

Novática, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática). **Novática** edita también **UPGRADE**, revista digital de **CEPIS** (Council of European Professional Informatics Societies), en lengua inglesa, y es miembro fundador de **UPENET** (**UPGRADE European Network**)

<<http://www.ati.es/novatica/>>
<<http://www.upgrade-cepis.org/>>

ATI es miembro fundador de **CEPIS** (Council of European Professional Informatics Societies) y es representante de España en **IFIP** (International Federation for Information Processing); tiene un acuerdo de colaboración con **ACM** (Association for Computing Machinery), así como acuerdos de vinculación o colaboración con **AdaSpain**, **AIZ** y **ASTIC**.

Consejo Editorial

Antoni Carbonell Noguera, Juan Manuel Cueva Lovelle, Juan Antonio Esteban Iriarte Francisco, Juan Crespo, Celestino Martín Alonso, Josep Molas i Bertrán, Olga Pallás Codina, Fernando Píera Gómez (Presidente del Consejo), Ramón Puigjaner Trepal, Miquel Sàrries Griño, Asunción Yturbe Herranz

Coordinación Editorial

Rafael Fernández Calvo <rfcalvo@ati.es>

Composición y autedición

Jorge Lázcan Gil de Ramales

Traducciones
Grupo de Lengua e Informática de ATI <<http://www.ati.es/gl/lengua-informatica/>>

Administración

Tomás Brunete, María José Fernández, Enric Camarero, Felicidad López

Secciones Técnicas: Coordinadores

Administración Pública electrónica
Gumersindo García Arribas, Francisco López Crespo (MAP)

<gumersindo.garcia@map.es>, <flc@ati.es>

Arquitecturas

Jordi Tubella Murgadas (DAC-UPC) <jordit@ac.upc.es>

Victor Viñals Yufiera (Univ. de Zaragoza) <victor@unizar.es>

Auditoría ética

Marina Tourinho Trullitro, Manuel Palao García-Suelto (ASIA)

(Escuela Superior de Informática, UCLM) <manuel@palao.com>

Bases de datos

Coral Calero Muñoz, Mario G. Piattini Velthuis

(Escuela Superior de Informática, UCLM) <Coral.Calero@uclm.es>, <mplattini@inf-cr.uclm.es>

Derecho e tecnologías

Isabel Hernández Colados (Fac. Derecho de Donostia, UPV) <ihernando@legaltek.net>

Elena Davara Fernández Marcos (Davara & Davara) <edavara@davara.com>

Enseñanza Universitaria de la Informática

Joaquín Ezepeleta Mateo (CPS-UZAR) <ezepeleta@posta.unizar.es>

Cristóbal Pareja Flores (DSP-UCM) <cpareja@sip.ucm.es>

Gestión del Conocimiento

Juan Baiget Solé (Cap Gemini Ernst & Young) <juan.baiget@ati.es>

Informática y Filosofía

José Corco Jorinúa (UC) <jcorco@unica.edu>

Esperanza Marcos Martínez (ESCET-URJC) <cuca@escet.urjc.es>

Informática Gráfica

Miguel Chover Solís (Universitat Jaume I de Castellón) <chover@lsi.uji.es>

Roberto Vivó Hernández (Eurographics, sección española) <rvivo@dsic.upv.es>

Ingeniería del Software

Javier Dolado Cosín (DLSI-UPV) <adolado@si.ehu.es>

Luis Fernández Sanz (PDSI-EJ-UEM) <lfern@dpriis.esi.uem.es>

Inteligencia Artificial

Federico Barber Sanchis, Vicente Botti Navarro (DSIC-UPV)

<fvotti_barber@fira.upv.es>

Interacción Persona-Computador

Julio Abascal González (FI-UPV) <julio@si.ehu.es>

Jesús Lorés Vidal (Univ. de Lleida) <jesus@eup.udl.es>

Internet

Alonso Álvarez García (TID) <alonso@ati.es>

Llucen Pagès Casas (Indra) <pages@ati.es>

Lengua e Informática

M. del Carmen Ugarte García (IBM) <cuarte@ati.es>

Lenguajes Informáticos

Andrés Martín López (Univ. Carlos III) <amarin@it.uc3m.es>

J. Angel Velázquez Hurtado (ESCET-URJC) <a.velazquez@escet.urjc.es>

Librerías e Informática

Alonso Escolano (FIR-Univ. de La Laguna) <aescolano@ull.es>

Xavier Gómez Guinovart (Univ. de Vigo) <xgg@uvigo.es>

Manuel Palomar (Univ. de Alicante) <mpalomar@disi.ua.es>

Mundo estudiantil

Adolfo Vázquez Rodríguez (Rama de Estudiantes del IEEE-UCM)

<a.vazquez@ieee.org>

Profesión Informática

Rafael Fernández Calvo (ATI) <rfcalvo@ati.es>

Miquel Sàrries Griño (Ayto. de Barcelona) <msarries@ati.es>

Redes y servicios telemáticos

Luis Gujardo Colombia (DCOM-UPV) <lgujardo@doom.upv.es>

José Solís Pareta (DAC-UPC) <pareta@ac.upc.es>

Seguridad

Javier Arellito Bertolin (Univ. de Deusto) <jarellito@eside.deusto.es>

Javier López Muñoz (ETSI Informática-UMA) <jlm@lcc.uma.es>

Sistemas de Tiempo Real

Alejandro Alonso Muñoz, Juan Antonio de la Puente Añaró (DIT-UPM)

<aalonso.igunte@di.upm.es>

Software Libre

Jesus M. González Barahona, Pedro de las Heras Quirós

(GSYC-URJC) <frob.gheras@gsyc.esct.urjc.es>

Teconología de Objetos

Jesus García Molina (DIS-UM) <jmolina@correo.um.es>

Gustavo Rossi (LIFIA-UNLP, Argentina) <gustavo@sol.info.unlp.edu.ar>

Technologías para la Educación

Juan Manuel Dodero Barro (UC3M) <dodero@inf.uc3m.es>

Technologías y Empresa

Pablo Hernández Medrano (Bluemart) <pablohm@bluemart.biz>

TIC para la Gestión
Valentín Masero Vargas (DI-UNEX) <vmasero@unex.es>

TIC y Turismo

Andrés Aguayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga)

<(aguayo, guevara)@cc.uma.es>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos. **Novática** permite la reproducción de todos los artículos, a menos que lo impida la modalidad de © o copyright elegida por el autor, debiéndose en todo caso citar su procedencia y enviar a **Novática** un ejemplar de la publicación.

Coordinación Editorial, Redacción Central y Redacción ATI Madrid
Padilla 66, 3º, dcha., 28006 Madrid
Tfn. 914029391; fax 913093685 <novatica@ati.es>

Composición, Edición y Redacción ATI Valencia
Av. del Reino de Valencia 23, 46005 Valencia
Tfn./fax 963303032 <secretgati@ati.es>

Administración y Redacción ATI Cataluña
Ciudad de Granada 131, 08018 Barcelona
Tfn. 934125225; fax 934127113 <secretgen@ati.es>

Redacción ATI Andalucía
Isaac Newton, s/n, Ed. Sadiel,
Isla Cartuja 41092 Sevilla, Tfn./fax 954460779 <secretand@ati.es>

Redacción ATI Aragón
Lagasca 9, 3-B, 50006 Zaragoza.
Tfn./fax 976235181 <secretara@ati.es>

Redacción ATI Asturias-Cantabria <gp-astucant@ati.es>

Redacción ATI Castilla-La Mancha <gp-clmancha@ati.es>

Redacción ATI Galicia
Recinto Ferial s/n, 36540 Silleda (Pontevedra)
Tfn. 986581413; fax 986580162 <secretgati@ati.es>

Suscripción y Ventas
<<http://www.ati.es/novatica/interes.html>>, o en ATI Cataluña o ATI Madrid

Publicidad
Padilla 66, 3º, dcha., 28006 Madrid
Tfn. 914029391; fax 913093685 <novatica.publicidad@ati.es>

Imprenta
Derra S.A., Juan de Austria 66, 08005 Barcelona.
Dedición legal: B 13.134-1975 -- ISSN: 0211-2124; CODEN NOVACE

Portada: Antonio Crespo Foix / © ATI 2005
Diseño: Fernando Agrieta / © ATI 2005

editorial	> 02
Las patentes de Software: el gran revolcón	
Andalucía: otra Ley de Colegios excluyente e inoperante	
en resumen	> 02
El Software Libre en el diván	
<i>Rafael Fernández Calvo</i>	
monografía	
El Software Libre como objeto de estudio	
(En colaboración con UPGRADE y con la cooperación del proyecto europeo CALIBRE)	
Editores invitados: <i>Jesús M. González Barahona, Stefan Koch</i>	
Presentación. El Software Libre al microscopio	> 03
<i>Jesús M. González Barahona, Stefan Koch</i>	
CALIBRE en la cresta de la ola europea del Software de Código Abierto	> 05
<i>Andrea Deverell, Par Agerfalk</i>	
¿Será el movimiento del Software Libre el nuevo escalón en el modelo de organización de la producción en el sector TI?	> 06
<i>Nicolas Jullien</i>	
Debian 3.1 (Sarge) como caso de estudio de medición del Software Libre: resultados preliminares	> 11
<i>Juan José Amor Iglesias, Jesús M. González Barahona, Gregorio Robles Martínez, Israel Herráiz Taberner</i>	
El análisis institucional aplicado al estudio del Software Libre como "bien comunal"	> 15
<i>Charles M. Schweik</i>	
Sobre proyectos de Software Libre / Código Abierto de "puerta cerrada": enseñanzas del enfoque de selección de desarrolladores para Firefox de Mozilla	> 23
<i>Sandeep Krishnamurthy</i>	
Agilidad y desarrollo de Software Libre	> 27
<i>Alberto Sillitti, Giancarlo Succi</i>	
secciones técnicas	
Enseñanza Universitaria de la Informática	
Propuesta de objetivos formativos para el primer curso de las Ingenierías Informáticas y de algunas estrategias docentes para conseguirlos	> 31
<i>Fermín Sánchez Carracedo, Ricard Gavalà Mestre</i>	
Gestión del Conocimiento	
Escenarios del conocimiento: conocimiento orgánico e inorgánico	> 36
<i>Juan Baiget Solé</i>	
Ingeniería del Software	
La gestión de la diversidad de procesos por los informáticos: reflexiones	> 38
<i>Daniilo Caivano, Corrado Aaron Visaggio</i>	
Aspectos pragmáticos en el Desarrollo por el Usuario Final	> 45
<i>José Antonio Macías Iglesias</i>	
Lenguajes informáticos	
Mono: mucho más que una implementación libre de .Net	> 48
<i>Jordi Mas i Hernández</i>	
Lingüística computacional	
Procesamiento y aplicaciones de los corpus paralelos	> 50
<i>Xavier Gómez Guinovart</i>	
Redes y servicios telemáticos	
Extensión del servicio A/V Streaming de CORBA. Modelo empírico basado en el control de tráfico	> 55
<i>Antonio Javier García Sánchez, Felipe García Sanchez, Pablo Pavón Mariño, Joan García Haro</i>	
Encaminamiento inter-dominio con calidad de servicio basado en Overlay Entities distribuidas y QBGP	> 61
<i>Marcelo Yannuzzi, Alexandre Fonte, Xavier Masip Bruin, Edmundo Monteiro, Sergi Sánchez López, Marilía Curado, Jordi Domingo Pascual</i>	
Referencias autorizadas	> 68
sociedad de la información	
Programar es crear	
Un evento que mejora cada año: el Concurso Universitario de Programación de la Comunidad Autónoma de Madrid (CUPCAM)	> 74
<i>Adolfo Vázquez Rodríguez</i>	
Dominó Solitario (CUPCAM 2005, problema A)	> 75
<i>Antonio Fernández Anta</i>	
asuntos interiores	
Coordinación editorial / Programación de Novática	> 76
Normas de publicación para autores / Socios Institucionales	> 77

Monografía del próximo número:
"Estandarización y normalización en Seguridad"

Nicolas Jullien

Groupe M@rsouin (Môle Armoricaín de Recherche sur la SOciété de l'information et les Usages d'INternet)

<Nicolas.Jullien@enst-bretagne.fr>

¿Será el movimiento del Software Libre el nuevo escalón en la evolución de la organización de la producción en el sector TI?

© Se permite la reproducción de este artículo siempre que sea completa, no tenga modificaciones y se cite al autor.

Traducción: José Alfonso Accino Domínguez (Grupo de Lengua e Informática de ATI)

1. Introducción

La difusión de productos de Software Libre¹ termina cambiando la manera en que se producen, distribuyen y venden los programas, y por tanto puede originar profundos cambios en la organización industrial de la TI Tecnología de la Información (TI). No sería, ni mucho menos, un fenómeno excepcional, dado que la estructura industrial en el campo de las TI ha sufrido dos grandes cambios en los últimos quince años². Teniendo en cuenta este hecho, podemos pensar si no estaremos en vísperas de una nueva estructura industrial y si no estará basada en una organización libre. Para ello, en la primera parte mostraremos que la industria de TI presenta hoy las mismas características ya vistas en las evoluciones anteriores. En la segunda parte mostraremos los argumentos que nos hacen creer que la organización libre se está convirtiendo en el modelo dominante de organización en la industria informática

2. Algunas características de la industria informática

2.1. Particularidades económicas

Ante todo, un programa de software puede considerarse como un "bien público" ya que³:

- "No es causa de rivalidad, en el sentido de que no escasea y, una vez producido, todo el mundo puede beneficiarse de él.
- No es exclusivo, en el sentido de que una vez producido, es imposible impedir que la gente acceda a ese bien".

Además, este bien no se gasta con el uso, de manera que se puede comprar una sola vez para todos. La segunda característica del producto informático es que no está hecho de una pieza, sino de la superposición de varios componentes: hardware (con una pieza específica, el microprocesador), el sistema operativo y los programas. Esto implica una coordinación entre los diferentes productores, o que un solo productor fabrique todos los componentes.

La tercera característica, que en realidad es una consecuencia de las dos primeras, es que los productos informáticos y especialmente el software están sujetos a "retornos de adopción crecientes", por usar el término de [1]. Este ha definido cinco casos de "retornos de adopción crecientes" que tienen un impacto directo que va desde el usuario individual hasta el mercado en su conjunto, y los cinco

Resumen: la difusión del Software Libre representa una de las transformaciones más importantes de la industria de Tecnología de la Información (TI) en los últimos años; tampoco puede decirse que sea la menos sorprendente. En este artículo intentaremos situar esta difusión en su contexto histórico. En primer lugar mostraremos que la industria de TI presenta hoy las mismas características que ya se vieron en anteriores transformaciones y presentaremos las razones que explican por qué pensamos que el software libre puede convertirse en un modelo dominante de organización en la industria informática.

Palabras clave: código abierto, economía industrial, evolución del sector TIC, Software Libre.

Autor

Nicolas Jullien defendió su tesis doctoral sobre Economía del Software Libre en 2001. Actualmente se encarga de coordinar un grupo de investigación sobre los usos de la TI en Bretaña (Francia) llamado M@rsouin (Môle Armoricaín de Recherche sur la SOciété de l'information et les Usages d'INternet, <<http://www.marsouin.org>>). También gestiona el proyecto europeo CALIBRE sobre software libre para GET (Groupe des Écoles des Télécommunications, <http://www.get-telecom.fr/fr_accueil.html>), grupo francés participante en el proyecto.

están presentes en la industria del software:

- **Efecto aprendizaje**, que significa que se aprende a utilizar un programa, pero también un lenguaje, haciendo difícil cambiar a otro que nos ofrezcan.
- **Efecto red**: las opciones que toman las personas con las que intercambiamos tienen un impacto en nuestra evaluación de la calidad de un bien. Por ejemplo, aunque un determinado editor de texto no sea el más adecuado para sus necesidades, es posible que usted lo elija porque todo el mundo le envía documentos en ese formato, de manera que lo necesita para leer esos textos,
- **Economía de escala**: dado que la producción de piezas de ordenador implica costes fijos sustanciales, el coste medio por unidad disminuye cuando la producción aumenta. Especialmente, éste es el caso del software, donde hay casi únicamente costes fijos (esto es consecuencia de sus características de bien público).
- **Retornos crecientes de información**: se habla más de Linux porque está ampliamente distribuido,
- **Interrelaciones tecnológicas**: como ya se ha dicho, un componente de software no funciona solo, sino con otras piezas de software. Lo que da 'valor' a un sistema operativo es la cantidad de programas disponibles para ese sistema. Cuanta más gente escoja un sistema operativo, mayor disponibilidad habrá de aplicaciones para ese sistema, y viceversa.

Todo ello significa que esta industria tiene cuatro características originales, según [31].

- Siendo el software un "bien público", sus costes de desarrollo y producción no dependen del tamaño de la población de usuarios, por lo que este grupo puede ampliarse a un coste, si no nulo, sí prácticamente despreciable en comparación con los costes de desarrollo.
- El ritmo de innovación es rapidísimo, porque dado que el producto no se destruye con el uso, sólo se pueden conseguir nuevas ventas con productos innovadores o al menos diferentes; esto tiene como resultado la reducción de la vida del producto.

Estas dos características llevan a una competencia feroz, a unas políticas de precios agresivas y a que las marcas intenten imponer sus soluciones como estándares, a fin de beneficiarse del monopolio.

Las otras dos características son consecuencias del "efecto red" y de las "interrelaciones tecnológicas":

- Las empresas que poseen un programa tienen un incentivo para desarrollar nuevos componentes de software que complementen el que ya tienen, pero son incapaces de responder al amplio espectro de la demanda relacionada con el componente original (especialmente si hablamos de programas clave, tales como sistemas operativos). Por tanto, al mismo tiempo, aparecen nuevas marcas para responder a las nuevas necesidades.
- La consecuencia es que los estándares juegan un papel muy importante porque hacen posible que bienes complementarios

funcionen juntos. También aquí, el hecho de controlar un programa, si ello supone controlar un estándar, es un activo. Pero para satisfacer todas las demandas se tienen que publicar las características de estándar o por lo menos una parte del mismo ...

Conocer estas características y sus consecuencias nos ayuda a comprender las transformaciones de esta industria desde su aparición a mediados del siglo pasado.

2.2. Progreso tecnológico, nuevos mercados, desintegraciones verticales y nuevos "regímenes de competencia"

Como veremos, cada período se caracteriza por una tecnología que ha permitido a las empresas proponer nuevos productos a nuevos consumidores.

1. Un concepto tecnológico dominante ...:

en el primer período (desde mediados de la década de 1940 a mediados de la de 1960) no había diferencia real entre hardware y software, y los ordenadores eran 'únicos', unos productos de investigación contruidos para un único proyecto. Gracias al progreso tecnológico (miniaturización de transistores, compiladores y *sistemas operativos*), en el segundo período (desde 1960 a principio de 1980), el ámbito de uso se extendió en dos direcciones: la reducción de tamaño y precio de los ordenadores, que hizo crecer el número de empresas que se los podían permitir, y el aumento de potencia de computación, que permitió que un mismo ordenador se utilizara para diferentes tareas.

Pero el principal cambio que caracterizó este período fue que el mismo programa podía instalarse en diferentes ordenadores (de una misma gama), permitiendo que el programa evolucionara, creciera en tamaño y sirviera a un número creciente de usuarios. El ordenador se convirtió en un bien 'clásico' que se podía cambiar cuando no sirviera o fuera demasiado viejo, pero sin perder la inversión en software. Con la llegada del microprocesador, comenzó el tercer período a finales de los 70. Una vez más, el ámbito de uso se extendió en dos direcciones: aumento de potencia y reducción de tamaño y precio en ordenadores de baja gama. El concepto tecnológico dominante era que el mismo programa podía empaquetarse y distribuirse a distintas personas y organizaciones, de la misma manera que cualquier otro bien tangible.

2. ... para un uso dominante: en el primer período los ordenadores era una herramienta de cálculo, o de investigación, para centros de investigación (a menudo militares). En el segundo período (de los 60 a los 80) se habían convertido en herramientas para proceso de información centralizado para las organizaciones (estadísticas, pago de salarios, etc) y el tamaño de las que podían tener acceso a este

recurso se hizo cada vez menor durante este período. El tercer período es el del proceso de información personal, pero de tipo profesional.

3. ... y un tipo dominante de retorno creciente de adopción: siendo una herramienta para especialistas, donde cada proyecto permitía a los fabricantes y usuarios comprender mejor las posibilidades de tales máquinas, el primer período estuvo dominado por el aprendizaje y el uso, y por tanto con costes significativos en I+D. En el segundo período, este efecto de "aprendizaje por el uso" no desapareció, ya que los usuarios podían llevarse sus programas de fabricación casera cuando cambiaban de ordenador. Esta posibilidad también generó el nuevo "*retorno creciente de adopción*" dominante: las interrelaciones tecnológicas.

Como de hecho un programa se desarrollaba y trabajaba con un solo sistema operativo, una vez iniciada la relación comercial con un productor se hacía difícil para un cliente romperla. A cambio, este cliente ya no necesitaba ni siquiera comprender el hardware de la máquina. Como en el segundo período, este efecto no desapareció en el tercero, pero éste está dominado por la economía de escala debido a la distribución de ordenadores y, especialmente, a la organización de la producción de PC⁴ pero sobre todo gracias al desarrollo de programas estandarizados [25].

Estas características tecnológicas proporcionan elementos que ayudan a comprender mejor la estructura de la industria informática: los "*retornos crecientes de adopción*" proporcionan una posición dominante a las compañías que los controlan.

En el primer período, cuanto más se participaba en un proyecto, más posibilidades se tenían de proponer innovaciones para el siguiente, gracias al conocimiento acumulado. Esto explica la rápida emergencia de siete compañías dominantes (en los EE.UU.). El segundo período lo inició IBM con la salida de la Serie 360, la primera familia de ordenadores que compartía un mismo sistema operativo. Al final de este período IBM era la firma dominante (incluso denunciada por abuso de posición monopolística) aunque los recién llegados, HP y Digital, habían ganado una posición significativa con los miniordenadores.

Una vez que estas compañías instalaban un ordenador para un cliente, las interrelaciones tecnológicas hacían que este cliente tuviera que afrontar unos costes importantes si cambiaba a otra familia que utilizase otro sistema operativo. Y cuantos más clientes tuvieran, más podían invertir en I+D para mejorar la eficacia de su familia de ordenadores, pero también más podían gastar en marke-

ting para captar nuevos clientes. Una vez más, esto favoreció la concentración en el negocio de fabricación.

En el tercer período, nuevamente, los ganadores fueron los que controlaban los elementos clave del ordenador en términos de interrelación tecnológica: los sistemas operativos todavía, pero también los microprocesadores. Fueron las compañías que se llevaron la mayor parte de los beneficios de la economía de escala, ya que la competencia hizo caer los precios en los demás sectores, en particular en las máquinas que antes habían sido fuente de beneficios elevados, pero también en otros componentes.

Si esta estandarización es uno de los elementos clave que hace posible la distribución de ordenadores, también genera algunas ineficiencias ya que el control de tales estándares por una sola compañía la lleva a abusar de su posición dominante o monopolista. Esta sospecha se dió a finales de los setenta con respecto a IBM y hoy Microsoft ha sido denunciada por abusos de posición dominante. No es nuestra intención debatir la realidad de estas prácticas, pero la existencia de tales procesos demuestra que hay quien cree que la redistribución de los retornos crecientes de adopción no se está haciendo eficientemente.

3 ¿Estamos en vísperas de un nuevo avance en la historia de la industria de las TI?

3.1. Necesidad de productos "de masas y a medida"

3.1.1. Nuevas tecnologías

Durante los noventa, con la llegada de Internet, la principal evolución técnica en TI fue, naturalmente, la generalización de las redes de ordenadores, tanto *dentro* como *fuera* de las organizaciones. La miniaturización también hizo posible la aparición de una nueva gama de productos 'nómadas': organizadores (Psion y Palm), reproductores musicales, teléfonos móviles. Todo esto entra dentro de la constante evolución de los productos TI. Se ha pasado de una única máquina, dedicada a una tarea conocida de antemano, y reservada para toda la organización, a múltiples máquinas conectadas que se usan para realizar diferentes tareas que cambian con el tiempo y que están integradas dentro de varias organizaciones. La conexión en red, el intercambio entre sistemas heterogéneos y la comunicación entre estas máquinas se han convertido en algo crucial.

Paralelamente a esta transformación, las tecnologías de programación de software también han evolucionado [17: 126-128]: la llegada de lenguajes de programación orientada a objetos (C++, Java) permitió la reutilización de componentes software ya desarrollados. Esto ha llevado al concepto de "software modular": la idea es desarrollar un conjunto de pequeños programas soft-

ware (módulos o componentes software) cada uno de los cuales tendría una función específica y que se podrían asociar y utilizarse con cualquier máquina ya que sus interfaces de comunicación serían estándar.

3.1.2. El nuevo retorno creciente de adopción

Por tanto, la difusión de Internet y el aumento de los intercambios fuera de la organización han convertido el factor red en el "retorno creciente de adopción" dominante.

3.1.3. Nuevos usos dominantes

Estos programas y estos materiales están a menudo producidos por diferentes firmas, para distintos usuarios. Es necesario entonces que estas firmas garanticen su disponibilidad en el futuro, a pesar de los cambios de versión.

En efecto, entre las empresas clientes la demanda se ha hecho cada vez más heterogénea debido a la conexión en red de distintos sistemas y la necesidad de los usuarios que trabajan en ellas de compartir las mismas herramientas. Los programas software (y más específicamente los paquetes software) tienen que adaptarse a las necesidades y conocimientos de cada individuo sin perder los beneficios de la economía de escala o la estandarización de los programas en los que se basa la solución.

Resulta lógico entonces que las empresas cliente deban buscar más soluciones abiertas que les garanticen mayor control. Por ejemplo, Internet no ofreció un 'protocolo' con vistas a permitir la transmisión simple de datos, puesto que ya existía, sino uno lo bastante simple y flexible como para permitirle imponerse por sí mismo como el estándar para intercambio. Tanto es así que Horn [19] defiende la idea de que podemos haber entrado en una nueva fase productiva: la producción "de masas y a medida".

3.1.4. ¿Por una nueva organización industrial?

Sin embargo, estas relaciones de servicio no han demostrado ser bastante eficientes. Cuando se examinan las encuestas de satisfacción que se hacen referentes a productos de TI², se comprueba que la gente está satisfecha con el ordenador en sí, pero no con el servicio post-venta, especialmente con los programas software. La tendencia principal expuesta en la encuesta "01 Informatique" es que el cliente quiere mejor soporte, antes y después de la venta. También quiere que se le ayude a resolver sus dificultades y que sus necesidades queden satisfechas.

Hemos encontrado todos los elementos presentes en vísperas de un nuevo período en la organización de las TI: algunas transformaciones técnicas correspondientes a la evolución de la demanda y para la cual la actual organización industrial se muestra relativamente ineficaz. Si admitimos que estamos en

el comienzo de una nueva organización industrial o "régimen competitivo", podemos preguntarnos cuáles serían las características de dicho régimen.

3.2. ¿Puede ser libre la próxima organización industrial?

Defenderemos la idea de que la innovación del Software Libre atañe al proceso de desarrollo de software. Proporciona a la industria dos 'herramientas' relacionadas: un sistema para producir lo que [32] ha denominado "bienes industriales públicos" y para organizar un desarrollo e implementación *normalizados*⁶ de los cuales la industria carecía. Esto debería hacer posible redefinir las relaciones de servicio y, de esta forma, hacer que la organización industrial evolucionase.

3.2.1. Producción libre: una forma de organizar la producción de bienes industriales públicos, respetando las 'normas'...

Más que productos de investigación públicos, los programas libres fueron, primero y ante todo, herramientas desarrolladas por usuarios expertos para satisfacer sus propias necesidades. La baja calidad de los paquetes de software cerrado y, especialmente, la dificultad de hacerlos evolucionar fue una de las razones fundamentales de la iniciativa de Richard Stallman⁷. Estos usuarios expertos están detrás de muchas iniciativas de desarrollo de software libre (entre las cuales están Linux, Apache o Samba) y las han mejorado. Hay que hacer notar que, en lo que se refiere a estos buques insignia de los programas software, este modelo de organización ha obtenido notables resultados en términos de calidad y rápidas mejoras⁸.

Esto se debe indudablemente a la libre disponibilidad del código fuente, que permite a los usuarios a menudo probar el software, estudiar su código y corregirlo si encuentran errores. A mayor número de colaboradores, mayor es la posibilidad de que uno de ellos encuentre un error y sepa cómo corregirlo. Pero también son programas libres las herramientas (lenguajes) y reglas de programación que hacen posible esta lectura. Todo ello contribuye a garantizar unos elevados umbrales mínimos de robustez para ese software. Otros programas libres ampliamente distribuidos son herramientas de desarrollo de programas (compiladores, tales como GCC C/C++, o entornos de desarrollo como Emacs o Eclipse). Las razones son dobles:

- son herramientas utilizadas por profesionales informáticos, que pueden y están interesados en desarrollar o adaptar sus herramientas de trabajo,
- son las primeras herramientas que se necesitan para desarrollar software y su eficiencia es muy importante para la eficiencia del programa. Por esta razón los primeros productos de la FSF fueron tales programa-

mas, y especialmente el compilador GCC. El trabajo cooperativo, el hecho de que el software sea a menudo una colección de proyectos a pequeña escala que evolucionan simultáneamente, requiere también que la interfaz de comunicación sea hecha pública y *normalizada*⁹. El código abierto facilita la comprobación de esta compatibilidad y, si es necesario, la modificación del software. Hay que destacar que, a fin de evitar que se reproduzcan las divergencias entre las diferentes versiones de Unix, las firmas informáticas han creado organizaciones que deben garantizar la compatibilidad de las distintas versiones y distribuciones de Linux. Deben también publicar recomendaciones técnicas sobre cómo programar las aplicaciones para que puedan funcionar con este sistema, en el mismo espíritu del estándar POSIX¹⁰.

El hecho de que las empresas usen programas libres puede verse como la creación de herramientas profesionales para coordinar colectivamente la creación de componentes y bloques de programas software que sean al mismo tiempo fiables y, especialmente, *'normalizados'*. Hasta el presente, esta base colectiva, normalizada, estaba ausente en la industria de la tecnología de la información [11].

Esta normalización de los componentes utilizados para construir «productos de masas a medida» ayuda a mejorar la calidad de la producción porque los servicios basados en ellos pueden ser de mejor calidad.

3.2.2. ... permitiendo el desarrollo de una industria de servicios más eficiente¹¹.

Para demostrar que se puede construir una industria de servicios más eficiente y duradera sobre productos de software libre, tenemos que analizar dos puntos: desde el punto de vista de la empresa, 1) que estas ofertas son más interesantes que las existentes y puede haber negocio en ella, 1.bis) que este negocio es financieramente sostenible; y desde una perspectiva global, 2) que a largo plazo proporciona suficientes incentivos a los actores para contribuir al desarrollo de tales bienes públicos para mantener el dinamismo de la innovación.

El negocio

Hay negocio basado en el software libre. Como en los programas 'privados'¹² clásicos, es necesario definir las necesidades, encontrar el software que las satisface, instalarlo y, a veces, adaptarlo desarrollando módulos complementarios. Una vez instalado, es necesario seguir su evolución (actualizaciones de seguridad, nuevas funcionalidades...). Hay que tener en cuenta que los usuarios --ya sean éstos empresas, instituciones o usuarios individuales-- no siempre son lo bastante competentes para evaluar, instalar o seguir la evolución de estos programas. Tampoco saben siempre cómo adaptarlos a sus propias necesidades. Todo

esto exige la presencia de especialistas de estos programas en la empresa, lo que no siempre es fácil: y la mayoría de los usuarios de negocios no los necesita a tiempo completo. Por esta razón algunos partidarios del software libre argumentan que "para ello hay que crear compañías y que esta actividad debería ser rentable" durante mucho tiempo [27].

Naturalmente, la inexistencia de gastos de licencias proporciona claramente una ventaja competitiva al software libre, pero esto por sí solo no justifica su adopción: esta solución debe demostrar que a largo plazo es más económica ofreciendo sin embargo los mismos niveles de calidad. Los fabricantes de soluciones privadas utilizan este indicador para defender sus ofertas¹³.

Consideremos las ventajas específicas del software libre. Ya hemos dicho que los programas libres más maduros eran de muy alta calidad. Esto facilita la relación entre los productores de una solución basada en software y los que la utilizan. Los productores pueden garantizar más fácilmente, mediante contrato, la fiabilidad de los programas libres que usan porque pueden evaluar su calidad gracias a las normas que han fijado durante la fase de desarrollo. Hay una red de asistencia disponible para ellos y ellos, por su parte, pueden también intervenir en estos programas software. Además, el hecho de que el código fuente del programa sea accesible y que la evolución de estos programas no esté controlada por una firma pueden dar seguridad a quien los adopta: la solución respeta y seguirá respetando los estándares. Por tanto seguirá siendo interoperable con los otros programas que utilice.

El que haya un fondo común de bloques software debería cambiar también la competencia entre las firmas de servicio hacia relaciones a largo plazo y el mantenimiento de los programas. Resultaría más difícil para ellos pretender que el mal funcionamiento de un programa que han instalado y parametrizado se debe a un error del programa. Esto puede animar a las empresas a mejorar los servicios al cliente y permitimos afirmar que, en este campo, las soluciones libres son competitivas.

Esta organización 'teórica' ¿proporciona modelos de negocio rentables a las empresas de software libre? Este es, indudablemente, el punto más delicado a defender hoy. Hay pocos ejemplos de empresas rentables y muchas, todavía, ni siquiera han equilibrado el balance. Sin embargo, podemos señalar los siguientes puntos:

■ Con respecto a los costes de producción, gracias a los módulos de construcción, el coste de desarrollo de software se reparte más ampliamente a lo largo del tiempo, pareciéndose más así a una estructura de producción de servicios en la que las funcionalidades inexistentes se desarrollan sólo cuando es ne-

cesario. La contribución de las empresas de servicios no se relaciona con la producción de la totalidad del software, sino con la de los componentes para clientes que prefieren software libre para no depender de su proveedor. Más aún, un componente que ha sido desarrollado para un cliente puede ser reutilizado para satisfacer las necesidades de otro cliente. Un "agujero de seguridad" que ha sido detectado con un cliente puede corregirse para todos los clientes de la firma. En consecuencia, las empresas monopolizan parte de las economías de escala generadas por el uso colectivo de un programa software. A cambio, garantizan la distribución de sus innovaciones y correcciones, que es uno de los roles tradicionales del editor de software. Pero tradicionalmente, éstos financian su actividad produciendo y vendiendo nuevas versiones del programa.

Puede decirse que las empresas de servicio que basan su oferta en programas libres proponen conocimiento libre 'codificado', es decir, software, para vender el conocimiento 'tácito' que poseen: la forma en que los programas funcionan internamente, la capacidad de sus desarrolladores para hacer contribuciones que funcionan, conseguir que quienes controlan la evolución del software acepten estas contribuciones, etc. Estas empresas son las mejor capacitadas para sacar ventajas de los beneficios ligados al aprendizaje generado por el desarrollo y mejora del software.

■ Debido a estos efectos de aprendizaje y a lo difícil que resulta difundir el conocimiento tácito que hay que dominar para seguir e influenciar la evolución de un programa libre, este papel estará inevitablemente limitado a un pequeño número de empresas. Éstas proporcionarán conjuntamente especialistas en software y los pondrán a disposición de las empresas cliente. Habrán creado marcas comerciales fuertes, reconocidas por el usuario, por desarrolladores de software y conocidas por otros clientes. Esto hará posible disminuir la presión de la competencia, asegurando así por tanto sus márgenes de beneficio. Si bien es difícil medir los incentivos para innovar, tal competencia debería también estimular a estos productores a contribuir al desarrollo del software que utilizan.

La contribución al desarrollo de software

Ante todo, es una forma de darse a conocer y de demostrar a los clientes su competencia como desarrolladores. Ya que cada cliente tiene distintas necesidades, es importante para las empresas dominar una cartera amplia de programas software, así como contribuir al desarrollo de programas estándar que se puedan usar en la mayoría de las ofertas. Deben ser capaces de presentar a sus clientes realizaciones que se relacionen con sus problemas: no es tanto cuestión de dominar productos técnicos como de ser capaces de seguir e incluso controlar su evolución, para garantizar al cliente que, a

largo plazo, el producto satisfará sus necesidades. Y es más fácil de seguir la evolución de estos programas si se toma parte en el proceso de innovación, como es más fácil comprender las innovaciones de otras personas [6].

En un mercado basado en el valor creciente de la experiencia técnica, esta actividad de contribución refuerza la imagen de una empresa respecto a su experiencia y capacidad para reaccionar, dos cualidades que le permiten destacar una oferta especial así como mejorar su reputación (por medio de la marca comercial) e incrementar sus márgenes. Por otra parte, esto reforzará, una vez más, la tendencia a centrarse en actividades específicas ya que es necesario reducir los costes de investigación y, por tanto, incrementar el número de proyectos y clientes.

Una fuente más importante de innovación debería proceder de los usuarios. Ya que es importante conseguir que las modificaciones de programa sean incluidas en la versión oficial (para no tener que rehacerlas para cada nueva versión del programa) la mayor parte de las nuevas funcionalidades desarrolladas por o para un usuario deberían ser redistribuidas a todos. De paso, esto también proporciona incentivos a las compañías de servicios para participar en el desarrollo del software más evolucionado. Si quieren ser capaces de proponer accesorios para sus clientes, tienen que ser conocidos como colaboradores 'autorizados'¹⁴.

4. Conclusión: seleccionar el panorama económico adecuado

Si bien el movimiento del software libre parece ser la próxima etapa de una tendencia histórica y se puede describir el modelo económico global, está bastante claro que los modelos de negocio que deberían aparecer y estructurar este nuevo período no están bien definidos todavía. Todo ello refuerza la necesidad de más análisis de estos modelos, un análisis ya iniciado por [8][23], pero tenemos que centrarnos en la relación productor-comunidad y en las ventajas competitivas de gestionar un proyecto libre. Esto significa también comprender mejor cómo funciona(n) la(s) organización(es) de producción libre(s), el incentivo de los desarrolladores para participar en esta producción y medir la productividad de la organización libre.

Ésta es la agenda de investigación del proyecto europeo CALIBRE, <<http://www.calibre.ie>>.

Agradecimientos

Este trabajo ha sido financiado por la RNTL (*Réseau National des Technologies Logicielles*, Red Nacional de Tecnologías Software, <<http://www.telecom.gouv.fr/mtl/>>). El informe final está disponible en: <http://www-eco.enst-bretagne.fr/Etudes_projets/RNTL/rapport_final/>.

Referencias

- [1] **W. B. Arthur.** Self-reinforcing mechanisms in economics. En **P. W. Anderson, K. J. Arrow, and D. Pines**, editors, *The Economy as an Evolving Complex System*. SFI Studies in the Sciences of Complexity, Addison-Wesley Publishing Company, Redwood City C.A, 1998.
- [2] **W. B. Arthur.** Competing technologies, increasing returns and lock-in by historical events: The dynamics of allocations under increasing returns to scale. *Economic Journal*, 99: 116-131, 1999. <http://www.santafe.edu/arthur/Papers/Pdf_files/EJ.pdf>.
- [3] **C. Y. Baldwin y K. B. Clark.** *The architecture of cooperation: How code architecture mitigates free riding in the open source development model*. Harvard Business School, 43 páginas, 2003. <<http://open-source.mit.edu/papers/baldwinclark.pdf>>.
- [4] **J. Bessen.** Open source software: Free provision of complex public goods. *Research on Innovation*, 2002. <<http://www.researchoninnovation.org/online.htm#oss>>.
- [5] **P. Breton.** *Une histoire de l'informatique*. Point Sciences, Le Seuil, Paris, 1990.
- [6] **W. M. Cohen y D. A. Levinthal.** Innovation and learning: The two faces of r&d. *Economic Journal*, 99: 569-596, 1989.
- [7] **M. Coris.** Free software service companies: the emergence of an alternative production system within the software industry? En [23], pp. 81-98, 2002.
- [8] **L. Dahlander.** Appropriating returns from open innovation processes: A multiple case study of small firms in open source software. School of Technology Management and Economics, Chalmers University of Technology, 24 páginas, 2004. <<http://open-source.mit.edu/papers/dahlander.pdf>>.
- [9] **J. De Bandt.** *Services aux entreprises: informations, produits, richesses*. Economica, Paris, 1995.
- [10] **M. Delapierre, L.-A. Gerard-Varet, y J.-B. Zimmermann.** Choix publics et normalisation des réseaux informatiques. Technical report, Rapport BNL, diciembre 1980.
- [11] **G. Dréan.** *L'industrie informatique, structure, économie, perspectives*. Masson, Paris, 1996.
- [12] **J. Gadray.** La caractérisation des biens et des services, d'adam smith à peter hill: une approche alternative. Technical report, IFRESI, Lille. document de travail, 1998.
- [13] **C. Genthon.** *Croissance et crise de l'industrie informatique mondiale*. Syros, Paris, 1995.
- [14] **C. Genthon.** Le cas Sun Microsystems. ENST Bretagne, 2000. <<http://www-eco.enst-bretagne.fr/Enseignement/2A/1999-2000/EST201/sun/sun00.htm>>, material de curso.
- [15] **C. Genthon.** Le libre et l'industrie des services et logiciels informatiques. RNTL, 2001. <http://www-eco.enst-bretagne.fr/Etudes_projets/RNTL/workshop1/genthon.pdf>, *workshop*.
- [16] **L.-A. Gérard-Varet y J.-B. Zimmermann.** Concept de produit informatique et comportement des agents de l'industrie. En el coloquio "Structures économiques et économétrie", mayo 1985.
- [17] **F. Horn.** *L'économie du logiciel. Tome 1: De l'économie de l'informatique à l'économie du logiciel. Tome 2: De l'économie du logiciel à la socio-économie des "mondes de production" des logiciels*. PhD, Université de Lille I, mention: économie industrielle, 570 páginas, 2000. <http://www-eco.enst-bretagne.fr/Etudes_projets/RNTL/documents_universitaires.html>.
- [18] **F. Horn.** Company strategies for the freeing of a software source code: opportunities and difficulties. En [23], pp. 99-122, 2002.
- [19] **F. Horn.** *L'économie des logiciels*. Repères, La Découverte, 2004.
- [20] **N. Jullien.** Linux: la convergence du monde Unix et du monde PC. *Terminal*, 80/81: 43-70. Número especial, *Le logiciel libre*, 1999.
- [21] **N. Jullien.** *Impact du logiciel libre sur l'industrie*

informatique. PhD, Université de Bretagne Occidentale / ENST Bretagne, mention: sciences économiques, 307 páginas, noviembre 2001. <http://www-eco.enst-bretagne.fr/Etudes_projets/RNTL/documents_universitaires.html>.

[22] **N. Jullien.** Le marché francophone du logiciel libre. *Systèmes d'Information et Management*, 8 (1): 77-100, 2003.

[23] **N. Jullien, M. Clément-Fontaine, y J.-M. Dalle.** New economic models, new software industry economy. Technical Report, RNTL (French National Network for Software Technologies) project, 202 páginas, 2002. <http://www-eco.enst-bretagne.fr/Etudes_projets/RNTL/>.

[24] **K. Lakhani y E. von Hippel.** How open source software works: Free user to user assistance. *Research Policy*, 32: 923-943, 2003. <<http://opensource.mit.edu/papers/lakhanivonhippelusersupport.pdf>>.

[25] **D. C. Mowery**, editor. *The International Computer Software Industry, A comparative Study of Industry Evolution and Structure*. Oxford University Press, 1996.

[26] **L. Muselli.** Licenses: strategic tools for software publishers? En [23], pp. 129-145, 2002.

[27] **J. Ousterhout.** Free software needs profit. *Communications of the ACM*, 42 (4): 44-45, abril 1999.

[28] **E. S. Raymond.** The Cathedral and the Bazaar, 1998. <<http://www.tuxedo.org/~esr/writings/cathedral-bazaar/>>.

[29] **E. S. Raymond.** Homesteading the Noosphere, 1998.

<<http://www.tuxedo.org/~esr/writings/homesteading/>>.

[30] **E. S. Raymond.** *The Cathedral & the Bazaar; Musing on Linux and Open Source by Accidental Revolutionary*. O'Reilly, Sebastopol, California, 1999.

[31] **G. B. Richardson.** Economic analysis, public policy and the software industry. En *The Economics of Imperfect Knowledge - Collected papers of G.B. Richardson*, volumen 97-4. Edward Elgar, DRUID Working Paper, abril 1997.

[32] **P. Romer.** The economics of new ideas and new goods. Annual Conference on Development Economics, 1992, Banque Mondiale, Banque Mondiale, Washington D. C., 1993.

[33] **C. Tzu-Ying y L. Jen-Fang.** A comparative study of online user communities involvement in product innovation and development. National Cheng Chi University of Technology and Innovation Management, Taiwan, 29 páginas, 2004. <<http://opensource.mit.edu/papers/chanlee.pdf>>.

[34] **J.-B. Zimmermann.** Le concept de grappes technologiques. Un cadre formel. *Revue économique*, 46 (5): 1263-1295, septiembre 1995.

[35] **J.-B. Zimmermann.** Un régime de droit d'auteur: la propriété intellectuelle du logiciel. *Réseaux*, 88-89: 91-106, 1998.

[36] **J.-B. Zimmermann.** Logiciel et propriété intellectuelle: du copyright au copyleft. *Terminal*, 80/81: 95-116. Special Issue, *Le logiciel libre*, 1999.

Notas

1 Hay que recordar que en inglés la palabra *free* significa tanto "libre" como "gratis", lo que a veces provoca malentendidos, y que el término *Open Source* puede resultar ambiguo. En cualquier caso, nos estamos refiriendo a aquellos programas cuyo código fuente puede ser accedido por el usuario, estándole permitido modificarlo y redistribuir el programa y sus modificaciones.

2 Nuestra idea de la historia de la TI sigue en gran medida los trabajos de [5][13][11]. El análisis de la organización de la industria de TI está basado también en [16], presentado en [10]. Por último, nuestro análisis de la economía e industria del software debe mucho a [25][19], cuyos trabajos son, en mi opinión,

una referencia en este campo. Recomendamos su lectura a quienes quieran saber más sobre estos temas.

3 Definición tomada de <<http://www.investordictionary.com/definition/public+good.aspx>>.

4 Al "abrir" el hardware del PC, IBM permitió a los competidores producir máquinas similares y a los fabricantes de componentes distribuir sus productos a diferentes productores. Esto ha hecho aumentar la competencia, en términos de precio, pero también en términos de eficiencia de los componentes. A cambio, la distribución de PC ha permitido a los productores aumentar el volumen de componentes vendidos y por tanto disminuir su precio, ya que esta producción tiene, esencialmente, costes fijos (la I+D y la construcción de la capacidad de producción).

5 Esto no es nuevo; véase, por ejemplo, la encuesta sobre satisfacción realizada durante tres años por "01 informatique", revista semanal para las grandes organizaciones francesas (núm. 1521 en 1998, 1566 en 1999 y 1612 en 2000). Hay otras encuestas aún más duras, como las que aparecen en [9][11, pp. 276 y ss.].

6 Entendido en el sentido en que lo define la teoría económica, es decir, un sistema abierto que permite a los actores negociar las características de un componente/producto/interfaz y garantizar que el diseño del producto respetará estas características.

7 Stallman 'inventó' el concepto de software libre con la creación de la licencia GNU/GPL y la *Free Software Foundation*, la organización que lo produce; véase <<http://www.fsf.org/gnu/thegnuproject.html>>. Véase también <<http://www.gnu.org/prep/standards.html>> para las recomendaciones técnicas sobre cómo programar software libre.

8 Sobre el modo en que se estructura el desarrollo libre, además de [28][29][30], nos podemos referir también a [24][21]. Véase [33] para una encuesta y un análisis de la eficacia de la implicación de la comunidad de usuarios *on-line* [4] y [3] para un análisis teórico del impacto de la arquitectura de código abierto en la eficiencia del desarrollo libre. El último argumenta que el desarrollo libre puede ser visto como una nueva 'institución' (pp. 35 y ss.). En cuanto a los tests de rendimiento, nos podemos referir a <<http://gnet.dhs.org/stories/bloor.php3>> para sistemas operativos. Los resultados de numerosas comparaciones están disponibles en <<http://www.spec.org>> y <<http://www.kegel.com/nt-linux-benchmarks.html>> (este último se refiere principalmente a NT/Linux).

9 En el sentido de que respetan los formatos públicos cuya evolución se decide colectivamente.

10 Es el *Free Standard Group*, <<http://www.freestandards.org/>>. Entre otros, son miembros de este comité Red Hat, Mandriva, SuSE/Novell, VA Software, Turbo Linux, y también IBM, SUN o Dell, etc.

11 Este análisis teórico se basa en un estudio de estrategias comerciales de compañías, en Francia, que afirman vender servicios o productos basados en software libre [22].

12 Preferimos este término al de 'propietario' ya que todos los programas tienen un poseedor. Aquí 'privado' significa que el poseedor no comparte el programa con otros, como en la distribución clásica de software.

13 A esto se le denomina TCO (*Total Cost of Ownership*). Actualmente, Microsoft defiende la idea de que aunque sus programas son más caros que los libres, tienen un costo total (TCO) menor, ya que es más fácil encontrar empresas que los instalen, su evolución está garantizada, controlada por una marca, etc.

14 Empresas como Easter Eggs en Francia cobran hoy por hacer modificaciones en programas libres que son luego aceptadas e integradas en la distribución oficial.