

Johnson (2007)

Interacción Humano Computador

Deyson Victor Diaz Ticona, Kevin Yunior Ccorimanya Paucar, Denilson Flores Valdivia, Marco Antonio Ponce de Leon

System Engineering School
System Engineering and Informatic Department
Production and Services Faculty
San Agustin National University of Arequipa

2020-09-16



Content

- 1 Analisis de Tareas
- 2 Principios de Interaccion Humano - Computador
- 3 References

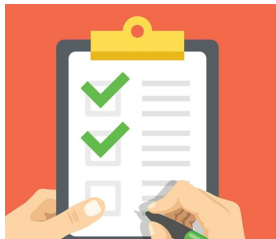
Analisis de Tareas

Describir en detalle cómo analizar los objetivos y tareas de los usuarios. Por ahora es suficiente decir que un buen análisis de tareas responde a estas preguntas:

- ¿Qué objetivos quieren alcanzar los usuarios al utilizar la aplicación?
- ¿Qué conjunto de tareas humanas pretende admitir la aplicación?
- ¿Qué tareas son más importantes y cuáles son las menos importantes?
- ¿Cuáles son los pasos de cada tarea?
- ¿Cuáles son el resultado y el producto de cada tarea?
- ¿De dónde proviene la información de cada tarea?
- ¿Cómo se utiliza la información que resulta de cada tarea?
- ¿Qué herramientas se utilizan para realizar cada tarea?
- ¿Qué problemas tienen las personas para realizar cada tarea? ¿Qué tipo de errores son comunes? ¿Qué los causa? ¿Qué tan dañinos son los errores?

Analisis de Tareas

Una vez que se responden estas preguntas (observando y / o entrevistando a las personas que hacen las tareas que la herramienta admitirá), el siguiente paso es comenzar a dibujar posibles interfaces de usuario. El siguiente paso es diseñar un modelo conceptual para la herramienta que se centra en las tareas y objetivos de los usuarios. Después de haber diseñado un modelo conceptual centrado en tareas, tan simple como sea posible, y lo más consistente posible, puede diseñar una interfaz de usuario que minimiza el tiempo y la experiencia necesarios para utilizar la aplicación para convertirse en un proceso automático.



Principios de Interaccion Humano - Computador

- Principio 1: Centrarse en los usuarios y sus tareas, no en la tecnología
- Principio 2: Considere la función primero, la presentación después
- Principio 3: Conforme a la visión de la tarea de los usuarios
- Principio 4: Diseño para el caso común
- Principio 5: No complique la tarea de los usuarios
- Principio 6: Facilitar el aprendizaje
- Principio 7: Entregue información, no solo datos
- Principio 8: Diseño para la capacidad de respuesta
- Principio 9: Pruébalo con los usuarios; entonces arrégalo

Principio 1: Centrarse en los usuarios y sus tareas, no en la tecnología

- Entender a los usuarios
- Entender las tareas
- Considere el contexto en el que funcionará el software

Para un correcto cumplimiento de los objetivos, la tecnología toma un lugar secundario y se prioriza el entendimiento del usuario y la eficacia del sistema.



Principio 2: Considere la función primero, la presentación después

- Desarrollar un modelo conceptual que cumple de manera completa el objetivo del mismo.

El cumplimiento de la tarea siempre será la prioridad.



Principio 3: Conforme a la visión de la tarea de los usuarios

- Lucha por la naturalidad
- Utilice el vocabulario de los usuarios, no el suyo
- Mantenga los componentes internos del programa dentro del programa
- Encuentre el punto correcto sobre el equilibrio entre potencia y complejidad

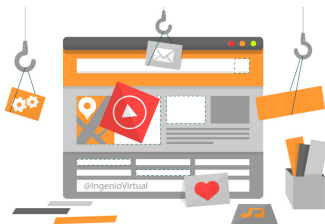
Para poder brindar algo optimo, el lenguaje a utilizar para los usuarios debe ser lo más simple posible, no resultar forzado para ellos, y debe ser sencilla y práctica.



Principio 4: Diseño para el caso común

- Hacer que los resultados comunes sean fáciles de lograr
- Dos tipos de "comunes": "cuántos usuarios" y "con qué frecuencia"
- Diseño para casos básicos; no te preocupes por los casos de "borde"

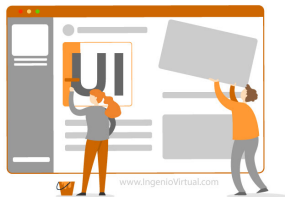
Como ya se mencionó, la practicidad y el nivel de interacción deben ser simples y prácticos para el entendimiento e interacción del usuario.



Principio 5: No complique la tarea de los usuarios

- No les dé problemas adicionales a los usuarios
- No hagas que los usuarios razonen por eliminación

Para el usuario, debe ser sencillo interactuar con el sistema, por esa razón las tareas que se le proporciona deben ser claras y exactas, no interponer varias acciones o tareas que puedan confundir a los usuarios.



Principio 6: Facilitar el aprendizaje

- Piense "de afuera hacia adentro", no "de adentro hacia afuera"
- Consistencia, consistencia, consistencia
- Proporcionar un entorno de bajo riesgo

Para el que interacciona, las tareas asignadas deben ser simples, que no comprometan la funcionalidad del sistema.



Principio 7: Entregue información, no solo datos

- Diseñe las exhibiciones cuidadosamente; conseguir ayuda profesional
- La pantalla pertenece al usuario
- Preservar la inercia de la pantalla



Principio 8: Diseño para la capacidad de respuesta

- Reconocer las acciones del usuario al instante
- Informar a los usuarios cuando el software está ocupado y cuando no
- Libera a los usuarios para que hagan otras cosas mientras esperan
- Animar el movimiento de forma suave y clara
- Permitir a los usuarios abortar operaciones prolongadas que no desean
- Permitir a los usuarios estimar cuánto tiempo tomarán las operaciones
- Trate de permitir que los usuarios establezcan su propio ritmo de trabajo

Para el usuario, las acciones deben ser comprensibles, para esto, la información que se le brinda debe estar a un nivel simple de entendimiento, con una retroalimentación moderada, y poder darle la opción de cancelar y medir el proceso de trabajo.

Principio 9: Pruébalo con los usuarios; entonces arréglalo

- Los resultados de las pruebas pueden sorprender incluso a los diseñadores experimentados
- Programe tiempo para corregir los problemas encontrados por las pruebas
- Las pruebas tienen dos objetivos: informativos y sociales.
- Hay pruebas para cada momento y propósito



References

- Introduction - xiii, Jhonson J. (2014). Designing with the Mind in mind. 2nd. edition.