

EL SOFTWARE CONVIVENCIAL. UNA PERSPECTIVA DEL USUARIO FINAL SOBRE SOFTWARE LIBRE Y EL CÓDIGO ABIERTO¹

CARL MITCHAM
University of Colorado
cmitcham@mines.edu

Recibido: 03/07/2007 Aceptado: 13/09/2007

Resumen: El software es un tipo de invención que nace en un contexto determinado. No toda innovación acaba teniendo éxito. En este sentido es posible analizar este contexto para valorar su viabilidad futura. Por otro lado, se puede entender que el software libre representa la emergencia de un nuevo paradigma ético y social en convivencia con otras formas de “libertad”. Se examina cómo el software libre es coherente con la tradición ética de los diversos colegios ingenieriles y deontologías. Finalmente se propone la producción de un “software convivencial” que atienda más a los usuarios.

Palabras clave: ética ingenieril, invención, innovación, tecnología convivencial.

Abstract: Free Software is a kind of invention born inside a concrete context. Not every innovation becomes successful. So it is possible to examine such context to evaluate a future viability for Free Software. Free software is studied as a new ethical and social paradigm towards freedom. There is an exam about how free software is compliant with the ethical tradition from engineering institutions and deontological proposals. Finally there is a proposal of a “convivial software” that would take into account final users.

Keywords: Engineering ethics, invention, innovation, convivial technology

Introducción

La idea del software libre y el código abierto nació dentro de una comunidad técnica aunque tal vez con ciertas influencias no técnicas (Turner, 2006). En cualquier caso las implicaciones que ha tenido para el mundo del usuario final han sido muy considerables. Lawrence Lessig (1999) y Steven Weber (2004), por ejemplo, han analizado las perspectivas que estos movimientos han tenido respectivamente para la ley y la economía política. Científicos informáticos, ingenieros de software y hackers han promocionado constantemente sus beneficios no técnicos. La comunidad del software libre ha visto el bazar del código abierto productivo tanto desde el punto de vista del usuario final a la vez que en bienes para el usuario de modos que el mercado del software propietario no lo es (ver, por ejemplo., las antologías de DiBona et al., 1999; Feller et al., 2005; y DiBona et al., 2006). En todos estos casos, el acercamiento general ha

¹Traducción de Andoni Alonso

sido el de proporcionar una descripción narrativa de alguno de los aspectos del fenómeno del software libre y el código abierto explorando cómo lo que primeramente pudo aparecer como comportamientos no estándar ha conducido a la construcción social de nuevas normas con implicaciones para el ordenamiento, legal, político y económico y que potencialmente son beneficiosas para la sociedad. (Tales narrativas acuden a cronologías que generalmente comienzan desde los años 50 del siglo pasado hasta el comienzo del siglo XXI, tal como se esboza de forma simplificada en la cronología que aparece en el apéndice).

Desde luego, en la base de tales narrativas se encuentra que la apertura ha emigrado desde el software a una plétora de términos relacionados tales como publicaciones de “acceso abierto”, desarrollo de “contenido libre” (como en la Wikipedia), “software de enseñanza abierta”, e incluso “cultura de código abierto”. Curiosamente, sin embargo, ha habido pocos enlaces entre los ideales del código abierto y la filosofía política popperiana de la sociedad abierta (Popper, 1945) y solo algunas discusiones limitadas entre las relaciones de la sociología clásica mertoniana de cómo funciona la ciencia cuando ésta es guiada por un conjunto apropiado de normas científicas (Merton, 1942), la cual se denomina en la actualidad como ciencia “libre” o abierta (vid Keltly, en Feller et al., 2005). Las relaciones entre los métodos del código abierto y la ontología de los sistemas abiertos (vid., e.g., Bertalanffy, 1968) son otras conexiones que quizás merecen un desarrollo ulterior.

Esta argumentación, sin embargo, tomará un rumbo diferente. Su propósito es comenzar con el usuario final y desde su perspectiva obtener un reflejo del movimiento del software libre y el código abierto no tanto desde dentro a afuera sino más bien desde fuera a adentro. Hacerlo así exige colocar este movimiento en una perspectiva histórico-filosófica más amplia. Así, yendo más allá de los detalles técnicos, el propósito consistirá en potenciar y apoyar el movimiento del software libre y el código abierto mostrando cómo es el heredero de una tradición profesional de idealismo ético y potencialmente en relación con cuestiones importantes de la historia de las relaciones entre ciencia, tecnología y sociedad.

Este acercamiento tiene obvias afinidades con el diseño centrado en el uso y el usuario final (vid., e.g., Lieberman et al., 2006; y Seffah et al., 2005) aunque se ha mantenido que tales métodos de diseño tienden a tratar al usuario como otro factor más en la construcción tecnológica en marcha, esto es como clientes o consumidores en vez de como el verdadero objetivo de la funcionalidad del software. En cambio, la argumentación que propongo tiene más en común con la perspectiva centrada en el usuario de Robert Johnson (1998) o Peter-Paul Verbeek (2005) con su análisis de los artefactos tecnológicos

en general; tanto Johnson como Verbeek, sin embargo, adoptan respectivamente posiciones retóricas y postfenomenológicas de las cuales me sustraigo.

El esfuerzo por concentrarse en el software desde el punto de vista del usuario final lleva a acuñar un término nuevo. Desde esta perspectiva lo que importa no es tanto la disponibilidad directa del código fuente y la libertad creativa para manipular lo que podría denominarse como software convivencial, expresión más profunda y rica en significados que la de “uso amigable”. Términos emparentados podrían ser “software sin fronteras” o incluso “software humanitario” a fin de enfatizar el deseo de romper las ataduras de la soberanía del software o del nacionalismo tecnicalizado. Hay más cosas en cuanto a la apertura que la transparencia del software o el derecho a jugar con el código. Desde una perspectiva no tecnológica, la invención y el desarrollo del software no es más que otro ejemplo de la invención y de la construcción del desarrollo tecnológico. El software es simplemente un nuevo ejemplo de tecnología y el esfuerzo para asegurar que tal tecnología se desarrolla y utiliza de manera socialmente responsable tiene una amplia historia. Por lo tanto mi argumentación comienza con observaciones sobre la historia de la tecnología. Esto conducirá a una serie de reflexiones críticas en el desarrollo de la ética profesional de la ingeniería y a una discusión del movimiento alternativo de la tecnología. Finalmente concluirá proponiendo algunas políticas y criterios para tener en cuenta cuando se imagina el diseño de un software convivencial.

La invención tecnológica en un contexto social

La creación del software es un tipo de invención. Pero es de tal tipo que tiene sus propias características problemáticas. Escribir y adaptar código fuente es una actividad alejada en más de una ocasión del usuario final. Es una actividad que requiere altos niveles de pensamiento analítico y abstracto. Resulta útil considerar tales rasgos desde una perspectiva histórica. Los seres humanos no pueden vivir sin técnica. Fabrican por necesidad refugios y vestimenta y deben asegurar la comida, y prepararla. Por ello es razonable re-describir al *homo sapiens* como *homo faber*. Pero del mismo modo que el pensamiento del *homo sapiens* no se limita a la racionalidad -ya que pensar incluye el uso creativo del lenguaje tanto en las artes como en las humanidades- asimismo el hacer de *homo faber* toma muchas formas. Aunque puede que la necesidad sea la madre de la invención, ésta es una madre de una gran familia con muchos retoños.

De acuerdo con Alfred North Whitehead y José Ortega y Gasset, quienes escribieron de forma independiente en las primeras décadas del siglo XX, la historia de la tecnología atravesó un umbral crítico a finales del siglo XIX

cuando su proceso se vio racionalizado. Lo que Whitehead llama “la invención del método de invención”, esto es, la invención no es sólo otro artefacto técnico sino las técnicas para promocionar la invención en sí misma, fue “la gran invención del siglo XIX”, un acontecimiento que transformó la técnica de forma esencial y también la relación entre técnica y sociedad (Whitehead, 1925, p. 141). Sustituyó las técnicas artesanales por una tecnología perseguida racionalmente.

Thomas Edison, quizás el mayor inventor de todos los tiempos, recompensado con el récord de 1093 patentes, jugó un papel fundamental al inventar lo que llamó como “la fábrica de investigación”. Respecto a los motivos que le condujeron a tal factoría, o lo que hoy llamaríamos Investigación y Desarrollo (I+D), cuenta la siguiente reveladora anécdota. De acuerdo con él, cuando sólo tenía 21 años y trabajando en la industria informática de su época, las comunicaciones telegráficas, su primera invención patentada fue un dispositivo para contar eléctricamente los votos de una asamblea legislativa. En lugar del voto por nombramiento o los sistemas de voto escrito, que eran muy lentos y empleaban mucho tiempo, todos los miembros del parlamento podrían tener unos interruptores eléctricos en sus escaños: apretando uno se registraría un voto afirmativo, apretando el otro un voto negativo. Así se podrían contar rápidamente los votos y se podrían determinar los resultados. Pero este invento fue un fracaso; no se vendió. El Congreso de Estados Unidos al que Edison se lo ofreció no quiso comprarlo. Más aún, se mostró positivamente en contra de él. La razón es que con frecuencia los políticos querían saber cómo votaban sus colegas antes de decidir su propio voto o querían un momento para reconsiderarlo mientras que se hacía manualmente el recuento de los votos escritos. En una votación por nominación, los miembros podían “pasar” mientras que alguien en particular votaba, para saber qué votarían los otros antes de decidirse finalmente. Así era posible negociar incluso en medio de la votación. Esta posibilidad se veía impedida por el sistema de voto eléctrico.

En respuesta a este fallo para vender esta máquina eléctrica de registro de voto, Edison prometió nunca volver a inventar algo al menos que supiera que la gente para quien estaba destinado realmente la deseaba. Desde entonces prometió que “dedicaría su talento inventivo sólo para las cosas que realmente tenía una demanda real, genuina” (Dyer y Martín, 1929, vol. 1, p. 103). Edison no estaba enamorada de la práctica de la tecnología que Samuel Florman (1970), en una apología de la ingeniería creativa, denomina como “los placeres existenciales” de la experiencia tecnológica. Lo que esta historia revela no es sólo un aspecto del plan de negocios de Edison sino algo sobre el propio proceso sistemático de invención. El proceso que hace posible lo que podría denominarse

como “invención alienada”. Karl Marx había identificado antes la alienación como una característica clave del proceso de producción en masa. Ahora, en la fábrica de Investigación y Desarrollo, la alienación ha pasado de la producción tecnológica a la producción de tecnología. Esto dará lugar a lo que se ha venido a llamar las dificultades de la transferencia tecnológica, esto es, la transferencia del laboratorio al mercado.

Es difícil pensar en los artesanos tradicionales inventando productos no queridos o superfluos. Tradicionalmente, los artesanos se encuentran tan encajados social y culturalmente que sus habilidades inventivas se aplican naturalmente a bienes que tienen un completo significado cultural y social. Sus contribuciones que pudiéramos confundir como anticipaciones de una economía de consumo podrían ser decoraciones y ornamentos que subrayan la integración cultural al unir tecnología, religión, política y arte. Las novedades no venían a la existencia repentinamente, no bajo un propósito consciente o sistemático, sino de forma lenta, como acumulaciones de adquisiciones sobre un período extenso de tiempo. En los laboratorios de investigación y desarrollo, sin embargo, los grandes procesos de separar o disgregar los elementos de la cultura, característico de la modernidad (1944), se apodera del proceso de invención y lo descontextualiza de la sociedad y la cultura. El Renacimiento, la Reforma y la Ilustración introdujeron distancia entre, por ejemplo, el arte, la religión, la política y la economía, convirtiéndose todas ellas en instituciones semiautónomas cuya integración, consecuentemente, había de ser reconstruida y no simplemente asumida. Así que también el proceso de invención fue entonces desenraizado de todas las conexiones implícitas con un nexo sociocultural y así se descontextualizó. Puesto que Edison no tenía un conocimiento tácito o intuitivo de la vida política, inventó algo que no encajaba con esa vida y juró que en el futuro realizaría algo que llamaríamos una investigación previa del mercado, antes de inventar nada.

Desde luego durante un tiempo la reciente invención del método de inventar podía apoyarse en la memoria residual de una necesidad humana contextualizada. Las mayores invenciones de Edison no requirieron una investigación consciente del mercado a fin de tener éxito. La luz eléctrica, la grabación de la voz y la imagen en movimiento respondían todas a deseos humanos casi míticos aunque muchos de esos deseos primordiales y posibles dispositivos para satisfacerlos hubieron de ser mitológicamente criticados. Sus grandes innovaciones tecno-económicas en el sistema de energía eléctrica fue una respuesta directa hacia el entusiasmo popular por las luces eléctricas. Pero consecuentemente, especialmente en casos particulares, se debían investigar cada

vez más los contextos sociales y culturales de uso previamente por medio de la investigación del mercado o se debían crear por medio de la publicidad.

Ortega (1939), poco después de Whitehead y sin referirse a Edison, indicó certeramente que una vez que aparece un método descontextualizado de invención, los inventores corren el peligro de estar cada vez más alienados de la vida. Se vuelven inventores solamente, interesados en la invención por la invención misma, nada más que inventores, atrapados en los placeres de la invención, pero por ello desprovistos de los placeres comunes de la existencia. Convierten el arte por el arte en la invención por la invención. La división entre experto y ciudadano, que es una brecha digital con dimensiones mayores que la simple accesibilidad al producto, se abre a los pies del especialista tecnocientífico y a la espalda del usuario final en democracia. Es una brecha que solamente se puede eliminar del mismo modo que se creó, con la conciencia: por medio del desarrollo sistemático de formas que relacionen ambos mundos.

Las comparaciones con el desarrollo del software son obvias, sin duda. La cultura hacker tiene la tendencia de volverse sobre sí misma. Se define a los hackers por sus habilidades técnicas y su placer en resolver problemas, tanto como su compromiso para compartir esas soluciones. Reinventar la rueda no es solamente un gasto de energía, tampoco es divertido. Lo divertido es inventar algo nuevo y compartir esa alegría. Los ingenieros de software escriben código que es técnicamente agradable y placentero compartir.

Éstas son las palabras de Linus Torvalds, otro inventor de 21 años quien había subido la versión 0.01 del kernel del Linux y anunció sin alharacas la versión 0.02 (el 5 de octubre de 1991):

¿Te acuerdas con pena de los buenos días del Minix-1.1., cuando los hombres eran hombres y escribían sus propios drivers para sus periféricos? No tienes un buen proyecto y te mueres por hincarle el diente a un (sistema operativo) que puedas modificar...? ¿Ya no quedan noctámbulos que hagan funcionar un programa de primera? Entonces este mensaje puede ser para ti.

En el medio de ese placer tecnológico, seguramente se encuentra la medida de la distancia con los intereses de los usuarios finales. En un correo electrónico anterior (25 de agosto de 1991) Torvalds escribía: “toda sugerencia es bienvenida pero no puedo prometer que la incluya”. En un libro posterior, éste consideró El significado de la vida” distinguiendo tres motivaciones humanas básicas: supervivencia, relaciones sociales y entretenimiento. “Estando asegurada la supervivencia”, Torvalds mantenía que Linux “atrajo a la gente tanto por el entretenimiento de un reto intelectual y las motivaciones sociales

incluidas en ser parte de crearlo todo” (Torvalds, 2001). Tal como el avaricioso Microsoft nos recuerda con frecuencia, la comercialización propietaria del software no se originó sólo por avaricia. Como bien sabemos la comercialización tiene sus propios impedimentos. El mercado mismo ha de ser creado y diseñado para beneficiar el bien común, porque está en constante peligro de ser capturado por los intereses privados. El software libre o el movimiento de código abierto es un intento de cambiar el equilibrio de poder en la economía de mercado, pero es un cambio que puede exigir ajustes complementarios ente aquellos que participan en este tipo de creatividad ingenieril y sus continuaciones empresariales.

El ideal ingenieril

En el mismo período que Thomas Edison estaba inventando el método de invención, los ingenieros estaban inventando su profesión, y su ética ingenieril. Edison mismo no fue un ingeniero licenciado profesionalmente, fue, igual que muchos hackers, un autodidacta. Pero del mismo modo que creó un método de invención, otros ingenieros se encontraban en el proceso de crear un método para producir ingenieros, un método que sistematizara el proceso de auto-educación.

La invención de una ingeniería profesional y su correlativa ética profesional ingenieril tiene su propia y compleja historia. Para este artículo baste señalar cómo la invención de la educación ingenieril fue una respuesta para la necesidad masiva de ingenieros, causada por la producción en masa y la invención en masa. Más aún, la emergencia de la ética de la ingeniería profesional puede entenderse como una respuesta novedosa ante el problema de la transferencia tecnológica de Edison, ante la tendencia de la invención tecnológica y el diseño de separarse del bien público para el usuario final. En vez de las respuestas comerciales ante la investigación de mercados y la publicidad, o del propio esfuerzo de Edison, la ingeniería profesional comenzó a plantear compromisos institucionales que pudieran evitar esos peligros.

Permítase abocetar esta emergencia de la profesión ingenieril y su correlativa ética como un argumento en tres fases. (la narración que se presenta más adelante se basa en una exploración previa en Mitcham, 1994.) La primera fase ocurre entre el siglo XVIII y el XIX, cuando la ingeniería civil surge del entorno militar. Los primeros ingenieros fueron militares, diseñadores de fortificaciones y diseñadores u operadores de “ingenios de guerras”, tales como catapultas o arietes. Cuando los ingenieros surgieron de las sombras de lo militar, se volvieron ingenieros civiles. En Inglaterra, a partir de los clubes informales gastronómicos, apareció la primera Sociedad de Ingenieros Civiles en 1771,

luego la Institución de Ingenieros Civiles en 1818 a la que se le otorgó un estatuto real en 1828. La Institución de Ingenieros Mecánicos apareció dos décadas después.

Cuando se concedió el Estatuto Real, la Institución de los Ingenieros Civiles se pidió a uno de sus miembros, Thomas Tredgold, que formulara una definición de la ingeniería. El resultado fue la ya clásica descripción de la ingeniería como “el arte de dirigir las grandes fuentes de energía de la naturaleza para el uso y conveniencia del ser humano”. Pero la verdad es que los ingenieros no trabajan para los seres humanos en general, la mayoría lo hace para corporaciones privadas y por tanto se vuelven fácilmente deudores de los intereses concretos de quien paga sus salarios. En esto se diferencian de otros profesionales como médicos o abogados, quienes por lo general llevan sus propias compañías.

Los resultados éticos son obvios. Los ingenieros carecen de un nivel de autonomía profesional de la que disfrutaban médicos y abogados. Desde luego, habiendo nacido de lo militar, hay una tendencia en los ingenieros de adoptar algunas formas de ética militar. La ética de los militares es obedecer a la autoridad. La ética implícita de los civiles tiende hacia la lealtad a la compañía. Esto es, a primera vista, los ingenieros asumen tranquilamente, a pesar de sus ideales de servir a la humanidad, que han jurado obediencia inmediata a sus empleadores.

Una segunda fase del argumento respecto a la ética ingenieril aparece a principios del siglo XX. Las sociedades ingenieriles profesionales prepararon el borrador de los primeros códigos éticos de la ingeniería profesional. Estos códigos simplemente hacían explícito lo que ya estaba implícito: la importancia de la compañía y la lealtad profesional. Pero enseguida se vio la inadecuación de esta postura. Los médicos tenían ya la obligación moral respecto a la salud. Los abogados perseguían el ideal moral de la justicia. Ni médicos ni abogados veían posible que sus pacientes o clientes pudieran ordenarles qué hacer: en vez de ello, como consejeros expertos, daban instrucciones a quienes ellos servían en el nombre de un ideal mayor al que sí servían. ¿No existía un ideal sustantivo para los ingenieros también al cual deberían servir?

Durante los años 20 y 30 se debatieron propuestas respecto a la posibilidad de que ese ideal sustantivo para los ingenieros fuera la eficiencia. El problema respecto a esta propuesta es doble: primeramente es incapaz de reconocer el grado en el que la eficiencia depende de un contexto, esto es, varía según qué inputs o outputs se consideren para calcular el promedio de ambos. En segundo lugar tiende a justificar un tipo de gobierno tecnocrático. Esto es, mina la democracia.

Finalmente, una tercera fase del argumento, los ideales inadecuados de lealtad y eficiencia se sustituyeron por una nueva formulación de la responsabilidad ingenieril. Inmediatamente tras la II Guerra Mundial, en los EE.UU., los códigos éticos de los ingenieros profesionales comenzaron a discutir “la seguridad pública, la salud y el bienestar” como las primeras obligaciones profesionales de los ingenieros. Durante el curso de cinco décadas, desde finales de 1940 hasta finales de los 90, esta argumento se convirtió en la perspectiva consensuada. Una redefinición de la trayectoria y su proyección es que los ingenieros tienen un deber *plus respicere* de tomar en cuenta algo más que la ingeniería (Mitcham, 1994, p. 164).

Este consenso se refleja en el “Software Engineering Code of Ethics and Professional Practice” (1999), que declara que los ingenieros de software deben adherirse a ocho principios “de acuerdo con un compromiso (general) hacia la salud, la seguridad y el bienestar del público” (los ingenieros de software hacen sólo un pequeño cambio al modificar el orden de la seguridad por encima del de la salud). Desde luego, el primero de estos principios es que “los ingenieros de software deben actuar consistentemente con su interés público”:

En el comienzo del siglo XXI, el conocimiento de esos códigos y la reflexión en las cuestiones asociadas con su aplicación y práctica se han vuelto un requerimiento en los currícula de los estudios de ingeniería. De acuerdo con el Accreditation Board for Engineering and Technology o ABET, la agencia profesional e independiente que acredita todos los programas educativos de ingeniería, se requieren once resultados para cualquier rama de la educación ingenieril. Y uno de esos resultados, junto al conocimiento de las matemáticas y la ciencia o las habilidades en proyectar, es “la comprensión de una responsabilidad ética y profesional”.

El movimiento del software libre y abierto constituye una contribución fundamental para la ética ingenieril. Desde luego, dado que una de las formas mejores de enseñar ética ingenieril, es por medio de los casos de estudio, el movimiento del software libre y gratuito es un buen candidato para enseñar cómo los ingenieros tienen en cuenta la ética. Se puede argumentar muy fácilmente que el código abierto es un medio necesario para proteger la seguridad pública, la salud y el bienestar en general y para practicar al menos dos de los ocho principios especificativos.

El principio 3, por ejemplo, mantiene que “los ingenieros de software deben asegurar que sus productos y sus modificaciones correspondientes cumplan los más altos estándares posibles. Tal como Eric Raymond ha sostenido largamente en *The Cathedral and the Bazaar*, esto requiere, en muchas ocasiones, una disponibilidad de código abierto. El principio 5 declara aún más

que “los ingenieros y administradores de software deben suscribir y promover un enfoque ético ante la administración, desarrollo y mantenimiento del software”. Es difícil ver cómo sea posible esto sin un código abierto y la libertad de alterar y adaptar tal código. Por tanto, se puede argumentar firmemente que el “Software Engineering Code of Ethics and Professional Practice” incluye un apoyo hacia el movimiento del software libre y abierto. (Desde luego se puede mantener que incluso el más genérico código ético de la “ACM Code of Ethics and Professional Conduct” apunta hacia la misma dirección.

Pero hacer de la seguridad, salud y bienestar públicos un valor clave también indica la importancia del público y eleva la cuestión, de nuevo, del papel final de usuario. Dentro de los diversos campos de la ética aplicada, respecto a la cual la ética informática y del software son sólo dos ejemplos, han existido sutiles al tiempo que evidentes movimientos para implicar al público en las decisiones técnicas (vid., e.g., las discusiones sobre el diseño colaborativo examinadas por Scrivener et al., 2000.). Existe, por supuesto, una fuerte resistencia a la idea desde varios lugares. Después de todo, dice el argumento ¿cómo pueden los no expertos decir a quienes lo son cómo realizar su trabajo? ¿No minará la interferencia pública la autonomía profesional técnica? ¿No aparece un peligro de politizar la ciencia y la ingeniería?

A pesar de los peligros, existen al menos tres argumentos correlacionados para promover una adecuada implicación del público en la toma de decisiones técnicas. (Mitcham, 1999) La primera es que la implicación pública propicia la democracia y la inteligencia civil. En el grado que los ciudadanos dejen a los expertos decidir por ellos, dejarán de ser miembros activos del cuerpo político: pero en el grado en que trabajen con los expertos a ayudarles a tomar las decisiones sobre los objetivos adecuados para la investigación científica y el desarrollo tecnológico, aprenderán sobre la ciencia y la ingeniería, sobre sus poderes y limitaciones, y reforzaran sus vidas como ciudadanos democráticos.

El segundo argumento es que la participación pública de hecho puede también promocionar una toma de elección tecnológica mejor. Todo conocimiento no es conocimiento técnico. El conocimiento local, el conocimiento del no experto, lo que con frecuencia se denomina como conocimiento indígena, cuando se utiliza apropiadamente, puede hacer que las decisiones técnicas sean mejores. Un buen ejemplo se encuentra en la investigación del HIV/SIDA (vid. Feenberg, 1995, cap. 5). Fueron los activistas de esta enfermedad los que ayudaron a incrementar los fondos de la investigación y los desviaron del trabajo básico en inmunología hacia terapias mucho más prácticas e inmediatas para los que estaban afectados por la

enfermedad. Sin la implicación de estos activistas del HIV/SIDA no se hubieran salvado tantas vidas por la investigación científica al respecto.

Finalmente, el tercer argumento es que simplemente la gente afectada por las decisiones debe tener derecho a influir en ellas. Muchas decisiones científicas e ingenieriles tienen un impacto fundamental en nuestras formas de vida. Los revolucionarios democráticos una vez proclamaron “no pago de impuestos sin representación” (no taxation without representation). Hoy se podría proclamar con la misma legitimidad: “ninguna invención sin representación”. Hoy en día, tal como Steven Goldman (1992) ha mantenido, debería ser igualmente legítimo decir “ninguna invención sin representación”.

Una de las instancias más claras de este salto de que “los profesionales técnicos conocen mejor” a “los profesionales técnicos aconsejan” ha ocurrido más claramente en la profesión médica. La ética médica tradicional enfatizaba que el médico era quien tomaba las decisiones para el paciente. Hoy este modelo está en transición hacia uno en el que el médico trata de educar a los pacientes a fin de que estos puedan realizar decisiones libres e informadas sobre el curso de los tratamientos que sufrirán o no. El médico es el experto, sí, Pero el médico no es la autoridad final. La pericia médica está al servicio de los pacientes quienes determinan ellos mismos cómo esa pericia ha de afectar, en última instancia, a su vida.

El ideal de una tecnología convivencial

La anécdota de Edison trata el problema de la transferencia de tecnología desde el laboratorio al mercado. Aquí, la separación entre la invención y la necesidad del usuario final resulta ser intensificada y racionalizada con nociones de autonomía técnica y los placeres existenciales de la invención de vanguardia, y las respuestas del marketing y la publicidad no revelan el problema sino que más bien lo oscurecen. El desarrollo de una ética ingenieril profesional y personal es en cambio una forma de enfrentarse con este problema oscuro. Pero el problema de la alienación respecto a las preocupaciones del usuario final se vuelve más claro con otro tipo de transferencia tecnológico, esto es, la que se produce entre un país o sociedad y otros.

El 20 de enero de 1949, en su discurso inaugural, Harry S. Truman, el primer presidente de los EE. UU. tras la guerra mundial, declaró que habiendo usado la ciencia y la tecnología para ganar el mayor conflicto militar de la historia, ahora EE.UU. tenía una nueva responsabilidad: “Debemos embarcarnos (dijo Truman) en un gran programa nuevo para hacer disponibles los beneficios

de nuestros avances científicos y progreso industrial para la mejora y el crecimiento de las áreas subdesarrolladas (citado en Sachs, 1992, p. 6)

Con estas palabras y consecuentes acciones, Truman no sólo inauguró su presidencia sino la “era del desarrollo”. Para el resto del siglo XX, los Estados Unidos y sus aliados trataron sistemáticamente de transferir tecnología desde los países desarrollados a los subdesarrollados, a fin de llevar a todas las sociedades a un marco común. Las discusiones del siglo XX sobre la economía de la globalización no son más que permutaciones recientes de la ideología del desarrollo.

Sin embargo para mitades de los setenta se había vuelto obvio que estas estrategias del desarrollo inicial a fin de transferir tecnología avanzada de un país a otro fallaban a menudo por razones similares a las que experimentó Edison con su dispositivo de recuento de votos. Lo que los países donantes querían dar a menudo causaba resistencia o era asumido de forma ineficiente por los países receptores. Los sistemas de electricidad y agua se colapsaban, las nuevas carreteras y maquinaria no tenían mantenimiento. La distancia entre los pobres y ricos no se redujo, más bien se amplió.

Aparecieron cuatro soluciones para ello: promocionar la investigación en ciencias sociales respecto al problema y aumentar la burocracia del desarrollo. Los productos resultantes de ello funcionaron a menudo como estudios de mercado internacionales.

La segunda fue promocionar la educación, como corolario adjunto para la transferencia internacional de tecnología. La educación en cuestión, sin embargo, funcionó a menudo más como publicidad que como otra cosa. En muchos casos simplemente facilitó la inmigración de una recientemente entrenada elite desde los países pobres a los ricos, un tipo de explotación del capital social, de forma paralela a la explotación de los recursos naturales.

Una tercera respuesta fue simplemente recortar la ayuda al desarrollo, con la creencia de que los países receptores simplemente tenían que hacerlo por su cuenta. Tal como un libro proclamaba, “el subdesarrollo es un estado mental” (Harrison, 1985) queriendo decir, desde luego que el problema era el estado mental de los pobres y no de los ricos.

La cuarta respuesta fue ofrecer propuestas de lo que se llamó, de varias formas como tecnologías intermedias, apropiadas o alternativas. Las tecnologías intermedias se diseñaron para funcionar a medio camino entre herramientas manuales y máquinas de alta tecnología. Un ejemplo: sustituir el transporte con animales no con motores de combustión interna sino por bicicletas. Tales tecnologías intermedias fueron también apropiadas para los contextos sociales y culturales porque introdujeron cambios menores, menos desestabilizadores y su

funcionamiento era más fácil de entender. El funcionamiento de una bicicleta es mucho más transparente que el de un motor de combustión interna.

Reflexionando de nuevo sobre el propio mundo de la alta tecnología, especialmente tras la crisis energética de los setenta y enfrentándose a las cada vez más graves problemas medioambientales de la polución, esas tecnologías se vieron útiles no sólo para los países subdesarrollados o en vías de desarrollo sino también para países altamente desarrollados. Ahora se denominan tecnologías alternativas. Los senderos suaves de las energías renovables, viento y sol en vez de carbón, petróleo y energía nuclear, se propusieron como mejores tanto para los pobres como para los ricos y también como una nueva base para crear marcos comunes de desarrollo.

La declaración más influyente del movimiento por la tecnología intermedia y apropiada fue el libro de E.F. Schumacher *Lo pequeño es bello* (1975) Aunque el libro de Schumacher es el más ampliamente conocido, consideramos que hay un análisis más profundo en el libro de Ivan Illich *La convivencialidad* (1973). Desde luego durante un período corto de tiempo, el término “herramientas convivenciales” o “tecnología convivencial” fue un suplemento para las tecnologías apropiadas, intermedias o alternativas. Aunque el término ha sido apartado por la historia, ahora lo traemos a colación para relacionarlo con el software, a fin de oponerse ante las formas más genéricas de pensar en la tecnología.

En *La convivencialidad* Illich comienza notando lo siguiente respecto a muchas tecnologías: comienzan siendo un medio para un fin específico pero con frecuencia acaban subvirtiendo ese fin a causa de un fenómeno de “contraproductividad”. El estudio de caso de Illich fue la medicina. Lo que comenzó siendo un medio para la salud se ha convertido finalmente en la causa de un nuevo tipo de enfermedad o afección iatrogénica o causada por el médico. De acuerdo con Illich, el primer umbral en la historia de la medicina científica ocurrió cuando los médicos de hecho hicieron algo beneficioso por sus pacientes en un 50% de los casos. Este umbral se atravesó en la primera o segunda década del siglo XX.

Pero apareció un segundo umbral se atravesó cuando se pasó a que más del 50% de las ocasiones los médicos trataron a los pacientes por enfermedades o afecciones a las que la propia práctica médica había contribuido. Tales enfermedades o afecciones pueden oscilar entre los resultados de prácticas equivocadas o efectos negativos secundarios de la medicación, las infecciones causadas por los tratamientos, enfermedades adquiridas en hospitales, enfermedades que ni siquiera existían antes de la evolución de las bacterias en

respuesta a los antibióticos y otras. De acuerdo con las estadísticas de Illich, este segundo umbral estaba a punto de alcanzarse o se había alcanzado ya.

Para Illich, la respuesta adecuada a esta contraproduktividad de este segundo umbral es la moderación: no más sino menos medicina. La tecnología convivencial debe ser una tecnología limitada adecuadamente, una tecnología que no nos abruma con su presencia de tal modo que pensamos que no podemos vivir sin ella. Ésta es una argumentación que se ha reiterado en el curso de más de un examen crítico del sistema contemporáneo de medicina (vid. e.g. , Callahan, 1998).

La respuesta a la transferencia de tecnología de un país a otro, de acuerdo con el argumento de Illich, llamaría al ejercicio de una distancia crítica. Quizás no debería transferirse toda tecnología. Antes de intentar hacerlo, en vez de realizar un estudio de mercado sobre qué quiere la gente, quizás debería hacerse una evaluación de la propia tecnología. No todas son inherentemente beneficiosas. Muchas conllevan implicaciones políticas si no consecuencias negativas imprevistas que deben considerarse cuidadosamente (vid. Winner, 1980). Es más importante que la tecnología misma lo que podíamos denominar, en una modestamente irónica adaptación de Illich, su cociente de convivencialidad. A este qué punto una tecnología hace posible o perfecciona a un ser humano (*vivere*) con su (*con*) presencia artefactual.

Uno puede entender en general cómo funciona un reloj mecánico, análogo. Sin los detalles uno apenas puede tener una pista de cómo funciona un reloj digital o electrónico. Ambos ofrecen la mercancía de lo que podríamos llamar un tiempo “bien templado”. Pero en el primer caso el mecanismo de ofrecimiento es imaginable incluso aunque no se entienda del todo. En el segundo, el mecanismo desaparece tras un velo de sofisticación tecnológica. La brecha digital se vuelve una característica ontológica de los artefactos de alta tecnología.

Ciertamente algo parecido ocurre en el cambio entre la máquina de escribir al procesador de textos del ordenador.

Conclusión e implicaciones

Cuando se va en esa dirección se nota el salto de la herramienta a la máquina y de ahí a los dispositivos computerizados que tienden a disminuir la integración entre humanos y artefactos. Las herramientas requieren el input de la energía humana y su guía. El martillo se mantiene en la mano, es movido por los músculos del brazo y guiado por el ojo vigilante que, a su vez informa del movimiento de la mano.

La máquina, por contra, recibe su energía y movimiento de una fuente no humana, aunque el ser humano permanece como la fuente inmediata de guía. El automóvil es movido por un motor de combustión interna pero requiere un conductor humano. La forma de convivencialidad o de “con-vivir” manifiesta en el coche es bastante distinta a la experimentada con el martillo. El coche comienza a tomar un tipo de independencia o semi-autonomía, si el conductor quita las manos del volante, el automóvil puede continuar el trayecto por su cuenta. El coche es menos una extensión del cuerpo humano y más un artefacto energizado que se enfrenta al cuerpo, si no es una cápsula que encierra temporalmente al cuerpo y lo transporta.

En las máquinas computerizadas esta trayectoria hacia vivir más *alejadas* de que *con*, alcanza un nuevo nivel. Los dispositivos computerizados tales como los cajeros automáticos no están movidos por la energía humana sino por electricidad y son usados no directamente por seres humanos sino por programas. La guía humana existe tras el escenario, desde luego, pero tan profundamente escondida o alejada que el artefacto parece tener vida independiente. Vivir con cajeros automáticos se vuelve una cada vez menor implicación corporal, al contrario con las herramientas y más aún vivir entre de las plantas y los animales. La estructura de la convivencialidad técnica sufre profundas alteraciones.

Una investigación posterior por una tecnología convivencial se puede encontrar en la obra de Albert Borgmann, otro filósofo profundamente implicado con las interacciones entre tecnología y cultura. De acuerdo con Borgmann, la separación entre el inventor y el usuario final está reificada en el artefacto material mismo, en la separación entre la maquinaria de un dispositivo y la mercancía que ofrece. Esta mercantilización de la cultura no es sólo la creación de una economía de compra inmediata sino una donde los bienes que se compran son mercancías cuya función es fundamentalmente opaca.

Uno puede entender sin necesidad de detalles cómo funciona un reloj mecánico, análogo. Sin los detalles uno apenas puede tener una pista de cómo funciona un reloj digital o electrónico. Ambos ofrecen la mercancía de lo que podríamos llamar un tiempo “bien templado”. Pero en el primer caso el mecanismo de ofrecimiento es imaginable incluso aunque no se entienda del todo. En el segundo, el mecanismo desaparece tras un velo de sofisticación tecnológica. La brecha digital se vuelve una característica ontológica de los artefactos de alta tecnología. Convivir con la máquina de escribir implica extender con sencillez la inteligencia en el mundo del artefacto. Uno no se siente tonto o estúpido o se enfrenta a una construcción opaca que empequeñece las habilidades cognitivas comunes. En vez de esto, mirar y usar una máquina de

escribir ofrece cierta satisfacción. Uno desarrolla rápidamente un sentido de competencia y seguridad en su presencia.

En la presencia de los procesadores de textos informáticos, en cambio, lo opuesto es casi la norma: uno se siente estúpido, incluso irritado, incapaz de entender que está pasando en realidad y en ocasiones incapaz de hacer nada. ¿Podría el software libre y el código abierto enfrentarse a este problema? ¿Podría el software libre y el código abierto diseñar un procesador de textos que como la máquina de escribir promoviera en vez de inhibir la convivencialidad? ¿Y no sería ese software inherentemente más disponible para la transferencia entre países del mismo modo que una bicicleta lo es más que un automóvil?

La historia de la tecnología y la invención de Edison del método de invención han servido para apuntar el problema de la transferencia tecnológica del laboratorio al mercado. El problema nace de una tentación a la que la comunidad del software no es inmune: la tentación de convertirse en una élite apartada del usuario final no técnico al cual deberían servir.

La ética ingenieril que se ha visto puede ser una respuesta parcial a este problema. Y se ha mantenido que el compromiso de la ética profesional hacia la seguridad, salud y bienestar públicos apoya la idea de un movimiento libre y de código abierto. Finalmente se ha considerado el reto del desarrollo y la cuestión de la transferencia tecnológica desde las naciones ricas a las pobres para ver los modos en que el movimiento libre y de código abierto podría considerarse como otra forma de los movimientos intermedios, apropiados y alternativos.

Permítasenos concluir con un breve comentario de cuáles deberían ser algunas de esas propuestas prácticas. Como se ha apuntado al principio, desde la perspectiva del usuario final lo que importa no es tanto la directa disponibilidad del código fuente sino la convivencialidad del programa, un término más rico en significado que el de “software amigable”. Hay más cosas respecto a la libertad de expresión o a la apertura que la cuestión técnica de la transparencia del código fuente.

Para resumir esto de la manera *más* sintética, sugeriríamos que el software convivencial debería ser juzgado en términos de estos tres criterios:

- 1) Estabilidad: la necesidad de continuas actualizaciones coarta la convivencialidad. Es como tener un amigo que insiste en reinventarse a sí mismo. Es difícil estar a gusto con un software cada vez más nuevo o con amigos siempre nuevos. Un bien perdido en el software es la estabilidad del antiguo que fuerza a uno a actualizar. Como ejemplo personal, desde que comencé a usar WordPerfect a mediados de los años 80, me he visto forzado a actualizar y reaprender al menos cinco

veces., una media de una vez cada tres o cuatro años. Compárese con mi máquina de escribir. Compré una cuando iba a la escuela y la usé continuamente con arreglos menores y limpiezas durante treinta años. El cambio de la máquina al procesador fue una mejora positiva de mis herramientas de escritura. Ninguna actualización de software ha llegado a parecerse y, de hecho, han sido negativas. Parece que la cuestión es actualizar siempre y siempre, obligando al usuario a reaprender nuevas cualidades y detalles técnicos con un lapso de un año como máximo. El ritmo que se impone entonces es demencial. Y al mismo tiempo se cae en la idea de una “obsolescencia programada”, que transforma al usuario como mero consumidor al albur del mercado y no de sus necesidades.

- 2) **Transparencia.** No es posible entender de golpe cómo funciona un procesador. Esto es verdad en muchos aspectos. En un nivel, la opacidad aparece en los iconos de las barras de herramientas que las notas de ayuda del menú sensible al contexto no siempre resuelve y lo mismo pasa con la personalización de la configuración por defecto. En otro nivel ocurre con qué pasa tras la escena, por decirlo así con el formateado. Al menos WordPerfect permite a los usuarios acceder al código y editarlos. Microsoft Word elimina esta posibilidad para el usuario siendo mucho más misterioso y difícil ajustarlo. Y todavía hay un tercer nivel de transparencia; a los usuarios les gustaría entender qué pasa cuando un procesador se congela y exige apagar el ordenador e inicializarlo. Los mensajes de error son completamente opacos (desde luego es difícil imaginar que esos mensajes de error signifiquen algo para la comunidad técnica sin tener a mano materiales de referencia). ¿Podría diseñarse un programa de procesadores de textos que se entendiera sin tener que poseer una licenciatura en ingeniería informática?
- 3) **Simplicidad:** La mayoría de los programas de procesamiento de textos incluyen muchas más características que lo que de hecho el usuario necesita. La barra de herramientas está abarrotada. Nunca uso “formato rápido” ni “trazo de dibujo” y me he encontrado con menos personas que técnicos que saben usarlas. Me desharía muy contento de “zoom” y “cambio de vista” Pero no me puedo librar de ellas o incluso esconderlas para simplificar la pantalla.

Respecto a la crítica fácil de que no se tiene que usar todas estas características empotradas en los sistemas operativos, ciertamente es así. Pero

complican la vida del usuario final, la vida de la mayoría de nosotros. Sería interesante vivir con un ordenador cuya pantalla tuviera la estética de los Shakers o del mobiliario escandinavo. ¿Por qué esto no es una opción digital? ¿No es una de las características que el software libre y el código abierto abren como posibilidad por vez primera?

En resumen, por lo tanto se sugeriría que el movimiento para el código abierto y libre se dirigiera hacia el ideal de un software convivencial que sea estable, transparente y simplificable. Tal software podría mediar en las brechas de la transferencia de tecnología y en la brecha digital entre los laboratorios y los mercados, y entre países desarrollados y en desarrollo, e incluso quizás transformar nuestro propio desarrollo. El software libre y el código abierto debe tener el objetivo no solo de ser capaz al poder tecnológico sino a la vida buena. Finalmente, la búsqueda de la convivencialidad en la cultura de código abierto en el software debería tener implicaciones para otras formas de cultura de código abierto, desde el acceso abierto a la ciencia abierta. Esto es, en vez de concentrarse en la estructura de la producción de conocimiento para después fijarse en sus implicaciones sociales, sería razonable que de vez en cuando se rewertiera el análisis; trabajar desde lo que podría ser más útil a los humanos, hacia las preocupaciones del usuario final incorporadas en los procesos de producción de conocimiento mismo. El concepto normativo de convivencialidad, que trae una idea de cómo vivir mejor, aguantando a estos medios de vida, puede tener beneficios más allá de los ordenadores y de la tecnología informática.

BIBLIOGRAFÍA

- ABET. (1998) *Engineering Criteria 2000*. La versión más reciente de este documento, con fecha nueva para este año académico se encuentra disponible como “Criteria for Accrediting Engineering Programs,” www.abet.org. El documento original se incluye como apéndice en Lisa R. Lattuca, Patrick T. Terenzini, and J. Fredericks Volkwein, *Engineering Change: A Study of the Impact of EC2000* (Baltimore: ABET, 2006).
- ACM. (1992) “Association for Computing Machinery Code of Ethics and Professional Conduct.” Disponible en www.acm.org.
- ACM. (1999) “Software Code of Engineering and Professional Practice. Disponible en www.acm.org.
- Bertalanffy, Ludwig von. (1968) *General System Theory: Foundations, Development, Applications*. New York: Brazillier, 1968.

- Borgmann, Albert. (1984) *Technology and the Character of Contemporary Life: A Philosophical Inquiry*. Chicago: University of Chicago Press, 1984.
- Callahan, Daniel. (1998) *False Hopes: Overcoming the Obstacles to a Sustainable, Affordable Medicine*. New York: Simon and Schuster, 1998.
- DiBona, Chris, Sam Ockman, and Mark Stone, eds. (1999) *Open Sources: Voices from the Open Source Revolution*. Beijing: O'Reilly, 1999.
- DiBona, Chris, Danese Cooper, and Mark Stone, eds. (2006) *Open Sources 2.0: The Continuing Evolution*. Beijing: O'Reilly, 2006.
- Dyer, Frank Lewis, and Thomas Commerford Martin. (1929) *Edison: His Life and Inventions*. 2ª edición 2 vols. New York: Harper, 1929.
- Feenberg, Andrew. (1995) *Alternative Modernity: The Technical Turn in Philosophy and Social Theory*. Berkeley: University of California Press, 1995.
- Feller, Joseph, Brian Fitzgerlad, Scott A. Hissam, and Karim R. Lakhani, eds. (2005) *Perspectives on Free and Open Source Software*. Cambridge, MA: MIT Press, 2005.
- Florman, Samuel. (1976) *The Existential Pleasures of Engineering*. New York: St. Martin's Press, 1976.
- Goldman, Steven. (1992) "No Innovation without Representation: Technological Action in a Democratic Society," en Stephen H. Cutcliffe, Steven L. Goldman, Manuel Medina y José Sanmartin, eds., *New Worlds, New Technologies, New Issues* (Bethlehem, PA: Lehigh University Press, 1992), pp. 148-160. Versión española: "Ninguna innovación sin representación: La actividad tecnológica en una sociedad democrática," en José Sanmartin, Stephen H. Cutcliffe, Steven L. Goldman, y Manuel Medina, eds., *Nuevas Mundos, Nuevas Technologies, Nuevas Perspectivas* (Barcelona, Spain: Anthropos, 1992), pp. 269-286.
- Harrison, Lawrence E. (1985) *Underdevelopment Is a State of Mind: The Latin American Case*. Lanham, MD: University Press of America, 1985.
- Illich, Ivan. (1973) *Tools for Conviviality*. New York: Harper and Row, 1973. Versión española: *La convivencialidad* (Barcelona: Seix Barral, 1974).
- Johnson, Robert R. (1998) *User-Centered Technology: A Rhetorical Theory for Computers and Other Mundane Artifacts*. Albany, NY: State University of New York Press, 1998.
- Lieberman, Henry, Fabio Paternò, y Volker Wulf, eds. (2006) *End User Development*. Dordrecht: Springer, 2006.
- Lessig, Lawrence. (1999) *Code: And Other Laws of Cyberspace*. New York: Basic Books, 1999.
- Lovins, Amory B. (1977) *Soft Energy Paths: Toward a Durable Peace*. San Francisco: Friends of the Earth, 1977.

- Merton, Robert. (1942) "Science and Technology in a Democratic Order," *Journal of Legal and Political Sociology*, vol. 1 (1945), pp. 115-126. Reimpreso como "The Normative Structure of Science," in Robert K. Merton, *The Sociology of Science: Theoretical and Empirical Investigations*, ed. Norman W. Storer (Chicago: University of Chicago Press, 1973).
- Mitcham, Carl. (1994) "Engineering Design Research and Social Responsibility," en Kristin S. Shrader-Frechette, *Ethics of Scientific Research* (Lanham, MD: Rowman and Littlefield, 1994), pp. 153-168.
- Mitcham, Carl. (1999) "Why the Public Should Participate in Technical Decision Making," en René von Schomberg, ed., *Democratising Technology: Theory and Practice of a Deliberative Technology Policy* (Hengelo, Netherlands: International Centre for Human and Public Affairs, 1999), pp. 39-50.
- Ortega y Gasset, José. (1939) *Meditación de la técnica*. en *Obras completas*, 1ª ed. (Madrid: Revista de Occidente, 1945-1947), vol. 5, pp. 317-375.
- Polanyi, Karl. (1944) *The Great Transformation*. New York: Farrar and Rinehart, 1944.
- Popper, Karl. (1945) *The Open Society and Its Enemies*. London: Routledge, 1945. Ediciones revisadas y ampliadas: (1950), 2ª rev. (1952), 3ª ed. rev. (1957), 4º ed. rev. (1962), y 5ª ed. (1966).
- Raymond, Eric S. (1999) *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Beijing: O'Reilly, 1999.
- Schumacher, E. F. (1973) *Small Is Beautiful: Economics as if People Mattered*. New York: Harper and Row, 1973.
- Scrivener, Stephen A.R., Linden J. Ball, y André Woodcock, eds. (2000) *Collaborative Design: Proceedings of CoDesigning 2000*. London: Springer, 2000.
- Sachs, Wolfgang, ed. (1992) *The Development Dictionary: A Guide to Knowledge as Power*. London: Zed Books, 1992.
- Seffah, Ahmed, Jan Gulliksen, and Michel C. Desmarais, eds. (2005) *Human-Centered Software Engineering: Integrating Usability in the Software Development Lifecycle*. Dordrecht: Springer, 2005.
- Torvalds, Linus. (1992) "LINUX's History." July 31, 1992. (Una colección de correos electrónicos con notas explicativas disponibles desde varios sitios web. Tomadas en www.cs.cmu.edu/~awb/linux.history.html February 28, 2007.)
- Torvalds, Linus, with David Diamond. (2001) *Just for Fun: The Story of an Accidental Revolutionary*. New York: HarperCollins, 2001.

Turner, Fred. (2006) *From Counterculture to Cyberworld: Stewart Brand, the Whole Earth Network, and the Rise of Digital Utopianism*. Chicago: University of Chicago Press, 2006.

Verbeek, Peter-Paul. (2005) *What Things Do: Philosophical Reflections on Technology, Agency, and Design*. University Park, PA: Penn State University Press, 2005.

Weber, Steven. (2004) *The Success of Open Source*. Cambridge, MA: Harvard University Press, 2004.

Whitehead, Alfred North. (1925) *Science and the Modern World*. New York: Macmillan, 1925.

Winner, Langdon. (1980) "Do Artifacts Have Politics?" *Daedalus*, vol. 109, no. 1 (Winter 1980), pp. 121-136. Reimpreso en Langdon Winner, *The Whale and the Reactor: A Search for Limits in an Age of High Technology* (Chicago: University of Chicago Press, 1986), pp. 19-39.

Apéndice: Los movimientos de software libre y código abierto. Una cronología selectiva

(adaptados de opensource.org, openknowledge.org, [wikipedia](http://wikipedia.org), etc.)

Años 50 Software se distribuye habitualmente como código fuente y sin restricciones

Años 60, la cultura hacker, enfatizando la apertura y compartir el software comienza a emerger en los centros de cálculo de Stanford, Berkeley, Carnegie Mellon y el MIT.

1969 El software ARPANET se hace disponible libremente, factor clave para transformar Internet.

1970s El código abierto permanece en distribución libre entre ingenieros informáticos y de telecomunicaciones

1978 Donald E. Knuth (Stanford Univ.) comienza a trabajar en TeX, un programa de proceso de textos que se distribuye como software libre.

1980 Scribe (CMU program de formateo de texto) se privatiza, así como el LISP (MIT).

1983 Richard Stallman escribe el GNU (Gnu's Not Unix) Manifiesto, pidiendo la vuelta a la compartición pública del software y del código fuente

1984 Stallman dimite del MIT para crear GNU un sistema operativo libre compatible con Unix.

1985 Stallman crea la Free Software Foundation para apoyar su trabajo y el de otros.

1989 Stallman crea la GNU General Public License (GPL) que define las protecciones copyleft. El Copyleft permite usar, modificar y distribuir el software, pero restringe la posibilidad de colocar un copyright en él o en sus modificaciones.

1990 Stallman recibe la MacArthur Fellowship y comienza a trabajar en el kernel GNU, el HURD.

1991 William y Lynne Jolitz (Univ. of California Berkeley) describen cómo migrar de BSD (Berkeley Software Distribution) Unix a los PCs i386, haciendo posible que BSD sea un sistema operativo libre

Linus Torvalds, un estudiante de la University of Helsinki, libera un kernel de tipo Unix-con el código abierto (Linus+Unix). Vid. Torvalds y David Diamond, *Just for Fun: The Story of an Accidental Revolutionary* (New York: Harper Collins, 2001).

1992 U.S. Air Force ofrece a la New York University un contrato para un compilador de código abierto para el Ada 95 basado en GNU y llamado GNAT.

1994 Marc Ewing comienza la distribución the Red Hat GNU/Linux . Aparece el primer número de *Linux Journal*.

1996 Primer congreso en Software Libremente distribuible, Cambridge, Massachusetts, USA.

1997 Eric Raymond publica "The Cathedral and the Bazaar," sosteniendo que las licencias de código abierto dan lugar a una mayor calidad y un software más barato. (Este ensayo se convertirá en el título para un libro publicado en 1999.)

- 1998 Netscape anuncia que abrirá el código del Netscape Navigator 5.0. Eric Raymond, Bruce Perens y Tim O'Reilly sostienen que el movimiento de software libre necesita una mejor estructura y de ahí la iniciativa Open Source para certificar licencias libres o de código abierto.
Linus Torvalds y Linux aparece en la portada de *Forbes* (10 de Agosto).
- 1999 Red Hat Linux y VA Linux se hacen públicos.
- 2002 La Junta de Extremadura crea el Linex como sistema operativo para su red de ordenadores y distribuye Linex para uso público.
- 2004 Publicación de Steve Weber's *The Success of Open Source* (Cambridge, MA: Harvard University Press).