

Human Factors and Usability in User Interface Design

Jim W. Berney

BellSouth Telecommunications, Inc.
Information Systems - Presentation Services
675 W. Peachtree Street
Room 13L71 SBC
Atlanta, Georgia 30375
404-529-0346

ABSTRACT

Over the past several years, the focus of software design has evolved to better ensure that user needs are met and that systems are more intuitive and user friendly. Software designers are turning towards human factors workers to address ergonomics and issues regarding human memory limitations, ability to learn, and better understanding user needs. This evolution is a shift from systems oriented design to User Centered Design (UCD). This paper provides an overview of how human factors can enhance design and how users play an important role in defining and verifying product design early and throughout the development process.

1. Common Pitfalls in Software Design

Some of the common software design myths and misconceptions discussed below have initiated a focus back to the user. Human factors and user centered design techniques offer design solutions that emphasize a focus on early and continuous user participation. These misconceptions must be addressed up-front and buy-in for human factors must be obtained from the development team in order to ensure a successful project.

- Users don't know what they want - Usually, users know exactly what they want, however its your challenge to figure it out. Thorough user interviews and careful analysis of the tasks performed by the user are good first steps in product design.
- We don't have time to talk to our users - You actually don't have time not to talk to your users. Not only talk to users and spend time on-site learning their job, but make them a part of the design team.
- We have too many user groups to talk to them all - You should take the time to identify and understand both the primary and secondary user communities before attempting product design.
- We know what the users really want - "Know thy user" is becoming a common phrase and simply means; understand all characteristics of your user community, the tasks and job to be performed and how the product will be used in the field.
- We can do it right the first time - Design is an iterative process that is enhanced and refined with each iteration. Rarely is design perfect or optimal the first time through.
- That's not a problem, since the user has not been trained - Theoretically, if the product is intuitive and easy to use, training should either not be necessary or minimal. Training should not compensate for poor or insufficient design.

2. Common System Problems

Designers must accept that product design is iterative and that multiple design iterations should be planned into the development schedule. Often tight schedules do not allow development of working prototypes to support iterative design and testing. Tight schedules can also cause user testing to be postponed or completely eliminated until after deployment.

Systems often require the user to remember and process too much information. Cryptic codes, unnecessary information, unneeded process steps, and unfamiliar terms are typical examples of how the user's working memory and ability to complete a task is burdened.

Systems are often not tolerant to minor input errors. Design systems to avoid user errors or at least to detect and allow correction. Do not force the user to reenter information, restart a task, or recover/back-out data. Messages should also be informative and in a

language natural to the user.

Systems where a careful task analysis was not performed, often do not provide the needed information and do not display data in a familiar and natural format. Systems may also lack needed functionality, force users to perform tasks unnaturally, and cause unwanted changes to the user's organizational structure and practices.

3. Human Factors

Human factors is the study of human behavior, cognition, and ability. Research performed in human factors such as learning, memory recall, building system metaphors, consistency, and effects of system feedback are transferred into product design. Human factors techniques can be used to profile the user community, build product requirements, and ensure that valid data is collected and accurately interpreted. Often human factors research is delivered as design guidelines, style guides, and conformance check lists of do's and don'ts in interface design. The most effective way to incorporate human factors is to include a human factors worker(s) on the design team to assist with user interviews, requirements gathering, and user interface design. This person should be involved end-to-end on the project and participate in every design, review, and test session. Central human factors groups lack the advantage of being part of the team and thus are difficult to involve midstream in the project.

4. User Centered Design Methods

User Centered Design (UCD) methods focus on early and continuous user involvement in product design. UCD is an iterative approach to define/verify user requirements and constraints, design the user interface, evaluate the design using "real users", and then feed the evaluation results back into the product design. Below are the principle techniques of User Centered Design:

4.1 User Characterization

Only the actual end user can tell you if the product is useful and meets their needs and expectations. Human factors workers use the phrase, "Know thy user!" to emphasize the need to perform early user interviews and job-site visits to accurately determine product requirements. Early user involvement in requirements specification and product design helps to ensure that product needs are not misinterpreted or specified incorrectly by a non-user.

Both the primary and secondary user communities are identified during this phase. Secondary users are usually occasional users and may include users from other departments, maintenance technicians, and training course developers. Avoid stereotypes and do not assume you understand all your users without thorough interviews and observation of them using existing or similar products. User education, experience, special skills, and physical handicaps all induce diversity within a user group.

The user's job description, turnover rate, and their organization's reward structure are important. For example, user groups with a high turnover rate need easy to learn, intuitive interfaces that require minimal training. If the users receive bonus or commission as compensation, the product must not inhibit their ability to do their job and meet their goals.

It is important to note whether use of your product is voluntary or mandatory. If the system is not "mission critical" or part of the user's job requirement, other products may be substituted if your design does not meet their needs. Motivation to use the system also has an important impact on the user's tolerance to poor design.

4.2 Task Analysis

Whereas there are formal methods to identify actions, data, and steps to perform a task, most task analysis is performed by observing and interviewing users, then recording requirements for each task. Below are examples of the types of information that should be gathered for a task:

- data required to start and complete a task; data generated by the task
- task decision points and data needed to support the decision
- dependencies on other tasks
- frequency and importance of the task
- physical, cognitive, and environmental demands on the user
- speed and accuracy in which the user must complete the task
- can the user control the pace and priority of the task?

4.3 Situational Analysis

During situational analysis, you should try to identify situations that may arise after the software is deployed that could affect your design. Discovering these situations upfront, allow you to modify your design to avoid making expensive changes once the user receives your product. For example:

- equipment selected by the user doesn't work as expected or performance is poor

- functionality or data needed to complete a task is not available or difficult to access
- user is confronted with too much information to remember or to process
- interruptions (e.g., poor response time, system failure, interruptions from other users) make it difficult to complete tasks
- product induces unsolicited changes in the user's environment (e.g., reorganization, new policies, changes in the customer base) causing a negative reaction to your product.

4.4 Usability Evaluation

Throughout the development process, the interface should be evaluated by real users to determine its usability and to ensure that needed functionality is accessible. Early when determining system requirements, objectives for usability should be defined. Usability objectives identify at what point the system is usable and acceptable by the user. Length of time to complete a task, training time, system performance, and number of errors while performing a task are examples of usability objectives. Product design is iteratively evaluated and enhanced until the usability objectives are met.

4.5 Early Prototyping

The design team should transform paper and pencil sketches of the interface into a working prototype early in design. Prototypes are valuable tools to convey design and can be shown to users to verify requirements. Whereas prototypes should accurately reflect the deployment environment, it is not critical that the exact hardware/software environment be used. For example, the OSF (Open Software Foundation) Motif and Microsoft Windows GUIs (Graphical User Interface) are similar environments. User Interface Management Systems (CASE tools that simplify GUI design and code generation) can be effective prototyping tools. Draw and paint packages can also be used by non-programmers to illustrate design.

5. Participatory Design

Participatory design is a User Centered Design technique that brings together users, developers, human factors workers, etc. to design the user interface. Bellcore has developed a technique called PICTIVE (Plastic Interface for Collaborative Technology Initiatives through Video Exploration). PICTIVE is a non-software based prototyping tool that uses readily available office supplies (e.g., Post-It (TM) Notes, markers, colored paper) to build interface objects. These objects can be icons, common dialog windows, or any GUI component used in the design. Easy to manipulate paper objects remove

system imposed barriers and position all members of the design team as "peer designers". Each member contributes their area of expertise or experience. PICTIVE is most productive if implemented after initial requirements analysis is complete. PICTIVE can also be used

to verify or iteratively enhance existing interface designs. PICTIVE design sessions can be video taped for later review or interpretation.

6. Usability Evaluation

6.1 Overview

Usability is the evaluation of a product by actual users to assess its ease of use, ability to do the job and meet user expectations, and its overall acceptability. Usability does not test users, rather users evaluate your product. Usability helps verify requirements and constraints and identify design faults, successes, and missing product functionality. Iterative evaluations provide the feedback needed to enhance your prototype or product until it is acceptable and meets usability/product objectives. Usability can improve productivity by eliminating unnecessary steps, improving task flow, and by ensuring that all information needed to complete the task is provided.

Whether or not usability is ever performed, the product will be evaluated as soon as it is released to your user/customer. The objective is to evaluate early in the development cycle, when fixes are less expensive and user acceptance and perception of the product is not affected. In other words, do not let your customer or competition tell you that your product does not meet their needs and that it is difficult to use.

One of my favorite usability examples (or lack of) is the two dollar bill. The US Treasury commissioned an artist to engrave the most elaborate plates ever produced for US currency. When the two dollar bill was released, no one used it, why? The simple answer was that there was no room in the cash drawer. Customers gave the two dollar bill to the clerk, the clerk stuffed the bill under the cash drawer, and the bill was sent back to the bank, never being recirculated. This is an example of a good idea that lacked user buy-in and acceptance. The public had not asked for the two dollar bill, and there was no immediate need from the public's perspective. Therefore, the public's motivation to use the bill was very low, eliminating their incentive to overcome the cash drawer problem. The current denominations were sufficient, so why use the new bill?

6.2 Benefits of Usability

Usability benefits the project team, the developer, and the user/customer. The user develops a feeling of ownership from participating in design and usually forms a positive

attitude toward your product. They feel that "the developers are really listening to us and our needs". For the project team, usability is a team builder that opens communications between project team members. This is especially valuable when large teams are divided between cities and departments.

Designers and developers also benefit since usability can improve the user's buy-in and acceptance of the product. Usability will also identify missing product functionality, design flaws, and identify redundant, unclear, or poorly designed tasks. Users almost always give unexpected reactions and perspectives about your product. There is nothing more enlightening than watching a real user try out your design. Product faults and user reactions are usually obvious and are witnessed by everyone observing the evaluation.

Usability can reduce training by making design more intuitive and by ensuring that terminology and tasks are within the user's domain. Usability can also help identify where either too much or too little training is provided. The design team should avoid "Blame and Train", where the user is blamed for not knowing how to use the system, then provided more training as compensation. The team should not label usability problems as training issues until design fixes and alternatives are explored. Training should be the last alternative. Often the longest sections in the training manual correspond to the most poorly designed tasks.

6.3 Usability Evaluation Techniques

Every component of your product should be evaluated. Typical components include the user interface, training, installation/configuration, on-line help/documentation, paper manuals, and third-party products used in conjunction with your product. The first step in usability is to form a usability team. This team typically consists of users, developers, designers, training, and any other key design team members. This team coordinates the evaluation, determines what will be evaluated, and analyzes the results.

Always assume that multiple evaluations will be needed for each product component or module and that changes will be made. Each iteration of usability reveals an additional level of design issues and problems. Only the obvious problems are found during the first evaluation, such as unclear wording, poorly written messages, and unintuitive or unnatural design. More complex process problems are not revealed until the second or third evaluation. Design changes made as a result of usability should be verified before moving on to another module.

It is critical that "real" users evaluate "real" tasks. Never substitute individuals who represent the user or someone from the project team. Task scenarios should be prepared that represent real-world tasks routinely performed by the user.

Simulate the user's actual environment as closely as possible. Their work space, type of workstation, keyboard, mouse, lighting, noise levels, and interruptions should all be considered and built into the evaluation. A telephone should be provided if the user must talk to a customer or consult with another organization to complete the scenario. Usability results may be invalid or misinterpreted if the environment is not accurately simulated.

User observation and statistical measurement are the two most frequently used methods to evaluate usability. Users are observed while performing task scenarios typical in their job. Measurements can supplement observation by building data collection into the software to record events such as the number of errors, help access, or time to complete a task. Measurements may also be made during the evaluation by observers or after the evaluation by reviewing the video. The usability process also helps ensure that valid data is collected by providing a more controlled environment. This is in contrast to casual interviews between developers and users to review design.

Usability evaluations can be either "informal" or "formal". Informal evaluations are one on one sessions between the user and the designer, where the user may be asked to review and comment on design or to complete task scenarios. Informal evaluations are most appropriate to iteratively evaluate and refine individual interfaces and product components. Formal evaluations are much more structured and resource intensive. They are most appropriate to verify end-to-end design of a prototype, principle product modules, or the entire product. Formal evaluations are usually videotaped and project team members are invited to observe the evaluation.

As the user works through each scenario, they are asked to "talk aloud" so that the usability team can follow the user's thought process and encourage comments. Following each scenario, the user is debriefed and asked to expand on problems and comment on overall design and ease of use. Post-evaluation questionnaires can also be used to obtain impressions and perceptions about the product design.

7. Usability Experiences

7.1 Obtaining Valid User Feedback

It is sometimes difficult to ensure that you are obtaining valid data on which to base design changes. You must always focus on the product objectives, evaluation objectives, user profiles, simulation of the user's environment, and the task scenarios to ensure that data is valid and interpreted correctly.

Users of critical mission products use the product daily (frequent users) and are considered experienced or experts in their job. When building such a system, there are no frequent users to perform the evaluation. Often up-front intuitiveness is given too much attention when actually task functionality, expert modes, and short cuts should be evaluated. Diversity within user groups also make it more difficult to find users with the exact job experience needed for the evaluation. Other constraints such as travel, time away from the job, and relying on a staff person or supervisor to select your evaluators make it difficult to obtain users that match the user profile established for your product.

7.2 Training

Deciding how much training should be provided prior to an evaluation is difficult and should be based on the objectives established for the evaluation. If your objective is to measure how intuitive the product is for first time users, then no training is provided. However, if you intend to measure product usability for a frequent, expert user, you must bring your user to the level of a typical user before starting the evaluation. Both training and practice with the product is needed before you begin. If you are developing training for your product (e.g., on-line, self paced), let the user evaluate the training, then perform the task scenarios for usability. If the system is changing the user's environment (e.g., 3270 to graphical windowing), you must first train the evaluators on the new environment so that it does not interfere with your evaluation. Focus on the profile of the "actual user in the field" in order to decide if a problem is caused by a lack of training or if it is a design problem.

7.3 Usability Team

It is critical that your usability team contain key members of the design team. Whereas everyone is busy with design and development, members of the usability team should be responsible for design and empowered to make changes. Remember, that one of the up-front rules of usability is to "be prepared to make change". Finally, design changes recommended during an evaluation should be a "team" decision. One member of the usability team should not influence everyone else. Everyone should leave the evaluation with a list of changes and design issues that need to be addressed.

7.4 Usability Coordinator

Rather than coordinating usability through a central human factors group, each project team or department should have a usability champion or coordinator. This person is closer to the work and is responsible for obtaining users and maintaining buy-in for

usability, especially when product schedules get tight. The central human factors group can act as consultants that help facilitate key evaluations, analyze usability results, and maintain and enhance the usability process.

7.5 Executive Sponsorship for Usability

It is important to get an "executive sponsor" to support usability within a project and within your company. This person should have overall responsibility for the project and can postpone deployment if the product is not ready. The sponsor also sets an acceptance level and priority for usability within the project. They ensure that problems are fixed and the product is not released with major usability problems.

7.6 Using Video in Usability

Video is often used in formal usability evaluations to record events for later review. Video is also a large part of the expense of building a usability lab. We have found the value of video to be questionable except in the following cases:

- Video allows project team members not present during the evaluation to see problems first hand. However, time must be set aside to go through many hours of tape. An edited version is often all that is needed.
- Video allows everyone involved in the evaluation and observing to watch the user without being in the same room. Typically the user is in a separate room with the "usability team" and observers in separate rooms.
- A 10 minute summary tape is valuable to show management and project team members that could not directly observe. The summary should contain clips that represent the tone of the evaluation and typical problems encountered.

8. Summary

Usability is a critical component of product design and should not be considered as "kinda nice". Every product will be evaluated at sometime. Evaluate early and often while its less expensive to make design changes. Real users need to evaluate your product, not substitutes from the design or project team. Avoid "blame and train" during product design and evaluation. Training should not compensate for poor design. If the task is hard to train, the design team most likely failed to understand the task. Tasks that are natural to the user should not require a lot of training.

Usability should be better integrated into the product design process, not an after thought when time is available or when a design failure is obvious. Where CASE tools are used, integrate User Centered Design and usability into the CASE methodology. James Martin emphasizes the need to evaluate the prototype, transactions, and the integrated system for usability as a way to "reduce" development time and enhance design.

Finally, those who are unsure about the value of User Centered Design and usability should watch an evaluation or design session in process. It is a sobering experience to watch a user totally unable to use your design because it did not match their needs or background.

REFERENCES

1. Paul Booth - "An Introduction to Human-Computer Interaction"
2. Michael J. Muller - "The PICTIVE Approach to Participatory Design", Proceedings of the Bellcore/BCC Symposium on User Centered Design, SR-OPT-022130, November 4-6, 1991