

No Hay FUD* Como un FUD Antiguo

Por el Dr. Glyn Moody

Publicado 14:11, 20 11 de enero en: <http://blogs.computerworlduk.com/open-enterprise/2011/01/theres-no-fud-like-an-old-fud/index.htm>

El Economista ha estado escribiendo artículos mal informados sobre el código abierto desde hace años - [he analizado](#) un ejemplo particularmente atroz en el 2006. Así que es difícil decir si los defectos en esta reseña de un [libro nuevo](#) se han reducido a la antipatía, o si son inherentes al título que describe, "El Código Mezclado". En lo que respecta a este último aspecto, la siguiente información no inspira confianza:

Por su tono académico, es improbable que el libro vaya a volar de las estanterías, incluso en zonas con una gran cantidad de piratas informáticos (que está seguro de que se ofendieron por el hecho de que **los autores recibieron dinero de Microsoft para su investigación**, de hecho el archi-enemigo de el movimiento de código abierto, -a pesar de que aseguran a los lectores que los fondos llegaron sin condiciones).

Como yo no he leído el libro - y prefiero no pagar £ 25.95 para el dudoso placer de descubrir dónde vienen los errores - Voy a limitarme a hacer frente a los argumentos expuestos en la revisión economista en lugar de preocuparme sobre dónde su origen.

En primer lugar, tenemos lo siguiente:

Los desarrolladores de código abierto, por ejemplo, se cree ampliamente que todo son voluntarios que aman escribir software. Si bien esto puede haber sido cierto en los primeros días de la informática, la motivación y los antecedentes de los programadores actuales es mucho más variada. Muchos de ellos trabajan para empresas que desarrollan tanto código abierto y software propietario y los combinan en todo tipo de modelos de negocio. Casi el 40% de las empresas encuestadas pertenecen a esta categoría.

Bueno, no, en realidad: esto ha sido evidente por años. Cualquier persona que todavía cree que este material se está escribiendo en los dormitorios no ha estado prestando atención claramente.

Esto también viene a ser bastante meh-digna:

La encuesta también indica que los dos mundos de software son mucho más "mezclados" que a sus respectivos campeones les gustaría. Más de una cuarta parte de las empresas felizmente mezcla y combina ambos tipos, en particular en los países más pobres.

Lo que significa que la gente no arranca cada pieza de software propietario en su negocio para sustituirla por completo con código abierto, pero tienden a utilizar una combinación de ambos? Bueno, ¿Quién lo hubiera pensado? En realidad, más en serio, creo que esa cifra de "más de un cuarto" subestima seriamente la práctica, que me imagino que es mucho más cercano al 100% en estos días. Si se trata de una figura auto-reportada es probablemente más un indicio de que muchos CTOs no saben realmente lo que está pasando en sus departamentos de Tecnología Informática IT (que ha sido así por mucho muchos años cuando se trata de código abierto.)

Pero mi principal preocupación aquí es con la siguiente sección:

"Sin embargo, la constatación que los defensores de código abierto que menos le gusta es que los programas libres no son siempre más barato. Para estar seguro, el costo inicial del software propietario es más alta (aunque los programas de código abierto no son siempre gratis). Pero las empresas que utilizan este tipo de programas gastan más en cosas como aprender a utilizarlos y hacerlos trabajar con otro software."

Sí, es una **vieja variante que Miedo Incertidumbre y Duda FUD** de edad que el software libre no es realmente libre (oh Dios mío, ¿verdad?) Es lo mismo que Microsoft trató de hacernos creer hace unos diez años y luego abandonó cuando se dio cuenta de que nadie dijo que era cuando se tiene en cuenta todos los factores como el pago de los salarios. Pero dejemos esto de lado también, que no es novedad para nadie, veamos la idea central de esta encarnación actual de ese Miedo Incertidumbre y Duda FUD:

"Las empresas que utilizan este tipo de programas gastan más en cosas como aprender a utilizarlos y hacerlos trabajar con otro software"

Así que ocurre lo mismo con la primera parte significa que aprender a usar una nueva pieza de software de código abierto es inherentemente más difícil que aprender a usar una nueva pieza de software propietario? Yo no he visto una sola pieza de la investigación que sugiera esto. Lo que he visto ampliamente documentado es que las personas que actualmente utilizan Microsoft Office, por ejemplo, les resulta más difícil aprender a usar OpenOffice, por ejemplo, que seguir utilizando Microsoft Office. ¿Lo cuál es, por supuesto, un pedazo de sabiduría firmemente ubicada en el corazón mismo de la Tierra de los Sangrantes.

Por lo tanto, pasando rápidamente de la esperanza de que puede haber una cuestión de fondo aquí, tenemos la segunda afirmación: "que las empresas gastan más en hacer trabajar el código abierto con el "software". Pero espera, ¿qué podría ser, a que "otros programas" se refieren? Ya que no es de código abierto (porque es "otro", no el de código abierto) es claramente software propietario, de modo que **el problema se reduce a hacer trabajar los programas de código abierto con el software propietario. ¿Y por qué podría ser eso?**

Bueno, podría ser porque el software cerrado es, por definición, **CERRADO**, con los fabricantes que son generalmente inútiles cuando se trata de proporcionar información que pueda ayudar a otros a trabajar con sus productos (porque quieren mantener a sus super-duper "secretos", en secreto). En otras palabras, **el problema no radica en código abierto pero con el software de código cerrado, lo que lo hace innecesariamente difícil de convivir en las aplicaciones de otros proveedores (sea usando código abierto o no).**

De hecho, es este problema - la dificultad de utilizar otro software, posiblemente abierto con aplicaciones propietarias - que **es una de las razones claves por las que las empresas quieren migrar al código abierto, para evitar este tipo de LOCK-IN*** para romper el círculo vicioso. Pero decir que el código abierto es más caro debido a los problemas causados por el software propietario que está tratando de reemplazar es rico de verdad. Lo correcto sería decir que **el software propietario tiene costos ocultos que se manifiestan cuando las compañías tratan de utilizar un nuevo software, por**

ejemplo, el código abierto. En otras palabras, es al revés.

Para ser justos, en el artículo de El Economista se menciona la solución a este problema:

*"los gobiernos deben asegurarse de que las dos formas de software **COMPITAN** en igualdad de condiciones y puedan mezclarse eficientemente. Una forma de hacerlo sería promover estándares abiertos para garantizar que los titulares de propiedad no abusen de una posición dominante."*

Lo que - o el libro - no tienen en cuenta es que esto es precisamente lo que las empresas de software propietario han estado luchando desesperadamente para evitar. La batalla por la definición de estándares abiertos en el [Marco Europeo de Interoperabilidad 2.0](#) es sólo el ejemplo más reciente.

En otras palabras, una vez más **son los titulares de código cerrado el problema aquí, tratando de inclinar el campo de juego a su favor, y para asegurarse de que todavía hay costos asociados con hacer que el código abierto trabaje con sus ofertas de código cerrado por insistiendo en los derechos de licencia u otras condiciones** (¿asi cómo puede ser eso un estándar "abierto"?)

Estas dificultades deben ser vistas como otra razón de peso para preferir el software libre y verdaderos estándares abiertos que permiten el uso de la principal software de licencia libre, la **GNU Licencia Pública General GPL**. Esto **aseguraría los proveedores verdaderamente COMPITAN en igualdad de condiciones**, no sesgados por un bloqueo en los enfoques de propiedad patentada. Las soluciones de código abierto obligan a los vendedores de software propietario a reducir sus márgenes hasta el hueso si quieren competir en estos términos completamente justos, y los clientes se benefician de este proceso de selección darwiniana.

***FUD** - Miedo Incertidumbre y Duda

***Lock-in** - Tiene no directa traducción al Español. -Consideremoslo un Anglicanismo del finales del siglo XX y principios del XXI- Sonaría muy tonto e incomprensible algo como "candado", pero mucho mas claro como "PRISION". -Esta palabra viene a significar literalmente la forzosa sumisión a formatos electrónicos que hacen muy difícil escaparse de ellos. -Por ejemplo una persona u organización que ha usado Microsoft Office durante mucho tiempo. Cuando sale una nueva versión es FORZADO a PAGAR por ella por que Microsoft SIEMPRE ha creado y creará incompatibilidades artificiales, y echarle la culpa a la competencia. -Lease los Comes vs Microsoft para ver como esto ha sido, es, y será una de sus tácticas para crear dependencia en sus productos. Por ello viene a ser un instrumento de dominación. El **Colonialismo Digital**, del que debemos romper cadenas AHORA para bienestar nuestro y de las futuras generaciones.