



# Harvard Business Review

América Latina

## El imperio de la colaboración

por Philip Evans y Bob Wolf

---

Julio 2005

---

Reimpresión R0507H-E

Los grandes esfuerzos grupales no tienen por qué ser milagrosos o fortuitos. Un ambiente diseñado para producir transacciones abundantes y baratas desata colaboraciones que atraviesan las barreras organizacionales.

# El imperio de la colaboración

por Philip Evans y Bob Wolf

Los líderes corporativos que buscan el crecimiento, el aprendizaje y la innovación pueden encontrar la respuesta en un lugar sorprendente: la comunidad de software de fuente abierta. Sin saberlo quizás, las personas que le trajeron el sistema operativo Linux son virtuosos en la práctica de nuevos principios de trabajo, que producen equipos energizados y bajos costos. Tampoco están solos.

Cualquiera sea la forma en que se le mida, Linux es un producto poderosamente competitivo. Se estima que hay más servidores que funcionan con Linux que con cualquier otro sistema operativo. Ha sobrepasado a UNIX en tanto oferta comercial, y sus ventajas se extienden más allá del costo y de la calidad, y van hacia la rapidez con que se le aumenta y mejora. Aunque sus partidarios discuten sobre sus limitaciones técnicas y el tratamiento de su propiedad intelectual, sí están de acuerdo en que el éxito del producto no se puede separar de su modo de producción único. Específicamente, Linux es la creación de una comunidad esencialmente voluntaria y auto-organizada compuesta por miles de programadores y empresas. La mayoría de los líderes estaría dispuesta a vender a sus abuelas con tal de contar con fuerzas de

trabajo que colaboren de forma tan eficiente, creativa y sin fricciones como los autodenominados “hackers” de Linux.

Pero Linux es software, y el software es un poco extraño. Toyota, sin embargo, es una empresa como cualquier otra; eso sí, como cualquiera que ha sido clasificada sistemáticamente como una de las organizaciones de mejor desempeño en el mundo. El fabricante de automóviles es hace tiempo un líder en calidad y en la producción ligera, y el éxito del híbrido Prius ha establecido su reputación como innovador. Hemos descubierto que los métodos de gestión de Toyota se parecen, en algunos de sus aspectos fundamentales, al funcionamiento de la comunidad Linux; el Sistema de Producción de Toyota (TPS, por sus siglas en inglés) le debe parte de su cacareada receptividad a rasgos de fuentes abiertas. De hecho, Toyota se está transformando en un híbrido que comprende una jerarquía convencional y una red del tipo Linux que se auto-organiza.

(A lo largo de este artículo, usamos el término “Linux” para referirnos a la comunidad de software de fuentes abiertas, que desarrolló y continúa refinando el sistema operativo y otros programas de fuentes abiertas. Usamos

el término “Toyota” para referirnos al TPS, que se compone de Toyota y sus proveedores directos en Japón y Estados Unidos, el “nivel uno” en la jerga automotriz).

Toyota es notablemente similar a Linux en la forma en que amalgama características clave tanto de los mercados como de las jerarquías. Al igual que los mercados, las comunidades de Toyota y Linux pueden auto-organizarse, pero a diferencia de ellos, no usan efectivo o contratos en las coyunturas cruciales. Al igual que las jerarquías, Toyota y Linux disfrutaban de bajos costos de transacción, pero a diferencia de ellas, sus miembros pueden pertenecer a muchas organizaciones diferentes (o a ninguna) y no están constreñidos por roles y responsabilidades específicos y predefinidos. Además, tal como ocurre en las jerarquías, los miembros comparten un propósito común, pero ese propósito emana de una automotivación, más que de los incentivos externos o de las sanciones que por lo general imponen las jerarquías. En estos aspectos, Toyota y Linux representan lo mejor de ambos mundos. El análisis de sus características comunes sugiere la forma en que las organizaciones de alto desempeño siguen siendo productivas e inventivas, incluso bajo condiciones durísimas. Creemos que esas lecciones pueden mejorar de manera significativa la forma en que el trabajo es realizado en la mayoría de las organizaciones.

## Martes, 2 de diciembre de 2003

Cerca de la medianoche, Andrea Barisani, administrador de sistemas en el departamento de física de la Universidad de Trieste, descubrió que un atacante había asaltado el servidor Gentoo Linux de la institución. Descubrió que la fisura estaba radicada en un punto vulnerable del kernel de Linux y que había otra en rsync, un mecanismo de transferencia de archivos que replica datos entre computadoras de forma automática. Se trataba de un ataque serio; cualquier penetración a rsync podría comprometer a los archivos de miles de servidores alrededor del mundo.

Barisani despertó a algunos colegas, quienes lo pusieron en contacto con Mike Warfield, un investigador senior de Internet Security Systems, en Atlanta, y con Andrew “Tridge” Tridgell, un conocidísimo programador Linux, en Australia, sobre cuya tesis doctoral estaba basado rsync. Ellos transfirieron el mensaje de Barisani (transformándolo en anónimo por razones de seguridad) a otro australiano, Martin Pool, quien trabajaba para Hewlett-Packard en Canberra y había sido uno de los líderes en el desarrollo de rsync. Aunque Pool ya no era responsable de rsync (nadie lo era), comenzó inmediatamente a usar el teléfono y a enviar e-mails. Sus preguntas estaban dirigidas a Warfield y a Dave Dykstra (otro de los

que había contribuido tempranamente al desarrollo de rsync y que estaba basado en California) y les inquirió acerca de sus vulnerabilidades. Luego ayudó a Barisani a rastrear la falla, línea por línea.

Al llegar la mañana, hora de Trieste, Pool y Barisani habían encontrado la ubicación precisa de la grieta. Pool se puso en contacto con el actual grupo de desarrollo de rsync, mientras que Barisani se conectó con la asociación informal de aficionados y profesionales que empaquetan Gentoo Linux y envió una temprana advertencia al sitio de Gentoo. Entonces, Pool y Paul “Rusty” Russell (un coterráneo de Canberra empleado en IBM) trabajaron durante la noche australiana para programar un parche, y en cinco horas los usuarios-desarrolladores de Gentoo comenzaron a probar la primera versión. Mientras tanto, Tridge elaboró una descripción de la vulnerabilidad y de su remedio, asegurándose (instado por Pool) de darle el crédito a Barisani y Warfield por sus esfuerzos tras bambalinas. El jueves en la tarde, hora de Canberra, el anuncio y el parche se colocaron en el sitio Web de rsync y se distribuyeron a los usuarios de Linux alrededor del mundo.

Unos pocos días después de la emergencia, ya recuperadas las horas perdidas de sueño, Barisani se ofreció como voluntario para colaborar con Warfield en el establecimiento de un sistema de servidores deliberadamente vulnerables, como un señuelo para que el cracker revelase su identidad.

Nadie autorizó o dirigió este esfuerzo. Nadie –aficionado o profesional– recibió una remuneración por participar o fue sancionado por no haberlo hecho. Nadie arriesgó su empleo si fallaban en detener el ataque. Nadie se calló por completo por miedo a las responsabilidades legales. De hecho, se mantuvo informada a toda la comunidad de usuarios acerca de los acontecimientos. A pesar de la necesidad de contar con la más alta seguridad, un grupo de unas 20 personas, prácticamente desconocidas entre sí, empleadas por una decena de empresas diferentes, viviendo en casi la misma cantidad de zonas horarias y alejándose mucho de sus descripciones de cargo, logró en cerca de 29 horas lo que a colegas en cubículos adyacentes les podría haber tomado semanas o meses.

Es tentador descartar esto como un ejemplo de la locura de los hackers, admirable, por cierto, pero totalmente alejado de los negocios reales. Considere, sin embargo, otro relato.

## Sábado, 1 de febrero de 1997

A las 4:18 de la madrugada estalló un incendio en la planta N° 1 de Kariya, perteneciente a Aisin Seiki, un im-

---

*Philip Evans* (evans.philip@bcg.com) es vicepresidente senior y *Bob Wolf* (wolf.bob@bcg.com) un ejecutivo, de la oficina en Boston, del Boston Consulting Group. Evans es coautor de *Volando en pedacitos: Cómo se transforma la estrategia de negocios en la nueva economía de la información* (Oxford University Press, México, 2000).

portante proveedor japonés de repuestos para automóviles. En minutos, el edificio y prácticamente toda la maquinaria especializada que estaba en su interior quedaron destruidos. La fábrica N°1 de Kariya produce 99% de las válvulas que inyectan líquido de frenos, llamadas válvulas P, para las operaciones japonesas de Toyota, partes requeridas por todo vehículo que es fabricado por la empresa. Y a Toyota, fiel a sus principios de entrega puntual, sólo le quedaba inventario para un día. El Sistema Japonés de Producción de Toyota se enfrentaba a la posibilidad de un cierre total que podía durar meses.

En cosa de horas, los ingenieros de Aisin se reunieron con sus contrapartes en Toyota y los otros proveedores de nivel uno de Toyota. El grupo acordó improvisar tanta producción como fuese posible. A medida que la noticia se propagó a través de la red de proveedores, algunos de los proveedores de nivel dos se ofrecieron voluntariamente para asumir roles de liderazgo. Aisin envió copias de planos de las válvulas a todos los proveedores que las solicitaron y distribuyó todas las herramientas, materias primas y trabajo en proceso que no había sufrido daños y que pudieron rescatarse. Los ingenieros de Aisin y de Toyota ayudaron a establecer improvisadas líneas de producción en 62 locaciones distintas, incluyendo salas de máquinas desocupadas, el propio taller de prototipos de Toyota, e incluso una instalación para máquinas de coser de propiedad de Brother. Denso, el mayor proveedor de Toyota, se ofreció como voluntario para administrar la complicada logística del despacho de las válvulas a Aisin para su inspección, y luego a las paralizadas líneas de ensamblaje de Toyota.

Todos se sorprendieron cuando Kyoritsu Sangyo un pequeño proveedor de nivel dos de electrodos de soldadura, fue el primero en entregar válvulas de calidad de producción a Toyota: 1.000 de ellas y sólo 85 horas después del incendio. Otros lo siguieron rápidamente, y Toyota comenzó a reabrir las líneas de ensamblaje el miércoles. Unas dos semanas después de la detención, toda la cadena de suministro estaba en plena producción. Seis meses más tarde, Aisin distribuyó una guía de respuesta a emergencias, que contenía lecciones sacadas de la experiencia y recomendaba procedimientos para responder a ese tipo de situaciones en el futuro.

No hubo un individuo u organización que planifi-

## Construir redes humanas vibrantes

Las empresas que estén cimentando una colaboración de alto desempeño, deberían seguir estos principios:

### **Despliegue una tecnología de colaboración impregnante.**

Manténgalo simple y abierto: “Pequeños trozos unidos sin apretar”, según la agradable frase de David Weinberger, coautor de *Cluetrain Manifesto*. Las herramientas deberían trabajar juntas a través de estándares comunes y ser tan compatibles como sea posible. Piense en opciones, no en integración; en adaptabilidad, no en eficiencia estática.

**Mantenga el trabajo visible.** A menos que exista una buena razón para no hacerlo, deje que todos vean el verdadero trabajo de los otros. Deje que las personas aprendan a filtrar y a clasificar por sí mismas. No haga abstracciones, no resuma ni encauce. El alimento no procesado es bueno. Póngalo al alcance de todos.

**Construya comunidades de confianza.** Cuando las personas se tienen confianza, es más probable que colaboren libre y productivamente. Cuando las personas confían en sus organizaciones, es más probable que rindan ahora, anticipando recompensas futuras. Y cuando las organizaciones se tienen confianza, es más probable que compartan propiedad intelectual sin atorarse en asuntos legales.

**Piense de manera modular.** La reingeniería era cosa de pensamiento lineal: el manejo del proceso de extremo a extremo, en vez de funciones discretas. Esa aproximación a las cosas promueve la eficiencia enfocada, pero inhibe la variedad y la adaptabilidad. La modularidad es lo inverso: el sacrificio de la eficiencia estática a cambio del valor recombinante de las opciones. Piense en equipos así como en procesos modulares. Mientras más finos, mejor.

**Estimule a los equipos.** Celebre los sacrificios hechos por los equipos en bien de todo el negocio, incluyendo a clientes y proveedores. Desmonte los indicadores de desempeño y las recompensas individualizadas que hacen que las personas se enfrenten entre sí. Las transacciones baratas entre muchos alimentan más innovación que los incentivos caros en provecho de unos pocos. Recompense al grupo, y el grupo lo recompensará a usted.

case estos esfuerzos; más bien, personas y empresas ayudaron en lo que podían. Los competidores colaboraron. En ese momento, a nadie se le retribuyó por contribuir. Meses más tarde, Aisin compensó a las otras empresas por los costos directos de las válvulas que habían entregado. Toyota le entregó honorarios a cada proveedor de nivel uno, basado en sus ventas actuales al fabricante de automóviles, animándolos –pero sin obligarlos– a hacer lo mismo con sus propios proveedores de nivel dos.

Son pocas las comunidades que parecen ser más distintas entre sí que el mundo anárquico, adepto a la caféina e hirsuto de los hackers, y el mundo disciplinado, adicto al té y pulcro de la ingeniería automotriz japonesa. Pero

los paralelos entre estos relatos son llamativos. En ambos hubo individuos que se encontraron y asumieron determinados roles sin un plan o una estructura establecida de comando y control. Una extensa red humana se organizó a sí misma en cosa de horas y se agrupó para luchar contra una amenaza. Personas, equipos y empresas trabajaron en conjunto sin contratos legales o pagos negociados. Y a pesar de no existir garrotes autoritarios ni zanahorias financieras, esas personas trabajaron *como nunca* para resolver el problema.

Ahora bien, obviamente éstas fueron respuestas de emergencia. Pero una mirada a las operaciones diarias de la comunidad Linux y del TPS revela que esas respuestas constituyeron meras intensificaciones de la forma en que las personas ya estaban trabajando

### Obsesión, interacción y un toque leve

Las reglas de los mercados son acerca del dinero y los contratos. Las reglas de las jerarquías tratan de la autoridad y la responsabilización. Pero al centro de las comunidades de Linux y Toyota existen reglas acerca de tres cosas completamente diferentes: cómo trabajan en conjunto los individuos y los pequeños grupos; cómo y cuán extensamente se comunican; y cómo los líderes los guían hacia una meta común.

**Una disciplina de trabajo común.** Las comunidades de Linux y Toyota están compuestas por ingenieros, de manera que sus miembros tienen las mismas destrezas que sus colegas y hablan el mismo lenguaje. Pero estos grupos son mucho más disciplinados y rigurosos en su enfoque hacia el trabajo que otras comunidades de ingenieros. Ambos hacen énfasis en lo granular: prestan atención a los detalles pequeños, eliminan los problemas en su origen, y podan cualquier cosa que parezca excesiva, ya sea que se trate de trabajo, códigos o materiales. Los miembros de Linux comparten una obsesión por escribir códigos mínimos, compilando la salida de cada día antes de proceder al siguiente, y extirpando las fallas de programación a medida que avanzan. Por su parte, los ingenieros de TPS son persistentes en la aplicación de cortos ciclos de ensayos y errores, enfocándose sólo en una cosa a la vez, observando y ubicándose al interior de los procesos actuales. Ambos grupos llevan esos principios a extremos evidentes. Los programadores de Linux reducen los códigos a sus partes más pequeñas, no por buscar la eficiencia computacional sino por la elegancia. Los ingenieros de Toyota rechazan los estampados en el capó del Lexus –aunque no presente fallas y esté completamente dentro de las especificaciones– porque según ellos son poco elegantes.

**Comunicación generalizada y granular.** En las comunidades de Linux y Toyota, la información acerca de los problemas y las soluciones se comparte de manera amplia y frecuente, así como en pequeños incrementos. La mayor parte de las comunicaciones de los hackers

de Linux no es e-mail punto a punto sino que mediante publicaciones en listas abiertas de distribución de correo electrónico (Listserv) que permiten la búsqueda dentro de ellas. Cualquiera puede revisar la historia de las versiones de los códigos y los debates en la Listserv; no los resúmenes ejecutivos o extractos, sino que toda la actividad en bruto. Cada contribución de códigos es probada hasta niveles de estrés por cientos de personas. Tal como lo expresa una famosa metáfora de fuente abierta: “Ante mil ojos, todos los errores son poco profundos”. La contribución media al kernel de Linux es de apenas de una docena de líneas de códigos. La actual versión alpha se recopila cada 24 horas, de manera que los hackers concilian sus esfuerzos casi continuamente. Si alguien trabajara aisladamente durante seis meses en la contribución más brillante, probablemente ésta sería rechazada por no ser compatible con los esfuerzos de los demás.

Análogamente, la filosofía Toyota de mejoramiento continuo está compuesta de mil colaboraciones pequeñas. Es sabido que a los ingenieros de Toyota se les instruye para que “pregunten cinco veces por qué”, para rastrear una cadena de causas y efectos hasta llegar a la raíz del problema. No se trata de un cliché anodino acerca del pensar en profundidad. De hecho es todo lo contrario. El mérito del precepto radica precisamente en su superficialidad. Decir que B es la causa de A es simplista; todas las complejidades de múltiples interacciones quedan reducidas a una causa y efecto únicos. Pero la cadena de pensamientos que se requiere para descubrir que C “es la causa” de B, y que D “es la causa” de C, lleva rápidamente a un nuevo dominio, probablemente al de otra persona. Así que en vez de elaborar complejas soluciones dentro de sus propios dominios, los ingenieros deben buscar soluciones simples más allá de ellos. “Efectuar los porqué de los porqué”, como se le conoce a la práctica, no dice relación en absoluto con la profundidad, sino con la amplitud.

Tal como ocurre con Linux, los protocolos de comunicación de Toyota imponen esta disciplina. Cada reunión trata sólo de un tema y apunta hacia un resultado específico, aunque ello signifique que las mismas personas se reúnan más de una vez en el día. Las lecciones se escriben en un formato estándar en una única hoja de papel A3. Además, todos aprenden a elaborar estos informes, incluyendo el pliegue en el documento que permite ver los puntos principales y esconde los detalles.

**Los líderes como conectores.** En cada nivel, los líderes de Linux y de TPS desempeñan tres roles cruciales. Ellos instruyen a los miembros de la comunidad, a menudo mediante el ejemplo, en las disciplinas que acabamos de describir. Articulan metas claras y simples para cada proyecto, basadas en su visión estratégica. Y conectan a las personas, en virtud de estar ellos mismos muy bien conectados. Los mejores programadores en Linux pro-

cesan a diario más de 300 o 400 e-mails. Fujio Cho, el presidente de Toyota, maneja un número similar de interacciones diarias, el que trasciende la cadena normal de mando.

Ninguna de las dos comunidades trata el liderazgo como una disciplina distinta del hacer. Más bien, la credibilidad y, por lo tanto, la autoridad de los líderes derivan de su competencia en la práctica. Los contenidos de las comunicaciones inconexas de los líderes no eran tanto *acerca* del trabajo, sino que *eran* trabajo en sí (cuando Linus Torvalds, el creador de Linux, despacha sus decenas de e-mails diarios, escribe casi tanto en el lenguaje de programación C como lo hace en inglés).

Ocasionalmente, los líderes sí tienen que efectuar actos tradicionales de liderazgo, como arbitrar en conflictos. Ésa es, sin embargo, una excepción y hasta cierto punto se la considera una falla del sistema. El supuesto por defecto es que, tanto como sea posible, los ejecutivos no gestionan en un sentido tradicional: la red humana se gestiona a sí misma. En Linux, las prioridades de desarrollo no las decide un CEO sino que miles de hackers que expresan su voto eligiendo en qué trabajar y en qué no. Ese tipo de auto-gestión radical no ocurre en Toyota, excepto en casos de emergencias. Pero incluso en las operaciones diarias un trabajador de producción cualquiera que advierte un problema de calidad, puede detener la línea, y los equipos de proyectos poseen amplias atribuciones para usar recursos, tomar decisiones de compras y seguir las prioridades que ellos mismos se fijaron.

Considerados en su conjunto, estos tres principios alimentan un sistema que está en continua adaptación. Una y otra vez, las ideas se formulan en paquetes apretados, susceptibles de ser probados; son comunicadas con mínima atenuación mediante conexiones establecidas, directas, y de persona a persona; y cuando los enlaces están ausentes, son creados cuando ello es necesario por los líderes-ejecutores ampliamente conectados. Esto es disciplina, pero no la disciplina de la conformidad, generada por controles e incentivos. Más bien, se parece a la disciplina de la ciencia. Tal como en las comunidades científicas, estos sistemas descansan sobre procedimientos comunes, reglas comunes para la comunicación y las pruebas, y metas comunes que son comprendidas con claridad. El comportamiento individual es rigurosamente cauto, pero el logro colectivo está marcado por la innovación continua y radical.

## Lo que conocen y cómo lo conocen

En el corazón de Linux y del TPS existe entonces un conjunto de prácticas de trabajo, comunicación y liderazgo que contribuye a una nueva forma de colaboración. Esta colaboración depende también de dos componentes de infraestructura: un fondo de conocimiento compartido y herramientas universalmente disponibles

para hacer que ese conocimiento circule.

**Propiedad intelectual común.** La Licencia Pública General bajo la cual Linux es publicado requiere que todos los distribuidores hagan que sus códigos fuentes estén disponibles libremente, de manera que otros puedan enmendarlo libremente. Este principio viral evita que quede encerrado en productos propietarios. A su vez, esta transparencia rompe la distinción entre productor y usuario. Un “cliente” sofisticado como Andrea Barisani es en realidad un usuario-desarrollador, que arregla defectos y agrega características para su propio beneficio y luego las comparte con todos los otros. Un rol como éste es imposible cuando un código propietario es licenciado por un proveedor comercial. De manera similar, la cadena de suministro de Toyota está basada en el principio de que mientras el conocimiento de un producto (como la copia de un plano) sea propiedad intelectual de alguien, el conocimiento del proceso es compartido. Eso rompe algunas distinciones entre empresas. Los proveedores de Toyota comparten de forma regular extensas lecciones sobre mejoramiento de procesos, tanto vertical como lateralmente, incluso con sus competidores. En Japón los proveedores suelen trabajar exclusivamente para un único fabricante original de equipamientos, por lo que el beneficio colectivo de esa información compartida permanece dentro de la cadena de suministro de Toyota. Pero incluso en Estados Unidos, donde Toyota es sólo uno de varios clientes para la mayoría de sus proveedores de nivel uno, el fabricante automotriz hace lo mismo a través de la Bluegrass Automotive Manufacturers Association, que disemina las mejores prácticas a todos los miembros.

**Tecnología simple y penetrante.** Aunque la información es la savia vital de las comunidades Linux y TPS, sus sistemas de circulación son sorprendentemente rudimentarios. Los desarrolladores de Linux producen software de vanguardia usando tecnología de comunicación no más sofisticada que el e-mail y las Listservs, pero esas herramientas terrenales son usadas por todos. De hecho, se le asigna un valor tan alto a la universalidad que los e-mail de texto simple son la norma, más que los formateados, lo cual asegura que los mensajes aparecerán de la misma forma para todos los destinatarios. Toyota también prefiere tecnología simple y penetrante. En un sistema *kanban*, un recipiente vacío alerta sobre la necesidad de reponer partes; un trozo de cinta de ducto en el piso de la línea de montaje asigna las horas de término para las tareas que se realizan en un vehículo en movimiento. Los problemas de control de calidad en la línea de montaje se anuncian mediante buscapersonas y monitores de TV. Pero todos reciben la alarma. Incluso Ray Tanguay, a la cabeza de Toyota Canadá, recibe un mensaje en su buscapersonas cuando se encuentra una falla en el último despacho de automóviles Lexus en el muelle en Long Beach, California,

o en una bahía de servicio en cualquier lugar de Norteamérica.

## El poder de la confianza y del aplauso

Estos tipos de colaboración extremadamente ricos y flexibles tienen consecuencias psicológicas positivas en los participantes y poderosos y lucrativos efectos en sus organizaciones. Esas consecuencias son conocimientos comunes provechosos, la capacidad de organizar equipos de forma modular, una motivación extraordinaria y altos niveles de confianza.

**Valioso conocimiento semántico.** Una rigurosa disciplina de trabajo, una propiedad intelectual común y el hecho de compartir constantemente se combinan para distribuir el conocimiento ampliamente y de forma relativamente pareja a través de las redes humanas. Ese conocimiento no sólo incluye la información formal y sintáctica que se encuentra en las bases de datos, sino que también el semánticamente provechoso conocimiento ambiguo sobre los contenidos y procesos que es la moneda de cambio de la colaboración creativa. ¿Qué queremos decir al mencionar que el brillo de un estampado de la carrocería no tiene suficiente lustre? Este tipo de pregunta sin respuesta fácil se discute y resuelve continuamente en un millar de colaboraciones de pequeños equipos. Los pensamientos matizados y los enriquecidos vocabularios comunes que emergen, vuelven a alimentar el fondo común de conocimientos, en donde se encuentran disponibles para seguir siendo refinados por toda la comunidad.

**Equipos modulares.** La modularidad es un principio del diseño en virtud del cual un proceso o producto complejo se divide en partes simples conectadas mediante reglas estándares. En los arreglos modulares de equipos humanos, cada equipo se enfoca en tareas pequeñas y simples que conforman un todo mayor. La modularidad permite que una organización efectúe múltiples experimentos paralelos, realizando muchas pequeñas apuestas en lugar de unas pocas grandes. Los proveedores de Toyota se organizaron de esta forma para fabricar válvulas P, operando en parte bajo dirección, pero principalmente ofreciéndose voluntariamente para hacer lo que cada uno conocía mejor. El grupo Gentoo, los expertos en seguridad de Tridge, y el círculo de ex-alumnos de rsync de Pool eran módulos pre-existentes y que se traslapaban, que mezclaban y hacían calzar roles, según lo requería la emergencia.

Cuando esbozamos un mapa de los patrones diarios de colaboración a lo largo de todo el esfuerzo de desarrollo del kernel de Linux, encontramos que tales arreglos modulares son penetrantes y hasta cierto punto están anidados uno dentro de otro. Esto crea un tipo de organigrama dinámico de la organización; un organigrama que nadie diseñó, pero que le permite a la comunidad expandirse y adaptarse sin colapsar en el caos.

**Motivación intrínseca.** Las comunidades de Linux y TPS

no vinculan las transacciones clave con el dinero. A pesar de los débiles incentivos financieros, logran un nivel de motivación más alto que el que se encuentra en ambientes convencionales. Los psicólogos han descubierto que las zanahorias monetarias y los garrotes de responsabilización motivan a la gente a desempeñar tareas estrechas y específicas, pero por lo general la desaniman a ir más allá de ellas. La admiración y el aplauso son estímulos mucho más eficaces para un comportamiento que vaya mucho más allá. “La reputación personal del desarrollador está adjunta a cada versión”, le explicó Linus Torvalds a Robert Cringely, columnista en tecnología, en 1998. “Si se está fabricando algo para regalárselo al mundo, algo que representa tu filosofía de la computación ante millones de usuarios, siempre producirás el mejor producto”.

Los psicólogos también hacen énfasis en la importancia motivacional de la autonomía. Los programadores de Linux deciden por sí mismos cómo y dónde contribuir, y gozan de la satisfacción de producir algo cuya calidad no es definida por un departamento de marketing o por contadores, sino que por sus propios estándares exigentes. El coautor de este artículo, Robert Wolf, y Karim Lakhani del MIT, encuestaron a más de 800 usuarios-desarrolladores, y más de la mitad dijo que su trabajo en las fuentes abiertas es el esfuerzo más valioso y creativo de sus vidas profesionales.

Por cierto, el TPS no ofrece una autonomía tan extrema y los empleados no trabajan gratis. Pero en comparación con sus contrapartes en el sector automotriz, los trabajadores de TPS gozan de menos controles, de más estímulos a la iniciativa individual, de menos indicadores relacionados con el desempeño individual, y de aplausos más fuertes de sus pares. La recompensa que tuvo el equipo de Kyoritsu Sangyo cuando entregó el primer lote de válvulas P fue el orgullo profesional y corporativo, no los honorarios de Toyota. Ese mismo orgullo es experimentado por un trabajador subalterno de la línea de montaje, cuando tiene la confianza de sus pares para experimentar con mejoramientos a los procesos y detener la línea si algo sale mal.

**Altos niveles de confianza.** Cuando la información fluye libremente, la reputación, más que la reciprocidad, se transforma en la base para la confianza. Desempeñándose bajo un escrutinio constante –lo cual es desafiante pero no hostil–, los trabajadores saben que sus reputaciones están en riesgo, y eso sirve como una garantía de buen comportamiento, el equivalente a los contratos en un mercado o a las auditorías en una jerarquía. De ahí la obsesión que existe en la comunidad Linux por el reconocimiento a las contribuciones a los códigos y a la inclusión de direcciones personales de e-mail en los campos de comentario de las listas de distribución. De ahí el generoso crédito público otorgado a Barisani y a Warfield. De ahí la celebración colectiva a los heroicos esfuerzos de Kyoritsu Sangyo.

Con sus reputaciones en juego, es menos probable que la gente actúe de forma oportunista. Con la misma información disponible para todos, es menos probable que una parte explote la ignorancia de la otra. Y con un vocabulario y una forma de trabajar comunes, se producen menos malentendidos. Esos factores hacen aumentar la confianza, que es el capital social fundamental de estas comunidades.

La confianza importaría menos si es que no existiese un costo para abandonar estas redes, o si las transacciones fuesen de tamaños radicalmente diferentes (puesto que ello tentaría a las personas o empresas a romper las reglas cuando surgiese una gran oportunidad). Pero tanto en la comunidad Linux como en la de Toyota, el ingreso a los círculos interiores es un privilegio arduamente ganado, y ambas operan sobre la base de muchos intercambios pequeños.

Además, donde la confianza es la moneda de cambio, la reputación es una fuente de poder. En una red poco densa, tal como en la mayoría de los mercados y jerarquías, el poder deriva de la capacidad de controlar o de intermediar en el flujo de información y, por lo tanto, a menudo de la capacidad de restringirlo. En una red densa, sin embargo, la información sencillamente fluye alrededor de lo que sería el punto de obstrucción. Bajo esas circunstancias, hay más poder por el hecho de ser una fuente de información que por el hecho de ser un sumidero de ella. En consecuencia, los individuos están motivados para maximizar la visibilidad de su trabajo y sus conexiones a aquellos que, a su vez, están ampliamente conectados. Esto también alimenta la densidad de la información de la red.

## Transacciones baratas y muchas de ellas

Hasta aquí hemos discutido el contenido del trabajo. Pero los modelos TPS y Linux también cambian la economía del trabajo abaratando los costos transaccionales. Así se hace rentable para las organizaciones efectuar más transacciones, y de menor tamaño, tanto internas como externas, e incrementar de esta manera el ritmo y la flexibilidad que son típicos de las organizaciones de alto desempeño.

Las fuentes clásicas de los costos

transaccionales son la vulnerabilidad mutua al enfrentarse a la incertidumbre, los conflictos de intereses y el acceso desigual a la información. Gastamos dinero en efectivo en la negociación, supervisión y restitución para reducir esas imperfecciones. Tanto los mercados como las jerarquías incurren en esos costos transaccionales. Usando una metodología desarrollada por J.J. Wallis y Douglass North, estimamos que en el año 2000 los costos transaccionales por sí solos representaron ¡más de la mitad del PIB no gubernamental de Estados Unidos! Gastamos más dinero negociando y haciendo cumplir transacciones que cumpliéndolas.

En las comunidades Linux y Toyota, los acuerdos no se hacen cumplir mediante la sanción de un contrato legal o por la autoridad de un jefe, sino mediante la confianza mutua, lo que disminuye drásticamente los costos transaccionales. Esto no es nuevo: los equipos de personas en los lugares de trabajo convencionales operan en todas partes sobre la base de la confianza.

Lo que es nuevo es cuán ampliamente se puede expandir la confianza, incluso a personas que no se conocen, o aún entre aquellos que tienen intereses que compiten entre sí. Aisin confió en sus proveedores rivales al

entregarles copias de los planos de las válvulas P. Los hackers de rsync intercambiaron información sensible con personas que jamás habían conocido. Los proveedores de componentes de Toyota comparten a diario el conocimiento de procesos, confiando en que Toyota no lo usará para hacer bajar los precios. Los hackers de Linux confían entre sí para efectuar enmiendas no coordinadas y simultáneas a los códigos base.

Aún más, la tenencia común de la propiedad, lo que ocurre con ciertos tipos de propiedad intelectual en estas comunidades, hace disminuir las apuestas monetarias entre los dueños conjuntos. Los costos transaccionales simplemente disminuyen porque hay menos que negociar. En la comunidad Linux, los costos transaccionales se acercan a cero. Hewlett-Packard le pagó a Martin Pool para que fuese un ingeniero de Linux, pero eso no implica que se le debía pagar a HP por el margen de los trabajos nocturnos de Pool con rsync. En la comunidad Toyota, los costos transaccionales, sin ser cero, se han reducido radicalmente. Cuando la planta Aisin

## Reconocer como corresponde

La comunidad Linux usa un formato particular —un “archivo de créditos”— para reconocer las contribuciones de sus miembros. Por ejemplo, si reconociéramos en el formato Linux las contribuciones individuales para este artículo, ello se vería así:

n: Mark Blaxill  
e: blaxill.mark@bcg.com  
d: Exploración de la economía de las fuentes abiertas  
s: Boston Consulting Group

n: Paul Carlile  
e: carlile@bu.edu  
d: Discusión de los paralelos Linux/Toyota  
s: Boston University

n: Karim Lakhani  
e: lakhani@mit.edu  
d: Discusión de los paralelos Linux/Toyota  
d: Estudio a hackers libres / fuentes abiertas  
s: MIT



Seiki quedó destruida, Toyota y sus proveedores no se demandaron entre sí ni improvisaron contratos de emergencia. Sencillamente, se pusieron a trabajar, confiando en que al final recibirían una compensación justa. Jeffrey Dyer, un profesor de estrategia en la Brigham Young University, estima que los costos transaccionales entre Toyota y sus proveedores de nivel uno son sólo la octava parte de los de General Motors, una disparidad que él atribuye a distintos niveles de confianza.

## Un modelo para muchos

Reúna todos estos elementos y obtendrá un círculo virtuoso. Una red densa, que se auto-organiza, crea las con-


diciones para que se genere confianza a gran escala. La confianza a gran escala hace bajar los costos transaccionales. Los bajos costos transaccionales, a su vez, posibilitan muchas transacciones pequeñas, lo cual crea una red auto-organizada que se va profundizando en forma acumulativa.

Una vez que el sistema alcanza la masa crítica, se nutre de sí mismo. Mientras más grande sea el sistema, más ampliamente se compartirán los conocimientos, el lenguaje y el estilo de trabajo. Mientras más grande sea el capital reputacional de los individuos, más fuerte será el aplauso y también la motivación. El éxito de Linux constituye evidencia del poder de este círculo virtuoso. El

éxito de Toyota constituye evidencia de que también es poderoso en empresas convencionales, que buscan maximizar las utilidades.

La comunidad de Linux y el Sistema de Producción de Toyota son notablemente diferentes. El hecho de que obtengan tanto de maneras tan similares indica algunos principios que otros pueden seguir.

- La disciplina de la ciencia es sorprendentemente adaptable a la organización del trabajo corporativo, e incluso inter-corporativo.
- Bajo algunas circunstancias, la confianza es un sustituto viable de los contratos de mercado y de la autoridad jerárquica, no sólo en pequeños equipos sino también en comunidades grandes.
- A lo largo de las cadenas de suministro, las organizaciones que pueden sustituir contratos por confianza se benefician más de la colaboración que lo que pierden en poder de negociación.
- Los bajos costos transaccionales logran más innovación que la lograda por altos incentivos monetarios.

Estos principios atienden a la necesidad de crecimiento e innovación de las empresas de maneras que no lo hacen los modelos organizacionales tradicionales. Tal vez la eficacia de estas colaboraciones sugiera la emergencia, en el largo plazo, de algo completamente nuevo. No se trata de mercados. No se trata de jerarquías. Se trata de una poderosa combinación de ambos, y de la huella de una sociedad que funciona en red. 

## Beneficiarse del 80% desaprovechado

El Principio de Pareto dicta que las empresas obtienen el 80% de su valor a partir de 20% de sus productos, clientes o ideas. Debido a los altos costos transaccionales la parte larga de esa curva –ese 80% de generadores inciertos de valor– no puede ser explorado. Así es que en nombre del foco de la empresa, esa parte larga de la curva se corta, se segmenta o se desecha. Y con ello mueren innovaciones potencialmente rentables.

Las organizaciones que reducen los costos transaccionales pueden abarcar al 80% desaprovechado. Pueden responder a débiles señales de mercado, explotar pequeños segmentos y experimentar con combinaciones tecnológicas poco probables. Pueden efectuar cien pequeñas apuestas en lugar de unas pocas apuestas grandes.

Por ejemplo, Detroit consideró que los vehículos híbridos constituirían un producto intermedio poco interesante; los ejecutivos de las empresas automotrices estadounidenses prefirieron la investigación, hasta ahora no exitosa, de la tecnología de células de combustible. Mientras tanto, Toyota estaba fabricando el Prius. Este híbrido está ahora en la segunda generación, y Toyota espera vender 300.000 en el mundo este año. Los bajos costos transaccionales de Toyota y su predilección por la colaboración a pequeña escala le ayudaron a mantener abiertas 80 opciones discretas para el motor híbrido hasta sólo seis meses antes de la entrega del diseño final. Los fabricantes convencionales habrían tenido que congelar esas variables de diseño por lo menos dos años antes.

Es en los intersticios de la red humana, más que en las mentes de unos pocos *Wunderkinder* (niños maravilla), donde nacen las verdaderas innovaciones. Y es así, porque los costos transaccionales restringen la innovación y constriñen las oportunidades de compartir diferentes ideas, habilidades y prejuicios.

“Las personas de Detroit tienen mucho más talento que las personas de Toyota”, afirma con una modestia excesiva Fujio Cho, presidente de Toyota. “Pero nosotros tomamos personas con talento promedio y las hacemos trabajar como equipos espectaculares”. El innovador es, en otras palabras, la red.