

checklists y guías de usabilidad

Cuatro principios de diseño

1. Foco temprano y continuo en el usuario
2. Testeo temprano y continuo por el usuario
3. Diseño iterativo
4. Diseño integrado

Métodos para llevar a cabo un testeo temprano y continuo por el usuario.

Ej. de Checklist

- Esquemas, o bocetos preliminares en pocos escenarios.
- Comenzamos a escribir el manual de usuario, que es una guía en el proceso de desarrollo.
- Comenzamos a usar simulaciones para ver las funciones y organización de la interfaz.
- Usar maquetas.
- Se hicieron demostraciones tempranas.
- Invitamos a personas a hacer comentarios del sistema.
- Invitamos a pensar en voz alta a futuros usuarios.
- Usamos métodos de corredor y vidriera.

Métodos para llevar a cabo un testeo temprano y continuo por el usuario.

Ej. de Checklist

- Usamos foros para obtener *feedback*
- Hicimos un testeo formal de prototipo
- Comparamos nuestros resultados con los objetivos establecidos de comportamiento.
- Alcanzamos nuestros objetivos de comportamiento
- Tratamos de que la gente encuentre bugs en nuestro sistema.
- Hicimos estudios de campo
- Hicimos seguimiento del uso del sistema

Guía de Apple para realizar tests con usuarios.

- Presentarse y explicar qué es testado y aclarar que es **testado el producto, no el usuario**.
- Establecer la duración del test y permitir la interrupción del mismo cuando quiera el usuario.
- Pedirles que piensen en voz alta, sobre lo que ven y sobre lo que están razonando.
- Explicar lo que el participante va a hacer, sin mostrar lo testado.
- Tratar que no haya ayuda o interferencias.
 - Si hay ayudas, especificar antes dónde y cuándo.
- Concluir la experiencia explicándole qué se busca con el test y contestando preguntas del usuario.
- Evitar pensar que los errores en el test son del usuario.
- Buscar patrones de comportamiento, no guiarse por excepciones.
- Analizar los resultados de forma multidisciplinaria y por los distintos responsables (“marketing”, documentación, programación, calidad, etc.)

Diseño Integrado

- Todos los aspectos de usabilidad deben evolucionar en paralelo
 - Interfaz de usuario, manual de usuario, los materiales de lectura, los sistemas de ayuda, etc.
- **Todos** los aspectos de la usabilidad están bajo un único foco, persona o **equipo**.

Métodos para llevar a cabo un Diseño Integrado.

- Diseño integrado es una aproximación al objetivo de usabilidad.
- La principal dificultad es organizacional.
 - Se debe reconocer la importancia de la usabilidad, que esta incluye muchos factores y que comienza desde el principio.
 - Requiere la existencia de un grupo, con los suficientes recursos para controlar la usabilidad y para **crear lo necesario** para que ésta sea buena.
- Una *checklist*.

Métodos para llevar a cabo un Diseño Integrado.

Ej. de Checklist

- Consideramos todos los aspectos de usabilidad en el diseño inicial.
- Una persona (con el apropiado número de gente) tiene la responsabilidad en la usabilidad.
- Manual de usuario.
- Manuales para grupos subsidiarios
- Identificación de las funciones requeridas.
- Interfaz de usuario.

Métodos para llevar a cabo un Diseño Integrado.

Ej. de Checklist

- Asegurar la adecuada confiabilidad y respuesta.
- Sistema de ayuda, material de entrenamiento, líneas de ayuda (*hot-lines*), videocintas, etc.
- Instalación
- Adecuación del sistema al usuario
- Sistema de mantenimiento
- Grupo de soporte de usuarios.

evaluación

- Un coprocesador de comportamiento altamente impredecible **-el usuario-** es agregado, en la realidad
- No hay datos sobre este coprocesador, por lo que hay que abandonar la idea de construir algoritmos basados en principios o axiomas.
- Una **aproximación empírica** es fundamental. Esta aproximación es muy superior a la necesaria para saber si el sistema “funciona”.

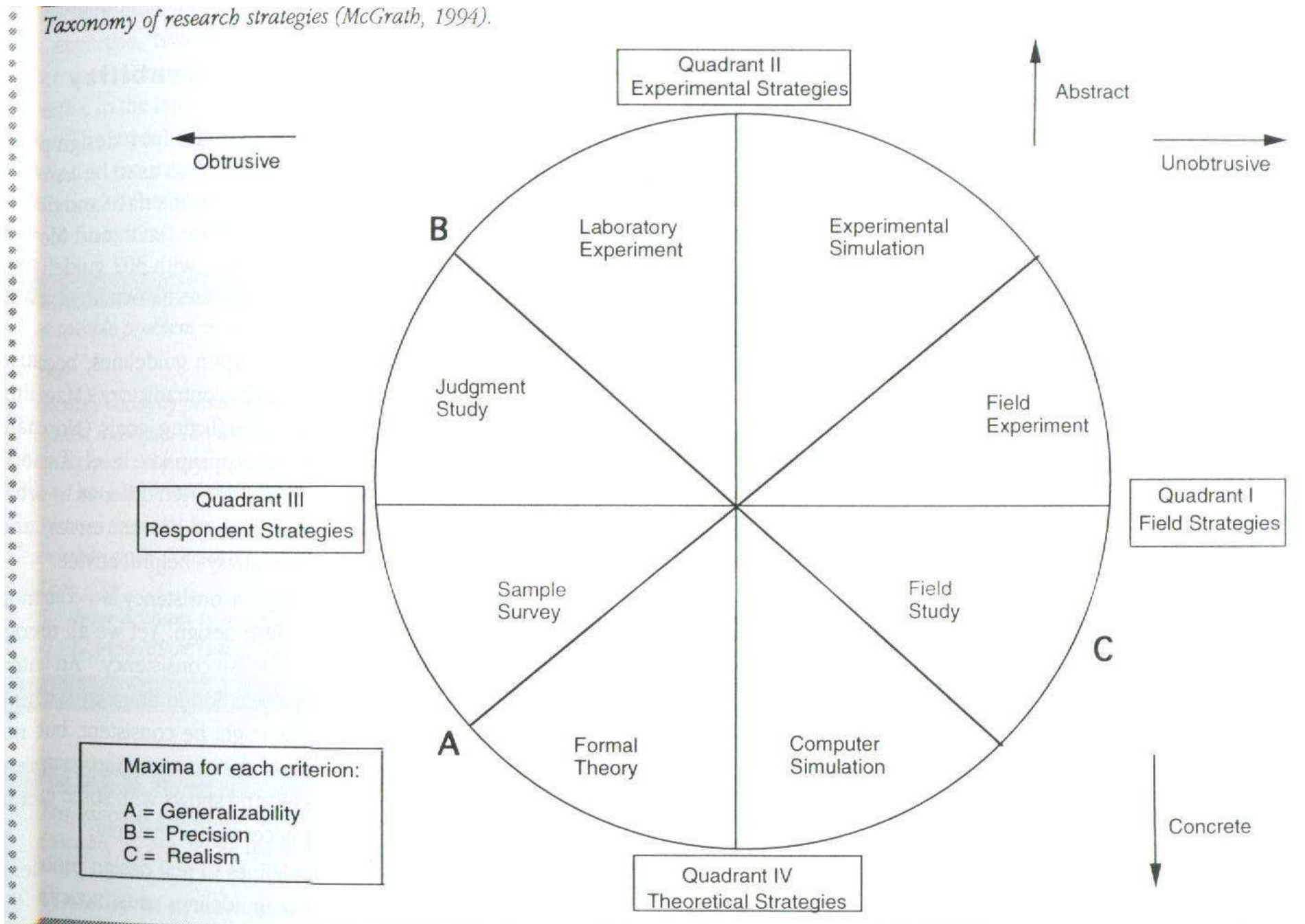
John Gould and Clayton Lewis

Evaluando sistemas y su interacción

- Consiste en evaluar los estados y acciones de las personas en el uso de las computadoras.
- Todos los métodos realizan un **balance** entre:
 - Generalización** de las evidencias medidas en la población,
 - Precisión** de las medidas de los comportamientos,
 - Realismo** del contexto donde se hace la medida en comparación con donde se quiere aplicar.

Evaluando sistemas y su interacción

Diseño y Evaluación



Ejemplo de guía de Nielsen

- 1) **visibilidad del estatus del sistema:** El sistema debe tener siempre informado a los usuarios de lo que está ocurriendo, con el feedback apropiado dentro del tiempo razonable.
- 2) **correspondencia entre el sistema y el mundo real:** El sistema debe hablar el lenguaje del usuario. Debe seguir convenciones del mundo real, hacer que la información aparezca en orden lógico y natural.
- 3) **control del usuario y libertad:** los usuarios a menudo eligen funciones del sistema por error y necesitarán una “salida de emergencia” para dejar el estado no buscado sin necesitar ir a través de un lenguaje extendido. Se debe soportar ***undo*** y ***redo***.

Ejemplo de guía de Nielsen

- 4) consistencia y estándares: Los usuarios no deben preguntarse si diferentes acciones, palabras, situaciones o acciones significan la misma cosa. Seguir las convenciones de plataforma.
- 5) prevención de errores: Mejor que mensajes de error es un cuidadoso diseño que prevenga la ocurrencia de problemas.
- 6) reconocimiento sobre recuerdo: Hacer visible los objetos, las acciones y las opciones. El usuario no tendría que recordar información de una parte del diálogo a otra. Instrucciones para el uso del sistema debería ser visible o fácilmente recuperable cuando se lo requiera.

Ejemplo de guía de Nielsen

- 7) **Flexibilidad y eficiencia de uso:** Aceleradores - invisibles para los usuarios novatos- pueden acelerar la interacción del usuario experto de forma que el sistema sirva tanto al novato como al experto.
- 8) **Estética y diseño minimalista:** Los diálogos no deben contener información irrelevante o raramente necesaria. Cada unidad extra de información compite con la información relevante y disminuye su visibilidad relativa.
- 9) **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores:** Los mensajes de error deberían ser expresados en lenguaje llano, indicar de forma precisa el problema y sugerir una solución de forma constructiva.

Ejemplo de guía de Nielsen

- 10) Ayuda y documentación: Aun pensando que es mejor si el sistema puede ser usado sin documentación, puede ser necesario proveer ayuda y documentación. Cualquier información de este tipo debería ser fácil de buscar, debería estar focalizada en las tareas del usuario, listar pasos concretos, y no ser demasiado larga.

ejemplos

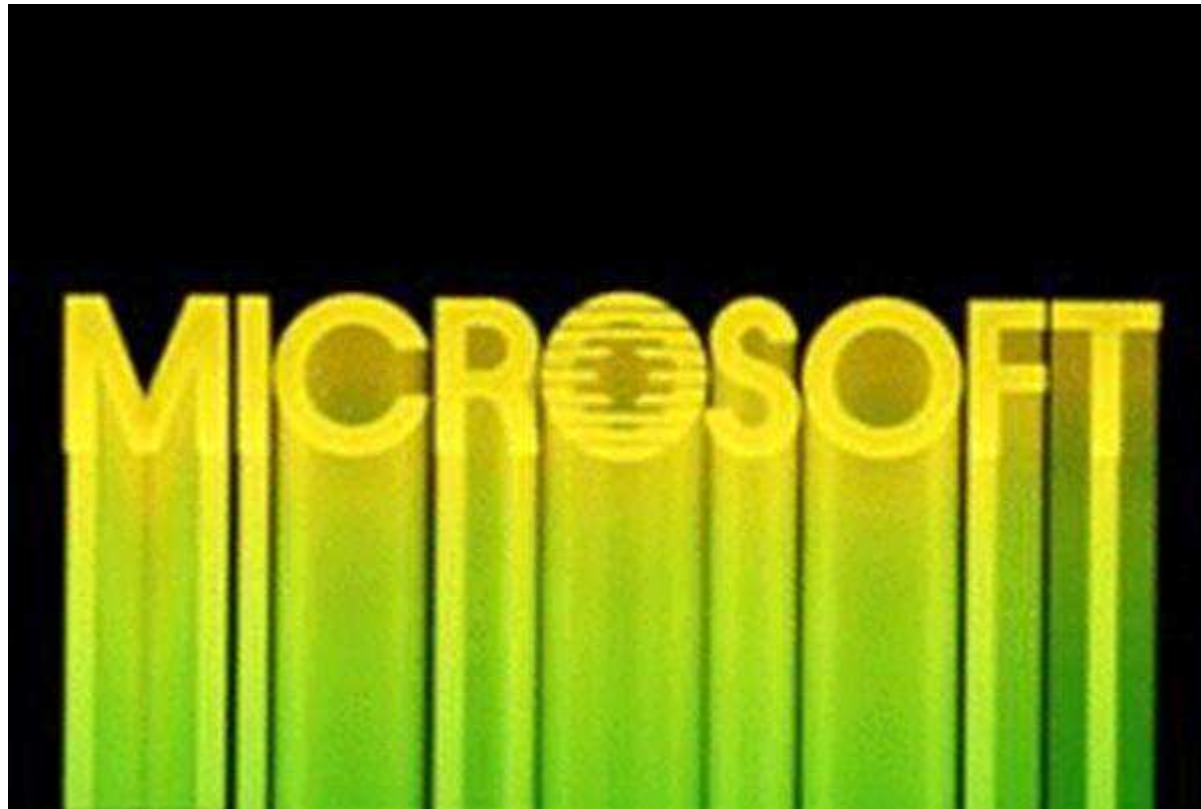
Microsoft y Apple

Extraído de:

Official Guidelines for User Interface Developers and Designers
(Microsoft)

Apple Human Interface Guidelines

Ambos documentos están disponibles en la Web y vinculadas desde el sitio del curso.



Checklist de usabilidad

- Su aplicación se instala fácilmente en un número mínimo de pasos.
- La instalación de su aplicación no requiere hacer “restart” del sistema.
- Los usuarios no tienen que leer el “readme file” antes de usar la aplicación.
- Los archivos de datos generados por el usuario son almacenados por defecto en la carpeta **My Documents**.
- Su aplicación evita nombres de archivo crípticos visibles a los usuarios.
- Su aplicación no crea carpetas por fuera de la carpeta de **Archivos de Programa**.

Checklist de usabilidad

- Su aplicación no escribe archivos en el directorio raíz del disco duro.
- Si su aplicación utiliza un cachè de disco, que dicho cachè sea contemplado al ejecutar “Disk Cleanup”.
- Su aplicación no instala íconos en el escritorio de Windows sin el permiso del usuario.
- Si su aplicación corre al arrancar el sistema, se carga sin desplegar splash screens ni ventanas de diálogo.

Checklist de usabilidad

- Su aplicación no utiliza el área de *taskbar notification* ni para indicar el estado de la aplicación, ni para lanzar aplicaciones o utilitarios. La utiliza solamente para alertar al usuario de un cambio importante.
- Su aplicación aplica apropiadamente las opciones de color que el usuario seleccionó en “Display properties” en el Panel de Control.
- Su aplicación es accesible por teclado.
- Su aplicación trabaja correctamente si el usuario incrementa el tamaño por defecto de la fuente.
- Su aplicación soporta el conjunto estándar de “keyboard shortcuts”, donde sea aplicable.

Checklist de usabilidad

- El proceso de desinstalación de su aplicación no deja archivos remanentes u otras entradas al registro que los archivos creados por el usuario.
- Su aplicación no utiliza jerga ininteligible en los textos de la interfaz de usuario. Utilice términos técnicos sólo si son claramente comprensibles por el usuario.
- Su aplicación se ajusta apropiadamente cuando el usuario cambia la resolución de pantalla, así como para configuraciones de múltiples-monitores.
- ...etcétera

Consideraciones de Microsoft sobre la usabilidad

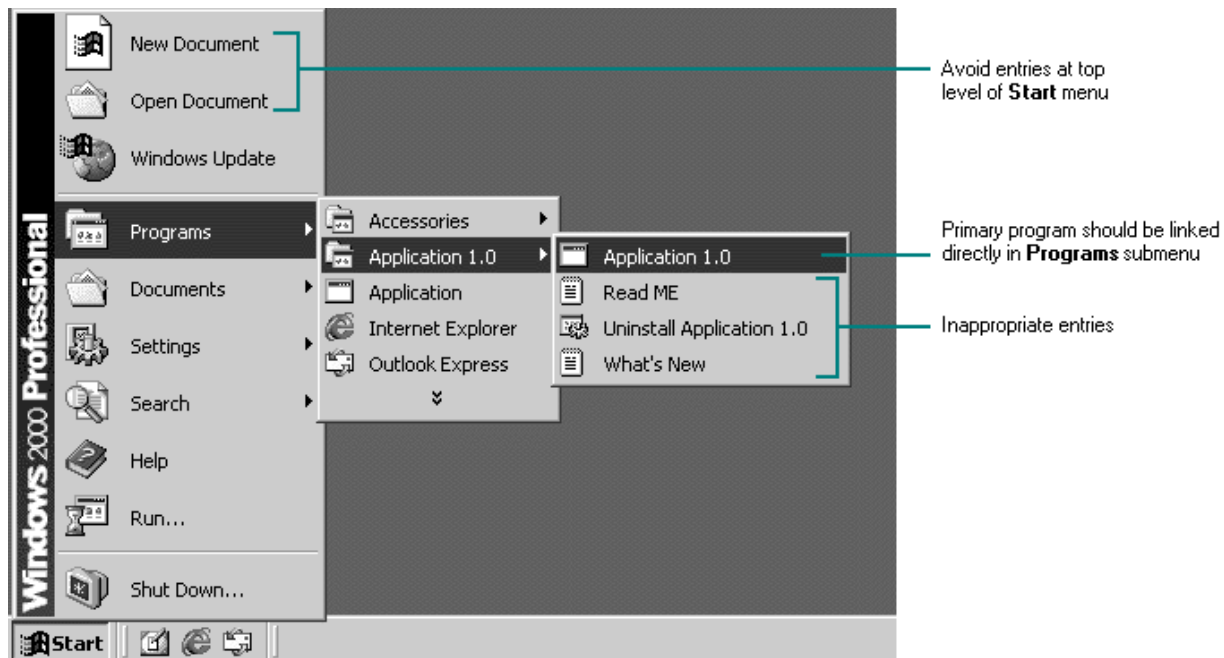
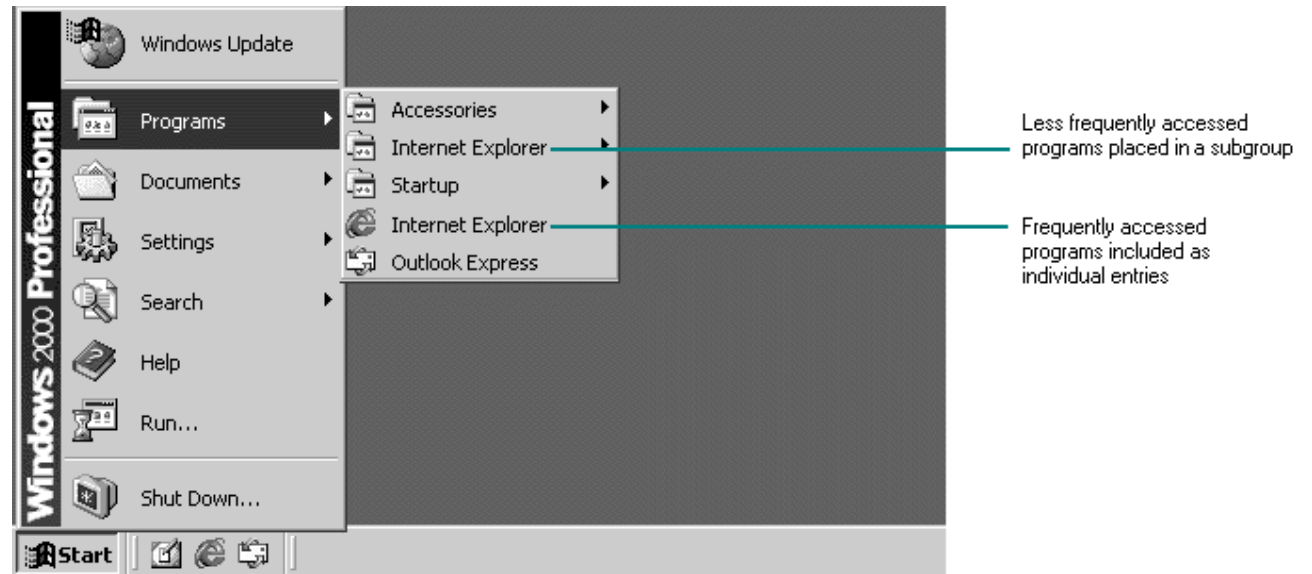
- Los aspectos de **usabilidad** en el diseño de una aplicación son esenciales para el **éxito** de esta.
- Las computadoras son aún difíciles de usar. Los usuarios solicitan que se comporten de una forma conocida, familiar, simple.
- Entregar simplicidad no consiste en remover funcionalidades, sino que implica trabajar de manera intensa. Aún una simple interfaz requiere una inversión significativa de esfuerzo y código.

Áreas clave para la usabilidad

- Diseños confusos y torpes del “*setup*” y del “*uninstall*”.
- *Setup* complejo y desordenado.
- Una primer experiencia frustrante con la aplicación.
- La opción de desinstalar incompleta o inexistente.
- Uso inadecuado del *filesystem*.
- No utilizar “My documents” como el destino inicial para alojar almacenar archivos del usuario.
- Alojar en lugares inadecuados los “program files”.
- Almacenamiento temporal disperso.
- Nombres de archivos crípticos.
 - Registrar el tipo de archivo. Registrar una descripción del tipo amigable. Registrar un ícono asociado al tipo de archivo.

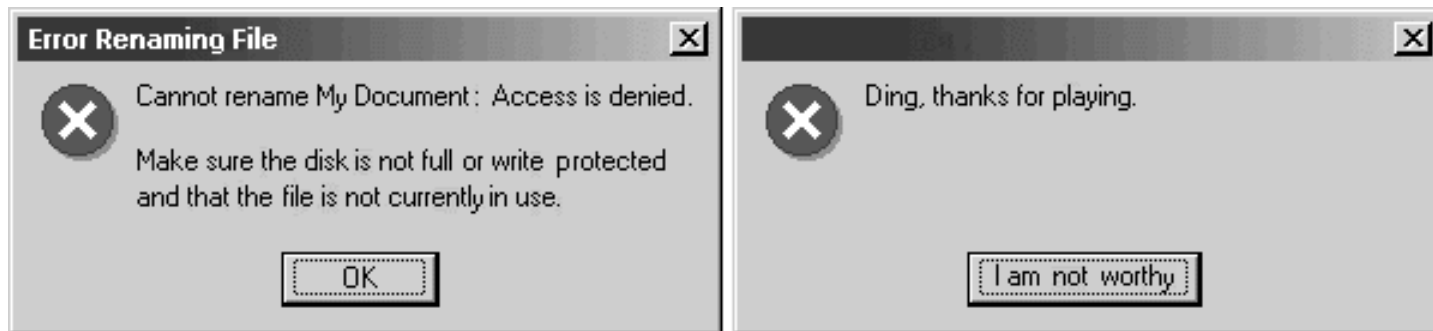
Áreas clave para la usabilidad

Abuso del “Start Menu”



Áreas clave para la usabilidad

- Evitar la creación de íconos en el escritorio.
- Evitar el abuso en el área del notificación del estatus.
- Pobreza en el diseño de los mensajes. Evitarlos en lo posible.



What is displayed on screen

What the user sees

- Deshabilitar comandos u opciones no permitidas.
- Proveer formas de feedback menos intrusivas, como los “balloon-style ToolTips”.

Áreas clave para la usabilidad

- Pasar por alto el “system locale” (idioma seleccionado para el Sistema Operativo).
- Falta de adecuado soporte de teclado.
- Diseños complejos de las segundas ventas (ventanas de diálogo).
- Falta de soporte en las preferencias del Font y Color de usuario.
- Uso erróneo del Color. (No adaptarse a los colores que seteo el usuario para el display).

Principios de diseño y de metodología

- **El control es del usuario.**
 - El usuario debe sentirse en control del software en lugar de sentirse controlado por el software.
- **Reflejar los modelos mentales del usuario.**
 - Experiencias tanto de la vida real como con computadoras.
 - Familiar, Simple, disponible, descubrible.
- **Manipulación Directa**
 - Diseñar el software de manera de que el usuario pueda manipular directamente la representación de información que realiza el software (existen acciones implícitas).
- **Consistencia**
 - Permite transferir conocimiento ya existente a nuevas tareas. El usuario no pierde tiempo recordando las diferencias en la interacción. La interfaz se hace familiar y predecible.
 - Consistente con el sistema de ventanas, consigo mismo, con versiones anteriores, con las expectativas de la gente.

Principios de diseño y de metodología

- **Dejar espacio para el error (forgiveness)**

- Los usuarios prefieren explorar y aprender por prueba y error. Una interfaz efectiva facilita esto. Sólo se proveen las opciones autorizadas, alerta sobre situaciones peligrosas y, mejor aún, hace acciones reversibles o recuperables.

- ***Feedback***

- Siempre brindar respuesta a las acciones del usuario. Ayuda a confirmar que el SW responde a la entrada y comunica detalles que distinguen la naturaleza de la acción.

- **Simplicidad**

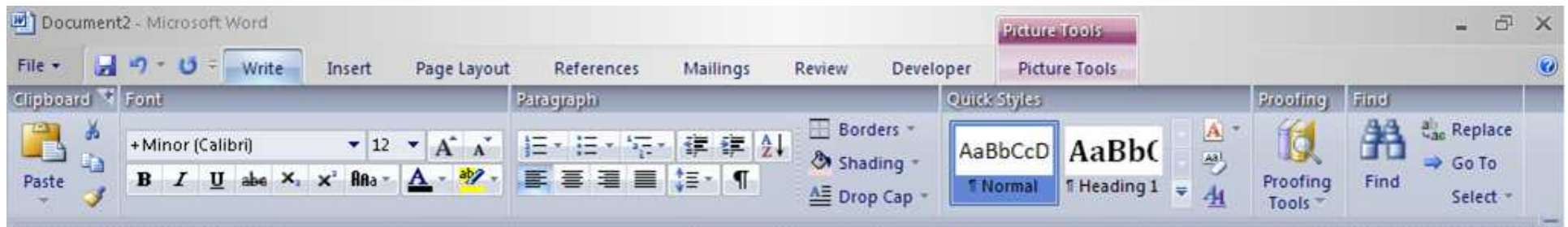
- **Simple pero no simplista**, fácil de aprender y de usar. Debe proveer acceso a todas las funcionalidades de una aplicación. Maximizar funcionalidad y mantener la simplicidad son opuestos a balancear.

Principios de diseño y de metodología

- **Integridad estética**

- Que siga principios de buen diseño visual. Debe ser placentero mirarlo aunque se lo mire por mucho tiempo.
- Mantener **simples** los temas gráficos.
- No utilizar símbolos arbitrarios para representar conceptos.
- Aspectos que incrementan la calidad:
 - Íconos de la mejor calidad posible.
 - Textos con *antialiasing*.
 - Tamaño y tipo de texto consistente en toda la aplicación.
 - El tamaño de los controles debe ser consistente.

Ejemplo: Word12



Ejemplo: Word12


Clinical Trial Agreement.docx - Microsoft Word

File Write Insert Layout References Mailings Review & Comment

Clipboard Font Paragraph Quick Text Formatting Proofing Find

Clipboard +Minor (Corbel) 11 A A Aa B I U A ab

Properties: Clinical Trial * = Required field

 Study Title: Clinical Trial - Sample Test Study ID Number: 1304W-P Sponsor's Protocol Number: 10-421 Look Up Sample Size: 200

Reviewer: Lauren Antonoff Date of Review: 6/9/2005 Related Studies: <http://contoso/clinical trials/1304W-P/State List.xls>
<http://contoso/clinical trials/1304W-P/Study Overview.ppt>

6/9/2005 Clinical Trials Agreement (#1304W-P)

CLINICAL TRIALS AGREEMENT

INTRODUCTION

Contoso recognizes the importance of the pharmaceutical industry in the clinical development of new anti-cancer agents. Contoso wishes to foster collaboration with industry whenever possible. As part of its mission to improve cancer care, Contoso shares with industry the important goal of defining the contribution of a new drug or biologic in the treatment of cancer. Contoso therefore recognizes and supports the need of a private sponsor to focus at the appropriate time on clinical trials which lead to a New Drug Application (NDA) or a Biologics License Application (BLA) since an NDA and a BLA, once approved, are the vehicles through which new anti-cancer therapies become widely available to cancer patients. Thus Contoso considers it appropriate for the investigators sponsored by Contoso to do clinical trials of interest to, and partially supported by a pharmaceutical firm, provided that the trials have scientific merit and are consistent with the overall goals of the investigators and Contoso.

Inasmuch as Contoso coordinates a large volume of clinical research with new anti-cancer agents, industry recognizes Conto's need to be aware of industry's plans for the clinical development of new

Page: 1 of 14 Words: 5,920 View SOAP Task

Principios de diseño y de metodología

- **En lo posible no utilizar interfaces modales.**
 - Permitir que el usuario haga lo que quiera todo el tiempo. Evitar generar modos que bloqueen la interfaz.
 - Sólo hacerlo en caso de:
 - tener necesidad de apretar el botón del mouse por mucho tiempo;
 - tener necesidad de apretar el botón de shift por mucho tiempo.
 - *Wizards* que guíen al usuario a lo largo de una tarea.

Conceptos básicos

- **Diseño centrado en los datos.**
 - El usuario puede navegar por los datos y editarlos directamente, en lugar de primero localizar el editor apropiado.
 - El usuario se concentra en la información en lugar de concentrarse en las aplicaciones.
- **El Objeto como metáfora.**
 - Los objetos de la naturaleza y sus propiedades se utilizan para facilitar la comprensión y para que el usuario se olvide de la tecnología subyacente. Los objetos poseen:
 - **Propiedades** – definen apariencia o estado: color, tamaño.
 - **Operaciones** – indican lo que puede ser realizado: mover, copiar, borrar.
 - **Relaciones** – indican el contexto en el cual existen. Colecciones, restricciones y compuestos.

A is for Apple.

It's the first thing you should know about personal computers.

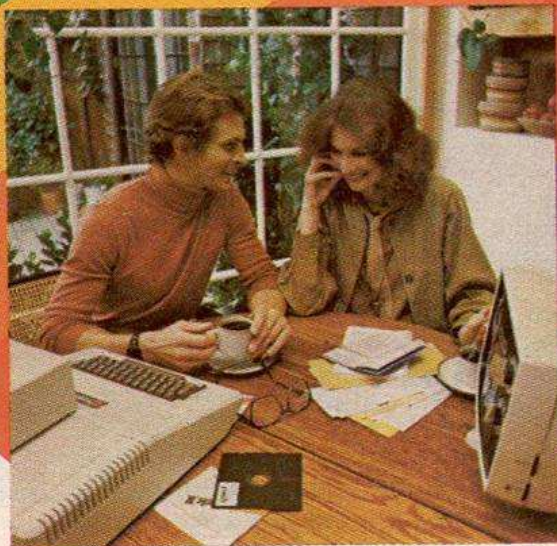
Discover the special thrill of personal computing — with Apple! It's a home computer, a business computer, a classroom computer, *your* computer. No wonder tens of thousands have already

chosen Apple. Join the excitement in your local computer store. Call for the one nearest you. **800-538-9696.**

In California call **800-662-9238.**



apple® computer
10260 Bandley Drive
Cupertino, CA 95014



Principios de diseño

- **Involucrar al usuario en el proceso de diseño.**
 - Conocer su audiencia.
 - Conocer sus usuarios.
 - Conocer las aplicaciones que compiten con la suya.
 - Crear escenarios típicos en los que su aplicación es usada.
 - Testar su aplicación con personas que caigan dentro de su objetivo.
 - A pesar de que el desarrollador conoce mejor la aplicación que los clientes, tratar que este último defina los “settings” y temas de diseño.
 - Analizar las tareas de los usuarios.
 - Conocer los modelos conceptuales que estos manejan en sus tareas.
 - Construir prototipos.
 - Testar las ideas de diseño fundamentales con los usuarios.
 - Observar a los usuarios.
 - Ver, escuchar con cuidado las reacciones frente a los prototipos.
 - Testar áreas claves del producto.
 - Cambiar el diseño en base a las sugerencias. Iterar.

Principios de diseño

- **Al tomar decisiones de diseño:**
 - Sopesar costos y beneficios de cada funcionalidad de la interfaz.
 - Cada vez que se agrega una funcionalidad ocurre que:
 - La aplicación crece.
 - La aplicación se hace más lenta.
 - La interfaz se hace más compleja.
 - En lugar de usar recursos en mejorar lo existente se agregan cosas nuevas.
 - La documentación y la ayuda se hacen más extensas.
 - Se corre el riesgo de empeorar las características ya existentes.
 - Se aumenta el tiempo requerido para validar el comportamiento de la aplicación.
- **Evitar el agregado de funcionalidades en cascada.**
 - Los rasgos deben formar una unidad para la mayor usabilidad.
- **Dirigirse al 80% de la audiencia potencial.**

Características de un gran software

- **Alto rendimiento**

- Rendimiento es la medida percibida de cuán rápido o eficiente es su software. Es crítico para el éxito de cualquier software. Aún cuando utilice el mejor algoritmo, puede parecer lento si gasta mas tiempo procesando datos que respondiendo al usuario.
- Utilice métricas para identificar problemas. No se base en su criterio.
- Evite esperar hasta el final para ajustar el rendimiento.
- Utilice hilos (threads) para aumentar la velocidad de respuesta.
- Optimizar las operaciones de I/O en disco o CD.
- Optimizar el uso de memoria.
- Evite cargar código que aún no va a utilizar.

Características de un gran software

- **Facilidad de uso**

- Utilice metáforas que representen ideas concretas y familiares. Haga obvias sus metáforas para que las expectativas del usuario se correspondan con la implementación.
- Enfocarse en soluciones, no en “features”. No agregar cosas solo porque la competencia las tiene.
- Asegurarse que el “packaging” indica claramente los requerimientos del sistema y contiene todo lo que el usuario necesita para comenzar.
- Establecer un *seteo* por defecto inteligente.
- No abrumar con demasiada información. Acceso progresivo a la misma, mostrándola a medida que se precisa y dejando que el usuario seleccione si quiere ver información inútil para él.
- Contemple que múltiples usuarios pueden estar utilizando la aplicación al mismo tiempo con los mismos datos.
- Proveer mensajes de error útiles. Qué sucedió, por qué y cuáles son las acciones posibles.

Características de un gran software

- **Confiabilidad**

- Un programa confiable presenta la información de la forma esperada o deseada, hace lo posible por mantener la integridad de los datos, puede manejar situaciones complejas sin colgarse.
- La interfaz se comporta de manera predecible (consistencia).
- Disminuye o elimina la pérdida de información al importar datos.
- Testar el software en variedad de condiciones inusuales.
- Asegurarse que el “packaging” indica claramente los requerimientos del sistema y contiene todo lo que el usuario necesita para comenzar.
- Anticiparse a los errores y manejarlos de manera elegante.

Características de un gran software

- **Adaptabilidad**

- Adaptarse al entorno. Si, por ej., se cae la red, que siga funcionando.
- Que tenga inteligencia suficiente para seguir un camino seguro para los datos (si se cae la red, que avise y salve localmente los datos).
- Minimizar los aspectos a asumir acerca del hardware.
- Contemplar los cambios en la resolución y cantidad de colores de la pantalla.

- **Facilidad de uso**

- Permita que los usuarios exploren las funcionalidades de su aplicación sin causar daño irreversible a sus datos. Undo, Redo.
- Internacionalizar su software.
- Hacer que su aplicación sea accesible para personas con discapacidades.
- Proveer documentación apropiada para su software.

Características de un gran software

- **Interoperabilidad**

- Importar y exportar archivos con otras aplicaciones.
- Leer archivos de otros formatos.
- Que los datos creados sean legibles por otras aplicaciones.
- Que los archivos sean legibles en distintas plataformas.
- Utilizar protocolos estándar para intercambio de datos (XML)
- Utilizar extensiones en los nombres de los archivos para permitir el intercambio con otras plataformas.

- **Movilidad (para uso de la aplicación en laptops)**

- Evitar preguntar por eventos. Si no hay eventos, que esté *idle* (p.54)
- Permitir que el usuario ingrese toda la aplicación en el disco duro, para no requerir el CD.
- Minimizar accesos al disco duro.
- Que sea adaptable a cambios frecuentes en la red.
- Ser adaptable al cambio de monitor, de resolución, etc.

Comunicación Visual Efectiva

Basado en:

**Principles of effective visual communication
for graphical user interface design**

Aaron Marcus

Del libro:

Human-Computer Interaction

Escrito y editado por Baecker, Grudin, Buxton, Greenberg

Introducción

La forma ingenieril de acceder al diseño...

- ... guías
- ... métodos
- ... métricas
- ... principios de diseño
- ... etc.

diseño sistemático.

Introducción

Algunas personas plantean que los problemas de diseño de interfaz de usuario están resueltos, ya que han emergido estándares en la industria para manejadores de ventanas.

Introducción

Otros sostienen que los paradigmas existentes (Windows, Macintosh, etc.) no resuelven la tarea esencial de **comunicar visualmente** los datos y funciones de un producto.

Introducción

La industria y la academia, deberían producir, entonces, principios de diseño sistemático que provean guías para **soluciones simples, claras y consistentes** en el diseño de menús, ventanas, íconos, cajas de diálogo y paneles de control que no son explícitamente especificados por los actuales paradigmas en el área de las interfaces gráficas.

Si bien se mantiene el lenguaje, se genera una gramática más refinada.

GUIs y diseño gráfico

Una GUI puede aportar:

- Una imagen mental comprensible (¿metáfora?)
- Organización apropiada de datos, funciones, tareas y roles (¿modelo cognitivo?)
- Un esquema de navegación eficiente entre estos datos y las funciones, tareas y roles.
- Calidad en la apariencia (el *look*)
- Una efectiva secuencia de interacción. (el *feel*)

GUIs y diseño gráfico

El diseño gráfico puede ayudar a las GUIs a alcanzar su potencial de comunicación.

El diseño gráfico sistemático orientado a la información,

es el uso de la tipografía, símbolos, color y gráficos (estáticos y dinámicos) para **expresar** hechos, conceptos, emociones etc., **de mejor forma posible.**

¿Cuál es la mejor forma posible?

El lenguaje visible

comunicando mensajes y contenidos

- El **lenguaje visible** consiste en todas las técnicas gráficas utilizadas para comunicar:
 - **Esquemas** (disposición): formatos, proporciones, grillas, organización 2D y 3D.
 - **Tipografía.**
 - **Color y textura.**
 - **Imágenes.** signos, íconos y símbolos, desde lo abstracto a lo fotográfico.
 - **Animación.**
 - **Secuenciación:** una aproximación general visual a la narrativa (visual storytelling)
 - **Sonido:** abstracto, vocal, concreto, musical (*earcons* (!!))
 - **Identidad visual.** Consistencia.

Principios en el diseño de interfaces de usuario

El diseño sistémico se basa en principios fundamentales (¿indiscutibles?), probados.

- 1 **Organizar:** proveerle al usuario una clara y consistente estructura conceptual
- 2 **Economizar:** maximizar la efectividad de un conjunto mínimo de indicadores (cues).
- 3 **Comunicar:** hacer corresponder la presentación a las capacidades del usuario.

Principio 1: organizar

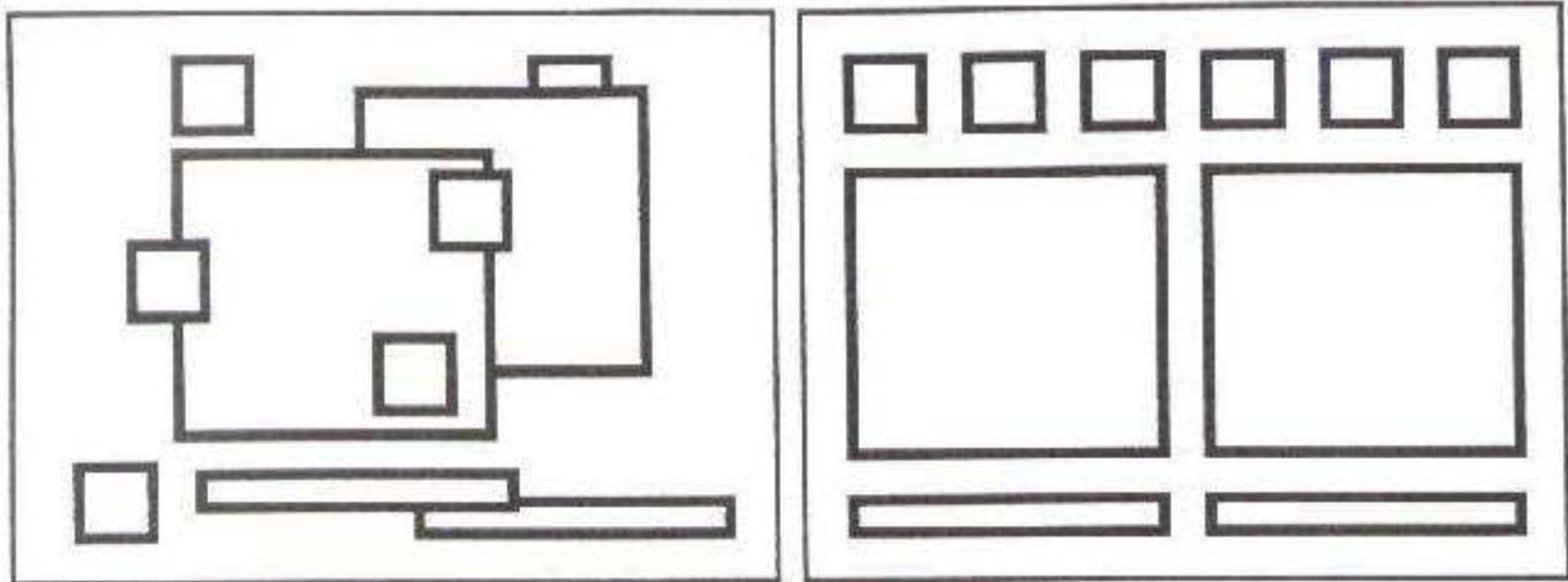


Figure 1: Chaotic Screen, Ordered Screen

Principio 1: Organizar

*Diseñar es planificar y organizar,
ordenar, relacionar y controlar.*

*En breve, abarca todos los medios
para oponerse al desorden y a los accidentes*

*Por tanto, es una necesidad humana que
califica el pensamiento y la acción de las personas*

Josef Albers - artista de la Bauhaus

- *La facilitación de esta emergencia de genios era una de las intenciones de la Bauhaus cuando se diseñó el famoso curso de fundación básica. El curso fue luego adoptado por infinidad de instituciones que se consideraron progresistas y modernas.*

Y no era que los ejercicios fueran malos, era la ideología la que fallaba. La moraleja de todo esto es que los 200.000 dólares en los estudios en Estados Unidos se invierten en el derecho de ser filtrado para dejar lugar a los genios.

- *luis camnitzer*

Consistencia

- **Consistencia interna:** Observar las mismas reglas y convenciones para todos los elementos de una GUI.

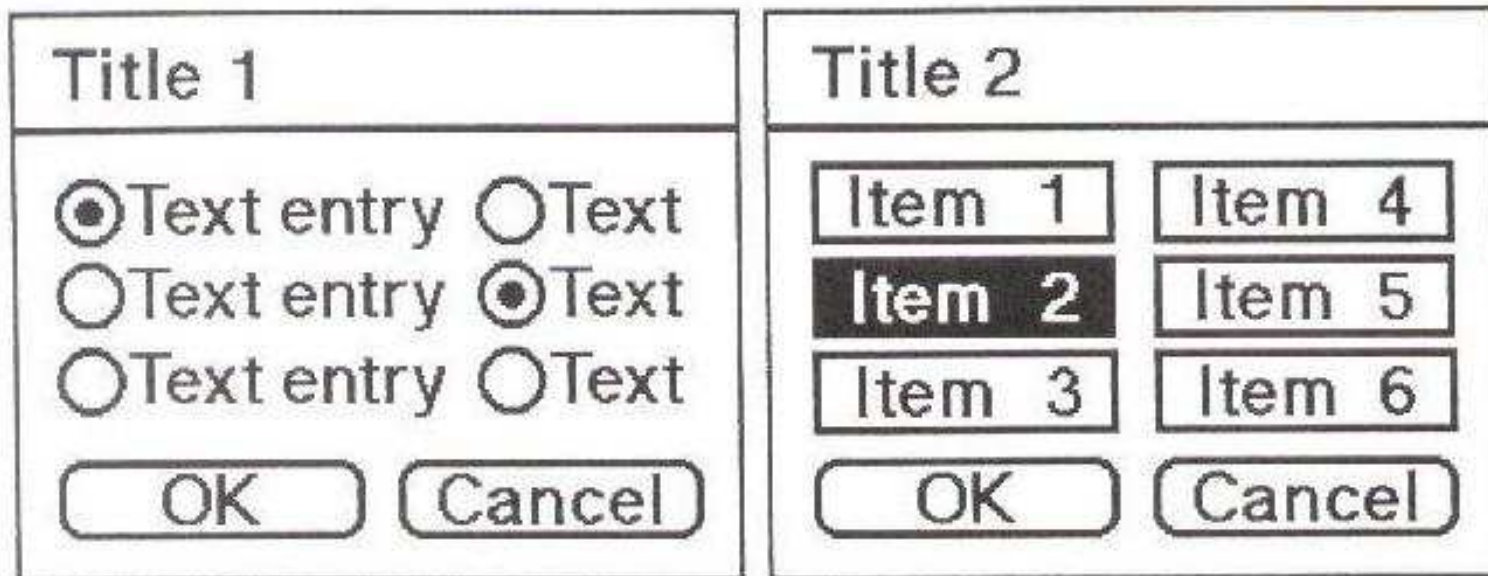


Figure 2: Internal Consistency: Dialogue Boxes

Consistencia

- **Consistencia externa:** Hay que seguir a las convenciones generadas por otras plataformas y por las convenciones culturales.

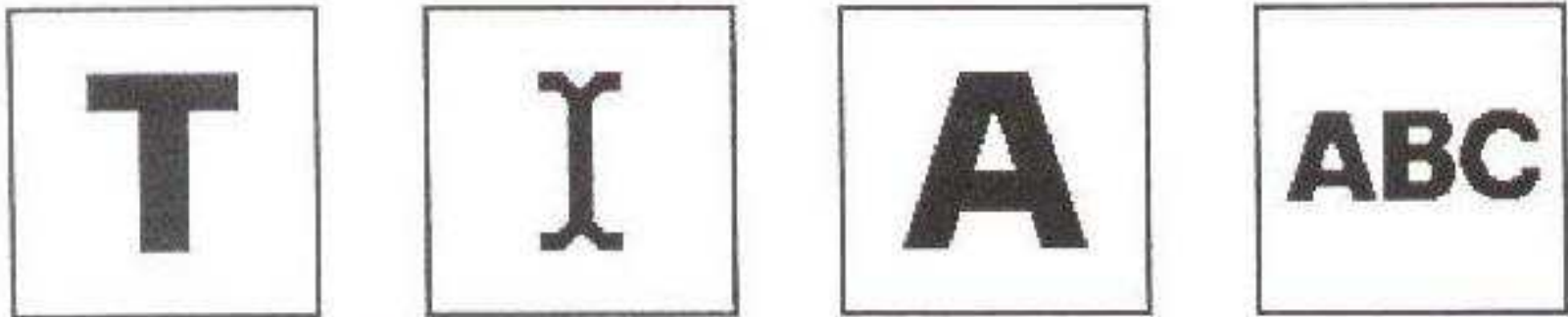


Figure 3: External Consistency for Text Tool Icons

Consistencia

- **Consistencia con el mundo real:** Se tiene que ser consistente con la experiencia del mundo real.

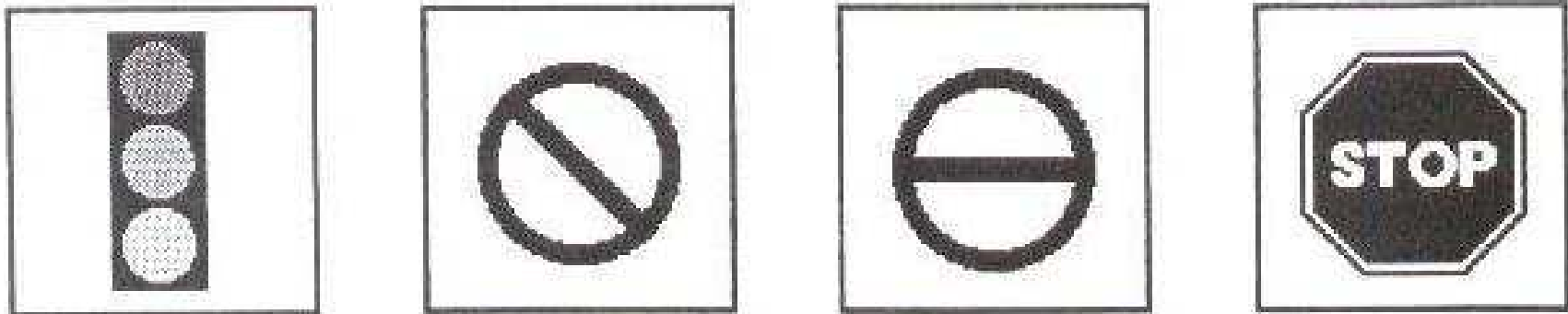
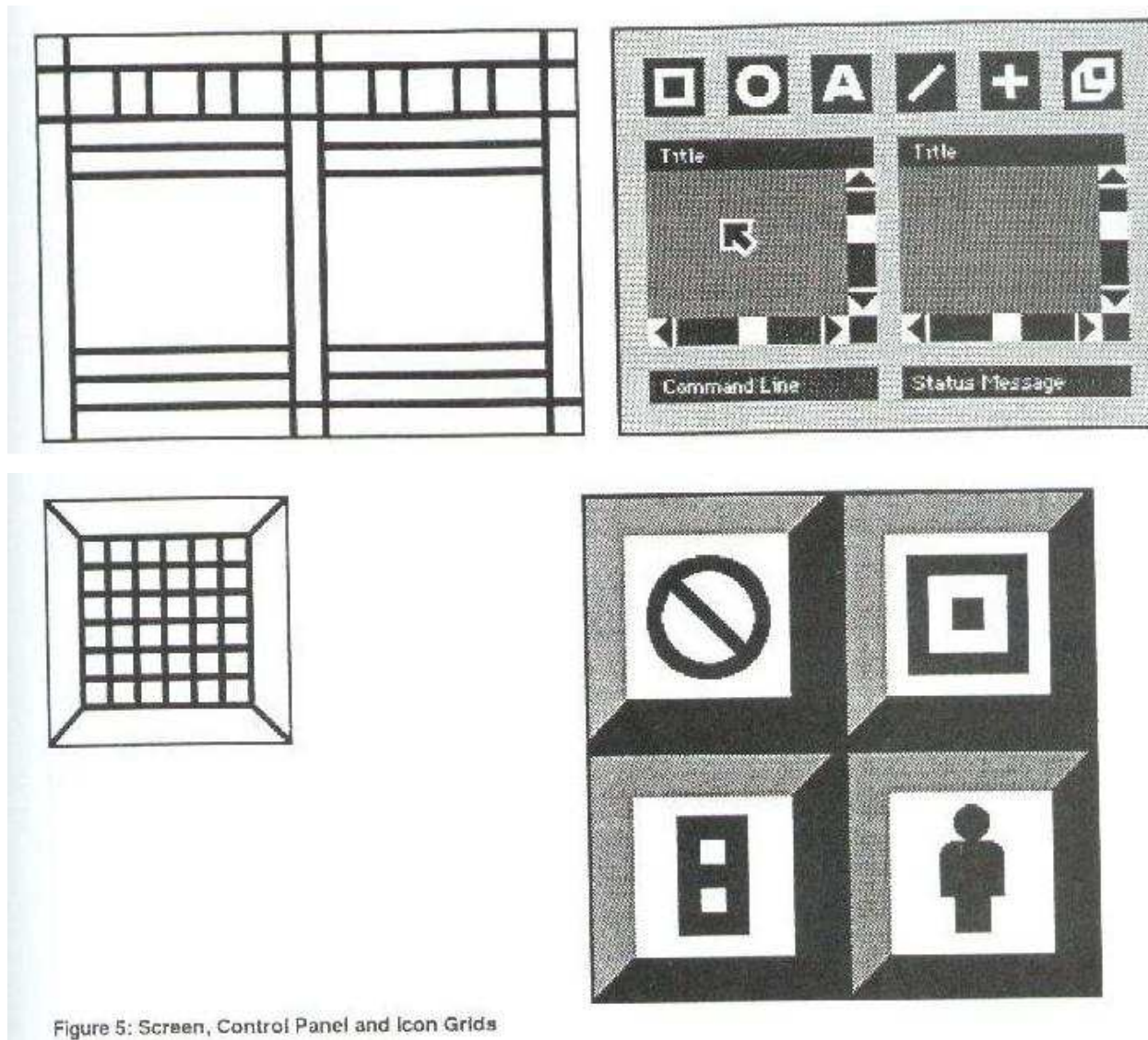


Figure 4: Real World Consistency: Highway Signage for Screen Display Icons

Consistencia

- ¿Cuándo ser inconsistente?
- **Innovación.** Hay que tener una buena razón (!): cuando hay una medida clara del beneficio de la innovación.
- “Luchar por la consistencia sin ser esclavo de ella”
 - muchas veces hay que ser esclavo y punto :)

Disposición (screen layout)



Disposición (screen layout)

- Existen tres formas principales de organizar: usar una estructura de grilla, estandarizar la disposición, y agrupar elementos relacionados.
- Estructura de grilla: puede ayudar a ubicar ítems en menús, componentes principales de la ventana, paneles de control y diálogos.
- Medidas cuidadosas deben hacerse para considerar los contenidos a desplegar, los espacios entre los grupos de textos, la resolución de pantalla, etc.
- El número de grandes divisiones debe cumplir la regla de Miller (7 ± 2).
- Utilizar grilla incluso dentro de íconos para ayudar a localizar las piezas individuales de los pictogramas o ideogramas.

Relaciones

- Establecer una clara relación (Gestalt!) que permita vincular elementos relacionados y desvincular elementos no relacionados.

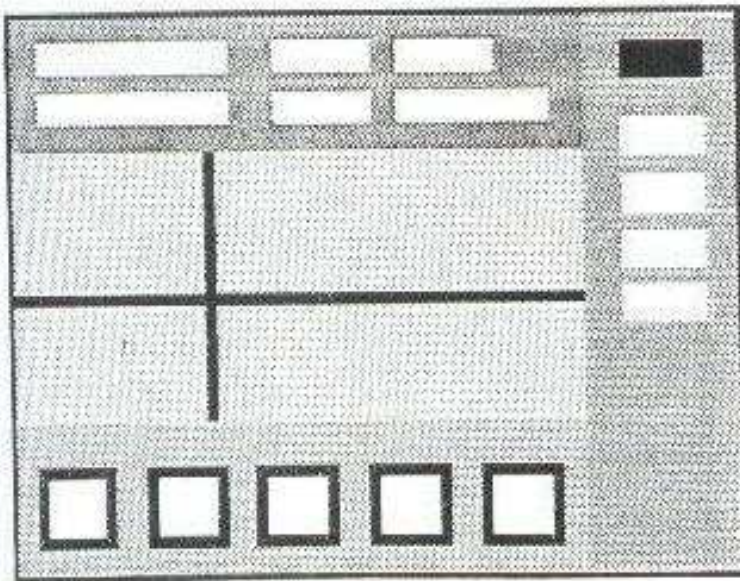
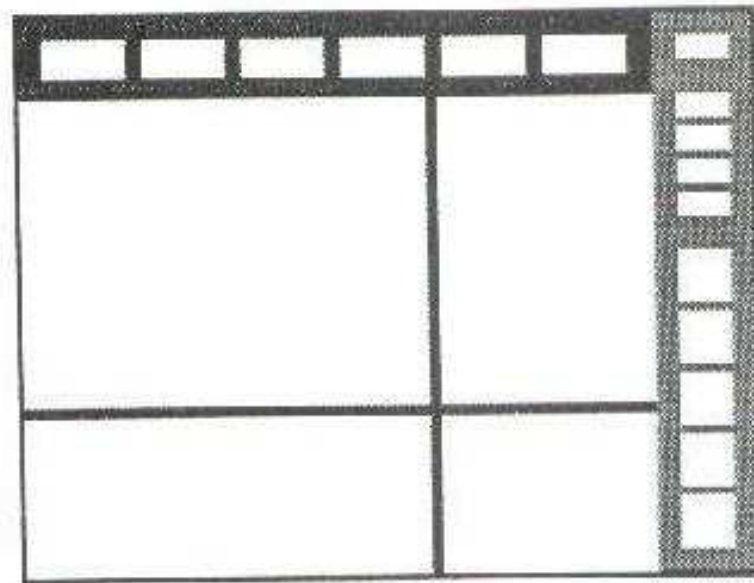


Figure 6: Relationships



Navegabilidad

- Proveer un foco inicial a la atención del usuario, dirigir la atención a los ítems importantes, secundarios o periféricos. Asistir a la navegación a través del material.

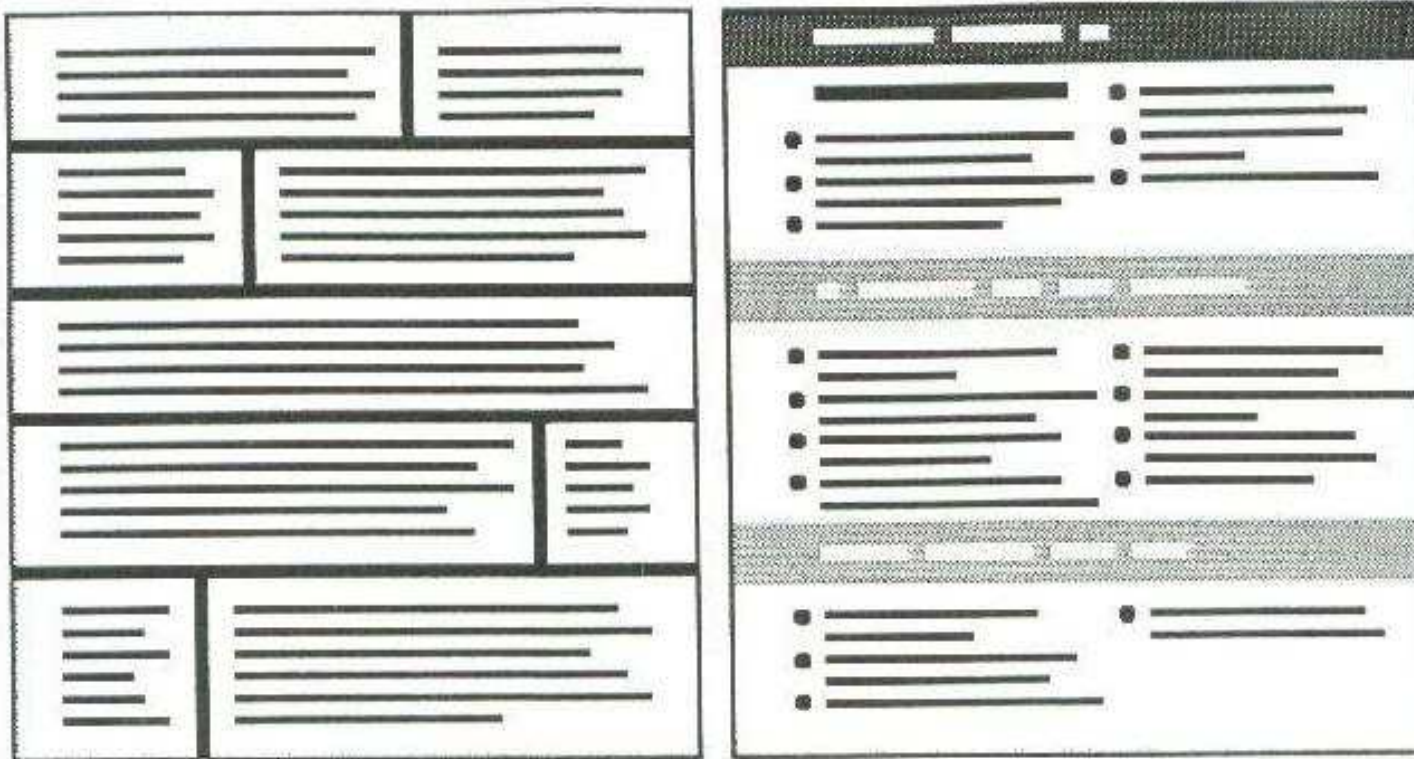


Figure 7: Navigation

Principio 2: Economizar

- *¿Cuántos controles precisa un aparato?*

Cuantos menos controles haya, luce más fácil de usar, y es más fácil de encontrar los controles relevantes...

Para que algo luzca fácil, minimiza el número de controles.

Don Norman

Economizar se compone de cuatro temas: **simplicidad, claridad, grado de distinción y énfasis.**

Principio 2: Economizar

- Less is more.
 - *motto* adoptado por Ludwig Mies van der Rohe

Simplicidad

- Incluir solamente aquellos elementos que son esenciales para la comunicación. Deberíamos ser lo menos molestos posible.
- Mantener la interfaz de la forma más simple.
- KISS. Pero no asumir que la gente es estúpida: todos tenemos capacidades limitadas para absorber información rica en contenido.
- Simplicidad es una virtud, ¿a menos que se quiera confundir, entretener o embellecer?
- Los componentes de la GUI deben ser *modestos* y no **atraer** demasiado la atención del usuario.
- el usuario debería ser **inconsciente** del **trabajo** de la GUI para **transmitir** significado.

Simplicidad

TITLE 1	
<input type="radio"/> Item 1	<input type="radio"/> Item 3
<input checked="" type="radio"/> Item 2	<input type="radio"/> Item 4
<input type="radio"/> Item 5	
<input type="radio"/> Item 6	<input type="button" value="OK"/>
<input type="button" value="CANCEL"/>	

Title 1	
<input type="button" value="Item 1"/>	<input type="button" value="Item 4"/>
<input checked="" type="button" value="Item 2"/>	<input type="button" value="Item 5"/>
<input type="button" value="Item 3"/>	<input type="button" value="Item 6"/>
<input type="button" value="OK"/>	<input type="button" value="Cancel"/>

Figure 8: Complicated and Simpler Designs

Claridad

- Diseñar todos los componentes para que su significado no sea ambiguo.

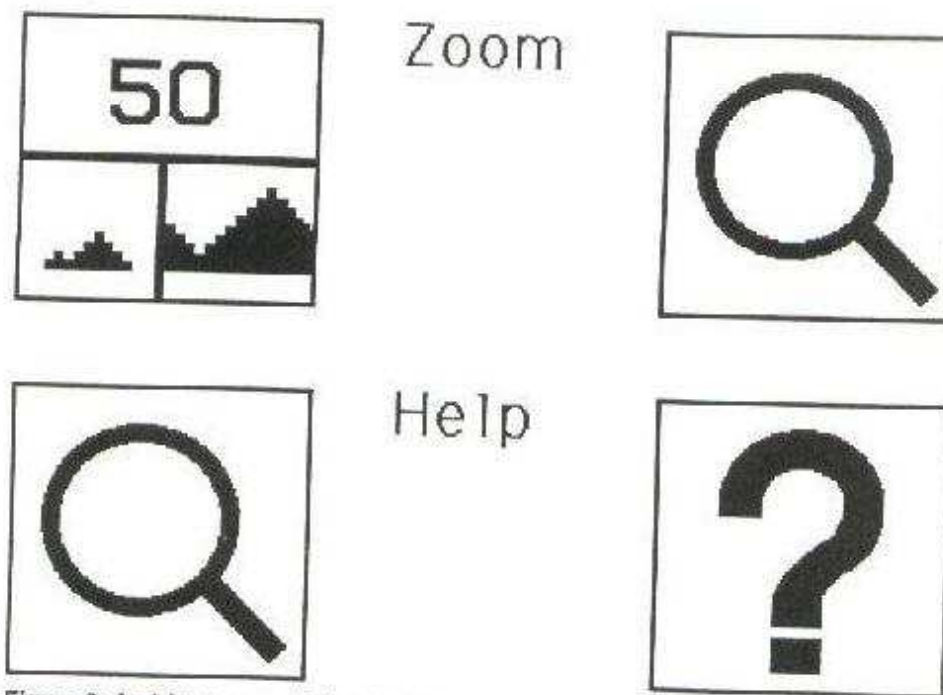
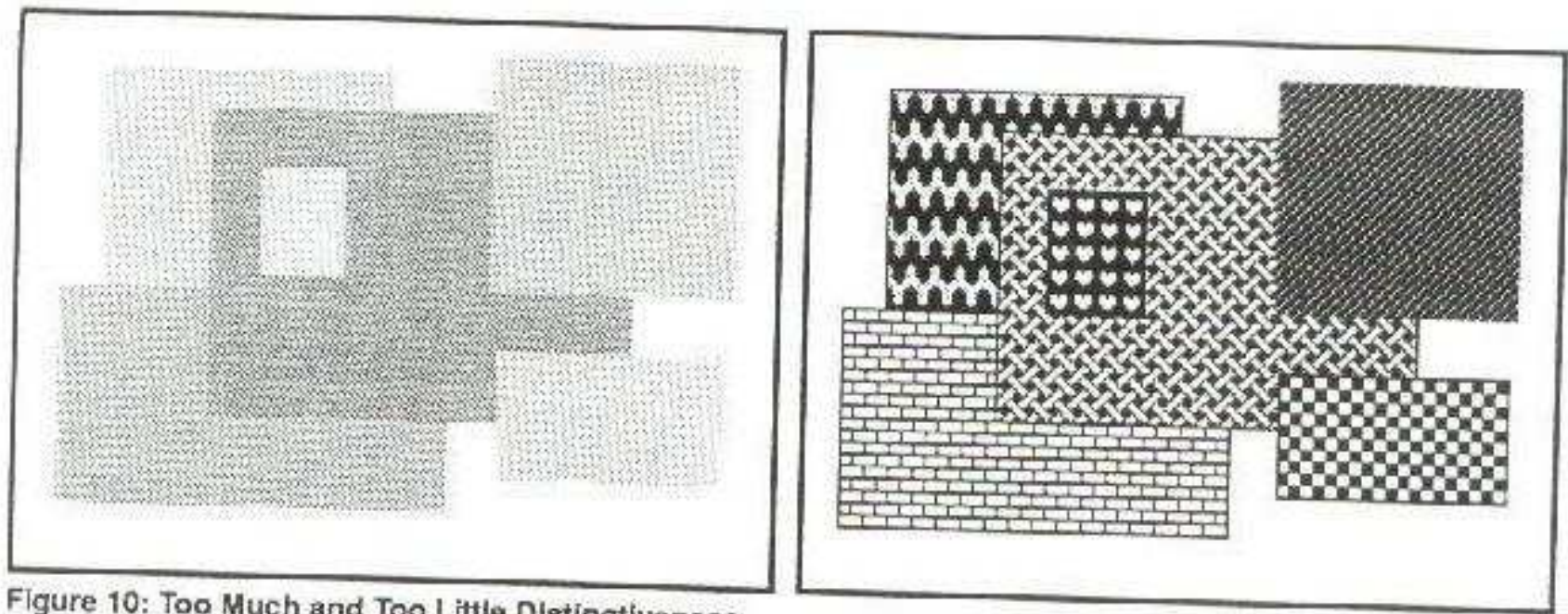


Figure 9: Ambiguous and Clear Icons

Grado de distinción

- Distinguir propiedades importantes de los elementos esenciales.



Énfasis

- Hacer que los elementos importantes sean fácilmente percibidos. No enfatizar elementos no críticos y minimizar la confusión para que la información crítica no esté oculta.
- **Distinguir propiedades importantes de los elementos esenciales.**

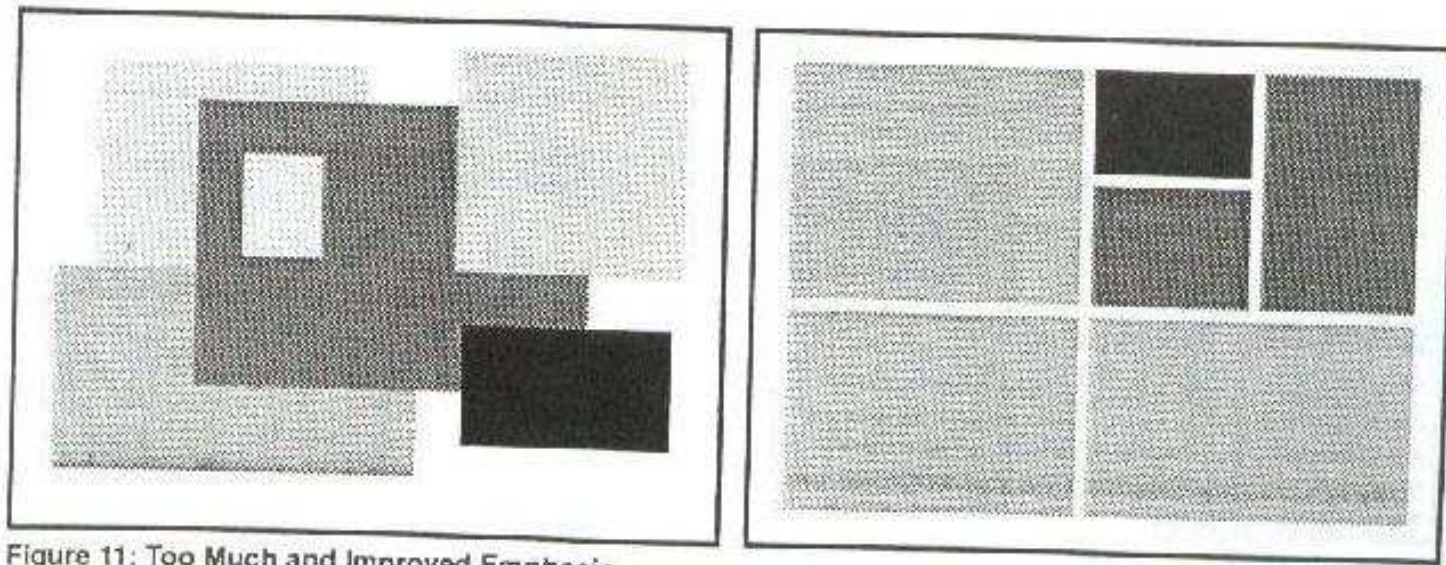


Figure 11: Too Much and Improved Emphasis

Principio 3: Comunicar

La comunicación es un proceso social, dentro de un contexto específico, en el cual signos son producidos y transmitidos, percibidos y tratados como mensajes en los cuales el significado puede ser inferido.

Sol Worth

Factores a considerar: legibilidad, comprensibilidad, tipografía, simbolismo, vistas múltiples y color/textura.

Legibilidad

Diseñar caracteres individuales, símbolos y elementos gráficos que sean fácilmente advertidos y distinguibles.

Illegible

Text set in
Old English

Legible

Text set in
Univers

Illegible

Large
Medium
Small

Legible

Large
Medium
Small

Figure 12: Illegible and Legible Texts Based on Fonts and Relative Sizes

Legibilidad

Diseñar caracteres individuales, símbolos y elementos gráficos que sean fácilmente advertidos y distinguibles.

Fonts, íconos, símbolos de los paneles de control.

Las técnicas de visualización deben ser apropiadas a la tecnología de pantalla (resolución, color, animación).

La pantalla negra de fondo en habitaciones iluminadas causa reflejos que distraen y que disminuyen la legibilidad.

Pantallas claras en habitaciones oscuras pueden deslumbrar.

Comprensibilidad

Que lo desplegado sea comprensible, fácil de identificar e interpretar.

Unreadable: Design components to be easy to interpret and understand. Design components to be inviting and attractive

Readable

Design components to be easy to interpret and understand.

Design components to be inviting and attractive

Figure 13: Unreadable and Readable Texts

Comprensibilidad

Que lo desplegado sea comprensible, fácil de identificar e interpretar.

Usar un número limitado de fuentes (tipografías) para distinguir entre diferentes clases de información.

Dentro de cada fuente, elegir un conjunto de estilos, marcas de puntuación y símbolos.

Univers
Helvetica
Times Roman
Palatino
Courier

ABCDEFGHIJKLM
ABCDEFGHIJKLM
AAaBBbCCcDDd

Uchbφ=>IQmΞ

Figure 14: Typefaces and Typestyles

Typesetting (ajuste de tipos)

- No mas de tres fuente en un máximo de tres tamaños, no importa qué aplicación.
- Líneas de texto deberían tener entre 40 y 60 caracteres máximo, y las palabras deberían estar espaciadas correctamente.
- El texto debe estar en el formato apropiado: texto hacia la izquierda, números a la derecha, evitar listas con texto centrado, evitar líneas de texto cortas justificadas.
- Líneas justificadas con “ancho fijo”, y textos en mayuscula enlentecen la lectura.

Simbolismo

- No solo preocuparse por los caracteres tipográficos.
- Íconos, símbolos, mapas, diagramas, fotografías usadas en el despliegue también deben ser cuidadosamente seleccionadas y refinadas para comunicar los contenidos deseados.

Vistas múltiples

Brindar múltiples perspectivas en el despliegue de estructuras complejas y procesos. Un importante principio de buen diseño es hacer uso de estas múltiples vistas:

- Múltiples formas de representación.
- Múltiples niveles de abstracción.
- Vistas alternativas simultáneas.
- Vínculos y relaciones cruzadas.
- Metadatos, metatextos, metagráficos.

Vistas múltiples

Nation	State	City
USA		Dallas
	Texas	

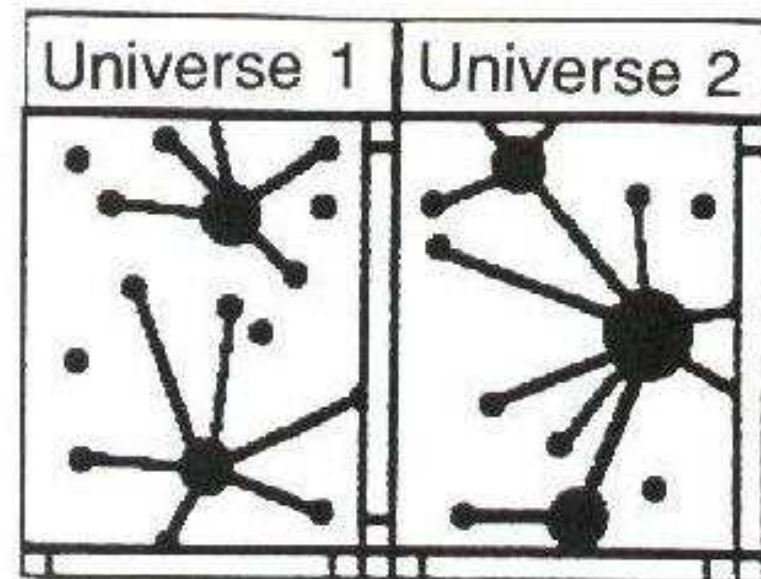


Figure 15: Verbal and Visual Multiple Views