fidelity* Digitalprototypen, dem User lediglich eine Aufgabe zu stellen und ihn dann sich selbst zu überlassen um ihn zu beobachten. Manchmal benötigt man auch einen Moderator, der den User durch den Testablauf leitet. Bei *low fidelity* Prototypen jedoch, zu denen meist Papierprototypen zählen, muss mehr Aufwand betrieben werden. Wie schon im vorigen Punkt erwähnt, ist es notwendig jemanden den Part des Computers bzw. Gerätes übernehmen zu lassen. Bei Paper Prototyping sitzt diese Person meist neben oder gegenüber der Testperson und übernimmt stillschweigend alle interaktiven Tätigkeiten.

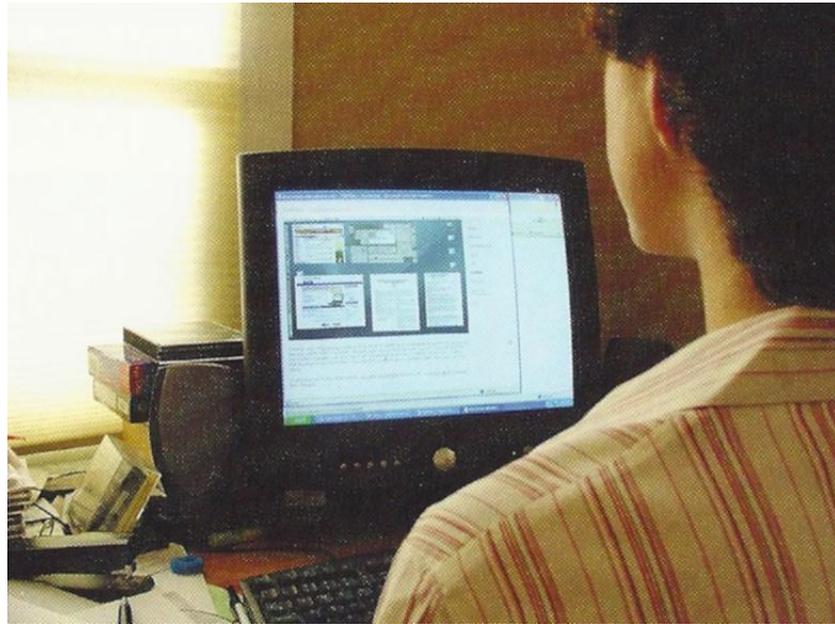


Abbildung 2.13: Testings werden am besten im geplanten Umfeld der Endnutzer durchgeführt. Ein digitaler Prototyp sollte z.B. beim Benutzer zu Hause getestet werden.

Dieses Prinzip ist aber nicht nur eine Eigenheit von Papierprototypen. Eine der ersten Anwendungen fand sich schon 1971 mit dem Testen eines elektronischen Flugticket-Schalters am Chicago O'Hare Airport (Erdmann and Neal, 1971). Da es von echten Kunden, mit echtem Geld zur Anschaffung echter Tickets für damals aktuelle Flüge verwendet wurde, musste es nicht nur von der *user-interaction*-Perspektive, sondern auch von der organisatorischen und finanziellen Seite her arbeiten. Doch da das System noch nicht fertig gestellt war und man es trotzdem einem Test unterziehen wollte, musste eine Person im Hintergrund alle Tätigkeiten, wie am normalen Ticketschalter, managen. Auch wenn es aus Sicht des Kundenservices teuer und ineffizient schien, brachte es wichtige Informationen über das Design, die *Usability* und die Akzeptanz mit sich (Buxton, 2007).

Diese Art des Testens nennt man auch *Wizard of Oz Technique*, die ihren Ursprung in der Kindergeschichte *The Wonderful Wizard of Oz* von L. Frank Baum findet. (siehe [Abbildung 2.14](#)) Sie handelt von einem kleinen Mädchen, Dorothy, welches zusammen mit ihrem Hund Toto von einem Wirbelsturm von ihrer Farm in Kansas in das Land der Munchkins getragen wird. Verzweifelt, mit dem Wunsch wieder nach Hause zu kommen, traf sie auf die Gute Hexe des Nordens, die der gelandeten Dorothy riet, in die Smaragdstadt zu gehen und dort den Zauberer von Oz um Hilfe zu bitten. Mit neu gefundenen Freunden zog Dorothy los und fand tatsächlich den Zauberer, der sich jedem in einer anderen

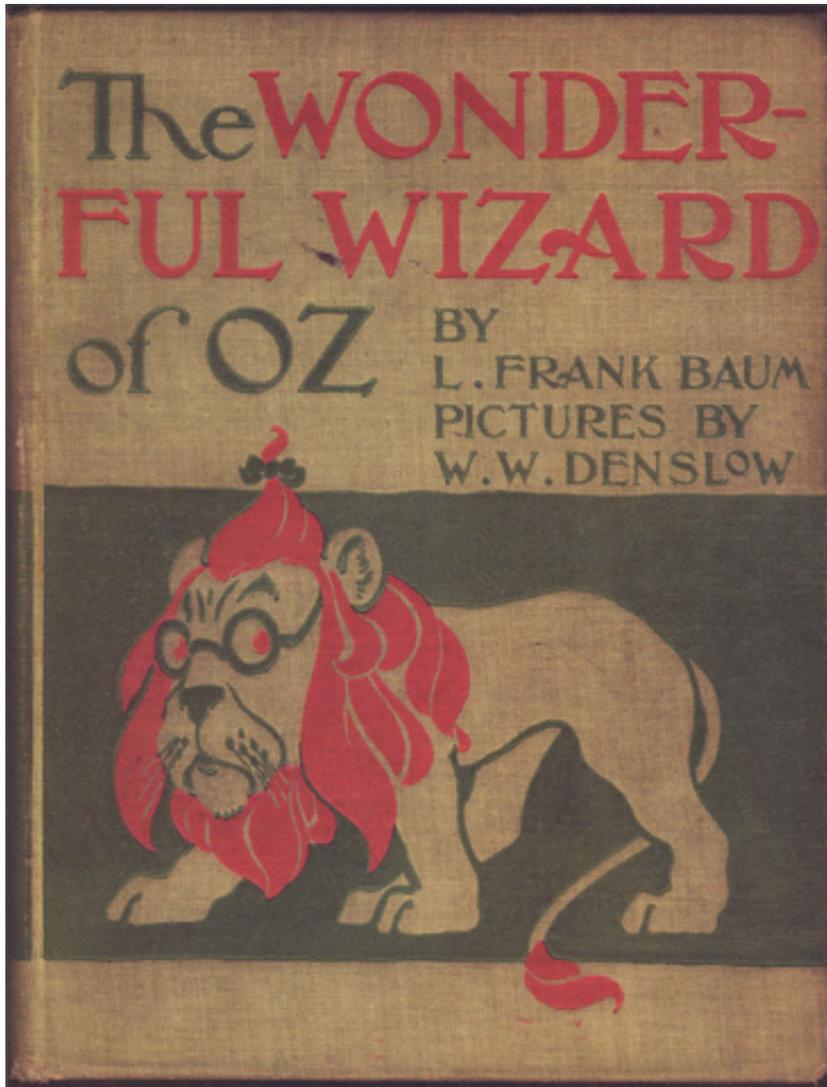


Abbildung 2.14: Das Cover der Erstedition des Buches, das einen unerwarteten Einfluss auf das Thema Interaction Design zur Folge hatte. Die Wizard of Oz Technik beschreibt eine Interaktion zwischen Mensch und Maschine in der eine Person die Reaktionen eines Systems erzeugt.

Gestalt zeigte. Er versicherte zu helfen, unter der Bedingung, dass einer die Böse Hexe des Westens tötet. Nach einigen Abenteuern schafften sie dies auch und kehrten zurück zum Zauberer, um das Versprechen, das er ihnen gegeben hatte, einzulösen. Nach anfänglichem Zögern seitens des Zauberers von Oz, sich zu zeigen, brachten sie ihn schlussendlich dazu und mussten feststellen, dass er gar kein großer Zauberer war, sondern nur ein alter Mann, den durch seine Ankunft mit einem Heißluftballon jeder als Magier ansah. (Baum, 2008)

Dorothy und ihr Hund Toto fanden natürlich einen anderen Weg nach Hause, die Geschichte zeigte uns aber, dass hinter dem großen *Wizard of Oz* ein schüchterner alter Mann steckte, der mit seinem Trick so überzeugend war, dass er alle glauben ließ er sei jemand, der er gar nicht war. Und das ist auch die Quintessenz:

*»To her the Wizard was real, and therefore so were all her experiences.«*

(Buxton, 2007)

Folgt man also dem Beispiel des Zauberers, kann man im User Testing Systeme »herbei beschwören«, die den Benutzern reale wirkungsvolle Erfahrungen bieten, bevor das richtige System im eigentlichen Sinne existiert. Die wichtigsten Punkte, die man dabei beachten sollte, sind folgende:

Eigenschaften von Testsystemen:

- Es ist das Detailreichtum der Erfahrung und nicht die des Prototyps, Skizze oder Technologie, die wichtig ist, im Sinne der Ideenfindung und des Anfangsdesigns.
- Man kann alles verwenden, um diese Erfahrungen »zu beschwören«.
- Umso früher damit begonnen wird, desto wertvoller sind sie.
- Es ist einfacher, billiger, schneller und zuverlässiger einen alten Mann, ein Mikrofon und Lautsprecher zu finden als einen richtigen Zauberer. Dies gilt auch für die meisten Systeme: Fake it before you build it.

(Buxton, 2007)

Die Zielsetzung ist also nicht, das tatsächliche System zu bilden, sondern etwas nach zu ahnen, das Benutzer wirklich erfahren können. Das ermöglicht, Design Konzepte in ihrer Tätigkeit zu erforschen und sie viel früher im Prozess zu erfahren als sonst

möglich. Solch ein System sollte preiswert, schnell zu verwirklichen und verwerfbar sein und nur soviel Detailreichtum wie nötig besitzen, um seinen Zweck zu erfüllen. Es sollte somit alle Eigenschaften aufweisen, die auch Skizzen kennzeichnet, jedoch einer Regel unbedingt unterliegen:

*»Generally the last thing that you should do when beginning to design an interactive system is write code.«*

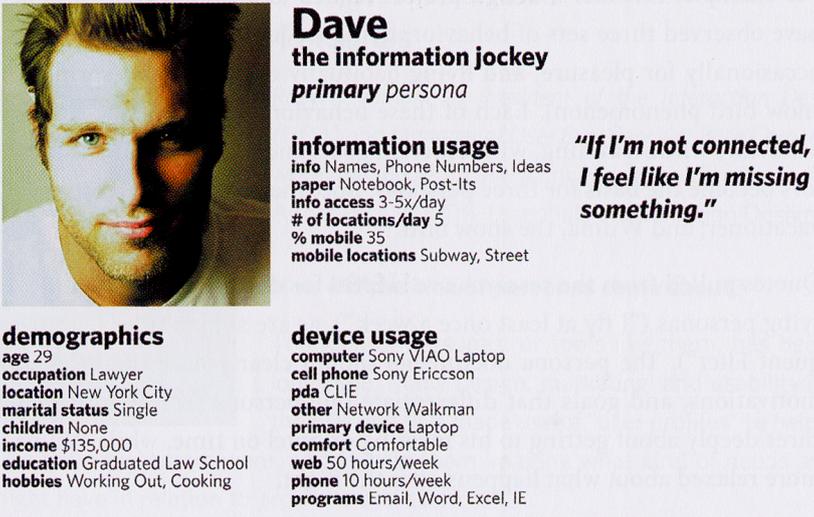
(Buxton, 2007)

Mit den richtigen Testsystemen und klaren Aufgabenstellungen ist es für die Designer einfacher bestimmte Verhaltensmuster aus den Benutzererfahrungen zu erkennen. Oft werden zusätzlich im Laufe des Testings zu den vordefinierten Aufgaben, die Zeit und die Fehlerquote gemessen bzw. der Weg, wie ein User ein Problem löst, aufgezeichnet. Dies soll messbare Qualitätsmerkmale über das Design liefern und bei den abschließenden Rücksprachen mit den Benutzern Orientierungshilfe bieten. Gewöhnlich werden 5-12 Testpersonen in ein Testing miteinbezogen. (Dumas and Redish, 1999) Meist sind es aber weniger, da man sich an einen fixen Projektzeitplan halten muss. Deswegen werden oft auch nur *Quick and Dirty* – Tests mit 1-2 Usern durchgeführt, um schnelles Feedback zu einer Designidee zu bekommen (Sharp et al., 2002).

Skizzieren, das Erstellen von Prototypen und Testen sind wichtige Methoden, um Ideen zu sammeln, sie zu einem Design zusammenzutragen und durch Erfahrungen zu verbessern bevor das endgültige Produkt erstellt wird. Viele Designer arbeiten mit ihnen effektiv und uneingeschränkt, jedoch ist es manchmal hilfreich zu wissen, von wem und wie genau ein Produkt verwendet werden soll. Dies wird nun abschließend im letzten Punkt erläutert.

### 2.2.5 Personas & Szenarien

Der Begriff Personas umfasst eine Sammlung an typischen Personen, die mit einem Produkt oder Service zu tun haben. Sie zeigen Designern, dass sie für bestimmte Personen mit bestimmten Eigenschaften und Vorlieben designen und nicht schlicht für »die Benutzer«. Designer erfinden aus diesem Grund verschiedene Personas mit Hilfe von Beobachtungen und Unterhaltungen mit Usern. Zitate von diesen sind z.B. sehr nützlich, um Personas zu beschreiben (»Ich fliege mindestens einmal pro Woche«), oder um einfache Überschriften (»Der häufige Passagier«) zu finden. Personas fassen also Personen zusammen, die ähnliche Ziele, Beweggründe und Verhalten teilen. Ein Persona Dokument sollte diese Eigenschaften klar aufzeigen, um eine Persona von einer anderen unterscheiden zu können (siehe [Abbildung 2.15](#)).



**Dave**  
the information jockey  
primary persona

**information usage**  
info Names, Phone Numbers, Ideas  
paper Notebook, Post-Its  
info access 3-5x/day  
# of locations/day 5  
% mobile 35  
mobile locations Subway, Street

**demographics**  
age 29  
occupation Lawyer  
location New York City  
marital status Single  
children None  
income \$135,000  
education Graduated Law School  
hobbies Working Out, Cooking

**device usage**  
computer Sony VIAO Laptop  
cell phone Sony Ericsson  
pda CLIE  
other Network Walkman  
primary device Laptop  
comfort Comfortable  
web 50 hours/week  
phone 10 hours/week  
programs Email, Word, Excel, IE

*"If I'm not connected,  
I feel like I'm missing  
something."*

Abbildung 2.15: Personas verwandeln »die Benutzer« in identifizierbare menschliche Wesen.

Im Allgemeinen sollte man die Anzahl an Personas gering halten – am besten zwischen eins und sieben. Nach ca. sieben Personas fällt es einem schwer, sich an alle zu erinnern bzw. sie unterscheiden zu können. Vor allem wird es mit zunehmender Anzahl immer schwieriger ein Design zu finden, welches allen Ansprüchen genügt. Man muss sich deswegen stets darum bemühen sich an das Kernverhalten zu richten und nicht an alle verschiedenen Facetten.

Nachdem eine Sammlung an Personas zusammengestellt wurde, sollte man ein Foto für jede finden. Bilder helfen mehr als alles andere, eine persönliche Note hinzuzufügen und um sich leichter an diese zu erinnern. So lange Personas nicht veröffentlicht werden, können Fotos auch von Services wie z.B. *Yahoo Personals* oder anderen Internetseiten verwendet werden. Personas an sich sind schlicht unbrauchbar, wenn sie nicht zusammen mit Szenarien eingesetzt werden. In Szenarien können sie benutzt werden, um Systemeigenschaften auf Nutzen und Eignung zu testen. Während viele Designer Personas nützlich finden, lehnen sie manche gänzlich ab. Diese Designer sehen Personas als eine Art künstliche Blockade zwischen dem Produkt und dessen Benutzer. Personas können jedoch, wenn sie auf Forschungen basieren und die richtigen Charakteristiken (Verhalten, Motivationen und Ziele) zeigen als wertvolles Tool dienen. Ob man Personas nun verwenden will, hängt schlussendlich vom Designer und der Größe des Projektes ab. (Saffer, 2007)

*Szenarien* hingegen bieten einen schnellen und effektiven Weg, um sich Design Konzepte vorzustellen. Man kann sie sich als Prototypen, welche lediglich aus Wörtern bestehen, vorstellen,

da sie simple Geschichten bilden, welche fertig gestellte Produkte beschreiben. Die zentralen Darsteller dieser Geschichten sind Personas, die durch sie einen Kontext erhalten und zum Leben erwachen. Das Durchlaufen eines Szenarios mit verschiedenen Personas eignet sich somit perfekt zur Darstellung, Analyse und Planung der Auswirkungen eines Systems auf das Verhalten der User. Man betrachte folgendes Beispiel eines Szenarios:

*»Sarah logs onto her BigGrocery account. She sees her order from last week and decides to use it again for this week's order. She removes a few items by dragging them off her BigGroceryList. Her total cost adjusts appropriately. She has all the groceries she wants now, so she clicks the Deliver button. Her saved credit card account is charged, and her confirmation page tells her to expect the groceries in about an hour.«*

(Saffer, 2007)

Dieses Szenario nahm nur ein paar Minuten in Anspruch, um es zu schreiben, aber es würde Stunden benötigen, um ein Storyboard<sup>2</sup> zu erstellen, Tage, um es realistisch darzustellen und Wochen um einen Prototypen anzufertigen. Mit der Hilfe von Szenarien sind Designer in der Lage, mit Wörtern zu skizzieren. (Saffer, 2007)

Szenarien müssen aber nicht unbedingt textlich verfasst werden. Es können auch Storyboards, Videonachbildungen oder reale Situationen erfunden werden, um bestimmte Benutzeraktivitäten zu unterstützen. Der Detailreichtum kann ebenfalls variieren. Das obige Szenario ist z.B. ein sehr allgemein gehaltenes, welches viel detaillierter verfasst hätte werden können. Ein detailliertes Szenario beschreibt z.B. zusätzlich die Systemfunktionalität und dessen Systemkomponenten wie Hardware, Software oder die User-Interface Elemente, die auftreten.

Im Allgemeinen beschreiben Szenarien aber konkrete Anwendungsfälle und konzentrieren sich dabei auf die Fragen: »Was passiert?«, »Wie passiert es?« und »Warum passiert es?« Sie befassen sich also mit Tätigkeiten zukünftiger Benutzer, was gleichzeitig die Designentwicklung antreibt und somit das Endergebnis bereichert. Szenarien sind deswegen oft offen und lückenhaft beschrieben, um den Entwicklern zu helfen mehr zu hinterfragen und damit Möglichkeiten zu eröffnen. (Carroll, 1995)

<sup>2</sup> Storyboards haben ihren Ursprung in der Filmindustrie. Sie bestehen aus Bildern (potentiellen Screenshots) und dazugehörigen Beschreibungen. Zusatzinformationen über den Ablauf werden durch Bildübergänge erreicht. »Storyboarding is a way to look at the film without spending a lot of money . . . It's not the ultimate film, but it represents a first chance to look at it« – Production Illustrator Marty Kline (Braa and Ruvik, 1989)

Susanne Bødker fasst Szenarien wie folgt zusammen: Szenarien sind ein Weg, um Anwendungen und deren Nutzen für User zu beschreiben und konzentrieren sich dabei auf die Tätigkeiten (inklusive deren Reihenfolgen) der User und das daraus resultierende Feedback des Produkts. Szenarien helfen, die Arbeitspraxis der Benutzer zu reflektieren und die Änderungen dieser beim Einsatz neuer Artefakte aufzuzeigen. (Bødker, 1991)

Ob und wie Szenarien benutzt werden, hängt wie bei Personas von den Designern ab. Tatsache ist jedenfalls, dass heutzutage Szenarien helfen und meist sogar benötigt werden, um User Trainings, Dokumentationen und User Tests zu konzipieren (Carroll, 1995). Da sie aber meist unvollständig und lückenhaft sind, sollten sie ständig hinterfragt und vervollständigt werden.

### 2.3 SPEZIELLE EIGENSCHAFTEN

Durch das Anwenden von Designmethoden bilden sich gezwungenermaßen nachweisbare Verhaltensmuster. Sie formen spezielle Eigenschaften, die Designer und ihre Arbeit charakterisieren.

#### 2.3.1 Designvokabular

In einer Studie über Industrial Designer, die an Konzeptskizzen arbeiteten, fanden Pan et al. (Pan et al., 2002) heraus, dass die Designer die Formen ihres Designs auf verschiedene Weisen verbal beschrieben. Ihre Sprache war nicht klar, einheitlich oder im Allgemeinen verständlich für Außenstehende. Designer haben ein »creative vocabulary, which has rich meanings in design communication« (Pan et al., 2002). In Hinsicht auf globale Kollaborationen zwischen einzelnen Arbeitsgruppen ist der Wunsch nach einem gemeinsamen Designvokabular naheliegend. Laut Hill (Hill et al., 2001) baut der Erfolg von Designteams stark auf die folgende Fähigkeit der Teilnehmer auf: »negotiate different design perspectives and specialities«. »Similarities in voice« ist das Um und Auf bei Teammitgliedern mit verschiedenen Disziplinen und Hintergründen.

Das Definieren eines gemeinsamen Designvokabulars scheint aber doch eher utopisch. Bucciarelli meinte dazu, dass, auch wenn Teilnehmer die gleiche gemeinsame Sprache (wie z.B. Englisch) verwenden, die Sprache auf verschiedene Weisen angewendet werden kann, wodurch es scheint, als würde ein Teilnehmer eine andere Sprache sprechen:

*»Not different in the sense that for you, as a foreigner, a translation would make meanings clear ... but different in that the concepts and ideas and relationships among the*

*things of an object world require new leaning, like the learning of a foreign language.«*

(Bucciarelli, 2002)

Im Grunde bedeutet dies für Designteams (speziell für globale), dass sie nicht nur eine sprachliche Hürde von z.B. Personen unterschiedlicher Herkunft meistern müssen, sondern auch von Personen, die von komplett unterschiedlichen Objektwelten stammen - abhängig von kulturellem Hintergrund, der Ausbildung und professionellen Disziplinen.

Larsson hält *Negotiation* als adäquateren Begriff, um die Kommunikation zwischen den einzelnen Mitgliedern eines Designteams zu beschreiben, da die Teilnehmer aktiv den Kontext und Inhalt der Situationen formen und die Informationen nicht passiv mit eindeutigen Bedeutungen und definierten Settings übertragen. (Larsson, 2003)

Die Wissenskonstruktion ist ein kollaborativer Prozess, der wie bereits Bucciarelli beobachtete, aus einer inhärenten Mehrdeutigkeit besteht, welche Designer vor die folgende, alles andere als einfache Aufgabe stellt:

*»frequently bringing the results of their object world efforts, which no doubt will conflict, into coherence if design is to proceed - and they must do this without a shared proper language.«*

(Bucciarelli, 2002)

#### 2.3.1.1 Storytelling

Eine Art, um Verständnis unter Designern zu vermitteln, ist das Erzählen von Geschichten. Geschichten ermöglichen Designern gezielter zu erklären, was sie gerade machen im Gegensatz zu dem, was sie machen sollten - falls sie sich an normative Designmodelle halten. Auch wenn die Geschichten nicht direkt ein Problem lösen, helfen sie dennoch, ein gemeinsames Vokabular aufzubauen, um über das Design sprechen zu können. Sie kreieren *»a new language capable of describing aspects of the evolving design«* (Lloyd, 2000).

Vereinbarte Geschichten ermöglichen Designer verschiedener Sparten, ein gemeinsames Verständnis zu kreieren, auf das sie sich beziehen können, ohne dabei konkreten Thematiken zustimmen zu müssen. Es sind konkrete Beispiele, auf die sich Personen mit verschiedenen Hintergründen beziehen können. Da uns das formlose Naturell von Geschichten dazu bringt sie nicht zu ernst zu nehmen, dienen sie als eine gemeinsam verfügbare Wissensquelle - die Freiraum für Interpretation und Fragestellungen aller

Beteiligten zulässt (Erickson, 1996). Wie in Orrs Studie über Service Techniker (Orr, 1986) beobachtet wurde, dienen Geschichten auch dem Zweck, Erfahrungen zu erhalten und zu verbreiten. Somit bieten sie Grundgerüst für Problemlösungsaktivitäten.

Zusätzlich bilden sie ein Transportmittel für Gedanken, da sie durch ihre Nutzung nicht nur anderen etwas beschreiben, sondern auch uns selbst (Norman, 1994).

### 2.3.1.2 *Indexical Expressions*

Neben Geschichten, die nützlich bei der Verständnisvermittlung sind, benutzen Designer auch *Repräsentationen*, wie Markierungen oder Symbole, die etwas Bestimmtes darstellen und ihnen bei ihren Argumentationen helfen (Norman, 1994). Repräsentationen bieten auf die gleiche Weise wie Geschichten konkrete Beispiele, welche die Gedanken der Designer abbilden. Dabei benutzen sie vorhandene Tools, die die Verständnisvermittlung zwischen den Teammitgliedern im jeweiligen Moment am besten unterstützen.

In Hinsicht auf ein gemeinsames Designvokabular muss man zur Kenntnis nehmen, dass einige Teile des Vokabulars keine strikten Definitionen haben. Designer und auch andere Personen benutzen *Indexical Expressions* wenn sie kommunizieren. Eines muss man sich dabei aber vor Augen halten:

*»[Indexical Expressions] cannot straightforwardly be repeated or reused outside the context in which they originated, without changing their meaning.«*

(Kristoffersen and Ljungberg, 1999)

*Markierungen und Symbole verlieren außerhalb des Kontextes ihre Bedeutung.*

Beispielsweise können Wörter wie »hier/dort«, »dies/das«, »sein/seine« oder »jetzt/dann« nicht angemessen gedeutet werden, ohne zu wissen unter welchen Umständen sie benutzt wurden (Larsson, 2003).

Designvokabular dient aber nicht nur zur Verständigung, sondern ist vielmehr ein Hilfsmittel, das Designer bei ihren Arbeitsmethoden begleitet und mit diesen zusammenspielt.

### 2.3.2 *Der kollaborative Charakter*

Designkollaboration wird durch die soziale Welt geformt. Aus diesem Grund ist es unmöglich, Designspezifikationen und Artefaktbeschreibungen zu interpretieren, ohne die soziale Situation, in der sie erstellt wurden, zu verstehen (Brown and Duguid, 2000). Da Design immer mit Kommunikation und Interaktion - zwischen Einzelpersonen und Gruppen in komplexen sozialen

Situationen - zusammenhängt, können die technischen Ergebnisse nicht vom sozialen Charakter der Designaktivitäten getrennt werden. Er ist durchgehend in Meetings, Diskussionen, Argumenten, Debatten und Interpretationen vorhanden und macht das dichtverflochtene Gewebe von Design aus. Minneman beschreibt in diesem Zusammenhang Design als »*social construction of a technical reality*« (Minneman, 1991).

Kollaborative Wissenskonstruktion ist wichtig für ein erfolgreiches Design. Zudem ist es auch wichtig, Mehrdeutigkeiten zu bewahren, um Teammitgliedern Freiheiten zu ermöglichen.

*»[It is important to provide team members with] the freedom to manoeuvre independently within objects worlds and providing room for the recasting of meaning in the negotiations with others.«*

(Bucciarelli, 1994)

Eine erfolgreiche Kollaboration zeichnet sich durch die Bildung eines gemeinsamen Verständnisses aus. Clark und Brennan benutzen den Begriff *Grounding*, um diesen Vorgang zu beschreiben, in dem das Kommunizierte auch verstanden wird. Sie erklären weiters: »*all collective actions are built on common ground and its accumulation*«, womit sie betonen, dass Gemeinsamkeiten kontinuierlich von Moment zu Moment durch die Interaktion der Teammitglieder aufgebaut werden (Clark and Brennan, 1991). Diese Interaktionen treten in verschiedenen Formen auf und beeinflussen den kollaborativen Designprozess. Short et al. nennen in (Short et al., 1991) beispielsweise Aspekte der visuellen Kommunikation, welche soziale Interaktionen beeinflussen:

*»In normal face-to-face interaction, the participants exchange in addition to the verbal material, a range of non-verbal cues such as facial expression, direction of gaze, posture, dress and physical distance.«*

(Short et al., 1991)

Um also ein Verständnis über Designkollaboration zu bekommen, müssen Gemeinsamkeiten in Hinsicht auf Relevanz und Bedeutung der in kollaborativen Designaktivitäten vorgebrachten Informationen erlangt werden (Hill et al., 2001).

## ZUSAMMENFASSUNG

Design ist ein alltäglicher und weitgehender Begriff, der sowohl Bedeutung als Tätigkeit und als Repräsentation in sich trägt. Es gibt viele verschiedene Designdisziplinen, welche zur Unterscheidung von Kompetenzen professioneller Designer dienen. Ein wichtiger Bereich für Softwaredesign ist (User) Interface Design bzw. Interaction Design, welche sich besonders mit den subjektiven und qualitativen Aspekten bei der Interaktion von Menschen und Computern beschäftigen. Es gibt verschiedenste Methoden, die benutzt werden können, um Designprobleme zu verstehen und um nötige Lösungsstrategien zu entwickeln. Skizzen bieten durch ihre minimale Detailstufe und die kurze Herstellungszeit Mehrdeutigkeiten, die zum Weiterforschen und Kommentieren anregen. Dieser soziale Prozess ist wichtig in der Ideenfindung am Anfang einer Designphase. Prototypen verbinden alle Teile eines Designs zu einer Einheit und verschaffen im (User) Testing neue Einblicke. Beobachtungen von Verhaltensmustern innerhalb der Testings bewegen schlussendlich dazu das Design iterativ zu überarbeiten. Es kann zusätzlich auch nützlich sein, sich durch Personas und Szenarien eine Art Leitfaden zusammen zu stellen, der durch das Beleuchten von Tätigkeiten potentieller Benutzer zustande kommt. Dieser sollte aber ständig vervollständigt werden, da er meist Lücken aufweist. Design hängt immer mit der Kommunikation und Interaktion von Personen zusammen und ist somit eine »soziale Konstruktion einer technischen Realität«. Das folgende Kapitel soll nun näher auf den kollaborativen Charakter eingehen.

Kollaborative Aktivitäten sind ein wichtiger Anwendungsbereich für elektronische Technologien (Kvan, 2000). Designer arbeiten immer häufiger nicht mehr alleine für sich, sondern mit anderen Designern, aber auch Leuten aus anderen Disziplinen zusammen. Da komplexe Probleme mehr Expertise und Wissen erfordern, als eine einzelne Person haben kann, sind Teilnahme, Kommunikation und Kollaboration seitens aller involvierten Akteure notwendig. Die intellektuelle, kulturelle und soziale Vielfalt, die dem Projekt dadurch zugutekommt, ist eine wichtige Quelle für soziale Kreativität (Fischer, 2005).

Für das Verständnis von kollaborativem Design ist es wichtig, die Bedeutung von Kollaboration zu erörtern. Es folgen nun verschiedene Aktivitäten, die gemeinhin unter dem Begriff Kollaboration zusammengefasst werden. Zur Lösung von gegebenen Problemen, beispielsweise dem Design eines neuen Produkts, bilden Individuen Teams und Gruppen, um voneinander zu profitieren (Steiner, 1972). Von kollaborativem Erfolg kann dann gesprochen werden, wenn die Gruppe etwas erreicht, das ihre Mitglieder alleine nicht hätten erreichen können. Kollaboration kann als gemeinsames Lösen von Problemen verstanden werden: Individuen haben ein oder mehrere gemeinsame Ziele und arbeiten zusammen, um befriedigende Lösungen zu finden. Thomas Kvan definiert den Begriff Kollaboration durch Bezugnahme auf das Oxford Englischwörterbuch:

»Indeed, the Oxford English Dictionary defines collaborate as “to co-operate, especially in literary, artistic or scientific work”, deriving from the Latin words *col labore*, to work along side one another.« (Kvan, 2000)

Weiters schreibt er, dass kollaboratives Design eine weitaus anspruchsvollere Aktivität ist, als einfach nur ein Projekt in einem Team abzuschließen. Es muss ein gemeinsames, ganzheitliches und kreatives Resultat produziert, nicht bloß eine gewisse Arbeit erledigt werden (Kvan, 2000).

Der folgende Abschnitt behandelt die sozialen Aspekte und Probleme, die kollaboratives Design mit sich bringt.

### 3.1 SOZIALE ASPEKTE & PROBLEME

Kollaboratives Design von mehreren Personen aus verschiedenen beruflichen Disziplinen birgt sehr viele und komplexe soziale Aspekte. Jede dieser Personen hat einen eigenen Charakter, verschiedene kulturelle, soziale und politische Ansichten und vor allem eigene Interessen und Sichtweisen.

**Abbildung 3.1** zeigt den sogenannten »Tree Swing«<sup>1</sup> Sketch, der auf witzige Weise die Problematik von kollaborativem Design aufzeigt. Man sieht darin die verschiedenen Visionen einer Baumschaukel der jeweiligen Teammitglieder. Jede einzelne Darstellung der Schaukel stereotypisiert eine ins Projekt involvierte Rolle (Kilker, 1999).

Designprojekte im technologischen Bereich umfassen immer mehr Teammitglieder aus unterschiedlichen Berufsgruppen. Egal ob diese Personen in einem Meeting zusammensitzen oder ihre Arbeit per Videokonferenz erledigen, sie alle haben verschiedene Perspektiven auf Projekt, Anforderungen und Lösung. Jeder muss seine eigene Sichtweise argumentieren und versuchen, so gut wie möglich durchzusetzen, ähnlich dem Sketch in **Abbildung 3.1**. Julian Kilker bezeichnet diese verschiedenen Rollen als »design identities«<sup>2</sup>. Jede dieser »design identities« hat eine eigene Meinung, eigene Standpunkte und eine eigene Einstellung gegenüber Technologie und Design. All diese Faktoren beeinflussen das Design, den Prozess und das Endprodukt der Projektarbeit (Kilker, 1999).

Die Theorie der sozialen Identität ist eine sozialpsychologische Perspektive, die erläutert, wie Menschen soziale Identitäten (beispielsweise »Entwickler«, »Manager« oder »Programmierer«) annehmen, halten und verwenden (Hogg and Abrams, 1988).

Aufbauend auf Sherifs Untersuchungen der realistischen Konflikttheorie (Sherif, 1966), ist die Theorie der sozialen Identität nützlich für die Erforschung der Art und Weise, wie Individuen in Gruppen zusammenarbeiten (Kilker, 1999). Personen sind Teil einer sozialen Gruppe, wenn sie eine gemeinsame Definition ihrer selbst als Teil dieser Gruppe vertreten (Hogg and Abrams, 1988).

Sherif hat Konflikte in Gruppen und Teams ausgiebig untersucht. Dabei hat er für seine Zwecke die Testpersonen so in Gruppen aufgeteilt, dass ein möglichst kompetitives Klima innerhalb des jeweiligen Teams entstand (Sherif, 1966). Die Theorie der sozialen Identität fokussiert im Gegensatz zu diesen designierten Teams auf möglichst realistische Gruppenkonstellationen.

<sup>1</sup> zu Deutsch: Baumschaukel

<sup>2</sup> zu Deutsch: Designidentitäten

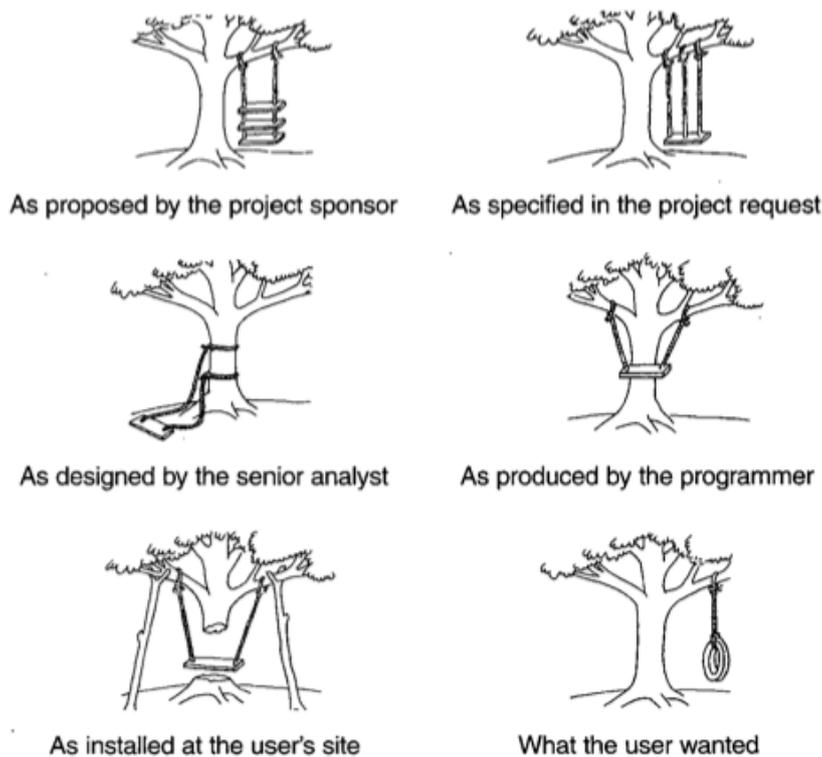


Abbildung 3.1: Der »Tree Swing« (Baumschaukel) Sketch zeigt die Fallen, in die man bei kollaborativem Design laufen kann. Wenn jeder eine andere Vorstellung vom Produkt hat, dann leidet das Endresultat.

Idealisierte soziale Identitäten spielen bei Interaktionen innerhalb einer Gruppe eine wichtige Rolle. Menschen brauchen sie, um sich selbst und andere zu definieren und Vergleiche herzustellen. Jeder hat ein stereotypes Bild der einzelnen involvierten Berufsdisziplinen im Kopf, basierend auf Bildung und Erfahrung. Der Theorie der sozialen Identität zufolge, betonen Leute im eigenen Team Ähnlichkeiten zwischen sich und den Kollegen, gegenüber anderen Gruppen heben sie jedoch Unterschiede hervor. Generell kommen stereotype Sichtweisen eher in multidisziplinären (heterogenen) Teams zum Vorschein, als in homogenen Konstellationen.

Im sozialen Wettbewerb tendieren soziale Identitäten zur Polarisierung. Die Individuen ordnen sich und den anderen eine gewisse Identität und deren Ideale zu und kategorisieren damit jedes Gruppenmitglied. Solche polarisierenden Identitäten sind beispielsweise Liberale oder Konservative in politischer Hinsicht, Vertreter von qualitativen oder quantitativen Methoden in wissenschaftlicher Hinsicht oder die stereotypischen Interpretationen von PC oder Mac Benutzern.

Stereotype sind von Natur aus simplifiziert und stilisiert. Kein Mensch entspricht genau so einem verzerrten Bild, denn jeder hat Ansichten, Ideale und Eigenschaften, die sich davon unterscheiden.

Obwohl es im Bereich Design viele unterschiedliche soziale Identitäten gibt, werden im folgenden zwei typische Identitäten behandelt: eine vertritt technologiezentrierte, die andere sozialzentrierte Ansichten und Methoden. Es werden die Kollaboration und die damit verbundenen Implikationen zwischen diesen beiden Charakteren geprüft (Kilker, 1999).

Die technologiezentrierte Identität bevorzugt ein gut geplantes, systematisches und lineares Design, das Pahl und Beitz als »systematischen Ansatz« bezeichnen (Pahl and Beitz, 1988), während die sozialzentrierte Identität benutzerzentriertes und partizipatives Design, sowie eine iterative Vorgehensweise vorzieht.

Tabelle 3.1 zeigt die unterschiedlichen, polarisierenden Ideale dieser beiden Designidentitäten nach Kilker (Kilker, 1999). Donald Norman nennt diese auseinander laufenden Perspektiven »menschenzentriert« und »maschinenzentriert«. Erstere ordnet den Menschen der Maschine über, denn er ist der Maschine durch Kreativität, Einfallsreichtum und Flexibilität überlegen. Computer hingegen sind dumm, einfallslos und starr. Die maschinenzentrierte Perspektive sieht den Menschen dagegen als unorganisiert, emotional und unlogisch, während Maschinen als geordnet, objektiv und logisch betrachtet werden (Norman, 1994).

Neben der stereotypischen Sichtweise der Kollegen im Projekt, hat auch jeder Designer ein stereotypisches Bild vom designierten

FAKTOR	TZI	SZI
Charakteristiken guten Designs	Fokus auf technologische Charakteristiken: Effiziente »clevere« Lösung, geringer Instandhaltungsaufwand. Das beste Design kann quantitativ ermittelt werden und das Ziel ist es, dieses zu implementieren	Fokus auf soziale Charakteristiken: Funktionierte in der gegebenen sozialen Umgebung und befriedigt die Bedürfnisse der Benutzer. Es gibt mehrere Perspektiven von gutem Design und das Ziel ist es, die beste Kombination zu finden.
Designprozess	Systemorientiert, basierend auf Spezifikationen	Iterativ, benutzerorientiert. Kollaboration verbessert Design.
Verantwortung der Designer	Designer sind nicht verantwortlich für soziale Verwendung von Technologie, die Verantwortung liegt beim Benutzer.	Designer sollen den sozialen Kontext der Technologie berücksichtigen.
Die Rolle von Evaluierung in Design	Quantitativ: gibt an, ob die Implementierung die Anforderungen der Spezifikation erfüllt. Benutzer gehen nicht rein logisch vor, ihre Evaluierung ist oft widersprüchlich und bringt wenig.	Qualitativ und ethnographisch: zeigt Diskrepanzen zwischen Spezifikation und Bedürfnissen der Nutzer auf. Die Benutzer haben höchste Priorität und sollten deshalb in den Designprozess einbezogen werden.
Wie wird das andere Ideal wahrgenommen?	Sozialzentrierten Idealen fehlt es an konkreten Zielen. Befürworter verlangen oft substanzielle Designänderungen aufgrund von Trivialitäten.	Technologiezentrierte Ideale sind starr und unflexibel. Befürworter sind zwar technisch kompetent, weigern sich aber, soziale Aspekte zu berücksichtigen.

Tabelle 3.1: Die unterschiedlichen Designideale von technologiezentrierten und sozialzentrierten Identitäten.

TZI = Technologiezentriertes Ideal

SZI = Sozialzentriertes Ideal

Endbenutzer. Problematisch ist jedoch bereits die Begrifflichkeit »der (End)benutzer«, da Technologien von sehr vielen und sehr unterschiedlichen Individuen in verschiedenen Kontexten genutzt werden (Agre, 1995). Daher muss der Begriff »Benutzer« explizit und implizit während des Designprozesses erörtert und ein authentisches Bild vom tatsächlichen Endbenutzer geschaffen werden. Designer können dies direkt, durch Diskussion mit und Observation von potenziellen Nutzern oder indirekt und hypothetisch, durch Stereotypisierung von Szenarios und Fallstudien erreichen.

Eine hypothetische Benutzergruppe wäre beispielsweise der »naive user«, der, laut Annahme, der Technologie komplett fremd ist. Technologiezentriertes Design würde davon ausgehen, dass dieser Benutzer in der selben Welt wie der Programmierer lebt und dass rationale und formale Methoden eingesetzt werden können, um die Interaktionen zwischen diesem Menschen und der Maschine zu modellieren. Das Gegenlager, sozialzentriertes Design, würde mit der Annahme starten, technische Systeme könnten nicht ohne ein umfassendes Verständnis des sozialen Kontexts entwickelt werden und die Zielgruppe müsste aktiv auf die neue Technologie vorbereitet werden.

Soziale Identitäten sind in zweierlei Weisen eine Herausforderung bei der Kollaboration in einem Team: zum einen ist es für die involvierten Personen schwierig, ihre unterschiedlichen Perspektiven und Meinungen zu vereinbaren, und zum anderen hat auch jeder der Beteiligten eine andere Vorstellung davon, wie der Designprozess und die Kollaboration selbst ablaufen sollen.

Jeder Mensch ist immer Teil von mehreren sozialen Gruppen. In einem kollaborativen Designkontext ist jeder ein Teil des Projektteams und zusätzlich noch Teil seiner Berufsgruppe und des damit verbundenen sozialen Kontexts. Der Zwiespalt zwischen diesen beiden Gruppen kann dazu führen, dass die Qualität der Zusammenarbeit leidet. Wenn Teammitglieder zu sehr die Rolle ihrer Berufsgruppe einnehmen, kann das in festgefahrenen Diskussionen resultieren (Kilker, 1999). Wird hingegen die Identität als Teammitglied zu sehr in den Vordergrund gerückt, kann dies dazu führen, dass die eigene Perspektive nicht mehr genügend eingebracht und dadurch wertvoller Input verloren geht (Irving, 1982).

Sherif hat durch seine Forschung herausgefunden, dass die Kollaboration zwischen polarisierenden Identitäten durch die Einführung eines gemeinsamen Gegners, oder schärfer gesagt, durch die Einführung eines Feindes angekurbelt werden kann. Besser ist es hingegen, übergeordnete Ziele für die Gruppe einzuführen, deren Erreichung für das Team äußerst reizvoll ist und die für das einzelne Individuum absolut unerreichbar sind (Sherif, 1966).

Viele Mitglieder in Designteams sind in kollaborativem Arbeiten ungeübt. Dieses Problem rührt daher, dass technische Ausbildungen darauf abzielen, optimale Lösungen zu finden, anstatt auch Kompromisse und Mittelwege in Betracht zu ziehen. Sozialzentrale Ausbildungen lehren hier eine andere Vorgehensweise. Sie propagieren während kollaborativer Arbeit auch zu diskutieren und zu verhandeln, das technologiezentrierte Lager empfindet diese Methoden hingegen als ineffektiv. Man betrachte dazu einen beispielhaften Kommentar aus einer Online-Gruppendiskussion über Hypertextstandards<sup>3</sup>, in dem er sich für seinen harschen Tonfall in einer vorhergehenden Aussage rechtfertigt:

»I try to discourage what I see as bad practice, I'm stronger about it here because I think we're all technical people, I know as an engineer I much prefer direct talk to soft sell. And my experience is most engineers aren't overly sensitive - they can take heat as well as give it without making it a personal issue.«<sup>4</sup>  
(Kilker, 1999)

Aus dieser Perspektive betrachtet wirkt der sozialzentrierte Designprozess ineffizient. Soziale Interaktionen sollten direkt, effizient und vernünftig sein, um technische Probleme zu lösen. Ein guter Ingenieur muss dieser Ansicht nach Kritik einstecken können, um zur optimalen technischen Lösung zu gelangen.

Es wird klar, dass Mitglieder eines heterogenen Designteams sich nicht nur durch ihre Designidentitäten, sondern auch durch ihre Kollaborationstechniken unterscheiden. Konflikte sind daher vorprogrammiert und einige davon sind auch tatsächlich hilfreich. Sind die Ideale jedoch zu verschieden und lassen sich nicht vereinbaren, leidet der Designprozess und folglich die Qualität des Produkts (Kilker, 1999).

---

<sup>3</sup> Unter Hypertext wird in der Regel ein Text mit aktivierbaren Elementen verstanden, die im einfachsten Fall dazu dienen, direkt zu anderen Stellen im Text oder zu anderen Texten zu springen

<sup>4</sup> zu Deutsch: »Ich versuche immer von schlechten Methoden abzuraten. Ich tue das in dieser direkten Form, weil ich denke wir sind hier Techniker unter uns. Als Ingenieur bevorzuge ich, Dinge direkt und konkret anzusprechen und meine Erfahrung hat mir gezeigt, dass die meisten Ingenieure nicht sonderlich zimperlich oder sensibel sind. Sie können hitzige Diskussionen genauso gut verdauen, wie anzetteln. Keiner nimmt das persönlich.«

### 3.2 ANALYSE EINES KOLLABORATIVEN DESIGNMEETINGS

Nigel und Anita Cross haben Teamwork und implizite soziale Prozesse in einem Experiment untersucht (Cross and Clayburn Cross, 1995). Dazu haben sie ein Team aus drei Personen zusammengestellt und eine Videoobservation durchgeführt. Kerry, Ivan und John arbeiteten alle für die selbe Design-Consulting-Agentur und hatten sehr ähnliche Jobrollen innerhalb dieser Agentur. Sie waren hierarchisch gesehen also alle ungefähr gleichgestellt und brachten keine vordefinierten Rollen mit sich. Aufgabe des Workshops war es, einen Fahrradgepäckträger zu entwickeln. Bei ihrer Analyse konzentrierten sich Nigel und Anita Cross auf die folgenden Kernpunkte:

- Rollen und Beziehungen
- Planung und Tätigkeiten
- Sammeln und Teilen von Informationen
- Analyse und Verständnis des Problems
- Generierung & Adaptierung von Konzepten
- Vermeidung und Lösung von Konflikten

#### 3.2.1 Rollen und Beziehungen

Die Untersuchungen haben ergeben, dass im Team verschiedene Rollen eingenommen wurden. Teilweise wurden Rollen durch das Team geformt und zugewiesen, andere potenzielle Rollen, in die Individuen schlüpfen wollten, wurden vom Team ignoriert und konnten sich dadurch nicht vollständig entwickeln. Die nachfolgenden Auszüge aus dem Transkript zeigen die Entwicklung und Festigung von Rollen und Beziehungen zwischen den Teammitgliedern.

Nach dem Lesen der Anforderungen schlägt Kerry vor damit zu beginnen, das Design des bestehenden Prototypen zu analysieren.

Kerry what do we need? I guess we should look at their existing prototype, huh?

Auszug 3.1: Kerry schlägt vor, den bestehenden Prototypen zu reviewen.

John möchte zuerst lieber sicherstellen, dass alle die Problematik korrekt verstanden haben und vom selben reden.

John yeh, em, let me think; we could also just sort of like try to quantify the problem, because - what's your understanding of the problem first of all?

Auszug 3.2: John möchte sicherstellen, dass jeder die Problemstellung genau verstanden hat.

Diese Klärung der Problematik wird dann tatsächlich vom Team durchgeführt. Kerrys Vorschlag wurde somit ignoriert.

Während dieser Phase der Klärung schlägt Kerry vor, Informationen aus dem Report der Benutzerevaluierung zu nehmen.

Kerry they're not pleased with it so far, and the users' tests have some - in fact it would be nice if we could see those users' tests to em see what the shortcomings were.

Auszug 3.3: Kerry weist erneut auf die Evaluierung des bestehenden Prototypen hin.

Dieser Vorschlag wird von den anderen beiden Teammitgliedern ignoriert. Kurz nachher, während der Ablaufplanung, erwähnt Ivan den Begriff »Information« und Kerry weist erneut auf den Report der Benutzerevaluierung und seine Nützlichkeit hin. Wieder erreicht sie die anderen beiden damit nicht und ihre Anregung wird nicht umgesetzt.

Ivan information or  
 Kerry yeah we wanna look at the em customer feedback or the users' testing  
 John oh-yeah, so maybe, yeah, wherever that comes in in this list...

Auszug 3.4: Erneut wird Kerrys Anliegen von den anderen ignoriert.

Etwas später lenkt Kerry wieder ein und fragt erneut nach dem Report. Diesmal geht Ivan darauf ein.

Ivan I think I'd also like to get the information on em  
 Kerry the user testing  
 Ivan the user testing

Auszug 3.5: Diesmal hat Kerry Erfolg und Ivan geht auf ihren Vorschlag ein.

Nach der Klärungsphase empfiehlt Ivan, einen zeitlichen Ablaufplan festzulegen. John und Ivan beginnen sogleich mit dieser Planung. Ivan beginnt dann, die diversen Dokumente auf dem Tisch auszulegen und zu ordnen und John nützt die Gelegenheit, um Ivan zum Verantwortlichen für die Ablaufplanung zu ernennen.

Ivan	...let's get this stuff sorted out
John	OK you were talking about schedule stuff before, do you wanna
Ivan	yeah, I think we should uh just figure out
John	just set some time limits for ourselves

Auszug 3.6: John schlägt vor, Ivan sollte sich um den zeitlichen Ablauf kümmern.

Etwas später, als Ivan die Planung durchführt, wird diese Rolle nochmals von John bestätigt.

Ivan	five-thirty we'll move on to the final cost and presentation... let's leave ourselves a little bit of time
Kerry	mm mm
John	Ivan's gonna be Mister Schedule
Ivan	yeah [...] on time, under budget

Auszug 3.7: John bestätigt Ivan erneut in seiner Rolle.

Ivan hat somit die Rolle des Ablaufplaners übernommen und wird sie für die restliche Sitzung beibehalten.

Es wird schnell klar, dass Kerry Schwierigkeiten hat, ihre bevorzugte Vorgangsweise durchzusetzen. Ivan scheint mit seiner Rolle als Ablaufplaner recht glücklich zu sein und hat keine Einwände gegen diese Aufgabe. John scheint viel Einfluss darauf zu haben, wie sich die Dinge im Team entwickeln und was gemacht wird. Die Auszüge zeigen die Muster der Rollen und Beziehungen, die sich mehr oder minder durch die gesamte Sitzung ziehen. Trotzdem werden Rollen kurzzeitig getauscht und jeder der drei übernimmt irgendwann einmal die Führung der Gruppe. Jeder verkörpert diese Rolle jedoch auf seine eigene Art und Weise.

Die Autoren weisen darauf hin, dass es nötig wäre, die Analyse weiter fortzuführen und zusätzliche, wichtige Faktoren zu berücksichtigen. Zum einen wäre es interessant zu analysieren, was einzelne Teammitglieder machen, wenn sie sich nicht gerade im Mittelpunkt der Teambeschäftigung befinden. Sie könnten

entweder inaktiv sein, dem Geschehen aus einer Distanz beiwohnen, oder parallele Aktivitäten durchführen. Außerdem wäre es wichtig, Körperhaltung, Körpersprache, Gestik und Mimik genauer zu prüfen. Ebenso wichtig bezeichnen sie eine eingehende Analyse von Gelächter und Witzen, die zur Vermeidung von Konflikten eingesetzt werden.

### 3.2.2 Planung und Tätigkeiten

Das von Cross und Cross untersuchte Team legte großen Wert auf die Planung von Ablauf und Tätigkeiten. Dies scheint eine logische Vorgehensweise zu sein, aber oft gehen Teams anders an die Sache heran. Gerade in frühen, konzeptionellen Phasen von Design, ist die Arbeit ungeplant, intuitiv und spontan. Designer legen häufig ein sogenanntes opportunistisches Verhalten an den Tag. Das bedeutet, sie unterlassen geplante Aktivitäten und gehen spontanen Ideen nach, sobald sie ihnen einfallen. Im Team kann das aber zu Schwierigkeiten führen, denn die Aktivitäten der einzelnen Mitglieder müssen koordiniert und abgestimmt werden.

Die Planungsphase im Team von Kerry, Ivan und John beginnt als Ivan vorschlägt, einen Ablaufplan zu erstellen, anstatt aus dem Stegreif zu agieren.

Ivan	should we uh prepare a schedule and then just sorta stick to it, or should we uh just start working?
John	no, it's probably a good idea to try to quantify our amount of time; the kinda time we have left (laugh)

Auszug 3.8: Ivan möchte den Ablauf zuerst planen.

Ivan und John listen daraufhin die folgenden Aktivitäten auf einem Blatt Papier auf:

- Quantifizieren des Problems
- Konzepte generieren
- Konzepte verfeinern
- Das beste Konzept auswählen
- Design
- Präsentation
- Testen

Sie gehen hier nach einem konventionellen Modell vor, das beide offensichtlich gut finden. Kerry hingegen schreibt eine alternative Vorgehensweise auf:

- Verstehen
- Observieren
- Evaluieren

Kerry macht die beiden anderen jedoch nicht auf ihre Liste aufmerksam, es scheint eine persönliche Sache zu sein, genauso wie ein Diagramm, das sie zuvor für sich selbst aufgezeichnet hat. Dies deutet darauf hin, dass sie die Designaufgabe lieber anders angegangen wäre.

Die Rolle des zeitlichen Koordinators wurde bereits Ivan zugewiesen und falls notwendig, weist er auf Zeitknappheit hin und treibt die anderen beiden voran.

Ivan we have to start making decisions, we're already at five-fifteen.

Auszug 3.9: Ivan weist darauf hin, dass es bereits spät ist.

Ivan OK um keep moving along, we have er fifteen minutes to finish our design.

Auszug 3.10: Ivan versucht die anderen voranzutreiben.

Manchmal kommt es vor, dass einer der beiden anderen auf den Zeitplan hinweist. Während Kerry und John einige strukturelle Implikationen des Designs diskutieren, bemerkt John, dass das Team sich auf die Ideenfindungsphase zubewegt und möchte noch einmal auf die geplante Aktivität zurückkommen.

Kerry	it'd be nice if you could have the forks coming more like that, so that
Ivan	right
John	mm mm
Kerry	the bouncing
John	it sounds like
Kerry	these'd have to be self-attached
John	it sounds like in a way we're starting to move on to ideation already, but uh have we kinda fleshed these major things out?

Auszug 3.11: John bemerkt, dass das Team sich zu schnell nach vorne bewegt.

Nachdem die Gruppe die Designanforderungen aufgelistet hat, schlägt Ivan vor, die Ideenfindung zu beginnen. John stimmt dem zu, will aber vorher noch prüfen, ob wirklich alles klar ist.

Ivan	OK shall we move into uh idea-ideation?
John	yeah; I think we've... have we covered the uh all the stuff that (inaudible)? I'll just read some of this out loud 'cos I (reads from marketing report)

Auszug 3.12: John checkt nochmals alles ab, bevor die Ideenfindung beginnen kann.

Trotz der konkreten Ablaufplanung kommt es immer wieder zu unvorhergesehen Abweichungen. Dies passiert durch ein Teammitglied, das durch eine Aussage oder Aktion die beiden anderen in eine andere Aktivität hineinzieht.

Während John die Anforderungen auf dem Whiteboard auflistet, leitet Kerry die Aufmerksamkeit der beiden plötzlich auf das Design des bestehenden Prototypen.

Kerry	(referring to drawing) now this looks like a snap-in feature.
-------	---

Auszug 3.13: Kerry macht auf den Prototypen aufmerksam.

Ivan steigt darauf ein und die beiden diskutieren die strukturellen Aspekte des Designs. Sie analysieren, welchen Einfluss Bewegung und Erschütterung auf den Gepäckträger haben. Kerry macht einen Vorschlag zum Design der Halterung.

<p>Kerry it'd be nice if you could have the forks coming more like that.</p>
--

Auszug 3.14: Kerry teilt den anderen mit, wie sie sich die Halterung vorstellt.

Während der Diskussion über alternative Materialien bittet John um ein Messband und beginnt, den Gepäckträger abzumessen. Dadurch wird die zuvor begonnene Aktivität unterbrochen und diesem neuen Aspekt nachgegangen. Es sind nun alle Teammitglieder abgedriftet und der Fokus der Diskussion hat sich verschoben.

<p>John do you have a ruler or a measuring tape anywhere? Just wondering how big this rack is; ooh... this looks like</p>
<p>Ivan eighteen</p>
<p>John em eyeballing it looks like seventeen</p>
<p>Ivan OK</p>
<p>John seventeen there, and em shall we record some of this stuff?</p>

Auszug 3.15: John nimmt die Maße des Gepäckträgers.

Später bemerkt John, dass das Abmessen des Gepäckträgers dazu geführt hat, dass die ursprüngliche Aktivität nicht fortgeführt wurde und rechtfertigt sich dafür.

<p>John I kinda disrupted our materials discussion but I was</p>
<p>Ivan that's OK.</p>
<p>John when Kerry started talking about sorta the strength issue I was thinking well how big are we looking at?</p>

Auszug 3.16: John rechtfertigt die Unterbrechung.

Kerry führt die Diskussion dann sogleich auf den ursprünglichen Pfad zurück.

Etwas später, als die Diskussion etwas ins Stocken gerät, erwähnt Kerry einen Punkt, der sie beschäftigt: Wie könnte eine »gute Verbindung« zwischen dem Gepäckträger und dem Fahrradrahmen ausschauen? John erkundigt sich daraufhin beim Leiter des Experiments nach der Größe des Gepäcks, das auf dem Gepäckträger transportierbar sein soll. Der Leiter gibt John eine Zeichnung des Gepäcks mit Detailinformationen. Kerry und Ivan diskutieren das Design der Halterung des Gepäckträgers

und John sieht sich die Zeichnungen an. Als er bemerkt, dass der Text darauf niederländisch ist, unterbricht er die anderen.

John	OK who reads Flemish or Dutch or whatever this is?
Ivan	where is it?
John	there's not too much on there (laugh)

Auszug 3.17: John unterbricht die Diskussion der anderen beiden.

John nimmt dann wieder an der Diskussion teil und die Zeichnungen werden beiseite gelegt. Die Information über die Dimensionen des Gepäcks konnte nicht gewonnen werden.

Als Ivan vorschlägt, die Ideenfindung zu beginnen, willigt John ein, prüft aber zuvor noch, ob alles geklärt ist. Er liest dazu die Anforderungen laut vor. Währenddessen beschäftigen sich seine zwei Kollegen anderweitig. Ivan steht auf und geht hinüber zum Fahrrad, um dieses etwas genauer zu inspizieren. Kerry trinkt ihren Kaffee aus und wirft den Plastikbecher in den Mülleimer. Danach nimmt sie ein Gepäckstück und geht damit zum Fahrrad hin. Ivan nimmt das Fahrrad von seinem Stand herunter und Kerry positioniert das Gepäck hinter dem Sattel. All dies passiert ohne ein Wort. Scheinbar gibt es ein stillschweigendes Übereinkommen von Kerry und Ivan. Sie ignorieren John, der die Anforderungen fertig vorliest und sich dann zu den anderen gesellt. Er schlägt vor, das Gepäck mit etwas zu füllen.

John	So I think we've covered that... is there anything we could put like some weight into this? to kinda give us the right
------	--

Auszug 3.18: John gesellt sich zu den anderen.

Obwohl die Gruppe sich schon zuvor geeinigt hat, in die Ideenfindungsphase überzugehen, weiß keiner konkret, wie man dabei am besten vorgehen könnte.

John schlägt vor den Gepäckträger im mittleren Rahmenbereich, zwischen den Pedalen anzubringen. Kerry weist darauf hin, dass das eher unpraktisch wäre. Dann schlägt John vor, den Träger vorne an der Lenkstange anzubringen. Ivan schreibt die Vorschläge auf das Whiteboard. Das Team befindet sich nun in einer Phase, in der sie unterschiedliche Positionierungen für den Träger diskutieren, obwohl diese Aktivität nicht ausdrücklich geplant wurde.

Abweichungen vom Plan passieren nicht nur durch gewisse Impulse einzelner Teammitglieder hin. In manchen Fällen driftet

die Gruppe ab und rutscht so ab in andere Aktivitäten. Beispielsweise bittet Kerry während der Diskussion von Gewicht und Dimensionen des Gepäcks um Informationen zu bestehenden Gepäckträgern, damit sie Vergleiche anstellen kann. Wenn diese Information zur Verfügung steht, wird sie zu einer Quelle von Designideen.

Kerry	this looks a lot like the little backpack frame doesn't it?
Ivan	yeah... you see we've been, it seems like mentally trying to just, because of a similarity in size and shape between the two, thinking of ways to er use the same product for the same thing but I dunno that we necessarily - I mean we're on a target for \$55, I mean if they're able to make that for 42,95 ... and if we just add a plastic part

Auszug 3.19: Kerry und Ivan inspizieren einen Gepäckträger.

Die Diskussion entwickelt sich zu einem Gespräch, in dem Ivan seine persönlichen Erfahrungen mit Kindersitzen fürs Fahrrad mitteilt. Nach einer Weile holt John die Diskussion zurück zur eigentlichen Frage nach dem Gewicht des Gepäcks.

John	can I interject here?
Ivan	yeah
John	on our weight spec, if you just wanna sorta look at these they have weights; yeah they're between 430 grams to er 630 grams.

Auszug 3.20: John holt die Diskussion zurück.

Durch dieses Abdriften und ungeplante Aktivitäten, bzw. Diskussionen, ist es häufig schwer zu sagen, was gerade im Teamwork passiert. Diese Tatsache hat direkten Einfluss auf die Richtlinien für das Design von Systemen zur Unterstützung von kollaborativem Design, denn diese müssen eine solche Arbeitsweise ermöglichen und fördern.

### 3.2.3 *Sammeln und Teilen von Informationen*

Kerry ist die, die es für nützlich hält, verfügbare Informationen zu sammeln und zu inspizieren. Durch diesen Einwand meldet sie sich freiwillig, die Rolle des Informationssammlers zu übernehmen, ähnlich wie Ivan, der die Ablaufplanung übernommen

hat. Trotz anfänglicher Schwierigkeiten, ihren Wunsch durchzusetzen, erhält sie Informationen zum Zielpreis des Produkts, zum Benutzerreport und zum bestehenden Prototypen. Sie unterbricht das Auflisten der funktionalen Anforderungen und erfragt den Zielpreis.

John	... cost target, we don't really know what that is (laugh)
Kerry	low
John	low but
Kerry	maybe they have a - do we have that information, let's see do we ask for em - do we have any specification on what the uh reasonable price range is?

Auszug 3.21: Kerry erkundigt sich nach dem Zielpreis.

Nachdem einige Informationen gesammelt wurden, analysiert das Team die verschiedenen Unterlagen. Jeder liest ein bestimmtes Dokument und Sachen, die wichtig erscheinen, werden den anderen laut vorgelesen. Man erkennt hier keine konkrete Struktur in der Vorgangsweise, außer der Auflistung von Anforderungen am Whiteboard.

Während man bespricht, ob und wie der Gepäckträger abnehmbar sein soll, kommt das Thema Diebstahl auf. Kerry schaut in den Report der Marktstudie.

Ivan	was theft er an issue?
Kerry	er let's see - user marketing research?

Auszug 3.22: Kerry konsultiert den Report.

Während Ivan und John fortfahren, liest Kerry im Report weiter. »Diebstahlsicher« wird als Anforderung mit aufgenommen. Tatsächlich wird das Thema Diebstahl im Report gar nicht abgehandelt, aber Kerry teilt dies nicht explizit mit.

Die Spezifikation besagt, dass der Gepäckträger für einen »Hi-Star« Rucksack konzipiert werden muss. Trotzdem gibt es für Ivan und John Unklarheiten. Kerry löst die Missverständnisse auf.

John OK I missed that

Ivan which part did you miss?

John oh the fact that - I thought I picked up that they were going to, that they were conceiving of making an internal frame pack but em I guess that's not what they're saying; you're saying that they make external frame packs currently?

Kerry mm hmm they make external

John does it say that they want to stick with that?

John well it doesn't say anything about going uh external or internal so that I think that you raised a good point.

Kerry they just, yeah

Ivan yeah that we have that freedom right now.

John OK maybe we could get something that we're gonna propose to them that if it has any advantage in this application, right?

Ivan sure.

John OK.

Kerry but they wanna use it with this external frame backpack it looks like

Ivan right with this, well let's see.

Kerry because the HiStar, this this is a best-selling backpack, the mid-range HiStar.

Ivan right and they have their best-selling bike, right.

Kerry they've decided to develop an accessory for the HiStar.

Auszug 3.23: Kerry räumt Missverständnisse aus dem Weg.

Im späteren Verlauf der Sitzung, während Details eines Konzepts skizziert werden, fällt Kerry und Ivan auf, dass sie vergessen haben eine funktionale Anforderung zu berücksichtigen.

Kerry	what, the rack has to fold?
John	yes the rack has to fold.
Kerry	where does it say that?
John	it says that in our spec.
Ivan	where?
Kerry	our spec?
John	says right here
Kerry	(reads from brief) should fold down or - stacked away easily.

Auszug 3.24: Kerry und Ivan haben eine funktionale Anforderung vergessen.

Die Fehler und Missverständnisse charakterisieren die ineffiziente Strategie zur Sammlung von Informationen. Dies wurde jedoch wahrscheinlich durch die knappe Zeit des experimentellen Workshops und die Tatsache, dass die Teilnehmer sich bereits kannten, beeinflusst. Trotzdem ist das Vertrauen, das die Teilnehmer in eigene Erfahrungen und Wissen haben, eventuell eine Hürde für unterstützende Systeme. Mit Falschinterpretationen und Missverständnissen muss immer gerechnet werden. Das bedeutet, dass man nicht von einem allgemeinen Verständnis in kollaborativen Settings ausgehen darf.

#### 3.2.4 *Analyse und Verständnis des Problems*

Die erste gemeinsame Aktivität im Team ist der Versuch ein gemeinsames Verständnis des Problems zu schaffen. Erreicht werden soll dies durch Auflistung der Anforderungen und Spezifikationen in der Gruppe. Auf diese Art wird auch das Verständnis der einzelnen Teammitglieder offengelegt. Außerdem gibt es auch Versuche, die Problematik durch Ausformulierung zu verinnerlichen.

Während der anfänglichen Diskussion und Klärung des Problems listet John auf dem Whiteboard ein Set von Anforderungen auf. Dies ist die primäre Aktivität, um die Problematik zusammenzufassen und mit den anderen zu teilen. Kerry zeichnet außerdem noch ein Diagramm für sich auf, das das Problem grafisch darstellt. Dies ist ein Beispiel für das Ausformulieren, um das Problem selbst besser zu verstehen. Durch das Zeichnen wird das Konzept klarer und die Anforderungen können besser verinnerlicht werden.

Das Ausformulieren des Problems passiert auch in verbaler Form. Während der Auflistung der Anforderungen zum Beispiel, geraten Kerry und Ivan in eine Diskussion über die Erwartungen der Benutzer an das Endprodukt.

Ivan	do they talk about how the people wanna use it? The uh do these, do the vacations - they take long bicycle trips and then take short feet off uh short trips off by foot.
Kerry	mm mm
Ivan	em so they use the bike to get where they're going and then do a little hiking; sounds like the bike becomes the...
John	so you
Ivan	it sounds like they wanna really ride the bicycle and just temporarily go to work or something but you wanna be able to ride the bicycle.
Kerry	right mm mm
John	does it sound like...
Kerry	ride it through the country and then you get to the base of the hill and you wanna take your backpack and summit the mountain or something.
Ivan	mm hmm
John	so...
Ivan	sou you want like a real...
Kerry	and it's an off-road bike so you'd need a real rugged raugged attachment or a rigid attachment.
Ivan	mm mm

Auszug 3.25: Kerry und Ivan diskutieren die Erwartungen der Benutzer.

Es besteht ein hoher Kontrast zwischen Auflistung und Ausformulierung während der Analyse. Durch Auflisten entsteht eine externalisierte Spezifikation, dies führt aber nicht unbedingt zu einer Konzeptualisierung oder besserem Verständnis der Problematik, wie das hingegen bei der Ausformulierung passiert.

### 3.2.5 Generierung & Adaptierung von Konzepten

Die Designer müssen einige Konzepte generieren und daraus gemeinsam einen Designvorschlag formen. Dazu werden anfangs viele grobe Konzepte entworfen, von denen später einige wenige zu detaillierteren und robusteren Konzepten weiterentwickelt

werden. Die Teammitglieder machen jeweils zwei Arten von Vorschlägen: Generelle Prinzipien und Ideen, die sich im finalen Design widerspiegeln sollten und konkrete Konzepte, wie das Design ausschauen und funktionieren soll.

Ein Designvorschlag beginnt als grober Sketch und es muss noch viel Arbeit hineingesteckt werden, um ein konkretes Konzept daraus zu formen. Zu diesem Prozess tragen alle Teammitglieder bei und gemeinsam formulieren sie die Idee zu einem konkreten Entwurf aus.

Kerry	maybe if you could flip it out and it becomes a bike lock, 'cos you know - lock up your bike while you go on a hike, that'd be kind of a neat feature so you could justify some extra cost maybe.
Ivan	right, right.
John	kick-stand alternative (laugh)
Ivan	pull it around your tyre and now you can stand the bike up.

Auszug 3.26: Gemeinsam werden Konzepte formuliert.

Neben der Zusammenarbeit zur Konkretisierung von Konzepten versuchen einzelne Teammitglieder auch andere von Konzepten zu überzeugen, die sie persönlich für die besten halten (meistens ein Konzept, das ihnen selbst eingefallen ist). Es ist sehr üblich, dass Designer eine emotionale Bindung zu gewissen (eigenen) Entwürfen aufbauen. Im Experiment gab es zwei Beispiele, in denen Teammitglieder versuchten, andere von gewissen Konzepten zu überzeugen. John redete seinen Kollegen ein, sein Konzept sei das sinnvollste und Kerry und Ivan versuchten im Gegenzug John davon zu überzeugen, ihre Idee zu verfolgen.

John	so it's either a bag or maybe it's like a little vacuum-formed tray kinda for it to sit in.
------	---

Auszug 3.27: John hat eine Idee.

Das Konzept wird gleich aufgegriffen, diskutiert und um zusätzliche Ideen erweitert. John versucht direkt, sein Konzept auf die Liste der Lösungen zu bringen.

John	I think tray is sorta a new one on the list it's not a subset of bag.
------	---

Auszug 3.28: John will seine Idee durchsetzen.

Außerdem bestätigt er, dass er eine emotionale Bindung zu dieser Idee hat.

John yeah, I, I really like that tray idea (laugh)

Auszug 3.29: John ist bereits emotional gebunden.

Dieses emotionale Zugeständnis wird nochmals verstärkt durch Johns Aussage, die Auswahl des besten Konzepts sei ein Beliebtheitswettbewerb.

John I think all design eventually comes down to a popularity contest.  
Ivan (writes on board) tray  
Kerry I hate that idea.  
John (laugh)

Auszug 3.30: Kerry antwortet auf Johns Aussage mit einem Witz.

Sobald eines der Konzepte tatsächlich ausgewählt werden muss, nominiert John eilig sein »Tray« Konzept.

John well OK well we know we we like this tray idea, right?

Auszug 3.31: John will sein Konzept in die Endrunde bringen.

### 3.2.6 Vermeidung und Lösung von Konflikten

Es ist ganz normal, dass bei kollaborativem Design irgendwann Unstimmigkeiten und Konflikte aufkommen. Jeder hat eine eigene Meinung, ein eigenes Verständnis der Problematik, einen anderen Zugang und eigene Vorlieben. Am Beispiel von John, der sein eigenes Design durchsetzen will, sieht man wie ein solcher Konflikt aussehen kann. Dieser Wettbewerb der unter den Teammitgliedern entsteht, kann auch sehr viel härter umkämpft sein als im Experiment beobachtet. Nichtsdestotrotz müssen die Designer sich einigen und die Konflikte lösen, um die Aufgabe erfolgreich abzuschließen.

Während über die Verstellmöglichkeit des Gepäckträgers diskutiert wird, kommt es zu einer Unstimmigkeit zwischen den Teammitgliedern.

John	you know one of the things that seems problematic, and it would be great from a manufacturing standpoint if you could get around it, is this distance is going to vary with frame size all over the map.
Ivan	right.
Ivan	but so, y'know, we were talking about maybe those legs could extend before so that you could get some adjustability on your rack, maybe you need that anyway just so that you can adjust to different rack styles - like a telescoping tube here.
John	mm mm
Kerry	I don't think you really need it.
Ivan	(laughter) OK.
Kerry	because this is a 26 inch wheel or whatever, it's pretty standard and so if this distance - you're right it just does vary a lot, but what's gonna change is maybe your angle on your on your rack is gonna change er er what really is gonna happen is is this is gonna be a fixed distance because if we go onto the braze-on or something down here and and you want to make sure that there's clearance here and then as the bike grows it might pivot up a little bit more.

Auszug 3.32: Das Team ist sich nicht einig.

Später kommt John wieder auf die Verstellmöglichkeit zurück und bringt das Schlagwort »good human factors« ins Spiel. Kerry und Ivan lehnen Johns Argument nicht ab, lassen sich aber nur auf ein sehr unverbindliches Zugeständnis ein. Kerry zuckt mit den Schultern und Ivan sagt bloß OK. John bemerkt die Zweifel der anderen und beschwichtigt, dass das kein Fakt, sondern nur seine Meinung sei. Ivan schlägt daraufhin vor, dieses Thema bei der Finalisierung des Designs noch mal aufzugreifen und schiebt so die Diskussion darüber vorerst aus dem Weg.

John	I think good human factors says it should be adjustable so that people can find the position they like.
Kerry	(shrugs her shoulders)
Ivan	OK.
John	em that's my opinion.
Ivan	whatever idea we come up with I think we can...
John	opinion not fact (laugh)
Ivan	we can look at ways of making it adjustable.

Auszug 3.33: Kerry und Ivan finden Johns Einwand nicht so gut.

### 3.2.7 Schlussfolgerungen aus der Studie

Das Team hat es geschafft, die Aufgabe in der relativ kurzen Zeit des Experiments erfolgreich abzuschließen. In einem Gespräch nach dem Workshop bestätigten alle, dass die recht zufrieden mit dem Ergebnis der Sitzung waren und dass sie alle Spaß an der Zusammenarbeit hatten. Laut den Dreien waren die sozialen Prozesse, die während des Experiments passierten für keinen frustrierend. Ton und Umgang miteinander empfanden alle immer angemessen.

Es ist klar, dass Teamarbeit ein sozialer Prozess ist und soziale Interaktionen, Rollen und Beziehungen müssen in der Analyse von Designaktivitäten berücksichtigt werden. Diese Studie hat gezeigt, dass viele Aspekte der Teamarbeit von sozialen Faktoren beeinflusst wurden. Gleich zu Beginn beispielsweise, als Kerrys Versuche, eine alternative Vorgangsweise durchzusetzen, komplett ignoriert wurden. Der Wechsel zwischen geplanten und ungeplanten Aktivitäten zeigt ebenfalls diesen Einfluss sozialer Faktoren. Auch die persönliche Bindung zu bestimmten Konzepten und der Wettbewerb um Durchsetzung dieser Konzepte sind auch Indikatoren dafür. Die Lösung dieser aufgetretenen Konflikte passierte ebenfalls auf sozialer Ebene.

Diese Beobachtungen sind für die Analyse von Design in kollaborativem Umfeld sehr wichtig und sollten immer berücksichtigt werden. Designprozesse, speziell im Umfeld von Ingenieuren werden oft als rein technischer Natur angesehen: als eine Sequenz von Aktivitäten, die auf rationalen Vorgangsweisen zur Lösung rein technischer Problemstellungen dienen. Mittlerweile gibt es jedoch einige Studien, die das kollaborative Design als kognitiven Prozess wahrnehmen und den Faktor Mensch mit einbeziehen. (Cross and Clayburn Cross, 1995)

## ZUSAMMENFASSUNG

Problemstellungen und Aufgaben, denen sich Designer heutzutage stellen müssen, werden immer anspruchsvoller und hängen von vielen verschiedenen Faktoren ab. Daher ist es notwendig, diverse Experten mit spezifischem Wissen zu vereinen, um solche vielschichtigen Herausforderungen zu bewältigen. »Kollaboratives Design« ist eine weitaus komplexere Aktivität, als schlichtweg das Lösen einer bestimmten Arbeitsaufgabe innerhalb einer Gruppe. Soziale Prozesse in einem Team, bestehend aus unterschiedlichen Personen verschiedener Fachbereiche, sind höchst kompliziert. Gruppenmitglieder nehmenb verschiedene soziale Identitäten an, die gewissen Stereotypen entsprechen. Jeder Mensch braucht solche Identitäten, um sich und andere zu definieren. Sie ermöglichen Solidarität zwischen ähnlichen und Abgrenzung von unterschiedlichen Identitäten.



Menschen arbeiten mit Dokumenten in zwei verschiedenen Welten: der elektronischen Welt des Computers und der physikalischen Welt am Schreibtisch. Jede Welt hat ihre Vor- und Nachteile (siehe [Tabelle 4.1](#)), welche uns dazu bewegen die »richtige« für bestimmte Aufgaben zu wählen ([Wellner, 1993](#)).

In mancher Hinsicht scheint es aber, als wäre Papier bald obsolet. Manche prophezeien das papierlose Büro schon in wenigen Jahren. Das Eigenartige dabei ist nur, dass sich die Leute ungern von Papier trennen. Studien belegen, dass der Papierkonsum in Büros seit 1970 um das 6-fache anstieg und derzeit jährlich um 20% steigt ([Seybold, 1992](#)). Papier hat ebenso wie elektronische Dokumente Eigenschaften, die Menschen nicht aufgeben wollen. Das macht sie in Hinsicht auf computerbasierende Alternativen »unverwüstlich«. ([Luff et al., 1992](#))

Wie wichtig Papier (im Zusammenhang mit Skizzen) für Designer ist, wurde auch im Kapitel [DESIGNTHEORIE](#) beschrieben. Welchem Umstand dies das alte Medium zu verdanken hat, lässt sich laut Sellen und Harpers Arbeit *The Myth of the Paperless Office* ([Sellen and Harper, 2003](#)) nicht durch einfache Merkmalgegenüberstellungen herausfinden. Nur Studien und Observationen können das kognitive Verhalten bei der Benutzung von Papier und digitalen Artefakten beschreiben.

Das nun folgende Kapitel soll die Stärken und Schwächen traditioneller und digitaler Arbeitsweisen gegenüberstellen und mittels verschiedenster Studien eine Erklärung zum oben genannten Phänomen finden.

#### 4.1 SKIZZIEREN AUF PAPIER

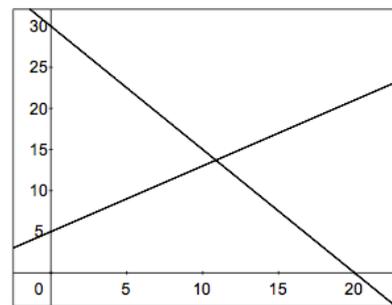
Skizzieren fördert eine schnelle und formlose Informationsverarbeitung. Eine Telefonnummer würde man beispielsweise eher auf ein Whiteboard in der Nähe des Schreibtischs schreiben, als sie in den Computer einzugeben, schnelle Berechnungen oder *To-Do-Lists* auf einem Blatt Papier, da sie leicht zu erstellen sind und das Ergebnis portabel ist. Larkin und Simon verglichen *schematische* Darstellungen mit *Darstellungen in der Wortebene* (geschrieben oder gesprochen) und zeigen damit dass räumliche Darstellungen oft mehr aussagen, als die gleiche Information in einem ausführlich geschriebenen Statement ([Larkin and Simon, 1987](#)). Wie effizient Information verarbeitet werden kann, hängt

ELEKTRONISCHE DOKUMENTE	PAPIERDOKUMENTE
<ul style="list-style-type: none"> <li>• schnelles Editieren</li> <li>• schnelles Kopieren</li> <li>• schnelles Senden</li> <li>• schnelles Freigeben</li> <li>• schnelles Ablegen</li> <li>• schnelles Abrufen</li> <li>• erlaubt Stichwortsuche</li> <li>• erlaubt Rechtschreibprüfung</li> <li>• erlaubt sofortige Berechnungen</li> </ul>	<ul style="list-style-type: none"> <li>• 3-dimensional</li> <li>• überall akzeptiert</li> <li>• billig</li> <li>• portabel</li> <li>• geläufig</li> <li>• hochauflösend</li> <li>• einfach zu lesen</li> <li>• fühlbar</li> <li>• man kann beide Hände &amp; Finger zum bearbeiten verwenden</li> <li>• man kann mit einem Stift darauf kritzeln</li> </ul>

Tabelle 4.1: Gegenüberstellung der Eigenschaften von elektronischen Dokumenten und Papierdokumenten.

$$P = 0.8 Q + 5$$

$$P = -1.5 Q + 30$$



(a) Geschriebene Darstellung (in der Wortebene).

(b) Graphische (schematische) Darstellung.

Abbildung 4.1: Die schematische Darstellung (a) und die Darstellung in der Wortebene (b) eines mathematischen Verhältnisses. Beide Darstellungsformen zeigen die gleiche Information.

technisch betrachtet davon ab, wieviel »Rechenaufwand« von Nöten ist, um Vorgaben so zu übersetzen, damit Verständnis entsteht. Um beispielsweise einen mathematischen Ausdruck eines Verhältnisses zu beschreiben, würde sich eine schematische Darstellung besser eignen. Der gleiche Ausdruck als Darstellung in der Wortebene würde eine längere Berechnungszeit benötigen (siehe [Abbildung 4.1](#)).

Wenn Personen skizzieren, fassen sie relevante Informationen zusammen und lassen die irrelevanten aus ([Tversky, 2002](#)). Zeichnen erlaubt Personen Papier als externen Speicher zu verwenden, um ihre kognitive Belastung zu verringern. Schematische Darstellungen stellen ein vereinfachtes Abbild von Informationsbereichen dar ([Tversky et al., 2000](#)). Per Hand gezeichnete Landkarten bzw. Wegbeschreibungen, werden z.B. mit simplen Formen wie Rechtecke oder Kreise für Gebäude, Linien oder Pfeile für Wege und sich überschneidende Linien für Kreuzungen gezeichnet ([Tversky and Lee, 1999](#)). Die Form und Krümmung von Gebäuden und Linien werden lediglich geschätzt, da Menschen bei genau gezeichneten Karten Schwierigkeiten haben, die kleinen Unterschiede zwischen der Karte und der echten Welt abzugleichen. Räumliche Darstellungen sind ein beeindruckendes Hilfsmittel beim Lernen. Skizzieren erlaubt Lernenden Konzepte, so wie sie verstanden wurden, aufzuzeichnen und diese von erfahrenen Personen auf Inkonsistenzen und Fehlern überprüfen zu lassen ([Forbus, 2008](#)).

Goel führte in ([Goel, 1995](#)) eine Reihe an Experimenten durch, um die kognitiven, handlungsauffordernden Merkmale von Skizzen zu erforschen. In seinen Studien verglich er traditionelle Papier-und-Stift Skizzen mit Skizzen, die mit Hilfe von computerbasierten Zeichenprogrammen erstellt wurden. Die teilnehmenden Designer bekamen Anweisungen, ein Artefakt auf beide Arten zu designen. Dabei kam ein modifiziertes Softwaretool zum Einsatz, das lediglich strukturierte Eingaben, wie gerade Linien, Rechtecke und Ellipsen unterstützte. Die Ergebnisse zeigten, dass das strukturierte Computerprogramm bezüglich der Ideenfindung nicht an die Schnelligkeit vom unstrukturierten Papier-und-Stift System herankam. Skizzieren unterstützt Designer bei der schnellen Entwicklung vieler unterschiedlicher Ideen. Diesen explorativen Charakter von Skizzen beschrieb Goel als *Quertransformationen*, die sich im Gegensatz zu den *vertikalen Transformationen*, welche sich durch Verfeinern von Ideen auszeichnen, deutlich abgrenzen. Bei den beiden Transformationstypen stützt sich Goel auf die von Newman und Landay in ([Newman and Landay, 2000](#)) beschriebenen Aktivitäten verschiedener Designstufen. Am Anfang des Designprozesses sind Designer mit der Ideengenerierung und -findung beschäftigt (Quertrans-

*Skizzieren mit einem strukturierten Softwaretool kommt nicht an die Schnelligkeit von unstrukturierten Papier-und-Stift Skizzen heran.*

formationen). Später konzentrieren sich Designer schrittweise Überarbeitungen und Änderungen (vertikale Transformationen).

#### 4.2 SKIZZIEREN AM COMPUTER

Heutige stiftbasierte Hardware orientiert sich vorwiegend am Schreiben und Lesen von Text. Wenn man erwägt, diese auch für Designaktivitäten einzusetzen, sollte man sie mit den Werkzeugen, die Designer in der Praxis benutzen, vergleichen.

Designer können schnell Zeichnungen auf Papier anfertigen und zwischen den einzelnen Seiten blättern. Sie können ebenfalls die Zeichnungen nebeneinander an die Wand heften, um sie zu vergleichen. Die Stiftspitze zeichnet dabei sofort bei Berührung auf dem Papier und es können zwei Hände benutzt werden um das Blatt Papier nach Belieben zu drehen bzw. um Hilfsmittel wie ein Lineal zu halten.

Am einfachsten wäre es, wenn computerunterstütztes Skizzieren genau die selben Eigenschaften aufweisen würde. Jedoch mangelt es der derzeitigen Hardware an Portabilität, schnellem Reaktionsvermögen und dem vertrauten Gefühl traditioneller Designpraktiken. Die kommerzielle Entwicklung und Forschung verbessert kontinuierlich Hardware auf diesem Sektor, so unterstützen viele Geräte wie z.B. PDAs, Tablet PCs oder interaktive Wandinstallationen bereits eine Art von Stift- oder Touchinput. Während stiftbasierte Systeme seit Jahrzehnten bei Forscher im Einsatz sind, verbreiten sie sich erst seit Mitte der 90er Jahre und werden stetig billiger.

»Skizzierhardware« kann in zwei Gruppen unterteilt werden:

- Hardware, die nur Eingaben unterstützt und
- Hardware, die Eingaben und Ausgaben unterstützt.

Tablets geben den Benutzern die Möglichkeit, mit Hilfe eines Eingabestifts zu schreiben bzw. zu zeichnen. Manche beschränken sich dabei aber lediglich auf die Eingabe und zeigen nicht gleichzeitig das Gezeichnete. Geräte, die Ein- und Ausgaben unterstützen, haben sich besonders bei den Tablet PCs durchgesetzt. Zusätzlich gibt es auch Systeme, die das gezeichnete indirekt »scannen«. Z.B. entwickelte PARC das ScanScribe System, welches Zeichnungen analysiert, die mit herkömmlichen Stift und Papier erzeugt wurden und erstellt daraus eine mit dem Computer aufgebesserte Version. (Johnson et al., 2009)

##### 4.2.1 *Electronic Ink*

Wenn man eine Zeichnung mit künstlerischem Ausdruck zu Papier bringt, gibt es viele Faktoren die das Endergebnis beein-

flussen. So z.B. die Stärke des Schreib- bzw. Zeichengeräts, sowie die Materialeigenschaften des Papiers. In vielen Branchen, wie beispielsweise in der Computeranimation, ist es noch immer üblich, Illustrationen zuerst auf Papier zu zeichnen und danach schrittweise zu digitalisieren. Es wurden zwar einige Versuche gestartet, direkt auf digitaler Ebene zu zeichnen, jedoch wird dies nach wie vor hauptsächlich nur dort verwendet, wo der Computer nicht nur optional benutzt wird. (Henzen et al., 2005)

Viele Tablets sind druckempfindlich. Designer benutzen oft stärkere Linien, um Objektgrenzen hervorzuheben und dünne Linien um dezente Texturen, Schatten oder Rundungen anzudeuten. Geräte, die Druck messen können, erlauben Designer dickere oder dunklere Striche zu zeichnen, ohne vorher in einen anderen Zeichenmodus wechseln zu müssen.

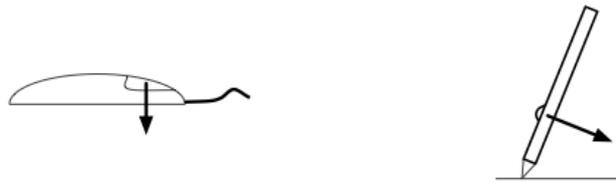
Die Druckmessung wurde auf verschiedene Weisen umgesetzt. Z.B. durch zwei übereinander liegenden leitfähigen Schichten mit entgegengesetzten Stromrichtungen. Die Schichten berühren sich nicht und sind oft durch eine nicht leitende Flüssigkeit abgeschirmt. Wenn etwas (ein Stift oder Finger) die obere Schicht berührt, verändert sich die Spannung und die Position kann mittels Interpolation an den Kanten errechnet werden. Andere berührungsempfindliche Oberflächen messen die elektrischen Eigenschaften der Dinge, die sie berühren. Darum funktionieren auch Handschuhe auf manchen Laptop Trackpads nicht. Wacom™ Tablets und gleichartige induktive Geräte, benötigen spezielle Stifte, die in einem vom Tablet generierten elektromagnetischen Feld mitschwingen. Wieder andere Tablets erkennen akkustische oder optische Störungen zur Positionsberechnung.

Es gibt Oberflächen, die lediglich die Koordinaten eines einzigen Berührungsortes liefern können und Oberflächen, die mehrere erkennen. Diese werden auch Multitouchoberflächen genannt. Multitouchsysteme, wie z.B. *Hans's Frustrated Total Internal Reflection technique* (Han, 2005) und *Microsoft's Surface System* (Microsoft, 2010) befinden sich derzeit im Aufschwung, werden aber mit den Fingern oder Händen benutzt und bieten deswegen völlig andere Anwendungserlebnisse als Stiftbasierte Oberflächen.

#### 4.2.2 *Der Unterschied zwischen Eingabestift und Maus*

Egal welche Abtasttechnologie man wählt, alle oben genannten Geräte erlauben Benutzern eine Eingabevariante, die näher an das traditionelle Schreiben herankommt, als die Maus jemals könnte. Obwohl Stift- und Mauseingaben viele Eigenschaften teilen (beide erlauben Benutzer im 2D Raum zu interagieren), haben sie einige grundlegende Unterschiede.

Mauseingaben liefern Daten über die *Bewegung*, Stifteingaben liefern Daten über die *Position* (Hinckley, 2003). Mit anderen Wor-



- (a) Drücken einer Maustaste erzeugt eine Kraft, die orthogonal auf die darunterliegende Oberfläche wirkt.
- (b) Drücken einer Stifttaste erzeugt eine ungewollte Stiftspitzenbewegung.

Abbildung 4.2: Entstehende Kräfte beim Drücken einer Maustaste (a) und einer Stifttaste (b) eines Tablets.

ten produzieren Mäuse relative *Änderungen* in der  $(x,y)$  Position und Stifte direkte, absolute  $(x,y)$  Positionen. Benutzer können Tablets aber auch so konfigurieren, damit sie sich wie Mäuse verhalten und relative Positionen erzeugen.

Die Form der Geräte ist ebenfalls extrem wichtig. Ein Stift zwingt die Benutzer die Feinmotorik ihrer Finger einzusetzen um die Stiftspitze zu steuern, wohingegen die Hand- und Unterarmmuskeln die Maus steuern. Finger können zwar auch zur Maussteuerung benutzt werden, aber nicht mit der gleichen Fertigkeit. Je nach Art der Arbeit ist ein Stift ergonomisch besser als eine Maus oder umgekehrt.

Tablets können aber mehr als nur die Stiftposition erkennen. Manche Geräte können auch den Aufpressdruck, den Winkel oder die Rotation des Stifts messen. Das andere Ende des Stifts wird auch oft als alternativer Modus (z.B. als ein Radierer) benutzt.

Einige Stifte haben noch zusätzliche Buttons. Während Buttons unentbehrliche Bestandteile von Mäusen sind, können sie entlang des Stiftgehäuses schwer benutzbar sein (Plimmer, 2008). Die Kraft, die bei einem Buttonklick an der *Maus* entsteht, ist orthogonal zu der Oberfläche, auf der die Maus benutzt wird und beeinflusst die Zielgenauigkeit nur unwesentlich. Ein Knopfdruck am *Stift* hingegen kann die Stiftspitze ungewollt bewegen, was das Zeigen auf ein bestimmtes Objekt und das gleichzeitige Drücken einer Taste erschwert (siehe [Abbildung 4.2](#)). Zudem muss der Benutzer beim Drücken einer Taste üblicherweise den Stift erst so drehen, damit sich sein Finger wieder über der Taste befindet (was durchaus öfter vorkommt). Diese Aktion lenkt ab und ist auf längere Zeit gesehen unbequem. (Johnson et al., 2009)

*Buttons an  
Tabletstiften können  
schwer benutzbar  
sein und sich auf  
längere Zeit  
unangenehm  
auswirken.*

### 4.2.3 Interaktionstechniken für stiftbasierte Skizziersysteme

Bei der Entwicklung neuer Technologien ist es wichtig das Ausmaß der kognitiven Belastung des Tools zu berücksichtigen. Aus Oviatt's Studie (Oviatt et al., 2006) über Mathematikstudenten geht hervor, dass das Arbeiten mit stiftbasierte Applikationen auf Tablet PCs erheblich schlechter funktioniert als mit herkömmlichen Stift und Papier. Durch die Benutzung der Tablets benötigten die Studenten mehr Zeit bei der Lösung von Mathematikaufgaben, was dazu führte dass sie sich mit der Technologie nicht anfreunden konnten. Anthony et al. führten eine ähnliche Studie durch, in der sie Eingaben von Studenten verglichen, die Gleichungen beinhalteten. Zur Eingabe dienten ein *Standard WIMP<sup>1</sup> Interface* und ein Eingabetablet (Anthony et al., 2005).

Sie kamen zur Erkenntnis, dass die Studenten Stifteingaben bevorzugten und handgeschriebene Gleichungen schneller und genauer niederschreiben konnten, als mit Keyboard und Maus. Papier und Stift wurden dem elektronischen Schreiben zwar vorgezogen, jedoch waren ihnen Stifte im Allgemeinen lieber als Tastatur- und Mauseingaben. Beide Studien kamen zum selben Schluss: Je mehr Aufmerksamkeit Benutzer aufbringen müssen um das Tool zu verwenden, desto weniger Aufmerksamkeit schenken sie dem eigentlichen Problem.

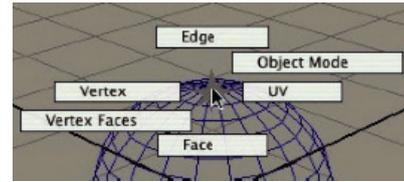
*Umso mehr Aufmerksamkeit man (Software-)Tools schenken muss, desto weniger Aufmerksamkeit bekommt das eigentliche Problem.*

Zur Reduzierung der kognitiven Belastung müssen demnach natürlichere Interaktionstechniken entwickelt werden. Verschiedenste Methoden erwiesen sich bis dato durchaus brauchbar zum effizienten Arbeiten mit Stiftapplikationen. Kontextmenüs sind ein gutes Beispiel dafür (Kurtenbach and Buxton, 1991). Traditionelle Menüs zeigen eine Liste an Optionen, welche das Menü nach unten und rechts wachsen lassen. Bei Tablets, die Eingaben und Ausgaben unterstützen (z.B. Tablet PCs) kann dies dazu führen, dass Benutzer das Menü mit der Hand verdecken. »Torten«-Menüs (eine Art Kontextmenü) erscheinen zentriert um die Stiftposition und zeigen die Optionen kreisförmig (siehe [Abbildung 4.3](#)). Die Hand ist zwar bei dieser Art von Menü noch immer teilweise im Weg, aber zumindest ist ein Teil des Menüs dadurch sichtbar. Der Vorteil dieser Methode ist, dass sich Personen Gesten anlernen können, ohne die Menüeinträge jedesmal erneut lesen zu müssen. Voraussetzung dafür ist jedoch, dass sich die Menüeinträge niemals ändern. Eventuell schwindet auch der Bedarf eines visuellen Menüs durch die Gesten. Tortenmenüs werden in Applikationen wie Maya (Autodesk, 2010), Spielen wie The Sims (Electronic Arts, 2010) oder als Erweiterungen in Webbrowser verwendet. Der Ansatz funktioniert zwar gut, bringt

<sup>1</sup> Die Abkürzung WIMP wurde 1980 von Merzouga Wilberts ins Leben gerufen und steht für Window, Icon, Menu, Pointing Device. Es bezeichnet das Grundkonzept moderner grafischen Benutzerschnittstellen (GUIs).



(a) Tortenmenüerweiterung für den Firefox Webbrowser.



(b) Autodesk Maya Tortenmenü.

Abbildung 4.3: Tortenmenüs in Firefox (a) und Maya (b).

aber auch zwei Nachteile mit sich. Zum einen ist es nicht klar wie man das Menü aufrufen kann und zum anderen müssen Gesten erst entdeckt, erlernt und ins Gedächtnis gerufen werden.

Ramos et al. (Ramos et al., 2004) erforschte weiters die Benutzung von Aufpressdruckdaten bei stiftbasierenden Interaktionen. Zu der zweidimensionalen  $(x,y)$  Stiftposition kommt die Druckstärke des Stiftes als dritte Dimension hinzu, die der Benutzer frei verändern kann. Zum Beispiel kann ein leichter Druck ein Menü zeigen und ein harter Druck ein anderes Menü. Druck kann eine effektive Eingabeart sein, wenn sie richtig (mit haptischen und visuellen Feedback) eingesetzt wird.

Hinckley et al. (Hinckley et al., 2005) empfiehlt die Benutzung von *Begrenzer* um Kontextmenüs aufzurufen. *Begrenzer* sind Eingaben, die Benutzer normalerweise nicht zeichnen würden aber dennoch leicht zu merken sind. Ein Beispiel zeigt ihr System Scriboli, in dem ein »Rattenschwanz« (eine Schleife am Ende einer Selektionsgeste) als *Begrenzer* benutzt wird. Diese flüssige Bewegung lässt den Benutzer Zielobjekte kennzeichnen, um auf sie anschließend ein Kommando anwenden zu können.

Wieder andere haben Interface Idiome<sup>2</sup> zur Unterstützung von Stift- & Skizzeneingaben entwickelt. *Gedrics* sind auf Gesten basierende Icons für stiftbasierte Applikationen (Geißler, 1995). Jedes *Gedric* ist mit einer Klasse an Aufgaben, wie »ändere die Schriftarteigenschaften« in einem Texteditor, verknüpft. Um ein Kommando durchzuführen, muss der Benutzer eine Geste auf ein *Gedric* zeichnen. Das System erkennt die Geste und gibt dem Icon anschließend die auf die Geste verknüpfte Bedeutung, welche durch Darauftippen aktiviert wird. Zeichnet man z.B. eine schräge Linie auf einem »Schriftart«-*Gedric* stellt das System einen markierten Text kursiv. Zeichnet man eine vertikale Linie von unten nach oben, wird die Schriftgröße erhöht.

<sup>2</sup> Ein Idiom im Sinne der Softwaretechnik ist ein programmiersprachenspezifisches Muster. Ein Idiom beschreibt, wie man bestimmte Aspekte von Komponenten oder Beziehungen zwischen ihnen mit den Mitteln einer bestimmten Programmiersprache implementiert (Buschmann, 1998).

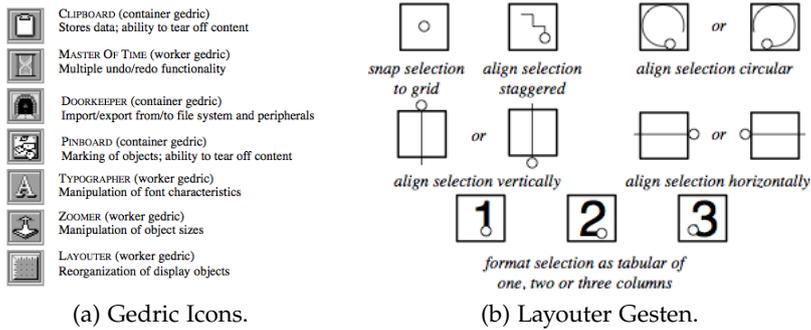


Abbildung 4.4: Beispiele für Gedric Icons (a) und Gesten einer Layoutfunktion (b).

So komfortabel Operationen auf Gedrics auch wirken, erfordern sie trotzdem eine Eingewöhnungsphase, in der Benutzer lernen müssen ihre Absichten in Gesten umzuwandeln. (vgl. [Abbildung 4.4](#))

Während Point & Click Aktionen für Aufgaben zu Anwendungen, die üblicherweise mit der Maus ausgeführt werden, ausgelegt sind, raten Accot und Zhai zur *Kreuzen-Technik* zur Stiftinteraktion ([Accot and Zhai, 2002](#)). Das experimentelle Zeichenprogramm *CrossY* ([Apitz and Guimbretiere, 2004](#)) zeigt Interface Widgets<sup>3</sup>, die Kreuzen erlauben. Um ein CrossY Button zu aktivieren, muss der Benutzer eine Linie von einer Seite des Buttons zur anderen zeichnen und somit den Button überqueren. Kreuzen macht es möglich, mehrere Aktionen in einer flüssigen Bewegung abzuhandeln. Zum Beispiel kann man die Farbe und Stärke der Electronic Ink des Stiftes gleichzeitig ändern, in dem man den Stift über benachbarte CrossY Widgets bewegt (siehe [Abbildung 4.5](#)).

So wie Skizzen mehrdeutig sein können, können auch stiftbasierte Interaktionen, wie das Drücken eines Buttons oder die Auswahl eines Menüpunktes, mehrdeutig sein. Wenn der Benutzer mit Hilfe des Stiftes, mit einem Objekt interagieren will, kann es sein, dass das System sein Ziel erst eindeutig machen muss. Zum Beispiel kann beim Drücken eines Buttons der Stift unabsichtlich zu einem anderen Button rutschen. Dieser Umstand wird auch *Target Ambiguity*, oder laienhaft übersetzt *Ziel Mehrdeutigkeit* genannt ([Mankoff, 2000](#); [Mankoff et al., 2000](#)).

Eine Alternative, um gegen die Mehrdeutigkeit vorzugehen, ist sie im Voraus zu bekämpfen. Pegasus und Chateau ([Igarashi and Hughes, 2001, 2003](#)) demonstrieren ein *Empfehlungssystem*, das vorhersagt, was der Benutzer zeichnen wird und alle möglichen Aktionen aufzeigt. Diese Technik eignet sich besonders, wenn die

<sup>3</sup> Widgets (dt. Steuerelemente) sind Interaktionselemente in einer grafischen Benutzeroberfläche (GUI), beispielsweise eine Schaltfläche oder eine Bildlaufleiste.

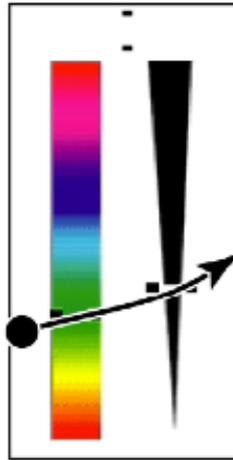


Abbildung 4.5: CrossY Interface Widgets erlauben Benutzer mehrere Parameter mit einer einzigen Bewegung zu ändern – in diesem Beispiel die Stiftfarbe und -stärke.

Umgebung eine regelmäßige Grammatik aufweist oder strukturierte Eigenschaften wie Symmetrie genutzt werden soll. Tsang et al. benutzten ein Empfehlungssystem in ihrem Programm, das Formen wie Flugzeugaußenhüllen modellieren kann (Tsang et al., 2004). Das System bietet ein Overlay, das den Benutzer durch zusätzlichen Skizzierinput leitet. Benutzer zufolge führt dies zu präziseren Input. Das System benutzt die angefangenen Skizzen als Vorlagen, vergleicht diese mit Datenbankeinträgen um ähnliche Zeichnungen zu finden und empfiehlt zusätzliche Geometrie, die vielleicht dazu passen könnte.

Bae's 3D Kurvenmodellierungssystem zeigt wie verschiedene kaligraphische Interaktionstechniken zusammen benutzt werden können um eine flüssige Skizzierumgebung zu schaffen (Bae et al., 2008). Das sog. *ILoveSketch* System imitiert ein physikalisches Skizzierbuch. Die Eingaben erfolgen per Stift und können optional durch Drücken von physikalischen Buttons durch die andere Hand verändert werden (siehe [Abbildung 4.6](#)). Das Interface verzichtet auf herkömmliche on-screen Buttons, Scrollbars und Menüs. Stattdessen gibt der Benutzer Kommandos durch Gesten, die oft vom Kontext abhängen. Zum Beispiel kann der Benutzer zu älteren Zeichnungen zurück »blättern«, in dem er an einer Ecke zieht. Um ein Objekt zu löschen, muss der Benutzer eine »Durchstreichen«-Geste anwenden. Viele Techniken in *ILoveSketch* betreffen Herausforderungen beim Skizzieren von 3D Objekten. Zum Beispiel errechnet das System automatisch aus der Arbeit des Benutzers die richtige Blickrichtung durch Drehungen und Verschiebungen.



Abbildung 4.6: ILoveSketch zeigt ein »natürliches« Skizziersystem, welches mehrere kaligraphische Interaktionstechniken miteinander vereint (Bae et al., 2008).

Alvarado fasste in (Alvarado, 2004) eine Liste aus sieben Designguidelines zur Entwicklung von *Sketch Recognition User Interfaces* (Benutzeroberflächen zur Skizzenerkennung), kurz SkRUIs zusammen:

1. Zeige Skizzenerkennungsergebnisse erst, wenn der Benutzer fertig ist mit dem Zeichnen
2. Sorge für offensichtliche Hinweise, um das Skizzieren von der Erkennung zu unterscheiden
3. Beschränke die Erkennung auf einen einzelnen Bereich, solange keine automatische Bereichserkennung machbar ist
4. Integriere stiftbasiertes Editieren
5. Skizzieren und Editieren sollten bestimmten Stiftbewegungen zugrunde liegen
6. SkRUIs benötigen große Buttons
7. Der Stift muss stets in real-time reagieren.

Während WIMP Interfaces seit Mitte der 1980er Jahre weit verbreitet benutzt werden, müssen stiftbasierte Interfaces erst Fuß fassen. Um bessere Interaktionsrichtlinien, Interface Design Patterns und Toolkits zu entwickeln, muss die Forschung weiterhin Skizzierapplikationen entwickeln und evaluieren.

#### 4.2.4 Das Modusproblem

Oft interpretiert das User Interface Eingaben unterschiedlich, je nach dem in welchen Modus sich das Programm gerade befindet. Zum Beispiel hat ein Zeichenprogramm Eingabemodi wie *Auswahl*, *Linie zeichnen* oder *Füllwerkzeug*. So ein Programm erlaubt dem Benutzer beispielsweise rechteckige Bereiche zu selektieren oder, wenn das Stifttool ausgewählt ist, zu zeichnen - beides indem der Benutzer eine Maustaste drückt und die Maus bewegt. Das Programm interpretiert die Eingaben hinsichtlich dem ausgewählten Werkzeug.

Manchmal ist sich der Benutzer jedoch nicht im Klaren, in welchen Modus er sich befindet oder weiß nicht, wie er in einen anderen Modus umschalten kann. Die Handhabung der Modi führt oft zu kognitiver Überlastung der Benutzer, da sich diese auf das ausgewählte Tool konzentrieren, anstatt auf ihre Arbeit. Dabei spricht man auch vom sog. *Modusproblem* (Tesler, 1981), welches seit den Anfängen interaktiver Systemen besteht und nicht nur auf Skizziersoftware beschränkt ist.

Mehrere  
Programmmodi  
führen oft zu  
kognitiver  
Überlastung.

*Sketchpad*<sup>4</sup>, das wohl erste nennenswerte elektronische Skizziersystem, behalf sich mit physikalischen Steuerelementen (Knöpfe, Schalter, Regler), durch die der Benutzer mit seiner linken Hand zu den verschiedenen Modi schalten konnte (siehe [Abbildung 4.7](#)) (Sutherland, 1964).

Im *GRAIL*<sup>5</sup> System können Benutzer nicht explizit in einen anderen Modus wechseln, um Text und Grafiken zu bearbeiten. Stattdessen versucht das System die Absichten des Benutzers aus einer Analyse des Gezeichneten bzw. dessen Kontext zu erschließen (siehe [Abbildung 4.8](#)) (Ellis et al., 1969).

Diese zwei früh entwickelten Systeme stellen zwei Extreme gegenüber, die zeigen, wie man mit dem Modusproblem umgehen kann. *Sketchpad*'s Ansatz baut auf explizite Modiwechsel abseits des Stiftes auf, *GRAIL*'s Ansatz auf implizite Modiwechsel, errechnet durch Stifteingaben.

4 *Sketchpad* war ein interaktives Designsystem aus dem Jahr 1963, das Technikern erlaubte Modelle mit Hilfe eines Lichtstiftes auf einem Display zu erstellen. Der Benutzer konnte dabei verschiedene Bedingungen (wie z.B. »erstelle eine Linie parallel zu dieser Linie und erhalte das Verhältnis) auf das Gezeichnete anwenden. (Sutherland, 1964)

5 *RAND*'s *GRAIL* (GRAphical Input Language) System aus dem Jahr 1968, interpretierte Stifteingaben in einer besonderen visuellen Programmiersprache, um Flussdiagramme zu erstellen (Ellis et al., 1969). Benutzer konnten via Eingabetablet semantisch wertvolle Modelldaten (Rechtecke, Pfeile, Schrift) und Kommandos (Lösche eine Linie, Bewege ein Rechteck, etc.) zeichnen und *GRAIL* errechnete anschließend automatisch die bestimmungsgemäße Information.

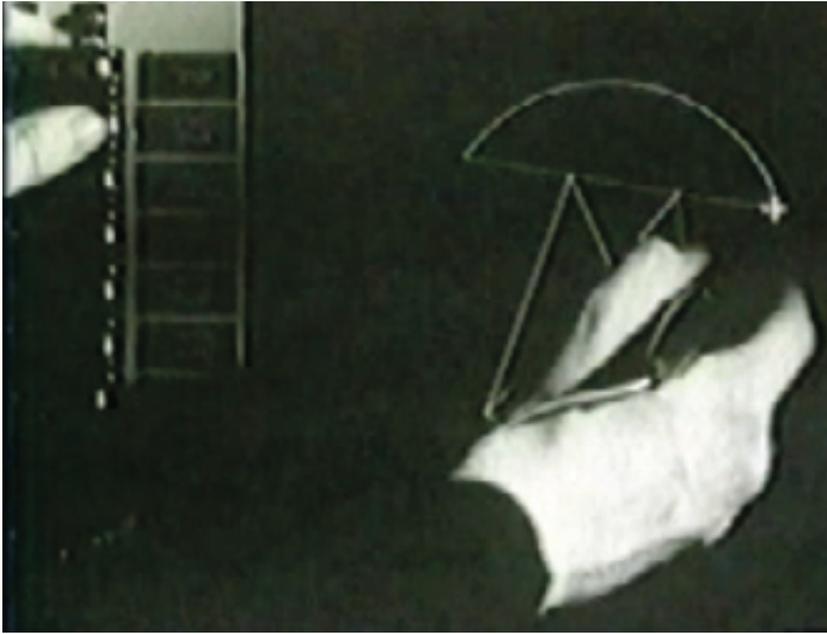
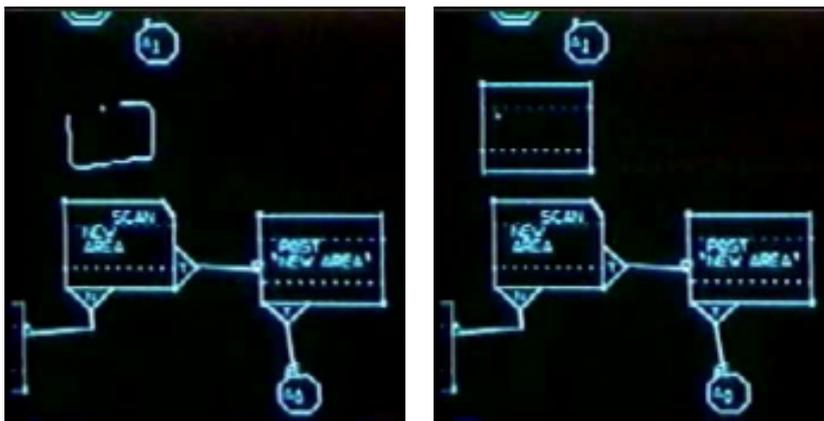


Abbildung 4.7: Sketchpad unterstützt Benutzer beim Erstellen von Designzeichnungen mittels Stift (rechte Hand) und verschiedener Modi/Bedingungen (erreichbar über die Knöpfe auf der linken Seite).



(a) Input

(b) Output

Abbildung 4.8: GRAIL analysiert die Benutzereingaben (a) und errechnet automatisch die bestimmungsgemäße Information (b) - hier z.B. ein Rechteck.

Es scheint keinen »richtigen« Weg geben, um dem Modusproblem entgegenzuwirken. Implizite Modiwechsel scheinen natürlicher, aber nur wenn das System die Benutzereingaben richtig interpretiert. Interpretationstechniken sind fehleranfällig. Viele Systeme ermöglichen deswegen eine Kombination dieser zwei Arten oder verhängen Zeichenkonventionen.

Saund and Lank erforschten automatische Interpretationen und Modiwechsel, basierend auf Eingaben von Benutzern und bereits Gezeichneten (Saund and Lank, 2003). Ihr *Inferred-Mode Protocol* beschreibt einen Ansatz zur Analyse des Gezeichneten, um herauszufinden, ob Aktionen mit Absicht durchgeführt wurden oder nicht. War eine Aktion unklar, wird ein Mediator eingesetzt, der Methoden vorschlägt, um die Unklarheit zu beseitigen.

Li et al. verglichen in (Li et al., 2005) verschiedene Moduswechseltechniken für stiftbasierende User Interfaces. Dabei wurden folgende Techniken miteinbezogen:

- Tasten am Stift,
- Drücken und Halten,
- Benutzung der schwachen Hand, um einen physikalischen Button zu drücken,
- ein neue durckbasierte Methode und
- Benutzung des »Radierers« des Stiftes

*Explizite Modiwechsel durch physikalische Buttons sind die schnellsten, am Fehler unanfälligsten und beliebtesten.*

Interessanterweise war die Methode, in der die schwache Hand verwendet wurde, die schnellste, am Fehler unanfälligsten und die beliebteste. Die »Drücken und Halten« Methode, wurde auch von Schilit et al. (Schilit et al., 1998) verwendet, die dies auch »Rast«-Geste bezeichneten. Microsoft Windows for Tablet PCs benutzt diese Geste beispielsweise, um die Rechtsklick-Kontextmenüs aufzurufen.

Bei Skizziersystemen tritt das Modusproblem vorwiegend auf, da es mehrere Typen an Stifteingaben gibt. Manche Eingaben sollen auf einer Seite angezeigt werden, da sie Wörter, Zeichnungen oder andere Modellelemente (*Model Operations*) darstellen. Andere sollen im Hintergrund weiterverarbeitet werden, da sie Selektionen oder Kommandos (*Environment Operations*) angeben.

*Flow Selection* (Johnson et al., 2006) erlaubt Benutzern mit Hilfe der Rast-Geste nahtlos vom Zeichen- zum Selektionsmodus zu wechseln. Anschließende Operationen, wie Verschieben eines Teilabschnitts einer Linie, werden durch Bewegung des Stiftes ausgeführt, ohne den Stift vorher abzusetzen. Wieviel Einzelpunkte vom darunter liegenden Objekt selektiert werden - man spricht dabei auch von der Selektionstärke - hängt vom Abstand zur

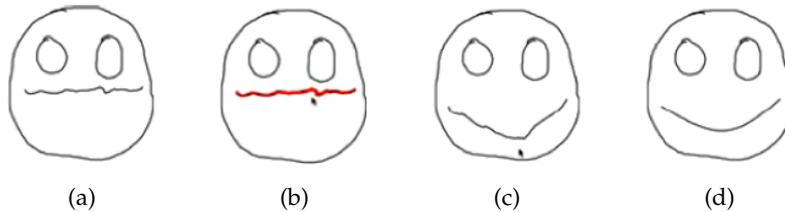


Abbildung 4.9: Modiwechsel der Flow Selection geschehen durch Halten und Bewegen des Stiftes (Johnson et al., 2006). Der Benutzer positioniert dazu den Stift am gewünschten Objekt (a), bewegt es (b) ohne dabei den Stift zu heben (c). Der Benutzer hält den Stift anschließend an der Position bis die Kurve geglättet wurde (d) bevor er den Prozess durch heben des Stiftes abschließt.

Stiftposition ab und wie lange der Benutzer den Stift an der selben Stelle ruhen lässt. Die Selektionstärke wird bei Operationen wie Verschieben oder Glättung verwendet (vgl. [Abbildung 4.9](#)).

#### 4.3 DIE BEDEUTUNG VON GESTIK

*Design is an "activity of the mind . . . grounded in mechanisms that evolved for interaction with the environment".*

— M. Wilson (Wilson, 2002)

Die Kombination von Gesten und Sprache bilden die Grundlage zur menschlichen dialogorientierten Interaktion. Sie sind somit auch ein wichtiger Bestandteil von Design (vgl. obiges Zitat von Wilson). Um diese Modalitäten auch richtig in HCI anwenden zu können, man ihre Wechselwirkung zur Kommunikation verstehen.

Was sind eigentlich Gesten? Wenn ein Student mit einer Kravatte in die Klasse kommt und der Professor keine trägt, geben beide eine Aussage über ihre Einstellung zu der Klasse ab.

Solche Handlungen nennt man auch »Nonverbale Kommunikation«. Ein breites Spektrum an Verhaltensweisen zählt dazu: die Wohn- und Arbeitsumgebungen, die wir schaffen; den Abstand, den wir zwischen uns und unseren Gegenüber einnehmen; ob wir unseren Körper bewegen, Augenkontakt herstellen, oder unsere Stimme erheben; all dies spielt zusammen und sendet Signale aus.

Die traditionelle Sicht auf Kommunikation teilt sich in die verbale und nonverbale Komponente bzw. ihr Zusammenspiel. Adam Kendon (1980) war einer der ersten, der sich mit dieser Ansicht beschäftigte. Er behauptet, dass sich mindestens eine

Form des nonverbalen Verhaltens - Gestik - von einer Konversation nicht trennen lässt. David McNeill zeigte 1992 in seinen bahnbrechenden Studien über Gestik und Sprache, dass Handbewegungen, die wir beim Sprechen produzieren, fest mit dem Gesprochenen, bezüglich Timing, Bedeutung und Funktion verstrickt sind. Wenn man Gestik ignoriert, ignoriert man einen Teil der Konversation.

Gestik ist ein Begriff, der für einen großen Bereich steht. Betrachtet man z.B. lediglich Handbewegungen, kann man nicht einmal von einer eigenen Kategorie sprechen. (Goldin-Meadow, 2003)

Ekman und Friesen veröffentlichten 1969 einen Entwurf um nonverbales Verhalten zu klassifizieren und identifizierten dabei fünf Typen:

- *Illustratoren*, dies sind Verhaltensweisen, die das Gesagte untermalen, illustrieren oder verdeutlichen;
- *Adaptoren*, dies sind Verhaltensweisen, die der Erregungsabfuhr oder der Selbststimulierung dienen können;
- *Emblemen*, dies sind Verhaltensweisen, die das gesprochene Wort ersetzen;
- *Regulatoren*, dies sind Verhaltensweisen, die Interaktion steuern;
- *AffektDarstellungen*, dies sind Verhaltensweisen, die Affekte, Stimmungen und Emotionen ausdrücken.

(Schäfer, 2003)

Im folgenden wird eine dieser fünf betrachtet: Illustratoren, auch *Gestikulation* von Kendon (1980) und *Gestik* von McNeill (1992) genannt. All diese Begriffe beschreiben Handbewegungen, die im direkten Bezug zu Gesprochenen stehen. Diese können das Tempo einer Rede vorgeben, auf Referenten verweisen, oder symbolischen Charakter haben, um den Inhalt einer Rede zu verdeutlichen. (Goldin-Meadow, 2003)

Bezüglich Skizzen ermöglicht es Gestik, Meetingteilnehmer Details zu Zeichnungen zu erklären und zu interpretieren. Zusätzlich können hypothetische Modifikationen vorgenommen werden. Gestik bietet einen Mechanismus um Aktivitäten, Größen, Verbindungen, Richtungen oder Blickrichtungen zu kommunizieren. Dantec schildert in (Le Dantec, 2009) die Erfahrungen, die er in einem Architekturdesignmeeting mit Gestik gemacht hat. Über das gesamte Meeting hinweg waren Gesten das primäre Mittel um Designprobleme zu lösen. Trotz der unbeständigen Natur von Gesten, wurden sie wiederholt und effektiv eingesetzt um

*Gesten sind das primäre Mittel um Designprobleme zu lösen.*

komplexe Konzepte zu kommunizieren, ohne zuvor ein spezielles Training absolviert zu haben. Die Effektivität von Gesten und die Art, wie sie zwischen allen Teilnehmer fungierten, zeigte wie mehrere Teilnehmer mit eigenen Spezialisierungen ein verständliches Kommunikationsmittel fanden und so zum gemeinsamen Design beitrugen.

In Dantecs beschriebenen Szenario wurden Gesten auch dazu verwendet, um über Merkmale von Designs zu sprechen, die im zweidimensionalen Design nicht eindeutig hervorgingen. So machte z.B. ein Teilnehmer große schwungvolle Gesten, um die Form und Platzierung kleiner Fenster in einer Kapelle zu beschreiben (siehe [Abbildung 4.10](#)). Die Gesten zeigten mit der Hilfe von den Skizzen ein klares Bild von Form und Größe, boten zudem aber noch mehr. Sie beinhalteten eine metaphorische Qualität, da sie ausdrücken konnten, wie der Raum Ruhe erzeugt. Somit bekräftigten sie den Zweck des Gebäudes, der als Ort der Trauer gilt (siehe [Auszug 4.1](#)). Die Tatsache, dass die Nutzung von Gesten, physikalische Eigenschaften und metaphorische Qualitäten beschreiben kann, bestätigen auch die Forschungsergebnisse von Casasanto und Lozano, die die Rolle von Gesten beim Erarbeiten von abstrakten Konzepten untersuchten ([Casasanto and Lozano, 2006](#)).

Obwohl Gesten typischerweise nicht als Medium gesehen werden, die Informationen speichern (da sie keine permanenten Spuren hinterlassen), fanden Tang & Leifer in ([Tang and Leifer, 1988](#)) einen Beweis, dass Gesten Informationen effektiv in Stücke zerteilen und ins Gedächtnis zurückrufen können - insbesondere dann, wenn die Gesten von anderen nachgeahmt oder schriftlich bzw. skizzenhaft zu Papier gebracht werden. In einer Designstudie beobachteten sie diesen Umstand. Durch Nachahmung einer Geste und dessen Benennung mit der Phrase »slide and tap«, blieb eine Idee in den Köpfen der Teilnehmer verankert.

In der selben Studie wurden diese und auch andere Funktionen von Gesten beobachtet und statistisch erfasst. Die gerade erwähnte Funktion *Speichern von Information* wurde einmal verzeichnet, das *Verdeutlichen von Ideen* 24 mal, *Hinweisen auf Ideen* neun mal und *Aufmerksamkeit erlangen* 46 mal (vgl. [Abbildung 1.3](#)). Wie man durch die Statistik erkennen kann, spielen Gesten eine wichtige Rolle in kollaborativen Situationen, da sie hauptsächlich dazu benutzt werden, um anderen Personen Aktionen zu demonstrieren und die Aufmerksamkeit auf bestimmte Orte zu lenken.

Tang & Minneman konzentrierten sich in ([Tang and Minneman, 1991](#)) auf die Benutzung von Handgesten in Verbindung mit Zeichnungen. Handgesten treten oft in Verbindung mit Skizzen auf, um Informationen zu verdeutlichen. Aus diesem Grund

entwickelten sie VideoDraw, ein kollaboratives Skizziersystem, das das Zusammenspiel von Skizzen und Gesten unterstützt. Das VideoDraw Setup besteht aus zwei Videokameras, die die Arbeitsflächen der Teilnehmer aufnimmt und direkt an den jeweilig gegenüberliegenden Bildschirm überträgt (vgl. [Abbildung 4.11](#)). Die Bildschirme fungieren als Whiteboard, auf das jeder Teilnehmer direkt mit Marker zeichnen kann. Die Zeichnungen *und* Handgesten der Teilnehmer werden somit durch die Kamera erfasst und zum anderen Teilnehmer übertragen.

Handgesten können in VideoDraw beispielsweise eingesetzt werden, um einem Teilnehmer die geplante Bedienung eines User Interfaces zu erklären. Die Effektivität dieser Art von Gesten hängt von der Aufrechterhaltung der Verbindung zwischen den Händen und den Skizzen am VideoDraw Screen ab.

Adam	that wasn't the idea I was anticipating that the hearses would be parked here [sketches]
Anna	were there OK that's fine yeah
Adam	exactly as they are at the moment that the coffin would be drawn out here and they would simply [points] walk it in I wasn't thinking that they'd try and park
Anna	no that's OK
Adam	in there
Anna	yes well that's what they're wondering how that would work then so we'd work I wasn't quite aware
Adam	we'd work it exactly the same way as the present system I mean maybe this should be made more obvious by perhaps a different colour in the paving or something [sketches] I mean what I'm trying to say here is that that's the vehicular line [sketches]
Anna	yes
Adam	and that these areas [points] are for people to mill about in and you've got a place for people to stand
Anna	yes they will probably want to know how [points] how + how far that is from the because they're going to be possibly carry the coffins in and most of the men are sort of in their seventies and eighties [laughs] carrying the coffin

Auszug 4.1: Transkript eines Designmeetings. Skizzen und Gestik dienen als Ausdrucksmittel.

Zusätzlich übermittelt VideoDraw auch Gesten mehrerer Hände und/oder Finger. Diese Kommunikation der Gesten ist reichhaltiger als die der meisten Computersysteme (typischerweise

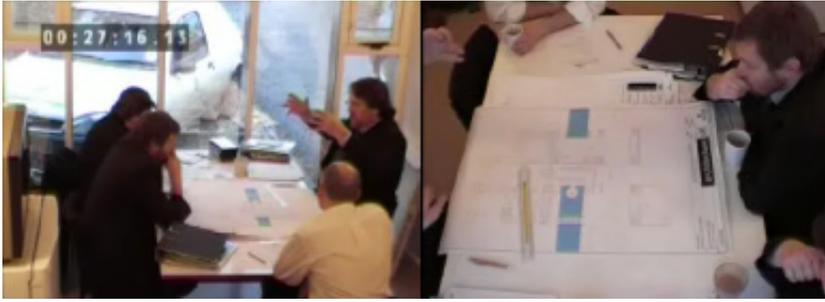


Abbildung 4.10: Ein Teilnehmer gestikuliert um Gebäudemerkmale zu beschreiben.

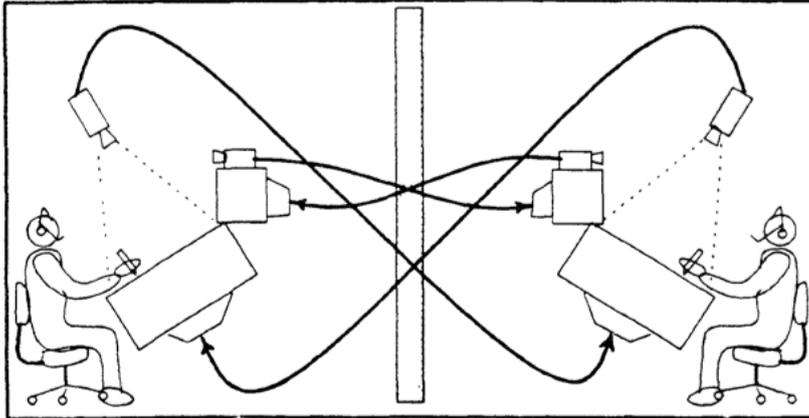


Abbildung 4.11: Schematische Darstellung von VideoDraw in einem Zwei-Personen-Szenario an verschiedenen Orten.

stellen diese lediglich einen mausgesteuerten Cursor zu Verfügung). VideoDraw überträgt ebenso das Gefühl von Dreidimensionalität. Benutzer verfügen über »räumliche« Gesten und können sogar alltägliche Gegenstände ins Kamerabild holen, wodurch eine dreidimensionale Wahrnehmung von Raumverhältnissen den anderen Teilnehmern vermittelt werden kann.

Durch Observationen von Tang & Minnemans System lassen sich vier Hauptmerkmale zusammenfassen.

VideoDraw:

- übermittelt Handgesten unter den Teilnehmern,
- verursacht keine problematischen Verzögerungen in der Interaktion,
- bietet eine neuartige Wahrnehmung von räumlichen Beziehungen zwischen den Teilnehmern und ihrer Zeichenoberfläche und
- erlaubt mehreren Teilnehmern gleichzeitiges Arbeiten auf der gleichen Arbeitsfläche.

VideoDraw erlaubt Teilnehmern gewohnte Tätigkeiten auf eine neue Art zu erleben. Es ermöglicht Benutzern ihre Zeichenoberfläche mit anderen Teilnehmern am selben oder an einem anderen Ort zu teilen - ohne dass dabei Verwirrungen in der Interaktion entstehen. Es ermöglicht zudem die Hände der Teilnehmer näher aneinander zubringen, als es in der Realität je möglich wäre, ohne die Teilnehmer gegenseitig zu behindern. (Tang and Minneman, 1991)

#### 4.4 CASE STUDY - DIGITAL AND PAPER MEDIA

2010 führten Hinckley et al. eine Studie durch, in der sie mittels Observationen herausfinden wollten, wie Personen mit Papier, Stiften und Hilfsmittel arbeiten. Sie stellten den Probanden die Aufgabe, Ideen zu einem hypothetischen Kurzfilm zu illustrieren. Zur Verfügung standen ihnen Notizblöcke, Scheren, Stifte, Klebeband und 20 Seiten an inspirierendem Material. Acht Personen nahmen an der Studie teil und unterstützten so Hinckley et al. auf der Suche nach Verhaltensmustern in Sachen Gestik und Arbeitsplatzstrukturierung. Neun Verhaltensweisen (V1-V9) stachen besonders hervor:

- V1) Teilnehmer klemmten den Stift zwischen die Finger der starken Hand, wenn sie vom Schreiben zum Schnipselverschieben wechselten (Abbildung 4.12a).
- V2) Teilnehmer hielten Schnipsel mit einem Finger der schwächeren Hand an einer Stelle (Abbildung 4.12a).
- V3) Teilnehmer tendierten dazu, Schnipsel mit der schwächeren Hand festzuhalten, wenn sie darauf schrieben (Abbildung 4.12b).
- V4) Eine häufig eingesetzter Handgriff war das Halten eines Schnipsels mit Daumen und Zeigefinger während dem Schreiben (Abbildung 4.12b).
- V5) Teilnehmer benutzten nur Teile des inspirierenden Materials. Sie schnitten Schnipsel über dem Notizblock indem sie das Material mit der schwächeren Hand hielten und die starke Hand zuschnitt (Abbildung 4.12c).
- V6) Teilnehmer schoben das Notizbuch nahe zu ihrem Körper, wenn sie zu den darüber liegenden Hilfsmittel fassten (Abbildung 4.12d).
- V7) Das Anhäufen von Schnipsel war ein häufiges Verhalten. Benutzer formten Stapel aus »interessanten« Objekten, während sie die übrigen in der schwächeren Hand hielten.

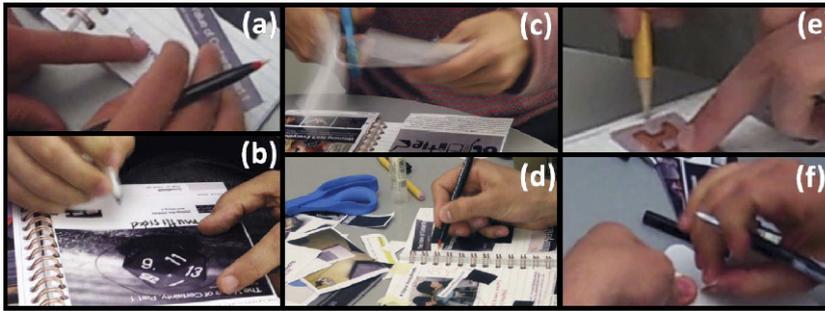


Abbildung 4.12: Verhaltensweisen bei der Benutzung von Notizbüchern. *a)* Benutzer klemmen Stifte zwischen ihre Finger, während sie Objekte bearbeiten. *b)* Daumen und Zeigefinger halten ein Objekt fest, während darauf geschrieben wird. *c)* Papierschnipsel fallen auf die Arbeitsfläche. *d)* Benutzer greifen oft nach Hilfsmitteln und neuen Inhalten, die über dem Notizbuch liegen. *e)* Zeichnen eines Rahmens mit einer Schablone, durch Halten der Schablone mit der Hilfshand und Nachfahren der Kanten mit dem Stift. *f)* Reißen einer Seite durch Festhalten des Papiers mit dem Daumen und Ziehen mit den Fingern der anderen Hand. Der Stift ist währenddessen wiederum zwischen den Fingern eingeklemmt.

- V8) Einige missbrauchten Schnipsel als Schablone, um einen Rahmen um ein Objekt zu zeichnen ([Abbildung 4.12e](#)).
- V9) Das Reißen von Papier war eine zweihändige Arbeit, die mit den Fingern vollzogen wurde ([Abbildung 4.12f](#)).

(Hinckley et al., 2010)

Piper & Hollan suchten im Weiteren den direkten Vergleich von Papier zu digitalen Medien und starteten dazu eine Untersuchung, in der sie Paare von Studenten mit Papier und digitalen Materialien auf Tabletop-Umgebungen arbeiten ließen. Die digitale Umgebung umfasste ein Tabletop-Display (vgl. [Fußnote 11](#) in [Abschnitt 1.3](#)) mit einer *Single Display Groupware (SDG)*<sup>6</sup> - die andere enthielt Papier, Hilfsmittel und einen Tisch.

20 Studenten einer neurowissenschaftlichen Einführungsveranstaltung nahmen fünf Wochen an der Untersuchung teil. Die Studenten mussten mit den vorgegebenen Systemen arbeiten und am Ende der Lehrveranstaltung in der Lage sein, die Anatomie des Gehirnes aufzusagen, komplexe Systeme (wie z.B. die Entladung einer Nervenzelle) zu verstehen und zu beschreiben,

<sup>6</sup> Single Display Groupware (SDG) ist eine Spezialform von Groupware (vgl. [CSCW & GROUPWARE](#)) und beschreibt eine Displaytechnologie, die Eingaben von mehreren Benutzern gleichzeitig unterstützt (Stewart, 1997). SDG<sup>1</sup> führt zu höherem Engagement und gerechteren Aufgabenteilung (Stewart et al., 1999).

sowie verschiedene Graphen und Schaltungen von Gehirnaktivitäten zu erstellen. Jede Teiluntersuchung umfasste folgende drei, vom Professor ausgewählte, Aktivitäten:

1. das Beschriften der Gehirnanatomie,
2. das Auseinandersetzen mit einem dynamischen System und
3. das Zeichnen eines Graphen oder einer Schaltung.

Das Ziel jeder Teiluntersuchung war, die Studenten auf ein kommendes Examen vorzubereiten. Die Arbeitsmittel beider Gruppen (analog, sowie digital Arbeitende) wurden soweit wie möglich angeglichen. Lediglich die Größe der Arbeitsfläche unterschied sich von der 79cm langen Bildschirmdiagonale zu den ca. 35cm Diagonale eines Blatt Papiers im A4 Format.

Papier- und Digitalmedien haben einzigartige und ergänzende Merkmale für kleine Lerngruppen. Die Studenten, die mit Papiermaterialien gearbeitet hatten, machten detaillierte Notizen und erarbeiteten meist ernsthaftere Arbeitsstrategien. Auf der anderen Seite diskutierten die Studenten, die am Tabletop-Display arbeiteten, ihre Ideen, bevor sie sich die Antworten ansahen, wiederholten Aktivitäten und schnitten besser im Endexamen ab. (Piper and Hollan, 2009)

Während die folgenden Ergebnisse der Studie auf die Aktivitäten von Lerngruppen ausgelegt sind, können trotzdem auch allgemeine kognitive und soziale Merkmale in Bezug auf die Arbeit in beiden Welten (traditionell & digital) entnommen werden.

#### 4.4.1 *Kognitive Merkmale*

Papier ist das traditionelle Medium für Schulungsunterlagen. In Lerngruppen interagieren Studenten üblicherweise mit Papierdokumenten. Dadurch, dass die Teilnehmer der Studie keine vorgehende Erfahrung mit einem kollaborativen Tabletop Umgebung hatten, hatten sie anfangs geringe Schwierigkeiten sich an das System zu gewöhnen. Jedoch förderte die Neuheit des digitalen Mediums einen gewissen Freiraum zur Interaktion. Munter kritzelten die Studenten mit den digitalen Stiften darauf los, während der Enthusiasmus bei den Studenten, die mit den traditionellen Mitteln arbeiteten sich in Grenzen hielt. Ebenso, ermutigten der geringe Aufwand und die geringen Konsequenzen (wenn etwas falsches gezeichnet wird, kann es leicht wieder gelöscht werden) die Studenten spontan Zeichnungen zu erstellen um die Diskussion zu unterstützen.

Diese spontanen Skizzen können den Lernprozess ebenso in kognitiver Art unterstützen; die Skizzen bieten eine externe Abbildung der Diskussionen und bieten eine Grundlage um ein

Verständnis zu entwickeln. Studenten, die mit Papierdokumenten arbeiteten, waren eher dazu geneigt die vorgegebenen Diagramme zu wenden, um auf der Rückseite zu zeichnen, anstatt direkt auf dem Bild. Die digitalen Medien ermutigten die Studenten nicht nur direkt auf ein Diagramm zu zeichnen, sondern auch die Radierfunktion öfter einzusetzen. Notizen auf Papierdokumenten wurden weniger ausradiert. Dadurch konnten die Studenten, die mit Papier arbeiteten, zurück gehen und ihre alten Notizen und Skizzen durchsehen, was bei den digitalen Materialien oft nicht mehr möglich war. Natürlich muss dies nicht zwangsweise so ablaufen. Die Formbarkeit digitaler Medien erlaubt zahlreiche interessante Lösungsansätze. Die Möglichkeit digitale Annotierungen zu löschen führt zu kognitiven Folgen, die nähere Untersuchungen benötigen.

Digitale Tische können ein dynamischeres und umfassenderes Erlebnis ermöglichen, als traditionelle Papierdokumente. Das hat positive wie negative Konsequenzen. Studien über das Lernen mit Diagrammen zeigen, dass digital animierte Diagramme Personen dazu zwingen, das Diagramm wahrzunehmen, es richtig zuzuordnen und dann Diagrammänderungen durchzuführen, was zu einer höheren kognitiven Belastung führt, als statische Papierdiagramme (Price, 2002). Andererseits ermöglichen große Touchscreens und interaktive Möglichkeiten, Formen an Interaktionen, die sich von den Papiermaterialien unterscheiden. Kognitionstheorien und die neuesten empirischen Forschungen zeigen, dass uns die körperliche Belastung hilft und zwingt abstrakte Konzepte zu verstehen (Clark, 1996; Johnson, 1987; Nunez, 1999; Varela et al., 1991). Das Zeichnen oder Nachverfolgen von Graphen mit dem Finger, ist ein konkreter kognitiver Prozess, der möglicherweise zu einem besseren Verständnis und Verinnerlichung eines abstrakten Konzeptes führt (Goldin-Meadow, 2003). Auszug 4.2 zeigt zwei Aussagen von Studenten, die mit digitalen Dokumenten arbeiteten. Piper & Hollan glauben, dass konkrete Wahrnehmung eine zentrale Rolle im Verstehen kognitiver Aspekte im Zusammenhang mit Multitouchoberflächen spielen. (Piper and Hollan, 2009)

Student D1	Want to redraw it together to help memorize it?
	...
Student D2	You want to try tracing it [the answer key]? It's good practice.

Auszug 4.2: Zwei typische Aussagen beim Arbeiten mit einem digitalen Tabletop-System.

#### 4.4.2 Soziale Merkmale

Tabletop-Displays erlauben durch ihre Größe und ihre gemeinsame Nutzbarkeit, einen gleichmäßigen Zugang zu Materialien und erlauben gleichzeitiges Arbeiten. Aber verbessert dies den Lernprozess? Es gibt interessante Unterschiede zwischen den von Piper & Hollan beobachteten parallelen Arbeitsweisen mit digitalen Dokumenten und seriellen Arbeitsweisen mit Papierdokumenten. Zum einen bedeutet paralleles Arbeiten, dass mehrere den gleichen und direkten Zugang zu Materialien haben. Zum anderen kann dies aber dazu führen, dass man wichtige Teile der Aktivitäten bzw. des Problemlösungsprozess verpasst. Eine serielle Arbeitsweise hat positive und negative Auswirkungen auf den Lernprozess. Während sich beide Studenten gemeinsam auf eine Aufgabe konzentrieren, macht einer den Hauptteil der Arbeit. Der Partner nimmt eine passive Rolle an.

Idealerweise würde die Lernumgebung Studenten dazu bewegen, sich gleichermaßen zu beteiligen und den Fokus auf die Aufgabe zu richten. Tabletop-Technologien erlauben Unterrichtenden zumindest die Darstellung der Lehrmittel soweit zu verändern, damit soziale Barrieren vermieden werden und eine ausgeglichene Beteiligung herrscht. (Piper and Hollan, 2009)

#### ZUSAMMENFASSUNG

Traditionelle und digitale Medien haben unterschiedliche Stärken und Schwächen. Papierdokumente sind billig, portabel und perfekt geeignet, um mit beiden Händen und einfachen Geräten, wie z.B. Stiften, daran zu arbeiten. Digitale Dokumente können hingegen mit Leichtigkeit editiert und schnell abgelegt bzw. abgerufen werden. Trotzdem kommt das Arbeiten mit einem strukturierten Softwaretool nicht an die Schnelligkeit von unstrukturierten Papier-und-Stift Skizzen heran.

Um auf Computer ähnlich gut skizzieren zu können, wurden Eingabegeräte wie Tablets geschaffen. Die Interaktion mit Tabletstiften unterscheidet sich jedoch stark von der der Maus. Aus diesem Grund wurden eigene Interaktionstechniken für stiftbasierte Skizziersysteme entwickelt. Skizziersysteme besitzen oft mehrere Programmmodi um die Funktion der Stifte zu bestimmen. Dabei kommt es oft zu kognitiven Überlastungen bei den Benutzern, welche mit Hilfe einiger Strategien vermieden werden sollten.

Gestik ist ebenso ein wichtiger Faktor im kollaborativen Designprozess und kann bei digitalen Arbeitsweisen berücksichtigt werden. Weitere Faktoren und Eigenschaften von kollaborativen computerbasierten Systemen werden nun im nächsten Kapitel erläutert.

Der Begriff »Computer Supported Cooperative Work« (CSCW) bezeichnet ein multidisziplinäres Forschungsgebiet, das die kooperative Zusammenarbeit mehrerer Gruppen untersucht und Technologien zu ihrer Unterstützung entwickelt. CSCW existiert seit den frühen achtziger Jahren. Um herauszufinden, wie die Technik Menschen bei ihrer Zusammenarbeit unterstützen kann, organisierten Irene Greif und Paul Cashman im Jahre 1984 einen Workshop für Personen, die sich mit der Arbeitsweise von Menschen auseinandersetzten (Grudin, 1994). Unter diesen Personen befanden sich Spezialisten aus verschiedenen wissenschaftlichen Bereichen, wie zum Beispiel Ökonomie, Sozialpsychologie, Anthropologie, Ethnologie und Pädagogik. Dieser Workshop war der Versuch der Techniker, Teamarbeit besser zu verstehen und folglich unterstützende Technologien entwickeln zu können (Grudin, 1994; Rama and Bishop, 2006). Seither hat CSCW sich zu einem nahezu riesigen wissenschaftlichen Forschungsgebiet entwickelt, dem sich heute unzählige Experten widmen. Trotzdem sind sich die Wissenschaftler nicht immer ganz einig bei der Definition des Begriffes CSCW. Die Bedeutung von »Cooperative Work« erscheint nicht eindeutig und führt häufig zu unterschiedlichen Interpretation seitens wissenschaftlicher Autoren (Gerlicher, 2007). Einige setzen »Cooperation« gleich mit »Collaboration«, andere hingegen unterscheiden die beiden Begriffe sehr strikt. Dillenbourg et al. definieren die Begriffe beispielsweise so:

»Cooperation and collaboration do not differ in terms of whether or not the task is distributed, but by virtue of the way in which it is divided; in cooperation the task is split (hierarchically) into independent sub-tasks; in collaboration cognitive processes may be (heterarchically) divided into intertwined layers. In cooperation, coordination is only required when assembling partial results, while collaboration is “...„ a coordinated, synchronous activity that is the result of a continued attempt to construct and maintain a shared conception of a problem.«

(Dillenbourg et al., 1995)

Die beiden Begriffe unterscheiden sich also nicht darin, ob Arbeit auf mehrere Individuen aufgeteilt wird oder nicht, sondern

in der Art und Weise, *wie* sie aufgeteilt wird. Bei »Cooperation« wird die Arbeit in einzelne, unabhängige Module aufgeteilt, die zuerst abgewickelt und danach wieder zu einem Ganzen zusammengesetzt werden können. Koordination wird in diesem Fall hauptsächlich beim Zusammenfügen der Module benötigt. »Collaboration« hingegen ist eine koordinierte, synchrone Aktivität und resultiert aus dem fortwährenden Versuch, eine gemeinsame Auffassung eines Problems zu konstruieren und zu erhalten.

In dieser Arbeit werden die beiden Begriffe synonym verwendet, da sich beide aus dem Englischen als »Zusammenarbeit« übersetzen lassen. Um den Begriff CSCW jedoch besser zu verstehen, sollen weitere Definitionen von »Cooperative Work« begutachtet werden.

Die Ökonomie kennt den Begriff »Cooperative Work« schon sehr lange und Marx definiert ihn als mehrere Individuen, die gemeinsam in einer koordinierten Art und Weise an einem oder mehreren, zusammengehörigen Produktionsprozessen arbeiten (Marx, 1867). Im deutschen Sprachraum wurde diese Definition im 19. Jahrhundert häufig durch andere Autoren wiederverwendet, jedoch gibt es sehr viele Formen von »Cooperative Work« und es gibt in der Literatur keine klare Trennung zwischen Begriffen wie »Cooperative Work«, »Collaborative Work«, »Collective Work« und »Group Work« (Bannon and Schmidt, 1990).

Die Definition von »Cooperative Work« nach Bannon und Schmidt besagt, dass sie im Allgemeinen aus Arbeitsprozessen besteht, die ein Produkt oder ein Service hervorbringen. Ein typisches Merkmal dieser Arbeitsform ist die Planung der Abwicklung, die vor Beginn der eigentlichen Arbeit durchgeführt wird. »Cooperative Work« beinhaltet *direkte, indirekte, verteilte* und *kollektive* Arten der Interaktion. *Kollektiv* in einer Gruppe geleistete Arbeit, sprich »Group Work« ist lediglich eine spezielle Form von »Cooperative Work«. Es ist aber ebenfalls möglich, dass »Cooperative Work« von einer geographisch *verteilten* Gruppe halb autonomer Personen geleistet wird, die bei der Abwicklung strategisch nach eigenem Ermessen vorgehen. Zusätzlich kann »Cooperative Work« *indirekt*, über den Arbeitsprozess selbst, als auch *direkt*, durch Kommunikation zwischen den einzelnen Gruppenmitgliedern durchgeführt werden. (Bannon and Schmidt, 1990)

Die Erkenntnisse, die bei der Erforschung von CSCW gewonnen werden, verwendet man dazu, sinnvolle Groupware zu konzipieren. Groupware bezeichnet also die technische und praktische Umsetzung, die auf den Theorien der CSCW-Forschung basiert. Alle Systeme, Applikationen und Werkzeuge, die CSCW unterstützen, können somit unter dem Begriff Groupware zusammengefasst werden (Koch, 2008; Gerlicher, 2007). Häufig werden diese Systeme auch als »kollaborative Software« bezeichnet (Bannon and Schmidt, 1990). Gerlicher definiert Groupware wie folgt:

»Der Begriff Groupware bezeichnet ein aus Software und eventuell spezifischer Hardware bestehendes System, das die Zusammenarbeit im Team durch die Schaffung von Kommunikations- und/oder Koordinationslösungen unterstützt oder ermöglicht.«

(Gerlicher, 2007)

Michael Koch definiert Groupware als:

»[...] Nevertheless, there are some technologies and tools [Groupware] that facilitate shaping [...] socio-technical systems. For these technologies and tools the term Groupware is used. In contrast to traditional computer systems that are primarily designed for a single user, the major goal of Groupware is to assist a group of users in communicating, in collaborating, and in coordinating their activities.«

(Koch, 2008)

Groupware unterscheidet sich also von »Ein-Benutzer-Systemen« darin, dass sie für mehrere Benutzer konzipiert ist und versucht, die höchste Effizienz und Flexibilität für die Teamarbeit dieser Benutzer zu gewährleisten. Gerlicher nennt die folgenden Gründe, warum Groupware eingesetzt wird:

- Verbesserung der Kommunikation in der Gruppe
- Ermöglichen von Kommunikation, wo sie sonst nicht möglich wäre
- Ermöglichen von Telearbeit (für geographisch entfernte Benutzer)
- Einsparung von Reisekosten
- Bündelung von Fachwissen und Gedankenaustausch in der Gruppe
- Bildung von Gruppen mit gemeinschaftlichem Interesse
- Reduktion von zeitlichem und finanziellen Aufwand bei der Koordination von Gruppen
- Problemlösung in der Gruppe
- Erlangung neuer Kommunikationsmöglichkeiten, beispielsweise dem anonymen oder strukturierten Informationsaustausch

(Gerlicher, 2007)

Demgegenüber steht die Schwierigkeit, erfolgreiche Groupware Systeme zu entwickeln. Im Gegensatz zu single-user Software ist dies äußerst komplex und es gibt viele kritische Faktoren. Zum einen sind die technischen Voraussetzungen diffiziler, weitaus entscheidender ist jedoch, dass Groupware von der Gruppe lebt und daher ist die Akzeptanz durch die Zielgruppe maßgebend für den Erfolg von Groupware (Gerlicher, 2007).

CSCW soll aber nicht nur Technologien und Werkzeuge zur Unterstützung von Kollaboration bereitstellen, sondern auch soziotechnische Systeme entwickeln (Koch, 2008). In den 1950er Jahren wurden Experimente durchgeführt, bei denen technische Systeme in mehrere verschiedene soziale Gruppen eingeführt wurden. Dabei stellte man fest, dass die Technologieeinführung in den Gruppen sehr unterschiedliche Auswirkungen hatte. Dadurch kamen die Wissenschaftler zum Schluss, dass das technische, als auch das soziale System auf diese Zusammenführung hin optimiert werden müssen, um ein erfolgreich funktionierendes, einheitliches Gesamtsystem hervorzubringen (Koch, 2008). Wenn diese Optimierung nicht passiert und bei der Einführung der Technik die sozialen Aspekte und Eigenheiten nicht berücksichtigt werden, kann das Ergebnis nur suboptimal sein. Soziale Prozesse sind die Basis für die Entwicklung und das Design von neuen Technologien und Systemen und umgekehrt strukturieren diese Technologien die Möglichkeiten des sozialen Austauschs. Für den Erfolg von neuen Technologien ist es unerlässlich, diese sozialen und technischen Aspekte bei Entwicklung und Design mit einzubeziehen (Mumford, 2000).

Das Forschungsgebiet CSCW versucht also soziale Interaktion in Gruppen und Teams zu verstehen und technische Systeme zur Unterstützung selbiger zu konzipieren, zu entwickeln und zu evaluieren (Koch, 2008).

## 5.1 KLASSIFIKATION VON CSCW

CSCW kennt zwei Dimensionen: Raum und Zeit. Dies sind die Kriterien, nach denen CSCW und Groupware eingeteilt werden. **Abbildung 5.1** illustriert die Kategorisierung von CSCW und ordnet den vier sich aus der Überschneidung der Dimensionen ergebenden, Quadranten einige Beispiele zu.

Der erste Quadrant umfasst alle Systeme und Technologien, bei denen die Benutzer zur selben Zeit am selben Ort sein müssen. Beispielsweise könnten sich zwei Personen in einem Meetingraum treffen und gemeinsam Konzepte an einem digitalen Whiteboard ausarbeiten. Sie teilen zeitliche und räumliche Komponenten miteinander.

		TIME	
		Same Time (Synchronous)	Different Time (Asynchronous)
SPACE	Same Space	<b>1<sup>st</sup> Quadrant</b> Spontaneous collaborations, formal meetings, classrooms	<b>2<sup>nd</sup> Quadrant</b> Design rooms, Project scheduling
	Distributed	<b>3<sup>rd</sup> Quadrant</b> Video conferencing, net meetings, phone calls	<b>4<sup>th</sup> Quadrant</b> Emails, blogging, authoring, voice mails, fax

Abbildung 5.1: Die Grafik illustriert die Unterteilung von CSCW in vier Kategorien, die sich durch ihre räumlichen und zeitlichen Eigenschaften unterscheiden. Der Faktor Zeit definiert, ob Zusammenarbeit in einer Gruppe synchron oder asynchron passiert und der Faktor Raum definiert, ob sie am selben oder an geographisch distanzierten Orten stattfindet.

Der zweite Quadrant kategorisiert all jene Technologien, die Personen am selben Ort zu unterschiedlichen Zeiten nutzen können. Ein Designraum mit elektronischen Geräten könnte ein Beispiel dafür sein. Angenommen, eine Person überlegt sich ein Konzept und zeichnet Skizzen auf ein Tabletop Display. Zu einem späteren Zeitpunkt, als die Person den Raum bereits verlassen hat, kommt eine andere Person herein, arbeitet mit den Skizzen weiter und entwickelt eigene Konzepte, die sie wiederum auf dem Tabletop festhält.

Zum dritten Quadranten gehören alle Systeme, die zur selben Zeit an verschiedenen Orten eingesetzt werden können. Das wohl bekannteste dieser Systeme ist das Telefon. Es kann nur von zwei Personen gleichzeitig benutzt werden, aber die Personen können sich dazu an beliebigen Orten befinden.

Im vierten Quadranten befinden sich alle Technologien, die zu unterschiedlichen Zeiten an unterschiedlichen Orten zum Einsatz kommen. E-Mail, das elektronische Nachrichtensystem, ist ein gutes Beispiel solcher Groupware. Um auf diesem Wege zu kommunizieren, müssen zwei Personen weder gleichzeitig damit interagieren, noch sich am selben Ort befinden.

Gerlicher definiert zusätzlich eine weitere Kategorie in [CSCW](#), auf die im Abschnitt [Multisynchrone Systeme](#) näher eingegangen werden soll, nachdem die vier grundlegenden Kategorien im folgenden näher definiert werden.

### 5.1.1 Asynchrone Systeme

Unter asynchroner Groupware versteht man Systeme, die keine Echtzeitanforderungen erfüllen müssen. Das bedeutet, dass die Nutzung dieser Systeme im Normalfall zeitversetzt erfolgt. Die wohl bekannteste asynchrone Groupware-Anwendung ist die digitale Übermittlung von Textnachrichten in Form einer *E-Mail*. Diese Nachrichten können an und von mehreren Personen versendet, empfangen und weitergeleitet werden. Dabei müssen Absender und Empfänger nicht gleichzeitig online sein, denn die *E-Mail* kann zu jedem beliebigen Zeitpunkt vom Empfänger abgerufen werden.

Eine andere, jedoch der *E-Mail* sehr ähnliche Form asynchroner Groupware, sind *Newsgroups* und *Mailinglisten* (Gerlicher, 2007). Sie dienen dem Nachrichtenaustausch in einer größeren Gruppe an Benutzern. *Newsgroups* zeigen Nachrichten nur dann an, wenn ein Benutzer diese direkt anfordert, *Mailinglisten* hingegen werden automatisch an all jene Personen weitergeleitet, welche die entsprechende Liste abonniert haben.

Auch *Workflow-Management-Systeme* werden von Gerlicher (Gerlicher, 2007) zur asynchronen Groupware gezählt, da sie die Arbeit unterschiedlicher Personen in einer Organisation regeln. Sie dienen der Steuerung arbeitsteiliger Prozesse. Man bezeichnet sie auch als *Geschäftsprozess-Management-Systeme*.

Das World Wide Web (WWW) ist ein *Hypertext-basiertes System* und gehört ebenfalls zur asynchronen Groupware (Gerlicher, 2007), da es einer sehr großen Anzahl an Personen ermöglicht, auf digitalem Wege Informationen auszutauschen. *Hypertext-basierte Systeme* nutzen das Internet oder ein bestimmtes Intranet, um ihre Dienste über einen Webbrowser zugänglich zu machen. Solch eine Webschnittstelle kann prinzipiell für fast jede Art der asynchronen Groupware implementiert werden.

Schlussendlich nennt Gerlicher noch *Gruppenkalender* als asynchrones Groupware System. Der *Gruppenkalender* ist ein Tool, das sehr häufig in größeren Unternehmen eingesetzt wird. Er ermöglicht die Terminplanung und Koordination von vielen Personen. Terminkonflikte werden automatisch erkannt und das System kann eigenständig jene Zeiträume finden, zu denen jeder erforderliche Teilnehmer eines Meetings frei zur Verfügung steht. Dies setzt jedoch eine gute Datenpflege aller Benutzer voraus und wird teilweise als unangenehmer Eingriff in die Privatsphäre empfunden (Gerlicher, 2007).

### 5.1.2 Synchroner Systeme

Dem gegenüber stehen synchrone Groupware Systeme, die sich dadurch charakterisieren, dass Benutzer zeitgleich auf Daten und Informationen zugreifen (Gerlicher, 2007). Ein Beispiel dafür sind *elektronische Tafeln*, denn sie erlauben mehreren Nutzern, die sich an unterschiedlichen Orten befinden, auf eine für alle sichtbare Fläche zu zeichnen. Häufig werden solche Tafeln bei Besprechungen eingesetzt, bei denen die Teilnehmer nicht im selben Raum sind. Dadurch wird es möglich, Konzepte und Ideen besser zu kommunizieren und für die anderen greifbarer zu machen.

*Application Sharing Systeme* zählen ebenfalls zur synchronen Groupware und gestatten das Teilen von Drittapplikationen mit anderen Personen (Gerlicher, 2007). Man bezeichnet diese Anwendungen im Englischen auch als *Remote Desktop*<sup>1</sup>. Dadurch wird es möglich, mit geografisch distanzierten Kollegen an jedem beliebigen Programm zu kooperieren. Beide Benutzer sehen die Applikation, die auf einem der Computer läuft und können mit ihr interagieren.

Weitere synchrone Systeme sind *Video- und Multimediale Konferenzsysteme* (Gerlicher, 2007). Erstere übertragen Video und Audio an mehrere verteilte Computer und werden eingesetzt, um virtuelle Meetings abzuhalten. Die Teilnehmer befinden sich meist an verschiedenen Orten und können so fast genauso kommunizieren, als würden sie sich in einem Raum befinden. Letztere bieten zusätzlich die Möglichkeit, multimediale Inhalte (beispielsweise Präsentationen) an die Teilnehmer der Konferenz zu übertragen. Systeme zur Entscheidungsfindung sind oft ein integraler Bestandteil solcher Software und bieten Werkzeuge für Ideenfindung und Brainstorming. Über auf diesem Wege generierte Konzepte kann dann abgestimmt und so die Spreu vom Weizen getrennt werden.

*Chat-Systeme*, ebenfalls der synchronen Groupware zuzuordnen, sind weitgehend bekannt und werden sehr häufig eingesetzt. Benutzer können durch diese Anwendungen in Echtzeit untereinander Textnachrichten austauschen und so eine Konversation führen, als säßen sie sich gegenüber.

### 5.1.3 Multisynchrone Systeme

*Versionsverwaltungssysteme* werden von Gerlicher als multisynchrone Systeme definiert (Gerlicher, 2007). Häufig werden diese in Softwareprojekten und auch bei der Erstellung von umfangreichen Dokumenten, beispielsweise Büchern eingesetzt. Das System übernimmt die Verwaltung der Dateien und sichert die

<sup>1</sup> Zu Deutsch: Entfernter Schreibtisch. Gemeint ist damit die virtuelle Schreibtischfläche des Betriebssystems.

Konsistenz der Daten. Es wird dadurch möglich, dass mehrere Personen gleichzeitig an der selben Datei arbeiten. Tatsächlich haben aber alle Benutzer eine eigene Kopie der Datei und das System synchronisiert die Dateien im Nachhinein selbständig. Ein weiterer großer Vorteil von *Versionsverwaltungssystemen* ist die Nachvollziehbarkeit von Änderungen (Gerlicher, 2007) an einzelnen Dateien. Das System legt für jede neue Version einer Datei ein Backup ab, das zu einem späteren Zeitpunkt bei Bedarf wiederhergestellt werden kann. Es ist zu jedem Zeitpunkt klar, wer welche Änderungen an einer bestimmten Datei durchgeführt hat.

Es wird deutlich, dass CSCW und Groupware sehr weitläufig sind und viele verschiedene Bereiche betreffen. Daher ist es bis heute immer noch eine große Herausforderung geblieben, bedienbare, effiziente und intuitive Systeme zu entwickeln, die die Arbeit der Benutzer tatsächlich optimieren und keinen Mehraufwand verursachen. Der folgende Abschnitt widmet sich den Problemfeldern, die damit zusammenhängen.

## 5.2 DESIGN VON GROUPWARE

Eine formale Beschreibung von Groupware Lösungen zu geben ist äußerst schwierig, da sie stark von den Menschen bzw. den organisatorischen Umständen abhängen von und in denen sie eingesetzt werden (Suchman, 1995). Obwohl also alle Groupware Systeme mit eigenen Anforderungen und Gegebenheiten zurecht kommen müssen, gehen Entwickler beim Design normalerweise von bekannten und bewährten Lösungen aus und versuchen dann, diese für die neue Situation anzupassen (Herrmann et al., 2003). Diese Vorgehensweise ist natürlich völlig legitim, kann aber zu Schwierigkeiten führen wenn es nicht gelingt, die Expertise erfahrener Entwickler auf die Kollegen zu übertragen. Eine gute Dokumentation ist daher von großer Bedeutung, denn sie erlaubt es den Entscheidungsträgern im Projekt, schon sehr früh mit Endbenutzern über Funktionalität und Design zu diskutieren. Herrmann et al. schlagen daher vor, sogenannte Patterns<sup>2</sup> zu verwenden (Herrmann et al., 2003).

Die Idee, Wissen und Problemlösungen in Form von Patterns zu archivieren, gibt es schon länger und Gamma et al. haben dieses Konzept der Patterns von urbaner Architektur auf Softwarearchitektur umgemünzt und eine Kollektion von Patterns für objektorientierte Softwareentwicklung veröffentlicht (Gam-

<sup>2</sup> aus dem Englischen: »Muster« beschreiben zum einen ein Problem, das immer wieder in einer gegebenen Umgebung auftritt und bieten eine generische Lösung dafür. Diese Lösung muss so formuliert werden, dass sie jedes mal (nahezu) unverändert eingesetzt werden kann. (Herrmann et al., 2003)

ma et al., 1995). Im Bereich HCI bietet Jan Borchers einige gute Beispiele für User Interface Patterns (Borchers, 2000).

Da Groupware stark abhängig vom Umfeld ist, müssen Patterns in dieser Domäne nicht nur technische, sondern auch soziale Aspekte berücksichtigen. Dazu gehören unter anderem soziale Unternehmensstrukturen, involvierte Personen und ihre Rollen im Unternehmen. Groupware Patterns sollten also anwendungsorientiert sein und dabei helfen, soziotechnische Systeme zu verstehen und zu designen (Herrmann et al., 2003; Eason, 1988). Laut Herrmann et al. bestehen Groupware Patterns aus statischen und dynamischen Aspekten. Statische Aspekte beschreiben die technische Struktur des Patterns und dynamische Aspekte repräsentieren die sozialen Prozesse, die involviert sein können. Da der dynamische Teil jedoch sehr variabel ist und schwer generalisiert werden kann, müssen und können Groupware Patterns nicht vollständig sein. Diese Unvollständigkeit wird bewusst beibehalten, damit menschliches Verhalten und menschliche Eigenheiten nicht als fix definiert angenommen werden. Diese Faktoren sind überaus instabil und müssen auch so behandelt werden (Herrmann et al., 2003).

Für eine ausreichend vollständige und anschauliche Beschreibung von Groupware Patterns propagieren Herrmann et al. Fotos, Screenshots und Kurzvideos, die typische Situationen zeigen, in denen das jeweilige Pattern zum Einsatz kommt. Das technische System soll also in einem sozialen Kontext gezeigt werden, um Eigenheiten und Herausforderungen besser vermitteln zu können. Da Patterns häufig auf elektronischem Wege, beispielsweise im Internet oder auf digitalen Datenträgern veröffentlicht werden, ist eine multimediale Darstellung die beste Form der Veranschaulichung. Zusätzlich sollte es ein Diagramm geben, das eine formale und schematische Beschreibung des Lösungsansatzes bietet. Gegenüber normalen Textbeschreibungen haben Diagramme den Vorteil, dass sie einprägsamer und leichter wiedererkennbar sind. Außerdem können sie auf verschiedene Arten gelesen werden und sind daher flexibler. Es müssen also Repräsentationen der Patterns geschaffen werden, die eine ausreichend detaillierte Beschreibung von Problem und Lösung bieten und gleichzeitig jedoch genügend Freiräume und Flexibilität beibehalten, um die menschlichen Faktoren nicht von vornherein zu stark einzuschränken. Bei der Verwendung von Groupware Patterns muss man sich jedoch im Klaren darüber sein, dass das Verhalten und die Eigenheiten der Benutzer von soziotechnischen Systemen niemals konkret vorausgesagt bzw. angenommen werden können und somit Patterns keine allumfassende, immer gültige Lösung, sondern nur einen Ansatz darstellen, der sich unter gewissen Bedingungen erfolgreich bewährt hat (Herrmann et al., 2003).

Ein Begriff, der in den Fachbereichen [CSCW](#) und [HCI](#) sehr häufig vorkommt ist »Awareness«. Er bezeichnet das Bewusstsein eines Benutzers über die momentanen Aktivitäten und Zustände der anderen Benutzer ([Dourish and Bellotti, 1992](#); [Hornecker et al., 2008](#)). Bei Groupware, die den Benutzern ermöglicht, gleichzeitig in der selben Arbeitsumgebung zu arbeiten, ist Awareness ein äußerst wichtiges Kriterium für eine hohe Usability. Die Zusammenarbeit der Personen kann nur dann ausreichend zufriedenstellend sein, wenn jeder immer über die anderen Bescheid weiß. Ohne diese Information wäre es gar nicht oder nur schwer möglich, die folgenden kollaborativen Aktivitäten durchzuführen:

- Koordinieren von Tätigkeiten
- Aufteilung der Benutzer in Teilgruppen zur Erledigung von Subtasks
- Diskussion über die Tätigkeit
- Die Aktionen anderer Benutzer vorhersehen und darauf reagieren
- Andere Benutzer bei ihren Aktivitäten unterstützen

([Gutwin and Greenberg, 1999](#))

Gutwin und Greenberg haben ein konzeptuelles Framework zur Aufrechterhaltung von Awareness in Groupware Systemen entwickelt. Der erste Teil des Frameworks teilt das Konzept der Awareness in mehrere Komponenten auf. Diese Komponenten sind in der ersten Spalte von [Tabelle 5.1](#) zu sehen und beantworten die Fragen: *Wer, Was, Wo, Wann* und *Wie?* Alle Teilnehmer sollten also wissen, mit wem sie arbeiten, was die anderen gerade machen, wo sie gerade arbeiten und welche Ereignisse wann eintreten. Diese Informationen sind bei jeglicher kollaborativen Arbeit äußerst wichtig und Designer von Groupware sollten immer versuchen, diese Informationen in ihren Systemen so gut wie möglich zum Benutzer zu transportieren. Erst wenn die Teilnehmer über diese Dinge Bescheid wissen, können sie effizient zusammenarbeiten.

Wenn sich die teilnehmenden Benutzer im selben Raum gegenüber sitzen, ist Awareness natürlich viel einfacher zu erreichen, als bei geographisch verteilten Systemen. Die Schwierigkeit, andere Benutzer und ihre Aktivitäten zu verfolgen, ergibt sich aus der eingeschränkten Wahrnehmung, die solche Systeme zwangsläufig mit sich bringen. Mit ihrem Framework versuchen Gutwin und Greenberg die fehlenden Informationen wiederherzustellen, aufgrund der knappen Platzverhältnisse auf Displays muss aber sehr genau evaluiert werden, welche Informationen wirklich

KATEGORIE	ELEMENT	FRAGESTELLUNG
Wer	Präsenz	Befindet sich jemand in der Arbeitsumgebung?
	Identität	Wer nimmt teil?
	Urheberschaft	Wer ist das?
		Wer macht das?
Was	Aktion	Was macht die Person?
	Intention	Welches Ziel verfolgt die Person?
	Artefakt	An welchem Objekt arbeitet die Person?
Wo	Ort	Wo arbeitet die Person?
	Blickrichtung	Wohin schaut die Person?
	Sichtfeld	Was sieht die Person?
	Reichweite	Was liegt in ihrer Reichweite?
Wann	Ereignishistorie	Wann ist das Ereignis eingetreten?
Wie	Aktivitätshistorie	Wie kam diese Aktivität zustande?
	Artefakthistorie	Wie kam das Artefakt in diesen Zustand?

Tabelle 5.1: Optimale Awareness garantiert, dass jeder Benutzer zu jedem Zeitpunkt alle diese Fragen beantworten kann.

genügend Relevanz besitzen, um den Platz dort einzunehmen (Gutwin and Greenberg, 1999).

Um ein besseres Verständnis für notwendige Designkriterien zu erlangen, haben Prante et al. drei verschiedene Groupware Applikationen für Ideenfindung im Team untersucht und Stärken und Schwächen ermittelt. Auf dieser Studie aufbauend, formulieren sie drei Anforderungen an Groupware dieser Art. Die folgenden Anforderungen an Groupware zur Ideenfindung haben sich also herauskristallisiert:

*Wenn das System stets nur einem Benutzer erlaubt, digitale Inhalte zu bearbeiten, wird die Effizienz drastisch gesenkt.*

- *Ermöglichung der gleichzeitigen Bearbeitung von Inhalten*  
Wenn das System stets nur einem Benutzer erlaubt, digitale Inhalte zu bearbeiten, wird die Effizienz drastisch gesenkt. Ähnlich wie bei Brainstormings, bei denen auch immer nur eine Person etwas zum Gesamten beitragen kann, entsteht bei diesen Technologien ein Flaschenhals. Daher ist die erste und wichtigste Anforderung an ein Groupware System die Vermeidung dieser äußerst kritischen Schwäche.
- *Möglichkeiten zur Strukturierung der Ideen*  
Die Studie hat gezeigt, dass die Effizienz des Prozesses der Ideenfindung deutlich gesteigert werden kann, wenn den Benutzern die Möglichkeit geboten wird, ihre Ideen und Konzepte auf der digitalen Arbeitsfläche zu strukturieren. Dadurch kann zuerst sehr frei und uneingeschränkt agiert werden. In einer zweiten Phase werden dann Struktur und Gliederung hinzugefügt.
- *Keine Einschränkung von Vorgehensweisen*  
Durch die Observation der Benutzer bei der Ideenfindung haben Prante et al. herausgefunden, dass es keine allgemeine Vorgehensweise gibt. Die Gruppen haben sehr unterschiedliche Formen der Interaktion und Kooperation gezeigt und daher auch andere Ansprüche an das System gestellt. Häufig erschienen die Aktionen chaotisch und frei von jeder Struktur. Im Laufe der Sitzungen wurden diese Aktionen dann klarer und Gliederungen hinzugefügt. Aufgrund dieser Tatsache, ist es äußerst wichtig für diese Art von Groupware, dass alle Formen des Vorgehens möglich und effizient durchführbar sind.

(Prante et al., 2002)

Auch Tang hat aus seiner Studie der Eigenheiten der Zusammenarbeit in kollaborativen Zeichenprogrammen Designkriterien für CSCW Software abgeleitet (Tang, 1989). Diese werden in [Tabelle 5.2](#) dargestellt.

Bei der Recherche, die der Entwicklung ihrer Roomware *iLand* (Streitz et al., 1998) voranging, entdeckten Streitz et al. einige

DESIGN CRITERIA	REASONS
1) Provide ways of conveying and supporting gestural communication. Gestures should be clearly visible, and should maintain their relation with objects within the work surface and with voice communication.	<ul style="list-style-type: none"> <li>• gestures are a prominent action</li> <li>• gestures are typically made in relation to objects on the work surface</li> <li>• gestures must be seen if they are to be useful</li> <li>• gestures are often accompanied by verbal explanation</li> </ul>
2) Minimize the overhead encountered when storing information.	<ul style="list-style-type: none"> <li>• only one person usually records information</li> <li>• other participants should not be blocked from continuing private or group work while information is being stored</li> </ul>
3) Convey the process of creating artifacts to express ideas.	<ul style="list-style-type: none"> <li>• the process of creation is in itself a gesture that communicates information</li> <li>• speech is closely synchronized with the creation process</li> <li>• artifacts in themselves are often meaningless</li> </ul>
4) Allow seamless intermixing of work surface actions and functions	<ul style="list-style-type: none"> <li>• a single action often combines aspects of listing, drawing and gesturing</li> <li>• writing and drawing alternates rapidly</li> <li>• actions often address several functions</li> </ul>
5) Enable all participants to share a common view of the work surface while providing simultaneous access and a sense of close proximity to it	<ul style="list-style-type: none"> <li>• people do not see the same things when orientation differs</li> <li>• simultaneous activity is prevalent</li> <li>• close proximity to the work surface encourages simultaneous activity</li> </ul>
6) Facilitate the participants natural abilities to coordinate their collaborations	<ul style="list-style-type: none"> <li>• people are skilled at coordinating communication</li> <li>• we do not understand the coordinating process well enough to mechanize it</li> </ul>

Tabelle 5.2: Tangs Designkriterien zur Erstellung von Multi-User Zeichenprogrammen.

wichtige Anforderungen und Kriterien für Design und Konzipierung von CSCW Systemen, sprich Groupware. Für ihre Studie untersuchten sie fünf Arbeitsgruppen mit insgesamt 80 Mitgliedern, indem sie mündliche Interviews führten, schriftliche Fragebögen ausfüllen ließen und selbst die Teamarbeitsräume inspizierten.

Kreative Teams stellen sich als optimalen Ort für Meetings einen möglichst großen, variabel gestaltbaren Raum vor, der den Charakter einer Landschaft oder eines Platzes hat. Das folgende Zitat aus der Studie untermauert dies:

»Meetings werden nicht mehr abgehalten, indem man sich in einem Raum trifft, sondern indem man die Umgebung und die Situation dafür schafft.«

(Streitz et al., 1998)

Trotzdem hält sich auch der Anspruch nach höchster Flexibilität und Mobilität. Das bedeutet, dass die Teilnehmer der kreativen Sitzungen gerne die Möglichkeit hätten, sich ins Freie zu bewegen und dort ihre Sitzung uneingeschränkt oder sogar unter besseren Bedingungen fortzuführen.

Die befragten Personen wünschen sich von solch allumfassender Groupware wie *iLand* einen sehr einfachen und direkten Zugang zu jeglicher Art von Informationen. Seien dies interne Unternehmensdaten oder externe Informationen von kommerziellen Datenbankanbietern oder Kunden. Diese Daten sollten barrierefrei zugänglich sein und möglichst in multimedialer Form aufbereitet werden. Besonders wichtig wären sogenannte Ideenpools, in denen nach Themen geordnete Materialien vorhanden sein sollten.

Für ein sinnvolles und effizientes kreatives Arbeiten in der Gruppe ist es von äußerster Wichtigkeit, Flexibilität und Individualität zu bewahren. Jedes Team entwickelt eigene Kreativitätstechniken und CSCW Systeme müssen sich entsprechend adaptieren können. Zur Zeit der Untersuchung, im Jahre 1998, war ein Großteil dieser Kreativitätstechniken papierbasiert. Dadurch ergaben sich Problemfelder wie beispielsweise Unlesbarkeit, Unordentlichkeit, mangelnder Platz und die eingeschränkte Flexibilität in Hinblick auf Löschen, Ändern und Umorganisieren von Informationen. Hier besteht große Hoffnung in computerbasierte Techniken, die diese Einschränkungen aufheben sollen.

Ein weiteres wichtiges Kriterium ist die Möglichkeit der adäquaten Präsentation von Ideen und Ergebnissen der kreativen Arbeit. Kreative Teams legten hier einen großen Wert auf flexible Alternativen zur herkömmlichen Frontalpräsentation. Sie wünschten sich einen partizipativen Präsentationsstil, bei dem die Zuhörer direkt eingebunden werden. So sollten Inhalte besser vermittelt und leichter verstanden werden.

»Wir haben das  
kreative Potential,  
nicht unsere  
Rechner!«

Die Befragten nannten die Visualisierung von Daten und Informationen als wichtigen Bestandteil von kreativen Sitzungen. Sie waren sich einig, dass eine gute und einfache Visualisierung von Konzepten und Ideen inspirierend wirken würde. Einerseits verstehe man sie dadurch besser, andererseits wären sie die Quelle neuer Ideen. Eine gute Visualisierung bewerteten sie als elementaren Bestandteil der Kommunikation von Konzepten innerhalb der Gruppe.

Kreative Teams leisten aber nicht nur kreative Arbeit, sondern müssen genauso Planung und Verwaltung von Projektarbeit durchführen. Für diese Art der Tätigkeit wünschen sie sich Groupware, die diesen zusätzlichen Aufwand möglichst minimiert, beispielsweise durch computerbasierte Unterstützung für Projekt-, Zeit- und Agenda-Management (Streitz et al., 1998).

### 5.3 PROBLEMFELDER

In seiner Studie über die Schwierigkeit des Designs und der Evaluierung von Groupware, nennt Jonathan Grudin drei grundlegende Problemfelder:

- Das Missverhältnis zwischen denen, die einen zusätzlichen Aufwand betreiben müssen und jenen die einen tatsächlichen Nutzen haben
- Das Treffen von Entscheidungen durch leitende Personen, basierend auf ihrer Intuition
- Das Unterschätzen der Schwierigkeiten, die mit einer Evaluierung von CSCW-Systemen verbunden sind

(Grudin, 1988)

Am Beispiel einer Software zur automatischen Planung von Sitzungen zeigt Grudin, wie ein ungleiches Verhältnis zwischen einzelnen Nutzern einer Groupware entstehen kann.

Wenn ein Mitarbeiter eine Sitzung mit anderen vereinbaren möchte, so muss er der Software nur die gewünschten Teilnehmer mitteilen. Das Programm vergleicht dann eigenständig die elektronischen Kalender dieser Personen und bucht einen Termin zu jener Zeit, in der noch alle Teilnehmer frei sind. Voraussetzung hierfür ist, dass jeder Mitarbeiter im Unternehmen einen elektronischen Kalender führt. Jene Personen, die in leitenden Positionen tätig sind, führen sehr wahrscheinlich einen Kalender oder haben eine Sekretärin, die diese Aufgabe übernimmt. Probleme ergeben sich dann, wenn leitende Mitarbeiter Termine mit Angestellten vereinbaren möchten und diese häufig keinen elektronischen Kalender führen. Das System glaubt daher, dass ein Termin zu jeder Zeit möglich wäre und bucht unter Umständen

*Groupware muss so konzipiert werden, dass jeder Benutzer einen Nutzen hat und zusätzliche Aufwände so gering wie möglich ausfallen.*

einen ungünstigen Zeitpunkt für entsprechende Teilnehmer. Um dieser Problematik vorzubeugen, müssten alle Mitarbeiter, vom einfachen Angestellten bis hin zum Geschäftsführer, einen elektronischen Kalender führen. Angestellte hätten dadurch einen zusätzlichen Aufwand, jedoch wenig bis gar keinen Nutzen vom System. Dieser Umstand kann sehr leicht dazu führen, dass die Software in der Praxis keine Akzeptanz unter den Benutzern findet und dadurch scheitert.

Beim Design von multi-user Software verlassen sich laut Butler et al. Entscheidungsträger häufig auf ihre Intuition, die wiederum auf persönlichen Erfahrungen, zumeist im single-user Bereich, beruht (Butler et al., 1987). Es mag relativ einfach sein, ein Gespür dafür zu entwickeln, wie gut die »User Experience«<sup>3</sup> eines Textverarbeitungs- oder Tabellenkalkulationsprogramm ausfällt, jedoch wird Groupware von einem breiten Spektrum an Personen verwendet und alle diese Benutzer haben unterschiedliche Hintergründe und Berufe, ungleiches technisches Know-How und abweichende Zugänge zum Programm. Die Software wird sehr wahrscheinlich scheitern, wenn die Intuition der Entscheidungsträger die Komplexität ignoriert, die durch solch eine Gruppendynamik zustande kommt.

Die von Grudin angeführte Software zur automatischen Planung von Sitzungen bezeichnet er selbst als anfällig für ein solches Scheitern durch falsche Intuition von leitenden Mitarbeitern. Er begründet dies so, dass jene leitenden Mitarbeiter selbst mit großer Wahrscheinlichkeit bereits einen elektronischen Kalender führen und daher vordergründig nur den eigenen Nutzen erkennen. Mangelnde Empathie kann zu Folge haben, dass der zusätzliche Aufwand, der einfachen Angestellten entsteht, nicht berücksichtigt wird.

Die Evaluierung von Groupware erfordert eine spezielle Vorgangsweise, basierend auf den Methoden der Psychologie und Anthropologie. Laut Grudin fehlen diese Fähigkeiten zum Zeitpunkt der Untersuchung (1988) in den meisten Entwicklungs- und Forschungsteams, da qualifizierte Personen aus den Bereichen »Human Factors Engineering«<sup>4</sup> und kognitiver Psychologie noch selten vertreten sind. Zusätzlich ist der mit der Evaluierung verbundene zeitliche und finanzielle Aufwand erheblich höher,

<sup>3</sup> User Experience bezeichnet das Nutzungserlebnis, das eine Person empfindet, während sie ein Produkt, einen Service, eine Software, etc. benutzt, bzw. in Anspruch nimmt. Ziel des User Experience Designs ist es, dieses Anwendungserlebnis für den Benutzer möglichst angenehm und positiv zu gestalten und all seine Erwartungen zu erfüllen.

<sup>4</sup> Als Human Factors Engineering bezeichnet man jene Disziplin, die sich mit den Fähigkeiten und Grenzen des Menschen beschäftigt und diese Erkenntnisse auf das Design von Produkten, Prozessen, Systemen und Arbeitsplätzen anwendet. Dazu gehört das Überprüfen auf »Usability«, das Erstellen von Nutzerprofilen und die Entwicklung von Benutzerdokumentation und Trainingsprogrammen.

da immer ganze Gruppen an Probanden getestet werden müssen. (Grudin, 1988)

Markus und Connolly greifen Grudins Überlegungen auf und erweitern diese. Ihren Behauptungen zufolge, kann Groupware auch dann scheitern, wenn gar keine Asymmetrie herrscht zwischen jenen Personen, die von dem System profitieren und jenen, die nur zusätzlichen Aufwand haben. Weiters sehen sie auch Entscheidungsträger in Groupware Projekten, die eine gute Intuition für den kollektiven Nutzen haben und richtige Entscheidungen treffen, nicht als ausreichenden Schutz gegen das Scheitern der Groupware. Sie argumentieren, dass der Nutzen eines CSCW Systems für einen Benutzer zu stark vom Verhalten anderer Benutzer abhängt. (Markus and Connolly, 1990)

Zu den Vorteilen, die Groupware bieten kann, zählen Markus und Connolly finanziellen Nutzen, zeitliche Einsparungen oder Verminderung von Arbeitsaufwand, als auch weniger rationale Nutzen, wie die Freude an der Benutzung neuer Technologien. Als Nachteile nennen sie finanzielle Auslagen, zeitliche Verzögerungen und zusätzlichen Aufwand durch etwaige Fehler, sowie Verwirrung und Ärgernisse bei der Benutzung der Groupware.

Zwischen diesen Vor- und Nachteilen können zwei verschiedene Formen gegenseitiger Abhängigkeiten entstehen. Die eine Form ist die Abhängigkeit in der Nutzung des Systems (»Usage«). Diese entsteht, wenn die Fähigkeit zur Nutzung des Systems einer Person von der Nutzung des Systems einer anderen Person abhängt. Ein Beispiel dafür wäre ein User, der Daten aus einer Datenbank abfragen möchte. Er kann dies nur tun, wenn bereits andere Personen Daten in der Datenbank abgelegt haben. Auch ein elektronisches Meeting-Planungssystem kann beispielhaft genannt werden: Die Nutzbarkeit hängt für jeden von der Nutzung des Systems durch die anderen Personen ab.

Situationen in denen die Nutzung des Systems durch bestimmte Personen positive oder negative Folgen (»Payoff«) für andere Benutzer hat, nennen Markus und Connolly die zweite Form der gegenseitigen Abhängigkeit. Die Autoren nennen diese beiden Abhängigkeiten »Usage interdependency« und »Payoff interdependency«.

Ihre Argumentation untermauern sie anhand dreier Beispiele. Das erste handelt von einer Abteilung in einem Unternehmen, in der ein Computer für alle Mitarbeiter zur Verfügung steht. Auf diesem ist ein Textverarbeitungsprogramm installiert, das alle Angestellten für ihre Arbeit benötigen. Die Maschine ist daher vollkommen ausgelastet und zu Stoßzeiten ist es äußerst schwierig für Mitarbeiter, Zugang zum Programm zu bekommen.

Der Abteilungsleiter entscheidet daher, einen zweiten Computer anzuschaffen und teilt dies der Belegschaft mit. In dieser

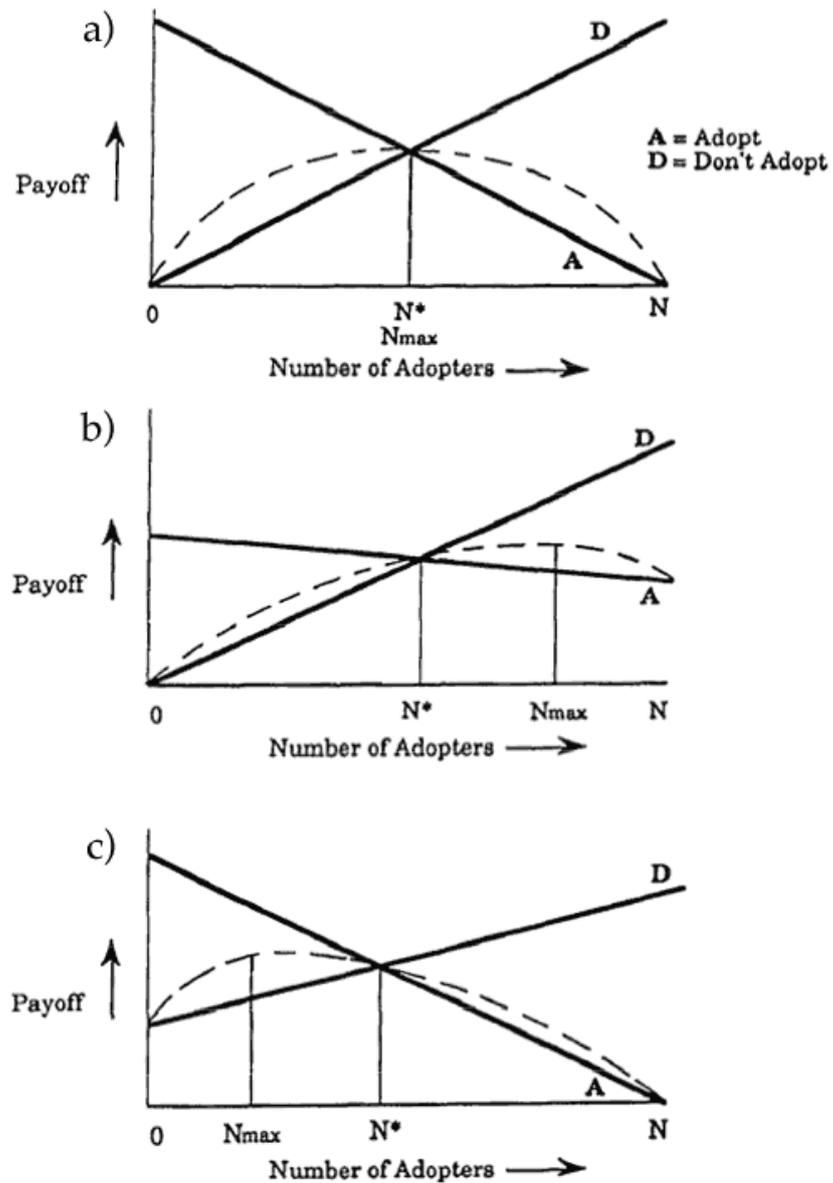


Abbildung 5.2: Die Grafiken illustrieren drei Szenarios der Entwicklung des tatsächlichen Nutzen («Payoff») für die Abteilung durch Einführung eines zweiten Computers mit einem Textverarbeitungsprogramm.

Situation ist es wahrscheinlich, dass das Verhalten der Mitarbeiter den maximalen Nutzen der zweiten Maschine verhindert. Jeder hat zwei Optionen offen und kann entweder auf den neuen Computer umsteigen ( $A$  für »Adopt«) oder beim alten bleiben ( $D$  für »Don't Adopt«). [Abbildung 5.2a](#) zeigt eine theoretische Verteilung des tatsächlichen Nutzen der Einführung des zweiten Computers. Auf der horizontalen Achse ist die Anzahl der Mitarbeiter dargestellt, die auf das neue Gerät wechseln.  $N$  bezeichnet die Gesamtanzahl aller Mitarbeiter. Die Linien  $A$  und  $D$  sind linear invers zueinander und stellen die Auslastung der beiden Geräte dar. Diese ergibt sich aus der Anzahl der jeweiligen Nutzer. Die vertikale Achse zeigt den Nutzen für die gesamte Abteilung. Theoretisch erreicht man einen maximalen Nutzen für die Abteilung, wenn beide Geräte gleich ausgelastet sind, in der Praxis jedoch ist dies oft nicht der Fall.

Es gibt viele Faktoren, die den Verlauf von  $A$  und  $D$  beeinflussen können, beispielsweise unterschiedliche Rechenleistungen und Performance der beiden Computer oder der Aufwand, der mit dem Erlernen eines neuen User Interface am neuen Rechner verbunden ist. [Abbildung 5.2b](#) und [Abbildung 5.2c](#) zeigen zwei weitere mögliche Szenarios, bei denen der maximale Nutzen dann erreicht wird, wenn der größere Teil der Belegschaft entweder beim alten Gerät bleibt oder zum neuen wechselt.

Dieses Beispiel zeigt, dass es notwendig ist, spezielle Situationen genau zu analysieren und entsprechend darauf zu reagieren. So läge es hier am Abteilungsleiter, die optimale Auslastung der beiden Geräte zu finden und zu forcieren. Die gegenseitige Abhängigkeit, die in solch einem Szenario entsteht, ist der Grund für diese unterschiedlichen Ergebnisse ([Markus and Connolly, 1990](#)).

Als zweites Beispiel behandeln Markus und Connolly ein System zur automatischen Planung von Meetings, genauso wie Jonathan Grudin in seiner Studie ([Grudin, 1988](#)). Dieses System wird für eine Gruppe von Personen in einem Unternehmen eingeführt, die untereinander alle ungefähr gleich vielen Meetings beiwohnen. Damit sei gesichert, dass theoretisch jeder gleich viel vom System profitieren kann.

In diesem Szenario sind »Adopter« jene Personen, die regelmäßig ihre Termine in das System eintragen und pflegen. Sobald Termine eingetragen werden, sind sie für alle zugänglich. Daher haben »Non-Adopter« einen Vorteil: sie haben keinen zusätzlichen Aufwand, da sie nichts eintragen, trotzdem verfügen sie über alle Informationen im System. Diese Situation kann auch als »multi-person prisoner's dilemma« bezeichnet werden ([Schelling, 1987](#)).

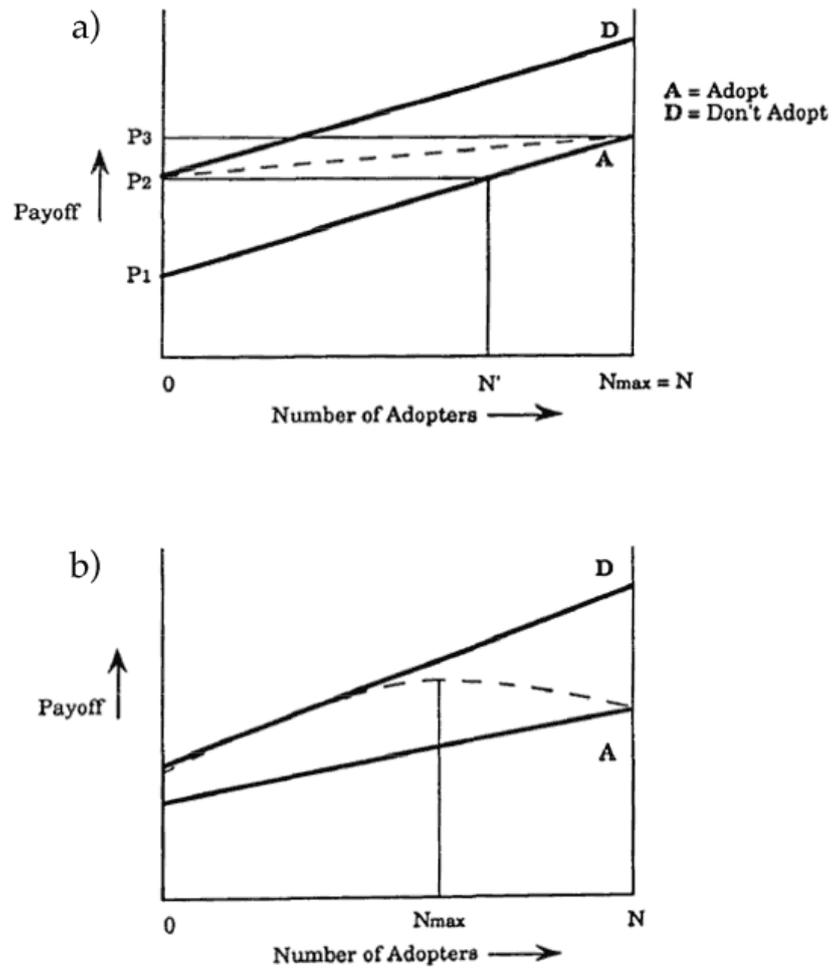


Abbildung 5.3: Die Grafiken illustrieren zwei Szenarios der Entwicklung des tatsächlichen Nutzen («Payoff») für die Abteilung durch Einführung eines automatischen Meeting Planers.

In [Abbildung 5.3a](#) wird der Nutzen von »Adoptern« und »Non-Adoptern« als zwei parallele Linien dargestellt. Die Differenz dieser beiden Werte bezeichnet den zusätzlichen Aufwand, den nur »Adopter« haben. Deswegen ist ihr Nutzen geringer. Man erkennt sofort, dass es irrelevant ist, wie viele Personen das System nutzen und wie viele nicht, denn diejenigen, die es nicht nutzen, werden immer einen Vorteil haben. Dies bedeutet aber auch, dass es für jeden das Beste wäre, das System nicht zu nutzen, weshalb es zu Schwierigkeiten bei der Einführung eines solchen Systems kommen könnte. In so einem Fall läge es wiederum am Abteilungsleiter, entsprechende Maßnahmen zu ergreifen, damit sich dieses ungleiche Verhältnis verbessert.

[Abbildung 5.3b](#) zeigt eine Situation, in der sich ausreichen Mitarbeiter zusammenschließen und gemeinsam in einer Koalition das System nutzen. Sie schaffen dadurch den selben kollektiven Nutzen für »Adopter«, wie wenn keiner das System verwenden würde. Dies setzt natürlich voraus, dass die Koalition über genügend Mitglieder verfügt, da die Rechnung sonst nicht aufgeht.

Im letzten Beispiel erläutern Markus und Connolly die Relevanz der kritischen Masse. Sie untersuchen dazu ein neu eingeführtes Kommunikationssystem, das es den Mitarbeitern erlaubt, intern mittels E-Mail zu kommunizieren. Vor Einführung dieses neuen Systems standen den Angestellten das Telefon, face-to-face Meetings und interne Post als Kommunikationsmittel zur Verfügung.

Es ist offensichtlich, dass der Nutzen dieses neuen Mediums für die ersten »Adopter« relativ gering ist. Genauso wie das Telefon macht das E-Mail System wenig Sinn, wenn es nicht von jedem genutzt wird. Gibt es aber genügend Personen die es nutzen, steigt der Nutzen für alle stark an, unter Umständen sogar so weit, dass er den Nutzen anderer Kommunikationskanäle übersteigt.

[Abbildung 5.4a](#) zeigt den Punkt der kritischen Masse dort, wo die »Payoff« Linien von »Adoptern« und »Non-Adoptern« sich schneiden. Der tatsächliche Nutzen für das Kollektiv wird durch die gestrichelte Linie dargestellt. Für das einzelne Individuum ist es vor Erreichen der kritischen Masse vorteilhafter, nicht auf das neue System umzusteigen, doch nach Erreichen der kritischen Masse ist das Umsteigen vorteilhafter. Der steigende Nutzen vor der kritischen Masse für »Non-Adopter« ergibt sich aus der Annahme, dass durch mehr »Adopter« weniger Personen die alten Kommunikationsmittel nutzen und diese dadurch entlastet werden. »Non-Adopter« finden also leichter freie Räume für Meetings und kommen auf telefonischem Wege leichter zu anderen Mitarbeitern durch, da deren Linien seltener besetzt sind. In dieser Situation erreicht der Nutzen für das Kollektiv seinen Höchststand, wenn alle Mitarbeiter das neue System verwenden.

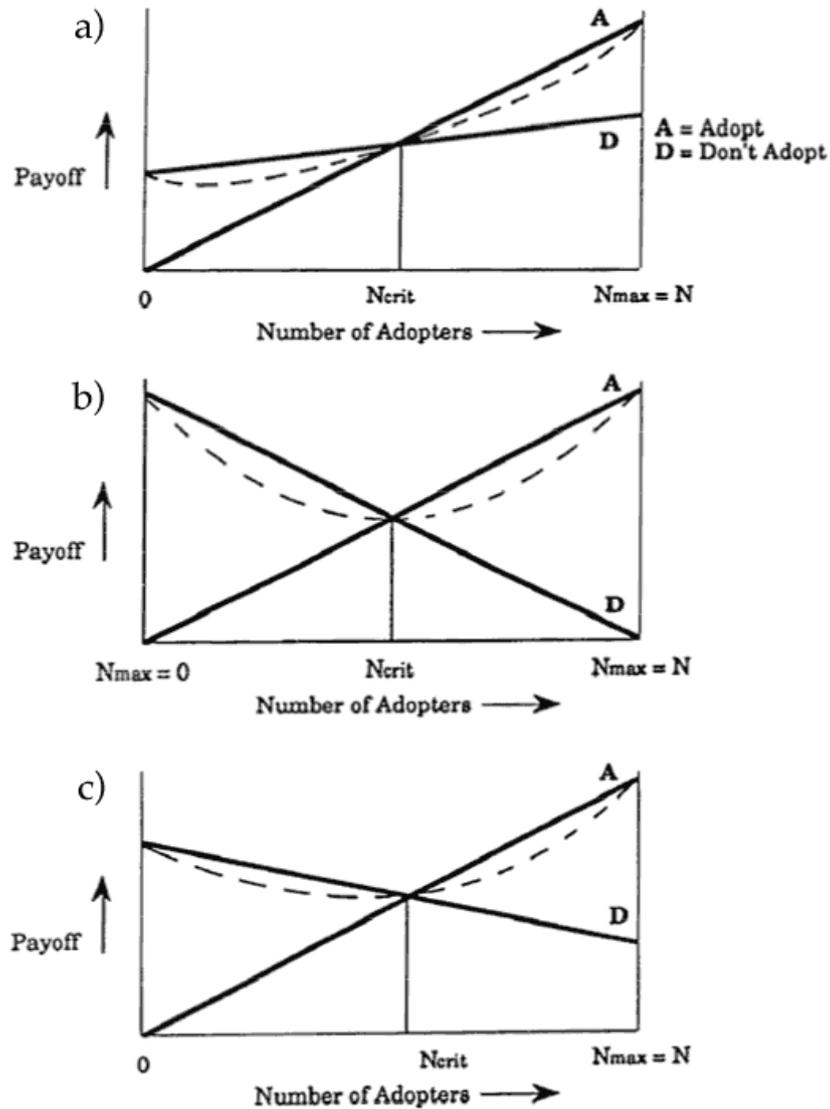


Abbildung 5.4: Die Grafiken illustrieren drei Szenarios der kritischen Masse unter verschiedenen Umständen.

Abbildung 5.4b stellt einen sinkenden Nutzen für »Non-Adopter« dar, während die Anzahl der »Adopter« steigt. Diese Situation könnte beispielsweise eintreten, wenn Personen wichtige Informationen nicht erhalten, nur weil sie das neue E-Mail System nicht nutzen. Höchst interessant ist in dieser Illustration die Tatsache, dass, wenn niemand das neue System verwendet, ein gleich hoher Nutzen für das Kollektiv entsteht, wie wenn alle es verwenden.

Abbildung 5.4c geht von einem noch pessimistischeren Szenario aus und zeigt den maximalen Nutzen wenn jeder Mitarbeiter auf das neue System umsteigt. Wichtig ist hier jedoch die Tatsache, dass die Belegschaft besser dran ist, wenn keiner umsteigt,

als wenn nur einige, sprich weniger Personen als für die kritische Masse notwendig, umsteigen. Das bedeutet, solange die Rate der Umsteiger sich nicht 100% nähert, ist es vorteilhafter, wenn keiner der Mitarbeiter umsteigt.

Aus diesen designierten Szenarios wird ersichtlich, dass die gegenseitigen Abhängigkeiten, denen Groupware unterliegt, eng verknüpft sind mit Nutzen und Schaden, die das jeweilige System anrichten kann. Diese Eigenheiten müssen beim Design von Groupware beachtet werden, damit der Erfolg selbiger nicht gefährdet ist. (Markus and Connolly, 1990)

#### ZUSAMMENFASSUNG

In der Wissenschaft gibt es verschiedene Definitionen und Auslegungen des Begriffes »Computer Supported Cooperative Work«. Die Wurzel dieser Uneinigkeit findet sich schon tiefer, bei der genauen Definition und Unterscheidung von »Collaborative Work«, »Collective Work« und »Cooperative Work«. Viele Wissenschaftler verwenden die Begriffe synonym, andere hingegen bestehen auf einer klaren Trennung der Termini. Für die Zwecke dieser Arbeit reicht jedoch eine synonyme Verwendung aus, denn letztendlich bezeichnet der Begriff CSCW ein multidisziplinäres Forschungsgebiet, das kooperative Zusammenarbeit innerhalb Gruppen untersucht und Technologien zur Förderung und Unterstützung selbiger entwickelt. Unter »Groupware« versteht man alle Systeme, egal ob Hard- oder Software, welche die in CSCW gewonnenen Erkenntnisse in die Praxis, zum Zwecke der Verbesserung von Teamwork, umsetzen. Design und Entwicklung solcher Systeme ist eine schwierige Aufgabe und wird von vielen verschiedenen Faktoren beeinflusst, die über Erfolg oder Misserfolg entscheiden. Es gibt Frameworks und Patterns, die versuchen formale Richtlinien zu geben, jedoch bildet jedes System und vor allem das Umfeld, in dem es eingesetzt wird, einen einzigartigen Kontext, der genau analysiert und berücksichtigt werden muss.



Teil II

PRAXIS



Die vergangenen Kapitel zeigen viele Beispiele, wie und mit welchen Hilfsmitteln Gruppenmeetings abgehalten werden können. Seien es traditionelle Medien wie Papier und Flip-Charts, oder elektronische Medien wie computerunterstützte Whiteboardsysteme - sie alle finden Einsatz in Designsessions und bieten verschiedenste Vor- und Nachteile.

Im folgenden wird *Scribbler* vorgestellt, ein elektronisches Interface, das die kollaborative Arbeit an virtuellen Artefakten unterstützt. Ähnlich wie die Autoren der bereits im ersten Kapitel beschriebenen Systeme, wurde mit *Scribbler* versucht, ein System zu kreieren, das die beiden Design-Arbeitswelten - geprägt durch traditionelle und elektronische Medien - mit einander verbindet und ihre Vorteile herausarbeitet. Im Gegensatz zu den bisherigen Ansätzen ist *Scribbler* ein elektronisches Skizziersystem, das ein barrierefreies Arbeiten ermöglicht.

Um dies zu zeigen, wird zuerst kurz die Ausgangssituation geschildert und danach die Anforderungen an ein kollaboratives Skizziersystem, die sich aus der Literaturrecherche ergeben haben, präsentiert. Danach wird die Vorgehensweise zur Erstellung des Prototyps erklärt, welche selbst einige Designmethoden aus [Kapitel 2](#) beinhaltet. Anschließend folgt eine Präsentation der simplen Oberfläche von *Scribbler* und dessen Funktionsumfang mitsamt technischer Hintergründe. Schlussendlich werden die Anforderungen und Merkmale gegenübergestellt und die aus User-Tests gewonnenen Erkenntnisse beschrieben.

## 6.1 AUSGANGSSITUATION & RAHMENBEDINGUNGEN

Die Technisierung vieler Berufe führte dazu, dass gewisse Arbeitsgegenstände ein elektronisches Abbild bekamen. Im Gegensatz zu früher, fertigen Grafikdesigner heutzutage Präsentationszeichnungen oft an einem digitalen Medium an. Neuere Berufssparten wie Interfacedesign setzen die tagtägliche Arbeit mit digitalen Artefakten voraus. Wie schon in den vorigen Kapiteln erwähnt, ist (im Besonderen) Design eine kollaborative Tätigkeit, wofür sich Gruppen in Meetings treffen, um gemeinsam neue Lösungen zu erarbeiten. Ein bewährtes und wichtiges Instrument dazu sind Stifte zur Erstellung von Skizzen, um Gedanken auszudrücken.

Ein wesentlicher Bestandteil moderner Meetingräume ist ein zentraler Bildschirm oder Projektor, der als Präsentationsmedium

eingesetzt wird. Zudem ist meistens eine Zeichenfläche vorhanden, wie z.B. Whiteboards oder Flip-Charts, um kollaborativ erarbeitete Ideen festzuhalten und neue Ideen zu explorieren.

Das Institut für Gestaltungs- und Wirkungsforschung an der TU Wien verfügt über so einen Meetingraum. Darin befinden sich mehrere Tische, die in der Mitte des Raumes zusammengestellt wurden und um die sich Sitzmöglichkeiten befinden. Neben einem Flip-Chart befindet sich ein 54" großer Bildschirm in der Mitte einer Wand, auf den alle Meetingteilnehmer von ihren Sitzplätzen aus blicken können. Mit dem Screen ist ein Computer (Apple Mac Mini) verbunden, auf den Daten über ein drahtlos Netzwerk gespielt werden können. Eine zusätzliche drahtlose Tastatur und Maus ermöglichen eine bequeme Steuerung. Zusätzliche Peripherie kann an der Rückseite des Computers angeschlossen werden.

Das Kerngebiet des Instituts ist **HCI**. Hier wird der Meetingraum oft dazu verwendet, Designs für interaktive Systeme, Applikationen oder Webinhalte zu erarbeiten. Erwartungsgemäß sind nur bereits umgesetzte Inhalte digital vorhanden. Diese Problematik macht den Einsatz der wichtigsten Designmethode - Skizzieren - auf dem Artefakt fast unmöglich. Das Ziel der vorliegenden Arbeit war es nun ein Tool zu entwickeln, das es ermöglicht, auf jegliche präsentierte Inhalte ob Webseiten, Programmoberflächen oder Bilder, »kritzeln« zu können. *Scribbler* soll so die Barriere zwischen der realen Welt und der digitalen Welt aufbrechen.

## 6.2 ANFORDERUNGEN AN KOLLABORATIVE SKIZZIERSYSTEME

In den vorhergehenden Kapiteln wurde ein Überblick über das Verbesserungspotential vieler existierender Systeme gegeben. Wichtige literarische Größen auf diesem Gebiet, wie beispielsweise Lee, Olsen, Tang, Larsson, Johnson oder Prante, wiesen auf wesentliche Erkenntnisse hin, die ein solches System formen. Diese Erkenntnisse können vier großen Einflussfaktoren zugeordnet werden: *Hardware*, *Software*, *Kollaboration* und *Skizzieren* und bieten eine Art Checkliste für kollaborative Systeme. [Abbildung 6.1](#) zeigt eine Aufstellung dieser Erkenntnisse, in Verbindung zu ihren Einflussfaktoren.

### *Hardware*

Ein ideales Setting für ein kollaboratives Skizziertool wäre laut Lee ein Grafiktablet in Verbindung mit einem Flatscreen. Dies wäre die bestmögliche Annäherung an eine Technologie, die

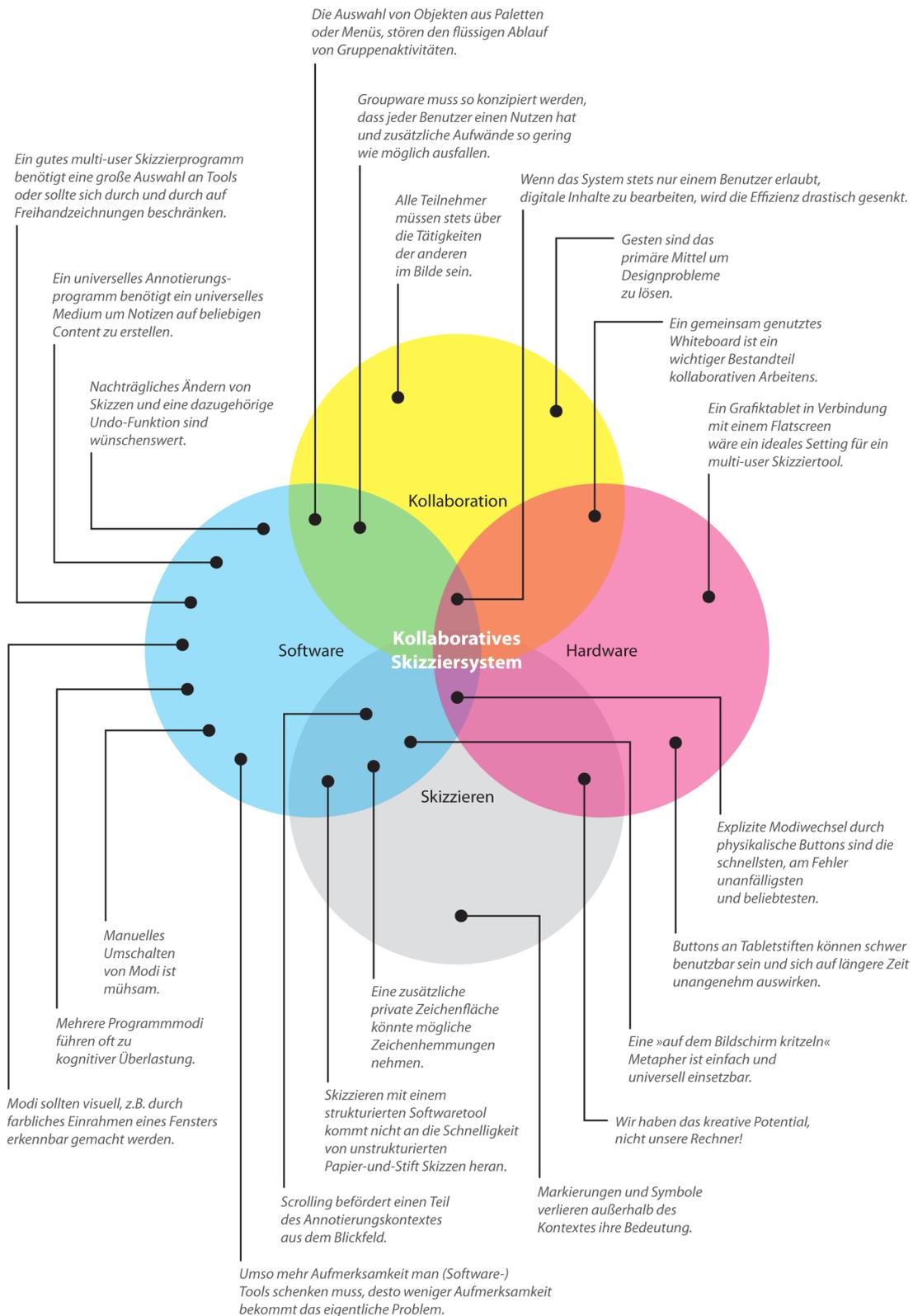


Abbildung 6.1: Anforderungen eines kollaborativen Skizzersystems, bestehend aus den vier Einflussfaktoren: Hardware, Software, Kollaboration & Skizzieren.

Stift und Papier ersetzen könnte. Sie würde ebenso zur Umsetzung eines Whiteboard Systems beitragen, was einen wichtigen Bestandteil kollaborativen Arbeitens ausmacht. Buttons an Tabletstiften können schwer benutzbar sein und sich auf längere Zeit unangenehm auswirken, deswegen sollte davon abgeraten werden diese mit Funktionen zu belegen.

### *Software*

Ein gutes multi-user Skizzierprogramm benötigt eine große Auswahl an Tools oder sollte sich ausschließlich auf Freihandzeichnungen beschränken. Welcher Ansatz besser funktioniert, müsse laut Lee ausprobiert werden. Je mehr Aufmerksamkeit ein Softwaretool erfordert, desto weniger Aufmerksamkeit kann jedoch dem eigentlichen Problem zukommen. Folglich sollte solch ein System so einfach wie möglich aufgebaut sein.

Die Verwendung mehrerer Programmmodi führt oftmals zu kognitiver Überlastung. Aus diesem Grund sollte der aktuelle Modus visuell, z.B. durch farbliches Einrahmen eines Fensters, erkennbar gemacht werden. Zudem ist das manuelle Umschalten von Modi mühsam. Explizite Modiwechsel durch physikalische Buttons sind dennoch die schnellsten, am Fehler unanfälligsten und beliebtesten.

Will man ein universelles Annotierungstool schaffen, das über alle verwendeten Programme eines Benutzers funktioniert, benötigt man auch ein universelles Medium, um Notizen auf beliebigem Content zu erstellen. Außerdem wären das nachträgliche Ändern von Skizzen und eine dazugehörige Undo-Funktion wünschenswert.

### *Kollaboration*

Gemeinsam genutzte Systeme müssen so konzipiert werden, dass jeder Teilnehmer einen Nutzen hat. Wenn das System nur einem Benutzer erlaubt digitale Inhalte zu bearbeiten, wird die Effizienz drastisch gesenkt. Zusätzliche Aufwände sollten so gering wie möglich ausfallen. Die Auswahl von Objekten aus Paletten oder Menüs, stören beispielsweise den flüssigen Ablauf von Gruppenaktivitäten. Die Teilnehmer der Kollaboration sollten ebenso stets über die Tätigkeiten der anderen im Bilde sein. Dabei spielen Gesten eine wichtige Rolle.

### *Skizzieren*

Das Zeichnen mit einem strukturierten Softwaretool ersetzt nicht das Zeichnen von unstrukturierten Papier-und-Stift Skizzen. Eine »auf dem Bildschirm kritzeln« Metapher ist somit einfach und universell einsetzbar. Skizzen verlieren außerhalb des Kontextes ihre Bedeutung. Daher muss darauf geachtet werden, dass der

Kontext stets bewahrt wird. Scrolling befördert z.B. einen Teil des Annotierungskontextes aus dem Blickfeld und benötigt somit zusätzliche Aufmerksamkeit bei der Umsetzung. Außerdem könnte eine zusätzliche, private Zeichenfläche mögliche Zeichenhemmungen von Teilnehmern nehmen. Schlussendlich haben die Benutzer das kreative Potential - ein Skizziersystem muss sich entsprechend adaptieren können.

### 6.3 VORGEHENSWEISE

Aufgrund des Bedarfs nach einem elektronischen Skizziersystem wurde zuerst eine Suche nach vergleichbaren Systemen veranlasst. Eine Brainstorming-Sitzung diente dazu, verschiedene Begriffe zu sammeln und zu explorieren. Viele wissenschaftliche Datenbanken stellen eine Fülle an Literatur bereit, die wiederum Hinweise zu bestehenden kollaborativen Skizziertools liefert. Nach der Lektüre und Analyse der gesammelten Werke konnte ein vorläufiges Anforderungsprofil an solche Systeme erstellt werden.

Eine wichtige Designfrage stellte sich schon zu diesem Zeitpunkt: Welche Eingabetechnologie sollte für *Scribbler* eingesetzt werden? [Abbildung 6.2](#) zeigt Aufzeichnungen zu der wohl wichtigsten hardwarespezifischen Frage. Nachträglich digitalisierte Skizzen, würden das System unabhängig von notwendigen Gerätschaften machen. Diese Voraussetzung käme der Flexibilität des Systems zu Gute. Jedoch ist es technisch aufwändig und fehleranfällig, gezeichnete Objekte nachträglich zu erkennen. Es müsste Mustererkennung zum Einsatz kommen, und das Ergebnis könnte unter Umständen nicht zufriedenstellend ausfallen. Würden Zeichnungen direkt auf digitalem Wege angefertigt, z.B. via Tablets, wäre die Flexibilität des Systems zwar eingeschränkt, das Ergebnis jedoch besser.

Im Hinblick auf *Scribbler* wurde zusammen mit dem Institut entschieden, auf die zweite Variante zu setzen, da Tablets bereits vorhanden waren und für einen ersten Prototypen ausreichen sollten.

Nachdem das grobe Konzept geklärt war, folgte der nächste Schritt, das Verstehen des eigentlichen Skizzierens. Es wurden dazu verschiedene Szenarios exploriert, um einen Einblick in die Praxis der Thematik zu erlangen. Anschließend wurden mehrere Mockups erstellt und das Skizzierverhalten von verschiedenen Probanden analysiert. Das Konzept konnte dadurch nach und nach verfeinert werden, doch aussagekräftigere Tests waren notwendig. Aus diesem Grund wurde bald mit der Anfertigung von Prototypen mit größerem Funktionsumfang begonnen. Die Mockups konnten auch nur Ein-Benutzer-Szenarios abdecken, weshalb schon bald mit der Implementierung begonnen wurde.

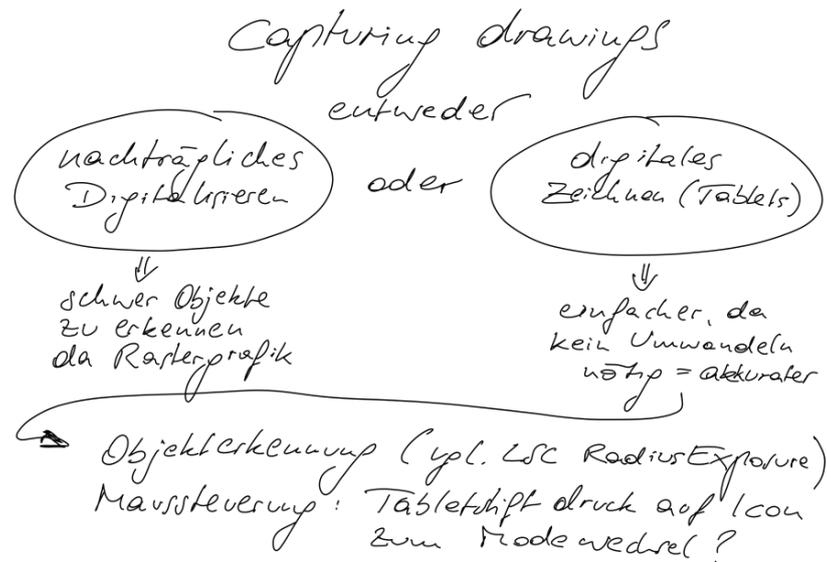


Abbildung 6.2: Aufzeichnung zur frühesten Designfrage. Sollten Skizzen nachträglich digitalisiert werden oder sollte gleich auf digitales Zeichnen gesetzt werden?

- ♣ *Anmerkung: Zu diesem Zeitpunkt kam erstmals der Begriff »Scribbler« auf, der fortan als Codename für das Projekt benutzt wurde.*

Während der Implementierungsphase konnten Funktionalität und Verhalten des Systems durch interne Tests kontinuierlich verbessert werden. Mit Hilfe von potentiellen Nutzern wurde die finale Beta-Version von *Scribbler* einem Review unterzogen.

In den folgenden Punkten werden die angesprochenen und verwendeten Designmethoden (vgl. [Abschnitt 2.2](#)) beschrieben und mit Hilfe verschiedener Bilder veranschaulicht. Die Beschreibung des Designprozesses soll so angereichert und durch konkrete, praktische Beispiele verständlicher aufbereitet werden.

### 6.3.1 Recherche

Recherche ist nicht direkt den Designmethoden zuzuordnen, stellt aber trotzdem einen wichtigen Teil in der Entwicklung eines Produkts dar. Recherche ermöglicht Einsicht in den bestehenden Markt und die aktuelle Forschung. Des weiteren wird ein Einblick in deren Funktionsweisen erlangt.

Durch die Literaturrecherche wurden einige vergleichbare Systeme gefunden, die jedoch die große Einschränkung hatten, dass Content und Input nicht miteinander verknüpft wurden. Es gibt zwar Annäherungen, jedoch bringen diese den Inhalt nur in eine statische Scheinbeziehung mit den Eingaben. Dies verdeutlichte die Notwendigkeit eines Skizziersystems wie *Scribbler*, das über die Grenzen des Instituts hinausgehen könnte.

Zusätzlich wies die Literatur bereits auf Probleme hin, die durch die notwendige Selektion eines Werkzeuges entstehen konnten (vgl. [4.2.4 Das Modusproblem](#)).

Das Erlangen neuer Sichtweisen des Systems durch die ausgiebige Recherche resultierte schließlich im vorläufigen Anforderungsprofil.

### 6.3.2 Skizzieren

Alleiniges bzw. gemeinsames Skizzieren waren die wichtigsten Instrumente in der Anfangsphase von *Scribbler*. Skizzen ermöglichten die Veranschaulichung bzw. Generierung neuer Ideen und dienten somit auch als Diskussionsbasis.

Aus der Literaturrecherche geht hervor, dass meistens ein Sitzungsteilnehmer als Schreiber fungiert und die wichtigsten, besprochenen Fakten in Stichwörtern und kurzen Sätzen zusammenfasst (vgl. [Abbildung 6.2](#)). Skizzen wurden manchmal angefertigt um bestimmte Vorgehensweisen aus der Literatur zu illustrieren. Dies führte meist zu kollaborativen Arbeitsvorgängen, bei denen zusammen an einer Skizze gearbeitet wurde.

Wie bereits erwähnt, wurden später Skizzen zu wichtigen Szenarios erstellt. Eine farbliche Kennzeichnung war dabei oft sehr hilfreich. Zudem konnten Zugehörigkeiten oder zeitliche Abläufe durch Hinzufügen von Nummerierungen leicht angedeutet werden (siehe [Abbildung 6.3](#)).

Abschließend zeigt [Abbildung 6.4](#) ein Beispiel einer detaillierten Skizze, die dazu verwendet wurde, technische Abläufe zu verstehen. Sie entstand im Laufe der Entwicklung, in der ständig neue Ideen aufkamen, die durch Skizzen innerhalb des Teams erforscht wurden.

### 6.3.3 Prototyping

Das Erarbeiten von Prototypen war keine leichte Aufgabe, da das Setting an sich, durch die verschiedenen Einflussfaktoren, kompliziert erschien. Dennoch konnten anfangs simple Prototypen, sogenannte Mockups angefertigt werden, um vorerst detaillierte Szenarios zu erstellen und später das Skizzierverhalten einiger Probanden zu beobachten. Mockups sind low-fidelity Prototypen, beispielsweise Papierprototypen. Dieser »Wegwerfcharakter« impliziert eine einfache Erstellung und rudimentäre Funktionalität, die für frühe Tests ausreicht.

[Abbildung 6.5](#) zeigt ein Mockup, das im Wesentlichen nur aus einem Screenshot eines WIMP-Systems bestand (in diesem Fall Mac OS X mit zwei offenen Programmfenstern) und in Verbindung mit einem Tablet PC benutzt wurde. Die konzipierten

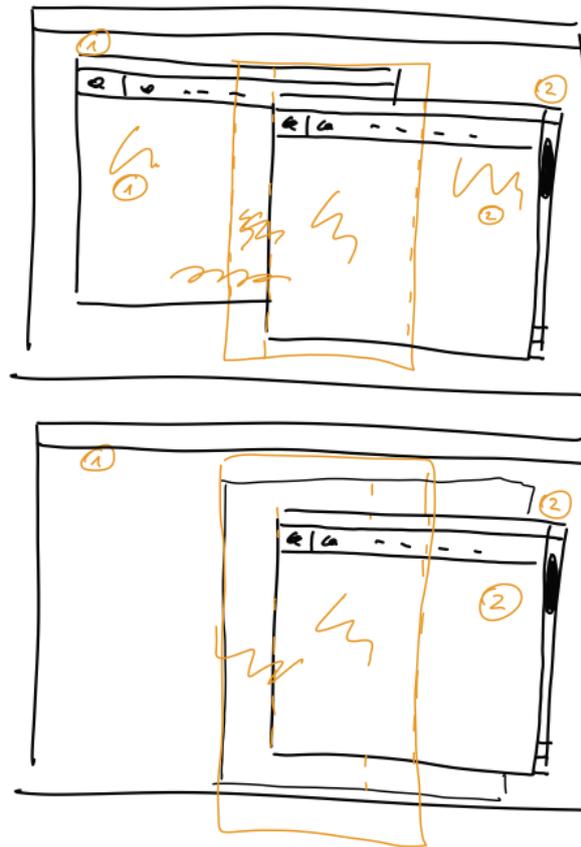


Abbildung 6.3: Skizzierte Szenarien in Scribblr. Zugehörigkeiten wurden durch eine Nummerierung gekennzeichnet.

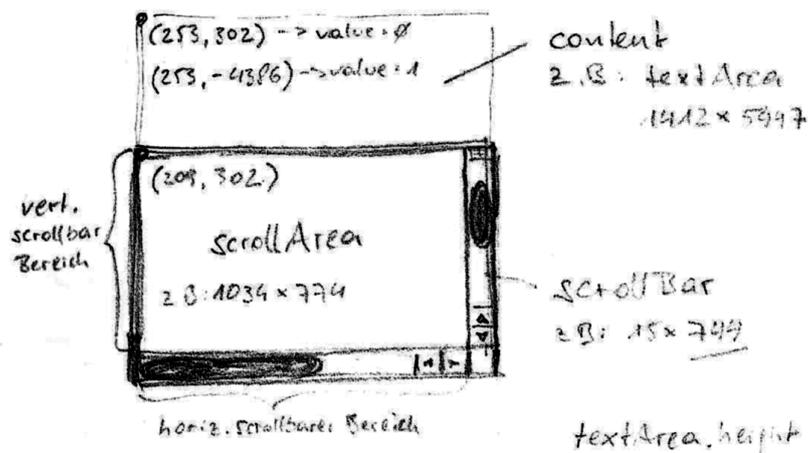


Abbildung 6.4: Detaillierte Skizze eines technischen Ablaufs, um diesen besser zu verstehen.

Szenarios, wurden durch den höheren Detailgrad konkreter, wodurch mehrere Funktionen im Voraus geplant werden konnten. Später wurden diese verwendet, um erste Tests durchzuführen. Der Prototyp beschränkte die Interaktionen der Benutzer auf das Skizzieren. So konnte beispielsweise festgestellt werden, welche Plätze zum Zeichnen gesucht werden, wie bestimmte Elemente markiert werden oder Zusammenhänge von Inhalten hergestellt werden.

Darauf folgende Prototypen als programmierte Versionen von *Scribbler* wurden ebenfalls getestet.

#### 6.3.4 (User) Testing

Die erstellten Prototypen vor der Implementierung des Programms, ermöglichten die Testung eines kleinen Funktionsteils mit Usern. Wichtige Systemverhaltensweisen bzw. Interaktionen konnten aber nur durch zusätzliche Testdurchgänge während der Umsetzung, auf Verständlichkeit und Anklang geprüft werden.

Einen Vorteil stellte die geringe Vorbereitungszeit dar. Es mussten keine bestimmten Anweisungen oder Aufgaben gestellt werden. Zum einen sollte die reale Welt in Bezug auf das Skizzieren mit Papier und Stift nachgebildet werden. Zum anderen sollten keine aufwändigen Aufgaben notwendig sein, sodass Probanden einfach »loskritzeln« konnten.

Es bestand außerdem die Möglichkeit, *Scribbler* innerhalb einer Gruppe von fünf Personen zu testen, die an einem bestimmten Design arbeiteten und dazu den in [Abschnitt 6.1](#) beschriebenen Meetingraum vom Institut nutzten (siehe [Abbildung 6.6](#)).

Nach Fertigstellung der Betaversion konnte *Scribbler* zusätzlich am Beginners-Day<sup>1</sup> der TU Wien vorgeführt und mit vielen erstsemestrigen Studenten getestet werden. Abschließend konnten auch ausführliche Tests mit Designern durchgeführt werden, die in [Abschnitt 6.5](#) ausführlich beschrieben sind.

---

<sup>1</sup> Der Beginners-Day ist eine Infoveranstaltung für alle Studienanfängerinnen und Studienanfänger in den Informatik-, Wirtschaftsinformatik- und Lehramtsstudien. (TU Wien, 2010)





Abbildung 6.6: Scribbler-Testing in einer Designsession. Fünf Personen überarbeiten mittels Scribbler ein Designvorschlag.

#### 6.4 SCRIBBLER AT A GLANCE

Durch die verschiedenen Designmethoden veränderte sich die Funktionalität von *Scribbler* öfters. Aus diesem Grund wird das Konzept erst in diesem Kapitel angeführt.

*Scribbler* ermöglicht neben der herkömmlichen Arbeit an einem Desktop (*WIMP*) System, mittels Stiften auf jedes beliebige Programmfenster zu zeichnen. Das System läuft im Hintergrund und aktiviert sich bei Bedarf. Es unterscheidet dabei generell zwei Hauptmodi: den Zeichen- und den Mausmodus. In letzterem kann die Maus wie gewohnt zur Navigation zwischen Programmfenster eingesetzt werden. Die Verwendung eines Tabletstifts hingegen wechselt in den Zeichenmodus. Die Benutzer können so auf das momentan aktive Fenster einer Drittapplikation »kritzeln«. Da besonders geübte Tabletbenutzer die Stifte auch als Ersatz für die Maus verwenden, ermöglicht *Scribbler* auch das explizite Wechseln der beiden Modi durch Drücken einer Tablettaste.

Da beim Skizzieren natürlich auch Fehler auftreten können, bietet *Scribbler* eine »Undo« bzw. »Redo« Funktion. Versehentliche Aktionen können so rückgängig gemacht oder wiederhergestellt werden. Zudem können einzelne Striche gelöscht bzw. »ausradiiert« oder der gesamte Zeichenbereich geleert werden.

Diese Grundfunktionalität soll vor allem Designer die Möglichkeit bieten, bestehende Arbeiten und Designs zu besprechen, Anmerkungen in direkter Verbindung zu digitalen Artefakten

zu verfassen und kontextbezogene Erweiterungen zu zeichnen. Weil natürlich nicht immer schon vorhandene Designs vorliegen, bietet *Scribbler* aber auch eine Whiteboard Funktion. Hierbei wird eine weiße Fläche über den gesamten Bildschirm eingeblendet, auf dem Benutzer Raum zum freien Skizzieren bekommen.

Um erarbeitete Skizzen zu speichern, besitzt *Scribbler* eine Screenshot Funktion. Nur durch Speichern eines Bildes, kann aus unserer Sicht der Kontext zu den Zeichnungen bewahrt werden. Eine Unterbrechung, jedoch nicht Beendigung von *Scribbler*, erfolgt durch eine integrierte Standby Funktion.

Die folgenden Punkte sollen nun nähere Informationen zur Hardware, dem grafischen User Interface und dem logischen Programmaufbau geben.

#### 6.4.1 Hardware

*Scribbler* benötigt ein relativ komplexes Hardware-Setting. Für die Ausgabe wird ein großes Display oder ein Beamer benutzt, während es für die Eingabe erforderlich ist, dass jeder Benutzer über ein Tablet mit Stift verfügt. Diese Tablets und das Ausgabegerät werden mit einem Rechner verbunden, auf dem das Apple Betriebssystem OS X 10.6 »Snow Leopard« läuft. Ältere Versionen von OS X werden von *Scribbler* nicht unterstützt. Für ein optimales Nutzungserlebnis sollte der Computer über eine gute Rechenleistung verfügen. Bei der Umsetzung des Prototypen fiel die Entscheidung auf die Nutzung von Tablets aus der Modellreihe Intuos3 von Wacom. Diese Geräte verfügen über jeweils vier Buttons an beiden Seiten neben der Zeichenoberfläche, die mit verschiedenen Funktionen belegt werden können. Über diese Tasten können die folgenden Aktionen durchgeführt werden:

- *Wechsel des Interaktionsmodus für den Tabletstift*  
Ermöglicht das Umschalten von Zeichen- in den Mausmodus und umgekehrt. (Dies betrifft nur den Tabletstift, die Maus selbst ist immer im Mausmodus und kann nicht zum Zeichnen verwendet werden)
- *Whiteboard*  
Blendet eine weiße Fläche über den gesamten Screen ein oder aus. Auf dieser kann ebenfalls gezeichnet werden.
- *Zurücksetzen*  
Löscht alle Zeichnungen des momentan aktiven Fensters.
- *Sichern*  
Erstellt einen Schnappschuss des Bildschirm samt den Zeichnungen des aktiven Fensters und legt diesen in einer Datei am Schreibtisch ab.



Abbildung 6.7: Die Anordnung der *Scribbler* Funktionen auf den Tasten des Tablets. Der Fotoapparat links oben erstellt ein Bildschirmfoto, das Verbotssymbol auf der rechten, oberen Taste löscht alle Zeichnungen des aktiven Fensters, die Maus auf der genoppten, mittleren Taste wechselt den Interaktionsmodus und das Whiteboard auf der unteren Taste blendet selbiges ein oder aus.

[Abbildung 6.7](#) zeigt die Anordnung der Buttons auf den Tablets. Die mittlere Taste ist mit einer Noppe versehen und bietet dadurch ein haptisches Feedback. Dadurch kann der Button ertastet werden ohne dass ein Hinsehen von Nöten ist. Sinnvollerweise sollte diese Taste mit der wichtigsten Funktion belegt werden. In *Scribbler* ist dies der Wechsel des Interaktionsmodus, denn dieser ermöglicht, den vollen Interaktionsumfang mit nur einem Eingabegerät (dem Stift) zu nutzen. Die anderen beiden wichtigen Funktionen, sprich das Bildschirmfoto und das Whiteboard, liegen auf den zwei großen Tasten links oben und unten. Das Zurücksetzen aller Zeichnungen wird relativ selten gebraucht und liegt daher auf der kleinen Taste rechts oben. Damit Benutzer diese Anordnung nicht auswendig lernen müssen, wurden bei den Tests entsprechende Icons an den Tasten angebracht. Auf der rechten Seite der Zeichenoberfläche des Tablets sind die selben Tasten spiegelverkehrt angebracht, damit auch Linkshänder *Scribbler* optimal verwenden können.

Neue Wacom Tablets werden üblicherweise mit nur einem Stift ausgeliefert. In den Testsettings wurden jedem Benutzer mehrere Stifte zur Verfügung gestellt. Diese verfügen alle über eine Hardware-ID und können dadurch von *Scribbler* eindeutig identifiziert werden. So ist es möglich, jedem Stift eine eigene Farbe zuzuordnen. [Abbildung 6.8](#) zeigt vier Stifte mit unterschiedlichen Farben. Um dies klar zu machen, wurde jeder Stift mit einem entsprechend gefärbten Sticker versehen. Der Wechsel zwischen Farben war dadurch so intuitiv wie das Wechseln zwischen zwei Buntstiften.

Jeder Stift hat einen Kippschalter an der Seite. Dieser erfüllt in *Scribbler* die Funktionen »Undo« und »Redo«, die die letzte Aktion rückgängig machen oder wiederherstellen. Stifte werden vom System erst dann wahrgenommen, wenn die Stiftpitze ein bis eineinhalb Zentimeter über das Tablet gehalten werden, wie [Abbildung 6.9](#) demonstriert. Dies ist auf die Hardware zurückzuführen und es kann kein Einfluss darauf genommen werden. Ebenso wird das Drücken des Kippschalters am Stift nur dann registriert, wenn der Stift sich in der entsprechenden Entfernung zum Tablet befindet. Am hinteren Ende eines Stiftes befindet sich ein weiterer Button, der üblicherweise eine Radierfunktion bietet, ähnlich einem mit Radiergummi versehenen Bleistift. Auch *Scribbler* belegt diese Taste mit einer Radierfunktion, jedoch können damit nur komplette Linien gelöscht werden.

#### 6.4.2 Grafisches User Interface

*Scribbler* ist ein Dienst, der im Hintergrund läuft und lediglich das Betriebssystem an Funktionalität anreichert. Daher hat es ein sehr subtiles GUI, das nahezu vollständig transparent bleibt



Abbildung 6.8: Jeder Stift ist eindeutig vom System identifizierbar und hat eine eigene Farbe zugeordnet. Zur Verdeutlichung, wurde jedem Stift ein entsprechend gefärbter Sticker aufgeklebt.

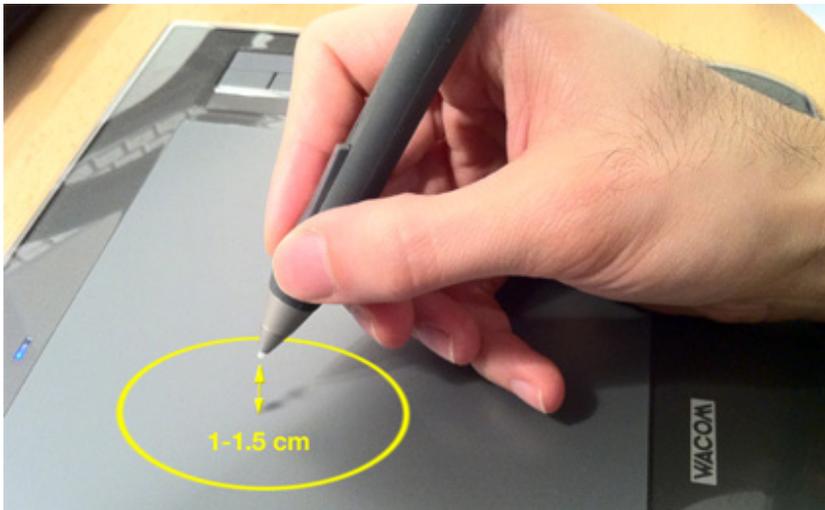


Abbildung 6.9: Die Hardware erkennt Stifte erst dann, wenn die Spitze ein bis eineinhalb Zentimeter über das Tablet gehalten wird. Das Betriebssystem empfängt dann ein »Proximity Event«, das von *Scribbler* weiterverarbeitet wird.

und Benutzer nicht von ihrer eigentlichen Arbeit ablenkt. Es besteht nur aus einem sternförmigen Icon in der Systemleiste des Betriebssystems, siehe [Abbildung 6.10a](#). Dahinter verbirgt sich ein Menü, dargestellt in [Abbildung 6.10c](#), das die Funktionen von *Scribbler* zur Verfügung stellt. Diese können fast alle auch über die Tasten am Tablett durchgeführt werden. Eine Ausnahme bilden der erste Menüeintrag, der *Scribbler* temporär in einen Standby-Zustand und somit inaktiv setzt, und der letzte Menüeintrag, der das Programm vollständig beendet.

Benutzer sollen das Programm gar nicht bemerken, müssen aber dennoch ein gewisses Maß an Feedback erhalten, um sinnvoll mit *Scribbler* arbeiten zu können. Dass der Zeichenmodus aktiv ist, wird Benutzern beispielsweise dadurch mitgeteilt, dass statt dem normalen Pfeil-Cursor (vgl. [Abbildung 6.12a](#)) ein Fadenkreuz-Cursor (vgl. [Abbildung 6.11](#)) angezeigt wird. Zusätzlich markiert *Scribbler* das aktive Fenster (einer Drittapplikation) mit einem blauen Rahmen. Den Benutzern wird so kommuniziert, welchem Fenster ihre Zeichnungen zugeordnet werden. Man kann sich also vergewissern, ob tatsächlich auf das gewünschte Fenster gezeichnet wird.

*Scribbler's* herausragendstes Feature, das es von allen bisherigen ähnlichen Systemen abhebt, ist der »Sticky Mode«. Diese Funktion bewirkt, dass Zeichnungen immer dort bleiben, wo sie hingehören. Wenn ein Benutzer auf ein Fenster zeichnet und dieses nachträglich verschiebt, erkennt *Scribbler* die Änderung der Fensterposition und verschiebt die Zeichnungen ebenfalls, sodass sie wieder genau dort über dem Fenster liegen, wo sie ursprünglich gezeichnet wurden.

Neben der Fensterposition kann *Scribbler* auch erkennen, wenn der Benutzer in einem Fenster die Scrollposition ändert. Zeichnet der Benutzer beispielsweise auf ein Element einer Webseite und scrollt dann nach unten oder oben auf der Seite, verschiebt *Scribbler* wiederum automatisch die Zeichnungen, sodass sie immer an der selben Stelle, relativ zum Inhalt bleiben.

Benutzer erhalten so direktes Feedback vom GUI, da die Zeichnungen in Echtzeit neu positioniert werden und dadurch scheinbar am Fenster haften. Außerdem färbt sich das Programm-Icon in der Systemleiste schwarz (siehe [Abbildung 6.10b](#)), sobald diese innovativen (nahezu magischen) Funktionen zum Einsatz kommen; ein kleines Detail im Sinne einer besseren User Experience.

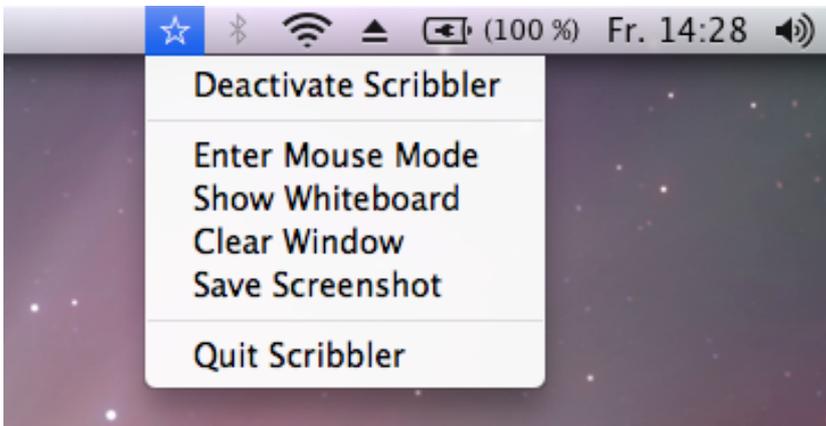
### 6.4.3 Programmlogik

*Scribbler* wurde auf das von Apple entwickelte *Cocoa*-Framework aufgebaut und genießt somit alle Vorteile einer objektorientierten Programmiersprache. Zusätzlich bietet *Mac OS X* eine Schnitt-



(a) Scribbler Icon

(b) When magic happens



(c) Scribbler Menu

Abbildung 6.10: Scribbler Menu Icon

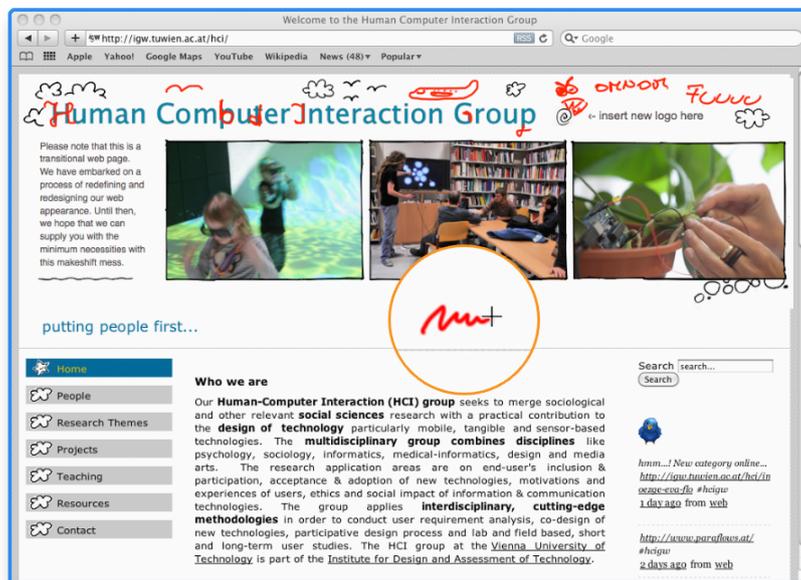


Abbildung 6.11: Skizzieren in Scribbler

Listing 6.1: Global and Local Event Monitoring

```

[NSEvent addGlobalMonitorForEventsMatchingMask: (
    NSLeftMouseDownMask | NSKeyDownMask | NSKeyUpMask |
    NSTabletProximityMask | NSMouseEnteredMask |
    NSLeftMouseDownMask | NSOtherMouseDownMask |
    NSRightMouseDownMask | NSOtherMouseDownMask)
    handler:^(NSEvent *incomingEvent) {

    ...

}];

[NSEvent addLocalMonitorForEventsMatchingMask: (
    NSOtherMouseDownMask | NSRightMouseDownMask |
    NSKeyDownMask | NSKeyUpMask | NSTabletProximityMask)
    handler:^(NSEvent *incomingEvent) {

    ...

}];

```

stelle für Bedienungshilfen, sog. Accessibility Daten, die einen wichtigen Teil des Systems ausmachen. Im folgenden Kapitel wird näher auf die Grundfunktionalität von *Scribbler* eingegangen.

*Warum weiß Scribbler, wann ein Benutzer zeichnen will?*

*Scribbler* fängt sogenannte globale und lokale Eingabe-Events ab. Events sind vom Benutzer ausgelöste Ereignisse, die durch Interaktion mit Eingabegeräten hervorgerufen werden. So ist ein Druck auf die linke Maustaste beispielsweise ein *LeftMouseDownEvent*, oder das Halten einer Taste auf der Tastatur ein *KeyDownEvent*. Geschehen die Aktionen im eigenen Programm, spricht man von lokalen Events. Ereignen sich die Aktionen in anderen Programmen, spricht man von globalen Events. Auf die selbe Weise erkennt das System auch *TabletProximityEvents*. Wie bereits im Punkt 6.4.1 *Hardware* erwähnt, entsteht ein *TabletProximityEvent* wenn ein Stift in die Nähe - mit 1-1.5 cm Abstand - zu einem Tablet kommt (vgl. *Abbildung 6.9*). *Scribbler* wechselt somit automatisch in den Zeichenmodus.

♣ *Anmerkung: Voraussetzung dafür ist, dass zuvor nicht explizit in den Mausmodus gewechselt wurde.*

*Listing 6.1* zeigt die Aufrufe, die benötigt werden um globale und lokale Events zu empfangen.

*Wie erkennt Scribbler das aktive Programmfenster?*

Das »aktive« Fenster wird in *Mac OS X* auch *Key Window* genannt und bezeichnet das am weitesten vorne liegende Fenster, das gerade den Fokus hat. Leider bietet *Cocoa* oder auch andere Frameworks keine direkte Abfrage nach dem *Key Window*. Es gibt zwar eine sogenannte *WindowList*, in der Informationen über alle offenen Programmfenster abgelegt sind, jedoch ohne Kennzeichnung des aktiven Fensters. *Scribbler* hat daher einen eigenen Mechanismus, um das aktive Fenster zu finden.

Dazu greift es auf die Daten des aktiven Programms zu und sucht in der *WindowList* nach allen dazugehörigen Einträgen. Da die Einträge der *WindowList* stets nach der Reihenfolge ihres Erscheinens sortiert sind, nimmt *Scribbler* den Eintrag an erster Stelle und hat somit das aktive Fenster gefunden.

*Woher weiß Scribbler, wann und wohin sich ein Fenster verschoben hat?*

Immer wenn ein Benutzer ein Programmfenster verschiebt, muss er ein Eingabegerät benutzen. Eine Fensterverschiebung wird naturgemäß durch ein Mausevent, ein sogenanntes *LeftMouseDraggedEvent* ausgelöst. Da *Scribbler* diese Events empfängt, kann im Anschluss eines solchen Events überprüft werden, ob sich das entsprechende *Key Window* verschoben hat.

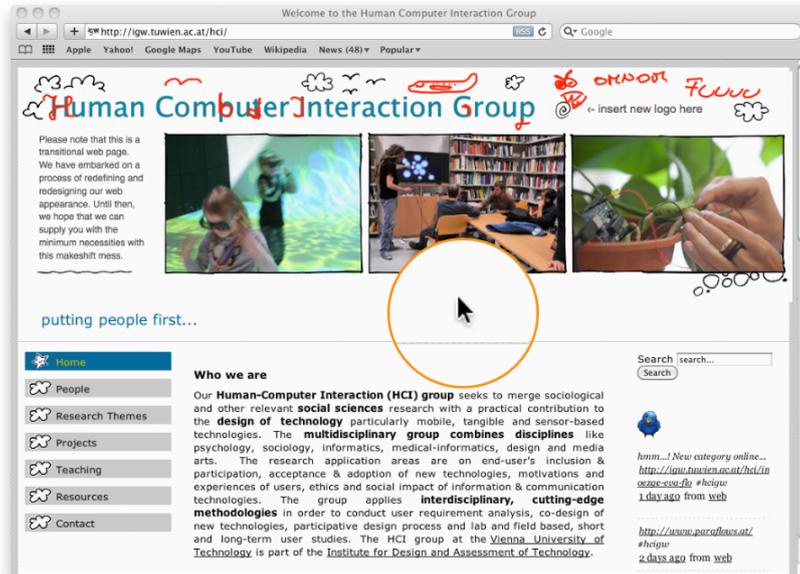
Dazu bedient sich *Scribbler* an den Informationen, die zu jedem Fenster in der *WindowList* gespeichert werden. Somit können allgemeine Informationen, wie z.B. der Programmname oder auch spezifische, wie die Fenstergröße und -position (zusammengefasst in sog. *Bounds*) gewonnen werden. [Listing 6.2](#) zeigt die Funktionen zur Gewinnung des Programmnamens und der *Windowbounds*.

*Wie können sich die Skizzen beim Scrolling mitbewegen?*

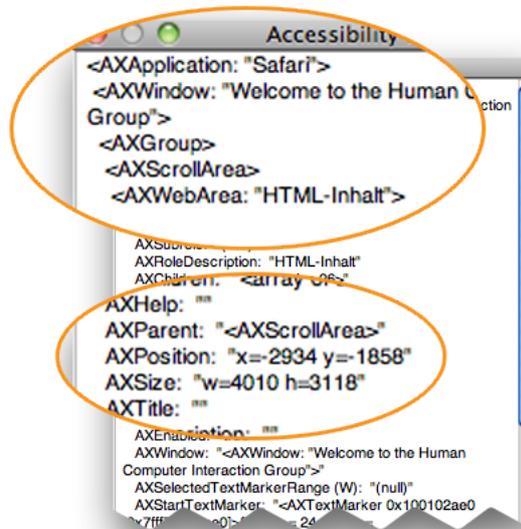
Im Prinzip ist Scrolling dasselbe wie eine Fensterverschiebung: Digitale Inhalte bewegen sich über den Bildschirm. Scrollingaktionen kann man zwar über die bewährten *Events* ansteuern, aber es ist weitaus komplizierter herauszufinden, welche Elemente sich wie weit bewegt haben. Aus Sicherheitsgründen darf ein Programm nicht auf Informationen anderer Programme zugreifen. Wie macht dies dann aber *Scribbler*?

Hier kommen die anfangs erwähnten Bedienungshilfen bzw. Accessibility Daten ins Spiel. Bedienungshilfen sind Funktionen, die für behinderte Menschen eingeführt worden sind und ausdrücklich in den Systemeinstellungen<sup>2</sup> aktiviert werden müssen.

<sup>2</sup> Erreichbar unter *Mac OS X*: Systemeinstellungen - Bedienungshilfen - »Zugriff für Hilfsgeräte aktivieren«.



(a) Mouse Position



(b) Accessibility Inspector

Abbildung 6.12: Accessibility Daten zur aktuellen Mausposition

Listing 6.2: Key Window Handling

```

// get keyWindow App Name
- (NSString*) getKeyWindowsApplicationName: (
    NSMutableDictionary*)windowInfos {

    return [windowInfos objectForKey:(id)kCGWindowOwnerName];
}
// get keyWindow Bounds
- (NSRect) getKeyWindowBounds: (NSMutableDictionary*)
    windowInfos {

    CGRect rect;
    CFDictionaryRef ref = (CFDictionaryRef)[windowInfos
        objectForKey:(id)kCGWindowBounds];
    CGRectMakeWithDictionaryRepresentation(ref, &rect);

    return *(NSRect *)&rect;
}

```

*Anmerkung: Scribbler benötigt den »Zugriff für Hilfsgeräte« um an Scrollinginformationen anderer Programme zu kommen.*



Unterstützt das aktive Programm Bedienungshilfen, können Accessibility Daten abgerufen werden. Je nach Programmherausgeber unterscheidet sich die Anzahl an verfügbaren Daten. Nicht jedes Programm, das Bedienungshilfen implementiert hat, stellt auch essentielle Daten zur Berechnung von Scrollinginformationen bereit. [Listing 6.3](#) zeigt allgemeine Informationen, die jedoch bei jeder Accessibilitykonfiguration vorhanden sind. So kann durch Bedienungshilfen beispielsweise, auf einfachem Weg das aktive Fenster ermittelt werden. Jedoch nur wenn das aktive Programm Accessibility Daten unterstützt. Welche Accessibility Daten von einem Programm unterstützt werden, kann mittels dem Dienstprogramm *Accessibility Inspector* herausgefunden werden, das alle verfügbaren Accessibility Daten zu der aktuellen Mausposition anzeigt (siehe [Abbildung 6.12](#)).

[Abbildung 6.12b](#) zeigt außerdem alle für Scrolling relevanten Einträge. Besitzt *AXParent* das Attribut *AXScrollArea*, existiert ein scrollbarer Bereich innerhalb des aktiven Fensters. Aus den *AXPosition* und *AXSize* Einträgen kann man mittels einfacher Mathematik die aktuelle Scrollingposition errechnen und somit die Bewegung innerhalb des Fensters.

## 6.5 USER REVIEW

Der Prototyp von *Scribbler* wurde mit verschiedenen potenziellen Nutzern getestet. Darunter drei Designer aus den Bereichen

Listing 6.3: Laden von Accessibility Daten

```

_systemWideElement = AXUIElementCreateSystemWide();

//Get the app that has the focus
AXUIElementCopyAttributeValue(_systemWideElement, (
    CFStringRef)kAXFocusedApplicationAttribute, (CTypeRef*)&
    _focusedApp);

//Get the window that has the focus
if(AXUIElementCopyAttributeValue((AXUIElementRef)_focusedApp,
    (CFStringRef)NSAccessibilityFocusedWindowAttribute, (
    CTypeRef*)&_focusedWindow) == kAXErrorSuccess) {
    ...
}

```

Produkt-, Interaction-, und User Experience Design, die sich an *Scribbler* in einem reduzierten Setting, also mit nur einem Tablet versuchten. Zusätzlich setzte eine Arbeitsgruppe am Institut für Gestaltungs- und Wirkungsforschung der technischen Universität Wien *Scribbler* in einem kollaborativen Setting bei einem Projekt-Meeting ein.

Bei allen drei Sitzungen wurden die Aktionen der Testpersonen durch eine Videokamera gefilmt und ihre Kommentare auf einem Tonträger festgehalten. Die drei Designer wurden zusätzlich interviewt, und die Transkripte sind im Appendix im Kapitel **GESPRÄCHE MIT DESIGNERN** zu finden.

Es folgt nun positives und kritisches Feedback, das aus diesen Tests hervorgegangen ist.

#### 6.5.1 Positives Feedback

*Scribbler* wurde von den Testpersonen durchwegs positiv aufgenommen und alle waren interessiert an Konzept und Funktionsweise. Die Arbeitsgruppe am Institut für Gestaltungs- und Wirkungsforschung der Technischen Universität Wien konnte den Prototypen produktiv bei ihrem Projekt-Meeting einsetzen und die Idee hinter *Scribbler* wurde als sinnvoll erachtet.

Peter, Produktdesigner und Dozent an der Universität für Angewandte Kunst Wien, bewertete das Konzept von *Scribbler* sehr gut.

Peter	Für mich ist das auf jeden Fall eine Traumsituation, da ich jetzt schon Notizen zu Präsentationen mache. Nur das Problem ist eben jetzt, dass ich die Notizen auf Zetteln mache und mir dazuschreiben muss, zu welcher Zeichnung die Notizen gehören, damit ich sie nachträglich wieder zuordnen kann.
-------	--

Auszug 6.1: Peter, Produktdesigner, über *Scribbler*.

Peter könnte sich vorstellen, das Tool im Unterricht im kollaborativen Setting einzusetzen und erklärt, in welchen spezifischen Situationen *Scribbler* nützlich wäre und in welchen nicht.

Peter	Ja das wäre super, wenn ich das im Unterricht einsetzen könnte. Es wäre spitze wenn jeder die Möglichkeit hätte, mitzuarbeiten. Wobei ich nicht ganz sicher bin ob Darstellungstechnik das passende Unterrichtsfach für <i>Scribbler</i> ist, weil da geht es konkret um das Zeichnen und dann reichen Grob-skizzen leider nicht aus, sondern man muss schon detaillierte Zeichnungen anfertigen. Anders sieht es da bei Designvisualisierungen aus, die auch in meinem Unterricht vorkommen. Da wäre es wirklich cool, die Studenten direkt einzubinden. Jeder könnte auch Stichworte dazuschreiben und ähnlich einem Brainstorming vernetzen. Das könnte ich mir gut vorstellen.
-------	--

Auszug 6.2: Peter über den Einsatz von *Scribbler* im Unterricht.

Jedem Stift eine eigene Farbe zuzuweisen gefällt Peter gut. Für seine Zwecke benötigt er mehr als nur eine Farbe und dies erscheint ihm eine gute Lösung. Er weist jedoch auch darauf hin, dass er gerne Kontextmenüs in Zeichenprogrammen verwendet und so schneller zwischen verschiedenen Farben wechseln kann.

Clemens	Du zeichnest derzeit mit der Farbe Magenta. Du hast vorher gemeint für Grobskizzen reicht dir eine Farbe oder?
Peter	Also wenn ich in der Strukturierungsphase bin, wäre es schon gut wenn ich mehrere Farben hätte.
Clemens	Ok. Dazu haben wir eigentlich mehrere Stifte mit unterschiedlichen Farben angedacht.
Peter	Wirklich wahr? <i>(Peter greift zu einem anderen Stift und probiert ihn aus)</i>
Peter	Wahnsinn. Ja, das finde ich super wenn man das so löst.
Thomas	Wir haben uns gedacht wir halten uns daran, möglichst realitätsnahe zu bleiben. Beim Zeichnen auf Papier würdest du auch einen anderen Stift zur Hand nehmen.
Peter	Verstehe. Was ich irrsinnig gerne verwende sind Untermenüs bzw. Pop-upmenüs. Das gibt es bei verschiedenen Programmen. Wenn ich z.B. auf die Stifftaste drücke geht ein Untermenü auf. So wie bei »Autodesk Maya« oder »Sketchbook«. Das sind Programme in denen ich z.B. immer zeichne. Und da hätte man im Untermenü auch die Möglichkeit auf eine Farbpalette, oder vielleicht auch 2-3 verschiedene Strichstärken. Dazu drück ich auf die Stifftaste, fahre mit dem Stift auf das Menü, lass wieder aus, das Menü ist wieder weg und ich habe die neue Einstellung. Beim Zeichnen ist das super; das ist etwas was mir z.B. in Photoshop abgeht.

Auszug 6.3: Jedem Stift wird eine eigene Farbe zugewiesen.

Das Whiteboard, das in *Scribbler* eingeblendet werden kann und dazu dient, auf eine weiße Fläche statt einem bestimmten Fenster zu zeichnen, wurde von allen Testpersonen positiv bewertet. Es gibt Situationen, in denen man schnell etwas aufkritzeln möchte, das nicht mit dem Fensterkontext in Verbindung steht, beispielsweise eine spontane Idee für ein Konzept oder ein Design. In diesen Momenten bietet *Scribbler* die passende Funktion, und die Idee kann so abseits vom aktuellen Geschehen festgehalten und später wieder abgerufen werden.

Durch *Scribbler* ist es möglich, auf mehrere nebeneinander angeordnete Fenster des Desktops zu zeichnen und dadurch semantische Verbindungen herzustellen. Dadurch können Inhalte auf eine ganz neue Art und Weise präsentiert werden. Derzeit ist es so, dass Zeichnungen sich immer an das aktive Fenster,

beispielsweise das Browserfenster heften und ihre Position relativ zur Fenster- und Scrollposition mitbewegen.

Thomas	Habe ich es richtig verstanden, dass du es gerne so hättest, dass wenn mehrere gleichzeitig zeichnen, jeder auf sein eigenes Fenster zeichnet und die jeweiligen Zeichnungen auf dem eigenen Fenster kleben bleiben? Momentan hängen alle Zeichnungen - egal von wem gezeichnet - nur auf dem aktivem Fenster. Sobald du das Fenster verschiebst, verschiebst du die Zeichnungen mit. <i>(Peter denkt nach)</i>
Peter	Weiß ich nicht. Also für mich funktioniert das derzeit von der Überlegung her recht gut. Aber ich müsste es erst länger ausprobieren, um auch die wirklichen Stärken zu finden.

Auszug 6.4: Zeichnungen heften sich an das aktive Fenster, auch wenn außerhalb dessen gezeichnet wird.

Als Produktdesigner zeichnet Peter sehr viel und er tut dies auch während Besprechungen. Das Zeichnen hilft ihm mit den anderen zu kommunizieren und bringt seine Kreativität in Schwung.

Peter	Was hier (in <i>Scribbler</i> ) auch gut funktioniert, ist das Verdeutlichen von Ideen. Ich bin jemand, der irrsinnig gerne zeichnet zum Reden. Ich könnte so z.B. ein paar Punkte rausholen und einen Teil, der hier im Bereich unten schwer zu erkennen ist, noch einmal rauszeichnen.
-------	--

Auszug 6.5: Zeichnen um besser zu kommunizieren.

Deshalb sieht er für *Scribbler* gute Einsatzmöglichkeiten bei Brainstormings und in Ideenfindungsphasen. Dadurch können Konzepte und Ideen schnell und effizient vermittelt und Inspiration gefördert werden. Ebenfalls denkbar wäre für ihn der Einsatz von *Scribbler* in Graphic-Recording-Meetings. Es handelt sich dabei um Sitzungen, bei denen eine Person jegliche verbale Kommunikation in der Gruppe als Skizzen festhält.

Peter	Das ist eine recht interessante Geschichte. Es handelt sich um Leute, die Meetings mit zeichnen. Das passiert analog auf einem Blatt Papier mit Stift. Nehmen wir an da sitzen mehrere Techniker und andere in ein Projekt verwickelte Personen, die Konzepte verbal besprechen und dann gibt es einen Zeichner, der, während die Leute ihre Ideen artikulieren, diese direkt zu Papier bringt und aufzeichnet. Darauf baut dann die Diskussion weiter auf und die Teilnehmer können gleich auf die Skizzen eingehen und sie weiter entwickeln oder verwerfen. Am Ende kann man anhand der Bilder nachvollziehen, worüber gesprochen worden ist. Das wäre sicherlich auch ein Gebiet, bei dem man <i>Scribbler</i> oder ähnliche Anwendungen zum Einsatz bringen könnte.
-------	--

Auszug 6.6: Ein Einsatz bei Graphic-Recordings wäre denkbar.

*Scribbler* könnte diese Art von Meetings unterstützen, indem die Skizzen und Zeichnungen, die die verbale Kommunikation der Sitzung abbilden, auf digitalem Wege erstellt werden. Dadurch wäre es leichter, Kopien anzufertigen und für jeden Teilnehmer bereitzustellen. Außerdem könnten sie in digitaler Form auch gleich per E-Mail versendet werden. *Scribbler* könnte außerdem das Meeting sinnvoll ergänzen. Beispielsweise, die Teilnehmer besprechen verschiedene Webseiten oder Produkte der Konkurrenz, so könnte der Zeichner diese Seiten öffnen und direkt darauf die Skizzen des Meetings festhalten. Auf diese Weise würde sich ein konkreter Kontext manifestieren. Folglich könnten mehr Informationen gesammelt werden.

### 6.5.2 *Kritisches Feedback*

Die Testdurchläufe mit den potentiellen Benutzern haben nicht nur positive, sondern auch kritische Aspekte von *Scribbler* enthüllt. Einige davon waren uns bereits vor dem Testen bewusst und wurden durch die Tests und Interviews bestätigt. Andere Kritikpunkte waren komplett neue Einsichten und haben uns auf wichtige Dinge aufmerksam gemacht. Die verschiedenen kritischen Anmerkungen werden im folgenden verschiedenen Kategorien zugeordnet, um ein besseres Verständnis zu bieten.

#### 6.5.2.1 *Unvollständigkeit des Prototypen*

Als *Scribbler* von der Arbeitsgruppe am Institut für Gestaltungs- und Wirkungsforschung in einem kollaborativen Setting eingesetzt wurde, wollten alle Teilnehmer sofort die Zeichenfunktion

ausprobieren. Gleichzeitig griffen alle zum Stift und bemerkten dann, dass der Prototyp jeweils nur einem und nicht mehreren Benutzern zu zeichnen erlaubt. Schon zu Beginn des Projektes wurde deutlich, dass dies eines der wichtigsten Features von *Scribbler* sein würde, aber es war nicht möglich, dieses in der kurzen Entwicklungszeit zu implementieren. Dieses Feature erfordert die Implementierung multipler Cursor und ist technisch aufwändig. Fehlt es jedoch, kommt es zu einem »Flaschenhalseffekt«, der die Effizienz von Meetings stark einschränkt. Personen können nicht parallel zeichnen, bzw. arbeiten und sind so ständig gezwungen, die Aktionen der anderen abzuwarten. Hinzu kommt ein erhöhter Aufwand der Absprache untereinander zur Koordination der Tätigkeiten. Der Test hat gezeigt, dass die Teilnehmer des Meetings sich ständig absprechen müssen, wer wann zeichnen kann. Diese zusätzliche Koordination ist in herkömmlichen Meetings überhaupt nicht notwendig und stellt einen klaren Nachteil hinsichtlich der Effizienz der Sitzung dar. Die Befürchtung, die bereits vor dem Test aufgekommen war, wurde sehr schnell bestätigt. Bei einer Weiterentwicklung von *Scribbler* muss diesem kritischen Feature höchste Priorität zugeordnet werden, denn es entscheidet über Erfolg oder Misserfolg des gesamten Systems.

Häufig kam es vor, dass Testpersonen unabsichtlich den Kipp- schalter am Stift drückten. Dies führte dazu, dass die letzte Aktion rückgängig gemacht wurde. Dieses Problem war uns schon während der Implementierungsphase bei internen Tests aufgefallen. Personen, die selten oder nie Tablet und Stift als Eingabegerät verwenden, passiert dieses unabsichtliche Drücken der Undo-Taste immer wieder. Der Produktdesigner, der täglich mit dem Tablet arbeitet, hatte hingegen keine Schwierigkeiten. Um *Scribbler* für eine breite Masse benutzbar zu machen, muss bei einer Weiterentwicklung darauf geachtet werden, dass den Tasten des Stifts keine Funktionen zugewiesen werden. Stattdessen sollten diese sinnvoll auf den Tasten des Tablets selbst angeordnet werden.

Es wurde sehr schnell deutlich, dass die Einsatzmöglichkeiten von *Scribbler* begrenzt sind. Die wenigen Zeichenfunktionen reichen lediglich für grobe Skizzen und Kritzeleien aus. *Scribbler* muss hier gezielt an Funktionalität angereichert werden und zumindest optional Features wie Strichstärke oder Druckempfindlichkeit anbieten.

Thomas	Sind diese primitiven Zeichenmöglichkeiten ausreichend, oder fehlt dir da was?
Peter	Nein, für Darstellungstechnik reichen diese Möglichkeiten bei weitem nicht aus. Man braucht da Dinge wie Strichstärke, gerade Linien, etc. Das sind Sachen, die Photoshop kann und die sind auch wirklich notwendig. Aber bei solchen groben Sachen, wie ich sie hier jetzt am Screen gezeichnet habe kann das schon reichen. Wichtig ist natürlich auch die Transparenz von Linien. Beim Skizzieren beginnt man ja mit ganz leichten Strichen, die das grobe Grundgerüst darstellen und zeichnet dann mit mehr Druckstärke drüber, sodass die Linien deutlicher werden und die Skizze konkreter wird.

Auszug 6.7: Die Einsatzmöglichkeiten sind begrenzt.

Der Prototyp war noch fehleranfällig und stürzte gelegentlich ab oder zeigte unerwartetes Verhalten. Dies wurde von den Teilnehmern zwar kritisiert, alle zeigten jedoch Verständnis für die unfertige Teilimplementierung. Selbstverständlich muss die finale Version von *Scribbler* stabil laufen und darf keine Daten verlieren.

#### 6.5.2.2 Technische Probleme

Eine große technische Hürde stellt für *Scribbler* das Speichern von Daten zur späteren Wiederverwendung dar. *Scribbler* hat natürlich keinen Einfluss auf andere Programme, hängt aber vollständig von deren Kontext ab. Das bedeutet, dass Zeichnungen in *Scribbler* nur dann sinnvoll sind, wenn die Fenster von Drittprogrammen darunter liegen. Die Problematik der Abspeicherung wird hier sehr schnell deutlich: *Scribbler* kann die Fenster und Anordnung von Drittprogrammen nicht speichern. Im Prototypen gibt es eine Funktion, die ein Bild des Desktops mit allen darauf liegenden Zeichnungen erstellt und in einer Datei am Schreibtisch ablegt. Dieses Rasterbild kann jedoch nicht weiterverwendet werden. Außerdem müssen bei längerem Arbeiten mit *Scribbler* sehr viele Bilder angefertigt werden, um alle Zeichnungen abzubilden. Dabei füllt sich der Desktop sehr schnell und zu einem späteren Zeitpunkt müssen Benutzer sich mit unzähligen Dateien plagen. Es muss hier dringend ein passendes Konzept gefunden werden, das die Bedürfnisse besser deckt als einfache Rasterbilder.

Das Schreiben ist eine weitere Schwäche von *Scribbler*. Die prototypische Implementierung ist technisch nicht optimiert und das System zeichnet nur wenige Bewegungspunkte des Stifts am Tablet auf. Dadurch entstehen ungenaue Linien und Handschrift wird deutlich unlesbar. Gerade wenn Benutzer versuchen, auf

kleine Flächen zu schreiben, scheitern sie in den meisten Fällen. *Scribbler* und der darunter liegende Code muss soweit optimiert werden, dass es möglich wird, auf Fenster in annehmbarer Größe zu schreiben, weil dies ein sehr übliches Szenario darstellt.

Die Entscheidung, Tablets ohne integriertes Display als Hardware für *Scribbler* zu benutzen, fiel aus der Annahme heraus, dass Personen in kollaborativen Settings weniger miteinander interagieren würden, wenn jeder auf seinen eigenen Bildschirm schaut. Ziel war der Einsatz eines einzelnen großen Displays, um eine größere Gemeinsamkeit zu schaffen und Teamwork zu fördern. Vielen Benutzern fällt jedoch die Abstraktion von Tablet hin zu einem entfernten Display schwer. Sie schaffen es nur sehr schwer, ihre Bewegungen mit dem Stift am Tablet so zu koordinieren, dass am Display auch tatsächlich die beabsichtigten Striche gezeichnet werden. Bereits das Einkreisen von gewissen Elementen am Bildschirm kann schwierig erscheinen und Frust beim Benutzer erzeugen. Diese Problematik verschärft sich, wenn das Display sich nicht frontal, sondern seitlich zum Benutzer befindet. Sitzt dieser zusätzlich vor dem Tablet, wird es fast unmöglich, *Scribbler* sinnvoll zu nutzen.

### 6.5.2.3 Konzeptionelle Probleme

*Scribbler* setzt den Einsatz von spezieller Hardware voraus. Benötigt wird ein großes Display oder ein Beamer, ein starker Rechner mit dem Betriebssystem OS X und mehrere Wacom Tablets. Diese Komponenten sind zum einen teuer und zum anderen keine üblichen Geräte, die die meisten Betriebe im Inventar führen. Aus diesen Gründen ist der Einsatz von *Scribbler* nicht ohne weiteres möglich, beispielsweise vor Ort beim Kunden. Dieser Aufwand steht in keinem Verhältnis zum effektiven Nutzen innerhalb eines Meetings steht.

Zed	[...] Beim kollaborativen Zusammenarbeiten mit dem Kunden glaub ich nicht dass es funktionieren würde. Der Kunde kann das nicht bedienen, allein der Umgang mit dem Tablet ist eher komplex. Hinzu kommt die notwendige Hardware, die ja auch erst ein mal angeschafft werden muss und transportiert werden muss. Man benötigt dann ja mehrere Tablets.
-----	---

Auszug 6.8: Notwendige Hardware schränkt ein.

Für die Interaction- und User Experience Designer scheint *Scribbler* keine wirklich brauchbaren Szenarios und Einsatzmöglichkeiten zu bieten.

Zed	Aber ich kann mir auch den praktischen Nutzen nicht so recht vorstellen. Zu sagen, ich würde das System wirklich irgendwo einsetzen... ich weiß nicht. Das Szenario fehlt mir. Beim Kunden fällt das nämlich komplett flach.
-----	--

Auszug 6.9: Der praktische Nutzen erschließt sich nicht jedem.

Gewisse zusätzliche Features und Optimierungen würden das *Scribbler* Konzept jedoch interessanter für sie erscheinen lassen.

Thomas	Was wäre notwendig, um es für dich nutzbar zu machen?
Zed	Dieser komischer Hovereffekt des Stifts am Tablet bereitet mir Schwierigkeiten. Das müsste man auf jeden Fall irgendwie lösen. Sodass man das Gefühl bekommt, dass man wirklich exakt arbeiten kann mit dem Ding. Das fehlt mir. Schön wäre natürlich auch, wenn man die Übersetzung von Tablet auf Screen lösen könnte, wobei das natürlich nur dann geht, wenn der Screen im Tablet integriert ist. Auch cool wäre, wenn man über das Internet zusammenarbeiten könnte, denn so ist man an dieses Setting im Raum gebunden, nicht mal über ein lokales Netzwerk hat man da irgendwelche Freiheiten.

Auszug 6.10: Notwendige Funktionalität.

Die Entscheidung, Linien in *Scribbler* als Vektorobjekte zu speichern, ging mit einigen negativen Konsequenzen einher. So ist zum Beispiel die Radiergummifunktion eingeschränkt. *Scribbler* kann immer nur eine ganze Linie komplett löschen. Folglich muss der Benutzer eine Linie vollständig löschen und neu zeichnen. Angenommen der Benutzer möchte wirklich nur eine Linie

kürzen, die etwas zu lang geworden ist, so muss er sie löschen und neu zeichnen.

Clemens	[...] Die Möglichkeit zu Radieren gibt es eigentlich auch - die hintere Taste am Stift dient dazu. Doch da das Programm vektorbasierend ist, radiert es nur den letzten Strich. Ist diese Funktion zu wenig für deinen Gebrauch?
Peter	Also wenn ich wirklich an einem Produkt arbeite, um eine Form herauszuarbeiten, wäre es zu wenig ja. Aber um einfache Sachen, wie z.B Notizen einzufügen oder Ideen zu formulieren, funktioniert es schon. Man müsste aber vielleicht anfangen das Programm häufiger zu verwenden, um ein gutes Feedback abgeben zu können.

Auszug 6.11: Der Radierer löscht nur ganze Linien.

#### 6.5.2.4 Wünsche der Benutzer

Offensichtliche Verbesserungen, wie Druckempfindlichkeit, Strichstärke und konkretere Zeichenmöglichkeiten wurden von fast allen Teilnehmern angesprochen. Zusätzlich gab es einige Einwände und Ideen zu Features, die das Nutzungserlebnis von *Scribbler* verbessern könnten. Essenziell für ein effizientes Arbeiten mit *Scribbler* erscheint die Möglichkeit, Zeichnungen nachträglich am Bildschirm noch verschieben zu können. So können erst Ideen generiert und danach Ordnung und Struktur geschaffen werden.

Thomas Ich habe gesehen, du würdest dir auch wünschen dass du einzelne Teile von Zeichnungen wo anders hinschieben könntest oder? Also dass du einen bestimmten Bereich ausschneiden und verschieben könntest.

Peter Ja, das wäre vielleicht speziell bei der Ideenfindung, wo alles durcheinander steht, oder auch bei Skizzen in einem Brainstorming technischer Natur interessant. Weil meistens ist der zweite Schritt dann der, dass ich anfangs die Ideen zu ordnen. Also es wäre gut, um den ersten Prozess der Ideenfindung nicht zu stoppen bzw. ihm eine Hürde zu geben, sondern gleich weiterarbeiten zu können. Und da ist es eben notwendig das Ganze in eine Ordnung zu bringen. Dann hätte ich wirklich eine Lösung wo alle gemeinsam arbeiten können.

Auszug 6.12: Es wäre gut, wenn man Zeichnungen verschieben könnte.

Pi und Zed hatten die Idee, zusätzlich ein Trackpad anstatt einer Maus zusammen mit dem Tablet als Eingabegeräte zu nutzen. Das Tablet wäre nur für Schreiben und Zeichnen zuständig, während das Trackpad zur Positionierung des Cursors genutzt werden könnte. Eventuell würde man so die Steuerung für Benutzer optimieren, die im Umgang mit Tablets ungeübt sind.

Pi Vielleicht ist die relative Projektion des Tablets dann sinnvoll, wenn man zwei Eingabegeräte verwendet. Dann setze ich mit der Maus den Zeiger dorthin, wo ich ihn möchte und kann sofort dort los zeichnen, egal wo am Tablet sich mein Stift befindet. [...] Oder man könnte auch überhaupt zwei Tablets haben, das eine zur Steuerung mit Gesten und das andere zum Zeichnen.

[...]

Zed Da würde sich doch das Magic Pad von Apple anbieten.

Auszug 6.13: Ein Trackpad statt einer Maus könnte die Eingabe erleichtern.

Von allen drei Designern wurde der Wunsch nach einer Online-Funktion geäußert. Sie alle haben schon öfter die Erfahrung gemacht, mit anderen Kollegen über das Internet zusammenzuarbeiten und es würde sich anbieten, *Scribbler* dafür zu nutzen. Für die lokale Benutzung würden sich die beiden auch wünschen,

dass nicht nur die Zeichnungen des aktiven Fensters sondern jene von inaktiven Fenstern im Hintergrund angezeigt werden. Um Chaos auf dem Bildschirm zu vermeiden könnte man beispielsweise Zeichnungen im Hintergrund in Grau und mit verschiedenen Alpha-Werten, entsprechend dem Z-Index des zugehörigen Fensters anzeigen.

Zed	Das heißt es hängt alles an einem Fenster dran... hmhm, kannst du mal bitte das hintere Fenster aktivieren und es so verschieben, dass man die Fenster und Zeichnungen dahinter sieht?
Thomas	Hmm, das geht so nicht. Es werden immer nur die Zeichnungen auf dem aktiven Fenster angezeigt.
Zed	Ah, hmm. Ich glaub das irritiert mich, dass die Zeichnungen verschwinden. Vielleicht sollte man die ausgegraut anzeigen, so wie die inaktiven Fenster selbst.
Clemens	Es besteht dann die Gefahr, dass ein Chaos entsteht, wenn zu viele Linien angezeigt werden.
Zed	Ja stimmt, aber man könnte doch mit verschiedenen Alpha-Werten arbeiten, je nach dem, wie weit hinten sich ein Fenster befindet.
Thomas	Oh, ja das klingt nach einer guten Idee.
Pi	Man könnte dadurch den Kontext wahren und sehen was man bereits gemacht hat.

Auszug 6.14: Es sollten nicht nur Zeichnungen des aktiven Fensters dargestellt werden.

Zudem kam bei ihnen der Wunsch nach einer besseren Darstellung der Zugehörigkeit von Zeichnungen zu Fenstern auf. Eine Idee wäre Farbe dafür einzusetzen, jedoch würde dies wiederum die Freiheit beim Zeichnen einschränken, da keine beliebigen Farben mehr gewählt werden könnten. Auch eine Darstellung, wer welche Zeichnungen gemacht hat, sollte es geben. Im Testsetting, das die Arbeitsgruppe an der Universität eingesetzt hat, wurde dies so gelöst, dass jedem Tablet eine Farbe zugewiesen wurde. Zu jedem gab es mehrere Stifte, die unterschiedliche Helligkeitswerte mit dem Farbwert des Tablets kombinierten. So konnte zumindest etwas Variation eingebracht werden. Im kollaborativen Setting wäre es ebenfalls wünschenswert, wenn mehrere Benutzer gleichzeitig auf verschiedene Fenster zeichnen könnten und die Zeichnungen dem jeweiligen darunter liegenden Fenster zugeordnet werden könnten. Dies stellt natürlich eine große technische Hürde dar und bei einer Weiterentwicklung von *Scribbler* muss die Machbarkeit erst exploriert werden.

Peter, der Produktdesigner, würde außerdem ein Kontextmenü benötigen, das ihm ein paar Optionen anbietet. Gängige Skizzierprogramme bieten diese Funktion an und ermöglichen so eine sehr hohe Effizienz bei der Arbeit.

## 6.6 ACHIEVEMENTS

Abschließend werden kurz die im Abschnitt [6.2 Anforderungen an kollaborative Skizziersysteme](#) beschriebenen Anforderungen an ein kollaboratives Skizziertool, den umgesetzten Merkmalen von *Scribbler* gegenübergestellt. Die Merkmale von *Scribbler* werden wiederum in die vier Einflussfaktoren *Hardware*, *Software*, *Kollaboration* und *Skizzieren* eingeteilt. [Abbildung 6.13](#) veranschaulicht diese durch eine farbliche Kennzeichnung der ursprünglichen Erkenntnisse aus der Literatur. In Grün gehaltene Erkenntnisse wurden in der Betaversion von *Scribbler* bereits behandelt, rot eingefärbte konnten (noch) nicht umgesetzt werden.

Wie die Abbildung erkennen lässt, konnte ein Großteil der Anforderungen umgesetzt werden, konkret 17 Punkte (77,27%). Die restlichen 5 Punkte (22,73%) blieben vorwiegend durch technische Probleme bis dato ungelöst.

### *Hardware*

Durch das Benutzen von Grafiktablets in Verbindung mit einem Flatscreen, wurde laut Lee ein solides Setting für das System geschaffen und die Weichen für eine Whiteboardfunktion gelegt. Die zwei Hauptmodi von *Scribbler*, der *Zeichenmodus* und der *Mausmodus*, können explizit durch Tabletbuttons gewechselt werden. Beobachtungen zeigten, dass dies den Arbeitsfluss beschleunigte.

Das derzeit schwerwiegendste Problem von *Scribbler* ist jedoch die Tatsache, dass das System nur einem Benutzer erlaubt, digitale Inhalte zu bearbeiten und nicht allen gleichzeitig. Die Userreviews zeigten ganz klar den Wunsch nach dieser Funktion. Nur so können Koordinationsschwierigkeiten vermindert werden.

Ebenso wurde deutlich, dass besonders Benutzer mit wenig Tableterfahrung, Probleme bei der Benutzung der Buttons an den Tabletstiften haben.

### *Software*

*Scribbler* sollte so simpel wie möglich aufgebaut sein und verzichtet auf einen großen Funktionsumfang. Daher beschränkt es sich auf einfache Freihandzeichnungen. So soll sichergestellt sein, dass die Aufmerksamkeit bei der Benutzung auf das Designproblem an sich und nicht auf das System selbst gerichtet wird.

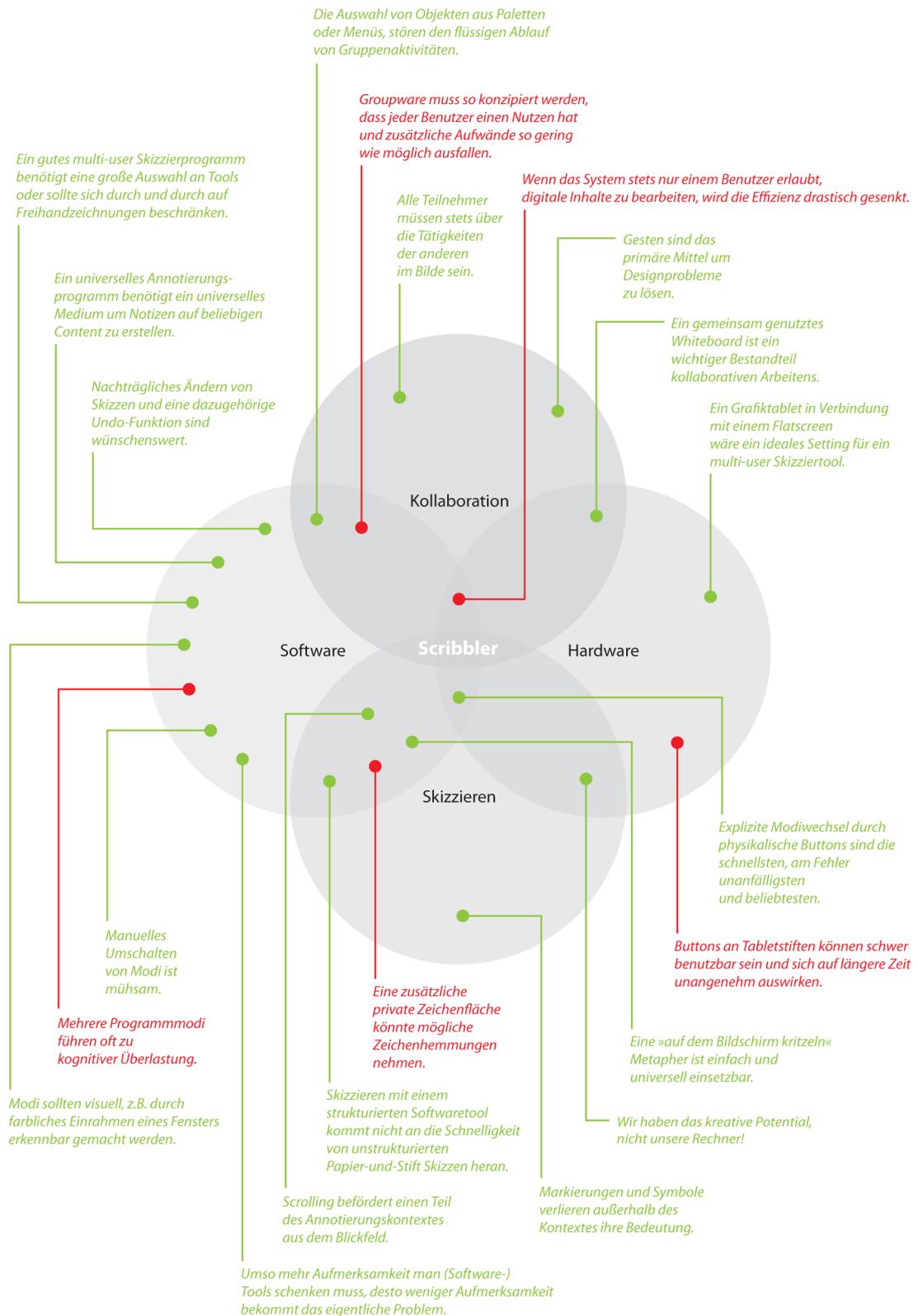


Abbildung 6.13: Anforderungsauswertung für Scribbler. Alle umgesetzten Anforderungen sind in Grün gehalten. (Noch) nicht umgesetzte Anforderungen sind rot markiert.

Durch die *Accessibility* Funktion von *Mac OS X* (vgl. [6.4.3 Programmlogik](#)) konnte *Scribbler* zu einem universellen Medium werden, das ermöglicht, auf beliebigen Content zu zeichnen. Es konnte ein Annotierungsprogramm entwickelt werden, das es in dieser Form noch nie zuvor gegeben hat.

Das Umschalten der Modi kann wie vorher beschrieben explizit durch Tabletbuttons geschehen, oder auch implizit durch das Wechseln zwischen der Maus und einem Tabletstift. Die implizite Methode verursachte keine Probleme. Für den expliziten Wechsel wurde die Kennzeichnung des Modus durch eine farbliche Fensterumrandung eingeführt. Da die Umrandung aber erst erschien, wenn ein Stift in der Nähe des Tablets war, kam es trotzdem zu Verwirrungen.

Schlussendlich bietet *Scribbler* eine Undo-Funktion die sich im Gegensatz zur Radierfunktion als äußerst hilfreich herausstellte.

### *Kollaboration*

Durch die hardware-spezifischen Einschränkungen war es unmöglich, die Bedürfnisse aller Benutzer abzudecken. Die Koordinationsaufwände stiegen dadurch ebenfalls. Aus diesem Grund war es aber stets klar, wer gerade auf einem digitalen Artefakt zeichnet. Folglich waren alle über die Tätigkeiten der anderen im Bilde.

Durch das Fehlen unnötiger Paletten oder Menüs konnte ebenfalls ein flüssigerer Ablauf von Aktivitäten gesichert werden. Da sich alle Mitglieder der jeweiligen Testsessions im selben Raum aufhielten und durchgehend Sichtkontakt behielten, waren Gesten fortwährend vorhanden und konnten so zur Lösung eines Problems beitragen.

### *Skizzieren*

*Scribbler* ist ein unstrukturiertes Skizzier- und Annotationstool. Durch die Verbindung mit den Grafiktablets kommt es durchaus an die Schnelligkeit von Papier-und-Stift Skizzen heran. Während der Laufzeit von *Scribbler* wird stets der Kontext zu den Skizzen bewahrt. Somit stellt Scrolling auch kein Problem dar.

Gerade die Tests am *Beginners' Day* zeigten, dass durchaus Hemmungen beim Erstellen von Zeichnungen auftreten. Private Zeichenflächen könnten dagegen helfen, fehlen derzeit aber noch im Prototypen.

## ZUSAMMENFASSUNG

Die umfangreiche wissenschaftliche Recherche enthüllte viele verschiedene Systeme, die als Inspiration und Wegweiser für die Entwicklung von *Scribbler* dienten. Durch die Erfahrungen, die Wissenschaftler in der Vergangenheit gemacht haben, konnte ein klares Anforderungsprofil an kollaborative Skizzier- und Annotationssoftware geschaffen werden. *Scribbler* wurde mit dieser Prämisse vor Augen entwickelt. Es handelt sich dabei um ein technisches System, das Menschen in kollaborativen Meetings beim Skizzieren und Annotieren unterstützen soll. Am sinnvollsten kann *Scribbler* von Designern, zur Optimierung des Designprozesses, oder von Lehrenden, zur besseren Veranschaulichung von Lehrinhalten, genutzt werden. *Scribbler* unterscheidet sich insofern von anderen Systemen, als dass es eine semantische Verknüpfung zwischen Drittapplikationen und ihnen zugehörigen Zeichnungen, bzw. Skizzen schafft. Das Gezeichnete haftet am darunter liegenden Programmfenster und sogar an darin befindlichen Inhalten, wie beispielsweise Webseiten. Daher kann damit ganz anders gearbeitet werden, als mit Systemen, die einfach ein statisches Bild über darunter liegende Fenster legen. *Scribbler* verfügt über einen innovativen Charakter und bietet Funktionen, die es zuvor in dieser Form noch nicht gegeben hat. Dies gebührt nicht zuletzt dem aktuellsten Mac Betriebssystem OS X 10.6, das zum ersten mal Entwicklern die dafür notwendigen Funktionen zur Verfügung stellt. Neben diesen positiven Eigenschaften gibt es auch einige Nachteile, beispielsweise das teure und umfangreiche Hardware-Setting oder das Fehlen umfangreicher Zeichenoptionen, das es für den Einsatz als komplexes Skizziertool unbrauchbar macht. Das Projekt *Scribbler* hat den aktuellen Stand von Technik und Wissenschaft genutzt und ein neuartiges System entwickelt, das im Gegenzug neue Eigenheiten und Aspekte hervorgebracht hat, die bei einer Weiterentwicklung in Betracht gezogen werden müssen.



*Never doubt that a small group of thoughtful, committed  
people can change the world.  
Indeed, it is the only thing that ever has.*

— Margaret Mead (Buxton, 2007)

## EPILOG

---

Wir versuchten mit dieser Arbeit ein Grundverständnis für Designmethoden und CSCW Systeme zu bilden, um die Arbeitsweisen von Designer in Bezug auf digitale Medien kennenzulernen. Durch das nähere Betrachten von kollaborativen Designpraktiken und den Unterschieden, die beim Einsatz der wichtigsten Designmethode (die des Skizzierens) bei traditionellen und digitalen Arbeiten auftreten, konnten wir die Diskrepanzen erkennen, die beim Arbeiten mit digitalen Artefakten entstehen können.

Bestehende Systeme zeigten uns, dass zwar schon viele Versuche unternommen wurden, kollaboratives Arbeiten auf digitale Medien umzusetzen, man dabei jedoch immer wieder auf Barrieren stieß, da es bisher noch keinem gelang, eine effiziente und transparente Schnittstelle zwischen analogen und digitalen Objekten zu schaffen.

Unsere praktischen Erfahrungen mit *Scribbler* zeigten, dass noch einige Verbesserungen auf diesem Sektor notwendig sind, um diese Barrieren aufbrechen zu können. Jedoch bewegten wir uns mit unserem System einen Schritt in die richtige Richtung.



Teil III  
APPENDIX



Der folgende Abschnitt umfasst Transkripte, von uns geführten Interviews mit Designern aus verschiedenen Designsparten. Sie bieten Einblicke in Arbeitssituationen und Arbeitsweisen in den Bereichen *Produktdesign* und *Interaktionsdesign* bzw. Erfahrungen und Anmerkungen zu Scribbler.

*Anmerkung: Die Abschrift und alle dazugehörigen Materialien, wurden von den jeweiligen Interviewpartnern gestattet. Unter Rücksicht auf Anonymität werden lediglich Vornamen oder Spitznamen zur Unterscheidung verwendet. Von Thomas Nägele und Clemens Sagmeister werden stets die Vornamen verwendet.*



#### A.1 INTERVIEW MIT EINEM PRODUKTDESIGNER

*Peter* ist von Beruf Produktdesigner und arbeitet zusätzlich als Professor an der Universität für angewandte Kunst Wien.

*Thomas*: Bitte erklär uns doch erst ein mal, wie du arbeitest, sprich, wie du ein bestimmtes Projekt abwickelst.

*Peter*: Also es beginnt grundsätzlich so: du kriegst einen Auftrag vom Kunden, ein Briefing. Dort stehen die Anforderungen an dich als Designer, in welche Richtung die Arbeit gehen soll. Danach gibt es eine zwei bis drei Wochen lange Phase, in der du nur Ideen generierst. Ich mache in dieser Phase viele, relativ grobe Skizzen mit einem Kugelschreiber auf Papier, um mich an das Thema anzunähern und gewisse Ideen mal aufzuzeichnen. Die Ideenfindung passiert ja meist über die Zeichnung, das bedeutet man zeichnet irrsinnig viel, ca. 50 Blätter Papier und durch das Visualisieren ergeben sich neue Ideen. Im nächsten Stadium verfeinert man dann einige Skizzen, koloriert sie etwa mit Photoshop und gewisse Details herausarbeitet oder man geht direkt schon über auf das Arbeiten mit digitalen Mitteln. Es besteht natürlich die Gefahr, dass man zu weit ins Detail geht obwohl das in dieser Phase noch niemanden interessiert, da muss man aufpassen.

*Thomas*: Arbeitest du dann mit Pen und Tablet?

*Peter*: Ja genau, am Rechner arbeite ich eigentlich nur selten mit der Maus. Hauptsächlich Pen und Tablet. Photoshop und Illustrator ausschließlich mit Pen und Tablet.

Gut, wenn man dann eine ganze Reihe an Ideen und recherchiert hat, beginnt man die Präsentation vorzubereiten. Das Ganze mache ich in InDesign, dort hab ich bereits meine Vorlagen

und füge dann Texte und Bilder ein und ich versuche das so zu gestalten, dass eine gedruckte Version der Präsentation selbsterklärend ist und der Kunde die Möglichkeit hat, sich das später noch mal anzusehen. Bei der Präsentation selbst kommt ja immer so viel Information in kurzer Zeit auf die Hörer zu, dass es gar nicht möglich ist, alles vollständig aufzufassen. Daher muss man sich das nachträglich auch in Ruhe noch mal ansehen können.

*Thomas:* Verteilst du die Ausdrücke dann schon zu Beginn oder wartest du bis nach der Präsentation?

*Peter:* Das kommt auf die Anzahl der Teilnehmer an. Bei größeren Gruppen muss man ja einen Beamer verwenden, da warte ich bis nach der Präsentation, wenn es hingegen wenig Leute sind, dann arbeite ich sehr gerne nur mit den Ausdrucken und präsentiere direkt am Tisch vor den Leuten. In diesem Fall notiert man natürlich auch Sachen auf der Printversion.

*Clemens:* Gibt es in diesem Stadium auch schon andere Artefakte als Skizzen? Ich meine, verwendest du auch schon andere Materialien als Papier, z.B. Plastilin etc.?

*Peter:* Das kommt auf das Projekt drauf an. Wenn du Designs hast, die davon leben, wo es z.B. auf die Ergonomie drauf ankommt, speziell bei Produkten die gut in der Hand liegen müssen, dann kommt das schon vor. Styropor eignet sich in solchen Fällen oft gut.

*Clemens:* Aber das Skizzieren kommt schon immer vorher oder?

*Peter:* Genau.

*Clemens:* Sind Artefakte beim Designprozess wichtig? Ich meine Modelle, Skizzen, etc.

*Peter:* Ja natürlich, äußerst wichtig. Du entwickelst ja Formen und Gebilde, die du sonst gar nicht beschreiben und dadurch auch nicht anderen kommunizieren könntest.

*Clemens:* Wie sieht es mit Farben aus beim Skizzieren?

*Peter:* Nun, beim skizzieren am Papier mache ich eigentlich nur Schattierungen, keine Farben. Diese füge ich dann später digital hinzu, hauptsächlich mit Photoshop. Es ist nämlich immer etwas schwierig beim Gespräch mit dem Kunden. Wenn man Farbe reinbringt, dann rutscht die Diskussion oft ab und der Kunde versteift sich auf die Farben, obwohl diese eigentlich nur exemplarisch sind und der Fokus der Diskussion eigentlich woanders liegen sollte.

*Thomas:* Kannst du die Skizzen einfach abgeben, sodass der Kunde sich diese selbst ansehen kann oder ist es wichtig, dass du dabei stehst und eine Erklärung bietest?

*Peter:* Ich persönlich erkläre immer meine Designs. Ob das auch anders rum funktionieren kann weiß ich jetzt nicht so recht. Vielleicht wenn man schon in einer weit fortgeschrittenen Phase des Designs ist und alle Beteiligten schon die selben konkreten Vorstellungen vom Resultat haben, könnte man eventuell technische Zeichnungen oder ähnliches auch kommentarlos abgeben.

*Clemens:* Arbeitest du eigentlich auch im Team mit anderen Designern oder hauptsächlich nur alleine?

*Peter:* Nun ja, ich hab schon viele verschiedene Situationen erlebt. Im Moment ist es so, dass ich Produktdesign alleine mache, aber ich habe auch oft schon mit andern Designern zusammengearbeitet. Meistens haben wir dann Screen-Sharing via Internet verwendet und zusätzlich noch telefoniert. Skype kann ja beides und hat sich als nützliches Tool zur Zusammenarbeit herausgestellt. Während der Besprechung skizziert gleich einer der beiden mit, z.B. auf Photoshop, der andere kann das live mitverfolgen und gemeinsam werden die Konzepte so ausgearbeitet.

*Thomas:* Könntest du dir vorstellen, dass zwei Designer gleichzeitig an der selben Skizze arbeiten?

*Peter:* Hmm, meistens ist es schon so, dass jeder für sich skizziert und der andere entweder zuschaut und Inputs liefert oder die Konzepte nachher verglichen und elaboriert werden.

*Clemens:* Bringt das parallele Arbeiten mehr Ideen hervor?

*Peter:* Ja, natürlich.

*Clemens:* Meinst du würden weniger Ideen produziert, wenn beide an der selben Skizze arbeiten würden?

*Peter:* Das ist schwierig zu sagen. Vielleicht würden weniger grundlegend verschiedene Konzepte zustande gebracht werden, aber beim gemeinsamen Ausarbeiten werden die Ideen beider Designer sofort in einem Konzept vereint, das dann andererseits schon besser durchdacht ist. Diese Arbeitsform kann ich mir aber eigentlich nur in einer sehr frühen und groben Ideenfindungsphase vorstellen, denn gleichzeitig am selben zu zeichnen ist sicherlich sehr schwierig.

*Clemens:* Es muss ja nicht direkt gleichzeitig passieren, sondern in Absprache und abwechselnd.

*Peter:* Ja gut, wenn immer nur einer der beiden zeichnet, es dann weiter reicht und der nächste daran weiter zeichnet, so eine Arbeitsform kann ich mir durchaus gut vorstellen.

*Thomas:* Du unterrichtest ja auch Darstellungstechnik an der Universität für angewandte Kunst in Wien. Wie sehen deine Vorlesungen aus, wie vermittelst du dein Wissen?

*Peter:* Nun, ich zeige die Unterlagen am Beamer anhand von PDFs. Darin sind unterschiedliche Arbeitsschritte und Techniken von gewissen Designs zu sehen. Außerdem habe ich immer Hand-Outs mit Beispielen, die ich den Studenten gebe. Auf diesen können sie dann selbst Details erarbeiten, häufig auch in Teams. Generelle wäre es im Unterricht natürlich sehr praktisch wenn man direkt digital vorzeichnen könnte. Das habe ich so noch nie gemacht.

*(Peter beginnt mit Scribbler auf verschiedenen Fenstern zu zeichnen)*

*Thomas:* Könntest du dir vorstellen, dass auch Studenten gleichzeitig mit dir mitarbeiten, sprich auch direkt am selben Screen zeichnen?

*Peter:* Ja das wäre super, wenn ich das im Unterricht einsetzen könnte. Es wäre spitze wenn jeder die Möglichkeit hätte, mitzuarbeiten. Wobei ich nicht ganz sicher bin ob Darstellungstechnik das passende Unterrichtsfach für Scribbler ist, weil da geht es konkret um das Zeichnen und dann reichen Grobskizzen leider nicht aus, sondern man muss schon detaillierte Zeichnungen anfertigen. Anders sieht es da bei Designvisualisierungen aus, die auch in meinem Unterricht vorkommen. Da wäre es wirklich cool, die Studenten direkt einzubinden. Jeder könnte auch Stichworte dazuschreiben und ähnlich einem Brainstorming vernetzen. Das könnte ich mir gut vorstellen.

*Thomas:* Sind diese primitiven Zeichenmöglichkeiten ausreichend, oder fehlt dir da was?

*Peter:* Nein, für Darstellungstechnik reichen diese Möglichkeiten bei weitem nicht aus. Man braucht da Dinge wie Strichstärke, gerade Linien, etc. Das sind Sachen, die Photoshop kann und die sind auch wirklich notwendig. Aber bei solchen groben Sachen, wie ich sie hier jetzt am Screen gezeichnet habe kann das schon reichen. Wichtig ist natürlich auch die Transparenz von Linien. Beim Skizzieren beginnt man ja mit ganz leichten Strichen, die das grobe Grundgerüst darstellen und zeichnet dann mit mehr Druckstärke drüber, sodass die Linien deutlicher werden und die Skizze konkreter wird.

*Peter:* Kennt ihr Graphic-Recording?

*Clemens, Thomas:* Nein.

*Peter:* Das ist eine recht interessante Geschichte. Es handelt sich um Leute, die Meetings mit zeichnen. Das passiert analog auf einem Blatt Papier mit Stift. Nehmen wir an da sitzen mehrere Techniker und andere in ein Projekt verwickelte Personen, die Konzepte verbal besprechen und dann gibt es einen Zeichner, der, während die Leute ihre Ideen artikulieren, diese direkt zu Papier

bringt und aufzeichnet. Darauf baut dann die Diskussion weiter auf und die Teilnehmer können gleich auf die Skizzen eingehen und sie weiter entwickeln oder verwerfen. Am Ende kann man anhand der Bilder nachvollziehen, worüber gesprochen worden ist. Das wäre sicherlich auch ein Gebiet, bei dem man Scribbler oder ähnliche Anwendungen zum Einsatz bringen könnte.

*Clemens:* Bei uns am Institut gibt es öfter Meetings in der Bibliothek, in der mehrere Tische in U-Form stehen und zu einem großen Screen gerichtet sind, wo alle hinsehen können. Wenn jetzt z.B. einer ein Design für eine Webpage macht, kann er das allen zeigen, und für die anderen soll es dann möglich sein mittels Tablets auf das Design zu zeichnen und zu sagen das gefällt mir das nicht usw. Also das war die Grundidee, warum wir begonnen haben das Programm zu entwickeln. Und jetzt suchen wir eben auch Parallelen zu anderen Designsettings, um herauszufinden ob es auch dort Einsatz finden könnte. Du hast uns dazu schon ein wenig Input gegeben. Vielleicht könnten wir jetzt kurz die derzeitigen Features von Scribbler durchbesprechen. Du hast vielleicht vorher nicht bemerkt, bzw. wir habens dir noch nicht gesagt: Die Möglichkeit zu Radieren gibt es eigentlich auch - die hintere Taste am Stift dient dazu. Doch da das Programm vektorbasierend ist, radiert es nur den letzten Strich. Ist diese Funktion zu wenig für deinen Gebrauch?

*Peter:* Also wenn ich wirklich an einem Produkt arbeite, um eine Form herauszuarbeiten, wäre es zu wenig ja. Aber um einfache Sachen, wie z.B Notizen einzufügen oder Ideen zu formulieren, funktioniert es schon. Man müsste aber vielleicht anfangen das Programm häufiger zu verwenden, um ein gutes Feedback abgeben zu können.

*Clemens:* Ich habe bemerkt, dass du bis jetzt nur direkt auf die Bilder bzw. Fenster gezeichnet hast und Notizen auch nur direkt darauf geschrieben hast. Aber du könntest auch außerhalb - wo Platz ist - zeichnen. War das nur Zufall?

*Peter:* Das war nur Zufall ja. Das ist ganz gut, dass man außerhalb zeichnen kann und somit Sachen verknüpfen kann.

*(zeichnet Pfeile von einem Fenster zum anderen)*

Ich bin ein Typ, der mir gerne Zusammenhänge und Strukturen schafft. Ich überlege gerade, wo man jetzt wirklich verschiedene Vorschläge hat.

*(öffnet seine Unterlagen vom Universal Serial Bus (USB)-Stick)*

Ich schau jetzt einmal ob ich noch irgendwo etwas habe mit verschiedenen Entwürfen.

*(öffnet eine Reihe von Bildern)*

Das war auch so eine Entwurfsphase, wo es verschiedene Vorschläge gibt, wie man das umsetzen könnte. Wenn ich mir das

jetzt überleg, dass das in einem Gespräch stattfindet. Da ist es eben darum gegangen verschiedene Formen zu finden, wie man den Inhalt transportieren kann. Das war zb. die erste Idee.

*(schreibt »1. Idee« auf ein Bild)*

*Peter:* Was hier [in Scribbler] auch gut funktioniert, ist das Verdeutlichen von Ideen. Ich bin jemand, der irrsinnig gerne zeichnet zum Reden. Ich könnte so z.B. ein paar Punkte rausholen und einen Teil, der hier im Bereich unten schwer zu erkennen ist, noch einmal rauszeichnen.

*Clemens:* Du zeichnest derzeit mit der Farbe Magenta. Du hast vorher gemeint für Grobskizzen reicht dir eine Farbe oder?

*Peter:* Also wenn ich in der Strukturierungsphase bin, wäre es schon gut wenn ich mehrere Farben hätte.

*Clemens:* Ok. Dazu haben wir eigentlich mehrere Stifte mit unterschiedlichen Farben angedacht.

*Peter:* Wirklich wahr?

*(Peter greift zu einem anderen Stift und probiert ihn aus)*

Wahnsinn. Ja, das finde ich super wenn man das so löst.

*Thomas:* Wir haben uns gedacht wir halten uns daran, möglichst realitätsnahe zu bleiben. Beim Zeichnen auf Papier würdest du auch einen anderen Stift zur Hand nehmen.

*Peter:* Verstehe. Was ich irrsinnig gerne verwende sind Untermenüs bzw. Popupmenüs. Das gibt es bei verschiedenen Programmen. Wenn ich z.B. auf die Stifttaste drücke geht ein Untermenü auf. So wie bei »Autodesk Maya« oder »Sketchbook«. Das sind Programme in denen ich z.B. immer zeichne. Und da hätte man im Untermenü auch die Möglichkeit auf eine Farbpalette, oder vielleicht auch 2-3 verschiedene Strichstärken. Dazu drück ich auf die Stifttaste, fahre mit dem Stift auf das Menü, lass wieder aus, das Menü ist wieder weg und ich habe die neue Einstellung. Beim Zeichnen ist das super; das ist etwas was mir z.B. in Photoshop abgeht.

*Thomas:* Gut, da würde man dann weg von dem Ansatz mit mehreren Stiften gehen, weil es natürlich viel schneller ist.

*Peter:* Ja, aber es funktioniert dann schon intuitiv. Man kann es echt gut benutzen. Wieviele Stifte kann man da nehmen?

*Thomas:* Unendlich viele.

*Peter:* Aha. Also das finde ich auch eine gute Sache mit unendlich vielen Farben.

*Clemens:* Mit der Maus navigierst du nicht besonders viel. Machst du alles nur mit dem Stift?

*Peter:* Wenn ich arbeite, arbeite ich ausschließlich mit dem Stift ja. Mit dem Stift und der Tastatur. Meine Arbeitssituation sieht immer so aus, dass ich das Tablet vor mir habe und eine Box das Tablet auf der Hinterseite leicht anhebt, sodass ich es gleich in der richtigen Schräge habe. Die Tastatur habe ich links daneben stehen, womit ich leicht Shortcuts erreichen kann.

*Thomas:* Das heißt du zeichnest am Liebsten in der Schräge?

*Peter:* Genau. Also leicht angehoben ja. So zeichne ich wesentlich entspannter. Ich habe zusätzlich ein A3 Tablet und dadurch werde ich gezwungen größere Bewegungen zu machen, was gut für meinen Rücken ist.

*Clemens:* Um noch einmal auf die Farben zurückzukommen: Wenn du dir vorstellst es arbeiten 5 Personen gleichzeitig an einem Design, dann braucht man vielleicht eine Unterscheidung vom jeweilig Gezeichneten. Wir dachten daran, dass jeder eine eigene Farbe bekommen könnte. Glaubst du das wäre problematisch, weil du vorher auch angemerkt hast, dass Farben oft mehr aussagen?

*Peter:* Das mit den Farben würde ich so sehen: Vom Einsatz her, würde ich sagen dass du damit keine Präsentationszeichnungen machen kannst [wo man mehrere Farben benötigt]. So eine Zeichnung zu machen ist echt nicht einfach. Dazu braucht man Zeit und Konzentration. Dass daran mehrere Leute arbeiten kann ich mir nur schwer vorstellen. Zur Besprechung und Ideenfindung kann ich mir es sehr gut vorstellen.

*Thomas:* Ich habe gesehen, du würdest dir auch wünschen dass du einzelne Teile von Zeichnungen wo anders hinschieben könntest oder? Also dass du einen bestimmten Bereich ausschneiden und verschieben könntest.

*Peter:* Ja, das wäre vielleicht speziell bei der Ideenfindung, wo alles durcheinander steht, oder auch bei Skizzen in einem Brainstorming technischer Natur interessant. Weil meistens ist der zweite Schritt dann der, dass ich anfangs die Ideen zu ordnen. Also es wäre gut, um den ersten Prozess der Ideenfindung nicht zu stoppen bzw. ihm eine Hürde zu geben, sondern gleich weiterarbeiten zu können. Und da ist es eben notwendig das Ganze in eine Ordnung zu bringen. Dann hätte ich wirklich eine Lösung wo alle gemeinsam arbeiten können.

*Thomas:* Wie machst du das wenn du mit Papier arbeitest? Schneidest du Skizzen aus und klebst sie irgendwo neu auf?

*Peter:* Nein. Also ich hab einen bestimmten Workflow wie ich an die Sache heran gehe: Wenn ich im Ideenfindungsprozess bin, mach ich zuerst einmal Cluster bzw. Assoziationsketten, wo ich aus dem Bauch heraus das Thema beschreibe, um einmal

abzuchecken, was dazu in meinem Kopf ist. Danach nehme ich ein neues Blatt Papier her und beginne die verschiedenen Skizzen zu kategorisieren indem ich die Skizzen geordnet neu zeichne.

*Thomas:* Wird das dann auch genauer?

*Peter:* Das wird auch genauer ja. Anfangs habe ich einen Pool von Gedanken und Assoziationen und die kann ich dann her nehmen und nach verschiedenen Kategorien einteilen.

*Clemens:* Wäre das ein Vorteil für dich, wenn du eine elektronische Unterstützung hast, wo du deine Zeichnungen sofort neu ordnen könntest?

*Peter:* Jaja. Na das wäre super. Das ist zum Beispiel immer das Problem wenn du mit »Illustrator« oder »Photoshop« arbeitest; da ist das sehr umständlich. In dem Bereich könnte ich mir das ganz gut vorstellen. Vor allem wenn verschiedene Leute mitarbeiten, dann würde »Illustrator« usw. mit der Werkzeugleiste wahrscheinlich überhaupt nicht funktionieren. Das würde für die Ideenfindung super funktionieren, wenn wirklich 3-4 Leute auf einen Screen arbeiten und das auch weiterverwenden können.

*Clemens:* Könntest du zusätzlich auch eine reine weiße Fläche zum Zeichnen brauchen? Also eine Whiteboard Funktion? Oder reicht es dir, auf Fenster zeichnen zu können?

*Peter:* Ich finde es grundsätzlich gut, wenn sich jeder ein eigenes Fenster hernehmen und dann direkt auf die vorhandenen Designs zeichnen kann. Ein zusätzliches Whiteboard wäre aber auch super.

*Thomas:* Ein weiterer interessanter Punkt für uns ist, wie man das Gezeichnete abspeichern könnte. Momentan gibt es dafür einen simplen Screenshot. Ist es aus deiner Sicht notwendig, etwas so abzuspeichern, damit man an einem späteren Zeitpunkt daran wieder weiter arbeiten kann, oder reicht dir ein Screenshot?

*Peter:* Das ist schwer zu sagen. Dazu müsste ich das Programm in meinen Workflow einbinden. Natürlich wäre es eine tolle Sache, wenn man beim Öffnen wieder genau die selbe Ansicht hat, auf der man zuvor gearbeitet hat. Aber ich kann jetzt nicht sagen, ob das ein wahnsinns Vorteil wäre. Dazu müsste ich es ausprobieren.

*Thomas:* Habe ich es richtig verstanden, dass du es gerne so hättest, dass wenn mehrere gleichzeitig zeichnen, jeder auf sein eigenes Fenster zeichnet und die jeweiligen Zeichnungen auf dem eigenen Fenster kleben bleiben? Momentan hängen alle Zeichnungen - egal von wem gezeichnet - nur auf dem aktivem Fenster. Sobald du das Fenster verschiebst, verschiebst du die Zeichnungen mit.

*Peter:* (denkt nach)

Weiß ich nicht. Also für mich funktioniert das derzeit von der Überlegung her recht gut. Aber ich müsste es erst länger ausprobieren, um auch die wirklichen Stärken zu finden.

*Thomas:* Hättest du Lust das Programm im Unterricht oder in der Arbeit auszuprobieren?

*Peter:* Zum Ausprobieren wäre es sicher toll. Interessant wäre es auch wenn man übers Netz auf einen Bildschirm zugreifen könnte. Weil ich sehr oft so arbeite, dass ich nicht im selben Raum sitze, mit den Leuten mit denen ich zusammenarbeite. Das wäre dann aber wahrscheinlich wieder ein eigenes Programm. Aber gerade für den Ideenfindungsprozess und Besprechungen wäre es sehr interessant.

*Clemens:* Zum Schluss würden wir gerne noch deine Meinung zu zukünftig angedachten Features hören. Da wir mit elektronischen Artefakten arbeiten, die durch den gemeinsam genutzten Bildschirm weiter von einem entfernt sind, entsteht eine Hürde bei der Gestik. Deswegen planen wir eine Zeigefunktion zu integrieren. Glaubst du würde ein am Screen angezeigter Laserpointer oder - vielleicht abstrakter - eine neigbare, durch den Tabletstift steuerbare Hand bei dem Problem helfen?

*Peter:* Würde ich toll finden ja. Speziell bei Präsentationen kommt es immer wieder zu Verwirrungen wenn jemand etwas zu einem Design anmerkt, weil die anderen nicht wissen von was genau gesprochen wird. Natürlich könnte man jedem einfach einen Laserpointer in die Hand drücken. Aber ich kann mir das auch gut vorstellen wenn man den Cursor z.B. durch eine Hand austauscht, oder zu jedem Cursor den Namen dazu schreiben kann. Man müsste dann eben Umschalten können zwischen der Zeichenfunktion und der Zeigefunktion.

*Clemens:* Eine weitere Idee von uns wäre den Screen für andere Computer im Netzwerk zu öffnen, damit Benutzer z.B. ein Fenster mit eigenen Content von ihrem Notebookscreen auf den großen Bildschirm ziehen können, auf das dann jeder zeichnen kann. Das könnte von Vorteil sein, da die meisten ihre eigenen Unterlagen nur bei sich am Rechner haben und man somit die Informationen für alle leicht zugänglich machen kann. Denkst du wäre das ein brauchbarer Ansatz?

*Peter:* Was mir dazu einfällt ist, dass man bei Präsentationen oft mehrseitige Dokumente hat. Interessant wäre es nun die Seiten wie im »Acrobat Reader« auf der Seite in einer Miniaturansicht anzuzeigen, damit dann jeder der auf ein bestimmtes Thema zugreifen möchte (und eine gute Präsentation hat eben die ganzen Themen drinnen), die Seite rausziehen kann.

*(denkt kurz nach)*

Im Prinzip kann man das natürlich auch jetzt schon über eine Dateienverwaltung machen. Aber das wäre vielleicht ein nettes Feature, weil die Leute dann aktiv bei der Präsentation mitarbeiten können. Und wenn man dann auf einer rausgezogenen Seite gezeichnet hat und sie wieder minimiert, dann wandern die Zeichnungen mit usw. So wie es jetzt schon mit dem aktiven Fenster funktioniert.

*Thomas:* Stören dich eigentlich die nachträglichen Korrekturen, die bei den gezeichneten Strichen vorgenommen werden?

*Anmerkung:* Um Rechenleistung zu sparen, werden Linien von Scrib-  
bler während dem Zeichnen als Verkettung mehrerer Geraden dargestellt und erst nachträglich nach dem Absetzen des Stifts zu runden Kurven umgerechnet.

*Peter:* Nein, überhaupt nicht. Natürlich würde es bei genauen Arbeiten nicht funktionieren, aber für die Ideenskizzen usw. ist es überhaupt kein Problem.

*(probiert genaues Zeichnen aus)*

Mit ein bisschen Übung bekommt man aber auch das hin.

*Clemens, Thomas:* Danke, das waren unsere Fragen. Hast du sonst noch Fragen oder Anregungen?

*Peter:* Nein, das hat mir jetzt echt Spaß gemacht. Es ist super - ich müsste nur anfangen damit zu arbeiten um genaueres sagen zu können. Aber ich hoffe ich habe euch helfen können. Für mich ist das auf jeden Fall eine Traumsituation, da ich jetzt schon Notizen zu Präsentationen mache. Nur das Problem ist eben jetzt, dass ich die Notizen auf Zettel mache und mir dazuschreiben muss zu welcher Zeichnung die Notizen gehören, damit ich sie nachträglich wieder zuordnen kann.

*Thomas:* Wäre das Programm deiner Meinung nach schon reif genug, um es in einer deiner Arbeitssituationen ausprobieren zu können?

*Peter: (denkt nach)* Ob ich es wirklich sofort einsetzen würde bei einer Präsentation, weiß ich jetzt nicht. Was den Unterricht anbelangt, müsste ich meine Arbeit umstellen und sie darauf anpassen. Wo ich mir den Einsatz gut vorstellen kann, ist im privaten Rahmen wo ich in Teams arbeite.

*Clemens, Thomas:* Super. Vielen Dank.

## A.2 INTERVIEW MIT ZWEI USER EXPERIENCE DESIGNERN

*Pi* und *Zed* sind von Beruf User Experience Designer und arbeiten hauptsächlich für Unternehmen, die im Bereich Telekommunikation tätig sind.

*Clemens*: Bitte erzählt uns kurz was ihr macht und wie ihr arbeitet. In welchen Situationen macht ihr Gebrauch von Skizzen?

*Pi*: wir skizzieren hauptsächlich auf Flipcharts beim Kunden selbst. Heute beispielsweise saßen wir zusammen in einem Meetingraum und haben Bedienungsabläufe am Flipchart gezeichnet. Das Kundenmeeting wird eigentlich oft zu einem Design Workshop auch wenn das so gar nicht geplant ist. Manchmal dreht man die Situation auch konkret so hin, dass es zu einem Design Workshop wird, weil sich der Kunde die Dinge dann besser vorstellen kann.

*Clemens*: Zeichnen die Kunden auch selbst?

*Pi*: Ja das kommt durchaus vor. Ich zeichne bestimmte Konzepte auf und Kunden schreiben oder zeichnen selbst etwas dazu.

*Clemens*: Also braucht ihr das Zeichnen hauptsächlich vor Ort beim Kunden?

*Zed*: Nein es gibt natürlich auch die Situationen in denen wir hier gemeinsam im Team skizzieren: auf Whiteboards, Blöcken oder auch auf tischgroßen Papieren.

*Clemens*: Ok, wollt ihr noch kurz erklären, was ihr allgemein macht?

*Zed*: Ja, wir sind beide Geschäftsführer der intuio User Experience Consulting GmbH, die wir gemeinsam im Jahre 2008 gegründet haben. Wir leben von Ablauf- und Interaktionsdesign, wo Sketches und Visualisierung von Abläufen und Zeichnungen eine große Rolle spielen, natürlich auch in Kollaboration mit Kollegen und Kunden.

*Clemens*: Super. Wir haben euch ja schon erzählt, was Scribbler ist. Vielleicht probiert ihr es einfach mal aus.

*(Zed nimmt den Stift in die Hand und beginnt zu zeichnen)*

*Zed*: Also ich hatte leider noch nie so ein Wacom Tablet zum probieren. Ich bin da etwas ungeschickt... Ah, so funktioniert das also, wenn man den Stift aufsetzt wird sofort gezeichnet, ich dachte dann würde nur der Cursor bewegt und man zeichnet mit mehr Druck oder einer bestimmten Taste.

*(Übergibt den Stift an Pi, der auch probieren will)*

*Pi*: Aha, er reagiert schon wenn man in die Nähe kommt. Für mich wirkt die absolute Projektion des Tabletbereichs auf

den Bildschirm nicht sehr intuitiv. Ich würde den Cursor lieber »schieben«, wie bei einer Maus.

*Thomas:* Bei Bedarf kann man die Projektion auch relativ einstellen.

*Pi:* Ah, ok. Wozu sind eigentlich diese Tasten hier gut? (Zeigt auf die zwei Tasten am Stift)

*Clemens:* Das sind die Funktionen Undo und Redo.

*Zed:* Aah, ich dachte die wären zum zeichnen... Gibt es sonst noch Tastenfunktionen?

*Clemens:* Ja, diese Tasten hier am Tablet sind mit Funktionen belegt. Man kann einen Screenshot machen, alle Linien löschen, ein Whiteboard einblenden und in den Mausmodus wechseln. Im Mausmodus zeichnet der Stift nicht, sondern kann als Maus verwendet werden. Außerdem ist ein Radierer am hinteren Ende des Stifts.

*Zed:* Ah, sehr gut.

*Pi:* Also das Radieren find ich weitaus besser als die Undo Funktion.

*Zed:* Kannst du eigentlich schreiben? Ich mein, mit dem Stift Dinge auf den Screen schreiben.

*(Pi versucht zu schreiben)*

*Zed:* Nicht so gut oder?

*(Pi hat Schwierigkeiten)*

*Clemens:* Warum tust du dich schwer beim Schreiben?

*Pi:* Hmm, warum ist das so..? Moment... erstens sitze ich nicht gerade zum Tablet... Ich versuche jetzt, auf dieser Linie gerade zu schreiben. Ah, ungenau wird das,

*Zed:* Ist die Übersetzung ungenau?

*Pi:* Naja, wenn du nahe hinkommst reagiert der Cursor ja schon und diese leichte Verschiebung bevor man wirklich aufsetzt, stört mich.

*Zed:* Ja, daher würde ich lieber kontrollieren können, wann ich tatsächlich zu schreiben beginne. Darf ich mal probieren?

*(Nimmt den Stift)*

*Zed:* Uah... Ja das ist wirklich schwierig. Ich komm nicht genau hin, wie ich mir das vorstelle. Wieso wird das so schief?

*Thomas:* Das liegt daran, dass du schief zum Tablet sitzt.

*Zed:* Ach deswegen. Ja es ist schwierig.

*Clemens:* Also weil der Cursor nicht dort ist, wo man ihn erwarten würde.

*Pi:* Genau.

*Zed:* Mir fehlt die Übersetzung vom Tablet auf den Screen. Vielleicht wäre eine relative Projektion doch besser.

*Clemens:* Ok, probieren wir es aus.

*(Stellt die Projektion um)*

*Clemens:* Würde es helfen, wenn der Stift erst auf Druck reagieren würde?

*Zed:* Ja, denke schon. Vielleicht wäre ein Tablet mit integriertem Screen besser, obwohl die auch diesen Hover Effekt haben.

*Clemens:* Dass man sieht, wo man aufsetzt, sollte es erleichtern.

*Zed:* Ja, du hast Recht. Die Übersetzung, oder anders gesagt, das Mapping fällt natürlich weg aber trotzdem würde der Hover Effekt stören.

*(Clemens hat inzwischen die Projektion umgestellt)*

*Pi:* Aah, wesentlich besser! Probier mal.

*(Gibt Zed den Stift)*

*Zed:* Ja, der Cursor wirkt nun ruhiger. Ah, schau her, die Linien werden ja nachträglich geglättet. Wenn ich schnell zeichne, wird die Linie erst kantiger. Das ist für schnelle Schreiben wahrscheinlich etwa ungünstig.

*Thomas:* Das liegt daran, dass weniger Punkte registriert werden, wenn man den Stift schnell bewegt.

*Pi:* Gut, es ist ja auch eher zum Zeichnen gedacht als zum Schreiben. Obwohl, beim Skizzieren muss man schon immer auch ein paar Notizen dazu machen.

*Zed:* Ja das ist schon sehr wichtig. Also ich finde das Hauptproblem nach wie vor die Übersetzung, aber das ist bestimmt auch Übungssache. Das Problem hat man ja auch mit der Maus, aber die benutzt jeder schon seit Jahren und kann daher gut damit umgehen. Trotzdem fände ich die Wacom Tablets mit integriertem Screen die bessere Lösung.

*(Pi zeichnet außerhalb des Browserfensters)*

*Zed:* Ah, jetzt sind wir außerhalb des Fensters. Das bedeutet die Zeichenfläche ist nicht auf das Fenster beschränkt. Versuch mal ein wirkliches Interface Redesign zu skizzieren auf dieser Website, damit wir sehen, ob das Programm den eigentlichen Zweck erfüllt.

*(Pi zeichnet)*

*Zed:* Ja man sieht, wenn man zu gewissen gezeichneten Dingen etwas dazuschreiben möchte und es ist nur wenig Platz vorhanden, dann wird's schwer. Zum Schreiben ist das leider zu ungenau.

*Pi:* So, wenn ich jetzt eine Linie vom Ende der Website bis oben zum Anfang ziehen möchte, dann müsste ich gleichzeitig scrollen können.

*Thomas:* Puh, das geht leider nicht.

*(Pi zeichnet eine Linie, setzt ab, scrollt nach oben und verlängert dann die Linie)*

*(Pi versucht wieder zu schreiben)*

*Zed:* Schreiben ist nichts. Schreiben ist fast unmöglich.

*(Pi versucht einen Textabsatz einzukreisen, es gelingt ihm jedoch nicht besonders gut)*

*Zed:* Warum geht das nicht, warum funktioniert das nicht so wie man das möchte?

*Clemens:* Das Schreiben?

*Zed:* Nein, das Einkringeln von bestimmten Dingen. Die Kreise werden so schief und verzogen.

*Thomas:* Es ist schwierig, am Tablet zu zeichnen, während man auf das Display schaut, das muss man üben.

*(Pi zeichnet inzwischen einen Strich und erreicht dabei mit dem Stift den Rand des Tablets, am Screen geht der Strich jedoch nicht ganz bis zum Displayrand)*

*Pi:* Ah, schau, jetzt bin ich nach oben gefahren und jetzt steh ich hier an.

*Thomas:* Ja das kommt durch die relative Projektion, die wir am Tablet eingestellt haben.

*Pi:* Ach so.

*Clemens:* Vielleicht sollten wir doch wieder umstellen.

*Pi:* Nein das passt schon so.

*Zed:* Können wir die Beschleunigung runter drehen?

*Clemens:* Ja, klar.

*(Verringert die Beschleunigung)*

*Zed:* Ok, das ist jetzt sehr langsam. (lacht)

*Clemens:* Gut, ich stelle mal wieder um auf absolute Projektion.

*Pi:* Ok.

*(Stellt die Projektion um)*

*Pi:* Ja es ist einfach schwierig mit dieser Hardware. Die Umsetzung find ich super, aber durch die Eigenheiten der Hardware wird die Benutzbarkeit beeinträchtigt.

*Clemens:* Was haltet ihr von der nachträglichen Glättung der Linien? Stört euch das?

*Pi:* Also ich hab es gar nicht bemerkt.

*Zed:* Ich finde es nett, es muss auch sein, glaub ich. Mich stört aber die Abtastfrequenz, weil wenn man schnell zeichnen möchte, dann funktioniert das nicht optimal.

*Clemens:* Wenn wir das Programm auf einem stärkeren Rechner laufen lassen, geht es besser.

*Zed:* Ah, ja.

*Thomas:* Die eigentlichen Key-Features haben wir ja noch gar nicht gesehen. Ich spreche davon, dass Gezeichnetes an Fenstern und Inhalten haften bleibt. Das bedeutet, dass man Fenster verschieben und darin scrollen kann und die Zeichnungen automatisch mit wandern.

*Zed:* Das Highlighting des aktiven Fensters, auf das gezeichnet wird, irritiert mich etwas. Weil es jedes mal kommt, wenn ich mit dem Stift zum Tablet hingehe. Hmmm... das müsste glaub ich nicht immer da sein. Außer es ändert sich das aktive Fenster.

*(Wechselt zwischen Programmen und bemerkt, dass die Zeichnungen der inaktiven Programme nicht angezeigt werden)*

*Zed:* Wenn man umschaltet (zwischen den Programmen), dann ist die Linie außerhalb dieses Fensters auch weg, richtig?

*Thomas:* Ja, genau.

*Pi:* So, wenn ich wieder zum anderen Programm wechseln möchte, muss ich jetzt extra in den Mausmodus umschalten.

*Clemens:* Naja, du kannst auch die Maus direkt benutzen.

*Pi:* Aaah, es funktioniert beides.

*Zed:* Aaah, das ist sinnvoll.

*Pi:* Ja, mir hat die ganze Zeit so was wie Gestensteuerung gefehlt oder so. Aber jetzt wenn ich mit dem Stift zeichnen und mit der Maus steuern kann, dann ist das gut.

*Pi:* Vielleicht ist die relative Projektion des Tablets dann sinnvoll, wenn man zwei Eingabegeräte verwendet. Dann setze ich mit der Maus den Zeiger dorthin, wo ich ihn möchte und kann sofort dort los zeichnen, egal wo am Tablet sich mein Stift befindet.

*Zed:* Aah, ja das ist eine gute Idee.

*Pi:* Oder man könnte auch überhaupt zwei Tablets haben, das eine zur Steuerung mit Gesten und das andere zum Zeichnen.

*Zed:* Da würde sich doch das Magic Pad von Apple anbieten.

*(Zed beobachtet, wie Pi zeichnet und zwischen Fenstern wechselt)*

*Zed:* Das heißt es hängt alles an einem Fenster dran... hmmm, kannst du mal bitte das hintere Fenster aktivieren und es so verschieben, dass man die Fenster und Zeichnungen dahinter sieht?

*Thomas:* Hmm, das geht so nicht. Es werden immer nur die Zeichnungen auf dem aktiven Fenster angezeigt.

*Zed:* Ah, hmm. Ich glaub das irritiert mich, dass die Zeichnungen verschwinden. Vielleicht sollte man die ausgegraut anzeigen, so wie die inaktiven Fenster selbst.

*Clemens:* Es besteht dann die Gefahr, dass ein Chaos entsteht, wenn zu viele Linien angezeigt werden.

*Zed:* Ja stimmt, aber man könnte doch mit verschiedenen Alpha-Werten arbeiten, je nach dem, wie weit hinten sich ein Fenster befindet.

*Thomas:* Oh, ja das klingt nach einer guten Idee.

*Pi:* Man könnte dadurch den Kontext wahren und sehen was man bereits gemacht hat.

*Clemens:* Gut, wenn aber jetzt angenommen auf drei Fenstern gezeichnet wurde und die Zeichnungen ragen über die jeweiligen Fenster hinaus, wäre es dann nicht schwierig den Überblick zu behalten, welche Zeichnungen zu welchem Fenster gehören?

*Pi:* Könnte man nicht auch jedem Fenster eine bestimmte Farbe geben, sodass z.B. alle Zeichnungen von Fenster A rot und alle Zeichnungen von Fenster B blau sind?

*Thomas:* Da muss man aufpassen und bedenken, dass die Inhalte der Fenster ja nicht zwangsläufig weiß sein müssen, wie es bei vielen Webseiten der Fall ist. Es könnte dann passieren, dass die Zeichnungen zu wenig Kontrast zum Fenster haben.

*Pi:* Ja, das stimmt.

*Thomas:* In einem Testsetting hatten wir daher mal mehrere Stifte und jeder hatte eine bestimmte Farbe zugeordnet. Man konnte dann mit beliebigen Farben zeichnen.

*Pi:* Ah, ja. Außerdem könnte man Farben auch nutzen um verschiedene Benutzer zu repräsentieren.

*Thomas:* Genau.

*Zed:* Wie sieht es eigentlich mit der Strichstärke aus? Kann man die anpassen?

*Thomas:* Im Prototypen haben wir das nicht umgesetzt. Prinzipiell würde die Strichstärke über den Druck des Stifts auf dem Tablet geregelt. Photoshop kann das zum Beispiel.

*Zed:* Ok. Das wäre sicher ein nettes Feature.

*(Denkt nach)*

*Zed:* Aber in der Praxis kann ich mir das schwer vorstellen *(Anm.: den Einsatz von Scribbler)*.

*Thomas:* Warum?

*Zed:* Weil es einfach noch so unrund läuft in der Bedienung. Beim kollaborativen Zusammenarbeiten mit dem Kunden glaub ich nicht dass es funktionieren würde. Der Kunde kann das nicht bedienen, allein der Umgang mit dem Tablet ist eher komplex. Hinzu kommt die notwendige Hardware, die ja auch erst ein mal angeschafft werden muss und transportiert werden muss. Man benötigt dann ja mehrere Tablets.

Ja das sind so die Gründe, die mir einfallen. Aber hauptsächlich liegt es an der unausgereiften Bedienung. Es müsste auch eine Möglichkeit geben, die Sachen abzuspeichern, vielleicht nicht nur ein Screenshot.

*Thomas:* Ja, es gibt ein paar Überlegungen unsererseits, wie man das sinnvoll speichern könnte. Eine Variante sähe so aus, dass Scribbler sich merkt zu welchem Fenster eine Zeichnung gehört und würde anbieten diese zu laden, sobald erkannt wird, dass der User jenes Fenster geöffnet hat. Die zweite Variante würde einen Screenshot der Fenster machen und die Zeichnungen separat speichern. Der Screenshot könnte dann halb transparent über den Desktop gelegt werden, sodass der Benutzer darunter die ursprünglichen Fenster wieder so anordnen kann, wie sie hin gehören.

*Zed:* Uh... das stelle ich mir schwierig vor. Da müsste ich echt alles wieder genau hinkriegen wie es war. Wenn es sich um komplexe Zeichnungen handelt ist das aufwändig und zeitintensiv. Total mühsam. Fraglich auch wie das dann aussieht wenn ich was in einer anderen Bildschirmauflösung öffne.

*Thomas:* Stimmt, das hatten wir so noch nicht bedacht.

*Pi:* Gut, die Vektorzeichnungen könnte man ja relativ einfach skalieren, wenn man die Bildschirmauflösung kennt.

*Clemens:* Das Problem ist natürlich, dass wir keinen Einfluss auf andere Programme nehmen können. Daher war der Screenshot die einfachste und natürlichste Lösung. Trotzdem wollen wir noch andere Konzepte ausarbeiten. Es gäbe auch die Möglichkeit ein Photoshop File zu sichern, das verschiedene Layer mit den einzelnen Fenstern und Zeichnungen beinhaltet.

*Zed:* Aber ich kann mir auch den praktischen Nutzen nicht so recht vorstellen. Zu sagen, ich würde das System wirklich irgendwo einsetzen... ich weiß nicht. Das Szenario fehlt mir. Beim Kunden fällt das nämlich komplett flach.

*Thomas:* Glaubt ihr würde es besser funktionieren, wenn man Tablets mit integrierten Displays einsetzen würde? Der Benutzer sieht ja dann worauf er zeichnet und diese Übersetzung von Tablet auf Beamer fiel flach.

*Zed:* Da muss ich jetzt aber auch noch mal nachfragen, denn es gibt ja beispielsweise am iPad schon einige Apps, mit denen man skizzieren kann. Wo wäre dann der Mehrwert? Obwohl, ich glaub ich kenn keine kollaborativen Tools dieser Art.

*Pi:* Aber mit dem Finger zu zeichnen ist Wahnsinn, das kannst du vergessen.

*Zed:* Da gibt es doch auch so eigene Stifte, die man kaufen kann.

*Clemens:* Leider funktionieren die nicht wirklich. Ich hatte mal die Gelegenheit einen auszuprobieren und muss sagen es war enttäuschend. Man muss viel zu fest drücken, das ist nicht wirklich ergonomisch beim Zeichnen.

*Zed:* Ah, ok schade.

*Pi:* Ich glaube die Touch-Technologie vom iPad ist einfach nicht geeignet für solche Dinge.

*Clemens:* Liegt das Problem von Scribbler dann eigentlich eher bei der Hardware?

*Pi:* Sehe ich schon so, ja.

*Zed:* Na, ich muss auch sagen, ich sehe das Anwendungsszenario für mich einfach nicht.

*Pi:* Gut, wir sind halt auch keine Grafiker, ich weiß nicht ob die der Sache nicht besser gegenüber stünden.

*Zed:* Richtig.

*Clemens:* Würdet ihr eine normale Flipchart-Applikation bevorzugen?

*Pi:* Naja, beim Erstentwurf, wenn es noch nichts gibt, dann ist natürlich eine weiße Fläche das sinnvollste zum Sketchen. Aber sicher, es wäre natürlich schon fein, wenn man bereits gefertigte Wireframes beim Kunden zeigt und dort dann ad-hoc Notizen und Skizzen dazu machen könnte. Das wäre sicherlich ein nettes Anwendungsszenario.

*Zed:* Wer notiert? Du oder der Kunde?

*Pi:* Wäre beides denkbar, wobei ich mir nicht sicher bin ob was Gescheites herauskommt wenn der Kunde da rumkritzelt.

*Zed:* Da kommt nur Blödsinn raus (*lacht*).

*Pi:* Ja bei diesen kollaborativen Anwendungen ist natürlich auch immer die Frage, wie geordnet dieses Zusammenarbeiten von statten geht.

*Clemens:* Wir haben auch eine Whiteboard-Funktion angedacht, sodass man einfach mal auf eine weiße Fläche zeichnen kann.

*Zed:* Auf jeden Fall, ja. Das finde ich wichtig. Ich glaube sogar, dass das eher ein Anwendungsszenario produzieren könnte, denn Fenster darunter liegen zu haben und auf dem Desktop zu zeichnen, das ist glaub ich selten notwendig. Soll das ganze eigentlich auch online durchführbar sein oder nur im lokalen Netzwerk?

*Clemens:* Eigentlich nur lokal, naja, nicht mal im Netzwerk, denn es hängen ja alle Tablets am selben Rechner.

*Zed:* Ah, ok. Also das kann ich mir dann wirklich nur eher auf so einem Tabletop-System vorstellen, das groß genug ist, dass mehrere Leute gleichzeitig arbeiten können. Ich denke das wäre die Einstiegsbarriere gering genug. Aber diese Setup hier, da bin ich ehrlich gesagt skeptisch.

*Pi:* Cool wäre natürlich, wenn man das so mit einer GUI-Widget-Library verbinden könnte. Dass man beispielsweise GUI-Elemente grob zeichnen kann und das System erkennt, sagen wir mal eine DropDown Liste und ersetzt das gezeichnete mit einer schönen Vektorgrafik. Dann könnte man sehr schnell und einfach GUIs aufzeichnen und hätte dann schon schön elaborierte Entwürfe.

*Zed:* Das ist dann eigentlich Mustererkennung.

*Pi:* Genau, im Prinzip kannst du dir dann deine Applikation zusammen zeichnen. So was würde ich schon cool finden.

*Clemens:* Und so wie das System ist, ohne die Mustererkennung meinst du geht das nicht?

*Pi:* Naja, es sieht dann halt so aus (*zeigt auf eine schlecht gezeichnete DropDown Liste*). Ich denke dabei an die Präsentation beim Kunden. Da muss das dann schon gut aussehen. Oder auch anders: Wir hatten heute erst ein Meeting bei dem wir Entwürfe auf ein Whiteboard gezeichnet haben und einer muss danach hergehen, die Skizzen abfotografieren und schöne Wireframes daraus machen. Diesen Arbeitsschritt würde man sich dann natürlich sparen.

*Zed:* Ich habe den Eindruck, dass das so nur von Experten benutzt werden kann, weil die Technik da noch eine große Hürde darstellt. Aber auch beim Erkennen von GUI Elementen muss

ich wissen, wie muss ich das Zeichnen, damit es erkannt wird. Außerdem muss ich alle Elemente auswendig kennen, denn irgendwo alle auf einen Blick hab ich ja nicht. Generell stelle ich den Nutzen vom System, so wie es jetzt ist, in Frage. Die Einstiegshürden, verbunden mit den Hardwarekosten... ich bin nach wie vor skeptisch.

*Thomas:* Was wäre notwendig, um es für dich nutzbar zu machen?

*Zed:* Dieser komischer Hovereffekt des Stifts am Tablet bereitet mir Schwierigkeiten. Das müsste man auf jeden Fall irgendwie lösen. Sodass man das Gefühl bekommt, dass man wirklich exakt arbeiten kann mit dem Ding. Das fehlt mir. Schön wäre natürlich auch, wenn man die Übersetzung von Tablet auf Screen lösen könnte, wobei das natürlich nur dann geht, wenn der Screen im Tablet integriert ist. Auch cool wäre, wenn man über das Internet zusammenarbeiten könnte, denn so ist man an dieses Setting im Raum gebunden, nicht mal über ein lokales Netzwerk hat man da irgendwelche Freiheiten.

*Pi:* Gut, es ist offensichtlich: Mit dem Stift kann ich immer noch nicht besser zeichnen, als mit der Maus. Da muss die Technologie verbessert werden. Klar ist das Übungssache, aber ein Bleistift und ein Blatt Papier sind einfach nach wie vor einfacher zu handhaben.

*Clemens:* Wir haben noch eine Idee, wie man dieses Übersetzungsproblem von Tablet auf Screen eventuell lindern könnte. Man weiß ja nie so recht, wo man sich gerade befindet am Screen. Bei der Literaturrecherche sind wir auf ein System gestoßen, das die Hände der Benutzer beim Zeichnen filmt und halbtransparent am Screen darstellt, sodass man sieht, wo sich die eigene Hand befindet.

*Pi, Zed:* Aaah, gute Idee.

*Clemens:* Also könntet ihr euch vorstellen, dass das helfen würde.

*Pi, Zed:* Ja.

*Pi:* Aber ist das nicht eigentlich ein Workaround für etwas, das eigentlich umgekehrt gelöst werden sollte? Ich meine, man projiziert die Hände des Users auf den Screen aber eigentlich möchte er den Screen bei seiner Hand haben, so wie bei einem Blatt Papier. Das ist eine technische Lösung weil es so einfacher geht.

*Clemens:* Der Vorteil wäre natürlich, dass zwei Personen am selben Ort zeichnen könnten, ohne dass sich die Hände in die Quere kommen.

*Zed:* Ah, gut, das ist aber kein Problem, das ich gelöst haben will, glaub ich.

*Clemens:* Ihr habt vorhin angesprochen, dass Kunden gerne vorne am gebeamteten Bild Sachen zeigen. Was würdet ihr von einer Zeigefunktion halten? Z.B. könnte man einen digitalen Laserpointer implementieren oder ähnliches?

*Zed:* Ich glaube die Tatsache, dass man aufstehen und nach vorne gehen muss, um etwas zu zeigen ist schon wichtig und gut so. Von der Ferne mit einem digitalen Laserpointer drauf zu zeigen ist einfach etwas anderes.

*Clemens:* Und wenn man sich überlegt, dass die Hand eingeblendet wird am Screen, damit man mit dem Finger über die Distanz zeigen kann?

*Zed:* Ich glaube nicht, dass das zum Zeigen gut wäre, aber es könnte das Übersetzungsproblem beim Zeichnen lindern oder sogar lösen.

*Pi:* Ja das könnte schon funktionieren. Es ist natürlich fraglich, was da wieder für ein Overhead zusammenkommt. Einerseits ist es sicherlich viel Rechenaufwand, mehrere Kamerabilder live, halbtransparent übereinander zu legen. Andererseits muss man auch schauen, wie viel Clutter da am Screen entsteht und ob das ganze dann noch übersichtlich ist. Fraglich. Müsste man testen. Die Hände kommen ja höchstwahrscheinlich auch alle von der selben Seite, sprich rechts unten und dann liegen alle genau übereinander. Ich weiß nicht.

*Clemens:* Ja, wiederum fraglich wie viele Personen dann wirklich gleichzeitig zeichnen.

*Pi:* Stimmt.

*Clemens:* Gut, das wars dann eigentlich. Danke, dass ihr euch für uns Zeit genommen habt.

*Pi, Zed:* Gerne.

*Thomas:* Danke.



## LITERATURVERZEICHNIS

---

- Accot, J. and Zhai, S. More than dotting the i's — foundations for crossing-based interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, CHI '02, pages 73–80, New York, NY, USA, 2002. ACM.
- Adler, A., Gujar, A., Harrison, B. L., O'Hara, K., and Sellen, A. J. A diary study of work-related reading: design implications for digital reading devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '98, pages 241–248, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.
- Agre, P. E. Conceptions of the User in Computer Systems Design. In *The social and interactional dimensions of human-computer interfaces*. Cambridge University Press, 1995.
- Akin, O. How do architects design? *Artificial Intelligence and Pattern Recognition in Computer Aided Design*, pages 65–103, 1978.
- Alvorado, C. Sketch recognition user interfaces: Guidelines for design and development. In *Proceedings of AAAI Fall Symposium on Intelligent Pen-based Interfaces*, 2004.
- Anthony, L., Yang, J., and Koedinger, K. R. Evaluation of multimodal input for entering mathematical equations on the computer. In *CHI '05 extended abstracts on Human factors in computing systems*, CHI '05, pages 1184–1187, New York, NY, USA, 2005. ACM.
- Apitz, G. and Guimbretiere, F. CrossY: a crossing-based drawing application. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, UIST '04, pages 3–12, New York, NY, USA, 2004. ACM.
- Autodesk. Autodesk Maya, 2010. URL <http://autodesk.com/>.
- Bae, S., Balakrishnan, R., and Singh, K. ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST '08, pages 151–160, New York, NY, USA, 2008. ACM.
- Ballay, J. M. An experimental view of the design process. *System Design: Behavioral Perspectives on Designers, Tools, and Organizations*, pages 65–82, 1987.

- Bannon, L. and Schmidt, K. CSCW: four characters in search of a context. *Studies in computer supported cooperative work*, 1990.
- Bardram, J. E. and Bossen, C. A web of coordinative artifacts: collaborative work at a hospital ward. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, GROUP '05, pages 168–176, New York, NY, USA, 2005. ACM.
- Barfield, L. *The user interface: concepts & design*. Human computer interaction. Addison-Wesley, 1993.
- Bastéa-Forte, M. and Yen, C. Encouraging contribution to shared sketches in brainstorming meetings. In *CHI '07 extended abstracts on Human factors in computing systems*, CHI '07, pages 2267–2272, New York, NY, USA, 2007. ACM.
- Baum, L. F. *The Wonderful Wizard of Oz: by L. Frank Baum*. MobileReference.com, 2008.
- Bentley, R., Hughes, J. A., Randall, D., Rodden, T., Sawyer, P., Shapiro, D., and Sommerville, I. Ethnographically-informed systems design for air traffic control. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work, CSCW '92*, pages 123–129, New York, NY, USA, 1992. ACM.
- Benyon, D., Turner, P., and Turner, S. *Designing interactive systems: people, activities, contexts, technologies*. Addison-Wesley, 2005.
- Bly, S. A. A use of drawing surfaces in different collaborative settings. In *Proceedings of the 1988 ACM conference on Computer-supported cooperative work, CSCW '88*, pages 250–256, New York, NY, USA, 1988. ACM.
- Bødker, S. *Through the interface: a human activity approach to user interface design*. L. Erlbaum, 1991.
- Bonsiepe, G. Die sieben Säulen des Design. Design braucht keine Manifeste, sondern Fundamente. *form+zweck*, 6, 1992.
- Borchers, J. O. A pattern approach to interaction design. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, DIS '00, pages 369–378, New York, NY, USA, 2000. ACM.
- Braa, K. and Ruvik, E. Edb-støtte til kreativt samarbeid i filmproduksjon [Computer-Support for Creative Cooperation in Film Production], 1989.
- Brereton, M. and McGarry, B. An observational study of how objects support engineering design thinking and communication: implications for the design of tangible media. In *Proceedings*

- of the SIGCHI conference on Human factors in computing systems, CHI '00, pages 217–224, New York, NY, USA, 2000. ACM.
- Brinck, T. and Gomez, L. M. A collaborative medium for the support of conversational props. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work, CSCW '92*, pages 171–178, New York, NY, USA, 1992. ACM.
- Brown, J. S. and Duguid, P. *The Social Life of Information*. Harvard Business School Press, 2000.
- Bucciarelli, L. L. *Designing engineers*. Inside technology. MIT Press, 1994.
- Bucciarelli, L. L. Between thought and object in engineering design. *Design Studies*, 23(3):219 – 231, 2002.
- Buschmann, F. *Pattern-orientierte Software-Architektur: ein Pattern-System*. Addison-Wesley, 1998.
- Butler, K., Bennet, J., and Whiteside, J. Engineering objectives for usability, 1987.
- Buxton, B. *Sketching user experiences: getting the design right and the right design*. Interactive Technologies. Elsevier/Morgan Kaufmann, 2007.
- Carroll, J. M. *Scenario-based design: envisioning work and technology in system development*, chapter The Scenario Perspective on System Development, pages 2–6. Wiley, 1995.
- Casasanto, D. and Lozano, S. Metaphor in the Mind and Hands. In *Proceedings of 28th Annual Conference of the Cognitive Science Society*, pages 142–147. Lawrence Erlbaum Associates, 2006.
- Clark, H. H. *Using Language*. Cambridge University Press, 1996.
- Clark, H. H. and Brennan, S. E. Grounding in Communication. In Baecker, R. M., editor, *Readings in Groupware and Computer-Supported Cooperative Work: Assisting Human-Human Collaboration*, pages 222–233. Morgan Kaufmann, 1991.
- Crampton, C. MUSK - a Multi-User Sketch Program. *Proceedings of the European UNIX systems User Group*, pages 17–29, 1987.
- Cross, N. and Clayburn Cross, A. Observations of teamwork and social processes in design. *Design Studies*, pages 1–28, 1995.
- Dewan, P. and Choudhard, R. Flexible user interface coupling in a collaborative system. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology, CHI '91*, pages 41–48, New York, NY, USA, 1991. ACM.

- Dillenbourg, P., Baker, M. J., Blaye, A., and O'Malley, C. The evolution of research on collaborative learning. *CHI '07 Proceedings - Programming by Professionals*, 1995.
- Dix, A., Finlay, J., and Abowd, G. D. *Human-computer interaction*. Pearson/Prentice Hall, 2004.
- Dourish, P. and Bellotti, V. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work, CSCW '92*, pages 107–114, New York, NY, USA, 1992. ACM.
- Dumas, J. S. and Redish, J. *A practical guide to usability testing*. Lives of Great Explorers Series. Intellect, 1999.
- Eason, K. *Information technology and organisational change*. Taylor & Francis/Hemisphere, Bristol, PA, USA, 1988.
- Electronic Arts. The Sims, 2010. URL <http://thesims.ea.com/>.
- Ellis, T. O., Heafner, J. F., and Sibley, W. L. *The GRAIL Project: An experiment in man-machine communications*. Number RM-5999-ARPA. The RAND Corporation, 1969.
- Erdmann, R. L. and Neal, A. S. Laboratory vs. Field Experimentation in Human Factors: An Evaluation of an Experimental Self-Service Airline Ticket Vendor. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 13:521–531, 1971.
- Erickson, T. Design as storytelling. *interactions*, 3:30–35, 1996.
- Fischer, G. Distances and Diversity: Sources for Social Creativity. *C&C 05*, pages 1–9, 2005.
- Forbus, K. D. Exploring spatial cognition through sketch understanding, 2008. URL <http://conference.spatial-cognition.de/sc08/tutorials/T-1>.
- Gamma, E., Helm, R., Johnson, R., and Vlissides, J. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.
- Gedenryd, H. *How designers work - making sense of authentic cognitive activities [Elektronisk resurs]*. Cognitive Science, 1998.
- Geißler, J. Gedrics: The next generation of icons. In *Proceedings of the 5th International Conference on Human-Computer Interaction, INTERACT'95*, 1995.
- Gerlicher, A. Computer-Supported Cooperative Work (CSCW) — kollaborative Systeme und Anwendungen. *Kompendium Medieninformatik*, pages 143–195, 2007.

- Goel, V. *Sketches of thought*. Bradford Books. MIT Press, 1995.
- Goldin-Meadow, S. *Hearing gesture: how our hands help us think*. The Belknap Press of Harvard University Press, 2003.
- Goodman, N. *Languages of art: an approach to a theory of symbols*. Bobbs-Merrill, 1968.
- Greenberg, S. and Bohnet, R. GroupSketch: A multi-user sketch-pad for geographically-distributed small groups. In *Proceedings of Graphics Interface '91*, pages 5–7, 1991.
- Greenberg, S., Roseman, M., Webster, D., and Bohnet, R. Human and technical factors of distributed group drawing tools. *Interacting with Computers*, 4(3):364–392, 1992a.
- Greenberg, S., Roseman, M., Webster, D., and Bohnet, R. Issues and Experiences Designing and Implementing Two Group Drawing Tools. *Readings in Computer Supported Cooperative Work*, pages 1–13, 1992b.
- Greenberg, S. and Buxton, B. Usability evaluation considered harmful (some of the time). *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, 2008.
- Grudin, J. Why CSCW applications fail: problems in the design and evaluation of organizational interfaces. *CSCW '88: Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, 1988.
- Grudin, J. CSCW: Its History and Participation, 1994. URL <http://research.microsoft.com/en-us/um/people/jgrudin/past/Papers/IEEE94/IEEEComplastsub.html>.
- Gutwin, C. and Greenberg, S. The effects of workspace awareness support on the usability of real-time distributed groupware. *Transactions on Computer-Human Interaction (TOCHI)*, 6(3), 1999.
- Hailpern, J., Hinterbichler, E., Leppert, C., Cook, D., and Bailey, B. TEAM STORM: demonstrating an interaction model for working with multiple ideas during creative group work. *C&C 07: Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition*, 2007.
- Halasz, F. G., Moran, T. P., and Trigg, R. H. Notecards in a nutshell. In *Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface*, CHI '87, pages 45–52, New York, NY, USA, 1987. ACM.

- Han, J. Y. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology, UIST '05*, pages 115–118, New York, NY, USA, 2005. ACM.
- Harrison, S. and Minneman, S. A Bike In Hand. In Cross, N., Christians, H., and Dorst, K., editors, *Analysing Design Activity*, pages 417–436. John Wiley & Sons, 1996.
- Heath, C. and Luff, P. Collaboration and control: Crisis management and multimedia technology in London Underground Line Control Rooms. *Computer Supported Cooperative Work (CSCW)*, 1:69–94, 1992.
- Hegarty, M. Mental animation: Inferring motion from static displays of mechanical systems. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, page 1102, 1992.
- Heiser, J. and Tversky, B. Descriptions and depictions of complex systems: Structural and functional perspectives. 2002.
- Henzen, A., Ailenei, N., Fiore, F. D., Van Reeth, F., and Patterson, J. Sketching with a low-latency electronic ink drawing tablet. In *Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia, GRAPHITE '05*, pages 51–60, New York, NY, USA, 2005. ACM.
- Herrmann, T., Hoffmann, M., Jahnke, I., Kienle, A., Kunau, G., Loser, K. U., and Menold, N. Concepts for usable patterns of groupware applications. *GROUP '03: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*, 2003.
- Heskett, J. *Design - A Very Short Introduction*. Oxford University Press, 2005.
- Hill, A., Song, S., Dong, A., and Agogino, A. M. Identifying Shared Understanding in Design Teams Using Document Analysis. In *Proceedings of the 13th International Conference on Design Theory and Methodology, DETC2001/DTM-21713*, 2001.
- Hinckley, K. Input technologies and techniques. In Jacko, J. A. and Sears, A., editors, *The human-computer interaction handbook*, pages 151–168. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 2003.
- Hinckley, K., Baudisch, P., Ramos, G., and Guimbretiere, F. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '05*, pages 451–460, New York, NY, USA, 2005. ACM.

- Hinckley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H., and Buxton, B. Manual deskterity: an exploration of simultaneous pen + touch direct input. In *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems, CHI EA '10*, pages 2793–2802, New York, NY, USA, 2010. ACM.
- Hogg, M. A. and Abrams, D. *Social Identifications: a Social Psychology of Intergroup Relations and Group Processes*. Routledge, 1988.
- Hornecker, E., Marshall, P., Dalton, N., and Rogers, Y. Collaboration and interference: awareness with mice or touch input. *CSCW '08: Proceedings of the 2008 ACM conference on Computer supported cooperative work*, 2008.
- Hutchins, E. *Cognition in the wild*. MIT Press, 1995.
- IBM Corporation. Optimistisches Sperren, 1996. URL <http://publib.boulder.ibm.com/infocenter/wchelp/v6r0m0/index.jsp?topic=/com.ibm.commerce.admin.doc/concepts/cpmoptlock.htm>.
- Igarashi, T. and Hughes, J. F. A suggestive interface for 3D drawing. In *Proceedings of the 14th annual ACM symposium on User interface software and technology, UIST '01*, pages 173–181, New York, NY, USA, 2001. ACM.
- Igarashi, T. and Hughes, J. F. Smooth meshes for sketch-based freeform modeling. In *Proceedings of the 2003 symposium on Interactive 3D graphics, I3D '03*, pages 139–142, New York, NY, USA, 2003. ACM.
- Irving, J. L. *Groupthink: Psychological Studies of Policy Decisions and Fiascoes*. Cengage Learning, 1982.
- Ishii, H., Kobayashi, M., and Arita, K. Iterative design of seamless collaboration media. *Communications of the ACM*, 37(8), 1994.
- Johnson, G., Gross, M. D., and Do, E. Y. Flow selection: a time-based selection and operation technique for sketching tools. In *Proceedings of the working conference on Advanced visual interfaces, AVI '06*, pages 83–86, New York, NY, USA, 2006. ACM.
- Johnson, G., Gross, M. D., Hong, J., and Yi-Luen Do, E. Computational Support for Sketching in Design: A Review. *Found. Trends Hum.-Comput. Interact.*, 2:1–93, 2009.
- Johnson, M. *The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason*. University of Chicago Press, 1987.

- Kidd, A. The marks are on the knowledge worker. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence, CHI '94*, pages 186–191, New York, NY, USA, 1994. ACM.
- Kilker, J. Conflict on Collaborative Design Teams: Understanding the Role of Social Identities. *IEEE Technology and Society Magazine*, pages 1–10, 1999.
- Knuth, D. E. Computer Programming as an Art. *Communications of the ACM*, 17(12):667–673, December 1974.
- Koch, M. CSCW and Enterprise 2.0 - towards an integrated perspective. In *Proc. Conf. Bled eConference eCollaboration*, pages 416–427, 2008.
- Kosslyn, S. M. *Image and mind*. Harvard University Press, 1980.
- Kristoffersen, S. and Ljungberg, F. An empirical study of how people establish interaction: implications for CSCW session management models. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit, CHI '99*, pages 1–8, New York, NY, USA, 1999. ACM.
- Kurtenbach, G. and Buxton, W. Issues in combining marking and direct manipulation techniques. In *Proceedings of the 4th annual ACM symposium on User interface software and technology, UIST '91*, pages 137–144, New York, NY, USA, 1991. ACM.
- Kvan, T. Collaborative design: what is it? *Automation in Construction*, pages 1–7, 2000.
- Larkin, J. H. and Simon, H. A. Why a Diagram is (Sometimes) Worth Ten Thousand Words. *Cognitive Science*, 11(1):65 – 100, 1987.
- Larsson, A. Making sense of collaboration: the challenge of thinking together in global design teams. In *Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work, GROUP '03*, pages 153–160, New York, NY, USA, 2003. ACM.
- Lawson, B. *How designers think: the design process demystified*. Architectural Press, 1997. ISBN 9780750630733.
- Le Dantec, C. A. Situated design: toward an understanding of design through social creation and cultural cognition. In *Proceeding of the seventh ACM conference on Creativity and cognition, C&C '09*, pages 69–78, New York, NY, USA, 2009. ACM.
- Lee, J. J. Xsketch: a multi-user sketching tool for X11. *SIGOIS Bull.*, 11:169–173, 1990.

- Lera, S. Synopses of some recent published studies of the design process and designer behaviour. *Design Studies*, 4(2):133–140, 1983.
- Li, Y., Hinckley, K., Guan, Z., and Landay, J. A. Experimental analysis of mode switching techniques in pen-based user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '05, pages 461–470, New York, NY, USA, 2005. ACM.
- Lloyd, P. Storytelling and the development of discourse in the engineering design process. *Design Studies*, 21(4):357 – 373, 2000.
- Luff, P., Heath, C., and Greatbatch, D. Tasks-in-interaction: paper and screen based documentation in collaborative activity. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, CSCW '92, pages 163–170, New York, NY, USA, 1992. ACM.
- Mankoff, J. Providing integrated toolkit-level support for ambiguity in recognition-based interfaces. In *CHI '00 extended abstracts on Human factors in computing systems*, CHI '00, pages 77–78, New York, NY, USA, 2000. ACM.
- Mankoff, J., Abowd, G. D., and Hudson, S. E. OOPS: a toolkit supporting mediation techniques for resolving ambiguity in recognition-based interfaces. *Computers & Graphics*, 24(6):819 – 834, 2000.
- Markus, M. and Connolly, T. Why CSCW applications fail: problems in the adoption of interdependent work tools. *CSCW '90: Proceedings of the 1990 ACM conference on Computer-supported cooperative work*, 1990.
- Marx, K. *Das Kapital. Kritik der politischen Ökonomie. Band 1*. Dietz Verlag, 1867.
- Microsoft. Surface, 2010. URL <http://www.microsoft.com/surface/>.
- Miller-Jacobs, H. M. *Rapid prototyping: an effective technique for system development*, pages 273–286. Academic Press Professional, Inc., San Diego, CA, USA, 1991.
- Minneman, S. L. *The Social Construction of a Technical Reality: Empirical Studies of Group Engineering Design Practice*. PhD thesis, Stanford University, CA, USA, 1991.
- Moggridge, B. *Designing interactions*. MIT Press, 2007.
- Mumford, E. A Socio-Technical Approach to Systems Design. *Requirements Engineering*, 5:125–133, 2000.

- Newman, M. W. and Landay, J. A. Sitemaps, storyboards, and specifications: a sketch of Web site design practice. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, DIS '00, pages 263–274, New York, NY, USA, 2000. ACM.
- Nokes, S. *The definitive guide to project management: the fast track to getting the job done on time and on budget*. Financial Times Prentice Hall, 2003. ISBN 9780273663973.
- Norman, D. A. *Things that make us smart: defending human attributes in the age of the machine*. Number Nr. 842 in A William Patrick Book. Addison-Wesley, 1994.
- Nunez, R. E. Could the future taste purple? Reclaiming mind, body and cognition. *Journal of Consciousness Studies*, 6:41–60, 1999.
- Olsen, Jr., D. R., Taufer, T., and Fails, J. A. ScreenCrayons: annotating anything. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, UIST '04, pages 165–174, New York, NY, USA, 2004. ACM.
- Orr, J. E. Narratives at work: story telling as cooperative diagnostic activity. In *Proceedings of the 1986 ACM conference on Computer-supported cooperative work*, CSCW '86, pages 62–72, New York, NY, USA, 1986. ACM.
- Oviatt, S., Arthur, A., and Cohen, J. Quiet interfaces that help students think. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, UIST '06, pages 191–200, New York, NY, USA, 2006. ACM.
- Pahl, G. and Beitz, W. *Engineering Design: A Systematic Approach*. The Design Council, 1988.
- Pan, T. Y., Newman, R. M., Porter, S., and Tovey, M. Verbal language and Sketching. In *Common Ground: Proceedings of the Design Research Society International Conference at Brunel University*. Staffordshire University Press, 2002.
- Panko, R. R. Office Work. *Information Technology & People*, 2(3): 205–238, 1993.
- Perry, M. and Sanderson, D. Coordinating joint design work: the role of communication and artefacts. *Design Studies*, 19(3):273 – 288, 1998.
- Piper, A. M. and Hollan, J. D. Tabletop displays for small group study: affordances of paper and digital materials. In *Proceedings of the 27th international conference on Human factors in computing*

- systems*, CHI '09, pages 1227–1236, New York, NY, USA, 2009. ACM.
- Plimmer, B. Experiences with digital pen, keyboard and mouse usability. *Journal on Multimodal User Interfaces*, 2:13–23, 2008.
- Prante, T., Magerkurth, C., and Streit, N. Developing CSCW tools for idea finding -: empirical results and implications for design. *CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 1–10, 2002.
- Preece, J., Rogers, Y., Sharp, Benyon, Holland, and Carey. *Human-Computer Interaction*. Ics Series. Addison-Wesley Pub. Co., 1994.
- Price, S. Animated Diagrams: How Effective are Explicit Dynamics for Learners? In *Proceedings of ICLS 2002*, pages 344–351, 2002.
- Rama, J. and Bishop, J. A survey and comparison of CSCW groupware applications. *SAICSIT '06: Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, 2006.
- Ramos, G., Boulos, M., and Balakrishnan, R. Pressure widgets. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, pages 487–494, New York, NY, USA, 2004. ACM.
- Randall, D., Harper, R., and Rouncefield, M. *Fieldwork for design: theory and practice*. Computer supported cooperative work. Springer-Verlag, 2007.
- Robinson, M. Design for unanticipated use... In *Proceedings of the third conference on European Conference on Computer-Supported Cooperative Work*, pages 187–202, Norwell, MA, USA, 1993. Kluwer Academic Publishers.
- Saffer, D. *Designing for interaction: creating innovative applications and devices*. Voices that matter. New Riders, 2007.
- Sagmeister, C. Design. Bachelor's thesis, Institute for Design and Assessment of Technology, Vienna University of Technology, Vienna, Austria, 2008.
- Saund, E. and Lank, E. Stylus input and editing without prior selection of mode. *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 1–4, 2003.
- Schelling, T. C. *Micromotives & Macrobehavior*. W. W. Norton, 1987.

- Schilit, B. N., Golovchinsky, G., and Price, M. N. Beyond paper: supporting active reading with free form digital ink annotations. In *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '98*, pages 249–256, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.
- Schmidt, K. and Wagner, I. Coordinative artefacts in architectural practice. In M., B.-F., A.M., P.-D., K., S., and P., Z., editors, *Proceedings of the Fifth International Conference on the Design of Co-operative Systems, COOP 2002*, pages 257–274. IOS Press, 2002.
- Schneider, B., Schmid, J., and Christen, D. *Design - eine Einführung: Entwurf im sozialen, kulturellen und wirtschaftlichen Kontext*. Birkhäuser, 2008.
- Schäfer, E. *Nonverbale Kommunikation*, 2003.
- Schön, D. A. *The reflective practitioner: how professionals think in action*. Basic Books, 1983.
- Sellen, A. J. and Harper, R. H. R. *The Myth of the Paperless Office*. MIT Press, Cambridge, MA, USA, 2003.
- Seybold, A. The DOCIT. *Andrew Seybold's Outlook on Professional Computing*, 11(2), 1992.
- Shapiro, D., Hughes, J. A., Randall, D., and Harper, R. Visual representation of database information: The flight data strip in air traffic control. In Tauber, M., Mahling, D., and Arefi, F., editors, *Cognitive Aspects of Visual Languages and Visual Interfaces*. Elsevier Science, 1994.
- Sharp, H., Rogers, Y., and Preece, J., editors. *Interaction Design: Beyond Human-Computer Interaction*. John Wiley, 2002.
- Sherif, M. *In Common Predicament: Social Psychology of Intergroup Conflict and Cooperation*. Houghton Mifflin, 1966.
- Short, J., Williams, E., and Christie, B. Visual Communication and Social Interaction. In Baecker, R. M., editor, *Readings in Groupware and Computer-Supported Cooperative Work: Assisting Human-Human Collaboration*, pages 153–164. Morgan Kaufmann, 1991.
- Snyder, C. *Paper prototyping: the fast and easy way to design and refine user interfaces*. The Morgan Kaufmann series in interactive technologies. Morgan Kaufmann, 2003.
- Star, S. L. and Griesemer, J. R. Institutional Ecology, 'Translations' and Boundary Objects. *Social Studies Of Science*, 19(3):387–420, 1989.

- Steiner, I. D. *Group process and productivity*. Social psychology. Academic Press, 1972.
- Stewart, J. E. Single display groupware. In *CHI '97 extended abstracts on Human factors in computing systems: looking to the future*, CHI '97, pages 71–72, New York, NY, USA, 1997. ACM.
- Stewart, J. E., Bederson, B. B., and Druin, A. Single display groupware: a model for co-present collaboration. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, CHI '99, pages 286–293, New York, NY, USA, 1999. ACM.
- Streitz, N., Rexroth, P., and Holmer, T. Anforderungen an interaktive Kooperationslandschaften für kreatives Arbeiten und erste Realisierungen. *Proc. of D-CSCW'98*, pages 237–250, 1998.
- Streitz, N., Tandler, P., Müller-Tomfelde, C., and Konomi, S. Roomware: Toward the Next Generation of Human-Computer: Interaction based on an Integrated Design of Real and Virtual Worlds. *Human-Computer Interaction in the New Millenium*, Addison Wesley, pages 551–576, 2001.
- Streitz, N., Prante, T., Müller-Tomfelde, C., Tandler, P., and Magerkurth, C. Roomware©: the second generation. *CHI '02 extended abstracts on Human factors in computing systems*, pages 506–507, 2002.
- Suchman, L. Making work visible. *Communications of the ACM*, 38(9), 1995.
- Sutherland, I. E. Sketch pad a man-machine graphical communication system. In *Proceedings of the SHARE design automation workshop*, DAC '64, pages 6.329–6.346, New York, NY, USA, 1964. ACM.
- Tandler, P. Architecture of beach: The software infrastructure for roomware environments. *CSCW 2000: Workshop on Shared Environments to Support Face-to-Face Collaboration*, 2000.
- Tandler, P., Streitz, N., and Prante, T. Roomware-moving toward ubiquitous computers. *Micro, IEEE*, 22(6):36–47, 2002.
- Tang, J. C. Findings from observational studies of collaborative work. *International Journal of Man-Machine Studies*, 34(2):143 – 160, 1991.
- Tang, J. C. Listing, Drawing, and Gesturing in Design: A Study of the Use of Shared Workspaces by Design Teams. Xerox PARC Research Report SSL-89-3, Department of Mechanical Engineering, 1989.

- Tang, J. C. and Leifer, L. J. A framework for understanding the workspace activity of design teams. *Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, pages 244–249, 1988.
- Tang, J. C. and Minneman, S. Videodraw: a video interface for collaborative drawing. *Transactions on Information Systems (TOIS)*, 9(2):1–15, 1991.
- Taylor, H. A. and Tversky, B. Descriptions and depictions of environments. *Memory and Cognition*, 20(5):483–496, 1992.
- Tesler, L. The smalltalk environment. *Byte Magazine*, 6:90–147, 1981.
- Tohidi, M., Buxton, W., Baecker, R., and Sellen, A. J. Getting the right design and the design right. In *Proceedings of the SIGCHI conference on Human Factors in computing systems, CHI '06*, pages 1243–1252, New York, NY, USA, 2006. ACM.
- Tsang, S., Balakrishnan, R., Singh, K., and Ranjan, A. A suggestive interface for image guided 3D sketching. In *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '04*, pages 591–598, New York, NY, USA, 2004. ACM.
- TU Wien. Beginners' Day, 2010. URL <http://www.informatik.tuwien.ac.at/lehre/services/beginners/>.
- Tuikka, T. and Kuutti, K. Thinking Together in Concept Design for Future Products - Emergent Features for Computer Support. In *Cognitive technology*, Lecture notes in computer science, pages 40–54. Springer, 2001.
- Tversky, B. What do Sketches Say about Thinking? AAI Technical Report SS-02-08, Stanford University, Menlo Park, CA, 2002.
- Tversky, B. and Lee, P. Pictorial and Verbal Tools for Conveying Routes. In Freksa, C. and Mark, D., editors, *Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science*, volume 1661 of *Lecture Notes in Computer Science*, pages 752–752. Springer Berlin / Heidelberg, 1999.
- Tversky, B., Zacks, J., Lee, P., and Heiser, J. Lines, Blobs, Crosses and Arrows: Diagrammatic Communication with Schematic Figures. In Anderson, M., Cheng, P., and Haarslev, V., editors, *Theory and Application of Diagrams*, volume 1889 of *Lecture Notes in Computer Science*, pages 389–408. Springer Berlin / Heidelberg, 2000.
- Ullman, D. G., Stauffer, L. A., and Dietterich, T. G. Toward Expert CAD. *Computers in Mechanical Engineering*, 6(3):56–70, 1987.

- Varela, F. J., Thompson, E., and Rosch, E. *The embodied mind : cognitive science and human experience*. MIT Press, 1991.
- Vertelney, L. and Booker, S. *The Art of Human-Computer Interface Design*, chapter Designing the Whole-Product User Interface. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.
- Vyas, D., Heylen, D., and Nijholt, A. Physicality and Cooperative Design. In *5th Joint Workshop on Machine Learning and Multimodal Interaction*, Lecture notes in computer science, pages 325–337. Springer-Verlag, 2008.
- Wang, H. and Blevis, E. Concepts that support collocated collaborative work inspired by the specific context of industrial designers. *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, 2004.
- Webster, D. E. Mapping the design information representation terrain. *Computer*, 21:8–23, 1988.
- Wellner, P. Interacting with paper on the DigitalDesk. *Commun. ACM*, 36:87–96, 1993.
- Wilson, M. Six views of embodied cognition. *Psychonomic Bulletin & Review*, 9(4):625–636, 2002.
- Wolf, C. G., Rhyne, J. R., and Briggs, L. K. Communication and Information Retrieval with a Pen-based Meeting Support Tool. *CSCW 92 Proceedings*, pages 1–8, 1997.



## INDEX

---

1-9

3-D Navigator, 40

### A

Artefakte, 50

Awareness, 137

### B

Boundary Objects, 52

Bounding Box, 22

### C

ClearBoard, 38

CommChairs, 35

ConnecTable, 36

Conversational Props, 52

ConverTable, 40

CrossY, 113

CSCW, 129

Asynchrone Systeme, 134

Designkriterien, 37, 140

Klassifikation, 132

Multisync. Systeme, 135

Synchrone Systeme, 135

### D

Design, 43

- disziplinen, 46

- kollaboration, 76

- meetings, 86

- methoden, 49

- phase, 44

- theorie, 43

- vokabular, 74

als Prozess, 44

als Repräsentation, 46

Arbeitsfelder, 46

getting it right, 58

DynaWall, 35

### E

Eingabestifte, 109

Electronic Ink, 23, 108

Druckmessung, 109

### F

Flow Selection, 118

### G

Gestik, 119

Begriff, 120

Handgesten, 120

GRAIL, 116

Group Drawing Tools, 15

GroupDraw, 17

GroupKit, 19

GroupSketch, 15

Groupware, 130

Anforderungen, 140

Design, 136

Patterns, 136

Problemfelder, 143

### H

Hintergrundmasken, 65

Human Computer Interaction, 27

### I

Identität, 80

- polarisierende, 82

- soziale, 80

- technologiezentrierte, 82

Idiome, 112

IloveSketch, 114

Indexical Expressions, 76

InteracTable, 36

### K

Kollaboratives Design, 79

Analyse, 97

Informationen, 94

Konflikte, 100

Konzepte, 98

Rollen, 86

Soziale Aspekte, 80

Tätigkeiten, 89

## L

Likert-Skala, 27

## M

Mäuse, 109

Masken, *siehe* Hintergrundmasken

Model-View-Controller, 22

Modellierung, 56

Modus, 116

- problem, 116

- wechseltechniken, 118

More is more, 40

## N

Notizen, 21

## P

Personas, 71

Prototyping, 60

Digital, 61

High-fidelity, 60

Low-fidelity, 57

Papier, 62

Physikalisch, 62

## R

Roomware, 35

## S

Screen crayons, 20

Scribbler, 155

Achievements, 188

at a Glance, 165

Ausgangssituation, 155

GUI, 168

Hardware, 166

Programmlogik, 170

User Review, 175

Vorgehensweise, 159

Sketching for Experience, 56

Sketchpad, 116

Skizzen, 52

am Computer, 108

auf Papier, 105

Eigenschaften, 54

Inhalte, 54

Verwertung, 55

Skizzierhardware, 108

Skizziersysteme

Anforderungen, 156

Designguidelines, 115

Interaktionstechniken, 111

Kognitive Belastung, 111

Sperrmodelle, 10

Storrtelling, 75

Storyboards, 73

Szenarien, 71

## T

Tabletop-Display, 40

Tablets, 108

Team Storm, 31

Testing, 67

Testsysteme, 70

## U

User Experience, 144

User Testing, *siehe* Testing

## V

VideoDraw, 121

## W

Wasserfallmodell, 44

We-Met, 11

WIMP, 111

Wizard of Oz Technique, 68

WYSIWIS, 15

## X

Xerox Star, 21

XGroupSketch, 17

XSketch, 8