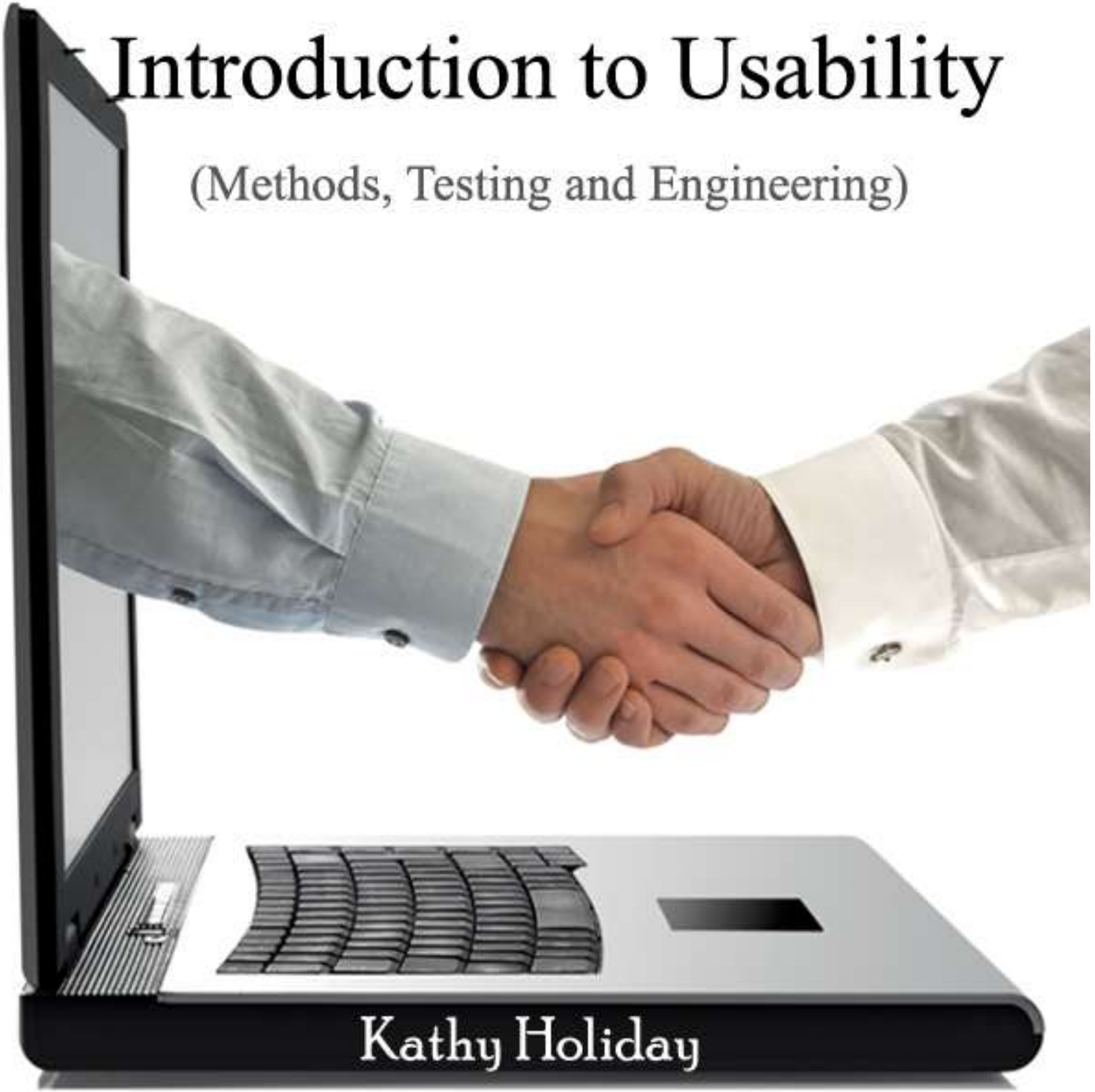
Introduction to Usability

(Methods, Testing and Engineering)



Kathy Holiday

First Edition, 2012

ISBN 978-81-323-1590-2

Published by:
Learning Press
4735/22 Prakashdeep Bldg,
Ansari Road, Darya Ganj,
Delhi - 110002
Email: info@wtbooks.com

Table of Contents

Chapter- 1

Introduction to Usability

In design, **usability** is the study of the ease with which people can employ a particular tool or other human-made object in order to achieve a particular goal. This can include endeavors as varied as consumer electronics, communication, and knowledge transfer objects (such as a cookbook, a document or online help) and mechanical objects such as a door handles or a hammer.

Usability includes the methods of measuring usability and the study of the principles behind an object's perceived efficiency or elegance.

In human-computer interaction and computer science, usability studies the elegance and clarity with which the interaction with a computer program or a web site (web usability) is designed.

Usability differs from user satisfaction insofar as the former also embraces usefulness.

Introduction

The primary notion of usability is that an object designed with a generalized users' psychology and physiology in mind is, for example:

- More efficient to use — it takes less time to accomplish a particular task
- Easier to learn — operation can be learned by observing the object
- More satisfying to use

Complex computer systems are finding their way into everyday life, and at the same time the market is becoming saturated with competing brands. This has led to usability becoming more popular and widely recognized in recent years as companies see the benefits of researching and developing their products with user-oriented instead of technology-oriented methods. By understanding and researching the interaction between product and user, the *usability expert* can also provide insight that is unattainable by traditional company-oriented market research. For example, after observing and interviewing users, the usability expert may identify needed functionality or design flaws that were not anticipated. A method called "contextual inquiry" does this in the naturally occurring context of the users own environment.

In the user-centered design paradigm, the product is designed with its intended users in mind at all times. In the user-driven or participatory design paradigm, some of the users become actual or de facto members of the design team.

The term *user friendly* is often used as a synonym for *usable*, though it may also refer to accessibility. Usability is also used to describe the quality of user experience across websites, software, products and environments.

There is no consensus about the relation of the terms ergonomics (or human factors) and usability. Some think of usability as the software specialization of the larger topic of ergonomics. Others view these topics as tangential, with ergonomics focusing on physiological matters (e.g., turning a door handle) and usability focusing on psychological matters (e.g., recognizing that a door can be opened by turning its handle).

Usability is also very important in website development (web usability). According to Jakob Nielsen, "Studies of user behavior on the Web find a low tolerance for difficult designs or slow sites. People don't want to wait. And they don't want to learn how to use a home page. There's no such thing as a training class or a manual for a Web site. People have to be able to grasp the functioning of the site immediately after scanning the home page—for a few seconds at most." Otherwise, most casual users will simply leave the site and continue browsing—or shopping—somewhere else.

Definition

ISO defines usability as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." Usability is a qualitative attribute that assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease-of-use during the design process. Usability consultant Jakob Nielsen and computer science professor Ben Shneiderman have written (separately) about a framework of system acceptability, where usability is a part of "usefulness" and is composed of:

- Learnability: How easy is it for users to accomplish basic tasks the first time they encounter the design?
- Efficiency: Once users have learned the design, how quickly can they perform tasks?
- Memorability: When users return to the design after a period of not using it, how easily can they re establish proficiency?
- Errors: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- Satisfaction: How pleasant is it to use the design?

Usability is often associated with the functionalities of the product (cf. ISO definition, below), in addition to being solely a characteristic of the user interface (cf. framework of system acceptability, also below, which separates *usefulness* into *utility* and *usability*). For example, in the context of mainstream consumer products, an automobile lacking a

reverse gear could be considered *unusable* according to the former view, and *lacking in utility* according to the latter view.

When evaluating user interfaces for usability, the definition can be as simple as "the perception of a target user of the effectiveness (fit for purpose) and efficiency (work or time required to use) of the Interface". Each component may be measured subjectively against criteria e.g. Principles of User Interface Design, to provide a metric, often expressed as a percentage.

It is important to distinguish between usability testing and usability engineering. Usability testing is the measurement of ease of use of a product or piece of software. In contrast, usability engineering (UE) is the research and design process that ensures a product with good usability.

Usability is an example of a non-functional requirement. As with other non-functional requirements, usability cannot be directly measured but must be quantified by means of indirect measures or attributes such as, for example, the number of reported problems with ease-of-use of a system.

Intuitive interfaces

The term intuitive is often listed as a desirable trait in usable interfaces, often used as a synonym for learnable. Some experts such as Jef Raskin have discouraged using this term in user interface design, claiming that easy to use interfaces are often easy because of the user's exposure to previous similar systems, thus the term 'familiar' should be preferred. As an example: Two vertical lines "||" on media player buttons do not intuitively mean "pause" - they do so by convention. Aiming for "intuitive" interfaces (based on reusing existing skills with interaction systems) could lead designers to discard a better design solution only because it would require a novel approach. This position is sometimes illustrated with the remark that "the only intuitive interface is the nipple; everything else is learned."

Bruce Tognazzini even denies the existence of "intuitive" interfaces, since such interfaces must be able to intuit, i.e. "perceive the patterns of the user's behavior and draw inferences." Instead, he advocates the term "intuitable", i.e. "that users could intuit the workings of an application by seeing it and using it." He continues, however, "But even that is a less than useful goal since only 25 percent of the population depends on intuition to perceive anything."

Investigation

The key principle for maximizing usability is to employ iterative design, which progressively refines the design through evaluation from the early stages of design. The evaluation steps enable the designers and developers to incorporate user and client feedback until the system reaches an acceptable level of usability.

The preferred method for ensuring usability is to test actual users on a working system. Although, there are many methods for studying usability, the most basic and useful is user testing, which has three components:

- Get some representative users.
- Ask the users to perform representative tasks with the design.
- Observe what the users do, where they succeed, and where they have difficulties with the user interface.

It's important to test users individually and let them solve any problems on their own. If you help them or direct their attention to any particular part of the screen, you will bias the test. Rather than running a big, expensive study, it's better to run many small tests and revise the design between each one so you can fix the usability flaws as you identify them. Iterative design is the best way to increase the quality of user experience. The more versions and interface ideas you test with users, the better.

Usability plays a role in each stage of the design process. The resulting need for multiple studies is one reason to make individual studies fast and cheap, and to perform usability testing early in the design process. Here are the main steps:

- Before starting the new design, test the old design to identify the good parts that you should keep or emphasize, and the bad parts that give users trouble.
- Test competitors' designs to get data on a range of alternative designs.
- Conduct a field study to see how users behave in their natural habitat.
- Make paper prototypes of one or more new design ideas and test them. The less time you invest in these design ideas the better, because you'll need to change them all based on the test results.
- Refine the design ideas that test best through multiple iterations, gradually moving from low-fidelity prototyping to high-fidelity representations that run on the computer. Test each iteration.
- Inspect the design relative to established usability guidelines, whether from your own earlier studies or published research.
- Once you decide on and implement the final design, test it again. Subtle usability problems always creep in during implementation.

Don't defer user testing until you have a fully implemented design. If you do, it will be impossible to fix the vast majority of the critical usability problems that the test uncovers. Many of these problems are likely to be structural, and fixing them would require major rearchitecting. The only way to a high-quality user experience is to start user testing early in the design process and to keep testing every step of the way.

ISO standards

ISO/TR 16982:2002

ISO/TR 16982:2002 ("Ergonomics of human-system interaction -- Usability methods supporting human-centered design") is a standard providing information on human-centred usability methods which can be used for design and evaluation. It details the advantages, disadvantages and other factors relevant to using each usability method.

It explains the implications of the stage of the life cycle and the individual project characteristics for the selection of usability methods and provides examples of usability methods in context.

The main users of ISO/TR 16982:2002 will be project managers. It therefore addresses technical human factors and ergonomics issues only to the extent necessary to allow managers to understand their relevance and importance in the design process as a whole.

The guidance in ISO/TR 16982:2002 can be tailored for specific design situations by using the lists of issues characterizing the context of use of the product to be delivered. Selection of appropriate usability methods should also take account of the relevant life-cycle process.

ISO/TR 16982:2002 is restricted to methods that are widely used by usability specialists and project managers.

It does *not* specify the details of how to implement or carry out the usability methods described.

ISO 9241

ISO 9241 is a multi-part standard covering a number of aspects for people working with computers. Although originally titled Ergonomic requirements for office work with visual display terminals (VDTs) it is being retitled to the more generic Ergonomics of Human System Interaction by ISO. As part of this change, ISO is renumbering the standard so that it can include many more topics. The first part to be renumbered was part 10 (now renumbered to part 110).

Part 1 is a general introduction to the rest of the standard. Part 2 addresses task design for working with computer systems. Parts 3–9 deal with physical characteristics of computer equipment. Parts 110 and parts 11–19 deal with usability aspects of software, including Part 110 (a general set of usability heuristics for the design of different types of dialogue) and Part 11 (general guidance on the specification and measurement of usability).

Usability considerations

Usability includes considerations such as:

- Who are the users, what do they know, and what can they learn?
- What do users want or need to do?
- What is the general background of the users?
- What is the context in which the user is working?
- What has to be left to the machine?

Answers to these can be obtained by conducting user and task analysis at the start of the project.

Other considerations

- Can users easily accomplish their intended tasks? For example, can users accomplish intended tasks at their intended speed?
- How much training do users need?
- What documentation or other supporting materials are available to help the user? Can users find the solutions they seek in these materials?
- What and how many errors do users make when interacting with the product?
- Can the user recover from errors? What do users have to do to recover from errors? Does the product help users recover from errors? For example, does software present comprehensible, informative, non-threatening error messages?
- Are there provisions for meeting the special needs of users with disabilities? (accessibility)
- Are there substantial differences between the cognitive approaches of various users that will affect the design or can a one size fits all approach be used?

Examples of ways to find answers to these and other questions are: user-focused requirements analysis, building user profiles, and usability testing.

Discoverability

Even if software is usable as per the above considerations, it may still be hard to *learn* to use. Other questions that must be asked are:

- Is the user ever expected to do something that is not obvious? (e.g. Are important features only accessible by right-clicking on a menu header, on a text box, or on an unusual GUI element?)
- Are there hints and tips and shortcuts that appear as the user is using the software?
- Should there be instructions in the manual that actually belong as contextual tips shown in the program?
- Is the user at a disadvantage for not knowing certain keyboard shortcuts? A user has the right to know all major and minor keyboard shortcuts and features of an application.

- Is the learning curve (of hints and tips) skewed towards point-and-click users rather than keyboard users?
- Are there any "hidden" or undocumented keyboard shortcuts, that would better be revealed in a "Keyboard Shortcuts" Help-Menu item? A strategy to prevent this "undocumented feature disconnect" is to automatically generate a list of keyboard shortcuts from their definitions in the code.

Lund, 1997 Usability Maxims

When evaluating the design and usability of a website, one must consider the following:

- Know the user, and YOU are not the user.
- Things that look the same should act the same.
- The information for the decision needs to be there when the decision is needed.
- Error messages should actually mean something to the user and tell the user how to fix the problem.
- Every action should have a reaction.
- Everyone makes mistakes, so every mistake should be fixable.
- Don't overload the user's buffers.
- Consistency, consistency, consistency.
- Minimize the need for a mighty memory.
- Keep it simple.
- The user should always know what is happening.
- The more you do something, the easier it should be to do.
- The user should control the system. The system should not control the user. The user is the boss and the system should show it.
- Eliminate unnecessary decisions and illuminate the rest.
- The best journey is the one with fewest steps. Shorten the distance between the user and the goal.
- The user should be able to do what the user wants to do.
- If I made an error, let me know about it before I get into REAL trouble.
- You should always know how to find out what to do next.
- The idea is to empower the user, not speed up the system.
- Things that look different should act different.

Also note that these are presented in a descending order determined by their mean rating of importance.

Designing for Usability

Any system that is designed for people should be easy to use, easy to learn, easy to remember, and helpful to users. Therefore, when designing for usability, three principles of design, as identified by John Gould and Clayton Lewis, should be followed: early focus on users and tasks, empirical measurement, and iterative design.

Early Focus on Users and Tasks

The design team should be user driven and direct contact with potential users is recommended. Several evaluation methods including personas, cognitive modeling, inspection, inquiry, prototyping, and testing methods may be used to gain an understanding of the potential users.

Usability considerations such as who the users are and their experience with similar systems must be examined. As part of understanding users, this knowledge must "be played against the tasks that the users will be expected to perform." This includes the analysis of what tasks the users will perform, which are most important, and what decisions the users will make while using your system. Designers must understand how cognitive and emotional characteristics of users will relate to a proposed system.

One way to stress the importance of these issues in the designers' minds is to use personas, which are made-up representative users. Another more expensive but more insightful way to go about it, is to have a panel of potential users work closely with the design team starting in the early formation stages.

Empirical Measurement

Two important factors of empirical measurement are stressed: testing the system early on, and testing the system on real users using behavioral measurements. This includes testing the system for both learnability and usability. It is important in this stage to use quantitative usability specs such as time and errors to complete tasks and number of users to test, as well as examine performance and attitudes of the users testing the system. Finally, "reviewing or demonstrating" a system before the user tests it can result in misleading results. The emphasis of empirical measurement is on measurement, both informal and formal, which can be carried out through a variety of evaluation methods.

Iterative Design

Iterative design is a design methodology based on a cyclic process of prototyping, testing, analyzing, and refining a product or process. Based on the results of testing the most recent iteration of a design, changes and refinements are made. This process is intended to ultimately improve the quality and functionality of a design. In iterative design, interaction with the designed system is used as a form of research for informing and evolving a project, as successive versions, or iterations of a design are implemented.

Iterative Design Process

The iterative design process may be applied throughout the new product development process. However, changes are easiest and less expensive to implement in the earliest stages of development. The first step in the iterative design process is to develop a prototype. The prototype should be evaluated by a focus group or a group not associated with the product in order to deliver non-biased opinions. Information from the focus

group should be synthesized and incorporated into the next iteration of the design. The process should be repeated until user issues have been reduced to an acceptable level.

Specific Application: Human Computer Interfaces

Iterative design is commonly used in the development of human computer interfaces. This allows designers to identify any usability issues that may arise in the user interface before it is put into wide use. Even the best usability experts cannot design perfect user interfaces in a single attempt, so a usability engineering lifecycle should be built around the concept of iteration.

The typical steps of iterative design in user interfaces are as follows:

1. Complete an initial interface design
2. Present the design to several test users
3. Note any problems had by the test user
4. Refine interface to account for/fix the problems
5. Repeat steps 2-4 until user interface problems are resolved

Iterative design in user interfaces can be implemented in many ways. One common method of using iterative design in computer software is software testing. While this includes testing the product for functionality outside of the user interface, important feedback on the interface can be gained from subject testing early versions of a program. This allows software companies to release a better quality product to the public, and prevents the need of product modification following its release.

Iterative design in online(website) interfaces is a more continuous process, as website modification, after it has been released to the user, is far more viable than in software design. Often websites use their users as test subjects for interface design, making modifications based on recommendations from visitors to their sites.

Iterative design use

Iterative design is a way of confronting the reality of unpredictable user needs and behaviors that can lead to sweeping and fundamental changes in a design. User testing will often show that even carefully evaluated ideas will be inadequate when confronted with a user test. Thus, it is important that the flexibility of the iterative design's implementation approach extends as far into the system as it is able to. Designers must further recognize that user testing results may suggest radical change that requires the designers to be prepared to completely abandon old ideas in favor of new ideas that are more equipped to suit user needs. Iterative design applies in many fields, from making knives to rockets, As an example consider the design of an electronic circuit that must perform a certain task, and must ultimately fit in a small space on a circuit board. It is useful to split these independent tasks into two smaller and simpler tasks, the functionality task, and the space and weight task. A breadboard is a useful way of

implementing the electronic circuit on an interim basis, without having to worry about space and weight.

Once the circuit works, improvements or incremental changes may be applied to the breadboard to increase or improve functionality over the original design. When the design is finalized, one can set about designing a proper circuit board meeting the space and weight criteria. Compacting the circuit on the circuit board requires that the wires and components be juggled around without changing their electrical characteristics. This juggling follows simpler rules than the design of the circuit itself, and is often automated. As far as possible off the shelf components are used, but where necessary for space or performance reasons, custom made components may be developed.

Benefits

When properly applied, iterative design will ensure a product or process is the best solution possible. When applied early in the development stage, significant cost savings are possible.

Other benefits to iterative design include:

1. Serious misunderstandings are made evident early in the lifecycle, when it's possible to react to them.
2. It enables and encourages user feedback, so as to elicit the system's real requirements.
3. The development team is forced to focus on those issues that are most critical to the project, and team members are shielded from those issues that distract them from the project's real risks.
4. Continuous, iterative testing enables an objective assessment of the project's status.
5. Inconsistencies among requirements, designs, and implementations are detected early.
6. The workload of the team, especially the testing team, is spread out more evenly throughout the lifecycle.
7. This approach enables the team to leverage lessons learned, and therefore to continuously improve the process.
8. Stakeholders in the project can be given concrete evidence of the project's status throughout the lifecycle.

Chapter- 2

Evaluation and Inspection Methods

There are a variety of methods currently used to evaluate usability. Certain methods make use of data gathered from users, while others rely on usability experts. There are usability evaluation methods that apply to all stages of design and development, from product definition to final design modifications. When choosing a method you must consider the cost, time constraints, and appropriateness of the method.

Cognitive modeling methods

Cognitive modeling involves creating a computational model to estimate how long it takes people to perform a given task. Models are based on psychological principles and experimental studies to determine times for cognitive processing and motor movements. Cognitive models can be used to improve user interfaces or predict problem errors and pitfalls during the design process. A few examples of cognitive models include:

Parallel Design

With parallel design, several people create an initial design from the same set of requirements. Each person works independently, and when finished, shares his/her concepts with the group. The design team considers each solution, and each designer uses the best ideas to further improve their own solution. This process helps to generate many different, diverse ideas and ensures that the best ideas from each design are integrated into the final concept. This process can be repeated several times until the team is satisfied with the final concept.

GOMS

GOMS (**Goals, Operators, Methods, and Selection rules**) is a kind of specialized human information processor model for human computer interaction observation. Developed in 1983 by Stuart Card, Thomas P. Moran and Allen Newell, it was explained in their book *The Psychology of Human Computer Interaction*. Following these initial steps, additional models for analysis evolved and are heavily used in the engineering-oriented usability community.

Overview

GOMS reduces a user's interaction with a computer to its elementary actions (these actions can be physical, cognitive or perceptual). Using these elementary actions as a framework an interface can be studied. There are several different GOMS variations which allow for different aspects of an interface to be accurately studied and predicted.

For all of the variants, the definitions of the major concepts are the same. **Goals** are what the user intends to accomplish. **Operators** are actions that are performed to get to the goal. **Methods** are sequences of operators that accomplish a goal. There can be more than one method available to accomplish a single goal, if this is the case then **selection rules** are used to describe when a user would select a certain method over the others. Selection rules are often ignored in typical GOMS analyses. There is some flexibility for the designers/analysts definition of all of these entities. For instance, one person's operator may be another's goal. The level of granularity is adjusted to capture what the particular evaluator is examining.

Advantages of GOMS Overall

The GOMS method is not necessarily the most accurate of human-computer interface interaction measurement methods, but it certainly has its advantages. A GOMS estimate of a particular interaction can be calculated with little effort, at little cost, and in a short amount of time if the average Methods-Time Measurement data for each specific task has been previously measured experimentally to a high degree of accuracy. With a careful investigation into all of the detailed steps necessary for a user to successfully interact with an interface, the time measurement of how long it will take a user to interact with that interface is a simple calculation. Summing the times necessary to complete the detailed steps provides an estimate for how long it will take a user to successfully complete the desired task.

Weaknesses of GOMS Overall

All of the GOMS techniques provide valuable information, but they all also have certain drawbacks. None of the techniques address user unpredictability - such as user behaviour being affected by fatigue, social surroundings, or organizational factors. The techniques are very explicit about basic movement operations, but are generally less rigid with basic cognitive actions. It is a fact that slips cannot be prevented, but none of the GOMS models allow for any type of error. Further, all of the techniques work under the assumption that a user will know what to do at any given point - so they apply only to expert users, not novices.

Functionality of the system is not considered, only the usability. If functionality were considered, the evaluation could make recommendations as to which functions should be performed by the system (i.e. mouse snap). User personalities, habits or physical restrictions (for example disabilities) are not accounted for in any of the GOMS models.

All users are assumed to be exactly the same. Recently some extensions of GOMS were developed, that allow to formulate GOMS models describing the interaction behavior of disabled users.

Except for KLM, the evaluators are required to have a fairly deep understanding of the theoretical foundations of GOMS, CCT (Cognitive Complexity Theory), or MHP (Model Human Processor). This limits the effective use of GOMS to large entities with the financial power to hire a dedicated human computer interaction (HCI) specialist or contract with a consultant with such expertise.

Variations

The plain, or "vanilla flavored" GOMS first introduced by Card, Moran and Newell is now referred to as CMN-GOMS. Keystroke Level Modeling (KLM) is the next GOMS technique and was also introduced by Card, Moran and Newell in their 1983 book. This technique makes several simplifying assumptions that make it really just a restricted version of GOMS. The third major variant on the GOMS technique is the 'Natural GOMS Language' or NGOMSL. This technique gives a very strict, but natural, language for building GOMS models. The final variation of GOMS is CPM-GOMS. This technique is based on the Model Human Processor. The main advantage of CPM-GOMS is that it allows for the modeling of parallel information processing by the user, however it is also the most difficult GOMS technique to implement.

Summary of CMN-GOMS Application

The CMN-GOMS method assumes that information is comprehended by a user in the following manner:

- Eyes/ears perceive information
- Information enters perceptual processor
- Information enters the visual/auditory image store
- Information is stored in the working memory and long term memory
- Information is analyzed in the cognitive processor and a desired reaction (motor function) is chosen
- Desired motor function is activated in the motor processor
- Desired motor function is applied by user's body

All measurements are provided in the following form: middleman[fastman, slowman]. The "middleman" term is the most typical time it would take to complete the action, or the time that is most representative of the average user (not the mean, average, or median, but the mode: the time that is most often measured). The fastman is a "best case" scenario. It is the reasonably best possible statistic. Note that, despite the name, it is not necessarily always the fastest time. It is instead the time that is expected to be the best a user could possibly do. The slowman time is, contrarily, a "worst case scenario."

In CMN-GOMS, the following Methods-Time Measurement data should be used:

- Eye fixation = 230[70, 700] milliseconds
- Eye movement = 30 milliseconds
- Perceptual Processor = 100[50, 200] milliseconds
- Cognitive Processor = 70[25, 170] milliseconds
- Motor Processor = 70[30, 100] milliseconds

Also important in CMN-GOMS is the time it takes to apply the motor function once it is processed. For this, a user can apply Fitt's Law.

Summary of KLM Application

The Keystroke Level Model is a less accurate, but faster application than CMN-GOMS. It is especially useful when determining time it takes to type a phrase, correct a realized error, or select something with a mouse. It uses the following average times as measured by Card, Moran and Newell:

- Press a key or button
    - Best typist = .08 seconds
    - Good typist = .12 seconds
    - Average skilled typist = .20 seconds
    - Average non-secretary = .28 seconds
    - Typing random letters = .50 seconds
    - Typing complex codes = .75 seconds
    - Worst typist = 1.2 seconds
- Point with a mouse (excluding click) = 1.1 seconds
- Move hands to keyboard from mouse (or vice-versa) = .4 seconds
- Mentally prepare = 1.35 seconds

Typing a word, assuming a subject’s hands are already on the keyboard, would therefore be calculated by multiplying the number of letters in the word by the value given above to “press a key or button.” Note that categorizing the subject into an accurate typing skill level impacts the estimated measurement greatly.

Importance of Assumptions in GOMS Analysis

Accurate assumptions are vital in GOMS analysis. Before applying the average times for detailed functions, it is very important that an experimenter make sure he or she has accounted for as many variables as possible by using assumptions. Experimenters should design their GOMS analysis for the users who will most likely be using the system which is being analyzed. Consider, for example, an experimenter wishes to determine how long it will take an F22 Raptor pilot to interact with an interface he or she has used for years. It

can probably be assumed that the pilot has outstanding vision and is in good physical health. In addition, it can be assumed that the pilot can interact with the interface quickly because of the vast hours of simulation and previous use he or she has endured. All things considered, it is fair to use fastman times in this situation. Contrarily, consider an 80-year-old woman with no flight experience attempting to interact with the same F22 Raptor interface. It is fair to say that the two people would have much different skill sets and those skill sets should be accounted for subjectively.

Accounting for Errors

The only way to account for errors in GOMS analysis is to predict where the errors are most likely to occur and measure the time it would take to correct the predicted errors. For example, assume an experimenter thought that in typing the word “the” it was likely that a subject would instead incorrectly type “hte.” The experimenter would calculate the time it takes to type the incorrect word, the time it takes to recognize that a mistake has been made, and the time it takes to correct the recognized error.

An experimenter should not, however, assume that an error will occur every time a subject does an action. James Reason calculated probabilities that an error will occur. According to Reason, a skill error is defined as an unconscious, automatic action resulting in an error (for example a mistyped key, a key hit the wrong number of times, a skipped key, etc.). A skill error will occur with a probability of .006 for young users and .011 for old users. A rule error, contrarily, is defined as following a series of steps and either making a mistake applying good rules incorrectly or applying bad rules at wrong times. Simple rule errors occur with a probability of .036 for young users and .024 for old users. Complex rule errors occur with a probability of .156 for young users and .324 for old users.

A concrete application of this idea is a GOMS model for keyboard navigation in Web pages. This model contains a probability for a focus loss during navigation inside the page using the TAB key.

Successful Applications of GOMS

A successful implementation of CPM-GOMS was in *Project Ernestine* held by New England Telephone. New ergonomically designed workstations were compared to old workstations in terms of improvement in telephone operators' performance. CPM-GOMS analysis estimated a 3% decrease in productivity. Over the four month trial 78,240 calls were analysed and it was concluded that the new workstations produced an actual 4% decrease in productivity. As the proposed workstation required less keystrokes than the original it was not clear from the time trials why the decrease occurred. However CPM-GOMS analysis made it apparent that the problem was that the new workstations did not utilize the workers' slack time. Not only did CPM-GOMS give a close estimate, but it provided more information of the situation.

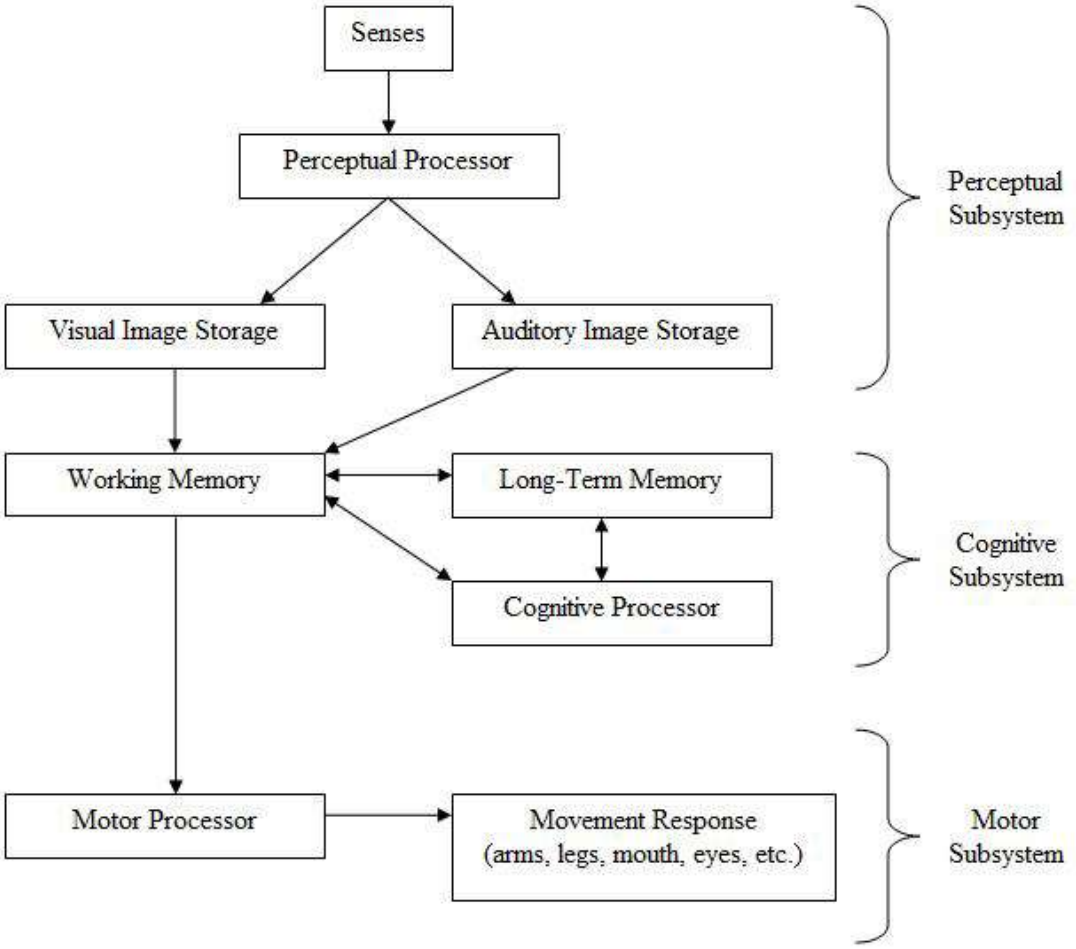
Software Tools

There exist various tools for the creation and analysis of Goms-Models. A selection is listed in the following:

- GOMSED (Goms-Editor - in german)
- QGoms (Quick-Goms)
- CogTool KLM-based modelling tool

Human Processor Model

Sometimes it is useful to break a task down and analyze each individual aspect separately. This allows the tester to locate specific areas for improvement. To do this, it is necessary to understand how the human brain processes information. A model of the human processor is shown below.



Many studies have been done to estimate the cycle times, decay times, and capacities of each of these processors. Variables that affect these can include subject age, aptitudes, ability, and the surrounding environment. For a younger adult, reasonable estimates are:

Parameter	Mean	Range
Eye movement time	230 ms	70-700 ms
Decay half-life of visual image storage	200 ms	90-1000 ms
Perceptual processor cycle time	100 ms	50-200 ms
Cognitive processor cycle time	70 ms	25-170 ms
Motor processor cycle time	70 ms	30-100 ms
Effective working memory capacity	7 items	5-9 items

Long-term memory is believed to have an infinite capacity and decay time.

Keystroke level modeling

Keystroke level modeling is essentially a less comprehensive version of GOMS that makes simplifying assumptions in order to reduce calculation time and complexity.

Inspection methods

These usability evaluation methods involve observation of users by an experimenter, or the testing and evaluation of a program by an expert reviewer. They provide more quantitative data as tasks can be timed and recorded.

Card sorting

Card sorting is a simple technique in user experience design where a group of subject experts or "users", however inexperienced with design, are guided to generate a category tree or folksonomy. It is a useful approach for designing information architecture, workflows, menu structure, or web site navigation paths.

Card sorting has a characteristically low-tech approach. The concepts are first identified and written onto simple index cards or Post-it notes. The user group then arranges these to represent the groups or structures they are familiar with.

Groups may either be organised as collaborative groups (focus groups) or as repeated individual sorts. The literature discusses appropriate numbers of users needed to produce trustworthy results.

A card sort is commonly undertaken when designing a navigation structure for an environment that offers an interesting variety of content and functionality, such as a web site. In that context, the items to be organized are those that are significant in the environment. The way that the items are organized should make sense to the target audience and cannot be determined from first principles.

The field of information architecture is founded upon the study of the structure of information. If an accepted and standardized taxonomy exists for a subject, it would be natural to simply apply that taxonomy as a means of organizing both the information in the environment and any navigation to particular subjects or functions. Card sorting is applied when:

- The variety in the items to be organized is so great that no existing taxonomy is accepted as organizing the items.
- The similarities among the items make them difficult to divide clearly into categories.
- Members of the audience that uses the environment may differ significantly in how they view the similarities among items and the appropriate groupings of items.

Basic method

To perform a card sort:

1. A person representative of the audience is given a set of index cards with terms already written on them.
2. This person puts the terms into logical groupings, and finds a category name for each grouping.
3. This process is repeated across a population of test subjects.
4. The results are later analyzed to reveal patterns.

Open card sorting

In an **open card sort**, participants create their own names for the categories.

This helps reveal not only how they mentally classify the cards, but also what terms they use for the categories.

Open sorting is **generative**; it is typically used to discover patterns in how participants classify, which in turn helps generate ideas for organizing information.

Closed card sorting

In a **closed card sort**, participants are provided with a predetermined set of category names. They then assign the index cards to these fixed categories.

This helps reveal the degree to which the participants agree on which cards belong under each category.

Closed sorting is **evaluative**; it is typically used to judge whether a given set of category names provides an effective way to organize a given collection of content.

Reverse card sorting

In a *reverse card sort* or card-based classification, an existing structure of categories and sub-categories is tested. Users are given tasks and are asked to complete them navigating a collection of cards. Each card contains the names of subcategories related to a category, and the user should find the card most relevant to the given task starting from the main card with the top-level categories. This ensures that the structure is evaluated in isolation, nullifying the effects of navigational aids, visual design, and other factors.

Reverse card sorting is *evaluative*; it's used to judge whether a predetermined hierarchy provides a good way to find information.

Analyzing card-sort results

Various methods can be used to analyze the data. The purpose of the analysis is to extract patterns from the population of test subjects, so that a common set of categories and relationships emerges. This common set is then incorporated into the design of the environment, either for navigation or for other purposes.

Card sorting is an established technique with an emerging literature.

Online (remote) card sorting

There are a number of tools available to perform card sorting activities with survey participants via the internet. The perceived advantages of remote card sorting are that it allows a larger group of participants to be reached at a lower cost. The software can also assist in the process of analyzing card sort results. The advantages of a remote card sort must be traded off against the lack of personal interaction between card sort participants and the card sort administrator, which may produce valuable insights.

Tree testing

Tree testing is a usability technique for evaluating the findability of topics in a website. It is also known as **reverse card sorting** or **card-based classification**.

A large website is typically organized into a hierarchy (a "tree") of topics and subtopics. Tree testing provides a way to measure how well users can find items in this hierarchy.

Unlike traditional usability testing, tree testing is not done on the website itself; instead, a simplified text version of the site structure is used. This ensures that the structure is evaluated in isolation, nullifying the effects of navigational aids, visual design, and other factors.

Basic method

In a typical tree test:

1. The participant is given a "find it" task (e.g., "Look for brown belts under $25").
2. They are shown a text list of the top-level topics of the website.
3. They choose a heading, and are then shown a list of subtopics.
4. They continue choosing (moving down through the tree) — drilling down, backtracking if necessary – until they find a topic that satisfies the task (or until they give up).
5. The participant does several tasks in this manner, starting each task back at the top of the tree.
6. Once several participants have completed the test, the results are analyzed for each task.

Analyzing the results

The analysis typically tries to answer these questions:

- Could users successfully find particular items in the tree?
- Could they find those items directly, without having to backtrack?
- If they couldn't find items, where did they go astray?
- Could they choose between topics quickly, without having to think too much?
- Overall, which parts of the tree worked well, and which fell down?

Tools

Tree testing was originally done on paper (typically using index cards), but can now also be conducted using specialized software.

Heuristic Evaluation

A **heuristic evaluation** is a discount usability inspection method for computer software that helps to identify usability problems in the user interface (UI) design. It specifically involves evaluators examining the interface and judging its compliance with recognized usability principles (the "heuristics"). These evaluation methods are now widely taught and practiced in the New Media sector, where UIs are often designed in a short space of

time on a budget that may restrict the amount of money available to provide for other types of interface testing.

Introduction

The main goal of heuristic evaluations is to identify any problems associated with the design of user interfaces. Usability consultant Jakob Nielsen developed this method on the basis of several years of experience in teaching and consulting about usability engineering.

Heuristic evaluations are one of the most informal methods of usability inspection in the field of human-computer interaction. There are many sets of usability design heuristics; they are not mutually exclusive and cover many of the same aspects of user interface design.

Quite often, usability problems that are discovered are categorized—often on a numeric scale—according to their estimated impact on user performance or acceptance. Often the heuristic evaluation is conducted in the context of use cases (typical user tasks), to provide feedback to the developers on the extent to which the interface is likely to be compatible with the intended users' needs and preferences.

The simplicity of heuristic evaluation is beneficial at the early stages of design. This usability inspection method does not require user testing which can be burdensome due to the need for users, a place to test them and a payment for their time. Heuristic evaluation requires only one expert, reducing the complexity and expended time for evaluation. Most heuristic evaluations can be accomplished in a matter of days. The time required varies with the size of the artifact, its complexity, the purpose of the review, the nature of the usability issues that arise in the review, and the competence of the reviewers. Using heuristic evaluation prior to user testing will reduce the number and severity of design errors discovered by users. Although heuristic evaluation can uncover many major usability issues in a short period of time, a criticism that is often leveled is that results are highly influenced by the knowledge of the expert reviewer(s). This "one-sided" review repeatedly has different results than performance testing, each type of testing uncovering a different set of problems.

Nielsen's heuristics

Jakob Nielsen's heuristics are probably the most used usability heuristics for user interface design. Nielsen developed the heuristics based on work together with Rolf Molich in 1990. The final set of heuristics that are still used today were released by Nielsen in 1994. The heuristics as published in Nielsen's book *Usability Engineering* are as follows

- **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 user's 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.

Gerhardt-Powals' cognitive engineering principles

Although Nielsen is considered the expert and field leader in heuristics, Jill Gerhardt-Powals also developed a set of cognitive principles for enhancing computer performance. These heuristics, or principles, are similar to Nielsen's heuristics but take a more holistic approach to evaluation. Gerhardt Powals' principles are listed below.

- **Automate unwanted workload**:
  - free cognitive resources for high-level tasks.
  - eliminate mental calculations, estimations, comparisons, and unnecessary thinking.

- **Reduce uncertainty**:
  - display data in a manner that is clear and obvious.

- **Fuse data**:
  - reduce cognitive load by bringing together lower level data into a higher-level summation.

- **Present new information with meaningful aids to interpretation**:
  - use a familiar framework, making it easier to absorb.
  - use everyday terms, metaphors, etc.

- **Use names that are conceptually related to function**:
  - Context-dependent.
  - Attempt to improve recall and recognition.
  - Group data in consistently meaningful ways to decrease search time.

- **Limit data-driven tasks**:
  - Reduce the time spent assimilating raw data.
  - Make appropriate use of color and graphics.

- **Include in the displays only that information needed by the user at a given time**.
- **Provide multiple coding of data when appropriate**.
- **Practice judicious redundancy**.

Usability Inspection

Usability inspection is the name for a set of methods where an evaluator inspects a user interface. This is in contrast to usability testing where the usability of the interface is evaluated by testing it on real users. Usability inspections can generally be used early in the development process by evaluating prototypes or specifications for the system that can't be tested on users. Usability inspection methods are generally considered to be cheaper to implement than testing on users.

Usability inspection methods include:

- Cognitive walkthrough (task-specific)
- Heuristic evaluation (holistic)
- Pluralistic walkthrough

Pluralistic Inspection

Pluralistic Inspections are meetings where users, developers, and human factors people meet together to discuss and evaluate step by step of a task scenario. As more people inspect the scenario for problems, the higher the probability to find problems. In addition, the more interaction in the team, the faster the usability issues are resolved.

Consistency Inspection

In consistency inspection, expert designers review products or projects to ensure consistency across multiple products to look if it does things in the same way as their own designs.

Activity Analysis

Activity analysis is a usability method used in preliminary stages of development to get a sense of situation. It involves an investigator observing users as they work in the field. Also referred to as user observation, it is useful for specifying user requirements and studying currently used tasks and subtasks. The data collected is qualitative and useful for defining the problem. It should be used when you wish to frame what is needed, or “What do we want to know?”

Chapter- 3

Inquiry and Prototyping Methods

The following usability evaluation methods involve collecting qualitative data from users. Although the data collected is subjective, it provides valuable information on what the user wants.

Task Analysis

Task analysis means learning about users' goals and users' ways of working. Task analysis can also mean figuring out what more specific tasks users must do to meet those goals and what steps they must take to accomplish those tasks. Along with user and task analysis, we often do a third analysis: understanding users' environments (physical, social, cultural, and technological environments).

Focus Groups

A focus group is a focused discussion where a moderator leads a group of participants through a set of questions on a particular topic. Although typically used as a marketing tool, Focus Groups are sometimes used to evaluate usability. Used in the product definition stage, a group of 6 to 10 users are gathered to discuss what they desire in a product. An experienced focus group facilitator is hired to guide the discussion to areas of interest for the developers. Focus groups are typically videotaped to help get verbatim quotes, and clips are often used to summarize opinions. The data gathered is not usually quantitative, but can help get an idea of a target group's opinion.

Prototyping methods

Rapid Prototyping

Rapid prototyping is a method used in early stages of development to validate and refine the usability of a system. It can be used to quickly and cheaply evaluate user-interface designs without the need for an expensive working model. This can help remove hesitation to change the design, since it is implemented before any real programming begins. One such method of rapid prototyping is paper prototyping.

Testing methods

These usability evaluation methods involve testing of subjects for the most quantitative data. Usually recorded on video, they provide task completion time and allow for observation of attitude.

Remote usability testing

Remote usability testing (also known as unmoderated or asynchronous usability testing) involves the use of a specially modified online survey, allowing the quantification of user testing studies by providing the ability to generate large sample sizes. Additionally, this style of user testing also provides an opportunity to segment feedback by demographic, attitudinal and behavioural type. The tests are carried out in the user's own environment (rather than labs) helping further simulate real-life scenario testing. This approach also provides a vehicle to easily solicit feedback from users in remote areas.

Thinking Aloud

Think-aloud protocol (or think-aloud protocols, or TAP) is a method used to gather data in usability testing in product design and development, in psychology and a range of social sciences (e.g., reading, writing and translation process research). The think-aloud method was introduced in the usability field by Clayton Lewis while he was at IBM, and is explained in *Task-Centered User Interface Design: A Practical Introduction* by C. Lewis and J. Rieman. The method was developed based on the techniques of protocol analysis by Ericsson and Simon.

Think aloud protocols involve participants thinking aloud as they are performing a set of specified tasks. Users are asked to say whatever they are looking at, thinking, doing, and feeling, as they go about their task. This enables observers to see first-hand the process of task completion (rather than only its final product). Observers at such a test are asked to objectively take notes of everything that users say, without attempting to interpret their actions and words. Test sessions are often audio and video taped so that developers can go back and refer to what participants did, and how they reacted. The purpose of this method is to make explicit what is implicitly present in subjects who are able to perform a specific task.

A related but slightly different data-gathering method is the **talk-aloud protocol**. This involves participants only describing their action but not giving explanations. This method is thought to be more objective in that participants merely report how they go about completing a task rather than interpreting or justifying their actions.

As Hannu and Pallab state the thinking aloud protocol can be divide in two different experimental procedures: the first one, is the concurrent thinking aloud protocol, collected during the decision task; the second procedure is the retrospective thinking aloud protocol gathered after the decision task.

RITE Method

RITE Method, for Rapid Iterative Testing and Evaluation, typically referred to as "RITE" testing, is an iterative usability method. It was defined by Michael Medlock, Dennis Wixon, Bill Fulton, Mark Terrano and Ramon Romero. It has been publicly championed by Dennis Wixon while working in the games space for Microsoft. It has many similarities to "traditional" or "discount" usability testing. The tester and team must define a target population for testing, schedule participants to come in to the lab, decide on how the users' behaviors will be measured, construct a test script and have participants engage in a verbal protocol (e.g. think aloud). However it differs from these methods in that it advocates that changes to the user interface are made as soon as a problem is identified and a solution is clear. Sometimes this can occur after observing as few as one participant. Once the data for a participant has been collected the usability engineer and team decide if they will be making any changes to the prototype prior to the next participant. The changed interface is then tested with the remaining users.

Initially it was documented as being used in the PC games business, but it in all truth has probably been in use "unofficially" since designers started prototyping products and watching users use the prototypes. Since its official definition and naming its use has rapidly expanded to many other software industries.

Subjects-in-Tandem

Subjects-in-tandem is pairing of subjects in a usability test to gather important information on the ease of use of a product. Subjects tend to think out loud and through their verbalized thoughts designers learn where the problem areas of a design are. Subjects very often provide solutions to the problem areas to make the product easier to use.

Other methods

Cognitive walkthrough

The **cognitive walkthrough** method is a usability inspection method used to identify usability issues in a piece of software or web site, focusing on how easy it is for new users to accomplish tasks with the system. Whereas **cognitive walkthrough** is task-specific, heuristic evaluation takes a holistic view to catch problems not caught by this and other usability inspection methods. The method is rooted in the notion that users typically prefer to learn a system by using it to accomplish tasks, rather than, for example, studying a manual. The method is prized for its ability to generate results quickly with low cost, especially when compared to usability testing, as well as the ability to apply the method early in the design phases, before coding has even begun.

Introduction

A cognitive walkthrough starts with a task analysis that specifies the sequence of steps or actions required by a user to accomplish a task, and the system responses to those actions. The designers and developers of the software then walkthrough the steps as a group, asking themselves a set of questions at each step. Data is gathered during the walkthrough, and afterwards a report of potential issues is compiled. Finally the software is redesigned to address the issues identified.

The effectiveness of methods such as cognitive walkthroughs is hard to measure in applied settings, as there is very limited opportunity for controlled experiments while developing software. Typically measurements involve comparing the number of usability problems found by applying different methods. However, Gray and Salzman called into question the validity of those studies in their dramatic 1998 paper "Damaged Merchandise", demonstrating how very difficult it is to measure the effectiveness of usability inspection methods. However, the consensus in the usability community is that the cognitive walkthrough method works well in a variety of settings and applications.

Walking through the tasks

After the task analysis has been made the participants perform the walkthrough by asking themselves a set of questions for each subtask. Typically four questions are asked:

- **Will the user try to achieve the effect that the subtask has?** Does the user understand that this subtask is needed to reach the user's goal?
- **Will the user notice that the correct action is available?** E.g. is the button visible?
- **Will the user understand that the wanted subtask can be achieved by the action?** E.g. the right button is visible but the user does not understand the text and will therefore not click on it.
- **Does the user get feedback?** Will the user know that they have done the right thing after performing the action?

By answering the questions for each subtask usability problems will be noticed.

Common Mistakes

In teaching people to use the walkthrough method, Lewis & Rieman have found that there are two common misunderstandings:

1. The evaluator doesn't know how to perform the task themself, so they stumble through the interface trying to discover the correct sequence of actions -- and then they evaluate the stumbling process. (The user should identify and perform the **optimal** action sequence.)

2. The walkthrough does not test real users on the system. The walkthrough will often identify many more problems than you would find with a single, unique user in a single test session.

History

The method was developed in the early nineties by Wharton, et al., and reached a large usability audience when it was published as a chapter in Jakob Nielsen's seminal book on usability, "Usability Inspection Methods." The Wharton, et al. method required asking four questions at each step, along with extensive documentation of the analysis. In 2000 there was a resurgence in interest in the method in response to a CHI paper by Spencer who described modifications to the method to make it effective in a real software development setting. Spencer's streamlined method required asking only two questions at each step, and involved creating less documentation. Spencer's paper followed the example set by Rowley, et al. who described the modifications to the method that they made based on their experience applying the methods in their 1992 CHI paper "The Cognitive Jogthrough".

Benchmarking

Benchmarking is the process of comparing one's business processes and performance metrics to industry bests and/or best practices from other industries. Dimensions typically measured are quality, time, and cost. Improvements from learning mean doing things better, faster, and cheaper.

Benchmarking involves management identifying the best firms in their industry, or any other industry where similar processes exist, and comparing the results and processes of those studied (the "targets") to one's own results and processes to learn how well the targets perform and, more importantly, how they do it.

The term benchmarking was first used by cobblers to measure people's feet for shoes. They would place someone's foot on a "bench" and mark it out to make the pattern for the shoes. Benchmarking is most used to measure performance using a specific indicator (cost per unit of measure, productivity per unit of measure, cycle time of x per unit of measure or defects per unit of measure) resulting in a metric of performance that is then compared to others.

Also referred to as "best practice benchmarking" or "process benchmarking", it is a process used in management and particularly strategic management, in which organizations evaluate various aspects of their processes in relation to best practice companies' processes, usually within a peer group defined for the purposes of comparison. This then allows organizations to develop plans on how to make improvements or adapt specific best practices, usually with the aim of increasing some aspect of performance. Benchmarking may be a one-off event, but is often treated as a continuous process in which organizations continually seek to improve their practices.

Popularity and benefits from benchmarking

In 2008, a comprehensive survey on benchmarking was commissioned by The Global Benchmarking Network, a network of benchmarking centers representing 22 countries. Over 450 organizations responded from over 40 countries. The results showed that:

1. Mission and Vision Statements and Customer (Client) Surveys are the most used (by 77% of organisations) of 20 improvement tools, followed by SWOT analysis(72%), and Informal Benchmarking (68%). Performance Benchmarking was used by (49%) and Best Practice Benchmarking by (39%).
2. The tools that are likely to increase in popularity the most over the next three years are Performance Benchmarking, Informal Benchmarking, SWOT, and Best Practice Benchmarking. Over 60% of organizations that are not currently using these tools indicated they are likely to use them in the next three years.

Collaborative benchmarking

Benchmarking, was originally invented as a formal process by Rank Xerox, is usually carried out by individual companies. Sometimes it may be carried out collaboratively by groups of companies (e.g. subsidiaries of a multinational in different countries). One example is that of the Dutch municipally-owned water supply companies, which have carried out a voluntary collaborative benchmarking process since 1997 through their industry association. Another example is the UK construction industry which has carried out benchmarking since the late 1990s again through its industry association and with financial support from the UK Government.

Procedure

There is no single benchmarking process that has been universally adopted. The wide appeal and acceptance of benchmarking has led to various benchmarking methodologies emerging. The seminal book on benchmarking is Boxwell's *Benchmarking for Competitive Advantage* published by McGraw-Hill in 1994. It has withstood the test of time and is still a relevant read. The first book on benchmarking, written and published by Kaiser Associates, is a practical guide and offers a 7-step approach. Robert Camp (who wrote one of the earliest books on benchmarking in 1989) developed a 12-stage approach to benchmarking.

The 12 stage methodology consisted of 1. Select subject ahead 2. Define the process 3. Identify potential partners 4. Identify data sources 5. Collect data and select partners 6. Determine the gap 7. Establish process differences 8. Target future performance 9. Communicate 10. Adjust goal 11. Implement 12. Review/recalibrate.

The following is an example of a typical benchmarking methodology:

1. **Identify your problem areas -** Because benchmarking can be applied to any business process or function, a range of research techniques may be required. They include: informal conversations with customers, employees, or suppliers; exploratory research techniques such as focus groups; or in-depth marketing research, quantitative research, surveys, questionnaires, re-engineering analysis, process mapping, quality control variance reports, or financial ratio analysis. Before embarking on comparison with other organizations it is essential that you know your own organization's function, processes; base lining performance provides a point against which improvement effort can be measured.
2. **Identify other industries that have similar processes -** For instance if one were interested in improving hand offs in addiction treatment he/she would try to identify other fields that also have hand off challenges. These could include air traffic control, cell phone switching between towers, transfer of patients from surgery to recovery rooms.
3. **Identify organizations that are leaders in these areas -** Look for the very best in any industry and in any country. Consult customers, suppliers, financial analysts, trade associations, and magazines to determine which companies are worthy of study.
4. **Survey companies for measures and practices -** Companies target specific business processes using detailed surveys of measures and practices used to identify business process alternatives and leading companies. Surveys are typically masked to protect confidential data by neutral associations and consultants.
5. **Visit the "best practice" companies to identify leading edge practices -** Companies typically agree to mutually exchange information beneficial to all parties in a benchmarking group and share the results within the group.
6. **Implement new and improved business practices -** Take the leading edge practices and develop implementation plans which include identification of specific opportunities, funding the project and selling the ideas to the organization for the purpose of gaining demonstrated value from the process.

Cost of benchmarking

The three main types of costs in benchmarking are:

- **Visit Costs -** This includes hotel rooms, travel costs, meals, a token gift, and lost labor time.
- **Time Costs -** Members of the benchmarking team will be investing time in researching problems, finding exceptional companies to study, visits, and implementation. This will take them away from their regular tasks for part of each day so additional staff might be required.
- **Benchmarking Database Costs -** Organizations that institutionalize benchmarking into their daily procedures find it is useful to create and maintain a database of best practices and the companies associated with each best practice now.

The cost of benchmarking can substantially be reduced through utilizing the many internet resources that have sprung up over the last few years. These aim to capture benchmarks and best practices from organizations, business sectors and countries to make the benchmarking process much quicker and cheaper.

Technical Benchmarking/Product Benchmarking

The technique initially used to compare existing corporate strategies with a view to achieving the best possible performance in new situations (see above), has recently been extended to the comparison of technical products. This process is usually referred to as "Technical Benchmarking" or "Product Benchmarking". Its use is particularly well developed within the automotive industry ("Automotive Benchmarking"), where it is vital to design products that match precise user expectations, at minimum possible cost, by applying the best technologies available worldwide. Many data are obtained by fully disassembling existing cars and their systems. Such analyses were initially carried out in-house by car makers and their suppliers. However, as they are expensive, they are increasingly outsourced to companies specialized in this area. Indeed, outsourcing has enabled a drastic decrease in costs for each company (by cost sharing) and the development of very efficient tools (standards, software).

Types of benchmarking

- **Process benchmarking -** the initiating firm focuses its observation and investigation of business processes with a goal of identifying and observing the best practices from one or more benchmark firms. Activity analysis will be required where the objective is to benchmark cost and efficiency; increasingly applied to back-office processes where outsourcing may be a consideration.
- **Financial benchmarking -** performing a financial analysis and comparing the results in an effort to assess your overall competitiveness and productivity.
- **Benchmarking from an investor perspective-** extending the benchmarking universe to also compare to peer companies that can be considered alternative investment opportunities from the perspective of an investor.
- **Performance benchmarking -** allows the initiator firm to assess their competitive position by comparing products and services with those of target firms.
- **Product benchmarking -** the process of designing new products or upgrades to current ones. This process can sometimes involve reverse engineering which is taking apart competitors products to find strengths and weaknesses.
- **Strategic benchmarking -** involves observing how others compete. This type is usually not industry specific, meaning it is best to look at other industries.
- **Functional benchmarking -** a company will focus its benchmarking on a single function to improve the operation of that particular function. Complex functions such as Human Resources, Finance and Accounting and Information and Communication Technology are unlikely to be directly comparable in cost and

efficiency terms and may need to be disaggregated into processes to make valid comparison.

- **Best-in-class benchmarking -** involves studying the leading competitor or the company that best carries out a specific function.
- **Operational benchmarking -** embraces everything from staffing and productivity to office flow and analysis of procedures performed.
- **Energy benchmarking -** developing an accurate model of a building's energy consumption with the purpose of measuring reductions in usage.

Metric Benchmarking

Another approach to making comparisons involves using more aggregative cost or production information to identify strong and weak performing units. The two most common forms of quantitative analysis used in metric benchmarking are data envelope analysis (DEA) and regression analysis. DEA estimates the cost level an efficient firm should be able to achieve in a particular market. In infrastructure regulation, DEA can be used to reward companies/operators whose costs are near the efficient frontier with additional profits. Regression analysis estimates what the average firm should be able to achieve. With regression analysis firms that performed better than average can be rewarded while firms that performed worse than average can be penalized. Such benchmarking studies are used to create yardstick comparisons, allowing outsiders to evaluate the performance of operators in an industry. A variety of advanced statistical techniques, including stochastic frontier analysis, have been utilized to identify high performers and weak performers in a number of industries, including applications to schools, hospitals, water utilities, and electric utilities.

One of the biggest challenges for Metric Benchmarking is the variety of metric definitions used by different companies and/or divisions. Metrics definitions may also change over time within the same organization due to changes in leadership and priorities. The most useful comparisons can be made when metrics definitions are common between compared units and do not change over time so improvements can be verified.

Meta-Analysis

In statistics, a **meta-analysis** combines the results of several studies that address a set of related research hypotheses. In its simplest form, this is normally by identification of a common measure of "effect size", for which a weighted average might be the output of a meta-analyses. Here the weighting might be related to sample sizes within the individual studies. More generally there are other differences between the studies that need to be allowed for, but the general aim of a meta-analysis is to more powerfully estimate the true "effect size" as opposed to a smaller "effect size" derived in a single study under a given single set of assumptions and conditions.

Meta-analyses are often, but not always, important components of a *systematic review* procedure. Here it is covenient to follow the terminology used by the Cochrane

Collaboration, and use "meta-analysis" to refer to statistical methods of combining evidence, leaving other aspects of 'research synthesis' or 'evidence synthesis', such as combining information from qualitative studies, for the more general context of systematic reviews.

History

The first meta-analysis was performed by Karl Pearson in 1904, in an attempt to overcome the problem of reduced statistical power in studies with small sample sizes; analyzing the results from a group of studies can allow more accurate data analysis. However, the first meta-analysis of all conceptually identical experiments concerning a particular research issue, and conducted by independent researchers, has been identified as the 1940 book-length publication *Extra-sensory perception after sixty years*, authored by Duke University psychologists J. G. Pratt, J. B. Rhine, and associates. This encompassed a review of 145 reports on ESP experiments published from 1882 to 1939, and included an estimate of the influence of unpublished papers on the overall effect (the *file-drawer problem*). Although meta-analysis is widely used in epidemiology and evidence-based medicine today, a meta-analysis of a medical treatment was not published until 1955. In the 1970s, more sophisticated analytical techniques were introduced in educational research, starting with the work of Gene V. Glass, Frank L. Schmidt and John E. Hunter. The online Oxford English Dictionary lists the first usage of the term in the statistical sense as 1976 by Glass. The statistical theory surrounding meta-analysis was greatly advanced by the work of Nambury S. Raju, Larry V. Hedges, Harris Cooper, Ingram Olkin, John E. Hunter, Jacob Cohen, Thomas C. Chalmers, and Frank L. Schmidt.

Advantages of meta-analysis

Advantages of meta-analysis (eg. over classical literature reviews, simple overall means of effect sizes etc.) include:

- Shows if the results are more varied than what is expected from the sample diversity
- Derivation and statistical testing of overall factors / effect size parameters in related studies
- Generalization to the population of studies
- Ability to control for between-study variation
- Including moderators to explain variation
- Higher statistical power to detect an effect than in ‘n=1 sized study sample’

Steps in a meta-analysis

1. Search of literature

2. Selection of studies (‘incorporation criteria’)

- Based on quality criteria, e.g. the requirement of randomization and blinding in a clinical trial
- Selection of specific studies on a well-specified subject, e.g. the treatment of breast cancer.
- Decide whether unpublished studies are included to avoid publication bias.

3. Decide which dependent variables or summary measures are allowed. For instance:

- Differences (discrete data)
- Means (continuous data)
- Hedges' *g* is a popular summary measure for continuous data that is standardized in order to eliminate scale differences, but it incorporates an index of variation between groups:

$\delta = \frac{\mu_t - \mu_c}{\sigma}$, in which μ_t is the treatment mean, μ_c is the control mean, σ^2 the pooled variance.

4. Model selection

Meta-regression models

Generally, three types of models can be distinguished in the literature on meta-analysis: simple regression, fixed effect meta-regression and random effects meta-regression.

Simple regression

The model can be specified as

$$y_j = \beta_0 + \beta_1 x_{1j} + \beta_2 x_{2j} + \cdots + \varepsilon$$

Where y_j is the effect size in study *j* and β_0 (intercept) the estimated overall effect size. The variables $x_{i.}(i = 1 \ldots k)$ specify different characteristics of the study, ε specifies the between study variation. Note that this model does not allow specification of within study variation.

Fixed-effect meta-regression

Fixed-effect meta-regression assumes that the true effect size θ is normally distributed with $\mathcal{N}(\theta, \sigma_\theta)$ where σ^2_θ is the within study variance of the effect size. A fixed effect meta-regression model thus allows for within study variability, but no between study variability because all studies have the identical expected fixed effect size θ, i.e. $\varepsilon = 0$. ***Note that for the "fixed-effect" no plural is used (in contrast to "random-effects") as only ONE true effect across all datasets is assumed.***

$$y_j = \beta_0 + \beta_1 x_{1j} + \beta_2 x_{2j} + \cdots + \eta_j$$

Here $\sigma^2_{\eta_j}$ is the variance of the effect size in study *j*. Fixed effect meta-regression ignores between study variation. As a result, parameter estimates are biased if between study variation can not be ignored. Furthermore, generalizations to the population are not possible.

Random effects meta-regression

Random effects meta-regression rests on the assumption that θ in $\mathcal{N}(\theta, \sigma_i)$ is a random variable following a (hyper-)distribution $\mathcal{N}(\theta, \sigma_\theta)$.

$$y_j = \beta_0 + \beta_1 x_{1j} + \beta_2 x_{2j} + \cdots + \eta + \varepsilon_j$$

Here $\sigma^2_{\varepsilon_j}$ is the variance of the effect size in study *j*. Between study variance σ^2_{η} is estimated using common estimation procedures for random effects models (restricted maximum likelihood (REML) estimators).

Applications in modern science

Modern statistical meta-analysis does more than just combine the effect sizes of a set of studies. It can test if the outcomes of studies show more variation than the variation that is expected because of sampling different research participants. If that is the case, study characteristics such as measurement instrument used, population sampled, or aspects of the studies' design are coded. These characteristics are then used as predictor variables to analyze the excess variation in the effect sizes. Some methodological weaknesses in studies can be corrected statistically. For example, it is possible to correct effect sizes or correlations for the downward bias due to measurement error or restriction on score ranges.

Meta-analysis can be done with single-subject design as well as group research designs. This is important because much of the research on low incidents populations has been done with single-subject research designs. Considerable dispute exists for the most appropriate meta-analytic technique for single subject research.

Meta-analysis leads to a shift of emphasis from single studies to multiple studies. It emphasizes the practical importance of the effect size instead of the statistical significance of individual studies. This shift in thinking has been termed "meta-analytic thinking". The results of a meta-analysis are often shown in a forest plot.

Results from studies are combined using different approaches. One approach frequently used in meta-analysis in health care research is termed 'inverse variance method'. The average effect size across all studies is computed as a *weighted mean*, whereby the weights are equal to the inverse variance of each studies' effect estimator. Larger studies

and studies with less random variation are given greater weight than smaller studies. Other common approaches include the Mantel–Haenszel method and the Peto method.

A recent approach to studying the influence that weighting schemes can have on results has been proposed through the construct of *gravity*, which is a special case of combinatorial meta-analysis.

Signed differential mapping is a statistical technique for meta-analyzing studies on differences in brain activity or structure which used neuroimaging techniques such as fMRI, VBM or PET.

Weaknesses

Meta-analysis can never follow the rules of hard science, for example being double-blind, controlled, or proposing a way to falsify the theory in question. It is only a statistical examination of scientific studies, not an actual scientific study, itself.

A weakness of the method is that sources of bias are not controlled by the method. A good meta-analysis of badly designed studies will still result in bad statistics. Robert Slavin has argued that only methodologically sound studies should be included in a meta-analysis, a practice he calls 'best evidence meta-analysis'. Other meta-analysts would include weaker studies, and add a study-level predictor variable that reflects the methodological quality of the studies to examine the effect of study quality on the effect size.

File drawer problem

Another weakness of the method is the heavy reliance on published studies, which may create exaggerated outcomes, as it is very hard to publish studies that show no significant results. For any given research area, one cannot know how many studies have been conducted but never reported and the results filed away.

This file drawer problem results in the distribution of effect sizes that are biased, skewed or completely cut off, creating a serious base rate fallacy, in which the significance of the published studies is overestimated. For example, if there were fifty tests, and only ten got results, then the real outcome is only 20% as significant as it appears, except that the other 80% were not submitted for publishing, or thrown out by publishers as uninteresting. This should be seriously considered when interpreting the outcomes of a meta-analysis.

This can be visualized with a funnel plot which is a scatter plot of sample size and effect sizes. There are several procedures available that attempt to correct for the file drawer problem, once identified, such as guessing at the cut off part of the distribution of study effects.

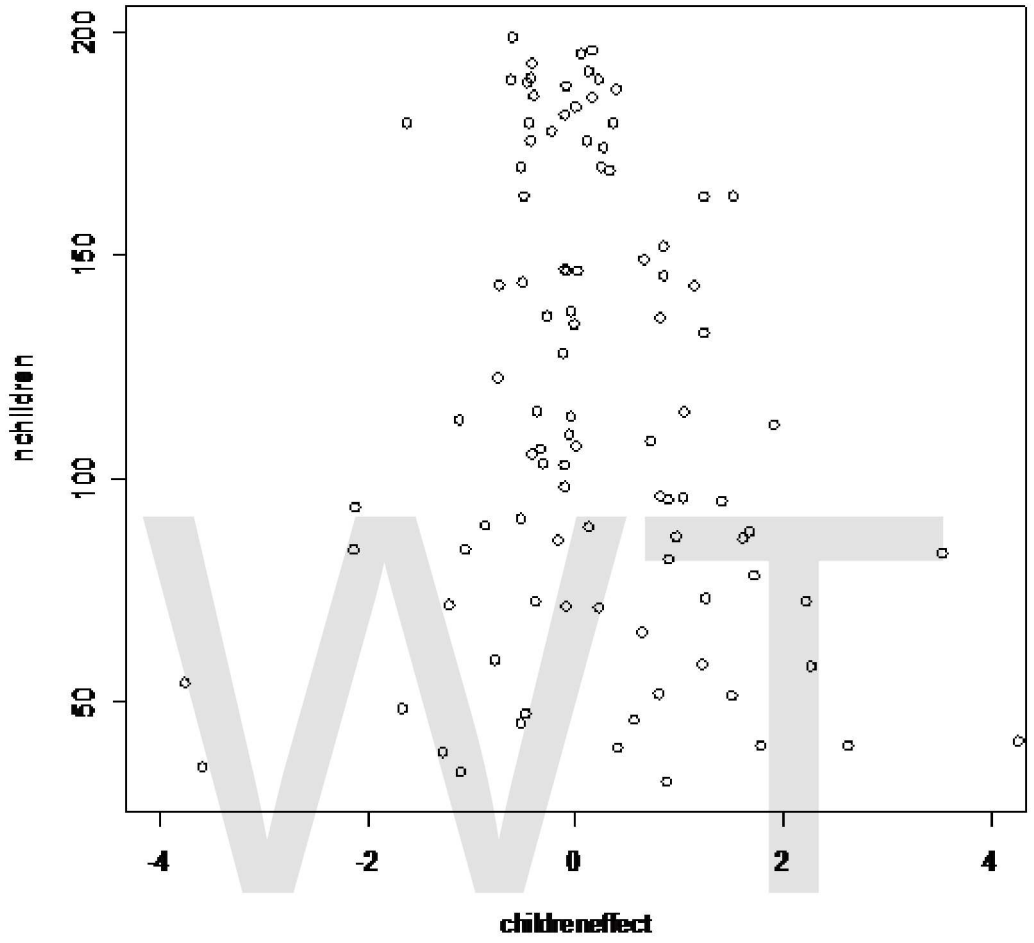
Other weaknesses are Simpson's Paradox (two smaller studies may point in one direction, and the combination study in the opposite direction); the coding of an effect is subjective; the decision to include or reject a particular study is subjective; there are two different ways to measure effect: correlation or standardized mean difference; the interpretation of effect size is purely arbitrary; it has not been determined if the statistically most accurate method for combining results is the fixed effect model or the random effects model; and, for medicine, the underlying risk in each studied group is of significant importance, and there is no universally agreed-upon way to weight the risk.

The example provided by the Rind et al. controversy illustrates an application of meta-analysis which has been the subject of subsequent criticisms of many of the components of the meta-analysis.

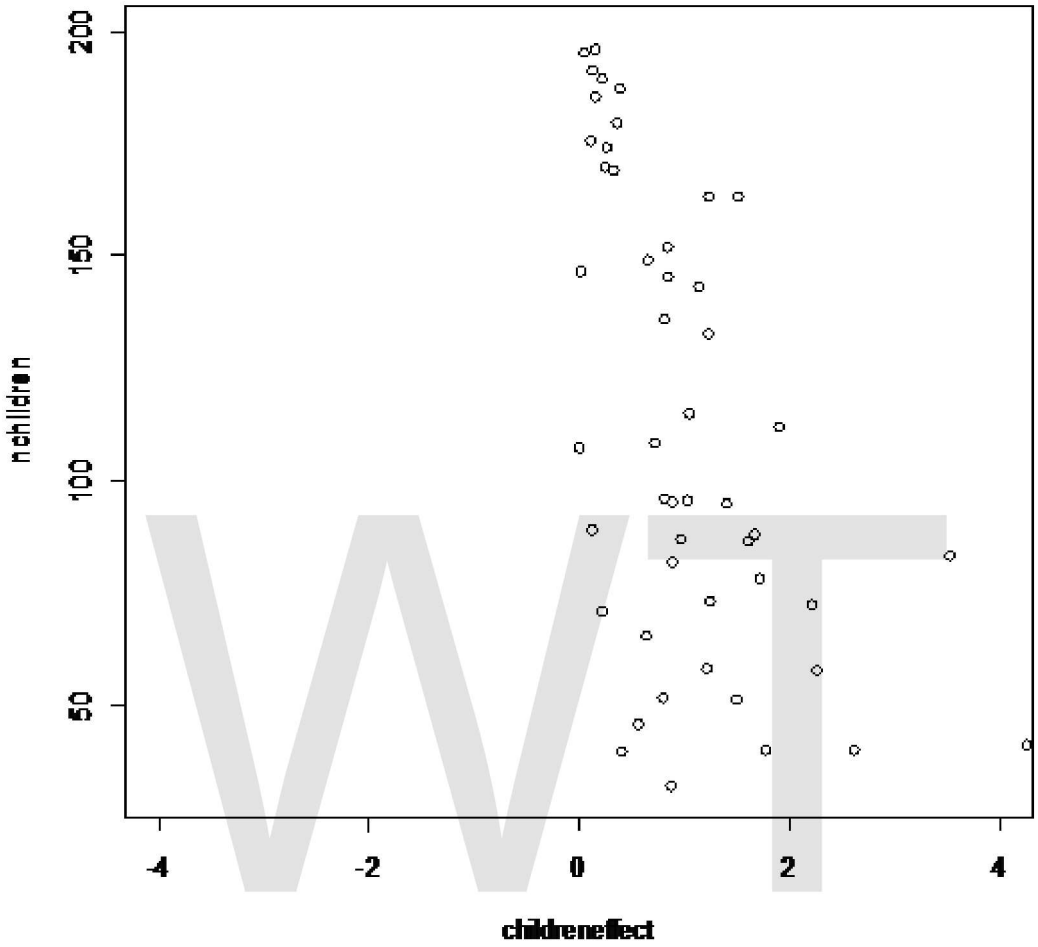
Dangers of agenda-driven bias

The most severe weakness and abuse of meta-analysis often occurs when the person or persons doing the meta-analysis have an economic, social,or political agenda such as the passage or defeat of legislation. Those persons with these types of agenda have a high likelihood to abuse meta-analysis due to personal bias. For example, researchers favorable to the author's agenda are likely to have their studies "cherry picked" while those not favorable will be ignored or labeled as "not credible". In addition, the favored authors may themselves be biased or paid to produce results that support their overall political, social, or economic goals in ways such as selecting small favorable data sets and not incorporating larger unfavorable data sets.

If a meta-analysis is conducted by an individual or organization with a bias or predetermined desired outcome, it should be treated as highly suspect or having a high likelihood of being "junk science". From an integrity perspective, researchers with a bias should avoid meta-analysis and use a less abuse-prone (or independent) form of research.



A funnelplot expected without the file drawer problem



A funnelplot expected with the file drawer problem

Benefits of usability

The key benefits of usability are:

- Higher revenues through increased sales
- Increased user efficiency and satisfaction
- Reduced development costs
- Reduced support costs

Corporate integration

An increase in usability generally positively affects several facets of a company's output quality. In particular, the benefits fall into several common areas:

- Increased productivity
- Decreased training and support costs
- Increased sales and revenues
- Reduced development time and costs
- Reduced maintenance costs
- Increased customer satisfaction

Increased usability in the workplace fosters several responses from employees. Along with any positive feedback, “workers who enjoy their work do it better, stay longer in the face of temptation, and contribute ideas and enthusiasm to the evolution of enhanced productivity." In order to create standards, companies often implement experimental design techniques that create baseline levels. Areas of concern in an office environment include (though are not necessarily limited to):

- Working Posture
- Design of Workstation Furniture
- Screen Displays
- Input Devices
- Organizational Issues
- Office Environment
- Software Interface

By working to improve said factors, corporations can achieve their goals of increased output at lower costs, while potentially creating optimal levels of customer satisfaction. There are numerous reasons why each of these factors correlates to overall improvement. For example, making a piece of software’s user interface easier to understand would reduce the need for extensive training. The improved interface would also tend to lower the time needed to perform necessary tasks, and so would both raise the productivity levels for employees and reduce development time (and thus costs). It is important to note that each of the aforementioned factors are not mutually exclusive, rather should be understood to work in conjunction to form the overall workplace environment.

Conclusion

Usability is now recognized as an important software quality attribute, earning its place among more traditional attributes such as performance and robustness. Various academic programs focus on usability. Several usability consultancy companies have emerged, and traditional consultancy and design firms offer similar services.

Professional development

Usability practitioners are sometimes trained as industrial engineers, psychologists, kinesiologists, systems design engineers, or with a degree in information architecture, information or library science, or Human-Computer Interaction (HCI). More often though they are people who are trained in specific applied fields who have taken on a usability

focus within their organization. Anyone who aims to make tools easier to use and more effective for their desired function within the context of work or everyday living can benefit from studying usability principles and guidelines.

For those seeking to extend their training, the Usability Professionals' Association offers online resources, reference lists, courses, conferences, and local chapter meetings. The UPA also sponsors World Usability Day each November.

Related professional organizations include the Human Factors and Ergonomics Society (HFES) and the Association for Computing Machinery's special interest groups in Computer Human Interaction (SIGCHI), and Computer Graphics and Interactive Techniques (SIGGRAPH).

The Society for Technical Communication also has a special interest group on Usability and User Experience (UUX). They publish a quarterly newsletter called *Usability Interface*.

Chapter- 4

Universal Usability and Usability Testing

Universal Usability

Universal usability refers to the design of information and communications products and services that are usable for every citizen. The concept has been advocated by Professor Ben Shneiderman, a computer scientist at the University of Maryland, College Park. He also provided a more practical definition of universal usability – "having more than 90% of all households as successful users of information and communications services at least once a week." The concept of universal usability ("usable by all") is closely related to the concepts of universal accessibility ("accessible by all") and universal design ("design for all"). These three concepts altogether cover, from the user's end to the developer's end, the three important research areas of information and communications technology (ICT): use, access, and design.

Challenges to universal usability

There are three major challenges to universal usability:

1. Supporting a broad range of hardware, software, and network access. With the advance of ICT, users' hardware, software, and network configurations are changing. The variety of ICT products creates complex systems with a broad range of hybridity. For example, would a software product be usable to users running Windows XP on a Centrino laptop with broadband Internet access and to those who have Windows 98 on a Pentium II desktop with 56K dial-up?
2. Accommodating individual differences among users, such as age, gender, disabilities, literacy, culture, income, and so forth. Individual differences can be roughly categorized into three types: physical, cognitive, and socio-cultural. In the field of HCI, research attempts have been centering on accommodating physical and cognitive differences by isolating various specific factors such as spatial ability, speed of movement, eye-hand coordination, and so forth. However, previous literature has demonstrated that individual differences are difficult to pin down and difficult to generalize from one context to another.
3. Bridging the knowledge gap between what users know and what they need to know about a specific system. Two issues need to be resolved: 1) Building a user

model to access individual user's background knowledge on a specific system; 2) Integrating the mechanism of evolutionary learning.

Principles of universal usability design

The key to universal usability is recognizing the diversity of user population and user needs. There is no "average" user on whom a system should be based. Although in some cases it is possible to accommodate technology variety and individual differences in one system, multi-layer designs are the most promising approach to achieving universal usability. That is, when a single design cannot accommodate a large fraction of the user population, multiple versions or adjustment controls should be available to users. For example, a novice user can be provided with only a few options; after gaining confidence and experience, the user can choose to progress to higher levels of tasks and the accompanying interface.

Sarah Horton has developed a set of universal usability guidelines for web design. The basic principles are:

- Design simply: Design simple sites, emphasizing important elements and using simple structures and clean, standards-based markup.
- Build well: Take full advantage of these inherent properties, such as fallbacks, flexibility, and user control, to construct universally usable Web sites.
- Favor HTML over other formats: Html is the best format for universal usability. Provide documents in nonstandard formats, such as PDF and Flash, only as an alternative to accessible html.

Harry Hochheiser and Ben Shneiderman have also developed the Universal Usability Statement Template, which describes a Web site's content, browser requirements, network requirements, and other characteristics that may influence its usability.

Electronic Curb-Cuts

The analogy "curb-cut" has been used by advocates of universal usability to explain how ICT products designed for disabled users can be beneficial to all users. Sidewalk curb-cuts are added to accommodate wheelchair users, but the benefits extend to baby carriage pushers, delivery service workers, bicyclists, and travelers with roller bags. In the context of ICT design and development, universal usability is often tied to meeting the needs of people with disabilities. The adaptability needed for users with physical, visual, auditory, or cognitive disabilities is likely to benefit users with differing preferences, tasks, hardware, etc. Hence, electronic curb-cuts – system functions that are designed for people with disabilities – may be usable by everyone in various usage situations. It might be expensive to transform an existing system to meet universal usability standards, but the extra cost of integrating electronic curb-cuts into a new system can be minimalized.

Current Research Development

Current trends in the Universal Usability research include:

- Multimodal or adaptive user interface
- Universal usability of commercial and e-government websites
- Interface solutions for older adult users and users with disabilities
- Contextualization of universal usability

Usability testing

Usability testing is a technique used to evaluate a product by testing it on users. This can be seen as an irreplaceable usability practice, since it gives direct input on how real users use the system. This is in contrast with usability inspection methods where experts use different methods to evaluate a user interface without involving users.

Usability testing focuses on measuring a human-made product's capacity to meet its intended purpose. Examples of products that commonly benefit from usability testing are foods, consumer products, web sites or web applications, computer interfaces, documents, and devices. Usability testing measures the usability, or ease of use, of a specific object or set of objects, whereas general human-computer interaction studies attempt to formulate universal principles.

History of usability testing

A Xerox Palo Alto Research Center (PARC) employee wrote that PARC used extensive usability testing in creating the Xerox Star, introduced in 1981.] Only about 25,000 were sold, leading many to consider the Xerox Star a commercial failure.

The Inside Intuit book, says (page 22, 1984), "... in the first instance of the Usability Testing that later became standard industry practice, LeFevre recruited people off the streets... and timed their Kwik-Chek (Quicken) usage with a stopwatch. After every test... programmers worked to improve the program.") Scott Cook, Intuit co-founder, said, "... we did usability testing in 1984, five years before anyone else... there's a very big difference between doing it and having marketing people doing it as part of their... design... a very big difference between doing it and having it be the core of what engineers focus on.

Goals of usability testing

Usability testing is a black-box testing technique. The aim is to observe people using the product to discover errors and areas of improvement. Usability testing generally involves measuring how well test subjects respond in four areas: efficiency, accuracy, recall, and emotional response. The results of the first test can be treated as a baseline or control measurement; all subsequent tests can then be compared to the baseline to indicate improvement.

- *Performance* -- How much time, and how many steps, are required for people to complete basic tasks? (For example, find something to buy, create a new account, and order the item.)
- *Accuracy* -- How many mistakes did people make? (And were they fatal or recoverable with the right information?)
- *Recall* -- How much does the person remember afterwards or after periods of non-use?
- *Emotional response* -- How does the person feel about the tasks completed? Is the person confident, stressed? Would the user recommend this system to a friend?

What usability testing is not

Simply gathering opinions on an object or document is market research rather than usability testing. Usability testing usually involves systematic observation under controlled conditions to determine how well people can use the product.

Rather than showing users a rough draft and asking, "Do you understand this?", usability testing involves watching people trying to *use* something for its intended purpose. For example, when testing instructions for assembling a toy, the test subjects should be given the instructions and a box of parts. Instruction phrasing, illustration quality, and the toy's design all affect the assembly process.

Methods

Setting up a usability test involves carefully creating a scenario, or realistic situation, wherein the person performs a list of tasks using the product being tested while observers watch and take notes. Several other test instruments such as scripted instructions, paper prototypes, and pre- and post-test questionnaires are also used to gather feedback on the product being tested. For example, to test the attachment function of an e-mail program, a scenario would describe a situation where a person needs to send an e-mail attachment, and ask him or her to undertake this task. The aim is to observe how people function in a realistic manner, so that developers can see problem areas, and what people like. Techniques popularly used to gather data during a usability test include think aloud protocol and eye tracking.

Hallway testing

Hallway testing (or **Hall Intercept Testing**) is a general methodology of usability testing. Rather than using an in-house, trained group of testers, just five to six random people, indicative of a cross-section of end users, are brought in to test the product, or service. The name of the technique refers to the fact that the testers should be random people who pass by in the hallway.

Remote testing

Remote usability testing (also known as unmoderated or asynchronous usability testing) involves the use of a specially modified online survey, allowing the quantification of user testing studies by providing the ability to generate large sample sizes. Similar to an in-lab study, a remote usability test is task-based and the platforms allow you to capture clicks and task times. Hence, for many large companies this allows you to understand the WHY behind the visitors intents when visiting a website or mobile site. Additionally, this style of user testing also provides an opportunity to segment feedback by demographic, attitudinal and behavioural type. The tests are carried out in the user's own environment (rather than labs) helping further simulate real-life scenario testing. This approach also provides a vehicle to easily solicit feedback from users in remote areas.

Expert Review

Expert Review is another general method of usability testing. As the name suggests, this method relies on bringing in experts with experience in the field (possibly from companies that specialize in usability testing) to evaluate the usability of a product.

Automated Expert Review

Similar to Expert Reviews, **Automated Expert Reviews** provide usability testing but through the use of programs given rules for good design and heuristics. Though an automated review might not provide as much detail and insight as reviews from people, they can be finished more quickly and consistently. The idea of creating surrogate users for usability testing is am ambitious direction for the Artificial Intelligence community.

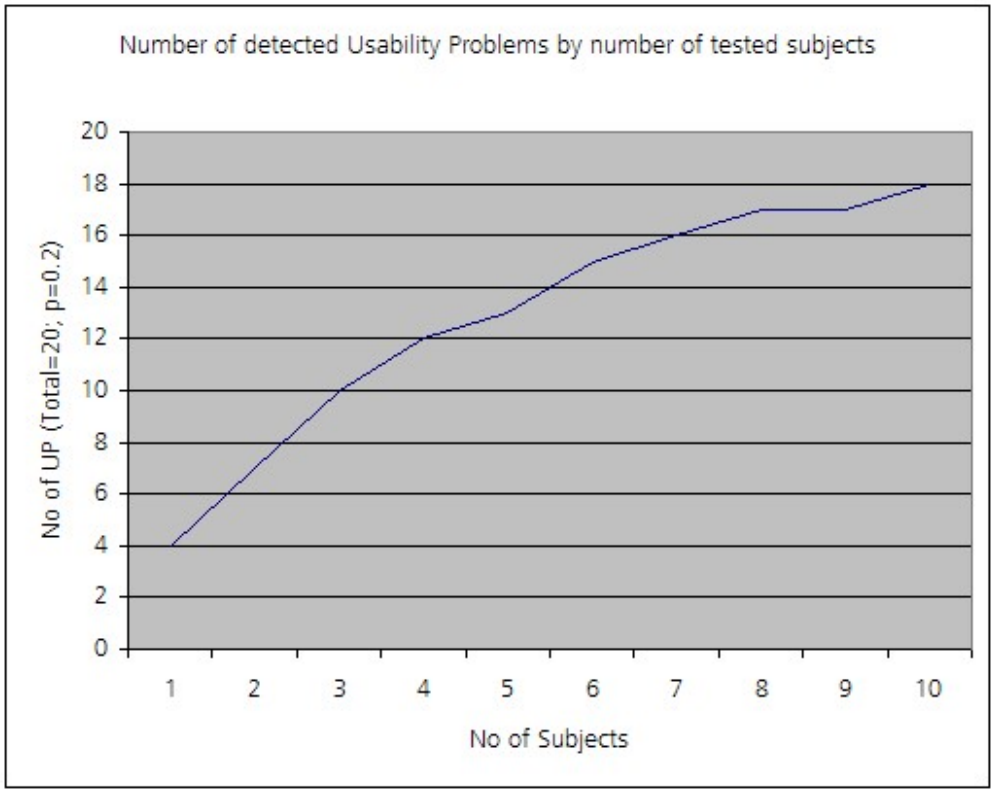
How many users to test?

In the early 1990s, Jakob Nielsen, at that time a researcher at Sun Microsystems, popularized the concept of using numerous small usability tests—typically with only five test subjects each—at various stages of the development process. His argument is that, once it is found that two or three people are totally confused by the home page, little is gained by watching more people suffer through the same flawed design. "Elaborate usability tests are a waste of resources. The best results come from testing no more than five users and running as many small tests as you can afford." Nielsen subsequently published his research and coined the term heuristic evaluation.

The claim of "Five users is enough" was later described by a mathematical model which states for the proportion of uncovered problems U

$$U = 1 - (1 - p)^n$$

where p is the probability of one subject identifying a specific problem and n the number of subjects (or test sessions). This model shows up as an asymptotic graph towards the number of real existing problems (see figure below).



In later research Nielsen's claim has eagerly been questioned with both empirical evidence and more advanced mathematical models. Two key challenges to this assertion are:

1. since usability is related to the specific set of users, such a small sample size is unlikely to be representative of the total population so the data from such a small sample is more likely to reflect the sample group than the population they may represent
2. Not every usability problem is equally easy-to-detect. Intractable problems happen to decelerate the overall process. Under these circumstances the progress of the process is much shallower than predicted by the Nielsen/Landauer formula.

Most researchers and practitioners today agree that, although testing 5 users is better than not testing at all, a sample size larger than five is required to detect a satisfying number of usability problems.

System Usability Scale

The **System Usability Scale** (SUS) in systems engineering is a simple, ten-item attitude Likert scale giving a global view of subjective assessments of usability. It was developed by John Brooke at Digital Equipment Corporation in the UK in 1986 as a tool to be used in usability engineering of electronic office systems.

The usability of a system, as defined by the ISO standard ISO 9241 Part 11, can be measured only by taking into account the context of use of the system — i.e., who is using the system, what they are using it for, and the environment in which they are using it. Furthermore, measurements of usability have several different aspects:

- effectiveness (can users successfully achieve their objectives)
- efficiency (how much effort and resource is expended in achieving those objectives)
- satisfaction (was the experience satisfactory)

Measures of effectiveness and efficiency are also context specific. Effectiveness in using a system for controlling a continuous industrial process would generally be measured in very different terms to, say, effectiveness in using a text editor. Thus, it can be difficult, if not impossible, to answer the question "is system A more usable than system B", because the measures of effectiveness and efficiency may be very different. However, it can be argued that given a sufficiently high-level definition of *subjective* assessments of usability, comparisons can be made between systems.

SUS has generally been seen as providing this type of high-level subjective view of usability and is thus often used in carrying out comparisons of usability between systems. Because it yields a single score on a scale of 0-100, it can be used to compare even systems that are outwardly dissimilar. This one-dimensional aspect of the SUS is both a benefit and a drawback, because the questionnaire is necessarily quite general. Recently, Lewis and Sauro suggested a two-factor orthogonal structure, which practitioners may use to score the SUS on independent Usability and Learnability dimensions. At the same time, Borsci, Federici and Lauriola by an independent analysis confirm the two factors structure of SUS, also showing that those factors (Usability and Learnability) are correlated.

The SUS has been widely used in the evaluation of a range of systems. Bangor, Kortum and Miller have used the scale extensively over a ten year period and have produced normative data that allow SUS ratings to be positioned relative to other systems. They propose an extension to SUS to provide an adjective rating that correlates with a given score.

World Usability Day

World Usability Day or "Make Things Easier Day" promotes the value of usability, usability engineering, user-centered design, universal usability, and every user's responsibility to ask for things that work better. It was initiated, and is still sponsored by, the Usability Professionals' Association, a group dedicated to "those who promote and advance the development of usable products, reaching out to people who act as advocates for usability and the user experience." In 2008, there were over 200 events in over 43 countries.

World Usability Day is held annually on the second Thursday of November. In 2009, this is November 12. Each year focuses on a different theme. For 2009 the theme is "Designing for a Sustainable World" using the "Cradle to Cradle" approach.

From the World Usability Day official website:

"World Usability Day 2009 is approaching design from Cradle to Cradle. Coming from a user-centric perspective and looking beyond form and function, we are exploring the impact design has on our World. The ‘Cradle to Cradle’ approach is to start the design with the premise of using materials that can fully enter a new life cycle by either going back to nature or going back into the design process as a new product. This holistic approach to sustainable design shows how usability can apply to all of what we do and build. Designing for a Sustainable World focuses on how our products and services impact our world. We look at all products and services, whether they are buildings, roads, consumer products, business, services or healthcare systems; throughout their life cycle. The impact focuses on - our environment, energy, water, soil, and more. Have the materials and processes that have been used been recycled and are they re-usable? Are they user and environmentally friendly? These are questions we all must consider as we design, purchase, use and dispose of products each and every day."

Historic events highlights

Details on all historic events can be accessed and searched on the World Usability Day official website.

2005

- The first World Usability Day
- over 36 hours of content in 115 events around the world
- 35 countries
- Approximately 8,000 people attended events worldwide
- over 200 online participants from 16 countries

2006

- 39 countries

Additionally, World Usability Day New England 2006 focused on universal usability to enhance learning, effectiveness, and understanding for people of all abilities.

2007

- Theme: Healthcare
- 41 countries involved
- 10,000 volunteers
- approximately 40,000 participants

2008

- Theme: Transportation
- 43 countries
- over 50,000 participants and volunteers

2009

- Theme: Designing for a Sustainable World

Chapter- 5

Usability Engineering

Usability engineering is a field that is concerned generally with human-computer interaction and specifically with making human-computer interfaces that have high usability or user friendliness. In effect, a user-friendly interface is one that allows users to effectively and efficiently accomplish the tasks for which it was designed and one that users rate positively on opinion or emotional scales. Assessing the usability of an interface and recommending ways to improve it is the purview of the Usability Engineer. The largest subsets of Usability Engineers work to improve usability of software graphical user interfaces (GUIs), web-based user interfaces, and voice user interfaces (VUIs).

Several broad disciplines including Psychology, Human Factors and Cognitive Science subsume usability engineering, but the theoretical foundations of the field come from more specific domains: human perception and action; human cognition; behavioral research methodologies; and, to a lesser extent, quantitative and statistical analysis techniques.

When usability engineering began to emerge as a distinct area of professional practice in the mid- to late 1980s, many usability engineers had a background in Computer Science or in a sub-field of Psychology such as Perception, Cognition or Human Factors. Today, these academic areas still serve as springboards for the professional practitioner of usability engineering, but Cognitive Science departments and academic programs in Human-Computer Interaction now also produce their share of practitioners in the field.

The term **usability engineering** (in contrast to interaction design and user experience design) implies more of a focus on assessing and making recommendations to improve usability than it does on design, though Usability Engineers may still engage in design to some extent, particularly design of wire-frames or other prototypes.

Standards and Guidelines

Usability engineers sometimes work to shape an interface such that it adheres to accepted operational definitions of user requirements. For example, the International Organisation for Standardisation-approved definitions usability are held by some to be a context-dependent yardstick for the effectiveness, efficiency and satisfaction with which specific users should be able to perform tasks. Advocates of this approach engage in task analysis,

then prototype interface design, and usability testing on those designs. On the basis of such tests, the technology is (ideally) re-designed or (occasionally) the operational targets for user performance are revised. [Dillon, 2000].

The National Institute of Standards and Technology has collaborated with industry to develop the Common Industry Specification for Usability - Requirements which serves as a guide for many industry professionals. The specifications for successful usability in biometrics were also developed by the NIST. Usability.Gov provides a tutorial and wide general reference for the design of usable websites.

Usability, especially with the goal of Universal Usability, encompasses the standards and guidelines of design for accessibility. Some primary guidelines for web accessibility are:
1. The Web Accessibility Initiative Guidelines
2. The Section 508 government guidelines applicable to all public-sector websites.
3. The ADA Guidelines for accessibility of state and local government websites.
4. The IBM Guidelines for accessibility of websites.

Methods and Tools

Usability Engineers conduct usability evaluations of existing or proposed interfaces and their findings are fed back to the Designer for use in design or redesign. Common usability evaluation methods include:

- usability testing (Gold standard of Usability Engineering, but the most involved and expensive method)
- interviews
- focus groups
- questionnaires
- cognitive walkthroughs
- heuristic evaluations
- RITE method
- cognitive task analysis
- contextual inquiry
- Think aloud protocol

Usability testing, the gold standard, is when participants are recruited and asked to use the actual or prototype interface and their reactions, behaviors, errors, and self-reports in interviews are carefully observed and recorded by the Usability Engineer. On the basis of this data, the Usability Engineer recommends interface changes to improve usability.

There are a variety of online resources that make the job of the Usability Engineer a little easier. Some examples of these include:
1. The Web Metrics Tool Suite is a product of the National Institute of Standards and Technology. This toolkit is focused on evaluating the HTML of a website versus a wide range of usability guidelines and includes:

- Web Static Analyzer Tool (WebSAT) - checks web page HTML against typical usability guidelines
- Web Category Analysis Tool (WebCAT) - lets the usability engineer construct and conduct a web category analysis
- Web Variable Instrumenter Program (WebVIP) - instruments a website to capture a log of user interaction
- Framework for Logging Usability Data (FLUD) - a file format and parser for representation of user interaction logs
- FLUDViz Tool - produces a 2D visualization of a single user session
- VisVIP Tool - produces a 3D visualization of user navigation paths through a website
- TreeDec - adds navigation aids to the pages of a website

2. The Usability Testing Environment (UTE) produced by Mind Design Systems is available freely to federal government employees. According to the official company website this tool

"consists of two tightly-integrated applications. The first is the UTE Manager which helps a tester set up test scenarios (tasks) as well as survey and demographic questions. The UTE Manager also compiles the test results and produces customized reports and summary data which can be used as quantitative measures of usability observations and recommendations. The second UTE application is the UTE Runner. The UTE Runner presents the test participants with the test scenarios (tasks) as well as any demographic and survey questions. In addition, the UTE Runner tracks the actions of the subject throughout the test including clicks, keystrokes, and scrolling."

3. The UsableNet Liftmachine is a product of UsableNet.com and implements the section 508 Usability and Accessibility guidelines as well as the W3C Web Accessibility Initiative Guidelines.

It is important to remember that online tools are only a useful tool, and do not substitute for a complete Usability Engineering analysis.

Research Resources

Some well-known practitioners in the field are Donald Norman, Jakob Nielsen, and John M. Carroll. Nielsen and Carroll have both written books on the subject of usability engineering. Nielsen's book is aptly titled *Usability Engineering*, and was published in 1993. Carroll wrote "Making Use: Scenario-Based Design of Human-Computer Interactions" in 2000, and co-authored "Usability Engineering: Scenario-Based Development of Human-Computer Interaction" with Mary Beth Rossen in 2001. Some other field leaders are Alan Cooper, Larry Constantine and Steve Krug the author of "*Don't Make Me Think! A Common Sense Approach to Web Usability*".

There are many books written on Usability Engineering. A few of the more popular recently published books are as follows:

1. 1993 - **Usability engineering** by Jakob Nielsen - 362 pages
2. 1999 - **Web site usability: a designer's guide** by Jared M. Spool, Tara Scanlon - 157 pages
3. 1999 - **The Usability Engineering Lifecyle: A Practitioner's Handbook** by Deborah J. Mayhew - 542 pages
4. 2000 - **Usability Engineering** by Xristine Faulkner - 244 pages
5. 2001 - **Usability Evaluation and Interface Design: Cognitive Engineering ...** by Michael James Smith, Richard John Koubek, Gavriel Salvendy, Don Harris - 1592 pages
6. 2002 - **Usability engineering: scenario-based development of human-computer interaction** by Mary Beth Rosson, John Millar Carroll - 422
7. 2003 - **The human-computer interaction handbook: fundamentals, evolving technologies ...** by Julie A. Jacko, Andrew Sears - 1277 pages
8. 2007 - **Usability Engineering: Process, Products, and Examples** by Laura M. Leventhal, Julie Barnes - 314 pages
9. 2008 - **Adoption-centric Usability Engineering: Systematic Deployment, Assessment ...** by Ahmed Seffah, Eduard Metzker - Computers - 450 pages
10. 2008 - **The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies ...** by Andrew Sears, Julie A. Jacko - 1358 pages

Interaction Design Pattern

Interaction design patterns are a way to capture optimal solutions to common usability or accessibility problems in a specific context. They document interaction models that make it easier for users to understand an interface and accomplish their tasks.

History

Patterns originated as an architectural concept by Christopher Alexander. Patterns are ways to describe best practices, explain good designs, and capture experience so that other people can reuse these solutions.

Design patterns in computer science are used by software engineers during the actual design process and when communicating designs to others. Design patterns gained popularity in computer science after the book *Design Patterns: Elements of Reusable Object-Oriented Software* was published. Since then a pattern community has emerged that specifies patterns for problem domains including architectural styles and object-oriented frameworks. The Pattern Languages of Programming Conference (annual, 1994—) proceedings includes many examples of domain specific patterns.

Applying a pattern language approach to interaction design was first suggested in Norman and Draper's book *User Centered System Design* (1986). The Apple Computer's Macintosh Human Interface Guidelines also quotes Christopher Alexander's works in its recommended reading.

Interaction Design Pattern Libraries

Alexander envisioned a pattern language as a structured system in which the semantic relationships between the patterns create a whole that is greater than the sum of its parts, much like the way that grammatical relationships between words make language meaningful. While some collections of patterns attempt to create the structural relationships needed to form a language, many others are simply an assemblage of patterns (and thus are more appropriately termed pattern libraries.)

Many online pattern libraries for interaction design exist today:

- Designing Interfaces Pattern Language by Jenifer Tidwell; based on earlier work such as:
  - UI Patterns and Techniques repository and
  - Common Ground pattern language
- Patterns in Interaction Design library by Martijn van Welie
- Patternry by Janne Lammi
- UI Patterns library by Anders Toxboe
- Quince UX Patterns Explorer by Infragistics

Elements of an interaction design pattern

For patterns to be helpful to the designers and developers who will make use of them, they need to be findable and readable.

Common Elements

Though pattern descriptions vary somewhat, many pattern libraries include some common elements:

- **Pattern Name:** Choosing a clear and descriptive name helps people find the pattern and encourages clear communication between team members during design discussions.
- **Pattern Description:** Because short names like "one-window drilldown" are sometimes not sufficient to describe the pattern, a few additional lines of explanation (or a canonical screenshot) will help explain how the pattern works.
- **Problem Statement:** Written in user-centered language, this communicates what the user wants to achieve or what the challenge is to the end-user.
- **Use When:** "Context of use" is a critical component of the design pattern. This element helps people understand situations when the design pattern applies (and when it does not.)
- **Solution:** The solution should explain "how" to solve the problem, and may include prescriptive checklists, screenshots, or even short videos demonstrating the pattern in action.

- **Rationale:** Providing reasons "why" the pattern works will reinforce the solution, though time-pressed developers may prefer to ignore this explanation.
- **Examples:** Each example shows how the pattern has been successfully applied This is often accompanied by a screenshot and a short description.
- **Comments:** Including a place for team members to discuss the use of the pattern helps maintain an active resource and keeps the team engaged.

Optional Elements

Pattern libraries can also include optional elements, depending on the needs of the team using them. These may include:

- **Implementation Specifications:** A style guide with detailed information about font sizes, pixel dimensions, colors, and wording for messages and labels can be helpful for developers.
- **Usability Research:** Any supporting research from usability tests or other user feedback should be captured. This can also include feedback from developers, customer service, or the sales team.
- **Related Patterns**: The pattern library may include similar patterns, or it may be organized into a hierarchy with parent and child patterns.
- **Similar Approaches**: Since there are likely to be many possible solutions to this problem, teams may want a place to capture similar alternatives.
- **Source Code**: If the code is modular enough to be reused, then it can be included in the library as well.

Reasons to use design patterns

Benefits of using interaction design patterns include:

- Teaching novices some best practices and common approaches
- Capturing collective wisdom of designers across many uses and scenarios
- Giving teams a common language, reducing misunderstandings that arise from different vocabulary
- Reducing time and costs in the design and development lifecycle
- Making usable designs the "path of least resistance"
- Eliminate wasted time spent "reinventing the wheel"
- Ensuring users have a consistent and predictable experience within an application or service

Advantages over design guidelines

Guidelines are generally more useful for describing requirements whereas patterns are useful tools for those who need to translate requirements to specific software solutions. Some people consider design guidelines as an instance of interaction design pattern as

they are also common approach of capturing experience in interaction design. However, interaction design patterns usually have the following advantages over design guidelines:

1. Abstract guidelines, like the *Eight Golden Rules of Interface Design* by Shneiderman, do not suggest how to solve a problem like many interaction design pattern, and cannot be used for interdisciplinary communication. Furthermore, guidelines do not provide an explanation as to why a particular solution works.
2. Concrete guidelines, like Macintosh Human Interface Guidelines, are too tailored to a specific interface, and therefore are not as effective when applied to other interfaces (especially non-Macintosh interfaces).
3. Other problems with guidelines are that they tend to be too numerous which makes it difficult for designers to apply the right guidelines. Also guidelines assume an absolute validity while usually they can only be applied in a particular context. A result of that is also that guidelines often tend to conflict just because they lack describing a context.

Guidelines and patterns are not necessarily conflicting, and both can be used in conjunction to identify the problem and then create a valid solution.

Machine translation software usability

The sections below give objective criteria for evaluating the usability of machine translation software output.

Stationarity or Canonical Form

Do repeated translations converge on a single expression in both languages? I.e. does the translation method show stationarity or produce a canonical form. Does the translation become stationary without losing the original meaning? This metric has been criticized as not being well correlated with BLEU (BiLingual Evaluation Understudy) scores

Adaptive to colloquialism, argot or slang

Is the system adaptive to colloquialism, argot or slang? The French language has many rules for creating words in the speech and writing of popular culture. Two such rules are: (a) The reverse spelling of words such as *femme* to *meuf*. (This is called verlan.) (b) The attachment of the suffix *-ard* to a noun or verb to form a proper noun. For example, the noun *faluche* means "student hat". The word *faluchard* formed from *faluche* colloquially can mean, depending on context, "a group of students", "a gathering of students" and "behavior typical of a student". The Google translator as of 28 December 2006 doesn't derive the constructed words as for example from rule (b), as shown here:

Il y a une chorale falucharde mercredi, venez nombreux, les faluchards chantent des paillardes! ==> *There is a choral society falucharde Wednesday, come many, the faluchards sing loose-living women!*

French argot has three levels of usage:

1. *familier* or friendly, acceptable among friends, family and peers but not at work
2. *grossier* or swear words, acceptable among friends and peers but not at work or in family
3. *verlan* or ghetto slang, acceptable among lower classes but not among middle or upper classes

The United States National Institute of Standards and Technology conducts annual evaluations of machine translation systems based on the BLEU-4 criterion . A combined method called IQmt which incorporates BLEU and additional metrics NIST, GTM, ROUGE and METEOR has been implemeneted by Gimenez and Amigo .

Well-formed output

Is the output grammatical or well-formed in the target language? Using an interlingua should be helpful in this regard, because with a fixed interlingua one should be able to write a grammatical mapping to the target language from the interlingua. Consider the following Arabic language input and English language translation result from the Google translator as of 27 December 2006 . This Google translator output doesn't parse using a reasonable English grammar:

وعن حوادث التدافع عند شعيرة رمي الجمرات -التي كثيرا ما يسقط فيها العديد من الضحايا- أشار الأمير نايف إلى إدخال "تحسينات كثيرة في جسر الجمرات ستمنع بإذن الله حدوث أي تزاحم". ==> And incidents at the push Carbuncles-throwing ritual, which often fall where many of the victims - Prince Nayef pointed to the introduction of "many improvements in bridge Carbuncles God would stop the occurrence of any competing."

Semantics preservation

Do repeated re-translations preserve the semantics of the original sentence? For example, consider the following English input passed multiple times into and out of French using the Google translator as of 27 December 2006:

Better a day earlier than a day late. ==> *Améliorer un jour plus tôt qu'un jour tard.* ==> To improve one day earlier than a day late. ==> *Pour améliorer un jour plus tôt qu'un jour tard.* ==> To improve one day earlier than a day late.

As noted above and in, this kind of round-trip translation is a very unreliable method of evaluation.

Trustworthiness and Security

An interesting peculiarity of Google Translate as of 24 January 2008 (corrected as of 25 January 2008) is the following result when translating from English to Spanish, which shows an embedded joke in the English-Spanish dictionary which has some added poignancy given recent events:

Heath Ledger is dead ==> *Tom Cruise está muerto*

This raises the issue of trustworthiness when relying on a machine translation system embedded in a Life-critical system in which the translation system has input to a Safety Critical Decision Making process. Conjointly it raises the issue of whether in a given use the software of the machine translation system is safe from hackers.

It is not known whether this feature of Google Translate was the result of a joke/hack or perhaps an unintended consequence of the use of a method such as statistical machine translation. Reporters from CNET Networks asked Google for an explanation on January 24, 2008; Google said only that it was an "internal issue with Google Translate". The mistranslation was the subject of much hilarity and speculation on the Internet.

If it is an unintended consequence of the use of a method such as statistical machine translation, and not a joke/hack, then this event is a demonstration of a potential source of critical unreliability in the statistical machine translation method.

In human translations, in particular on the part of interpreters, selectivity on the part of the translator in performing a translation is often commented on when one of the two parties being served by the interpreter knows both languages.

This leads to the issue of whether a particular translation could be considered *verifiable*. In this case, a converging round-trip translation would be a kind of verification.