

¿Hasta dónde hay que perseguir la pureza técnica?

Suficientemente bueno

Por Daniel Mordecki

La lista de los productos que "los clientes no supieron valorar" es muy larga. El mercado es un juez cruel a la hora de determinar quién vive y quién muere, y la defensa de la pureza técnica no parece ser la clave que dirime la contienda.
14 de mayo de 2006

Miremos el PS/2. Trajo con sigilo el diskette de 3^{1/4} de doble cara, que persiste hasta nuestros días, la memoria de 72 pines, la pantalla VGA, el BIOS configurable por software y el MicroCanal, un bus de comunicaciones interno que se dio el lujo de equipar a Deep Blue, la computadora que le ganó a Gari Kasparov, el campeón mundial de ajedrez, en mayo de 1997. El PS/2 era un producto técnicamente ejemplar, y su legado tecnológico es tan grande como imponente su fracaso comercial. Le ganaron la partida propuestas como la arquitectura EISA o la tarjeta de video Hércules, así como algunas otras tecnologías que pese a su proliferación, siempre tuvieron el gusto retroamargo de la mediocridad.

Fijemos la atención ahora en LISA, ¡Parió la interfase gráfica!. ¿Qué más se puede decir de una computadora que trajo al mundo las ventanas, el ratón y el escritorio? Se puede decir que fue un fracaso comercial rotundo, que no logró convencer a sus clientes y que por ello tuvo una cortísima vida: lanzada en 1979, murió en silencio y casi en el olvido en 1986, aplastada por el avance glamoroso de Macintosh.

Perseguir la pureza técnica atenta contra la idea de suficientemente bueno y esto es negativo a la hora de llegar al mercado con un producto de software.

Y así llegamos a Windows. Primero MacOS, después OS/2, ahora Linux. La lista de ventajas técnicas de todos y cada uno de ellos es extensísima frente a un sistema inestable, inseguro, propietario, las más de las veces ilógico e incomprensible, poco elegante en sus soluciones y por sobre todas las cosas técnicamente seguidista y copión. Pero la realidad pura y dura, la que manda dice que el noventa y ??? de las computadoras del mundo usan Windows. Y la cifra, mal que nos pese, parece casi casi inamovible, como grabada en granito.

Suficientemente bueno

Voy a arriesgar una hipótesis, a proponer un denominador común que en mayor o menor medida está en todos estos fracasos¹: *Perseguir la pureza técnica atenta contra la idea de*

¹ Cuando decimos fracaso no hablamos de productos que no andan ni de empresas que pierden plata, sino de productos que soñaron con todo el mercado y se quedaron apenas con migajas.

suficientemente bueno y esto es negativo a la hora de llegar al mercado con un producto de software.

Es sin duda una hipótesis arriesgada porque la calidad técnica tiene sus beneficios tanto funcionales como económicos, para quién desarrolla un producto y para quién lo usa. Por eso figura la palabra "pureza" y no "calidad". También porque luchar por alcanzar la perfección es una virtud deseable en las organizaciones. Pelear por cada detalle, ser intolerante con lo que sabemos que no está bien, o que puede estar mejor. Planificar, ejecutar y analizar lo hecho para volver a comenzar una y otra vez, puliendo, limando, perfeccionando es una de las claves para una organización sustentable y exitosa.

El mercado masivo elige productos excelentes en algunos aspectos y suficientemente buenos en el resto. Y por sobre todas las cosas ignora algunos preciosismos técnicos destinados al nichos de los apologistas

Pero también es una hipótesis realista. Porque el mercado, la justicia incontrastable del mercado se aburre de ponérselo delante de nuestras narices y solo la obstinación religiosa nos impide verlo.

El mercado masivo elige productos excelentes en algunos aspectos y suficientemente buenos en el resto. Y por sobre todas las cosas ignora algunos preciosismos técnicos destinados al nichos de los

apologistas. Tal vez sea esto lo que más nos duele y nos es más difícil de tragar. Porque la mayoría de los que estamos en el negocio de la informática somos apologistas fanáticos, y nuestra generosidad nos dice que lo que es bueno para nosotros debe ser bueno para los demás.

Un programador le comenta a otro:

- La moneda soporta decimales

y el otro ametralla sin mediar respiro:

- ¿La cantidad de dígitos es configurable?

El sentimiento de culpa del primero es inconmensurable, infinito. ¿Cómo se puede cometer un error semejante? Fijar en dos la cantidad de decimales y programar un sistema en el que la moneda o tiene o no tiene decimales es casi una herejía, bah, es una HEREJÍA a secas. Sin embargo el mercado no hizo aún su pregunta, una sola pregunta: ¿en qué país del mundo se usan más de 2 decimales? Corolario. Si hay un país en así en el mundo: ¿existe la más remota posibilidad de vender el software a ese país?

La diferencia en el nivel de complejidad de una opción frente a la otra es palpable, sensible. Problemas en el diseño de la base de datos, en la definición de las variables, en la utilización de las mascarar de presentación y captura de datos, sin hablar del incremento en la complejidad del testing. Pero la culpa, nuestra culpa, como buenos fieles de la cofradía de devotos del byte no es menos palpable, y si pudiéramos pediríamos unos días más para hacerlo configurable,

Tal vez debemos comenzar a traducir "mejor" como "más equilibrado" y abandonar la costumbre de traducirlo como "más sofisticado".

como debe ser. ¿Y si pedimos tiempo para modificar otra cosa, y mechamos unas horitas para hacer la cantidad de decimales configurable? Total ...

Tal vez la verdad sea que no existe una pureza tecnológica absoluta. Tal vez tengamos que conformarnos con el hecho de que la sofisticación y complejidad debe ser proporcional y relativa al fin y uso del producto que la implementa y no al revés. Tal vez debamos comenzar a traducir "mejor" como "más equilibrado" y abandonar la costumbre de traducirlo como "más sofisticado".