

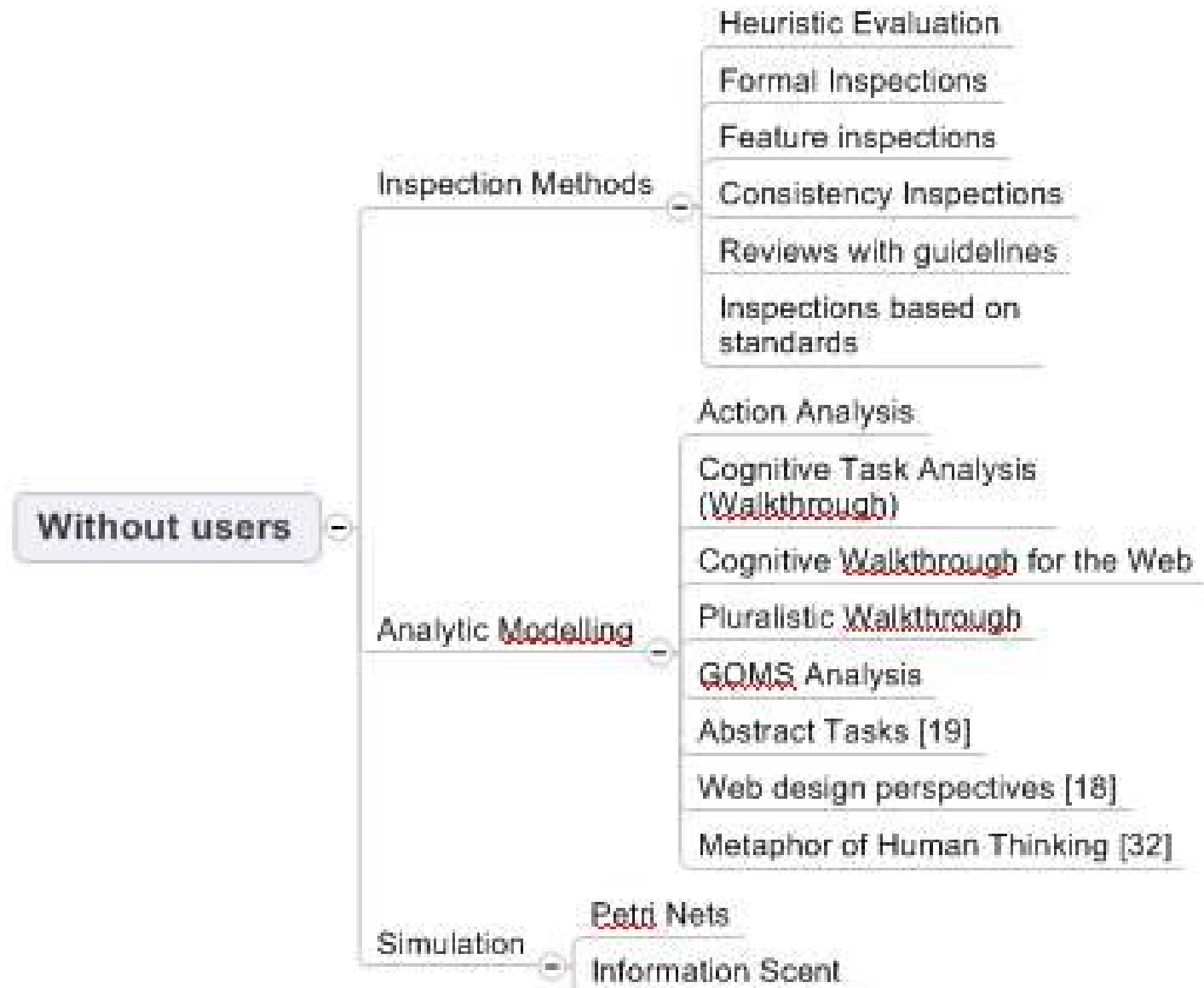
Gestión Calidad Software



GRADO II
ITINERARIO ING. SW
OPT. 4º CURSO 2º C.

UT sin usuarios

Técnicas



Técnicas de Modelado Analítico

<https://hcibib.org/tcuid/chap-4.html>

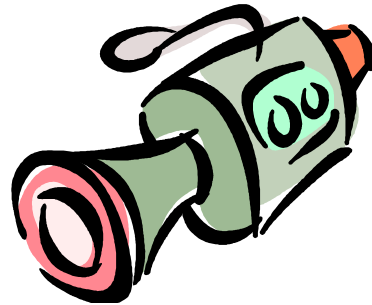
■ Paseo cognitivo

- Método orientado a la tarea: parte del análisis de tarea del UCA. Forma formalizada de imaginar los pensamientos y acciones de tus usuarios cuando usan tu interfaz POR PRIMERA VEZ (DIFERENCIA RESPECTO A ANÁLISIS DE ACCIÓN)
- Enfatiza temas cognitivos: learnability, memorability etc.
 - requiere perfiles de usuario y escenarios (ponerse en la piel de una persona concreta con unos objetivos concretos)
- El grupo de expertos va explorando las funcionalidades del sistema simulando el comportamiento paso a paso para una tarea determinada. Para cada paso intenta justificar por qué el usuario será capaz de realizar dicho paso. Para ello debe dar respuestas plausibles a cuatro preguntas:
 - ¿Pensará que es posible realizar la tarea?
 - ¿Verá la opción?
 - ¿Asociará la acción con el efecto?
 - ¿Percibirá que se está acercando a su meta?
- Dos fases: preparatoria (se selecciona la interfaz, los usuarios tipo, tareas y acciones que conforman la tarea) y de análisis (se simula el comportamiento del usuario en la aplicación)
- Crítica: Puede ser muy lento (bajo ratio coste/beneficio)
- LA EMPATÍA TIENE UN PAPEL RELEVANTE EN EL MÉTODO, ya que el evaluador debe motivar cada acción del usuario en base a su conocimiento sobre el usuario y el comportamiento de la aplicación.

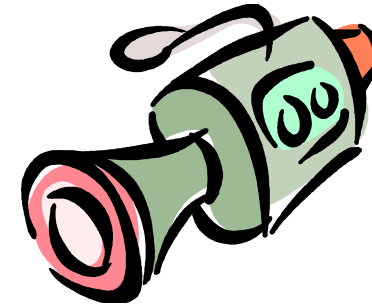
UT sin usuarios

Analíticas: paseo cognitivo

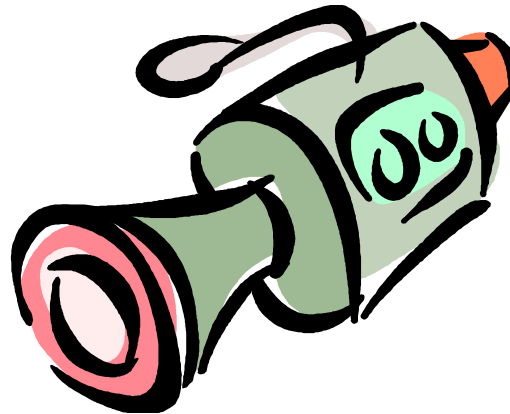
- Vídeo: Paseo cognitivo + observación de usuario



Cognitive Walkthrough
Itunes, part 1 (9:55; English)
<http://www.youtube.com/watch?v=Ro77wQq0sWo>



Cognitive Walkthrough
Itunes, part 2 (9:55, English)
http://www.youtube.com/watch?v=P72XbgQ-Z_k



Cognitive Walkthrough
www.thomann.com (accessed January 2013) (4:58)

UT sin usuarios

Analíticas: paseo cognitivo

- **Ventajas:**
 - Independencia de los usuarios finales
 - Uso de un prototipo totalmente funcional, lo que ayuda a la identificación efectiva de problemas reales que surgen de la interacción con el sistema
- **Inconvenientes:**
 - No hay usuarios finales
 - Requiere un prototipo completamente funcional
 - Posible bias debido a la selección incorrecta de tareas
 - Posible énfasis excesivo en detalles de poca importancia
 - Centrado en problemas de interacción relacionados con la cognición: learnability, memorability, etc.

- Paseo cognitivo plural
 - El 'paseo cognitivo plural' (pluralistic cognitive walkthrough) implica a un grupo examinando la interfaz. El grupo está compuesto por un experto y una serie de **usuarios finales**, desarrolladores, etc. Todos realizan las tareas y van discutiendo cada paso. Esta versión se puede realizar más pronto en el ciclo de desarrollo.

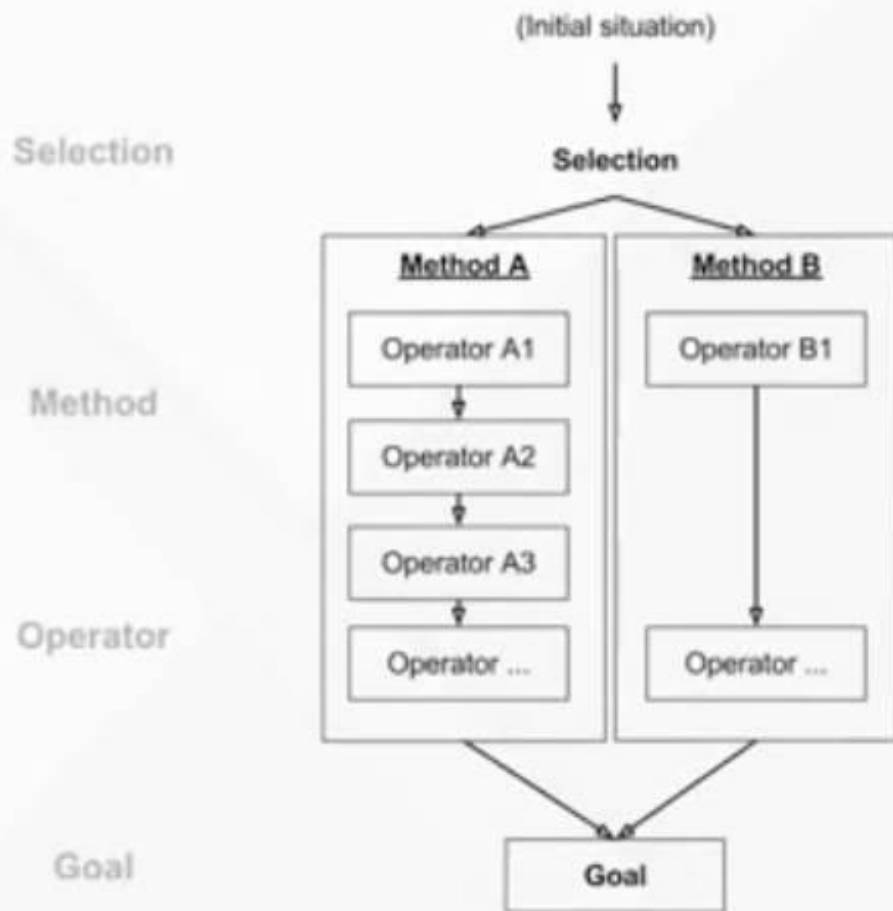
- Todos los métodos relacionados con el paseo cognitivo tienen ciertas características:
 - Usados sólo para evaluación formativa
 - Pueden ser usados con cualquier nivel de prototipo
 - No pueden ser usados hasta que no obtienes un cierto nivel de experiencia de dominio
 - No son buenos para evaluar tareas cognitivas (e.g. tareas que incluyen una toma de decisiones compleja como aplicaciones de e-commerce)
 - Tampoco sirven para evaluar tareas con metas que no están muy bien definidas, e.g. 'familiarízate con la interfaz y encuentra un artículo interesante que leer'.

■ Análisis de Acción

- Consiste en la descomposición de tareas en acciones elementales, y el estudio de cada acción. Centrados por tanto en task performance (y a veces tb error rate)
- Dos modalidades: formal y back-of-the-envelope.
 - **Formal:** El evaluador observa la secuencia de acción de usuarios reales (keystroke level analysis). Para ello divide la tarea en acciones individuales, del tipo move-mouse-to-menu o type-on-keyboard, y calcula los tiempos necesarios para realizar cada acción.
 - **Back-of-the-envelope:** Menos precisa, proporciona resultados tb menos precisos, pero es más rápida de ejecutar.
- Ventajas:
 - Predicción precisa de los tiempos asociados a las tareas.
 - Identificación de cuellos de botella en las tareas.
 - Conocimiento profundo del comportamiento del usuario en relación con la tarea
- Inconvenientes
 - Consume mucho tiempo. Requiere una experiencia muy elevada.
 - Su representación es muy complicada, incluso para tareas muy sencillas; en muchos casos las representaciones sólo pueden ser interpretadas por los que realizaron el análisis

■ GOMS Family: Goals – Operators – Methods – Selection Rules

- Modelan el rendimiento de una tarea mediante su descomposición
 - Goals: user goals
 - Operadores: acciones perceptuales, motoras o cognitivas de los usuarios
 - Métodos: formas alternativas de realizar la tarea (grupos de metas y operadores)
 - Selection rules: reglas que ayudan a elegir entre los distintos métodos



- **GOMS Family: Goals – Operators – Methods – Selection Rules**
 - Centrados en la TASK PERFORMANCE, y basados en tablas de tiempos muy validadas.
 - Suelen ser ejecutados durante el diseño temprano
 - Son cuantitativos (rendimiento, ratio de errores, learnability, ...)
 - Pueden ser también usados tanto para predecir rendimiento como para explicar mejoras de rendimiento detectadas durante un test de rendimiento (el modelo en que se basan puede ser usado tanto como modelo predictivo como explicativo)

- E.g. Tarea: recupera el artículo 'Why GOMS? De Bonnie John en ACM DL

Goal Structure

- Goal: Retrieve article from ACM DL
 - Goal: Go to ACM
 - Goal: Enter ACM URL
 - Goal: Submit URL
 - Goal: Go to DL
 - Goal: Locate DL link
 - Goal: Select the link
 - Goal: Select method
 - [Method: Search method
 - Goal: Search for article
 - Goal: Enter search parameters
 - Goal: Submit search
 - Goal: Identify article from results
 - Goal: Select the article]
 - [Method: Browse method - <take home exercise>]
 - Goal: Save article to disk
 - Goal: Initiate save action
 - Goal: Select location
 - Goal: save article to that location

- ¡GOMS asume que tienes un modelo válido!

- Build initial model based your own understanding of a task's execution
 - record other users performing the task
 - compare predicted versus actual sequence
 - refine and iterate

■ E.g. Método GOMS sencillo— Fingerstroke/Keystroke-Level Model

- Proporcionan estimadores de tiempo para cada operador. Fijaos en que incluye tanto tiempos de acciones físicas como cognitivas.

Operator	Brief description	Time, sec
T (tapping)	Inputting the repetitive commands, which is generally located in a designated area on the screen	0.31
P (pointing)	A relatively long movement from one position to the other position	0.43
D (dragging)	Dragging or moving an object on the touch screen	0.17
F (flicking)	Dragging as quickly and shortly as possible	0.12
R (response time)	System response time about input signal	Variable
M (mental thinking time)	Mentally preparing for physical acting	1.35

* Lee, A., Song, K., Ryu, H. B., Kim, J., & Kwon, G. (2015). Fingerstroke time estimates for touchscreen-based mobile gaming interaction. *Human Movement Science*, 44, 211-224.



- Calculadora Sencilla de Keystroke Level Model
 - <http://courses.csail.mit.edu/6.831/2009/handouts/ac18-predictive-evaluation/klm.shtml>

KEYSTROKE LEVEL MODEL (KLM) CALCULATOR

Enter an action string below to calculate its cost in the Keystroke Level Model.

Actions:

Time: 5.50 sec

Keystroke	0.28	sec
Button	0.1	sec
Point	1.1	sec
Home	0.4	sec
Mental	1.2	sec

- CogTool

- <https://cogtool.wordpress.com/overview-of-cogtool/>
- <https://github.com/cogtool/cogtool/releases/tag/1.2.2>

■ E.g. Método GOMS más complejos: MGOMSL

- Permiten crear modelos no sólo de la interacción sino también de procesos de toma de decisiones (algoritmos que tienen en cuenta finite capacity of short time memory, tiempo necesario para recuperar info de long-term memory, etc.)
- Permiten no sólo estimar el tiempo de compleción de tarea, sino tb predecir secuencias de operadores, el tiempo de aprendizaje, consistencia de la interfaz, ...
- Los métodos de la familia GOMS no son muy populares para la evaluación de apps móviles.

■ GOMS Family: Goals – Operators – Methods – Selection Rules

■ Ventajas:

- Predicen el rendimiento humano antes de tener que comprometerse con un diseño particular o tener que realizar estudios empíricos
- No requiere habilidades especiales
- Se ha validado en muchos estudios (funciona: rango de error inferior al 20%).
- Permite detectar cuellos de botella
- ESPECIALMENTE ÚTIL PARA TAREAS REPETITIVAS (e.g. Operadores telefónicos)

■ Inconvenientes:

- Asume un comportamiento habilidoso (usuarios expertos) y sin errores, racional y dirigido a la meta
- No existe una receta formal sobre como realizar la descomposición de tarea
- Puede requerir mucho tiempo para ser ejecutado (¿coste/ beneficio?).

■ Alternativa: Informal Action Analysis (Back of the envelope)

- Explicado en el libro online Task-Centered UI Design <https://hcibib.org/tcuid/>
- Reduce la precisión del GOMS: divide la tarea en acciones básicas (seleccionar un ítem de menú, escribir un nombre de fichero, etc.). que al menos tomen 2 o 3 segundos cada una. Después analiza cada una de esas acciones básicas para ver si se puede optimizar. ¿Cosas que deben ser aprendidas/recordadas? ¿Cosas que pueden ser realizadas fácilmente? En el proceso se busca encontrar formas de optimizar la experiencia.

■ Ejemplo

- Buscar artículo en ACM Digital Library

Action Analysis Example

- Enter URL String
- Press “Enter” key
- Find “digital library link”
- Select the link
- [assume search method]
- Enter title of article into search field
- Select “Search”
- Find “Why GOMS” link
- Select the link
- Select “Save” button
- Select folder location
- Select “Save” button on dialog

12 Steps = [24, 36] seconds

My actual time = 28 seconds

■ Ventajas:

- Más rápida de ejecutar
- Más fácil para el evaluador principante
- Buena para aplicaciones donde la productividad es menos crítica

■ Inconvenientes:

- Menos exacta (mayor variabilidad)
- Más difícil comparar diseños alternativos que están cerca en cuanto al rendimiento predicho.

■ Ejemplo

- <https://www.coursera.org/lecture/ui-testing/case-example-action-analysis-Sq0Es>

- The task:

- Joe needs a new treadmill for his office (his previous one is broken). He would like to shop for a treadmill designed to work with a standing/walking desk (which he already has).
- After his frustration getting poor support from the small company he bought his previous treadmill from, he's decided he wants the backing of a large retailer.
- His task is to find a treadmill that meets his key requirements:
 - Can handle someone up to 300 pounds using it 3-5 hours a day
 - Can handle walking speeds in 0.1 MPH increments from 1 MPH up to at least 2.5 MPH
 - As large a tread as possible (width first, but also length)
 - Good reviews (and many reviews)
 - Under \$1200



Técnicas de Simulación

- Los modelos de simulación van un paso más allá de las técnicas analíticas: pretenden automatizar (predecir) las interacciones de los usuarios en base a teorías formales, a menudo con un alto componente matemático.
 - Si la teoría es suficientemente formal a nivel matemático como para hacer predicciones, podemos incluso comparar distintas interfaces de manera muy rápida en base a ese modelo matemático sin tener que realizar estudios comparativos.
 - Cuando la gente utiliza simulaciones, a menudo lo hace en conjunción con Monte Carlo Optimization.
- .

- La simulación se ha utilizado sobre todo para técnicas de entrada de información, debido a que el rendimiento del sistema motor de las personas es probablemente el área mejor cuantificada de HCI.
 - Ejemplo: Sistemas como Swipe (un tipo de shape writers system), i.e. la construcción de un keyboard donde las personas puedan introducir una palabra completa de un solo trazo, fue posible gracias a modelos formales y aproximaciones basadas en la optimización.
 - <http://www.shuminzhai.com/>



- La simulación también se puede usar para tareas cognitivas de más alto nivel, e.g. Modelos del comportamiento de web surfing de las personas (Pete Pirolli y sus colegas en PARC): permiten estimar qué link es más probable que un usuario presione simplemente examinando los tags del link relevantes.
 - ¿Os animáis a buscar información sobre cómo lo hacen?

Métodos de Inspección

- Evaluación Heurística (Técnicas de Checklist) (Nielsen and Molich, 1990)
 - Consiste en proporcionar una serie de principios o heurísticas de diseño a un grupo de entre 3 y 5 expertos. Ellos revisarán tu diseño de manera individual según esas heurísticas, y después agregarán resultados.
 - Define 10 principios de Usabilidad:
 - Visibilidad del estado del sistema (feedback)
 - Coherencia entre el sistema y el mundo (comprensión)
 - Control de usuario y libertad (acción)
 - Consistencia y estándares (comprensión)
 - Prevención de errores (feedback)
 - Reconocimiento mejor que Recuerdo (acción)
 - Flexibilidad y Eficiencia de uso (acción)
 - Estética y diseño minimalista (comprensión)
 - Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores (feedback)
 - Ayuda y documentación (feedback)

- No son las únicas reglas que existen!
 - Las reglas de Norman del libro 'The design of Everyday Things'
 - Los 16 principios de Tognazzini
 - Las ocho reglas de oro de Shneiderman
 - ...

- Ítems de los formularios de Evaluación Heurística se pueden extraer de muchas fuentes
 - Libros (Gold-standard)
 - E.g. Libro: **Visual Design: Ninety-five things you need to know. Told in Helvetica and Dingbats.**
 - Proyectos de investigación
 - **E.g. tesis doctoral: Evalweb**
 - Estándares y recomendaciones nacionales o internacionales
 - ISO, ANSI. E.g. ISO 9241-151 (web)
 - Recomendaciones de Expertos (Jacob Nielsen, HFI, etc.)
 - E.g. <https://www.nngroup.com/articles/>
 - Webs o blogs reputados
 - E.g. <https://webstandards.hhs.gov/> (web)
 - Publicaciones en revistas y conferencias relevantes
 - E.g.: Publicaciones de CHI
 - *ACM Conference on Human Factors in Computing Systems*
 - Recomendaciones específicas en función de hardware, medio o plataforma de tu aplicación
 - E.g Guías de estilo de Android
 - Experiencia personal y conocimiento de dominio

UT sin usuarios

Inspección: evaluación heurística

- Se pueden definir a distintos niveles de abstracción Si las heurísticas están orientadas a la solución, el recomendador puede generar una serie de recomendaciones a partir de su evaluación.
- Ejemplo:

An Example of A Heuristic

**If onboarding is used, the user can choose to skip it
(put the user in control)***

Users should be able to skip the onboarding sequence if they wish not to be interrupted by it or do not perceive a benefit in swiping through it. Put the user in control of their time and initial experience of the app. Onboarding should only ever appear on first app use, or until the user is signed-in.

- En la práctica, este tipo de heurísticas (orientadas a la solución) funcionan bien con dominios bien estudiados, como e-commerce

UT sin usuarios

Inspección: evaluación heurística

- Las heurísticas tb pueden definirse a más alto nivel, e.g. Interaction or Experience-oriented heuristics.

- E.g.

H2: Cross-platform service access*

Pragmatic aspects of the heuristic H2:

- the user can do all the things she/he wants with the service on their mobile device
- it should be easy to move between using the service with a mobile device and PC
- ...

Hedonic aspects of the heuristic H2:

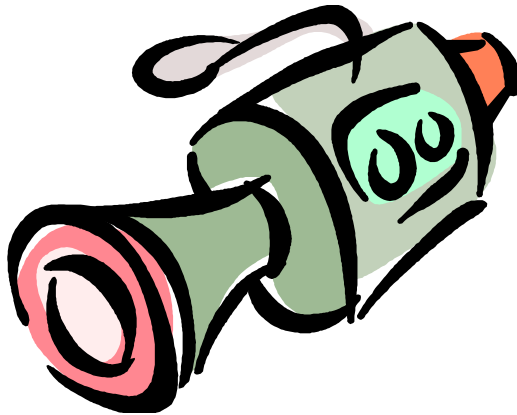
- it feels exciting to use either of the platforms in the usage contexts where the need for using the service arises
- task flow between the platforms/devices should be pleasant and natural.

- En comparación con las solution-oriented heuristics, permite encontrar problemas de distinto tipo, incluyendo posibles problemas de interacción

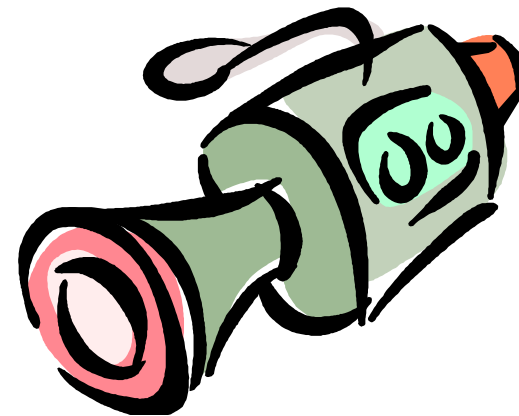
UT sin usuarios

Inspección: evaluación heurística

- Ejemplos de evaluación heurística (23:58)



Heuristic Evaluation
The ten Nielsen Heuristics (5:45)
<http://www.youtube.com/watch?v=hWc0Fd2AS3s>



Heuristic Evaluation
www.renfe.es (accessed January 2011)

UT sin usuarios

Inspección: Evaluación Heurística

- A la hora de preparar la ejecución de una evaluación heurística, se pueden desarrollar checklists más elaborados que los principios generales de Nielsen o de cualquier otro autor.
- Estas heurísticas pueden estar además adaptadas al contexto concreto de nuestra aplicación.
 - Make a usability checklist:
http://www.youtube.com/watch?v=yCa0g_4jfBw

Usability checklist para móviles:

- Heuristic Evaluation of Mobile Usability: A Mapping Study (2014)
- Heuristic Evaluation on Mobile Interfaces: A New Checklist (2014)
- Perspectives on usability guidelines for smartphone applications: An empirical investigation and systematic literature review (2018)

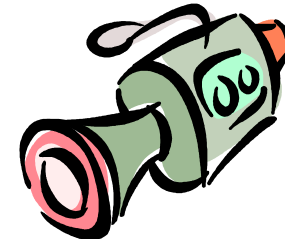
UT sin usuarios

Inspección: Evaluación Heurística

■ Evaluación Heurística mediante checklist

■ Ventajas:

- Aplicación de principios reconocidos y aceptados
- Intuitividad
- Facilidad para integrarse en fases tempranas
- Rapidez (cost-effective)
- Identificación efectiva de problemas mayores y menores
- Si utilizamos solution-oriented checklists, no es necesaria mucha experiencia ni de dominio ni de tecnología.



Usability.gov: Heuristic Evaluation
<http://www.youtube.com/watch?v=XWdqzN7USse>

■ Inconvenientes:

- Las heurísticas deben ser cuidadosamente seleccionadas para reflejar el sistema específico que está siendo evaluado (validez de la heurística)
- Requiere entre 3 y 5 evaluadores (no tan barato)
- Sensibilidad a la experiencia de dominio y de usabilidad de los evaluadores (sobre todo si son heurísticas definidas a un nivel de abstracción elevado)
- Separación de los usuarios, y por tanto dificultad para identificar necesidades desconocidas
- Posible énfasis excesivo en detalles de poca importancia
- **El checklist debería estar validado**

UT sin usuarios

Inspección: evaluación heurística

Proceso de validación empírica de una checklist de evaluación heurística

(Extraído de *Research-Derived Web Design Guidelines for Older People* Sri Kurniawan, Panayiotis Zaphiris)

- Parte de una checklist donde se incluyan criterios contrastados de usabilidad relevantes para tu proyecto (junto con sus referencias a las fuentes)
- Realiza un ejercicio de card sorting con usuarios (unos 15-20) para que organicen los criterios->haz un clustering
- Realiza un focus group con expertos (unos cinco) para que, mediante un affinity diagramming (card sorting grupal) reorganicen, añadan, refundan, reescriban, etc. Ítems y categorías
- Realiza un experimento (2 webs del tipo para el que se ha desarrollado el checklist, y 2 checklists, la original y la modificada)
 - Medir info objetiva (grado de acuerdo) y subjetiva (qué les parece la checklist)
- Dale la lista más robusta a un conjunto de usuarios finales de la web y realiza una evaluación basada en la checklist con ellos (con ayuda del investigador si hace falta). Grábala para registrar dudas, etc. Después de la evaluación pide que evalúen cada guideline en términos de Usefulness (de 1 a 5) y que justifiquen brevemente su rating. Por último pedirles que sugieran nuevas guías si piensan que falta alguna.

- La **estimación heurística** es una variante en la que a los inspectores se les pide que estimen la usabilidad relativa de dos o más diseños en términos cuantitativos (rendimiento de usuario esperado)
- Otra variante es la **Perspective-based Heuristic Evaluation**, donde cada experto se centra en un aspecto específico de la app. Se asume que de esta manera se incrementa la eficiencia.

■ Inspecciones Formales

- Combinan inspecciones individuales y de grupo en un proceso de seis pasos con roles bien definidos y mezclando elementos de evaluación heurística y una forma simplificada de paseos cognitivos.
- Meta: identificar el máximo número de defectos en la interfaz tan eficientemente como sea posible. El proceso de revisión incluye modelos de rendimiento a la hora de realizar tareas (*task performance*) y heurísticas, así como definiciones de perfiles de usuario y escenarios (que idealmente ya estarán definidos desde la fase de UCA).
- Ventajas:
 - Involucran a los diseñadores en el proceso, y los hacen más conscientes de las necesidades de usabilidad
 - Más eficiente y barato
- Inconvenientes:
 - Se pierde la visión de distintos stakeholders

**¿Cuándo usar
cada técnica?**

UT sin usuarios

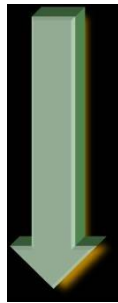
Cuándo usar cada técnica



Principales Técnicas

Evaluación Heurística

Inspección Formal



Principales Técnicas

Heuristic Evaluation

Cognitive Walkthrough

Pluralistic Cognitive Walkthrough

Action Analysis

Inspecciones formales