

Diseñar bien es efectivo y rentable

Los objetivos del Diseño

Por Daniel Mordecki

Cuando la disciplina de diseño se asume como diseño de la interacción, los objetivos quedan automáticamente determinados y son una poderosa herramienta para alinear los sitios y sistemas con los objetivos del negocio
14 de octubre de 2001

Los 5 objetivos del diseño

Partiendo del diseño de sitios o sistemas como Diseño de la Interacción¹ podemos determinar al menos 5 objetivos para el diseño:

- ✓ Definir el producto final
- ✓ Acotar y minimizar los costos
- ✓ Poner foco en el usuario
- ✓ Sacar la presión que el diseño implica para el equipo de programación
- ✓ Hacer creíbles y "cumplibles" los cronogramas

El conocimiento a priori de objetivos genéricos para el diseño de sitios y aplicaciones, más allá del objetivo específico de diseñar dicha aplicación o sitio, nos permite saber si vamos o no por buen camino. Preguntando por cada objetivo en concreto podemos determinar fácilmente si estamos o no diseñando en forma acertada.

Vale la pena resaltar una vez más que el diseño **precede** a la programación. Primero se diseña, luego se programa y después se decora. Confundir decoración con diseño es un error común y que se paga caro. Los objetivos aquí planteados tienen que conseguirse por tanto **antes** de que se escriba la primera línea de programación.

Vale la pena resaltar una vez más que el diseño **precede** a la programación. Primero se diseña, luego se programa y después se decora.

Definir el producto final

Definir el producto final antes de empezar a programar es algo tan obvio en la teoría como difícil de encontrar en la práctica.

Otras ciencias más maduras que la informática, como por ejemplo la arquitectura y la ingeniería civil, trabajan el tiempo necesario para cumplir con este objetivo antes de comenzar con las

¹ Ver: Diseño: los usuarios al poder I en <http://www.mordecki.com>

tareas de campo. Todas las herramientas son válidas: maquetas, planos, memorias descriptivas, animaciones y un bagaje enorme de técnicas permiten dar visiones completas del producto terminado a los clientes, a los inversores, a los constructores, a los albañiles, subcontratistas y todos los participantes de la obra sin necesidad de poner el primer ladrillo. El ejemplo es válido para trasladarlo a la industria informática. En ésta, apenas se conocen los primeros esbozos del sistema, se comienza a programar, con la ilusión de que programar y diseñar en paralelo ahorra tiempo y dinero. Lamentablemente muchas veces es considerado entre los programadores una viveza, un rasgo de picardía, obviar el trabajo previo de diseño y documentación

Lamentablemente (saltar el diseño) muchas veces es considerado entre los programadores una viveza, un rasgo de picardía, obviar el trabajo previo de diseño y documentación

Sin embargo definir correctamente el producto final tiene dos consecuencias altamente positivas: permite trazar el camino y permite manejar las expectativas de los clientes.

Trazar el camino

Recién al definir el producto final, es posible trazar el camino, desarrollar un plan del proyecto. Sin un destino, sin saber a dónde nos dirigimos, no se puede marcar un camino. Un viejo adagio dice que “todos los caminos son buenos para quién no sabe a dónde va” y parece haber sido escrito especialmente para esta ocasión.

El desarrollo de código, que se basa de la definición previa de las estructuras y modelos de datos, congela el proyecto. La supuesta posibilidad de que sobre la marcha se pueden ir modificando las características de los sistemas, es apenas una ilusión óptica que desconoce la realidad de que el diseño de software comprende decisiones sobre un cumulo de equilibrios y balances, y que para escribir la primera línea de código es necesario haber tomado todas y cada una de esas decisiones. Esto se puede hacer en forma explícita, separando el diseño de la programación o en forma implícita, asumiendo que “programamos un poco y después vemos”.

Solo la realización del diseño antes de la programación brinda a la gerencia del proyecto la capacidad de definir el camino a seguir por el proyecto.

Manejar las expectativas de los clientes

No es posible tener clientes satisfechos, más allá de algún golpe de suerte, si no somos capaces de manejar las expectativas de los clientes.

Todos nosotros queremos que nuestros clientes queden contentos, satisfechos con el producto que les entregamos, sean estos clientes internos, externos o ambos.

No es posible tener clientes satisfechos, más allá de algún golpe de suerte, si no somos capaces de manejar las expectativas de dichos clientes. La idea vulgar de que un cliente queda satisfecho cuando el producto es “bueno” es esencialmente falsa. Un cliente queda satisfecho cuando su percepción sobre el producto que se le entrega es igual o mejor que las expectativas que tenía.

La acumulación de features en una lista no permite manejar las expectativas de cliente alguno. Siguiendo con el ejemplo de la Arquitectura, la definición de features de una vivienda, dice bastante, pero no alcanza para generar expectativas: “Apartamento de 3 dormitorios, 2 baños, 110 m2, vista al frente, piso de cerámica importada, calefacción central” abarca una cantidad enorme de apartamentos completamente distintos. Por eso, entre otras cosas, la arquitectura brinda herramientas adicionales para que sus clientes se formen una idea mucho más acabada del producto que recibirán. En el diseño de sitios y aplicaciones ocurre exactamente lo mismo. Mientras que una “excelente búsqueda” para uno puede significar una generosa dotación de operadores lógicos, de cercanía, sinónimos, etc. para otro puede significar velocidad. Para un tercero puede implicar capacidad de distinguir entre los idiomas de diferentes documentos y para un cuarto capacidad de búsqueda no solo en textos, sino también en imágenes, sonidos, videos y otros formatos de archivos. Sin diseño, es decir, sin un correcto manejo de las expectativas de los clientes, solo la suerte puede salvarnos de que se defrauden cuando vean el producto terminado.

Acotar y minimizar los costos

Las leyes de Murphy rezan que un proyecto informático consume el 90% de los recursos en el primer 90% del trabajo y otro 90% de recursos en el 10% restante. Un corolario afirma que no vale presupuestar 180% para evitar el problema. Lamentablemente, esto se acerca peligrosamente a la realidad.

La lista de desvíos en los presupuestos y cronogramas de proyectos informáticos es extensísima y abarca desde pequeños proyectos hasta proyectos enormes, como por ejemplo el retraso de 2 años en el proyecto de la versión 4 de Windows, que terminó saliendo al mercado bajo el nombre de Windows 95.

Las dificultades en determinar costos reales para el desarrollo de los sistemas, derivan principalmente de la falta de diseño. La carencia de diseño determina que no se pueda trazar un camino, un cronograma confiable, y sin éste no hay presupuesto. Esto se agrava enormemente con los problemas que genera el hecho de que los clientes (internos y externos) se defrauden al ver los resultados del trabajo a medida que van apareciendo. Esto impone cambios que se suman a las definiciones tardías del sistema, que van contra las definiciones tomadas implícitamente al empezar a programar. Las pulseadas y roces que supone este proceso de interacción y sucesivas aproximaciones, genera en algunos casos verdaderas batallas campales de interna empresarial. Cuando los costos comienzan a dispararse, el departamento de sistemas comienza a acusar al resto de la empresa por las indefiniciones, y el resto de la empresa pide la cabeza de sistemas por los retrasos. El daño ya está hecho.

La carencia de diseño determina que no se pueda trazar un camino, un cronograma confiable, y sin éste no hay presupuesto

La medicina para este mal, el de que los costos se disparen es preventiva, y no curativa

El complemento de este proceso es el hecho de que la descripción de los sistemas en base a features se estima en horas hombre por parte de los programadores. El presupuesto se desarrolla en base a una lista de features, las horas hombre que lleva cada feature y el precio de dichas horas. De esta manera,

los programadores determinan unilateralmente que partes son más importantes y cuáles no lo son. Cuando los costos se disparan, comienza la hora de los recortes y allí se eliminan sin piedad los features más costosos. Este proceso perverso, lejos de ayudar, solo agrega leña al fuego, perjudicando enormemente la calidad del producto final y avivando la discusión de las distintas áreas que promovieron en su momento la inclusión de los features que ahora se eliminan.

La medicina para este mal es preventiva, y no curativa. El diseño de la interacción permite definir adecuadamente cada una de las áreas de interacción con los clientes y usuarios del sistema, dar a estos una visión completa de lo que obtendrán y en base a esto priorizar las áreas funcionales del sistema. También el diseño de la interacción permite desarrollar el camino desde la situación actual hacia el destino conocido, determinar el trabajo de cada parte no solo del área de sistemas, sino el involucramiento de todos los participantes del proyecto y en base a esto desarrollar un presupuesto óptimo, y además realista.

Poner foco en el usuario

Salvo en el discurso, el usuario no está representado en el proceso de desarrollo de sitios y sistemas. Es una realidad que hay que reconocer antes de poder corregirla. Los folletos declararán a gritos que la interfase es intuitiva y el sistema fácil de usar. Es más, expresiones rimbombantes como “interfase humana” “interacción natural” y similares plagan las descripciones y listas de features de los distintos productos. Sin embargo, los usuarios comunes y corrientes odian las computadoras y sienten que las computadoras los odian a ellos.

expresiones rimbombantes como “interfase humana” “interacción natural” y similares plagan las descripciones y listas de features de los distintos productos. Sin embargo, los usuarios comunes y corrientes odian las computadoras y sienten que las computadoras los odian a ellos.

La disociación entre intenciones y realidad tienen también una de sus causas profundas en las carencias de diseño en la creación de sitios y aplicaciones. La intención de poner foco real en el usuario es sincera, pero ante la presión de los plazos, la presión de los costos y la presión de la competencia, los desarrollos terminan recorriendo caminos que los alejan de las buenas intenciones y los acercan a la realidad: faltan las definiciones, el tiempo se termina y cada vez el desarrollo se parece más a un “vale todo”. Una prueba contundente de esta realidad es la documentación: en general, cuando está disponible, es pésima. Apenas colma la línea del contrato que decía: “documentación para el usuario”. Si los usuarios odian la computadora, mucho más odian la documentación.

Determinar con la mayor claridad posible el producto final, y con esta definición determinar el camino a seguir, previene de las desviaciones que las presiones imponen y acerca los sistemas al usuario. Siempre es posible torcer el camino hacia un derrotero erróneo, alcanza apenas con tomar una o dos decisiones equivocadas durante el proceso de desarrollo. Pero mientras que el diseño no garantiza que el usuario se va a sentir a gusto con el sitio o sistema, que va a alcanzar sus objetivos, la inversa sí es garantizada: la ausencia de diseño de la interacción genera sitios y sistemas hechos a la medida de los programadores, las presiones y las pulseadas entre los distintos departamentos.

Sacar la presión que el diseño implica para el equipo de programación

Quien alguna vez trabajó como programador conoce la desesperación de tener que desarrollar un sistema pobremente especificado. Mientras que las primeras armas como programador nos hacen creer que eso es una oportunidad para brillar, la vida nos enseña que es una carta blanca para criticar impunemente nuestro trabajo, incluir funcionalidad que no estaba prevista, obligarnos a hacer cambios enormes que de otro modo no hubieran sido necesarios.

Los emails, apuntes de las reuniones y todas las armas que tengamos serán insuficientes: tal o cual función es obvio que debió ser incluida y a pesar de que en su momento nadie oyó nuestras preguntas, nadie puso interés en las reuniones, nadie llenó los formularios, nosotros tenemos que programar. Los programadores asumen así, por omisión del resto del equipo la pesada tarea de diseñar: una tarea para la que no están capacitados y que se les obliga a desempeñar en las peores condiciones.

El diseño tiene teoría, objetivos, fundamentos, técnicas y metodología distinta que la programación. Es más, en muchos casos, los caminos son opuestos. Por ejemplo, mientras que la programación supone el análisis permanente y exhaustivo de la situaciones de borde, el diseño de la interacción prácticamente las ignora.

Obligar a los programadores a diseñar es en primer lugar injusto y en segundo lugar extremadamente contraproducente: se diseña muy mal y se quita tiempo a los programadores para que hagan el trabajo que realmente saben y tienen que hacer.

Hacer creíbles y cumplibles los cronogramas

Si bien este item podría haberse incluido dentro del item de costos por su parentesco, merece un capítulo aparte por sus consecuencias.

Las causas son prácticamente las mismas, pero las consecuencias son distintas: mientras que los problemas de costos impactan en la rentabilidad de los proyectos, la credibilidad es un intangible que cala mucho más hondo y los problemas de credibilidad tienen consecuencias mucho más profundas para las empresas. Las desviaciones en costos se arreglan con más dinero, la falta de credibilidad no.

Los propios cambios y definiciones tardías agravan el proceso de indefiniciones, que muchas veces se vuelve literalmente infinito

El proceso que ya describimos, donde se comienza a programar sin tener claro el producto final, tomando implícitamente decisiones de diseño que congelan la capacidad de incorporar determinada funcionalidad, generando un proceso de revisión permanente, de aproximación por ensayo y error, genera márgenes de incertidumbre mucho más importantes que los que se deberían asumir. Los propios cambios y definiciones tardías agravan este proceso, que muchas veces se vuelve literalmente infinito. Así es que es frecuente encontrar decisiones que indican que en una fecha determinada, el producto sale al aire “tal como esté” dando una muestra elocuente de que hace mucho que el

proyecto está fuera de control.

Si bien se puede creer que este es un problema del Área de Sistemas, asumir que Sistemas no es creíble es una forma miope de verlo. Hoy en día, sin sistemas no hay negocio. Sin nuevos sistemas no hay nuevos negocios. La falta de credibilidad en el desarrollo de los sistemas se vuelven falta de credibilidad en el desarrollo de negocios, falta de credibilidad en la empresa.

Diseño y Análisis de sistemas

Vale para terminar hacer algunos comentarios entre la relación del diseño y el análisis de sistemas.

Los programadores tendemos a confundir a ambos, dado que tanto un buen análisis como un buen diseño conduce a un buen sistema. Sin embargo no son equivalentes. El análisis, las técnicas y metodologías del análisis informático parten de la base de que el diseño ya está hecho. El análisis informático toma un problema, define para él una estrategia de desarrollo, un modelo de datos asociado y permite la realización de una codificación adecuada. El diseño de la interacción parte de la realidad para especificar el problema.

Diseño de la Interacción y Análisis de Sistemas son entonces tareas imprescindibles pero disjuntas. El diseño precede al análisis.

Diseño de la Interacción y Análisis de Sistemas son entonces tareas imprescindibles pero disjuntas. El diseño precede al análisis.

A modo de conclusión

La incertidumbre y falta de credibilidad que el desarrollo de sitios y sistemas impone a las empresas, la impotencia ante el empantanamiento de los proyectos, que de herramienta de cambio se vuelven lastre que mantiene el status quo, que de arma competitiva se vuelven desventaja, merecen un cambio profundo en la forma de concebir y encarar estos procesos.

Las carencias en diseño de la interacción aparecen como uno de los caminos para revertir esta tendencia. No es el único. Tal vez ni siquiera sea el mejor. Pero está al alcance de la mano, con objetivos claros, técnicas definidas y e implementaciones realizables. No supone gastos sino todo lo contrario: no nos cabe duda de que el diseño de la interacción es para la empresa una opción eficiente y rentable: un gran negocio.