Marketing de Software, análisis desde la administración de sistemas.

Ing. Juan Manuel Calvo
Director del Centro de Cómputos
Universidad del CEMA

marzo del 2001

Introducción

Discutiremos algunas de las técnicas de marketing utilizadas por los productores de software con el propósito de obtener el mayor beneficio de la venta de sus productos y como afectan estas técnicas a los usuarios de aplicaciones de software.

Código fuente y objeto

Las computadoras contienen dos elementos: el hardware y el software. El hardware son las partes de una computadora que son visibles, que se pueden tocar. El software en cambio es el juego de instrucciones que controla el hardware, en esencia el software no es nada mas que que una secuencia de unos y ceros que controla el envío de electricidad al hardware, piense que un uno enciende una llave y un cero la cierra, el hardware solo responde a secuencias de unos y ceros. Esas secuencias de unos y ceros se las llama forma ejecutable de un programa o código objeto.

El código objeto no se escribe directamente ya que sería extremadamente complejo hacerlo, se sigue un modelo donde las ideas o especificaciones de la aplicación a implementar son expresadas en un lenguaje simbólico llamado lenguaje de programación o código fuente y luego este código fuente es transformado en código objeto por un programa compilador.

Licenciamiento

Antes de 1980 las computadoras eran muy caras y su uso estaba restringido al gobierno, las universidades y las grandes corporaciones. El software estaba desarrollado a medida para las necesidades específicas de cada lugar. Durante esta época, los programadores compartían los avances y las soluciones que encontraban. Era más rentable económicamente compartir el resultado de las investigaciones y beneficiarse con no tener que desarrollar todas las soluciones independientemente que explotar los beneficios de la propiedad intelectual del software. En

estas épocas el mercado de software consistía en proveer servicios de programación a los compradores más que entregar un producto pre-hecho a usuarios finales.

El licenciamiento no existía, los programadores eran contratados para realizar su trabajo y el empleador utilizaba el programa que se había desarrollado, no existía el mercado para vender copias del mismo software[1].

Cuando aparecieron las computadoras personales, a pricipios de los 80, la computación se hizo alcanzable económicamente para un número cada vez mayor de personas y el número de computadoras que corrian la misma aplicación de software aumentó explosivamente. Había nacido el mercado masivo para el sofware.

Una de las consecuencias de las políticas de licenciamiento del software es que el usuario no adquiere la propiedad del producto, cuando adquiere un producto de software licenciado, solo adquiere un permiso para utilizar este producto, en las condiciones que especifica el contrato que define la licencia.

Los proveedores de software fueron adoptando políticas para proteger su cuota de mercado y sus beneficios. Manteniendo secreto el código de un programa una empresa puede prevenir que otros lo imiten o desarrollen productos competitivos dado los altos costos involucrados con la creación de software.

Luego no solo se mantuvo en secreto el código del progama sinó las interfases del programa con el mundo exterior, de esta manera los fabricantes de software pudieron restringir la comunicación entre aplicaciones a las que ellos realizaban.

En los primeros tiempos el gasto en computación era principalmente de hardware, el software era gratuito o se incluia en el precio del hardware. Con la evolución de la técnica, los costos del harware han estado disminuyendo en forma continua, mientras que los costos del software, de ser insignificantes, se han transformado en la parte más importante del gasto en tecnología informática. Es una idea extendida en muchas personas que en computación es más importante el hardware y que el software.

La publicidad de los fabricantes de software se enfoca principalmente en el número de características que tienen sus productos, lo avanzadas tecnologicamente y la facilidad con la que se pueden comenzar a utilizar. Si bien estas caracteristicas son importantes para el usuario final generalmente interesan poco al administrador de sistemas que se tiene que encargar de instalar, configurar y mantener en funcionamiento las aplicaciones.

No todo el software se distribuye manteniendo secreto el código fuente, las aplicaciones “open source” incluyen todo el código utilizado para generarlas y generalmente se pueden obtener gratuitamente.

Como los fabricantes defienden su mercado

El costo de producción de un paquete de sofware es insignificante, el mayor costo está en el desarrollo , este costo de desarrollo se amortiza por la venta de una gran cantidad de unidades del paquete. El mercado de software tiene tendencia al monopolio, el fabricante que más vende es el que obtiene las mayores ganancias, y dispone de dinero para invertir en desarrollo, marketing, etc. Este proceso de acumulación de ganancias puede ser muy rápido, un producto exitoso genera importantes ganancias al fabricante ya que el costo marginal de un paquete de software

es muy bajo. En cambio, cuando un producto de sofware pierde mercado y es desplazado por otro que lo sustituye los ingresos por ventas del fabricante sustituido disminuyen abruptamente. Por esta razón una vez que un fabricante ha logrado un producto exitoso trata de defender la porción de mercado obtenido a toda costa.

Las tecnicas que habitualmente se utilizan para defender el mercado pueden resumirse en: cambios permanentes, complejidad inecesaria y el uso de propiedad intelectual.

Cambios permanentes

Los fabricantes estan forzados a liberar al mercado periódicamente nuevas versiones de las aplicaciones, con el fin de que la competencia no pueda imitarlas. Es muy habitual encontranos que en cada nueva versión, se realicen cambios en los formatos de los archivos y los protocolos de comunicación, con el fin de dificultar que la competencia pueda desarrollar una aplicación similar.

Un ejemplo de esto fué Real Audio, esta empresa logró mantener su liderazgo en el streaming de audio y video en Internet cambiando el protocolo utilizado en la transmisión en cada nueva versión. Ya que el protocolo era secreto, no había ninguna empresa que pudiera desarrollar una aplicación compatible en el breve lapso de tiempo que transcurria entre versión y versión, la única manera de hacer una versión compatible es la ingeniería inversa, que es un procedimiento cuyos resultados no son predecibles.

Microsoft ha aplicado esta técnica en los protocolos de red que usan las máquinas para comunicarse entre si, cada versión de Windows ha traido un protocolo que es parcialmente incompatible con otros productos desarrollados independientemente.

En cambio los protocolos de internet que se establecieron en la década del 70 han permanecido estables, solo ha habido muy pequeñas modficaciones, permitiendo el desarrollo de múltiples productos que los emplean[2].

Relacionada con los cambios permanentes es la técnica de vincular las correciones de errores de programación (bugs) con nuevas características y distribuirlas como un paquete indivisible, es frecuente que las correcciones de bugs no se distribuyan aisladas, corregiendo un error o una falla de seguridad recien descubierta, sinó junto a nuevas características, las cuales a su vez introducen nuevos bugs, en un ciclo que se repite periodicamente. En cambio en las aplicaciones de código abierto casi siempre las correcciones de bugs están totalmente separadas de las nuevas características, y es posible solucionar un problema generado por un bug sin modificar el funcionamiento de un programa.

En los productos de código abierto las correcciones de bugs y las nuevas características se distribuyen por separado facilitando enormemente la tarea de mantenerlas actualizadas, como ejemplo el sistema operativo Linux las versiones cuyo segundo número de versión es par son las versiones estables y las diferentes revisiones solo corrigen problemas, en cambio las nuevas características solo se incluyen en las versiones de desarrollo donde el segundo número de versión es impar. De esta manera se separan claramente ambos tipos de actualización.

Complejidad innecesaria

La realidad nos dice que una aplicación simple y confiable puede ser copiada con facilidad por la competencia, en cambio si hacemos una aplicación de software extremadamente compleja, dificultamos el desarrollo de aplicaciones compatibles, esto induce a los fabricantes a desarrollar protocolos o formatos de archivos extremadamente complejos. El ejemplo mas conocido el Win32, que consiste en la interfase que conecta a las aplicaciones con el sistema operativo Microsoft Windows, la cual ha ido creciendo con cada nueva versión y es de una complejidad increíble.

Es interesante comparar la complejidad de algunas aplicaciones de código abierto con aplicaciones comerciales similares, encontrándose que las aplicaciones de código abierto generalmente tienen un código cuya longitud es de un órden de magnitud menor: Windows 2000 tiene unos 30 millones de líneas de programa, linux 2.4.x alrederdor de 3,5 millones, resultados similares obtendremos si comparamos VisualC++ con gcc o Microsoft Exchange con sendmail.

Uso de propiedad intelectual

Para el usuario que el fabricante se proteja con propiedad intelectual (patentes, trade secrets, non disclosure agreements) siempre es perjudicial ya que dificulta el proceso de migrar a otra aplicación, los datos se guardan en formatos que solo son entendidos por la aplicación que lo generó.

Para impedir el desarrollo de un producto compatible. Se ven obligados además a efectuar cambios frecuentemente para que las técnicas utilizadas no sean copiadas. Patentar un protocolo de comunicaciones impide a la competencia desarrollar un producto compatible.

Algo que está relacionado con la propiedad intelectual es la publicación de las especificaciones, tomemos como ejemplo los casos de los formatos .doc de los documentos de Microsoft Word y .pdf portable document format, ambos son formatos complejos que han sido publicados por sus fabricantes pero en condiciones muy distintas. Mientras el pdf ha sido documentado con exactitud[2] y ha permitido desarrollar aplicaciones compatibles (ghostscript[4], xpdf[5], pdflib[6]) con la aplicación del fabricante (adobe acrobat[7]), el formato doc solo es compatible con el Word de Microsoft, las otras implementaciones (starOffice[10], wvWare[8], wine[9]) no son compatibles en su totalidad y han tenido que recurrir a la ingeniería inversa para implementar partes que no están definidas en la especificación.

Es importante destacar que las aplicaciones de código abierto no pueden tener características propietarias ya que el código fuente está disponible.

Qué características debemos buscar en el software que adquirimos?

Las principales carácterísticas que debemos buscar en los productos de software que adoptamos son: archivos de formatos standard, esto nos facilitará intercambiar datos con otras aplicaciones o migrarlos, actualizaciones para resolver bugs o fallas de seguridad que no requieran instalar nuevas características, no uso de propiedad intelectual para proteger la aplicación, esto es de mayor importancia para el formato en que la aplicación guarda los datos.

Referencias

[1] Facilitating Collaborative Software Development: The Enforceability of Mass-Market Public Software Licenses Daniel B. Ravicher Virginia Journal of Law and Technology http://www.vjolt.net/vol5/issue3/v5i3a11-Ravicher.html.

[2] Internet Official Protocol Standards, RFC2800, Internet Engineering Task Force, http://www.faqs.org/rfcs/rfc2800.html.

[3] PDF Reference Manual http://partners.adobe.com/asn/developer/acrosdk/docs/pdfspec.pdf.

[4] Ghostscript, Ghostview and GSview, http://www.cs.wisc.edu/~ghost/.

[5] Xpdf, a PDF viewer for X, http://www.foolabs.com/xpdf/.

[6] Pdflib, a library for generating PDF on the fly, http://www.pdflib.com/pdflib/index.html.

[7] Adobe acrobat, http://www.adobe.com/products/acrobat/readstep.html.

[8] wvWare, http://www.wvware.com/wvWare.html.

[9] Wine, http://www.winehq.com.

[10] Star Office, http://www.sun.com/products/staroffice/.