

1

Designing the User Interface

*Strategies for Effective
Human-Computer Interaction*
Third Edition

AUG 4 1998

~~DEPARTMENT OF COMPUTER SCIENCE~~
UNIVERSITY OF TORONTO

Ben Shneiderman
The University of Maryland

 **ADDISON-WESLEY**

An imprint of Addison Wesley Longman, Inc.

Reading, Massachusetts • Harlow, England • Menlo Park, California
Berkeley, California • Don Mills, Ontario • Sydney • Bonn • Amsterdam
Tokyo • Mexico City

META-1012
IPR2025-00430

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed in initial caps or all caps.

Library of Congress Cataloging-in-Publication Data

Shneiderman, Ben.

Designing the user interface : strategies for effective human
-computer-interaction / Ben Shneiderman. -- 3rd ed.

800P A cm.
Includes bibliographical references and index.

ISBN 0-201-69497-2

1. Human-computer interaction. 2. User interfaces (Computer

QA76.9.H85S54 v.1998

004'.01'9--dc21

96-37974

CIP

Access the latest information about Addison-Wesley titles from our World Wide
Web site: <http://www.awl.com/cseng>

Chapter opener illustrations from art provided by Mark Kostabi. Reproduced
with permission.

Cover art © Boris Lyubner/SIS

Copyright © 1998 by Addison Wesley Longman, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a
retrieval system, or transmitted, in any form or by any means, electronic,
mechanical, photocopying, recording, or otherwise, without the prior written
permission of the publisher. Printed in the United States of America.

1 2 3 4 5 6 7 8 9 10-MA-0100999897

Preface to the Third Edition

Designing the User Interface is intended primarily for designers, managers, and evaluators of interactive systems. It presents a broad survey of designing, implementing, managing, maintaining, training, and refining the user interface of interactive systems. The book's second audience is researchers in human-computer interaction, specifically those who are interested in human performance with interactive systems. These researchers may have backgrounds in computer science, psychology, information systems, library science, business, education, human factors, ergonomics, or industrial engineering; all share a desire to understand the complex interaction of people with machines. Students in these fields also will benefit from the contents of this book. It is my hope that this book will stimulate the introduction of courses on user-interface design in all these and other disciplines. Finally, serious users of interactive systems will find that the book gives them a more thorough understanding of the design questions for user interfaces. My goals are to encourage greater attention to the user interface and to help develop a rigorous science of user-interface design.

Since publication of the first two editions of this book in 1986 and 1992, researchers in the field of human-computer interaction and practitioners of user-interface design have grown more numerous and influential. The quality of interfaces has improved greatly, and the community of users has grown dramatically. Researchers and designers could claim success, but user expectations are higher and the applications are more demanding. Today's interfaces are good, but novice and expert users still experience anxiety and frustration all too often. To achieve the goal of universal access, designers will have to continue to work harder. This book is meant to help them keep up the momentum, and thus to encourage further progress.

Keeping up with the innovations in human-computer interaction is a demanding task. Requests for an update to my second edition began shortly after its publication, but I had to wait until a sabbatical year allowed me to set aside enough time to complete this third edition. I've gone to the library, the World Wide Web, conferences, and colleagues to harvest information, and then returned to my keyboard to write. My first drafts were only a starting point to generate feedback from colleagues, practitioners, and students. The work was intense and satisfying.

New in the Third Edition

Comments from instructors who used the second edition were influential in my revisions of the structure. Since many courses include design, evaluation,

and construction projects, the chapters on development methodologies, evaluation techniques, and software tools were moved toward the beginning. Since direct manipulation is the dominant user-interface style, it is presented first, followed by menus, form fillin, and command languages. The material on computer-supported cooperative work has changed dramatically as research ideas and prototypes have become commercial tools. Information visualization is still in its early phases, but vigorous research and emerging commercial activity are widespread. The closing chapter on the rapidly growing World Wide Web is totally new.

Instructors wanted more guidelines and summary tables; these elements are now shown in boxes throughout the book. The Practitioner Summaries and Researcher Agendas remain popular; they have been updated. The references have been expanded and freshened with many new sources, with classic papers still included. Because some of the previously cited works were difficult to find, a much larger percentage of the references now are widely available sources. Figures—especially those showing screen designs—age quickly. In this edition, numerous new user interfaces are shown, many in full color.

Readers will see the dynamism of human-computer interaction reflected in the substantial changes to this third edition. Controversy continues about the future of speech input and output, natural-language interaction, anthropomorphic design, and agents. I emphasize empirical reports, try to present both sides fairly, and offer my opinions.

The presence of the World Wide Web has a profound effect on researchers, designers, educators, and students. I want to encourage intense use of the web by all these groups and to ease integration of the web into common practice. However, the volatility of the web is not in harmony with the permanence of printed books. Publishing website URLs in the book would have been risky, because changes are made daily. For these and other reasons, with the cooperation of my publisher and Prof. Blaise Liffick (Millersville University), we have established an ambitious web site (<http://www.aw.com/DTUI>) to accompany this book. It contains pointers to web sites related to each chapter's topics, updates on fast-changing topics, interesting reviews, and instructional support. Exercises, homework assignments, projects, and examination questions are just a few of the elements of this evolving site. Contributions from professionals, faculty, and students are making this resource increasingly valuable, and the community using it is lively and growing. I hope that every reader will visit the site, will participate in discussion groups, and will contribute to it. Send us your ideas and contributions.

Ways to Use This Book

I hope that practitioners and researchers who read this book will want to keep it on their shelves to consult when they are working on a new topic or seeking pointers to the literature.

Instructors may choose to assign the full text in the order that I present it, or to make selections from it. The opening chapter is a good starting point for most students, but instructors may take different paths depending on their disciplines. For example, instructors might emphasize the following chapters, listed by area:

- Computer science: 2, 5, 6, 13, 14, 15
- Psychology: 2, 4, 9, 10, 14
- Library and information science: 2, 4, 12, 15, 16
- Business and information systems: 3, 4, 14, 15
- Education technology: 2, 4, 11, 12, 14, 16
- Communication arts and media studies: 4, 11, 12, 16
- Technical writing and graphic design: 3, 4, 11, 12, 15, 16

The book's web site provides syllabi from many instructors, and offers supplemental teaching materials.

Acknowledgments

Writing is a lonely process; revising is a social one. I am grateful to the many colleagues and students who contributed their suggestions. My close daily partners at the University of Maryland have the greatest influence and my deepest appreciation: Gary Marchionini, Kent Norman, Catherine Plaisant, and Anne Rose. I give special thanks to Charles Kreitzberg and Jenny Preece for their personal and professional support. Other major contributors of useful comments include Richard Bellaver, Tom Bruns, Stephan Greene, Jesse Heines, Eser Kandogan, Chris North, Arkady Pogostkin, Richard Potter, Marilyn Saltzman, Michael Spring, Egemen Tanin, and Craig Wills. The many people and organizations that provided figures are acknowledged in the relevant captions.

I also appreciate the students around the world who sent me comments and suggestions. Their provocative questions about our growing discipline and profession encourage me daily.

Ben Shneiderman (ben@cs.umd.edu)

Contents

CHAPTER 1	HUMAN FACTORS OF INTERACTIVE SOFTWARE	3
1.1	Introduction	4
1.2	Goals of System Engineering	9
1.2.1	Proper functionality	11
1.2.2	Reliability, availability, security, and data integrity	12
1.2.3	Standardization, integration, consistency, and portability	13
1.2.4	Schedules and budgets	14
1.3	Goals of User-Interface Design	14
1.4	Motivations for Human Factors in Design	16
1.4.1	Life-critical systems	16
1.4.2	Industrial and commercial uses	16
1.4.3	Office, home, and entertainment applications	17
1.4.4	Exploratory, creative, and cooperative systems	17
1.5	Accommodation of Human Diversity	18
1.5.1	Physical abilities and physical workplaces	18
1.5.2	Cognitive and perceptual abilities	20
1.5.3	Personality differences	21
1.5.4	Cultural and international diversity	23
1.5.5	Users with disabilities	24
1.5.6	Elderly users	26
1.6	Goals for Our Profession	28
1.6.1	Influencing academic and industrial researchers	28
1.6.2	Providing tools, techniques, and knowledge for systems implementers	31
1.6.3	Raising the computer consciousness of the general public	31
1.7	Practitioner's Summary	32
1.8	Researcher's Agenda	32
CHAPTER 2	THEORIES, PRINCIPLES, AND GUIDELINES	51
2.1	Introduction	52
2.2	High-Level Theories	53
2.2.1	Conceptual, semantic, syntactic, and lexical model	54
2.2.2	GOMS and the keystroke-level model	55
2.2.3	Stages of action models	57
2.2.4	Consistency through grammars	58
2.2.5	Widget-level theories	60
2.3	Object-Action Interface Model	61
2.3.1	Task hierarchies of objects and actions	63

2.3.2	Interface hierarchies of objects and actions	64
2.3.3	The disappearance of syntax	65
2.4	Principle 1: Recognize the Diversity	67
2.4.1	Usage profiles	67
2.4.2	Task profiles	70
2.4.3	Interaction styles	71
2.5	Principle 2: Use the Eight Golden Rules of Interface Design	74
2.6	Principle 3: Prevent Errors	76
2.6.1	Correct matching pairs	77
2.6.2	Complete sequences	77
2.6.3	Correct commands	78
→ 2.7	Guidelines for Data Display	79
2.7.1	Organizing the display	80
2.7.2	Getting the user's attention	81
2.8	Guidelines for Data Entry	82
2.9	Balance of Automation and Human Control	83
2.10	Practitioner's Summary	89
2.11	Researcher's Agenda	90

CHAPTER 3 MANAGING DESIGN PROCESSES 95

3.1	Introduction	96
3.2	Organizational Design to Support Usability	97
3.3	The Three Pillars of Design	100
3.3.1	Guidelines documents and processes	100
3.3.2	User-interface software tools	102
3.3.3	Expert reviews and usability testing	103
3.4	Development Methodologies	104
3.5	Ethnographic Observation	107
3.6	Participatory Design	109
3.7	Scenario Development	111
3.8	Social Impact Statement for Early Design Review	113
3.9	Legal Issues	115
3.10	Practitioner's Summary	118
3.11	Researcher's Agenda	118

CHAPTER 4 EXPERT REVIEWS, USABILITY TESTING, SURVEYS, AND CONTINUING ASSESSMENTS 123

4.1	Introduction	124
4.2	Expert Reviews	125
4.3	Usability Testing and Laboratories	127
4.4	Surveys	132

- 4.5 Acceptance Tests 135
- 4.6 Evaluation During Active Use 145
 - 4.6.1 Interviews and focus-group discussions 145
 - 4.6.2 Continuous user-performance data logging 146
 - 4.6.3 Online or telephone consultants 147
 - 4.6.4 Online suggestion box or trouble reporting 147
 - 4.6.5 Online bulletin board or newsgroup 148
 - 4.6.6 User newsletters and conferences 148
- 4.7 Controlled Psychologically Oriented Experiments 149
- 4.8 Practitioner's Summary 150
- 4.9 Researcher's Agenda 151

CHAPTER 5 SOFTWARE TOOLS 155

- 5.1 Introduction 156
- 5.2 Specification Methods 157
 - 5.2.1 Grammars 158
 - 5.2.2 Menu-selection and dialog-box trees 160
 - 5.2.3 Transition diagrams 160
 - 5.2.4 Statecharts 162
 - 5.2.5 User-action notation (UAN) 163
- 5.3 Interface-Building Tools 166
 - 5.3.1 Design tools 168
 - 5.3.2 Software-engineering tools 169
- 5.4 Evaluation and Critiquing Tools 177
- 5.5 Practitioner's Summary 179
- 5.6 Researcher's Agenda 181

CHAPTER 6 DIRECT MANIPULATION AND VIRTUAL ENVIRONMENTS 185

- 6.1 Introduction 186
- 6.2 Examples of Direct-Manipulation Systems 187
 - 6.2.1 Command-line versus display editors versus word processors 187
 - 6.2.2 The VisiCalc spreadsheet and its descendants 191
 - 6.2.3 Spatial data management 192
 - 6.2.4 Video games 193
 - 6.2.5 Computer-aided design 197
 - 6.2.6 Office automation 199
 - 6.2.7 Further examples of direct manipulation 201
- 6.3 Explanations of Direct Manipulation 202

6.3.1	Problems with direct manipulation	204
6.3.2	The OAI model explanation of direct manipulation	205
6.4	Visual Thinking and Icons	207
6.5	Direct-Manipulation Programming	210
6.6	Home Automation	213
6.7	Remote Direct Manipulation	217
6.8	Virtual Environments	221
6.9	Practitioner's Summary	228
6.10	Researcher's Agenda	229

CHAPTER 7 MENU SELECTION, FORM FILLIN, AND DIALOG BOXES 235

7.1	Introduction	236
7.2	Task-Related Organization	237
7.2.1	Single menus	238
7.2.2	Linear sequences and multiple menus	247
7.2.3	Tree-structured menus	247
7.2.4	Acyclic and cyclic menu networks	252
7.3	Item Presentation Sequence	252
7.4	Response Time and Display Rate	254
7.5	Fast Movement Through Menus	255
7.5.1	Menus with typeahead: The BLT approach	255
7.5.2	Menu names or bookmarks for direct access	256
7.5.3	Menu macros, custom toolbars, and style sheets	257
7.6	Menu Layout	257
7.6.1	Titles	257
7.6.2	Phrasing of menu items	259
7.6.3	Graphic layout and design	259
7.7	Form Fillin	262
7.7.1	Form-fillin design guidelines	262
7.7.2	List and combo boxes	265
7.7.3	Coded fields	266
7.8	Dialog Boxes	268
7.9	Practitioner's Summary	270
7.10	Researcher's Agenda	270

CHAPTER 8 COMMAND AND NATURAL LANGUAGES 275

8.1	Introduction	276
8.2	Functionality to Support Users' Tasks	280

8.3	Command-Organization Strategies	282
8.3.1	Single command set	282
8.3.2	Command plus arguments	282
8.3.3	Command plus options and arguments	284
8.3.4	Hierarchical command structure	285
8.4	The Benefits of Structure	285
8.4.1	Consistent argument ordering	286
8.4.2	Symbols versus keywords	286
8.4.3	Hierarchical structure and congruence	287
8.5	Naming and Abbreviations	289
8.5.1	Specificity versus generality	289
8.5.2	Abbreviation strategies	290
8.5.3	Guidelines for using abbreviations	291
8.6	Command Menus	292
8.7	Natural Language in Computing	293
8.7.1	Natural-language interaction	294
8.7.2	Natural-language queries	296
8.7.3	Test-database searching	297
8.7.4	Natural-language text generation	300
8.7.5	Adventure and educational games	300
8.8	Practitioner's Summary	300
8.9	Researcher's Agenda	301
 CHAPTER 9 INTERACTION DEVICES		305
9.1	Introduction	306
9.2	Keyboards and Function Keys	307
9.2.1	Keyboard layouts	308
9.2.2	Keys	311
9.2.3	Function keys	312
9.2.4	Cursor movement keys	313
9.3	Pointing Devices	315
9.3.1	Pointing tasks	315
9.3.2	Direct-control pointing devices	316
9.3.3	Indirect-control pointing devices	319
9.3.4	Comparisons of pointing devices	323
9.3.5	Fitts' Law	325
9.3.6	Novel pointing devices	326
9.4	Speech Recognition, Digitization, and Generation	327
9.4.1	Discrete-word recognition	328
9.4.2	Continuous-speech recognition	331
9.4.3	Speech store and forward	332

- 9.4.4 Speech generation 333
- 9.4.5 Audio tones, audiolization, and music 335
- 9.5 Image and Video Displays 336
 - 9.5.1 Display devices 336
 - 9.5.2 Digital photography and scanners 339
 - 9.5.3 Digital video 339
 - 9.5.4 Projectors, heads-up displays, helmet-mounted displays 341
- 9.6 Printers 342
- 9.7 Practitioner's Summary 343
- 9.8 Researcher's Agenda 344

CHAPTER 10 RESPONSE TIME AND DISPLAY RATE 351

- 10.1 Introduction 352
- 10.2 Theoretical Foundations 354
 - 10.2.1 Limitations of short-term and working memory 355
 - 10.2.2 Sources of errors 356
- 10.3 Expectations and Attitudes 358
- 10.4 User Productivity 361
 - 10.4.1 Repetitive tasks 361
 - 10.4.2 Problem-solving tasks 362
 - 10.4.3 Summary 364
- 10.5 Variability 364
- 10.6 Practitioner's Summary 366
- 10.7 Researcher's Agenda 367

CHAPTER 11 PRESENTATION STYLES: BALANCING FUNCTION AND FASHION 371

- 11.1 Introduction 372
- 11.2 Error Messages 373
 - 11.2.1 Specificity 374
 - 11.2.2 Constructive guidance and positive tone 375
 - 11.2.3 User-centered phrasing 376
 - 11.2.4 Appropriate physical format 376
 - 11.2.5 Development of effective messages 377
- 11.3 Nonanthropomorphic Design 380
- 11.4 Display Design 384
 - 11.4.1 Field layout 387
 - 11.4.2 Empirical results 389
 - 11.4.3 Display-complexity metrics 391
- 11.5 Color 398

11.6	Practitioner's Summary	403
11.7	Researcher's Agenda	403

CHAPTER 12 PRINTED MANUALS, ONLINE HELP, AND TUTORIALS 409

12.1	Introduction	410
12.2	Reading from Paper versus from Displays	412
12.3	Preparation of Printed Manuals	414
12.3.1	Use of the OAI Model to design manuals	415
12.3.2	Organization and writing style	417
12.3.3	Nonanthropomorphic descriptions	421
12.3.4	Development process	423
12.4	Preparation of Online Facilities	425
12.4.1	Online manuals	428
12.4.2	Online tutorials, demonstrations, and animations	434
12.4.3	Helpful guides	436
12.5	Practitioner's Summary	437
12.6	Researcher's Agenda	438

CHAPTER 13 MULTIPLE-WINDOW STRATEGIES 443

13.1	Introduction	444
13.2	Individual-Window Design	448
13.3	Multiple-Window Design	455
13.4	Coordination by Tightly-Coupled Windows	458
13.5	Image Browsing and Tightly-Coupled Windows	462
13.6	Personal Role Management and Elastic Windows	468
13.7	Practitioner's Summary	472
13.8	Researcher's Agenda	472

CHAPTER 14 COMPUTER-SUPPORTED COOPERATIVE WORK 477

14.1	Introduction	478
14.2	Goals of Cooperation	479
14.3	Asynchronous Interactions: Different Time, Different Place	482
14.3.1	Electronic Mail	483
14.3.2	Newsgroups and network communities	485
14.4	Synchronous Distributed: Different Place, Same Time	488
14.5	Face to Face: Same Place, Same Time	494
14.6	Applying CSCW to Education	498
14.7	Practitioner's Summary	502
14.8	Researcher's Agenda	503

CHAPTER 15 INFORMATION SEARCH AND VISUALIZATION 509

- 15.1 Introduction 510
- 15.2 Database Query and Phrase Search in Textual Documents 513
- 15.3 Multimedia Document Searches 519
- 15.4 Information Visualization 522
- 15.5 Advanced Filtering 541
- 15.6 Practitioner's Summary 544
- 15.7 Researcher's Agenda 544

CHAPTER 16 HYPERMEDIA AND THE WORLD WIDE WEB 551

- 16.1 Introduction 552
- 16.2 Hypertext and Hypermedia 556
- 16.3 World Wide Web 560
- 16.4 Genres and Goals and Designers 562
- 16.5 Users and Their Tasks 565
- 16.6 Object-Action Interface Model for Web Site Design 567
 - 16.6.1 Design of task objects and actions 567
 - 16.6.2 Design of interface objects and actions 569
 - 16.6.3 Case study with the Library of Congress 571
 - 16.6.4 Detailed design issues 572
 - 16.6.5 Web-Page design 575
 - 16.6.6 Testing and maintenance of web sites 579
- 16.7 Practitioner's Summary 580
- 16.8 Researcher's Agenda 580

**AFTERWORD SOCIETAL AND INDIVIDUAL IMPACT OF
USER INTERFACES 585**

- A.1 Between Hope and Fear 586
- A.2 Ten Plagues of the Information Age 592
- A.3 Prevention of the Plagues 596
- A.4 Overcoming the Obstacle of Animism 597
- A.5 In the Long Run 600
- A.6 Practitioner's Summary 601
- A.7 Researcher's Agenda 601

Name Index 605

Subject Index 621

Designing the User Interface

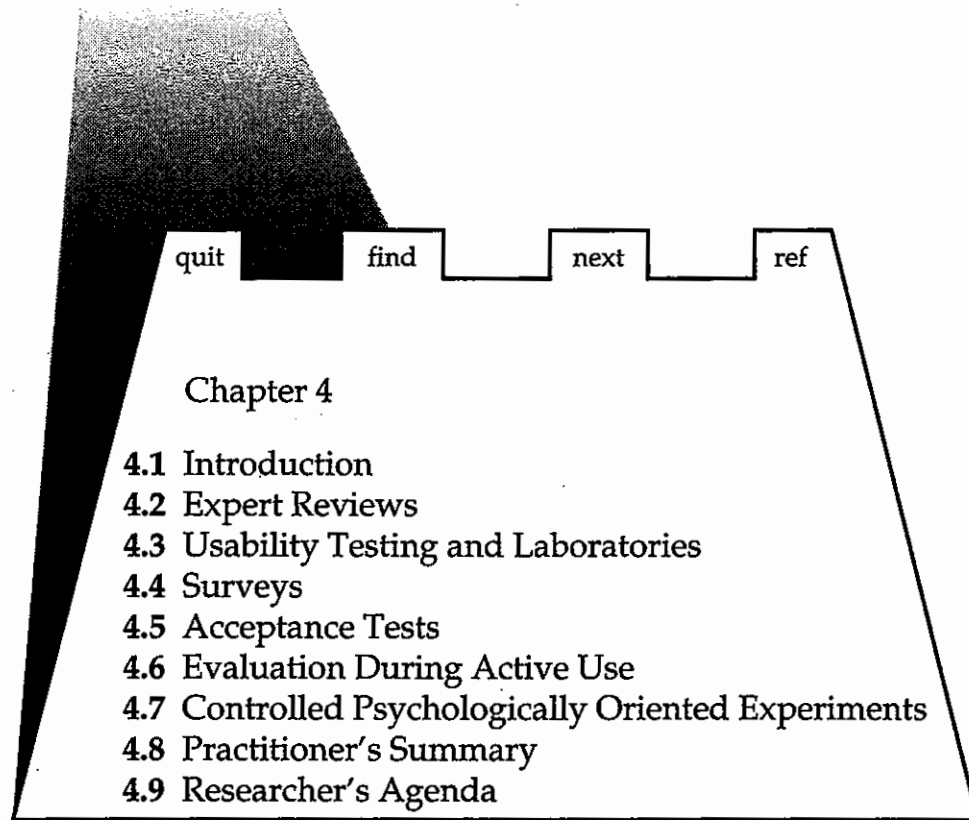
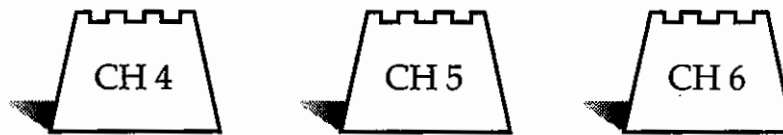
*Strategies for Effective
Human-Computer
Interaction*

Third Edition

Expert Reviews, Usability Testing, Surveys, and Continuing Assessments

The test of what is real is that it is hard and rough.
... What is pleasant belongs in dreams.

Simone Weil, *Gravity and Grace*, 1947



4.1 Introduction

Designers can become so entranced with their creations that they may fail to evaluate those objects adequately. Experienced designers have attained the wisdom and humility to know that extensive testing is a necessity. If feedback is the "breakfast of champions," then testing is the "dinner of the gods." However, careful choices must be made from the large menu of evaluation possibilities to create a balanced meal.

The determinants of the evaluation plan include (Nielsen, 1993; Hix and Hartson, 1993; Preece et al., 1994; Newman and Lamming, 1995)

- Stage of design (early, middle, late)
- Novelty of project (well defined versus exploratory)
- Number of expected users

- Criticality of the interface (for example, life-critical medical system versus museum-exhibit support)
- Costs of product and finances allocated for testing
- Time available
- Experience of the design and evaluation team

The range of evaluation plans might be from an ambitious two-year test with multiple phases for a new national air-traffic-control system to a three-day test with six users for a small internal accounting system. The range of costs might be from 10 percent of a project down to 1 percent.

A few years ago, it was just a good idea to get ahead of the competition by focusing on usability and doing testing, but now the rapid growth of interest in usability means that failure to test is risky indeed. The dangers are not only that the competition has strengthened, but also that customary engineering practice now requires adequate testing. Failure to perform and document testing could lead to failed contract proposals or malpractice lawsuits from users when errors arise. At this point, it is irresponsible to bypass some form of usability testing.

One troubling aspect of testing is the uncertainty that remains even after exhaustive testing by multiple methods. Perfection is not possible in complex human endeavors, so planning must include continuing methods to assess and repair problems during the lifecycle of an interface. Second, even though problems may continue to be found, at some point a decision has to be made about completing prototype testing and delivering the product. Third, most testing methods will account appropriately for normal usage, but performance with high levels of input such as in nuclear-reactor-control or air-traffic-control emergencies is extremely difficult to test. Development of testing methods to deal with stressful situations and even with partial equipment failures will have to be undertaken as user interfaces are developed for an increasing number of life-critical applications.

The Usability Professionals Association was founded in 1991 to exchange information among workers in this arena. The annual conference focuses attention on forms of usability evaluations and provides a forum for exchanges of ideas among the more than 4000 members.

4.2 Expert Reviews

While informal demos to colleagues or customers can provide some useful feedback, more formal expert reviews have proved to be effective (Nielsen and Mack, 1994). These methods depend on having experts available on staff or as consultants, whose expertise may be in the application or user-interface domains. Expert reviews can be conducted on short notice and rapidly.

Expert reviews can occur early or late in the design phase, and the outcomes can be a formal report with problems identified or recommendations for changes. Alternatively, the expert review could result in a discussion with or presentation to designers or managers. Expert reviewers should be sensitive to the design team's ego involvement and professional skill, so suggestions should be made cautiously: It is difficult for someone just freshly inspecting a system to understand all the design rationale and development history. The reviewer notes possible problems for discussion with the designers, but solutions generally should be left for the designers to produce. Expert reviews usually entail half day to one week, although a lengthy training period may be required to explain the task domain or operational procedures. It may be useful to have the same as well as fresh expert reviewers as the project progresses. There are a variety of expert-review methods from which to choose:

- *Heuristic evaluation* The expert reviewers critique an interface to determine conformance with a short list of design heuristics such as the eight golden rules (Chapter 2). It makes an enormous difference if the experts are familiar with the rules and are able to interpret and apply them.
- *Guidelines review* The interface is checked for conformance with the organizational or other guidelines document. Because guidelines documents may contain a thousand items, it may take the expert reviewers some time to master the guidelines, and days or weeks to review a large system.
- *Consistency inspection* The experts verify consistency across a family of interfaces, checking for consistency of terminology, color, layout, input and output formats, and so on within the interface as well as in the training materials and online help.
- *Cognitive walkthrough* The experts simulate users walking through the interface to carry out typical tasks. High-frequency tasks are a starting point, but rare critical tasks, such as error recovery, also should be walked through. Some form of simulating the day in the life of the user should be part of expert-review process. Cognitive walkthroughs were developed for interfaces that can be learned by exploratory browsing (Wharton et al., 1994), but they are useful even for interfaces that require substantial training. An expert might try the walkthrough privately and explore, but then there also would be a group meeting with designers, users, or managers to conduct the walkthrough and to provoke a discussion. This public walkthrough is based on the successful code walkthroughs promoted in software engineering (Yourdon, 1989).
- *Formal usability inspection* The experts hold courtroom-style meeting, with a moderator or judge, to present the interface and to discuss its merits and weaknesses. Design-team members may rebut the evidence about problems in an adversarial format. Formal usability inspections

can be educational experiences for novice designers and managers, but they may take longer to prepare and more personnel to carry out than do other types of review.

Expert reviews can be scheduled at several points in the development process when experts are available and when the design team is ready for feedback. The number of expert reviews will depend on the magnitude of the project and on the amount of resources allocated.

Comparative evaluation of expert-review methods and usability-testing methods is difficult because of the many uncontrollable variables; however, the studies that have been conducted provide evidence for the benefits of expert reviews (Jeffries et al., 1991; Karat et al. 1992). Different experts tend to find different problems in an interface, so three to five expert reviewers can be highly productive, as can complementary usability testing.

Expert reviewers should be placed in the situation most similar to the one that intended users will experience. The expert reviewers should take training courses, read manuals, take tutorials, and try the system in as close as possible to a realistic work environment, complete with noise and distractions. In addition, expert reviewers may also retreat to a quieter environment for detailed review of each screen.

Getting a *bird's-eye view* of an interface by studying a full set of printed screens laid out on the floor or pinned to walls has proved to be enormously fruitful in detecting inconsistencies and spotting unusual patterns.

The dangers with expert reviews are that the experts may not have an adequate understanding of the task domain or user communities. Experts come in many flavors, and conflicting advice can further confuse the situation (cynics say, "For every PhD, there is an equal and opposite PhD"). To strengthen the possibility of successful expert review, it helps to choose knowledgeable experts who are familiar with the project situation and who have a long-term relationship with the organization. These people can be called back to see the results of their intervention, and they can be held accountable. Moreover, even experienced expert reviewers have great difficulty knowing how typical users, especially first-time users, will behave.

4.3 Usability Testing and Laboratories

The emergence of usability testing and laboratories since the early 1980s is an indicator of the profound shift in attention to user needs. Traditional managers and developers resisted at first, saying that usability testing seemed like a nice idea, but that time pressures or limited resources prevented them from trying it. As experience grew and successful projects gave credit to the testing process, demand swelled and design teams began to compete for the scarce

resource of the usability-laboratory staff. Managers came to realize that having a usability test on the schedule was a powerful incentive to complete a design phase. The usability-test report provided supportive confirmation of progress and specific recommendations for changes. Designers sought the bright light of evaluative feedback to guide their work, and managers saw fewer disasters as projects approached delivery dates. The remarkable surprise was that usability testing not only sped up many projects, but also produced dramatic cost savings (Gould, 1988; Gould et al., 1991; Karat, 1994).

Usability-laboratory advocates split from their academic roots as these practitioners developed innovative approaches that were influenced by advertising and market research. While academics were developing controlled experiments to test hypotheses and support theories, practitioners developed usability-testing methods to refine user interfaces rapidly. Controlled experiments have at least two treatments and seek to show statistically significant differences; usability tests are designed to find flaws in user interfaces. Both strategies use a carefully prepared set of tasks, but usability tests have fewer subjects (maybe as few as three), and the outcome is a report with recommended changes, as opposed to validation or rejection of hypotheses. Of course, there is a useful spectrum of possibilities between rigid controls and informal testing, and sometimes a combination of approaches is appropriate.

The movement toward usability testing stimulated the construction of usability laboratories (Dumas and Redish, 1993; Nielsen, 1993). Many organizations spent modest sums to build a single usability laboratory, while IBM built an elaborate facility in Boca Raton, Florida, with 16 laboratories in a circular arrangement with a centralized database for logging usage and recording performance. Having a physical laboratory makes an organization's commitment to usability clear to employees, customers, and users (Nielsen, 1994) (Fig. 4.1). A typical modest usability laboratory would have two 10- by 10-foot areas, one for the participants to do their work and another, divided by a half-silvered mirror, for the testers and observers (designers, managers, and customers) (Fig. 4.2). IBM was an early leader in developing usability laboratories, Microsoft started later, but embraced the idea forcefully, and hundreds of software-development companies have followed suit. A consulting community that will do usability testing for hire also has emerged.

The usability laboratory is typically staffed by one or more people with expertise in testing and user-interface design, who may serve 10 to 15 projects per year throughout the organization. The laboratory staff meet with the user-interface architect or manager at the start of the project to make a test plan with scheduled dates and budget allocations. Usability-laboratory staff participate in early task analysis or design reviews, provide information on software tools or literature references, and help to develop the set of tasks for the usability test. Two to six weeks before the usability test, the detailed test plan is developed, comprising the list of tasks, plus subjective satisfac-

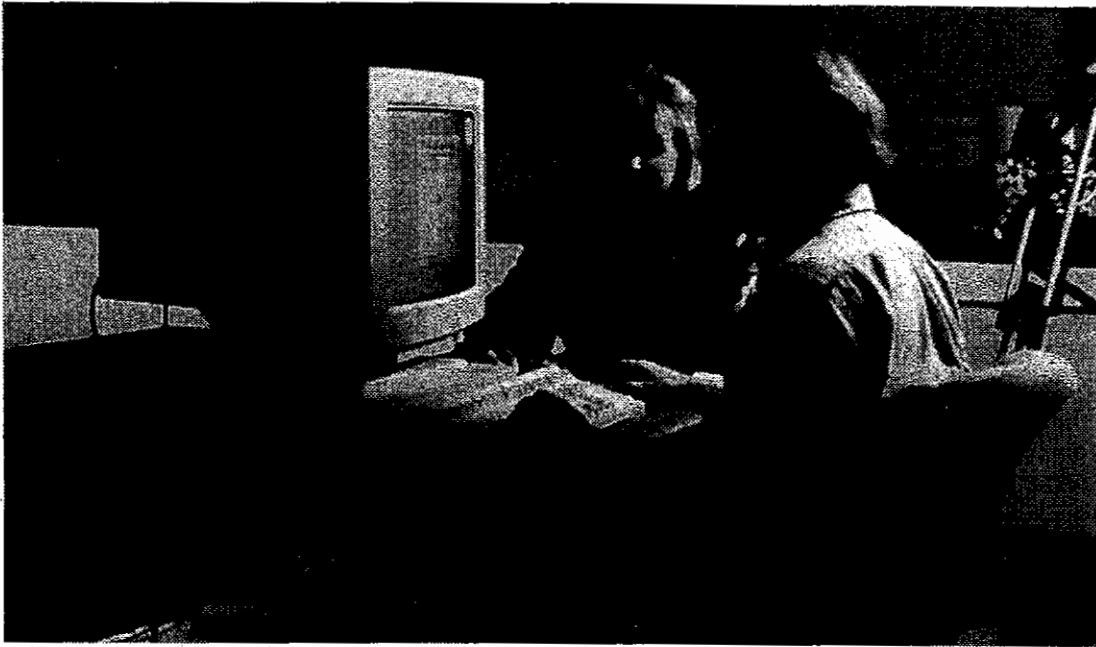


Figure 4.1

Usability lab test, with subject and observer seated at a workstation. Video recorders capture the user's actions and the contents of the screens, while microphones capture thinking-aloud comments. (Used with permission of Sun Microsystems, Mountain View, CA.)

tion and debriefing questions. The number, types, and source of participants are identified—sources, for example, might be customer sites, temporary personnel agencies, or advertisements placed in newspapers. A pilot test of the procedures, tasks, and questionnaires, with one to three subjects is conducted one week ahead of time, while there is still time for changes. This stereotypic preparation process can be modified in many ways to suit each project's unique needs.

After changes are approved, participants are chosen to represent the intended user communities, with attention to background in computing, experience with the task, motivation, education, and ability with the natural language used in the interface. Usability-laboratory staff also must control for physical concerns (such as eyesight, left- versus right-handedness, age, and gender), and for other experimental conditions (such as time of day, day of week, physical surroundings, noise, room temperature, and level of distractions).

Participants should always be treated with respect and should be informed that it is not *they* who are being tested; rather, it is the software and user interface that are under study. They should be told about what they will be doing (for example, typing text into a computer, creating a drawing using a mouse, or getting information from a touchscreen kiosk) and how long they will be expected to stay. Participation should always be voluntary, and

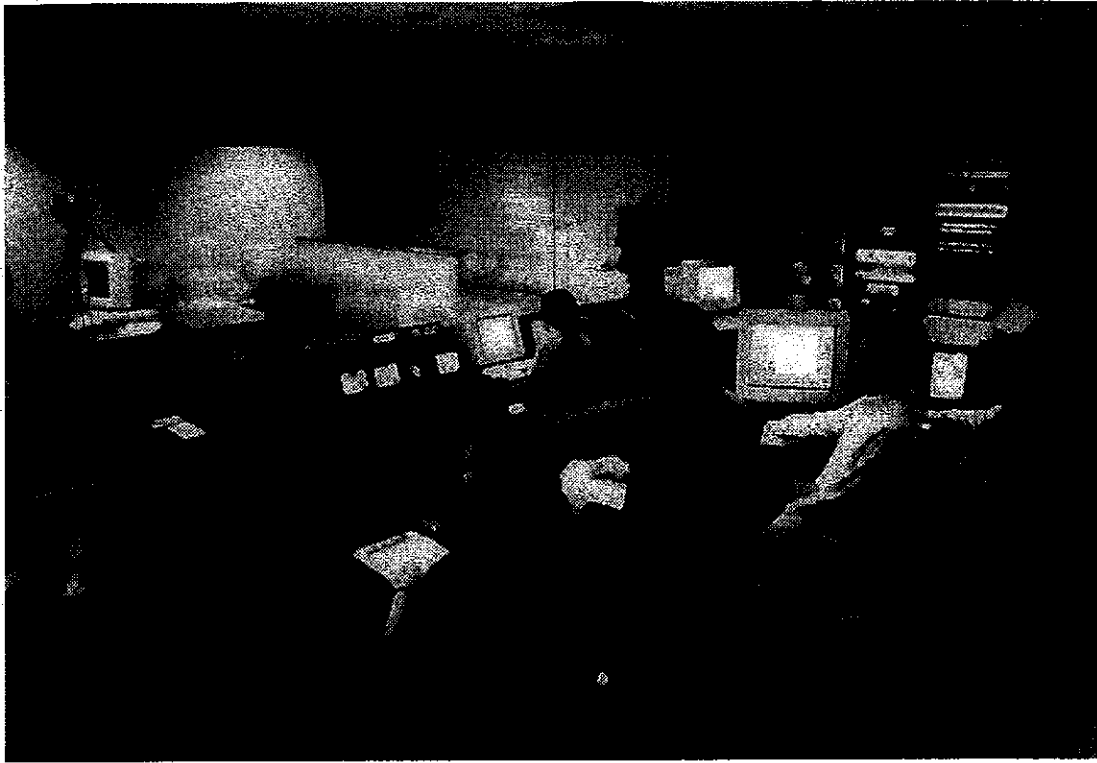


Figure 4.2

Usability lab control room, with test controllers and observers watching the subject through a half-silvered window. Video controls allow zooming and panning to focus on user actions. (Used with permission of Sun Microsystems, Mountain View, CA.)

informed consent should be obtained. Professional practice is to ask all subjects to read and sign a statement like this one:

- I have freely volunteered to participate in this experiment.
- I have been informed in advance what my task(s) will be and what procedures will be followed.
- I have been given the opportunity to ask questions and have had my questions answered to my satisfaction.
- I am aware that I have the right to withdraw consent and to discontinue participation at any time, without prejudice to my future treatment.
- My signature below may be taken as affirmation of all the above statements; it was given prior to my participation in this study.

An effective technique during usability testing is to invite users to think *aloud* about what they are doing. The designer or tester should be supportive of the participants, not taking over or giving instructions, but prompting and listening for clues about how they are dealing with the interface. After a suitable

time period for accomplishing the task list—usually one to three hours—the participants can be invited to make general comments or suggestions, or to respond to specific questions. The informal atmosphere of a thinking-aloud session is pleasant, and often leads to many spontaneous suggestions for improvements. In their efforts to encourage thinking aloud, some usability laboratories found that having two participants working together produces more talking, as one participant explains procedures and decisions to the other.

Videotaping participants performing tasks is often valuable for later review and for showing designers or managers the problems that users encounter (Lund, 1985). Reviewing videotapes is a tedious job, so careful logging and annotation during the test is vital to reduce the time spent finding critical incidents (Harrison, 1991). Participants may be anxious about the video cameras at the start of the test, but within minutes they usually focus on the tasks and ignore the videotaping. The reactions of designers to seeing videotapes of users failing with their system is sometimes powerful and may be highly motivating. When designers see subjects repeatedly picking the wrong menu item, they realize that the label or placement needs to be changed. Most usability laboratories have acquired or developed software to facilitate logging of user activities (typing, mousing, reading screens, reading manuals, and so on) by observers with automatic time stamping.

At each design stage, the interface can be refined iteratively, and the improved version can be tested. It is important to fix quickly even small flaws, such as of spelling errors or inconsistent layout, since they influence user expectations.

Many variant forms of usability testing have been tried. Nielsen's (1993) *discount usability engineering*, which advocates quick and dirty approaches to task analysis, prototype development, and testing, has been widely influential because it lowered the barriers to newcomers.

Field tests attempt to put new interfaces to work in realistic environments for a fixed trial period. Field tests can be made more fruitful if logging software is used to capture error, command, and help frequencies, plus productivity measures. Portable usability laboratories with videotaping and logging facilities have been developed to support more thorough field testing. A different kind of field testing supplies users with test versions of new software. The largest field test of all time was probably the beta-testing of Microsoft's Windows 95, in which reportedly 400,000 users internationally received early versions and were asked to comment.

Early usability studies can be conducted using paper mockups of screen displays to assess user reactions to wording, layout, and sequencing. A test administrator plays the role of the computer by flipping the pages while asking a participant user to carry out typical tasks. This informal testing is inexpensive and rapid, and usually is productive.

Game designers pioneered the *can-you-break-this* approach to usability testing by providing energetic teenagers with the challenge of trying to beat

new games. This destructive testing approach, in which the users try to find fatal flaws in the system or otherwise to destroy it, has been used in other projects and should be considered seriously. Software purchasers have little patience with flawed products and the cost of sending out tens of thousands of replacement disks is one that few companies can bear.

Competitive usability testing can be used to compare a new interface to previous versions or to similar products from competitors. This approach is close to a controlled experimental study, and staff must be careful to construct parallel sets of tasks and to counterbalance the order of presentation of the interfaces. Within subjects, designs seem more powerful because participants can make comparisons between the competing interfaces, so fewer participants are needed, although they will each be needed for a longer time period.

For all its success, usability testing does have at least two serious limitations: It emphasizes first-time usage and has limited coverage of the interface features. Since usability tests are usually two to four hours, it is difficult to ascertain how performance will be after a week or a month of regular usage. Within the typical two to four hours of a usability test, the participants may get to use only a small fraction of the features, menus, dialog boxes, or help screens. These and other concerns have led design teams to supplement usability testing with the varied forms of expert reviews.

4.4 Surveys

Written user surveys are a familiar, inexpensive, and generally acceptable companion for usability tests and expert reviews. Managers and users grasp the notion of surveys, and the typically large numbers of respondents (hundreds to thousands of users) offer a sense of authority compared to the potentially biased and highly variable results from small numbers of usability-test participants or expert reviewers. The keys to successful surveys are clear goals in advance and then development of focused items that help to attain those goals. Experienced surveyors know that care is also needed during administration and data analysis (Oppenheim, 1992).

A survey form should be prepared, reviewed among colleagues, and tested with a small sample of users before a large-scale survey is conducted. Similarly, statistical analyses (beyond means and standard deviations) and presentations (histograms, scatterplots, and so on) should also be developed before the final survey is distributed. In short, directed activities are more successful than unplanned statistical-gathering expeditions (no wild goose chases, please). My experience is that directed activities also seem to provide the most fertile frameworks for unanticipated discoveries.

Survey goals can be tied to the components of the OAI model of interface design (see Section 2.3). Users could be asked for their subjective impressions about specific aspects of the interface, such as the representation of

- Task domain objects and actions
- Interface domain metaphors and action handles
- Syntax of inputs and design of displays

Other goals would be to ascertain the user's

- Background (age, gender, origins, education, income)
- Experience with computers (specific applications or software packages, length of time, depth of knowledge)
- Job responsibilities (decision-making influence, managerial roles, motivation)
- Personality style (introvert versus extravert, risk taking versus risk averse, early versus late adopter, systematic versus opportunistic)
- Reasons for not using an interface (inadequate services, too complex, too slow)
- Familiarity with features (printing, macros, shortcuts, tutorials)
- Feelings after using an interface (confused versus clear, frustrated versus in control, bored versus excited)

Online surveys avoid the cost and effort of printing, distributing, and collecting paper forms. Many people prefer to answer a brief survey displayed on a screen, instead of filling in and returning a printed form, although there is a potential bias in the self-selected sample. One survey of World Wide Web utilization generated more than 13,000 respondents. So that costs are kept low, surveys might be administered to only a fraction of the user community.

In one survey, users were asked to respond to eight statements according to the following commonly used scale:

1. Strongly agree
2. Agree
3. Neutral
4. Disagree
5. Strongly disagree

The items in the survey were these:

1. I find the system commands easy to use.
2. I feel competent with and knowledgeable about the system commands.
3. When writing a set of system commands for a new application, I am confident that they will be correct on the first run.

4. When I get an error message, I find that it is helpful in identifying the problem.
5. I think that there are too many options and special cases.
6. I believe that the commands could be substantially simplified.
7. I have trouble remembering the commands and options, and must consult the manual frequently.
8. When a problem arises, I ask for assistance from someone who really knows the system.

This list of questions can help designers to identify problems users are having, and to demonstrate improvement to the interface as changes are made in training, online assistance, command structures, and so on; progress is demonstrated by improved scores on subsequent surveys.

In a study of error messages in text-editor usage, users had to rate the messages on 1-to-7 scales:

Hostile	1 2 3 4 5 6 7	Friendly
Vague	1 2 3 4 5 6 7	Specific
Misleading	1 2 3 4 5 6 7	Beneficial
Discouraging	1 2 3 4 5 6 7	Encouraging

If precise—as opposed to general—questions are used in surveys, then there is a greater chance that the results will provide useful guidance for taking action.

Coleman and Williges (1985) developed a set of bipolar semantically anchored items (pleasing versus irritating, simple versus complicated, concise versus redundant) that asked users to describe their reactions to using a word processor. Another approach is to ask users to evaluate aspects of the interface design, such as the readability of characters, the meaningfulness of command names, or the helpfulness of error messages. If users rate as poor one aspect of the interactive system, the designers have a clear indication of what needs to be redone.

The *Questionnaire for User Interaction Satisfaction* (QUIS) was developed by Shneiderman and was refined by Norman and Chin (Chin et al., 1988) (<http://www.lap.umd.edu/QUISFolder/quisHome.html>). It was based on the early versions of the OAI model and therefore covered interface details, such as readability of characters and layout of displays; interface objects, such as meaningfulness of icons; interface actions, such as shortcuts for frequent users; and task issues, such as appropriate terminology or screen sequencing. It has proved useful in demonstrating the benefits of improvements to a videodisc-retrieval program, in comparing two Pascal programming environments, in assessing word processors, and in setting requirements for redesign of an online public-access library catalog. We have

since applied QUIS in many projects with thousands of users and have created new versions that include items relating to website design and video-conferencing. The University of Maryland Office of Technology Liaison (College Park, Maryland 20742; (301) 405-4209) licenses QUIS in electronic and paper forms to over a hundred organizations internationally, in addition to granting free licenses to student researchers. The licensees have applied QUIS in varied ways, sometimes using only parts of QUIS or adding domain-specific items.

Table 4.1 contains the long form that was designed to have two levels of questions: general and detailed. If participants are willing to respond to every item, then the long-form questionnaire can be used. If participants are not likely to be patient, then only the general questions in the short form need to be asked.

Other scales include the Post-Study System Usability Questionnaire, developed by IBM, which has 48 items that focus on system usefulness, information quality, and interface quality (Lewis, 1995). The Software Usability Measurement Inventory contains 50 items designed to measure users' perceptions of their effect, efficiency, and control (Kirakowski and Corbett, 1993).

4.5 Acceptance Tests

For large implementation projects, the customer or manager usually sets objective and measurable goals for hardware and software performance. Many authors of requirements documents are even so bold as to specify mean time between failures, as well as the mean time to repair for hardware and, in some cases, for software. More typically, a set of test cases is specified for the software, with possible response-time requirements for the hardware-software combination. If the completed product fails to meet these acceptance criteria, the system must be reworked until success is demonstrated.

These notions can be neatly extended to the human interface. Explicit acceptance criteria should be established when the requirements document is written or when a contract is offered.

Rather than the vague and misleading criterion of "user friendly," measurable criteria for the user interface can be established for the following:

- Time for users to learn specific functions
- Speed of task performance
- Rate of errors by users
- User retention of commands over time
- Subjective user satisfaction

Table 4.1

Questionnaire for User Interaction Satisfaction (© University of Maryland, 1997)

Identification number: _____ System: _____ Age: _____ Gender: __ male __ female

PART 1: System Experience

1.1 How long have you worked on this system?

- | | |
|--|---|
| ☐ less than 1 hour | ☐ 6 months to less than 1 year |
| ☐ 1 hour to less than 1 day | ☐ 1 year to less than 2 years |
| ☐ 1 day to less than 1 week | ☐ 2 years to less than 3 years |
| ☐ 1 week to less than 1 month | ☐ 3 years or more |
| ☐ 1 month to less than 6 months | |

1.2 On the average, how much time do you spend per week on this system?

- | | |
|---|--|
| ☐ less than one hour | ☐ 4 to less than 10 hours |
| ☐ one to less than 4 hours | ☐ over 10 hours |

PART 2: Past Experience

2.1 How many operating systems have you worked with?

- | | |
|-------------------------------|--------------------------------------|
| ☐ none | ☐ 3-4 |
| ☐ 1 | ☐ 5-6 |
| ☐ 2 | ☐ more than 6 |

2.2 Of the following devices, software, and systems, check those that you have personally used and are familiar with:

- | | | |
|--|--|--|
| ☐ computer terminal | ☐ personal computer | ☐ lap top computer |
| ☐ color monitor | ☐ touch screen | ☐ floppy drive |
| ☐ CD-ROM drive | ☐ keyboard | ☐ mouse |
| ☐ track ball | ☐ joy stick | ☐ pen based computing |
| ☐ graphics tablet | ☐ head mounted display | ☐ modems |
| ☐ scanners | ☐ word processor | ☐ graphics software |
| ☐ spreadsheet software | ☐ database software | ☐ computer games |
| ☐ voice recognition | ☐ video editing systems | ☐ internet |
| ☐ CAD computer aided design | ☐ rapid prototyping systems | ☐ e-mail |

PART 3: Overall User Reactions

Please circle the numbers which most appropriately reflect your impressions about using this computer system. Not Applicable = NA.

- | | | | | |
|--------------------------------------|-------------------|--|----------------|----|
| 3.1 Overall reactions to the system: | terrible | | wonderful | |
| | 1 2 3 4 5 6 7 8 9 | | | NA |
| 3.2 | frustrating | | satisfying | |
| | 1 2 3 4 5 6 7 8 9 | | | NA |
| 3.3 | dull | | stimulating | |
| | 1 2 3 4 5 6 7 8 9 | | | NA |
| 3.4 | difficult | | easy | |
| | 1 2 3 4 5 6 7 8 9 | | | NA |
| 3.5 | inadequate power | | adequate power | |
| | 1 2 3 4 5 6 7 8 9 | | | NA |
| 3.6 | rigid | | flexible | |
| | 1 2 3 4 5 6 7 8 9 | | | NA |

Table 4.1 (continued)

PART 4: Screen

4.1 Characters on the computer screen	hard to read								easy to read	
	1	2	3	4	5	6	7	8	9	NA
4.1.1 Image of characters	fuzzy								sharp	
	1	2	3	4	5	6	7	8	9	NA
4.1.2 Character shapes (fonts)	barely legible								very legible	
	1	2	3	4	5	6	7	8	9	NA
4.2 Highlighting on the screen	unhelpful								helpful	
	1	2	3	4	5	6	7	8	9	NA
4.2.1 Use of reverse video	unhelpful								helpful	
	1	2	3	4	5	6	7	8	9	NA
4.2.2 Use of blinking	unhelpful								helpful	
	1	2	3	4	5	6	7	8	9	NA
4.2.3 Use of bolding	unhelpful								helpful	
	1	2	3	4	5	6	7	8	9	NA
4.3 Screen layouts were helpful	never								always	
	1	2	3	4	5	6	7	8	9	NA
4.3.1 Amount of information that can be displayed on screen	inadequate								adequate	
	1	2	3	4	5	6	7	8	9	NA
4.3.2 Arrangement of information can be displayed on screen	illogical								logical	
	1	2	3	4	5	6	7	8	9	NA
4.4 Sequence of screens	confusing								clear	
	1	2	3	4	5	6	7	8	9	NA
4.4.1 Next screen in a sequence	unpredictable								predictable	
	1	2	3	4	5	6	7	8	9	NA
4.4.2 Going back to the previous screen	impossible								easy	
	1	2	3	4	5	6	7	8	9	NA
4.4.3 Progression of work related tasks	confusing								clearly marked	
	1	2	3	4	5	6	7	8	9	NA

Please write your comments about the screens here:

PART 5: Terminology and System Information

5.1 Use of terminology throughout system	inconsistent								consistent	
	1	2	3	4	5	6	7	8	9	NA
5.1.2 Work related terminology	inconsistent								consistent	
	1	2	3	4	5	6	7	8	9	NA
5.2.3 Computer terminology	inconsistent								consistent	
	1	2	3	4	5	6	7	8	9	NA

Table 4.1 (continued)

5.2 Terminology relates well to the work you are doing?	never								always	
	1	2	3	4	5	6	7	8	9	NA
5.2.1 Computer terminology is used	too frequently								appropriately	
	1	2	3	4	5	6	7	8	9	NA
5.2.2 Terminology on the screen	ambiguous								precise	
	1	2	3	4	5	6	7	8	9	NA
5.3 Messages which appear on screen	inconsistent								consistent	
	1	2	3	4	5	6	7	8	9	NA
5.3.1 Position of instructions on the screen	inconsistent								consistent	
	1	2	3	4	5	6	7	8	9	NA
5.4 Messages which appear on screen	confusing								clear	
	1	2	3	4	5	6	7	8	9	NA
5.4.1 Instructions for commands or functions	confusing								clear	
	1	2	3	4	5	6	7	8	9	NA
5.4.2 Instructions for correcting errors	confusing								clear	
	1	2	3	4	5	6	7	8	9	NA
5.5 Computer keeps you informed about what it is doing	never								always	
	1	2	3	4	5	6	7	8	9	NA
5.5.1 Animated cursors keep you informed	never								always	
	1	2	3	4	5	6	7	8	9	NA
5.5.2 Performing an operation leads to a predictable result	never								always	
	1	2	3	4	5	6	7	8	9	NA
5.5.3 Controlling amount of feedback	impossible								easy	
	1	2	3	4	5	6	7	8	9	NA
5.5.4 Length of delay between operations	unacceptable								acceptable	
	1	2	3	4	5	6	7	8	9	NA
5.6 Error messages	unhelpful								helpful	
	1	2	3	4	5	6	7	8	9	NA
5.6.1 Error messages clarify the problem	never								always	
	1	2	3	4	5	6	7	8	9	NA
5.6.2 Phrasing of error messages	unpleasant								pleasant	
	1	2	3	4	5	6	7	8	9	NA

Please write your comments about terminology and system information here:

PART 6: Learning

6.1 Learning to operate the system	difficult								easy	
	1	2	3	4	5	6	7	8	9	NA
6.1.1 Getting started	difficult								easy	
	1	2	3	4	5	6	7	8	9	NA

Table 4.1 (continued)

6.1.2 Learning advanced features	difficult								easy	
	1	2	3	4	5	6	7	8	9	NA
6.1.3 Time to learn to use the system	difficult								easy	
	1	2	3	4	5	6	7	8	9	NA
6.2 Exploration of features by trial and error	discouraging								encouraging	
	1	2	3	4	5	6	7	8	9	NA
6.2.1 Exploration of features	risky								safe	
	1	2	3	4	5	6	7	8	9	NA
6.2.2 Discovering new features	difficult								easy	
	1	2	3	4	5	6	7	8	9	NA
6.3 Remembering names and use of commands	difficult								easy	
	1	2	3	4	5	6	7	8	9	NA
6.3.1 Remembering specific rules about entering commands	difficult								easy	
	1	2	3	4	5	6	7	8	9	NA
6.4 Tasks can be performed in a straightforward manner	never								always	
	1	2	3	4	5	6	7	8	9	NA
6.4.1 Number of steps per task	too many								just right	
	1	2	3	4	5	6	7	8	9	NA
6.4.2 Steps to complete a task follow a logical sequence	never								always	
	1	2	3	4	5	6	7	8	9	NA
6.4.3 Feedback on the completion of sequence of steps	unclear								clear	
	1	2	3	4	5	6	7	8	9	NA

Please write your comments about learning here:

PART 7: System Capabilities

7.1 System speed	too slow								fast enough	
	1	2	3	4	5	6	7	8	9	NA
7.1.1 Response time for most operations	too slow								fast enough	
	1	2	3	4	5	6	7	8	9	NA
7.1.2 Rate information is displayed	too slow								fast enough	
	1	2	3	4	5	6	7	8	9	NA
7.2 The system is reliable	never								always	
	1	2	3	4	5	6	7	8	9	NA
7.2.1 Operations	undependable								dependable	
	1	2	3	4	5	6	7	8	9	NA
7.2.2 System failures occur	frequently								seldom	
	1	2	3	4	5	6	7	8	9	NA
7.2.3 System warns you about potential problems	never								always	
	1	2	3	4	5	6	7	8	9	NA

Table 4.1 (continued)

8.3.5 Help defines specific aspects of the system	inadequately								adequately	
	1	2	3	4	5	6	7	8	9	NA
8.3.6 Finding specific information using the on-line help	difficult								easy	
	1	2	3	4	5	6	7	8	9	NA
8.3.7 On-line help	useless								helpful	
	1	2	3	4	5	6	7	8	9	NA

Please write your comments about technical manuals and on-line help here:

PART 9: On-line Tutorials

9.1 Tutorial was	useless								helpful	
	1	2	3	4	5	6	7	8	9	NA
9.1.1 Accessing on-line tutorial	difficult								easy	
	1	2	3	4	5	6	7	8	9	NA
9.2 Maneuvering through the tutorial was	difficult								easy	
	1	2	3	4	5	6	7	8	9	NA
9.2.1 Tutorial is meaningfully structured	never								always	
	1	2	3	4	5	6	7	8	9	NA
9.2.2 The speed of presentation was	unacceptable								acceptable	
	1	2	3	4	5	6	7	8	9	NA
9.3 Tutorial content was	useless								helpful	
	1	2	3	4	5	6	7	8	9	NA
9.3.1 Information for specific aspects of the system were complete and informative	never								always	
	1	2	3	4	5	6	7	8	9	NA
9.3.2 Information was concise and to the point	never								always	
	1	2	3	4	5	6	7	8	9	NA
9.4 Tasks can be completed	with difficulty								easily	
	1	2	3	4	5	6	7	8	9	NA
9.4.1 Instructions given for completing tasks	confusing								clear	
	1	2	3	4	5	6	7	8	9	NA
9.4.2 Time given to perform tasks	inadequate								adequate	
	1	2	3	4	5	6	7	8	9	NA
9.5 Learning to operate the system using the tutorial was	difficult								easy	
	1	2	3	4	5	6	7	8	9	NA
9.5.1 Completing system tasks after using only the tutorial	difficult								easy	
	1	2	3	4	5	6	7	8	9	NA

Please write your comments about on-line tutorials here:

Table 4.1 (continued)

PART 10: Multimedia

10.1 Quality of still pictures/photographs	bad								good	
	1	2	3	4	5	6	7	8	9	NA
10.1.1 Pictures/Photos	fuzzy								clear	
	1	2	3	4	5	6	7	8	9	NA
10.1.2 Picture/Photo brightness	dim								bright	
	1	2	3	4	5	6	7	8	9	NA
10.2 Quality of movies	bad								good	
	1	2	3	4	5	6	7	8	9	NA
10.2.1 Focus of movie images	fuzzy								clear	
	1	2	3	4	5	6	7	8	9	NA
10.2.2 Brightness of movie images	dim								bright	
	1	2	3	4	5	6	7	8	9	NA
10.2.3 Movie window size is adequate	never								always	
	1	2	3	4	5	6	7	8	9	NA
10.3 Sound output	inaudible								audible	
	1	2	3	4	5	6	7	8	9	NA
10.3.1 Sound output	choppy								smooth	
	1	2	3	4	5	6	7	8	9	NA
10.3.2 Sound output	garbled								clear	
	1	2	3	4	5	6	7	8	9	NA
10.4 Colors used are	unnatural								natural	
	1	2	3	4	5	6	7	8	9	NA
10.4.1 Amount of colors available	inadequate								adequate	
	1	2	3	4	5	6	7	8	9	NA

Please write your comments about multimedia here:

PART 11: Teleconferencing

11.1 Setting up for conference	difficult								easy	
	1	2	3	4	5	6	7	8	9	NA
11.1.1 Time for establishing the connections to others	too long								just right	
	1	2	3	4	5	6	7	8	9	NA
11.1.2 Number of connections possible	too few								enough	
	1	2	3	4	5	6	7	8	9	NA
11.2 Arrangement of windows showing connecting groups	confusing								clear	
	1	2	3	4	5	6	7	8	9	NA
11.2.1 Window with view of your own group is of appropriate size	never								always	
	1	2	3	4	5	6	7	8	9	NA

Table 4.1 (continued)

11.2.2 Window(s) with view of connecting group(s) is of appropriate size	never	1	2	3	4	5	6	7	8	9	always	NA
11.3 Determining the focus of attention during conference was	confusing	1	2	3	4	5	6	7	8	9	clear	NA
11.3.1 Telling who is speaking	difficult	1	2	3	4	5	6	7	8	9	easy	NA
11.4 Video image flow	choppy	1	2	3	4	5	6	7	8	9	smooth	NA
11.4.1 Focus of video image	fuzzy	1	2	3	4	5	6	7	8	9	clear	NA
11.5 Audio output	inaudible	1	2	3	4	5	6	7	8	9	audible	NA
11.5.1 Audio is in sync with video images	never	1	2	3	4	5	6	7	8	9	always	NA
11.6 Exchanging data	difficult	1	2	3	4	5	6	7	8	9	easy	NA
11.6.1 Transmitting files	difficult	1	2	3	4	5	6	7	8	9	easy	NA
11.6.2 Retrieving files	difficult	1	2	3	4	5	6	7	8	9	easy	NA
11.6.3 Using on-line chat	difficult	1	2	3	4	5	6	7	8	9	easy	NA
11.6.4 Using shared workspace	difficult	1	2	3	4	5	6	7	8	9	easy	NA

Please write your comments about teleconferencing here:

PART 12: Software Installation

12.1 Speed of installation	slow	1	2	3	4	5	6	7	8	9	fast	NA
12.2 Customization	difficult	1	2	3	4	5	6	7	8	9	easy	NA
12.2.1 Installing only the software you want	confusing	1	2	3	4	5	6	7	8	9	clear	NA
12.3 Informs you of its progress	never	1	2	3	4	5	6	7	8	9	always	NA
12.4 Gives a meaningful explanation when failures occur	never	1	2	3	4	5	6	7	8	9	always	NA

Please write your comments about software installation here:

An acceptance test might specify the following:

The subjects will be 35 secretaries hired from an employment agency. They have no word-processing experience, but have typing skills in the range of 35 to 50 words per minute. They will be given 45 minutes of training on the basic features. At least 30 of the 35 secretaries should be able to complete, within 30 minutes, 80 percent of the typing and editing tasks in the enclosed benchmark test correctly.

Another testable requirement for the same system might be this:

After four half-days of regular use of the system, 25 of these 35 secretaries should be able to carry out, within 20 minutes, the advanced editing tasks in the second benchmark test, and should make fewer than six errors.

This second acceptance test captures performance after regular use. The choice of the benchmark tests is critical and is highly system dependent. The test materials and procedures must also be refined by pilot testing before use.

A third item in the acceptance test plan might focus on retention:

After two weeks, at least 15 of the test subjects should be recalled and should perform the third benchmark test. In 40 minutes, at least 10 of the subjects should be able to complete 75 percent of the tasks correctly.

In a large system, there may be eight or 10 such tests to carry out on different components of the interface and with different user communities. Other criteria such as subjective satisfaction, output comprehensibility, system response time, installation procedures, printed documentation, or graphics appeal may also be considered in acceptance tests of complete commercial products.

If they establish precise acceptance criteria, both the customer and the interface developer can benefit. Arguments about the user friendliness are avoided, and contractual fulfillment can be demonstrated objectively. Acceptance tests differ from usability tests in that the atmosphere may be adversarial, so outside testing organizations are often appropriate to ensure neutrality. The central goal of acceptance testing is not to detect flaws, but rather to verify adherence to requirements.

Once acceptance testing has been successful, there may be a period of field testing before national or international distribution. In addition to further refining the user interface, field tests can improve training methods, tutorial materials, telephone-help procedures, marketing methods, and publicity strategies.

The goal of early expert reviews, usability testing, surveys, acceptance testing, and field testing is to force as much as possible of the evolutionary

development into the prerelease phase, when change is relatively easy and inexpensive to accomplish.

4.6 Evaluation During Active Use

A carefully designed and thoroughly tested system is a wonderful asset, but successful active use requires constant attention from dedicated managers, user-services personnel, and maintenance staff. Everyone involved in supporting the user community can contribute to system refinements that provide ever higher levels of service. You cannot please all of the users all of the time, but earnest effort will be rewarded by the appreciation of a grateful user community. Perfection is not attainable, but percentage improvements are possible and are worth pursuing.

Gradual system dissemination is useful so that problems can be repaired with minimal disruption. As more and more people use the system, major changes should be limited to an annual or semiannual system revision that is announced adequately. If system users can anticipate the change, then resistance will be reduced, especially if they have positive expectations of improvement. More frequent changes are expected in the rapidly developing World Wide Web environment, but a balance between stable access to key resources even as novel services are added may be the winning policy.

4.6.1 Interviews and focus-group discussions

Interviews with individual users can be productive because the interviewer can pursue specific issues of concern. After a series of individual discussions, *focus-group discussions* are valuable to ascertain the universality of comments. Interviewing can be costly and time consuming, so usually only a small fraction of the user community is involved. On the other hand, direct contact with users often leads to specific, constructive suggestions.

A large corporation conducted 45-minute interviews with 66 of the 4300 users of an internal message system. The interviews revealed that the users were happy with some aspects of the functionality, such as the capacity to pick up messages at any site, the legibility of printed messages, and the convenience of after-hours access. However, the interviews also revealed that 23.6 percent of the users had concerns about reliability, 20.2 percent thought that using the system was confusing, and 18.2 percent said convenience and accessibility could be improved, whereas only 16.0 percent expressed no

concerns. Later questions in the interview explored specific features. As a result of this interview project, a set of 42 enhancements to the system was proposed and implemented. The designers of the system had earlier proposed an alternate set of enhancements, but the results of the interviews led to a changed set of priorities that more closely reflected the users' needs.

4.6.2 Continuous user-performance data logging

The software architecture should make it easy for system managers to collect data about the patterns of system usage, speed of user performance, rate of errors, or frequency of requests for online assistance. Logging data provide guidance in the acquisition of new hardware, changes in operating procedures, improvements to training, plans for system expansion, and so on.

For example, if the frequency of each error message is recorded, then the highest-frequency error is a candidate for attention. The message could be rewritten, training materials could be revised, the software could be changed to provide more specific information, or the command syntax could be simplified. Without specific logging data, the system-maintenance staff has no way of knowing which of the many hundreds of error-message situations is the biggest problem for users. Similarly, staff should examine messages that never appear, to see whether there is an error in the code or whether users are avoiding use of some facility.

If logging data are available for each command, each help screen, and each database record, then changes to the human-computer interface can be made to simplify access to frequently used features. Managers also should examine unused or rarely used facilities to understand why users are avoiding those features. Logging of the Thomas system for access to U.S. Congress legislation revealed high-frequency terms, such as *abortion*, *gun control*, and *balanced budget* that could be used in a browse list of hot topics (Croft et al, 1995). Logging in an educational database identified frequently used as well and rarely used paths and features (Marchionini and Crane, 1994).

A major benefit of usage-frequency data is the guidance that they provide to system maintainers in optimizing performance and in reducing costs for all participants. This latter argument may yield the clearest advantage to cost-conscious managers, whereas the increased quality of the interface is an attraction to service-oriented managers.

Logging may be well intentioned, but users' rights to privacy deserve to be protected. Links to specific user names should not be collected, unless necessary. When logging aggregate performance crosses over to monitoring individual activity, managers must inform users of what is being monitored and how the information will be used. Although organizations may have a right to ascertain worker performance, workers should be able to view the results and to discuss the implications. If monitoring is surreptitious and is later discovered, resulting worker mistrust of management could be more

damaging than the benefits of the collected data. Manager and worker cooperation to improve productivity, and worker participation in the process and benefits, are advised.

4.6.3 Online or telephone consultants

Online or telephone consultants can provide extremely effective and personal assistance to users who are experiencing difficulties. Many users feel reassured if they know that there is a human being to whom they can turn when problems arise. These consultants are an excellent source of information about problems users are having and can suggest improvements and potential extensions.

Many organizations offer a toll-free number via which the users can reach a knowledgeable consultant; others charge for consultation by the minute. On some network systems, the consultants can monitor the user's computer and see the same displays that the user sees while maintaining telephone voice contact. This service can be extremely reassuring: Users know that someone can walk them through the correct sequence of screens to complete their tasks.

America Online provides live (real-time) chat rooms for discussion of user problems. Users can type their questions and get responses promptly. Many groups maintain a standard electronic-mail address of `staff@<organization>` that allows users to get help from whomever is on duty. My several successful experiences of getting quick help late at night from our departmental staff have remained firmly in my memory. On one occasion, they helped me to unpack a file in an unfamiliar format; on another, they recovered an inadvertently deleted file.

4.6.4 Online suggestion box or trouble reporting

Electronic mail can be employed to allow users to send messages to the maintainers or designers. Such an *online suggestion box* encourages some users to make productive comments, since writing a letter may be seen as requiring too much effort.

A Library of Congress website that invites comments gets 10 to 20 per day, including thoughtful ones such as this:

I find as I get searching through the various Web pages . . . that I am left with an unsatisfied feeling. I have been sitting in front of the PC for close to an hour . . . and have been stopped and/or slowed due to items that can be directly related to web server design.

First off, the entry pages are too big and disorganized. Those links that do exist do not have adequate enough descriptions to direct a user to the information they desire. In addition, the use of a search engine would greatly facilitate sifting through the abundance of information that is thrown at the user with any one of these links. Links should be short, sweet, and specific. Large amounts of material should not be included in one document on a busy server. . . .

Breaking up these larger documents into smaller, well organized documents may seem to create an additional burden on programming. However, if intelligence is used in the creation of such systems, it would not take much . . .

In fact, the search engine that this user wanted was available, but he could not find it, and larger documents were broken into smaller segments. A reply helped to get this user what he was seeking, and his message also led to design changes that made the interface features more visible.

An internet directory service for personal names, Knowbot Information Service, offers a gripe command with the invitation "Place a compliment or complaint in the KIS log file." Another service simply has a button labeled "Tell us what you think."

A large corporation installed a full-screen, fill-in-the-blanks form for user problem reports, and received 90 comments on a new internal system within three months. The user's identification number and name were entered automatically, and the user moved a cursor to indicate which subsystem was causing a problem and what the problem's seriousness was (showstopper, annoyance, improvement, other). Each problem report received a dated and signed response that was stored on a file for public reading.

4.6.5 Online bulletin board or newsgroup

Some users may have a question about the suitability of a software package for their application, or may be seeking someone who has had experience using an interface feature. They do not have any individual in mind, so electronic mail does not serve their needs. Many interface designers offer users an *electronic bulletin board* or *newsgroup* (see Section 14.3) to permit posting of open messages and questions. These newsgroups cover programming languages, software tools, or task domains. There are also mailing lists for interface designers, such as the one established on the internet by the Human Factors and Ergonomics Society's Computer Systems Technical Group (send electronic mail to list-serv@listserv.vt.edu with this line: subscribe cstg-L <your full name>)

Some professional societies offer bulletin boards by way of networks such as America Online, Prodigy, and CompuServe. These bulletin boards may offer information services or permit downloading of software.

Bulletin-board software systems usually offer a list of item headlines, allowing users the opportunity to select items for display. New items can be added by anyone, but usually someone monitors the bulletin board to ensure that offensive, useless, or repetitious items are removed.

4.6.6 User newsletters and conferences

When there is a substantial number of users who are geographically dispersed, managers may have to work harder to create a sense of community. *Newsletters* that provide information about novel interface facilities, sugges-

tions for improved productivity, requests for assistance, case studies of successful applications, or stories about individual users can promote user satisfaction and knowledge. Printed newsletters are more traditional and have the advantage that they can be carried away from the workstation. A printed newsletter has an appealing air of respectability. Online newsletters are less expensive and more rapidly disseminated. World Wide Web or CD-ROM newsletters are appealing if collections of images are included or large datasets are anticipated.

Personal relationships established by face-to-face meetings also increase the sense of community among users. *Conferences* allow workers to exchange experiences with colleagues, promote novel approaches, stimulate greater dedication, encourage higher productivity, and develop a deeper relationship of trust. Ultimately, it is the people who matter in an organization, and human needs for social interaction should be satisfied. Every technical system is also a social system that needs to be encouraged and nurtured.

By soliciting user feedback in any of these ways, managers can gauge user attitudes and elicit useful suggestions. Furthermore, users may have more positive attitudes toward the interface if they see that the managers genuinely desire comments and suggestions.

4.7 Controlled Psychologically Oriented Experiments

Scientific and engineering progress is often stimulated by improved techniques for precise measurement. Rapid progress in the designs of interfaces will be stimulated as researchers and practitioners evolve suitable human-performance measures and techniques. We have come to expect that automobiles will have miles-per-gallon reports pasted to the window, appliances will have energy-efficiency ratings, and textbooks will be given grade-level designations; soon, we will expect software packages to show learning-time estimates and user-satisfaction indices from appropriate evaluation sources.

Academic and industrial researchers are discovering that the power of the traditional scientific method can be fruitfully employed in the study of interfaces (Barnard, 1991). They are conducting numerous experiments that are uncovering basic design principles. The outline of the scientific method as applied to human-computer interaction might include these tasks:

- Deal with a practical problem and consider the theoretical framework.
- State a lucid and testable hypothesis.
- Identify a small number of independent variables that are to be manipulated.
- Carefully choose the dependent variables that will be measured.

- Judiciously select subjects, and carefully or randomly assign subjects to groups.
- Control for biasing factors (nonrepresentative sample of subjects or selection of tasks, inconsistent testing procedures).
- Apply statistical methods to data analysis.
- Resolve the practical problem, refine the theory, and give advice to future researchers.

The classic experimental methods of psychology are being enhanced to deal with the complex cognitive tasks of human performance with information and computer systems. The transformation from Aristotelian introspection to Galilean experimentation that took two millennia in physics is being accomplished in two decades in the study of human-computer interaction.

The reductionist approach required for controlled experimentation yields narrow but reliable results. Through multiple replications with similar tasks, subjects, and experimental conditions, reliability and validity can be enhanced. Each small experimental result acts like a tile in the mosaic of human performance with computer-based information systems.

Managers of actively used systems are also coming to recognize the power of controlled experiments in fine tuning the human-computer interface. As proposals are made for new menu structures, novel cursor-control devices, and reorganized display formats, a carefully controlled experiment can provide data to support a management decision. Fractions of the user population could be given proposed improvements for a limited time, and then performance could be compared with the control group. Dependent measures could include performance times, user-subjective satisfaction, error rates, and user retention over time.

Experimental design and statistical analysis are complex topics (Hays, 1988; Cozby, 1996; Runyon and Haber, 1996; Winer et al., 1991.) Novice experimenters would be well advised to collaborate with experienced social scientists and statisticians.

4.8 Practitioner's Summary

Interface developers evaluate their designs by conducting expert reviews, usability tests, surveys, and rigorous acceptance tests. Once systems are released, developers perform continuous performance evaluations by interviews or surveys, and by logging user performance in a way that respects the privacy of users. If you are not measuring, you are not doing human factors!

Successful system managers understand that they must work hard to establish a relationship of trust with the user community. In addition to pro-

viding a properly functioning system, computer service managers and information-systems directors recognize the need to create social mechanisms for feedback, such as online surveys, interviews, discussions, consultants, suggestion boxes, bulletin boards, newsletters, and conferences.

4.9 Researcher's Agenda

Researchers can contribute their experience with experimentation to developing techniques for system evaluation. Guidance in conducting pilot studies, acceptance tests, surveys, interviews, and discussions would benefit commercial development groups. Experts in constructing psychological tests would be extremely helpful in preparing a validated and reliable test instrument for subjective evaluation of interactive systems. Such a standardized test would allow independent groups to compare the acceptability of their systems. In addition, assessment methods for user skill levels with software would be helpful in job-placement and training programs.

Clinical psychologists, psychotherapists, and social workers could contribute to training online or as telephone consultants—after all, helping troubled users is a human-relationship issue. Finally, more input from experimental, cognitive, and clinical psychologists would help computer specialists to recognize the importance of the human aspects of computer use. What techniques can reduce novice user anxiety? How can life-critical applications for experienced professionals be tested reliably?

World Wide Web Resources

WWW

Prototyping and usability testing methods are covered with some information on evaluation methods, such as surveys. The full text of our QUIS is available online.

<http://www.aw.com/DTUI>

References

- Barnard, Phil, The contributions of applied cognitive psychology to the study of human-computer interaction. In Shackel, B. and Richardson, S. (Editors), *Human Factors for Informatics Usability*, Cambridge University Press, Cambridge, U.K. (1991), 151–182.
- Chin, John P., Diehl, Virginia A., and Norman, Kent L., Development of an instrument measuring user satisfaction of the human-computer interface, *Proc. CHI '88—Human Factors in Computing Systems*, ACM, New York (1988), 213–218.

- Coleman, William D. and Williges, Robert C., Collecting detailed user evaluations of software interfaces, *Proc. Human Factors Society—Twenty-Ninth Annual Meeting*, Santa Monica, CA (1985), 204–244.
- Cozby, Paul C., *Methods in Behavioral Research* (Sixth Edition), Mayfield, Mountain View, CA (1996).
- Croft, W. Bruce, Cook, Robert, and Wilder, Dean, Providing government information on the internet: Experiences with THOMAS, *Proc. Digital Libraries '95 Conference*, ACM, New York (1995). Also available at <http://www.csd1.tamu.edu/DL95/papers/croft/croft.html>
- Curtis, Bill, Defining a place for interface engineering, *IEEE Software*, 9, 2 (March 1992), 84–86.
- Dumas, Joseph and Redish, Janice, *A Practical Guide to Usability Testing*, Ablex, Norwood, NJ (1993).
- Gould, John, How to design usable systems. In Helander, Martin (Editor), *Handbook of Human–Computer Interaction*, North-Holland, Amsterdam, The Netherlands (1988), 757–789.
- Gould, John D., Boies, Stephen J., and Lewis, Clayton, Making usable, useful productivity-enhancing computer applications, *Communications of the ACM*, 34, 1 (January 1991), 75–85.
- Harrison, Beverly L., Video annotation and multimedia interfaces: From theory to practice, *Proc. Human Factors Society Thirty-Fifth Annual Meeting* (1991), 319–322.
- Hays, William L., *Statistics* (Fourth Edition), Holt, Rinehart and Winston, New York (1988).
- Hix, Deborah and Hartson, H. Rex, *Developing User Interfaces: Ensuring Usability Through Product and Process*, John Wiley and Sons, New York (1993).
- Jeffries, R., Miller, J. R., Wharton, C., and Uyeda, K. M., User interface evaluation in the real world: A comparison of four techniques, *Proc. ACM CHI91 Conf.* (1991), 119–124.
- Karat, Claire-Marie, A business case approach to usability. In Bias, Randolph, and Mayhew, Deborah (Editors), *Cost-Justifying Usability*, Academic Press, New York (1994), 45–70.
- Karat, Claire-Marie, Campbell, Robert, and Fiegel, T., Comparison of empirical testing and walkthrough methods in user interface evaluation, *Proc. CHI '92—Human Factors in Computing Systems*, ACM, New York (1992), 397–404.
- Kirakowski, J. and Corbett, M. SUMI: The Software Usability Measurement Inventory, *British Journal of Educational Technology*, 24, 3 (1993), 210–212.
- Landauer, Thomas K., *The Trouble with Computers: Usefulness, Usability, and Productivity*, MIT Press, Cambridge, MA (1995).
- Lewis, James R., IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use, *International Journal of Human–Computer Interaction*, 7, 1 (1995), 57–78.
- Lund, Michelle A., Evaluating the user interfaces: The candid camera approach, *Proc. CHI '85—Human Factors in Computing Systems*, ACM, New York (1985), 93–97.

- Marchionini, Gary and Crane, Gregory, Evaluating hypermedia and learning: Methods and results from the Perseus Project, *ACM Transactions on Information Systems*, 12, 1 (1994), 5-34.
- Newman, William M. and Lamming, Michael G., *Interactive System Design*, Addison-Wesley, Reading, MA (1995).
- Nielsen, Jakob (Editor), Special Issue on Usability Laboratories, *Behaviour & Information Technology*, 13, 1 & 2 (January-April 1994).
- Nielsen, Jakob, *Usability Engineering*, Academic Press, New York (1993).
- Nielsen, Jakob and Mack, Robert (Editors), *Usability Inspection Methods*, John Wiley and Sons, New York (1994).
- Oppenheim, Abraham N., *Questionnaire Design, Interviewing, and Attitude Measurement*, Pinter Publishers, New York (1992).
- Preece, Jenny, Rogers, Yvonne, Sharp, Helen, Benyon, David, Holland, Simon, and Carey, Tom, *Human-Computer Interaction*, Addison-Wesley, Reading, MA (1994).
- Runyon, Richard P. and Haber, Audrey, *Fundamentals of Behavioral Statistics* (Eighth Edition), McGraw-Hill, New York (1996).
- Wharton, Cathleen, Rieman, John, Lewis, Clayton, and Polson, Peter, The cognitive walkthrough method: A practitioner's guide. In Nielsen, Jakob and Mack, Robert (Editors), *Usability Inspection Methods*, John Wiley and Sons, New York (1994).
- Winer, B. J., Brown, Donald R., and Michels, Kenneth M., *Statistical Principles in Experimental Design*, McGraw-Hill, New York (1991).
- Yourdon, Edward, *Structured Walkthroughs* (Fourth Edition), Yourdon Press, Englewood Cliffs, NJ (1989).