¿Can we apply *Free Software* production model to learning contents?

Sergio Monge Benito

Universidad del País Vasco – Euskal Herriko Unibertsitatea.
Barrio Sarriena, s/n. 48940 Leioa (Bizkaia), Spain
tallerd3@euskalnet.net

Abstract. Free software movement has released into public domain high quality software at no cost. Software industry has been surprised by their decentralized production model, based in voluntary contributions from all-around the world volunteers, but now it is slowly accepting some of their approaches. In order to write my thesis about TIC application to secondary education in the Basque Country, I have researched coincidences and differences between free software production model and an hypothetic learning content production model based on the it.

¿Es aplicable el modelo de producción del *software* libre a contenidos educativos?

Sergio Monge Benito

Universidad del País Vasco – Euskal Herriko Unibertsitatea.
Barrio Sarriena, s/n. 48940 Leioa (Bizkaia).
tallerd3@euskalnet.net

Resumen: El movimiento del *software* libre ha logrado poner a disposición del gran público gran cantidad de *software* gratuito de calidad. Su modelo descentralizado de producción mediante la suma de aportaciones de voluntarios de todo el planeta ha sorprendido a la gran industria del *software*, que poco a poco adopta algunos de sus planteamientos. Para escribir mi tesis doctoral sobre "La escuela ante el cambio tecnológico", he indagado en las coincidencias y diferencias que existen entre el modelo de producción del *software* libre y un hipotético modelo de producción de contenidos educativos que trate de emularlo.

0.- Introducción

El origen de esta ponencia está en el caso de Berrikuntza (www.berrikuntza.net), un portal de Internet lanzado en el 2001 con la intención de servir de herramienta de colaboración entre docentes de primaria y secundaria de la CAV[1]. Fue diseñado por los responsables del Programa TIC [1] del Gobierno Vasco con la intención de aprovechar las ventajas que ofrecía Internet para la colaboración. Entre otras cosas, Berrikutza debía permitir a los docentes intercambiar contenidos didácticos. El portal fue un fracaso de participación en cuanto a intercambio de contenidos didácticos, si bien sirvió para ofrecer otros servicios a los docentes (como es el correo electrónico). En una serie de entrevistas realizadas a expertos, estos destacaron que no existía voluntad de colaborar entre los profesores. ¿Cómo es que una estrategia de colaboración telemática que se había demostrado útil en otros sectores no servía para la educación?

1.- El caso del *software* libre

En 1984, Richard Stallman abandonó su puesto de profesor en el MIT[2] para dedicarse por completo a la creación de un sistema operativo que concediera las cuatro libertades que él consideraba fundamentales [2]: libertad de uso, libertad de estudio y adap-

[1] Comunidad Autónoma Vasca.
[2] *Massachusetts Institute of Technology*.

tación (lo que exige tener el código fuente), libertad de redistribución y libertad para mejorar el programa y publicar esas mejoras. Un año después, fundó la FSF[3] e inició el proyecto GNU [3], una organización sin ánimo de lucro con el objetivo de producir *software* basado en estas premisas. A lo largo de estos veinte últimos años, cientos de grupos de programadores de todo el globo han colaborado para producir *software* de calidad y de libre distribución. El caso más conocido es el de *Linux*, el *kernel* (núcleo) del sistema operativo que vino a completar los esfuerzos de Stallman y la FSF. En la actualidad, podemos obtener alternativas de calidad y libres de costes de distribución de la mayoría de programas de uso común (sistema operativo, *suite* ofimática, retoque fotográfico, etc.).

El modo de trabajar de los programadores de *software* libre no ha pasado inadvertido para su sector, que poco a poco van examinado con interés su modo de trabajo y las posibilidades de negocio [4] que ofrece. Voluntarios de todo el mundo colaboran en un mismo proyecto y, aunque normalmente hay un líder por grupo de trabajo, la organización se parece más a la de un desordenado bazar [5] que a un organigrama jerárquico. El código fuente es accesible para todo el mundo y todo el mundo puede proponer mejoras. Las decisiones se toman normalmente con una mezcla de democracia y meritocracia, es decir, los que más aportan al proyecto tienen mayor poder de decisión sobre qué se incluye en él. Puesto que el código fuente es de acceso libre, cientos de usuarios/programadores realizan las pruebas del programa, lo que garantiza una facilidad para detectar errores que difícilmente puede permitirse el desarrollo de un programa propietario.

El movimiento que se inició con una opción ética [6] ha demostrado que sus modos de trabajo son tremendamente competitivos. Esto ha hecho que numerosas personas se hayan acercado examinar ese modelo de producción de bienes inmateriales (informacionales) por su eficacia y no por su opción ética. En Junio, por ejemplo, *The economist*[7] especulaba sobre las posibilidades que tendría el modelo de trabajo de código abierto para la investigación de nuevos fármacos.

Preparando mi tesis doctoral sobre el uso de las TIC[4] en la enseñanza secundaria de la CAV, me he encontrado con una preocupante escasez de contenidos diseñados especialmente para estas nuevas tecnologías. La pregunta que me hago es ¿podría el modelo de producción del *software* libre servir para corregir esa carencia de contenidos educativos adecuados a las TIC? ¿Podrían los propios profesores de secundaria trabajar juntos mediante las herramientas de las redes telemáticas para producir contenidos apropiados? Hay una serie de similitudes y diferencias entre el modelo de producción del *software* libre y un hipotético modelo de colaboración entre profesores de secundaria.

2.- Similitudes

La mayor parte de las similitudes provienen del objeto de producción. En ambos casos (*software* y contenidos educativos) estamos hablando de un bien inmaterial, de

[3] *Free Software Foundation*. Organización sin ánimo de lucro dedicada a la producción de *software* libre.
[4] Tecnologías de la Información y la Comunicación.

información, de conocimiento. Por tanto, la estrategia de compartir la información y hacerla lo más trasparente posible al grupo de producción es ventajosa en ambos casos. Todo el mundo estaría en posición de aportar lo que considerase oportuno y de sugerir cambios a la comunidad. Esta podría tomar las decisiones de incluir o no incluir determinado objeto de aprendizaje de manera democrática. Además, no serian necesario, como en el caso del *software*, que si determinado objeto de aprendizaje no se incluye en la versión *oficial* del contenido esos esfuerzos se pierdan, ya que se podría almacenar como un contenido opcional que los profesores pudieran decidir incluir o no en sus clases.

Además, el *software* y los contenidos educativos tienen otro elemento en común. Una de las razones por las cuales el modelo descentralizado de producción de los proyectos de código abierto funciona tan bien es que el *software* es modular. Es decir, se puede construir por partes, con pequeños grupos dedicados a distintas pequeñas partes del mismo. Los contenidos educativos también son modulares, separados muchas veces en temas, capítulos, epígrafes, etc. En la actualidad, el mundo del *eLearning* maneja el concepto de LO (L*earning Object* u objeto de aprendizaje) como unidad mínima de los contenidos educativos, aunque aún existe bastante controversia acerca de su definición [8, 9]. Los profesores podrían colaborar en base a sus áreas de especialidad en diferentes partes de un mismo objeto de aprendizaje.

3.- Diferencias.

La mayoría de las veces que se propone aplicar el modelo de producción del *software* libre para solucionar determinado problema se tiende a ser demasiado optimista a ese respecto y se olvida analizar las diferencias fundamentales que podrían hacer ese modelo no aplicable. En este caso, es decir, para la generación de contenidos mediante la colaboración entre profesores de secundaria, hay muchos obstáculos a superar antes de que el modelo sea aplicable. Citamos algunos de ellos:

3.1.- La comunidad.

En la producción de *software* libre, el concepto de comunidad tiene una importancia capital. Cuando se le pregunta a uno de estos voluntarios por sus motivaciones para programar *software* libre [10], dos motivaciones importantes son "compartir conocimiento y habilidades" (50%) y "participar en una nueva forma de cooperación" (alrededor de un tercio). La comunidad es el motor impulsa a los programadores a entregar su tiempo al proyecto y es a la vez la receptora de los beneficios del mismo. Un programador prolífico y hábil recibe el reconocimiento de su comunidad por ello. Esta comunidad se forja a nivel internacional y se relaciona a través de Internet con una complicada serie de costumbres y tabúes semiinconscientes que Eric S. Raymond ana liza en su artículo "Cultivando la noosfera [11]."

En el caso de los profesores de secundaria, no es habitual que su "comunidad" se extienda más allá del cetro en el que trabajan. Los lazos que le unen al resto de la comunidad docente (a nivel nacional, o autonómico) son más bien difusos. ¿Puede

desarrollarse una tarea colaborativa de producción de contenidos sin una comunidad que sancione y premie determinadas actitudes? Dado el carácter voluntario de las aportaciones en este tipo de escenarios de colaboración, la ausencia de una comunidad similar la del *software* libre que recompense las colaboraciones sería un primer obstáculo.

3.2.- Manejo de las herramientas tecnológicas.

Esta es una diferencia fundamental entre la comunidad *hacker*[5] y los profesores de secundaria. Lo es en varios niveles. En primer lugar y enlazando con lo anterior, la familiaridad en el manejo de las TIC es necesaria para forjar una comunidad virtual. Las comunidades virtuales a nivel mundial o incluso local tienen sus propias normas de comportamiento. Es una socialización secundaria en toda regla que requiere como elemento previo una cierta facilidad, naturalidad y costumbre a la hora de manejar las herramientas telemáticas. Difícilmente va entrar en esa dinámica de comunidad aquel que revise su correo solamente una vez al mes o quien no haya oído en la vida hablar de una lista de correo. Los programadores de *software* libre tienen, por profesión o dedicación, el dominio necesario de habilidades tecnológicas (y sociales) para socializar en ese medio virtual.

En cambio, no todos los profesores de secundaria tienen ese mismo gusto por las tecnologías. En el caso de la CAV, por ejemplo, la edad media del profesorado ronda los 50 años y muchos profesores, aún habiendo recibido formación tecnológica para manejar esas herramientas de comunicación que ofrece Internet, no las emplean en su vida cotidiana. Esto dificulta la creación de una comunidad virtual de docentes a nivel autonómico (o nacional), puesto que no es suficiente con que dispongamos redes tecnológicas de comunicación. Lo más importante es que dispongamos de redes humanas que estén dispuestas a colaborar.

Por otra parte, ese manejo de las herramientas tecnológicas también es necesario para la producción de contenidos adecuados a las TIC. A un programador se le supone su dominio de lenguajes informáticos y, del mismo modo, a un profesor se le supone su capacidad pedagógica. Pero un profesor puede que no disponga de conocimientos de lenguaje audiovisual para producir documentos en video que luego pueda utilizar en el aula. O puede que sea capaz de concebir un objeto de aprendizaje electrónico pero no sea capaz de programarlo. ¿Deben los profesores desarrollar esas habilidades para producir sus propios contenidos? ¿O debemos buscar nuevas estrategias de colaboración que incluyan a terceras partes? Desde luego, si quisiéramos utilizar el modelo de producción del *software* libre tal cual sería necesario que los profesores aportaran sus propios contenidos, diseñados y producidos por ellos mismos.

---

[5] Los medios de comunicación han dado un sentido peyorativo a esta palabra que no lo tenía originalmente. *Hackers* es la manera en que muchas comunidades de desarrolladores de *software* libre se denominan a sí mismas.

3.3.- Estándares.

Si examinamos los proyectos de *software* libre más conocidos [12], descubriremos que la mayoría del trabajo está programado en el mismo lenguaje. El 94,56% de *Linux* está programado en C, así como el 92% de Apache, el 86% de GNOME, el 62% de Red Hat Linux, o el 63% de Debian Linux. En los casos que no se usa C, se suele utilizar una evolución de este, C++ (82% de KDE, 54% de Mozilla y 85% de *Open Office*). Hay otros proyectos que utilizan otros de lenguajes de programación (como Java, o Perl, etc.), pero lo que coincide en todos los casos es que el núcleo del proyecto, la mayor parte de las líneas de código que lo componen, están programadas normalmente en un mismo lenguaje. Además, hay un programa muy extendido (CVS) que utilizan casi todos los proyectos para controlar las versiones del programa y coordinar la programación colaborativa. Los programadores de *software* libre tienen sus propios estándares, lo que les permite trabajar unidos en una misma dirección.

En la educación tenemos un grave problema: no tenemos un estándar para producir contenidos de manera que estos sean reutilizables y fácilmente adaptables. Un contenido educativo puede tener numerosas formas: texto para ver en pantalla, texto para imprimir, imágenes, videos, documentos sonoros, ejercicios diseñados para el *learning-by-doing*,... Es cierto que desde el mundo del *eLearning*, organismos e iniciativas como IMS[6], ADL[7] o IEEE[8] están haciendo esfuerzos considerables encaminados a diseñar estos estándares. En la actualidad, todos esos esfuerzos parecen converger hacia SCORM (*Sharable Content Object Reference Model*). En la versión de enero de 2004, SCORM incluía tres libros de especificaciones: un modelo de agregación de contenidos (CAM[9]), una especificación (RTE[10]) para la interacción en tiempo de ejecución entre las aplicaciones clientes y el LMS (*Learning Management System*) y un libro adicional para examinar detenidamente la navegación y la secuenciación (SN[11]).

Los esfuerzos de todos estos organismos y el "nacimiento" de SCORM suponen un importante paso adelante hacia la búsqueda de un estándar que sea ampliamente compartido por aquellos que trabajen en el *eLearning*. Con el tiempo, un estándar de este tipo podría popularizarse en todos los niveles de la comunidad educativa. No obstante, en la actualidad aún existe una gran controversia [8, 13] en torno a aspectos tan básicos como la definición exacta de la unidad básica en la generación de contenidos didácticos: el objeto de aprendizaje o *learning object*. Si los más avanzados en el uso de estándares (aquellos que se dedican al *eLearning*), aún mantienen abierto el debate en torno a ellos, no podemos esperar que los profesores de enseñanza secundaria sean capaces de trabajar en un lenguaje común.

No me extenderé en este tema que, puesto que tiene una importancia capital en la reutilización de materiales didácticos [13], seguramente estará muy presente en el resto de las ponencias. Tan sólo debo apuntar que, mientras no se disponga de forma-

---

[6] IMS Global Learning Consortium, Inc. [http://www.imsglobal.org/].
[7] Advanced Distributed Learning [http://www.adlnet.org/] .
[8] Institute of Electrical and Electronics Engineers, Inc. [http://www.ieee.org/].
[9] Content Aggregation Model.
[10] Run-Time Environment.
[11] Sequencing and Navigation.

tos estándar para la generación de contenidos educativos ampliamente aceptados y utilizados dentro de la comunidad docente de educación obligatoria, la creación de comunidades virtuales que colaboren para producir contenidos encontrará numerosos problemas. Los contenidos serán difícilmente reutilizables y requerirán un gran esfuerzo de adaptación por parte de los usuarios finales. A medida que herramientas como SCORM se popularicen y trasciendan la a aquellos interesados en el *eLearning* para abarcar otros niveles de la educación, este obstáculo dejará de serlo. Mientras tanto, es un posible freno a tener en cuenta a la hora de iniciar cualquier proyecto.

3.4.- Voluntarios universitarios.

En la mayoría de los proyectos de *software* libre, la aportación del entorno universitario ha sido notable. El mismo Linus Torvalds, por ejemplo, inició el desarrollo de Linux en su época de estudiante en la Universidad de Helsinki. El movimiento del *software* libre se beneficia de la disponibilidad de tiempo y energía de los universitarios de todo el mundo. Aunque los programadores del mundo empresarial también han tenido una notable parte en el desarrollo de *software* libre, es muy probable que muchos proyectos jamás hubieran acometido de no ser por esa energía adicional aportada desde el mundo universitario.

Cuando trasladamos este mismo fenómeno al terreno de los contenidos educativos, lo primero que salta a la vista es que los docentes de los cuerpos de primaria y secundaria normalmente ya están cargados de trabajo en el desarrollo de sus funciones (de un modo análogo al que los programadores en empresas comerciales lo están). ¿Podría obtener la comunidad docente ese mismo apoyo universitario que los programadores de *software* libre? ¿De que manera? Y si no pueden, ¿sería factible que su "empresa" (la administración) invirtiera parte del tiempo de los docentes en la creación de contenidos?

3.5.- Masa crítica.

Como cada vez que se desea cambiar las actitudes de un grupo grande, la estrategia del *software* libre requirió que un grupo pequeño (aunque suficientemente importante) de programadores con una actitud consistente y un ideario concreto se lanzaran a programar. El gesto desinteresado de ofrecer gratuitamente el esfuerzo de su trabajo y la continuidad de esa actitud generosa no pasó desapercibido para el resto de la comunidad. Los programadores de *software* libre comenzaron a ganar colaboradores y su comunidad creció hasta los niveles actuales. Esa pequeño grupo sirvió de referencia para el resto de la comunidad y marcó el camino que debía seguirse.

Para generar una comunidad similar entre los docentes de primaria y secundaria, sería necesario que surgiera un grupo de voluntarios dentro de sus mismas filas dispuestos acometer un proyecto bajo las premisa de compartir con toda la comunidad educativa. Dicho grupo debería tener en cuenta las limitaciones de aplicabilidad (apuntadas en este texto) del modelo del *software* libre para poder sortear los inconvenientes que surgieran en su camino. Sólo si consiguen mantener su actitud generosa de manera consistente y sortear dichos inconvenientes, podremos averiguar si el

modelo del *software* libre puede duplicarse porque su forma de compartir es inherente a la conducta humana o si se ha tratado de una excepción casual. No obstante, dicha iniciativa se enfrenta a barreras culturales más rígidas que en el caso de la comunidad del *software* libre. Mientras que para los *hackers* compartir el *software* siempre había sido costumbre habitual hasta la irrupción del *software* propietario, entre los docentes tenemos ese viejo dicho de "cada maestrillo su librillo." ¿Puede la comunidad docente superar esa tradición individualista en los albores del tercer milenio?

4.- Conclusiones.

El modelo de producción del *software* libre ha demostrado su viabilidad y eficacia. Una estrategia basada en compartir ha demostrado ser mucho más competitiva la tendencia del mundo comercial a mantener secretos. Estos buenos resultados nos dan esperanzas de que este mismo modelo se pueda aplicar a los contenidos educativos. No obstante, en el momento actual existen numerosos inconvenientes que dificultan la adopción inmediata de ese modelo en la educación secundaria obligatoria. Estos obstáculos no son, en ningún caso, insalvables pero si constituyen un punto sobre el que debemos reflexionar. También debemos estar atentos al mundo del *eLearning* que, trabajando para resolver sus propios problemas, puede que termine ofreciendo a los docentes de secundaria las herramientas que necesitan para colaborar en la creación de contenidos a través de Internet.

5.- Referencias:

1. Gobierno Vasco: "Programas de innovación educativa 2000-2003." Dirección de innovación educativa. Vitoria-Gasteiz 2000. También disponible *online*. [Consultado 27/9/2004] http://www.berrikuntza.net/edukia/ikt/aurkezpena/programacmplt_ikt_es.html?etapa_id=1
2. *Free Software Foundation*: "La definición de *software* libre" Documento *online* (2001). [Consultado 20/6/2004] http://www.gnu.org/philosophy/free-sw.es.html
3. Stallman, R. M.: "El proyecto GNU.". Documento *online* (1998) [Consultado 20/6/2004] http://www.gnu.org/gnu/thegnuproject.es.html
4. Raymond, E. S.: "El caldero mágico." Documento *online* (2000) [Consultado 30/6/2004] http://www.alanta.info/MagicCauldron.html
5. Raymond, E. S.: "La catedral y el bazar." Documento *online* (1998) [Consultado 30/6/2004] http://sindominio.net/biblioweb/telematica/catedral.html
6. Stallman, R. M.: "Por qué el *software* no debe tener propietarios." Documento *online* (1994) . [Consultado 20/6/2004] http://www.gnu.org/philosophy/why-free.es.html
7. The Economist: "An open-source shot in the arm?". June 12th (2004). 15-17.
8. Polsani, P. R. : "Use and abuse of Reusable Learning Objects." Documento *online* (2003). [Consultado 26/9/2004] http://jodi.ecs.soton.ac.uk/Articles/v03/i04/Polsani/
9. Downes, S.: "Learning Objects: Resources for Distance Education Worldwide." International Review of Research in Open Distance Learning. Documento *online* (2001). [Consultado 26/9/2004] http://www.irrodl.org/content/v2.1/downes.html
10. González Barahona, J.; Seoane, J.; Robles, G.: "Introducción al *software* libre." Eureka Media, Barcelona (2003). 98-99.

11. Raymond, E. S.: “Cultivando la noosfera.” Documento *online* (2000) [Consultado 30/6/2004] http://sindominio.net/biblioweb/telematica/noosfera.html
12. González Barahona, J.; Seoane, J.; Robles, G.: “Introducción al *software* libre.” Eureka Media, Barcelona (2003). 241-300.
13. Sicilia, M.A.; García, E.: “On the Concepts of Usability and Reusability of Learning Objects.” International Review of Research in Open Distance Learning. Documento *online* (2003). [Consultado 26/9/2004] http://www.irrodl.org/content/v4.2/sicilia-garcia.html