

User-Centered Origins:
An outline of the User-Centered Design Process
and the sources of current industry practices¹.

¹ Based on the course “User-Centered Web Design” developed and taught by Jess McMullin at Grant MacEwan College, Fall 2000.

Table of Contents

| | |
|--|----|
| Introduction..... | 3 |
| Week One | 3 |
| Introduction to User-Centered Design..... | 3 |
| User-Centered Design and the Bottom Line: UCD is a sound investment..... | 5 |
| UCD Origins..... | 5 |
| Week Two..... | 6 |
| Field Studies: Observation & Interviewing | 6 |
| Field Study Experience..... | 6 |
| Field Study Origins..... | 6 |
| Week Three..... | 7 |
| What to do with the Data? | 7 |
| Presentation and Analysis Origins..... | 8 |
| Week Four..... | 9 |
| Design discipline survey..... | 9 |
| Card Sorting – A Conceptual Modeling and Information Architecture tool | 10 |
| Low-fidelity Prototyping | 10 |
| Week Five..... | 11 |
| Usability Inspections..... | 11 |
| Week Six..... | 12 |
| Usability Testing..... | 12 |
| Conclusions..... | 13 |
| Acknowledgements..... | 14 |
| Appendix..... | 15 |
| Ten Usability Heuristics | 15 |

Introduction

User-centered design (UCD) is a methodology and philosophy for creating products that makes users and their goals the primary driver in development. Industry currently is embracing user-centered principles and practices at an accelerating pace, and some UCD practices are becoming standard in commercial development, especially in transaction-based online applications such as ecommerce. However, several challenges must be surmounted in order to increase the rate of acceptance of UCD in industry. Lack of people with UCD skills and experience is one such fundamental barrier. In order to help overcome that obstacle, I offered a six-week course on user-centered web design at a local college. In both teaching and learning this material it is informative to see the sources of UCD practices. This paper follows the curriculum I developed for the course, outlines the material taught, and traces these current best practices to their origins.

The calendar listing describes the course:

“Meeting user’s needs is crucial to successful, sustainable projects. By putting the needs of the user as the driving force in new media development, teams avoid the pitfalls of focusing on technology or design for its own sake. User centered design offers pragmatic processes and techniques that developers can incorporate into existing development efforts. This course teaches practical ways for understanding your audience, rapidly exploring design alternatives, and evaluating design options through observing real users. Hands-on exercises include field studies, low-fidelity prototyping, and discount usability testing. Adopting these methods can save overall development time, increase end-user satisfaction, and produce projects that truly meet user needs.”²

The course itself ran over six weeks for three hours one evening each week. The curriculum was a combination of in-class lecture and demonstration, and hands-on activities. The material was presented in the same order as the User Centered Design Process itself – this process-centered approach was chosen to emphasize the steps and activities at each stage in the UCD process.

Week One

Introduction to User-Centered Design

User-Centered Design is not a concrete methodology or process – different people and organizations have different versions of UCD. However, there are some common elements for all versions of the process: an early focus on users and tasks, empirical measurement of product usage, and iterative design whereby a product is designed, refined, and evaluated repeatedly³. In this course, User-Centered Design is represented as an iterative four-stage cycle of Understanding, Design, Prototyping, and Evaluation, as illustrated in Figure 1.

² Grant MacEwan College *Guide to Part-Time Learning – Fall 2000*

³ Gould, J. D. & Lewis, C. “Designing for Usability: Key Principles and What Designers Think.” *Communications of the ACM*, 2(3), March 1985, pp. 300-311

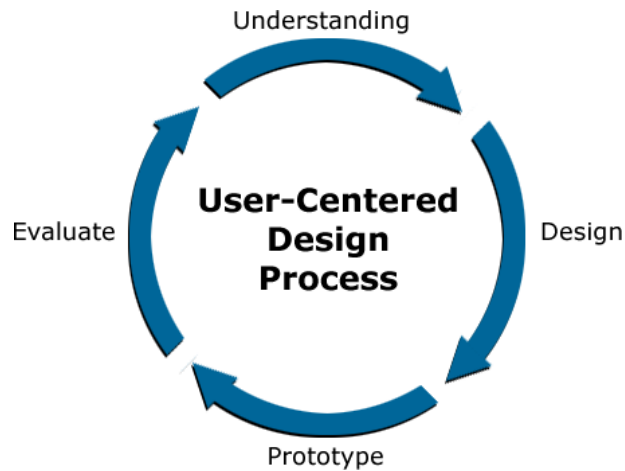


Figure 1. The User-Centered Design Process

Understanding includes understanding project goals and limitations; users, their goals, and the activities that accomplish those goals; and how users currently meet their needs without the product under development. In this course the emphasis is on understanding users through behavior-focused field research. Understanding produces empathy for users in the development team, and creates a foundation for a project that meets real needs instead of functionality added mainly for feature-list comparisons. The Understanding phase generates user experience requirements that are used to determine functionality during Design and act as benchmarks for performance in the Evaluation stage.

Design is a wide-ranging and varied stage, covering abstract conceptual design to concrete visual design and everything in between. Design translates requirements into functionality and form, and includes disciplines and practices such as Conceptual Design, Information Architecture, Interaction Design, Information Design, and Visual Design. Because of the wide scope of Design, this course does not address any design discipline in detail.

Prototyping takes abstract design concepts and makes them concrete. A prototype is a mockup or representation of the final system, with only some of the functionality of the finished product. A prototype generally begins as a low-fidelity mockup that represents the system, but isn't very realistic, such as a paper mockup of a website. With each pass through the UCD cycle, the realism of the prototype increases to reflect more functionality and become more like the final form of the product.

Evaluation works with a prototype to see how well it meets the user experience requirements from the Understanding phase. Evaluation is usually focused on usability evaluation – how usable is the product? Evaluation can also include broader measures of subjective satisfaction and user response. Typical evaluation methods involve expert review of the prototype and representative users trying to accomplish real goals using the prototype. Evaluation produces data that informs the team's understanding of the project and the target users, and the process returns to the Understanding phase.

The UCD process continues, with the product becoming more and more refined with each iteration. The process is ideally complete when a project meets all user experience requirements. However, more often a product cycle is finished when a project runs out of time or money. Even in that case, the UCD process produces better results than if the team designs a product without consideration of the intended user.

User-Centered Design and the Bottom Line: UCD is a sound investment.

User-Centered Design is a sound business decision, resulting in significant financial benefits for commercial organizations⁴. One study estimates the potential increase in ecommerce sales at \$19 billion for the year 2000 if UCD practices were adopted industry-wide⁵. Several studies have shown that UCD produces a cost benefit ratio between 1:10 and 1:100, *i.e.* for every dollar spent on UCD practices, the organization will realize from ten to one hundred dollars^{6,7}.

Some specific benefits to the organization include⁸

- increased sales and customer satisfaction
- a competitive edge over competitors who do not take usability as seriously
- advertising advantages
- better notices in the media
- reduced development and maintenance costs
- improved productivity and operational efficiency
- reduced training costs
- lower technical support costs
- reduced documentation costs
- litigation deterrence

User-Centered Design is one of the single greatest investments a product developer can make today.

UCD Origins

User-Centered Design evolved in the field of human-computer interaction, though the process can be applied to any product with end users. UCD was first articulated as such in *User-Centered System Design*⁹. The term User Centered System Design was coined by Paul Smolensky to reflect the initials of the University of California, San Diego (UCSD)

⁴ Bias, R. & Mayhew, D. (1994) *Cost-Justify Usability*.

⁵ Creative Good (2000), *The Dotcom Survival Guide*, url: <http://www.creativegood.com/survival/>

⁶ Gilb, T. (1988) *Principles of Software Engineering Management*. Addison Wesley, Reading, MA.

⁷ Pressman, R.S. (1992). *Software Engineering: A Practitioner's Approach*. McGraw Hill, NY.

⁸ Donahue, G., Weinschenk, S., & Nowicki, J. (1999) *Usability is Good Business*. Compuware Corporation. url: <http://www.compuware.com/intelligence/resources/usability.pdf>

⁹ Norman, D. A. & Draper, S. W. (Eds.), (1986) *User Centered System Design*, Lawrence Erlbaum Associates: Hillsdale, NY.

where the central work of the book was completed. Subsequently UCSD has been shortened to UCD. Other related terms are usability engineering and human factors engineering, which involve similar processes. The term user-centered design places the focus on the user, and so has become more widely used.

Week Two

Field Studies: Observation & Interviewing

Know users, their goals, and the activities that accomplish those goals are essential for creating outstanding user experiences. Traditionally in product development this information is obtained through focus groups and surveys. While these are valuable tools in determining user preferences, they do not address user behavior. Field study techniques that draw on the ethnographic tradition are more appropriate for understanding user needs.

Observation and one-on-one interviews are the key tools in gathering data about users in the context of use¹⁰. This process is often referred to in the industry as *contextual inquiry*, since observation and interviews are on site at the user's workplace or home. However, because contextual inquiry is part of the specific contextual design process developed by Beyer & Holtzblatt¹¹, the more generic term *field research* is used in this course. Under the best circumstances, a development team would want to interview four or five users from each major user group.

Field Study Experience

Students in the course observed the process of a book being checked out of the library, and interviewed the library technician on staff at the circulation desk in order to develop their observation and interviewing skills. They worked as a team taking on different roles of observer, interviewer and camera person, and used video, and Polaroid cameras in addition to taking notes.

Field Study Origins

Field studies are most often linked with anthropology – so-called “living with the natives” – but also with sociology and social psychology. Konrad Lorenz made an important contribution with the notion of observation without interference in the natural context of the organism. Early human-factors consultants like Frank and Lillian Gilbreth also often worked on the factory floors measuring movement and timing.

¹⁰ Hackos, J. T. & Redish, J. C (1998) *User and Task Analysis for Interface Design*, John Wiley & Sons: Toronto, ON.

¹¹ Beyer, H. & Holtzblatt, K. (1997) *Contextual Design: A Customer-Centered Approach to Systems Design*, Morgan Kaufman: San Francisco, CA.

Week Three

What to do with the Data?

With any field research a major challenge is handling the amount of data collected, and presenting it in a way that communicates what was learned during a site visit to the rest of the development team. Without the ability to pass along this information, the impact of any field research is limited and the understanding of the team is impaired. Accordingly, instruction in field techniques should be accompanied by suggestions for analysis and presentation to manage the data collected. This class discussed several tools to manage field study data. Scenarios, storyboards, task analysis, and personas were the main methods presented.

Scenarios

Scenarios are narratives that describe the story of the user and the activity – “what people do and experience as they try to make use of computer systems and applications”¹². This can be the story of what happens right now in the current system, or they can describe new possibilities for what will happen with the product under development. Scenarios make the process come alive for the development team. This method of presenting data is more engaging than a bulleted list of steps in an activity – it makes the user more real to the development team and captures the sense of the overall flow of an activity.

Storyboards

Storyboards are scenarios in comic-book form – quick sketches that represent the steps of an activity. Storyboards capture the richness of scenarios visually, and are a quick way for others to understand user processes. Comparing the story board of “what happens” to “what will happen” can be a powerful illustration of the savings realized, as people can immediately see the impact of streamlining a process – resulting in an easy-to-understand reduction in the number of steps in the storyboard. Storyboards can also be used to rapidly explore alternatives to the existing process.

Because scenarios and storyboards are *stories* they strike a chord with many people, and are easily understood without the training necessary to follow a particular notation like those that might be used in task analysis.

Task analysis

Task analysis is the structured breakdown of the activities a user engages in to reach a goal. Task analysis is useful for understanding the current flow of activity and in realizing new opportunities for improving that flow. Understanding of the necessary steps for completing an activity en route to a user goal is critical for new products that aim to help users achieve that same goal, or existing products that wish to improve the user experience.

¹² Carroll, J. M. (Ed.). (1995) *Scenario-Based Design: Envisioning Work and Technology in System Development* (p. 3), John Wiley & Sons: New York, NY.

Task analysis consists of analyzing observation and interview data collected during field research and noting what event triggers an activity, what discrete steps an activity has, and what outcome an activity produces, including subsequent tasks. Task analysis also notes decision points and branches, any dependencies in the process (other, outside information or events that are necessary to continue the activity) and other activities that may depend on the one under examination.

Task analysis may contain a varying degree of detail, depending on the needs of the project. Analysis may be coarse-grained, only breaking down a very large process into individual activities to examine task flow, or very fine-grained, examining individual mouse clicks.

Personas

A persona is a fictional user created to represent a user group^{13,14}. The development team creates this user as an aggregate of the real users they have observed and interviewed. A persona can be *more typical* than any real individual. A persona is a profile of this supertypical user with a name, photo, likes and dislikes, habits, background and expectations, and any other information that will help the development team identify with the user. Most importantly, personas list key goals for the user. Personas help avoid designing for one's self, or designing for needs that don't really exist. Design decisions can be made by answering the question "What would Kim [our persona] feel comfortable with in this situation?" instead of advancing a personal agenda. By having personas as references, consistency through the project is increased as the development team is focused on meeting the same user goals at the same level of expertise and experience.

Presentation and Analysis Origins

Storyboards and scenarios were used in film and theatre prior to being adopted as a tool in software development. Just as a scene could quickly be re-envisioned on the set, these tools allow teams to test alternate work flows to find the best solutions.

Task analysis has a deep history in human factors – in fact, task analysis might be the basis for the first human factors work, such as Frederick Taylor and the Gilbreth's time and motion studies in the early 20th century. More recently, Card, Moran and Newell's GOMS methodology is a keystroke-level analysis of users' work with computers¹⁵.

¹³ Cooper, A. (1999) *The Inmates Are Running the Asylum : Why High Tech Products Drive Us Crazy and How To Restore The Sanity*, SAMS: Indianapolis, IN.

¹⁴ Goodwin, K. (2000) "Taking Control Back From the Inmates" in *User Interface 2000 Proceedings*, User Interface Engineering: Boston, MA.

¹⁵ Card, S. K., Moran, T., & Newell, A. (1987) "The keystroke-level model for user performance time with interactive systems." In *Readings in Human-Computer Interaction: A Multidisciplinary Approach*. Baecker, R. M. & Buxton, W. A. S. (Eds.) (pp. 192-206) Morgan Kaufman: San Mateo, CA.

Personas were originally created by Alan Cooper, but draw from previous work in user profiling through developing mental models¹⁶ and even further back to Carl Jung's notion of archetypes.

Week Four

Design discipline survey

Design is a vast topic, and many other good resources exist to support it. This course surveys many design disciplines¹⁷, but does not examine any in detail.

Conceptual Modeling

Conceptual Modeling is the creation of the “big picture” of the system. It is an explicit acknowledgement of Donald Norman's notion of a Design Model¹⁸ and attempts to create a design model informed by real understanding of users – understanding achieved in the previous stage of the UCD process. Conceptual modeling deals with creating the vision for the project that will be translated into the product itself and communicate with the user what the product is all about. At a simple level, some would say this is metaphor creation. However, metaphors always break down (eventually) and limit the use of the product. So conceptual modeling is creating a “big picture” without a complete reliance on metaphor.

Information Architecture

Information Architecture is the arrangement, grouping, and labeling of content in a project. Information architecture determines the site or product structure in order to facilitate easy access to content, and is essential in developing good search tools. Richard Saul Wurman, an architect and graphic designer coined “Information architecture” in 1975¹⁹. Information architecture also has strong ties to library and information science²⁰. Note that someone with the job title “Information Architect” may have many responsibilities outside of practicing Information Architecture as described here.

Interaction Design

Interaction Design is the development of interaction between the user and the product, the flow of activity and the functionality that supports that flow. Interaction design overlaps the activities of information architecture in that both are concerned with the flow through the product – though information architecture focuses on the flow of content, and interaction design focuses on the flow of functionality, the two are entwined. If a product

¹⁶ Norman, D. A. (1987) “Some Observations on Mental Models.” In *Readings in Human-Computer Interaction: A Multidisciplinary Approach*. Baecker, R. M. & Buxton, W. A. S. (Eds.) (pp. 192-206) Morgan Kaufman: San Mateo, CA.

¹⁷ See Jesse James Garrett's survey of disciplines in *The Elements of User Experience* (2000). url: <http://www.jjg.net/ia/elements.pdf>

¹⁸ Norman, D. A. (1988) *The Design of Everyday Things* (pp. 12-17) Doubleday: Toronto, ON.

¹⁹ Richard Saul Wurman Interview url: <http://www.frontwheeldrive.com/wurman.html>

²⁰ Rosenfeld, L. & Morville, P. (1998) *Information Architecture for the World Wide Web*, O'Reilly: Cambridge, MA.

is primarily an information source offering content, then information architecture will have a strong impact on interaction design. If a product is a productivity application offering functionality, then interaction design will inform the information architecture.

Information Design

Information design is used in the sense of Edward Tufte's work²¹ on designing information presentation and layout to facilitate understanding²². Some people use information design to mean the same thing as information architecture, but the separation between organization and visual presentation of information is useful. However, there is strong overlap between the two practices. Information design is about communication, and is often taught in graphic design programs.

Visual Design

Visual Design is the visual treatment of the interface such as color schemes. Note that Information Design is significantly concerned with layout, so in this sense visual design is more the final cosmetic finish on a product. A visual designer will often be responsible for information design and visual design.

Card Sorting – A Conceptual Modeling and Information Architecture tool

“Card Sorting is a technique for exploring how people group items, so that you can develop structures that maximize the probability of users being able to find items.”²³ Card sorting involves having a representative user group index cards printed with labels or sample content into groups. These groupings reveal how users think about content – part of their conceptual model – and also inform the information architecture for the project. Card sorting can be analyzed using cluster analysis or simply manually inspected for trends if the number of items is low enough.

Card sorting originated with P-sorts and Q-sorts and other grouping experimental methodologies²⁴. Since it only requires index cards it is an easily available method for early design efforts.

Low-fidelity Prototyping

Prototypes represent the product. A high-fidelity prototype may look just like the finished product, while a low-fidelity prototype is usually a paper mockup. Paper prototypes are low cost, collaborative, easier to criticize, easier to make changes, and focus on the experience instead of the technology²⁵. Paper prototypes also provide an easy way to generate many design alternatives. Several good resources detail how to create a paper

²¹ Tufte, E. (1992) *The Visual Display of Quantitative Information*, Graphics Press: Cambridge, MA.

²² Garrett, J. J. (2000) *The Elements of User Experience*, url: <http://www.jjg.net/ia/elements.pdf>

²³ InfoDesign (2000) *Card Sorting*, url: <http://www.infodesign.com.au/usability/cardsorting.html>

²⁴ Cook, N. J. (1994) “Varieties of Knowledge Elicitation Techniques”, *International Journal of Human-Computer Studies*, **41** pp. 801-849.

²⁵ Klee, M. (2000) “Five Paper Prototyping Tips” *User Interface Engineering*. url: <http://world.std.com/~uieweb/paperproto.htm>

prototype^{26,27}, and also the rationale for paper prototyping²⁸. A prototype may be horizontal, covering the breadth of content, but not contain any depth; or it may be vertical, fully exploring one content branch or function, but not representing other content or functions²⁹. In class, paper prototyping allowed students to participate in hands-on user-centered design activities without a significant amount of training or materials.

Week Five

Usability Inspections

Usability is a primary goal of UCD, and consists of the following five components: ease of learning, efficiency, low error rates, easy to remember how to use (memorability), and subjectively pleasing³⁰. Generally one aspect will dominate the others – ATMs need to be easy to learn to use for first time “walk up and use” customers, while air traffic control needs extremely low error rates.

One way of evaluating the usability of an application is through a usability inspection – an expert review of the interface for usability problems. Two kinds of inspection are most common: the cognitive walkthrough and heuristic evaluation.

Cognitive Walkthrough

The cognitive walkthrough is a structured walkthrough of an interface^{31,32}. The evaluator works with a prototype and persona and walks through a scenario. At each step of an activity the evaluator asks the following four questions:

1. Does the user understand what action they should be trying to accomplish right now?
2. Is the interaction object visible to the user?
3. Does the user understand the connection between the desired action and the interaction object – do they recognize that the interaction object is the correct one to accomplish the desired action?
4. Does the interface present sufficient feedback to indicate success and to provide the user with understanding of the next appropriate action?

The evaluator tells a believable story of the application’s use when justifying each step by answering the questions. If the evaluator cannot reasonably describe how the persona

²⁶ Rettig, Marc (1994) “Prototyping for tiny fingers”, *Communications of the ACM* 37:4 pp. 21-27

²⁷ Hackos, J. T. & Redish, J. C (1998) *User and Task Analysis for Interface Design*, (pp. 375-404) John Wiley & Sons: Toronto, ON.

²⁸ Muller, M.J. (1991) “PICTIVE: An exploration in participatory design.” in *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp.225-231, ACM Press.

²⁹ Nielsen, J. (1987) “Using Scenarios to Develop User Friendly Videotex Systems.” in *Proceedings of NordData87, Joint Scandinavian Computer Conference*.

³⁰ Nielsen, J. (1994) *Usability Engineering* (p. 26) AP Professional: Boston, MA.

³¹ Lewis, C. & Rieman, J. (1993) *Task-Centered User Interface Design*. (Ch. 4) url: <http://home.att.net/~jrieman/jrtcdbk.html>

³² Wharton, C., Rieman, J., Lewis, C. & Polson, P. (1994) “The Cognitive Walkthrough Method: A Practitioner’s Guide” in Nielsen, J. & Mack, R. (Eds.) *Usability Inspection Methods* (pp. 105-140), John Wiley & Sons: New York, NY.

would be able to know the information in each step an issue with the interface is identified, though analysis continues as though the persona *could* successfully complete the problem step in order to detect potential problems with later stages of the activity.

Heuristic Evaluation

Heuristic evaluation is expert review of an interface against a short list of guidelines or heuristics^{33,34}. The evaluator works through an interface following a scenario, and identifies any usability issues and categorizes them according to the heuristics. If several evaluators all separately assess an interface using the same heuristics, they will find a majority of significant usability problems. Jakob Nielsen, who developed the method with Rolf Molich, has ten recommended heuristics for software evaluation³⁵. Note that these heuristics may not be suitable for some online applications.

If the development team has expertise in usability or UCD, then usability inspections provide an economical way to quickly evaluate an interface.

Week Six

Usability Testing

Usability testing is having real users perform real tasks and observing where they encounter difficulties with the interface³⁶. This methodology has its beginning in cognitive psychology studies of perception and cognition. Usability testing can occur in a formal lab with the user in a separate room observed through a one-way mirror and monitored by multiple video feeds from remote control cameras. This type of testing can produce very exact results. However, since the goal of usability testing is not to achieve statistically significant results, but to improve the usability of the product, these rigorous methods are not necessary. Instead, development teams can conduct discount usability tests, where a moderator simply sits with a user and asks them to use a prototype. This type of testing will find a significant number of usability issues with a substantially lower cost.

In both formal and discount settings, users may be asked to think out loud to inform observers why he or she is taking that particular action. Users may also respond to pre- and post-test questionnaires or other subjective measures.

Usability evaluation provides the connection between the user model and the design model – the design team is able to see how the system image supports an accurate user model and refine the product to increase user success and satisfaction and improve the user experience as a whole.

³³ Nielsen, J. (1994) “Heuristic Evaluation” in Nielsen, J. & Mack, R. (Eds.) *Usability Inspection Methods* (pp. 25-62), John Wiley & Sons: New York, NY.

³⁴ Nielsen, J. (1994) *Usability Engineering* (pp. 115-163), AP Professional: Boston, MA.

³⁵ See Appendix for a list of the heuristics

³⁶ Rubin, J. (1994) *The Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests* (p. 27-30), John Wiley & Sons: New York, NY.

Conclusions

User-Centered Design is a valuable practice in developing products. By focusing on users and their goals development teams can realize significantly more success³⁷. Providing education to create a skill base of UCD practitioners is an important part of implementing UCD in the industry. These students will have a better appreciation for the skills they learn with an understanding of the source of the techniques. This can also help them to validate UCD activities with others.

Going forward the practices outlined in this paper will give rise to new methods and techniques suited for new technical and social contexts, such as broadband or wireless. With an appreciation of the past, there is a strong foundation for that future, a future where technology fulfills user needs simply, elegantly, and with flair.

³⁷ Landauer, T. K. (1995) *The Trouble With Computers: Usefulness, Usability, and Productivity*. MIT Press: Cambridge, MA.

Acknowledgements

I would like to thank Dr. Karsten Loepelmann, who supervised this project with a light touch and treatment as a peer, trusting that things that need to be done would be done. I am indebted to Blaine Bertsch, the coordinator of the GMC New Media Outreach program for inviting me to teach the course. Once the course began the staff of the Jasper Place campus were always helpful, with particular credit to the assistance of Margo and Vern. I would also like to thank the students in the class, who all attended faithfully and were supportive and understanding of an instructor teaching the material for the first time. Finally, I would like to thank most of all my wife Heather for the sacrifices she has made as I have devoted time and energy to this project.

Appendix

Ten Usability Heuristics³⁸

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.

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.

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.

³⁸ Nielsen, J. (1994) "Heuristic Evaluation" in Nielsen, J. & Mack, R. (Eds.) *Usability Inspection Methods* (pp. 25-62), John Wiley & Sons: New York, NY.