XXX aniversario novática UPGRADE
Revista de la Asociación de Técnicos de Informática

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** (***UP**GRADE **E**uropean **NET**work*)

<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**, **AI2** y **ASTIC.**

Coordinación Editorial, Redacción Central y Redacción ATI Madrid
Padilla 66, 3º, dcha., 28006 Madrid
Tlfn.914029391; fax.913093685 <novatica@ati.es>
Composición, Edición y Redacción ATI Valencia
Av. del Reino de Valencia 23, 46005 Valencia
Tlfn./fax 963330392 <secreval@ati.es>
Administración y Redacción ATI Cataluña
Ciudad de Granada 131, 08018 Barcelona
Tlfn.934125235; fax 934127713 <secregen@ati.es>
Redacción ATI Andalucía
Isaac Newton, s/n, Ed. Sadiel,
Isla Cartuja 41092 Sevilla, Tlfn./fax 954460779 <secreand@ati.es>
Redacción ATI Aragón
Lagasca 9, 3-B, 50006 Zaragoza.
Tlfn./fax 976235181 <secreara@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)
Tlfn.986581413; fax 986580162 <secregal@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
Tlnf.914029391; fax.913093685 <novatica.publicidad@ati.es>
Imprenta
Derra S.A., Juan de Austria 66, 08005 Barcelona.
Depósito legal: B 15.154-1975 -- ISSN: 0211-2124; CODEN NOVAEC

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

sumario

Nº 175, mayo-junio 2005, año XXXI

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

Charles M. Schweik
Universidad de Massachusetts, Amherst (EE.UU.)

<cschweik@pubpol.umass.edu>

El análisis institucional aplicado al estudio del Software Libre como "bien comunal"

Traducción: Victoriano Giralt García (Servicio Central de Informática, Universidad de Málaga)

Resumen: *cualquiera que esté interesado en el Software Libre (SL) deseará conocer los motivos que conducen al éxito o al fracaso de los proyectos libres. El presente artículo es el inicio de un programa de investigación a cinco años, realizado con fondos de la Fundación Nacional para la Ciencia de los EE.UU.* (U.S. National Science Foundation)*, encaminado a identificar los principios de diseño que conducen al éxito de los proyectos de desarrollo de SL. Recientemente, los investigadores se han dado cuenta de que dichos proyectos se pueden considerar una forma de bienes comunales* (commons)*, que generan bienes públicos en forma de software. Esta conexión es importante, ya que existe un importante corpus de hallazgos teóricos y empíricos relacionados con los bienes comunales medioambientales que se han mantenido durante mucho tiempo y que podría aplicarse a los proyectos de SL y darles forma. Las instituciones -- entendidas como 'normas de uso' -- son un conjunto de variables básicas que se sabe que influyen en el resultado final de los planteamientos comunales (por ejemplo, bienes comunales de larga pervivencia o aquellos que caen presa de lo que G. Hardin ha dado en llamar "Tragedia de los bienes comunales"). Hasta el momento conocemos relativamente poco sobre el diseño institucional de los proyectos de SL y cómo evolucionan. El presente artículo presenta un marco que se utiliza con frecuencia para analizar el diseño institucional de los bienes comunales medioambientales que servirá de guia para futuras investigaciones empíricas en el campo de los proyectos de SL. Presentamos la trayectoria de estos proyectos y comentamos formas de medir su éxito o fracaso. El artículo termina presentando un conjunto de hipótesis de ejemplo en relación con los atributos institucionales de estos proyectos, que deberán ser verificadas.*

Palabras clave: *bienes comunales, instituciones, propiedad compartida, Software Libre.*

Autor

Charles M. Schweik es Profesor Adjunto del Dpto. de Conservación de Recursos Naturales y del Centro de Políticas y Administración Públicas de la Universidad de Massachusetts, Amherst (EE.UU.). Tiene un doctorado Políticas Públicas por la Universidad de Indiana (EE.UU.), es Licenciado en Administración Pública por la Universidad de Syracuse (EE.UU.), y es graduado en Informática. Uno de sus principales intereses en investigación es sobre el uso y gestión de la tecnología de la información pública. Durante más de seis años, entre la obtención de su grado en Informática y la licenciatura, trabajó como programador para IBM.

1. Introducción

Diversos artículos incluidos en la monografía del número de diciembre de 2001 de las revistas ***Novática*** y ***UPGRADE*** dedicada al Software Libre o Código Abierto [1] (al que llamaremos en adelante Software Libre o SL), evidenciaban que la composición de los equipos de desarrollo estaban pasando de estar compuestos completamente por voluntarios a incorporar miembros pagados por la industria, el gobierno u organizaciones sin ánimo de lucro [2]. Aunque el método de colaboración libre no es la panacea, hay suficientes éxitos para llegar a la conclusión que este modelo de desarrollo es viable e importante. Así mismo, un número mayor de proyectos libres han sido abandonados antes de conseguir los objetivos que se habían propuesto alcanzar cuando comenzaron [28]. Por tanto, una pregunta importante, identificada por diversos investigadores[3-8], es ¿qué factores conducen al éxito o el fracaso de los proyectos libres (PL)?

Recientemente, los proyectos de desarrollo de SL han sido identificados como una forma de "bienes comunales", donde grupos de desarrolladores voluntarios y miembros de equipos de profesionales pagados, de todas partes del mundo, colaboran para producir software que es un bien público [9-13][53]. Esta identificación nos da la oportunidad de conectar líneas de investigación sobre la gestión de recursos comunales naturales (hay resúmenes en [14][15]) con las investigaciones más tradicionales de sistemas de información relacionadas con la producción de software en general, y de SL en particular.

Ver los PLs como bienes comunales centra la atención en los atributos y temas referentes a la acción colectiva, la gobernanza y al sistema, a menudo complejo y cambiante de normas que ayudan a conseguir que los bienes comunales se mantengan en el tiempo [16]. La famosa frase de Hardin, "*La tragedia de los bienes comunales*" [17], describe entornos en los que los usuarios que comparten el bien (por ejemplo un pastizal) sobreexplotan el recurso, llevándolo a su destrucción. Para cada pastor, el añadir un animal a su rebaño añade valor, porque es un animal más para vender. La parte negativa aparece porque es un animal más pastando en los terrenos comunales. La opción más racional para cada pastor es añadir más animales, lo que conduce en último término a la sobreexplotación del pastizal.

En el caso del SL, gracias a su naturaleza digital, la sobreexplotación de los bienes comunales no es un problema, Sí lo es mantener (y quizá incrementar) un equipo de desarrolladores. En este entorno, la tragedia que se debe evitar es la decisión de abandonar el proyecto prematuramente, no por causa de un factor externo (como la aparición de una tecnología mejor que la que produciría el proyecto), sino por causa de alguna clase de problema interna al proyecto (tales como conflictos sobre su dirección, pérdida de apoyos financieros, etc.).[1]

Ya que éste es un punto importante, intentaré analizar esta tragedia concreta según la lógica de Hardin. En el entorno de los proyectos de SL, los desarrolladores (y, probablemente, los usuarios, los probadores, los documentalistas) sustituyen a los pastores en el papel de responsables de la toma decisiones. Lo que motiva a estas personas a participar es, en parte, el convencimiento previo de que el software que se producirá cubrirá una necesidad personal o de su organización. Sin embargo, la investigación de las motivaciones de los desarrolladores de SL [58] ha demostrado que reciben otros beneficios de su participación. Desde el punto de vista del desarrollador, merece la pena pasar una unidad de tiempo contribuyendo al proyecto porque: (1) hace que su nombre sea conocido y, así, aumentan las posibilidades de conseguir oportunidades de trabajo o consultoría en el futuro, (2) aprende nuevas técnicas a través de la lectura y revisión por sus iguales del código fuente hecho público por él, y/o (3) su emleador le paga por participar. De forma alternativa, sería lógico que dejase de contribuir con su tiempo porque no le gustan los derroteros que está

tomando el proyecto, o porque sus contribuciones no son aceptadas y no obtiene suficiente información de los motivos. En estos casos, la acumulación de insatisfacciones de los desarrolladores puede conducir a un abandono prematuro del proyecto, a causa de factores internos al mismo. La tragedia de los bienes comunales, en este contexto, consiste en la pérdida prematura de un equipo productivo, no a una sobreapropriación como en el famoso ejemplo de Hardin sobre los pastizales. Por tanto, una de las principales preocupaciones que deberán tener las organizaciones de tecnologías de la información (TI) que estén evaluando el SL como como política o como estrategia es la forma de mantener a largo plazo un esfuerzo entusiasta de producción y mantenimiento, y cómo evitar el abandono prematuro del proyecto.

En *Governing the Commons* [18], Elinor Ostrom puso de relieve que en ciertos terrenos comunales se evita la tragedia de Hardin — los terrenos llegan a tener una larga pervivencia --gracias a diseños institucionales creados por comunidades autogestionadas. Las instituciones, en este contexto, se pueden definir como un conjunto de normas --formales o no-- que se componen de mecanismos para vigilar, sancionar y resolver conflictos que ayudan a crear un conjunto de incentivos para la gestión del entorno comunal. En el entorno de los bienes comunales del SL, la evolución de las instituciones del proyecto puede ayudar a explicar porqué unos proyectos avanzan suavemente de alfa a beta hasta llegar a ciclos regulares y estables de entrega, con la creación y mantenimiento de grupos de desarrolladores y comunidades de usuarios cada vez mayores, mientras que otros proyectos son abandonados antes de alcanzar la madurez. Aunque la investigación demuestra que la inmensa mayoría de los proyectos de SL tienen un solo desarrollador o un pequeño grupo de ellos [48-52], yo opino que la influencia del diseño institucional será cada vez más crítica a medida que los proyectos crezcan (en el número de partes interesadas) y maduren. Más aún, el aumento de la participación de las empresas y Administraciones Públicas en el desarrollo de SL llevará sin duda a entornos institucionales más complejos. Ésta es la razón por la que creo que la atención al diseño institucional de los proyectos Libres s es de importancia capital a medida que las colaboraciones en SL (y otras "*colaboraciones en contenidos abiertos*", véase [13]) se hacen más habituales.

Este artículo describe algunos elementos de un programa de investigación a cinco años recién iniciado que estudiará los diseños institucionales de los proyectos de desarrollo de SL desde el punto de vista de los bienes comunales. Uno de los objetivos principales de la investigación es identificar los principios de diseño que contribuyen al éxito o al fracaso final de estos proyectos. El artículo se estructura de la siguiente forma. Primero, explicaré porqué los proyectos de desarrollo de SL son un tipo de bienes comunales o, más concretamente, un "régimen de propiedad comunal".

Después, describiré a lo que me refiero con el término instituciones y describiré un marco teórico que se utiliza bastante en Ciencias Sociales para estudiar el diseño institucional de los entornos de terrenos comunales. Posteriormente describiré la trayectoria general de los proyectos de desarrollo de SL y comentaré diversas formas de medir el éxito y el fracaso en las etapas de dicha trayectoria. Daré algunos ejemplos de hipótesis relacionadas con el diseño institucional que podrían ayudar, si se prueban empíricamente, a identificar principios de diseño para los proyectos de SL. Terminaré con un comentario sobre porqué esto debe preocupar a los profesionales de las TI.

2. El Software Libre son bienes públicos desarrollados por regímenes de propiedad comunal

Es posible ver el Software Libre desde dos perspectivas: el uso y el desarrollo. Consideraré primero la parte del uso. En Ciencias Sociales se definen cuatro categorías de bienes: recursos privados, públicos, clubes y comunes, que se diferencia en función de dos propiedades (**figura 1**) [22][21]: en primer lugar, el grado de dificultad para excluir a otros del uso o acceso al recurso; en segundo lugar, si el recurso tiene exclusividad, esto es, si yo tengo una unidad del bien, ¿impide esto que la utilicen otros al mismo tiempo?

Adaptada de [21: 7]

CAPACIDAD PARA EXCLUIR	EXCLUSIVOS: NO	EXCLUSIVOS: SI
Difícil	Bienes públicos	Recursos comunales
Fácil	Bienes de tipo Club	Bienes privados

Figura 1. Una clasificación general de los bienes (adaptado de [21: 7]).

El software propietario tradicional se puede clasificar como un bien de tipo club de la **figura 1**. La naturaleza digital del software (descargable de Internet o copiable de un CD-ROM) hace que no tenga exclusividad. El precio para obtenerlo (y las restricciones de copia de las "licencias con envoltorio de celofán") hace posible excluir a aquellos que no lo adquieren. Pero, en muchos casos, esta exclusión no tiene siempre éxito. Se asume a nivel general que se producen copias ilegales de software propietario, creando una forma diferente de club, con un acceso a éste basado más en la disposición a aceptar el riesgo de ser pillado que en un precio para acceder. Pero, independientemente de que la compañía pueda o no impedir con éxito el pirateo de su software, el software propietario, debido a su naturaleza digital, pertenece a la categoría de bienes tipo club.

El SL difiere del propietario en que las licencias libres (tales como la licencia GNU GPL, *General Public License*) permiten a los usuarios copiar y distribuir el software siempre que les plazca mientras que cumplan los requisitos de la licencia [54]. Estas licencias incluyen un mecanismo para actuar frente a la trasgresión de las normas especificadas, por tanto, la exclusión es teóricamente posible pleiteando de acuerdo con las leyes mercantiles o de propiedad intelectual [61-62], pero es poco probable en la mayoría de los casos [61]. Dado que el SL tampoco es exclusivo --se copia libremente en forma digital por Internet o en CD-ROM (como, por ejemplo, en el caso de una distribución de Linux)-- técnicamente, se debe clasificar como un bien de tipo club --un club cuya única cuota de entrada es el cumplimiento de la licencia. Ahora bien, como la distribución del SL es de alcance global sin costes monetarios asociados, muchos lo clasifican como un bien público [23-25][55].

Permítanme volver a considerar la producción del SL. Anteriormente he indicado cómo algunos consideran los proyectos de SL una cierta forma de "bienes comunales" [25][54]. McGowan [53] se refiere a estos bienes comunales como "espacios sociales" dedicados a la producción de software disponible y modificable libremente. Aunque estos proyectos conllevan colaboración, y en contra de lo que algunos podrían creer, en estos bienes comunales existen derechos de propiedad (*copyright*) y cuestiones de propiedad.

Raymond (según cita de McGowan) define como propietarios de un proyecto de SL a aquellos que tienen "*un derecho exclusivo, reconocido por la comunidad en general, para redistribuir versiones modificadas*" [53: 24]. Según Raymond, uno se convierte en el propietario de un proyecto bien porque (1) es la persona o el grupo que inició el proyecto desde cero; porque (2) es alguien que ha recibido, de manos del propietario original,

> Los proyectos de Software Libre se pueden considerar una forma de bienes comunales (*commons*) que generan bienes públicos en forma de software

la autoridad para dirigir los trabajos futuros de mantenimiento y desarrollo; porque (3) es la persona que se hace cargo de un proyecto que se considera abiertamente como abandonado y se toma el trabajo de localizar al autor, o autores, original y de conseguir permiso para ser propietario del mismo. McGowan añade una cuarta opción --la "apropiación hostil"-- cuando el proyecto puede ser secuestrado o "bifurcado" en virtud de los permisos de "nuevos trabajos derivados" que consiente la licencia. La bifurcación se produce con frecuencia cuando parte del equipo considera que el proyecto está encaminado técnica o funcionalmente en una dirección equivocada. Puede producirse una especie de motín y se crea un nuevo proyecto a partir de código fuente del proyecto original. El resultado es que aparecen dos versiones que compiten entre ellas [53].

Para algunos lectores la definición de Raymond de los propietarios de PLs puede resultar un tanto problemática. Esta definición engloba la visión libertaria de Raymond de los PLs, en la cual la comunidad en su conjunto define de alguna manera los derechos de propiedad y actúa colectivamente como una unidad para apoyar estos derechos. Para algunos, esta definición y actuación colectivas parecen más bien difíciles de creer. Una forma altrenativa de identificar o definir un propietario en entornos de SL es en virtud de la capacidad de una persona o equipo para iniciar o mantener un proceso de desarrollo colectivo coherente. Desde este punto de vista, la propiedad es más bien el resultado de barreras contra la expropiación y no requiere alguna forma mística de apoyo colectivo.

El lector debería observar que esta definición alternativa de la propiedad libre coincide con las cuatro formas de ser identificado como propietario de SL Libre según Raymond y McGowan citadas más arriba. Teniendo en cuenta los anteriores aspectos de la propiedad, la clave está aquí: los proyectos de Software Libre son una forma de "régimen de propiedad comunal" autogestionado, en el que los desarrolladores colaboran para producir producir un bien público [9][11][12][27][13][25][53][54]. Aunque el término utilizado con mayor frecuencia es "bienes comunales", los proyectos de SL se definen de forma más precisa como "regímenes de propiedad comunal". Definir los PLs como una forma de régimen de propiedad comunal nos da la oportunidad de conectar con el conocimiento acumulado a lo largo de los años sobre la gestión y administración de recursos naturales comunales en entornos de propiedad colectival (por ejemplo [14][15]). Recientemente, Weber indicó la importancia de la gobernanza y administración en los proyectos de SL cuando dijo: "*El proceso de las fuentes abiertas es un experimento en curso. Se está probando una mezcla imperfecta de liderazgo, mecanismos informales de coordinación y normas explícitas e implícitas, junto a estructuras formales de gobernanza que están evolucionando y lo hacen a un ritmo que ha bastado para mantener unidos sistemas sorprendentemente complejos*" [12: 189].

3. Un marco para el estudio de los diseños institucionales en los proyectos de Software Libre

Lo que Webber identifica como normas sociales, procesos informales de coordinación y estructuras formales de gestión, se corresponde con lo que en Ciencias Económicas y Políticas se denomina 'instituciones' [18][21][31]. Durante más de 40 años, los investigadores, incluido el autor [32-34], han utilizado el "Marco para el análisis institucional" (**figura 2**) para organizar las ideas sobre los casos de terrenos comunales [31: 8]. Este marco no se ha aplicado aún al estudio de los bienes comunales del SL, pero la lente analítica que nos da complementa otras investigaciones en curso sobre SL realizadas por investigadores de campos más tradicionales de los sistemas de información (por ejemplo, [35-38]).

Consideremos una situación en la que un analista está intentando comprender porqué un determinado proyecto de SL tiene vitalidad o porqué está perdiendo impulso. La **figura 2** muestra los PLs como sistemas dinámicos con retroalimentación. El analista podría empezar a estudiar el proyecto fijándose primero en los elementos de la izquierda: los atributos físicos, comunitarios y normativos.

Los atributos físicos hacen referencia a diversas variables relacionadas con el propio software o a parte de la infraestructura para coordinar al equipo. Incluyen el lenguaje o lenguajes de programación utilizados, el grado de modularidad de la estructura del código y el tipo de infraestructura de comunicación y gestión de contenidos que se utiliza.

Los atributos de comunidad hacen referencia al conjunto de variables relacionadas con las personas involucradas en el proyecto de SL, tales como si son voluntarios o se les paga por participar, si todos hablan o no el mismo idioma, y otros aspectos más difíciles de cuantificar relacionados con el capital social, tales como lo bien que se llevan los miembros del equipos, lo que se fian unos de otros [63], etc. Esta componente también incluye otros atributos no físicos del proyecto, tales como la situación financiera y las fuentes de estos fondos (por ejemplo, una fundación).

Las normas-de-uso se refieren al tipo de normas definidas con el fin de guiar el comportamiento de los participantes cuando se dedican a actividades cotidianas relacionadas con el desarrollo, mantenimiento o uso del SL. La licencia libre concreta que se use es un componente importante en la categoría de normas-de-uso. Pero yo espero que la mayor parte de los PLs --especialmente aquellos más maduros y con un mayor número de participantes-- tendrán definidos otros conjuntos de normas formales o informales o normas sociales que ayuden a coordinar y administrar el proyecto.

La parte central de la **figura 2**, Actores y Escenario de la Acción, indica un momento o período de tiempo durante el cual los atributos de la parte izquierda permanecen relativamente constantes y los actores (por ejemplo, desarrolladores de software, probadores, usuarios) involucrados en el proyecto de SL toman decisiones y llevan a cabo acciones (p.e.: programar, revisar código, decidir reducir o terminar su participación, etc.). La suma de estos actores que toman decisiones y llevan a cabo acciones se muestra como Patrones de Interacción en la **figura 2**. Un cúmulo de acciones tiene como resultado una cierta Consecuencia (parte derecha abajo, en la **figura 2**). Una consecuencia podría ser un cambio en los atributos físicos de los bienes comunales del SL (por ejemplo, la entrega de una nueva versión), un cambio en los atributos comunitarios del proyecto (por ejemplo, personas nuevas que se suman o personas que dejan el proyecto), un cambio en las normas-de-uso (por ejemplo, un nuevo sistema para resolver conflictos) o cualquier otra combinación de los mismos. En la **figura 2** este tipo de cambios se muestran por medio del sistema de retroalimentación desde las consecuen-

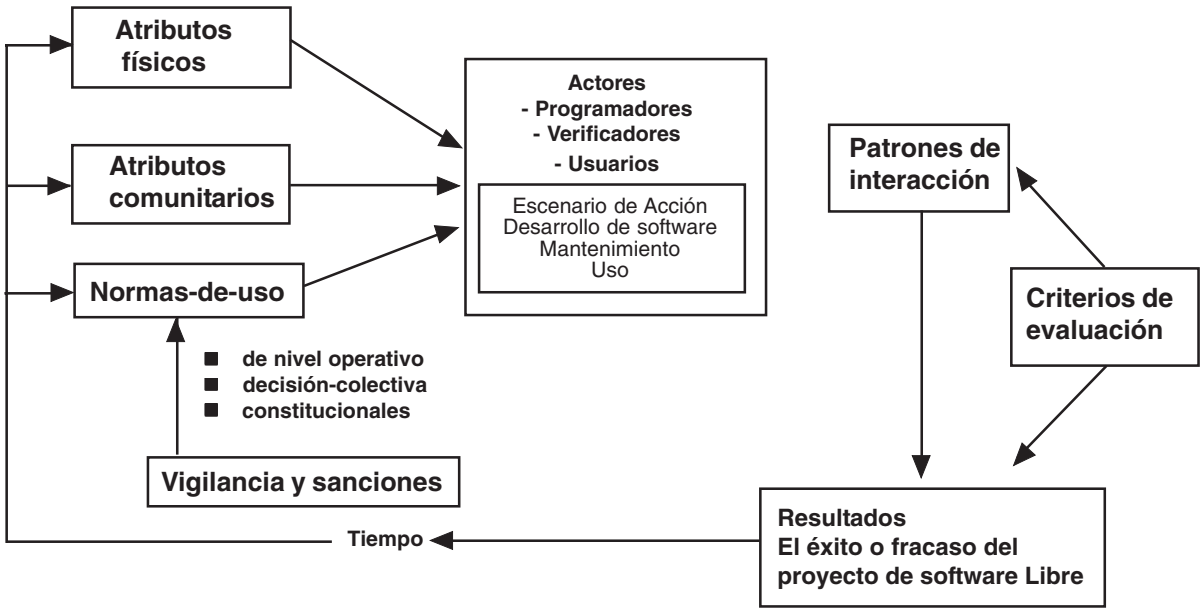

Figura 2. Un marco para el análisis institucional de los entornos de bienes comunales. Adaptado de [21:37].

cias hacia los tres conjuntos de atributos de la propiedad comunal en la parte izquierda de la figura, y se inicia un nuevo período de tiempo.

Buena parte del análisis institucional se dedica al estudio de las normas, desde leyes formales a normas informales de comportamiento, procedimientos de actuación normalizados y similares. En la categoría de normas-de-uso de la **figura 2** se contienen tres niveles anidados que, en su conjunto, influyen en las acciones que se llevan a cabo y en las consecuencias resultantes a estos diferentes niveles de análisis [39]. Las normas de nivel operativo afectan a las actividades cotidianas que realizan los participantes en los bienes comunales del SL. Puede tratarse de normas escritas formalmente, o, lo que es más común en entornos libres, pueden ser normas de comportamiento de la comunidad.

Un ejemplo de normas de nivel operativo pueden ser los procedimientos que se siguen para añadir nuevas funcionalidades a la siguiente revisión del software. Otro ejemplo pueden ser las normas para ascender a un desarrollador a una posición de mayor responsabilidad en la toma de decisiones dentro del proyecto. El nivel de Decisión-Colectiva representa el espacio de discusión donde los miembros del equipo con autoridad definen los objetivos del grupo y crean o revisan normas de nivel operativo que lleven al equipo hacia esos objetivos. Además, en este nivel existe un sistema de normas de Decisión-Colectiva que definen quién puede ser elegido para cambiar las normas de nivel operativo y especifican el proceso para cambiar estas normas [21]. Las normas de Decisión-Colectiva podrían, por ejemplo, determinar cómo puede cambiar el equipo el proceso de revisión del código antes de que el nuevo código pueda ser registrado o 'transferido' [40]. Las normas de nivel de Decisión-Constitucional especifican quién tiene derecho a cambiar las normas de Decisión-Colectiva y también definen los procedimientos para tales cambios. Por ejemplo, si el líder de de un PL decide dedicarse a una nueva actividad, las normas de Decisión-Constitucional definirían cómo se elige un sustituto. En cada nivel, pueden existir mecanismos para verificar que se cumplan las normas y mecanismos sancionadores para cuando no se cumplen.

Resumiendo, en cualquier momento del tiempo del ciclo de vida de un proyecto de SL, los programadores, usuarios y probadores tomarán decisiones sobre su participación en función de los atributos físicos, comunitarios e institucionales que presente el proyecto, así como en función de su percepción de hacia donde se encamina el proyecto y sus propias circunstancias personales. Los participantes toman decisiones y llevan a cabo acciones en tres niveles: operativo, de decisión-colectiva y, en menor medida, de decisión-constitucional. Una hipótesis a verificar en el curso de la presente investigación es que los sistemas de normas-de-uso se harán más complejos a medida que el proyecto de SL madura y aumenta el número de participantes. También espero que el diseño institucional se haga más complejo en situaciones en las que una o más organizaciones (p.e.: empresas, organizaciones sin ánimo de lucro o Administraciones Públicas) contribuyen recursos al proyecto.

Esto coincide con McGowan [53: 5] cuando dice: "*Las estructuras sociales necesarias para dar soporte a la producción de proyectos grandes y complejos son diferentes --si existen-- de las estructuras necesarias para dar soporte a los proyectos pequeños ...*".

4. La trayectoria de los proyectos de Software Libre

Ahora pasaré a la cuestión de cómo evaluar el componente "Resultados" del marco. Aunque la **figura 2** muestra un bucle de retroalimentación para llamar la atención sobre la naturaleza dinámica y evolutiva de estos proyectos [48, 45], no es demasiado adecuada para mostrar las propiedades longitudinales. Es por esto que incluyo la figura 3.

En trabajos previos, he argumentado que los proyectos de software Libre siguen una trayectoria en tres etapas (**figura 3**): (1) inicio; (2) la decisión de "convertirse en abiertos" y licenciarlos como SL; y (3) un periodo de crecimiento, estabilidad o abandono. La mayor parte de la investigación sobre SL se centra en proyectos que están en la etapa 3. Pero, algunas de las decisiones que se tomen en las etapas previas pueden ser factores decisivos que conduzcan a las consecuencias de crecimiento o abandono de la etapa 3.

Considérese la etapa 1 de la **figura 3**. En muchos casos, los proyectos de SL comienzan con conversaciones privadas de uno o unos pocos programadores que trabajan juntos para desarrollar una pieza 'central' de software. En este punto, el software puede no estar aún disponible con una licencia libre o ser descargable de Internet, y en algunos casos el equipo puede no haberse ni tan siquiera planteado ofrecerlo bajo una licencia de SL. Pero en esta etapa se pueden tomar decisiones críticas, tales como el nivel de modularidad del código, que podrían tener una influencia vital en lo bien que se

podría desarrollar en un entorno de propiedad comunal libre durante la etapa 3.

Aunque la idea el "grupo reducido y privado que empieza de cero" es, probablemente, lo que ve la mayoría como la fase de inicio de un proyecto de SL, existe al menos otra alternativa: "la suelta de software" [60]. En estos casos, el software se desarrolla inicialmente con un modelo más tradicional, propietario, de fuentes cerradas, dentro de una empresa. En algún momento — quizá después de años de trabajo — los responsables pueden tomar la decisión estratégica de no dar más servicio y, como consecuencia, ofrecer el código y licenciarlo como SL. Este caso puede ser más prevalente en años venideros si las empresas de software siguen considerando el SL como una pieza de su estrategia de negocios.

La fase de "convertirse en abiertos" (**figura** 3, etapa 2) probablemente sea corta, pero quizá no tan simple como podría parecer a primera vista. En esta etapa, los miembros del equipo deciden cuál será la licencia de SL, y, quizá más importante, crean un espacio de trabajo público y una infraestructura de colaboración (por ejemplo, un sistema de control de versiones, métodos para revisión, mecanismos de seguimiento de fallos, etc.) que den soporte al proyecto. Las plataformas como sourceforge.net o freshmeat.net han facilitado enormemente este paso, pero hay algunos proyectos que utilizan plataformas basadas en web desarrolladas por ellos mismos.

Debo señalar en este punto que, en algunos PLs, las etapas 1 y 2 se pueden combinar. Puede darse el caso, con bastante frecuencia, en el que un miembro fundador tiene una idea e inmediatamente propaga una petición de ayuda para encontrar socios que colaboren en el desarrollo del proyecto. Esta petición puede llevar aparejada de forma inmediatas la creación de una página del proyecto en la web o en un sitio de hospedaje como sourceforge.net.

Cualquiera que sea la forma en que un proyecto supera la etapa 2, el siguiente paso es la etapa 3 de la **figura 3**. Esta etapa describe la parte del ciclo de vida del proyecto durante la cual el software se desarrolla activamente y se utiliza con una licencia de SL estando disponible públicamente en Internet.

Muchos de los primeros estudios sobre proyectos de SL se centraron en casos que se incluyen en la categoría de éxitos de "gran crecimiento" (en términos de cuota de mercado o número de desarrolladores) como Linux o el servidor web Apache. Alcanzar esto niveles suele ser la medida de éxito que se espera o asume para estos proyectos en la literatura sobre SL. Sin embargo, los estudios empíricos sobre SLe han demostrado que la mayoría de los proyectos nunca alcanzan ese nivel y que muchos, quizá la mayor parte, involucran a un pequeño número de individuos [48-52].

Algunos de estos estudios pueden estar concentrados en proyectos que se encuentren en el principio de su ciclo de vida, con las personas trabajando para conseguir un alto nivel de crecimiento. Pero en otros casos, los miembros de un determinado proyecto pueden estar muy satisfechos de permanecer 'estables' y activos con un pequeño número de participante (**figura 3**, etapa 3: Grupo Pequeño). Ciertos proyectos de SL en el campo de la bioinformática podrían ser buenos ejemplos de este tipo de entornos [47]. Lo principal de la **figura 3** es que hay etapas importantes en la trayectoria de los proyectos de SL y que la medidas para determinar el éxito o el fracaso cambiarán con toda probabilidad durante las mismas. Más aún, también evolucionarán los atributos físicos, comunitarios e institucionales del proyecto.

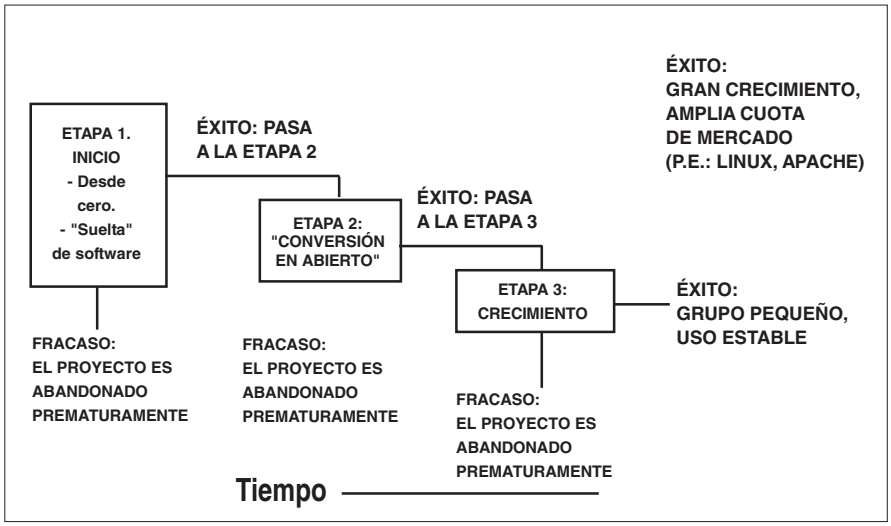

Figura 3. Etapas de los proyectos de SL y medida de los resultados (éxito).

5. Medida del éxito o fracaso de los proyectos de Software Libre a lo largo de su trayectoria

He indicado más arriba que el objetivo del presente proyecto de investigación es definir los "principios de diseño" que llevan a colaboraciones en SL exitosas. El concepto que busco explicar con el trabajo empírico que estoy comenzando es el éxito o fracaso de los PLs. Lo que sigue es la descripción de un método para medir el éxito o fracaso. Otros también han realizado investigaciones para cuantificar esto, y yo construyo usando sus importantes trabajos como base [3][4][8][41].

Para mis fines, una medición inicial del éxito o fracaso de una colaboración para un PL requiere hacer dos preguntas, por este orden. Primero, ¿tiene el proyecto un cierto nivel de actividad de desarrollo, o, desde el punto de vista del desarrollo el proyecto parece abandonado? Segundo, en el caso de proyectos que parezcan abandonados, ¿fueron abandonados por causas que el equipo no podía controlar? Permitanme comentar cada una de las preguntas.

5.1. ¿Muestra el proyecto signos de actividad o parece abandonado?

Diversos estudios han medido si un proyecto de software Libre está 'vivo' o 'muerto' [48], observando los cambios durante un cierto lapso de tiempo de las siguientes variables de atributos físicos del software (**figura 2**):

- Trayectoria de versiones (p.e.: paso de versión alfa a beta y de ésta a estable) [3]
- Número de versión [3][48]
- Líneas de código [48][43]
- Número de 'ingresos' en un repositorio del almacenamiento central [45]

Igualmente, el analista podría controlar los cambios en variables de los atributos comunitarios (**figura 2**), tales como:
- La puntuación de actividad o vitalidad en las medidas de plataformas de colaboración como Sourceforge.net o Freshmeat.net [3][8][48].

Está claro que si se producen cambios de estas medidas durante un cierto tiempo, el proyecto muestra un cierto nivel de actividad. Un punto fundamental será decidir cuánto tiempo será necesario o adecuado para considerar que un proyecto está muerto o abandonado. Considero probable que algunos de los proyectos de software más maduros pueden mostrar periodos de adormecimiento hasta que un usuario o desarrollador sugieren una nueva idea.

En consecuencia, el periodo de tiempo sin signos de actividad debiera ser relativamente largo antes de decidir que un proyecto está muerto, o, mejor aún, el analista debería buscar alguna prueba de que el proyecto ha sido

cerrado o abandonado en la propia documentación del mismo (p.e.: su sede web).

5.2. Si el proyecto parece abandonado ¿se produjo el abandono por causas ajenas al equipo?

Clasificar un proyecto como muerto no indica per se que el proyecto fracasara [63]. Algunos proyectos pueden no mostrar actividad debido a que han alcanzado su completa madurez de desarrollo: el software realiza el trabajo para el que se diseñó y no necesita mejoras. En estos casos, el proyecto se clasificaría como un éxito, no como un fracaso.

En otros casos, un proyecto se puede clasificar como muerto o abandonado pero ha llegado a ese estado por razones ajenas al equipo, tales como la aparición de una tecnología (rival) que es tecnológicamente superior o tienen una mayor aceptación (véase el ejemplo del Gopher frente a la WWW de la nota final 1). En estos casos, el proyecto, probablemente, no debería ser considerado como un fallo desde el punto de vista de la colaboración (aunque en ciertos casos bien que se podría considerar). Simplemente surgió una tecnología rival que hacía mejor el trabajo.

Por último, existirán casos en los que el proyecto no muestre signos de vida, el software no haya llegado a desarrollarse en todo su potencial y no se detectan factores externos que indujeran a los desarrolladores a abandonarlo. Yo los clasificaría como casos de abandono prematuro, debidos a que algún factor interno al proyecto hizo a la gente abandonar el esfuerzo antes de que alcanzase la madurez.

En consecuencia, pretendo utilizar, en esta investigación, las preguntas 5.1 y 5.2 para clasificar los proyectos como éxitos o fracasos. Los proyectos exitosos o bien mostrarán signos de vida o no mostrarán signos de más desarrollo porque han alcanzado la madurez.[2] Los proyectos abandonados por causas externas se eliminarán del conjunto de casos de estudio, ya que no se pueden clasificar ni como éxitos ni como fracasos. Se clasificarán como fracasos aquellos casos en los que parezca haber sido abandonados de forma prematura por algún tipo de problema interno al mismo.

Estas mediciones son fáciles de obtener en proyectos que se encuentren en la fase 3 (crecimiento) de la **figura 3**. Será más difícil identificar proyectos que se encuentren en las fases iniciales (etapa 1 y 2) para poderlos estudiar, pero cuando lo consiga, se deberían poder aplicar estos mismos conceptos.

6. Hacia la identificación de los "principios de diseño" de los bienes comunales del Software Libre

Hasta el momento, he intentado destacar cuatro puntos. Primero, que los proyectos de SL son una forma de régimen de propiedad comunal que tiene atributos físicos, comunitarios e institucionales que afectan su rendimiento. Segundo, que hay diversas maneras de medir el éxito o fracaso de estos proyectos pero que será importante una medida que determine si la colaboración se abandonó de forma prematura (fracaso) o se mantuvo hasta que el software alcanzó la madurez (éxito). Tercero, que los diseños ins-titucionales --las normas-de-uso-- son un aspecto que hasta el momento ha sido mayoritariamente soslayado por los estudios sobre el SL. Cuarto, que es deseable la identificación de unos "principios de diseño" que conducen al éxito de estos proyectos en las diferentes etapas de la **figura 3**, en la medida en que un mayor número de organizaciones ven el SL como una estrategia de las TI.

La identificación de los principios de diseño requerirá un estudio sistemático de proyectos de SL en las diferentes etapas de la **figura 3**, prestando atención a las medidas de éxito o fracaso adecuadas para cada una de ellas. Será necesario diseñar hipótesis en relación con los tres conjuntos de variables independientes --los atributos físicos, comunitarios e institucionales de la figura 1-- fundamentadas en trabajos sobre desarrollo de software tradicional, estudios más recientes explícitamente dedicados a los proyectos de SL, y en trabajos aplicables en relación con los terrenos o recursos naturales comunales.

Para no extendernos, terminaré este artículo dando algunas hipótesis en relación a los diseños institucionales de los proyectos de SL (normas-de-uso de la **figura 1**) y mostrando su relación con los estudios sobre recursos naturales comunales.

Los proyectos de SL tendrán un mayor éxito (no los abandonarán de forma prematura) si dan un cierto nivel de voz en la construcción de normas de nivel operativo a los participantes de nivel más bajo.

Se ha demostrado, en el entorno de los recursos naturales comunales, que los recursos se sostienen mejor cuando los usuarios tienen un ciertos derechos para definir y hacer cumplir sus propias normas de nivel operativo [18][14]. Aplicando esto a los proyectos de SL, si, por ejemplo, una autoridad superior impone normas de nivel operativo sin consultar a los que trabajan "en las trincheras", los trabajadores se desilusionarán y abandonarán el proyecto. Por el contrario, si los desarrolladores y usuarios de un proyecto de SL Libre pueden opinar sobre la definición y revisión de las normas de nivel operativo a medida que avanza el proyecto, la teoría de bienes comunales sugiere que estarán más dispuestos a participar a largo plazo.

Los proyectos de SL tendrán más éxito (no se abandonarán de forma prematura) si se han establecido mecanismos de decisión-colectiva para cambiar las normas de nivel operativo cuando sea necesario.

También se ha demostrado que los recursos naturales comunales de larga pervivencia, suelen tener diseños institucionales que permiten la adaptación de las normas cuando es necesario. Los sistemas con normas fijas fracasarán lo más seguro, porque la comprensión de la situación, en el momento en que fueron definidas, podía ser incorrecta, en cierta medida, o la situación para la que se definieron cambia en último término [15].

Los proyectos de SL tendrán un mayor éxito (no serán abandonados de forma prematura) si tienen establecidos sistemas para resolución de conflictos entre los miembros del grupo.

Estudios como el de Divitni et al. [56] y el de Shaikh y Cornford [57] tienen comentarios sobre los conflictos en el entorno del SL. El tipo de conflicto extremo es la 'bifurcación', descrita más arriba. Los entornos de bienes comunales que disponen de mecanismos para dirimir los conflictos suelen conseguir una pronta resolución unida a nuevo aprendizaje y comprensión dentro del grupo (15: 1909). Los proyectos que son incapaces de manejar los conflictos pueden verse abocados a situaciones disfuncionales donde no es posible seguir cooperando.

Los proyectos de software Libre tendrán un mayor éxito (no serán abandonados de forma prematura) si:

… tienen previstos sistemas que permiten vigilar las normas de nivel operativo.

… tienen un cierto nivel de sanciones graduales para aquellos que transgreden las normas.

… tienen personas encargadas de hacer cumplir las normas cuyos juicios se consideran legítimos y efectivos.

Las normas de nivel operativo funcionan sólo si se hacen cumplir. Las investigaciones sobre recursos naturales comunales han demostrado que los propios usuarios pueden establecer con frecuencia sistemas de vigilancia de bajo coste y que la eficacia es máxima cuando existen sanciones menores (inicialmente) para los trasgresores [15]. Sharma, Sugumaran y Rajgopalan [59] indican que en los proyectos de SL existe una cierta forma de sistemas de vigilancia y sanción. Sin embargo, se habla poco de este asunto en las publicaciones actuales sobre SL. La literatura sobre bienes comunales sugiere que la probabilidades de éxito serán mayores si existe un mecanismo de vigilancia del cumplimiento de las normas de nivel operativo así como un sistema de sanciones progresivas para frenar a los trasgresores.

> Los proyectos de Software Libre tendrán más éxito en la medida que exista un flujo constante y preestablecido de fondos que los mantengan

Un reproche por correo privado personal que, si no tiene éxito, da lugar a una reprimenda delante de todo el equipo, es un ejemplo de procedimiento sancionador gradual en el entorno del SL.

Además, los estudios sobre bienes comunales han demostrado también que el cumplimiento de las normas funciona mejor cuando los sancionadores son personas reputadas como efectivas y legitimadas [15]. Traducido al entorno libre, la imposición de sanciones efectivas a los trasgresores requiere que lo haga alguien que ha sido formalmente designado para ello o que es identificado como una autoridad legítima dentro del grupo.

7. Conclusiones

Se pretende que las hipótesis presentadas en la sección anterior ilustren lo que se necesita hacer para avanzar hacia la identificación de los principios de diseño en los entornos de bienes comunales de SL. He dado ejemplos que subrayan asuntos institucionales (normas-de-uso) porque creo que esta es un área que ha sido ignorada hasta el momento en la investigación sobre SL. Sin embargo, también se pueden diseñar hipótesis verificables para el resto de categorías de atributos de la parte izquierda de la **figura 1**. Por ejemplo, una que es obvia pero importante: los proyectos de SL tendrán más éxito en la medida que exista un flujo constante y preestablecido de fondos que los mantengan.

Puede ocurrir que, para muchos PLs, la atención al diseño institucional sencillamente no importe, porque el equipo de desarrollo está formado por un individuo o un pequeño grupo. En esta etapa pueden ser más importantes atributos físicos o comunitarios. Sin embargo, tengo la sospecha de que en los proyectos más grandes (en cuanto a número de líneas de código), o en los PLS a los que contribuyen más de una empresa u organización, el diseño institucional será un conjunto de variables de importancia mucho mayor.

En los próximos años, con fondos de la Fundación Nacional para la Ciencia de los EE.UU., llevaré a cabo un estudio sistemático de estos proyectos, con especial atención al diseño y evolución de las estructuras institucionales y sus asuntos relacionados.
¿Qué le importa esto a los lectores de ***Novática*** y ***UP****GRADE*? Volvamos al principio. Diversos artículos de la monografía del número de diciembre del año 2001 ponían de manifiesto la naturaleza cambiante del modo de participación en los proyectos de SL: cada vez son más los actores que no son voluntarios sino personas pagadas por sus organizaciones para contribuir al desarrollo del software. No es difícil imaginar un futuro en el cual las Administraciones Públicas y/o las empresas dedican recursos a trabajar juntas en un proyecto de SL (las empresas lo están haciendo ya).
La principal lección que se debe aprender de los recursos naturales comunales es que las instituciones importan. Anticipo que, a medida que maduren el SL y sus bienes comunales, los atributos institucionales se harán más importantes y visibles en cuanto que factores que llevan al éxito o al fracaso de estos proyectos.

Notas finales

1. Estoy en deuda con alguien anónimo que revisó este texto y me hizo ver que algunos proyectos abandonados no son tragedias. Esta persona me propuso el ejemplo de la tecnología Gopher, que se vió sobrepasada por la World Wide Web. Éste es un caso en el que un factor externo condujo al abandono prematuro del proyecto de software, pero no se consideraría una tragedia. Debo indicar también que la idea de la supresión de un proyecto se ha usado en el pasado en el desarrollo de software más tradicional [42], pero el término "abandono prematuro" encaja mejor que "supresión prematura" en el entorno de SL ya que, en muchos casos, no existe una organización formal que toma la decisión de finalizar el proyecto de forma anticipada.

2. Un añadido analítico de este proyecto será analizar el 'entusiasmo' de los proyectos exitosos --capturar el nivel de vida que muestra un proyecto (en función de la actividad de los desarrolladores o los usuarios). En otras palabras, mi intención última es desarrollar una medida del éxito que vaya más allá de una medición "vivo" frente a "muerto". Diversos estudios (por ejemplo, [3][4][8]) se han fijado en las medidas de entusiasmo, concentrándose en variables como el número de personas en el equipo de desarrollo formal o en el equipo de desarrollo amplio (por ejemplo, personas que informan de fallos), el número de transferencias de código, el número de descargas, etc. Otras posibles medidas del entusiasmo podrían incluir un examen del sentido del cambio en el número de participantes de los equipos de desarrollo formal y amplio. Sin embargo, se hace necesario un examen más exhaustivo de estas medidas --que va más allá del alcance del presente artículo. Pero es muy probable que cualquier medida de entusiasmo estará íntimamente relacionada con la etapa de desarrollo del proyecto. Por ejemplo, Dalle y sus colegas [44] indican que los proyectos más jóvenes y activos en sourceforge.net tienen más probabilidades de atraer desarrolladores con una tasa mayor que los proyectos más maduros con un corpus de código de mayor tamaño. Desde este punto de vista, las medidas de entusiasmo podrían parecer muy similares en un proyecto que está siendo abandonado de forma prematura y en otro que está alcanzando la madurez. Es por esto por lo que, en el presente artículo, sólo quiero mostrar mi intención de ir más allá en la investigación sobre cómo conceptualizar y usar las medidas de entusiasmo, pero está más allá del alcance del mismo hacerlo.

Agradecimientos

Este estudio se realizó con el apoyo de una beca de la *US National Science Foundation* (NSFIIS 0447623), pero los hallazgos, recomendaciones y opiniones que se expresan en él son responsabilidad de sus autores y no reflejan necesariamente las opiniones del organismo que los patrocina.