

Dagobert Soergel dsoergel@umd.edu www.dsoergel.com

Room 4101

Main Office mailbox 22

Off. 301-405-2037 / Home 703-823-2840 / Cell 703-585-2840

But mostly out of town, so use email

Description:

Identification, evaluation, and selection of software. Usability testing of software and of Web sites. Covers a wide spectrum of software used in a wide variety of contexts. Topics include identifying user needs; sources for software information; evaluation criteria, including human factors; evaluation methods; organizing the evaluation process; organizing a software collection and promoting its use.

Objectives

1. Students should be able to identify user needs and opportunities for software application in an organization or other context.
2. Students should be able to draw up a list of selection/evaluation criteria that reflect user needs as well as the present situation in the organization.
3. Students should be able to identify suitable software packages from a variety of sources
4. Students should be able to apply the criteria specified to the software packages found and arrive at a well-founded selection decision. In particular,
5. Students should be able to conduct simple usability studies for software and for Web sites.
6. Students should be able to develop a software collection policy and to manage a software collection and promote its use.
7. Students should be able to judge the quality of software reviews.
8. Students should be able to judge the quality and suitability of software information sources.

Methods

Lecture/discussion, readings, hands-on exercises, evaluation project, student presentations.

Midterm exam (only diagnostic). Grade based primarily on three-hour final exam, project, and reflective paper.

If you have a documented disability and wish to discuss academic accommodations with me, please contact me as soon as possible

Text: Nielsen, Jakob. **Usability Engineering**. San Diego: Academic Press; 1993. 362 p. ISBN: 0125184069. Reissued by Morgan Kaufmann Publishers 1994.

Each day:	.1 9:00-10:15,	.2 10:30-11:45,	.3 12:30-1:45	.4 2:00-3:15	.5 3:30 - 5:00
-----------	----------------	-----------------	---------------	--------------	----------------

1.1	Sa June 7	Introduction The software selection process - systems analysis
1.2		Functions/types of software
		Software evaluation and selection - general principles
1.3A		Software information sources (needed here for term project)
1.3B - 1.4		General software specification and evaluation criteria
1.5A		Software testing and evaluation procedures
1.5B		Organizing a software collection. Organizing the software review process
2.1	Su June 8	The user-system interface. Human-computer interaction
2.2		Continued
2.3		Display design (60 min)
2.4A		System messages, help, and error handling (40 min)
2.4B		Documentation (35 min)
	Sa June 21	Usability testing
3.1 - 3.2		Usability testing (discussion of Nielsen's book)
3.3 - 3.4		Usability testing of Web sites
3.5		Midterm exam
	Su June 22	Software in specific environments
4.1		Software in education and training
4.2		Software in personal information management
4.3 - 4.4		Final review
	June 23 - July 11	Work on project
	Sa July 12	Specific types of software (student reports)
5.1 - 5.5		Student reports
	Su July 13	
6.1 - 6.2		Student reports
6.3 - 6.5		Final exam 3 hours closed book Can deliver paper in hard copy
	M July 21	Paper due (12 midnight by email)

Each day:	.1 9:00-10:15,	.2 10:30-11:45,	.3 12:30-1:45	.4 2:00-3:15	.5 3:30 - 5:00
-----------	----------------	-----------------	---------------	--------------	----------------

1.1	Sa June 7	Introduction The software selection process - systems analysis <i>Soergel, Organizing Information, Chapter 6 Systems analysis</i>
1.2		Functions/types of software Software evaluation and selection - general principles
1.3A		Software information sources (needed here for term project) <i>Truett, Carol. Evaluating software reviews: A review of the reviews. Library Software Review; 1984. v.3, n.3, p. 371-378</i>
1.3B - 1.4		General software specification and evaluation criteria
1.5		Software testing and evaluation procedures
2.1		Organizing a software collection. Organizing the software review process
2.2	Su June 8	The user-system interface. Human-computer interaction
2.3		Continued <i>Pollitt, Steven A.; Smith, Martin P.; Braekevelt, Patrick A J; Treglown, Mark. HIBROWSE for EMBASE - the application of view-based searching techniques to Europe's most important biomedical database. INFO 96, 11th Annual Information Meeting, Tel Aviv, Israel, May 5 - 9, 1996</i>
2.4		Display design (60 min) <i>Some pages from Shneiderman, Human-Computer Interface. 1998</i> <i>Pages from 670 Lecture 6.2 Document design</i> <i>Some useful document design guidelines compiled by D. Soergel</i>
2.5A		System messages, help, and error handling (40 min) <i>Some pages from Shneiderman</i>
2.5B		Documentation (35 min) <i>Review and table of content of Thomas Barker book</i>

	<p>Sa Jun 21 Usability testing</p>
<p>3.1 - 3.2</p>	<p>Usability testing (discussion of Nielsen's book) <i>Several lists of usability heuristics</i> <i>Usability test plan elements from Nielsen</i> <i>Typical quantifiable usability measurements from Nielsen</i> <i>Usability methods from Nielsen, Usability Net, and Usability Methods Toolbox</i> <i>Examples of CAUSE tools from Nielsen</i></p>
<p>3.3 - 3.4</p>	<p>Usability testing of Web sites <i>Criteria for evaluating information sources</i> <i>Web usability inspections (slide presentation), Constantine & Lockwood, User Interface '99, San Francisco</i> <i>Some forms and reports from Constantine & Lockwood</i> <i>Some pages from usability.gov on usability methods to read</i> <i>p.83 - 90 More on methods and conduct of a study</i> <i>WAMMI Web usability questionnaire</i> <i>Stanford Library System Web usability questionnaire</i> <i>p. 95 - 98 Usability testing tools</i> <i>Pages from two Web usability reference sites</i> <i>Levi, Michael D.; Conrad, Frederick G. A heuristic evaluation of World Wide Web prototype. From Bureau of Labor Statistics (Reading)</i> <i>Melody Y. Ivory, Rashmi R. Sinha, Marti A. Hearst. Empirically Validated Web Page Design Metrics. Appearing in ACM SIGCHI'01, March 31-April 4, 2001, Seattle, WA (Reading)</i> <i>Melody Y. Ivory, Marti A. Hearst. Statistical profiles of highly rated Web pages. CHI 2002 (Reading)</i> <i>Web tango Automatic Web site evaluation</i></p>
<p>3.5</p>	<p>Midterm exam</p>
	<p>Su June 22 Software in specific environments</p>
<p>4.1</p>	<p>Software in education and training <i>Soloway, Elliot. Reading and writing in the 21st century. Communications of the ACM. 1993 May; v.36, n. 5: p. 23 - 27 (Reading)</i> <i>Wikipedia. Computer-supported collaborative learning (Reading)</i> <i>CSILE. Computer-supported intentional learning environments (Reading)</i></p>
<p>4.2</p>	<p>Software in personal information management <i>CALO (A Cognitive Assistant that learns and organizes) Slide presentation (Reading)</i></p>
<p>4.3 - 4.4</p>	<p>Final review</p>
<p>June 23 - July 11</p>	<p>Work on project</p>
<p>Sa July 12</p>	<p>Specific types of software (student reports)</p>

5.1	- 5.5	Student reports
Su July 13		
6.1	- 6.2	Student reports
6.3	- 6.5	Final exam 3 hours closed book Can deliver paper in hard copy
M July 21 Paper due (12 midnight by email)		

Term project

In the term project, an individual student or a team of students will apply the general principles discussed in the class to a specific type of software of their choosing. The numbers given are for individual students. So when it say 5 reviews, each student in a team of three would analyze 5 reviews for a total of 15 (or you can do some duplicates and compare results).

Note: Deliverables due on a date when there is no class are due by email

- 1 **Choose a type of software** and discuss with instructor. You may choose a scenario (such as a school, a company, a doctor's office) in which this software is to be used. Indicate in a paragraph why you are interested in this type of software.

Deliverable: Paragraph on your interest in type of software chosen. **Due Su June 8**

- 2 **Review literature and develop evaluation checklist**

- 2.1 **Assemble and read literature about this type of software and its use.** This should include literature on the function the software supports, the mental operations necessary to carry out the function, and how the program supports or takes over these operations. It should also include software reviews and articles on how to evaluate this type of software.

Deliverable: A bibliography **Due Th June 12**

- 2.2 **Prepare systematic notes of what you learned from the literature.** Develop an **evaluation checklist**, i.e. a checklist of criteria for describing and evaluating this type of software and an evaluation plan, which specifies the methods used for evaluation. (Type the checklist on a word processor so that it can be changed easily.)

Deliverables: Notes summarizing essential points from the literature **Due Su June 15**
Evaluation checklist Version 0

- 3 **Identify several programs of the chosen type** (3 - 5, more for a group). Identify reviews of these programs (1 or more reviews for each program for a total of 5 per group member) through a search in appropriate sources; prepare a short description and **preliminary evaluation** for each program.

- 3.1 **Search software and software review sources**

Describe your search:

- Sources used
- For each source:
 - Search criteria used / query formulation
 - How easy was the search, what other access would be desirable
 - Information obtained, adequacy of that information

Comment on the suitability of each source for a search for software of your type.

3.2 **Prepare a brief annotation for each of 5 reviews.** Each annotation on a separate page, giving

- Basic data on the program.
- The name and credentials of the reviewer and the source of the review.
- An abstract of the review that summarizes the main points (max 100 words).
- A brief critique of the review (strong points, weak points)

3.3 **Put the information together:**

For each program considered, make a summary sheet. Arrange review abstracts for each program after its summary sheet. The summary sheets provide the basis for selecting a program.

Amend the evaluation checklist and the evaluation plan; include amended version of checklist, mark items added with an *, items you would delete with a #.

From the best of these programs select one for in-depth evaluation. (A group may select more than one program for comparative evaluation or do a thorough evaluation of one program of sufficient complexity.) The program(s) selected must be available for testing (Own copy, in CLIS, or in the Computer Science Library, see note after assignment).

Deliverables:

All due Sa June 21

Search report with comments on the sources used

Annotations on five software reviews (use the form if convenient, but could be briefer)

Evaluation checklist Version 1

Summary sheets for 3-5 programs

Paragraph giving the rationale for selection of one program for thorough evaluation.

(½ page or less)

(For a team, there may be more than one program for in-depth evaluation.)

4 Do an **in-depth evaluation of one program** (possibly 2 or 3 for a team)

4.1 **Find five published reviews** (including posted on the Web) for each program selected for in-depth evaluation and prepare a comparative analysis of the quality of these reviews. Use the Software Review Evaluation Form for each review, then compare. (This is done by each student individually).

4.2 Then **conduct a thorough evaluation** of the program(s) selected, following the evaluation plan specified. The evaluation checklist and the evaluation plan may be amended in the process. Revisit your analysis of the five reviews in light of what you have learned from your own evaluation. How did the reviews help you in your own evaluation? What important results of your own evaluation were missing from the reviews? Do you agree with the statements and conclusions of the reviews?

- 4.3 **Prepare a class presentation** in class; the presentation should be based on handouts, to include at least the evaluation checklist with results for the program being evaluated and examples (screens, messages etc.) that illustrate the program's functioning and the reasons for the scores on the checklist. The presentation can include a demonstration of the program. The presentation should give the audience a good understanding of how the program operates, its strong and weak points — a good "feel" for the program.

Presentations will be on Sa July 12 and Su July 13.

Deliverable: Presentation with an outline that can be distributed
I will review the slides and give you feedback by F July 11

Due: Th July 10

- 4.4 Using feedback from the presentation, **prepare the final version of a written evaluation paper**

Deliverable: Final evaluation paper

Due: Monday, July 21

In addition to the Software Evaluation Project:

Each student individually:

Write an **essay of 3 - 5 pages summarizing your thoughts on software evaluation**

Deliverable: Reflective essay on software evaluation

Due: Monday, July 21

Sample project topics

Bibliographic software (for example, compare Library Master and ProCite)

Thesaurus management software

Classification software (Dewey for Windows, Classification Plus)

Software to support social work

Educational software (such as thinking skill software, story writing software)

Gardening software

Content management software

Software supporting any kind of business process

Software Review Evaluation Form

Software title:

Bibliographic data

Abstract of the review

Evaluation criteria

1	Basic information is given at beginning of text (best) or buried in text (worst). Technical specifications are given.	0 1 2 3 4
2	A set evaluation format with standard, clearly stated criteria is given (for example: performance, ease of use, error-handling, appropriateness, documentation).	0 1 2 3 4
3	A summary of each criterion or evaluation standard is given, on a rating scale.	0 1 2 3 4
4	The credentials of the reviewer are clearly stated. The credentials are good	0 1 2 3 4
5	Informative, evaluative comments, for example: all major functions are discussed, compatibility with other programs (exchange files easily, data import and export), usability by different classes of users, strengths and weaknesses of the software are given. See Software Evaluation Criteria for detail	0 1 2 3 4
6	The reviewer compares the program with similar programs of the same type. Feature-by-feature comparison table.	0 1 2 3 4

7	Available, or unavailable documentation is discussed.	0 1 2 3 4
8	A rebuttal to the review from the producer, publisher, or author is included in the review.	0 1 2 3 4
9	Illustrations depicting sample frames from the program are provided.	0 1 2 3 4
10	Feedback from actual users is included in the review.	0 1 2 3 4
11	URL is given to download this software.	0 1 2 3 4
12	Quality of writing and formatting, for example: Subheadings, small labeled chunks, major points pulled out, "Bottom line" section with advise, use of graphics and different fonts, screen shots are included.	0 1 2 3 4
13	Strong points of the review (or indicate in main text with +)	
14	Weak points of the review (or indicate in main text with -)	
15	My rating of the review.	0 1 2 3 4

Lecture outlines

Lecture 1.1

Introduction

Overview

Term project

The software selection process – systems analysis

See attached chapter from Soergel, Organizing Information

Lecture 1.2. Functions / types of software

Some programs serve primarily one function, others serve several functions in differing combinations. Program types are by no means clear cut. **This list is not complete.**

Programs used by end users; program modules used by programmer - software reuse

System software

- . Operating system (OS)
 - . Operating system for a network
 - . Small network OS
 - . Peer-to-Peer network OS
 - . Grid OS
 - . Web services framework / middleware
- . Operating system enhancement
- . Programming language (compiler or interpreter)
- . Debugger
- . Utility functions (often included with OS)
 - such as backup, disk organization, data compression

Information management

- . Integrated information management package
- . **Database management system (DBMS)**
 - . Object-oriented DBMS
 - . Relational DBMS (with or without programming language)
 - . Flat file manager
 - . Mailing list manager, contact manager
- . Data dictionary creation and maintenance
- . Knowledge Organization System (KOS) (ontology, thesaurus) creation and maintenance
- . Expert system shell
- . Library management
 - . Digital Library Management System (DLMS)
 - . Digital library software system
 - . Digital repository system
 - . Library automation

Web-related software (some of these repeated later in a functional context)

- Web browser software (HTML, XML with XSL style sheet capability)

- HTML and XML editors

- Web server software

- Intranet management (see also content management)

- Web-enabled database management software

- Middleware: Access to databases through Web pages

Information retrieval

- Online retrieval software (e.g., Dialog)

- Online front-end, meta search engine

- OPAC

- Bibliographic system, bibliography manager

- Indexing, esp. book indexing

- Text, image, and sound management. Content management

- Text retrieval

- Hypermedia system

- Multimedia system`

- Image management

- Sound management, sophisticated voice mail

- Geographic information system (GIS) software

- Support for qualitative research

- Information and referral

- Genealogy

- Personal information manager

Number processing and mathematics

- Spreadsheet

- Statistical computations

- Mathematical manipulation (e.g. symbolic differentiation and integration)

- . **Document generation and formatting**
 - . Text editor/Word processor
 - . Spell checker
 - . Grammar checker
 - . HTML/SGML editor, tagger, checker
 - . Document formatter
 - . HTML browser
 - . Computer typesetting
 - . Desktop publishing
 - . Graphics production
 - . Presentation manager
 - . CAI (Computer-Assisted Instruction) authoring program
 - . Music composition and presentation
 - . Text analysis software
 - . Automatic or computer-assisted translation
- . **Data conversion**
 - . Data conversion from one format to another (e. g., graphics formats or word processor formats)
 - . OCR (Optical Character Recognition)
 - . Automated speech recognition (ASR)
 - . Voice output
 - . Braille output
 - . Sign language output (does it exist?)

Computer-aided design (CAD), drawing

Work plan and project management BT Information management

- . Personal information manager. Group / collaborative information manager
- . Project manager
- . Simple calendar/appointment book. Group / collaborative scheduling
- . Contact manager

Computer-supported cooperative work (CSCW) NT Collaborative scheduling

Communication software

- . Fax
- . E-mail
- . Easier access to specific systems (e.g. AOL software)
- . Network access, e.g. WWW browser
- . FTP
- . Remote access to computer (incl. personal computers)

Process control

Business operation support (for organizations or the home) RT Information management

- . Business function
 - . Enterprise resource planning (ERP)
 - . Decision support
 - . Customer relations management (CRM)
 - . Payroll and accounting
 - . Home finance manager
 - . Tax preparation
 - . Supply chain management
 - . Ordering, including library acquisition
 - . Inventory control
 - . Library circulation
 - . Planning and control of operations RT Project management

- . Business support by specific business
 - . Home
 - . Will maker
 - . Professional practice
 - . Law practice
 - . Medical practice
 - . Library operations, library automation
 - . School operation
 - . Church/congregation operations
 - . Non-profit operations (esp. fund raising)

Education

- . Thinking skills and problem solving
- . Education - by method/function
 - . Learning through using (e.g., a game requiring addition; a drawing program requiring dealing with lengths and angles)
 - . Drill and practice
 - . Interactive textbook
 - . Support for subject-specific operations (mathematical operations, computations in physics and chemistry, simulation)
- . Education - by subject
 - . Education - mathematics
 - . Education - science
 - . Education - language arts
 - . Education - history and social studies
 - . Education - art
 - . Education - music
- . Education - by level
 - . Education - preschool
- . Education - by characteristics of students
 - . Students with special needs
- . Management of instruction

Games

- . Thinking/logic games, puzzles
- . Video games
 - . Adventure games
 - . Shooting games

Data-based applications. Databases with retrieval and processing software

Examples

- . Dictionary
- . Translation program
- . Bible with search program
- . City maps

Lecture 1.3A. Software information sources

Types of sources

Note: Three formats: Print, CD-ROM, online / Web.

- . **General information sources, esp. in computer science / computer applications**

- . . Journals and magazines, Web sites
- . . Bibliographies / bibliographic databases
- . . General Web search engines and directories. Search for *software* in combination with the subject. In Yahoo search for *software* to find categories under subjects. To find information on types of software, search for

software AND (criteria OR checklist OR evaluation)

- . **Information sources on software or computer or educational technology products**

- . . Review journals or magazines
- . . Software/product directories
- . . Catalogs from distributors, including e-commerce Web sites
- . . Promotional materials from software producers
- . . AOL, etc. forums, list serves of computer user groups. News groups. Often lots of messages about problems.

- . **Journals and Web sites in the subject area of the software**

Characteristics of sources. Evaluation

. Coverage

- . . Definition by
 - . . . Subject area;
 - . . . Type of computer and operating system;
 - . . . Commercial status of programs (regularly published, shareware, public domain);
 - . . . Country of publication;
 - . . . Time scope (only current or retrospective).
- . . Comprehensive vs. selective.
 - . . . If comprehensive: How complete?
 - . . . If selective: How good is the selection?
- . . Currency

. Information/detail given for each program

- . . Features covered. (See list of software evaluation criteria, both general criteria and criteria for specific types of programs.) Principle of selection of the features covered.
- . . Detail in which features are covered.
- . . Are evaluations given? If so, how trustworthy are the evaluations? (Evaluation procedures used, competence of the evaluators, individual evaluator or group consensus) Items considered in the evaluation.
- . . Are programs compared? How systematic is the comparison?
- . . Are descriptive/evaluative characteristics boiled down to descriptors/scores for easy comparison and searching? Quality of the index language for each characteristic.
- . . How is the information for each program and for program comparison arranged?

. Searchability

- . . By what elements can the database be searched? For print form: What indexes are provided. For CD-ROM and online: What data elements are searchable.
- . . Mechanics of searching

. Evaluate these characteristics in light of the requirements of a given search for software.

Lectures 1.3B - 1.4.**Software specification and evaluation criteria****Outline**

- A General Information
- B Capabilities
- C Program correctness. Bugs
- D Computer environment required/supported (hardware and software)
- E Cost and ease of use
 - . Installation cost and maintenance cost (capital cost)
 - . Daily operating cost
 - . Ease of use
- F Documentation. Vendor support. Error-handling
- G User interface features
- H Multi-user capability
- I Security
- K Input/output and working with other programs

A General Information

Name of program
 Producer/vendor
 General type
 Length of time on the market, last enhancement
 Number of sales
 Price
 Demonstration version (available, cost, limitation)
 Examples or built-in applications included

B Capabilities

List of capabilities specific to the type of program. See comparative evaluations in computer magazines and publications on selecting a specific type of software.

C Program correctness. Bugs

D Computer environment required/supported

Indicate minimum values, recommended level, and maximum usable level (e.g., with respect to main memory). (If more powerful hardware can be used, it often leads to improved performance.) Indicate connection of hardware requirements to software features. Programs that can be used in many environments are often preferable; for example, a DBMS that runs on many machines from mainframe to PC and under many operating systems can be used to manage a distributed database.

Computer environment requirements may be used in two ways:

Forward mode: Have computer environment, need software that runs in it

Backward mode: Have selected software, need to create computer environment for it. Used when buying new hardware or upgrading existing hardware.

D1 Hardware required/supported (minimum required - maximum usable)

- . Size of computer - mainframe, mini, micro
- . Make(s) and model(s)
- . Main memory
- . Disk requirements/support (also dependent on operating system, including enhancements)
- . Output devices supported (printer, plotter, monitor)
- . Output formats supported (esp. Web and PDA/cell phones)
- . Other hardware (e.g., soundboard)

D2 System software requirements/compatibility

- . Operating system
- . Network operating system (such as Novell)
- . Compiler, interpreter, DBMS, Spreadsheet, etc.

E Cost and ease of use

The cost of acquiring or producing a program is only a small part of the total system cost. Decisions based on purchase price alone may be quite shortsighted. Ease of use is a major determinant of total system costs. Other determinants are computer time and storage space used. All these often depend on specific program features used (e.g., sorting), and details should be recorded as part of the feature analysis.

E1 Installation cost and maintenance cost (capital cost)

- . Cost of acquisition (incl. cost of evaluation) or production of program
- . Cost of getting the program running in a given computer environment
- . Cost for building applications (for example, when using a DBMS)
- . Training cost (see also Ease of use)
 - . . Training cost for people defining applications (e.g., setting up an application using a DBMS)
 - . . Training cost for data entry personnel
 - . . Training cost for data use personnel
- . Maintenance cost

E2 Daily operating cost

Program efficiency depends on algorithm and programming language used. Is a function of file size, types of data, feature used, and computer system (hardware and system software) used (example: speed of hard disk, use of disk cache); need accurate data taking all of these into account

- . Computer time for daily use
- . Staff time for daily use
- . How fast does it run
 - . . How fast are batch updates and generation of reports – influences computer time needed
 - . . Response time in interactive use – influences computer time needed and staff time.
- . How many computer resources does it need (important in multi-tasking or networking environments)
 - . . Efficiency of storage (e.g., data compression)

For programs that manage information:

- . Cost for building the data base (very broadly defined to include, for example, text files and spreadsheet files)
 - . Staff and computer time for data entry
 - . Computer time for data base creation and maintenance
- . Cost for using
 - . Query input
 - . Computer time for data base searching and data analysis and formatting
- . Cost for data transmission

E3 Ease of use

Ease of use is a factor in both staff time and equipment use time

Ease of use is a complex concept; it includes aspects of learning (amount of learning required, ease of learning) and ease of performing operations; more precisely, one should look at the ease of performing operations as a function of how much the user has learned. Furthermore, each of these aspects should be considered for each of the program's functions; for example: how easy is it to learn the operation *block move* in WordPerfect as compared to Word. How easy is it to carry out the operation once learned. Further complication: there may be a slow way that requires no learning (such as going through a series of menus) and a fast way that requires learning (such as Ctrl-Key combination, often called a shortcut. one must first learn

- . Ease of performing operations
 - . Cognitive load — memory and thinking required
 - . Efficiency of performing operations (e.g., number of keystrokes or mouse clicks) (s.a. macros)
- . These two factors interact: a user willing to think may need fewer keystrokes
- . Amount and ease of learning
 - . Amount of learning required
 - . Some factors (see the Lecture on User Interface for more):
 - . Learning what can do vs. learning how these functions are invoked
 - . Intuitiveness
 - . Consistency within the program
 - . Similarity/consistency with other programs (can be achieved if customizable)
 - . Good documentation and online help reduce learning required
 - . Ease of learning (e.g., learning during use)
 - . Ease of remembering
 - . Programs that are intuitive are good on all three counts
 - . Customizability

- . Availability in user's native language ("localization" of software)
 - . Ease of use by general function
 - . Ease of system definition (includes, for example, setting up for producing certain types of documents with a word processor)
 - . How well can system definition be done by non-systems-people
 - . How well can system definition be revised by non-systems-people
 - . Ease of data entry
 - . Ease of data use
 - . Appropriateness of default settings
- The characteristics under E and F all affect ease of use and performance, thus cost.

Performance measurements

F Documentation. Vendor support. Error-handling

F1 Written documentation

- . Demo
- . Tutorial
 - . Printed tutorial
 - . Online tutorial
- . Reference manual
- . Quick reference chart
- . Other (list)
- . For each:
 - . Readability/clarity,
 - . Ease of finding specific points,
 - . Speed of finding important things,
 - . Separation by levels of complexity,
 - . Completeness,
 - . Correctness,
 - . Avoidance of redundancy

F2 On-line help (see also error messages)

- . Continuum of specificity: from very general to very specific to the situation (context-sensitive). Pressing the help key results in
 - . Very general: Display of a general index to help
 - . General: Display of an explanation of the general functional area the user works in when the help key is pressed, with short explanation of individual functions
 - . Somewhat specific: Explanation of the currently highlighted menu choice, regardless how specific. When called in response to an error: Standard explanation of the function in which the error occurred; repeating manual definition of the command in which an error occurred.
 - . Very specific: If called in response to an error - trying to analyze the error precisely and suggest specific corrective action
- . Method of "moving around" in online help (e.g., hypertext functionality)

F3 Vendor support

- . Medium used: Phone, automated fax, e-mail, Web site, bulletin board
- . Evaluation criteria
 - . Charges (for phone call, for service)
 - . Response time
 - . Correctness (problem fixed?)

F4 Error handling

- . Types of errors
 - . Error in system specification (in programming, in specifying database structure in a DBMS or a template in a spreadsheet)
 - . Error in system use
- . Quality of error detection
- . Error messages (see also online help)
 - . Very general (extreme: only message that error occurred; usually at least the place of the error, such as the line in a program)
 - . Identify the nature of the error (command or variable unknown, variable of wrong type, data value input not in authority list)
 - . How precisely is the error identified?
 - . Give more information about the error (exact location of error in the line, which variable is unknown, which file not open)
 - . Error correction - system guesses what user wants and asks user for confirmation (e.g., system finds the command closest to the string entered by the user or the closest value in an authority list)
- . Error messages defined by
 - . System
 - . User
- . Error recovery
 - . System can restore state before data were erroneously changed by operator mistake or system malfunction (e.g., restoration of a text to nullify an erroneous global replace).
- . Assistance in debugging of stored programs, report definitions, etc.

G User interface features (see Lectures 2.2 - 2.3 The user-system interface)

G1 Interaction style (give all that apply)

What combination of

- . Menu-driven
 - . System-defined menus
 - . User-defined menus
- . Script-driven
- . Direct commands (immediate mode)
- . Programming language
 - . Part of program
 - . Power of programming language, specifically flow-of-control commands
 - . Other programming languages can be used easily with the program
- . Nature of the command/programming language
 - . Specific vs. general commands
 - . Complexity of commands (general commands need more complex syntax)
 - . Brevity of commands (do long and short forms exist?)
 - . Consistency of command syntax (particularly important when defining macros for a recurring command part)
 - . Affinity of the command syntax to natural language (these two criteria may conflict)
- . The interaction style may differ from one program function to another. In a detailed analysis, indicate for each program function whether it is
 - . Achievable through menu selection
 - . Achievable through one direct command or a short sequence of direct commands (These commands may be stored as a program, but flow-of-control commands such as DO WHILE or IF..THEN..ELSE..ENDIF..are not used)
 - . Achievable through programming (if the programming language is reasonably powerful, almost anything is achievable through programming, and this need not be mentioned with every feature)

G2 History of user actions

- . Ability to edit or recall and edit input line in immediate mode
- . . Ability to edit input line before entering
- . . Ability to recall last entered line, edit, and re-enter (especially in case of an error)
- . Ability to save commands or groups of commands, queries, reports formats, macro definitions, game status, etc., on disk with a name for later execution or restoration to memory
- . . Ability to edit saved items
- . . Ability to do this for items generated by menu selections
- . Ability to annotate history items)
- . Ability to save the work space (for example, open documents)

G3 Defaults

- . Defaults used where appropriate
- . Sensible defaults

G4 Customization. Macro and function key definition

- . Configuration customizable by user
- . Ability to define macros
 - . . Keystroke macros
 - . . Result macros
 - . . Choice
- . Ability to edit macros, once defined

(One or both features may be available in programmable keyboards or through programs extending the operating system or in a and can then be used in any software compatible with the operating system extension)
- . Ability to define function keys (function key as macro name in direct mode)
 - . . How many? (function key alone or in combination with SHIFT, CTRL, ALT; can other keys be used as function keys?)
- . Characters with special meaning can be specified by user
- . "Literal" operator to use a special character literally rather than for its function

Ability to create turnkey systems for specific applications

For DBMS and interpreted programming languages: Run-time version available

H Multi-user capability. Network version

System keeps a file of users (a prerequisite for security)

System can run user-specific start-up program upon sign-on

System can be used by several users simultaneously

I Security

While security is most important in data base management systems, it is also an issue in text processing, and spreadsheets, and others. Security is especially important in a multi-user environment. Big issue on the Internet / Web.

I1 General issues

- . Clearance for accessing computer in the first place; keeping hackers out (firewall)
- . Adequacy of the security access scheme (usually user name and password)
- . User-changeable passwords. Check on password difficulty
- . Security of data transmission
- . Audit trail, reports of unsuccessful access attempts

I2 Access to data

The possibilities for stipulating security definitions in a DBMS can be described along two dimensions:

Function	Protecting data integrity			Restricting data distribution
	Data definition	Data entry	Data modification	Data use: viewing printing
System				
Data base				
Data field				
Specific record or data field by content				

Authorizing the user vs. authorizing the authorizer

Data use: Display data vs. using a data field for selecting data.

Using individual data vs. aggregate data.

To what extent can security stipulations be differentiated?

I3 Type of security definition

- . Directly by user name and password
- . By access codes assigned to users on the one hand and each level/function combination on the other

Example: User 1: ak1 User 2: blt File 1 for reading: gkm
 User 1 may read File 1, User 2 may not

Protection of files from access outside the program (e.g., encryption)

K Input/output and working with other programs

(The operating system or utilities may provide some of these capabilities)

K1 Input/output

- . What input and output devices are supported
- . Ability to specify the device for any input (e.g., through setting the value of an environment variable)
 - . Keyboard, disk, modem, mouse, scanner, etc.
- . Ability to specify the device for any output
 - . Screen
 - . Printer
 - . Disk
 - . Transmission port
 - . Other (specify)

K2 Data input and output formats. Data exchange with other programs. HTML support

- . Data import formats (esp. problematic for graphics)
 - . Delimited text
 - . Data format used by another program
- . Data export formats same as above
- . Other programs that can use any of these data export formats (or perhaps even files in the analyzed programs internal format)

K3 Integration with other components which interface directly.

- . Nature of integration, for example plug-ins

K4 Ability to use operating system commands from within the program being analyzed

K5 Ability to call other programs from within the program being analyzed (for complex data manipulation, results are communicated through parameters or a common data area)

K6 Relationship to other software

- . Availability of software enhancing capabilities (e.g., collections of macros or special keyboard definitions for WordPerfect)
- . For system software: application software written in it
 - . Are applications included with the package?

Lecture 1.5. Software testing and evaluation procedures

Testing vs. evaluation

Context of testing and evaluation

- . Purpose of evaluation
 - . Testing and evaluation by or for producer
 - . Formative evaluation. Purpose: Improve product
 - . Testing and evaluation from a consumer perspective
 - . Summative evaluation. Purpose: Support decision on which product to use

Perspectives of testing and evaluation

- . Program functionality – feature analysis
- . Program correctness: Does the program do what the programmer or his supervisor meant it to do?
- . Program performance: How efficiently does the program do its work? How much time does it take?
- . Program usability: How well can the program be used by the intended users?
- . Program impact. Does the program have the intended ultimate effect? (Learning, productivity, etc.)

Testing procedures

For feature analysis

- . . Check feature list against task requirements

For program correctness

- . . Test all features (batch or interactive). May need to develop a test protocol to be followed and record all results
- . . Need test data that cover all possibilities.
- . . Walking through all possible paths of menu choices
- . . Examine outputs carefully. May also need to examine what is written to a database.
- . . Check for program parts that have not been executed (there is testing software that

assists in doing this)

- . . Test in different hardware/software environments

- . . This kind of testing is also done during program development (debugging)

- . **For program performance**

- . . Test time for each feature or at least important features and/or features that can be expected to take a long time (such as a global change in a large document or spell check).

- . . Need to test with data sets of realistic size.

- . . May need to test for different hardware configurations.

- . **For program usability**

- . . Needs test with real users or at least people who might be real users - problem of sample selection. May need to include different types of users if the program is to be used by a heterogeneous user group.

- . . Record times needed to do certain tasks.

- . . Record features used and also features not used when they would have been useful.

- . . Record what people have difficulty with. Record when they used help.

- . . Record errors, whether the user was able to recover from the error, and how long it took to do so.

- . . Analyze data taking into account the characteristics of each test user.

- . . Recording key stroke data might be helpful.

- . . Such testing done in the laboratory (usability lab, possibly on prototypes) and in the field (beta test). Field tests will bring out many problems not thought of before. Continues into the phase when the program is actually sold and used (need mechanism to receive user feedback).

- . **For program impact**

- . . Most difficult. Need before-and-after studies. Useful also to software producers for marketing.

Forms for recording results

Lecture 2.1

Organizing a software collection. Organizing the software review process

Organizing a software collection

The software collection could be all software included in a directory, or all software in a school district, or in an academic library for student and faculty use, or in a public library for public use, or in a company or other organization. In an organization the programs may be physically in one location or distributed over many offices; they should still be recorded and cataloged.

In an organization, a software collection should be seen as part of all information resources; integration allows unfragmented service to users for all their information needs.

Steps

- . Determine user needs and background.
- . What kind of software
- . Based on that, develop selection criteria with weights (see previous lecture). May need different sets of selection criteria for different groups of users.
- . Identify software to acquire. Use appropriate software information sources (see separate lecture).
- . Do in-house testing and evaluation as needed.
- . Acquire the programs selected (consider discount stores and site licenses).
- . Catalog the programs acquired.
 - . If the collection is part of a library, do library cataloging as Machine-Readable Data File (MRDF) following Chapter 9 of AACR2 using the MARC format. A catalog record may already be in OCLC. Distributed copies may be handled as permanently checked out items.
 - . This level of cataloging is not sufficient for purposes of managing a software collection well. A parallel system (with data transfer) should be maintained, using a format for the cataloging data suitable for the organization (see attached examples). This system should be searchable as appropriate to the organization.

Related topic: **Software reuse**

Organizing the software review process

Magazines have individual reviewers. Some also have labs to do testing. Some do user surveys (much like Consumer Reports, usually not very scientific).

Dedicated organizations

- . Nationwide collaborative organizations like microSIFT.
- . Statewide organizations like MECC
- . School district
- . Review in house by specially assigned professionals or outside reviewers. In any case follow a common format and procedure.
- . Results are widely available.

Magazines and review organizations often get review copies of new programs.

In an organization:

- . Rely as much as possible on published reviews for preliminary selection. Get demos, tutorials (from the producer or third party), manuals, and third-party books for detailed examination. May need to try out somewhere.
- . Establish a software review committee or in other ways involve users in the process.

Lectures 2.2 - 2.3

The user-system interface. Human-computer interaction

Outline

- A Interaction styles
 - A1 Command-driven: User enters commands at system prompt
 - Formal command language
 - Natural language
 - A2 Script: User fills in blanks on a form
 - A3 Menu driven: User selects from displayed options
 - A4 Direct manipulation
 - A5 Graphic User Interface (GUI): A combination of approaches
 - A6 Parameters for characterizing interaction styles
 - A7 Actual interfaces use a combination of these approaches
- B Human-computer interaction principles
- C User characteristics
- D Interface display characteristics (hardware and software)
 - Importance of interfaces that can be customized (user-definable interface)

A Interaction styles

A1 Command-driven interaction style

A1.1 Formal command language

Often require a high cognitive load. Recall vs. recognition - user must recall the command (s) and initiate action. Cognitive load can be lessened through good design. The structure of the command language should reflect the structure of the task and use elements already familiar to the user.

A1.1.1 Command structure

Human learning and recall is facilitated by meaningful structure. People are good at recognizing familiar structures and encoding it in long term memory.

Simple commands - a one to one mapping between command and task.

Commands plus arguments

SHOW m1

SELECT FROM m1 FOR m1.CourseNo = "FDST101"

Commands plus options plus arguments

DIR /w

PRINT /3, LASRBDGT.ICF

Faceted structure. For a set of tasks this can provide a meaningful structure.

Example:	Action	Object	Destination
	COPY	FILE	FILE

Example:	In WordPerfect, Word, and other applications			
	→,	←,	Del	apply to an individual character,
	Ctrl-→,	Ctrl-←,	Ctrl-Del	apply to a whole word.

(In the second example, the object of the action is implied by the position of the cursor.)

A1.1.2 Guidelines for command language systems

Create an explicit model of objects and actions. To the extent possible, this model should correspond to the way people think.

Choose meaningful, specific, distinctive names or keystrokes.

Stay close to other command languages (incl. natural language) (but this may conflict with internal consistency).

Define a faceted structure where meaningful.

Provide consistent structure.

Counter-example (from Guru):

```
DEFINE table WITH file
USE file AS table
```

Support consistent abbreviation.

Offer frequent users capability of creating macros.

Consider command menus.

Limit number of commands and ways of accomplishing tasks.

A1.2 Natural language

Useful in that no computer-related syntax needs to be learned. However, in a computer environment it is a complex case of a command language. The user still has to initiate actions and must know what commands the system can perform and the order in which these need to be entered.

Example of a natural language command (extreme).

I want to take the file SMITH.DAT from the directory SUPPLIES of the disk in the ZIP drive and put its duplicate in the dir ACCOUNT under directory COMPANY on drive C.

(Think how this type of interface would work in air traffic control!)

Test a natural language system whether it can "understand" this?

Duplicate SMITH.DAT from SUPPLIES directory in ACCOUNTS directory.

If not, then perhaps command or menu or hybrid may be better.

Command language equivalent:

COPY G:\SUPPLIES\SMITH.DAT C:\COMPANY\ACCOUNT

Script equivalent:

COPY

FROM: DRIVE: _ : DIR: _____ FILE: _____

TO: DRIVE: _ : DIR: _____ FILE: _____

A2 Script interface

Display a form with blanks to be filled in. Many "dialog boxes" in Windows use a script.

A3 Menu interface

A menu presents available choices from which the user may select. Choices may be actions or objects. Choices can be represented as text or as icons.

Example: The WordPerfect File List screen shows a menu of files in a directory (text).

If one clicks on File, a menu of possible actions, also described in text, appears.

A toolbar is a menu of actions represented as icons; on mouse-over, a text appears.

Menu driven systems can lower cognitive load. People are better at recognition than recall. No long and complex commands to remember. User can concentrate on task rather than computer syntax.

Menu selection systems are considered useful for novice or intermittent users who have not had time to learn a complex command language (or who forget because of infrequent use).

Menus can be bothersome for experienced users of a system. They are generally slower than command-driven systems. An experienced user must usually go through an entire sequence of menus even if the destination is well known. Short cuts can be made available.

There are many characteristics of menus, each of which could be used as a basis for classifying menus. We will discuss some of these.

A3.1 Static vs dynamic menus

Static menus are always the same (in the example: the menu of actions), dynamic menus are based on the current status of the system (in the example: the menu of files in a directory).

A3.2 Method of selection

Direct selection by pointing: Position the cursor over the choice, using the mouse or arrow keys; selection may require, in addition, clicking (or double-clicking) or pressing Enter.

Symbolic selection: Enter appropriate characters (the name of the element to be selected) from the keyboard. Symbolic selection allows for type-ahead and reliable macro construction.

Many menus give the user a choice of direct and symbolic selection.

A3.3 Number of menu elements

Binary menus (choose 1 from 2)

Examples:

Do you want to leave Word Perfect (y, n):_

Press 1 to start printing, 2 to stop.

Multiple item menus (more than 2 items)

Examples:

Not ready error reading drive A

Abort, **R**etry, **I**gnore?_

EXAMINE, PRINT, DROP, OR HOLD

1. Edit
2. Create
3. Delete
4. View
5. Quit

A menu may take more than one screen:

1. Edit a copy
2. Copy a document
3. Create a document
4. Erase a document
5. Print a document
6. View index to documents

Type the number of your choice or M for more choices
Then press RETURN

On pressing M a second screen appears:

7. Alter line width
8. Change character set
9. Attempt recovery of damaged file
10. Reconstruct erased file
11. Set cursor blink rate
12. Run diagnostic

Type the number of your choice or P to return to Page 1.
Then press RETURN

A multiple item menu may allow for **multiple selection**, that is, the user can choose multiple items at a time.

Example: In the WordPerfect List Files Screen, one can select multiple files by pressing * for each and then choose an action (copy, move, print) to be applied to all files selected.

A3.4 Menu organization

Menu system should have comprehensible, memorable semantic organization **relevant** to the user's task.

Alphabetical vs conceptual organization

Conceptual organization can be based on the user task domain to make it meaningful. Related items will be displayed together (chunking) thus relieving cognitive load. This cannot be done with alphabetical organization of menu items. Often groups of items are separated graphically (boxes).

A3.5 Format of menus

Skeleton menu

Embedded menu - Hypertext

A3.6 Permanence of menus

Permanent menus. Always on display at top or bottom of screen (or on keyboard)

Example: The menu bar in many applications; the task bar in Windows; a template giving the role of function keys.

Transient menus. Displayed on request (for example, by making a selection on another menu), disappear once selection is made.

Examples: Pop-up menus and pull-down menus

A3.7 Sequence of menus

Several menus are needed when a task needs multiple pieces of information, each of which has its own menu. The choices may be independent of each other or they may depend on each other.

Linear sequence Useful for guiding a user through a complex decision making process. Often seen in software installation programs.

Tree structure Useful when a hierarchical structure exists which is known to the user

Subordinate menus shown only as a heading (often in WordPerfect 5)

Some options of the subordinate menu shown on the screen, but selection of the subordinate menu leads to a new screen (sometimes in WordPerfect 5)

Subordinate menus shown with all their options on the screen (usually in box). Selecting the subordinate menu "activates" the box and options can now be selected without going to a new screen (often used in WordPerfect 6). (More precisely, in WP6 many screens in which options are displayed in boxes function as a single menu with very many options for direct selection with the mouse and as a menu tree for symbolic selection with the keyboard.)

Network structure Useful when access to a particular item or menu is desirable from more than one previous menu. Disadvantages: users can easily become disoriented.

A3.8 Menu combined with other interaction styles

Differences between command driven and menu selection systems are not always clear cut.

Example: command driven system which offers a help screen of commands.

ENTER COMMAND (? for HELP)

If ? is pressed, a menu type help screen appears.

Load, Edit, Run, Delete, Print(etc.)

Menu systems with symbolic selection could be considered command language systems with the available commands displayed. Many menu systems do indeed offer shortcuts, key stroke combination that accomplish an action that might otherwise require multiple menu selections. Short cuts are often shown in the menus to facilitate learning. Short cuts can also be given in the form of buttons - direct selection.

A3.9 Menu selection guidelines

Use task semantics to organize menu structure (single, linear sequence, tree structure, network)

Try to give position in organization by graphic design, numbering, and titles

Items become titles in walking down a tree

Make meaningful groupings of items in a menu

Make meaningful sequences of items in a menu

Items should be brief and consistent in grammatical style

Permit type-ahead, jump-ahead, or other short cuts

Permit jumps to previous and main menu, at the user's option

Use consistent layout and terminology

Consider novel selection mechanisms and devices

Consider response time and display rate impact

Consider physical properties of the screen: size, resolution, color, time to redraw

Offer help facilities

(From Shneiderman, B. (1998). Designing the user interface. 3. ed. Modified)

Designing the User Interface: Strategies for Effective Human-Computer Interaction, 4/E

Ben Shneiderman, University of Maryland

Catherine Plaisant, University of Maryland

ISBN-10: 0321197860 ISBN-13: 9780321197863

Addison-Wesley: 2005 (Published: 03/31/2004), 672 p.

Provide commands or buttons ("shortcuts") at least for frequently used items

Show shortcuts in the menus

Shortcut keys should work not only from the working screen, but also from the menu

A4 **Direct manipulation**

Objects - or rather representations of objects - on the screen are manipulated using the mouse and/or cursor keys.

Examples:

Pinball construction set

Moving a file from one directory to another

Changing size or location of a window

Allows:

- Continuous representation of objects and actions of interest
- Physical actions rather than complex syntax
- Rapid incremental reversible operations whose impact is immediately visible

A5 **Graphic User Interface (GUI)**

Characterized by

Multiple boxes/windows. Windows can be opened and closed as needed

Menus, often with icon representation

"Pull-down" menus (usually a menu of actions). Once an action is selected, a "dialog box" may be opened with supplementary menu choices and/or a script for entering information

Direct manipulation

A6 **Parameters for characterizing interaction styles**

Amount of information on available choices/expected input presented to the user

Method of data input (symbolic, direct selection, direct manipulation)

Note: If options are not shown on the screen, the user must retrieve them from his head, online help/documentation, a crib sheet or template, or the manual. Need for display on the screen depends on availability of the information from these other sources.

B **Human-computer interaction principles**

A dialog between a person and a computer program to accomplish some task.

B1 **Models/metaphor**

Dialog takes place in some related setting. Choice of appropriate metaphor makes the task easier for the user. Metaphor must be appropriate for the user; a print shop metaphor used in a word processing program is probably not useful if the program is used by secretaries. Metaphors that get in the way

"What you see is what you get" (WYSIWYG) is desirable?

WYSIWYG: Here is an **example** of boldfaced text.

Non-WYSIWYG: Here is an ^Bexample^B of boldfaced text.

Computer metaphor vs. task metaphor

If user is not a programmer, task metaphor is more appropriate.

Example: Should a text editor operate on lines or on sentences/paragraphs?

Examples of task metaphor in interface design:

Electronic spreadsheet

Cataloguing data input form

Desk top organization of operating system

B2 **Data entry and display**

Data entry

Consistency

Minimal input actions

Minimal memory load on user

Compatibility of data entry with data display

Flexibility for user control of data entry

Data display

Consistency of data display

Efficient assimilation by user

Minimal memory load on user

Compatibility of data display with data entry

Flexibility for user control of data display

B3 System messages

System messages that guide the interaction - always given - discussed above

Number and level of such messages

Help messages

General help

Context-sensitive help. Degree of context-sensitivity

Error messages

Amount of information given

C **User characteristics**

C1 **Physical characteristics**

Especially handicaps

Eye-hand coordination

C2 **Personality characteristics**

C3 **Cognitive characteristics**

Information processing model of cognition

Working memory (short-term memory)

Time and attention (forgetting). Without rehearsal, information held in working memory appears to be forgotten within 10 to 20 seconds (much of it even sooner)

Space. Limit on the number of **unrelated** items can be held even when full attention is given. Miller's "magic number 7 plus or minus 2"

Chunking. Items may be combined into **chunks** of related items.

Example: Ask a user to remember this sequence of characters:

SCMESRINEAVMOMDNYD

or this:

COMMAND DRIVEN SYSTEM

A single object can act as a "chunk" that supports memory of its various attributes. People are much better at remembering few objects with many attributes than many objects with few attributes.

Important: Working memory is easily overloaded!

Long-term memory

Forgetting - Decay is very slow.

Space - Virtually unlimited.

Organization - Highly organized.

Learning - New information is mapped to existing knowledge structures.

Spatial and visual ability

User knowledge

Syntactic

Varied, device dependent, acquired by rote memorization, easily forgotten.

Semantic

Structured, device independent, acquired through meaningful learning, stable in memory.

Computer knowledge

Computer concepts

Computer actions

Task knowledge

D **Interface display characteristics (hardware & software)**

D1 **Physical factors**

Legibility/visibility

Illumination & contrast

Resolution & character design

Color vs monochrome

Response time

D2 **Means of presentation** (treated more fully later)

Textual messages, data output

Graphics output

Symbols/icons

Generally good for concrete information, less good for abstract concepts

Color

Task considerations. For imaging and solids modeling many colors and shades can be used and discriminated. For information coding this is much more limited. Color coding can be useful where

Symbol density is high

Operator must search for specified information

Examples of use of color in information coding:

RED System failure. (Disk full)

YELLOW System warning (Only 5% of account left)

GREEN System fully operational (3 file (s) copied)

References

Shneiderman, B.

Designing the user interface. 3. ed.

Reading, MA.: Addison-Wesley; 1998. 639 p. isbn: 0201694972

Galitz, Wilbert O.

The Essential Guide to User Interface Design : An Introduction to Gui Design. Principles and Techniques.

NewYork: Wiley; 1996. 640 p. ISBN: 0471157554

Mandel, Theo

The Elements of User Interface Design.

NewYork: Wiley; 1997. 640 p. ISBN: 0471162671

Lecture 2.4

Display design

Goals

Transmit information as efficiently as possible.

Solicit appropriate instruction from user with minimum chance of errors.

Consider

Task

User

Environment (for example, lighting conditions)

Capabilities of the display device

Capabilities of the computing environment (hardware and software) (Graphics displays may take a long time)

Cognitive Match Interface Design (COMIND)

A computer system must be designed so that it operates in a manner which parallels or complements the human thought process.

Use symbols with known or easily inferable meanings.

Use analogy where helpful but not where it is restricting.

Some guidelines (in addition to what is found in readings and elsewhere in the lecture)

Show information needed for the task.

Omit from display or de-emphasize what is not needed at the moment.

Make groupings helpful to task. Where appropriate, use familiar groupings

Make sequence of groups and sequence within group helpful to the task.

In a sequence of displays:

Maintain the context.

Rewrite only those parts of the screen that change.

Do not obscure needed information through new windows.

Message area (as in a demo) should be in same place unless that would obscure essential parts of the screen.

Lecture 2.5A

System messages, help, and error handling

System message: Message related to system use. Excludes display of actual data asked for, such as bibliographic citations or a phone number.

Functions of system messages

Prompt for user input

Give feedback to the user

Warnings: Give information on the consequences of actions (unless obvious) and ask for confirmation

Help

Report on system status (particularly important when there is an operation that takes lots of computer time)

Error message

General message guidelines

Be specific.

Be brief yet understandable (not cryptic).

Be polite. For example, express the systems readiness to accept input or the systems inability to process input rather than giving orders to the user or blaming the user for giving erroneous input. However, this is not always possible or may be awkward. People are used to reading "orders" in instructions.

In instructions: Mention each step clearly in its own main clause in the proper sequence.

Not: After doing A, do B but: Do A, then B

(Even worse: Press F10 after you entered the data)

Message parameters

The following message parameters should be adapted to the intended user group or, even better, user-selectable.

Message length

Message expression (see Display design, High level means of expression)

Can message texts be modified or created by the application system designer or user.

Help

Content of help

What vs. how

Help in deciding what to do

Help in how to do it, for example description of steps, coaches, wizards

System only vs. local database also

Only on the system (DBMS, word processor, etc.) in general

Also values from the database (as in pop-up list of possible values from a field - this may or may not be classified under help)

When help is given

Anticipatory help - elaborate message as input prompt when it is expected that the user will have difficulty with the next input. This may be programmed into the system based on experience in testing or the system may monitor the user's inputs and detect when the user is lost.

Help on request - brief message jogging the user's memory or more or less elaborate explanation.

Help after an error. Such help may be given in three ways: As part of the error message, through automatically displaying a help screen, or on explicit request.

Levels of context-sensitivity

Not at all context-sensitive - the system displays the general help index.

Somewhat context-sensitive - the system displays the help screen for the command that is half-finished on the input line or that contained an error or for the currently highlighted menu item (often a command) or the help screen for entering data in the field in which the cursor rests.

Very context-sensitive - the system displays help specifically geared to the problem, e.g., "a file name is needed here" rather than just displaying the command syntax. Even better: Display a menu of the applicable files.

Help vs. on-line documentation

Some systems have the entire manual online, accessible through a hypertext structure and possibly through a text-search capability. Help display becomes a starting point to navigate the hypertext.

Error handling

First principle: Avoid errors. For example: In selection from a menu, there can be no typo.

The more flexibility and control the user has, the more errors can occur. Most errors in using a programming language.

Functions

Detecting errors - syntactical, logical/semantic

User/programmer may need to be involved in detecting errors - what kind of support is offered (debugger)

Announcing errors to the user

Correcting errors and/or giving help

Error message guidelines

See copy from Shneiderman 1998, p. 375+.

Also: Identify the program that generated the error message. For example, while using WordPerfect, one may encounter an error message generated by DOS. Particularly important in the now common multi-tasking environments.

Further criterion: Correctness of error messages

Examples of bad messages

Bad error messages

Stacker

error: unable to exec

MS-DOS

Seek error. Abort **R**etry **F**ail

(Error in seeking file on the disk; one user interpreted this message as an instruction to her to seek for the error)

Unable to create directory entry *(Reason : the root directory has max. 512 entries)*

GURU

Cannot open file *(Reason for error: File is already open)*

One version of software for installing IOMEGA ZIP drive

No drive letter added *(meaning: installation was not successful, but no reason is given)*

Newer version will give reason and a fix, for example

Not enough drive letters available in DOS. Use the DOS LASTDRIVE command to add a drive letter

Other messages (or lack thereof)

Windows

Directory display does not tell how much space is free

Directory display does not give file name; extensions (default, can be changed)

Press any key to continue

Most frequently asked question at a help desk: Where is the any key

Better message: Press Enter to continue (Enter is better than space, since Enter often is used to signal the end of an input or to accept an option.)

Library Master

Database for structure model (none = no model):

Typing "none" produces the message "no database by that name". The user must simply press Enter to work without a model.

Lecture 2.5B

Documentation

Purposes - need for documentation

- (1) Make evaluation of the program possible.
- (2) Enable user and operator to run the program.
 - Installation (often through a specialist)
 - Customization (often through a specialist)
 - Learning / training
 - Daily use by end user for accomplishing infrequent tasks or trouble shooting
 - Maintenance by specialist
 - A really well designed program may be usable without documentation - discussion
- (3) Make modifications of the program possible.
- (4) Monitor the development of the program and support communication among programmers.

Parts/content

These items can come in various forms. For (1) and (2):

Reviews

Demo

User's guide or users manual - final form of the functional specifications

Consists of

Tutorial and/or introductory manual. Enough information for standard use of program. List of all error messages. Related programs. Possibly sample files the user can practice with. Tutorial gives instructions for the user to follow along.

Reference manual - more complete, all exceptions, etc. Operator's guide - for computer center operator.

Reference card - summary of commands to support the user's memory

Brief data sheet describing precisely the environment required by the program.

Performance data document: "A definitive statement of testing and verification procedures. This should include a listing of test data, a statement describing which program sections were tested by each data group, and a description of the verification procedure." (Yohe, 1974.12, Computing Surveys, p. 240)

For (3) and (4):

Technical description - **System manual**

Possibly a more technical version of the functional specifications.

Description of files and data structures in main memory.

Description of the logic and structure of the program (the program listing itself should serve this purpose to the extent possible).

Forms of documentation

Paper documentation

Audiotapes, videotapes (mostly demos or tutorials)

Presentation. Training course

Machine-readable documentation, incl. CD-ROM. Tutorials using computer-assisted instruction (CAI)

Online access to materials. Web site. Knowledge base accessible through the Web

Interactive documentation (answering questions)

Bulletin board, list serve, Wiki (through producer or through user group).

E-mail, instant messaging

Telephone support

Take over user's computer

Sources for documentation

Producer / publisher of the program

Third-party publishers (source of reviews, textbooks and reference manuals, training materials)

User group (may have list serve or Wiki)

In-house

Some evaluation criteria for documentation

Availability of the different components (consider all sources)

For interactive documentation: Toll-free or pay; hours of operation; wait time for establishing connection; response time for questions that are not answered immediately.
Knowledge level of staff

Coverage

Are all features of the program documented?

Is there a list of errors, indicating the reason for each error and what to do to correct it?

Level of detail

Clarity and usefulness of overall arrangement or online access and presentation structure

Clarity of the writing - all the usual criteria, especially vocabulary appropriate for the audience

Examples

How easy is it to find answers to specific questions? Does the index provide access by **problems users want to solve**? Perhaps a separate "problem index" in classified order for browsing, since the user may not be able to find just the right word to describe her problem.

Computer-searchable documentation. Access points, search possibilities. Accessible from within the program? Directly accessible after error?

Layout and typeface

See the list from Giril in the lecture on display.

Lectures 3.1-3.2

Usability testing (discussion of Nielsen's book)

Usability heuristics

In 7 minutes, write down as many of the usability heuristics as you can remember

A large, empty rectangular box with a thin black border, intended for the student to write down as many usability heuristics as they can remember within a 7-minute time limit.

Nielsen's updated usability heuristics www.useit.com/papers/heuristic/heuristic_list.html

Visibility of system status

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

Match between system and the real world

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.

User control and freedom

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.

Consistency and standards

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

Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

Recognition rather than recall

Minimize the user's memory load by making 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.

Flexibility and efficiency of use

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.

Aesthetic and 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.

Help users recognize, diagnose, and recover from errors

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

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.

Site Usability Heuristics for the Web

<http://user-experience.org/uefiles/writings/heuristics.html>

Jakob Nielsen's 10 usability heuristics appear below, with his description in bold and my Web-specific comment following.

The overriding theme for applying these heuristics to the Web is to use links effectively.

1. Visibility of system status

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

Probably the two most important things that users need to know at your site are "*Where am I?*" and "*Where can I go next?*"

Make sure each page is branded and that you indicate which section it belongs to. Links to other pages should be clearly marked. Since users could be jumping to any part of your site from somewhere else, you need to include this status on every page.

My Site Stress Test is an evaluation focused on this heuristic because it is so important on the Web.

2. Match between system and the real world

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.

On the Web, you have to be aware that users will probably be coming from diverse backgrounds, so figuring out their "language" can be a challenge.

3. User control and freedom

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.

Many of the "emergency exits" are provided by the browser, but there is still plenty of room on your site to support user control and freedom. Or, there are many ways authors can take away user control that is built into the Web. A "home" button on every page is a simple way to let users feel in control of your site.

Be careful when forcing users into certain fonts, colors, screen widths or browser versions. And watch out for some of those "advanced technologies": usually user control is not added until the technology has matured. One example is animated GIFs. Until browsers let users stop and restart the animations, they can do more harm than good.

4. Consistency and standards

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

Within your site, use wording in your content and buttons consistently. One of the most common cases of inconsistent wording I see deals with links, page titles and page headers. Check the titles and headers for your pages against the links that point to them. Inconsistent wording here can confuse users who think they ended up in the wrong spot because the destination page had a title that differed vastly from the link that took them there.

"Platform conventions" on the Web means realizing *your site is not an island*. Users will be jumping onto (and off of) your site from others, so you need to fit in with the rest of the Web to some degree. Custom link colors is just one example where it may work well for your site but since it could conflict with the rest of the Web, it may make your site hard to use.

And "standards" on the Web means following HTML and other specifications. Deviations from the standards will be opportunities for unusable features to creep into your site.

5. Error prevention

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

Because of the limitations of HTML forms, inputting information on the Web is a common source of errors for users. Full-featured, GUI-style widgets are on their way; in the meanwhile you can use JavaScript to prevent some errors before users submit, but you still have to double-check after submission.

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.

For the Web, this heuristic is closely related to system status. If users can recognize where they are by looking at the current page, without having to recall their path from the home page, they are less likely to get lost.

Certainly the most invisible objects created on the Web are server-side image maps. Client-side image maps are a lot better, but it still takes very well-crafted images to help users recognize them as links.

Good labels and descriptive links are also crucial for recognition.

7. Flexibility and efficiency of use

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.

Some of the best accelerators are provided by the browser. Like bookmarks.

Make pages at you're your site easy to bookmark. If a user is only interested in one corner of your site, make it easy for him to get there. Better than have him get frustrated trying to get from your home page to what he is looking for.

Do not use frames in a way that prevent users from bookmarking effectively.

Support bookmarking by not generating temporary URLs that have a short life span. If every week you come out with a new feature article for your site, make sure your URL lives on, even after the content is taken down. *Web Review* uses long-term locations by putting date information into the URLs. Or, you could re-use your URLs for the newer content.

Consider using GET instead of POST on your forms. GET attaches the parameters to the URL, so users can bookmark the results of a search. When they come back, they get their query re-evaluated without having to type anything in again.

All of these rules for "design to be bookmarked" also help you *design to be linked to*. If the contents of your site can easily be linked to, others can create specialized views of your site for specific users and tasks. Amazon.com's associates program is just one example of the value of being easy to link to.

8. Aesthetic and 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.

Extraneous information on a page is a distraction and a slow-down. Make rarely needed information accessible via a link so that the details are there when needed but do not interfere much with the more relevant content.

The best way to help make sure you are not providing too much (or too little) information at once is to use *progressive levels of detail*. Put the more general information higher up in your hierarchy and let users drill down deeper if they want the details. Likewise, make sure there is a way to go "up" to get the bigger picture, in case users jump into the middle of your site.

Make sure your content is written for the Web and not just a repackaged brochure. Break information into chunks and use links to connect the relevant chunks so that you can support different uses of your content.

9. Help users recognize, diagnose, and recover from errors

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

Errors will happen, despite all your efforts to prevent them. Every error message should *offer a solution* (or a link to a solution) on the error page.

For example, if a user's search yields no hits, do not just tell him to broaden his search. Provide him with a link that will broaden his search for him.

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.

Some of the more basic sites will not need much documentation, if any. But as soon as you try any complicated tasks, you will need some help for those tasks.

For the Web, the key is to not just slap up some help pages, but to *integrate the documentation* into your site. There should be links from your main sections into specific help and vice versa. Help could even be fully integrated into each page so that users never feel like assistance is too far away.

Metaphors

Think about metaphors in the context of the type of software you are working on. Write down your thoughts in preparation for a discussion.

A large, empty rectangular box with a thin black border, intended for the student to write their thoughts on metaphors in software design.

Usability test plan elements (Nielsen 1994, p. 170 - 171)**Arrange these elements into groups** (sent by email)

- The goal of the test: What do you want to achieve?
- Where and when will the test take place?
- How long is each test session expected to take?
- What computer support will be needed for the test?
- What software needs to be ready for the test.
- What should the state of the system be at the start of the test? What should the system/network load and response times be? If possible, the system should not be unrealistically slow (see the discussion of prototyping in Section 4.8), but neither should it be unrealistically fast because it is run on a system or network with no other users. One may have to artificially slow down the system to simulate realistic response times.
- Who will serve as experimenters for the test?
- Who are the test users going to be, and how are you going to get hold of them?
- How many test users are needed?
- What test tasks will the users be asked to perform?
- What criteria will be used to determine when the users have finished each of the test tasks correctly?
- What user aids (manuals, online help, etc.) will be made available to the test users?
- To what extent will the experimenter be allowed to help the users during the test?
- What data is going to be collected, and how will it be analyzed once it has been collected?
- What will the criterion be for pronouncing the interface a success? Often, this will be the "planned" level for the previously specified usability goals (see page 80), but it could also be a looser criterion such as "no new usability problems found with severity higher than 3."

Typical quantifiable usability measurements include (Nielsen 1994, p. 194)

Arrange this list into groups.

- The time users take to complete a specific task.
- The number of tasks (or the proportion of a larger task) various kinds that can be completed within a given time limit.
- The ratio between successful interactions and errors.
- The time spent recovering from errors.
- The number of user errors.
- The number of immediately subsequent erroneous actions.
- The number of commands or other features that were utilized by the user (either the absolute number of commands issued or the number of different commands and features used).
- The number of commands or other features that were never used by the user.
- The number of system features the user can remember during a debriefing after the test.
- The frequency of use of the manuals and/or the help system, and the time spent using these system elements.
- How frequently the manual and/or help system solved the user's problem.
- The proportion of user statements during the test that were positive versus critical toward the system.
- The number of times the user expresses clear frustration (or clear joy)
- The proportion of users who say that they would prefer using the system over some specified competitor.
- The number of times the user had to work around an unsolvable problem.
- The proportion of users using efficient working strategies compared to the users who use inefficient strategies (in case there are multiple ways of performing the tasks).
- The amount of "dead" time when the user is not interacting with the system. The system can be instrumented to distinguish between two kinds of dead time: response-time delays where the user is waiting for the system, and thinking-time delays where the system is waiting for the user. These two kinds of dead time should obviously be approached in different ways.
- The number of times the user is sidetracked from focusing on the real task.

Usability methods (Nielsen 1994, p. 224)

<i>Method Name</i>	<i>Lifecycle Stage</i>	<i>Users Needed</i>	<i>Main Advantage</i>	<i>Main Disadvantage</i>
Heuristic evaluation	Early design, "inner cycle" of iterative design	None	Finds individual usability problems. Can address expert user issues.	Does not involve real users, so does not find "surprises" relating to their needs.
Performance measures	Competitive analysis, final testing	At least 10	Hard numbers. Results easy to compare.	Does not find individual usability problems.
Observation	Task analysis, follow-up studies	3 or more	Ecological validity; reveals users' real tasks. Suggests functions and features.	Appointments hard to set up. No experimenter control.
Logging actual use	Final testing, follow-up studies	At least 20	Finds highly used (or unused) features. Can run continuously.	Analysis programs needed for huge mass of data. Violation of users' privacy.
Thinking aloud	Iterative design, formative evaluation	3-5	Pinpoints user misconceptions. Cheap test.	Unnatural for users. Hard for expert users to verbalize.
Questionnaires	Task analysis, follow-up studies	At least 30	Finds subjective user preferences. Easy to repeat.	Pilot work needed (to prevent misunderstandings),
Interviews	Task analysis	5	Flexible, in-depth attitude and experience probing.	Time consuming. Hard to analyze and compare.
Focus groups	Task analysis, user involvement	6-9 per group	Spontaneous reactions and group dynamics.	Hard to analyze. Low validity
User feedback	Follow-up studies	Hundreds	Tracks changes in user requirements and views.	Special organization needed to handle replies.

Overview of usability methods

www.usabilitynet.org/tools/methods.htm

Very cool

Insert usabilitynet

Another list of methods from The Usability Methods Toolbox <http://jthom.best.vwh.net/usability/>
This site gives a practical description of each method

Inquiry

- . Contextual inquiry
- . Ethnographic study / field observation
- . Interviews and focus groups
- . Surveys
- . Questionnaires
- . Journaled sessions
- . Self-reporting logs
- . Screen snapshots

Inspection

- . Heuristic evaluation
- . Cognitive walkthroughs
- . Formal usability inspections
- . Pluralistic walkthroughs
- . Feature inspection
- . Consistency inspection
- . Standards inspection
- . Guideline checklists

Testing

- . General concepts
- . Thinking aloud protocol
- . Co-discovery method
- . Question asking protocol
- . Performance measurement
- eye-tracking

Related techniques

- . Prototyping
 - . low-fidelity prototyping
 - . high-fidelity prototyping
 - . horizontal prototyping
 - . vertical prototyping
- . Affinity diagrams
- . Archetype research
- . Blind voting
- . Card-sorting
- . Education evaluation
 - . Stake model
 - . Stufflebeam (CIPP) model
 - . Scriven model
- . action research

Examples of CAUSE tools (Nielsen 1994, p. 264 - 265)

- Prototyping tools to rapidly construct a mock-up user interface for user testing.
- Tools for interactive construction of screen layouts, dialog boxes, icons, etc., by direct manipulation.
- Tools for interactive manipulation and easier use of formal notations, specifications, models, and task analysis techniques in order to lower the barriers to their use [Johnson and Johnson 1990].
- Hypermedia representation of user interface standards and guidelines, allowing designers to view animated examples of interaction techniques and to jump between related issues.
- Design rationale representations.
- Wizard of Oz support tools that allow the human simulating the advanced interface to construct replies more easily and that constrain those replies according to the rules of the experiment.
- Logging tools for use during a user test. Such tools typically allow an experimenter to record the time of various user actions or events and to annotate them with a prespecified set of codes as well as free-form comments. Often, the time stamps in the log are coordinated with the timing codes on a videotape of the experiment, thus allowing a playback tool to retrieve the taped record of the test, starting a minute or so before each event that needs to be studied in more detail.⁴
- Localization and translation support tools for international user interfaces.
- Keystroke and event loggers, either for use in user testing or for instrumentation of installed systems.
- Databases of user complaints and support line calls, as well as analysis tools to extract more general information from the database.

Other questions on Nielsen

Lectures 3.3-3.4.

Usability testing of Web sites

For information from 708V, Lecture 9

Evaluation of Internet resources

“Evaluation of information found on the Web is in most respects no different from evaluating information anywhere. The difference is that a lot of information on the Web is not filtered, There are no filters between you and the Internet, so the principle *caveat lector* (reader beware) applies.

All information, whether in print or electronic, needs to be evaluated by readers for authority, appropriateness, and other personal criteria for value. If you find information that is “too good to be true”, it probably is. Never use information that you cannot verify. Establishing and learning criteria to filter information you find on the Internet is a good beginning for becoming a critical consumer of information in all forms. “Cast a cold eye” (as Yeats wrote) on everything you read.

Question it. Look for other sources that can authenticate or corroborate what you find. Learn to be skeptical and then learn to trust your instincts.”

(Adapted from Elizabeth E. Kirk, www.jessebethel.net/imc/kirk_evalinfo.html, © 1996 last modified 6/99, a good site with a tutorial on evaluation)

Outline

Purpose of evaluation

Type of source evaluated

Subject field

Evaluation criteria

Purpose of evaluation

- . Who will use the information
 - . . Own use of information
 - . . Recommendation to specific audience
- . Use as primary or secondary source
 - . . The source is to be used as a primary source for obtaining substantive data
 - . . . Validity of the information, verifying facts
 - . . . Use to be made of the information. Mission-critical?
 - . . The source is to be used as a secondary source for obtaining pointer data

Type of source evaluated

- . Type of document / website
 - . . Strictly Web document / website
 - . . . By type, e.g.,
 - e-commerce site
 - . . . "Traditional" document (such as a journal article) made available through the Web
 - . . Medium (Text, image, sound, multimedia)
- . Other characteristics

Subject field**Evaluation criteria**

- . Depend on purpose, type of source, and subject field
- . Where to get information needed to apply each criterion

Information filtering

- . Remarks on definition. Information filtering vs. information retrieval
- . Filter software
 - . . presently emphasizing the rejection of documents with content considered objectionable or unsuitable for a given audience
 - . . could be applied using other criteria that can be assessed automatically
- . Collaborative filtering

Basic criteria for evaluating all forms of information, and how they apply to the Internet

From Kirk with many additions from other sources.

One perspective: How much of the evaluation/judgment can be automated?

Coverage

- What topics are covered?
- What does this page offer that is not found elsewhere?
- What is its intrinsic value?
- How in-depth is the material?

Rationale: 1. Web coverage often differs from print coverage
2. Frequently, it's difficult to determine the extent of coverage
3. Sometimes web information is just-for-fun or outright silliness

- Are scope and limits clearly stated? Title informative? Does the content fit the scope?
- If there is advertising, what is its relevance and proportion to the rest of the site?

Organization. Style and Functionality

- Are headings clear and descriptive, or do they use jargon unknown to the average user?
- Is text well-written with acceptable grammar and spelling? What is the quality of multimedia files? Writing style is appropriate for the intended audience
- Is the content organized by the needs of the user, or does it reflect an internal hierarchy?
- Site is laid out clearly and logically with well organized subsections
- Links to remote sites all work
- Search capability is offered if the site is extensive

Authority: Authorship and sponsorship

- Are author and/or sponsor indicated? Is the page signed?
- If not, is there any other way to determine its origin?
Look for a header or footer showing affiliation.
Look at the URL, for example, <http://www.fbi.gov>
Look at the domain. .edu, .com, .ac.uk, .org, .net

Rationale 1. Anyone can publish anything on the Web
2. It's often hard to determine a web page's authorship
3. Even if a page is signed, qualifications aren't usually given
4. Sponsorship isn't usually indicated

Authorship

Authorship is perhaps the major criterion used in evaluating information. Who wrote this? When we look for information with some type of critical value, we want to know the basis of the authority with which the author speaks. Here are some possible filters:

- In your own field of study, the author is a well-known and well-regarded name you recognize.
 - When you find an author you do not recognize:
 - the author is mentioned in a positive fashion by another author or another person you trust as an authority;
 - you found or linked to the author's Web/Internet document from another document you trust;
 - the Web/Internet document you are reading gives biographical information, including the author's position, institutional affiliation and address;
 - biographical information is available by linking to another document; this enables you to judge whether the author's credentials allow him/her to speak with authority on a given topic;
 - if none of the above, there is an address and telephone number as well as an e-mail address for the author in order to request further information on his or her work and professional background. An e-mail address alone gives you no more information than you already have. Mail-to link
 - You may need to trace back in the URL (Internet address) to view a page in a higher directory with background information
 - Sponsor/location of the site is appropriate to the material as shown in the URL
 - ~NAME in URL may mean a personal home page with no official sanction
- For a different point of view: I have heard it proposed that URLs that contain a tilde (~) are not as authoritative as web sites with their own domain, but the Internet is proof that dedicated amateurs can create valuable resources, and that not everything with a "brand name" is reliable.
(Carolyn Caywood c1995. <http://www6.pilot.infi.net/~carolyn/criteria.html>)

Sponsorship. Publisher. Third-party endorsement

The publishing body also helps evaluate any kind of document you may be reading. In the print universe, this generally means that the author's manuscript has undergone screening in order to verify that it meets the standards or aims of the organization that serves as publisher. This may include peer review. On the Internet, ask the following questions to assess the role and authority of the “publisher”, which in this case means the server (computer) where the document lives:

- Is the name of any organization given on the document you are reading? Are there headers, footers, or a distinctive watermark that show the document to be part of an official academic or scholarly Web site? Can you contact the site Webmaster from this document?
- If not, can you link to a page where such information is listed? Can you tell that it's on the same server and in the same directory (by looking at the URL)?
- Is this organization recognized in the field in which you are studying?
- Is this organization suitable to address the topic at hand?
- Can you ascertain the relationship of the author and the publisher/server? Was the document that you are viewing prepared as part of the author's professional duties (and, by extension, within his/her area of expertise)? Or is the relationship of a casual or for-fee nature, telling you nothing about the author's credentials within an institution?
- Can you verify the identity of the server where the document resides? Internet programs such as `dnslookup` and `whois` will be of help.
- Does this Web page actually reside in an individual's personal Internet account, rather than being part of an official Web site? This type of information resource should be approached with the greatest caution. Hints on identifying personal pages are available in *Understanding and Decoding URLs*.
- Endorsement by third party
- Certification, certified by. For example, Verisign

Point of view or bias. Objectivity

- Is bias minimal?
- Is the page designed to sway opinion?
- Is there any advertising on the page?

Rationale: 1. Frequently the goals of the sponsors/authors aren't clearly stated.
2. Often the Web serves as a virtual "Hyde Park Corner", a soapbox.

- Are biases and opinions clearly identified? Is there any conflict of interest?

Point of view or bias reminds us that information is rarely neutral. Because data is used in selective ways to form information, it generally represents a point of view. Every writer wants to prove his point, and will use the data and information that assists him in doing so. When evaluating information found on the Internet, it is important to examine who is providing the "information" you are viewing, and what might be their point of view or bias. The popularity of the Internet makes it the perfect venue for commercial and socio-political publishing. These areas in particular are open to highly "interpretative" uses of data.

Steps for evaluating point of view are based on authorship or affiliation:

- First, note the URL of the document. Does this document reside on the Web server of an organization that has a clear stake in the issue at hand?

If you are looking at a corporate Web site, assume that the information on the corporation will present it in the most positive light.

If you are looking at products produced and sold by that corporation, remember: you are looking at an advertisement.

If you are reading about a political figure at the Web site of another political party, you are reading the opposition.

- Does this document reside on the Web server of an organization that has a political or philosophical agenda?

If you are looking for scientific information on human genetics, would you trust a political organization to provide it?

- Many areas of research and inquiry deal with controversial questions, and often the more controversial an issue is, the more interesting it is. When looking for information, it is always critical to remember that everyone has an opinion. Because the structure of the Internet allows for easy self publication, the variety of points of view and bias will be the widest possible.

Accuracy or verifiability of details

- Is the information reliable and error-free?
- Is there an editor or someone who verifies/checks the information?

Rationale: 1.Anyone can publish anything on the Web
2.Unlike traditional print resources, web resources rarely have editors or fact-checkers
3.Currently, no web standards exist to ensure accuracy

Accuracy or verifiability of details is an important part of the evaluation process, especially when you are reading the work of an unfamiliar author presented by an unfamiliar organization, or presented in a non-traditional way. Criteria for evaluating accuracy include:

- For a research document, the data that was gathered and an explanation of the research method(s) used to gather and interpret it are included.
- The methodology outlined in the document is appropriate to the topic and allows the study to be duplicated for purposes of verification.
- Is the origin of the content documented?

The document relies on other sources that are listed in a bibliography or includes links to the documents themselves.

The document names individuals and/or sources that provided non- published data used in the preparation of the study.

- The background information that was used can be verified for accuracy.

Currency

- Is the page dated? Last update indicated?
- How current are the links? Have some expired or moved?

Rationale: 1. Publication or revision dates not always provided.

2. If a date is provided, it may have various meanings. For example,

It may indicate when the material was first written

It may indicate when the material was first placed on the Web

It may indicate when the material was last revised

Currency refers to the timeliness of information. In printed documents, the date of publication is the first indicator of currency. For some types of information, currency is not an issue: authorship or place in the historical record is more important (e.g., T. S. Eliot's essays on tradition in literature). For many other types of data, however, currency is extremely important, as is the regularity with which the data is updated. Apply the following criteria to ascertain currency:

- The document includes the date(s) at which the information was gathered (e.g., US Census data).
- The document refers to clearly dated information (e.g., "Based on 1990 US Census data.").
- Where there is a need to add data or update it on a constant basis, the document includes information on the regularity of updates.
- The document includes a publication date or a "last updated" date.
- The document includes a date of copyright.
- If no date is given in an electronic document, you can view the directory in which it resides and read the date of latest modification.

Referral to and/or knowledge of the literature

Referral to and/or knowledge of the literature refers to the context in which the author situates his or her work. This reveals what the author knows about his or her discipline and its practices. This allows you to evaluate the author's scholarship or knowledge of trends in the area under discussion. The following criteria serve as a filter for all formats of information:

- The document includes a bibliography.
- The author alludes to or displays knowledge of related sources, with proper attribution.
- The author displays knowledge of theories, schools of thought, or techniques usually considered appropriate in the treatment of his or her subject.
- If the author is using a new theory or technique as a basis for research, he or she discusses the value and/or limitations of this new approach.
- If the author's treatment of the subject is controversial, he or she knows and acknowledges this.
- Links

Links are relevant and appropriate

Don't assume that the linked sites are the best available. Be sure to investigate additional sites on the topic

Are there reviews of the site? How many other sites link to this one?

Some additional lists are attached. Good evaluation Web sites mentioned in text chapter.

Additional reference:

Nielsen, Jacob

Designing Web usability.

Indianapolis, IN: New Riders Publishing; 2001. 419p.

ISBN 1-56205-810-X. \$45 (get cheaper through www.isbn.nu)

Web-specific criteria

From Carolyn Caywood c1995. <http://www6.pilot.infi.net/~carolyn/criteria.html> (rev.12/99)

Access

- Is the site WAP (Wireless Application Protocol) enabled
- Is it written in standard html, or have proprietary extensions been used?
- Does it use standard multimedia formats?
- Must you download software to use it?
- Do parts of it take too long to load?
- Is it usually possible to reach the site, or is it overloaded?
- Is it stable, or has the URL changed?
- Is the URL stated in the text of the web page?
- Does the site use the words the average person would try in a search engine?
- Is it open to everyone on the Internet, or do parts require membership and/or fees?
- If there is a charge, can the library pay it on a subscription basis for multiple access points?
- Are any rules for use stated up front?

Design

- Color
- Typeface
- Layout
- Are the individual Web pages concise, or do you have to scroll forever?
- Do essential instructions appear before links and interactive portions?
- Do all the parts work?
- Is using the site intuitive, or are parts likely to be misunderstood?
- Can you find your way around and easily locate a particular page from any other page?
- Is the structure stable, or do features disappear between visits?
- Can you "back" out of the site, or do you get stuck looping between pages?
- Does it look and feel friendly?
- Is it conceptually exciting? Does it do more than can be done with print?
- Can the user interact in satisfying ways?
- Are interactions secured if they involve private information?

- If Java or ActiveX, extensions like frames, or plug-ins are employed, do they actually improve the site? How do they affect users with older browsers?
- Is the amount of bandwidth commensurate with the usefulness of the content, or are graphics or multimedia included simply to show off?
- Site is easy to navigate, including
Clearly labeled Back, Home, Go To Top icons/links
Internal indexing links on lengthy pages

Miscellaneous

- Is it a dead end, or are there relevant and useful links to take you further?
Are the content and the links clearly described and suitable to the expected audience? *
- Is the content amplified over time, or is one visit all it's worth?
- Is there community interest in the content or the site itself?

Insert slide presentation 1 XXX

From http://stats.bls.gov/ore/htm_papers/st960160.htm

Sidebar #7: Usability Principles (Heuristics)

1. **Speak the users' language.** Use words, phrases, and concepts familiar to the user. Present information in a natural and logical order.
2. **Be Consistent.** Indicate similar concepts through identical terminology and graphics. Adhere to uniform conventions for layout, formatting, typefaces, labeling, etc.
3. **Minimize the users' memory load.** Take advantage of recognition rather than recall. Do not force users to remember key information across documents.
4. **Build flexible and efficient systems.** Accommodate a range of user sophistication and diverse user goals. Provide instructions where useful. Lay out screens so that frequently accessed information is easily found.
5. **Design aesthetic and minimalist systems.** Create visually pleasing displays. Eliminate information which is irrelevant or distracting.
6. **Use chunking.** Write material so that documents are short and contain exactly one topic. Do not force the user to access multiple documents to complete a single thought.
7. **Provide progressive levels of detail.** Organize information hierarchically, with more general information appearing before more specific detail. Encourage the user to delve as deeply as needed, but to stop whenever sufficient information has been received.
8. **Give navigational feedback.** Facilitate jumping between related topics. Allow the user to determine her/his current position in the document structure. Make it easy to return to an initial state.
9. **Don't lie to the user.** Eliminate erroneous or misleading links. Do not refer to missing information.

Insert first presentation

<http://www.gotomedia.com/seattle00/usability>

Overview

"Ease of Use"

The understanding of how an individual user navigates, Finds information and interacts with your Web site. It is an one-on-one 'watch and learn' approach.

What is Usability Testing?

Overview

One to two day process with 4-8 participants
1-hour approx. per session
Pre-determined tasks performed
Test facilitator watches & takes notes
Sometimes video taped
Sometimes watched by development team
Results show what is working and not working

"Two distinct advantages to formal testing: it provides feedback for site development, and it is an educational opportunity for designers and engineers, allowing them to see firsthand the decision-making process of an average site visitor"

Formal vs. Informal Testing

Formal Testing

Formal facility
Moderator is an expert in human factors engineering
Pre-screened applicants
Formal test plan
Complex video monitoring practices

Informal Testing

Participant's work environment
Friends, co-workers or family members
Impartial monitor
Informal test plan
No video taping, only notes and observations

Cost Comparisons

- \$0 Informal testing, with friends/co-workers
- \$300 - \$500 Informal Testing, 2-days, outside participants
- \$1000 Semi-formal testing, outside company finds & pre-screens participants
- \$2,000 - \$10,000 Formal, professional testing with experts laboratory or formal testing environment
Human factors specialists, impartial and consistent. Trained to plan, set up, conduct, analyze and report results

Methods of Testing

Comparison of Methods

Usability Testing	Focus Groups	Online/Email Surveys	Remote Testing
4 - 8 participants assessment/verification	8 - 20 participants early stages of dvlpmnt	50 - 1000 participants used any time in cycle	4 - 8 participants assessment/verification
task/action oriented	representative users	broad-base of users	task/action oriented
one-on-one interaction	facilitator - group interaction	no direct interaction	remote interaction by phone and video
results based on observation	opinions and feedback	statistically significant feedback	variable results

Have Tangible Goals

- Don't try and test your entire site at once
- Identify problem areas & solution goals
- Develop specific tasks to fit goals

Use the Appropriate Method

Usability Testing

- Collects first hand data through observation
- Used throughout development process
- Task oriented, non-disputable result

Focus Groups

- Collects opinions and feedback
- Used early in the development cycle
- Representative user and market
- Good for overall layout, information design and navigation schemes

Surveys

- Used throughout process
- statistically significant feedback
- Conducted by phone, mail or email

Why Test?

Identify Problem Areas Early

Usability testing helps to identify problem areas within the site:

Settle disputes with data instead of guesses

Provide real feedback with actual users

Low-cost for valuable results

What is obvious to you might not be obvious to the user

Minimize risk prior to launch

The Earlier the Better

Conducting tests earlier in the process is cheaper and easier because:

Able to identify problems and make changes early in development process

Testing wireframes or paper prototypes early in process doesn't affect time or budget

Testing an HTML protosite is helps identify structural & navigational issues

User Centered Design

Overview

1. UCD is the practice of designing around the user identify specific usability needs and requirements
2. Referred to as human factors engineering or ergonomics
3. Focus on making the design fit the user, instead of user fitting design

Usability Goals & Objectives

Usefulness. Does it meet the goals of the user?

Effectiveness. Did the user accomplish the task?

Learnability. Is the site easier to navigate the second time around?

Attitude. Does the user like the site?

(Adapted from Jeffrey Rubin's Handbook of Usability Testing)

An Iterative Approach

"An iterative design and testing approach also allows one to make a steady and rapid progress on a project, to learn through empirical evidence, and to 'shape' the product to fit the end users' abilities, expectations and aptitude. - Jeffrey Rubin

Usability Testing Categories

Exploratory Test:

Early in development cycle
 Functional questions debated
 Evaluate effectiveness of specific area

Assessment Test:

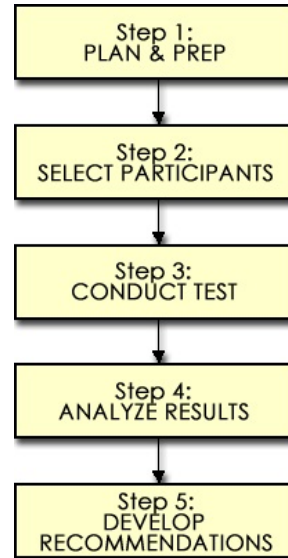
Easiest to perform
 Typical test for development cycle
 Early or midway into development cycle

Validation Test:

Late in development cycle
 Determine performance or usability
 Validation prior to release

Testing Overview

5-Step Process



Step1. Plan and prepare

Goals and Objectives

Have specific goals or objectives in mind
 Develop test plan (purpose, problem, profile)
 Create task list
 Test on co-worker or friend
 Keep it simple
 Plan on testing in user's own environment

Develop a Test Plan

Keep it Simple!

A simple test plan, a list of questions and good attitude are all you need in preparation. Test plans will vary depending on type of test and formality desired.

Basic Plan

A basic plan will contain the following:

1. Overall Goals/Objectives
2. Methodology
3. Target Profile
4. Testing Outline
5. Specific Task List
6. Final Evaluation/Reporting Structure

Create framework for your testing process.

Allows for communication of goals and objectives.

Does not need to be complex.

You will not be able to test everything.

Step 2. Select participants**Where to Find Participants**

Posting on job sites
Posting in college
Ads in Newspaper under "Internet"
Co-workers/Building co-workers
Friends & Relatives

Gather User Profiles

Sample User Grid (page attached)
Sample User Profile (page attached)

Create a Task List

Create list of tasks or questions which a general user can (hopefully) finish in about an hour.

Assign basic time frames to each task, which will vary from user to user.

Make sure tasks are not too simple, nor too difficult.

Pre-test with friends or co-workers to confirm.

Step3. Conduct test

Sample Test Script

Create an 'orientation script' or bullet list of items to mention prior to start of testing:

Greet participant

Give overview of session & goals

Participant is not being evaluated in any way

Don't be concerned with results

You will be watching & taking notes, but will not be interacting or answering questions

Afterwards participant will talk about the experience & asked to fill out a brief survey.

Moderator Tips

Tips for the moderator when conducting the session:

Keep neutral during testing

Don't give any hints

Don't rescue the individual

Respond if necessary in non-specific manner

Read level of frustration, 'fail task' only if all other options are taken

Conduct Test

What Do You Need?

Checkoff List

NDA ready for signing

Batteries charged for video camera (along with camera and tapes)

Printout of site for easier note taking

List of sequenced, predetermined tasks

Survey form ready for post-testing

Cash payment in envelope

Orientation Dos

Introduce yourself, explain why you are here

Explain your role as a silent observer

Explain you are not testing and there are no wrong answers

Explain participant can think aloud if desired

Keep session relaxed, use humor when appropriate

Stay impartial

Collect Basic Data

Could they complete the task?

Did they need help?

How much time did it take? (track time through testing)

Stumbling blocks (problems/obstacles)

Overall observations, commentary

Post Test Survey (sample attached)

Prepare survey on paper for user to fill in

Ask basic questions about how user felt overall about site – graphics, logic, copy, navigation, etc.

Rate questions on a scale of 1–10

Gather data about overall effectiveness of site

Step 4. Analyze results

Compile and Summarize Data

Gather data as you go

Transfer handwritten notes to computer

Write reports while test is still fresh in your mind

Summarize data in grid format showing results, problem areas and comments

Identify problem areas

Identify global or broad-scope issues

Identify specific task-oriented issues

Step 5. Develop recommendations

Compile and Recommend

Take all information and translate into recommendations.

Validate thoughts you had prior to testing, put it into action.

Concentrate on high-level functionality first (global nav, text links vs. graphic links, layout, etc.)

Focus on specific areas and recommendations for improved user experience.

Divide recommendations into short and long term goals.

Determine implementation plan, staged or phased tasks.

Final Report (sample page attached)

Executive Summary

Brief synopsis of major findings & recommendations

Methodology

Testing process, methods of obtaining data, user profiles, etc.

Results

Survey feedback, detailed session notes

Findings & Recommendations

General & specific information, short and long term changes

Appendices

Raw data, notes, samples of test material, background data, etc.

Please feel free to email me at: kelly@gotomedia.com

This presentation is available at: www.gotomedia.com/iceland/usability .

Links

WEB RESOURCES

WEB STYLE GUIDE

Basic Design Principles for Creating Web Sites

<http://info.med.yale.edu/caim/manual/contents.html>

SUN ON THE NET

Guide to Web Style

<http://www.sun.com/styleguide/>

USER TEST YOUR WEB SITE:

AN INTRODUCTION TO USABILITY TESTING

By Keith Instone

<http://www.instone.org/howtotest/introduction.html>

CONDUCTING YOUR FIRST USER TEST

By Keith Instone

<http://webreview.com/wr/pub/97/05/30/usability/index.html>

USER TESTING: HOW TO FIND OUT WHAT USERS WANT

By Jennifer Flemming

<http://www.ahref.com/guides/design/199806/0615jef.html>

BOOKS

HANDBOOK OF USABILITY TESTING

How to Plan, Design and Conduct Effective Tests

Jeffrey Rubin

WEB SITE USABILITY

A Designer's Guide

Jared M. Spool, et al

From: <http://www.acm.org/~perlman/question.html>

Web-Based User Interface Evaluation with Questionnaires

Gary Perlman <http://www.acm.org/~perlman/> perlman@acm.org

Introduction

Questionnaires have long been used to evaluate user interfaces (Root & Draper, 1983). For a handful of questionnaires specifically designed to assess aspects of usability, the validity and/or reliability have been established, including some in the following table.

Acronym	Instrument	Reference	Institution	Example
QUIS	Questionnaire for User Interface Satisfaction	Chin <i>et al</i> , 1988	Maryland	27 questions
PUEU	Perceived Usefulness and Ease of Use	Davis, 1989	IBM	12 questions
NAU	Nielsen's Attributes of Usability	Nielsen, 1993	Bellcore	5 attributes
NHE	Nielsen's Heuristic Evaluation	Nielsen, 1993	Bellcore	10 heuristics
CSUQ	Computer System Usability Questionnaire	Lewis, 1995	IBM	19 questions
ASQ	After Scenario Questionnaire	Lewis, 1995	IBM	3 questions
PHUE	Practical Heuristics for Usability Evaluation	Perlman, 1997	OSU	13 heuristics
PUTQ	Purdue Usability Testing Questionnaire	Lin <i>et al</i> , 1997	Purdue	100 questions

There are other questionnaires, including instruments from the HFRG (Human Factors Research Group):

SUMI: Software Usability Measurement Inventory

MUMMS: Measurement of Usability of Multi Media Software

WAMMI: Website Analysis and Measurement Inventory

FAQ: Frequently Asked Questions about Questionnaires in Usability Engineering (compiled by Jurek Kirakowski)

Also, see LaLomia & Sidowski, 1990, 1991, 1993 for reviews of measurements of satisfaction, literacy, aptitude, attitude, and anxiety.

NetRaker (netraker.com) provides online usability evaluation, with a free trial version.

Note: Although these instruments are online, many are copyrighted and some require a fee for non-educational use. See the references at the end of each form for more information.

<http://www.nomos.se/wammi/advdemo/graph.htm>

WAMMI Scales:

Attractiveness

An Attractive site is visually pleasant, and also offers much of direct interest to the intended users, whether it be functionality or information.

Controllability

If a site scores well on Controllability the users most probably feel they can navigate around it with ease. Poor usually means a poorly organized site.

Efficiency

When users give a high Efficiency rating they feel they can quickly locate what is of interest to them and they feel that the web site responds (possibly, the pages load) at a reasonable speed.

Helpfulness

A site which is high on Helpfulness corresponds with the users' expectations about its content and structure. A site low on Helpfulness can be misleading about its content.

Learnability

When Learnability is high, users feel they are able to start using the site with the minimum of introductions. Everything is easy to understand from the start. When Learnability is low, users feel that the site may be using concepts or terminology which are unfamiliar. More explanations are needed.

Global Usability

Global Usability centres round the concepts that a site must make it easy for users to access what they need or want from the site, that there is a good, understandable level of organisation, and that the site 'speaks the users language'.

From <http://hci.stanford.edu/cs377/schiano/SampleSurveyPilot.pdf>

Web Usability Questionnaire for Stanford Library System: Socrates

Using the scale provided, please indicate your experience with Stanford Web Library System "Socrates". How much do you agree or disagree with the following statements? For items that are not applicable, please check "Not Applicable."

Strongly Disagree Disagree Undecided Agree Strongly Agree Not Applicable

1. I feel in control when I use Socrates.
2. The organization of Socrates is inconsistent.
3. The interface of Socrates is pleasant.
4. I feel the colors and font sizes used on Socrates is easy to read.
5. It is easy to find explanations of how to use Socrates.
6. The speed of page display in Socrates is slow.
7. I am able to efficiently complete my work using Socrates.
8. I feel the speed with which I can accomplish my tasks in Socrates (e.g. searching for books, renew books) to be acceptable.
9. I am satisfied with the effectiveness of search results in Socrates.
10. The information provided in Socrates (e.g. Search Tips, Tutorial and User Guides, and FAQs.) is effective in helping me complete the tasks.
11. Socrates gives error messages that clearly tell me how to solve the problems.
12. Socrates has all the functions and capabilities I expect it to have.
13. Overall, it is easy to learn how to use Socrates.
14. Overall, I am satisfied with Socrates.

Background information

15. Age: 1) ___ < 20 2) ___ 20-30 3) ___ 30-40 4) ___ 40-50 5) ___ >50
16. Gender: 1) ___ Male 2) ___ Female
17. What is your major? _____
18. Which search choice do you most often use?
1) ___ Basic Search 2) ___ Browse Search 3) ___ Expert Search 4) ___ Reserves Catalog
19. How often do you use Socrates?
1) ___ Several times a day 2) ___ Once a day 3) ___ Several times a week 4) ___ Once a week
5) ___ Several times a week 6) ___ Once a month 7) ___ Less than once a month
20. What do you like most about Socrates?

21. What do you dislike most about Socrates?

22. Please add any additional comments you wish to make about Socrates:

Usability Tools: A Useful Start XXX check for newer

<http://www.webtechniques.com/archives/2000/08/stratrevu/>

WebSAT

Lift Online and Lift Onsite

Evaluate the HTML code of Web pages based on a set of usability guidelines

Max performs a statistical evaluation to generate a site rating

NetRaker Suite remotely surveys users to gather usability feedback.

Andrew Chak

Manager of user experience design at Derivion, responsible for the usability of electronic bill presentment and payment products. andrewchak@hotmail.com.

Your Web site's usability drives its success or failure. Usability ensures that visitors can understand how to interact with your site. If it's easier to buy something or find information at another site, why should users bother with yours?

The best way to improve your site's usability is to observe your users interacting with it and then incorporate their feedback into your design. However, if you're inexperienced with facilitating usability tests or lack the resources to conduct a test, consider employing some automated usability tools that provide a head start in designing a more usable site.

Keep in mind that these tools are only a starting point. Using them is like using Microsoft Word to check your grammar—they're good for highlighting potential problems, but you'll need to evaluate each identified issue for its appropriateness. The tools are good for checking routine site-design elements for consistency, and encouraging good design practices. Use them to complement directly observed usability tests.

WebSAT

National Institute of Standards and Technology (NIST)

zing.ncsl.nist.gov/webmet/sat/websat-process.html

Cost: Free.

Web Static Analyzer Tool (WebSAT) is one of the tools included in the WebMetrics suite developed by the National Institute of Standards and Technology (NIST). This tool highlights potential problems by checking the HTML of individual pages against a set of usability guidelines. WebSAT reviews your HTML to see if it's implemented in a usable manner. WebSAT is free, and can either be used online or downloaded to your PC or UNIX box to test intranet pages behind a firewall.

With WebSAT, you can enter up to five individual URLs to be checked against its usability guidelines. One guideline for navigation is that the user should always be able to get to another page. In the associated HTML check, WebSAT will look for at least one link on each page. After WebSAT's evaluation, it provides a report of problems found on each page entered.

WebSAT identifies potential usability problems according to six categories: accessibility, form use, performance, maintainability, navigation, and readability. The online site presents an extensive explanation of the usability guidelines for each of these categories, and links to online resources.

Accessibility—ensures that the page makes proper use of tags for visually impaired users.

Form Use—primarily checks for the presence of form Submit and Reset buttons. Deploying a Reset button may be unwise because it lets users erase their input.

Performance—analyzes the size and coding of graphics in relation to page download speeds.

Maintainability—checks for tags and coding information that would make this page easier to port to another server.

Navigation—does only a basic check on how links are coded, and does not provide any feedback on the site navigation and labeling effectiveness.

Readability—provides an interesting evaluation in terms of the density of links versus content.

In summary, WebSAT is useful for ensuring that your HTML follows good coding practices and for learning some basic usability guidelines. The reports could be improved by providing specific recommendations rather than just identifying problems. WebSAT's main limitation is that it can check only individual pages, which prevents you from evaluating pages in the context of an overall site, but its free pricing can't be beat.

Lift Online and Lift Onsite

UsableNet.com www.usablenet.com

Cost: Lift Online: \$1 per page with a maximum of \$50 for a report of your site. (Lift Online Pro will be available September 1, 2000.)

Lift Onsite: Yearly subscription will be \$199 with a special introductory price of \$149 before September 1, 2000.

UsableNet.com offers a service called Lift that analyzes your Web pages for potential usability problems. Lift is available as an online service called Lift Online Pro, and as a subscription-based software tool called Lift Onsite. Lift Online Pro will be available only after September 1, but I looked at a demo version called Lift Online for this review.

Lift Online performs HTML checks derived from usability principles, much as WebSAT does. The demo version lets you evaluate one page at a time and provides a detailed report on the page's usability issues. The report provides an overall rating and a prioritized list of potential problems.

Lift Online goes beyond WebSAT by providing specific code change recommendations. For example, if a page is deemed to be too heavy for a quick download, the report provides a breakdown of the page download size according to its basic size (the HTML code) and the image download size. Further detail lists all of the graphics on the page along with their file sizes. The report is made complete by providing the usability rationale behind each of the recommended changes. This detail helps you make quicker design changes.

An important feature of Lift Online Pro will be evaluation of complete sites. (I am basing this statement on what's described on UsableNet's site.) My impression is that you would enter the URL of a site's home page and that it would automatically spider through the site to check all subpages. You use Lift Online Pro to check for consistent implementation of items like background colors and images, font usage, and navigational elements.

Lift Onsite is a subscription-based software tool that you can run from your computer. The subscription lets you update the software as Usablenet.com develops additional usability evaluation rules. Lift Onsite provides the very compelling feature of directly fixing the HTML problems as you review them in the evaluation report. This is extremely convenient for making immediate improvements. Lift Online provides thorough recommendations for improving a site's pages. However, I prefer Lift Onsite to Lift Online because it lets you make improvements to your site's HTML as you're going through the evaluation results.

Max WebCriteria www.webcriteria.com

Cost: Contact for pricing.

WebCriteria uses an intelligent browsing agent named Max that crawls through your site to gather vital statistics and rate your site's usability. Max uses a statistical model to simulate a user's experience to arrive at ratings in three areas: accessibility, load times, and content.

Max's accessibility evaluation measures how quickly a user can find something on the site. The rating is given in terms of the estimated time it takes for users to find what they're searching for.

The first factor in calculating accessibility is the link scan time, which is based on the premise that links placed higher up on a page are more accessible. This calculation is contentious; links at the top aren't necessarily highly visible or accessible. Many users may have learned to scan over the tops of Web pages to ignore navigational items and banner ads to focus on the page's content.

The second factor in measuring accessibility is the browse path, which averages the shortest number of clicks between the starting page to any given page, illustrating how deep a user must drill down to access a page.

The calculated access time should be used only as an initial benchmark, because it doesn't take into account the use or effectiveness of the site's search engine. The analysis assumes that all links are equally seen and understood by users, which isn't always the case.

Max does a good job of reporting the average time it takes your site's pages to load. The report provides a useful comparison of download times at office versus home speeds. The report also provides a distribution of pages in terms of their download times. A nice feature of the report is that you can download it into a spreadsheet for further analysis.

In the third evaluation area—content—Max summarizes the percentage of different media elements (text, graphics, multimedia) and client-side technologies used (Flash, PDF, and so on) that comprise the site. The main use of this component is to compare your site against your competitors' for an appropriate balance of media elements and client-side technologies.

If you need only a quick numerical analysis to grade your site, Max is for you. This service is useful for evaluating your site at regular intervals so that you can track its performance over time. Max's main strength is in providing you with a performance benchmark, but it doesn't provide many suggestions for design changes.

NetRaker Suite

NetRaker Corp. www.netraker.com

Cost: Monthly subscription; contact for pricing.

NetRaker Suite comprises online tools that help identify usability problems and conduct market research. NetRaker lets you create online surveys that gather feedback as users interact with your site. Essentially, NetRaker provides you with a limited version of a remote usability test. The main advantage of this approach is that you can collect feedback in the context of your site's intended purpose as opposed to relying on generic HTML checks or statistical evaluations.

NetRaker provides a customizable set of usability survey templates and comprehensive guidelines to help you compose objective survey questions. The questions are randomly made available to your site's users, who are given the option to participate. The survey asks users to perform tasks on the site and then provide feedback on how easy the tasks were. A graphical summary is provided as users contribute to the results.

NetRaker also lets you conduct market research with similar surveys to gain an understanding of your users and their expectations. The SiteRaker download performance tool monitors your site's download speeds against those of your competitors, and provides great added value by monitoring site performance across nine major metropolitan areas, with global tests available in the next few months. With NetRaker, you can survey a number of users quickly to gather usability feedback. The automated process ensures that users are surveyed consistently. You can reuse the same survey to evaluate the site after significant design changes to see if it's improving.

NetRaker is definitely one of the best tools to identify usability issues, because it's based on direct user feedback. Of course, the pop-up survey might be a minor irritation, and even more so if the user has a slow modem connection. NetRaker should incorporate an option that would prevent the pop-up survey from appearing on a regular basis: Every time I go back to its site, the same pop up appears, which eventually becomes distracting.

A Final Word

Each of these usability tools can highlight potential problem areas within your site. WebSAT and UsableNet.com's Lift services provide usability-based HTML validations to ensure good coding practices. WebCriteria's Max provides numerical analysis that's useful for tracking performance with quantifiable metrics. NetRaker is definitely my preferred choice, providing direct user feedback that can be used to derive site design improvements.

As I mentioned earlier, these tools are best used as indicators of your site's usability, and should be complemented by directly observed usability tests. (Several usability-related sites are listed in "Online".) Once you've made your design changes, you should test your site again to make sure that each change is actually an improvement. As you continue refining your site based on user feedback, I'm sure that you'll find more of your users clicking to your site's success.

Empirically Validated Web Page Design Metrics

Melody Y. Ivory EECS Department UC Berkeley Berkeley, CA 94720-1776 ivory@cs.berkeley.edu	Rashmi R. Sinha SIMS/Psychology Department UC Berkeley Berkeley, CA 94720-5050 sinha@sims.berkeley.edu	Marti A. Hearst SIMS UC Berkeley Berkeley, CA 94720-4600 hearst@sims.berkeley.edu
--	---	--

Appearing in ACM *SIGCHI'01*, March 31-April 4, 2001, Seattle, WA, USA. Copyright 2001 ACM 1-58113-327-8/01/0003.

Abstract:

A quantitative analysis of a large collection of expert-rated web sites reveals that page-level metrics can accurately predict if a site will be highly rated. The analysis also provides empirical evidence that important metrics, including page composition, page formatting, and overall page characteristics, differ among web site categories such as education, community, living, and finance. These results provide an empirical foundation for web site design guidelines and also suggest which metrics can be most important for evaluation via user studies.

METHODOLOGY

This study computes quantitative web page attributes (e.g., number of fonts, images, and words) from web pages evaluated for the 2000 Webby Awards [20]. The Webby organizers place web sites into 27 categories, including news, personal, finance, services, sports, fashion, and travel. A panel of over 100 judges from The International Academy of Digital Arts & Sciences use a rigorous evaluation process to select winning sites.¹ Webby organizers describe the judge selection criteria as follows: "Site Reviewers are Internet professionals who work with and on the Internet. They have clearly demonstrable familiarity with the category in which they review and have been individually required to produce evidence of such expertise. The site reviewers are given different sites in their category for review and they are all prohibited from reviewing any site with which they have any personal or professional affiliation. The Academy regularly inspects the work of each reviewer for fairness and accuracy."

Judges rate web sites based on six criteria: content, structure & navigation, visual design, functionality, interactivity, and overall experience. Figure 1 shows the distribution of the overall criterion across all of the judged sites. We suspected that the six criteria were highly correlated, suggesting that there was one factor underlying them all. To test this hypothesis, we used a principles component analysis to examine the underlying factor structure. The first factor accounted for 91% of the variance in the six criteria. In the experiments reported below, we used both the overall Webby score and the extracted factor for doing discriminant classification.

For our study, we selected sites from six topical categories - financial, educational, community, health, service, and living - because these categories contained at least 100 information-centric sites (in which the primary goal is to convey information about some topic). We used the overall score to define two groups of sites for analysis: good (top 33% of sites), and not-good (remaining 67% of sites). Specifically, we wanted to determine if there are significant

differences between the groups - both overall and within each category. Furthermore, we wanted to construct models for predicting group membership. These models would enable us to establish concrete thresholds for each metric, evaluate them with user studies, and eventually provide guidance for design improvement. We also used the composite rating to group sites into two categories: top 33% of sites, and bottom 33% of sites. The cutoffs for both sets, based on the overall criterion (ranging from 1 to 10) are:

	Overall	Community	Education	Finance
Top	6.97	6.58	6.47	6.6
Bottom	5.47	5.66	5.38	5.8
	Health	Living	Services	
Top	7.9	6.66	7.54	
Bottom	6.4	5.66	5.9	

The following section introduces the metrics and describes the data collected for this analysis.

Web Page Metrics

Table: Web page metrics computed for this study.

Metric	Description
Word Count	Total words on a page
Body Text %	Percentage of words that are body vs. display text (i.e., headers)
Emphasized Body Text %	Portion of body text that is emphasized (e.g., bold, capitalized or near !')
Text Positioning Count	Changes in text position from flush left
Text Cluster Count	Text areas highlighted with color, bordered regions, rules or lists
Link Count	Total links on a page
Page Size	Total bytes for the page as well as elements graphics and stylesheets
Graphic %	Percentage of page bytes that are for graphics
Graphics Count	Total graphics on a page (not including graphics specified in scripts, applets and objects)
Color Count	Total colors employed
Font Count	Total fonts employed (i.e., face + size + bold + italic)

From a list of 42 web page attributes associated with effective design and usability [13], we developed an automated tool to compute the 11 metrics that we focus on in this study (see Table 1). (This subset was chosen primarily because it was the easiest to compute; we are in the process of extending the tool to compute a wider range of metrics.) The tool functions similarly to the

Netscape Navigator browser in processing web pages and cascading stylesheets; it has limited support for inline frames, but does not support framesets, applets, scripts or other embedded objects. We analyzed the accuracy of the computed metrics using a set of 5 pages with widely different features, such as use of stylesheets, style tags, and forms. Overall, the metrics are about 85% accurate with text cluster and text positioning counts range from 38% to 74% accuracy.

Data Collection

We used the metrics tool to collect data for 1,898 pages from the six Webby Awards categories. These pages are from 163 sites and from 3 different levels in the site - the home page, pages directly accessible from the home page (level 1), and pages accessible from level 1 but not directly accessible from the home page (level 2). We attempted to capture 15 level 1 pages and 45 level 2 pages from each site. Because not every website has many pages at each level, our collection consists of an average of 11 pages per site.

Some good usability (including Web usability) Web sites XXX check against earlier list

<http://elearningcenter.univie.ac.at/index.php?id=148>

www.usabilityfirst.com/

A very nice introductory site

www.eye-square.com/english/usability-en.htm?gclid=CIOcz436xYUCFTZfQodQHNMpg

High-level overview by a usability testing company.

Very nicely done

The Usability Methods Toolbox

<http://jthom.best.vwh.net/usability/>

This is a very nice site, dated 1998

<http://usableweb.com/>

Next update: Never. By Keith Instone. Usable Web is a stale collection of links about information architecture, human factors, user interface issues, and usable design specific to the World Wide Web

1995 - 2004 (basically to 2001)

<http://usableweb.com/about/> gives more Web sites, the best ones included here

www.usabilitynet.org/home.htm

A very comprehensive site funded by the European Union

www.usabilitynet.org/tools/methods.htm

Very cool

User interface guidelines

www.hcibib.org/hci-sites/GUIDELINES.html

Ivory, Melody Y.

"Automated Web Site Evaluation"

Kluwer Academic Publishers. 2003

<http://webtango.berkeley.edu/papers/ue-survey/p470-ivory.pdf>

Web Accessibility Initiative (WAI)

"W3C Web Content Accessibility Guidelines"

www.w3.org/TR/2004/WD-WCAG20-20040311/

www.useit.com

Lecture 4.1.

Software in education and training

Write down some criteria for software that trains for using a software package such as MS Access or Corel Draw

A large, empty rectangular box with a thin black border, intended for the student to write down their criteria for software training. The box occupies most of the lower half of the page.

Environment of use

- . Schools (from preschool to University)
- . Home
- . Any kind of organization - training uses

Types of educational applications (at all levels of education)

Many programs combine features from several types.

- . Programming - programming languages for use by students
 - . Programming and problem solving.
 - . Top-down and bottom up approaches to writing programs as a model for problem solving.
 - . Programming as a way to gain insights into a topic. Examples:
 - . Understanding parts of speech through writing a sentence building program.
 - . Writing a program to do Latin conjugation.
 - . See attached article Reading and writing in the 21st century.
- . Computer-supported instruction - software to support student's own work
 - . In general thinking and information organization skills
 - . Concept mapping software
 - . Creativity support software
 - . Games, puzzles
 - . In mathematics:
 - . Software to do symbolic computation, such as solving systems of multiple equations.
 - . Software to graph functions.

- . . In Science:
 - . . . Software to support scientific calculations.
 - . . . Software to support data reporting and analysis (DBMS, spreadsheet, statistical packages).
 - . . . Computerized instrumentation.
 - . . . Simulation of scientific processes.
- . . In social studies and history:
 - . . . Use existing databases to find data, create own databases, software for data analysis.
 - . . . Hypermedia systems. Electronic encyclopedias.
 - . . . Simulation of social processes (including economics)
- . . In language arts:
 - . . . Word processing
 - . . . Software to guide writing
 - . . . Dictionary / thesaurus
- . . In art: Drawing and painting software
- . . In music: Composing programs
- . Networking to support information gathering and communication among students and between students and teachers (including distance learning)
- . Computer-assisted instruction - software to transmit instructional content
- . Educational games
- . Drill and practice
 - . . Mathematics practice
 - . . Vocabulary drill
 - . . Typing
- . Administering tests, student assessment
- . Managing instruction

Special criteria for selecting educational and training software

- . Educational objectives supported
- . Fits into curriculum / educational approach
 - . . Replaces or complements other means of delivery
- . Transferability of things learned
- . Several difficulty levels
- . Difficulty level adjusted by program as student progresses
- . Feedback to student how well he or she is doing. See guidelines for error messages. Must give constructive criticism
- . Good error handling particularly important. Do not discourage student.
- . Involvement of student, learner control
- . Program includes tests to assess level of knowledge and ability of the student both at the beginning and during use and adjusts material and difficulty level accordingly
- . Program keeps records for several students. How many?
- . Content: Accuracy, proper sequence, good examples
- . Presentation: Capabilities of computer, incl. multimedia, fully used?
- . Can the content and other aspects of the program be modified by the teacher?
- . Pace controlled by student
- . Free of stereotypes of all kinds
- . Motivation
 - . . Intrinsic motivation (example: mathematical problem must be solved to know how to proceed in a game)
 - . . Extrinsic motivation (example: nice picture or sound upon completion of a mathematical problem)
- . Appeals / relates to experience of learner

- . Stimulates students' own ideas - supports students' work, lets student explore phenomena or historical data
- . Suitability for group work
- . Need for instruction / supervision in using the program
- . Documentation for teacher, explaining
 - . objectives
 - . prerequisites
 - . instructional approach
 - . the instructional use
 - etc.
- . Good use of graphics and sound. Use of multimedia
- . Cost, particularly for multiple users. Network installation permitted?
- . And others, see review forms from lecture on organizing software collections.

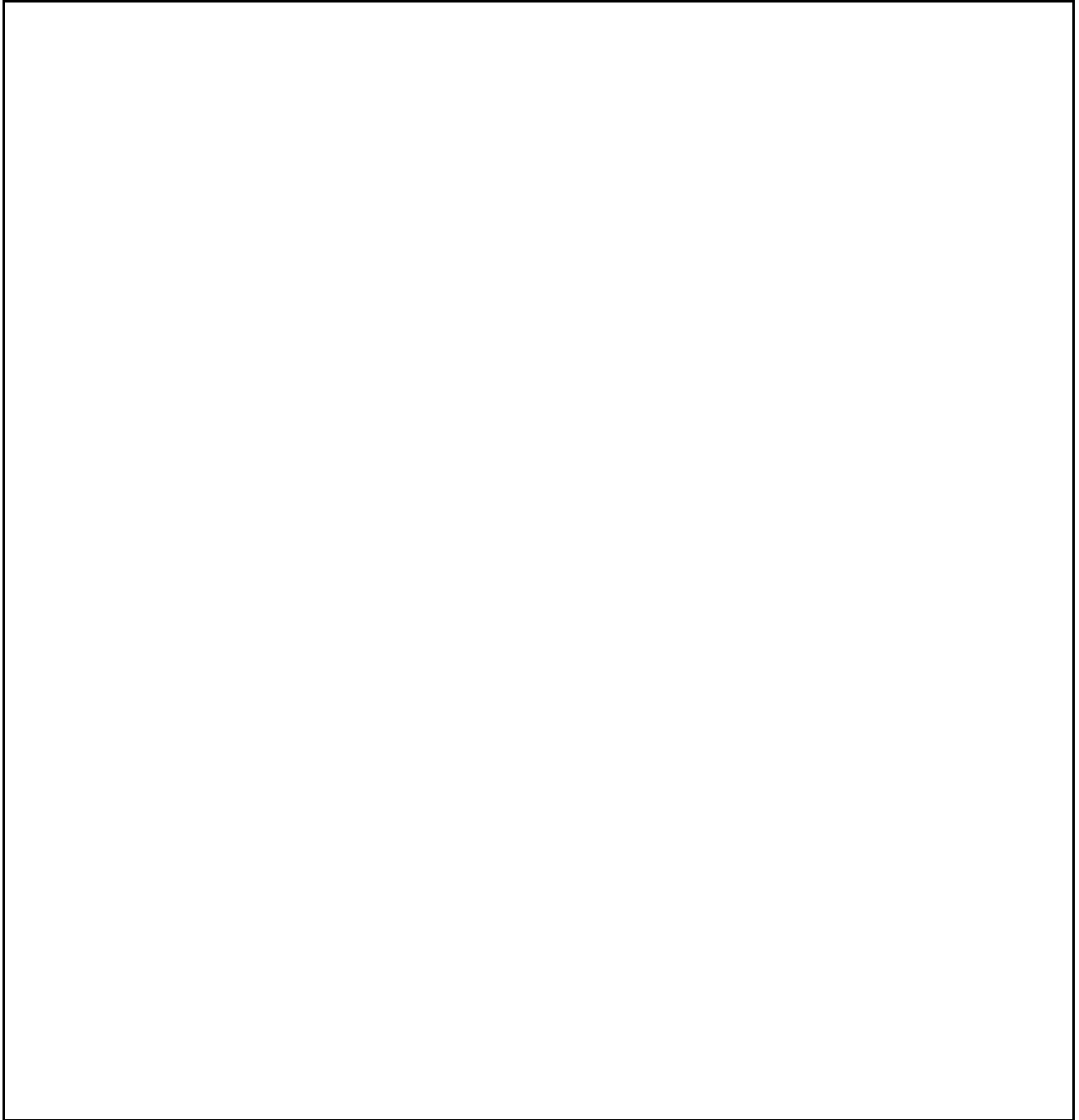
Organizational issues

- . Instructional media center should be in charge of **all** software
- . Where is software used - classroom vs. media center
- . Where is software kept
- . Process for selecting software - same as for other instructional materials
- . Educating and supporting teachers

Lecture 4.2.

Software in personal information management

Write down some functions you would like to see in an integrated personal information manager (7 min)

A large, empty rectangular box with a thin black border, intended for the student to write down functions they would like to see in an integrated personal information manager. The box occupies most of the lower half of the page.

Lecture 4.2A

Meta search engines

Definition

A system that, based on a single query formulation, searches multiple other systems and returns the results of these multiple searches. A meta search engine can reside on the user's computer (client) or on a computer the user's computer connects to (server).

Systems searched can be

Web search systems such as AltaVista or Yahoo

Other databases

Examples (this concept existed before the Web!)

micro-CSIN

Gateways to multiple online systems

Copernic 2001 (client-side, www.copernic.com)

Metacrawler (server-side, www.metacrawler.com/index_power.html)

BookWhere V4 (client-side, searches Z39.50-compatible databases)
<http://www.webclarity.info/bookwhere.html>

LC Z39.50 Gateway (server-side, <http://www.loc.gov/z3950/>)

Functions: Overview XXX need dots

Assist with query formulation and search scheduling

Search multiple systems

Assist with processing results

Functions: Detail

Assist with query formulation and search scheduling

- . Provide a thesaurus
 - . . Means for user additions to thesaurus
- . Directory of topics
- . Schedule topics for searching at certain times

Search multiple systems

- . Select systems to search or assist in selection
 - . . Store systems to be searched with each topic
- . Translate query formulation for each system
- . Make connection, send query, and receive results
 - . . Searching several systems in parallel
 - . . How many items can be retrieved from each site?
- . Needed for all of these functions: database of systems
 - . . Means for adding to this database and customizing each system "profile"

Assist with processing results

- . Finding duplicates
 - . . Same URL found from several systems
 - . . Mirrored URLs
- . Preparing abstract
- . Sorting by various criteria
 - . . Clustering
- . Indexing by user's descriptors

More examples and references

Sonia (Stanford) (see reading)

Meta Search Engines. by Nancy Garman. ONLINE, May 1999

<http://www.onlinemag.net/OL1999/garman5.html>

scroll down to find meta search tools listed by IQ score. suggestions/contact

<http://www.zdnet.com/searchiq/directory/multi.html>

Lecture 4.2B

Personal information management system and links to the Web

Ideally, a personal information system would integrate all information management functions of the user in one interface. With respect to the Web, the users own database would appear simply as part of the Web. Even word processing and graphics should be included, but one will have to settle for minimal functionality here with compatibility to dedicated software. Many of the functions discussed in the following exist in individual programs, and some programs integrate several of these functions, but no completely comprehensive program exists.

Wish list for a personal document management system XXX dots

1 General

- . **Several types of systems rolled into one.** Such a system could make use of other "agents", such as a translation Web site, a translation program residing on the user's machine, or a vocabulary server, preferably without the need for user intervention.
- . Lots of user-specifiable features, but usable out of the box with sensible defaults
- . Should include a calendar and personal workflow manager

2 **Handles all kind of documents (very broadly defined)**, both in full version (text and images, multimedia documents) and bibliographic records, as well as address records

- . Documents produced by many applications (word processors, graphics program, presentation manager, etc.)
- . Any kind of Web document, including PDF and PostScript text and any kind of multimedia document
- . e-mail with threading and categorizing
- . Data files
- . Bookmarks with easily visible annotations (bibliographic record)
- . Queries (manage a query database)
- . Databases and search systems (serving as a guide to databases)
- . Software
- . Persons and organizations with their addresses

3 System rules and conventions

- . Support for a comprehensive metadata schema (data dictionary), with templates that select elements required according to type of document
 - . . Metadata schema management
- . Support for thesaurus management and use
 - . . Thesaurus incorporates folder hierarchy
 - . . Comes with built-in thesaurus ready to use
 - . . Seamless access to thesauri and dictionaries available on the Web (through one or more vocabulary servers if available) both for ad-hoc use in queries and as a source of concepts, terms, and relationships in thesaurus management
 - . . Thesaurus management, both automatic addition and user editing
 - . . Flexible thesaurus search and display functions. Display of folder hierarchy

4 Searching

- . Support for query formulation, especially thesaurus support through the internal thesaurus and access to external thesauri
 - . . Relevance feedback: Generating new search terms or a whole query from a sample of relevant documents and (optionally) a sample of irrelevant documents. Can obtain sample information unobtrusively as the user marks/uses documents found.
- . Searching with one query any combination of
 - . . internal database
 - . . multiple Web search systems
 - . . multiple databases accessible through the Web or otherwise, Z39.50 or database-specific protocol
- . Intelligent mapping of query formulation to exploit features of each external system
- . Query translation into multiple languages using dictionaries available on the Web as needed
- . Accessing multiple sources in parallel, preferably in the background, with the possibility of displaying results as they become available
- . Scheduling searches for one-time or periodic execution
- . Processing and integrating results with functions described below before integrating into database or along with it.
- . Powerful search features for the internal database (system has no control over external databases)

5 Document acquisition and reformatting

- . Can acquire all types, preserving all content
- . . Problems to watch out for: Images included in Web pages, frames, dynamic Web pages, preserve hyperlinks, acquire URL with document, complex image and multimedia formats
- . . Cascading acquisition from Web: A page and all pages it links to, perhaps several levels deep (user-specified). Might want option to limit to links on the same site or by other criteria
- . . . Special case: Assemble an entire document from TOC (Table Of Content) page (for printing). System needs to be able to distinguish *includes* links that link to parts of a logical document from other types of links.
- . . Selective acquisition: Selected portion (text and images) from Web page directly to a designated document (one shortcut key: default document presently in effect; a second short cut key: ask for destination document) (Being able to designate as default clipboard would be a useful Windows feature)
- . Regular check for updates of documents. Update even segments selected from a Web page found and transferred to a page composed by the user (done by remembering sentences or other identifying features before and after the selection).
- . Integration of document in database at acquisition, with system-produced metadata (see below), optionally checked and augmented by user; for example, supply a new title (system should keep the original title as well). Assign default descriptors (minimally: file in default folder) set by user for a series of related transactions. Maintain date of acquisition
- . . Option to schedule actual file copy (“grab”) for later
- . Version control with drill-down to earlier versions (requires documents to be stored in the system rather than merely bookmarked)
- . Format conversion, esp. format conversion that enables free-text searching or other manipulation and use of text and images
- . . PDF or PostScript to text
- . . OCR (optical character recognition), including detection and OCR of text in embedded images
- . . Translation (see below) might be considered format conversion that enables free-text searching in one language.

6 Document processing and organization

- 6.1 . Functions that can be executed by the system and/or the user with contribution from each varying from 100% to 0%. These functions can be initiated by the system or by the user (independently from how much work each does on the function). When done by the user, the system should provide support
- . . Finding duplicates
 - . . . Duplicate URL's retrieved from searching multiple search systems (or even the same search system)
 - . . . Duplicate documents stored at different URL's (identifying candidate pairs based on title and other data, then complete comparison)
 - . . . Versions in different languages
 - . . Detecting and marking the structure of large unstructured documents (may be useful for preparing document representations)
 - . . Preparing document representations
 - . . . Descriptive metadata
 - . . . Subject indexing
 - Support through thesaurus access
 - . . . Classifying
 - Assigning documents to predefined categories / folders
 - When done by the user, should be possible by typing in descriptor/folder name. Should be able to designate several folders, to be shown in a small window with associated numbers/single letters, so that a document could be assigned to a folder by simply typing that number/letter (first-named folder is default, z means other folder)
 - Clustering
 - Prepare representations of classes
 - Select typical example of a class for display
 - . . Evaluating documents. Ranking'
 - . . Translation
- 6.2 . System support for user operations
- . . Marking or deleting individual documents or whole groups
 - . . Annotations, marked as sharable or private
 - . . Adding links

7 Display / views / reports

- . See also D Searching
- . Sorting by values from the record
- . Display most typical or most actionable element of a group (such as the most recent from a group of e-mail messages), with easy access to the other members of the group
- . Within folders, capability of viewing only selected types of documents
- . Putting together a presentation of selected Web and other documents with annotations (tool for an information specialist)
- . Display all types of documents correctly (including any type of multimedia document)
- . Print all types of documents correctly

8 Sharing database or parts with others

- . Sending e-mail or instant messaging with attachments
- . Send documents with or without annotations (system should ask, user might want to send even annotations marked private)

Lecture 4.2SupplA

Software in the home

Types of home applications

- . A wide variety of applications for children and adults, usually at a moderate level of sophistication and/or volume of data. The following list is by no means all-inclusive.
- . Simple word processing, database management, personal information managers, genealogy
- . Communications, connection to Prodigy, America Online, CompuServe, etc. Web browsers
- . Financial management
- . Household management
 - . . Recipes
 - . . Control lights and appliances
- . General information, reference works
- . Education (see software in education)
- . Hobby support (genealogy, stamp collecting)
- . Entertainment, games
- . Increased use of multimedia in home applications

Special selection criteria for home software

- . Hardware sometimes still limited
- . Ease of installation and use essential (no experts to ask as in an organization)
- . Low cost

Special characteristics / selection criteria for financial managers for the home

- . General principle: Multiple use of data input once.
- . Connection with a check-paying service such as Checkfree integrated with balancing checkbook.
- . Regular payments.
- . Input of expected or contemplated expenses for budgeting and cash-flow projection. Later activation for payment.
- . Keeping multiple accounts.
- . Apply a payment to a credit card or other loan both places: add to credit card balance, subtract from paying account balance.

- . Prepare tax return and tax projections (requires categorizing expenses properly).
- . Prepare a statement of financial situation, including net worth.
- . Financial planning and advise, including
 - . Retirement planning - compare different plans, including pension plans
 - . College costs and other usual large expenditures (orthodontist, major home maintenance)
 - . Compute life insurance needs.
- . Easily do computation needed for mortgage applications, college financial aid applications, and such.
- . Has a default classification of income categories and expense categories that can be modified.
- . Can figure out best credit card to use from closing date and interest rate.
- . Retrieval and display/report of transactions by multiple criteria.
- . Inventory and keeping track of assets
 - . Keeping track of investments.
 - . Inventory of possessions and related insurance needs

Special characteristics / selection criteria for games

- . Manual dexterity and reaction required
- . Mental effort required
- . Levels of difficulty - challenge
- . Entertainment value
- . Educational value
- . Played once or often
- . Level of violence
- . Variety of action
- . Quality of graphics and animation. Quality of sound
- . For programs such as a flight simulator: realism, number of airports
- . Number of live players vs. playing against the computer
- . Games on the Internet (free or fee-based)

Lecture 4.2SupplB

Software in public library service

Goal

- . Equalizing access
- . Support computer owners, including small businesses

Purposes

- . Software for actual use
 - . . Educational software
 - . . Entertainment software
 - . . . For both: Multimedia
 - . . Informational software
 - . . Word processing software, tax software, will maker, and similar programs that require only one-time entry of data or that can be used with data from a floppy disk
 - . . Access to networks - may need to work out system for paying fees
- . Software users can try before purchase
- . Software distribution (public domain software, shareware)

Selection criteria

- . See under educational software and software for home use. Also consider small business
- . For trying out, several programs of the same kind should be available, at least in demo versions.
- . Public domain software that users can copy and take home. Shareware

Acquisition

- . Donations (e.g. from local computer stores)

- . Purchase (public domain software is available very cheaply, but also consider disk space required)

Special problems

- . Multiple access - need computer room with network
- . All software should be stored on hard disk to avoid handling floppies - requires very large hard disk (Publication opportunity: software sampler on CD-ROM)
- . Online software catalog
- . Have fliers, empty boxes, manuals, access to reviews
- . Prevent unauthorized copying - needs operating system facility. For example, store programs as hidden files (using cryptic file names) with access strictly through the catalog

Lecture 4.2SupplC

Software in library operations

Since automation has been a hot topic of late -- at least my mailbox keeps filling with requests for the CMBLS list or other information -- I thought I would post a copy of the evaluation worksheet I created when examining various pieces of library automation software. Anyone can use it as a model -- adding, deleting, or changing information to suit your particular needs. Of course, since I'm a Cataloger, it has a bit of cataloger bias, so be warned!

Ron Bettencourt
Cataloger
B.M.C. Durfee High School
360 Elsbree Street
Fall River, MA 02720
Internet rbettencourt@umassd.edu

Some additions by DS. General software requirements deleted from the list to concentrate on features specific to library automation software.

Automation Worksheet

Note: Many of the criteria listed here apply also to bibliographic information managers.

Components of System

- . Database of users and user needs
- . Acquisitions
- . ILL requests to other libraries
- . Cataloging
- . Serials
 - . . Online Catalog
 - . . Authority control
- . Circulation of library-owned materials
- . ILL requests from other libraries
- . Circulation of materials obtained through ILL
- . Computerized reference, for example expert system for referral to other information sources
- . Gateway to other information sources, both locally mounted CD-ROM databases and online sources (possibly integrated with the online catalog)
- . SDI

Overall functionality

- . System is fully integrated with all components utilizing a central database?
- . All data can be input, modified, deleted online?
- . Bibliographic files can be loaded from other sources?
 - . . Batch load from MARC (Machine-Readable Cataloging) tapes and such?
 - . . Online upload from utilities such as Bibliofile, LaserQuest, etc. and MITINET/marc?
Individual records? Whole batch of records?
- . System supports all current MARC formats?
- . Central database while maintaining individuality of catalogs in a library system (as in CARL)?
- . Supports multiple languages?

System Description

- . Flexibility of system (e.g. screen formats, printing options, etc.)
- . LC extended character set/full ALA character set (Greek, Cyrillic, etc.). Chinese characters
 - . . Displays full set on processing terminals
 - . . . OPAC
 - . . . Printouts
- . System provides aids to record input (work forms, prompts)
 - . . Multiple work forms for different kinds of documents and other types of objects being cataloged
 - . . Dynamic work forms
- . Bar codes
 - . . Provision for smart bar codes to be created
- . How is information entered
- . Number of possible terminals
 - . . On-site
 - . . Dispersed
- . Response time under varying system loads (number of concurrent users, operations requested by users)
- . System reliability
- . How is back-up of data in system made
- . What is used as back-up if system is down or some system component is down
- . How is the system restored in event of failure? What happens to just recently added data?
- . Produces catalogs of special collections (e.g. Special Collections, foreign language materials, AV)

Bibliographic Database

- . Maximum database size
- . Bibliographic records are based on full MARC format for
 - . . Books
 - . . Serials
 - . . Music
 - . . Sound recordings

- . . Films
- . . Maps
- . . Computer files
- . . Web pages
- . . Manuscripts

- . How will the system adjust to comply with MARC format integration?
- . Accepts, stores, and uses full-MARC records
- . Options for reconfiguration or change of data
- . Methods of entering bibliographic data
 - . . Keyboard entry
 - . . MARC file load
 - . . Conversion options
 - . . OCR input
- . Fields include (if not full MARC)
 - . . Location
 - . . Call number
 - . . Main entry
 - . . Uniform title
 - . . Title
 - . . General Material Designation (GMD)
 - . . Statement of responsibility
 - . . Edition

 - . . Imprint
 - . . Collation
 - . . Series
 - . . Notes
 - . . LC card number
 - . . ISBN/ISSN
 - . . URL

- . . Bibliographic utility control number
- . . Subject headings
- . . Added entries
- . . Holdings
- . . Bar Code Number(s)
- . . Price
- . Possibility of adding new fields
- . Output formats available, e.g.
 - . . standard AACR2 format
 - . . Prints spine labels (multiple copies/volumes)
 - . . Prints processing labels
 - . . Prints bar code labels
 - . . Define own formats
- . Searching capabilities, esp. indexes included (not complete)
 - . . Names (Main entry)
 - . . Names (Added entries)
 - . . Titles (Primary in direct order)
 - . . Titles (Added, linking in direct order)
 - . . Series
 - . . Subject
 - . . Classification number
 - . . Multiple local numbers
 - . . Local, OCLC, etc. control numbers
 - . . ISBN/ISSN
 - . . LCCN
 - . . Publisher
 - . . Copy/item number
 - . . Keyword from names
 - . . Keyword from titles

- . . Keyword from series
- . . Keyword from journal title
- . . Keyword from subjects
- . . Keyword from contents
- . . Keyword from notes

- . All matching copies linked to one bibliographic record
- . All editions linked to a basic record
- . Location status information given at item (copy) level
- . System displays holdings for multiple locations
- . System accommodates multiple separate collections
- . System guards against duplicate bibliographic record creation
- . Provision for LC rule interpretation options
- . Record display indicates missing items

How are serials handled

- . Full MARC format
- . Holdings
- . Reports of missing issues

On-line Catalog (see also Searching)

- . Fully Integrated with other modules to allow display of current status?
- . Interaction style
 - . . Use of command language, scripts, menus, direct manipulation, graphics, hypermedia style
 - . . Full words or names or abbreviated search keys or both?
- . Record display includes full MARC fields, options, etc.
- . Preserves local data in record display
- . Record display includes circulation categories
- . Displays acquisitions, processing, circulation status
- . Search
 - . . Combination searches

- . . . author/title
- . . Boolean logic capabilities, ranking capabilities, etc.
- . . Indexes include (see above)
- . . Fixed field search qualifiers (language, date, etc.)
- . . Can items be selected by language?
- . . Retrieve "j" (juvenile) items from fixed field code?
- . . Can items be searched by type of format (all video cassettes?)
- . . Can items be selected by date
- . . Searchable terms
 - . . . Is "United States" a searchable term in subject and name fields?
 - . . . Is there a list of unsearchable stop words?
 - System allows for locally-defined search stop words?
 - . . . Allows truncated term entry in all searches
- . . Backtrack on a search
- . . Branch off into different directions from different locations in a search
- . . Displays search term(s) throughout search
- . . What happens with failed searches (spelling errors?, etc.)
- . Browsing order of catalog / arrangement of search results
 - . . View records in alphabetic order (like regular card catalog for browsing)
 - . . View records in call number order (like shelflist), etc.
 - . . View records by date (oldest first or newest first)
 - . . Can items be separated by type of format (books listed first, etc.)?
- . Self-service HOLD placement
- . OPAC includes user aids such as
 - . . help screens
 - . . . local help screens
 - . . . online tutorial
- . Instructions and help screens can be displayed in multiple languages (user choice)
- . Page forward/backward in OPAC
- . Auditing and reporting features

- . . Logging activity patterns at a terminal
- . Easy to use
- . . used by evaluator after NO instruction
- . . used by evaluator after BRIEF instruction
- . . used by evaluator after MUCH instruction

Authority system

- . Online interactive authority control system
- . Support MARC record authority format
- . Authorities
 - . . Personal name
 - . . Corporate name
 - . . Geographic name
 - . . Conference name
 - . . Uniform title
 - . . Series
 - . . Subject
- . Are they linked (e.g. items used as both authors and subjects treated separately but with a link)
- . Locally produced cross-references can be created?
- . Is the authority files at the same time used as an index? In other words, are authority files linked to each occurrence in the bibliographic file?
- . Global entry change at field level / subfield level
- . Scope, reference, and information notes are included
- . Can load LC subject authority and LC name authority files from magnetic tape or CD-ROM
- . Browse features in authority entry searching
- . Are all blind cross-references suppressed
- . Provision for multiple authority heading sources
- . Authority records matched against changes and updates in a revised file
- . Are "see" references automatically connected to the desired item so you press one key instead of typing in the correct heading

- . Authority record displays record source and cataloger
- . Multilingual authority list

Circulation information

- . Fully integrated with bibliographic database
- . Maximum size of patron database
- . Method of entering patron information
- . Method of deleting patron information
- . Global delete?
 - . Is it possible to delete entire Senior class with one command
 - . Does it delete if items are outstanding?
- . Supports bar wands
- . Backup of transaction files
- . Multiple security levels
- . Multiple patron types (student, faculty, etc.)
- . Multiple circulation types (overnight, etc.)
- . Is a patron with overdues "locked out" unless overridden?
- . Are holds and reserves possible (e.g. for special projects, etc.)
- . Reports of data (to be used, for example, for collection development)
 - . Statistical reports
 - . Number of circulations per item
 - . Items never circulated
 - . Status reports
 - . Inventory reports
 - . Reports on the age of certain portions of the collection?
 - . Do inventory reports indicate if item is out of place on the shelves?
 - . Print formats for printer
 - . Tables
 - . Charts
 - . Graphs

- . . Information on all items loaned to a single patron
- . . Provide information per item
- . . Provide information per classification range
- . . Provide information on number of items out in circulation
- . Overdue reports
 - . . Overdue notices (sorted by grade, homeroom, etc.)
- . Protects patron privacy

Networking

- . Is connection to public library (etc.) network possible
- . Is connection to local CD-ROM network possible
- . Web access
 - . . Through own Web interface
 - . . Z39.50 compatible

Expandability (full text options, support for digital library)

Sample database (for trying out the system) available? Size?

Management reports included

- . OPAC user frequent dead-end search terms
- . processing activities
- . file counts
- . System provides exceptions report, e.g. entry-matching problems
- . Custom reports

Lectures 4.3-4.5

Final Review

Question 1

A large school district wants to set up a systematic procedure, including a software testing lab, to evaluate educational software. They called you in as a consultant to devise a plan. Think of the various aspects to be considered in setting up this procedure and the lab and discuss each of them briefly.

Question 2

Assume you are a usability consultant. A large software company is thinking about developing a new product (a smart Web browser, an encyclopedia, a home finance manager, or whatever). They want to involve you in the entire development process from beginning to end and ask you to submit a plan. Outline such a plan.

Question 3

What is the purpose of usability heuristics?

How do you know that a given list of usability heuristics is actually useful?

How would you go about developing a set of 10 usability heuristics?

Question 4

List some important ideas you have learned in this course.

Student's questions

Review of topics

Functions/types of software

- . System software
- . Information management
 - . Database management system
 - . Information storage and retrieval
 - . Personal information manager
 - . Document generation and formatting
 - . Number processing and mathematics
 - . Data conversion
- . Work plan and project management
- . Communication software
- . Business operation support (for organizations or the home)
- . Education
- . Games
- . Data-based applications. Databases with retrieval and processing software

Software evaluation and selection - general principles

The software selection process

Software evaluation criteria

General software evaluation criteria

- A General Information
- B Capabilities
- C Program correctness. Bugs
- D Computer environment required/supported (hardware and software)
- E Cost and ease of use
 - . Installation cost and maintenance cost (capital cost)
 - . Daily operating cost
 - . Ease of use
- F Documentation. Vendor support. Error-handling
- G User interface features
- H Multi-user capability
- I Security
- K Input/output and working with other programs

The user-system interface

Throughout: Consistency, intuitiveness

A Interaction styles

- A1 Command-driven: User enters commands at system prompt
 - . Formal command language. Faceted structure for ease of learning
 - . Natural language
- A2 Script: User fills in blanks on a form
- A3 Menu driven: User selects from displayed options
 - . Static vs. dynamic menus
 - . Direct vs. symbolic selection
 - . Binary vs. multiple item menus
 - . Menu organization
 - . Skeleton vs. embedded menus
 - . Permanent vs. transient menus
 - . Sequence of menus
 - . Menus combined with other interaction styles
 - . Menu selection guidelines
- A4 Direct manipulation
- A5 Graphic User Interface (GUI): A combination of approaches. Icons
- A6 Parameters for characterizing interaction styles
 - . Amount of information on available choices/expected input presented to the user
 - . Method of data input (symbolic, direct selection, direct manipulation)
- A7 Actual interfaces use a combination of these approaches

B Human-computer interaction principles

- . Models/metaphors
- . Data entry and display
- . System messages

C User characteristics: Physical and cognitive, user knowledge

D Interface display characteristics (hardware and software)

Importance of interfaces that can be customized (user-definable interface)

Display design

- . Goals: Transmit information efficiently, solicit input from user with minimum chance of errors.
- . Consider: Task, user, capabilities of the display device and the computing environment
- . Cognitive Match Interface Design (COMIND), for example, use known symbols and analogies where cogent to the computer environment
- . High-level means of expression: Text, graphics, sound
- . Low-level means of expression: Typography, highlight or lead symbol, Graphical means for highlighting or de-emphasizing (such as bold, color), boxes for grouping

System messages, help, and error handling

- . Functions: Prompt for user input, feedback, warning, help, report on system status, error message
- . Message parameters: Length, expression (see under display)

- . Help
 - . . Content: What to do, how to do it
 - . . When: Anticipatory, on request, after an error
 - . . Levels of context-sensitivity
 - . . Content of help: Only on system in general or with values from the database (as in pop-up list of possible values from a field)
 - . . Help vs. on-line documentation
- . Error handling
 - . . Functions: Detecting errors, announcing errors to the user, correcting errors and/or giving help
- . Error message guidelines: Be specific, clear, polite

Documentation

- . Purposes: For user: Evaluation and operation; for programmer: maintenance/modification, coordination in large programming projects
- . Parts/content: Reviews, demo, user's guide or users manual (Tutorial, reference manual, reference card)
- . Forms of documentation: Paper, audiotapes, videotapes, presentation, training course, machine-readable (incl. CD-ROM, tutorials using computer-assisted instruction), interactive (fax, bulletin board, telephone)
- . Sources: Producer / publisher of the program, third-party publishers, user groups, in-house
- . Evaluation criteria for documentation: Availability (consider all sources), coverage, level of detail, overall arrangement, clarity of writing, examples, finding aids (index, access by problem to be solved), searchability by computer, layout and typeface

Addendum to general software criteria

- . Updates available
- . Adapts to changes in the world (e.g., new tax laws, different situation on the financial markets, changing trends in college costs)

Software testing procedures

- . Context/purpose: Formative evaluation for producer, summative evaluation for consumer
- . Perspectives: Correctness, performance, usability, effectiveness
- . Testing procedures: Walk through all features (complete test data), test time (realistic amount of data), test with real users, test for ultimate effect
- . Forms for recording results

Usability assessment

- . Formative: Identify usability problems
- . Summative: Give usability rating
- . Methods (see also attached summary from Nielsen)
 - . Usability walk-through by experts. Heuristic evaluation
 - . Testing with representative users
 - . Find users or respondents (sample size, representativeness)
 - . Define tasks to be performed or questions to be asked (representative of real environment)
 - . Define performance measures (error rates, time for task)
 - . Data collection
 - . From people actually using program: logging, observation, think-aloud (including audio- or videotaping for later leisurely analysis. Test and observation arrangements (special lab, observation during "natural" use of program).
 - . Questionnaires, interviews, focus groups
 - . User error reports and help requests

Software information sources and software collections

Software information sources

- . Types of sources: General in computer science, specific on software or computer or educational technology products
- . Formats: Print, CD-ROM, online
- . Characteristics of sources. Evaluation
 - . Coverage
 - . Information/detail given for each program
 - . Clarity, formalization, and consistency of evaluation criteria
 - . Searchability

Organizing a software collection

- . Integrated vs stand-alone
- . Steps: User needs, selection criteria, identify software to acquire, in-house testing and evaluation, acquire, catalog.

Organizing the software evaluation process

- . Commission reviewers. Have several reviewers and combine data. Centralized testing lab.

Usability

See attached copies

Variations on heuristic evaluation

Expert evaluation

Informal developer walk-through

Cognitive walk-through

Pluralistic usability walkthrough, joint design walkthrough

Collaborative usability inspection

Software in specific environments

Software in education

- . Environment of use: Schools, home, training in any kind of organization
- . Types of educational applications
 - . . Programming languages for use by students: Programming as a paradigm for problem solving, gaining insights in a subject by writing programs in that subject area (e.g., language processing)
 - . . Computer-supported instruction
 - . . Networking to support information gathering and communication among students and between students and teachers (including distance learning)
 - . . Computer-assisted instruction - software to transmit instructional content
 - . . Educational games
 - . . Drill and practice
 - . . Managing instruction, administering tests, student assessment
- . Special criteria for selecting educational software: Fit with educational philosophy and objectives, correctness, free of stereotypes, motivation, several difficulty level, learner control, documentation addressing these issues
- . Organizational issues

Software in the home

- . Types of home applications
- . Special selection criteria for home software: Limited hardware, ease of installation and use, low cost
- . Special characteristics / selection criteria for games: Manual dexterity and reaction required, mental effort, levels of difficulty - challenge, entertainment value, educational value, played once or often, level of violence, variety of action, quality of graphics and animation, quality of sound, realism (where applicable)

Software in public library service

- . Goals: Equalizing access, support computer owners
- . Purposes: Software for actual use (education, entertainment, information, word processing, tax, etc., access to networks), software try out, software distribution
- . Selection criteria
- . Special problems: Multiple access, online software catalog, make information on programs available, prevent unauthorized copying

Software in library operations

- . Components of System (All library functions, including database of users and user needs)
- . Overall functionality (e.g., integration, multilingual capability)
- . Bibliographic database (size, record formats or specific fields included, searchability)
- . On-line Catalog (interaction style, search functions)
- . Authority system
- . Circulation information
- . Management reports
- . Networking

Lectures 5.1 - 6.5**Student reports and Final exam**

		Student	Title
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			

