



Cómo citar el artículo

Serna Montoya, E.; Andrés Galvis, J. & Ortiz, I. M. (2014). Principios estratégicos del software libre y su relación con la reingeniería de procesos. Reflexión pedagógica. *Revista Virtual Universidad Católica del Norte*, 42, 197-206. Recuperado de <http://revistavirtual.ucn.edu.co/index.php/RevistaUCN/article/view/504/1043>

Principios estratégicos del software libre y su relación
con la reingeniería de procesos

Strategic Principles of Free Software and its Relation
with Process Reengineering

Principes stratégiques du logiciel libre et leur relation
avec la réingénierie des processus

Edgar Serna Montoya

Doctorando

Instituto Tecnológico Metropolitano

Grupo de investigación AEyCC

edgarserna@itm.edu.co

Medellín, Colombia

Jorge Andrés Galvis

Ingeniero de Sistemas

Universidad Nacional de Colombia

jagalvis@unal.edu.co

Medellín, Colombia

Iván Mauricio Ortiz

Ingeniero de Sistemas

Desarrollador independiente

maor474@gmail.com

Medellín, Colombia

Tipo de artículo: Investigación e innovación

Recibido: 14 de febrero de 2014

Evaluado: 21 de abril de 2014

Aprobado: 28 de abril de 2014

Resumen

En las últimas décadas las empresas han tenido que reestructurar sus procesos para mantenerse competitivas en el mundo globalizado. Para lograrlo tuvieron que aplicar estrategias que les permitieran alcanzar el desarrollo mínimo de adaptación, y de esta manera hacer frente a los cambios que el mercado les obliga, lo que las ha llevado a utilizar la reingeniería de procesos como alternativa de sostenibilidad. En este trabajo se describen algunas estrategias basadas en software libre que se pueden aplicar en la reingeniería de procesos para alcanzar ese objetivo. En dicho proceso se tiene en cuenta que el surgimiento del software libre permitió que las empresas incursionaran en nuevas formas de hacer esta reingeniería, ya que mediante el trabajo colaborativo redujeron los costos y el tiempo de muchos de sus procesos.

Palabras clave

Estrategias de software libre, Reingeniería de procesos, Reutilización, Trabajo colaborativo.

Abstract

In the last decades, companies have been forced to restructure their processes to remain competitive in a globalized world. To achieve this they had to use strategies which allowed them to reach the minimum adaptation development and in this way they could face changes imposed by the market. This has led companies to use process reengineering as a sustainability alternative. In this work we describe some strategies based on free software which can be applied in process reengineering

to achieve this goal. In this process is taken into account the emergence of free software which has allowed companies venturing into new ways of performing this reengineering because, by means of collaborative work, they reduced time and costs for many of their processes.

Keywords

Process reengineering, free software strategies, reuse, collaborative work.

Résumé

Pendant les dernières décennies les entreprises ont restructuré leur processus pour être compétitive dans le monde globalisé. Pour réussir ca ils ont appliqué des stratégies qui rendent possible pour eux réussir le développement minimal d'adaptation et par conséquent d'affronter les changements imposés par le marché, ce qui a résulté en la utilisation de la réingénierie des processus comme alternative de soutenabilité. Dans ce travail on décrit quelques stratégies en se basant sur logiciel libre, qui peuvent être appliquées dans la réingénierie des processus pour réussir cet objectif. Dans ce processus on considère que l'apparition du logiciel libre a permis que les entreprises rentrent dans nouvelles formes de faire cette réingénierie, étant donné que le travail collaboratif a réduit les coûts et le temps pour beaucoup de ses processus.

Mots-clés

Réingénierie des processus, stratégies de logiciel libre, réutilisation, travail collaboratif.

Introducción

Se considera ampliamente que la reingeniería de procesos es una herramienta de gestión, con la cual es posible rediseñar las estructuras organizacionales, en muchos casos desde cero, con el objetivo de mejorar la eficiencia, y que supone una tendencia para empresas menos jerarquizadas y basadas en el trabajo en equipo. Para realizar esta reingeniería se requiere aplicar uno o varios productos software, cuyo objetivo se oriente a mejorar los procesos involucrados (Pintendi, 2012). Entre las diversas alternativas se encuentra el software libre, que por su modelo de desarrollo y filosofía de distribución se convierte en una opción estratégica para el desarrollo tecnológico de las organizaciones, y consecuentemente para la reingeniería de sus procesos.

Las empresas que sobrevivieron a la crisis económica, petrolera y tecnológica de los años 80, y aquellas que tuvieron que enfrentar las revoluciones del siglo pasado, comprendieron que tenían que modificar sus estructuras para que en el futuro no fueran sorprendidas por situaciones semejantes. Una de las alternativas a las que debieron recurrir fue la de realizar reingeniería de procesos, lo que originó nuevas y revolucionarias formas de ver el presente y visionar el futuro. De esta forma lograron la competitividad que la revolución de internet y el mundo globalizado trajeron al inicio del siglo XXI.

Los principios del software libre fueron una opción, una filosofía de trabajo que presentaba para entonces un aceptable nivel de madurez, y que además se ofrecía de forma abierta para todos los interesados. Entre sus principios se encuentra el trabajo colaborativo, la comunidad abierta y el acceso a las experiencias y estudios de caso, todos posibles de implementar en un proyecto de reingeniería de procesos que sirva como alternativa a los establecidos desde otras ideologías relacionadas.

En este artículo se presenta la descripción de algunos de estos principios, y se describe la importancia de aplicarlos en la reingeniería de procesos en las organizaciones. En la primera parte del documento se presenta la metodología utilizada, luego se anotan las definiciones y alcances de la reingeniería de procesos, posteriormente se describen y analizan algunos principios estratégicos del software libre y su utilización en la reingeniería de procesos, y finalmente se presentan las conclusiones y las referencias que sustentan este trabajo.

Metodología

Para identificar los principios generales del software, y analizar su posible utilidad en la reingeniería de procesos, se hizo una búsqueda en la literatura; un proceso que incluyó la selección de las áreas y palabras clave, de las fuentes a las cuales consultar o contactar, y de la validez de los principios para la temática de trabajo. En la Tabla 1 se describen las áreas y palabras clave utilizadas en la consulta.

Tabla 1. Áreas y palabras clave de la búsqueda

Áreas	Palabras clave
Reingeniería de procesos	Implantación Ingeniería en Sistemas Computacionales Mejoramiento de la gestión Puestos de trabajo Competitividad Mercado
Software libre	Proyectos Empresas Principios Calidad Prospectiva Sostenibilidad en el tiempo
Estrategias del software libre	Reutilización Trabajo colaborativo Comunidad Acceso a experiencias Estudios de caso

Con los hallazgos de este proceso se hizo una compilación inicial para determinar la validez de los postulados y la adaptabilidad de los principios para la reingeniería de procesos. En las siguientes secciones se describe el resultado del análisis realizado.

Reingeniería de procesos

La reingeniería de procesos es una herramienta de gestión organizacional a la que varios investigadores han dado su propia interpretación y a la que han hecho aportes factibles de aplicar en las empresas. Davenport & Short (1990, p. 5) la describen como el análisis y el diseño de los flujos de trabajo y los procesos al interior y entre las organizaciones; Hammer & Champy (1993, p. 55) promueven el rediseño y proponen un replanteamiento radical a los procesos de negocio para alcanzar mejoras en costo, calidad, servicio y rapidez; Talwar (1993, p. 27) centra su trabajo en el replanteamiento, la reestructuración y la racionalización de la estructura del negocio, los procesos y los métodos de trabajo de los sistemas de gestión y sus relaciones exteriores, a través de lo cual es posible crear valor y entrega; Petrozzo & Stepper (1994, p. 138) consideran que implica el rediseño concurrente de los procesos, las organizaciones y quienes apoyan los sistemas de información, para lograr el mejoramiento en áreas como clientes, productos y servicios; Lowenthal (1994, p. 162) hace un replanteamiento al rediseño de los procesos operativos y a la estructura organizativa, centrándose en las competencias básicas de la organización para mejorar su desempeño; y Gadd & Oakland (1995, p. 20) afirman que para lograr los objetivos de la reingeniería de procesos es importante hacer referencia a los Sistemas de Información, porque aunque no dependen necesariamente de soluciones informáticas, se convierten en facilitadores para alcanzar las mejoras que buscan las empresas al rediseñar sus procesos.

Estrategias del software libre para la reingeniería de procesos

Procesos de Ingeniería de Software

Como se mencionó antes, el software libre es una de las opciones que tienen las empresas para realizar reingeniería de procesos, porque si el producto es libre puede trabajar con él, además de mejorarlo y adaptarlo a sus necesidades. Este principio no lo encuentran en otras opciones, porque cuando compra esos productos, su modificación puede hacer parte de los acuerdos con el fabricante (González et al., 2003, p. 58), además, no es fácil lograr el acceso a las fuentes en el nivel que ofrecen las herramientas libres. Otra cuestión es que no tienen perdurabilidad en el tiempo, o se incrementan los costos en cada versión, mientras que en el software libre las aplicaciones están disponibles para perdurar, debido a que el acceso a las fuentes facilita su mantenimiento y adecuación.

Por otro lado, generalmente el modelo de desarrollo en proyectos del software libre se estructura de manera más informal, porque el trabajo de los participantes es en su mayoría voluntario y sin recompensa económica directa, y los realizan con el afán de compartir y aportar sus conocimientos a través del trabajo colaborativo. Así mismo, la forma como se elicitan los requisitos depende tanto de la edad como del tamaño del proyecto, es decir, en las primeras etapas el creador del proyecto y el usuario suelen ser la misma persona; posteriormente, si el proyecto crece, este proceso se realiza mediante listas de correo, y se puede llegar a una clara distinción entre el equipo de desarrollo y los usuarios. Para proyectos grandes, con diversos usuarios y desarrolladores, la elicitación se realiza mediante la misma herramienta que utilizan para gestionar los errores del proyecto (González et al., 2003, p. 92).

El proceso del diseño es más global y sólo los grandes proyectos lo documentan de manera exhaustiva, mientras que para otros es probable que los desarrolladores principales sean los únicos que la posean a veces sólo en su mente, o que se adapte para versiones posteriores (González et al., 2003, p. 42). En los procesos tradicionales el diseño detallado no sólo implica limitaciones en cuanto a la reutilización de módulos, sino que tam-

bién sirve de soporte al momento de aceptar nuevos desarrolladores, pues éstos deben pasar necesariamente por un proceso de aprendizaje lento y costoso; por otro lado, está muy generalizado y su ausencia implica la pérdida de posibilidades de reutilización de código.

La implementación es la fase en la que se concentra el mayor esfuerzo por parte de los desarrolladores de software libre, y suelen aplicar el paradigma clásico de programación de prueba y error hasta que consiguen los resultados desde un punto de vista subjetivo. En el caso de proyectos grandes, como Mozilla, existen máquinas dedicadas exclusivamente a descargar los repositorios que contienen el código más reciente, para compilarlo en diferentes arquitecturas. Los errores se notifican a una lista de correos de desarrolladores, pero las pruebas automáticas no están muy arraigadas, porque generalmente son los usuarios quienes las realizan y presentan soluciones, lo que es ventajoso porque se reducen los costos.

El mantenimiento en el software libre también depende del proyecto, y en aquellos que requieren estabilidad, como núcleos de sistemas operativos, la estrategia es mantener versiones previas de los mismos, porque el cambio a una nueva puede resultar traumático. Por lo general, en la mayoría de estos proyectos no se corrigen los errores de las versiones previas, porque se recomienda utilizar versiones más recientes en las que posiblemente ese error ya no se encuentre; esto, teniendo en cuenta que los productos del software libre se piensan y diseñan de forma evolutiva (González et al., 2003, p. 94).

Las estrategias descritas hasta el momento se pueden aplicar en la reingeniería de procesos directamente, pero teniendo en cuenta aspectos como integración, adaptación a las necesidades reales de la empresa y migración, lo que implica analizar los aspectos de reemplazar un sistema por otro y capacitar a quienes lo utilizarán, debido a que un producto nuevo incluye necesariamente conocimientos, alternativas, herramientas y plataformas específicas. Por otro lado, las estrategias del software libre garantizan de cierta manera un ámbito de seguridad, que permite descubrir la vulnerabilidad mucho antes que en otros sistemas debido al trabajo colaborativo y a los estándares abiertos, los cuales inyectan mayor flexibilidad a los productos para adaptarlos a las nuevas necesidades.

Reutilización

El principio de la reutilización se refiere a implementar técnicas para garantizar que de un producto, proceso o desarrollo se pueda emplear parte o todo para la construcción de otro. Esto se puede aplicar en la reingeniería de procesos con el objetivo de reutilizar un trabajo anterior, de manera que se economice tiempo y costos. La decisión de cuándo utilizarla depende de los requisitos técnicos que deben cumplir los componentes, y de que el proceso sea económicamente razonable. Así mismo, debe pasar por las fases de prospección, adaptación y comparación (Neumann & Breidert, 2005, p. 48).

La reutilización podría contribuir a minimizar los recursos necesarios y el tiempo disponible en una reingeniería de procesos. Esto, teniendo en cuenta que las empresas destinan un amplio margen de tiempo para adaptar los productos a las nuevas necesidades (Hippel & Krogh, 2003, p. 212). Por otro lado, una de las características de este siglo es la inmediatez, lo que acrecienta la necesidad de economizar tiempo para desarrollar los proyectos. Si la reingeniería de procesos aplica la estrategia de reutilizar y tiene en cuenta las experiencias anteriores, propias o ajenas, sería posible economizar el tiempo suficiente para responder a las exigencias empresariales y sociales. Si el software libre puede entregar desarrollos constantemente, debido a que reutiliza código, procesos, procedimientos y personas, es posible que una reingeniería de procesos diseñada para reutilizar las experiencias anteriores, también pueda lograr que sus procesos se lleven a cabo en el tiempo y en los costos establecidos inicialmente, y que no se requiera prorrogas para entregar productos, que en la mayoría de ocasiones no cumplen con los requisitos del cliente.

Trabajo colaborativo

Las comunidades libres pueden ser grupos de usuarios, de desarrolladores, mixtos, de debates técnicos, de debates políticos, de organización y articulación o de personas que usan/developan un producto. Este es el caso de Mozilla y Mambo, en los que se reúnen desarrolladores, usuarios e interesados para debatir y perfeccionar cada herramienta en particular. Cuanto más visibles los proyectos, mayores serán las comunidades que se forman a su alrededor y más productivo será el trabajo colaborativo que cumplen. Aunque el papel central lo realizan los desarrolladores, también se suman los usuarios, quienes contribuyen para la evolución, la prueba, el mantenimiento y la distribución de los productos (Cortes, 2008, p. 39).

Los proyectos, en contextos diferentes al software libre, los realizan grupos de desarrolladores que están bajo contratos que les impiden la divulgación y el uso de la información relacionada con el producto en construcción. Esto se respalda en cuestiones como el secreto industrial y la propiedad intelectual. Los productos del software libre se desarrollan bajo estrategias de acceso abierto, el conocimiento se comparte y se distribuyen las versiones para su enriquecimiento desde las mismas comunidades. Uno de los desarrollos que más ha contribuido con este principio ha sido internet, porque proporciona la diseminación y expansión del trabajo colaborativo, y permite la creación de redes globales de usuarios, críticos y adeptos. También ha permitido el mejoramiento de las prácticas y las responsabilidades para los participantes, porque estas redes pueden ser sistemas complejos que se auto-organizan, o sistemas jerárquicos con reglas y niveles de acceso diversificado (Cortes, 2008, p. 57).

Tradicionalmente, en la reingeniería de procesos se exige inicialmente la actuación de un líder de proyecto, es decir, una persona con actuación rectora que posea la experiencia y el conocimiento del proceso, con el fin de decidir cuáles apoyos necesita para llevar a cabo lo establecido, pero teniendo en cuenta las prioridades y los rumbos específicos. En un ambiente de comunidad libre en línea, el líder infunde entusiasmo a los demás miembros, lo que contribuye a una actitud positiva que convoca a la participación desinteresada y mayoritaria. Los desarrolladores voluntarios trabajan por el logro de objetivos, los retos y el reconocimiento y posicionamiento en la comunidad. Esta especie de *status* motiva a que otros quieran lograr igual posición, y de esta forma su trabajo se convierte en un proceso de calidad constante para alcanzarla (Thompson, 2005, p. 160). En los procesos de software libre los méritos de los participantes tienen importancia, y aunque el concepto varía de una comunidad a otra generalmente involucra cuestiones como cantidad y calidad de código contribuido, sugerencias, participación activa, coherencia y opiniones constructivas en los debates.

En la reingeniería de procesos, un modelo como este podría lograr que muchas más personas se sumen al proyecto, que trabajen con mayor dedicación y responsabilidad, y que ninguno quiera desarrollar sus funciones individualmente, lo que redundaría en reducción de tiempo y costos, a la vez que se incrementa la calidad y la seguridad de los sub-productos. Se trata de una forma de organización que aprovecha economías de escala y de alcance, y que permite una corrección rápida de fallas, pues los entregables pueden ser inspeccionados públicamente, y las evaluaciones a las que se exponen colaboran a lograr la fiabilidad y credibilidad necesarias dentro del equipo de trabajo. Una característica adicional en este aspecto es la posibilidad de realizar alteraciones específicas de acuerdo con las necesidades individuales del usuario, lo que genera diversas versiones personalizadas que atienden perfectamente a cada característica demandada (Cortes, 2008, p. 98).

El desarrollo de las redes permite una comunicación global para apoyar la interacción social, la cooperación, la colaboración para el aprendizaje y la construcción y distribución del conocimiento (Friedman, 2005, p. 39), y de acuerdo con la filosofía del software libre, unirse a una comunidad en línea implica estar comprometido a participar e incorporar conocimiento formal para integrarlo a la práctica informal. Esto sucede con frecuencia en las comunidades que desarrollan y hacen soporte de software de código abierto (Lakhani & Hippe, 2003, p. 939). Para un proceso de reingeniería esta estrategia sería importante, porque los integrantes del proyecto tendrían un incentivo adicional al salarial, es decir, capacitación constante y mejoramiento de sus competencias relacionadas. Al respecto, se pueden mencionar las comunidades de práctica, que son grupos que comparten una preocupación, un conjunto de problemas o una pasión sobre un tema, y que profundizan su conocimiento y

pericia en esta área interactuando de forma permanente (Lave & Wenger, 1991, p. 118). Los miembros se reúnen regularmente para compartir información, conocimiento y asesoramiento, ya sea en comunidades pequeñas o grandes, de larga o corta duración, no reconocidas o institucionalizadas. Para que existan estas comunidades se requieren tres elementos: 1) un dominio del conocimiento, porque esto les permite reconocer la importancia de la comunidad y motiva a los miembros a participar, 2) una comunidad, que crea relaciones entre los miembros con base en el respeto mutuo y la voluntad de compartir ideas y experiencias y 3) una práctica, es decir, un conjunto de herramientas, términos, actividades y documentos compartidos por los miembros. A menudo, estas comunidades de práctica se basan en un modelo de aprendizaje determinado, lo que podría enriquecer la reingeniería de procesos, teniendo en cuenta que los objetivos se lograrían con mayor eficiencia y eficacia.

Otro principio de estas comunidades es que los miembros son coinvestigadores, coaprendices y compañeros de los líderes, quienes escuchan e infunden el respeto mutuo. A diferencia de otros modelos, los participantes no esperan a que el líder tome su papel como tal, porque entre sus iniciativas y responsabilidades pueden asumir el rol que mejor aporte a las soluciones y al equipo. Mientras en la reingeniería de procesos tradicional a los participantes se les asignan roles directamente, por lo que se asemejan a autómatas que responden a un código programado, en la estrategia del software libre a los integrantes del proyecto se les permite el desarrollo de capacidades no visibles, lo que genera una especie de deseo de superación constante para alcanzar las metas del proceso (Forte & Bruckman, 2006, p. 186).

El nivel de formación de los integrantes de las comunidades de trabajo colaborativo es comparable al de cualquier grupo de investigación en una universidad, y de acuerdo con Friedman (2005, p. 73), de un total de ocho comunidades en proyectos de software libre encuestadas, el 83% tenía al menos *título de pregrado*. Estos resultados son consistentes con los resultados en otros estudios (Hars & Ou, 2002, p. 30), en los que también se indaga por cuestiones como *tiempo de participación*, 61% por lo menos tres años; *aportes en grandes proyectos*, 43% había contribuido a proyectos como Apache, Debian, Drupal, Gentoo, Joomla, SeaMonkey, Thunderbird, Ubuntu y Linux; *nivel de satisfacción*, 70% informaron que contribuir a un proyecto era gratificante o muy gratificante; *funciones realizadas*, 37% desarrolladores de software, 23% profesores, 17% consultores, 15% jefes de proyecto y 6% estudiantes. Además, una tercera parte desarrolló las funciones como parte de su empleo y los demás lo hicieron de forma voluntaria. Así mismo, se indagó por las actividades realizadas en el proyecto: escribir y depurar código, editar código, apoyar a los usuarios a través de grupos de noticias, listas de correo o foros de discusión, traducir, documentar, corregir pruebas, controlar calidad, realizar pruebas de usabilidad, diseñar nuevos módulos/funcionalidades, actualizar páginas web, gestionar de proyectos y recaudar fondos.

Un proceso de reingeniería se podría enriquecer ampliamente si el trabajo colaborativo se implementara como filosofía para llevar a cabo las diferentes actividades programadas. De esta manera, el trabajo se lograría en menos tiempo y los costos rebajarían, los participantes lograrían mayor satisfacción en su labor, la empresa estructuraría una gestión más ambiciosa, el logro de prospectiva no sería un inconveniente, y la competitividad que requiere la organización para mantenerse en el mercado no sería difícil de lograr.

Conclusiones

La filosofía de apertura del software libre favoreció los procesos centrados en el capital humano, los conocimientos, la personalización y la adecuación de los productos, así como su evolución constante. En este sentido, sobresale el papel que juega la comunidad, que vela por la calidad y por la evolución de los productos. Esta iniciativa ha venido consolidando una propuesta que enmarca la mayoría de sectores con presencia de software, ofreciendo un mercado maduro, cualitativo y seguro sobre el cual presentar una estrategia empresarial al momento de hacer una reingeniería de procesos, tanto para el desarrollo de software como para los servicios complementarios.

Los principios estratégicos del software libre que se describen en este documento son un aporte de esta filosofía, y están disponibles para ser adaptados en cualquier proceso que los requiera. Una de las áreas en las que es posible integrarla es en la reingeniería de procesos, pues su finalidad es particularmente semejante a la que se busca en proyecto de software libre. Los procesos se podrían enriquecer y el logro de objetivos se alcanzaría en tiempos eficaces y costos racionales.

Implementar estas estrategias no debería ser complicado, porque sólo se requiere libertad de acción para que los integrantes se apropien de sus roles y apoyen el proceso de reingeniería. En la Tabla 2 se hace un resumen de las ventajas de las estrategias del software libre analizadas en este artículo.

Tabla 2. Ventajas de las estrategias del software libre

Estrategia	Ventajas
Procesos de Ingeniería de Software	Probados y experimentados Soportados en productos de calidad Fáciles de reproducir y adaptar Ampliamente documentados Actualización constante Discusión permanente Amplia contribución y participación
Reutilización	Ahorro de tiempo y costos Sustentada en experiencia previa Conocimientos documentados Errores depurados Flexibilidad Acceso a las fuentes
Trabajo colaborativo	Número de participantes Globalización Integración Roles definidos o propuestos Participación voluntaria Recompensas personales Conocimiento Experiencia compartida Soluciones a tiempo

Referencias

Debido a la temática de la investigación y a los objetivos trazados en la misma, los autores se enfocaron en analizar las publicaciones inmediatamente posteriores a la época dorada del inicio de las comunidades del software libre, y no incluyeron las referencias de períodos más recientes porque al analizarlas no ofrecían el aporte que buscaban.

Cortes, L. (2008). Ingeniería en Sistemas Computacionales. México, Tecnológico de Estudios Superiores del Oriente del Estado de México.

Davenport, T. & Short, J. (1990). The new industrial engineering: Information technology and business process redesign. *Sloan Management Review*, 31(4), 11-27.

Forte, A. & Bruckman, A. (2006). From Wikipedia to the classroom: Exploring online publication and learning. *Proceedings of the 7th international conference on learning sciences*, pp. 182-188, Indiana.

- Friedman, T.L. (2005). *The world is flat: A brief history of the globalized world in the twenty-first century*. New York, Allen Lane.
- Gadd, K.W. & Oakland, J.S. (1995). Discontinuous change in a total quality environment, D2D Ltd: A case study. *Business Process Reengineering and Management Journal*, 1(2), 7-27.
- González, J., Pascual, S. & Robles, G. (2003). *Introducción al Software Libre*. Cataluña, Universidad Abierta de Cataluña.
- Hammer, M. & Champy, J. (1993). *Reengineering the Corporation: A Manifesto for Business Revolution*. New York, Harper Business.
- Hars, A. & Ou, S. (2002). Working for free? Motivations of participating in open source projects. *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)*, 7, 25-39.
- Hippel, E. & Krogh, G. (2003). Open source software and the private collective innovation model: Issues for organization science. *Organization Science*, 209-223.
- Lakhani, K.R. & Hippel, E. (2003). How open source software works: "Free" user-to-user assistance. *Research Policy*, 32(6), 923-943.
- Lave, J. & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge, Cambridge University Press.
- Lowenthal, J. (1994). *Reengineering the Organization: A Step-By-Step Approach to Corporate Revitalization*. Milwaukee, ASQC Quality Press.
- Neumann, C. & Breidert, C. (2005). El reto de utilizar código abierto como estrategia de reutilización. *Revista de la asociación de técnicos de informática*, 117, 47-50.
- Petrozzo, D. & Stepper, J.C. (1994). *Successful Reengineering*. New York, Willey.
- Pintendi, G. (2012). Analysis of the relationship of the level of maturity process with the Success of Open Source Projects. *Revista Antioqueña de las Ciencias Computacionales y la Ingeniería de Software*, 2(2), 21-27.
- Talwar, R. (1993). Business re-engineering - A strategy-driven approach. *Long Range Planning*, 26(6), 22-40.
- Thompson, M. (2005). Structural and epistemic parameters in communities of practice. *Organization Science*, 16(2), 151-164.