

Human-Computer Interaction

Course Notes

Version of 01 Jun 2014



Ao.Univ.-Prof. Dr. Keith Andrews

IICM
Graz University of Technology
Inffeldgasse 16c
A-8010 Graz
kandrews@iicm.edu

Copyright 2014 Keith Andrews.

Contents

Contents	iii
List of Figures	viii
List of Tables	ix
Preface	xi
Credits	xiii
1 Human Computer Interaction	1
2 The Psychology of Usable Things	7
2.1 The Psychopathology of Everyday Things	8
2.2 The Psychology of Everyday Things	17
2.3 The Psychopathology of Computers	34
2.4 Interface Hall of Shame	36
2.5 User-Centered Design	40
3 Usability Engineering	43
3.1 Defining Usability	44
3.2 Usability Evaluation	45
3.3 The Usability Engineering Lifecycle	47
3.4 Planning Usability Activities	51
4 Know the User	53
4.1 Classifying Users	53
4.2 Research the Frames of Reference	55
4.3 Research the End User	56
5 Usability Benchmarking	59
5.1 Competitive Analysis	59
5.2 Set Usability Targets	60
5.3 Return on Investment	60
6 Goal-Oriented Interaction Design	61
6.1 Creating Personas	65
6.2 Video: Illustrating the Persona Design Process	71
6.3 Defining Goals for each Persona	73

6.4	Defining Scenarios for each Persona	73
6.5	Moving to a Design Solution	73
6.6	Getting Ideas from Your Users	75
6.7	Follow Conventions	75
7	Prototyping	81
7.1	Types of Prototype	82
7.2	Verbal Prototype	82
7.3	Low-Fidelity Paper Prototypes	82
7.4	High-Fidelity Paper Prototypes	85
7.5	Interactive Sketches	85
7.6	Working Prototypes	85
7.7	Implementation	87
8	Usability Inspection Methods	89
8.1	Heuristic Evaluation	90
8.2	Severity Ratings	98
8.3	Guideline Checking	99
8.4	Cognitive Walkthrough	101
8.5	Guideline Scoring	105
8.6	Action Analysis	107
9	Usability Testing Methods	109
9.1	Preparing for Usability Testing	112
9.2	Six Stages of Conducting a Test	118
9.3	Thinking Aloud	129
9.4	Co-Discovery	131
9.5	Formal Experiments	132
9.6	A/B Testing	137
9.7	Query Techniques	140
9.8	Usage Studies	144
10	Usability in Practice	147
10.1	Comparison of Evaluation Techniques	147
10.2	Discount Usability Engineering	147
10.3	Differences in Evaluation Practices	149
10.4	Usability Reports	154
10.5	Usability Consulting	154
11	Visual Design and Typography	157
11.1	Typography	157
11.2	Factors Influencing the Legibility of Text	158
12	Icon Design	165
12.1	Visual Association	166
12.2	Standard Parts of an Icon	166
12.3	Icon Design Principles	167
12.4	Cultural and International Issues	171

12.5 Do Not Always Use Icons	173
12.6 Iconic Language	174
12.7 The Icon Design Lifecycle	177
12.8 Designing Icons for Sun's Public Web Site	180
13 A Brief History of HCI	183
13.1 Early Interfaces	184
13.2 WIMP Interfaces	187
Bibliography	189

List of Figures

1.1	The Nature of Human-Computer Interaction	2
2.1	Video Recorder	9
2.2	Video Recorder Remote Control	9
2.3	Zeiss Slide Projector	10
2.4	The control panel in lecture theatre HS EDV	11
2.5	Audiovisual trolley with inputs at rear.	12
2.6	Warning label on audiovisual trolley.	13
2.7	Where is the toilet paper?	13
2.8	Ah, there it is!	14
2.9	Hidden Shower Control	14
2.10	Mop Sink	15
2.11	Mop Sink Label	15
2.12	Car Seat	16
2.13	Ambiguous Door Designs	18
2.14	Good Use of Affordances in Door Designs	18
2.15	Example of Ambiguous Affordances in Door Design	18
2.16	Good Use of Affordances in the Same Hotel	19
2.17	Seemingly Reasonable Door Affordances	19
2.18	Rubbish Bin Affordances	20
2.19	A Label as Big as the Control Panel	20
2.20	Arbitrary mapping of controls to hot plates	22
2.21	Paired cooker controls	22
2.22	A full, natural mapping of cooker controls	22
2.23	Lego Motorbike Kit	23
2.24	Assembled Lego Motorbike	24
2.25	Example of Expected Date Format	26
2.26	Beer Tap Handles	26
2.27	B17 Flying Fortress	27
2.28	B17 Cockpit	28
2.29	B17 Cockpit Control Knobs	28
2.30	B17F Manual	29
2.31	CFR 14 Control Knobs	29
2.32	Fridge Freezer	31
2.33	Fridge freezer controls and instructions	31
2.34	The apparent conceptual model for the fridge freezer	32
2.35	The actual conceptual model for the fridge freezer	32

2.36	Projecting a correct conceptual model	33
2.37	Scissors project a good conceptual model	33
2.38	A digital watch provides no obvious conceptual model	34
2.39	New Keyboard for Windows PCs	36
2.40	Internet Explorer 4.0 Cache Settings Panel	37
2.41	Internet Explorer 4.0 Certificate Authority Selection Panel	38
2.42	A Two-Item List Box in Visual Basic 5.0	38
2.43	A Two Thousand Item List Box	39
2.44	Multi-Row Tab Controls	39
2.45	Win95 Error Box	40
2.46	Macintosh Trash Can	40
2.47	Macintosh Keyboard with Eject Key	41
2.48	Mac OS X Trash Can	41
3.1	Usability Engineering Cartoon	43
3.2	Defining Usability in Context of System Acceptability	45
3.3	Usability Evaluation Methods by Purpose	48
3.4	The Usability Engineering Lifecycle	49
4.1	Categories of user experience	54
4.2	Riding the Learning Curves	55
6.1	Elastic user	65
6.2	Jumble Car	66
6.3	Cars to Match Their Drivers	66
6.4	The InFlight Seat Console	70
6.5	The InFlight Final Design	71
6.6	Users Form a Point Cloud	72
6.7	The Average User and Personas	72
6.8	Primary Personas	73
6.9	Parallel and Iterative Design	74
6.10	Lateral Thinking	75
6.11	Survey on Menu Organisation	76
6.12	Conventional Link Style	77
6.13	Non-Standard and Inconsistent Link Styles	78
6.14	Avis Asterisk Means Optional	79
6.15	Avis Asterisk Means Required	79
7.1	Paper Prototype of IICM on Air	83
7.2	Working Prototype of IICM on Air	83
7.3	Paper Prototype 1	84
7.4	Paper Prototype 2	84
7.5	Paper Prototype 3	85
7.6	An Interactive Sketch	86
7.7	Dimensions of Prototyping	87
8.1	Aggregated evaluations	93
8.2	Sample Banking System Dialogue	95

8.3	Aggregated Evaluations by Evaluator Experience	95
8.4	The Web Usability Checklist	106
9.1	Simple Usability Test Setup	112
9.2	Single Room, Single Camera Test Setup	113
9.3	Example Single Room, Single Camera Test	113
9.4	Alternative Single Room Setup	114
9.5	Observation Room with Electronic Monitoring	114
9.6	Classical Usability lab	115
9.7	Microsoft Usability lab	115
9.8	Morae Test Setup	116
9.9	Portable Usability Kit	117
9.10	Portable Usability Kit 2	117
9.11	Example Internal Task List	120
9.12	Orientation script for Harmony usability test	121
9.13	Background Questionnaire for Harmony Usability Test	122
9.14	Combined nondisclosure and consent form	123
9.15	A Generic Data Collection Form	124
9.16	Completed Data Collection Form	124
9.17	Example Test Checklist	126
9.18	A/B Test	137
10.1	Hotmail password hint question	152
10.2	Hotmail redesigned secret question	152
10.3	Hotmail compose new message	152
11.1	Serif and Sans Serif Fonts	158
11.2	Proportional Versus Fixed Width Fonts	159
11.3	Font Size Changes	159
11.4	All Upper Case Slows Reading	160
11.5	Lower and Mixed Case Words Have Distinctive Shapes	160
11.6	En and Em Word Spacing	160
11.7	Line Spacing	160
11.8	Characters per Line	161
11.9	Flush and Justified Text Styles	162
11.10	Using a Layout Grid	163
11.11	Text Right Up To Margins	164
11.12	Text with Ample Margins	164
12.1	The Standard Parts of an Icon	167
12.2	Visually Imbalanced Icons	168
12.3	Mixed Levels of Realism in Icons	169
12.4	Symbols for Men and Women	169
12.5	Typical Viewing Distances to Icons	170
12.6	Symbol silhouette conveys most information	170
12.7	Garish Multicolour Icons	171
12.8	Well-Balanced, Consistent Icons	172

12.9 Evolution of Microsoft Word icon bar	172
12.10Language-Dependent Text in Icons	172
12.11Culturally-Dependent Mailbox Icons	172
12.12Icons for Food and Drink areas	173
12.13Words for Food and drink Areas	173
12.14The Icon Design Lifecycle	178
12.15Test Setup for Icon Intuitiveness Test	178
12.16An Icon Intuitiveness Test in Progress	179
12.17Room Setup for Icon Test	179
12.18Icon Iterations for “Products and Solutions”	180
12.19Icon Iterations for “Sun on the Net”	182
13.1 Memex Overview	185
13.2 Memex Detail View	185
13.3 SketchPad	186
13.4 Augment/NLS Production Mouse	187

List of Tables

5.1	Survey of usability budgets	60
6.1	Differences Between Computers and Humans	62
6.2	Programmers Think and Behave Differently	63
6.3	Personal and Corporate Goals are Different	64
6.4	Four Main Passenger Personas	68
6.5	Five Main Employee Personas	69
8.1	Heuristic Evaluation Number of Problems Found	92
8.2	Heuristic Evaluation Aggregating Individual Results	93
8.3	Proportion of Evaluators by Experience	96
8.4	One of the 944 guidelines by Smith and Mosier.	99
8.5	Average Times for Typical Keystroke-Level Actions	107
9.1	Simple Coding Scheme for Event Logging	124
10.1	Comparison of Evaluation Techniques	148
12.1	Iconic language for Windows NT 4.0 documents	175
12.2	Iconic language for document and link icons in Harmony	176
12.3	First Round of Icon Designs for “Technology and Developers”	181
12.4	Second Round of Icon Designs for “Technology and Developers”	181
12.5	Third Round of Icon Designs for “Technology and Developers”	181

Preface

I first started teaching human-computer interaction at Graz University of Technology in 1993. These lecture notes have evolved since then, and have benefitted from my teaching courses on user interface design at FH Technikum Kärnten in Villach, web usability and advanced user interfaces at FH Joanneum in Graz, human-computer interaction at FH Hagenberg near Linz and numerous intensive courses at conferences and for industry.

I would like to thank my many tutors for their many helpful ideas and comments over the years. I would also like to thank all my students past and present for their many suggestions and corrections which have helped to massage these notes into their current form.

References in Association with Amazon

References with an ISBN number are linked to amazon.com (or amazon.co.uk or amazon.de) for easy lookup and possible purchase. Amazon pay me a small referral fee for each item you purchase after following such a link – the item itself does not cost you any more. If you find these notes useful and would like to contribute towards their maintenance, please purchase any book you might want *after* following a specific ISBN link from here.

Thanks and happy reading,

Keith

Credits

- The photographs in Figures 2.9, 2.10, 2.11, and 2.12 are used courtesy of Baddesigns.Com (Michael J. Darnell) [Darnell, 2010].
- The photographs in Figures 2.19 and 2.18 are used with kind permission of Karl Voit.
- The material in Section 2.4 is from the Interface Hall of Shame and is used with kind permission from Brian Hayes, Isys Information Architects Inc.
- The screenshot in Figure 6.14 is used with kind permission of Jared Spool [Spool, 2005].
- The paper prototype images of the Northumberland Bank interface in Section 3.3 are used with kind permission from Cliff Brown, Northumbria University.
- The material in Section 10.3 on Comparative Usability Evaluation is used with kind permission from Rolf Molich, DialogDesign, Denmark.
- The material in Section 12.8 on Icon Design is used with kind permission from Jakob Nielsen.
- Figures 13.1 and 13.2 are screen shots from the Memex animation [Adelman and Kahn, 1995], and are used with kind permission of Paul Kahn.
- Figure 13.4 is used with kind permission of Stanford University, Special Collections.

The following figures are used under the terms of the Sun Microsystems Copyright Notice (see page xiii):

- The photograph of Ivan Sutherland's SketchPad system used in Figure 13.3, taken from the sun.com web site (it is no longer there).

Sun Microsystems Copyright Notice

Copyright 1996 Sun Microsystems, Inc. 2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A. All rights reserved. Copyright in this document is owned by Sun Microsystems, Inc. Any person is hereby authorized to view, copy, print, and distribute this document subject to the following conditions:

1. The document may be used for informational purposes only.
2. The document may only be used for non-commercial purposes.
3. Any copy of this document or portion thereof must include this copyright notice.

Chapter 1

Human Computer Interaction

“ Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them. ”

[ACM SIGCHI Curricula for Human-Computer Interaction [Hewett et al., 2002, page 5]]

References

- ++ Cooper, Reimann, and Cronin; *About Face 3: The Essentials of Interaction Design*; Wiley, 2007. ISBN 0470084111 (com, uk) [Cooper, Reimann and Cronin, 2007]
- ++ Alan Cooper; *The Inmates are Running the Asylum*; Sams, 2004. ISBN 0672326140 (com, uk) [Cooper, 2004]
- ++ Steve Krug; *Rocket Surgery Made Easy*; New Riders, 2009. ISBN 0321657292 (com, uk) [Krug, 2009]
- ++ Steve Krug; *Don't Make Me Think!: A Common Sense Approach to Web Usability*; 3rd Edition, New Riders, 2013. ISBN 0321965515 (com, uk) [Krug, 2013]
- + Lazar, Feng, and Hochheiser; *Research Methods in Human-Computer Interaction*; Wiley, 2010. ISBN 0470723378 (com, uk) [Lazar, Feng and Hochheiser, 2010]
- + Kim Goodwin and Alan Cooper; *Designing for the Digital Age: How to Create Human-Centered Products and Services*; Wiley, February 2009. ISBN 0470229101 (com, uk) [Goodwin and Cooper, 2009]
- + Stone, Jarrett, Woodruffe, and Minocha; *User Interface Design and Evaluation*; Morgan Kaufmann, March 2005. ISBN 0120884364 (com, uk) [Stone et al., 2005]
- + Isaacs and Walendowski; *Designing from Both Sides of the Screen*; Sams, 2001. ISBN 0672321513 (com, uk) [Isaacs and Walendowski, 2001]
- o Jef Raskin; *The Humane Interface*; Addison-Wesley, March 2000. ISBN 0201379376 (com, uk) [Raskin, 2000]
- o Ben Shneiderman and Catherine Plaisant; *Designing the User Interface*; 5th Edition, Addison-Wesley, March 2009. ISBN 0321537351 (com, uk) [Shneiderman and Plaisant, 2009]

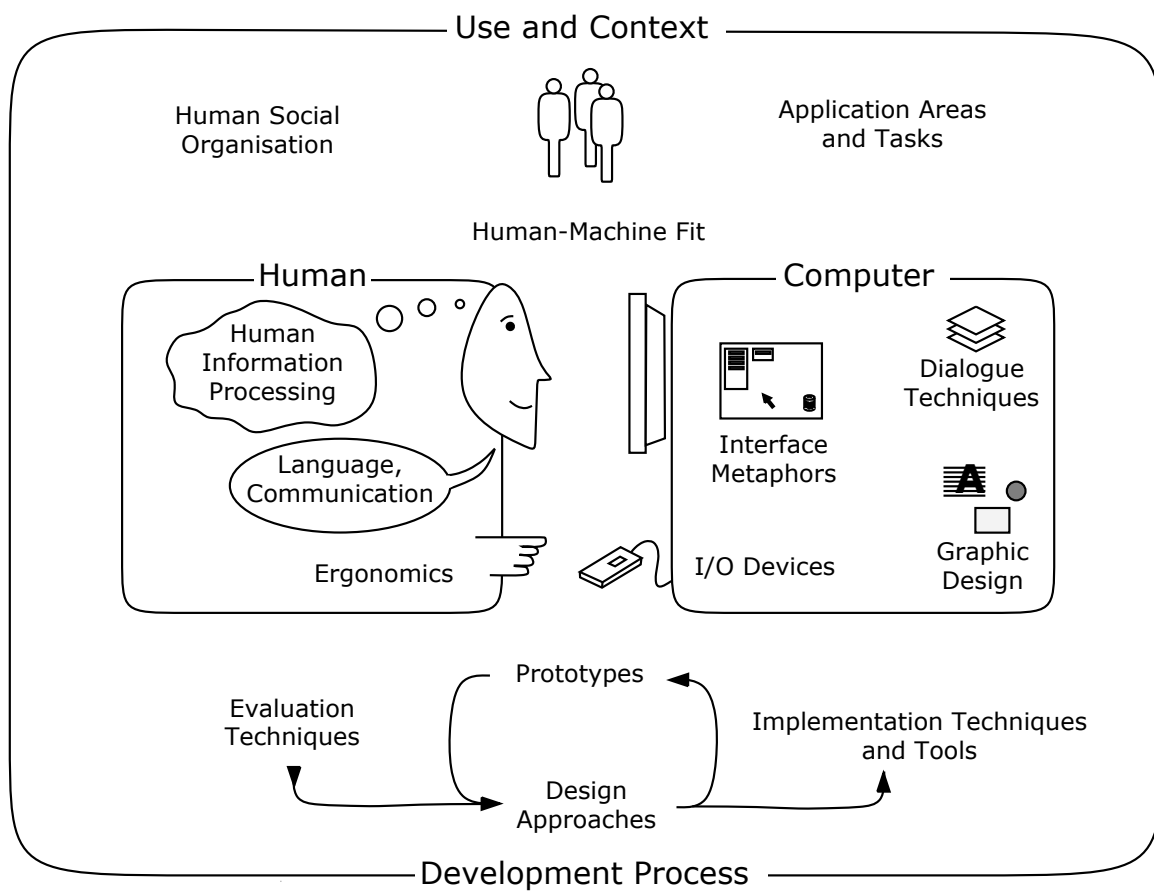


Figure 1.1: The nature of Human-Computer Interaction. Adapted from Figure 1 of the ACM SIG-CHI Curricula for Human-Computer Interaction [Hewett et al., 2002]

- Sharp, Rogers, and Preece; *Interaction Design*; 2nd Edition, Wiley, 2007. ISBN 0470018666 (com, uk)
- Jenny Preece et al; *Human-Computer Interaction*; Addison-Wesley, 1994. ISBN 0201627698 (com, uk) [Preece et al., 1994]
- Terry Winograd (Ed.); *Bringing Design to Software*; Addison-Wesley, April 1996. ISBN 0201854910 (com, uk)
- Helander, Landauer, Prabhu (Eds.); *Handbook of Human-Computer Interaction*; 2nd Edition, Elsevier, 1997. 1602 pages. ISBN 0444818626 (com, uk) [Helander, T. K. Landauer and v Prabhu, 1997]
- Jacko and Sears (Eds.); *The Human-Computer Interaction Handbook*; Lawrence Erlbaum, Sep. 2002. 1277 pages. ISBN 0805844686 (com, uk) [Jacko and Sears, 2002]
- Bruce Tognazzini; *Tog on Interface*; Addison-Wesley, 1992. ISBN 0201608421 (com, uk) [Tognazzini, 1992]
- Bruce Tognazzini; *Tog on Software Design*; Addison-Wesley, 1995. ISBN 0201489171 (com, uk) [Tognazzini, 1995]
- Baecker et al; *Human-Computer Interaction: Toward the Year 2000*; Morgan Kaufmann, 1995. ISBN 1558602461 (com, uk) [Baecker, Grudin et al., 1995]
- Baecker and Buxton; *Readings in Human-Computer Interaction*; Morgan Kaufmann, 1987. ISBN 0934613249 (com, uk) [Baecker and Buxton, 1987]
- John Anderson; *Cognitive Psychology and its Implications*; 6th Edition, Worth, 2004. ISBN 0716701103 (com, uk) [Anderson, 2004]
- Robert Bailey; *Human Performance Engineering*; 3rd Edition, Pearson Education, Jan. 1996. ISBN 0131496344 (com, uk) [Bailey, 1996]
- Wickens et al; *An Introduction to Human Factors Engineering*; 2nd Edition, Prentice Hall, 2003. ISBN 0131837362 (com, uk) [Wickens et al., 2003]
- William Calvin; *How Brains Think*; Basic Books, 1997. ISBN 046507278X (com, uk) [Calvin, 1997]

References in German

- + Steve Krug; *Web Usability: Rocket Surgery Made Easy*; Addison-Wesley, 2010. ISBN 3827329744 (com, uk) [Krug, 2010]
- + Steve Krug; *Don't make me think!: Web Usability: Das intuitive Web*; 2. Auflage, mitp Verlag, 2006. ISBN 3826615956 (com, uk) [Krug, 2006]
- + Martina Manhartsberger und Sabine Musil; *Web Usability - Das Prinzip des Vertrauens*; Galileo Press, 2001. ISBN 3898421872 (com, uk) [Manhartsberger and Musil, 2001]
- Florian Sarodnick and Henning Brau; *Methoden der Usability Evaluation*; Huber, 2006. ISBN 3456842007 (com, uk)

Standards

- + ISO 9241:2008 *Ergonomics of Human-System Interaction*, revised new family of standards. iso.org

See http://en.wikipedia.org/wiki/ISO_9241 and http://userfocus.co.uk/articles/ISO9241_update.html

- ISO 9241:1998 family *Ergonomics requirements for office work with visual display terminals (VDTs)*, 1998. Now obsolete.
- ISO 13407:1999 *Human-centred design processes for interactive systems*, 1999. Replaced by ISO 9241-210:2008.
- ISO 16071:2003 *Guidance on Accessibility for Human-Computer Interfaces*, 2003. Replaced by ISO 9241-171:2008.

Travis [2013] gives an overview of both the old and new ISO 9241 standard families.

Online Resources

- Wikipedia; *Human-Computer Interaction*; http://en.wikipedia.org/wiki/Human-computer_interaction
- ACM SIGCHI; <http://acm.org/sigchi/>
- User Experience Professionals Association; <http://uxpa.org/>
- Society for Technical Communication (STC); *Topics in Usability*; <http://stcsig.org/usability/topics/>
- BCS Interaction Specialist Group; *Usability News*; <http://usabilitynews.bcs.org/>
- Usability Body of Knowledge; <http://usabilitybok.org/>
- Usability.gov; <http://usability.gov/>
- UXmatters; <http://uxmatters.com/>
- IBM; *User-Centered Design*; <http://ibm.com/software/ucd/ucd.html>
- HCI Bibliography; <http://hcibib.org/>
- Newsgroup `news:comp.human-factors`
- ACM Digital Library; <http://acm.org/dl>
[Free access from IP addresses within TU Graz through ACM <http://acm.org/dl/>]
- IEEE Computer Society Digital Library; <http://computer.org/portal/web/csd1/>
[Free access from IP addresses within TU Graz through IEEE Explore <http://ieeexplore.ieee.org/>]

Journals

- *interactions*; ACM; ISSN 1072-5520 interactions.acm.org
- *Transactions on Computer-Human Interaction (TOCHI)*; ACM; ISSN 1073-0516 tochi.acm.org
- *International Journal of Human-Computer Interaction (IJHCI)*; Taylor & Francis; ISSN 1044-7318
<http://tandfonline.com/toc/hihc20/current>

- *International Journal of Human-Computer Studies (IJHCS)*; Elsevier; ISSN 1071-5819 <http://sciencedirect.com/science/journal/10715819>
- *Behaviour & Information Technology (BIT)*; Taylor & Francis; ISSN 0144-929X <http://tandfonline.com/toc/tbit20/current>
- *Journal of Usability Studies (JUS)*; User Experience Professionals Association ISSN 1931-3357 http://upassoc.org/upa_publications/jus/

Conferences

- Conferences sponsored by or in cooperation with ACM SIGCHI <http://sigchi.org/conferences/>
- CHI; Annual SIGCHI Conference: *Human Factors in Computing Systems*; chi2014.acm.org
- AVI; *Advanced Visual Interfaces*; avi2014.deib.polimi.it
- UXPA; *User Experience Professionals Association*; uxpa2014.org
- HCII; *Human-Computer Interaction International* hci-international.org
- South-CHI (formerly Usab); southchi.org

Video: The Front Desk

- Bruce Tognazzini and the BBC; *The Front Desk*; 30-minute video [[BBC, 1996](#)]

Chapter 2

The Psychology of Usable Things

“ When simple things need pictures, labels, or instructions, the design has failed. ”

[Don Norman, The Design of Everday Things, 1988 [Norman, 1992, page 9]]

References

- ++ Donald Norman; *The Design of Everyday Things*; MIT Press, 23 Dec 2013. ISBN (com, uk) [Norman, 2013]
- ++ Donald Norman; *The Design of Everyday Things*; Basic Books, 1992. ISBN 0465067107 (com, uk) [Norman, 1992] [This is a paperback reprint of the original hardbound book *The Psychology of Everyday Things* [Norman, 1988].]
 - Donald Norman; *Turn Signals Are the Facial Expressions of Automobiles*; Addison-Wesley, 1993. ISBN 020162236X (com, uk) [Norman, 1993]
 - Donald Norman; *Things That Make Us Smart*; Addison-Wesley, 1994. ISBN 0201626950 (com, uk) [Norman, 1994]
 - Lakoff and Johnson; *Metaphors We Live By*; Second Edition, University of Chicago Press, 1980. ISBN 0226468011 (com, uk) [Lakoff and M. Johnson, 1980]
 - Leonard Lee; *The Day the Phones Stopped Ringing*; Plume, 1992. ISBN 1556112866 (com, uk) [Lee, 1992] [Out of print]
 - Jeff Johnson; *GUI Bloopers 2.0*; Morgan Kaufman, 2007. ISBN 0123706432 (com, uk) [J. Johnson, 2007]
 - Jeff Johnson; *Web Bloopers*; Morgan Kaufman, 2003. ISBN 1558608400 (com, uk) [J. Johnson, 2003]
 - Vincent Flanders and Michael Willis; *Web Pages That Suck*; Sybex, 1998. ISBN 078212187X (com, uk) [Flanders and Willis, 1998]
 - Vincent Flanders and Dean Peters; *Son of Web Pages That Suck*; Sybex, 2002. ISBN 0782140203 (com, uk) [Flanders and Peters, 2002]

Online Resources

- Michael Darnell; *Bad Human Factors Designs*; <http://baddesigns.com/> [Darnell, 2010]

- Mark Hurst; *This Is Broken*; <http://thisisbroken.com/>
- Wired; *History's Worst Software Bugs*; <http://www.wired.com/software/coolapps/news/2005/11/69355>
- Computer Stupidities; <http://www.rinkworks.com/stupid/>
- Web Pages That Suck; <http://webpagesthatsuck.com/>
- Wikipedia; *Affordance*; <http://en.wikipedia.org/wiki/Affordance>
- Wikipedia; *Natural Mapping*; http://en.wikipedia.org/wiki/Natural_mapping

2.1 The Psychopathology of Everyday Things

Examples of where the design of everyday things went wrong.

Opening a Milk Carton

- Classic example from Austrian TV [ORF, 1987].
- Glass bottles were being replaced by new cartons.
- On live TV, a manager demonstrates how easy it is to open the new cartons. . .
- . . . but everything goes rather wrong!
- The original was broadcast live on the program “wir”, but was later rebroadcast in the outtake show “Hoppala” (hence the laughter over the original soundtrack).

Early Tractors

- Early tractors had a high centre of gravity and narrow wheel base.
- On rough, hilly surface → disaster!
- Used to be called “driver error”.
- More probably “design error”, since tractors today are designed with a low centre of gravity and wide wheel base.

The Frustrations of Everyday Life

Can you use all the functions of your:

- digital watch?
- mobile phone?
- washing machine?
- video recorder?



Figure 2.1: The most basic functionality of a video recorder, playing a tape, is easy to use. However, anything more advanced, such as programming a recording, can become rather difficult.



Figure 2.2: Some of the buttons on a VCR remote control are easy to understand, but others are unfathomable without the instruction manual.

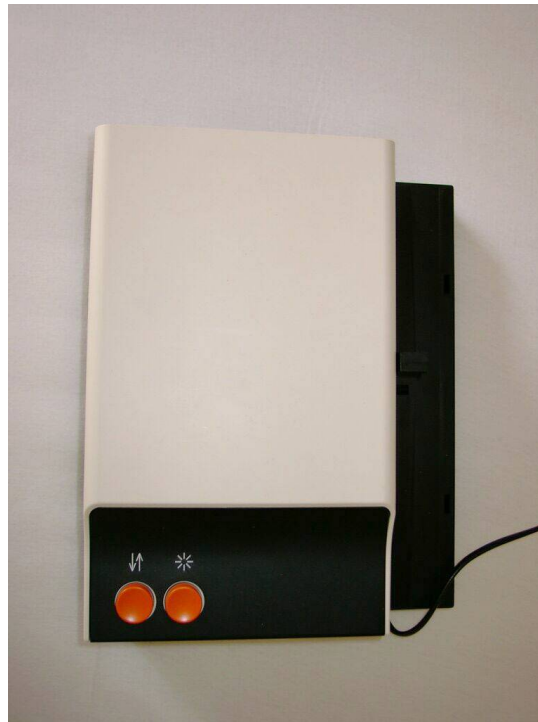


Figure 2.3: The Zeiss Ikon Perkeo 511 slide projector. A short press advances to the next slide, a long press moves back one slide, but how should first-time users know this? [Thanks to Horst Ortmann for providing the photograph.]

Zeiss Slide Projector

- Only one button to control the slide advance, see Figure 2.3.
- During lectures, sometimes the slides go forwards, sometimes they go backwards ...
- If you can find an instruction manual:
Short press = forward, long press = backward.
- What an elegant design, two functions with just one button!
- But how should first-time users know what to do?

The Louis-Laird Amphitheatre in the Sorbonne

- Magnificent murals on the ceiling.
 - But only the right way up for the lecturer.
- Electric projection screen.
 - Has to be lowered from a back room up a short flight of stairs, out of sight.

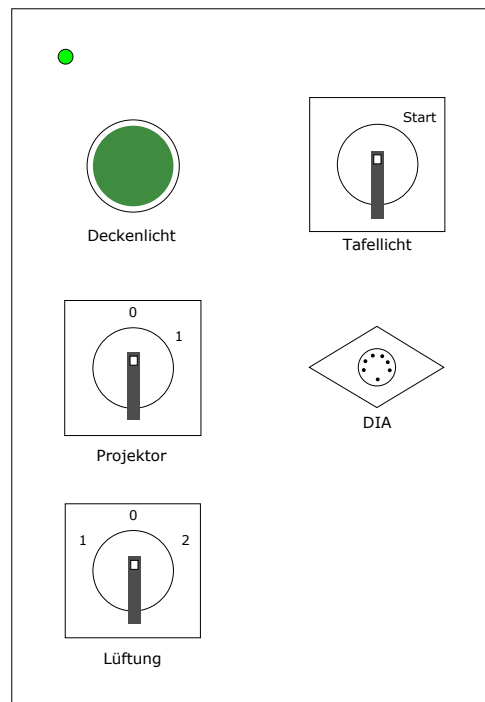


Figure 2.4: The control panel for the lecturer in the old lecture theatre HS EDV of Graz University of Technology.

Could Someone Please Turn the Lights Down

- Figure 2.4 shows the control panel for the lecturer at the front of the old lecture theatre HS EDV.
- I often had to assist guest speakers in turning the lights down (but not completely off).
- The problem is that four (!) separate controls are mapped to the single green button (Deckenlicht):
 - Depressing and releasing the green button either turns the lights completely on or completely off, depending on whether they are currently on or off.
 - Holding the green button down either dims or increases lighting, depending on whether it was last dimmed or increased.
- The air conditioning control (Lüftung) is also problematic: 0 is off, 1 is on, and 2 is off (!).

Clunky Connector

- The audiovisual control trolley (see Figure 2.5) at the front of lecture theatre HS EDV also caused me a major problem.
- Intending to hook up my laptop to the ceiling mounted projector, I unwittingly unscrewed the monitor cable connector from the outside of the trolley.
- This resulted in a dull clunking sound from inside the trolley.
- Unfortunately, the internal connector had only been secured by virtue of its being attached to the external connector!



Figure 2.5: The audiovisual trolley in lecture theatre HS EDV has input connectors at the rear. Unfortunately, unscrewing the external connector causes the internal connector to fall with a clunk somewhere inside the trolley!

- It took three days for a technician to arrive, disassemble the trolley, and reconnect everything.
- To remind myself and to warn others, I resorted to the last ditch technique of providing the label shown in Figure 2.6.

Where is the Toilet Paper?

- Fancy hotel, nice bathrooms, see Figure 2.7.
- Having sat down and done the business, where the heck is the toilet paper?
- Ah, there it is! Well-hidden, see Figure 2.8.

Shower Control

- Shower control: water either goes into the bath out of the faucet or comes out of the shower. See Figure 2.9.
- Sticker with instructions on the faucet.
- How do you make the water come out of the shower instead of the faucet?
- You have to reach *under* the faucet and pull the knob down!
- This example is from Baddesigns.Com [Darnell, 2010] <http://baddesigns.com/shower1.html>

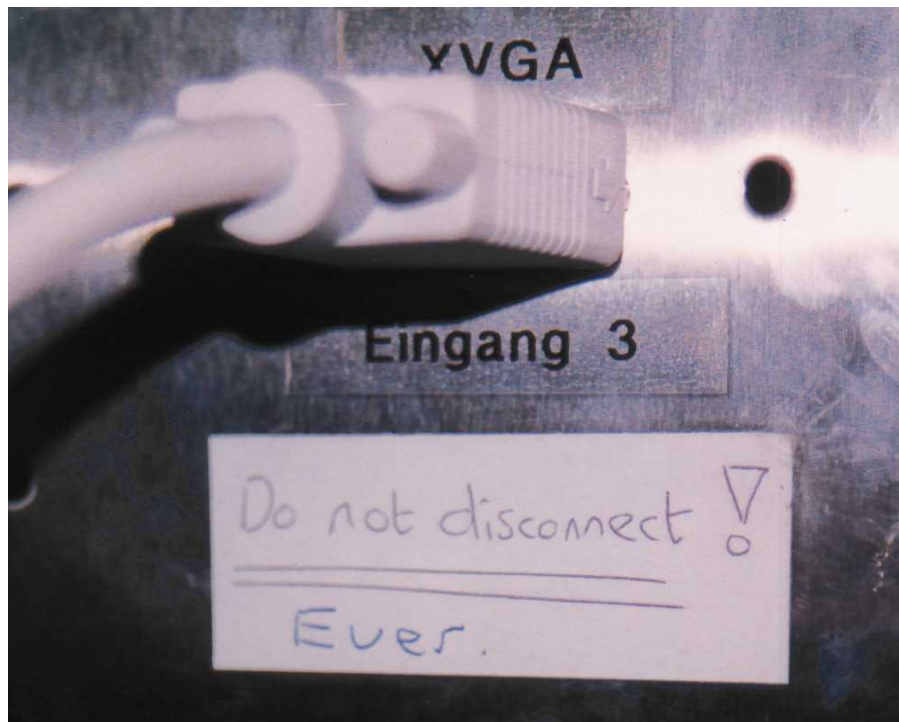


Figure 2.6: I provided this label as a warning to myself and others.



Figure 2.7: Can you see where the toilet paper is in this hotel bathroom?

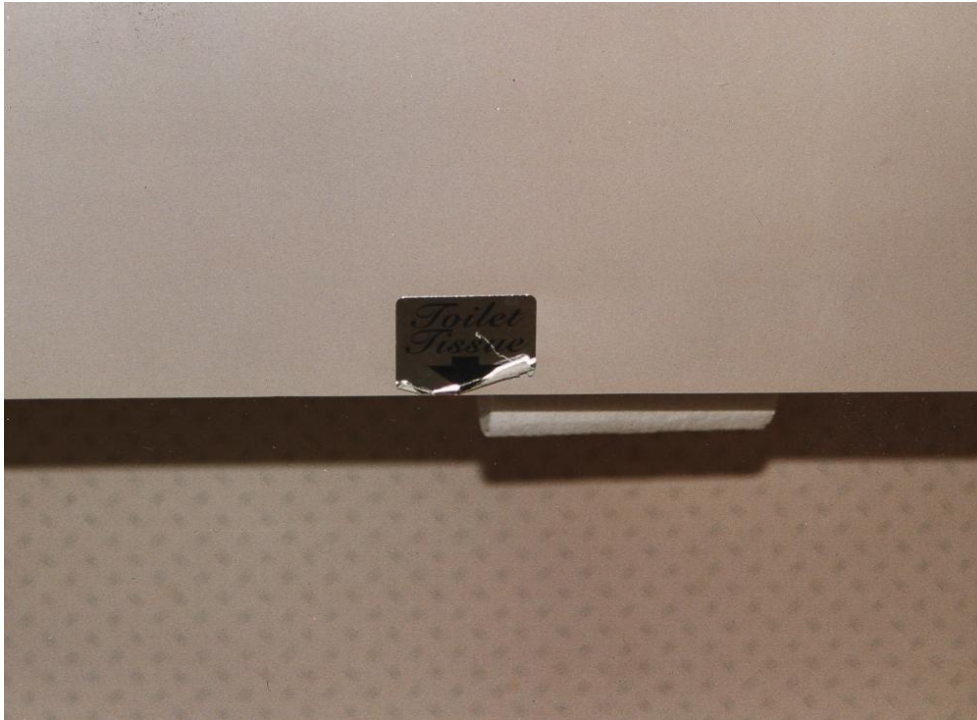


Figure 2.8: The toilet paper is well-hidden under the ledge, and is impossible to find without the sign.



Figure 2.9: Hidden shower control. To divert water to the shower, you have to pull down on the knob beneath the faucet. [Photograph courtesy of Baddesigns.Com [Darnell, 2010].]



Figure 2.10: The men's toilet in a restaurant in Santa Barbara. The fixture in the corner affords a certain activity. It is, however, not a urinal but a mop sink. [Photograph courtesy of Baddesigns.Com [Darnell, 2010].]

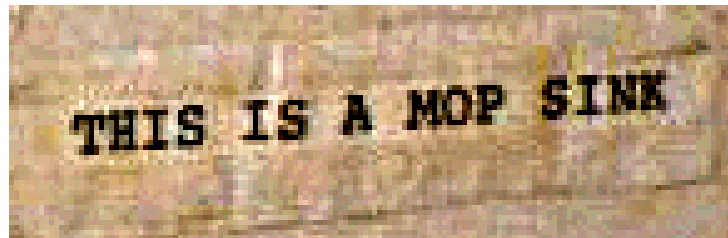


Figure 2.11: A close-up of the label. It reads "This is a Mop Sink". [Photograph courtesy of Baddesigns.Com [Darnell, 2010].]

This is a Mop Sink

- A picture from the men's toilet of a restaurant in Santa Barbara, see Figure 2.10.
- There is no urinal in the toilet.
- Where do you think most men relieve themselves?
- The label says "This is a Mop Sink", see Figure 2.11.
- The mop sink looks enough like a urinal to be used as one.
- This example is from Baddesigns.Com [Darnell, 2010] <http://baddesigns.com/mopsnk.html>



Figure 2.12: The lever beneath this mini-van seat does not work as expected. Instead of allowing the seat to slide backward or forward, pulling the lever detaches the seat from the floor to make room for cargo! [Photograph courtesy of Baddesigns.Com [Darnell, 2010].]

Car Seat

- A seat in a mini-van (people carrier), see Figure 2.12.
- What do you think happens when you pull the lever under the seat?
- Most normal-thinking people would expect the seat to slide backward or forward.
- Not in this mini-van. Pulling the lever detaches the seat from the floor to make room for cargo!
- This example is from Baddesigns.Com [Darnell, 2010] <http://baddesigns.com/carseat.html>

2.2 The Psychology of Everyday Things

Perceived and Real Affordances

Affordances are the range of possible (physical) actions by a user on an artefact:

- *Perceived Affordances* are the actions a user perceives to be possible.
- *Real Affordances* are the actions which are actually possible.

See [Norman, 1999] for a discussion of affordances and perceived affordances.

Real World Affordances

For physical objects, there can be both real and perceived affordances (and the two sets are not necessarily the same).

- Appearance indicates how to use something:
 - A chair affords (suggests) sitting.
 - Knobs are for turning.
 - Slots are for inserting things.
 - A button affords pushing.
- When perceived affordances are taken advantage of, the user knows what to do just by looking.

Figures 2.13 and 2.14 illustrate the perceived affordances of door handles.

Figures 2.18 illustrates affordances of rubbish bin lids.

Labels

- “When simple things need pictures, labels, or instructions, the design has failed!” Norman [1992, page 9]
- See Figure 2.19.

GUI Affordances

For screen-based interfaces, the computer hardware already has built-in physical affordances:

- Screen affords touching.
- Mouse affords pointing.
- Mouse buttons afford clicking.
- Keyboard affords typing.

Changing the shape of the cursor to indicate a clickable link is not an affordance (you can still click anywhere), but visual feedback.

Physically locking the mouse button on non-clickable areas is a real affordance.

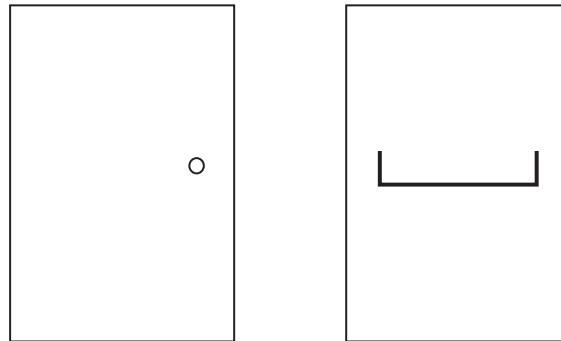


Figure 2.13: Ambiguous door designs. A knob affords turning, but do you push or pull? A horizontal bar affords pushing, but which side do you push on?

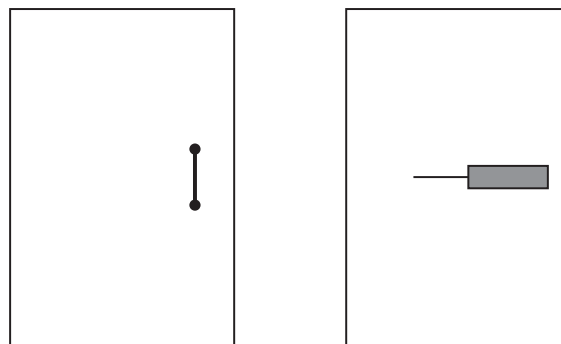


Figure 2.14: Good use of affordances in door designs. A vertical handle affords grasping and pulling. A flat panel affords pushing and the broadness indicates which side to push.



Figure 2.15: An example of ambiguous affordances in door design. The vertical handles mounted on both sides of the door suggest grasping and pulling. Unfortunately, from one side, the door has to be pushed! Note the signs above the handles.



Figure 2.16: Good use of affordances in the same hotel. This door is well designed. The vertical handle correctly suggests pulling, the flat bar correctly suggests pushing.



Figure 2.17: The affordances for this door seem reasonable within themselves. Context is everything. Hopefully, the door is kept locked!



Figure 2.18: Rubbish bin lid affordances. [Photograph taken at Hyatt Regency, Bellevue, WA, in Feb 2012. Used with kind permission of Karl Voit.]

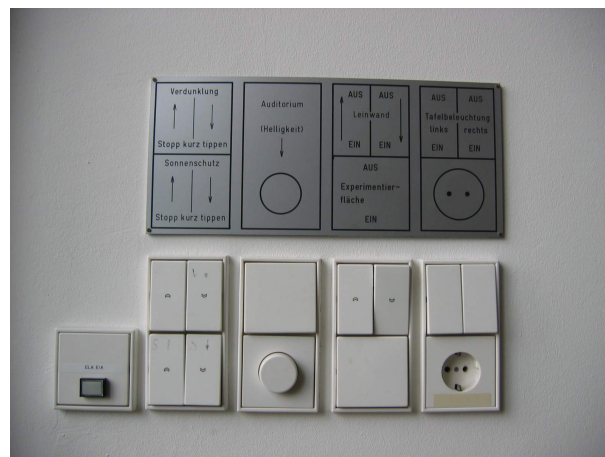


Figure 2.19: A label as big as the control panel. [Photograph taken at TU Chemnitz, Germany in March 2008. Used with kind permission of Karl Voit.]

Mappings

Mappings are the relationships between controls and their effects on a system.

Natural mappings take advantage of physical analogies and cultural standards.

Examples:

- Turn steering wheel clockwise to turn a car right. Actually, there are two mappings here:
 - which control affects steering,
 - which direction to turn it.
- Move a control up to move an object up.
- Use a louder sound to mean a greater amount.

Mapping of Cooker Controls

How should one arrange the hot plate controls on a cooker?

- Arbitrary Mapping (see Figure 2.20).
- Paired Mapping (see Figure 2.21).
- Full Natural Mapping (see Figure 2.22).

Adapted from Norman, *The Design of Everyday Things*, Figures 3.3, 3.4, and 3.5 [Norman, 1988].

Constraints

The difficulty of dealing with a novel situation is directly related to the number of possibilities.

Constraints are physical, semantic, cultural, and logical limits on the number of possibilities.

- *Physical* constraints such as pegs and holes limit possible operations.
- *Semantic* constraints rely upon our knowledge of the situation and of the world.
- *Cultural* constraints rely upon accepted cultural conventions.
- *Logical* constraints exploit logical relationships. For example a natural mapping between the spatial layout of components and their controls.

Where affordances suggest the range of possibilities, constraints limit the number of alternatives.

Constraints in Lego Motorbike

Motorbike toy with 12 parts. Constraints make its construction simple, even for adults!

- *Physical*: Front wheel only fits in one place.
- *Semantic*: The rider sits on the seat facing forward.

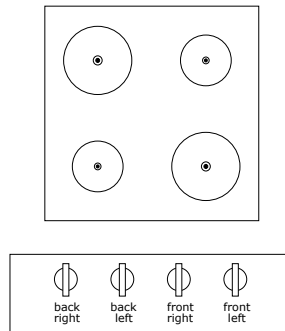


Figure 2.20: Arbitrary mapping of controls to hot plates. There are 24 possible arrangements, requiring the use of labels and memory.

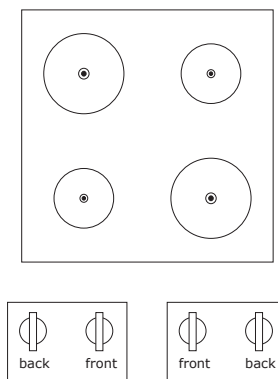


Figure 2.21: Paired cooker controls. Now there are only four possible arrangements, two on each side, but confusion can still occur.

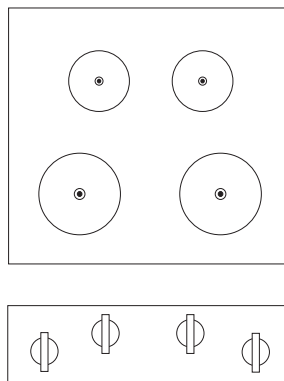


Figure 2.22: A full, natural mapping of cooker controls. There is no ambiguity, no need for learning or remembering, and no need for labels.

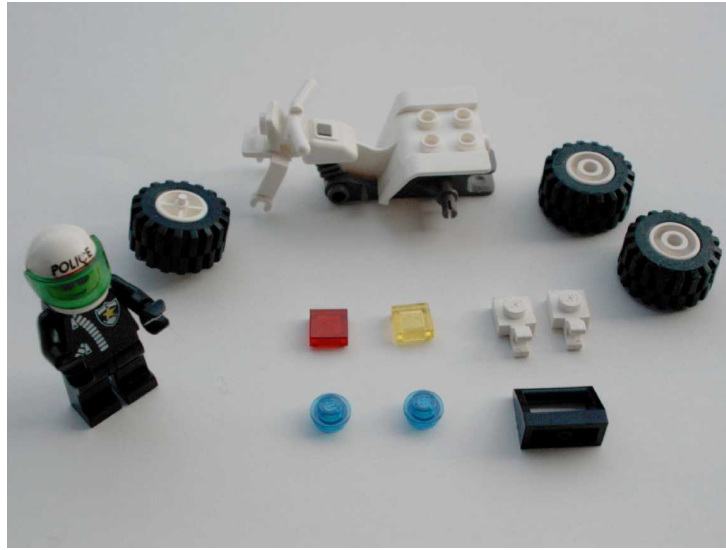


Figure 2.23: The design takes advantage of constraints to make its construction simple.

- *Cultural:* Red is a rear light, yellow a front light.
- *Logical:* Two blue lights, two white pieces, probably go together.

See Figures 2.23 and 2.24.

Conventions

Conventions are cultural constraints. They are initially arbitrary, but evolve and become accepted over time.

They can however still vary enormously across different cultures, for example:

- Light switches:

America	down is off
Britain	down is on
- Water taps:

America	anti-clockwise is on
Britain	anti-clockwise is off
- The colour red:

America	danger
Egypt	death
India	life
China	happiness

The Principle of Causality

Causality is the relation between two events, cause and effect, where the second occurs as a consequence of the first.

Apparent causality is when something which happens immediately after an action, *appears* to have been caused by that action. We associate the effect with the *apparent* cause.

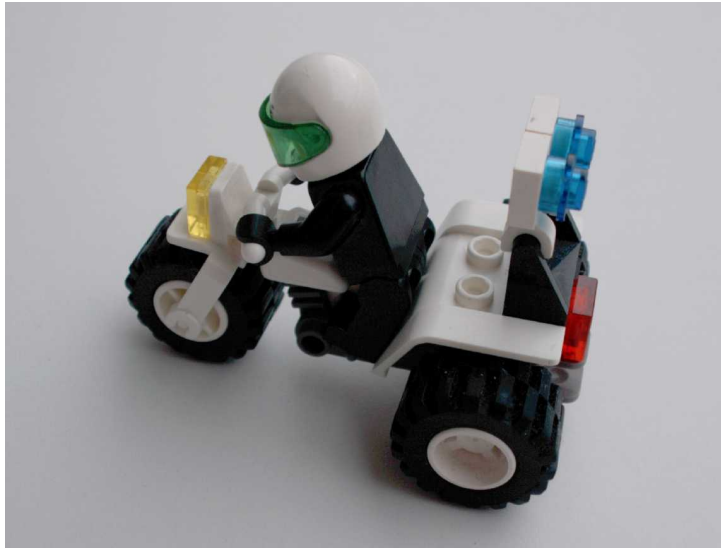


Figure 2.24: The assembled lego motorbike.

There are two kinds of *false* causality:

- Coincidental effects lead to superstition:
 - Touch a computer terminal just before it fails, and you are apt to believe you caused the failure.
 - Start an unfamiliar application, just before the computer crashes.
- Invisible effects lead to confusion:
 - When an action has no apparent result, you may conclude it was ineffective (and repeat it).
 - For example, repeatedly clicking the “Stop” button when the system is unresponsive.

→ There is a need for feedback!

The Structure of Human Memory

Short-Term Memory (STM)

Short-term memory is the memory of the present, used as working or temporary memory.

- Information is retained in STM automatically and is retrieved without effort.
- However, the amount of information in STM is severely limited: 7 ± 2 items [Miller, 1956]
- STM is extremely fragile – the slightest distraction and its contents are gone.

For example, STM can hold a seven digit phone number from the time you look it up until the time you use it, as long as no distractions occur.

Long Term Memory (LTM)

Long-term memory is the memory of the past.

- It takes time to put stuff into LTM and time and effort to get stuff out.
- Capacity is estimated at about 100 million items.

Knowledge in the Head and in the World

Not all of the knowledge required for precise behaviour has to be in the head. It can be distributed:

- partly in the head
- partly in the world
- and partly in the constraints of the world.

Placing Knowledge in the World

Having knowledge in the world reduces the load on human memory:

- An example of the input format can be provided in the interface:

Please enter the date (yyyy/mm/dd):

See Figure 2.25 for an example.



- Previously entered values can be used as defaults, so users do not have to remember items between screens.
- It is better if the designers of an interface place knowledge in the world.
- However, sometimes, users have to place knowledge in the world themselves to fix a broken interface.
- Control-room operators at a nuclear power plant fixed beer-tap handles to similar-looking knobs, so as to better distinguish between them. See Figure 2.26.

Wherever possible, also allow expert users to internalise knowledge for faster and more efficient performance (say by learning to type a date in a particular format, rather than having to use the provided calendar widget).

Shape Coding for Aircraft Control Knobs


- During the first years of World War 2, there were dozens of gear-up accidents when landing B-17, B-25, and P-47 aircraft [Koonce, 2002, page 95].
- Pilots frequently retracted the landing gear instead of the flaps after landing. See Figure 2.27.
- In 1943, Lt. Alphonse Chapanis was ordered to investigate and noticed that C-47 transport (DC3 Dakota) pilots suffered no such accidents [Roscoe, 1997, page 3].

Anmeldung

von  Format: TTMMJJJJ Uhrzeit  [Gruppenliste](#)

kein Anmeldebeginn-Zeitpunkt erlaubt

24.02.2014 - 07.03.2014 von 07:00 bis 09:00
 29.09.2014 - 03.10.2014 von 07:00 bis 09:00
 02.03.2015 - 06.03.2015 von 07:00 bis 09:00

bis  Format: TTMMJJJJ Uhrzeit: 23:59:59

☐ Web ANmeldung nicht erlauben

Figure 2.25: The intranet server at Graz University of Technology, TUGrazOnline, provides an example of the expected date format (TTMMJJJJ) in the interface next to the input field. Users do not have to remember the required format.

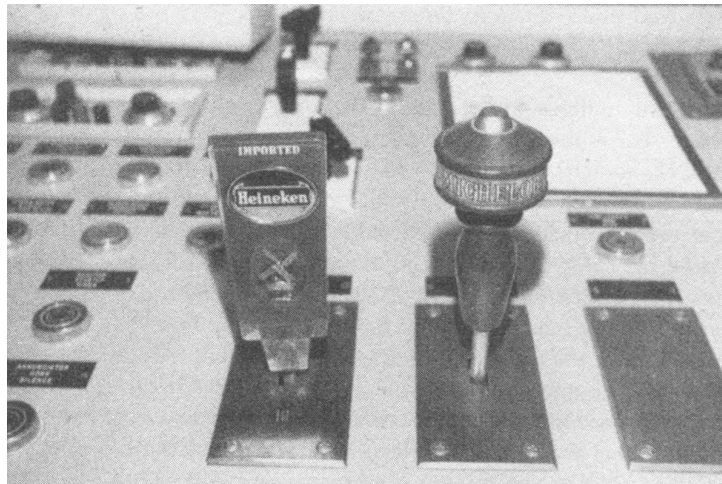


Figure 2.26: Beer tap handles mounted atop similar-looking knobs in the control room of a nuclear power plant to help operators distinguish between them. [Original photograph by Joseph L. Seminara, scanned from Norman [1988, page 95] with permission from Don Norman.]



Figure 2.27: The B17 Flying Fortress. [From Emgonzalez [2005], the image was placed in the public domain by the photographer.]

- In the B-17, the control knobs for flaps and landing gear were identical and located close to one another, as shown in Figures 2.28, 2.29, and 2.30.
- In the C-47, the flaps control was totally separate and was activated like a modern car handbrake.
- Chapanis realised that so-called “pilot errors” were really “designer errors” [Chapanis, 1999, page 16].
- As a quick fix, a small rubber wheel was attached to the end of the landing gear control and a wedge shape to the flap control.
- The gear-up landings ceased.
- After the war, the shape-coded wheel and flap controls were standardised and are still used today [CFR, 2008, pages 253–254], as shown in Figure 2.31.

To Err is Human

- People make errors routinely, you must design for error.
- Assume that any error, that can be made, will be made!
- Design explorable systems, where operations are easy to reverse.

Categories of Error

Two fundamental categories of error:

- *Slips* result from automatic behaviour, when subconscious actions toward a correct goal go wrong.

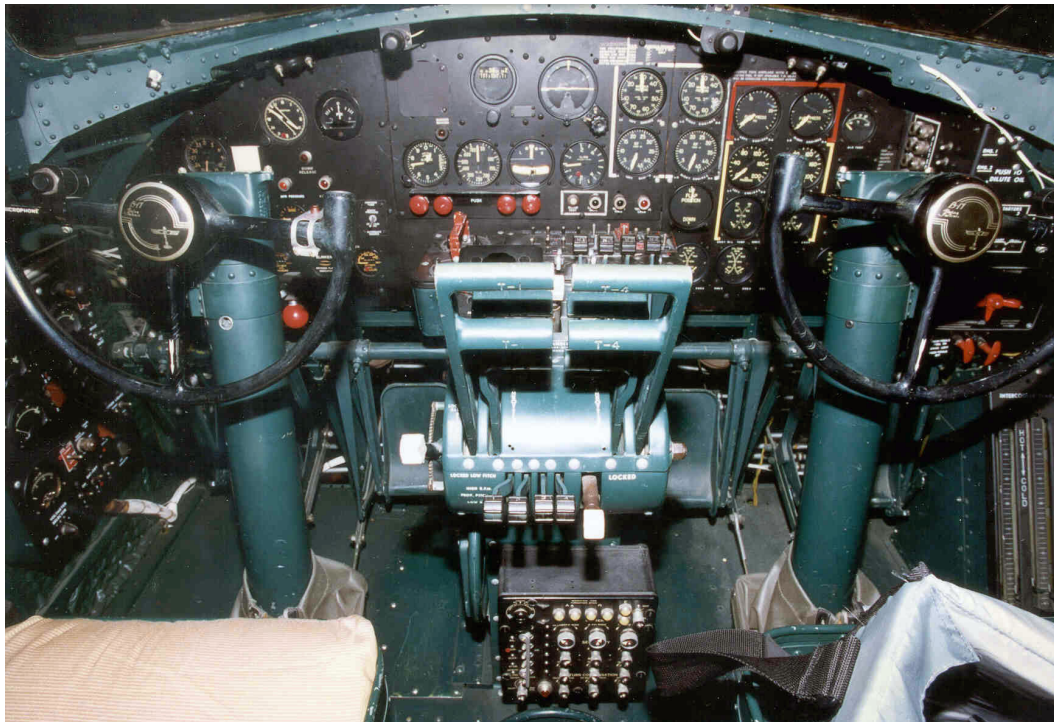


Figure 2.28: The cockpit of a B17G Flying Fortress. The landing gear and flaps control knobs are identical and are very close to one another. [From USAF [2006], image believed to be in the public domain.]



Figure 2.29: Close-up of the cockpit control knobs of a B17G Flying Fortress. The landing gear and flaps control knobs are identical and are very close to one another. [From USAF [2006], image believed to be in the public domain.]

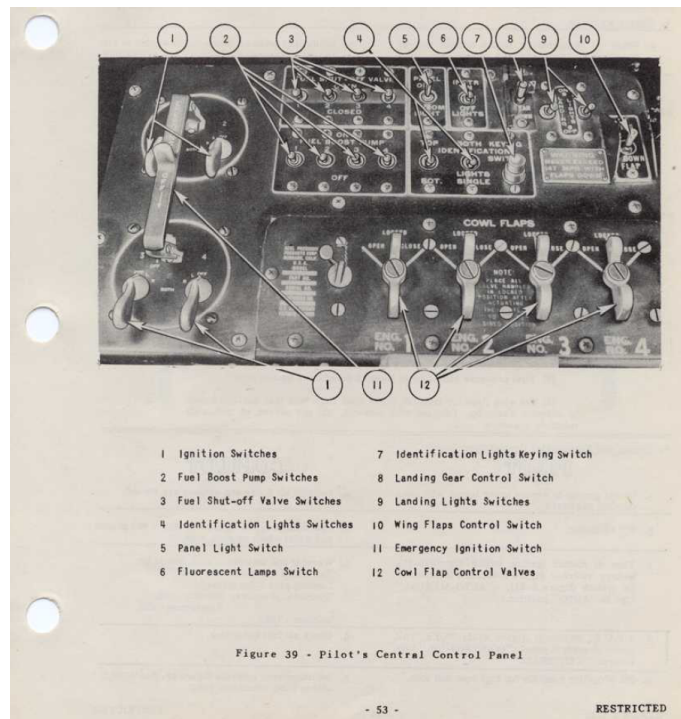


Figure 2.30: Page 53 from the B-17F Airplane Pilot's Flight Operating Instructions. Item 8 is the landing gear control switch and item 10 is the flap control. [From USAF [1942].]

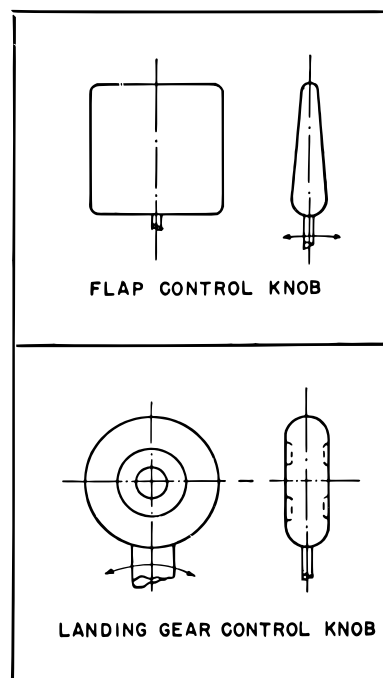


Figure 2.31: After WW2, the shape coding for landing gear and flaps control knobs was standardised. The landing gear control resembles a wheel and the flaps control resembles a flap. [From CFR [2008, page 254].]

- *Mistakes* result from conscious deliberations, which formed an inappropriate goal.

Conceptual Models

A *conceptual model* is a mental model of how something works, which is formed inside a person's head.

A user's conceptual model built up and influenced by numerous factors, including:

- familiarity with similar devices (transfer of previous experience)
- affordances
- mapping
- constraints
- causality
- instructions
- interacting with the device.

Conceptual models may be wrong, particularly if the above factors are misleading.

A Conceptual Model of a Fridge Freezer

A fridge-freezer with two compartments: the fridge for fresh food at the bottom and the freezer for frozen goods at the top, as shown in Figure 2.32:

- The two control dials (Figure 2.33) suggest a particular conceptual model (Figure 2.34) for operating the fridge freezer.
- Unfortunately, the apparent conceptual model does not match the way the fridge freezer actually works (Figure 2.35).

Adapted from [Norman, 1988], pages 17–18.

Projecting a Correct Conceptual Model

- Designers have their own conceptual model of a system, the *design model*.
- The *system image* is the actual implementation or embodiment of the design (including documentation, instructions, and labels).
- The *user's model* is built through interaction with the system.

The designer expects the user's model to be the same as the design model, however all communication takes place through the system image.

→ The system image should make the design model clear and consistent.



Figure 2.32: The fridge freezer.

NORMAL SETTINGS	C	AND	5	1 SET BOTH CONTROLS
COLDER FRESH FOOD	C	AND	6-7	2 ALLOW 24 HOURS
COLDEST FRESH FOOD	B	AND	8-9	TO STABILIZE
COLDER FREEZER	D	AND	7-8	
WARMER FRESH FOOD	C	AND	4-1	
OFF (FRESH FD & FRZ)			0	

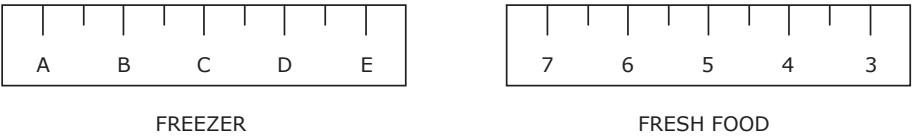


Figure 2.33: The fridge freezer controls and instructions.

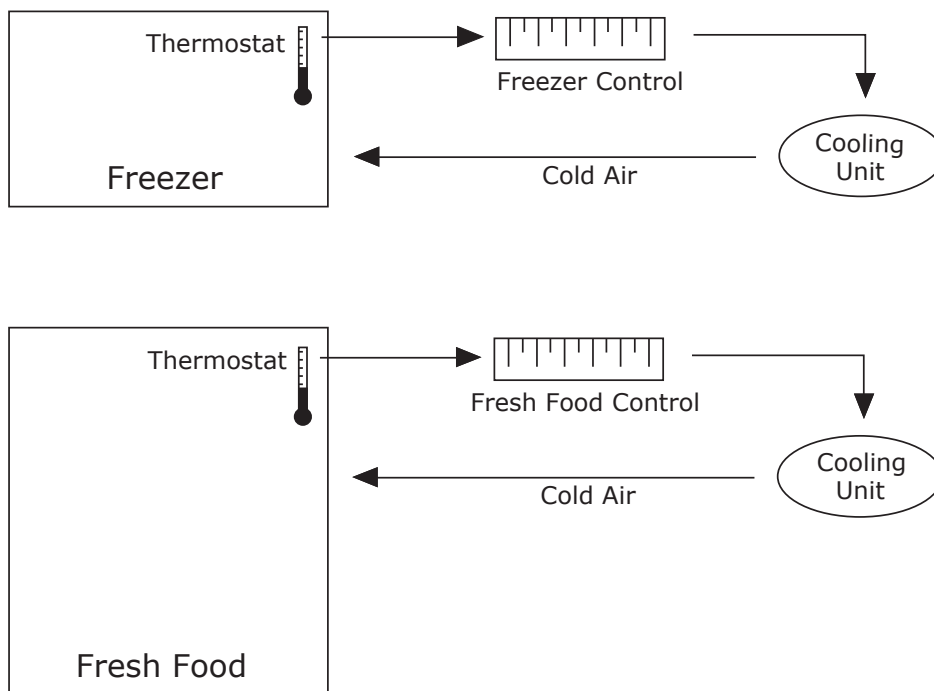


Figure 2.34: The apparent conceptual model for the fridge freezer, gleaned from the controls and instructions, is that each control is responsible for the temperature of the corresponding compartment.

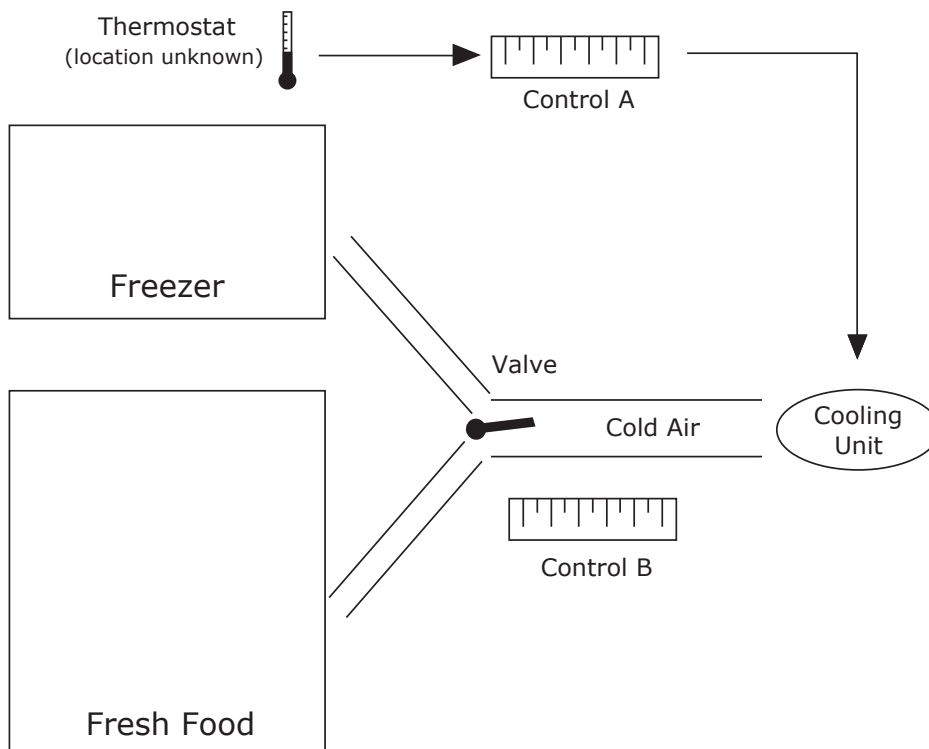


Figure 2.35: The actual conceptual model for the fridge freezer. In fact, there is only one thermostat and only one cooling unit. One control adjusts the thermostat setting, the other controls the relative proportion of cold air!

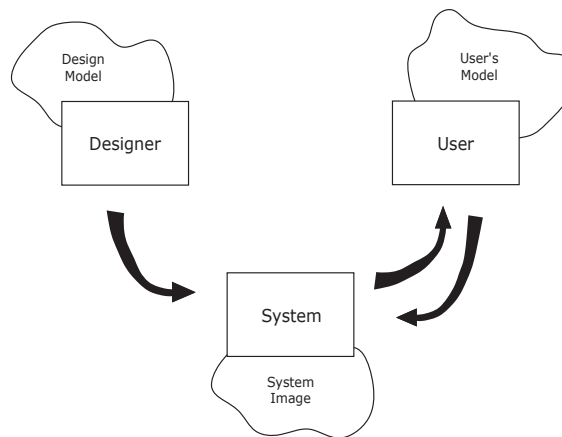


Figure 2.36: Projecting a correct conceptual model. Designers should take care to project an accurate conceptual model through the system image.



Figure 2.37: Scissors project a good conceptual model.

A Pair of Scissors Projects a Good Conceptual Model

- **Affordances:** holes for putting fingers in.
- **Constraints:** small hole for thumb, big hole for several fingers.
- **Mapping:** between holes and fingers suggested and constrained by appearance.
- **Conceptual Model:** operating parts are visible and their implications are clear.

A Digital Watch Projects No Visible Conceptual Model

- **Affordances:** four buttons to push – but what do they do?
- **Mapping:** no clear relationship between buttons and possible actions.
- **Transfer of Prior Knowledge:** little similarity to analog watches.
- **Conceptual Model:** must be learnt from instructions.

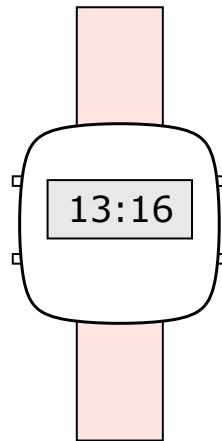


Figure 2.38: A digital watch provides no obvious conceptual model.

2.3 The Psychopathology of Computers

Usability “war stories” concerning computers ...

The PC Cup Holder

A supposedly true story from a Novell NetWire SysOp:

Caller: “Hello, is this Tech Support?”

Tech Rep: “Yes, it is. How may I help you?”

Caller: “The cup holder on my PC is broken and I am within my warranty period. How do I go about getting that fixed?”

Tech Rep: “I’m sorry, but did you say a cup holder?”

Caller: “Yes, it’s attached to the front of my computer.”

Tech Rep: “Please excuse me if I seem a bit stumped, it’s because I am. Did you receive this as part of a promotional, at a trade show? How did you get this cup holder? Does it have any trademark on it?”

Caller: “It came with my computer, I don’t know anything about a promotional. It just has ‘4X’ on it.”

At this point the Tech Rep had to mute the caller, because he couldn’t stand it. The caller had been using the load drawer of the CD-ROM drive as a cup holder, and snapped it off the drive.

This story was found at [Greenberg, 1997](#) and is attributed there to George Wagner g.wagner@sylvania.sev.org.

Dangerous Command Names

A widely used text editor (ed) used the character ‘.’ to select the current line of text, and ‘,’ to select the entire document for an operation.

- These two keys are adjacent on the keyboard → highly likely they will sometimes be mistaken.
- Intending to change one line

“A heavy poll is expected ...”

to

“A heavy turnout is expected . . .”

can easily change ‘poll’ to ‘turnout’ throughout the entire document.

- Such a case was reported in the British press: all the election documents of a candidate named Pollack were printed with the name Turnoutack.
- A “computer failure” was blamed.

This story is taken from [Newman and Lamming, 1995], pages 8–9.

Beware Unix Commands

- Intend to type: `rm *~` to remove Emacs backup files.
- Actually type: `rm * ~` which removes everything!
- And there is no undo . . .

The Terminal is Dead

Reported in the *Human Factors Society Bulletin*, 1981:

- The manager of a system installation for police departments reported that one day he received the call “your terminal is dead. Come and get it.”
- He suggested that the repair service should be contacted, but the caller insisted.
- The terminal had two bullet holes in it.
- Apparently, an officer got a “Do not understand” message on the screen once too often.

Phobos 1 Never Made it to Mars

From *Science* magazine, 1989, and reported by Norman in *CACM*, Jan. 1990 [Norman, 1990]:

“not long after the launch, a ground controller omitted a single letter in a series of digital commands sent to the spacecraft. And by malignant bad luck, that omission caused the code to be mistranslated in such a way as to trigger the test sequence”

- The test sequence, stored in ROM, was intended to be used only when checking the spacecraft on the ground.
- Phobos 1 went into a tumble from which it never recovered.
- The controller was moved to other duties.

Iran Air 655

Reported in [Lee, 1992]:

- In 1988, the USS *Vincennes* shot down an Iran Air A-300 Airbus with 290 people aboard.



Figure 2.39: A new keyboard designed to speed up access to the most commonly used feature on Windows PCs.

- The Aegis weapons system aboard the *Vincennes* had sophisticated software for identifying and tracking potential targets.
- However, the large-screen display did not show altitude information – altitude had to be read from separate consoles.
- The Airbus which had levelled off at 12 500 feet, was taken to be an F-14 fighter descending from 9 000 feet.
- Ironically, an escort ship with older equipment was able to read the plane’s altitude quite correctly, but could not intervene in time.

And Finally...

- A new keyboard designed to speed up access to the most commonly used feature on Windows PCs. [From the enemy.org web site <http://www.enemy.org/gallery/devices.shtml>, which no longer exists.]

Lessons

- Most failures of human-machine systems are due to poor designs which do not take account of peoples’ capabilities and fallibilities.
- These are often labelled as “computer failure” or “human error” rather than design failure.

2.4 Interface Hall of Shame

Examples of interface design mistakes, taken from the Interface Hall of Shame, which used to be at <http://www.iarchitect.com/shame.htm> but no longer exists. A partial mirror is available at <http://homepage.mac.com/bradster/iarchitect/shame.htm>

[Thanks to Brian Hayes from Isys Information Architects Inc. for permission to use them here.]

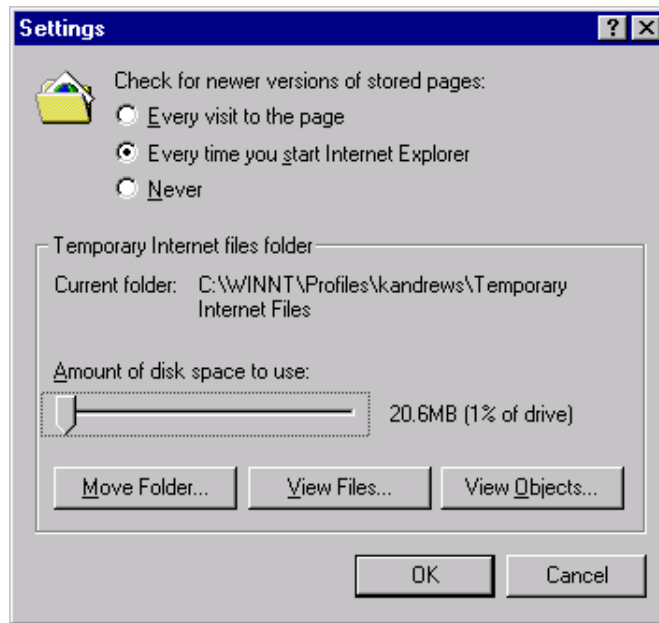


Figure 2.40: Internet Explorer 4.0 cache settings panel.

Smallest Setting is 1%

- The Internet Explorer 4.0 cache size can only be set in increments of 1% of the size of the hard disk, as shown in Figure 2.40.
- To quote from user Ross Cormier:

“The smallest setting is 1%. I have a 4 Gig drive, and don’t need 40 MB of cache thank you.”

Horizontal Scrolling

- Humans can scan written material faster from top to bottom rather than left to right.
- Vertically scrolling lists support single-item scrolling.
- The Internet Explorer 4.0 certificate authority selection panel uses horizontal scrolling, as shown in Figure 2.41.

Two Item List Box

- Visual Basic 5.0 uses a two (!) item list box.
- A drop down list or radio buttons would be much better.

Two Thousand Item List Box

- Do not put hundreds or thousands of items into a list box, either (see Figure 2.43).



Figure 2.41: Internet Explorer 4.0 certificate authority selection panel.

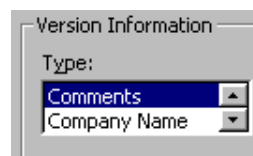


Figure 2.42: A two-item list box in Visual Basic 5.0.

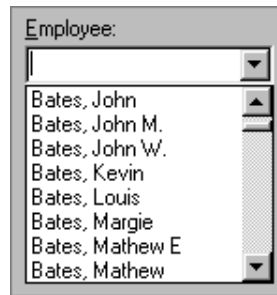


Figure 2.43: A two thousand item list box. Putting too many items into a list box is bad.

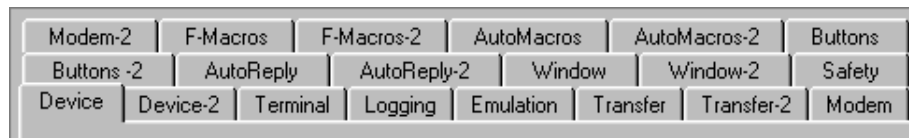


Figure 2.44: Zoc uses multi-row tab controls.

- The following message, posted in a Visual Basic programmers forum on 11th December 1996, is typical:

“I want to fill a list box with 2000 items ... This takes incredibly long ... over 20 minutes. Any ideas?”

Multi-Row Property Sheets

- Single-row property sheets (tab controls) are among the best user interface elements ever devised.
- Multi-row tab controls are perhaps one of the worst interface elements ever!
- Clicking one of the tabs other than from the bottom row, causes a major reorganisation of the complete set of tabs.

Figure 2.44 shows an example from Zoc, a communications program.

Stupid Error Messages

- Roy Child writes: *“I came across this message when trying to delete files from a nearly-full hard drive in Windows 95.”*

Avoid Breaking a Metaphor

- As a means of deleting files and documents, the Macintosh trash can is a perfectly intuitive metaphor.
- Unfortunately, the designers decided to extend the trash can metaphor to include the completely counterintuitive function of ejecting diskettes.

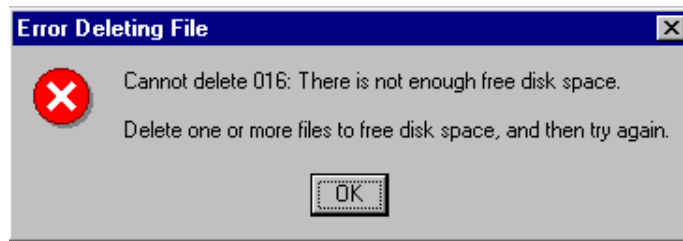


Figure 2.45: Deleting files from an almost full hard disk in Windows 95. OK, now I know what to do!

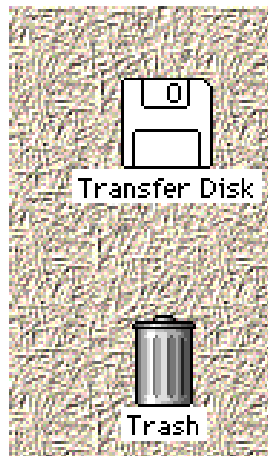


Figure 2.46: Ejecting a diskette on the Mac by throwing it into the trash unfortunately breaks the trash can metaphor.

- To eject a diskette, the user had to drag the diskette icon and drop it into the trash! See Figure 2.46.
- Later versions of the Mac keyboard have a dedicated eject button. See Figure 2.47.
- The interface was finally fixed in Mac OS X. When the user drags a diskette icon, the trash icon morphs into an eject icon. See Figure 2.48.

2.5 User-Centered Design

Science Finds, Industry Applies, Man Conforms.

[Motto of 1933 Chicago World's Fair]

People Propose, Science Studies, Technology Conforms.

[Don Norman's person-centered motto for 21st century]

System-Centered Design

- What can be built easily on this platform?
- What can I create from the tools available?
- What do I as a developer find interesting to work on?



Figure 2.47: Later versions of the Mac keyboard have a dedicated eject button.

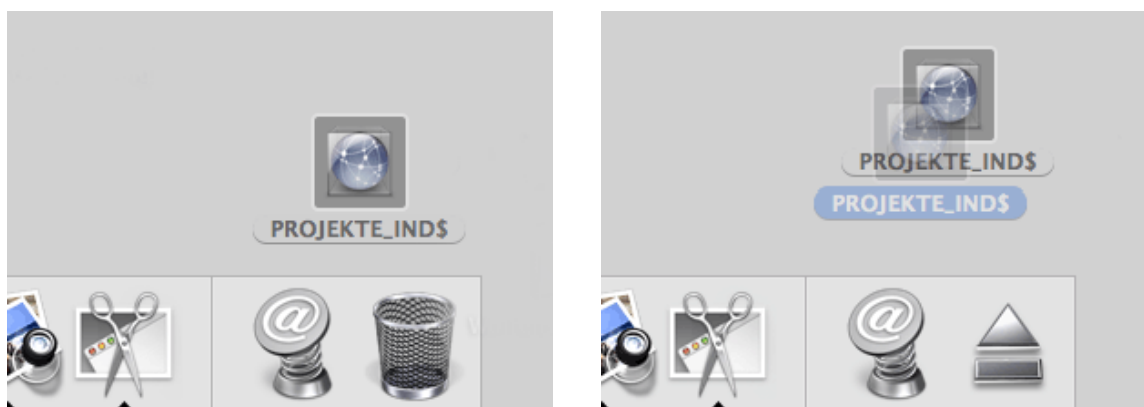


Figure 2.48: Mac OS X retains the trash can metaphor, but when the user drags a diskette icon, the trash can morphs into an eject icon. [Thanks to Tanja Kolrus for the screen shots.]

- What do I as a developer think users need?

User-Centered Design

The design is based upon a user's:

- abilities and needs
- context
- work
- tasks

Video: Simplicity Sells

- David Pogue; *Simplicity Sells*; TED 2006 talk, 21-minute video [[Pogue, 2006](#)] [04:00-12:26]

Chapter 3

Usability Engineering

*“ When the cook tastes the soup, that’s formative assessment.
When the guests taste the soup, that’s summative assessment. ”*

[Robert E. Stake, 1976. [Stake, 1976]]

Usability Engineering ... *iterative process* to improve usability of a system.

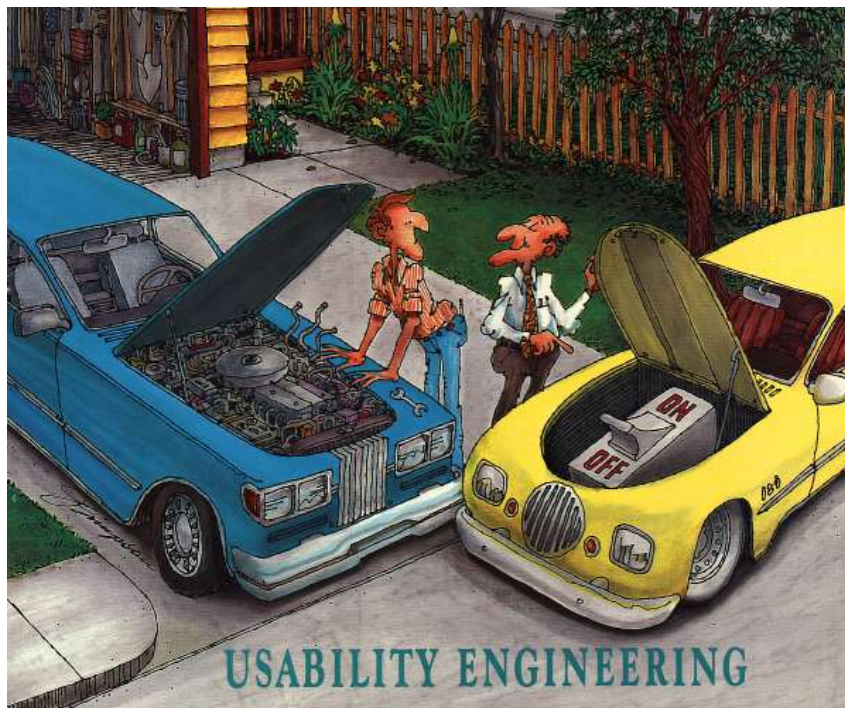


Figure 3.1: Usability engineering. By Jay Simpson, from the cover of IEEE Computer, March 1992.

References

- ++ Cooper, Reimann, and Cronin; *About Face 3: The Essentials of Interaction Design*; Wiley, 2007.
ISBN 0470084111 (com, uk) [Cooper, Reimann and Cronin, 2007]

- + Stone, Jarrett, Woodruffe, and Minocha; *User Interface Design and Evaluation*; Morgan Kaufmann, March 2005. ISBN 0120884364 (com, uk) [Stone et al., 2005]
- + Carol Barnum; *Usability Testing and Research*; Pearson Education, Oct. 2001. ISBN 0205315194 (com, uk) [Barnum, 2001]
- Jakob Nielsen; *Usability Engineering*; Morgan Kaufman, 1994. ISBN 0125184069 (com, uk) [Nielsen, 1993b]
- Deborah Mayhew; *The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design*; Morgan Kaufman, 1999. ISBN 1558605614 (com, uk) [Mayhew, 1999]
- Rosson and Carroll; *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*; Morgan Kaufman, 2001. ISBN 1558607129 (com, uk) [Rosson and Carroll, 2001]
- Larry Constantine and Lucy Lockwood; *Software for Use*; Addison-Wesley, 1999. ISBN 0201924781 (com, uk) [Constantine and Lockwood, 1999]
- ISO 9241-11 *Ergonomics requirements for office work with visual display terminals (VDTs), Part 11: Guidance on usability*, 1998. [ISO, 1998]
- ISO 13407 *Human-centred design processes for interactive systems*, 1999. [ISO, 1999]

Online Resources

- James Hom; *The Usability Methods Toolbox*. usability.jameshom.com
- Jakob Nielsen; *Nielsen Norman Group*. nngroup.com

3.1 Defining Usability

The ISO defines usability as “the extent to which a product can be used by *specified users* to achieve *specified goals* with *effectiveness*, *efficiency* and *satisfaction* in a *specified context of use*.” [ISO, 1998].

[The emphasis is mine, not part of the original definition.]

The three measurable usability attributes defined by ISO [1998] are:

- *Effectiveness*: accuracy and completeness with which users achieve specified goals.
- *Efficiency*: resources expended in relation to the accuracy and completeness with which users achieve goals.
- *Satisfaction*: freedom from discomfort, and positive attitudes towards the use of the product.

Usability in Context

Nielsen [1993b] defines usability in the context of overall system acceptability, as shown in Figure 3.2.

Six Usability Attributes

Combining the three ISO usability attributes with Nielsen's five usability attributes, leads to the following six usability attributes:

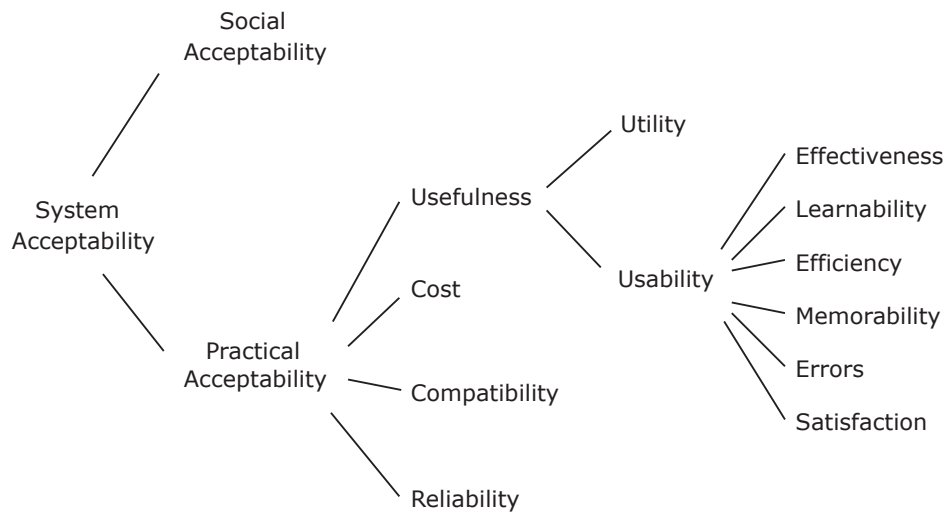


Figure 3.2: A model of the attributes of system acceptability, based on Figure 1 of [Nielsen, 1993b].

1. *Effectiveness*: completeness with which users achieve their goal.
2. *Learnability*: ease of learning for *novice* users.
3. *Efficiency*: steady-state performance of *expert* users.
4. *Memorability*: ease of using system intermittently for *casual* users.
5. *Errors*: error rate for minor and catastrophic errors.
6. *Satisfaction*: how satisfying a system is to use, from user's point of view.

Measuring Usability Attributes

- *Effectiveness*: decide on definition of success. For example, number of substitution words spotted in a text, or binary measure of success (order completed or not).
- *Learnability*: pick novice users of system, measure time to perform certain tasks. Distinguish between no/some general computer experience.
- *Efficiency*: decide definition of expertise, get sample expert users (difficult), measure time to perform typical tasks.
- *Memorability*: get sample casual users (away from system for certain time), measure time to perform typical tasks.
- *Errors*: count minor and catastrophic errors made by users while performing some specified task. For example, number of deviations from optimal click path.
- *Satisfaction*: ask users' subjective opinion (questionnaire), *after* trying system for real task.

3.2 Usability Evaluation

There are four types of evaluation, according to the purpose of the evaluation:

- *Exploratory* - how is it (or will it be) used?
- *Predictive* - estimating how good it will be.
- *Formative* - how can it be made better?
- *Summative* - how good is it?

My definitions [Andrews, 2008] are adapted from those of several authors [Stone et al., 2005; Rubin, 1994; Lockee, Moore and Burton, 2002; Ellis and Dix, 2006].

Exploratory Evaluation

Explores current usage and the potential design space for new designs.

- Done *before* interface development.
- Learn *which* software is used, *how often*, and *what for*.
- Collect *usage data* – statistical summaries and observations of usage.

Predictive Evaluation

Estimates the overall quality of an interface (like a summative evaluation, but a prediction made in advance).

- Done once a design has been done, but before implementation proceeds.

Formative Evaluation

Informs the design process and helps improve an interface during design.

- Done *during* interface development.
- Learn *why* something went wrong, not just that it went wrong.
- Collect *process data* – qualitative *observations* of what happened and why.

Summative Evaluation

Assesses the overall quality of an interface.

- Done once an interface is (more or less) *finished*.
- Either compare alternative designs, or test definite performance requirements.
- Collect *bottom-line data* – quantitative *measurements* of performance: how long did users take, were they successful, how many errors did they make.

Modified Soup Analogy

Extending Robert Stake's soup analogy [Stake, 1976; Lockee, Moore and Burton, 2002]:

“When the cook tastes other cooks' soups, that's *exploratory*.
When the cook assesses a certain recipe, that's *predictive*.
When the cook tastes the soup while making it, that's *formative*.
When the guests (or food critics) taste the soup, that's *summative*.”

Usability Evaluation Methods

The methods of usability evaluation can also be classified according to who performs them:

- *Usability Inspection Methods*
Inspection of interface design by usability specialists using *heuristics* and *judgement* (no test users).
- *Usability Testing Methods*
Empirical testing of interface design with real users.

Figure 3.3 illustrates some of the different inspection and testing methods, grouped by purpose and by who performs them.

3.3 The Usability Engineering Lifecycle

1. Know the User
2. Usability Benchmarking
3. Goal-Oriented Interaction Design
4. Iterative Design:
 - (a) Prototyping
 - (b) Formative Usability Evaluation (Inspection and/or Testing)
5. Summative Usability Evaluation
6. Follow-up Studies

The lifecycle is illustrated in Figure 3.4.

3.3.1 Know the User

- Qualitative research: observation of users and interviews.
- Classify users according to their characteristics.
- Draw up a *user profile* for each (potential) class of user, based on behavioural and demographic variables.
- Identify user goals and attitudes.

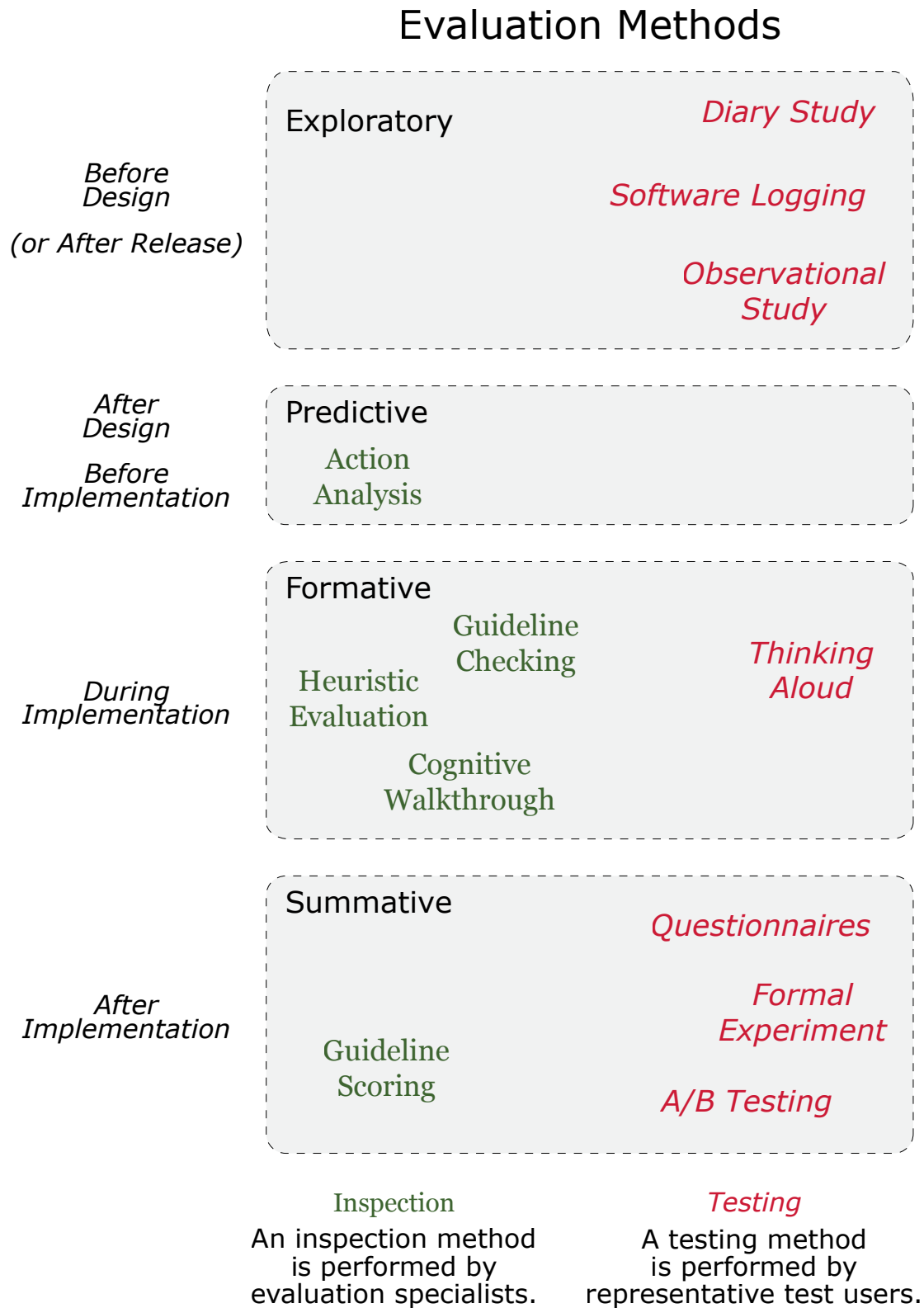


Figure 3.3: Nine common evaluation methods grouped by purpose and by who performs them.

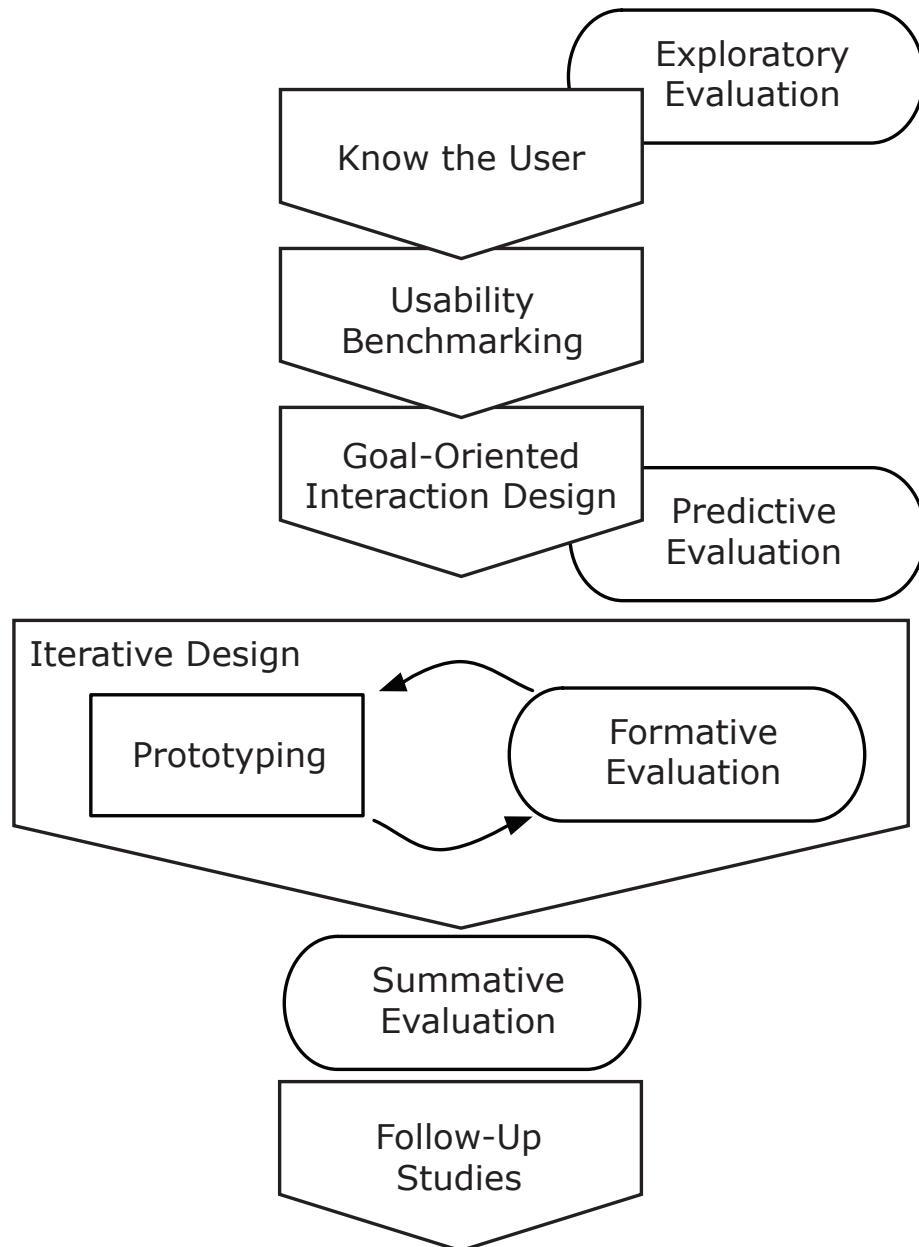


Figure 3.4: The usability engineering lifecycle. Adapted from a figure kindly provided by Martin Loitzl.

- Analyse workflow and context of work.
- Exploratory evaluation: which software is used, how is it used, and what is it used for.
- Draw up a set of typical *user scenarios*.

See Chapter 4.

3.3.2 Usability Benchmarking

- Analyse competing products or interfaces heuristically and empirically.
- Set measurable usability targets for your own interface.

See Chapter 5.

3.3.3 Interaction Design

Goal-oriented initial design of interface.

See Chapter 6.

3.3.4 Iterative Design

“Design, Test, Redesign.”

Build and evaluate prototype interface, then:

- *Severity ratings* of usability problems discovered.
- Fix problems → new version of interface.
- Capture *design rationale*: record reasons why changes were made.
- Evaluate new version of interface.

until time and/or money runs out.

A cycle of continuous improvement.

Building Prototypes

- Verbal description.
- Paper prototype.
- Working prototype.
- Implementation of final design.

See Chapter 7.

Formative and Summative Usability Evaluation

The usability evaluation methods are described according to who performs them:

- *Usability inspection methods*: Chapter 8.
- *Usability testing methods*: Chapter 9.

3.3.5 Follow-Up Studies

Important usability data can be gathered after the release of a product for the next version:

- Specific field studies (interviews, questionnaires, observation).
- Standard marketing studies (what people are saying in the newsgroups and mailing lists, reviews and tests in magazines, etc.).
- Analyse user complaints to hotline, modification requests, bug reports.
- Usage studies of longer-term use of product (see Section 9.8):
 - Diary studies.
 - Software logging: instrumented versions of software → log data.
 - Observational studies.

3.4 Planning Usability Activities

1. Prioritise activities.
2. Write down explicit plan for each activity.
3. Subject plan to independent review (e.g. colleague from different project).
4. Perform pilot activity with about 10% of total resources, then revise plan for remaining 90%. [*Always perform a pilot study!*]

Chapter 4

Know the User

“ I’m a very selfish designer: when I design software, I design it for me. And so my first task is to become you. ”

[Bruce Tognazzini, The Front Desk, BBC Video, 1996. [BBC, 1996, 00:21:28]]

Qualitative research is used to determine user characteristics, goals, and context of use.

References

- + JoAnn Hackos and Janice Redish; *User and Task Analysis for Interface Design*; Wiley, 1998. ISBN 0471178314 (com, uk) [Hackos and J. C. Redish, 1998]
- + Mike Kuniavsky; *Observing the User Experience: A Practitioner’s Guide to User Research*; Morgan Kaufmann, April 2003. ISBN 1558609237 (com, uk) [Kuniavsky, 2003]
- o Hugh Beyer and Karen Holtzblatt; *Contextual Design : A Customer-Centered Approach to Systems Designs*; Morgan Kaufmann, 1997. ISBN 1558604111 (com, uk) [Beyer and Holtzblatt, 1997]
- o Catherine Courage and Kathy Baxter; *Understanding Your Users*; Morgan Kaufmann, 2004. ISBN 1558609350 (com, uk) [Courage and Baxter, 2004]
- + Alan Cooper, Robert Reimann, and David Cronin; *About Face 3: The Essentials of Interaction Design*; Wiley, 2007. ISBN 0470084111 (com, uk) [Cooper, Reimann and Cronin, 2007]
- o Kate Gomoll, Ellen Story Church, and Eric Bond; *The Field Study Handbook*; UIE, June 2007. http://uie.com/reports/field_study_handbook/

4.1 Classifying Users

Users can be classified according to their:

- experience
- educational level
- age
- amount of prior training, etc.

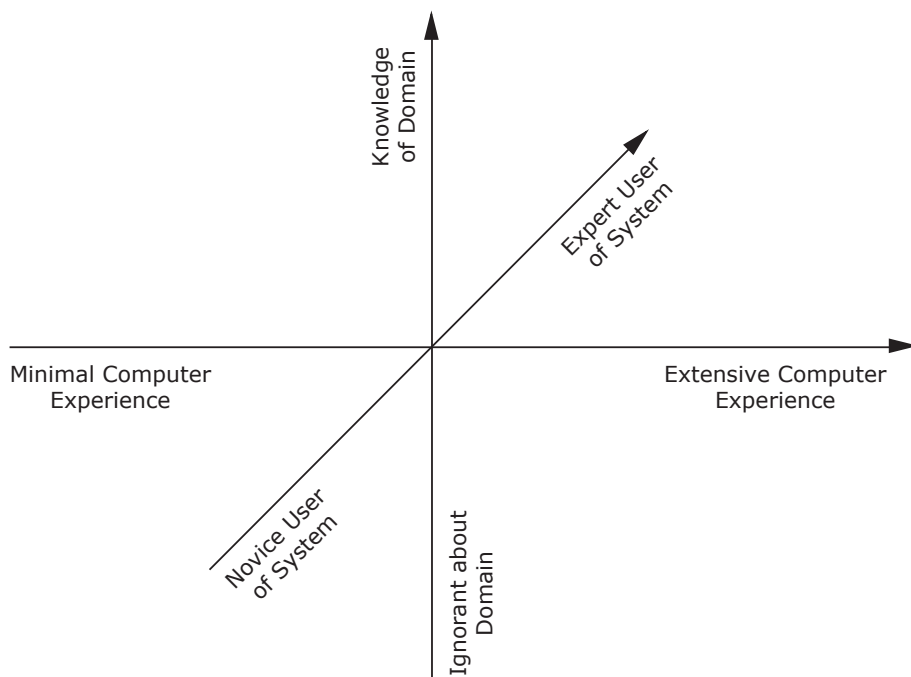


Figure 4.1: The three main dimensions on which user experience varies: experience of computers in general, understanding of the task domain, and expertise in using the specific system. From Figure 3 of [Nielsen, 1993b].

Categories of User Experience

User experience can be thought of along three dimensions, as shown in Figure 4.1.

Learning Curves

- Some systems are designed to focus on learnability.
- Others emphasise efficiency for proficient users.
- Some support both ease of learning and an “expert mode” (for example rich menus and dialogues plus a command/scripting language), and thus attempt to ride the top of the curves in Figure 4.2.

Most Users are Perpetual Intermediates

The experience level of people using computer software tends, like most population distributions, to follow the classical statistical bell curve (normal distribution).

In terms of using a software interface, the bell curve represents a snapshot in time:

- Beginners do not remain beginners for long.
- The difficulty of maintaining a high level of expertise means that experts fade over time.
- Most users gravitate over time towards intermediacy.

Most users are neither beginners nor experts: they are *perpetual intermediates*.

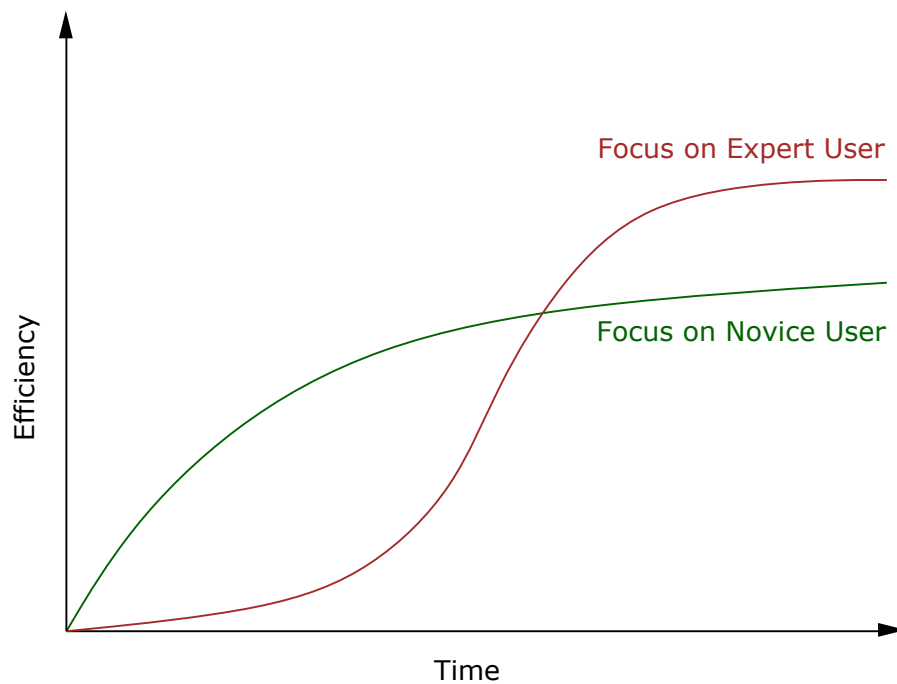


Figure 4.2: Learning curves for hypothetical systems focusing on the novice user (easy to learn, but less efficient to use) and the expert user (harder to learn, but then highly efficient). From Figure 2 of [Nielsen, 1993b].

4.2 Research the Frames of Reference

Conduct interviews with:

- Project staff (managers, programmers, marketing people) who are in charge of developing the software.
- Subject matter and domain experts.
- Customers (the purchaser of the product, not necessarily the same as the end user).

to determine values, expectations, issues, and constraints.

Interviewing Project Staff

- One-on-one interviews.
- Try to discover:
 - vision of the product.
 - budget and schedule.
 - technical constraints.
 - perceptions of who users might be.

Interviewing Subject Matter Experts (SMEs)

- Often hired externally by project manager.
- Provide knowledge of complex domains, regulations, industry best practice.
- Often lean towards expert user perspective (rather than intermediate).

Interviewing Customers

- Customers are the people who make the decision to purchase.
- For consumer products, customers are often the same as users.
- For business settings, customers are rarely actually the users of a product.
- Try to discover the customer's:
 - goals in purchasing the product
 - frustrations with current solutions
 - decision process for purchasing
 - role in installation and maintenance

4.3 Research the End User

The actual *users* of a product should always be the main focus of the design effort.

- Most people are incapable of accurately assessing their own behaviour [Pinker, 1999].
- Rather than talk to users about how they think they behave, it is better to observe their behaviour first-hand.
- And then ask clarifying questions in the context of use.

Ethnographic Interviews

A combination of immersive observation and directed interview techniques.

- Observe the user using their current tools in their normal environment.
- Interviewer assumes the role of an apprentice learning from the master craftsman (user).
- Alternate between observation of work and discussion of its structure and details.

Identifying Candidate Users

Designers must capture the range of user behaviours regarding a product.

- What sorts of people might use this product?

- How might their needs vary?
- What ranges of behaviour might be involved?
- Which kinds of environment might be involved?

Try to interview some people from each different group.

Examples

Whom would you interview if you were designing:

- An in-flight entertainment system?
- A corporate help desk?
- A complete hospital management system?
- A mobile phone with email capability?

Conducting an Ethnographic Interview

- In actual workplace/environment.
- 45-60 minutes.
- No third parties (supervisors or clients).
- Focus on understanding:
 - Overall goals
 - Current tasks
 - Constraints and exceptions
 - Problems needing solution (where does it hurt?)
 - Broader context
 - Domain issues
 - Vocabulary
- Ask permission to take a few photographs of the user and their workplace (for creating personas).

Patterns of Use

When interviewing users, we are trying to discover patterns of use:

- Business products: Patterns of use are generally based on job responsibilities.
- Consumer products: Patterns of use are generally based on lifestyle (age, gender, occupation, etc).

Being an Active Listener

A good interviewer is an active listener:

- Use open body language: lean forward, hand under chin, arms open, eye contact.
- Use minimal encouragers: brief verbal cues (hmmm, uh-huh, oh?), nodding, tilting head sideways.
- Ask open-ended questions (how, when, what, why) to encourage elaboration.
- Use closed questions (can you, will you, do you) with yes/no or simple fact answer to clarify your understanding.
- Summarise to check you understand the important points: “So it sounds like the key points are...”.

General Flow of Interview for Business Product

- Introductions.
- Why we’re here: We’ve been asked to design/improve X.
- What we’ll ask: your day, your background, your frustrations.
- Tell us about your responsibilities and your typical workday.
- Drill into specific tasks.
- How is existing product (if any) involved in those tasks.
- Relationships with other people and processes.
- Goals.
- Follow up on interesting points.
- Wrap-up.

Good General Questions

- What do you spend most of your time on? [task priority]
- What things waste your time? [opportunity]
- Where does it hurt? [opportunity]
- What makes a good work day? A bad one? [goals]
- What kind of training do you have? [support to provide]
- What are the most important things you do? [priorities, goals]
- What information helps you make decisions? [info to provide]

Chapter 5

Usability Benchmarking

Usability benchmarking:

- how usable is the competition?
- how much better should your interface be?
- what is your likely return on investment?

References

- Randolph Bias and Deborah Mayhew (Eds.); *Cost-Justifying Usability*; 2nd Edition, Morgan Kaufmann, April 2005. ISBN 0120958112 (com, uk) [Bias and Mayhew, 2005]
- Nielsen Norman Group; *Usability Return on Investment*; Nielsen Norman Group, 2003. <http://www.nngroup.com/reports/roi/>
- Eric Schaffer; *Institutionalization of Usability*; Addison-Wesley, 2004. ISBN 032117934X (com, uk) [Schaffer, 2004]
- John S. Rhodes; *Selling Usability: User Experience Infiltration Tactics*; CreateSpace, Feb 2009. ISBN 1442103736 (com, uk) [Rhodes, 2009]

Online Resources

- Jakob Nielsen; *Return on Investment for Usability Alertbox*, January 2003. <http://www.useit.com/alertbox/20030107.html>
- Peter Merholz and Scott Hirsch; *Report Review: Nielsen/Norman Group's Usability Return on Investment* Boxes and Arrows, July 2003. http://www.bboxesandarrows.com/archives/report_review_nielsennorman_groups_usability_return_on_investment.php
- Aaron Marcus; *Return on Investment for Usable User Interface Design: Examples and Statistics*; http://www.amanda.com/resources/ROI/AMA_ROIWhitePaper_28Feb02.pdf

5.1 Competitive Analysis

Competitive analysis of competing systems:

	<i>Q1</i>	<i>Median</i>	<i>Q3</i>
Project size (person-years)	11.0	23.0	58.0
Actual usability budget (% total)	4%	6%	15%
Ideal usability budget (% total)	6%	10%	21%
Actual usability effort (person-years)	1.0	1.5	2.0
Ideal usability effort (person-years)	1.7	2.3	3.8

Table 5.1: Survey of the usability budgets of 31 projects having some usability activities. From Table 1 of [Nielsen, 1993b].

- Determine the current state of the art and decide how far to raise the bar.
- Analyse competing products or interfaces heuristically (run a heuristic evaluation) or empirically (run a thinking aloud test or formal experiment).
- “Intelligent borrowing” of ideas from other systems.

5.2 Set Usability Targets

- Decide in advance on usability metrics and desired level of *measurable* usability (usability targets). For example:
 - The current system exhibits 4.5 errors per hour on average for an experienced user. The target for the new version is less than 3 errors per hour.
 - From competitive analysis, on the main competing web site, novice users take 8 mins. and 21 secs. on average to book a flight. The target for our new web site is 6 mins.

5.3 Return on Investment

Estimate *return on investment* (ROI) by performing a financial impact analysis:

- Compare potential savings based on *loaded cost* of users to to the estimated cost of the usability effort.

Jakob Nielsen concludes [Nielsen, 2003] that current best practices call for devoting about 10% of a project’s budget to usability.

See Table 5.1.

Chapter 6

Goal-Oriented Interaction Design

“ What should be in the designer’s mind at the start of an interface project? That’s simple. Nothing. ”

[Don Norman, The Front Desk, BBC Video, 1996. [BBC, 1996, 00:21:20]]

References

- ++ Alan Cooper et al; *About Face 3: The Essentials of Interaction Design*; Wiley, 2007. ISBN 0470084111 (com, uk) [Cooper, Reimann and Cronin, 2007]
- ++ Alan Cooper; *The Inmates are Running the Asylum*; Sams, 2004. ISBN 0672326140 (com, uk) [Cooper, 2004]
- + John Pruitt and Tamara Adlin; *The Persona Lifecycle*; Morgan Kaufmann, 2006. ISBN 0125662513 (com, uk) [Pruitt and Adlin, 2006]
- + Steve Mulder and Ziv Yaar; *The User Is Always Right: A Practical Guide to Creating and Using Personas for the Web*; New Riders, 2006. ISBN 0321434536 (com, uk) [Mulder and Yaar, 2006]
- + Ellen Isaacs and Alan Walendowski; *Designing from Both Sides of the Screen*; New Riders, Dec. 2001. ISBN 0672321513 (com, uk) [Isaacs and Walendowski, 2001]
- o Joel Spolsky; *User Interface Design For Programmers*; APress, June 2001. ISBN 1893115941 (com, uk) [Spolsky, 2001]

Online Resources

- o Alan Cooper; *Cooper Interaction Design*. <http://www.cooper.com/>
- o Kim Goodwin; *Getting from Research to Personas: Harnessing the Power of Data*; http://www.cooper.com/content/insights/newsletters/2002_11/getting_from_research_to_personas.asp
- o Kim Goodwin; *Perfecting Your Personas*; http://www.cooper.com/content/insights/newsletters/2001_07/perfecting_your_personas.htm
- o Dan Saffer; *Persona Non Grata*; <http://adaptivepath.com/publications/essays/archives/000524.php>
- o David Anderson; *uidesign.net*. <http://www.uidesign.net/>

<i>Computers</i>	<i>Humans</i>
Incredibly fast	Incredibly slow
Error free	Error prone
Deterministic	Irrational
Apathetic	Emotional
Literal	Inferential
Sequential	Random
Predictable	Unpredictable
Amoral	Ethical
Stupid	Intelligent

Table 6.1: There are incredible differences between computers and the humans who have to use them.

- Joel Spolsky; *User Interface Design For Programmers*; <http://www.joelonsoftware.com/uibook/fog0000000249.html>
- dreamstime; <http://www.dreamstime.com/>
- morgueFile; <http://morguefile.com/>
- stock.xchng; <http://www.sxc.hu/>
- OPENPHOTO; <http://openphoto.net/>
- Visipix; <http://visipix.com/>
- Flickr; <http://www.flickr.com/>
- Larry Bercow Photography; <http://www.bercowstudio.com/>
- Free Images; <http://www.freeimages.co.uk/>
- Google Image Search; <http://images.google.com/> (but beware of copyright!).
- Wikipedia free image resources; http://en.wikipedia.org/wiki/Public_domain_image_resources

Computers Versus Humans

Computers do not work like humans (see Table 6.1).

- One part of software, the inside, must clearly be written in harmony with the demands of silicon.
- Equally, the other side of software, the outside, must be written in harmony with the demands of human nature.

Programmers are Different

Programmers (“homo logicus”) think and behave differently from normal humans (homo sapiens) and most users. See Table 6.2.

<i>Programmer</i>	<i>User</i>
Wants control, will accept some complexity.	Wants simplicity, will accept less control.
Wants to understand, will accept some failure.	Wants success, will accept less understanding.
Concerned with all possible cases, will accept advance preparation.	Concerned with probable cases, will accept occasional setbacks.

Table 6.2: Programmers (“homo logicus”) think and behave differently from normal humans (homo sapiens). They are more in tune with the needs of computers.

Programmers are good at designing the inside of software, interaction designers should design the outside.

Interaction Design versus Interface Design

Interface design suggests an interface between code on one side and users on the other side, passing messages between them. It gives programmers a licence to code as they please, because the interface will be slapped on when they are done.

Interaction design refers to function, behaviour, and final presentation.

Goal-Oriented Interaction Design

Designing software based on an understanding of human goals.

What is a goal?

- A *goal* is a final purpose or aim, an objective.
- *Tasks* are particular ways of accomplishing a goal.

There may be multiple ways of achieving a goal.

Tasks are not Goals

- **Goal:** Get something to eat.
- **Task:** Go to the restaurant around the corner. Or
- **Task:** Call the pizza delivery service. Or
- **Task:** Go to the supermarket, buy ingredients, and cook for myself.

Too often, software designers focus on simplifying a task, rather than accomplishing a goal. Tasks are a means to an end, not an end in themselves.

Tasks Change with Technology

Tasks change with technology, goals do not:

<i>Personal Goals</i>	<i>Corporate Goals</i>
Do not look stupid. Not make mistakes. Get adequate amount of work done. Have fun.	Increase profit. Increase market share. Defeat competition. Hire more people. Go public.

Table 6.3: Personal and corporate goals are different. Both must be taken into account for software to succeed, but personal goals will always dominate.

- Year 2000
 - **Goal:** Get to work.
 - **Task:** Take the tram.
 - **Task:** Take a taxi.
 - **Task:** Drive in traffic.
- Year 3000
 - **Goal:** Get to work.
 - **Task:** Press the teleport button.
 - **Task:** Fly with jet pack.

Personal and Corporate Goals

Personal and corporate goals are different (See Table 6.3).

- Both are the highest expression of goals for their respective owners (both must be taken into account).
- But people are doing the work, and their personal goals will always take precedence (although they are rarely discussed, precisely because they are personal).

The Interaction Design Process

1. Interview users
2. Create personas
3. Define their goals
4. Create concrete scenarios
5. Move to a design solution

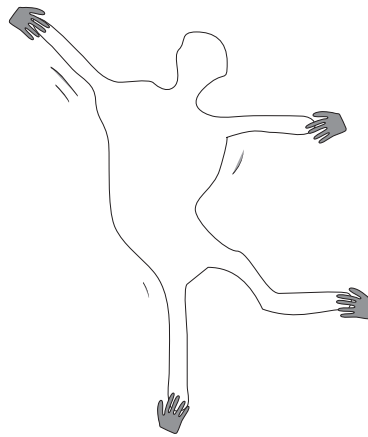


Figure 6.1: The term “user” is elastic and is liable to be bent and stretched by the programmer to the needs of the moment.

The Design Team

Two designers in core team:

- Designer: generates ideas, leads the process.
- Design Communicator: articulates half-formed ideas, writes design spec.

6.1 Creating Personas

From the insight you gained in your interviews, you now invent user archetypes to represent the main user groups of your product.

In other words, you make up pretend users and design for them.

The Elastic User

All too often, the term “user” is bent and stretched by the programmer to adapt to the needs of the moment (see Figure 6.1):

- When the programmer finds it convenient to dump the user into the Windows file system, the elastic user is defined as an accommodating, computer-literate power user.
- Other times, when the programmer finds it convenient to step the user through a difficult process with a mindless wizard, the elastic user is redefined as an obliging first-time novice.

Never refer to “the user”, instead refer to a very specific individual, a persona.

Do Not Design for the Average User

- Designing for the “average” user produces a design to please no-one, like the jumble car in Figure 6.2.
- Differentiate primary kinds of user and design for them, like the cars in Figure 6.3.

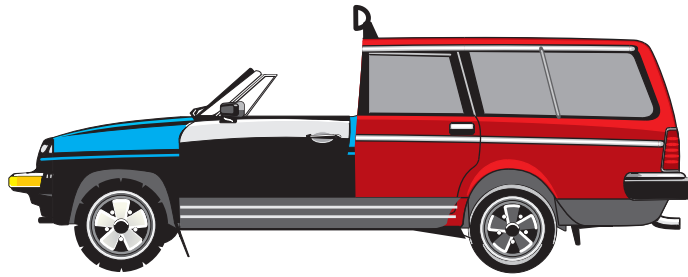


Figure 6.2: The jumble car was designed for the “average” driver.

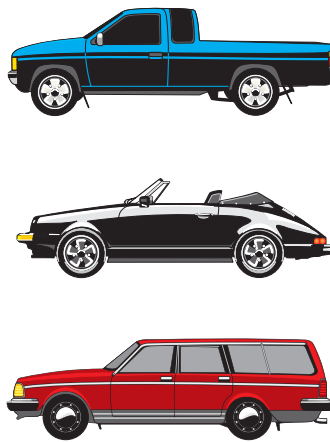


Figure 6.3: Cars are designed to appeal to different kinds of drivers with different needs and goals.

What is a Persona

A *persona* is a prototypical user:

- An imaginary, but very specific, example of a particular type of user.
- Not “real”, but hypothetical.

A persona is used to role-play through an interface design and check if the design would meet the needs of such a user.

A Good Persona

- A good persona is not “average”, but typical and believable.
- If the set of users interviewed were somehow plotted according to their characteristics as a cloud of points, the best ones to base personas on would be the ones around the edges.
- If our design satisfies the hard cases around the edges, the ones in the middle should be able to use the interface as well.

Define the Persona Precisely

- Specify a name, age, face, and quirky, believable detail.
- For faces, use stock photos from CD-ROM or the internet, or photographs taken during user interviews.
- It is more important to define the persona in great and specific detail, so that it cannot wiggle under the pressure of development, than that the persona be exactly the right one.

Finding Primary and Secondary Personas

- Start off with a larger set of personas.
- Combine or throw out redundant personas.
- A *primary* persona will not be satisfied with a design for someone else.
 - If there are multiple personas with radically different needs, there are multiple primaries.
 - Each primary gets their own interface.
- A *secondary* persona is mostly satisfied with a primary's interface, but has a specific additional need.

Case Study: In-Flight Entertainment System

Fictional example based on example in [Cooper, 2004], an inflight entertainment system called InFlight for Zoom Airways.

At each seat a touch-screen video console (see Figure 6.4):

- 36 films in five categories, 36 music channels, news, childrens shows, games, shopping.
- computers + large hard disks in front of the plane.
- true video on demand – each passenger can start, pause, and rewind programmes independently.

Case Study: Two Separate Interfaces

- One for the passengers in the seat console.
- A different one for employees in the attendant's station.

Case Study: Who are the Primary Passenger Personas?

The seat console interface has to satisfy Chuck, Erin, Marie, and Clevis, while at the same time not making any of them unhappy.

- Erin knows wanting to play games is something special, so she does not mind pressing a few extra buttons to get them.


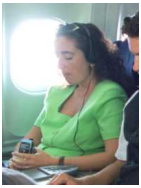

	<p>Name: Clevis McQuinn Age: 63 Class: Economy</p> <p>Clevis was born and still lives in a small town in Texas. He only flies once or twice a year to visit his daughter who lives in Boston.</p> <p>Clevis might be old, but he is still spry. He is slightly embarrassed about the touch of arthritis in his hands, but his mind is still very sharp.</p> <p>Clevis does not own a computer and does not know how to use one. He is firmly of the opinion that you can get by without one.</p> <p>Clevis had to start wearing glasses about 5 years ago, because his eyesight was starting to fail him.</p>
	<p>Name: Marie Dupart Age: 31 Class: Business</p> <p>Marie was born in France, but has been living and working in the USA for 6 years. She is bilingual, but English is her second language.</p> <p>Marie travels on business several times a year. She is a self-confident young woman, who is not afraid of modern gadgetry. She owns a PDA and an iPod.</p> <p>Marie does much of her shopping online. She is also very interested in the latest show business gossip in the entertainment media.</p>
	<p>Name: Chuck Burgstein Age: 52 Class: First</p> <p>Chuck is a resident of New York who flies almost every week. He is a member of the 100,000-mile club. He has an extremely hectic lifestyle and spends more than 100 nights a year in hotel rooms.</p> <p>Chuck expects service here and now and has little tolerance for condescending or time-consuming activities.</p> <p>Chuck has strong opinions, which he is not shy to express. Even if he is usually right, other people do find his in-your-face manner somewhat irritating.</p>
	<p>Name: Erin Scott Age: 9 Class: Economy</p> <p>Erin lives Austin, Texas and is going to stay with her aunt and uncle in upstate New York for two weeks.</p> <p>She is a little bit nervous, but also excited about travelling unaccompanied for the first time. Erin likes drinking fizzy orange pop. At home she will often spend hours and hours on her computer playing Sims2.</p>

Table 6.4: The four main passenger personas.

	<p>Name: Brent Coverham Age: 33 Position: Purser Brent is new to Zoom, having spent 7 years at another airline.</p>
	<p>Name: Amanda Surrey Age: 28 Position: Flight attendant After safety, Amanda must focus on assuring that each passenger has the best experience possible. She uses the InFlight console at the flight attendant's station to deal with any problems during flight.</p>
	<p>Name: Molly Springer Age: 41 Position: Ground staff Molly is responsible for uploading new content to the InFlight system.</p>
	<p>Name: Joseph C. Wong Age: 47 Position: Pilot Born in Seattle to Chinese parents, Joseph transferred to Zoom Airways from the military.</p>
	<p>Name: Max "Hoppy" Hooper Age: 51 Position: Mechanic Hoppy is an aircraft enthusiast who has been working at Zoom for 15 years. Hoppy services and maintains the InFlight system, fixing any bugs, wiring, etc.</p>

Table 6.5: The five main employee personas.

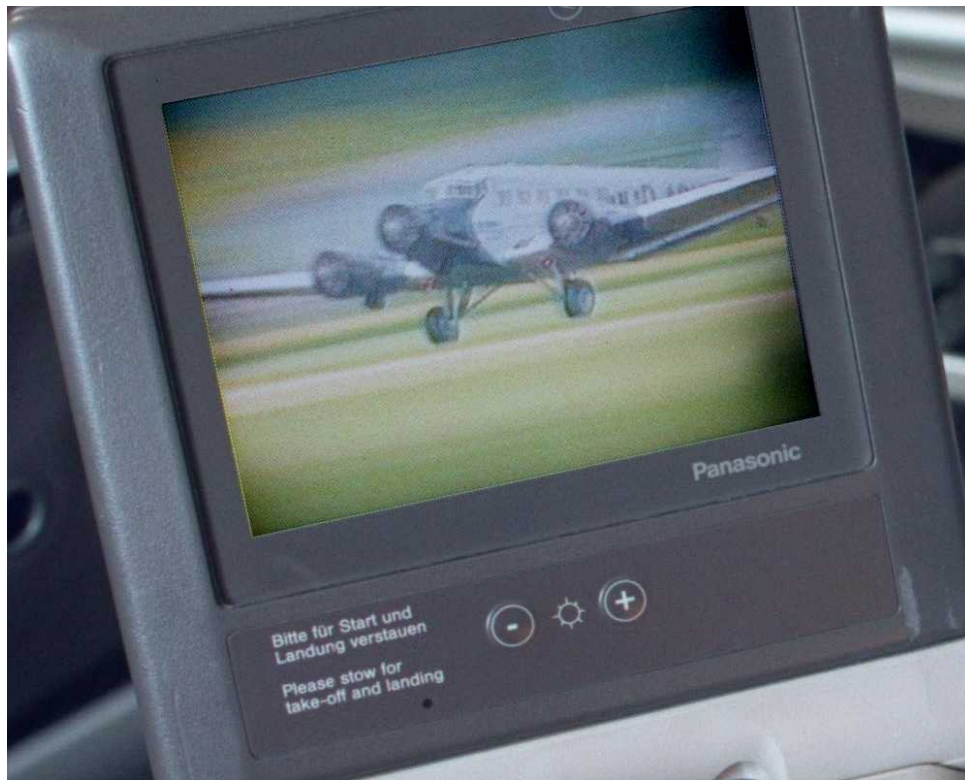


Figure 6.4: The InFlight seat console.

- Chuck knows his vast flying experience has earned him some shortcuts, but he does not mind investing a little effort into remembering those special commands.
- Marie is similar to Chuck, and both would be annoyed by time-consuming training screens for new users.
- Clevis is the golden nugget, the primary persona. A menu bar or dialogue box would instantly lose Clevis. With arthritis, any complex manipulation is out of the question. An interface designed for Clevis will be acceptable to all the others, as long as their special needs are accommodated somewhere in the interface.

Case Study: Designing for Clevis

Clevis can not and will not “navigate”, so there can be only one screen:

- Horizontal scrolling panoply of film posters and album covers.
- A large rotating knob (a “data wheel”) physically below the screen, which can be spun like a radio wheel.
- Clevis views the posters as if strolling past, no need to even think in terms of film categories.
- Navigation bar across bottom of screen, feedback where we are and with jump scrolling for Chuck.

See Figure 6.5.



Figure 6.5: The InFlight final design.

6.2 Video: Illustrating the Persona Design Process

- A 3d animation to illustrate the persona design process.
- Created by Keith Andrews and Ines Legnar in 2012.
- A set of 14 blue spheres represent real users.
- They are placed within three (exemplary) dimensions, according to the characteristics age, educational level, and computer experience.
- The users form a “point cloud”.
- A persona in the middle of the point cloud would represent the "average" user and would not make a good design target.
- Good candidate personas tend to represent the edge cases of user characteristics around the edges of the point cloud.
- Personas are given photographs to make them more believable.
- Finally, personas are divided into primary and secondary personas.

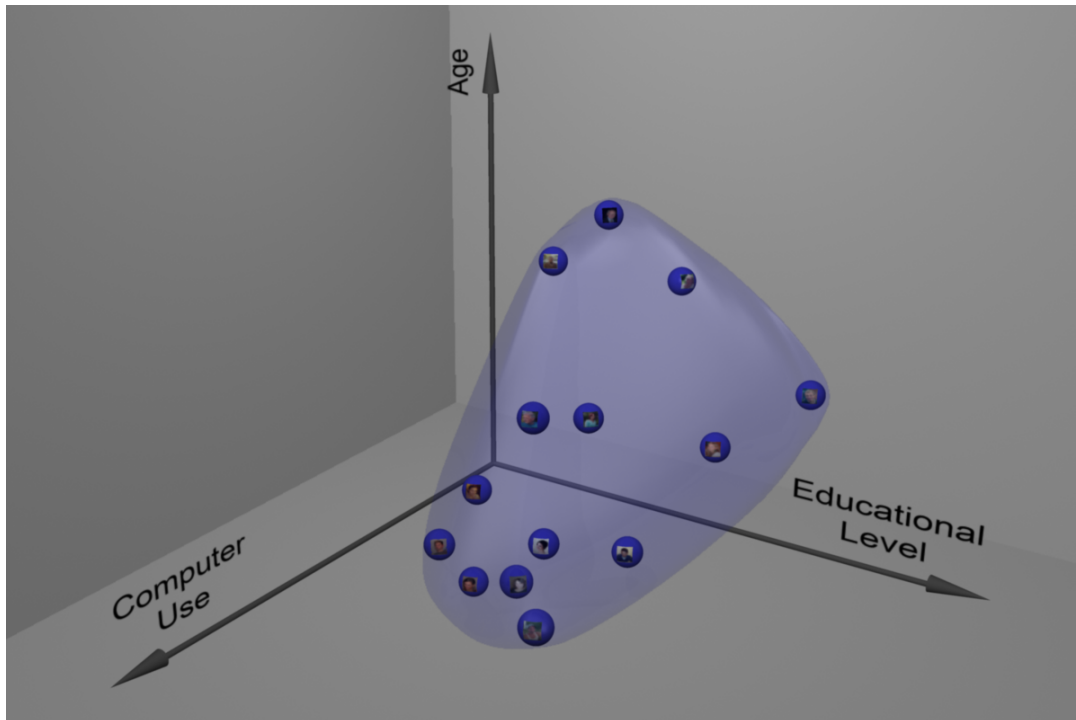


Figure 6.6: Users form a point cloud.

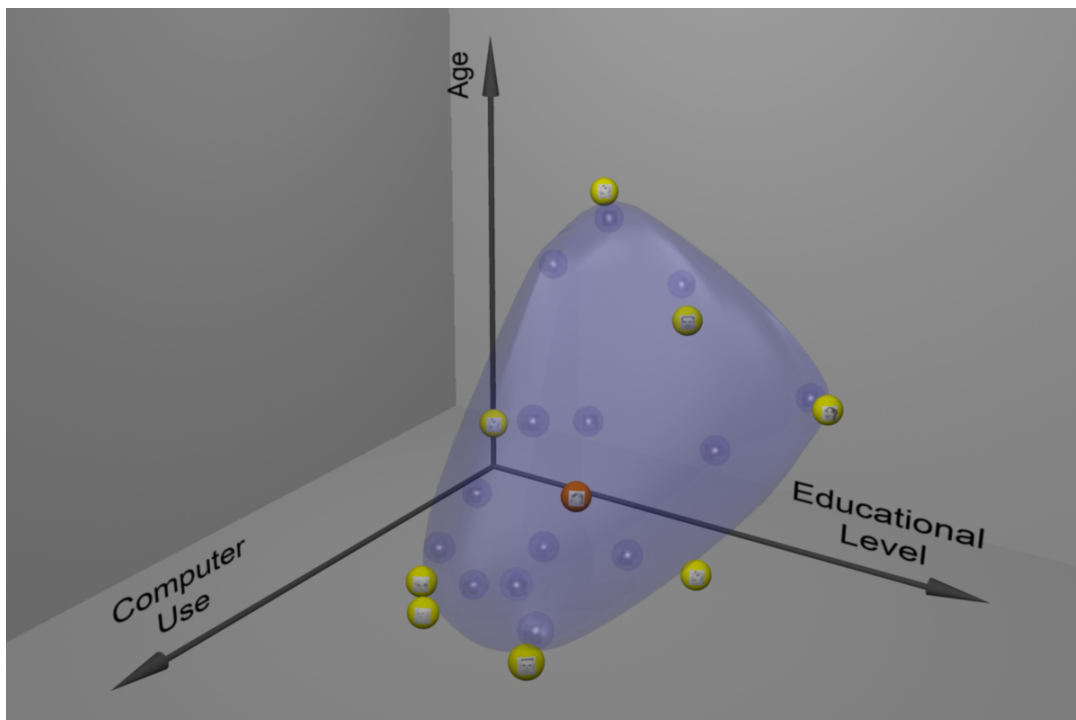


Figure 6.7: The average user and personas.

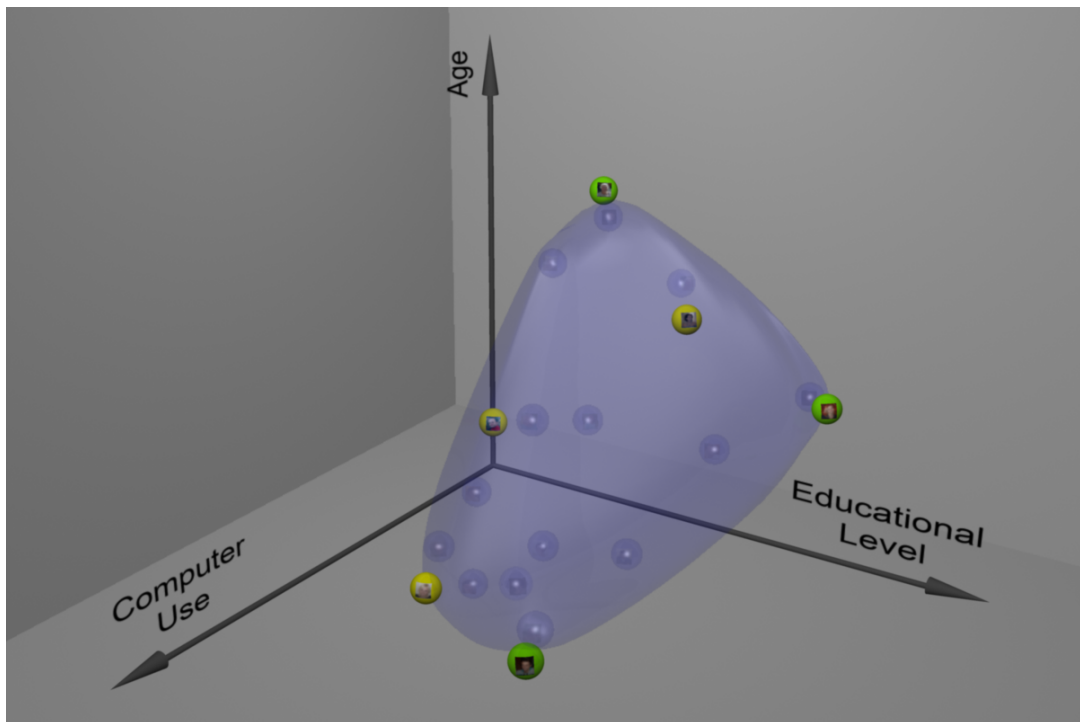


Figure 6.8: Primary personas.

6.3 Defining Goals for each Persona

Goals and personas co-exist. A persona exists to achieve his goals, a goal exists to give meaning to a persona.

Define the goals of each persona.

6.4 Defining Scenarios for each Persona

A scenario is a precise description of a persona using an interface to achieve a goal.

- **Daily Use Scenarios:** the primary actions the user will perform. These need the most robust design.
- **Necessary Use Scenarios:** More occasional, infrequent actions, which are necessary from time to time.
- **Edge Case Scenarios:** Loved by programmers, these can largely be ignored during the design process.

As the design progresses, play act the personas through the scenarios to test the validity of the design.

6.5 Moving to a Design Solution

Parallel Design

- If time and resources allow, explore design alternatives.

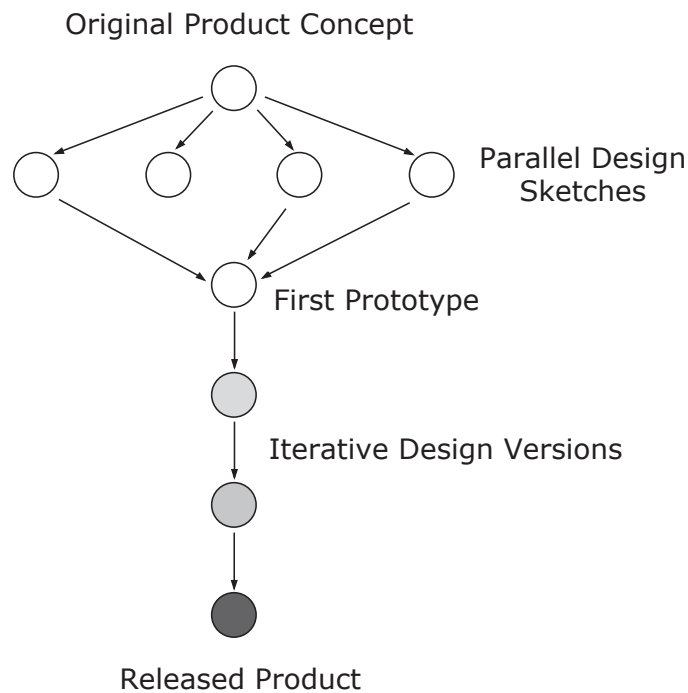


Figure 6.9: The relationship between parallel and iterative design. The first prototype is based on ideas from parallel design sketches. From Figure 8 of [Nielsen, 1993b].

- Have several design teams work *independently*, then compare draft designs (see Figure 6.9).

Brainstorm

- Meet away from usual workplace (different building, hut in the mountains).
- Brainstorm with mixed team (engineers, graphic designer, writer, marketing types, usability specialist, one or two representative users).
- Use plenty of paper. Cover the walls with it!
- Draw. Scribble. Use lots of coloured pens.
- Be stupid.
- Go crazy, build on the insane, think laterally (see Figure 6.10).
- Three rules during brainstorming:
 1. No one is allowed to criticise another's ideas.
 2. Programmers must not say it can't be implemented.
 3. Graphic designers must not laugh at programmers' drawings.
- Only *after* brainstorming, organise ideas and consider their practicality and viability.

Adapted from [Tognazzini, 1992], page 67.

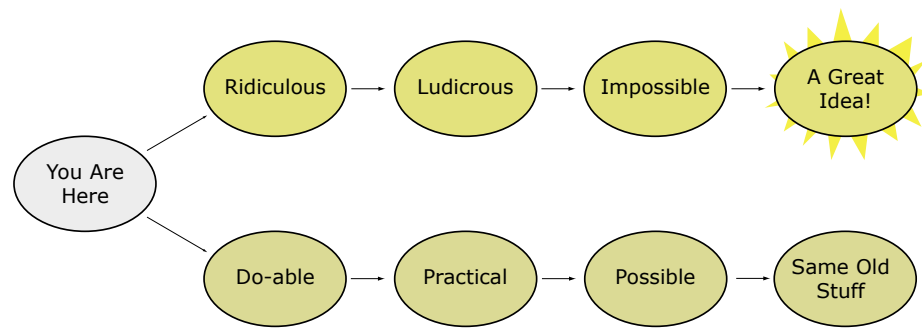


Figure 6.10: Lateral Thinking. Build on the crazy to generate new ideas [from Edward de Bono]

Four Techniques for Getting Unstuck

- **Pretend It's Magic:** If it were magic, it would do X. If we can not do X, how close can we get?
- **Pretend It's Human:** What response would the persona expect from a human?
- **Getting Another Brain:** 15-minute rule. If a team is stuck for more than 15 minutes, ask another designer for help.
- **Renaming:** Sometimes, a lack of common vocabulary is the problem. In this case, give elements new names in a silly theme (types of cheese, mountain ranges), define what they mean, and give them real names later.

Pretend It's Magic

What would these things do if they were magic?

- TV/VCR entertainment system
- Telephone/voicemail
- Calendar software
- Microsoft Windows

6.6 Getting Ideas from Your Users

Quick Surveys for Menu Organisation

A simple user survey can quickly tell you where users would expect to find certain functionality.

Figure 6.11 shows a survey to place word processing functions.

[This is, in fact, a quick and dirty version of a closed card sorting test.]

6.7 Follow Conventions

Following the standard way of doing things is generally good:

Under which pull-down menu would you most expect to find each of the word processing tasks below.

File Edit View Insert Format Utilities Macro Window

1. Search for a word: _____
2. Create a header: _____
3. Create an index entry: _____
4. Set up your document preferences: _____
5. Repeat an annotation: _____
6. Paste another file into your current file: _____
7. Use italics: _____
8. Show summary information about your document: _____
9. Switch to another document: _____
10. Show all annotations in the document: _____
11. Repaginate the document: _____
12. Set up your printer: _____

Figure 6.11: A quick user survey on menu organisation.

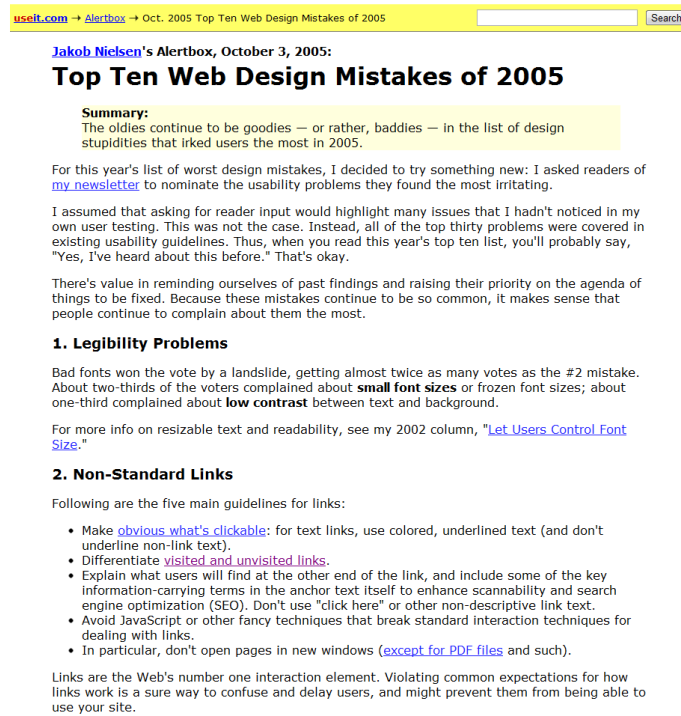


Figure 6.12: Jakob Nielsen's web site useit.com uses the conventional link style of blue underlined for unvisited links and purple underlined for visited links [Nielsen, 2005].

- Users can then *transfer knowledge* as they move between applications.
- Only break the convention for good reason.

Link Styles

- Users should not have to guess where they can click (or have to scan the screen with the mouse to see what is clickable).
- For textual links, use coloured and underlined text (and do *not* underline non-link text).
- Distinguish between visited and unvisited links.
- By convention, unvisited links are blue underlined, visited links are purple underlined. This is a good default choice. See Figure 6.12.
- If you do choose to use a different link style, at least use it *consistently* across your site. See Figure 6.13.

An Asterisk Means a Required Field

- In the context of a web form, a field marked with an asterisk (*) has come to mean a *required* field.
- Do not change this meaning arbitrarily, as Avis did on their web site in 2005, shown in Figure 6.14.
- These three fields were the only three optional fields on their site.

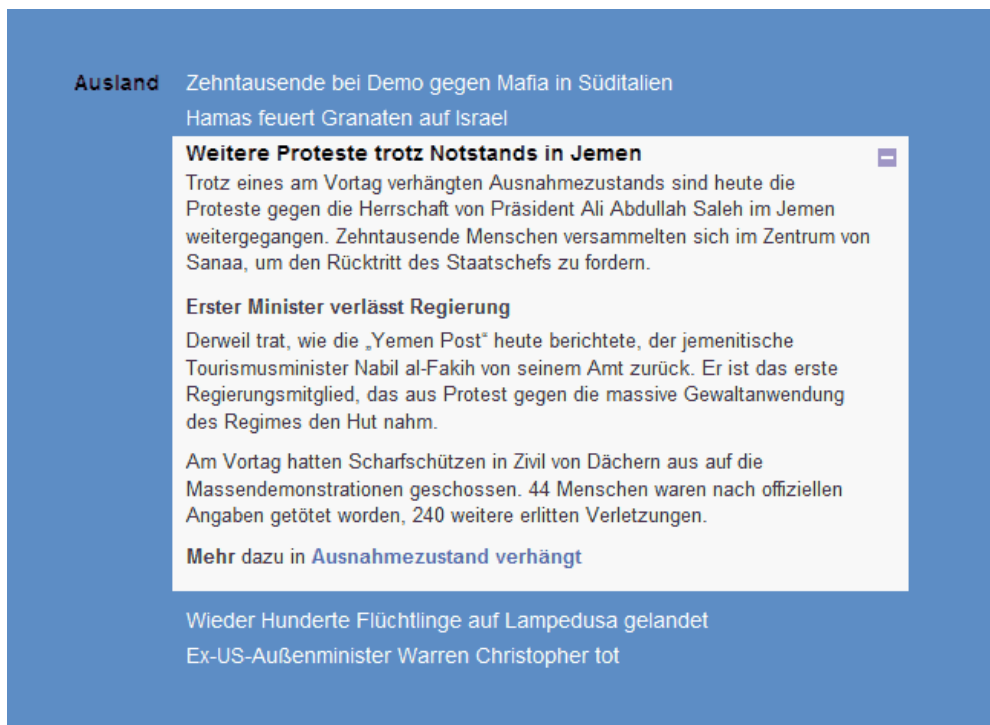


Figure 6.13: Non-Standard and inconsistent link styles at orf.at. There are three different styles for links: white, bold black, and bold blue-grey. Bold black is used inconsistently within itself: the first bold black text (Ausland) is not clickable, the second (Weitere Proteste...) is, the third (Erster Minister...) is not. Users can no longer predict what is a link and what is not.


- Rather than place asterisks meaning *required* next to dozens of other fields (a terrible waste of asterisks), Avis' web designers thought it reasonable to redefine the meaning of an asterisk to indicate an *optional* field.
- Breaking the convention caused unnecessary confusion to their users.
- Avis now (11 Mar 2011) follows the conventional meaning and uses an asterisk to indicate a *required* field, as shown in Figure 6.15.
- See Jared Spool's article [[Spool, 2005](#)].

Platform Conventions

Platform conventions are the accepted way of doing things on a particular platform:

- Macintosh Human Interface Guidelines, Apple Computer, Jan. 1993 [[Apple Computer, 1992](#)].
- Microsoft Windows User Experience, Microsoft Press, September 1999 [[Microsoft, 1999](#)].
- Java Look and Feel Design Guidelines, Sun Microsystems, July 1999 [[Sun Microsystems, 1999](#)].

They are often supported by *widget libraries* of shared code.



Personal Data

To sign up for My Avis and the Wizard program at once, simply complete and submit the following personal profile information. Upon completion, you'll receive a Wizard Member Number and be registered for My Avis Service. **Please note that information marked with an asterisk (*) is optional.**

Contact Information

Title:

First Name:

* Middle Initial:

Last Name:

* Company Name:

Address 1:

* Address 2:

City:


State:

Zip / Postal Code **(Please do not use any spaces or dashes):**

Country:

Telephone Number:

Figure 6.14: In 2005 the Avis web site used an asterisk to indicate an *optional* field, breaking the convention that an asterisk means *required*. This caused confusion to their users. [Image used with kind permission of Jared Spool [Spool, 2005]]



REGISTER FOR MY AVIS - STEP 1 of 2: Your Information

Simply complete and submit the following personal profile information. Upon completion, you'll receive an Avis Wizard Number and be registered for My Avis Service.

Your information is secure. [Privacy Policy](#)

IDENTITY INFORMATION All required fields have an asterisk. *

Full Name: *

Create Username: *

Password (cAsE sEnSiTiVe): * Must use 6 or more letters & numbers.

Retype Password: *

Password Reminder: *

Reminder Answer: *

Date of Birth: *

CONTACT INFORMATION

Country: *

Address Line 1: *

Address Line 2:

City: *

State / Province: *

Zip / Postal: *

Telephone: *

Email Address: *

Retype Email Address: *

Notes: A valid email address is required to register with Avis. We will use this email address to send an enrollment confirmation email.

Figure 6.15: Avis now (11 Mar 2011) follows the conventional meaning and uses an asterisk to indicate a *required* field.

Design Patterns

Look for *design patterns* for your particular situation: platform-independent tried-and-tested solutions to common design problems (for example, the shopping cart design pattern for an e-commerce web site).

Chapter 7

Prototyping

“ There’s a mantra at IDEO: “Never go to a meeting without a prototype.” At whatever stage of development, one week, one month, or 6 months. ”

[Tim Brown, President, IDEO, speaking at CHI 2004 in Vienna.]

Perform usability evaluation and obtain feedback as *early* as possible in the design cycle by building and evaluating prototypes. Finally, throw prototypes away and implement final design.

References

- ++ Carolyn Snyder; *Paper Prototyping*; Morgan Kaufmann, 2003. ISBN 1558608702 (com, uk) [Snyder, 2003]
- + Bill Buxton; *Sketching User Experiences*; Morgan Kaufmann, 2007. ISBN 0123740371 (com, uk) [Buxton, 2007]
- o Marc Rettig; *Prototyping for Tiny Fingers*; CACM, 37(4), 1994. [Rettig, 1994].
- o Carolyn Snyder; *Using Paper Prototypes to Manage Risk*; Software Design and Publisher Magazine, 1996. http://www.uie.com/articles/prototyping_risk/
- o Snyder and Lahti; *Paper Prototyping Tricks*; 12 min. videotape. User Interface Engineering, 1995. <http://www.uie.com/>
- o Jakob Nielsen; *Paper Prototyping: A How-To Training Video*; 32 min. video (DVD). Nielsen Norman Group, 2003. <http://www.nngroup.com/reports/prototyping/>

Online Resources

- o *Paper Prototypes: Still Our Favorite*; User Interface Engineering, 2000. http://www.uie.com/articles/paper_prototyping/
- o Matthew Klee; *Five Paper Prototyping Tips*; User Interface Engineering, 2000. http://www.uie.com/articles/prototyping_tips/

7.1 Types of Prototype

In increasing order of complexity:

- **Verbal Prototypes:** textual description of choices and results.
- **Paper Prototypes:**
 - *Low-Fidelity:* hand-drawn sketches.
 - *High-Fidelity:* more elaborate printouts.
- **Interactive Sketches:** interactive composition of hand-drawn sketches.
- **Working Prototypes:** interactive, skeleton implementation.

Then, throw prototypes away and implement final design.

7.2 Verbal Prototype

Simple textual description of choices and results.

7.3 Low-Fidelity Paper Prototypes

- Paper prototypes simulate screen and dialogue elements on paper.
- First hand-drawn sketches (lo-fi), later perhaps more elaborate printouts (hi-fi).
- Early usability feedback with throwaway designs: *maximum feedback for minimum effort!*
- Greeking (drawing squiggly lines) is used to represent text which would otherwise be a distraction.
- Figures 7.1 and 7.2 show paper and working prototypes for an online radio station.

Paper Prototype of Customer Information Terminal

- Design interface for customer information terminal to be placed in branches of the (fictional) Northumberland Bank.
- Specifically prototype the loans section: tasks include information about loans and calculation of loan repayments.
- Figures 7.3, 7.4, and 7.5 show three of the paper prototypes developed.
- Thanks to Cliff Brown, Northumbria University, for permission to use these images.

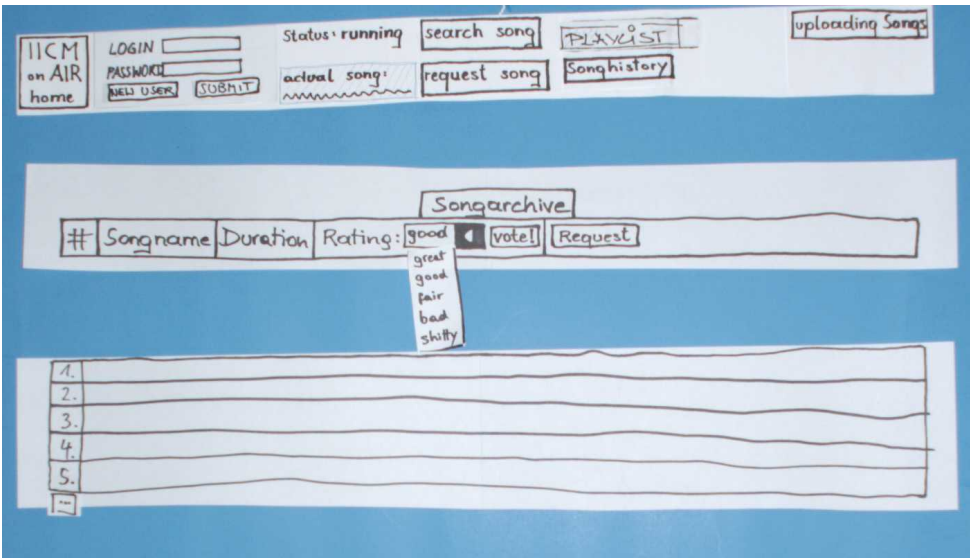


Figure 7.1: Paper prototype of IICM on Air.

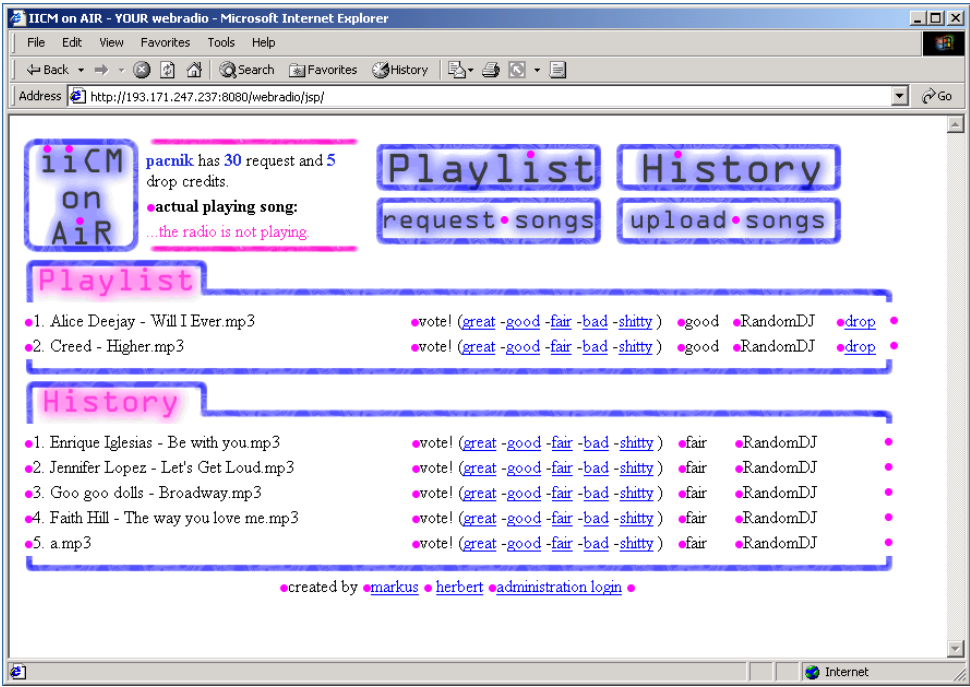


Figure 7.2: Working prototype of IICM on Air.

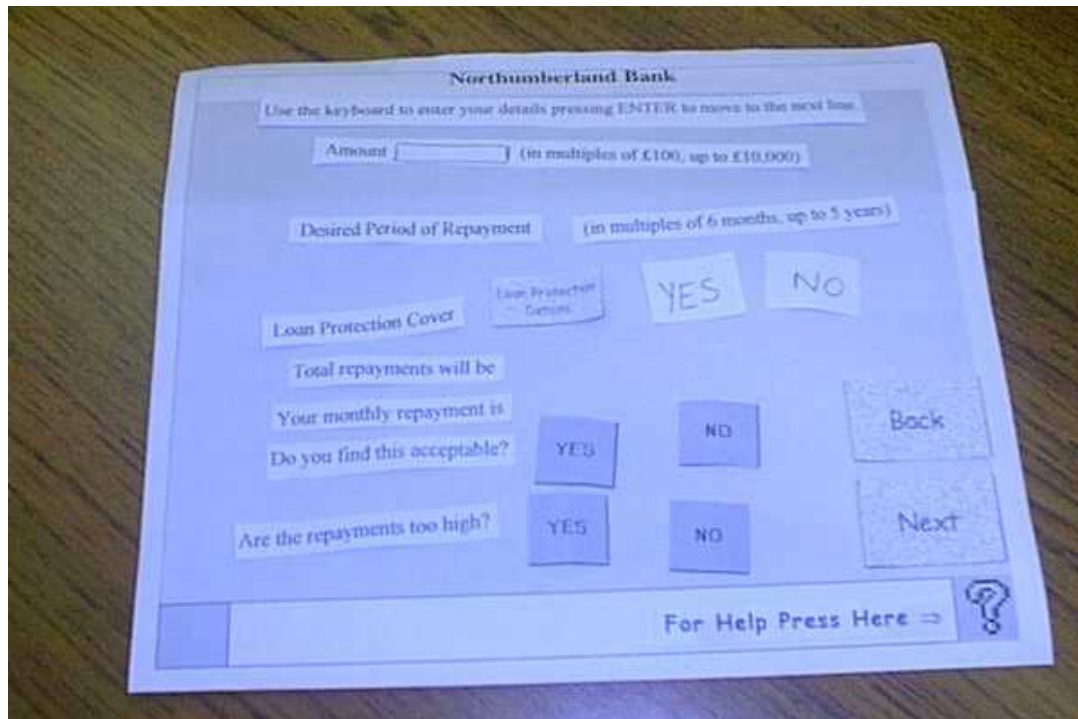


Figure 7.3: Paper prototype 1 for a customer information terminal. [Image used with kind permission of Cliff Brown, Northumbria University.]



Figure 7.4: Paper prototype 2 for a customer information terminal. [Image used with kind permission of Cliff Brown, Northumbria University.]

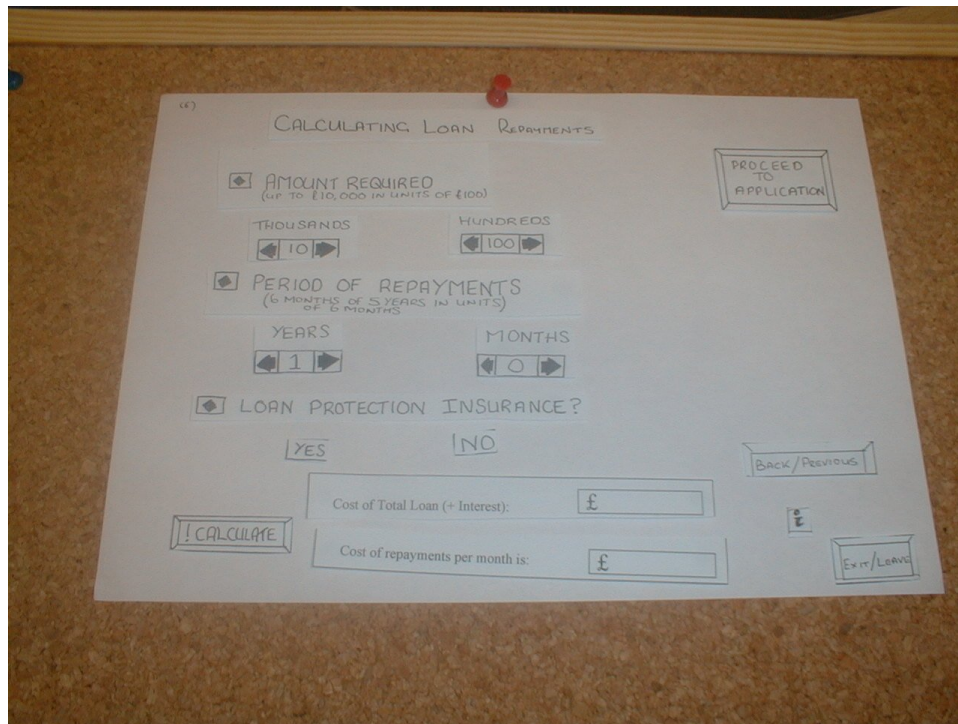


Figure 7.5: Paper prototype 3 for a customer information terminal. [Image used with kind permission of Cliff Brown, Northumbria University.]

7.4 High-Fidelity Paper Prototypes

- Elaborate screen designs created with drawing editors such as Adobe Illustrator or Corel Draw.
- Printed out in colour.
- The often look too much like a finished design, and not enough like a prototype.
- Users tend to comment on the choice of fonts and colours, rather than the flow through the application.

7.5 Interactive Sketches

- Scan in hand-drawn interface sketches.
- Assemble interactive prototype with clickable elements (say with Macromedia Director).
- Retains throwaway, casual look to encourage criticism and discussion, as shown in Figure 7.6.
- For example, Chris Edwards' Sketchy Thing [Edwards, 1999] and Marc Rettig's Interactive Sketch [Rettig, 1999]

7.6 Working Prototypes

- *Simple algorithms*: ignore special cases.

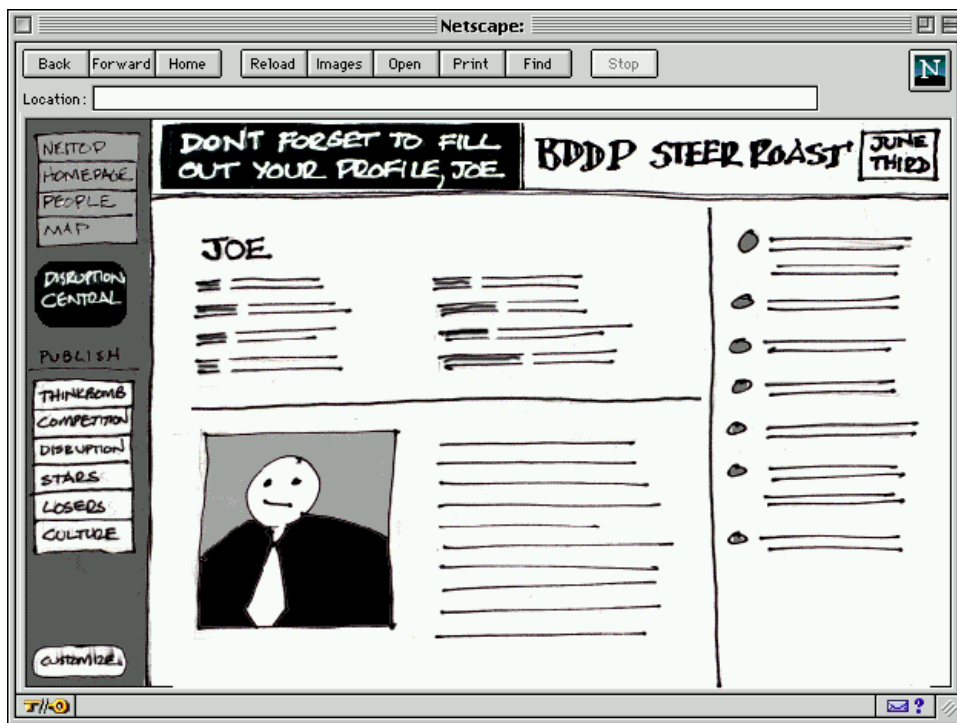


Figure 7.6: An interactive sketch made in Shockwave. Screen designs sketches are scanned and assembled into an interactive prototype with Macromedia Director. Screenshot taken from Chris Edwards' Sketchy Thing [Edwards, 1999].

- *Fake data*: similar data, images instead of video, etc.
- *Wizard of Oz*: human expert operating behind the scenes to simulate interface responses.

Prototyping Tools

- DENIM; <http://dub.washington.edu/denim/>
- Axure; <http://www.axure.com/>
- Balsamiq Mockups; <http://balsamiq.com/>
- Microsoft Expression SketchFlow;
http://www.microsoft.com/expression/products/Sketchflow_Overview.aspx
- HotGloo <http://www.hotgloo.com/>

Dimensions of Working Prototypes

Working prototypes cut down on either the number of features, or the depth of functionality of features:

- *Vertical Prototype*: in-depth functionality for a few selected features.
- *Horizontal Prototype*: full interface features, but no underlying functionality.
- *Scenario Prototype*: only features and functionality along the specific scenarios or paths through the interface which are to be evaluated.

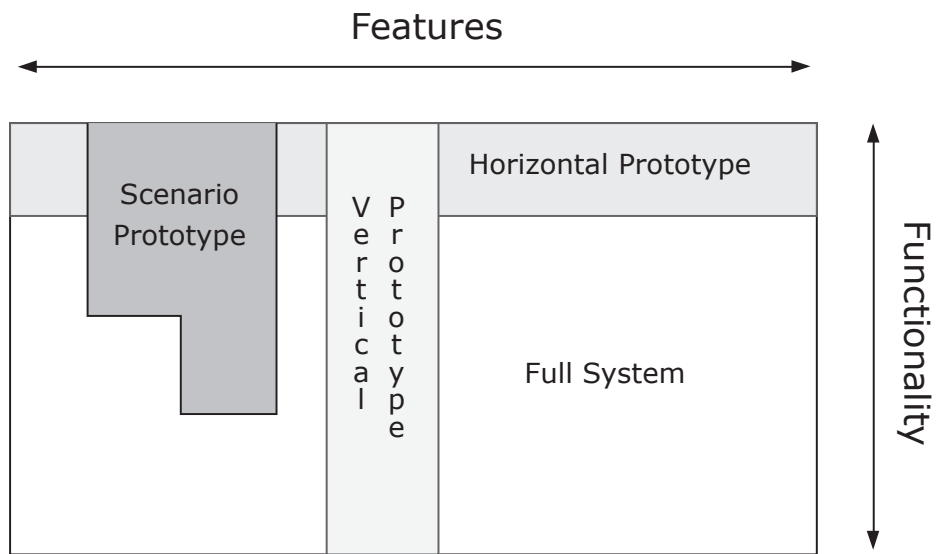


Figure 7.7: Working prototypes vary according to the breadth or depth of features implemented.
From Figure 9 of [Nielsen, 1993b].

These varieties of prototype are illustrated in Figure 7.7.

7.7 Implementation

Implement final design.

Competitive analysis of software components:

- Use existing interface framework as far as possible (Motif, MS-Windows, Java Swing) – saves a *lot* of work.
- Use existing components and applications rather than re-inventing the wheel.

Chapter 8

Usability Inspection Methods

Inspection of interface design using *heuristic* methods (based on analysis and judgement rather than experiment).

1. *Heuristic Evaluation*: A small team of evaluators inspects an interface using a small checklist of general principles and produces an aggregate list of potential problems.
2. *Guideline Checking*: An evaluator checks an interface against a detailed list of specific guidelines and produces a list of deviations from the guidelines.
3. *Cognitive Walkthrough*: A small team walks through a typical task in the mind set of a novice user and produces a success or failure story at each step along the correct path. [analyses learnability]
4. *Guideline Scoring*: An evaluator scores an interface against a detailed list of specific guidelines and produces a total score representing the degree to which an interface follows the guidelines.
5. *Action Analysis*: An evaluator produces an estimate of the time an expert user will take to complete a given task, by breaking the task down into ever smaller steps and then summing up the atomic action times. [analyses efficiency]

References

- Jakob Nielsen and Robert L. Mack (Eds.); *Usability Inspection Methods*; John Wiley, 1994. ISBN 0471018775 (com, uk) [Nielsen and Mack, 1994]

Would You Use Untested Software?

Would you knowingly use untested software?

- How many of you have written programs that are *used* by other people?
- How many of you have watched or observed users using your software?
- How many of you actually evaluated or tested your interface before it was used?
- In practice, most software developers do not actually conduct *any* kind of usability evaluation [due to perceived expense, lack of time, lack of expertise, lack of inclination, or lack of tradition].

8.1 Heuristic Evaluation

First described in [Nielsen and Molich, 1990].

Small team of evaluators (usually usability specialists) systematically checks interface design against small set of recognised usability principles (the “heuristics”).

Online Resources

- Jakob Nielsen; *Heuristic Evaluation*; <http://nngroup.com/topic/heuristic-evaluation/>

Usability Heuristics

Revised list of usability heuristics from [Nielsen, 1994]. The category names in brackets are the original names of the corresponding categories from [Nielsen and Molich, 1990], in those cases where they are different.

1. **Visibility of System Status** [Feedback]

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

For example: busy cursor [1–10s], progress indicator [>10s].

2. **Match Between System and the Real World** [Speak the Users’ Language]

The system should speak the users’ language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

Match users’ mental model. Beware of misleading metaphors.

3. **User Control and Freedom** [Clearly Marked Exits]

Users often choose system functions by mistake and will need a clearly marked “emergency exit” to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

4. **Consistency and Standards** [Consistency]

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

5. **Error Prevention**

Even better than good error messages is a careful design which prevents a problem from occurring in the first place.

For example: select file from menu rather than typing in name, confirmation before dangerous actions, beware of modes, avoid similar command names, warning if Caps Lock is activated when entering a password, etc.

6. **Recognition rather than Recall**

Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Users' short-term memory is limited. Provide examples, default values, easily retrievable instructions.

7. **Flexibility and Efficiency of Use** [Accelerators]

Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

For example: abbreviations, command keys, type-ahead, edit and reissue previous commands, menu of most recently used files, macros.

8. **Aesthetic and Minimalist Design** [Minimalist Design]

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. "Less is more"

9. **Help Users Recognise, Diagnose, and Recover from Errors** [Good Error Messages]

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Phrase error messages *defensively*, never blame the user. Multilevel messages. Link to help system.

10. **Help and Documentation**

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Liberal use of examples.

Limits on Response Times

- **0.1 sec.:** is the limit so that the system appears to react instantaneously.
Important for direct manipulation, virtual world navigation.
- **1 sec.:** is the limit so that the user's flow of thought stays uninterrupted.
Display a busy cursor if things will take longer than 1 sec.
- **10 secs.:** is the limit for keeping the user's attention on the task at hand.
Display a progress indicator if things will take longer than 10 secs.

From Section 5.5 of [Nielsen, 1993b] and [Nielsen, 1993a].

Performing a Heuristic Evaluation

- Design may be verbal description, paper mock-up, working prototype, or running system. [when evaluating paper mock-ups, pay special attention to missing dialogue elements!]
- Provide evaluators with checklist of usability heuristics.
- Optionally provide evaluators with some domain-specific training.
- Each evaluator works *alone* (\approx 1–2 hours).

<i>Evaluation Name</i>	<i>Number of Evaluators</i>	<i>Total Known Problems</i>	<i>Average No. Problems Found per Evaluator</i>
Teledata	37	52	51%
Mantel	77	30	38%
Savings	34	48	26%
Transport	34	34	20%

Table 8.1: The average number of problems found by individual novice evaluators in each of four heuristic evaluations. [Data extracted from Nielsen and Molich [1990, Table 2].]

- Interface examined in two passes: first pass focuses on general flow, second on particular elements in detail.
- Notes taken either by evaluator or evaluation manager.
- Make list of potential problems and list of positive findings.
- Take screen shots (in PNG) as you work - they may not be reproducible later.
- Independent findings are then aggregated into one large list (by evaluation manager). This is best done with a spreadsheet.
- The large list of potential problems is distributed to each evaluator.
- Each evaluator now assigns severity ratings individually to each problem in the large list (unseen by the other evaluators).
- The individual severity ratings are averaged to obtain the final severity rating for each problem.
- The long list is sorted in decreasing order of average severity. Now you know why a spreadsheet comes in handy.
- Group debriefing session to suggest possible redesigns.

How Many Problems are Found?

Nielsen and Molich [1990] ran four heuristic evaluations using “usability novices” as evaluators.

They compared the average number of problems found by each evaluator with the total number of known problems in each system, as shown in Table 8.1.

Aggregated Evaluations

- Individual evaluators found relatively few problems.
- Aggregating the evaluations (merging the problem lists) of several individuals produced much better results.
- See Table 8.2 and Figure 8.1.
- Group debriefing session to suggest possible redesigns.

<i>Aggregate:</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>5</i>	<i>10</i>
Teledata	51%	71%	81%	90%	97%
Mantel	38%	52%	60%	70%	83%
Savings	26%	41%	50%	63%	78%
Transport	20%	33%	42%	55%	71%

Table 8.2: The average proportion of usability problems found by various sized aggregates of novice evaluators in each of the four heuristic evaluations. [Data extracted from Nielsen and Molich [1990, Table 4].]

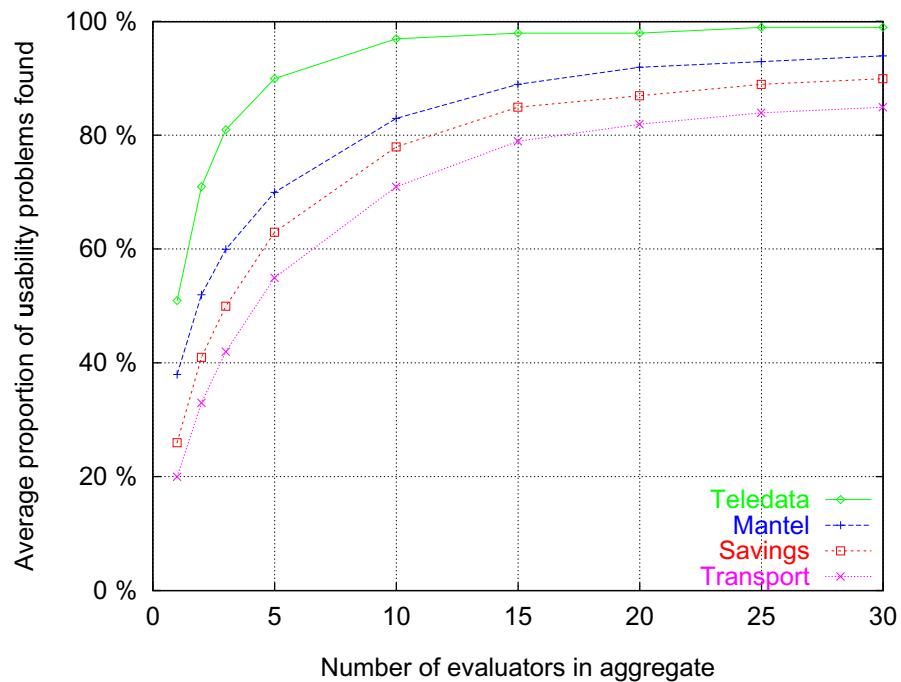


Figure 8.1: Average proportion of usability problems found by aggregates of size 1 to 30. [Results from Nielsen and Molich [1990, Figure 4].]

Experience of Evaluators

The experience of evaluators influences how many problems they find [Nielsen, 1992].

Study of one interface, the Banking System, a touch tone “voice response” telephone banking system, by 3 groups of evaluators:

- 31 “novice” evaluators: computer science students with no formal knowledge of UI or usability (no usability expertise).
- 19 “regular” specialists: people with UI and usability experience, but no expertise in voice-response systems (usability expertise).
- 14 “double” specialists: people with expertise both in usability and in telephone-operated interfaces (usability and domain expertise).

Task: transfer \$1000 from savings account to check account.

Sample Banking System Dialogue

Figure 8.2 shows the sample dialogue given to evaluators in the study.

- This dialogue actually took place!
- The problem was that the user did not have authority to make phone transfers.
- The results are summarised in Table 8.3.
- Aggregating the results of individual evaluators gives the plots shown in Figure 8.3.

Conclusions from Studies

From these two studies, we can conclude that:

- Single evaluator finds only small subset of potential problems.
- Several evaluators should work *independently*, then aggregate results.
- Evaluators pay more attention to major problems without neglecting minor ones.
- In the second study [Nielsen, 1992], for the Banking System interface, finding around 80% of usability problems required 2–3 double specialists, 3–5 regular usability specialists, and around 15 novice usability evaluators.
- This result does *not* generalise to all interfaces in all situations.

Heuristic Evaluation in Practice

When heuristic evaluation is used in standard industry practice today:

- It is often called an *expert review*.
- It is typically performed by 3–4 usability specialists.
- It might cost around € 2000 (see Section 10.5).

[First there is a short dialogue in which the user is identified by entering an identification number and access code – this is not part of the evaluation exercise]

1. S: *Enter one for account information, three for transfers between your own accounts, ...*
2. U: 3# {the user interrupts the system}
3. S: *Enter account to transfer from.*
4. U: 1234567890# {savings account number}
5. S: *Enter account to transfer to.*
6. U: # {an abbreviation for the checking account}
7. S: *Enter amount in cents.*
8. U: 100000#
9. S: *From account number twelve thirtyfour fiftysix seventyeight ninety to account number primary account, a transfer of one thousand dollars is to be made. Press one to confirm, zero to cancel.*
10. U: 1#
11. S: *You do not have access to this function.*

Figure 8.2: The sample Banking System dialogue used for the study. [Extracted from Nielsen [1992, Figure 1].]

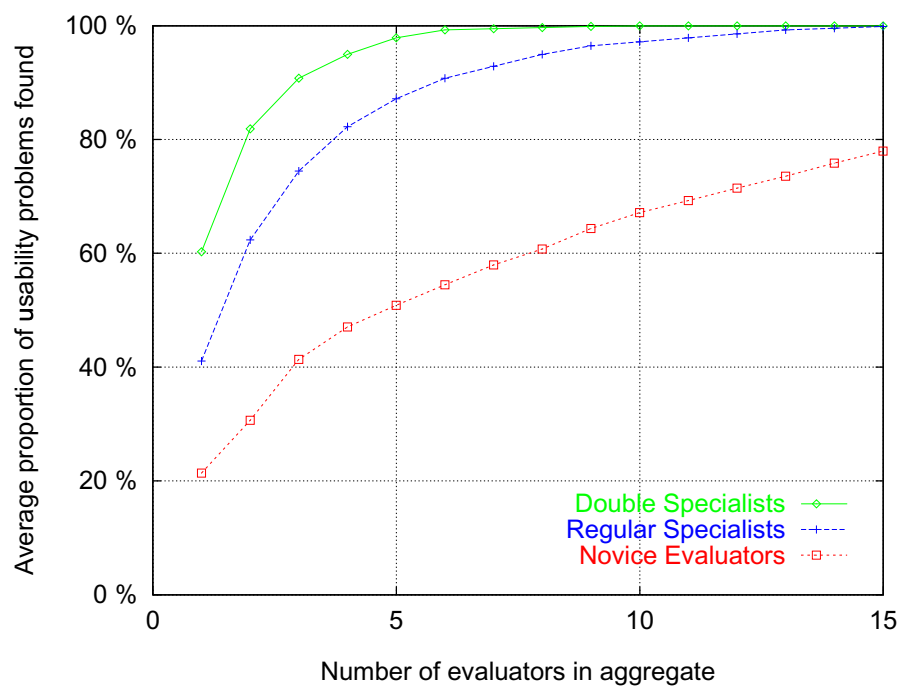


Figure 8.3: Average proportion of usability problems found by aggregates of novice evaluators, regular specialists, and double specialists. [Results from Nielsen [1992, Figure 2].]

<i>No.</i>	<i>Problem</i>	<i>Novice</i>	<i>Regular</i>	<i>Double</i>
<i>Major Usability Problems</i>				
1.	Error message appears too late.	68%	84%	100%
2.	Do not require dollar amount to be entered in cents.	68%	74%	79%
3.	The error message is not precise.	55%	63%	64%
4.	The error message is not constructive.	6%	11%	21%
5.	Replace term “primary account” with “checking account”.	10%	47%	43%
6.	Let users choose account from a menu.	16%	32%	43%
7.	Only require a # where it is necessary.	3%	32%	71%
8.	Give feedback as name of chosen account.	6%	26%	64%
Average for major problems		29%	46%	61%
<i>Minor Usability Problems</i>				
9.	Read menu item description before action number.	3%	11%	71%
10.	Avoid gap in menu numbers between 1 and 3.	42%	42%	79%
11.	Provide earlier feedback.	42%	63%	71%
12.	Replace use of 1 and 0 for accept and reject with # and *.	6%	21%	43%
13.	Remove the field label “number” when no number is given.	10%	32%	36%
14.	Change prompt “account” to “account number”.	6%	37%	36%
15.	Read numbers one digit at a time.	6%	47%	79%
16.	Use “press” consistently and avoid “enter”.	0%	32%	57%
Average for minor problems		15%	36%	59%
Average for all problems		22%	41%	60%

Table 8.3: Proportion of novice, specialist, and double specialist usability evaluators finding problems in the Banking System. [Results from Nielsen [1992, Table 1].]

Pros and Cons of Heuristic Evaluation

- ++ cheap
- + intuitive
- + usable early in development process
- + finds many problems
- + finds both major and minor problems
- may miss domain-specific problems

8.2 Severity Ratings

Severity ratings can help prioritise the fixing of usability problems.

- After evaluation sessions, a complete aggregate list of usability problems is given/sent to each evaluator.
- Working independently, evaluators assign severity rating [on scale of 0–4] to each problem (≈ 30 mins.).
- Severity rating of single evaluator is unreliable, mean of 3–5 evaluators is satisfactory.

See the discussion of severity ratings in [Nielsen and Mack, 1994], pages 47–55.

Five-Point Severity Scale

<i>Score</i>	<i>Severity</i>	<i>Fix Priority</i>
4	catastrophic problem	imperative
3	major problem	high
2	minor problem	low
1	cosmetic problem only	
0	not a problem at all	

Order of Criticality

To explicitly take problem frequency into account, assign *criticality* ratings.

Criticality = Severity Ranking + Frequency Ranking

<i>Severity</i>	<i>Frequency</i>
4 catastrophic	4 >90%
3 major	3 51–89%
2 minor	2 11–50%
1 cosmetic	1 1–10%
0 none	0 <1%

1.3 DATA ENTRY: Text

1.3/1 Adequate Display Capacity

Ensure that display capacity, i.e., number of lines and line length, is adequate to support efficient performance of text entry/editing tasks.

Example: For text editing where the page format of subsequent printed output is critical, the user's terminal should be able to display full pages of text in final output form, which might require a display capacity of 50-60 lines or more.

Example: For general text editing where a user might need to make large changes in text, i.e., sometimes moving paragraphs and sections, a display capacity of at least 20 lines should be provided.

Example: Where text editing will be limited to local changes, i.e., correcting typos and minor rewording, as few as seven lines of text might be displayed.

Comment: A single line of displayed text should not be used for text editing. During text editing, a user will need to see some displayed context in order to locate and change various text entries. Displaying only a small portion of text will make a user spend more time moving forward and back in a displayed document to see other parts, will increase load on the user's memory, and will cause users to make more errors.

Reference:

Elkerton Williges Pittman Roach 1982

Neal Darnell 1984

See also: 1.3/27

Table 8.4: One of the 944 guidelines by Smith and Mosier.

8.3 Guideline Checking

Guidelines ... specific advice about usability characteristics of an interface.

- An evaluator checks an interface against a detailed list of specific guidelines and produces a list of deviations from the guidelines.
- Whereas heuristic evaluation employs 10 broad principles, guideline checking often involves dozens (or hundreds) of more specific individual items on a checklist.

Example Sets of Guidelines

- Sidney Smith and Jane Mosier; *Design Guidelines for Designing User Interface Software*; The MITRE Corp., 1986. [944 guidelines] ISBN 9992080418 (com, uk) <ftp://ftp.cis.ohio-state.edu/pub/hci/Guidelines>
- C. Marlin Brown; *Human-Computer Interface Design Guidelines*; Ablex, NJ, 1988. ISBN 0893913324 (com, uk) [302 guidelines]
- Deborah Mayhew; *Principles and Guidelines in Software User Interface Design*; Prentice-Hall, 1991. ISBN 0137219296 (com, uk) [288 guidelines]

Table 8.4 shows one of the 944 guidelines from Smith and Mosier.

Pros and Cons of Guideline Checking

- + cheap

- + intuitive
- + usable early in development process
- time-consuming
- can be intimidating – often hundreds or thousands of specific guidelines.

8.4 Cognitive Walkthrough

Task-oriented walkthrough of interface, imagining novice users' thoughts and actions. Focuses explicitly on learnability.

- Design may be mock-up or working prototype.
- Analogous to structured walkthrough in software engineering.
- Based on *cognitive model* (CE+) of human exploratory learning.

First described in [Lewis, Polson et al., 1990], discussion and practical guide in [Wharton et al., 1994].

Other References

- Clayton Lewis and John Rieman; *Task-Centered User Interface Design: A Practical Introduction*; 1993. <http://hcibib.org/tcuid/>
- John and Packer; *Learning and Using the Cognitive Walkthrough Method: A Case Study Approach*; Proc. CHI'95, May 1995. <http://www.acm.org/sigchi/chi95/proceedings/papers/bej1bdy.htm>
- Rick Spencer; *The Streamlined Cognitive Walkthrough Method*; Proc. CHI 2000, [Spencer, 2000]

Exploratory Learning

Rather than read manual or attend course, users often prefer to learn new system by “trial and error” → *exploratory learning* [Carroll and Rosson, 1987]:

1. Start with rough idea of task to be accomplished.
2. Explore interface and select most appropriate action.
3. Monitor interface reactions.
4. Determine what action to take next.

The CE+ Model of Exploratory Learning

Based on psychological studies, the CE+ model describes exploratory learning behaviour in terms of 3 components:

- **Problem-Solving Component**

User chooses among alternative actions based on similarity between the expected consequences of an action and the current goal.

After executing selected action, user evaluates system response and decides whether progress is being made toward the goal. A mismatch results in an undo → “hill-climbing”.

- **Learning Component**

When above evaluation process leads to positive decision, the action taken is stored in long-term memory as a rule.

- **Execution Component**

User first attempts to fire applicable rule matching current context. If none found, problem-solving component is invoked.

Cognitive Walkthrough Preparation

- a) Identify user population.
- b) Define suite of *representative* tasks.
- c) Describe or implement interface or prototype.
- d) Specify correct action sequence(s) for each task.

Cognitive Walkthrough Steps

For each action in solution path, construct *credible* “success” or “failure” story about why user would or would not select correct action.

Critique the story to make sure it is believable, according to four criteria:

- a) **Will the user be trying to achieve the right effect?**

What is users’ goal – will they *want* to select this action?

- b) **Will the user know that the correct action is available?**

Is control (button, menu, switch, triple-click, etc.) for action *apparent* (visible)?

- c) **Will the user know that the correct action will achieve the desired effect?**

Once users find control, will they *recognise* that it is the correct control to produce the desired effect?

- d) **If the correct action is taken, will the user see that things are going ok?**

After correct action, will users realise progress has been made towards the goal (feedback)?

Note that CW always tracks the *correct* action sequence. Once the user deviates from the correct path their further progress is no longer considered.

Group Walkthrough

- Performed by mixed team of analysts (designers, engineers, usability specialist).
- Capture critical information on three group displays (flip charts, overheads):
 1. User knowledge (prior to and after action).
 2. Credible success or failure story.
 3. Side issues and design changes.
- Perhaps also videotape entire walkthrough.

Detailed Cognitive Walkthrough Example

Forwarding calls on a campus telephone system, from the perspective of a first time user; from [Wharton et al., 1994], pages 118–123.

- **Users:** New faculty, staff, guests, and visitors. For this evaluation assume that the user is a new university professor.
- **Task:** Cancel current forwarding and forward calls instead to a colleague with the extension 1234.
- **Interface:** Standard-size, touch-tone phone on desk. Overlay template includes the following information:

FWD *2
CNCL #2
SEND ALL *3

- **Correct Action Sequence:** The seven correct actions for accomplishing this task are:
 1. Pick up the receiver.
Phone: *dial tone*
 2. Press #2. {command to cancel forwarding}
Phone: *bip bip bip*
 3. Hang up the receiver.
 4. Pick up the receiver.
Phone: *dial tone*
 5. Press *2. {command to forward calls}
Phone: *dial tone*
 6. Press 1234.
Phone: *bip bip bip*
 7. Hang up the receiver.

Example Walkthrough Steps

1. Pick up the receiver.
Phone: *dial tone*
Success story:
Seems ok based on prior experience with phones.
2. Press #2.
Phone: *bip bip bip*
Failure story:
 - **Will the user be trying to achieve the right effect?**
How does the user even know that forwarding is in effect?
 - **Will the user know that the correct action is available?**
Probably yes, if forwarding is active, one must be able to cancel it. CNCL is visible on the template.

- **Will the user know that the correct action will achieve the desired effect?**

Might not recognise CNCL as the control to cancel forwarding. Might think that just pressing '2' is sufficient, instead of '#2'. Might try to press the buttons simultaneously, rather than sequentially.

- **If the correct action is taken, will the user see that things are going ok?**

How do first-time users know they have succeeded? After some experience, they will recognise the *bips* as confirmation, but will they at first?

3. Hang up the receiver.

Failure story:

- **Will the user be trying to achieve the right effect?**

Big trouble. How do you know you have to hang up before reestablishing forwarding?

4. etc.

Pros and Cons of Cognitive Walkthrough

- ++ finds task-oriented problems
- + helps define users' goals and assumptions
- + usable early in development process
- some training required
- needs task definition methodology
- applies only to ease of learning problems
- time-consuming

8.5 Guideline Scoring

- The interface is scored according to its conformance against a weighted list of specific guidelines.
- A total score is produced, representing the degree to which an interface follows the guidelines.

An example checklist for the evaluation of a web site is shown in Figure 8.4.

Pros and Cons of Guideline Scoring

- + cheap
- + intuitive
- must select and weight guidelines
- guidelines or weightings often domain-dependent

Web Technologies - Checklist Homepage Design / Usability

Homepage (URL)		Date		Nr.	
Tester		Size	kB	Score	%
Nr	Topic	Recommended Design	Strength	Points	You
1	Download time	50 kB (<10 sec for your customer)	***	3	
2	Window title	Start with Company Name	***	3	
3	Title tag line	What about	***	3	
4	Readable URL	Hackable URL	**	2	
5	Error page	Catch errors/dead links, to search	**	2	
6	Meta tags	For search engines	***	3	
7	Alt Information	Images, accessibility, Lynx	**	2	
8	Page width	770 pixel (620-1024)	**	2	
9	• Liquid vs. frozen layout	Liquid	**	2	
10	• Page length	<2 pages (1000-1600 px)	**	2	
11	Frames	No	***	3	
12	Logo placement	Upper left	***	3	
13	• Logo size	80x68 Pixel	**	2	
14	Search	Yes, in a box, always	***	3	
15	• Search placement	Upper part, right or left corner	***	3	
16	• Search box colour	White	***	3	
17	• Search button	Call it "Search" or "Go"	**	2	
18	• Width of search box	>=25 characters (30 best)	**	2	
19	• Type of search	Simple search (Link to advanced)	**	2	
20	Navigation	4 types: left, tabs, top, categories	**	2	
21	• Footer navigation links	Max. 7 links, single line	*	1	
22	• Sitemap link	Name "Site Map"	**	2	
23	Routing page	No	**	2	
24	Splash page	No	***	3	
25	Sign-In	"Account" or "Sign In"	*	1	
26	About the company	Always include it	***	3	
27	• About link	Call it "About <company>"	**	2	
28	• Contact information	Call it "Contact us"	**	2	
29	• Privacy policy	If you collect data	***	3	
30	• Name of privacy link	Call it "Privacy Policy"	**	2	
31	Job opening	Call it "Jobs" if you have it	**	2	
32	Help	If it is a complex site	*	1	
33	• Help placement	Upper right	**	2	
34	Auto-playing music	No	***	3	
35	Animation	No	**	2	
36	Graphics/illustration	5-15%	*	1	
37	Advertising	<= 3 ads	**	2	
38	Body text colour	Black	**	2	
39	• Body text size	12 points	*	1	
40	• Body text size frozen	No	***	3	
41	• Body text typeface	Sans-serif	*	1	
42	• Background colour	White	**	2	
43	• Link colour (unvisited)	Blue	**	2	
44	• Link colour (visited)	Purple	*	1	
45	• Link colour different	Yes (not light grey)	***	3	
46	• Link underlining	Yes (except in navigation bar)	**	2	
Score of URL:				100	

* Default Recommendation
 ** Strong Recommendation
 *** Essential Recommendation

from Nielsen, Homepage Usability
 adapted by alexander@nischelwitz.com V1.3

Figure 8.4: The Web Usability Checklist. The checklist covers 46 individual guidelines, each with a weighting of between one and three points, for 100 points total. [Adapted from [Nielsen and Tahir, 2001] by Alexander Nischelwitz, used with permission.]

8.6 Action Analysis

Quantitative analysis of actions to predict time skilled user requires to complete tasks, based on time estimates for typical interface actions. Focuses on performance of skilled user (efficiency).

Two flavours (levels of detail):

- a) Formal or “Keystroke-Level”
- b) Informal or “Back-of-the-Envelope”

There are good examples in [Lewis and Rieman, 1993] and [Raskin, 2000].

8.6.1 Keystroke-Level Analysis

Described by [Card, Moran and Newell, 1983].

- Developed from GOMS (Goals, Operators, Methods, Selection) modeling.
- Extremely detailed, may often predict task duration to within 20%, but very tedious to carry out.
- Used to estimate performance of *high-use* systems (e.g. telephone operator workstations).

Procedure for Keystroke-Level Analysis

- Break down tasks hierarchically into subtasks until reach fraction of second level of detail.
- Use average values for action times (determined through extensive published research) to predict expected performance for particular task. See Table 8.5.
- The analysts do not measure action times themselves, but refer to published tables.

	<i>Action</i>	<i>Time</i>
<i>Physical Movements</i>	One keystroke	0.28
	Point with mouse	1.50
	Move hand to mouse or function key	0.30
<i>Visual Perception</i>	Respond to brief light	0.10
	Recognise 6-letter word	0.34
	Move eyes to new location on screen	0.23
<i>Mental Actions</i>	Retrieve one item from long-term memory	1.20
	Learn one step of a procedure	25.00
	Execute a mental step	0.075
	Choose among methods	1.20

Table 8.5: Average Times for typical keystroke-level actions, in seconds. From [J. Olson and G. M. Olson, 1990], and cited by [Lewis and Rieman, 1993].

8.6.2 Back-of-the-Envelope Action Analysis

From [Lewis and Rieman, 1993]. “Back-of-the-Envelope” uses the analogy of sketching out a rough analysis on the back side of an envelope while somewhere away from your desk (“Milchmädchenrechnung” in German).

- List actions required to complete a task (as before), but in much less detail – at level of explaining to a user:

“Select Save from the File menu”

“Edit the file name”

“Confirm by pressing OK”

- At this level of analysis, every action takes at least 2 to 3 seconds (videotape a few users doing random tasks if you do not believe it takes this long!).
- Allows quick estimation of expected performance of interface for particular task.

Pros and Cons of Action Analysis

- + predicts efficiency of interface before building it
- some training required
- time-consuming

Chapter 9

Usability Testing Methods

“ Would you fly in an airplane that hasn’t been flight tested? Of course not. So you shouldn’t be using software that hasn’t been usability tested. ”

[Ben Shneiderman, The Front Desk, BBC Video, 1995.]

Empirical testing of interface design with representative users.

- *Thinking Aloud*: Test users verbalise thoughts while performing test tasks.
- *Co-Discovery*: Two test users explore an interface together. Insight is gained from their conversation while performing test tasks.
- *Formal Experiment*: Controlled experiment, face-to-face with test users, measurements and statistical analysis.
- *A/B Test*: Controlled experiment on (part of) actual user population, typically (remote) web site users, with measurements and statistical analysis.
- *Query Techniques*: Interviews and questionnaires.
- *Usage Studies*: Usage data is collected from a small number of users working on their own tasks in their natural environment over a longer period.

References

- + Jeffrey Rubin and Dana Chisnell; *Handbook of Usability Testing*; 2nd Edition, Wiley, 2008. ISBN 0470185481 (com, uk) [Rubin and Chisnell, 2008]
- + Joseph Dumas and Janice Redish; *A Practical Guide to Usability Testing*, Revised Edition; Intellect, 1999. ISBN 1841500208 (com, uk) [Dumas and J. Redish, 1999]
- o Thomas Landauer; *Research Methods in HCI*; In *Handbook of HCI*, Helander (Ed.), North-Holland, 1988. [T. Landauer, 1988]
- o Ericsson and Simon; *Protocol Analysis: Verbal Reports As Data*; MIT Press, May 1993. ISBN 0262550237 (com, uk) [Ericsson and Simon, 1993]
- o van Someren, Barnard, and Sandberg; *The Think Aloud Method*; Academic Press, 1994. ISBN 0127142703 (com, uk) [van Someren, Barnard and Sandberg, 1994]

- ++ Andy Field et al; *Discovering Statistics Using R*; Sage Publications, Mar 2012. ISBN 1446200469 (com, uk) [A. Field, Miles and Z. Field, 2012]
- ++ Andy Field; *Discovering Statistics Using SPSS*; Sage Publications, 3rd Edition, 2009. ISBN 1847879071 (com, uk) [A. Field, 2009]
- + Andy Field and Graham Hole; *How to Design and Report Experiments*; Sage Publications, 2002. ISBN 0761973834 (com, uk) [A. Field and Hole, 2002]
- Shaughnessy et al; *Research Methods In Psychology*; 6th Edition, McGraw Hill, April 2003. ISBN 0071198903 (com, uk) [Shaughnessy, E. B. Zechmeister and J. S. Zechmeister, 2003]

Online Resources

K. Anders Ericsson; *Protocol Analysis and Verbal Reports on Thinking* <http://www.psy.fsu.edu/faculty/ericsson/ericsson.proto.thnk.html>

Why do Usability Testing?

- More often than not, intuitions are wrong!
- People believe they understand behaviour of others based on their own experiences.
- This belief only lost through prediction then measurement (→ usability tests).
- Designers of system find it very easy to use.
- Experience changes one's perception of the world.
- It is almost impossible to “forget” an experience and put oneself in position of someone not having had the same experience.
- For example, Karl Dallenbach's famous photo, or hearing a clear version of a garbled voice recording.

Keyboard vs Mouse

From studies at Apple [Tognazzini, 1992]:

- Test users consistently report that keyboarding is faster than mousing.
- The stopwatch consistently proves that mousing is faster than keyboarding, an average of 50% faster.

“In one study of this phenomenon (Tognazzini, Tog on Interface, 1992.), users were asked to do the same task using the keyboard and the mouse. The keyboard was powerfully engaging, in the manner of many videogames, requiring the user to make many small decisions. The mouse version of the task was far less engaging, requiring no decisions and only low-level cognitive engagement.

Each and every user was able to perform the task using the mouse significantly faster, an average of 50% faster.

Interestingly, each and every user reported that they did the task much faster using the keyboard, exactly contrary to the objective evidence of the stopwatch.

The most obvious take-away message from this is that people's subjective beliefs as to what is or is not quick are highly-suspect. No matter how heart-felt the belief, until a stopwatch shows it is true, do not accept personal opinion about speed and efficiency as fact. Instead, user-test." [Tognazzini, 1999]

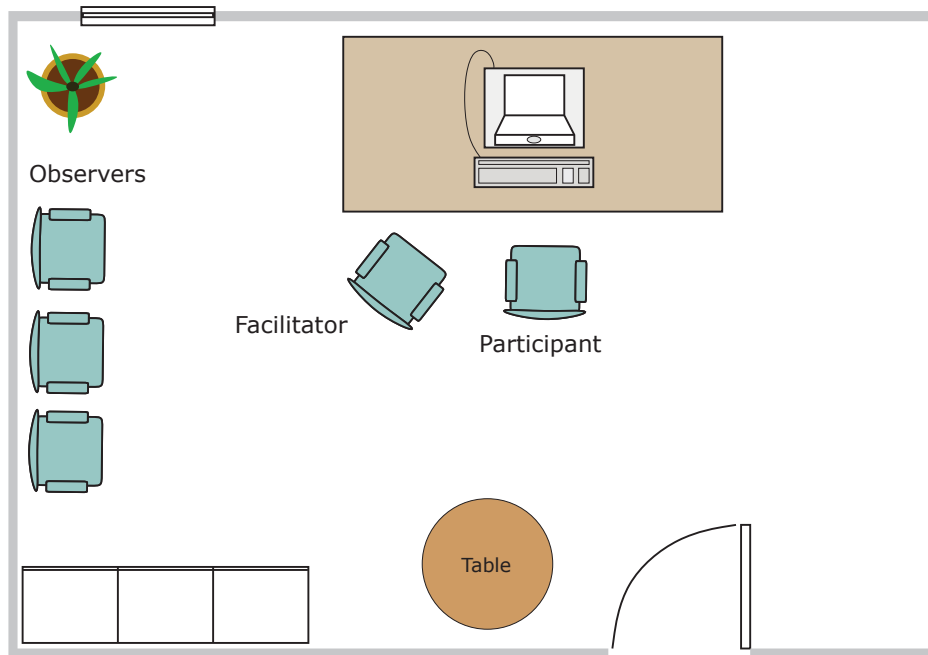


Figure 9.1: A simple usability test setup. This and the following test setup diagrams were inspired by those in [Rubin, 1994].

9.1 Preparing for Usability Testing

Test Environment

Ensure comfortable test environment:

- Organise *quiet* room.
- Put up sign “User test in progress – Do not disturb”.
- Disable telephones (fixed line and mobile).
- Ensure adequate lighting.
- Provide (non-alcoholic) refreshments.

Or use dedicated usability lab ...

Test Equipment

- Digital video camera (MiniDV, SD, DVD).
- Video tripod.
- Good microphone (table, lapel, or headset). A singer’s microphone is no good, because it must be held right next to the mouth.
- Headphones (to monitor sound).

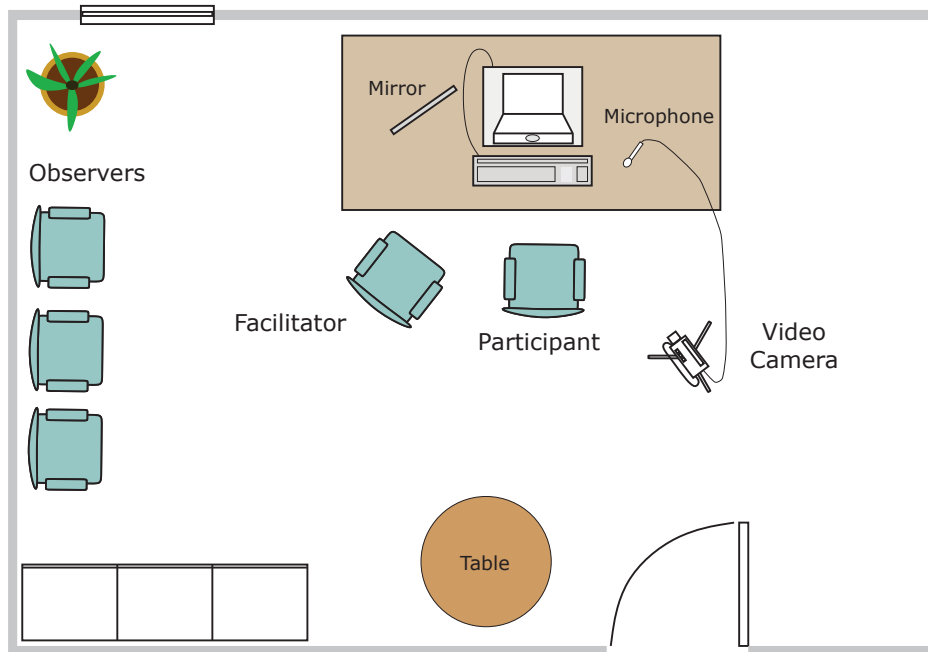


Figure 9.2: A typical single room, single camera usability test setup.



Figure 9.3: A simple usability test with a single video camera. A mirror is used to capture the user's facial expressions.

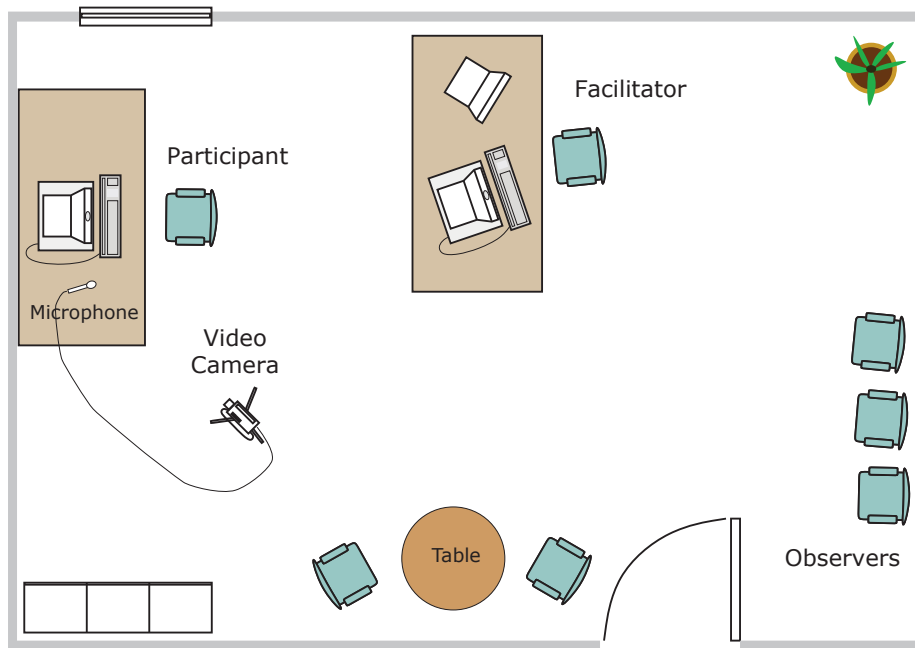


Figure 9.4: Single room test setup. Facilitator sits behind test participant monitoring video output and using logging software.

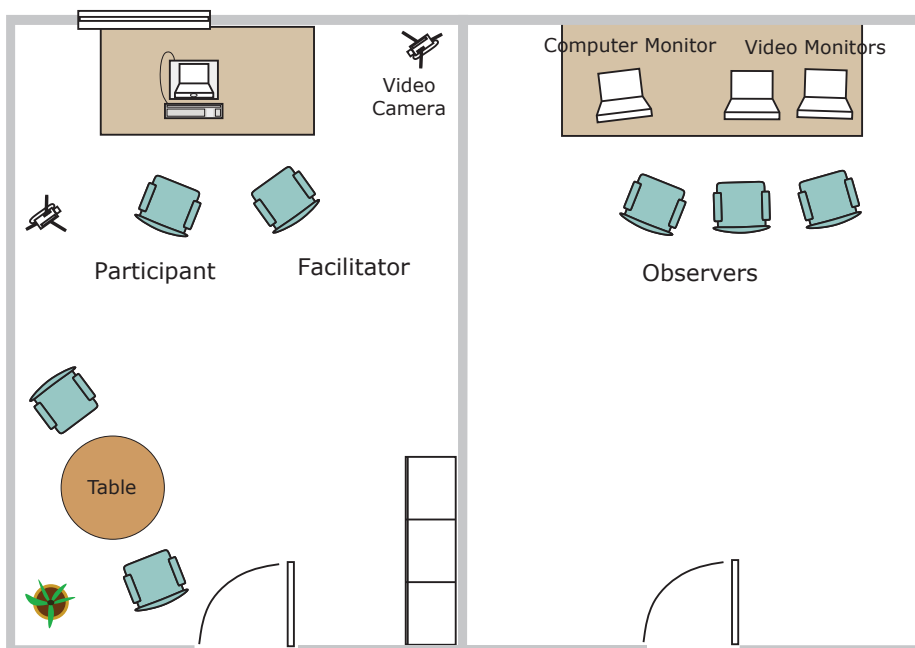


Figure 9.5: Observation room with electronic monitoring.

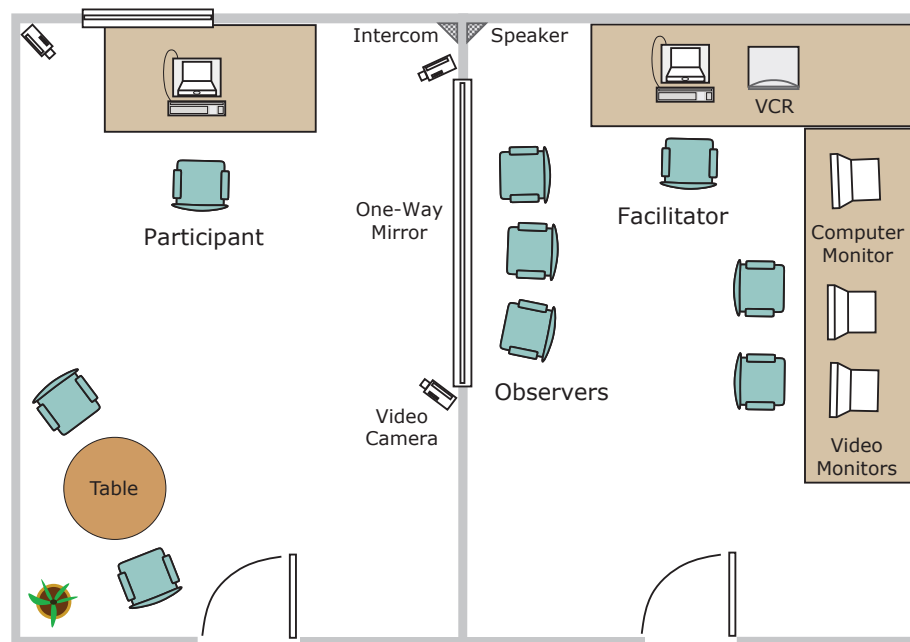


Figure 9.6: A classical usability lab, including an observation room with a one-way mirror.

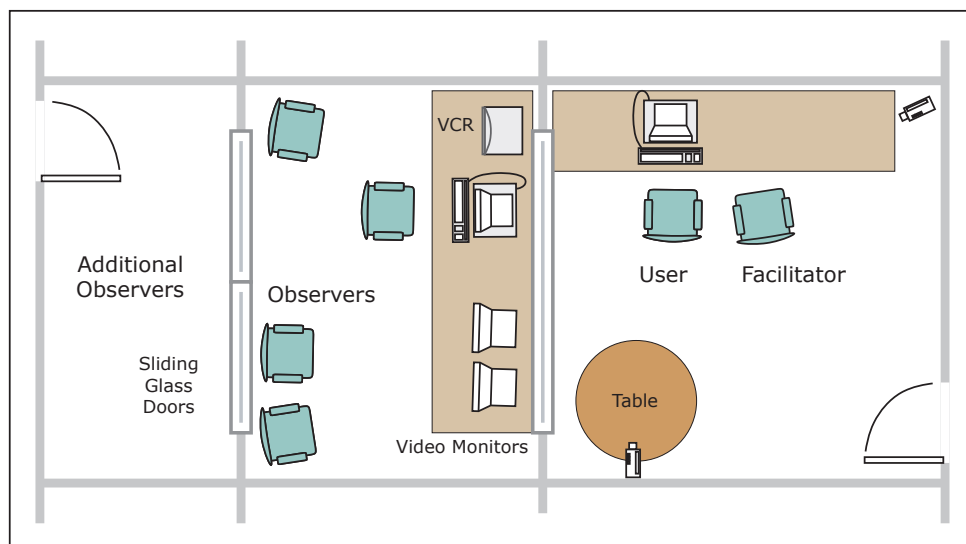


Figure 9.7: The standard usability lab at Microsoft headquarters in Seattle. There 25 such labs on two floors, side to side. Users enter from the right, developers enter from the left. [Adapted from diagram at [Microsoft, 2005].]

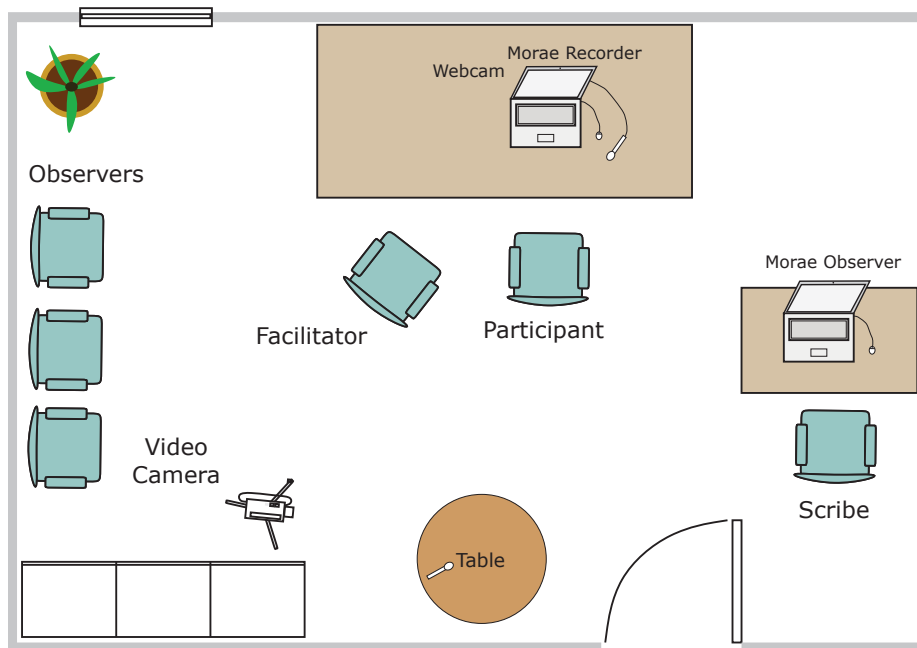


Figure 9.8: Morae software [TechSmith, 2008] installed on the test computer will capture the screen in one stream and the web cam of the user's face in a second, synchronised stream.

- Mirror (to capture user's facial expressions).
- Light (desk lamp or video lighting).
- Colour video monitor (to view camera image).
- DVD recorder.
- Video cart.
- Powerstrip, extension cables.
- "Do not disturb" sign.
- Refreshments.
- Logging software or forms.

Figures 9.9 and 9.10 show the portable usability kits used at Graz University of Technology.

Roles in the Test Team

- **Test Facilitator** (Administrator, Moderator, Manager, Monitor)

In overall charge of test, responsible for *all* interaction with test user (introduction, test, debriefing).

- **Data Logger** (Scribe)

Usability Kit Inventory List

- 1. Tripod Hama Profil 74
- 2. Rucksack LowePro
- 3. Headphones
- 4. Headphones extension cable
- 5. Microphone Phillips SBC ME570
- 6. Microphone extension cable
- 7. Headphone adapter
- 8. Usability kit inventory list
- 9. Video camera manual
- 10. Video camera power supply
- 11. Video camera mains cable
- 12. Microphone adapter cable
- 13. Video camera Sanyo Xacti HD1010
- 14. Video camera bag
- 15. Video camera remote control
- 16. Transcend SD HC Card 16gb



Figure 9.9: The portable usability kit used at Graz University of Technology. The inventory shows all of the components of the kit.



Setting up.



Packing.

Figure 9.10: The portable usability kit used at Graz University of Technology. On the left the kit has been set up ready for use. The righthand photos illustrate packing the kit.

Records activities and events of interest on paper, incl. time of occurrence.
[Assign shorthand codes to expected activities *before* test.]

- **Video Operator**

Responsible for recording *entire* test proceedings, incl. initial instructions and debriefing:

- Check camera angles so user and interface both clearly visible.
- Use *manual* focus (can't autofocus on computer screen).
- Turn off any data fields (time, date, etc.) in the camera.
- Ensure audio recording level is high enough.
- Label, copy, and edit tapes.

- **Computer Operator**

- Play the role of computer in a paper prototype.
- Reset the interface to a clean state for each new test user (clear the cache, history, cookies, etc.).
- Do *not* set the Home button to a web site under test, ask the user to type the URL.
- Restart after system crash, unexpected hang-ups, etc.

Test Users

- **Test Participants or Test Users**

- Users taking part in the test.
- Never ever call them *test subjects*!

9.2 Six Stages of Conducting a Test

1. Develop the Test Plan
2. Select and Acquire Participants
3. Prepare Test Materials
4. Run a Pilot Test
5. Conduct the Real Test
6. Analysis and Final Report

Note: *Always* do a pilot test!

9.2.1 The Test Plan

Main section headings:

- Purpose
- Problem Statement
- User Profile
- Method (Test Design)
- Task List
- Test Environment
- Data to be Collected
- Content of Report

Task List

- Prioritise tasks by frequency and criticality.
- Choose those most frequent and critical to test.
- Make a task list for test team internal use only.
- For each task:
 - Define any prerequisites.
 - Define successful completion criteria.
 - Specify maximum time to complete each task, after which help may be given.
 - Define what constitutes an error.
- Do *not* instruct the test user to return to the initial screen (home page) at the beginning of each task. If they do so of their own accord, that fine.

See Figure 9.11.

9.2.2 Selecting and Acquiring Participants

- Users are typically divided into different user groups, based on their characteristics and needs.
- Test each user group *separately*.
- Test at least 5 test users per user group.
- Choose *representative* test users who span the chosen user group.
- Acquire test users via employment agency, students, existing customers, internal personnel.
- Maintain a database of potential test users.
- Screening questionnaire (ensure users fit profile).

<i>Task</i>	<i>Description</i>	<i>Criteria</i>
1	Open the 3D landscape from with Harmony.	<i>PreReq:</i> Harmony Session Manager up and displayed. <i>Completed:</i> Landscape window open. <i>MaxTime:</i> 1 minute.
2	Find and open text document “Welcome to IICM”.	<i>PreReq:</i> Landscape window open at root collection. <i>Completed:</i> Text displayed in Text Viewer. <i>MaxTime:</i> 1 minute.
3	Navigate to City of Graz collection.	<i>PreReq:</i> Landscape window open. <i>Completed:</i> City of Graz collection in centre of view and opened. <i>MaxTime:</i> 2 minutes.
4	etc. ...	

Figure 9.11: An example internal task list for a usability test of Harmony.

9.2.3 Test Materials

- Orientation Script
- Background Questionnaire
- Nondisclosure and Consent Form
- Training Script (if any)
- Task Scenarios
- Data Collection Forms
- Debriefing Topic Guide
- Post Test Questionnaire
- Checklist

Orientation Script

- Introduce yourself and any observers by first name (no titles or job descriptions!).
- Explain that the purpose of the test is to collect input to help produce a better interface.
- Emphasise that system is being tested *not* user.
- Acknowledge software is new and may have problems.
- Do not mention any association you have with product (do mention if you are *not* associated with product).
- Explain any recording (reassure confidentiality).
- Say user may stop at any time.
- Say user may ask questions at any time, but they may not be answered until after the test is completed.

"Hi, my name is Keith. I'll be working with you in today's session. [Frank and Thomas here will be observing].

We're here to test a new product, the Harmony 3D Information Landscape, and we'd like your help.

I will ask you to perform some typical tasks with the system. Do your best, but don't be overly concerned with results – the system is being tested, and not your performance.

Since the system is a prototype, there are certainly numerous rough edges and bugs and things may not work exactly as you expect.

[I am an independent researcher hired to conduct this study, and have no affiliation with the system whatsoever]. My only role here today is to discover the flaws and advantages of this new system from your perspective. Don't act or say things based on what you think I might want to see or hear, I need to know what you really think.

Please do ask questions at any time, but I may only answer them at the end of the session.

While you are working, I will be taking some notes. We will also be videotaping the session for the benefit of people who couldn't be here today.

If you feel uncomfortable, you may stop the test at any time.

Do you have any questions?

If not, then let's begin by filling out a short background questionnaire and having you sign the nondisclosure agreement and consent to tape form."

Figure 9.12: An orientation script for testing the Harmony 3D Information Landscape.

- Invite questions.

See Figure 9.12.

Background Questionnaire

- Admin. data: date, test number, user number or id.
- General data: age (range), sex, educational level, ...
- Computer experience: total time, frequency of use, types of software, have used a GUI before, ...
- Application experience: total time, frequency of use, brand.

So that you can better understand the user's performance during the test. See Figure 9.13.

The facilitator should fill in the background questionnaire, asking the test user the questions.

If possible, copy and give to observers before the test proper starts.

Training Script

Exact written description of prior training:

- Demonstration of GUI.
- Demonstration of special interaction styles: mouse keys, drag-and-drop, etc.
- Walk-through of sample task.
- Demo of how to think aloud (for Thinking Aloud style tests).

Background Questionnaire

Date: _____ Test No.: _____ User No.: _____

General Information

Age: _____ Sex: _____

Education

- Please circle the highest educational grade you have achieved:
 Secondary School University Degree Doctorate
- If you are a student or graduate, please state your major area of study.

Computer Experience

- How long have you been using personal computers (years and months)? _____
- In a typical week, how many hours do you use a computer? _____
- Please circle the types of system you have used, followed by approximately how many months experience you have with them.

Months of Experience

Operating System

DOS	_____
OS/2	_____
Windows	_____
Unix	_____
Mac	_____

Application

Word Processor	_____
Database	_____
Spreadsheet	_____
CAD/CAM	_____

Experience of Related Systems

- How many hours per week do you spend accessing the Internet/World Wide Web? _____
- Do you have experience as a Web administrator or webmaster?
 If so, how many months? _____
- Have you used systems involving navigation through 3D worlds?
 If so, which systems? _____

Figure 9.13: A background questionnaire for the Harmony 3D Information Landscape usability test.

Thank you for participating in our product research. Please be aware that confidential information will be disclosed to you and that it is imperative that you do not reveal information that you may learn during the course of your participation. In addition, your session will be videotaped, to allow staff members who are not present to observe your session and benefit from your feedback.

Please read the statements below and sign where indicated. Thank you.

I agree that I will disclose no information about the product research conducted by ABC Company Inc. or about the specifications, drawings, models, or operations of any machine, devices, or systems encountered.

I understand that video and audio recordings will be made of my session. I grant ABC Company Inc. permission to use these recordings for the purposes mentioned above, and waive my right to review or inspect the recordings prior to their dissemination and distribution.

Please print name: _____

Signature: _____

Date: _____

Figure 9.14: A combined nondisclosure and consent form.

Task Scenarios

The task descriptions given to the test users.

- Simple introductory first task (early success).
- Realistic scenarios in typical order.
- If sequential ordering not crucial, randomise presentation order (→ counterbalances learning effect).
- Each task scenario on a separate sheet.
[do *not* hand the user all the tasks at once, but one at a time!]
- Do not guide participants through the task!
[Describe the goal rather than individual steps]

An Example Task Scenario

Task 3.	Using the 3D landscape, navigate to the collection about the city of Graz and open it.
---------	--

Data Collection Forms

- Define abbreviations for expected events → coding scheme. See Table 9.1.
- Use a probe mark like ⑦ to signal an event worth probing during debriefing.
- Paper or electronic data collection forms (or instrumented software). See Figure 9.15.

<i>Code</i>	<i>Event</i>
S	Start of task.
E	End of task.
O	Observation (problem).
G	Positive impression.
Q	Verbatim user comment.
X	Error or unexpected action.
H	User given help by facilitator.
P	Prompted by test facilitator.
T	Timeout, exceeded maximum time.
⊙	Probe this activity during debriefing.
C	Comment by facilitator.
*	Very important action.

Table 9.1: A simple coding scheme for logging events during a test.

Test:		User No.:
Date:	Time:	Page: of
<i>Task</i>	<i>Elapsed Time</i>	<i>Observations</i>

Figure 9.15: A generic data collection form.

Test: Edit HTML Document		User No.: 3
Date: 23 Apr 97	Time: 11:50	Page: 1 of 3
<i>Task</i>	<i>Elapsed Time</i>	<i>Observations</i>
1	04:25	X Opened wrong file. Found mistake. X Opened wrong file again. Self-corrected due to error message.
	06:00	P
	07:00	T
2	11:30	Q “I wish it were always that easy!”
	15:20	⊙ Very long hesitation, then correct action.
	16:15	E

Figure 9.16: A completed data collection form with a probe mark during task 2.

Debriefing Guide

- Things to discuss with user in any case.
- Structured interview.

Post Test Questionnaire

Collect feelings, opinions, suggestions (hard to observe in other ways), for example:

- Interface organisation matches real-world tasks?
- Too much or too little information on screens?
- Similar information consistently placed?
- Problems with navigation?
- Computer jargon?
- Appropriate use of colour?

Checklist

- Make chronological checklist of things to do, as shown in Figure 9.17.

9.2.4 Pilot Test

Always perform pilot tests of the *entire* test procedure.

You always find something you need to fix, such as:

- ambiguous instructions.
- unrealistic time estimates.
- ambiguous task completion criteria.
- misleading questionnaire questions.
- dead battery in microphone.
- blank DVDs incompatible with DVD recorder.

If you do not catch these things in a pilot test and one of these problems occurs with user number 1 of 10 scheduled at hourly intervals, it can ruin the whole test.

Maybe even run two pilot tests: once with colleagues, once with one or two real test users.

9.2.5 Conducting the Real Test

- Test facilitator handles *all* interaction with participant (other team members and observers remain completely quiet).

Test Checklist

1. Preparation:

- ☐ Reset interface for new user.
- ☐ Check that everything is ready in test room.
- ☐ Establish protocol for any observers.

2. Opening:

- ☐ Greet the participant.
- ☐ Go through orientation script and set the stage.
- ☐ Ask questions on background questionnaire: facilitator asks and fills out form.
- ☐ Ask user to read and sign consent and non-disclosure forms.

3. Test Session:

- ☐ Move over to testing area (computer).
- ☐ Start computerised session recording (Morae).
- ☐ Provide any prior training.
- ☐ Provide training of thinking aloud.
- ☐ User begins with tasks.
- ☐ User finishes last task.

4. Closing:

- ☐ Interview: how was it?
- ☐ Structured interview questions.
- ☐ Individual interview questions arising from test.
- ☐ Feedback questionnaire. User fills out form.
- ☐ Thank participant, provide any remuneration, show participant out.

5. Clean-Up:

- ☐ Summarise thoughts about this test.
- ☐ Organise data sheets and notes.
- ☐ Check and archive session recordings.

Figure 9.17: A test checklist.

- Do not prompt or bias user during test (beware of non-verbal signals).
- Only assist if user in severe difficulty (make note of when and what help given).
- Conduct debriefing interview or questionnaire.
- Save screen shots (in PNG) of any interesting problems and positive findings *right after the test*. They may not be reproducible again later.
- If you use screen capture software, you can extract stills from the screen video later on:
 - Configure the screen capture software to use a *lossless* codec.
 - Extract stills as PNG or another lossless format such as BMP (and then convert to PNG).
 - Do not extract a still image as JPEG, since that will introduce artefacts into the image.

Debriefing Interview

1. Let user speak thoughts first: “So, how was it?”.
2. Let them talk and talk, until they stop talking of their own accord.
3. Top-down: probe high-level issues from topic guide first, then more detailed questions about each task.
4. Probe specific issues arising from test notes ⑦ . See Figure 9.16.
5. Ask any questions passed to the facilitator from the observers (they should be written onto index cards by the observers).

See also Section 9.7.1.

9.2.6 Analysis and Final Report

- Compile and summarise data, for example:
 - Mean, median, range, and standard deviation of completion times.
 - Percentage of users performing successfully.
 - Bar chart of preference scores.
 - etc.
- Analyse data:
 - Identify errors and difficulties which arose.
 - Diagnose the source of each error.
 - Prioritise problems by their severity or criticality.

Final Report

- Title Page
- Description of Test Environment
 - Hardware, software version, test room, dates when tests were performed.
- Executive Summary
 - Concise summary of major findings, no more than a few pages.
- Description of Test
 - Updated test plan, method, training, and tasks.
- Test Person Data
 - Tabular summary of age, occupation, experience.
- Results
 - Tabular and graphical summaries of times taken, number of errors made, questionnaire responses, etc.
 - Discussion and analysis, amusing quotations.
- List of Positive Findings
- List of Recommendations List of problems discovered, in descending order of severity, and recommended improvements. For each recommendation:
 - diagnose why the problem occurred
 - illustrate it with a screen shot
 - rate its severity (0...4 scale)
 - indicate exactly how many test users experienced the problem
 - include a reference to timestamp(s) on the video tape
 - possibly include an appropriate user quotation
 - describe your suggested improvements
- Appendices (raw data and tables).
 - Background questionnaires, consent forms, orientation script, data collection forms, video and audio tapes, transcripts, etc.

Example Recommendation

R12. Sort Order Panel

(Severity 3.2)

- *Problem:* Users had problems understanding the sort order panel. In particular, the plus and minus icons used for increasing and decreasing order are non-intuitive.
- *Reference:* TP1, 00:08:15
“What does this plus mean?”
- *Recommendation:* redesign the icons, for example as sloping ramps.

9.3 Thinking Aloud

Test users are asked to verbalise their thoughts (“think aloud”) while performing tasks.

- provides wealth of process data
- relatively small number of test users (say 3 to 5)
- many vivid and colourful quotes.

Detecting Vocabulary Problems with Thinking Aloud

Example from Lewis and Rieman [1993, Section 5.5]:

- Menu-based administrative system for law offices.
- System messages extensively referred to “parameter”.
- Test users persistently misread “parameter” as “perimeter”.
- Hard to detect such problems just by watching people’s mistakes, much easier when they are thinking aloud.

The Thinking Aloud Method

Ask users to tell you:

- what they are trying to do
- things they read
- questions that arise in their mind
- things they find confusing
- decisions they make

Preparing the User

- Demonstrate thinking aloud for an unrelated task, e.g. looking up the films on tonight in the local cinemas (newspaper or online).
- Show user short video clip of a previous thinking aloud test.
- Have the user practice the technique using a different interface and unrelated task.
- Request questions be asked as they arise, but explain that you won’t answer them until after the test.

Test Facilitator’s Role

- Spontaneous comments from the user are best.

- If the user stops talking aloud, encourage them to keep up the flowing commentary with neutral, unbiased prompts:
 - non-committal “uh huh”
 - “Can you say more?”
 - “Please tell us what you are doing now?”
 - “I can’t hear what you are saying”
 - “What are you thinking right now?”
- Do *not* direct the user with specific questions like:
 - “Why did you do that?”
 - “Why didn’t you click here?”
 - “What are you trying to decide between?”

Do Not Ask Why Questions

- Maier [1931]. Problem: tie together two strings hanging from ceiling, too far apart to be grabbed. Solution: tie weight to one string, set it swinging, grab other string, wait for swinging string to come within reach. When Maier “accidentally” brushed against one string, people much more likely to find solution. When asked how arrived at solution, did *not* say Maier’s brushing against string had given them idea.
- Nisbett and Wilson [1977]. In market survey, most people preferred rightmost pair of three *identical* pairs of underwear. When asked why, people made up plausible (but wrong) reasons. Real reason is natural bias towards last of number of closely matched alternatives.

Specific “why” questions encourage plausible, but often unreliable, answers.

Listening Labs

Pre-defined tasks often neglect what individual users want to accomplish and sometimes miss larger strategic findings.

- Single user, thinking aloud.
- Environment simulates real-use setting.
- No preset tasks, but instead users set their own context and tasks.
- See [Hurst and Terry, 2000].

Pros and Cons of Thinking Aloud

- ++ finds many usability problems
- ++ finds *why* they occur (process data)
- + small number of test users (3 to 5)

- + usable early in development process
- + requires little facilitator expertise
- + generates colourful quotes
- - thinking aloud slows users down by about 17% [Ericsson and Simon, 1993, page 105]
- depending on the instructions given to the user, having to think aloud can change the user's problem-solving behaviour (they might think more before acting).
- cannot provide performance data (bottom-line data)

9.4 Co-Discovery

- Two test users explore an interface *together*.
- There is natural interaction and communication between the participants.

Pros and Cons of Co-Discovery

- ++ No unnatural thinking aloud.
- Need twice as many test users.
- - Validity issue: would the interface be used by two people working together in real life?

9.5 Formal Experiments

Controlled experiments with test users:

- Measurement and collection of quantitative data.
- Both objective measures (performance, success, ...) and subjective measures (ratings, overall preference).
- Summative evaluation: performed on (fully) implemented design(s).
- Followed by more or less rigorous statistical analysis.
- Two main uses:
 - assessing the absolute performance of an interface.
 - objectively comparing two (or more) alternative interface designs.

Formal experiments provide bottom-line data (measurements), but require larger numbers of test users for statistical accuracy (sometimes around 16 to 20, but often 50 or 100 or more).

Objective Measures

Collect objective, quantitative data by measuring or counting things such as:

- Time to complete specific task(s).
- Number of tasks completed within given time.
- Number of errors.
- Number of deviations (extra clicks) from optimal path.
- Accuracy (answer to question true or false).
- Ratio successful interactions : errors.
- Time spent recovering from errors.
- Number of commands/features used.
- Number of features user can remember after test.
- How often help system used.
- Time spent using help.
- Ratio positive : negative user comments.
- Number of times user sidetracked from real task.

Subjective Measures

Collect subjective, quantitative data, by asking users to give ratings and/or to express a preference (usually via a questionnaire), such as:

- The interface was easy to use (for example, on a scale of 0..6).
- The interface was cluttered (for example, on a scale of 0..6).
- Of the four interfaces, which was your favourite?

Validity

Validity: is measured data relevant to the usability of the real system in real world conditions?

Typical validity problems include:

- **Testing with the wrong kind of user**

For example, testing business students instead of managers for a management information system. However, testing business students will generally lead to better results than testing, say, mathematics students.

- **Testing the wrong tasks**

The results from testing a toy task in a prototype of a few dozen hypermedia documents may not be relevant to a planned system for managing tens of thousands of documents.

- **Not including time constraints and social influences**

Queues of people waiting in line, noise levels in the working environment, etc.

9.5.1 Testing Absolute Performance of One Interface

- One interface.
- Run an experiment to objectively determine whether the interface satisfies specific requirements.
- For example: measure how long it takes 20 expert users to perform task X.
- Result: an expert user can on average perform task X in 2 minutes 10 seconds \pm 6 seconds.

9.5.2 Comparing Two Alternative Interfaces

- Two interfaces, A and B.
- Run an experiment to objectively determine which interface is better, according to some criterion (efficiency, error rate, etc.).
- Two different ways of designing an experiment: *independent measures* (also called *between-groups* or *unrelated*) and *repeated measures* (also called *within-groups* or *related*).

Independent Measures (or Between-Groups) Experiment

- Two equally-sized groups of test users.
- *Randomly* assign users to two groups.
- Identical tasks for both groups.
- Group 1 uses only system A, group 2 only system B.

Pros and Cons of Between-Groups Experiment

- + no problems with learning effect.
- cannot ask users which they preferred.
- generally needs twice as many users.
- large individual variation in user skills (std. dev. $\approx 50\%$).

Repeated Measures (or Within-Groups) Experiment

- One group of test users.
- *Randomly* assign users to two equally-sized pools.
- Users perform *equivalent* tasks on *both* systems.
- Pool 1 uses system A first, pool 2 system B first.

Pros and Cons of Within-Groups Experiment

- + automatically controls for individual variability.
- + can ask users which they preferred.
- + generally needs fewer test users in total.
- transfer of skill between systems (learning effect).

Example Experimental Designs

Between-Groups		Within-Groups (Repeated Measures)	
<i>System A</i>	<i>System B</i>	<i>Participant</i>	<i>Sequence</i>
John	Dave	Elisabeth	A, B
James	Mariel	Sven	A, B
Mary	Ann	Amanda	A, B
Stuart	Phil	Claudia	A, B
Keith	Tony	Terry	A, B
Gary	Gordon	Nigel	A, B
Jeff	Ted	Barry	A, B
...
Bill	Edward	Ben	B, A
Charles	Thomas	Michael	B, A
Celine	Doug	Richard	B, A

Statistical Analysis

- Is there a statistically significant difference between system A and B? [hypothesis testing]
- How large is the difference? [point estimation, averages]
- How accurate are the results? [standard deviation, confidence intervals]

We are 95% certain that the time to perform task X is 4.5 ± 0.2 minutes.

System A is faster than system B at the level $p < 0.2$.

[20% chance that B is faster, but still choose A since odds are 4:1]

Sample Size (How Many Test Users?)

- Depends on desired confidence level and confidence interval.
- Confidence level of 95% often used for research, 80% ok for practical development.
- Nielsen: survey of 36 published usability studies [Nielsen, 1993b, pages 166–169]. Rule of thumb: 16-20 test users.
- If the differences are small, you might need 50 or 100 or more users to detect a statistically significant difference.

Case Study: Touchscreen Toggle Design

Catherine Plaisant, University of Maryland [Plaisant and Wallace, 1992].

- Home automation system with touchscreen display.
- Toggles (on/off switches) for lighting, climate control, security, etc.
- Six toggle designs: 1-Button, Rocker, 2-Button, Words, Slider, and Lever.
- Usability study with 15 novice users.

Pros and Cons of Formal Experiment

- ++ collects quantitative measurement data (bottom-line data)
- ++ collects both objective and subjective measures
- ++ allows comparison of alternative designs
- usable only later in development process
- requires facilitator expertise
- cannot provide why-information (process data)
- needs significant number of test users (20 or more)

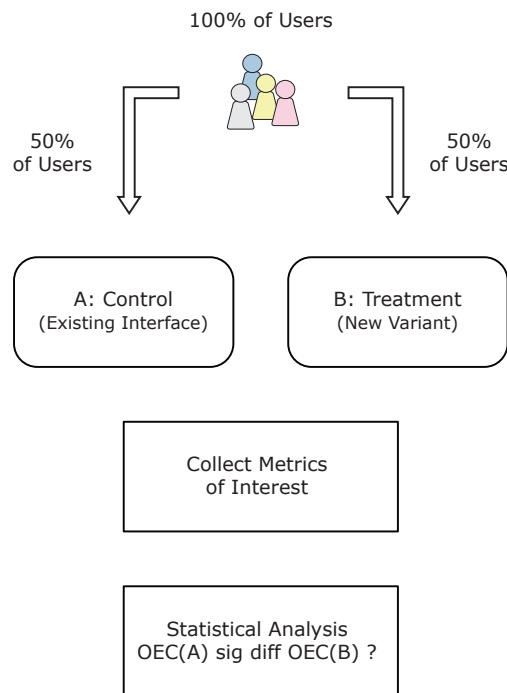


Figure 9.18: An A/B test is a controlled experiment on a web site involving its real live users. A proportion of users see a modified version and are then compared to a control group.

9.6 A/B Testing

Controlled experiments on a web site with its real live users.

- A proportion of visitors are randomly assigned to a variant (B) of the web site (they have a slightly different experience), the others see the standard web site (A, the control).
- A cookie is usually assigned, so that individual users always see the same variant.
- A metric (the *overall evaluation criteria*, or OEC) such as click-through rate is measured for each variant.
- The differences in OEC are examined for statistical significance.
- Originally used in marketing [Hopkins, 1923], where direct mail with variants of brochures elicited varying response rates (go with the best one).
- Online A/B testing was pioneered at Amazon.com.
- Also called *split testing*, *bucket testing*, and *multivariant testing*.

See Figure 9.18.

References

- ++ Ron Kohavi et al; *Controlled Experiments on the Web: Survey and Practical Guide*; Data Mining and Knowledge Discovery, Vol. 18, No. 1, Kluwer, Feb 2009, pages 140–181. doi:[10.1007/s10618-008-0114-1](https://doi.org/10.1007/s10618-008-0114-1) [Kohavi, Longbotham et al., 2009]

- ++ Bryan Eisenberg and John Quarto-vonTivadar; *Always Be Testing: The Complete Guide to Google Website Optimizer*; Sybex, Aug 2008. ISBN 0470290633 (com, uk) [Eisenberg and Quarto-vonTivadar, 2008]
- + Thomas Crook et al; *Seven Pitfalls to Avoid when Running Controlled Experiments on the Web*; Proc. KDD 2009, Paris, France, Jun 2009, pages 1105–1114. doi:10.1145/1557019.1557139 [Crook et al., 2009]
- ++ Ron Kohavi and Roger Longbotham; *Online Experiments: Lessons Learned*; IEEE Computer, Vol. 40, No. 9, Sept 2007, pages 103–105. doi:10.1109/MC.2007.328 [Kohavi and Longbotham, 2007]
- + Ron Kohavi et al; *Practical Guide to Controlled Experiments on the web: Listen to Your Customers not to the HiPPO*; Proc. KDD 2007, San Jose, California, Aug 2007, pages 959–967. doi:10.1145/1281192.1281295 [Kohavi, Henne and Sommerfield, 2007]
- o Claude Hopkins; *Scientific Advertising*; Lord & Thomas, 1923. Copyright expired. Freely available in PDF. [Hopkins, 1923]

Online Resources

- ++ Ron Kohavi; *Practical Guide to Controlled Experiments on the Web: Listen to Your Customers not to the HiPPO*; Industry Day talk at CIKM 2008 (Duration 23:55). Video. http://videlectures.net/cikm08_kohavi_pgtce/
- + Wikipedia; *A/B testing*; http://en.wikipedia.org/wiki/A/B_testing
- + *abtests.com*; <http://abtests.com/>
- + Anne Holland; *Which Test Won?*; <http://whichtestwon.com/>
- + Visual Website Optimizer; *IdeaFox*; <http://visualwebsiteoptimizer.com/ideafox.php>
- o Bryan Eisenberg; *A/B Testing for the Mathematically Disinclined*; ClickZ, 07 May 2004 <http://www.clickz.com/3349901>

Running an A/B Test

- Problems and biases in the design can be detected by running an A/A test.
- Multiple variants can be tested simultaneously (sometimes called A/B/N or A/B/Z tests).
- The user base must be large enough, so that there is a reasonable chance of reaching statistical significance.
- The test users are normal visitors to the site and usually unknowingly take part in an A/B test.

Examples of A/B Testing

- Google tested 41 shades of blue to find the best colour for links [Holson, 2009; Mayer, 2009].
- The BBC uses A/B testing for different design elements on its web sites [Hampson, 2010].

- Microsoft runs many A/B tests on its web sites [[Kohavi, Crook and Longbotham, 2009a](#); [Kohavi, Crook and Longbotham, 2009b](#)].
- Amazon has run many experiments to optimise its shopping cart [[Eisenberg, 2008](#)].
- Amazon even briefly experimented with different prices for the same product [[Martinez, 2000](#); [PBSJ, 2000](#)].

9.7 Query Techniques

Ask test users questions *after* having used system to perform representative tasks:

- Interview
- Questionnaire

Provide *subjective* data about users' view of system: preferences, impressions, attitudes.

Simple, cheap, and useful supplement to thinking aloud or formal experiment.

9.7.1 Interview

- Let user speak thoughts first: "So, how was it?".
- Top-down: probe high-level issues from topic guide first, then more detailed questions about each task.
- Probe specific issues arising from test notes.
- Review answers to post test questionnaire.
- Accept questions from any observers (should be written on a slip of paper for the facilitator to ask).

Pros and Cons of Interview

++ flexible – facilitator can probe interesting issues

- more time-consuming
- harder to analyse and compare

9.7.2 Questionnaire

- Written, structured form filled out by user.
- Quantitative and qualitative data.
- Electronic questionnaires such as WAMMI [WAMMI, 2000] or the GVU Web User Survey [GVU, 1998].
- Note, however, that designing truly unbiased questionnaires and surveys is a discipline of its own [Foddy, 1994; Dillman, 1999].

Pros and Cons of Questionnaire

++ can reach wide subject group (by post, email, Web forms)

++ easy to analyse and compare

+ less time-consuming than interview

- + easy to repeat (trends)
- less flexible than interview

Styles of Question

1. **General:** age (range), sex, occupation, etc.

2. **Open-Ended:** suggestions, comments.

I found the following aspects particularly easy to use (please list 0–3 aspects):

3. **Likert Scale:** judge agreement with specific statement (5, 6, or 7 point scale best).

Overall, I found the widget easy to use.

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

5 and 7 point scales offer the user a midpoint (a fence to sit on). Use a 6 point scale to force users to jump one way or the other. More than 7 point scales provide too little distinction between neighbouring points.

4. **Semantic Differentials:** sliding scale between opposing pairs of adjectives (5 or 7 point scale best).

Circle the number most closely matching your feelings about the interface.

Simple	3	2	1	0	1	2	3	Complex
Professional	3	2	1	0	1	2	3	Unprofessional
Reliable	3	2	1	0	1	2	3	Unreliable
Attractive	3	2	1	0	1	2	3	Unattractive

Externally, on the questionnaire, the user sees unbiased ratings from 3 down to 0 and back up to 3. Internally, for statistical analysis, you convert these ratings to scores between 0 points (the worst rating) and 6 points (the best rating).

5. **Overall Preference:** A vote for one item from a set of choices.

Overall, which hierarchy browser did you prefer?

Tree View (Windows explorer)	<input type="checkbox"/>
Hyperbolic browser	<input type="checkbox"/>
Treemap	<input type="checkbox"/>
Information pyramids	<input type="checkbox"/>

6. **Multi-Choice:** boxes to tick. [Tick just one box, tick multiple boxes, yes and no boxes]

Which methods do you use to get help (tick any that apply)?

Context-sensitive help	<input type="checkbox"/>
On-line manual	<input type="checkbox"/>
Printed manual	<input type="checkbox"/>
Google search	<input type="checkbox"/>
Ask a colleague	<input type="checkbox"/>

7. **Ranked:** place items in order.

Please rank the usefulness of these methods (1 most useful, 2 next, 0 if unused)?

Menu-selection	<input type="checkbox"/>
Button	<input type="checkbox"/>
Accelerator Key	<input type="checkbox"/>
Command line	<input type="checkbox"/>

Pros and Cons of Query Techniques

- + collect subjective user view of system
- + both quantitative and qualitative data
- + useful supplementary information
- + simple
- + cheap
- subjective data often unreliable

9.8 Usage Studies

Users actual activity is estimated or recorded and then analysed.

Purpose of a Usage Study

- Insight into how long users spend on each activity.
- Insight into which software is used for what purpose.
- Insight into which features of software packages are most used or unused.

9.8.1 Diary Studies

- Users are asked to keep a diary (or logbook) of their usage of a system over several days or weeks (self-reporting).
- Repeat for several users.
- The diary entries are converted into estimates of the amount of time spent on various activities.
- Statistically analyse the resulting data.

Pros and Cons of Diary Studies

- + anecdotal evidence (better than nothing?)
- subjective estimates made by users
- self-reporting is highly unreliable

9.8.2 Software Logging

- An instrumented version of the software logs all user interactions.
- Users must give their informed consent.
- Can recruit a larger sample of test users (20–50+).
- Gather and aggregate the various log files.
- Statistically analyse the resulting data.

Pros and Cons of Software Logging

- + objective log file data
- all the software of interest must be instrumented
- hard (impossible) to infer the user's intentions and motivations

9.8.3 Observational Studies

- Record one or more typical days of use of a system (screen capture and user video).
- Users must give their informed consent.
- Repeat for several users.
- Manually analyse the recordings and encode the activities (begin and end) in a timeline.
- Statistically analyse the resulting data.

References

- Byrne et al; *A Day in the Life of Ten WWW Users*; unpublished paper, 2000. [Byrne, John and Joyce, 2000]
- Byrne et al; *The Tangled Web We Wove: A Taskonomy of WWW Use*; [Byrne, John, Wehrle et al., 1999]

Finding Willing Users

Users are often reluctant to participate in a usage study, because they feel it is an invasion of their privacy.

Over several meetings and many hours, explain to potential test users:

- Exactly what will be recorded.
- That they can turn off the recording at any time.
- Exactly how the data will be analysed.
- That statistics will be aggregated on activities and software usage over several users.
- That statistics of individual user performance will not be aggregated.
- That individual users will not be identifiable in any reports or publications.
- That you will seek specific permission before showing or publishing and video, photographs, or screenshots.

Pros and Cons of Observational Studies

++ objective analysis of usage (not self-reporting)

- often difficult to find willing users

-- video analysis is extremely time-consuming

Chapter 10

Usability in Practice

References

- Michael Wiklund (Ed.); *Usability in Practice*; Academic Press, 1994. ISBN 0127512500 (com, uk) [Wiklund, 1994] [Out of print]
- ACM Interactions, Volume 9, Number 2, March 2002 <http://www.acm.org/dl>
- ACM Interactions, Volume 8, Number 2, March 2001.
- ACM Interactions, Volume 7, Number 2, March 2000.
- IEEE Software, 18(1), January/February 2001. Special issue on Usability Engineering. <http://www.computer.org/software/so2001/s1toc.htm>
- ISO 25062; *Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Common Industry Format (CIF) for usability test reports*.

Online Resources

- Jakob Nielsen; *Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier* http://www.useit.com/papers/guerrilla_hci.html
- Rolf Molich; *Comparative Usability Evaluation - CUE*; <http://www.dialogdesign.dk/cue.html>
- NIST; *Industry Usability Reporting*; <http://zing.ncsl.nist.gov/iusr/>
- Rolf Molich; *Usability Test Reports*; <http://www.dialogdesign.dk/utestreports.html>
- Jakob Nielsen; *Usability Laboratories: A 1994 Survey*; <http://www.useit.com/papers/uselabs.html>
- Microsoft Usability Group; <http://microsoft.com/usability/>

10.1 Comparison of Evaluation Techniques

10.2 Discount Usability Engineering

“Some usability evaluation is always better than none” [Jakob Nielsen]

	<i>Stage of Life-cycle</i>	<i>Obj./ Subj.</i>	<i>Qual./ Quant.</i>	<i>Data</i>	<i>Time</i>	<i>Equip-ment</i>	<i>Exper-tise</i>	<i>Insp./ Test</i>	<i>No. Usab. Special-ists</i>	<i>No. Users</i>
Exploratory Methods										
<i>Diary Studies</i>	compet. anal.	subj.	both	usage	medium	low	low	test	1–2	3–5
<i>Software Logging</i>	compet. anal.	obj.	quant.	usage	medium	medium	medium	test	1–2	20+
<i>Observ. Studies</i>	compet. anal.	observ.	both	usage	high	medium	medium	test	1–2 (+ 4–8 analysts)	3–5
Predictive Methods										
<i>Action Analysis</i>	design	subj.	quant.	bottom-line	med./high	low	med./high	insp	1–2	0
Formative Methods										
<i>Heur. Eval.</i>	all	subj.	qual.	process	low	low	medium	insp	3–5	0
<i>Guideline Checking</i>	all	subj.	qual.	process	medium	low	low	insp	1–2	0
<i>Cog. Walkthr.</i>	all	subj.	qual.	process	medium	low	high	insp	3–5	0
<i>Thinking Aloud</i>	design	observ.	qual.	process	medium	low/med.	low/med.	test	2–4	3–5
<i>Interviews</i>	design	observ.	qual.	process	medium	low	medium	test	1–2	>3
Summative Methods										
<i>Guideline Scoring</i>	finished system	subj.	quant.	bottom-line	medium	low	low	insp	1–2	0
<i>Formal Experiment</i>	compet. anal., finished system	obj.	quant.	bottom-line	high	medium	med./high	test	2–3	20–50+
<i>A/B Test</i>	finished system	obj.	quant.	bottom-line	medium	medium	med./high	test	2–3	1000s
<i>Questionnaires</i>	all	subj.	both	both	medium	low	medium	test	1–2	20+

Table 10.1: Comparison of evaluation techniques.

- *User and Task Analysis*: observe users on location, keep quiet, don't interfere.
- *Scenarios*: paper mock-ups, simple prototyping tools.
- *Heuristic Evaluation*: 3 evaluators.
- *Thinking Aloud*: pencil and paper notes, 2–3 users.

10.3 Differences in Evaluation Practices

“A recent survey shows that 80% of all Danish drivers think that their driving skills are above average.”

How about usability testers?

- Take a web site.
- Take N professional usability teams.
- Let each team evaluate the web site.
- Are the results similar?

Comparative Usability Evaluation (CUE)

- Rolf Molich, DialogDesign, Denmark.
- Tutorial and panel at CHI99.
- There have now been a total of 9 CUE studies since 1998.
- The original materials, team reports, papers, etc. are available at <http://dialogdesign.dk/cue.html>

Thanks to Rolf Molich for permission to use his materials in this section.

Four Comparative Studies

<i>Date</i>	<i>Name</i>	<i>Test Interface</i>	<i>Student Teams</i>	<i>Prof. Teams</i>
Oct 97	Student 1	9 Danish web sites	50	0
Dec 97	CUE 1	Windows calendar app	0	4
Oct 98	Student 2	9 Danish web sites	50	0
Dec 98	CUE 2	hotmail.com	2	7

All results point in the same direction: teams produce widely differing reports and results!

Student 1 and Student 2

- Introductory course in HCI at the Technical University of Copenhagen.
- 120 students per course.
- Fifty teams of one to three students.

- 9 Danish web-sites tested by four to nine teams with at least four test participants.
- Quality of evaluations and reports is acceptable considering that most teams used 20-50 hours.

Comparative Usability Evaluation 1 (CUE 1)

Four professional teams evaluated a Windows calendar program:

- Two US teams (Sun, Rockwell), one English (NPL) and one Irish (HFRG, Univ. Cork).
- Results published in a panel and a paper at UPA98 [Molich et al., 1998].

Comparative Usability Evaluation 2 (CUE 2)

Nine teams evaluated Hotmail:

- seven professional labs and two student teams,
- four from Europe, five from the USA.

Purpose of study was:

- a) to investigate the reproducibility of usability test results, and
- b) to survey the state-of-the art within professional usability testing.

not to pick a winner or make a profit.

CUE 2 Procedure

- Web-site address (hotmail.com) disclosed at start of three week evaluation period.
- Client scenario provided to teams.
- Email access to client through intermediary.
- Three weeks to evaluate using each team's standard approach.
- Deliver anonymised usability report.

Number of Problems Reported

<i>Number of Problems Found</i>	<i>CUE 1</i>	<i>CUE 2</i>
Total	141	300
by seven teams	-	1
by six teams	-	1
by five teams	-	4
by four teams	1	4
by three teams	1	15
by two teams	11	49
by only one team	(91%) 128	(75%) 226

Resources Used per Team

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>J</i>
<i>Person Hours</i>	136	123	84	(16)	130	50	107	45	218
<i># Usability Pros</i>	2	1	1	1	3	1	1	3	6
<i># Tests</i>	7	6	6	50	9	5	11	4	6

Usability Evaluation Reports

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>J</i>
<i># Pages</i>	16	36	10	5	36	19	18	11	22
<i>Exec. Summary</i>	y	y	n	n	n	y	n	y	y
<i># Screen Shots</i>	10	0	8	0	1	2	1	2	0
<i>Severity Scale</i>	2	3	2	1	2	1	1	3	4

Usability Results

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>J</i>
<i># Positive Findings</i>	0	8	4	7	24	25	14	4	6
<i># Problems Reported</i>	26	150	17	10	58	75	30	18	20
<i>% Exclusive</i>	42	71	24	10	57	51	33	56	60
<i>% Core</i> (100%=26)	38	73	35	8	58	54	50	27	31
<i>Person Hours</i>	136	123	84	(16)	130	50	107	45	218

Problem Reported by Seven Teams

- During registration process, users asked to enter a “password hint question” (see Figure 10.1).
- Most users did not understand the concept of a password hint.
- Some users entered their Hotmail password in the hint question box.
- Redesign as “secret question” with careful explanation (see Figure 10.2).

Language-Related Problem Reported by all European Teams

- To send a new message, users must select the “Compose” button in the left-hand panel (see Figure 10.3).
- European users did not understand the term “Compose”.
- “Create New Message” or “Write Mail” would be better.

Choose your password hint question and answer now. This will be used by us to help you in case you forget your password.

Please enter your password hint question:

Please enter your password hint answer:

Pressing the button below will confirm your entered information and activate your new Hotmail account!

OK

© 1996-1998 Hotmail. All rights reserved.

Figure 10.1: Part of the Hotmail registration process involved asking for a “password hint question”.

Account Information

Sign-In Name @hotmail.com Begin with a letter, and use only letters (a-z), numbers (0-9), the underscore (_), and **no spaces**.

Password Must be **at least eight (8) characters long**, may contain numbers (0-9) and upper and lowercase letters (A-Z, a-z), but **no spaces**. Make sure it is difficult for others to guess!

Re-enter Password

Secret Question Choose a question only you know the answer to and that has nothing to do with your password. If you forget your password, we'll verify your identity by asking you this question. [Writing an effective secret question](#)

Answer to Secret Question

Directories ☐ Hotmail Member Directory ☐ Internet White Pages Use the checkboxes to indicate whether you wish to be listed in these Internet directories. [More information about Directories.](#)

Figure 10.2: After testing, the dialogue was redesigned as a “secret question”.

msn Hotmail

HOTMAIL BECOME A WORDSMITH: USE SPELL TIP CHECK, DICTIONARY AND THESAURUS **msn Hotmail**

In-Box RELATED: [FindMessage](#) | [Reminders](#) | [Hotmail Member Directory](#)

Check for: [New Hotmail](#) | [POP Mail](#) - The Microsoft Network -

1 messages, 0 new

New	From	Date	Subject	Size
<input type="checkbox"/>	Hotmail Staff	Dec 03	Welcome New Hotmail User!	1k

☐ Select all displayed messages

Move To (Move Checked Messages to Selected Folder)

Delete Use checkboxes to select messages to move or delete

© 1996-1998 Hotmail. All Rights Reserved. [\[Contact Us\]](#) [\[Help\]](#)

Figure 10.3: The term “Compose” for writing a new message was not understood by European users.

Hotmail Summary of Findings

From the client's (Hotmail) point of view, of the 300 findings reported:

- New findings (4%)
- Validation of known issues (67%)
- Problems beyond Hotmail usability (29%)
 - business reasons for not changing.
 - beyond Hotmail control (partner sites).
 - problems generic to the web.

Current Boundaries to State-of-the-Art

- No specific expert user test, although four teams also recruited expert users.
- Few tests which required complicated setup, such as sending attachments.
- Little testing of boundary conditions, such as a large number of emails in the in-box.

Advice for a Usable Usability Report

- List problems with a severity rating
- State the number of users experiencing the problem.
- Include positive findings.
- Provide short executive summary.
- Keep it short.
- Distinguish clearly between:
 - personal opinions,
 - expert opinions,
 - user opinions, and
 - user findings.

Conclusions from CUE Studies

- There are overwhelmingly many usability problems.
- Many of them are serious.
- Limited overlap between team findings.
- Many of the teams found more problems than they actually reported. How do you select those to report?

- In most cases, no form of cost-effective testing will find all or most of the problems, or even most of the serious ones.
- Claims like “Method x finds at least 80% of all serious usability problems” are not in accordance with the results of the CUE studies.

10.4 Usability Reports

10.4.1 Common Industry Format (CIF)

Common Industry Format (CIF) for usability test reports.

CIF Summative (ISO 25062)

- Standard report format for formal experiments.
- Grew from the IUSR project at NIST [NIST, 2008] <http://zing.ncsl.nist.gov/iusr/>
- ISO 25062:2006; *Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Common Industry Format (CIF) for usability test reports*; ISO Standard, 31 Mar 2006. http://iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43046
CHF 146 ≈ € 90

CIF Formative

- Standard report format for thinking aloud tests.
- Project initiated by NIST: <http://zing.ncsl.nist.gov/iusr/formative/>
- Appears to have stalled.

10.4.2 UsabML

Usability Markup Language (UsabML).

- Standardised XML format for Heuristic Evaluation (HE) and Thinking Aloud test (TA) reports.
- Will potentially be extended to formal experiments and questionnaire surveys.
- Papers at HCSE 2010 and EICS 2010, web site usabml.fh-joaanneum.at.

10.5 Usability Consulting

- EU Usability Network [UsabilityNet, 2005]
- Austrian Directory of HCI&UE practitioners [Austrian Computer Society, 2008]

Pricing for External Heuristic Evaluation

- Nielsen Norman Group:
Around US\$ 38,000.
<http://www.nngroup.com/services/inspection.html>
- Interface Consult, Vienna (Martina Manhartsberger): from around € 900 (1 evaluator). http://usability.at/leistungen/experten_evaluation.html

Pricing for External Thinking Aloud Test

- Nielsen Norman Group:
Around US\$ 30,000.
<http://www.nngroup.com/services/testing.html>
- DialogDesign, Copenhagen (Rolf Molich):
TA test with 6 users, no video taping, on customer premises € 6,500 (48,000 DKK)
<http://www.dialogdesign.dk/Prices.htm>
- Catalysts, Linz:
TA test with 2 users € 1,008
<http://www.catalysts.cc/software/usabilitytest/>
- Interface Consult, Vienna (Martina Manhartsberger):
Usability Quick Test: TA test, 5 users, 20 mins. each, test on Tue, results on Thu, report as Power-point slides, € 1,074. <http://usability.at/leistungen/usability-schnell-test.html>

Chapter 11

Visual Design and Typography

General Visual Design References

- Kevin Mullet and Darrell Sano; *Designing Visual Interfaces*; Prentice Hall, 1994. ISBN 0133033899 (com, uk) [Mullet and Sano, 1994]
- Aaron Marcus; *Graphic Design for Electronic Documents and User Interfaces*; Addison-Wesley, 1992. ISBN 0201543648 (com, uk) [Marcus, 1992]
- Robin Williams; *The Non-Designer's Design Book*; 3rd Edition, Peachpit Press, 2008. [Williams, 2008]

11.1 Typography

Typography ... the style and appearance of text.

Font Sizes

Font sizes are traditionally expressed in printers' points (pt):

$$1 \text{ pt} = \frac{1}{72} \text{ inch} = \approx 0.35 \text{ mm}$$

$$1 \text{ pica} = \frac{1}{6} \text{ inch} = \approx 4.20 \text{ mm} = 12 \text{ pt}$$

Serif and Sans Serif Fonts

- A *serif* is a slight projection or embellishment at the end of a letter stroke, as shown in Figure 11.1.
- Examples of serif fonts include Times Roman and Georgia.
- A *sans serif* (French = without serif) font does not have such embellishments.
- Examples of sans serif fonts include Arial, Helvetica, and Verdana.
- When reading passages of text on paper (very high resolution), serif text may be easier to read than sans serif, but the evidence is not conclusive [Schriver, 1997, Chapter 5].

See Figure 11.1.

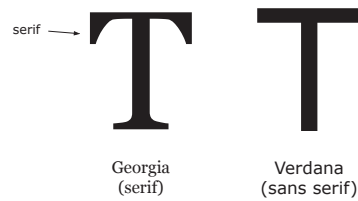


Figure 11.1: Georgia is a serif font, Verdana a sans serif font.

Typography References

- Paul Kahn and Krzysztof Lenk; *Principles of Typography for User Interface Design*; interactions, 5(6), 1998. [Kahn and Lenk, 1998]
- Robin Williams; *Non-Designer's Type Book*; 2nd Edition, Peachpit Press, 2005. ISBN 0321303369 (com, uk) [Williams, 2005]
- David Collier; *Collier's Rules for Desktop Design and Typography*; Addison-Wesley, 1990. ISBN 0201544164 (com, uk) [Collier, 1990]
- Spiekermann; *Stop Stealing Sheep*; 3rd Edition, Adobe Press, 2013. ISBN (com, uk) [Spiekermann, 2013]
- Jan Tschichold; *The Form of the Book*; Hartley and Marks, 1997. ISBN 0881791164 (com, uk) [Tschichold, 1997]
- Carter, Day, and Meggs; *Typographic Design: Form and Communication*; 5th Edition, Wiley, 2011. ISBN 047064821X (com, uk) [Carter, Day and Meggs, 2011-12-02]

11.2 Factors Influencing the Legibility of Text

11.2.1 Font Type

Proportional fonts consume less space and are more legible than fixed width fonts.

See Figure 11.2.

11.2.2 Font Size

- 10 pt is legible, 11 pt or 12 pt is better.
- The distinction between font sizes should be at least 2 pt.
[smaller changes cannot be discriminated by the eye]

For example, use:

- 12 pt for flowing text
- 10 pt for subscripts and footnotes
- 14 pt for section headings

Proportional fonts consume less space and are more legible than fixed width fonts. Proportional fonts consume less space and are more legible than fixed width fonts. Proportional fonts consume less space and are more legible than fixed width fonts.

Proportional fonts consume less space and are more legible than fixed width fonts. Proportional fonts consume less space and are more legible than fixed width fonts. Proportional fonts consume less space and are more legible than fixed width fonts.

Figure 11.2: Proportional versus fixed width fonts.

2 Sixteen Point Fonts Might Be Used for Titles

2.1 Fourteen Point Fonts for Section Headings

Twelve point is great for flowing text such as this^a. Remember that font size changes should be differentiated by at least two points.

^a10 pt might be used for subscripts and footnotes.

Figure 11.3: Font size changes should be differentiated by at least two points.

- 16 pt or larger for titles

as shown in Figure 11.3.

11.2.3 Case

Mixed case is more compact and legible than all upper case.

[Rehe: all upper case slows reading speed by about 12%] This is illustrated in Figure 11.4.

Word recognition is based partly on word shape. Lower and mixed case words have more irregular and thus more recognisable word shapes, as shown in Figure 11.5.

11.2.4 Character, Word, and Line Spacing

- Character spacing depends largely on font.
- Word spacing \approx width of an 'n'. See Figure 11.6.
- Line spacing \approx 2 pt. See Figure 11.7.

SETTING A LARGE AMOUNT OF TEXT IN CAPITAL LETTERS SERIOUSLY IMPAIRS LEGIBILITY. UPPER CASE SHOULD ONLY REALLY BE USED FOR EMPHASIS, BUT REMEMBER TO SET THE FONT SIZE A LITTLE SMALLER THAN THAT OF THE SURROUNDING TEXT.

Setting a large amount of text in capital letters seriously impairs legibility. Upper case should only really be used for EMPHASIS, but remember to set the font size a little SMALLER than that of the surrounding text.

Figure 11.4: Using all upper case slows reading speed.

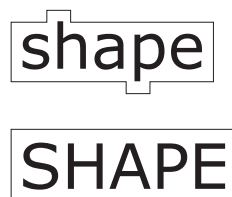


Figure 11.5: Lower and mixed case words have more recognisable word shapes.

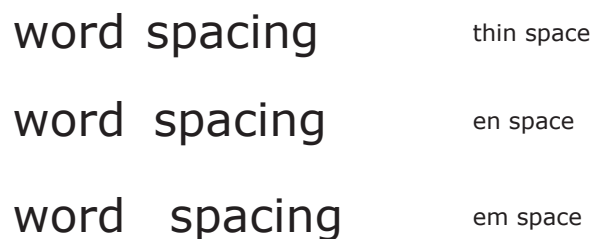


Figure 11.6: En and em word spacing.

The space between lines is important. Too much space and lines can float away. Too little space and lines become too densely packed to read comfortably. In the old days strips of lead “leading” were inserted between the lines of type. Nowadays, text processing systems make it easy to experiment with different values of leading.

The space between lines is important. Too much space and lines can float away. Too little space and lines become too densely packed to read comfortably. In the old days strips of lead “leading” were inserted between the lines of type. Nowadays, text processing systems make it easy to experiment with different values of leading.

Figure 11.7: Too much or too little space between lines makes reading difficult.

If a line is too long, readers have difficulty finding the beginning of the next line. The first example here has about 90 characters per line. For books, about 60 characters or ten words per line is about right. Newspaper columns typically have around 30 characters or five words per line.

If a line is too long, readers have difficulty finding the beginning of the next line. The first example here has about 90 characters per line. For books, about 60 characters or ten words per line is about right. Newspaper columns typically have around 30 characters or five words per line.

If a line is too long, readers have difficulty finding the beginning of the next line. The first example here has about 90 characters per line. For books, about 60 characters or ten words per line is about right. Newspaper columns typically have around 30 characters or five words per line.

Figure 11.8: For books, about 60 characters or ten words per line is about right. Newspaper columns typically have around 30 characters or five words per line.

11.2.5 Line Length

- \approx 10 words (60 chars.) per line for books.
- \approx 5 words (30 chars.) per line for newspapers.

See Figure 11.8.

11.2.6 Justification

Justification is the insertion of extra space within a line to create flush left *and* right margins.

- Justification without hyphenation slows reading due to gaps between words.
- Use either flush left, or justified with hyphenation.

See Figure 11.9.

11.2.7 Overall Text Environment

Maximum of:

- two typefaces,

Flush Left (Ragged Right)

Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.

Flush Right (Ragged Left)

Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.

Centered

Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.

Justified without Hyphenation

Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.

Justified with Hyphenation

Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.

Figure 11.9: Flush and justified text styles.

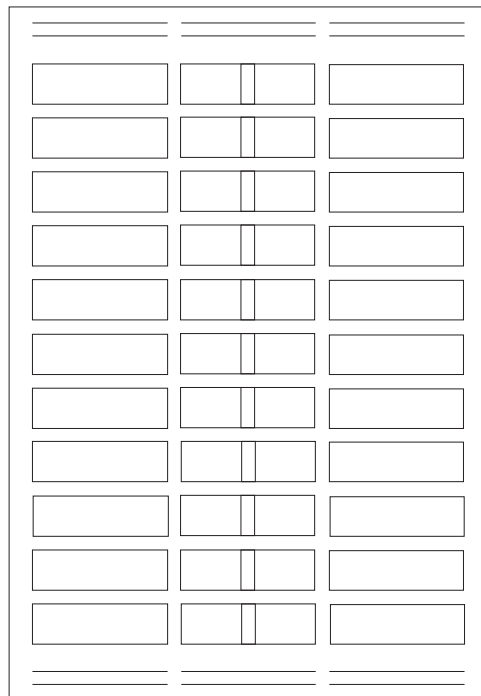


Figure 11.10: A layout grid ensures consistent location and extent of text, columns, titles, and illustrations.

- two slants (normal, italic),
- two weights (medium, bold),
- and four sizes (title, subtitle, text, footnote).

11.2.8 Layout

Use an underlying spatial grid to ensure consistent location and extent of text, columns, titles, illustrations, etc., as shown in Figure 11.10.

11.2.9 Margins

Avoid “word processor syndrome” (text right up to edge of the window) → leave ample margins.

See Figures 11.11 and 11.12.

Ample Margins

11.2.10 Typographic Distinction

Information is “any difference that makes a difference” [Gregory Bateson]

distinction \Leftrightarrow information

Use typographic distinction such as bold, italics, font change, etc. only if it conveys extra information.



Figure 11.11: Text right up to edge of the window is difficult to read “word processor syndrome”.



Figure 11.12: Ample margins all round make the text much easier to read.

Chapter 12

Icon Design

Icon . . . small visual symbol (broad definition)

“A picture is worth a thousand words.”

Well-designed icons:

- save screen space
- are recognised quickly in a busy visual environment
- are easily remembered
- help interfaces become international.

References

- + William Horton; *The Icon Book*; John Wiley, 1994. ISBN 0471599018 (com, uk) [Horton, 1994]
- + Jon Hicks; *The Icon Handbook*; Five Simple Steps, Dec 2011. ISBN 1907828044 (com, uk) [Hicks, 2011]
- Steve Caplin; *Icon Design*; Watson-Guptill Publications, April 2001. ISBN 0823025225 (com, uk) [Caplin, 2001]
- Gregory Thomas; *How to Design Logos, Symbols and Icons*; How Design Books, New Edition, 26 Apr 2003. ISBN 1581804563 (com, uk) [Thomas, 2003]
- Rudolf Modley; *Handbook of Pictorial Symbols*; Dover Publications, New York, 1976. ISBN 048623357X (com, uk) [Modley, 1976]
- Henry Dreyfuss; *Symbol Sourcebook*; Wiley, 1984. ISBN 0442218060 (com, uk) [Dreyfuss, 1984]

Online Resources

- Matt Queen; *Icon Analysis: Evaluating Low Spatial Frequency Compositions*; Boxes and Arrows, 25 Jul 2006. <http://boxesandarrows.com/icon-analysis>

- Uday Gajendar; *Learning to Love the Pixel: Exploring the Craft of Icon Design*; Boxes and Arrows, 13 Oct 2003. <http://boxesandarrows.com/learning-to-love-the-pixel-exploring-the-craft-of-icon-design/>
- symbols.com;
symbols.com
- The Noun Project;
thenounproject.com
- Open Iconic;
<https://useiconic.com/open/>
- Open Icon Library;
openiconlibrary.sourceforge.net
- The Icon Archive;
iconarchive.com
- Icon Finder;
iconfinder.com
- Mark James; *Silk Icons*;
<http://famfamfam.com/lab/icons/silk/>
- Nicolas Mollet; *Google Maps Icons*;
mapicons.nicolasmollet.com
- Realm Graphics Web Images
<http://www.ender-design.com/rg/>
- Anthony Thyssen; *Anthony's Icon Library*;
<http://www.ict.griffith.edu.au/anthony/icons/>

12.1 Visual Association

Which companies do these logos represent?



- Visual symbols are easy to remember.
- Powerful association, even though all colours were removed.

12.2 Standard Parts of an Icon

- Border

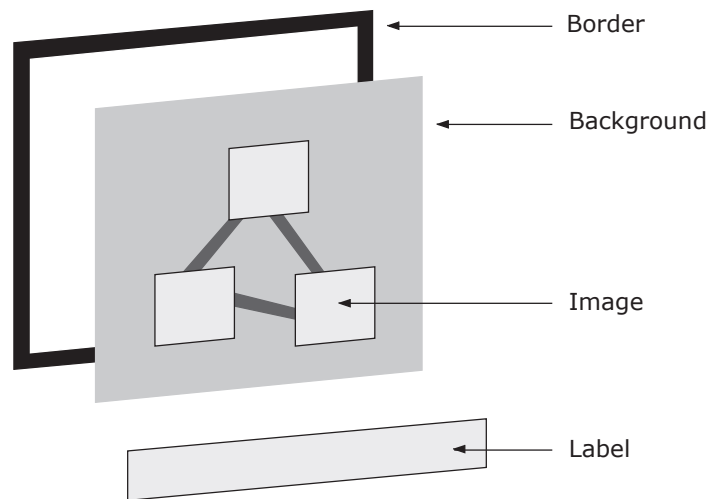


Figure 12.1: The standard parts of an icon.

- Background
- Image
- Label

See Figure 12.1.

12.3 Icon Design Principles

12.3.1 Coherency

- Design a set of icons *as a whole*.
- Icon set should be *consistent* in terms of size, colours, metaphor, level of realism (abstraction), etc.
- The icons in a set should be visually *balanced*.
- Visual distinctions should have significance – extraneous decorations distract.

Visually Imbalanced Icons

An early version of Paintbrush displayed visually incoherent icons, as can be seen on the left in Figure 12.2.

- The lower icons are simple and visually lightweight, the upper icons are very full (heavy).
- The redesigned icons in MS Paint (on the right) are more balanced and consistent.

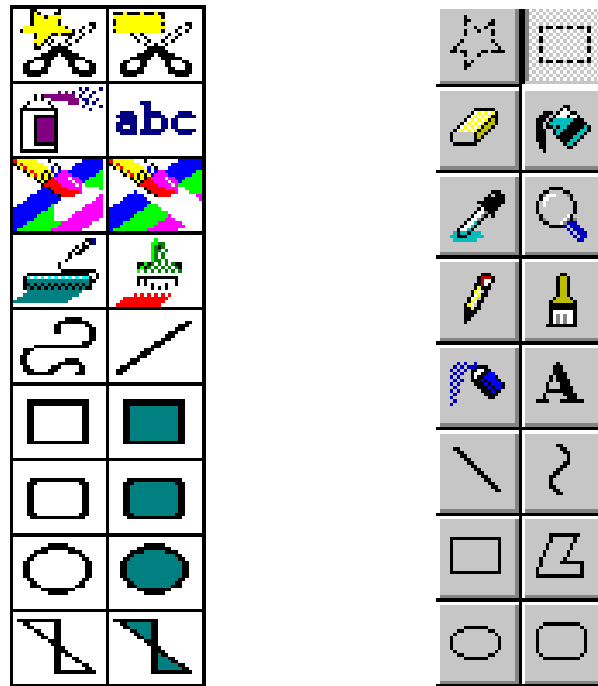


Figure 12.2: Visually imbalanced icons. On the left, the icon palette from an early version of Paintbrush for Windows. On the right, the redesigned icons from MS Paint.

Levels of Realism

From photorealistic to a simple silhouette:

- Photograph
- Drawing
- Caricature
- Outline
- Silhouette

However, stick to just *one* level of realism, unlike the mixture of levels of realism illustrated in Figure 12.3.

Symbols can also be drawn at different levels of abstraction, as shown in Figure 12.4.

12.3.2 Legibility

- As far as the pixel real estate allows, use large objects, bold lines, and simple areas.
- Take into account screen resolution and viewing distance.
- Good foreground/background contrast.
- Avoid arcs and oblique lines (“jaggies”).
- External shape (silhouette) conveys most info.

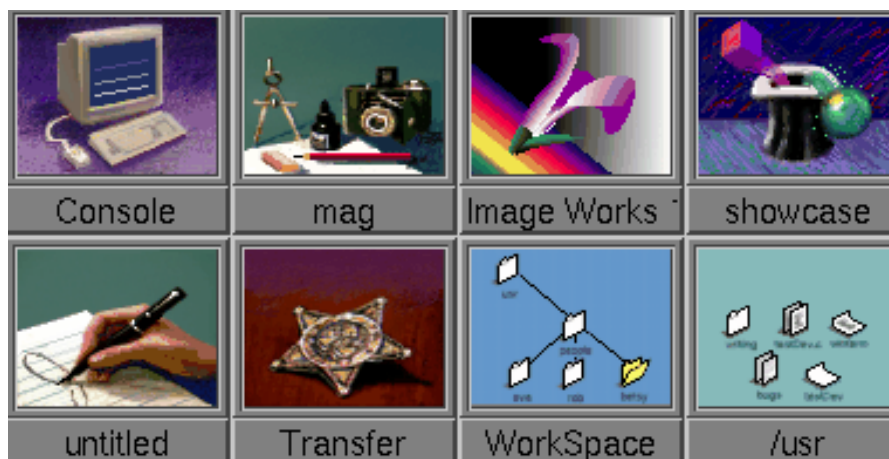


Figure 12.3: Mixed levels of realism in icons. Many of the icons from SGI's IRIX systems are photorealistic. However, some icons break the mould, like the abstract icons in the bottom right.

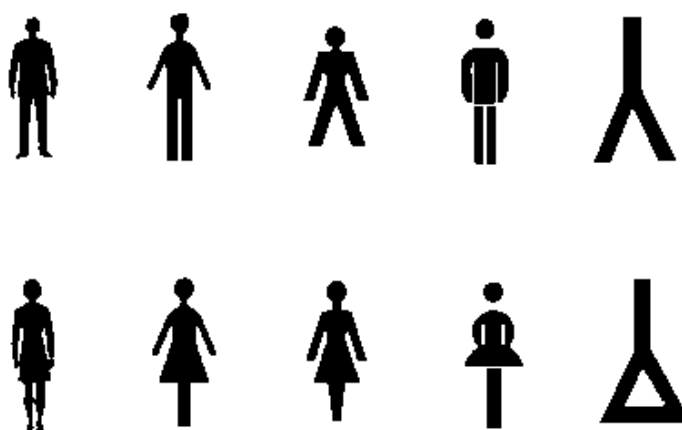


Figure 12.4: Symbols for men and women at different levels of abstraction.

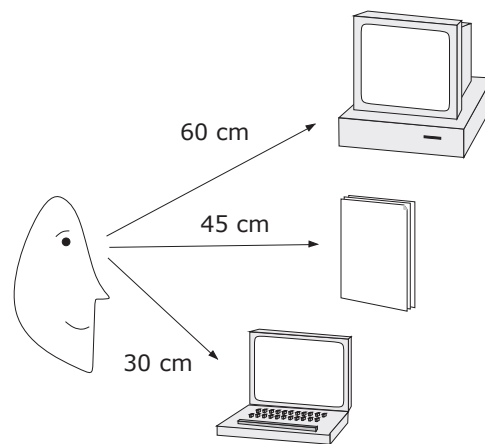


Figure 12.5: Typical viewing distances to icons.

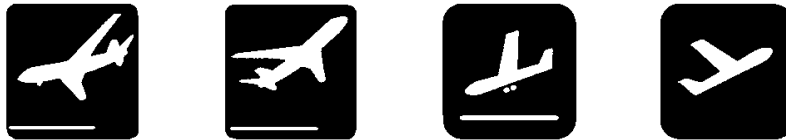


Figure 12.6: Simple aircraft outlines to denote arrivals and departures areas in an airport. Note how the silhouette conveys all of the information.

Typical Viewing Distances

- A desktop monitor: 60cm.
- A paper document: 45cm.
- A laptop screen: 30cm.

A 30×30 pixel icon will appear *much* larger at 30cm on an 11" 640×480 laptop screen, than at 60cm on a 17" 1280×1024 workstation monitor.

See Figure 12.5.

Silhouette Conveys Most Information

The external shape (silhouette) of a symbol conveys the most information.

See Figure 12.6.

12.3.3 Recognition and Recall

- Where possible, choose a metaphor familiar to the viewer.
- Use concrete objects wherever possible, abstract concepts and actions are difficult to visualise.
- Do you know of a good icon for “Shortcut”?



Figure 12.7: Garish multicolour icons competing for the user’s visual attention in a Windows 3.1 desktop.

- For years there was no good icon for the concept of “Undo”, but in the meantime there is a convention of a backwards curving arrow.
- Provide textual labels (at least by default).

12.3.4 Use Colour Sparingly

- Design first for black and white, add any colour later.
- Gratuitous use of colour overloads the viewer, use grey tones and one or two colours.

12.4 Cultural and International Issues

- Beware of using text or alphabetic characters inside an icon (as opposed to the label). Otherwise, different language versions of the icon will probably be needed. See Figure 12.10.
- Hand symbols, facial appearances, etc. vary immensely from culture to culture → don’t use them in icons.
- Beware also of metaphors dependent on a particular culture, e.g. the US mailbox for incoming mail. See Figure 12.11.

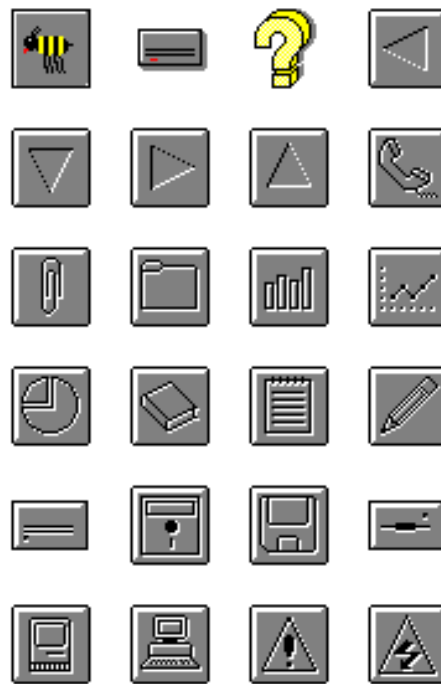


Figure 12.8: A well-balanced, consistent set of icons with restrained use of colour.

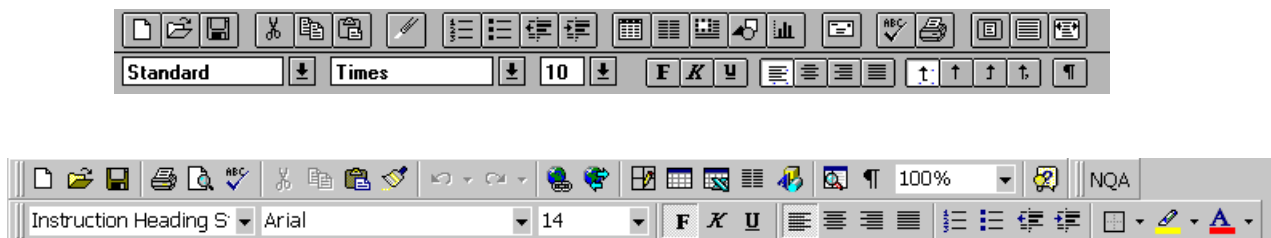


Figure 12.9: The evolution of the Microsoft Word icon bar from simple greyscale to limited use of colour.



Figure 12.10: Icons from the English and German versions of Word97. Text or alphabetic characters within an icon are language-dependent and will often have to be translated.

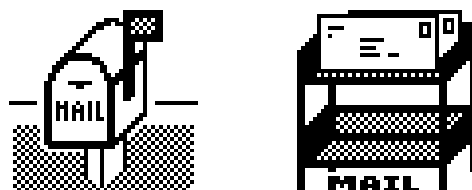


Figure 12.11: Two icons for incoming mail from different versions of Unix. The US mail box is unrecognisable for many Europeans; the letter tray is more universal. Note that these icons also use text within the icons proper which may need to be translated.

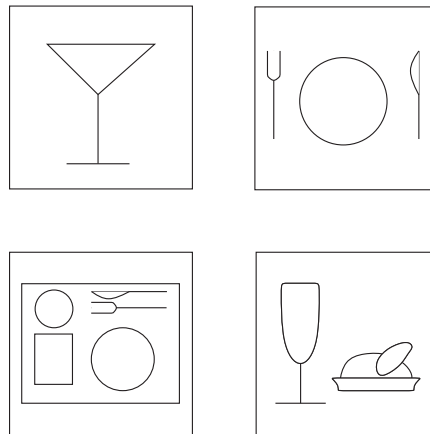


Figure 12.12: Icons for various types of food and drink area. What do the icons mean?

Bar

Snacks

Selfservice

Restaurant

Figure 12.13: Compared to the iconic representations in the previous figure, here words convey the subtle distinction better.

12.5 Do Not Always Use Icons

For more abstract concepts and subtle distinctions, verbal representations can sometimes work better than iconic representations.

See Figures 12.12 and 12.13.

12.6 Iconic Language

For larger sets of icons, it is often useful to develop an *iconic language*.

An iconic language is a systematic way of combining elementary symbols into more complex icons:

- *Vocabulary*: set of primitive symbols.
- *Grammar*: rules for combining them.

Example Iconic Language – Windows NT 4.0 (95)

Primitive symbols for application and document types are combined:

Document = Application + DocType [+ Template] [+ Assistant]

When a document is a template, or an assistant is provided, these symbols are added, as shown in Table 12.1.

Example Iconic Language – Harmony

Primitive symbols for document, link, and visited are combined for various types of document in the Harmony authoring tool for Hyperwave web servers [Andrews, 1996].

Document = [Link to] + DocType [+ Visited]

See Table 12.2.






















Elementary Symbols	
Document	
Assistant	
Template	
Document Types	
Text document	
Spreadsheet document	
Presentation document	
Database document	
Applications	
Word	
Excel	
Powerpoint	
Access	
Generated Icons	
Word text document	
Excel spreadsheet document	
Powerpoint presentation document	
Access database document	
Word template	
Powerpoint template	
Access template	
Word template assistant	
Powerpoint template assistant	
Access template assistant	

Table 12.1: Iconic language for Windows NT 4.0 documents.
















Elementary Symbols	
Document	
Link	
Visited	
Generated Icons	
Text document	
Visited text document	
Link to text document	
Visited link to text document	
Image document	
Visited image document	
Link to image document	
Visited link to image document	
3D model	
Visited 3D model	
Link to 3D model	
Visited link to 3D model	

Table 12.2: Iconic language for document and link icons in Harmony.

12.7 The Icon Design Lifecycle

Usability engineering lifecycle for icons:

Design, test, redesign.

See Figure 12.14.

Icon Design Iterations

- Start with simple black and white, hand-drawn sketches on paper (silhouette conveys the most information).
- Test and redesign until the basic symbols work.
- Add greys and perhaps colour. Design on computer. Print out colour versions of the designs, at the approximate real size.
- Test and redesign until the icons work.

Icon Intuitiveness Testing

Test the intuitiveness of (a set of) icons by running a simple thinking aloud usability test:

- Mount each icon design on a piece of card.
- Either tape up an area of the table with masking tape, into which the icons are placed, as shown in Figures 12.15 and 12.16. This is so the video camera remains focussed and test users are less tempted to pick up the icon cards.
Better still, use a stand to present icon designs at approximately the correct viewing angle and distance.
- The test facilitator places the icons one after the other in a random sequence into the taped up area or onto the stand.
- Use *thinking aloud* to capture:
 - the user's initial reaction
 - what they think the symbol *is*,
 - and what they think it might be intended to *represent*.

See Figure 12.16.

- At the end of the test, interview the test user in more detail. Provide an overview sheet/printout of all the icons to aid the discussion.

An Icon Test in Progress

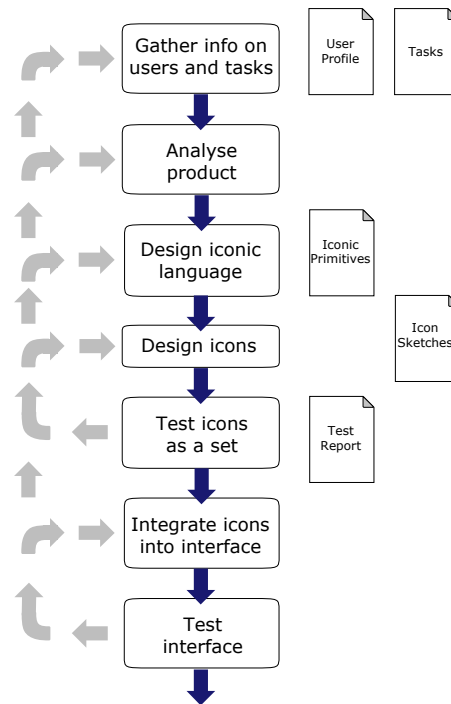


Figure 12.14: The icon design lifecycle. Based on the figure on page 270 of [Horton, 1994].

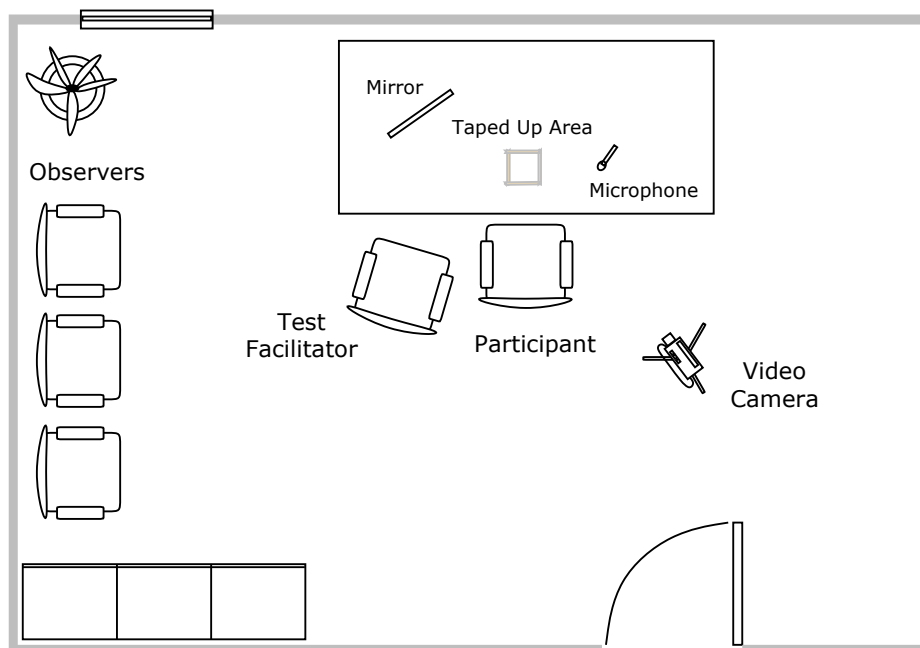


Figure 12.15: Test setup for an icon intuitiveness test.

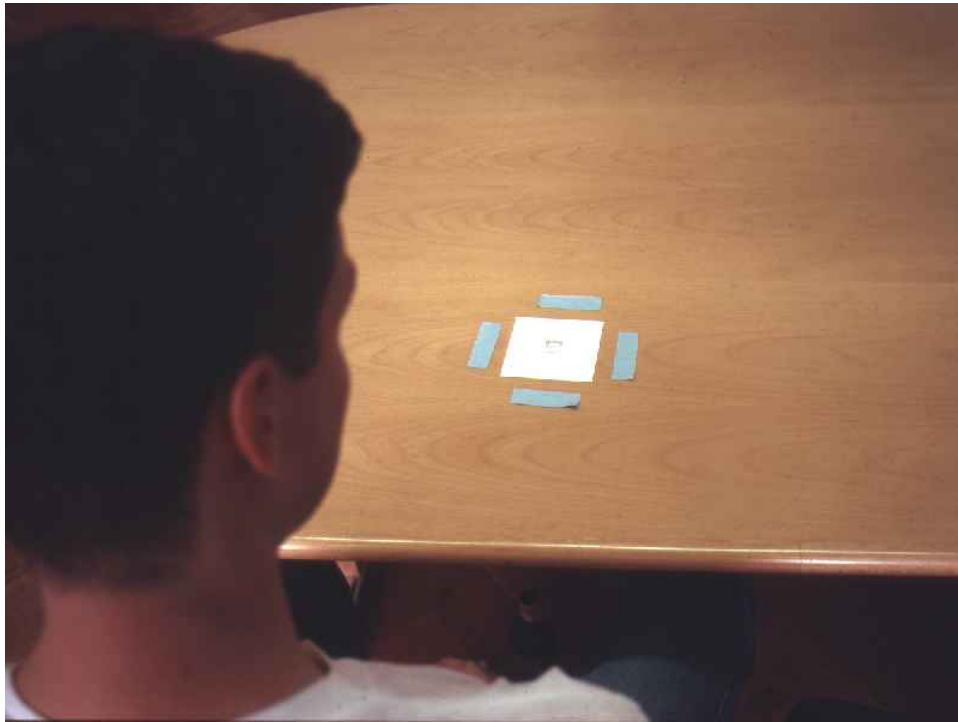


Figure 12.16: An icon intuitiveness test in progress. Each icon is printed out in colour and mounted on card. The taped up area of the desk keeps the icon in camera view.



Figure 12.17: Room setup for an icon test. Note the video camera and clip-on microphone.



Figure 12.18: Eight iterations of the icon design for “Products and Solutions”. Between each iteration, icon testing was carried out. [Thanks to Jakob Nielsen for permission to use these images.]

12.8 Designing Icons for Sun’s Public Web Site

- In the spring of 1995, Sun Microsystems’ public web site was redesigned.
- Icons were used to represent parts of the web site, such as “Technology and Developers”, “Products and Solutions”, and “Sun on the Net”.
- Thanks to Jakob Nielsen for permission to use the material in this section.

Icon Designs for Concept of “Technology and Developers”

Three rounds of icons were designed and tested to represent the concept of “Technology and Developers”, as can be seen in Tables 12.3, 12.4, and 12.5.

Icon Designs for Concept of “Products and Solutions”

Figure 12.18 shows the iterations of icon designs for the concept of “Products and Solutions”.

Icon Designs for Concept of “Sun on the Net”

Figure 12.19 shows the iterations of icon designs for the concept of “Sun on the Net”.




<i>Icon</i>	<i>Intended Meaning</i>	<i>User Interpretation</i>
	Chip, CD-ROM.	Too hard, finished product.
	Computer, chip.	Too hard, finished product.
	Construction worker.	Negative connotation of “under construction”.

Table 12.3: Initial set of black and white designs for an icon to represent the concept of “Technology and Developers” and their interpretation after user testing. [Thanks to Jakob Nielsen for permission to use these images.]




<i>Icon</i>	<i>Intended Meaning</i>	<i>User Interpretation</i>
	Developer, construction.	Liked most, but too much emphasis on hardware.
	Developer, power.	Liked idea of “harnessing the power”.
	Developer, computer.	

Table 12.4: Second round of icon sketches for “Technology and Developers”. [Thanks to Jakob Nielsen for permission to use these images.]




<i>Icon</i>	<i>Intended Meaning</i>	<i>User Interpretation</i>
	Developer, power, construction.	“Thunder and lightning”, “electric - looks painful”, “person being killed by technology”, “dance machine”, “why do Sun developers look bug-eyed?”.
	Power and cog wheels.	Lightning striking machinery.
	CD-ROM and cog-wheels.	Development and engineering.

Table 12.5: Third round of icons for “Technology and Developers”. The most promising of the black and white designs have been drawn on computer and colour has been added. [Thanks to Jakob Nielsen for permission to use these images.]

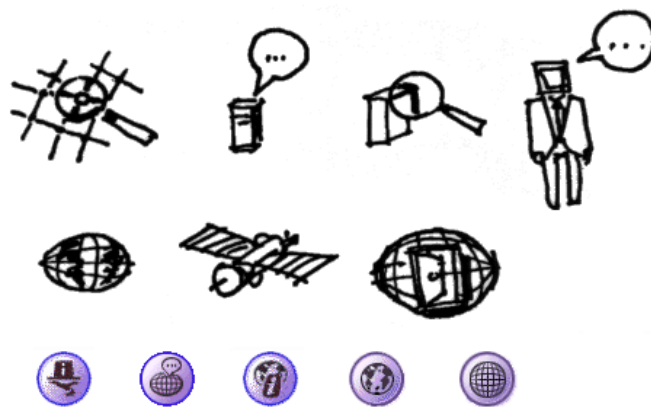


Figure 12.19: Twelve iterations of the icon design for “Sun on the Net”. Between each iteration, icon testing was carried out. [Thanks to Jakob Nielsen for permission to use these images.]

Chapter 13

A Brief History of HCI

References

- ++ Adele Goldberg; *A History of Personal Workstations*; Addison-Wesley, 1988. ISBN 0201112590 (com, uk) [Goldberg, 1988]
- + Brad Myers; *A Brief History of Human-Computer Interaction Technology*; ACM interactions, 1998. [Myers, 1998]

Online Resources

- Matthias Rauterberg ; *History of HCI*; <http://www.idemployee.id.tue.nl/g.w.m.rauterberg/presentations/HCI-history/index.htm>
- Wikipedia; *History of the Graphical User Interface*; http://en.wikipedia.org/wiki/History_of_the_graphical_user_interface
- Wikipedia; *WIMP (computing)*; [http://en.wikipedia.org/wiki/WIMP_\(computing\)](http://en.wikipedia.org/wiki/WIMP_(computing))
- Howard Rheingold; *Tools For Thought*; <http://www.rheingold.com/texts/tft/>
- DigiBarn Computer Museum; <http://www.digibarn.com/>
- Computer History Museum; <http://www.computerhistory.org/>
- IFIP WG 9.7; *History of Computing*; <http://www.comphist.org/>

Other Resources

- ++ Bob Cringely (real name Mark Stephens); *Triumph of the Nerds*;
Three-part TV series made for PBS in USA in 1996. Includes interviews with Steve Jobs, Bill Gates, and many others. The VHS version is 200 minutes, the DVD version is cut down to 150 minutes.

Available on youtube http://www.youtube.com/view_play_list?p=4D5CD637F73C24C7.

13.1 Early Interfaces

Memex (1945)

- Vannevar Bush, 1945 [Bush, 1945b].
- Memex = “memory extender”.
- Hypothetical design (never built) based on microfilm technology and mechanical levers integrated into a large desk.
- Vision of document repository with links between documents.
- “Trails” = chained sequence of links.
- Personal annotations.

Memex Chronology

- Bush started to formulate Memex ideas in 1932 [Nyce and Kahn, 1991, page 42].
- Draft sent to FORTUNE magazine on 07 Dec 1939 [Nyce and Kahn, 1991, page 52], but publication delayed by USA entering World War 2.
- Bush was presidential science advisor during World War 2.
- Memex article finally published in July 1945 in the Atlantic Monthly [Bush, 1945b].
- A condensed version, with new illustrations by Alfred D. Crimi, was printed on 10 Sep 1945 in LIFE Magazine [Bush, 1945a].
- Later also reprinted in Nyce and Kahn [1991] and Bush [1996].
- Second, revised article published in 1967 [Bush, 1967], also reprinted in Nyce and Kahn [1991].

SketchPad (1963)

- Ivan Sutherland, PhD Thesis, MIT, 1963.
- Drawing editor with built-in constraint solver.
- First graphical user interface: using a CRT and a lightpen (and various switches and knobs).
- First use of “windows” (virtual sheets of paper).
- First object-oriented program.
- Original PhD (scanned by MIT) [Sutherland, 1963a], reconstructed electronic edition produced by University of Cambridge [Sutherland, 1963b].
- Some video of SketchPad can be seen on Alan Kay’s video [Kay, 1987, at 00:04:06] and in the 1964 Science Reporter episode [Fitch, 1964].



Figure 13.1: The memex device. [Screen shot from the Memex animation [Adelman and Kahn, 1995], used with kind permission of Paul Kahn.]

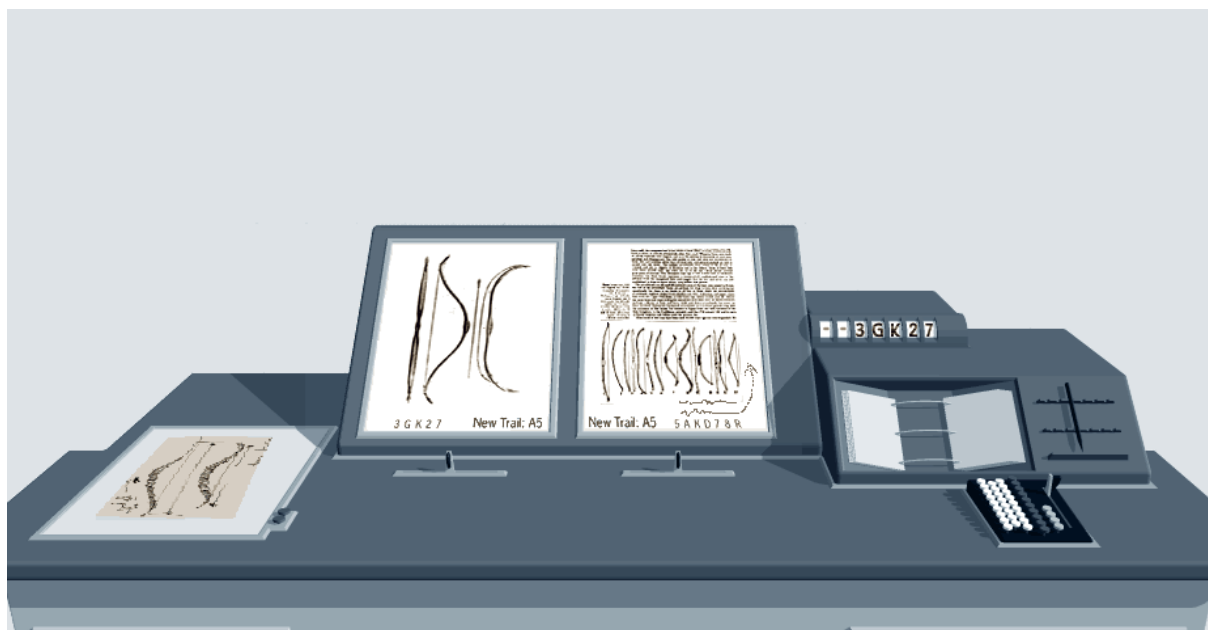


Figure 13.2: Part of a trail (A5) about the English long bow. Here a link between pages 3GK27 and 5AKD78R. [Screen shot from the Memex animation [Adelman and Kahn, 1995], used with kind permission of Paul Kahn.]



Figure 13.3: Ivan Sutherland's SketchPad system. [Image taken from the [sun.com](#) web site (it is no longer there) and used under the terms of the Sun Microsystems Copyright Notice (see page xiii).]

Augment/NLS (1968)

- Doug Engelbart, Augmentation Research Center (ARC) of Stanford Research Institute (SRI), 1968.
- First use of the mouse.
- First raster-scan (pixel-based) monitor.
- Mouse and five-chord keyboard for most interaction, keyboard only for entering longer pieces of text.
- First practical implementation of hypertext links.
- Drawing editor and image links.
- Interactive collaboration: multiple people could point and edit and see each other's pointers.
- Famous live demo "Mother of all Demos" [[Wikipedia, 2009](#)] at AFIPS Fall Joint Computer Conference (FJCC), San Francisco, 09 Dec 1968.

Engelbart and NLS Resources

- Augment concept paper [[Engelbart, 1962](#)].
- Comparative study of different input devices [[English, Engelbart and Berman, 1967](#)].
- The mouse patent [[Engelbart, 1970](#)], filed in 1967, expired in 1987.
- The Bootstrap Institute [[Engelbart, 2008](#)].

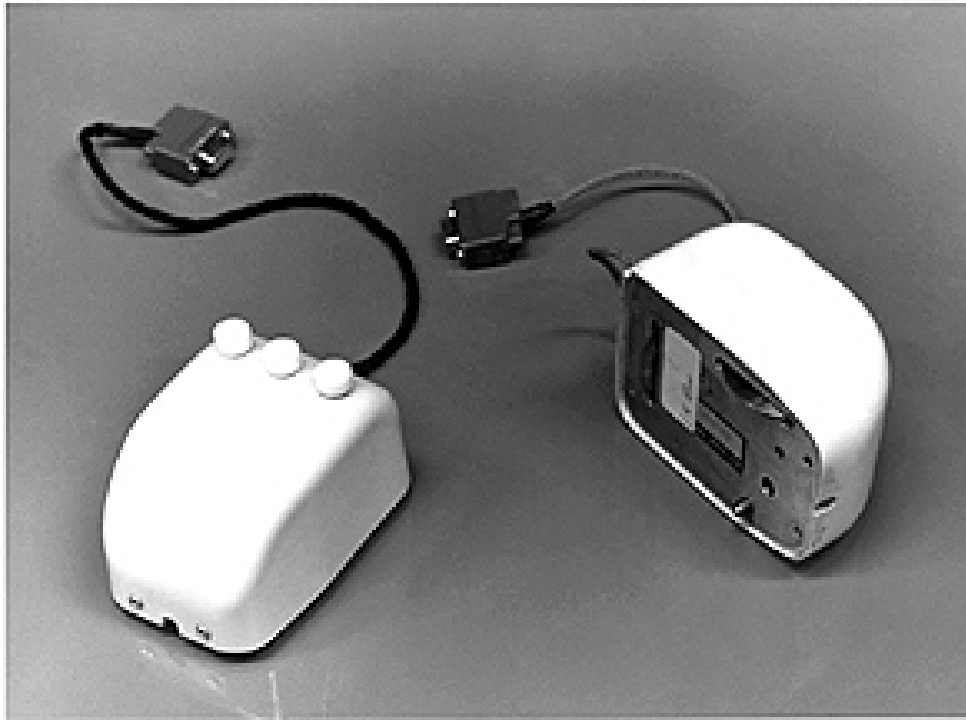


Figure 13.4: The Augment/NLS Production Mouse used by Doug Engelbart in the famous 1968 live demo. [Image used with kind permission of Stanford University, Special Collections.]

- Video interviews and background material at Invisible Revolution [[Hegland and Klijnsma, 2008](#)].
- Video of “The Mother of all Demos” can also be seen on Alan Kay’s video [[Kay, 1987](#), at 00:10:55].

13.2 WIMP Interfaces

WIMP stands for “window, icon, menu, pointing device”.

Xerox Alto (1973)

- Xerox Palo Alto Research Center (PARC) opened 01 Jul 1970.
- PARC’s Computer Science Lab had 50 or so of the best computer science researchers in the world (almost all with PhDs).
- “Bean-bag” culture.
- Xerox licensed the mouse from SRI in 1971.
- Xerox Alto was first built in 1973.
- The first “personal computer”, designed to be used by only one person (a very radical idea at the time).
- Several thousand Altos were built, but it was never released for sale commercially.
- The components alone cost over \$10,000 per machine.

Alto GUI

- Raster graphics display with pixels: black and white 808 × 606 pixels resolution.
- windows
- a mouse and a cursor
- pop-up menus
- Word processor (Bravo)

Bravo and BravoX

Bravo (and later BravoX) was a word processor developed for the Alto by Charles Simonyi and Butler Lampson:

- WYSIWYG (what you see is what you get)
- split screen
- bold and italic
- font families
- variable-width characters (proportional fonts)

Simonyi later joined Microsoft and led development of Microsoft Word.

Networked Desktop Environment

Also at PARC at the same time:

- laser printer
- ethernet

In other words, PARC had developed the first networked desktop office environment.

Xerox Star (1981)

- Designed as an office automation system.
- Computer as desktop of professionals in business organisations.
- Goal: computer invisible to the user.
- Easy-to-learn GUI.

Apple Lisa (1983)**Apple Macintosh (1984)****Microsoft Windows (1987)**

Bibliography

- Adelman, Ian and Paul Kahn [1995]. *Memex Animation*. 1995. <http://kahnplus.com/download/online/memex1a.zip> (cited on pages xiii, 185).
- Anderson, John [2004]. *Cognitive Psychology and its Implications*. 6th edition. Worth, 2004. ISBN 0716701103 (com, uk) (cited on page 3).
- Andrews, Keith [1996]. “Browsing, Building, and Beholding Cyberspace: New Approaches to the Navigation, Construction, and Visualisation of Hypermedia on the Internet”. PhD thesis. Graz University of Technology, Austria, 10th Sep 1996. <http://ftp.iicm.tugraz.at/pub/keith/phd/andrews-1996-phd.pdf> (cited on page 174).
- Andrews, Keith [2008]. *Evaluation Comes in Many Guises*. CHI 2008 Workshop on BEyond time and errors: novel evaluation methods for Information Visualization (BELIV’08). Florence, Italy, Apr 2008. <http://www.dis.uniroma1.it/beliv08/pospap/andrews.pdf> (cited on page 46).
- Apple Computer [1992]. *Macintosh Human Interface Guidelines*. Addison Wesley, 1992. ISBN 0201622165 (com, uk). <http://devworld.apple.com/techpubs/mac/pdf/HIGuidelines.pdf> (cited on page 78).
- Austrian Computer Society [2008]. *Austrian Directory to HCI&UE*. 2008. <http://www.ocg.at/ak/usability/directory.html> (cited on page 154).
- Baecker, Ronald M. and William A. S. Buxton, editors [1987]. *Readings in Human-Computer Interaction*. Morgan Kaufmann, 1987. ISBN 0934613249 (com, uk) (cited on page 3).
- Baecker, Ronald M., Jonathan Grudin et al., editors [1995]. *Human-Computer Interaction: Toward the Year 2000*. 2nd edition. Morgan Kaufmann, Apr 1995. ISBN 1558602461 (com, uk) (cited on page 3).
- Bailey, Robert W. [1996]. *Human Performance Engineering*. 3rd edition. Prentice Hall, 1996. ISBN 0131496344 (com, uk) (cited on page 3).
- Barnum, Carol M. [2001]. *Usability Testing and Research*. Longman, 2001. ISBN 0205315194 (com, uk) (cited on page 44).
- BBC [1996]. “The Front Desk”. In: *30-minute video*. Open University Learning Resources M206/03V. 1996. <http://worldcat.org/title/front-desk/oclc/222590055> (cited on pages 5, 53, 61).
- Beyer, Hugh and Karen Holtzblatt [1997]. *Contextual Design : A Customer-Centered Approach to Systems Designs*. Morgan Kaufmann, Aug 1997. ISBN 1558604111 (com, uk) (cited on page 53).
- Bias, Randolph G. and Deborah J. Mayhew [2005]. *Cost-Justifying Usability: An Update for the Internet Age*. 2nd edition. Morgan Kaufmann, Apr 2005. ISBN 0120958112 (com, uk) (cited on page 59).
- Bush, Vannevar [1945a]. “As We May Think”. *LIFE Magazine* 19.11 (10th Sep 1945), pages 112–124 (cited on page 184).

- Bush, Vannevar [1945b]. “As We May Think”. *Atlantic Monthly* 176.1 (Jul 1945), pages 101–108. <http://www.theatlantic.com/doc/194507/bush> (cited on page 184).
- Bush, Vannevar [1967]. “Memex Revisited”. In: *Science is Not Enough*. Edited by Vannevar Bush. New York: William Morrow, 1967, pages 75–101 (cited on page 184).
- Bush, Vannevar [1996]. “As We May Think”. *interactions* 3.2 (Mar 1996), pages 35–46. ISSN 1072-5520. doi:10.1145/227181.227186 (cited on page 184).
- Buxton, William A. S. [2007]. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann, Mar 2007. ISBN 0123740371 (com, uk) (cited on page 81).
- Byrne, Michael D., Bonnie E. John and Elizabeth Joyce [2000]. “A Day in the Life of Ten WWW Users”. Unpublished paper. 2000. <http://citeseer.ist.psu.edu/400156.html> (cited on page 145).
- Byrne, Michael D., Bonnie E. John, Neil S. Wehrle et al. [1999]. “The Tangled Web We Wove: A Taskonomy of WWW Use”. In: *Proc. Conference on Human Factors in Computing Systems (CHI’99)*. ACM. Pittsburgh, Pennsylvania, USA, May 1999, pages 544–551. ISBN 0201485591 (com, uk). doi:10.1145/302979.303154 (cited on page 145).
- Calvin, William H. [1997]. *How Brains Think*. Basic Books, 1997. ISBN 046507278X (com, uk) (cited on page 3).
- Caplin, Steve [2001]. *Icon Design: Graphic Icons in Computer Interface Design*. Watson-Guptill, 2001. ISBN 0823025225 (com, uk) (cited on page 165).
- Card, Stuart K., Thomas P. Moran and Allen Newell [1983]. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum, 1983. ISBN 0898598591 (com, uk) (cited on page 107).
- Carroll, John M. and Mary Beth Rosson [1987]. “The Paradox of the Active User”. In: *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. Edited by John M. Carroll. MIT Press, 1987. Chapter 5. ISBN 0262031256 (com, uk). <http://www.cs.vt.edu/~rosson/papers/paradox.pdf> (cited on page 101).
- Carter, Rob, Ben Day and Philip B. Meggs [2011-12-02]. *Typographic Design: Form and Communication*. 5th edition. Wiley, 2011-12-02. ISBN 047064821X (com, uk) (cited on page 158).
- CFR [2008]. *Code of Federal Regulations, Title 14, Part 23, Section 781*. US Federal Government. 1st Jan 2008. <http://frwebgate.access.gpo.gov/cgi-bin/get-cfr.cgi?TITLE=14&PART=23&SECTION=781&TYPE=PDF> (cited on pages 27, 29).
- Chapanis, Alphonse [1999]. *The Chapanis Chronicles: 50 Years of Human Factors Research, Education, and Design*. Aegean, Jun 1999. ISBN 0963617893 (com, uk) (cited on page 27).
- Collier, David [1990]. *Collier’s Rules for Desktop Design and Typography*. Addison-Wesley, 1990. ISBN 0201544164 (com, uk) (cited on page 158).
- Constantine, Larry L. and Lucy A.D. Lockwood [1999]. *Software for Use*. Addison-Wesley, 1999. ISBN 0201924781 (com, uk) (cited on page 44).
- Cooper, Alan [2004]. *The Inmates are Running the Asylum*. Sams, 2004. ISBN 0672326140 (com, uk) (cited on pages 1, 61, 67).
- Cooper, Alan, Robert M. Reimann and David Cronin [2007]. *About Face 3: The Essentials of Interaction Design*. Wiley, 2007. ISBN 0470084111 (com, uk) (cited on pages 1, 43, 53, 61).
- Courage, Catherine and Kathy Baxter [2004]. *Understanding Your Users*. Morgan Kaufmann, 2004. ISBN 1558609350 (com, uk) (cited on page 53).

- Crook, Thomas et al. [2009]. “Seven Pitfalls to Avoid when Running Controlled Experiments on the Web”. In: *Proc. 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2009)*. ACM, Paris, France, Jun 2009, pages 1105–1114. doi:10.1145/1557019.1557139. <http://exp-platform.com/Documents/2009-ExpPitfalls.pdf> (cited on page 138).
- Darnell, Michael J. [2010]. *Bad Human Factors Designs*. 2nd May 2010. <http://baddesigns.com/> (cited on pages xiii, 7, 12, 14–16).
- Dillman, Don A. [1999]. *Mail and Internet Surveys: The Tailored Design Method*. 2nd edition. Wiley, 1999. ISBN 0471323543 (com, uk) (cited on page 140).
- Dreyfuss, Henry [1984]. *Symbol Sourcebook*. Wiley, 1984. ISBN 0471288721 (com, uk) (cited on page 165).
- Dumas, Joseph and Janice Redish [1999]. *A Practical Guide to Usability Testing*. Revised Edition. Intellect, 1999. ISBN 1841500208 (com, uk) (cited on page 109).
- Edwards, Chris [1999]. *Sketchy Thing*. 1999. <http://advance.aiga.org/timeline/artifacts/artifact3.html> (cited on pages 85, 86).
- Eisenberg, Bryan [2008]. *Hidden Secrets of the Amazon Shopping Cart*. GrokDotCom Marketing Optimization Blog. 26th Feb 2008. <http://www.grokdotcom.com/2008/02/26/amazon-shopping-cart/> (cited on page 139).
- Eisenberg, Bryan and John Quarto-vonTivadar [2008]. *Always Be Testing: The Complete Guide to Google Website Optimizer*. Sybex, Aug 2008. ISBN 0470290633 (com, uk) (cited on page 138).
- Ellis, Geoffrey and Alan Dix [2006]. “An Explorative Analysis of User Evaluation Studies in Information Visualisation”. In: *Proc. AVI 2006 Workshop on BEyond time and errors: novel evaluation methods for Information Visualization (BELIV '06)*. Venice, Italy: ACM Press, May 2006, pages 1–7. ISBN 1595935622 (com, uk). doi:10.1145/1168149.1168152 (cited on page 46).
- Emgonzalez [2005]. *Boeing B-17 Flying Fortress*. Lone Star Flight Museum Air Show, Galveston, Texas. 13th Nov 2005. http://upload.wikimedia.org/wikipedia/commons/f/ff/B-17_Lone_Star_Flight_Museum_Airshow.jpg (cited on page 27).
- Engelbart, Douglas C. [1962]. *Augmenting Human Intellect: A Conceptual Framework*. AFOSR-3233 Summary Report. Stanford Research Institute, Oct 1962. http://www.liquidinformation.org/ohs/62_paper_full.pdf (cited on page 186).
- Engelbart, Douglas C. [1970]. *X-Y Position Indicator for a Display System*. US Patent 3541541. Filed 21 Jun 1967, issued 17 Nov 1970. Stanford Research Institute, Nov 1970 (cited on page 186).
- Engelbart, Douglas C. [2008]. *Bootstrap Institute*. 2008. <http://bootstrap.org> (cited on page 186).
- English, William K., Douglas C. Engelbart and Melvyn L. Berman [1967]. “Display-Selection Techniques for Text Manipulation”. *Transactions on Human Factors in Electronics* 8.1 (Mar 1967), pages 5–15. ISSN 0096-249X (cited on page 186).
- Ericsson, K. Anders and Herbert Alexander Simon [1993]. *Protocol Analysis: Verbal Reports As Data*. Revised Edition. MIT Press, May 1993. ISBN 0262550237 (com, uk) (cited on pages 109, 131).
- Field, Andy [2009]. *Discovering Statistics Using SPSS*. 3rd edition. Sage Publications, 2nd Mar 2009. ISBN 1847879071 (com, uk) (cited on page 110).
- Field, Andy and Graham Hole [2002]. *How to Design and Report Experiments*. Sage Publications, 2002. ISBN 0761973834 (com, uk) (cited on page 110).
- Field, Andy, Jeremy Miles and Zoe Field [2012]. *Discovering Statistics Using R*. Sage Publications, 31st Mar 2012. ISBN 1446200469 (com, uk) (cited on page 110).

- Fitch, John [1964]. *Computer Sketchpad*. WGBH Science Reporter. 00:29:57 video. 1964. http://www.wgbh.org/article?item_id=3360989 (cited on page 184).
- Flanders, Vincent and Dean Peters [2002]. *Son of Web Pages That Suck*. Sybex, 2002. ISBN 0782140203 (com, uk). <http://webpagesthatsuck.com/> (cited on page 7).
- Flanders, Vincent and Michael Willis [1998]. *Web Pages That Suck*. Sybex, 1998. ISBN 078212187X (com, uk). <http://webpagesthatsuck.com/> (cited on page 7).
- Foddy, William [1994]. *Constructing Questions for Interviews and Questionnaires: Theory and Practice in Social Research*. Cambridge University Press, 1994. ISBN 0521467330 (com, uk) (cited on page 140).
- Goldberg, Adele [1988]. *A History of Personal Workstations*. Addison-Wesley, Jan 1988. ISBN 0201112590 (com, uk) (cited on page 183).
- Goodwin, Kim and Alan Cooper [2009]. *Designing for the Digital Age: How to Create Human-Centered Products and Services*. Wiley, 24th Feb 2009. ISBN 0470229101 (com, uk) (cited on page 1).
- Greenberg, Saul [1997]. *Foundations and Principles of Human Computer Interaction*. An undergraduate HCI course. University of Calgary, 1997. <http://www.cpsc.ucalgary.ca/projects/grouplab/481/> (cited on page 34).
- GVU [1998]. *GVU's WWW User Surveys*. 1998. http://www.cc.gatech.edu/gvu/user_surveys/ (cited on page 140).
- Hackos, JoAnn T. and Janice C. Redish [1998]. *User and Task Analysis for Interface Design*. Wiley, 1998. ISBN 0471178314 (com, uk) (cited on page 53).
- Hampson, Mat [2010]. *A/B Testing*. BBC Web Developer Blog. 12th Jan 2010. <http://www.bbc.co.uk/blogs/webdeveloper/2010/01/ab-testing.shtml> (cited on page 138).
- Hegland, Frode and Fleur Klijnsma [2008]. *Invisible Revolution: The Doug Engelbart Story*. 2008. <http://invisiblerevolution.net/> (cited on page 187).
- Helander, Martin G., Thomas K. Landauer and Prasad v Prabhu, editors [1997]. *Handbook of Human-Computer Interaction*. 2nd edition. Elsevier, 1997. 1602 pages. ISBN 0444818626 (com, uk) (cited on page 3).
- Hewett, Thomas T. et al. [2002]. *ACM SIGCHI Curricula for Human-Computer Interaction*. ACM, 2002. ISBN 0897914740 (com, uk). <http://old.sigchi.org/cdg/> (cited on pages 1, 2).
- Hicks, Jon [2011]. *The Icon Handbook*. Five Simple Steps, Dec 2011. ISBN 1907828044 (com, uk) (cited on page 165).
- Holson, Laura M. [2009]. *Putting a Bolder Face on Google*. New York Times. 1st Mar 2009. <http://www.nytimes.com/2009/03/01/business/01marissa.html> (cited on page 138).
- Hopkins, Claude [1923]. *Scientific Advertising*. Lord & Thomas, 1923. ISBN 1607962365 (com, uk). <http://www.scientificadvertising.com/ScientificAdvertising.pdf> (cited on pages 137, 138).
- Horton, William [1994]. *The Icon Book*. Wiley, 1994. ISBN 0471599018 (com, uk) (cited on pages 165, 178).
- Hurst, Mark and Phil Terry [2000]. *The Dotcom Survival Guide*. Creative Good. 12th Jun 2000. <http://www.dad.be/library/pdf/creativegood.pdf> (cited on page 130).
- Isaacs, Ellen and Alan Walendowski [2001]. *Designing from Both Sides of the Screen*. Sams, 2001. ISBN 0672321513 (com, uk) (cited on pages 1, 61).

- ISO [1998]. *Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) – Part 11: Guidance on Usability (ISO 9241-11:1998)*. ISO. 1998. http://iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=16883 (cited on page 44).
- ISO [1999]. *Human-Centred Design Processes for Interactive Systems (ISO 13407:1999)*. ISO. 1999. http://iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=21197 (cited on page 44).
- Jacko, Julie A. and Andrew Sears, editors [2002]. *The Human-Computer Interaction Handbook*. Lawrence Erlbaum, 2002, page 1277. ISBN 0805844686 (com, uk) (cited on page 3).
- Johnson, Jeff [2003]. *Web Bloopers*. Morgan Kaufman, 2003. ISBN 1558608400 (com, uk) (cited on page 7).
- Johnson, Jeff [2007]. *GUI Bloopers 2.0*. Morgan Kaufman, Sep 2007. ISBN 0123706432 (com, uk) (cited on page 7).
- Kahn, Paul and Krzysztof Lenk [1998]. “Principles of Typography for User Interface Design”. *interactions* 5.6 (Nov 1998), pages 15–29. doi:10.1145/287821.287825 (cited on page 158).
- Kay, Alan [1987]. *Doing with Images Makes Symbols (Part 1)*. University Video Communications (UVC). 00:46:29 video. San Jose, California, USA, 1987. <http://www.archive.org/details/AlanKeyD1987> (cited on pages 184, 187).
- Kohavi, Ron, Thomas Crook and Roger Longbotham [2009a]. “Online Experimentation at Microsoft”. In: *Third Workshop on Data Mining Case Studies (DMCS 2009)*. Sep 2009. https://expplatform.sharepoint.com/Documents/Exp_DMCaseStudies.pdf (cited on page 139).
- Kohavi, Ron, Thomas Crook and Roger Longbotham [2009b]. “Online Experimentation at Microsoft”. In: *Microsoft ThinkWeek 2009*. Longer version of DMCS 2009 paper. Sep 2009. <https://expplatform.sharepoint.com/Documents/ExpThinkWeek2009Public.pdf> (cited on page 139).
- Kohavi, Ron, Randal M. Henne and Dan Sommerfield [2007]. “Practical Guide to Controlled Experiments on the web: Listen to Your Customers not to the HiPPO”. In: *Proc. 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007)*. ACM. San Jose, California, USA, Aug 2007, pages 959–967. doi:10.1145/1281192.1281295. <http://expplatform.com/Documents/GuideControlledExperiments.pdf> (cited on page 138).
- Kohavi, Ron and Roger Longbotham [2007]. “Online Experiments: Lessons Learned”. *Computer* 40.9 (Sep 2007), pages 103–105. ISSN 0018-9162. doi:10.1109/MC.2007.328. <http://kohavi.com/IEEEComputer2007OnlineExperiments.pdf> (cited on page 138).
- Kohavi, Ron, Roger Longbotham et al. [2009]. “Controlled Experiments on the Web: Survey and Practical Guide”. *Data Mining and Knowledge Discovery* 18.1 (Feb 2009), pages 140–181. ISSN 1384-5810. doi:10.1007/s10618-008-0114-1 (cited on page 137).
- Koonce, Jefferson M. [2002]. *Human Factors in the Training of Pilots*. CRC, 23rd May 2002. ISBN 0415253608 (com, uk) (cited on page 25).
- Krug, Steve [2006]. *Don’t Make Me Think! Web Usability - Das intuitive Web*. German. 2nd edition. mitp, Jul 2006. ISBN 3826615956 (com, uk) (cited on page 3).
- Krug, Steve [2009]. *Rocket Surgery Made Easy*. New Riders, 18th Dec 2009. 168 pages. ISBN 0321657292 (com, uk) (cited on page 1).
- Krug, Steve [2010]. *Web Usability: Rocket Surgery Made Easy*. German. Addison-Wesley Verlag, 16th Jun 2010. 192 pages. ISBN 3827329744 (com, uk) (cited on page 3).
- Krug, Steve [2013]. *Don’t Make Me Think, Revisited A Common Sense Approach to Web Usability*. 3rd edition. New Riders, 24th Dec 2013. 216 pages. ISBN 0321965515 (com, uk) (cited on page 1).

- Kuniavsky, Mike [2003]. *Observing the User Experience*. Morgan Kaufmann, 2003. ISBN 1558609237 (com, uk) (cited on page 53).
- Lakoff, George and Mark Johnson [1980]. *Metaphors We Live By*. 2nd edition. University Of Chicago Press, 1980. ISBN 0226468011 (com, uk) (cited on page 7).
- Landauer, Thomas [1988]. “Research Methods in HCI”. In: *Handbook of Human-Computer Interaction*. Edited by Martin G. Helander. North-Holland, 1988. Chapter 42, pages 905–928. ISBN 0444705368 (com, uk) (cited on page 109).
- Lazar, Jonathan, Jinjuan Heidi Feng and Harry Hochheiser [2010]. *Research Methods in Human-Computer Interaction*. Wiley, 2010. ISBN 0470723378 (com, uk) (cited on page 1).
- Lee, Leonard [1992]. *The Day the Phones Stopped Ringing*. Plume, 1992. ISBN 1556112866 (com, uk) (cited on pages 7, 35).
- Lewis, Clayton, Peter Polson et al. [1990]. “Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces”. In: *Proc. Conference on Human Factors in Computing Systems (CHI’90)*. ACM. Seattle, Washington, USA, Apr 1990, pages 235–242. doi:10.1145/97243.97279 (cited on page 101).
- Lewis, Clayton and John Rieman [1993]. *Task-Centered User Interface Design: A Practical Introduction*. Shareware book. University of Colorado, Boulder, 1993. <http://hcibib.org/tcuid/> (cited on pages 107, 108, 129).
- Lockee, Barbara, Mike Moore and John Burton [2002]. “Measuring Success: Evaluation Strategies for Distance Education”. *EDUCAUSE Quarterly* 25.1 (2002), pages 20–26. <http://www.educause.edu/ir/library/pdf/eqm0213.pdf> (cited on pages 46, 47).
- Maier, Norman R. F. [1931]. “Reasoning in Humans II. The Solution of a Problem and its Appearance in Consciousness”. *Journal of Comparative Psychology* 12.2 (Aug 1931), pages 181–194. <http://content.apa.org/journals/com/12/2/181> (cited on page 130).
- Manhartsberger, Martina and Sabine Musil [2001]. *Web Usability*. German. Galileo Press, Nov 2001. ISBN 3898421872 (com, uk) (cited on page 3).
- Marcus, Aaron [1992]. *Graphic Design for Electronic Documents and User Interfaces*. ACM Press (Addison-Wesley), 1992. ISBN 0201543648 (com, uk) (cited on page 157).
- Martinez, Michael J. [2000]. *Amazon Error May End ‘Dynamic Pricing’*. abc News. 29th Sep 2000. <http://abcnews.go.com/Technology/story?id=119399> (cited on page 139).
- Mayer, Marissa [2009]. “Design @ Google”. In: *AIGA Design Conference 2009*. AIGA. 9th Oct 2009. <http://www.aiga.org/content.cfm/video-makethink-2009-mayer> (cited on page 138).
- Mayhew, Deborah J. [1999]. *The Usability Engineering Lifecycle: A Practitioner’s Handbook for User Interface Design*. Morgan Kaufmann, May 1999. ISBN 1558605614 (com, uk) (cited on page 44).
- Microsoft [1999]. *Microsoft Windows User Experience*. Microsoft Press, 1999. ISBN 0735605661 (com, uk). <http://msdn.microsoft.com/library/books/winguide/welcome.htm> (cited on page 78).
- Microsoft [2005]. *Usability Research*. 2005. <http://microsoft.com/usability/> (cited on page 115).
- Miller, George A. [1956]. “The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information”. *The Psychological Review* 63.2 (Feb 1956), pages 81–97. <http://www.well.com/user/smalin/miller.html> (cited on page 24).
- Modley, Rudolf [1976]. *Handbook of Pictorial Symbols*. Dover Publications, 1976. ISBN 048623357X (com, uk) (cited on page 165).

- Molich, Rolf et al. [1998]. “Comparative Evaluation of Usability Tests”. In: *Proc. UPA98*. Usability Professionals Association. Washington, D.C., USA, Jun 1998. <http://www.dialogdesign.dk/cue.html> (cited on page 150).
- Mulder, Steve and Ziv Yaar [2006]. *The User Is Always Right: A Practical Guide to Creating and Using Personas for the Web*. New Riders, 2006. ISBN 0321434536 (com, uk) (cited on page 61).
- Mullet, Kevin and Darrell Sano [1994]. *Designing Visual Interfaces*. Prentice Hall, 1994. ISBN 0133033899 (com, uk) (cited on page 157).
- Myers, Brad A. [1998]. “A Brief History of Human-Computer Interaction Technology”. *interactions* 5.2 (Mar 1998), pages 44–54. ISSN 1072-5520. doi:10.1145/274430.274436 (cited on page 183).
- Newman, William M. and Michael G. Lamming [1995]. *Interactive System Design*. Addison-Wesley, 1995. ISBN 0201631628 (com, uk). <http://www.rxrc.xerox.com/publis/isd/home.htm> (cited on page 35).
- Nielsen, Jakob [1992]. “Finding Usability Problems Through Heuristic Evaluation”. In: *Proc. Conference on Human Factors in Computing Systems (CHI'92)*. ACM. Monterey, California, USA, May 1992, pages 373–380. doi:10.1145/142750.142834 (cited on pages 94–96).
- Nielsen, Jakob [1993a]. *Response Times: The Three Important Limits*. 1993. <http://www.useit.com/papers/responsetime.html> (cited on page 91).
- Nielsen, Jakob [1993b]. *Usability Engineering*. Morgan Kaufmann, 23rd Sep 1993. ISBN 0125184069 (com, uk) (cited on pages 44, 45, 54, 55, 60, 74, 87, 91, 135).
- Nielsen, Jakob [1994]. “Enhancing the Exploratory Power of Usability Heuristics”. In: *Proc. Conference on Human Factors in Computing Systems (CHI'94)*. ACM. Boston, Massachusetts, USA, Apr 1994, pages 152–158. doi:10.1145/191666.191729 (cited on page 90).
- Nielsen, Jakob [2003]. *Return on Investment for Usability*. The Alertbox. Jan 2003. <http://www.useit.com/alertbox/20030107.html> (cited on page 60).
- Nielsen, Jakob [2005]. *Top Ten Web Design Mistakes of 2005*. 2005. <http://www.useit.com/alertbox/designmistakes.html> (cited on page 77).
- Nielsen, Jakob and Robert L. Mack, editors [1994]. *Usability Inspection Methods*. Wiley, 1994. ISBN 0471018775 (com, uk) (cited on pages 89, 98).
- Nielsen, Jakob and Rolf Molich [1990]. “Heuristic Evaluation of User Interfaces”. In: *Proc. Conference on Human Factors in Computing Systems (CHI'90)*. ACM. Seattle, Washington, USA, Apr 1990, pages 249–256. doi:10.1145/97243.97281 (cited on pages 90, 92, 93).
- Nielsen, Jakob and Marie Tahir [2001]. *Homepage Usability: 50 Websites Deconstructed*. New Riders, Nov 2001. ISBN 073571102X (com, uk) (cited on page 106).
- Nisbett, Richard E. and Timothy D. Wilson [1977]. “Telling More Than We Can Know: Verbal Reports on Mental Processes.” *Psychological Review* 84.3 (Mar 1977), pages 231–259. <http://www.people.virginia.edu/~tdw/nisbett&wilson.pdf> (cited on page 130).
- NIST [2008]. *Industry Usability Reporting*. 2008. <http://zing.ncsl.nist.gov/iusr/> (cited on page 154).
- Norman, Donald A. [1988]. *The Psychology of Everyday Things*. Original, hardbound edition of The Design of Everyday Things. New York: Basic Books, 13th Jun 1988. ISBN 0465067093 (com, uk) (cited on pages 7, 21, 26, 30).
- Norman, Donald A. [1990]. “Human Error and the Design of Computer Systems”. *Communications of the ACM* 33.1 (Jan 1990), pages 4–7. doi:10.1145/76372.316030. http://www.jnd.org/dn.mss/commentary_huma.html (cited on page 35).

- Norman, Donald A. [1992]. *The Design of Everyday Things*. Paperback edition of The Psychology of Everyday Things. New York: Basic Books, 1992. ISBN 0465067107 (com, uk) (cited on pages 7, 17).
- Norman, Donald A. [1993]. *Turn Signals are the Facial Expressions of Automobiles*. Addison-Wesley, 1993. ISBN 020162236X (com, uk) (cited on page 7).
- Norman, Donald A. [1994]. *Things That Make Us Smart*. Addison-Wesley, 1994. ISBN 0201626950 (com, uk) (cited on page 7).
- Norman, Donald A. [1999]. “Affordance, Conventions, and Design”. *interactions* 6.3 (May 1999), pages 38–43. doi:10.1145/301153.301168. http://jnd.org/dn.mss/affordance_conv.html (cited on page 17).
- Norman, Donald A. [2013]. *The Design of Everyday Things*. 2nd edition. MIT Press, 23rd Dec 2013. ISBN 0262525674 (com, uk) (cited on page 7).
- Nyce, James M. and Paul Kahn [1991]. *From Memex to Hypertext: Vannevar Bush and the Mind’s Machine*. Academic Press, Dec 1991. ISBN 0125232705 (com, uk) (cited on page 184).
- Olson, Judith and Gary M. Olson [1990]. “The Growth of Cognitive Modeling in Human-Computer Interaction Since GOMS”. *Human-Computer Interaction* 5.2–3 (1990), pages 221–265 (cited on page 107).
- ORF [1987]. *Hoppala*. Originally from wir, rebroadcast on Hoppala outtake show, Austrian TV (ORF). 1987. <http://www.youtube.com/watch?v=2-p8YpR7rJc> (cited on page 8).
- PBSJ [2000]. *Bezos calls Amazon experiment ‘a mistake’*. Puget Sound Business Journal. 28th Sep 2000. <http://www.bizjournals.com/seattle/stories/2000/09/25/daily21.html> (cited on page 139).
- Pinker, Steven [1999]. *How the Mind Works*. Norton, Jan 1999. ISBN 0393318486 (com, uk) (cited on page 56).
- Plaisant, Catherine and Daniel Wallace [1992]. “Touchscreen Toggle Design”. In: *CHI’92 Technical Video Program*. University of Maryland. 1992. <http://www.acm.org/sigchi/video/chi92v.txt> (cited on page 135).
- Pogue, David [2006]. *Simplicity Sells*. TED 2006 Talk. 24th Feb 2006. http://www.ted.com/talks/david_pogue_says_simplicity_sells.html (cited on page 42).
- Preece, Jenny et al. [1994]. *Human-Computer Interaction*. Reading, MA: Addison-Wesley, 1994. ISBN 0201627698 (com, uk) (cited on page 3).
- Pruitt, John and Tamara Adlin [2006]. *The Persona Lifecycle*. Morgan Kaufmann, 2006. ISBN 0125662513 (com, uk) (cited on page 61).
- Raskin, Jef [2000]. *The Humane Interface*. Addison-Wesley, 2000. ISBN 0201379376 (com, uk) (cited on pages 1, 107).
- Rettig, Marc [1994]. “Prototyping for Tiny Fingers”. *Communications of the ACM* 37.4 (Apr 1994), pages 21–27. doi:10.1145/175276.175288 (cited on page 81).
- Rettig, Marc [1999]. *Interactive Sketch*. 1999. <http://www.enteract.com/~marc/intersketch/> (cited on page 85).
- Rhodes, John S. [2009]. *Selling Usability: User Experience Infiltration Tactics*. CreateSpace, 2009. ISBN 1442103736 (com, uk) (cited on page 59).
- Roscoe, Stanley N. [1997]. *The Adolescence of Engineering Psychology*. Volume 1. Human Factors History Monograph Series. Human Factors and Ergonomics Society, 1997. ISBN 0945289103 (com, uk). <http://www.hfes.org/Web/PubPages/adolescence.pdf> (cited on page 25).

- Rosson, Mary Beth and John M. Carroll [2001]. *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*. Morgan Kaufman, 2001. ISBN 1558607129 (com, uk) (cited on page 44).
- Rubin, Jeffrey [1994]. *Handbook of Usability Testing*. Wiley, 1994. ISBN 0471594032 (com, uk) (cited on pages 46, 112).
- Rubin, Jeffrey and Dana Chisnell [2008]. *Handbook of Usability Testing*. 2nd edition. Wiley, 2008. ISBN 0470185481 (com, uk) (cited on page 109).
- Schaffer, Eric [2004]. *Institutionalization of Usability*. Addison-Wesley, 2004. ISBN 032117934X (com, uk) (cited on page 59).
- Schrivver, Karen A. [1997]. *Dynamics in Document Design*. Wiley, 1997. ISBN 0471306363 (com, uk) (cited on page 157).
- Shaughnessy, John J., Eugene B. Zechmeister and Jeanne S. Zechmeister [2003]. *Research Methods in Psychology*. 6th edition. McGraw-Hill Education, May 2003. ISBN 0071113231 (com, uk) (cited on page 110).
- Shneiderman, Ben and Catherine Plaisant [2009]. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 5th edition. Addison-Wesley, Mar 2009. ISBN 0321601483 (com, uk) (cited on page 1).
- Snyder, Carolyn [2003]. *Paper Prototyping*. Morgan Kaufmann, 2003. ISBN 1558608702 (com, uk) (cited on page 81).
- Spencer, Rick [2000]. “The Streamlined Cognitive Walkthrough Method, Working Around Social Constraints Encountered in a Software Development Company”. In: *Proc. Conference on Human Factors in Computing Systems (CHI 2000)*. ACM. The Hague, The Netherlands, Apr 2000, pages 353–359. doi:10.1145/332040.332456 (cited on page 101).
- Spiekermann, Erik [2013]. *Stop Stealing Sheep & Find Out How Type Works*. 3rd edition. Adobe, 13th Dec 2013. ISBN 0321934288 (com, uk) (cited on page 158).
- Spolsky, Joel [2001]. *User Interface Design for Programmers*. Apress, 2001. ISBN 1893115941 (com, uk) (cited on page 61).
- Spool, Jared [2005]. *Avis: Trying Too Hard?* UIE Brainsparks. 5th Sep 2005. <http://www.uie.com/brainsparks/2005/09/05/avis-trying-too-hard/> (cited on pages xiii, 78, 79).
- Stake, Robert E. [1976]. *Evaluating Educational Programmes: The Need and the Response: A Collection of Resource Materials*. OECD, 1976. ISBN 9264115358 (com, uk) (cited on pages 43, 47).
- Stone, Debbie et al. [2005]. *User Interface Design and Evaluation*. Morgan Kaufmann, 2005. ISBN 0120884364 (com, uk) (cited on pages 1, 44, 46).
- Sun Microsystems, editor [1999]. *Java Look and Feel Design Guidelines*. Addison Wesley, Jul 1999. ISBN 0201615851 (com, uk). <http://java.sun.com/products/jlfd/dg/> (cited on page 78).
- Sutherland, Ivan E. [1963a]. “Sketchpad: A Man-Machined Graphical Communication System”. Original version scanned by MIT. PhD Dissertation. MIT, Jan 1963. <http://hdl.handle.net/1721.1/14979> (cited on page 184).
- Sutherland, Ivan E. [1963b]. “Sketchpad: A Man-Machined Graphical Communication System”. Archival version with preface by Alan Blackwell and Kerry Roddenphone. PhD Dissertation. MIT, Jan 1963. <http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-574.pdf> (cited on page 184).
- TechSmith [2008]. *Morae Customer Experience Software*. Nov 2008. <http://morae.com/> (cited on page 116).

- Thomas, Gregory [2003]. *How to Design Logos, Symbols and Icons*. How Design Books, 2003. ISBN 1581804563 (com, uk) (cited on page 165).
- Tognazzini, Bruce [1992]. *Tog on Interface*. Addison-Wesley, 1992. ISBN 0201608421 (com, uk) (cited on pages 3, 74, 110).
- Tognazzini, Bruce [1995]. *Tog on Software Design*. Addison-Wesley, 1995. ISBN 0201489171 (com, uk) (cited on page 3).
- Tognazzini, Bruce [1999]. *Maximizing Human Performance*. AskTog, March 1999. Mar 1999. <http://www.asktog.com/columns/023MaxHumanPerf.html> (cited on page 111).
- Travis, David [2013]. *ISO 9241 for Beginners*. Userfocus Ltd. May 2013. <http://userfocus.co.uk/articles/ISO9241.html> (cited on page 4).
- Tschichold, Jan [1997]. *The Form of the Book*. Hartley & Marks, 1997. ISBN 0881791164 (com, uk) (cited on page 158).
- UsabilityNet [2005]. *EU Usability Network*. 2005. <http://www.usabilitynet.org/> (cited on page 154).
- USAF [1942]. *B-17F Airplane Pilot's Flight Operating Instructions*. US Army Air Forces, 25th Dec 1942. <http://b17panels.com/id23.html> (cited on page 29).
- USAF [2006]. *Boeing B-17G Flying Fortress Cockpit*. National Museum of the U.S. Air Force, Dayton, Ohio. 20th Oct 2006. http://commons.wikimedia.org/wiki/File:B-17_Flying_Fortress_Cockpit.jpg (cited on page 28).
- Van Someren, Maarten W., Yvonne F. Barnard and Jacobijn A. C. Sandberg [1994]. *The Think Aloud Method*. Academic Press, 1994. ISBN 0127142703 (com, uk) (cited on page 109).
- WAMMI [2000]. *Web Usability Questionnaire*. 2000. <http://www.wammi.com> (cited on page 140).
- Wharton, Cathleen et al. [1994]. "The Cognitive Walkthrough Method: A Practitioner's Guide". In: *Usability Inspection Methods*. Edited by Jakob Nielsen and Robert L. Mack. Wiley, 1994, pages 105–140. ISBN 0471018775 (com, uk) (cited on pages 101, 103).
- Wickens, Christopher D. et al. [2003]. *Introduction to Human Factors Engineering*. 2nd edition. Prentice Hall, 2003. ISBN 0131837362 (com, uk) (cited on page 3).
- Wikipedia [2009]. *The Mother of All Demos*. 2009. http://en.wikipedia.org/wiki/The_Mother_of_All_Demos (cited on page 186).
- Wiklund, Michael E., editor [1994]. *Usability in Practice*. Academic Press, 1994. ISBN 0127512500 (com, uk) (cited on page 147).
- Williams, Robin [2005]. *Non-Designer's Type Book*. 2nd edition. Peachpit Press, 2005. ISBN 0321303369 (com, uk) (cited on page 158).
- Williams, Robin [2008]. *The Non-Designer's Design Book*. 3rd edition. Peachpit Press, 12th Feb 2008. ISBN 0321534042 (com, uk) (cited on page 157).