

El modelo de administración del conocimiento MACOSC-IASC

Octavio Orozco y Orozco, Computadoras Objetos y Comunicaciones S.A. de C.V., México

Resumen: La dificultad para adecuar los programas de código abierto limita la utilización de este bien público en las pequeñas organizaciones que desarrollan software, pues realizar esta actividad para código de tamaño considerable y especializado requiere de un uso intensivo del conocimiento organizacional que se considera necesario administrar. En este artículo argumento que la concepción de conocimiento organizacional y el modelo de su administración vigentes son insuficientes en este caso. Así, a partir de concebir al conocimiento organizacional como un sistema complejo de acciones en ejecución, conjeturo que para atender la necesidad de estas organizaciones es pertinente un modelo de administración del conocimiento conformado por tres sistemas de acciones denominados Uso, Adecuación y Contribución. Para probar la pertinencia del modelo, realicé el diseño mixto de un método de investigación-acción de sistemas complejos y lo utilicé para estudiar el complejo empírico. Asimismo, presento los resultados de intervenir con el método una de estas organizaciones en México para implementar el modelo. Concluyo, a partir de las explicaciones del funcionamiento del sistema complejo, que el modelo que propongo es pertinente para atender la necesidad de administrar el conocimiento en la organización de estudio.

Palabras clave: administración del conocimiento, sistemas complejos, software de código abierto

Abstract: The difficulty in adapting open source software limits the use of this public good in small software development organizations, as to carry on this activity for a sizable and specialized open source software requires intensive organizational knowledge that is deemed necessary to be managed. In this article I argue that the current conception of organizational knowledge and methods for its management are insufficient in such case. Thus, from conceiving organizational knowledge as a complex system of actions being executed, I hypothesize that to attend the need of these organizations it is pertinent to apply a model to manage organizational knowledge which consists of three systems of actions by the name of Usage, Adaptation and Contribution. To test the pertinence of the model I design a mixed methods research based on action-research and complex systems theory, and apply this research method to study the empirical complex. I then present the results of intervening, with the use of this method, one of those organizations in Mexico to implement the model. My conclusion is, based on the explanation of the inner workings of the complex system, that the model I propose is pertinent to fulfill the need of knowledge management in the subject organization.

Keywords: Knowledge Management, Complex Systems, Open Source Software

Introducción

Cualquier persona que sea usuario de Internet o de alguno de la gran mayoría de teléfonos móviles avanzados es, también y quizá sin saberlo, usuario de programas de código abierto. El amplio alcance de esta tecnología —considerada como un bien público, al que es posible acceder de manera libre— y el impacto que tiene en la sociedad son innegables porque los programas de código abierto son parte de los cimientos sobre los que operan cotidianamente Internet y dichos teléfonos.

En el caso de Internet estos cimientos consisten de tres tecnologías esenciales: protocolos básicos de comunicación *TCP/IP*, sistema de nombres de dominio *DNS* y correo electrónico *email*; esos cimientos continúan su evolución con sendos programas o software de código abierto: *BSD Sockets*, *BIND* y *sendmail*. Para el caso de la mayoría de los teléfonos móviles avanzados o *smartphones*, dichos cimientos son sistemas operativos de código abierto: Linux, en el caso de Android de Google y FreeBSD (BSD Berkeley Unix) en el caso de iOS de Apple. Si caracterizamos el alcance de esta tecnología con porcentajes, tenemos que: en el mercado de equipos de cómputo, un 14.7% de servidores —para el cuarto trimestre del 2009— en todo el mundo se vendió con Linux (IDC, 2010), mientras que el 89.2% de las supercomputadoras utilizaron como sistema operativo Linux (TOP500.Org, 2010). Para 2011, el 52.5% de los teléfonos celulares avanzados utilizaron un sistema



operativo con Linux (Econsultancy, 2011); y para el 2012, el 68.8% utilizaron Android y el 18.8% utilizaron iOS, por lo que el 87.6% del mercado global de 722 millones de teléfonos celulares avanzados utilizó software de código abierto (IDC, 2013). De esta forma, un usuario de servicios de correo electrónico, mensajes cortos, o de búsqueda en Internet, y un usuario de teléfonos celulares avanzados son prácticamente usuarios de software de código abierto (Orozco, 2011). La relevancia del software de código abierto estriba en que permite que cualquier organización se beneficie de su uso y contribuya a incrementar su valor. Utilizaré el nombre “software de código abierto” de manera consistente en este trabajo para denominar, con un atributo común, a un conjunto de programas que es amplio y relevante. El atributo común con el que caracterizo este software es que permite ejecutar, estudiar, modificar —como se juzgue pertinente— y redistribuir el código en el que está escrito el software. Este atributo se especifica usualmente en una parte de la licencia de uso del software y hay múltiples opciones con importantes diferencias en ese ámbito¹.

Sin embargo, la dificultad para adecuar los programas de código abierto limita utilizar ese bien público en las pequeñas organizaciones que desarrollan software, pues realizar esta actividad para código de tamaño considerable y especializado requiere de un uso intensivo del conocimiento organizacional que se considera necesario administrar. Así, estas pequeñas organizaciones en el caso de que se planteen como objetivo adecuar software de código abierto porque lo requieren para prestar sus servicios o proveer los bienes que son una parte esencial de su razón de ser, enfrentan la necesidad de administrar el conocimiento organizacional para lograr ese objetivo.

En este trabajo argumento que el modelo vigente de administración del conocimiento, que divide en una faceta explícita y una tácita al conocimiento organizacional, es insuficiente para las pequeñas organizaciones que desarrollan software, lo que las enfrenta a un problema porque dicho conocimiento dual les es inaccesible, en el caso de que requieran adecuar programas de código abierto (sección 1). Propongo entonces, una nueva concepción de conocimiento organizacional y un modelo de administración del conocimiento (sección 2), así como un método para investigar su implementación en una organización sujeto de estudio (sección 3). Integro el modelo MACOSC-IASC (sección 4) y presento el estudio empírico en el que describo el resultado de intervenir con él la organización (sección 5). Concluyo el trabajo con la discusión correspondiente.

El problema

Problema práctico

Cuando las pequeñas organizaciones que desarrollan software se proponen adecuar software de código abierto tienen la necesidad de administrar el conocimiento organizacional y enfrentan una disyuntiva al pretender hacerlo a partir de la concepción dual vigente, que lo divide en tácito y explícito.

La necesidad como problema práctico se hace evidente al considerar el límite o umbral en el tamaño del software, T_0 , que se ha definido empíricamente con la relación:

$$T_0 \geq 5000LC$$

Esto equivale a un desarrollo de software que consiste de 100 páginas de código fuente, en un lenguaje de programación de alto nivel, y toda la documentación asociada necesaria. También es equivalente a una carga de trabajo, para un solo desarrollador de software, de más de un año de acuerdo a la productividad promedio en la industria (Wang, 2008, 12-14). En el área de sistemas operativos de código abierto, una versión hecha pública en 2009 del kernel Linux conocida como linux-2.6.29 tiene un tamaño de $T_l = 7\,707\,587$ líneas de código abierto en lenguaje “C”. En el área de arquitecturas estándares de cómputo distribuido para la integración de aplicaciones empresariales, con el estándar de la industria CORBA (OMG, 2012a), una versión del ORB conocida como JacORB 2.3.1 tiene un tamaño de $T_j = 201\,168$ líneas de código abierto en lenguaje “Java”. Desarrollar y ade-

¹ Véase para introducirse al tema (Tucker, Morelli y De Silva, 2011: 16-18), (Goldman y Gabriel, 2005: 129); y para licencias específicas (Free Software Foundation, 2010b), (Open Source Initiative, 2013).

cuar esa cantidad de software hace indispensable involucrar a varios especialistas² y por ello se considera necesario administrar el conocimiento organizacional en las pequeñas organizaciones que desarrollan software, cuando el desarrollo implica adecuar este software. Sin embargo, en ese caso enfrentan una disyuntiva, pues necesitan administrar el conocimiento organizacional y por tres razones no es posible hacerlo con la concepción y modelo de conversión vigentes³.

La primera razón, es que no se cuenta usualmente —de manera vinculante y directa— con los desarrolladores que tienen el conocimiento tácito correspondiente para adecuarlo. La segunda razón, es que no se cuenta usualmente con el conocimiento explícito, en la forma de documentación del software, que propicie la adecuación del código en el que fue escrito.

Al expresar estas dos razones con el modelo de conversión vigente tenemos:

- La socialización es insuficiente para el caso en que es necesario realizar adecuaciones o localizaciones específicas. Este caso se explica porque no es el interés inmediato de los desarrolladores del software de código abierto resolver los problemas de la pequeña organización. No obstante, se reconoce que la socialización es útil para para el caso que es necesario resolver fallas o problemas en el software de código abierto⁴. Este caso se explica porque resolver los errores de los programas usualmente si es de interés para los desarrolladores del software de código abierto. Una alternativa reciente consiste en contratar —disponible en ciertos casos de software de código abierto— alguna empresa que se dedique a proveer servicios especializados, que van desde resolver de manera oportuna los errores reportados hasta realizar adecuaciones o localizaciones específicas. Por los costos involucrados, esta opción es, sin embargo, usualmente solo viable para organizaciones de mayor tamaño. Se podría caracterizar la socialización en este caso como una relación en la que se observa también una disyuntiva de asimetría de información⁵.
- La externalización es insuficiente porque los integrantes de la pequeña organización no son los creadores del software y —sobre todo al inicio del proyecto— no tienen los recursos para intentar documentarlo. Como ya se ha afirmado, los creadores del software, por su parte, usualmente no lo documentan. No obstante, se reconoce que hay algunos casos de software de código abierto en que se ha generado documentación, creada por algunos desarrolladores o por terceros, y que falta buen trecho por recorrer. Se puede afirmar que las adecuaciones más sofisticadas implican cambios a la arquitectura del software. También se registran esfuerzos en esta área para algunos casos de software de código abierto en que se ha generado alguna documentación sobre su arquitectura. (Véase, por ejemplo, Brown y Wilson, 2012). En este terreno, y respecto de la necesidad de conducir estudios empíricos para explorar la relación entre la arquitectura del software y los requerimientos de comunicación de los grupos de trabajo, en el contexto de la administración del conocimiento puede consultarse a Ali, Beecham y Mistrík (2010, 351).
- La combinación y la internalización son insuficientes por el tamaño y la falta de documentación del software. No obstante, se reconoce que hay algunos casos en que la documentación creada por terceros para productos similares podría ser utilizada, con el debido cuidado en el nivel de similitud y diferencia existentes. Véase por ejemplo lo propuesto por Brown y Wilson (2011). Sin embargo, es evidente que aquí también falta buen trecho por recorrer.

² Véase (The Linux Foundation, 2008) para una estimación del esfuerzo que requirió desarrollar Linux.

³ Vía el modelo de conversión de conocimiento: a) Tácito a tácito (socialización), b) Tácito a explícito (externalización), c) Explícito a explícito (combinación), d) Explícito a tácito (internalización). Véase (Nonaka y Takeuchi, 1995: 62-70).

⁴ Véase (Flichy, 2008: 3), sin olvidar que las soluciones que describe Flichy, ocurren con tiempos de respuesta que dependen de los desarrolladores del software y no del que confronta directamente el problema, en el que se detectó el error.

⁵ Véase (Nielsen 2008: 5-11). Sin embargo, considero que en ese trabajo permea la concepción dual vigente y además para el caso del software de código abierto, la pequeña organización que desarrolla software es solo una parte de una comunidad mucho más amplia que cruza horizontalmente a las organizaciones y esto obliga a considerar aspectos adicionales a los ahí mencionados.

La tercera razón tiene que ver con dos aspectos que surgen recurrentemente en la práctica administrativa del conocimiento organizacional con la concepción vigente. Primero, los creadores de esta concepción tomaron un curso que omitió la acción y privilegió una posición epistémica y ontológica; segundo, en la práctica se asocia al conocimiento organizacional con aspectos intangibles y con el criterio de rendimiento económico. Estos aspectos se sustentan —ceñidos a los autores estudiados— de la siguiente manera:

- Nonaka y Takeuchi reconocen la importancia de la acción y su relación esencial con el conocimiento⁶, sin embargo la omiten y se vuelcan en lo que denominan la dimensión epistemológica⁷ del conocimiento que sustentan en la “expansión práctica” del trabajo de Polanyi⁸. También plantean el proceso de creación del conocimiento organizacional, considerando que el conocimiento es creado solo por individuos y que la organización soporta y provee el contexto de dicha creación. Se percatan de esta peculiaridad del conocimiento y la llaman la dimensión ontológica⁹. Davenport y Prusak respecto del valor del conocimiento, afirman que se ubica cerca de la acción, que puede y debe ser evaluado por las acciones a las que conduce, pero que puede ser difícil trazar la ruta entre conocimiento y acción (Davenport y Prusak, 2000). Así, consideramos que se reconoce la importancia de la acción y que su omisión —y consecuente dificultad para enlazarla al conocimiento— muestran la insuficiencia de la concepción dual vigente.
- En la administración, se asocia al conocimiento con aspectos intangibles: “aunque incorpóreo, es efectivamente un activo generador de bienes” (Herrera, Ramirez y May, 2012: 273); “el capital invisible que poseen las organizaciones” (Solleiro *et al.*, 2009: 33). Le Blanc y Ermine intentan trascender este aspecto intangible y medir el capital intelectual o *knowledge capital*. De las hipótesis que han propuesto eligen la semiótica, en la que conjeturan que el conocimiento se percibe como un signo. Afirman entonces que el conocimiento es información que hace sentido en un contexto dado (Le Blanc y Ermine, 2007: 54). Esto, dicen, les permite resolver lo que identifican como una ambigüedad entre conocimiento e información. Con esta definición vinculan el capital intelectual *K*, que reside fuera del sistema de administración del conocimiento *S*, con la base de conocimiento *I*. Sin embargo, identifico dos problemas, el primero es que a pesar de redefinir conocimiento, recurren de nueva cuenta a la concepción dual de Nonaka y Takeuchi donde *I* es conocimiento explícito; *K*, tácito. Este último vuelve a ser intangible, porque *K* queda fuera de *S* (Le Blanc y Ermine, 2007: 53). El segundo problema es que para formalizar uno de los componentes críticos —el contexto— lo reducen, con el uso de grafos y la teoría de redes sociales, a la conexión entre los usuarios de la base de conocimiento *I* (Le Blanc y Ermine, 2007: 63-64). Beyerlein y Kennedy parten de la concepción dual para administrar el capital intangible; sin embargo, afirman que es imperativo encontrar maneras de desarrollar las diversas formas del capital intangible¹⁰; que un primer paso es hacerlas visibles vía mecanismos de registro y reporte¹¹; que la importancia de hacer visible las diversas formas de capital intangible es aparente cuando las empresas reducen su tamaño: a mayor capital intelectual y social de la persona que deja la organización el último día de empleo, mayor la pérdida para la empresa (Beyerlein y Kennedy, 2008: 409). Considero que ignorar o reducir deliberadamente la individualidad, la cultura organizacional, el contexto y la situación social

⁶ Véase (Nonaka y Takeuchi, 1995: 58,59), también su respectiva nota de pie 2: (Nonaka y Takeuchi, 1995: 90).

⁷ Al plantear que el conocimiento explícito es de tipo proposicional.

⁸ Véase (Nonaka y Takeuchi, 1995: 60) y (Polanyi y Sen, 2009: 15-25).

⁹ Sin embargo, asumen que como proceso, la creación del conocimiento organizacional, amplifica el conocimiento creado por los individuos y lo “cristaliza como parte de la red de conocimiento de la organización” (Nonaka y Takeuchi, 1995: 59).

¹⁰ Ellos caracterizan seis: Procesos internos y externos de conversión —The way assets are used, grown, stored, and combined determines how much value is produced—, intelectual —What you know—, social —Who you know—, organizacional —How you do it—, colaborativo —How you work together— y humano —How well you can do it— (Beyerlein y Kennedy, 2008: 402-403).

¹¹ Como se registra en este trabajo, esto usualmente se realiza con una perspectiva dual que aborda al conocimiento —en su faceta explícita— como información y consecuentemente utiliza de manera preponderante las tecnologías de información y comunicaciones con la pretensión de vincularlo con su faceta tácita, intangible.

al recurrir a aspectos inobservables, intangibles, muestra la insuficiencia de la concepción dual, para la administración del conocimiento organizacional.

- Prevalece en la concepción vigente el criterio de rendimiento económico, de competitividad excluyente, de capital intelectual, que privatiza el conocimiento como recurso. Para los esfuerzos administrativos de privatizar el conocimiento como capital intelectual, es notable que el productor del capital intelectual pueda salir caminando de la organización, inclusive que no sea un activo de la organización, véase (Drucker, 1974: 308). En este contexto, sin embargo, para el desarrollo de software de código abierto la administración del conocimiento no es guiada por el criterio de capital intelectual, véase (Federman, 2006: 89) y (Goldman y Gabriel, 2005: 1). Por lo que, para nuestro caso de interés, el criterio de rendimiento económico vigente en la práctica es insuficiente pues es necesario utilizar criterios adicionales, entre los que se encuentran promover como un bien público el conocimiento y el software de código abierto.

La pequeña organización enfrenta una disyuntiva a la hora de administrar el conocimiento organizacional con la concepción y modelo vigentes, por las tres razones expuestas sobre su insuficiencia. Considero que por ello no puede administrar con éstos, luego que ha de hacerlo con otros, lo que me lleva a formular la pregunta de investigación como sigue.

La pregunta de investigación

Se considera que los programas de código abierto cumplen con un conjunto de criterios específicos que deliberadamente eliminan cualquier límite para compartirlos, hacen posible que cualquier organización se beneficie de su uso y además participe, si desarrolla software, para incrementar el valor de dichos programas. También, que el desarrollo de software es una actividad que hace uso intensivo del conocimiento, que este requiere ser administrado y para ello se requieren métodos administrativos y organizacionales pertinentes. Finalmente, que la pequeña organización enfrenta una disyuntiva a la hora de administrar el conocimiento organizacional con la concepción y modelo vigentes, por las razones expuestas sobre su insuficiencia. Desde este enfoque planteo como pregunta de investigación:

¿Qué modelo para administrar el conocimiento es pertinente para la pequeña organización que desarrolla software, en el caso que se proponga adecuar programas de código abierto?

Una nueva concepción de conocimiento organizacional y el modelo MACOSC

Para responder la pregunta de investigación formulo una nueva concepción de conocimiento organizacional y un modelo de administración del conocimiento, a fin de atender la necesidad de las pequeñas organizaciones que desarrollan software, cuando pretenden adecuar software de código abierto.

Una nueva concepción de conocimiento organizacional

El análisis e interpretación integradores de diversos planteamientos¹² me permiten concebir al conocimiento organizacional como un sistema complejo de acciones eficaces, eficientes y reiteradas en ejecución por los diversos agentes involucrados en la organización:

- **Si** un agente X en un dominio D al ejecutar el sistema de acciones A consigue en grado elevado el objetivo O (con $O \subseteq \omega$) que se propone **y** lo hace en tiempo T (con $T \leq \tau$) **y** utiliza recursos P (con $P \subseteq \rho$) **y** minimiza los resultados no deseados o no propuestos E (con $E \subseteq \epsilon$), **entonces** X actúa eficaz y eficientemente (AEE) para lograr O en D .
- **Si** un agente X actúa eficaz y eficientemente (AEE) en D , un número de veces η (con $\eta > 1$), **entonces** X actúa eficaz, eficiente y reiteradamente ($AEER$) para lograr O en D .

¹² Véase (Orozco, O., 2013: 28-43).

- **Si** un agente X actúa eficaz, eficiente y reiteradamente (*AEER*) para lograr O en D , **entonces** X **conoce** como lograr O en D .

En este sistema de acciones A , los valores de: ω como el conjunto de objetivos de la organización, τ como el tiempo máximo en que se puede lograr el objetivo, ρ como los recursos máximos que se pueden utilizar para lograr el objetivo, ϵ como los resultados no deseados o no propuestos pero tolerables, y η como el número de veces mínimo que se debe ejecutar la acción; se definen al articular y ejecutar la acción eficaz, eficiente y reiterada, de los diversos agentes involucrados en la organización.

- **Si** una organización —considerada como un agente X — al ejecutar el sistema de acciones A , actúa eficaz, eficiente y reiteradamente (*AEER*) para lograr el objetivo O en D , **entonces** esa organización **conoce** como lograr O en D .
- **Si** un sistema de acciones A en ejecución, cumple dos condiciones *Co1*, *Co2*:
 - *Co1* - Las funciones de los elementos (las acciones de los diversos agentes —como subsistemas— involucrados en la organización) no son independientes, lo que determina la interdefinibilidad de los componentes
 - *Co2* - El sistema como totalidad es abierto, es decir, carece de fronteras rígidas; está inmerso en una realidad más amplia con la cual interactúa por medio de flujos de materia, energía, recursos económicos, políticas regionales, nacionales, etcétera**entonces** el sistema de acciones A en ejecución cumple las condiciones que caracterizan a un sistema como complejo¹³.
- **Si** un sistema de acciones A en ejecución, incluye tres causalidades *Ca1*, *Ca2* y *Ca3*:
 - *Ca1* - Lineal: Tal acción produce tales efectos
 - *Ca2* - Circular retroactiva: que estimula o disminuye los efectos
 - *Ca3* - Recursiva: los efectos son necesarios para la causa**entonces** el sistema de acciones A en ejecución, tiene todos los niveles de organización de los sistemas complejos¹⁴.
- **Si** una organización al ejecutar el sistema de acciones A , actúa eficaz, eficiente y reiteradamente (*AEER*) y el sistema de acciones A , cumple las condiciones *Co1* y *Co2* e incluye las causalidades *Ca1*, *Ca2* y *Ca3* para lograr el objetivo O en D , **entonces** **conoce** como lograr O en D y se considera ese **conocimiento organizacional** como un **sistema complejo**.

Por lo anterior, concibo al conocimiento organizacional como un sistema complejo de acciones eficaces, eficientes y reiteradas en ejecución por los diversos agentes involucrados en la organización. El sistema complejo emerge cuando el sistema de acciones en ejecución cumple las condiciones e incluye las causalidades descritas. La emergencia de este sistema complejo se observa, por ejemplo, cuando la organización logra el objetivo de mantener su acoplamiento estructural de tercer orden¹⁵, y logra en grado elevado prestar sus servicios o proveer los bienes que son una parte esencial de su razón de ser vía la ejecución del sistema¹⁶ de acciones, de forma abierta e interdefinida.

El modelo para la administración del conocimiento MACOSC

Considero que esta nueva concepción de conocimiento organizacional, en el dominio de las pequeñas organizaciones que desarrollan software, cuando el desarrollo involucra adecuar software de código abierto, me permite formular un modelo de administración del conocimiento organizacional como sistema complejo (MACOSC) que propicia atender su necesidad.

¹³ En el dominio de los sistemas complejos, son las dos condiciones que García utiliza para caracterizar a un sistema como complejo (García, 2008: 143-144).

¹⁴ Son las tres causalidades que Morin identifica en todos los niveles de organización complejos (Morin, 2004: 122-124).

¹⁵ Véase la definición de acoplamiento estructural en (Maturana y Varela, 2003: 64,67), y la de tercer orden en (Maturana y Varela, 2003: 121).

¹⁶ Compuesto del conjunto de las acciones de los diversos agentes involucrados.

El modelo se integra por tres subprocesos-estadios: **Uso**, **Adecuación** y **Contribución**. Estos consisten de acciones que se articulan de forma colectiva interdefinida y con interdependencia de funciones como se pretende ilustrar en la figura 1.

Subproceso -Estadio	Acciones
Uso	<i>Observar, Representar, Compilar, Verificar, Ejecutar, Aplicar.</i>
Adecuación	<i>Observar, Representar, Modificar, Compilar, Verificar, Ejecutar, Aplicar.</i>
Contribución	<i>Observar, Proponer, Negociar, Proveer.</i>

Figura 1: Los subprocesos-estadios del MACOSC

Fuente: Elaboración propia.

En la práctica administrativa el MACOSC se implementa como un sistema de acciones con el propósito de que la organización logre en alto grado su objetivo. Los tres subsistemas de acciones al ejecutarse conducen a la organización a un estadio correspondiente. Cada subsistema, como subproceso-estadio, se describe a continuación.

El subproceso-estadio **Uso** consiste de las acciones siguientes:

- **Observar**
Se observan los lineamientos de la licencia de uso de software. El resultado cuantificable es la confirmación de que se puede proceder con las acciones siguientes, en el entendido de que dicha licencia de uso armoniza con el objetivo particular de la organización. Se obtiene del sitio en Internet original¹⁷ el software en código fuente y se observa. El resultado cuantificable y observable es un inventario detallado.
- **Representar**
Se procesa el software con herramientas que permitan generar diversas representaciones. El resultado cuantificable y observable son las representaciones de las interrelaciones de los componentes en código fuente.
- **Compilar**
Se compila el software para generar los ejecutables y/o bibliotecas. El resultado cuantificable y observable es un programa compilado, ejecutable y/o un conjunto de bibliotecas, listos para ser invocados.
- **Verificar**
Se verifica que el software ejecutable y/o bibliotecas pasen las pruebas esenciales que constaten su funcionalidad “tal y como vienen”. El resultado cuantificable y observable es el número de pruebas que pasa el código al ejecutarlas.
- **Ejecutar**
Se ejecuta el software y/o bibliotecas en un entorno de “prueba” que permita observar su funcionalidad “tal y como vienen”. El resultado observable es que las pruebas previstas, en ese entorno, pasan sin errores.
- **Aplicar**
Se crea una aplicación inicial para confirmar que es posible usar el software “tal y como viene” para tomar la oportunidad o resolver el problema planteado en el objetivo particular de la organización. El primer resultado es que sea posible crear una aplicación sin la funcionalidad requerida. Los resultados subsecuentes agregan de forma iterativa mayor funcionalidad, con

¹⁷ También se utilizan los términos: Se “baja”, se hace download, etcétera.

maestría creciente, hasta lograr la funcionalidad requerida para un objetivo parcial. Se itera en este estadio para crear aplicaciones cada vez más demandantes que permitan explorar el subestadio *Aplicar* con todo detalle, hasta lograr en alto grado el objetivo propuesto.

El subproceso-estadio **Adecuación** consiste de las acciones siguientes:

- **Observar**
Se observa el software, en modo fuente y en modo ejecutable, para decidir cuáles elementos es necesario adecuar para que la organización pueda tomar la oportunidad o resolver el problema planteado de mejor manera. Se establecen como objetivos y se priorizan. El resultado es una lista de objetivos priorizados.
- **Representar**
Se procesa el software con herramientas que permitan generar representaciones del software acordes a los objetivos ordenados y priorizados. El resultado son representaciones ad-hoc al estadio cognoscitivo del agente involucrado.
- **Modificar**
Se modifica el software para alcanzar el objetivo siguiente de acuerdo a su prioridad. Se recorre parte del subproceso-estadio **Uso** una vez realizada la modificación. El resultado observable es el número de pruebas que el software modificado pasa¹⁸.
- **Compilar**
Se compila el software para generar los ejecutables y/o bibliotecas. El resultado cuantificable y observable es un programa compilado, ejecutable y/o un conjunto de bibliotecas, listos para ser invocados.
- **Verificar**
Se verifica que el software ejecutable y/o bibliotecas pasen las pruebas esenciales que constata la nueva funcionalidad. El resultado cuantificable y observable es el reporte de pruebas realizadas.
- **Ejecutar**
Se ejecuta el software en un entorno de “pruebas” que permita observar su nueva funcionalidad. El resultado cuantificable y observable es el reporte de pruebas realizadas.
- **Aplicar**
Se crea una aplicación inicial para confirmar que es posible usar el software ya adecuado para tomar la oportunidad o resolver el problema planteado en el objetivo particular de la organización. Se itera en este estadio para crear aplicaciones cada vez más demandantes que permitan explorar el subestadio *Aplicar* con todo detalle haciendo uso de la adecuación realizada. El resultado observable es el grado en que se logra el objetivo.

El subproceso-estadio **Contribución** consiste de las acciones siguientes:

- **Observar**
Se observa la red social de innovación vía los medios de comunicación e interacción que utilizan: listas de correo, sitios en internet. El resultado es la elección de un medio que se considere pertinente y una forma particular de abordar la red social de innovación.
- **Proponer**
Se propone la adecuación al software que se considere pertinente por el medio elegido. El resultado es la respuesta inicial de la red social de innovación.
- **Negociar**
Se negocia con la red social de innovación la responsabilidad sobre el mantenimiento de la adecuación propuesta al software. El resultado final es la decisión integrar la adecuación al código base o mantener de forma separada y pública dicha adecuación.

¹⁸ En adición a que pase, por supuesto, todas las pruebas anteriores que confirman que no se introducen —en principio— nuevos errores en el código.

- Proveer

Se provee el software que compone la adecuación de forma acorde al resultado de la negociación del paso anterior. El resultado observable es que la adecuación esté disponible al público en general, de acuerdo con los lineamientos de la licencia de uso del software.

En la figura 2 se pretende ilustrar la implantación del MACOSC en la organización, como parte de los procesos de administración del conocimiento ilustrados¹⁹ por las circunferencias AC.

La práctica administrativa de administración del conocimiento propicia entonces la emergencia del conocimiento organizacional como sistema complejo cuando se articula y ejecuta el sistema de acciones y cuando este cumple las condiciones e incluye las causalidades descritas. El análisis de sus procesos a través del tiempo hace observable los niveles de efectividad reiterada y la eficiencia alcanzados con el sistema de acciones así como en los efectos, deseados y no deseados, en el dominio²⁰.

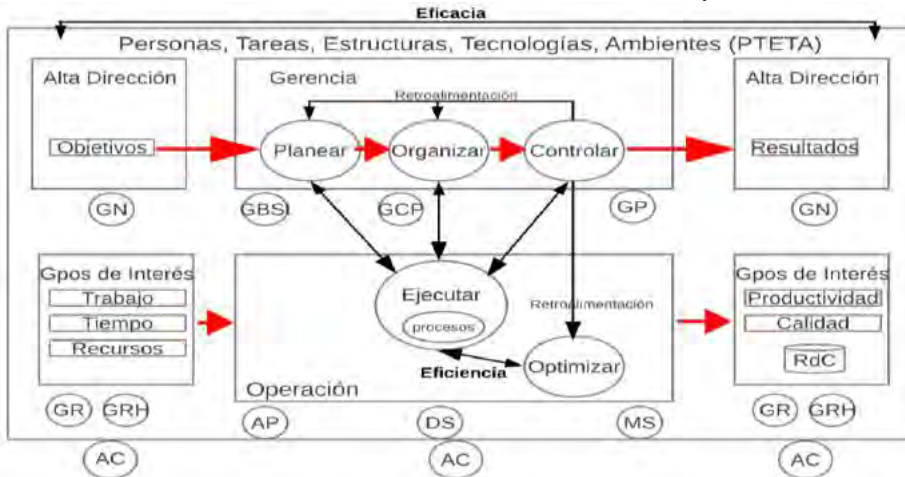


Figura 2: MACOSC en la Organización

Fuente: Adecuación a partir de Modelos de Procesos de la Teoría General de la Administración (Chiavenato, 2000: 945-946), y de los modelos de Oktaba et al. (2009, 29); Wang (2008, 855-864) y Carrillo (2008, 50-53).

En la figura 3 se pretende ilustrar el tránsito esperado de la organización, por estadios observables, durante la implantación del MACOSC, como parte de los procesos de administración del conocimiento.

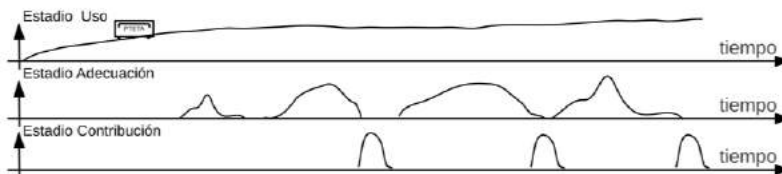


Figura 3: Tránsito esperado por los estadios Uso, Adecuación y Contribución al implantar el MACOSC en la organización.

Fuente: Elaboración propia.

¹⁹ Dónde GCP significa “Gestión de Cartera de Proyectos”; GBSI “Gestión de Bienes, Servicios e Infraestructura”; GR “Gestión de Recursos”; GRH “Gestión de Recursos Humanos”; MS “Mantenimiento de Software”; DS “Desarrollo de Software” y AP “Administración de Proyecto”.

²⁰ En la perspectiva del observador que interpreta.

Con la nueva concepción²¹ de conocimiento organizacional²² y con el modelo propuesto para la administración del conocimiento respondo a la pregunta de investigación. Esto me permite crear el problema de investigación con la siguiente hipótesis:

Si Las pequeñas organizaciones que desarrollan software, para el caso en que se propongan adecuar programas de código abierto, formulan sus acciones con el modelo MACOSC, entonces van a conseguir en grado elevado su objetivo.

El método

El problema de investigación requiere el estudio empírico de la pertinencia del modelo MACOSC, para ello realizo un diseño mixto que integra el método de investigación-acción y la teoría de sistemas complejos. La postura filosófica que adopto en esta investigación para realizar los trabajos de diseño es pragmática²³. Para justificar la decisión de utilizar en el diseño la investigación-acción, en primer lugar, considero como opinión experta la de Oktaba, quien reporta utilizar ese método en sus trabajos de investigación sobre procesos de ingeniería de software para pequeñas y medianas empresas. El sustento de esta consideración es que sus propuestas se han aceptado como estándares internacionales (Morales, Ventura, Oktaba y Torres, 2012), (Oktaba *et al.*, 2009). En segundo lugar considero que, desde la perspectiva de la informática, realizar investigación sobre sistemas de información con el método de investigación-acción se defiende apropiadamente en diversos artículos (Avison *et al.*, 1999a; Avison *et al.*, 1999b y Baskerville, 1999). También, es relevante utilizar el método investigación-acción pues, la intervención puede ser beneficiosa para la organización que participa en la investigación, lo que conlleva el potencial de fortalecer los vínculos entre las pequeñas organizaciones y los centros de investigación universitarios.

En este contexto, la justificación de la decisión de utilizar la teoría de sistemas complejos, desde la perspectiva de la epistemología genética, tiene dos aspectos. El primero es que considero como opiniones expertas la de Carrillo, quien reporta utilizar esta teoría de la complejidad en sus trabajos de investigación de gestión del conocimiento en entornos académicos (Carrillo, 2008), así como la de Amozurrutia, quien propone aplicarla en la construcción de sistemas adaptativos para el análisis social (Amozurrutia, 2011). El segundo aspecto es mi propósito de entrar en detalle y estudiar esta teoría al pretender aplicarla como columna vertebral teórica del diseño investigación-acción, para explorar su potencial como solución a varias recomendaciones reportadas en los artículos de Madeiros dos Santos y Horta respecto de la tendencia creciente del uso del diseño investigación-acción en ingeniería de software (Madeiros dos Santos y Horta, 2009), y en el de (Kock *et al.*, 1997) respecto a la carencia²⁴ de rigor científico del mismo método investigación-acción.

La investigación-acción de sistemas complejos IASC

Parto de las anteriores consideraciones para realizar el diseño de investigación, que denomino diseño mixto investigación-acción de sistemas complejos (IASC), con el objetivo de observar la pertinencia del MACOSC para la administración de conocimiento en una organización concreta que pretende adecuar programas de código abierto específicos. Este diseño consiste en un proceso iterativo incremental que contempla la múltiple realización de subprocesos de investigación-acción que se repiten, con maestría creciente, en cada iteración. Un subproceso de investigación-acción está compuesto por cuatro actividades fundamentales: diagnóstico de la oportunidad, planeación de la acción, intervención, aprendizaje reflexivo (Avison *et al.*, 1999a; Kock *et al.*, 1997).

²¹ Véase (Khun, 2007, 213: 238-246).

²² La organización como agente, el conjunto de relaciones que la caracterizan como sistema, se considera como una totalidad organizada en su acción eficaz y en las implicaciones entre acciones en un momento dado. Véase (Piaget, García *et al.*, 1997: 149).

²³ Véase (Laudan, 1993) y (Easterbrook *et al.*, 2008: 286).

²⁴ La falta de control sobre las variables (del ambiente, etc.) o rigor científico, con la consecuente: imposibilidad de generalizar los resultados, falta de imparcialidad, y riesgo de involucramiento del investigador y su efecto en los resultados.

La propuesta teórica de la investigación de sistemas complejos, sintetizada por García (2008: 181-190), Amozurrutia (2011, 227 y 319-324) y por Rivas *et al.* (2009, 39-97), se usa como columna vertebral de las cuatro actividades que componen cada subproceso de la forma siguiente:

Diagnóstico de la oportunidad

1. Actividad fundamental en donde se entrecruzan la perspectiva cualitativa y la cuantitativa. Exige aplicar ambas perspectivas; primero, para identificar variables cualitativas —grado de dependencia e interdependencia porque se pretende incrementar el grado de interdependencia y disminuir el de dependencia respecto a clientes y proveedores— que permitan iterativamente construir la *explicación causal* que dé cuenta de la funcionalidad del sistema complejo —el conocimiento organizacional como acción eficaz, eficiente y reiterada—, y segundo, para identificar aspectos mensurables en el *complejo empírico —la realidad*²⁵— que puedan ser atribuidos²⁶ a las variables cuantitativas —objetivos, sistemas de acciones ejecutados, resultados (objetivos logrados, recursos consumidos, efectos deseados y no deseados, etcétera)— en la teoría que se construye.
2. Identificar el objeto de estudio²⁷, los objetos empíricos de estudio²⁸, que estarán en correspondencia con la cadena de inferencias en la teoría. Crear la versión²⁹ del sistema complejo³⁰ que representa el *complejo empírico* de estudio. Proceder con la distinción y análisis de los niveles del sistema complejo:
 - a. Identificar niveles de organización de la “realidad estructurada”: es decir del conocimiento organizacional como un sistema complejo de acciones eficaces, eficientes y reiteradas en ejecución por los diversos agentes involucrados en la organización.
 - b. Identificar la interacción entre los niveles de organización —Uso, Adecuación y Contribución— del sistema complejo.
 - c. Calendarizar la evolución por las reorganizaciones sucesivas de cada nivel de organización del sistema complejo.

Planeación de la acción

1. Identificar los cursos alternativos de acción para aprovechar la oportunidad, mejorar la situación o resolver el problema.
2. Seleccionar el curso de acción que se considere más apropiado.
3. Establecer los instrumentos de observación y análisis del complejo empírico.

²⁵ (Véase García, 2008: 186), (Kant, 1995: 22-26), (Maass et al., 2012: 291).

²⁶ Véase (García, 2008: 189).

²⁷ En este caso la relación, representada en la teoría, entre los objetivos, las acciones y los resultados, para transitar por los tres subprocesos-estadios para la administración del conocimiento indicados en el MACOSC como unidad de análisis.

²⁸ García los denomina en su conjunto como una abstracción del complejo empírico (García, 2008: 185 - 186) que, en este caso, corresponde al cúmulo real de objetivos, acciones y resultados de una organización. Dicho cúmulo se abstrae, para sistematizarlo, como el conjunto compuesto por los objetivos expresados por la organización, las acciones indicadas para los subprocesos-estadios y los resultados reportados a la organización. Todas las acciones, que componen los subprocesos-estadios, se asocian con objetos empíricos: Los programas de código abierto, los requerimientos detallados, la listas de objetivos y la de resultados por mencionar algunos de los objetos empíricos de nuestra unidad de observación.

²⁹ Le llamo versión porque debemos generar una nueva “versión” del sistema complejo —el conocimiento organizacional siempre cambiante— para cada iteración en el ciclo básico de las cuatro actividades que componen cada subproceso: diagnóstico de la oportunidad, planeación de la acción, intervención, aprendizaje reflexivo.

³⁰ La intención es explicar, en cada iteración, de forma más precisa y atinada los procesos al interior del sistema complejo, tanto en las interrelaciones de sus partes como en su totalidad organizada. Para propiciar con maestría creciente la emergencia del conocimiento organizacional como un sistema complejo de acciones eficaces, eficientes y reiteradas en ejecución por los diversos agentes involucrados en la organización.

Intervención

1. Realizar el curso de acción elegido.
2. Observar los efectos sobre el complejo empírico.

Aprendizaje reflexivo

1. Evaluar los resultados del curso de acción realizado, desde una perspectiva científica, sobre las observaciones realizadas en el complejo empírico:
Agregar variables omitidas y eliminar variables innecesarias en la teoría, para así construir conocimiento³¹, respecto del objeto de estudio y una realidad mejor determinada.
2. Desde una perspectiva organizacional, evaluar los resultados del curso de acción realizado:
Identificar la consolidación de invariantes, que se consideren con valor organizacional³² y para los cuales sea posible establecer una *explicación causal* que da cuenta de la intervención³³.
3. Si al evaluar los resultados, se considera que puede ser conveniente realizar otro ciclo, tanto desde la perspectiva académica como organizacional, entonces hay que reiniciar un nuevo paso del proceso iterativo incremental que contemple la realización de los subprocesos: diagnóstico del problema, planeación de la acción, intervención, aprendizaje reflexivo. Por otro lado, en caso que se considere pertinente dar por terminado el proceso, hay que generar los reportes finales apropiados para la comunidad académica y para la organización.

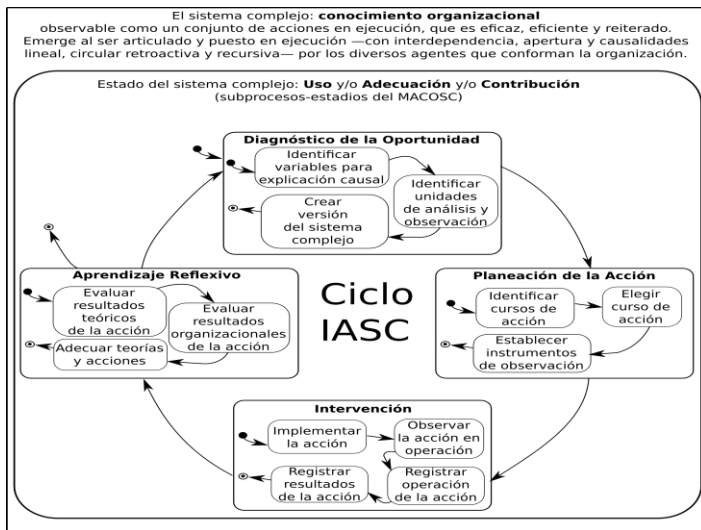


Figura 4: Detalle del diseño mixto IASC para intervenir la organización

Fuente: Elaboración propia.

³¹ Conocimiento en el sentido de acción eficaz, eficiente y reiterada de los agentes epistémicos, cognoscitivos que conforman el grupo de investigación.

³² Valor organizacional asociado a la(s) variable(s) dependiente(s). En este caso esta variable cuantitativa dependiente corresponde a los resultados que se representan en la tríada: objetivos, acciones, resultados. Con diversos índices e instrumentos pretendemos caracterizarlas y observarlas, para ser capaces de consolidar, el avance en el logro de los objetivos organizacionales respecto a los programas de código abierto, en términos de Uso, Adecuación y Contribución.

³³ Vía la(s) variable(s) independiente(s). En este caso el sistema de acciones formulado en el MACOSC. Con este modelo se pretende caracterizar, analizar y propiciar la diversidad de estadios por los que atraviesa la organización al realizar, en la práctica, la administración del conocimiento organizacional. Estas acciones deben ser commensurables tener similar adecuación empírica y, sin embargo, diferente nivel de respuesta funcional empírica, al resistir de mejor o peor manera, el enfrentamiento con contrastaciones exigentes. Contrastaciones en las que incluye, el grado en que propician la eficacia de los integrantes de la organización, para usar, adecuar y contribuir a los programas.

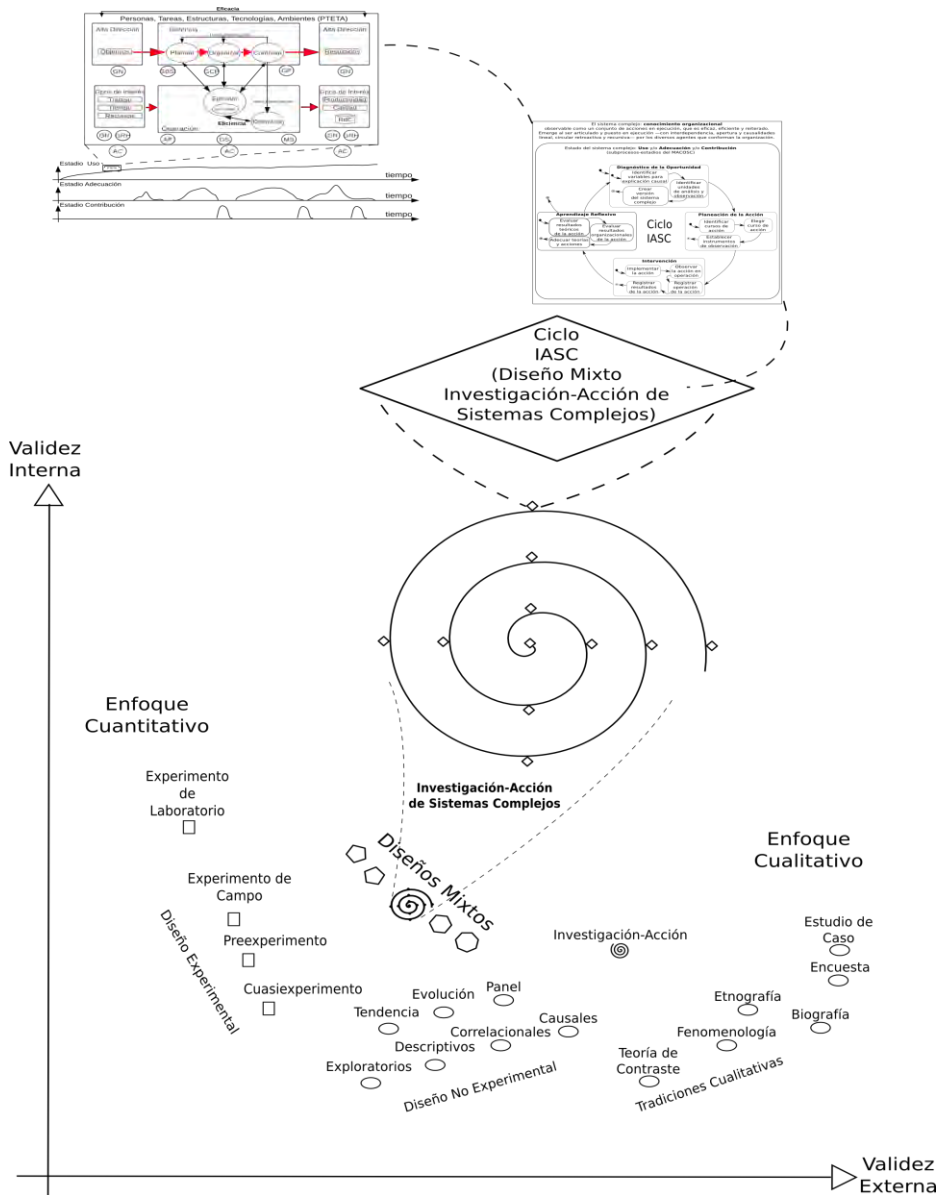


Figura 5: Ubicación del método IASC — Investigación-Acción de Sistemas Complejos

Fuente: *Elaboración propia a partir de Kerlinger y Lee (2002, 403-561), Hernández, Fernández y Baptista (2003, 182-296), García (2008, 79-80) e Easterbrook et al. (2008, 285-311).*

En la figura 4 muestro un esquema gráfico con el ciclo y fases que componen el diseño mixto investigación-acción de sistemas complejos IASC que utilicé para intervenir en la organización sujeto de estudio³⁴. Al realizar un ciclo del IASC se implementa un estadio del MACOSC y así se

³⁴ Se advierte que el diseño del método es un primer paso y que este no se propone como único o que responde puntualmente a todos los requerimientos de científicidad actualmente reconocidos. Sin embargo, pretende ser el inicio de un círculo virtuoso de estudio-transformación que asume que no es posible considerar efectos constantes en las acciones de los individuos y de la pequeña organización sujeto de estudio i.e. que la misma causa no necesariamente producirá el mismo efecto y que no es posible controlar de manera constante las variables de una situación. Véase (Poteete, Janssen y Ostrom, 2012: 71).

contribuye a la emergencia del conocimiento organizacional. Mientras que con el estado del sistema se pretende representar la implementación de alguno de los subprocesos-estadios por los que transita la organización (**Uso, Adecuación o Contribución**), con el sistema se pretende representar la emergencia del sistema complejo

Una representación gráfica de la ubicación del método IASC, se muestra en la figura 5. En el detalle que se muestra del método, la espiral corresponde al proceso iterativo incremental; asimismo, un rombo marca el inicio de un subproceso compuesto por las cuatro actividades que se realizan en la investigación al implementar el MACOSC para observar el conocimiento organizacional como un sistema complejo de acciones eficaces, eficientes y reiteradas en ejecución por los diversos agentes involucrados en la organización.

El modelo MACOSC-IASC

Finalmente, las acciones que se siguen para contrastar y *explicar causalmente* la hipótesis, desde una perspectiva general del diseño mixto IASC, se cruzan con la acciones para cada subproceso-estadio del MACOSC. De esta forma integro el modelo de administración del conocimiento MACOSC-IASC para las pequeñas organizaciones que desarrollan software, aplicable al caso que deseen adecuar programas de código abierto, como se muestra en la tabla de la figura 6. Para cada ciclo de investigación-acción IASC el sistema complejo emerge como conocimiento organizacional durante la implantación del modelo MACOSC en la organización sujeto de estudio³⁵ cuando se realizan las acciones propuestas en cada cruce MACOSC-IASC de manera eficaz, eficiente y reiterada.

El modelo de administración del conocimiento MACOSC-IASC		IASC Investigación-acción de sistemas complejos (Diseño mixto)																				
		Diagnóstico de la Oportunidad	Planeación de la acción <i>Observar, Representar, Modificar, Compilar, Verificar, Ejecutar, Aplicar, Proponer (Pp), Negociar, Proveer (Pv)</i>								Intervención <i>Observar, Representar, Modificar, Compilar, Verificar, Ejecutar, Aplicar, Proponer (Pp), Negociar, Proveer (Pv)</i>								Aprendizaje Reflexivo			
			O	R	M	C	V	E	A	Pp	N	Pv	O	R	M	C	V	E		A	Pp	N
MACOSC Modelo de administración del conocimiento organizacional como sistema complejo (Subprocesos-estadios)	Uso	Op 1																			Ar 1	
		Op 2																				Ar 2
		Op 3																				Ar 3
		... Op n																				... Ar n
	Adecuación	Op 1																				Ar 1
		Op 2																				Ar 2
		Op 3																				Ar 3
		... Op n																				... Ar n
	Contribución	Op 1																				Ar 1
		Op 2																				Ar 2
		Op 3																				Ar 3
		... Op n																				... Ar n

Figura 6: El modelo de administración del conocimiento MACOSC-IASC

Fuente: Elaboración propia.

El estudio empírico

En esta sección se describe la aplicación del diseño de investigación para el estudio empírico de la pertinencia del modelo de administración del conocimiento MACOSC-IASC en una pequeña organización (COC) que desarrolla software, para el caso en que se propone adecuar programas de código abierto (PCA). Se inicia con una breve descripción contextual y se describen las diversas iteraciones del estudio.

Perspectiva general de la pequeña organización sujeto de estudio

En 1995 COC inició la comercialización de servicios y tecnologías propietarias CORBA para implementar sistemas distribuidos en México. En noviembre de 1998, logró iniciar un proyecto piloto

³⁵ Es pertinente notar que con el modelo de administración del conocimiento MACOSC-IASC se observa la interacción entre el método IASC y la teoría representada por el modelo MACOSC.

en la empresa de telecomunicaciones de mayor tamaño del país. De junio de 1999 a agosto de 2000, concursó y logró la asignación de un contrato de venta de productos, transferencia tecnológica y consultoría para la implementación de un sistema distribuido de misión crítica en múltiples áreas de la empresa. El proyecto concluyó en enero de 2010. En junio de 2005 introdujo la tecnología en la segunda empresa de telecomunicaciones de mayor tamaño del país. En noviembre de 2007 inició un proyecto de misión crítica en esta empresa, que fue exitosamente implementado en la subsidiaria de México y posteriormente en las subsidiarias de diversos países de Centro América. El proyecto concluyó en febrero de 2008. Los sistemas distribuidos implantados en ambas empresas se encuentran actualmente en producción. En octubre de 2011 COC, debido al cambio recurrente en las condiciones comerciales de clientes³⁶ y proveedores³⁷, aceptó iniciar un proyecto piloto interno para evaluar la posibilidad de adoptar programas de código abierto que le permitieran por un lado disminuir el grado de dependencia y aumentar el de interdependencia como variables cualitativas, y por otro le permitieran implementar sistemas distribuidos de misión crítica. El primer paso sería utilizar el estándar de la industria CORBA.

Perspectiva general de CORBA

La arquitectura de sistemas distribuidos denominada *Common Object Request Broker Architecture* o CORBA es una especificación y se considera un estándar de la industria de tecnologías de información, que ha sido utilizada a escala global. Este estándar se define y aprueba por los miembros de la organización Object Management Group (OMG). Sin embargo, dichos grupos no son grupos de ingeniería de desarrollo de productos. La versión 1.0 de CORBA se liberó en octubre de 1991. En marzo de 2012 la OMG aprobó la versión 3.3 de CORBA (OMG, 2012a). Organizaciones independientes son las que proveen la implementación del estándar y lo promueven como un producto denominado Object Request Broker (ORB) al que asignan un nombre, por ejemplo: Orbix, VisiBroker, JacORB, etcétera. La tecnología CORBA es utilizada en diversos sectores de la industria; por ejemplo, las organizaciones que requieren sistemas de gran tamaño y resistencia la han utilizado para manejar volúmenes enormes de invocaciones para acceder a datos y servicios con la garantía de un alto nivel de continuidad y de disponibilidad. Por ello se consideran de misión crítica para la organización, véanse algunos casos de éxito en (OMG, 2006). En la actualidad CORBA se considera una tecnología madura, que está en la fase de reemplazo³⁸ o de complementación con *SOA* y *web services*. Sin embargo, diversas implementaciones de servidores de aplicaciones que permiten la integración de aplicaciones vía *web services*, *DDS*, etc., así como implementaciones del estándar para sistemas incrustados que requieren tiempos de respuesta en tiempo real y de alto rendimiento³⁹, utilizan en la actualidad para su implementación esta tecnología.

Perspectiva general de la investigación

Laboré en COC y creé una infraestructura CORBA (*framework*) de integración de aplicaciones empresariales (Orozco, 2001), que utilicé durante la realización de los principales proyectos descritos para implementar los sistemas distribuidos empresariales de misión crítica, en las empresas de telecomunicaciones de mayor tamaño en México. Renuncié a COC en enero de 2010, no obstante mantuve contacto con la dirección. Propuse, en julio de 2011, a la dirección de COC su participación como sujeto de estudio en la realización de la investigación. Aceptó participar en el estudio en octubre de 2011.

Para la investigación se planteó un doble objetivo en la fase de intervención, del diseño mixto investigación-acción de sistemas complejos IASC:

- Contrastar y explicar causalmente la hipótesis:

³⁶ Para reducir el costo de licencias y servicios.

³⁷ Para reducir comisiones y pretender tomar el control directo de los clientes.

³⁸ Véase por ejemplo el artículo de Henning (2008).

³⁹ Sistemas descritos en inglés como: Real-time, embedded, and high-performance systems.

Si La pequeña organización que desarrolla software (COC), para el caso en que se propone adecuar los programas de código abierto JacORB v 2.3.1, formula sus acciones con el modelo MACOSC, *entonces* va a conseguir en grado elevado su objetivo.

- Que la organización sujeto de estudio (COC), para el caso de los programas de código abierto JacORB versión 2.3.1, formule sus acciones con el modelo de administración del conocimiento (MACOSC) para lograr el objetivo de realizar la lista de adecuaciones que se consideran necesarias, para crear y utilizar servidores CORBA de alto rendimiento (SCAR) que sean funcionales en un entorno de producción organizacional.

Contrastación y explicación causal de la hipótesis

Para proceder a contrastar y *explicar causalmente* la hipótesis, utilicé el modelo de administración del conocimiento MACOSC-IASC. Participé de manera práctica y activa en la organización (COC) que decidió adoptar programas de código abierto, y se propuso adecuarlos a las condiciones particulares de sus problemas y oportunidades específicos. Así, con el objetivo de crear y utilizar servidores CORBA de alto rendimiento (SCAR), se implantó el modelo de administración del conocimiento MACOSC en la organización. Para realizar la investigación, se siguieron las cuatro actividades de cada subproceso, que conforman el proceso iterativo e incremental del IASC.

Se estableció como condición necesaria, para contrastar y explicar causalmente la hipótesis, la siguiente:

- Que por administración del conocimiento organizacional se entienda que en la organización se articulan y ejecutan los pasos propuestos en el MACOSC para acceder a los subprocesos-estadios:
- **Uso** de los programas de código abierto JacORB v2.3.1,
- **Uso** de los programas de código abierto git v1.7.4.1,
- **Adecuación** del control de versiones de JacORB v2.3.1 de CVS a git v1.7.4.1,
- **Contribución** a JacORB v2.3.1 a partir de los resultados obtenidos en las fases anteriores de **Uso** (JacORB, git) y en la fase de **Adecuación**.

Guiados por el objetivo de crear servidores CORBA de alto rendimiento (SCAR) que sean funcionales en un entorno de producción organizacional.

Para la aplicación del diseño *investigación-acción de sistemas complejos* IASC, este se cruza con el modelo de administración del conocimiento MACOSC. Se identifica como *unidad de análisis* al modelo MACOSC-IASC que resulta del cruce; y como *unidad de observación* al complejo empírico de estudio conformado por las tríadas: objetivos, acciones, resultados.

Plan de acción para la contrastación y explicación causal

Las acciones que se siguen para contrastar y *explicar causalmente* la hipótesis en el estudio empírico se visualizan en el modelo MACOSC-IASC, en la tabla de la figura 7 donde se muestra la lista correspondiente de acciones propuestas. Esta *explicación causal* es una *atribución*⁴⁰ a la realidad de transformaciones observadas, en el caso de que estén en correspondencia con la inferencia establecida en la hipótesis⁴¹.

Diagnóstico de la oportunidad

Entreverar la perspectiva cualitativa y cuantitativa

La perspectiva cualitativa tiene varias facetas para disminuir el grado de dependencia e incrementar el de interdependencia con clientes y proveedores.

- Mantener como guía de todo el proceso el objetivo de negocio de la pequeña organización que

⁴⁰ Esta atribución no implica ningún tipo de hipótesis, porque no otorga “realidad objetiva” a tales transformaciones (García, 2008: 189).

⁴¹ Véase la sección 2.2.

considerado como el sistema de acciones eficaces, eficientes y reiteradas en ejecución⁴⁴. Así la organización, el conjunto de relaciones que la caracterizan como sistema, se observa como una totalidad organizada en su acción eficaz y en las *implicaciones entre acciones* en un momento dado. El análisis de sus procesos a través del tiempo hace observable esta práctica administrativa en los niveles de efectividad reiterada y, posteriormente, de eficiencia en el sistema organizacional⁴⁵ para transitar por los subprocesos-estadios del MACOSC cruzado con el IASC, como se muestra en la figura 7.

Identificar niveles de organización de la “realidad estructurada”

Son tres niveles:

- Uso
- Adecuación
- Contribución

Identificar la interacción entre los niveles de organización de la “realidad estructurada”

Se consideran dos interacciones principales entre niveles:

- Uso - Adecuación y
- Adecuación - Contribución

Calendarizar la evolución, por reorganizaciones sucesivas de cada nivel de organización del sistema

Los periodos de tiempo en que transcurre el tránsito por los subprocesos-estadios:

- Uso de JacORB original: octubre de 2011 a diciembre de 2011
- Uso de git original: enero de 2012 a junio de 2012
- Adecuación del control de versiones de JacORB, de CVS a git: julio de 2012 a diciembre de 2012. Fecha esta última en que concluyó la investigación.

Planeación de las acciones

El plan de acción, guiado por la teoría e hipótesis inicial, consistió de identificar cursos alternativos de acción para aprovechar la oportunidad, seleccionar uno de ellos, y establecer los instrumentos de observación del complejo empírico.

Identificar cursos alternativos de acción para aprovechar la oportunidad

Se registraron las opciones de implementación de CORBA con diversos programas de código abierto. En la figura 7 se muestran las opciones propuestas.

Seleccionar el curso de acción para aprovechar la oportunidad

- Racionalización de la elección de JacORB
- Racionalización de la elección de git

El plan de acción guiado por la teoría se muestra en la tabla de la figura 7 en el modelo MACOSC-IASC.

⁴⁴ Por ejemplo: utilizar exitosamente SCAR, creados con JacORB, en producción; resolver requerimientos de problemas de los SCAR en producción.

⁴⁵ Como se pretende ilustrar en la figura 2.

Establecer los instrumentos de observación del complejo empírico

Consiste de un sistema de observación tecnicada que proporciona observaciones neutrales que permiten comparar la teoría en su evolución en el tiempo (González C. E., 2011). Para ello de la *unidad de observación* identificamos:

- Acciones de los agentes involucrados
- Objetivos de la organización
- Sistema de acciones (MACOSC) en ejecución
- Resultados de ejecutar el sistema de acciones (MACOSC)
- Programas de código abierto

Primera iteración del MACOSC-IASC

Los resultados y lecciones aprendidas en la primera iteración del estudio, al implantar el modelo MACOSC-IASC para intervenir la organización, con la pretensión de propiciar la emergencia del conocimiento organizacional como un sistema complejo de acciones en ejecución para crear SCAR con JacORB en el subproceso-estadio **Uso** son los siguientes:

- Resultados empíricos

La organización que interviene para implementar el modelo MACOSC para la administración del conocimiento organizacional accede al estadio **Uso** y concluye con éxito la acciones de ese estadio, que van de *Observar* hasta *Aplicar*. Se logra el objetivo organizacional de construir servidores de alto rendimiento (SCAR) con programas de código abierto JacORB v 2.3.1. En la figura 8 se muestra el comportamiento del SCAR.

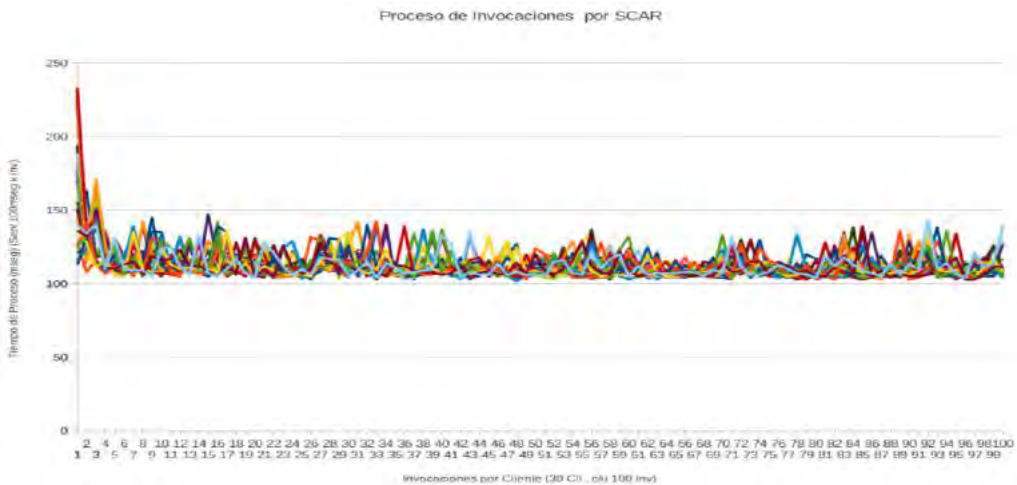


Figura 8: Comportamiento de Servidores CORBA de Alto Rendimiento (SCAR)

Fuente: *Elaboración propia.*

- Diferencias del subproceso-estadio **Uso** respecto de utilizar los PCA como “caja negra”: al controlar la construcción del ejecutable de los PCA gradualmente incrementa la independencia.
- Al concluir con éxito la actividad *Aplicar*, se vislumbra una alternativa emancipadora respecto del código propietario que afectará la relación con clientes y proveedores.
- Resultados teóricos

La consecuencia de corroborar la hipótesis para el estadio-subproceso **Uso** es que las acciones propuestas en el **MACOSC** son efectivas. Esto tiene como consecuencia que el

MACOSC es pertinente para la administración del conocimiento organizacional en la pequeña organización que desarrolla software en el caso que se propone hacer **Uso** de los programas de código abierto.

- Al concluir con éxito la actividad *Aplicar*, se confirma que se accede al subproceso-estadio **Uso**. Asimismo, se confirma la inferencia de la explicación causal.

Segunda iteración del MACOSC-IASC

Los resultados y lecciones aprendidas en la segunda iteración del estudio, al implantar el modelo MACOSC-IASC para intervenir la organización, con la pretensión de propiciar la emergencia del conocimiento organizacional como un sistema complejo de acciones en ejecución para realizar con git el control de versiones distribuido, en el subproceso-estadio **Uso** son los siguientes:

- **Resultados empíricos**
La organización que interviene para implementar el modelo MACOSC para la administración del conocimiento organizacional accede al estadio de **Uso** y concluye con éxito la acciones de ese estadio⁴⁶, que van de *Observar* hasta *Aplicar*. Se logra el objetivo organizacional de controlar un sistema de archivos con programas de código abierto git v 1.7.4.1.
 - Al concluir con éxito la actividad *Aplicar*, se vislumbra una alternativa de participación en la red de innovación que desarrolla git.
 - Diferencias del subproceso-estadio **Uso** respecto de utilizar los PCA como “caja negra”: Al controlar la construcción del ejecutable de los PCA, gradualmente incrementa la independencia.
- **Resultados teóricos**
La consecuencia de corroborar la hipótesis para el estadio-subproceso **Uso** es que las acciones propuestas en el **MACOSC** son efectivas. Esto tiene como consecuencia que el **MACOSC** es pertinente para la administración del conocimiento organizacional en la pequeña organización que desarrolla software en el caso que se proponga hacer **Uso** de los programas de código abierto git.
 - Al concluir con éxito la actividad *Aplicar*, se confirma que se accede al Subproceso-Estadio **Uso**. Asimismo, se confirma la inferencia de la explicación causal.El resultado de seguir el plan de acción, para la segunda iteración, guiado por la teoría se muestra en la tabla de la figura 9 en el modelo MACOSC-IASC.

Tercera iteración del MACOSC-IASC

Los resultados y lecciones aprendidas en la tercera iteración del estudio, al implantar el modelo MACOSC-IASC para intervenir la organización, con la pretensión de propiciar la emergencia del conocimiento organizacional como un sistema complejo de acciones en ejecución para realizar el control de versiones de JacORB con git, en el subproceso-estadio **Adecuación** son los siguientes:

- **Resultados empíricos**
La organización que interviene para implementar el modelo **MACOSC** para la administración del conocimiento organizacional, accede al estadio de **Adecuación** y concluye con éxito la acciones de ese estadio⁴⁷ que van de *Observar* hasta *Aplicar*. Resaltando que en estas acciones se incluye *Modificar*. Se logra el objetivo organizacional de controlar el sistema de archivos de código fuente de JacORB, lo que abarca por lo menos un período de un año hasta la liberación de JacORB v 2.3.1 con programas de código abierto git v 1.7.4.1.

⁴⁶ Véase 2.2 Modelo para la administración del conocimiento organizacional.

⁴⁷ Véase 2.2 Modelo para la administración del conocimiento organizacional.

- Al adecuar la construcción del ejecutable de los PCA a los procesos de la organización, gradualmente incrementará la interdependencia con la red de innovación que desarrolla JacORB.

Lo anterior se suma a lo ya logrado en el estadio anterior:

- Diferencias del subproceso-estadio **Uso** respecto a utilizar los PCA como “caja negra”: Al controlar la construcción del ejecutable de los PCA, gradualmente incrementa la independencia.
 - Al concluir con éxito la actividad **Aplicar**, se vislumbra una alternativa emancipadora respecto al código propietario, que afectará la relación con clientes y proveedores.
- Resultados teóricos

La consecuencia de corroborar la hipótesis para el estadio-subproceso **Adecuación** es que las acciones propuestas en el **MACOSC** son efectivas. Esto tiene como consecuencia que el **MACOSC** es pertinente para la administración del conocimiento organizacional en la pequeña organización que desarrolla software en el caso que se proponga hacer la **Adecuación** de controlar las versiones de los programas de código abierto JacORB con git.

- Al concluir con éxito la actividad *Aplicar*, se confirma que se accede al Subproceso-Estadio **Adecuación**. Asimismo, se confirma la inferencia de la explicación causal.

El resultado de seguir el plan de acción, para las tres iteraciones, guiadas por la teoría se muestra en la tabla de la figura 9 (véase más abajo) en el modelo MACOSC-IASC.

Discusión

A partir de argumentar la pertinencia de la propuesta de una nueva concepción del conocimiento organizacional y su consecuente modelo de administración del conocimiento MACOSC, que contempla la necesidad de estas organizaciones, y de argumentar la pertinencia de la propuesta de un diseño de investigación-acción de sistemas complejos IASC, que es utilizado para realizar el estudio del complejo empírico observado con la nueva concepción, concluyo que:

- Concebir al conocimiento organizacional como un sistema complejo de acciones eficaces, eficientes y reiteradas en ejecución por los diversos agentes involucrados en la organización, sistema complejo que emerge cuando el sistema de acciones en ejecución cumple las condiciones e incluye las causalidades descritas, permite que en la práctica administrativa dicho conocimiento organizacional sea observable en la dimensión colectiva en el nivel reiterado de eficacia y eficiencia del sistema de acciones.
- La creación y diseño del método investigación-acción de sistemas complejos IASC, me permitió abordar la investigación de la pertinencia del modelo de administración del conocimiento MACOSC, que se implantó en la organización sujeto de estudio.

A partir de la evidencia que el estudio empírico ofreció, con los resultados de intervenir con el modelo MACOSC-IASC una de estas organizaciones en México, y a partir de las explicaciones del funcionamiento del sistema complejo, concluyo que:

- Al implantar el MACOSC en la organización sujeto de estudio, ésta accedió a los subprocesos-estadios de **Uso** y **Adecuación**. Se corroboró la hipótesis en este caso.

La consecuencia de corroborar la hipótesis, para los subprocesos-estadios de **Uso** y **Adecuación**, es que las acciones propuestas en el modelo MACOSC propiciaron la consecución en grado elevado del objetivo organizacional de usar y adecuar los programas de código abierto. Lo anterior tiene como consecuencia que el modelo fue pertinente para la administración del conocimiento en la organización sujeto de estudio.

REFERENCIAS

- Ali, N., Beecham, S., Mistrik, I. (2010). *Architectural Knowledge Management in Global Software Development: A Review*. ICGSE '10, Proceedings of the 2010 5th IEEE International Conference on Global Software Engineering, pp. 347-352, IEEE Computer Society Washington, DC, USA.
- Amozurrutia, J. A. (2011). *Complejidad y Ciencias Sociales: un modelo adaptativo para la investigación interdisciplinaria*. CEIICH, UNAM, México.
- Avison, D., Francis, L., Myers, M., Nielsen, P. A. (1999). *Action Research*. Communications of the ACM, Vol. 42 No. 1, pp. 94-97.
- Avison, D., Baskerville, R., Myers, M., Wood-Harper, T. (1999). *IS Action Research: Can We Serve Two Masters?* ICIS 99, Proceedings of the 20th international conference on Information Systems, AIS Atlanta, GA, USA, pp. 582-585.
- Baskerville, R. (1999). *Investigating Information Systems with Action Research*. Communications of AIS, Vol. 2, article 19.
- Beyerlein, M., Kennedy, F. (2008). "Managing Intangible Capital." En *21st Century Management A Reference Handbook*, Wankel, C. (ed.). (Volume 2, pp. 401-410). California, USA: SAGE Publications.
- Brown, A., Wilson, G. (Editors). (2011). *The Architecture of Open Source Applications. Volume I: Elegance, Evolution and a Few Fearless Hacks*. Disponible en: <http://www.aosabook.org/en/index.html>. (Se accedió al sitio 02-enero-2013).
- (2012). *The Architecture of Open Source Applications. Volume II: Structure, Scale, and a Few More Fearless Hacks*. Disponible en: <http://www.aosabook.org/en/index.html>. (Se accedió al sitio 02-enero-2013).
- Carrillo, L. P. (2008). *Sociedad del Conocimiento, Academia, administración, complejidad y tecnología*. UNAM - Facultad de Ciencias Políticas y Sociales. México: SITESA.
- Chiavenato, I. (2000). *Introducción a la Teoría General de la Administración*. Quinta Edición. México: McGraw-Hill Interamericana.
- Davenport, T., Prusak L. (2000). *Excerpt of Working Knowledge: How organizations Manage What They Know*. Ubiquity, Volume 2000 Issue August, Article No. 6, ACM, N.Y., USA. <http://ubiquity.acm.org/article.cfm?id=348775>.
- Drucker, P. (1974). *Management*. New York, USA: Harper and Row Publishers.
- Easterbrook et al. (2008). "Selecting Empirical Methods for Software Engineering Research." En *Guide to Advanced Empirical Software Engineering*, Shull et al. (eds.), (pp. 285-311). London, UK: Springer-Verlag.
- Econsultancy.com, Ltd. (2011). *Android takes 52.5% of smartphone market*. Econsultancy, Ltd., <http://econsultancy.com/uk/blog/8279-android-doubles-market-share-to-take-52-5-of-smartphone-market>. (Se accedió al sitio el 24-Ene-2012).
- Federman, M. (2006). "The Penguinist Discourse: A Critical Application of Open Source Software Project Management to Organization Development." *Organization Development Journal* 24(2).
- Flichy, P. (2008). *Internet, un outil de la démocratie?* La Vie Des Idees. Disponible en: <http://www.laviedesidees.fr/Internet-un-outil-de-la-democratie.html?lang=fr>. (Se accedió al sitio 02-enero-2013).
- Free Software Foundation Inc. (1991). *GNU General Public License*. Free Software Foundation Inc. <http://www.gnu.org/licenses/gpl.txt>. (Se accedió al sitio el 30-Abr-2010).
- García, R. (2008). *Sistemas complejos: conceptos, método y fundamentación epistemológica de la investigación interdisciplinaria*. Barcelona: Gedisa.
- Goldman, R., Gabriel, R. P. (2005). *Innovation Happens Elsewhere: Open Source as Business Strategy*. CA, USA: Morgan Kaufmann Publishers, Elsevier. Disponible en: <http://www.dreamsongs.com/IHE/>. (Se accedió al sitio 02-enero-2013).
- González, C. E. (2011). *Commensurar la ciencia: observaciones tecnificadas y comparabilidad*. Tesis de Maestría, Posgrado en Filosofía de la Ciencia, UNAM, México.

- Henning, M. (2008). *The Rise and Fall of CORBA*. Communications of the ACM, pp. 53-57, Vol. 51, Num. 8, NY, USA.
- Hernández, R., Fernández, C., Baptista, P. (2003). *Metodología de la investigación*. 3a. Ed., México: McGraw-Hill.
- Herrera, L. V., Ramírez, P. y May, G. (2012). “Organizaciones artísticas en México: modelo e indicadores de capital intelectual”. *Contaduría y Administración* 57(3), 259-275. Disponible en: <http://contaduriayadministracionunam.mx/articulo-2-266-32.html>. (Se accedió al sitio 02-enero-2013).
- IDC, Inc. (2010). *Worldwide Server Market Rebounds Sharply in Fourth Quarter as Demand for Blades and x86 Systems Leads the Way, According to IDC*. IDC, Inc. Documento número: prUS22224510, (Feb 2010), Research Press Release <http://www.idc.com/>. (Se accedió al sitio el 05-May-2010).
- (2013). *Worldwide Quarterly Mobile Phone Tracker*. IDC, Inc., (14 febrero 2013), Research Press Release <http://www.idc.com/getdoc.jsp?containerId=prUS23946013#UUJ75VHP5Ls>. (Se accedió al sitio el 14-febrero-2013).
- Kant, I. (1995). *Crítica a la razón pura*. Alfaguara, Madrid, España.
- Kerlinger, F., Lee, H. (2002). *Investigación del comportamiento: métodos de investigación en Ciencias Sociales*. 4a. Ed., (trad. Pineda, L., Mora, I.), (rev. Balbás, C., Vadillo, G.). México: McGraw-Hill.
- Khun, T. (2007). *La estructura de las revoluciones científicas*. FCE, (trad. Solís, C.), Tercera edición (2006), primera reimpresión, México.
- Kock N.F. Jr., McQueen R. J., Scott, J. L. (1997). “Can action research be made more rigorous in a positivist sense? The contribution of an iterative approach.” *Journal of Systems and Information Technology* 1(1), 1 – 23.
- Laudan, L. (1993). *La ciencia y el relativismo*. Alianza Editorial, Madrid.
- Le Blanc, B. Ermine, J-L (2007). “A Shannon's Theory of Knowledge. Ser. on Innovation and Knowledge Management Series.” En *Creating collaborative advantage through knowledge and innovation*, Suliman Al-Hawamdeh (ed.), 51-67. World Scientific Pub. Disponible en: <http://books.google.com.mx/books?id=qWQCfXpec68C>.
- The Linux Foundation. (2008). *Estimating the Total Development Cost of a Linux Distribution*. Disponible en: <http://www.linuxfoundation.org/sites/main/files/publications/estimatinglinux.html>. (Se accedió al sitio 02-enero-2013).
- Madeiras dos Santos, S., Horta, G. (2009). *Action Research Use in Software Engineering: an Initial Survey*. ESEM 2009 Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement; IEEE Computer Society Washington, DC, USA.
- Maass, M., Amozurrutia, J., Almaguer, P., González, L., Meza, M. (2012). *Sociocibernética, cibercultur@ y sociedad*. CEIICH, UNAM, México.
- Maturana, H., Varela, F. (2003). *El árbol del conocimiento - Las bases biológicas del entendimiento humano*. Argentina: Lumen - Editorial Universitaria.
- Morales, M., Ventura, T., Oktaba, H., Torres, R. (2012). *From MoProSoft Level 2 to ISO/IEC 29110 Basic Profile: Bridging the Gap*. XV Ibero-American Conference on Software Engineering, CIBSE2012, Buenos Aires, Argentina. Disponible en: http://cibse.inf.pucrio.br/CIBSEpapers/pdf_counter.lp?cibse=CIBSE12&file_name=paper_7.pdf. (Se accedió al sitio 02-enero-2013).
- Morin, E. (2004). *Introducción al pensamiento complejo*. Primera reimpresión. México: Editorial Gedisa Mexicana.
- Nielsen T. M. (2008). *The Evolving Nature of Work Teams. Changing to Meet the Requirements of the Future*. en Wankel, C. (ed.): *21st Century Management A Reference Handbook*, Volume 2: 490-499. SAGE Publications Inc., California, USA.
- Nonaka, I., Takeuchi, H. (1995). *The Knowledge-Creating Company - How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press.

- Oktaba, H., Piattini, M., Pino, F.J., Orozco, M.J., Alquicira, C. (2009). *COMPETISOFT Mejora de Procesos Software para Pequeñas y Medianas Empresas y Proyectos*. México: Alfaomega Grupo Editor.
- Object Management Group Inc. (2012). *CORBA Success Stories*. Object Management Group Inc. <http://www.corba.org/success.htm> (Se accedió al sitio 27-Jul-2012)
- (2012). *UPDATED CORBA 3.3 RTF report* Report of the CORBA/ZIOP Revision Task Force to the OMG Platform Technical Committee, 12 March 2012, <http://www.omg.org/cgi-bin/doc?ptc/12-03-15.pdf>, MA, USA. <http://www.omg.org/cgi-bin/doc?ptc/2012-03-15> (Se accedió al sitio 27-Jul-2012)
- Open Source Initiative Inc. (2013). *Open Source Licenses*. Open Source Initiative Inc. Disponible en: <http://www.opensource.org/licenses>. (Se accedió al sitio 02-enero-2013).
- Orozco, O. (2001). *Simple Enterprise Application Integration Framework*. Conferencia impartida en: IONA World 2001, Miami, FL, USA.
- (2011). *Representación y gestión del conocimiento de los programas de código abierto*. En Memorias del Primer Congreso de Alumnos de Posgrado de la UNAM, Área de Ciencias Sociales, Aporte al Conocimiento, poster 558, Coordinación de Estudios de Posgrado, UNAM. http://www.posgrado.unam.mx/publicaciones/1congreso/03_2_CS.pdf.
- (2013). *Un modelo de administración del conocimiento para las pequeñas organizaciones que desarrollan software, aplicable al caso de los programas de código abierto*. Tesis de Doctorado en Ciencias de la Administración, Programa de Posgrado en Ciencias de la Administración, Facultad de Contaduría y Administración, UNAM. <http://132.248.9.195/ptd2013/mayo/075288730/Index.html>.
- Piaget, J., García, R. et al. (1997). *Hacia una lógica de significaciones*. Barcelona: Gedisa.
- Polanyi, M. y Sen, A. (2009). *The tacit dimension with a new foreword by Amartya Sen*. USA: The University of Chicago Press.
- Poteete, A. R., Janssen, M. A. y Ostrom E. (2012). *Acción colectiva, bienes comunes y múltiples métodos en la práctica*. trad. Niles L. B., Merino, L. UNAM, CEIICH, CRIM, FCPS, FE, IIEc, IIS, PUMA; IASC, CIDE, Colsan, CONABIO, CCMS, FCE, UAM, México.
- Rivas, L., et al. (2009). *Efectos de la teoría de la complejidad en la gestión ambiental en México*. Primera Edición. Instituto Politécnico Nacional, Centro Mario Molina, México.
- Solleiro et al. (coordinador). (2009). *Gestión del conocimiento en centros de investigación y desarrollo de México, Brasil y Chile*. Programa de Investigación sobre Economía del Conocimiento en América Latina y el Caribe IDRC-Flacso, FLACSO México, Centro Internacional de Investigaciones para el Desarrollo, México.
- TOP500.Org (2010). *Top 500 Supercomputer Sites*. TOP500.Org <http://www.top500.org/stats/list/34/osfam>. (Se accedió al sitio el 03-May-2010).
- Tucker, A., Morelli, R., De Silva, Ch. (2011). *Software Development an Open Source Approach*. USA: CRC-Press.
- Wang, Y. (2008). *Software Engineering Foundations - A Software Science Perspective*. USA: Auerbach Publications.

SOBRE EL AUTOR

Octavio Orozco y Orozco: Entró en la dignidad de Doctor en Ciencias de la Administración, con mención honorífica, en noviembre de 2013; el grado le fue otorgado por la Universidad Nacional Autónoma de México. Actualmente es Director General de la empresa Computadoras, Objetos y Comunicaciones S.A. de C.V.