

COOPERACIÓN EN EL DESARROLLO DE PROYECTOS DE SOFTWARE LIBRE

Investigación

Fecha de recepción:
6 de junio de 2013

Fecha de aprobación:
30 de junio de 2013

Byron Cuesta Quintero
M.Sc. en Software Libre
Grupo de Investigación en Pensamiento Sistémico, Escuela de Ingeniería de Sistemas
Universidad Autónoma de Bucaramanga
bcuesta@unab.edu.co

Jorge Andrick Parra Valencia
Ph.D. en Ingeniería de Sistemas
Grupo de Investigación en Pensamiento Sistémico, Escuela de Ingeniería de Sistemas
Universidad Autónoma de Bucaramanga
japarra@unab.edu.co

Resumen

Este artículo presenta un conjunto de elementos pertinentes a los proyectos de software libre y a teorías de la cooperación, a partir de reconocer cómo desarrolladores distribuidos geográficamente en el mundo encuentran la motivación para crear comunidad alrededor del software libre y desarrollar software exitoso. También se estudió casos de éxito a nivel mundial para mostrar cómo los participantes, a través de una comunicación mediada por computador, resuelven conflictos de racionalidad individual y grupal usando escenarios de confianza y reciprocidad sobre el uso de recursos comunes que permiten la cooperación para el desarrollo y mantenimiento del software libre.

Palabras Clave

Cooperación, Desarrolladores, Sistemas, Software Libre.

Abstract

This paper presents a set of elements for the relevant cooperation to free software projects and theories of cooperation, from geographically distributed developers recognized as the world find the motivation to create community around free software and successful software development. Successful cases worldwide was also studied to show how participants through a

computer-mediated communication to resolve conflicts using individual and group rationality scenarios of trust and reciprocity on the use of common resources that allow cooperation for development and maintenance free software.

Key Words

Cooperation, Developers, Systems, Free Software.

Introducción

El desarrollo de proyectos de software libre se realiza en un ambiente distribuido de cooperación en red, donde la mayoría de los desarrolladores son voluntarios que encuentran una motivación principalmente filosófica en pasar su tiempo desarrollando software libre. (González Barahona, J., et al. 2003). De esta manera, se forman comunidades globales organizadas por desarrolladores que contribuyen con horas de trabajo representadas en esfuerzos individuales que suman en el desarrollo del software y que utilizan la comunicación mediada por computador, con el uso de herramientas que hacen efectiva esa colaboración como las listas de correo, el IRC (Internet Relay Chat) y los sistemas de control de versiones (CVS). (Markus et al., 2000).

La disponibilidad de código fuente y la interactividad que puede tener un desarrollador

al comunicarse mediante el uso de sistemas y herramientas telemáticas representan el éxito de los proyectos de software libre. Esto lo afirma (Raymond, 1999), en "La catedral y el bazar" y además explica como en el modelo de desarrollo libre con respecto al propietario se considera a los usuarios desarrolladores y así, puede darse el caso de que algún desarrollador encuentre un error y lo solucione o que una persona diferente a quien descubra el error sea la que lo entienda y corrija.

El software libre se asocia a un bien común en el que la acción colectiva representada en el esfuerzo de muchos desarrolladores está presente para su creación ó mantenimiento (Schweik & English, 2007). Por ser un bien común no se puede excluir a un individuo de los beneficios del bien, así tenga poca ó nada de motivación por contribuir voluntariamente en su producción ó mantenimiento. Estos individuos se enfrentan a situaciones de cooperación social y se les conoce como free-riders (Ostrom, 1990). En este contexto los free-riders son programadores, probadores o documentadores que utilizan software libre, pero se enfrentan a situaciones donde no contribuyen en su mantenimiento. Estas situaciones de racionalidad individual y colectiva es lo que se denomina un dilema social, siendo uno de los ejemplos más famosos de modelos de dilemas sociales llamado "la tragedia de los comunes" (Hardin, 1968). Hardin describe a un grupo de pastores que comparten una parcela como recurso común y que al tomar decisiones individuales en poner más vacas de las posibles pueden llevar al peor resultado para todos. Si se contextualiza con el software libre, no sería un problema de sobre apropiación de recursos comunes, sino con el hecho de que un proyecto de software libre quede abandonado por falta de recursos humanos disponibles para desarrollar y mantener el software (Schweik, 2005).

Metodología

Las comunidades de software libre entienden que el acceso al código fuente de forma libre representa el punto más alto en la creación del software, permitiendo a los desarrolladores bajo un sistema de licenciamiento abierto estudiarlo, modificarlo e inclusive distribuirlo con las mejoras realizadas bajo los mismos términos del software original u otros. (Stallman, 2002), (Perens, 1999). El código fuente es realizado por personas y

como tal es un producto de la mente humana, lo que se denomina la propiedad intelectual, pero esto no es un obstáculo para que en las comunidades los desarrolladores sean capaces de compartir el software dejando a un lado las acciones individuales para tomar acciones en las que todos cooperan para un beneficio común. (De Joode et al., 2003).

En la participación de proyectos de desarrollo de software libre, se encuentran desarrolladores inexpertos y expertos. Generalmente trabajan voluntariamente, pero también pueden recibir pago de empresas que patrocinan el desarrollo de software libre. Los inexpertos, se encuentran motivados por adquirir habilidades a través de reconocer el código fuente de otros y producir nuevo código que es sometido a un proceso de revisión y de esa evaluación demostrar competencias que le permitan relacionarse y conseguir de a poco reputación. (Feller & Fitzgerald, 2001). Los expertos, hacen parte de un grupo importante y reducido de desarrolladores que contribuyen en el desarrollo de la mayor parte del código principal de los proyectos. Esto se puede evidenciar en proyectos como el Kernel de Linux (Dempsey, et. AL., 1999) y GNOME (Koch & Schneider, 2002).

Se han realizado estudios utilizando Sourceforge.net, que es un sitio que aloja el más grande repositorio de software libre, donde se demuestra que el tamaño de los equipos de desarrollo de software libre generalmente son pequeños, lo que facilita la cooperación (Schweik & English, 2007).

En el desarrollo de proyectos de software libre también se presentan situaciones en la que algunos participantes quieren sacar provecho de los aportes realizados por otros participantes sin contribuir al mantenimiento del proyecto del software en curso. A estos participantes se les denomina free-riders y se caracterizan por no querer contribuir con tiempo y recursos al proyecto, lo que repercute en el éxito o abandono de los participantes del proyecto. (Baldwin & Clark, 2005).

Teorías de la Cooperación

Los dilemas sociales son el estudio de la tensión entre el individuo y la racionalidad colectiva, en

donde el comportamiento de una persona afecta la situación de otras personas. (Kollock, 1998). Kollock da una categorización a los dilemas sociales. Los dilemas donde la desertión está por encima de la cooperación y es la estrategia dominante para cada uno de los actores. Los dilemas donde la cooperación es mutua, que es un equilibrio óptimo, y la desertión es mutua, que es un equilibrio deficiente, siendo la clave el saber si podemos confiar en los demás y los dilemas en donde cada persona intenta no desertar y esperar a que el otro coopere para obtener el mejor resultado. Estos dilemas son altamente expuestos en estudios de cooperación y particularmente en la teoría del dilema del prisionero. (Axelrod, 1984)

Si un individuo contribuye en la construcción de un bien público se puede enfrentar a una situación en la que otros no contribuyan y si quieren sacar provecho del bien (free-riders). De esta manera los individuos buscan garantías en sus contribuciones para evitar que no haya cooperación (Ostrom, 2001).

El estudio de los dilemas sociales muestra como las personas pueden querer obtener el mejor resultado posible sin cooperar. Igualmente, si los miembros de un grupo contribuyen aun bien público que beneficie a todos replica en una de identidad de grupo y en una interdependencia basada en confianza que representa cooperación entre los sujetos. (Walker & Ostrom, 2007)

La figura 1, muestra cómo se pueden alcanzar niveles altos de cooperación entre un grupo de individuos que propenden por el bien colectivo antes que el individual a partir de relacionar tres condiciones iniciales: La confianza, que representa las expectativas que los individuos tienen sobre el comportamiento de los demás; La reciprocidad, que son las normas que los individuos aprenden de la socialización y las experiencias de la vida y la reputación en como los individuos crean identidades que proyectan sus intenciones y normas. (Ostrom, 1997).

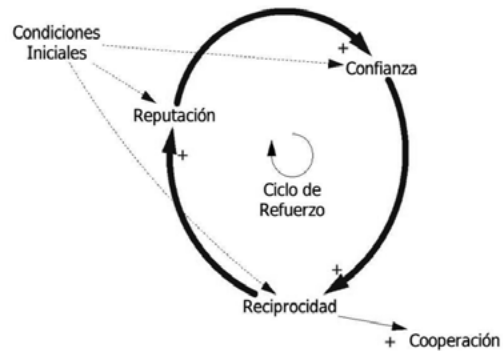


Figura 1. Condiciones iniciales para la cooperación Fuente: (Parra, 2010).

Cooperación en el Desarrollo de Software Libre

El software libre se desarrolla bajo un modelo en el que al compartir el código fuente, los desarrolladores cooperan a partir de una rigurosa revisión por pares, donde se aprovecha la depuración en paralelo que genera como resultado la innovación y el rápido avance en el desarrollo de productos de software (Hertel & Niedner & Herrmann, 2003).

La tabla 1, muestra que los repositorios de los proyectos de software libre representan una fuente importante para reconocer el aporte de los participantes en el desarrollo del software. Una de las métricas utilizadas para verificar el aporte de un desarrollador es el número de líneas de código (LOC) agregadas a un archivo. De esta manera, los estudios demuestran que proyectos como el Kernel de Linux, Gnome y Apache, fueron desarrollados por pequeños grupos de programadores que lograron una fácil comunicación y cooperación, siendo estos responsables de la mayor parte del código desarrollado en el proyecto. (Dempsey & Weiss & Jones & Greenberg, 1999), (Koch & Schneider, 2002), (Mockuset al., 2000).

Elliott & Scacchi muestran un estudio sobre la resolución de conflictos en las comunidades

virtuales de proyectos de software libre, a partir de los valores del trabajo cooperativo y las creencias en el software libre. Para este caso se analizaron los registros de mensajes IRC que están representados en resúmenes (kernelcousins) de las discusiones de los desarrolladores sobre dos casos: el primero, el conflicto y debate sobre el uso de un gráfico no diseñado con herramientas libres y el segundo el uso de una herramienta no libre para la documentación. En ambos casos demuestran cómo se resuelven los conflictos a través de comunicaciones mediadas por computador y fundamentadas en la identidad de grupo de la comunidad virtual (Elliott & Scacchi, 2004).

PROYECTO DE SOFTWARE LIBRE	DESCRIPCIÓN DE ELEMENTOS DE COOPERACIÓN
LINUX	Linux es un núcleo libre de un sistema operativo basado en Unix. El desarrollo se basaba principalmente en una lista de correo. En la lista de correo no sólo se discutía, sino que también se desarrollaba. Así, la vida del proyecto se ve reflejada en la lista, y los parches se envían a la misma. En contra de lo que uno pudiera esperar que se enviara el parche como adjunto, Linux prefiere que se mandara el código en el cuerpo del mensaje para que él y los demás pudieran comentarlo. (González Barahona, J. et al., 2003). Otro estudio es el basado en el repositorio bibliográfico (formalmente SunSITE.unc.edu y MetaLab.unc.edu) donde se analizan los archivos de metadatos (Linux Software Maps (LSM)) generados e información demográfica de los participantes, donde se evidencia como los desarrolladores colaboran entorno a una comunidad y se demuestran los esfuerzos globales de cooperación de los participantes para construir el código. (Dempsey & Weiss & Jones & Greenberg, 1999).
GNOME	GNOME es un proyecto de software construido para un entorno de escritorio para usuarios finales. De este proyecto Koch & Schneider analizaron las asociaciones entre los programadores y los archivos en el repositorio CVS, y explican el bajo nivel de cooperación sobre los archivos básicos. Para los archivos más grandes, más programadores activos hacen un mayor porcentaje de su trabajo y dentro de este círculo reducido de personas, la cooperación es posible. (Koch & Schneider, 2002), (Koch & Schneider, 2002b), (Germán, 2002).
APACHE	El servidor web Apache utilizó un modelo donde la mayoría de los módulos, fueron escritos por un grupo central correspondiente a 15 desarrolladores que contribuyeron y compartieron el código, con el fin de evitar hacer cambios conflictivos en el mismo. De esta manera en lugar de que un solo individuo escribiera todo el código para un módulo dado, el grupo central tuvo un nivel suficiente de confianza mutua para realizar contribuciones en varios módulos de código según fuera necesario. (Mockuset al., 2000), (Roberts, J. et al., 2006).
GNU Enterprise (GNUe)	GNUe es una organización virtual para el desarrollo de software libre de aplicaciones de negocios para planificación de recursos empresariales (ERP). Elliott & Scacchi, muestran que en el proyecto GNUe la comunicación mediada por ordenador con elementos como: Internet Relay Chat (IRC), el sistema de control de versiones (CVS) y las listas de correo, facilitan el trabajo en equipo, la resolución de conflictos y cooperar para construir comunidad. Así, mismo muestran que la construcción de una comunidad en línea mediante una cultura organizacional sugiere principios de diseño, ya que los participantes no tienen la oportunidad de convivir cara a cara en su vida laboral cotidiana y muestran que los conflictos se resuelven simultáneamente a través de IRC, facilitando el trabajo en equipo ya que la gente al mismo tiempo podría cooperar para resolver un problema técnico. (Scacchi, 2003), (Elliott & Scacchi, 2004).
FreeBSD	FreeBSD es un sistema operativo derivado de BSD Unix. Es el resultado del desarrollo de software de código abierto y el mantenimiento esfuerzo realizado por más de 600 personas ubicadas en todo el mundo. (Feller & Fitzgerald, 2003). El modelo está fuertemente basado en dos herramientas: el sistema de versiones CVS y el sistema de informe de error GNATS (GNU project's issue-tracking software). Alrededor de estas dos herramientas gira todo el proyecto, como se puede comprobar por el hecho de que se ha creado una jerarquía a partir de las mismas. Así, sobre los comitters, aquellos desarrolladores con derecho a escritura al CVS es sobre quien recae toda la soberanía del proyecto (González Barahona, J. et al., 2003).

Tabla 1. Elementos de cooperación en proyectos de Software Libre

Resultados

La tabla 2 muestra el resultado asociado a los elementos basados en las teorías de la cooperación y la cooperación en el software libre, donde se comparan y se describen en detalle cada uno de ellos de forma que se señalan las similitudes con respecto al problema de la cooperación y la resolución de conflictos en el desarrollo de software libre. Este resultado sigue la línea de investigación expuesta en (Cuesta & Parra, 2013).

TEORÍAS DE LA COOPERACIÓN	COOPERACIÓN EN EL SOFTWARE LIBRE	DETALLE
Evitar individuales. (Axelrod, 1984), (Kollock, 1998), (Kollock, 1999).	estrategias de las listas de correo. (Kollock & Smith, 1996).	Lograr que todos obtengan un beneficio positivo si se coopera en el uso de recursos comunes
Bienes comunes. (Kollock, 1998).	Código fuente como un recurso común. (González Barahona, et al., 2003).	Cooperar para disfrutar del bien por igual y hay un beneficio inmediato que produce un costo compartido por todos
Desarrollar una identidad de grupo. (Kollock, & Smith, 1996), (Ostrom, 2001).	Comunidades virtuales. (Elliott & Scacchi, 2004), (Dempsey & Weiss & Jones & Greenberg, 1999).	Creación y refuerzo de las normas compartidas sobre intereses y recursos comunes
Crear grupos pequeños. (Mockuset al., 2000), (Ostrom, 2001).	Trabajo en grupos pequeños. (Koch & Schneider, 2002).	Permiten una fácil comunicación y la cooperación.
Comunicar ideas cara a cara. (Ostrom, 1997).	Comunicación mediada por computador. (Feller & Fitzgerald, 2003), (Elliott & Scacchi, 2004).	Para encontrar intereses comunes y evitar que no haya cooperación cuando se trabaja en la construcción de un bien público.
Promover la reciprocidad para obtener reputación y generar confianza en grupo. (Rheingold, 1993), (Ostrom, 1997).	Cambiar roles de los participantes continuamente. (Mockuset, et al., 2002).	Compromiso mutuo, para que los individuos cooperen
Free-riders. (Ostrom, 1990), (Ostrom, 2001).	Uso del software sin realizar ningún esfuerzo por contribuir a la comunidad. (Elliott & Scacchi, 2004)	Si un individuo contribuye en la construcción de un bien público se puede enfrentar a una situación en la que otros no contribuyan y si quieren sacar provecho del bien

Tabla 2. Elementos comunes para las teorías de cooperación y cooperación en software libre

El conjunto de elementos de cooperación que son el resultado de la tabla 2, muestra los aportes de diferentes autores sobre cómo los individuos se motivan para contribuir en el desarrollo de un bien común y como las comunidades virtuales de software libre definen comportamientos para cooperar y resolver conflictos. De esta manera, se reconoce que la existencia de un conflicto sobre el mantenimiento de recursos comunes se representa en aquello que es bueno para el individuo y lo que es bueno para el grupo (Axelrod, 1984). La posibilidad que una persona este motivada a contribuir a un grupo sin recibir nada a cambio, es una motivación prevista por la reciprocidad que se espera del grupo en una cooperación generalizada sobre un bien público (Kollock, 1999). Las contribuciones de los individuos al grupo se incrementan en la medida en que la contribución es visible para la comunidad, con lo cual se gana prestigio y reconocimiento. Esta reputación se basa en la reciprocidad, la interacción continua, la identidad de grupo y los límites de las comunidades en línea (Rheingold, 1993). Para (Ostrom, 1997), existen condiciones iniciales que al conjugarlas se puede alcanzar niveles altos de cooperación con beneficios para todos los individuos de un grupo a partir de la reciprocidad que genera reputación y la confianza para colaborar mutuamente (Figura 1).

La tabla 1 representa elementos de cooperación en proyectos de software libre, el principal elemento es el acceso libre al código fuente y como a partir de este recurso común se construye comunidad sobre Internet (González Barahona, J. et al, 2003). Para construir comunidad es necesario definir una identidad de grupo a partir de normas donde se establecen límites para los participantes (Kollock, & Smith, 1996), en donde se pueden resolver conflictos como lo demuestran estudios basados en repositorios sobre discusiones relacionadas en los proyectos donde los desarrolladores muestran actividad en listas de correo e IRC (Elliott & Scacchi, 2004). La cooperación alcanza mayores niveles en grupos pequeños de desarrollo sobre los archivos grandes, debido a que se facilita la comunicación (Koch & Schneider, 2002), (Mockuset al., 2000). Los resultados sugieren un conjunto de elementos que demuestran la importancia de la cooperación en el desarrollo de software libre.

El software libre se desarrolla a partir de un grupo de personas conectadas en red con intereses comunes alrededor de un recurso público, que permite encontrar la interacción repetida con los demás y ganar reputación a partir de normas de confianza para obtener un beneficio colectivo en un esfuerzo global de cooperación. (Kollock & Smith, 1996), (Feller & Fitzgerald, 2003).

El tamaño del grupo de desarrollo de los proyectos de software libre repercute en la facilidad para que sus participantes cooperen, siendo más difícil en la medida en que aumenta el tamaño. Al ser más grande, crecen los costos en una comunicación mediada por computador, en donde la resolución de conflictos puede generar en el abandono del proyecto (Ostrom, 2001), (Koch & Schneider, 2002).

Se evidencia que las comunidades virtuales de software libre han podido desarrollar y mantener con éxito muchos proyectos a pesar de la tentación de los free-riders en poco ó ningún esfuerzo por contribuir a la comunidad (Ostrom, 1990), (Elliott & Scacchi, 2004).

Referencias Bibliográficas

Axelrod, R. (1984). *The Evolution of Cooperation*, Basic Books, NY.

Cuesta, B. & Parra, J. (2013). *Curso E-learning para el mejoramiento de las competencias de cooperación en el desarrollo de proyectos de software libre orientado a los lineamientos metodológicos de la Programación Extrema*. Tesis de Maestría, Universidad Autónoma de Bucaramanga Colombia. Maestría en Software Libre.

De Joode, V. & De Bruijn; R. & Van Eeten, M. (2003). *Protecting the Virtual Commons: Self-organizing open source communities and innovative intellectual property regimes*, Information Technology & Law Series, T.M.C. Asser Press, The Hague.

Feller, J. & Fitzgerald, B. (2001). *Understanding Open Source Software Development*. Addison-Wesley, Reading, MA.

González Barahona, J. (2003). *Introducción al software libre: El desarrollador y sus motivaciones*. Universitat Oberta de Catalunya. pp. 91 - 245.

Hardin, G. (1968). *The Tragedy of the Commons*. Science, New Series, Vol. 162, No. 3859. pp. 1243-1248.

Koch, S. & Schneider, G. (2002). *Effort, co-operation and co-ordination in an open source software project: GNOME*, Information Systems Journal 12(1), pp. 27-42.

Koch, S. & Schneider, G. (2002b). *Results from software engineering research into open source development projects using public data*, gkeitsfeld Informationsverarbeitung und Informationswirtschaft, En: H.R. Hansen, W.H. Janko (Hrsg.), Wirtschaftsuniversität Wien (núm. 22), 2002.

Kollock, P. & Smith, M. (1996). *Managing the Virtual Commons: Cooperation and Conflict in Computer Communities*. Pp. 109-128 in *Computer-Mediated Communication: Linguistic, Social, and Cross-Cultural Perspectives*, edited by Susan Herring. Amsterdam: John Benjamins.

Kollock, P. (1998). *Social dilemmas: The anatomy of cooperation*. Annual Review of Sociology, 24 (1), pp. 183-214.

Kollock, P. (1999). *"The Economies of Online Cooperation: Gifts and Public Goods in Cyberspace"*. In Smith, Marc; Kollock, Peter. *Communities in Cyberspace*. London: Routledge. pp. 220-239

Markus, M. & Manville, B. & Agres, C. (2000). *What Makes a Virtual Organization Work?*, Sloan Management Review, 42 (1) 13-26.

Ostrom, E. (1990). *Governing the commons: The evolution of institutions for collective action*. Cambridge University Press.

Ostrom, E. (1997). *A Behavioral Approach to the Rational Choice Theory of Collective Action: Presidential Address, American Political Science Association*. The American Political Science Review, Vol. 92, No. 1. pp. 1-22.

Ostrom, E. (2001). *Social dilemmas and human behavior*. Economics in Nature. Social Dilemmas, Mate Choice and Biological Markets. pág. 23-41.

Perens, B. (1999). *The Open Source Definition*, in *Open Sources: Voices from the Open Source Revolution*, DiBona, C., S. Ockman and M. Stone (Eds) O'Reilly & Associates, Sebastopol, pp. 171-189.

Raymond, E. (1999). *The Cathedral and the Bazaar*. Knowledge, Technology & Policy, Fall 99, Vol. 12 Issue 3. pp. 23-49.

Schweik, C. (2005). *"An Institutional Analysis Approach to Studying Libre Software "Commons": Upgrade"* The European Journal for the Informatics Professional.

Schweik, C. & English, R. (2007). *Tragedy of the FOSS Commons? Investigating the Institutional Designs of Free/Libre and Open Source Software Projects*, IEEE Computer Society Washington, DC, USA.

Stallman, R. (1998). *"The GNU Operating System and the Free Software Movement"*, in *Open Sources: Voices from the Open Source Revolution*, C. DiBona, S. Ockman, and M. Stone, Eds.: O'Reilly Books, pp. 53-70.

Stallman, R. (2002). *Free Software, Free Society: Selected Essays of Richard M. Stallman*, GNU Press, Boston

Walker, J. & Ostrom, E. (2007). *Trust and reciprocity as foundations for cooperation: Individuals, institutions, and context*. In *Capstone Meeting of the RSF Trust Initiative at the Russell Sage Foundation*.