

# Buenas prácticas en el diseño de software

## Guión

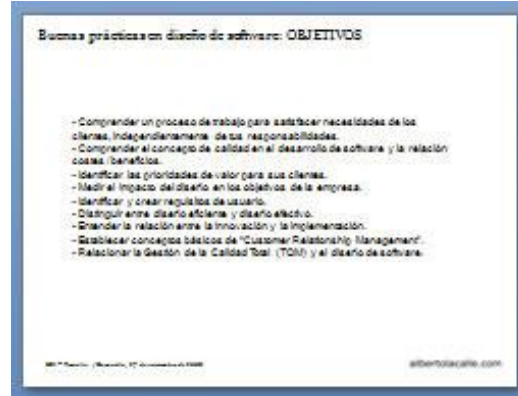
- Introducción
- Conceptos clave
- Test de usuarios
- Metodología y procesos de diseño
- Ejemplos y casos de uso.
- Preguntas y dudas

## Objetivos

- Explicar un proceso de trabajo para satisfacer necesidades de los clientes.
- Definir el concepto de calidad en el desarrollo de software.
- Identificar las prioridades de valor para los clientes.
- Medir el impacto del diseño en los objetivos de la empresa.
- Identificar y crear requisitos de usuario.
- Distinguir entre diseño eficiente y diseño efectivo.
- Establecer conceptos básicos de «Customer Relationship Management».
- Relacionar la Gestión de la Calidad Total (TQM) y el diseño de software.

## Introducción

#i0 y 1 La siguiente introducción tiene como **objetivo aclarar el marco donde se mueve el diseño de software en la empresa.**



#i2



Hecho: «la mayor parte de los proyectos son un caos, apenas se evalúan económicamente y la racionalidad es una excepción».

«Éxito» (15-30%): el proyecto se ha completado a tiempo, en el presupuesto y con las funcionalidades especificadas originalmente.

El proceso de diseño de un producto de software es complejo porque:

- Hay un elevado número de factores (internos y externos) que deben ser manipulados e incorporados.
- Los objetivos a cumplir pueden ser amplios.
- Hay personas, organizaciones y procesos involucrados.

#i2



Todas las empresas existen para generar VALOR.

Pero el valor es un concepto relativo, donde cada interesado lo entiende de forma distinta.

Valor para un negocio es el beneficio, para un empleado la seguridad laboral o un buen sueldo, para un gobierno que la empresa pague impuestos y cumpla las leyes...

El DISEÑO se centra en el VALOR para EL CLIENTE / USUARIO.

#i3



La empresa privada no puede sobrevivir si no proporciona valor a sus clientes.

ENTENDER LA PERCEPCIÓN de valor de los CLIENTES es un REQUISITO inicial en cualquier estrategia de negocio.

*Valor percibido = calidad percibida en relación al precio*

## #i4.1



Los clientes evalúan la CALIDAD según las necesidades de USO y de ESTIMA.

El diseño de software se ocupa de las necesidades TANGIBLES o de USO.

El diseño es por tanto un trabajo fundamentalmente ÚTIL (que trae o produce provecho), lo que algunos hoy llaman «usable».

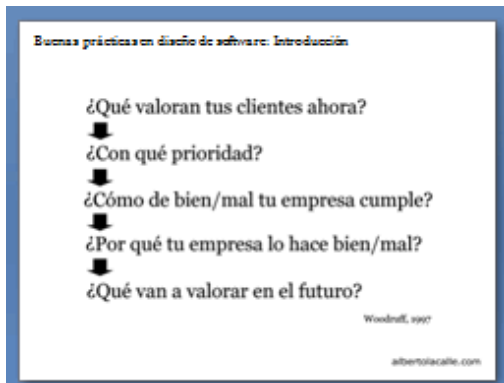
## #i4.2



Si no hablas el lenguaje del valor, sólo podrás competir por precio.

// caso: diseño gráfico es arte //

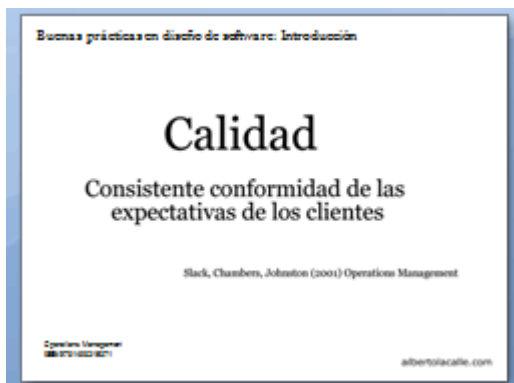
#i5



Identificar las prioridades de VALOR de los clientes es necesario para **invertir** o **medir el rendimiento de un producto**.

Sin una respuesta clara con DATOS y HECHOS a estas respuestas, cualquier diseño estará basado únicamente en SUPOSICIONES / INTUICIÓN, pero no en proceso metodológico.

#i7



La CALIDAD implica:

**Consistente:** (cumplir la especificación no es un hecho aislado, sino un proceso diseñado y controlado que aseguren el cumplimiento).

- a) Un proceso intuitivo NUNCA podrá serlo.
- b) Medir el impacto del proyecto.

**Conformidad:** (necesidad de cumplir las especificaciones).

//apenas existen las especificaciones de usuario en los proyectos o son de baja calidad //

#i8



La CALIDAD implica:

**Expectativas:** ni necesidades (requisitos básicos) ni deseos (todo, cualquier cosa).

Para la gestión total de la calidad (Total Quality Management), cumplir las expectativas de los clientes, significa más:

- Implica VER LAS COSAS DESDE LA PERSPECTIVA DEL CLIENTE. Ej: Caso del «servidor».
- Esto implica ver a la empresa como un **proceso**, que afecta a toda la organización, contra la visión tradicional de **funciones/competencias/departamentos**.

//comentar gráfico //

#i9

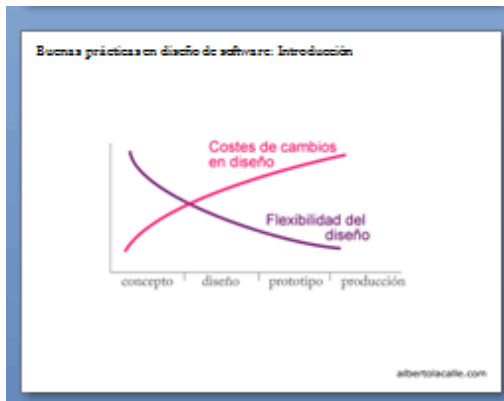


2: La mayoría de las empresas entienden el diseño como un ejercicio de estilo (modelo agencia).

3: El diseño es un proceso. Se usa para mejorar la EFICIENCIA, (y que son las técnicas que hoy voy a comentar).

4: Diseño como innovación, dirigiendo todas las actividades para satisfacer las necesidades de los usuarios.

#i10



Los costes de rectificar errores se incrementan cuanto más tiempo permanezcan en el proceso.

El **diseño de un producto y su proceso de creación no pueden separarse**, especialmente en los servicios, donde el proceso es el servicio.

Este aspecto está relacionado con el **proceso de desarrollo** (incremental y no en cascada) y las **competencias** de los componentes de un equipo.

## Conceptos clave



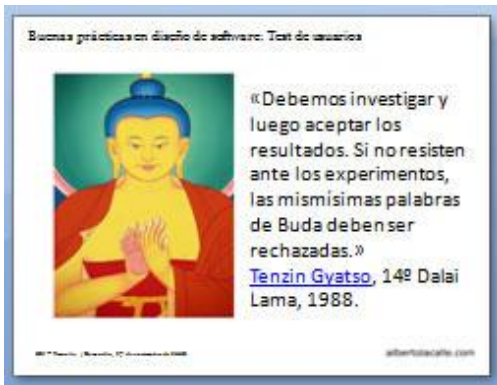
Conceptos:

- Diseño centrado en el usuario / usabilidad / experiencia del usuario.
- Diseño de la información / AI / diseño de contenidos / accesibilidad.
- Diseño de interfaz.
- Diseño de interacción.
- Diseño orientado a objetos / lenguajes de modelado vs. Natural / proceso dirigido por los **casos de uso**.
- Diseño de procesos / estándares.
- Diseño visual / gráfico.

//confirmar perfil audiencia//

## Test de usuarios

#t1



No a los dogmas ni a los «expertos» ni a los «gurús» ni al «porque yo lo valgo».

#t2.1



El testing es un análisis EMPÍRICO que verifica las especificaciones de requisitos en condiciones concretas.

Incluye los defectos de forma y los «bugs» (defectos de software/errores de programación).

- Testing: es el método de verificación más riguroso.
- Inspección: usando los sentidos o con medidas mecánicas.
- Demostración: probar que algo se puede hacer.
- Análisis: modelos matemáticos, simuladores, cálculos...
- Certificación: Basado en un certificado que alguien otorga.

//poner ejemplos//



## #t2.2



//pregunta: ¿no usar código fuente estándar es o no un «bug»?

- ñ vs. &ntilde;
- <title>nada</title>
- <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

//

Depende de lo que MIDAS obtienes un producto con una determinada calidad.

## #t2.3



**EMPÍRICO:** hechos experimentales y no en OPINIONES.

«intuitivo, «no es fácil de usar», «fresco», «altamente funcional», «diseño limpio», «diseño arriesgado», «aprueba en criterios de usabilidad», «de fácil uso», «claramente orientado a sus clientes».

#t3



Diferencia entre un **test de usabilidad/usuarios** y los tradicionales: «**beta testing**», «**testing**» o «**pruebas de software**»:

1. **El objetivo se especifica de forma concreta.**
2. Los participantes **son usuarios reales.**
3. Los participantes **realizan tareas reales.**
4. **Se observa y graba** lo que hacen y dicen.
5. **Se analizan los datos, diagnostican problemas y recomiendan cambios** para solucionarlos.

Un proceso que vale para **analizar cualquier producto** (productos de consumo, hardware, software, productos médicos...).

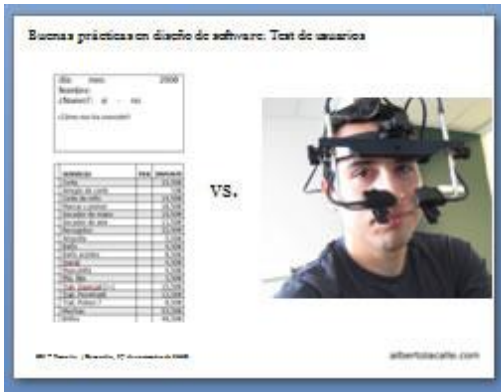
#t4



Matriz de verificación de requisitos.

La mayor parte de los test (si se hacen), se producen al final del proceso cuando está parcial o totalmente terminado el producto. En esa fase sólo los errores críticos se pueden arreglar, pero no el VALOR / CALIDAD del producto.

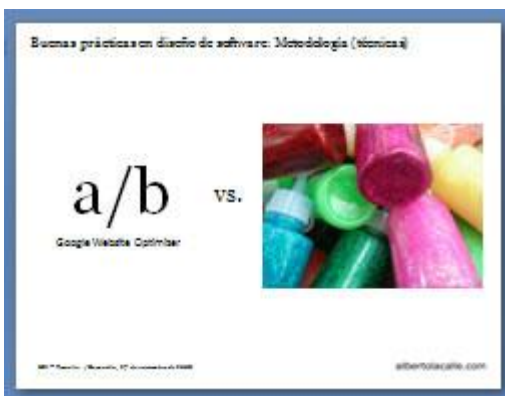
#t5



Valor + Metodología y no sólo la herramienta.

// gratis y tecnología punta //

#t6



// gratis y aclara diseños visuales y efectividad de contenidos//

## Metodología, procesos y casos de uso

El software no es ARTE, sino un producto de consumo, es por tanto un PRODUCTO industrial más.

#m1



Lo que los clientes quieren y compran, no es un software, sino los beneficios que ese producto produce.

Cuanto más maduro es el mercado, hay cada vez menos diferencia, y la tecnología no la genera por sí misma.

albertolacalle.com

#m2



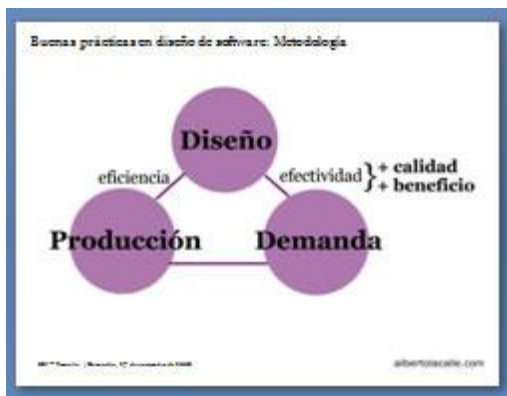
//comentar caso//

#m3



//comentar caso//

#m4



Diseño Eficiente (minimizar los costes de producción) //Código fuente de 350KB a 100KB //.

Diseño Efectivo (proporcionar los atributos demandados por los clientes en mercado).

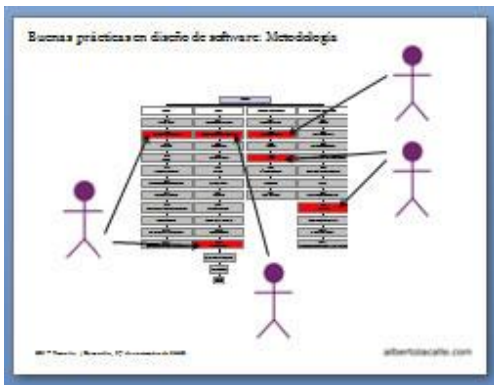
Se presta poca atención en definir el VALOR del cliente al planificar productos.

#m5



//comentar caso//

#m6



//comentar caso//

### **10 técnicas y ningún método**

A cada tipo de producto corresponde un tipo de ingeniería, entendida como el conjunto de métodos, técnicas y herramientas.

#m7 (1)



Análisis de NEGOCIO: Sin objetivos, no hay test que valga.

- ¿La inversión va a ser RENTABLE?: ¿cómo? ¿cuándo?.
- ¿Hay otras alternativas?.
- ¿Qué opción es la más rentable?: tiempo, dinero, recursos.

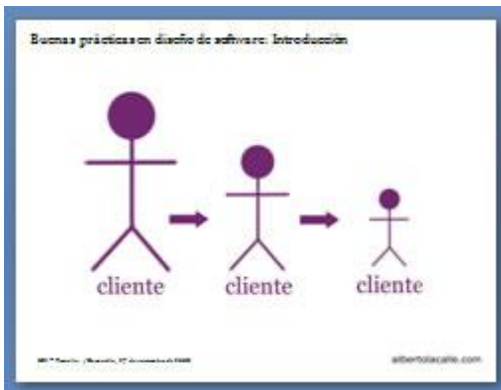
// Marqués de Riscal //

#m8 (2)



Los requisitos de usuario son fundamentales y deben incluir CASOS DE USO que describan las interacciones y PERFILES de usuario.

#m9 (3)



Identificar a **todos los usuarios** permite tomar decisiones con criterio sobre su grado de implicación en un proyecto para que tenga éxito.

¿Quién es realmente tu usuario?: medir riesgos

//caso del cliente del cliente del cliente: «todo para el pueblo pero sin el pueblo»//

#m10 (4)

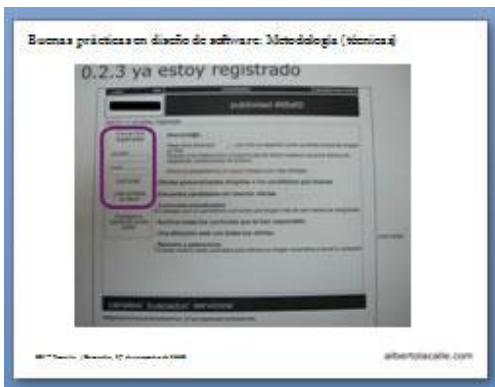




Recopila la mayor cantidad de información posible sobre los clientes que sean de UTILIDAD para poder dar **valor**. **Administrar la relación con los clientes**, «customer relationship management» **CRM**, es parte de una estrategia de negocio centrada en el cliente.

#m11 (5)

REUTILIZACION (componentes / marcos / patrones de diseño): No construyas/analices de nuevo algo que ya existe dentro o fuera de tu empresa.



//piezas de coches, casas...//

#m12 (6)



Ver es comprender: no escribas páginas y páginas de requisitos, potencia los requisitos VISUALES.

// Caso: «tiene pocas páginas para lo que vale, escribe más» //



#m13



Con planos, mapas de procesos.

#m14



Diagramas (ej: caso de uso).

#m15



Escenarios: Un escenario es una descripción narrativa informal (Carroll 2000) en forma de historia de actividades humanas o tareas.

Entender qué hace la gente, es un buen punto de partida para saber cómo funciona un producto.

//ej: caso Avis <http://albertolacalle.com/hci/escenario.htm> //

#m16



O prototipos.

#m17 (7)

Esta tabla de priorización de requisitos relaciona 'cómo' (cómo) con 'competencia' (competencia). El eje vertical 'cómo' incluye 'qué' y 'cliente'. El eje horizontal 'competencia' incluye '1', '2', '3'. La tabla muestra valores numéricos y letras que indican la prioridad de los requisitos.

		cómo			competencia		
		a	b	c	1	2	3
qué	cliente	10	4				SA
		5	1	7		B	KA
	importancia		45	35			A
	dificultad		2	5			

Legenda:  
 a: imposible  
 A: fuerte  
 B: marginal  
 S: ...

Utiliza metodología para priorizar los requisitos.

**Quality function deployment (QFD):** Método para transformar las demandas de los usuarios en calidad de diseño, priorizando las características del producto. De uso extensivo en Toyota.

// explicar tabla //

#m18



La forma de valorar el cómo, DEBE emplear un criterio de evaluación.

#m19 (8)



MEJORA CONTINUA / LECCIONES APRENDIDAS, usando auditorías al final de cada proyecto. Con consecuencias en la cultura / estructura / procesos / responsabilidades... de todos los implicados.

#m20



// comentar caso call-center //

#m21



No es natural para las empresas estar orientadas a sus clientes: inercia estructural / cultural. Ejemplo: las políticas internas dictan que no es una buena idea decirle a tu jefe que lo que publica es basura. // comentar caso //

## #m22 (9)



## Estructura de trabajo:

- Equipos multidisciplinares.
- Responsabilidad de los empleados en los resultados
- Jerarquías planas y descentralizadas.
- Formación reglada (ej: «arquitecto de experiencias»).
- Respeto profesional (ej: «me gusta, le gusta, te gusta»).

albertolacalle.com

**Conclusión**

## #c1



- Las decisiones deben apoyarse en datos con un método científico.
- Orientación a largo plazo.
- Los problemas de calidad se producen por malas gestiones, no por malos empleados.
- La calidad es un proceso infinito.
- Logro de resultados «¿y esto para qué vale?».
- Cooperación y colaboración interna y externa (empleados / clientes).
- Orientación al cliente.
- Ética profesional.