

# Contribución de capacidades de innovación en la industria de Software a través de la creación de interfases: estudio de caso de empresas mexicanas<sup>1</sup>

---

Fecha de recepción: 23 de marzo de 2006 • Fecha de aprobación: 10 de mayo de 2006

**José Luis Sampedro Hernández**

## Resumen

**E**n el presente trabajo se mostrará evidencia de los procesos de desarrollo y diseño de software y de sus implicaciones en la acumulación de capacidades tecnológicas de empresas de software; asimismo se mostrará evidencia en cómo, a través de la construcción de interfases, el conocimiento tecnológico de empresas desarrolladoras y usuarias ayuda o inhibe esa generación y acumulación. El estudio se centra en empresas mexicanas que desarrollan 'software hecho a la medida'. La estrategia tecnológica que empresas mexicanas han seguido ha sido el desarrollo y uso de software propietario (SP), sin embargo, hay empresas que están migrando al desarrollo y uso de software de fuente abierta (SFA). A diferencia de las empresas que desarrollan y diseñan SP, las de SFA pueden abrir un abanico mayor de posibilidades, pueden diversificar transversalmente su mercado, generar conocimiento y acelerar el proceso de acumulación de capacidades tecnológicas.

**Palabras clave:** Interfase, Conocimiento, Capacidades Tecnológicas, Software.

Universidad Autónoma  
Metropolitana, Unidad Xochimilco.  
sampedro@yahoo.com.mx

---

<sup>1</sup> Este trabajo es parte de una investigación más amplia. Aquí sólo se reportan algunos de los resultados más importantes de la tesis doctoral que se desarrolla en el marco del Doctorado en Ciencias Sociales, Área Desarrollo Tecnológico, en la Universidad Autónoma Metropolitana, Unidad Xochimilco, México D. F.

## Introducción

El objetivo de esta investigación es explicar y analizar el proceso de construcción de *interfases* entre empresas desarrolladoras y usuarias de software y en cómo, a través de las *interfases*, generan conocimiento y acumulan capacidades tecnológicas. Se mostrará evidencia de los procesos de desarrollo y diseño de software y de sus implicaciones en la acumulación de capacidades tecnológicas de empresas de software; asimismo, se mostrará evidencia en cómo, a través de la construcción de interfases, el conocimiento tecnológico de empresas desarrolladoras y usuarias de software ayuda o inhibe esa generación y acumulación.

El estudio se centra en empresas mexicanas que desarrollan 'software hecho a la medida'.<sup>2</sup> Algunas de ellas han generado conocimiento y acumulado capacidades tecnológicas a través de interfases simples y estables, donde el flujo de información predomina sobre el flujo de conocimiento; otras han generado interfases complejas, lo que les permite generar mayores flujos de conocimiento y potenciar la acumulación de capacidades tecnológicas. La estrategia que estas empresas mexicanas han seguido ha sido la producción y uso de SP, sin embargo, hay empresas que están migrando al desarrollo y uso SFA. A diferencia de las empresas que desarrollan y diseñan SP, las de SFA pueden abrir un abanico de posibilidades mayor, pueden diversificar transversalmente su mercado, generar conocimiento y acelerar el proceso de acumulación de capacidades tecnológicas.

La estructura de este trabajo es la siguiente: en la primera sección se explican los conceptos de interfase, conocimiento y de acumulación de capacidades tecnológicas. En la siguiente sección se presenta la estrategia de investigación. En la tercera sección se describe la trayectoria tecnológica e industrial del software. En la cuarta sección se presentan las características de la industria de software en México. En la quinta sección se presentan los resultados de la investigación. Finalmente las conclusiones.

---

<sup>2</sup> En el mercado de software comercial desde la década de los 50's se pueden encontrar distintas variedades de software de acuerdo a las plataformas de hardware desarrolladas. En Estados Unidos, Europa y Asia hay al menos tres variedades de software: software empacado, especializado (servicios y productos relacionados) y hecho a la medida. En los últimos años se ha difundido a gran velocidad el software hecho a la medida, y sobre este tipo de software tanto propietario como de fuente abierta se presenta evidencia empírica en este trabajo. El *software hecho a la medida* lo redefino como una aplicación o programa hecho con base en los requerimientos del usuario, puede ser particular a una empresa o genérico a un sector.

## I. Marco analítico-conceptual

### I.1 La interfase como punto de partida en la generación de conocimiento tecnológico y acumulación de capacidades tecnológicas

El concepto de interfase se inserta en el contexto de paradigma tecno-económico (PTE) a nivel microeconómico. En el contexto de cambio tecnológico, el PTE significa coordinación entre productores y usuarios de artefactos funcionalmente definidos (commodities) que se están intercambiando en el mercado, muestra una definición de commodity, un diseño o una interfase entre dos agentes (Andersen, 1991:121)<sup>3</sup>

El concepto de interfase no está muy distante del concepto de relación proveedor-usuario (P-U) acuñado por Lundvall (1985, 1988, 1992). El concepto de relación P-U resalta la importancia del aprendizaje por interacción entre los agentes. Los proveedores aprenden haciendo (learning-by-doing) y los usuarios por el uso (learning-by-using), y la interacción P-U permite la retroalimentación de sus aprendizajes (learning-by-interactive). Cuando las empresas se vinculan intercambian información tecnológica, conocimientos, habilidades o know how específico sobre productos, procesos y patrones de organización productiva. Esta interacción supone aprendizaje interactivo entre proveedores y usuarios lo que permite aumentar el potencial innovativo y competitivo de ambas.

Según Lundvall (1985, 1988, 1992), la relación P-U se basa en distintos tipos de relaciones que dependen de: (i) las relaciones simétricas y asimétricas entre los agentes, (ii) la estandarización de la tecnología, frecuencia y duración del intercambio, (iii) los diversos espacios (económicos y culturales) o contextos nacionales e institucionales, (iv) la dimensión organizacional y, (v) los flujos de conocimiento intra e inter empresa. Lundvall estudia este proceso interactivo con diferentes niveles de agregación, primero a nivel microeconómico y segundo a nivel de sistema nacional de innovación.<sup>4</sup> Andersen (1991) conceptualiza esta

---

<sup>3</sup> El concepto de PTE difiere del concepto de Paradigma Tecnológico, peste es visto como un medio de coordinación entre productores de conocimiento científico dentro de una comunidad científica. El concepto de paradigma tecnológico fue rediseñado por Dosi (1982) y adaptado a contextos de cambio tecnológico, donde las relaciones que determinan una innovación está dada por la coordinación entre los productores de conocimiento científico.

<sup>4</sup> Véase Lundvall 1985, 1988 y 1992, donde explica las condiciones microeconómicas e institucionales que deben imperar en un Sistema Nacional de Innovación.

relación P-U bajo un contexto de PTE a nivel micro,<sup>5</sup> y a esa relación le llama interfase.

### ***Interfase y PTE con un enfoque micro***

Con base en Andersen (1996:45) una interfase se define como una relación entre dos agentes a través del cual fluyen diferentes tipos de información. Si una interfase está simplificada y estandarizada la información necesaria para cada uno de los agentes será delimitada. Pero el proceso innovativo presupone una interacción *rica en información* y por ello presupondría interfases no simplificadas ni estandarizadas sino complejas.

En Andersen (1991 y 1996), el juego entre productores y usuarios tiene un rol fundamental en la composición de la definición del PTE,<sup>6</sup> el cual es una especificación aceptada y estable de la interfase entre productores y usuarios de artefactos tecnológicos. Desde esta perspectiva el paradigma es visto como una definición agregada de la interfase P-U, la cual toma la forma de especificaciones de commodities. Las especificaciones entre productor y usuario de un commodity pueden ser generalizadas y crear estructuras sobre las que se basa la relación de los agentes. Los cambios radicales pueden romper con las especificaciones básicas y crear nuevas condiciones para los productores y usuarios.

En el argumento anterior está incorporada la idea de estabilidad de las interfases bajo dos principios. El primero es el *commodity abstraction* que minimiza las necesidades de flujos de información, donde las interrelaciones simples y relativamente estables son de fundamental importancia para la funcionalidad del sistema económico.<sup>7</sup> Bajo este principio se crean rutinas entre

---

<sup>5</sup> A diferencia de Perez (1986, 2002 y 2003) y Freeman y Perez (1988) que definen el PTE a nivel macro, Andersen (1991, 1996) conceptualiza el PTE a nivel micro.

<sup>6</sup> Desde la perspectiva de este autor, el paradigma tecnológico, (como lo acuñó Dosi, 1982), debe incluir más explícitamente la interacción entre productores y usuarios de mercancías a los cuales la tecnología es más o menos accesible.

<sup>7</sup> La estabilidad se puede ver en términos del diseño básico de la mercancía a la cual se adaptan las formas de descubrir y solucionar problemas en un contexto donde las relaciones se simplifican, cambiar una interfase establecida es difícil porque está arraigada en las normas y en la organización de ambas partes que componen la interfase. El mundo se caracteriza por un nivel alto de reproducibilidad de rutinas y de interfases estables, lo cual permite que los agentes acumulen experiencia que puede funcionar como conocimiento re-utilizable (Andersen, 1996:47-48).

los agentes involucrados en la interfase; si las rutinas se desarrollan bajo el supuesto de una interfase estable, hay pocas posibilidades sobre un cambio de estrategia. Una tecnología puede ser madura y estandarizada, en tal caso una interfase puede ser creada y fijada y permanecer por mucho tiempo. En este sentido, se crea un sendero cuasi-irreversible de mantenimiento y evolución.<sup>8</sup>

El segundo principio es el de *aprendizaje interactivo* (interactive learning) entre productores y usuarios, una forma de interacción necesaria en la creación de nuevos productos, pero la cual al mismo tiempo incrementa las necesidades de flujos de información. Bajo este principio la interfase no puede ser estable como en el principio de commodity abstraction, ya que el aprendizaje interactivo contiene fuertes vínculos y efectos que no se encuentran por completo en el intercambio económico de commodities, al menos que se incluya información en común. Por eso las interfaces tienden a ser distintas.<sup>9</sup>

## 1.2 Interfase y conocimiento

El paradigma puede ser considerado como medio para delimitar las necesidades de los flujos de información entre los actores, de este modo proveen una mayor condición para la existencia de sistemas tecno-económicos complejos. Pero, desde la perspectiva de Andersen (1991, 1996), el juego entre productores y usuarios tiene un rol mucho más fundamental en una definición de paradigma tecnológico. Los problemas relevantes son la base de conocimiento y los procedimientos respecto a la eficiencia productiva. El conocimiento de la empresa es sólo una parte limitada de lo que existe en su medio ambiente (conocimiento

---

<sup>8</sup> El problema no es de inflexibilidad técnica, sino muestra el *lock in* de un sendero seleccionado. Ante un problema de inflexibilidad el mercado por sí mismo puede crear un *lock in* de una tecnología inferior. Arthur (1989) señala este problema en un escenario de largo plazo y diferentes formas de 'retorno' entre dos tecnologías posibles, señala que los eventos poco significantes pueden llegar a ocasionar reacciones importantes en el mercado e inducir en la selección de una tecnología inferior que presenta altas tasas de retorno. Las interfaces no son seleccionadas por el mercado sino adaptadas a las características propias de la tecnología, creando relaciones particulares entre los agentes de acuerdo al grado de madurez de la tecnología seleccionada.

<sup>9</sup> No todas las interfaces son igual de importantes para el desarrollo de un nuevo producto. Algunos usuarios juegan un rol activo y crean nuevos productos. Otros recogen los beneficios de las mejoras de los productos sin jugar un rol activo en el proceso de innovación. Otros aceptan los productos estandarizados que se encuentran en el mercado, aun si estos no son adecuados para sus necesidades, y entonces aceptan el principio de commodity abstraction (Andersen, 1996:55).

localizado), y es una parte del sistema de conocimiento tecno-económico basado en una extensa división del trabajo entre diferentes poseedores de conocimiento.

**Conocimiento tácito versus codificado.** En términos de Nonaka y Takeuchi (1994:65), el conocimiento de una empresa incluye elementos tácitos y codificados que se encuentran distribuidos en diversas áreas. El conocimiento tácito es personal y de contexto específico, difícil de formalizar y comunicar. A diferencia del conocimiento explícito o codificado, que sólo se puede transmitir a través del lenguaje formal y sistemático, el conocimiento tácito implica una dimensión cognitiva y una dimensión técnica.<sup>10</sup>

Las características de la tecnología afectan la forma en que las empresas pueden tener acceso al conocimiento relevante. Cuando el conocimiento es tácito, complejo y es parte de un sistema también complejo, los medios informales de transmisión del conocimiento son más importantes tales como los círculos de trabajo, la capacitación, monitoreo sistemático, etcétera. Con estas condiciones la *proximidad geográfica* entre los agentes es importante para garantizar una adecuada transferencia y acumulación de conocimiento. Pero, cuando el conocimiento es estandarizado, codificado, simple e independiente, son más importante los medios formales de transmisión del conocimiento como las publicaciones internas para difundir los avances de la empresa, manuales de proceso y producto, patentes, etcétera. En estas condiciones, la proximidad geográfica no es fundamental para facilitar la transmisión del conocimiento (Maleaba y Orsenigo, 1990, 1995 y 1996). En este sentido, el conocimiento puede ser codificado en un contexto, pero tácito en otro, o bien complementarios.

### **I.3 Acumulación de capacidades tecnológicas**

La literatura sobre aprendizaje y acumulación de capacidades tecnológicas ha mostrado que los procesos de aprendizaje y el desarrollo de algunas capacidades internas permite a las empresas mejorar su productividad e innovar a nivel de producto, procesos y organización a lo largo del tiempo (Maxwell, 1981; Bell,

---

<sup>10</sup> Los elementos de la dimensión cognitiva se centran en los modelos mentales: esquemas, paradigmas, perspectivas, creencias, criterios, los cuales ayudan a los individuos a percibir y definir su mundo. Los elementos de la dimensión técnica contienen know how, oficios y habilidades concretas (Nonaka y Takeuchi, 1994:66)

1984; Bell y Pavitt, 1995; Lall, 1987, 2000; Hobday, 1995, 2000, 2001; Dodgson, 1993).

Según Bell y Pavitt (1995) las capacidades tecnológicas permiten hacer un uso efectivo del conocimiento tecnológico, incluyen la capacidad para generar y administrar el cambio técnico a través de cuatro procesos: (i) por el proceso de innovación y de difusión, (ii) por el progreso técnico que generan productores y usuarios de tecnología, (iii) por el aprender haciendo, y (iv) por la estabilidad tecnológica del ciclo de vida del producto.

Los análisis económicos han dado importancia a los procesos de acumulación de capacidades tecnológicas a través de los procesos de aprendizaje y han mostrado el cambio técnico como resultado de la construcción gradual de una base mínima de conocimiento tecnológico que permite llevar a cabo actividades de innovación. La acumulación de capacidades implica más que la acumulación de destrezas y know how para operar nuevos procesos o para producir productos con ciertas especificaciones; las empresas acumulan diferentes formas de conocimiento, destrezas y experiencia requerida para generar continuamente el cambio incremental, con lo cual mejoran el desarrollo original de la tecnología actual y modifican sus insumos, productos y procesos en respuesta a los cambios dados en el mercado (Dutrénit, 2000; Dutrénit y Vera-Cruz, 2001; Vera-Cruz, 2004)

En las actividades técnicas, económicas y sociales hay una amplia gama de interfases creadas entre diversos actores. Desde la perspectiva del cambio técnico son importantes aquellas interfases que contribuyen a la mejora y al cambio incremental de procesos tecnológicos, productos y servicios. En este sentido, el concepto de interfase es de fundamental importancia para entender los tipos de relación, interacción y vinculación que una empresa genera con diversos actores tales como empresas oferentes y usuarias, universidades, instituciones de gobierno, entre otros. Las empresas no actúan aisladamente y necesitan de algún tipo de relación simple o compleja con el mercado y los actores existentes, el dinamismo y complejidad de la relación crea flujos de información y en algunos casos se pueden generar también flujos de conocimiento. Cuando la empresa puede asimilar grandes flujos de información y hay mecanismos adecuados (como una interfase) para transformar esos flujos en conocimiento, entonces la empresa puede acumular capacidades tecnológicas, es por eso que la creación de interfases podría permitir una mayor acumulación de esas capacidades.

## II. Estrategia de investigación

En este trabajo sugiero como mejor estrategia de investigación el *estudio de caso múltiple-exploratorio*. El tipo de *investigación exploratoria* aborda un problema, una cuestión o un tema nuevo de los cuales se sabe poco, se examina qué teorías y conceptos son los más apropiados, se elaboraran nuevos conceptos y de ser necesario nuevas metodologías, lo que implica expandir la frontera de conocimiento con la 'esperanza' de averiguar algo (Yin, 1994:5; Phillips y Pugh, 2001:70). Con este tipo de investigación se busca comprender por métodos cualitativos la realidad tal y como otros la experimentan.

Como parte del diseño de investigación el *caso* son empresas de software de fuente abierta y empresas de software propietario, y dada la naturaleza del estudio de caso múltiple, esta investigación tiene dos *unidades de análisis*. La primera hace referencia a la construcción de interfases y la segunda hace referencia a los procesos de desarrollo y diseño de software.

El tipo de información analizada es cualitativa y los resultados presentados no se sustentan sobre los *casos* sino sobre las *unidades de análisis*. El análisis empírico se sustenta en 17 entrevistas realizadas a líderes de proyecto y desarrolladores de software ubicados en 14 empresas de software propietario y software de fuente abierta. El trabajo de campo se realizó entre marzo de 2004 y marzo de 2005. Las empresas consideradas en el trabajo de campo se ubican en las ciudades de México, Guadalajara, León y Monterrey.

## III. El software como factor de discontinuidad dentro de las Tecnologías de Información (TI)

La industria de TI está conformada por procesos que incorporan capital, trabajo calificado y conocimiento. En especial, el sector de software tiene como insumo principal el conocimiento que está socialmente incorporado en individuos, empresas y regiones, es un proceso único que traslada ese conocimiento sobre una estructura social. Parte de ese conocimiento es tácito, indefinido, no codificado y no siempre es explícito entre los actores involucrados en el proceso (Eischen, K., 2002)



### III.1 Del software empaçado al software de fuente abierta

Durante la década de los cuarenta y cincuenta se descubrieron la mayoría de los conceptos básicos que se aplicaron al diseño de software en las décadas de 1960 a 1980.<sup>11</sup> El software era desarrollado y dominado por los fabricantes de hardware, incentivados por el complemento económico que representaba y porque eran ellos quienes tenían conocimiento sobre los lenguajes y herramientas de aplicación. Los usuarios, escasos en conocimientos para el desarrollo de software, no tenían la habilidad para entender y solucionar los problemas específicos de los sistemas que rentaban. El software era específico a cada tipo de máquina y a cada usuario. Pero a finales de la década de los cincuenta los usuarios de computadoras empezaron a desarrollar aplicaciones para su propio uso, cooperaron con los fabricantes de computadoras en el intercambio de rutinas de programación y métodos de organización con el fin de mejorar las actividades de desarrollo de software.

En 1964 IBM introduce la familia de computadoras S/360 que unifica el sistema operativo de la línea de productos de IBM. Por primera vez un proveedor ofrecía un grupo de máquinas de tamaños diferentes, en las cuales el software (sistema operativo) que funcionaba en una operaba en otra. Más tarde, en 1969 IBM establece el precio y venta del software independiente de las computadoras, de forma que incide en el surgimiento de vendedores de software independiente en la industria. Esto origina una rápida estandarización tecnológica tanto del hardware como del software. Debido a que las máquinas se estaban estandarizando y eran compatibles el software funcionaba en más de una máquina.<sup>12</sup> Entonces se pasó del software desarrollado por lo clientes a la instalación de paquetes de software desarrollados por empresas independientes.

A principios de la década de los setenta ya había un gran número de empresas independientes de software, según Steinmueller (1996:27), en Estados Unidos había entre 1500 y 2800 empresas. Con la existencia de empresas de

---

<sup>11</sup> En 1957 se introduce el lenguaje FORTRAN (Formula Translator) y en 1960 se introduce COBOL (Common Business-Oriented Language), dos de los primeros y más importantes lenguajes desarrollados.

<sup>12</sup> En este periodo el software se limitaba a productos que sólo funcionaban en tipos específicos de máquinas y sólo algunas empresas de software se especializaban en su desarrollo y venta. La estandarización de las computadoras permitiría después una estandarización de programas de software.

software independiente la industria experimentó una extensiva división del trabajo y una fuerte especialización. En esta década la industria tuvo un periodo de crecimiento de la productividad en el desarrollo de software. Sin embargo se acentuaron los problemas naturales de una industria reciente, sobre todo los relacionados con el desarrollo del software mismo y con el mantenimiento de las aplicaciones para computadoras centrales (mainframes) y minicomputadoras. Los problemas para administrar los esfuerzos de programación interno y los altos costos de mantenimiento, así como los problemas de medición del desempeño en las actividades de desarrollo y la calidad han sido desde entonces grandes retos para la industria.

En la década de los ochenta la industria del software parecía llegar a su madurez. Llegaron empresas independientes que rápidamente se consolidaron en el mercado con diseños dominantes usando una nueva plataforma de hardware.<sup>13</sup> La PC IBM, introducida en el mercado en 1981, combinó un razonable nivel de poder computacional y un sistema operativo homogéneo que facilitaría el desarrollo de aplicaciones. Con la introducción de la PC se lograron tres cosas: (i) establecer un sistema operativo estándar, (ii) que la información procesada por un usuario pudiera ser usada o compartida por otro con un mismo sistema operativo, y (iii) focalizarse sobre un producto de tal forma que se acumulan habilidades facilitando el uso de los productos.

La introducción de la PC permitió la integración de tendencias tecnológicas en software y el trabajo en red. El software y la computación evolucionaron hacia un mayor empleo de las redes. En 1990, empresas e instituciones públicas crearon redes internas basadas en la PC, llamadas *intranets*. A mediados de esa década, las empresas utilizaban Internet<sup>14</sup> para enlazar las redes internas (Intranet) y las redes privadas entre sí. El software para lograr eso provenía de proyectos patrocinados por el gobierno, como

---

<sup>13</sup> En esta época, el desarrollo del software también se asocia al desarrollo tecnológico de chips y a la capacidad de memoria. Los chips proporcionaban la rapidez del procesamiento, mientras que la memoria y los componentes relacionados aumentaban el volumen de datos que se podían mover a través de un sistema. A medida que se ampliaba la capacidad de memoria de una computadora sucedía lo mismo con la cantidad de software que funcionaba en la máquina y más complejo sería el software: contenía más funciones y habilidades para procesar una mayor cantidad de datos.

<sup>14</sup> Internet se creó a partir del desarrollo de ARPANET por parte del gobierno estadounidense en la década de los 60's a los 80's.

APARNET, o de proyectos independientes de software como Novell, Oracle y Lotus. En este sentido, el desarrollo de software en los 90's estuvo influenciado por el crecimiento en las capacidades de redes. Estos programas de aplicación, sin embargo, no han eliminado las necesidades de los usuarios por desarrollar aplicaciones de software específico a sus necesidades.

Con la introducción de la PC la habilidad de los proveedores de software dependió de algunas nuevas plataformas tecnológicas básicas, principalmente de Microsoft e Intel. Sin embargo, UNIX combinó componentes de un sistema operativo y de un conjunto de herramientas de aplicación, desarrollado principalmente por universidades e ingenieros de Bell Laboratories, proporcionando una arquitectura abierta.<sup>15</sup> En este sentido UNIX brindaba herramientas que funcionaba en cualquier tipo de computadora o plataforma,<sup>16</sup> dando al usuario independencia hacia cualquier proveedor individual, como IBM o DEC. Con el tiempo, los componentes de hardware apoyaron este enfoque, especialmente los chips RISC. A pesar de esto, Microsoft e Intel mantienen su éxito debido a que dominan una gran parte de las computadoras de escritorio y de las herramientas de software que se utilizan en todo el mundo.

#### **IV. Características de la industria de software en México**

##### **IV.1 El segmento de SP en México**

Actualmente no hay una caracterización del sector de software en México, los datos del número de empresas o tamaño en términos de empleo y ventas no son exactos. Se estima que en el país hay entre 1000 y 1500 empresas establecidas formalmente. Sin embargo, se han hecho estudios que permiten darnos una idea acerca de la estructura que podría presentar la industria de software en el país. Una encuesta realizada por AMITI (2001) a 206 empresas desarrolladoras de software propietario muestra una estructura atomística: el 87 por ciento de las empresas nacionales son micro y pequeña, con 60 empleados

---

<sup>15</sup> Llamado ahora open source o software de fuente abierta. Uno de los sistemas operativos más representativos en software de fuente abierta es Linux.

<sup>16</sup> Las principales plataformas (computadoras) comerciales han sido -en secuencia evolutiva-: computadoras centrales (mainframes), minicomputadoras, estaciones de trabajo (workstation) y microcomputadoras tales como las PC's o PDA's.

en promedio en cada empresa;<sup>17</sup> 6.7 por ciento son mediana empresa, con 175 empleados en promedio; 5.3 por ciento son gran empresa, con 600 empleados en promedio; y solamente una empresa cuenta con alrededor de 1500 empleados.

Se ha establecido un programa de fomento a la industria de software con el objeto de hacerla participar en los mercados internacionales. A través del Programa para el Desarrollo de la Industria del Software (PROSOFT)<sup>18</sup> se pretende alcanzar en el año 2013 una capacidad instalada de 5 mil millones de dólares anuales entre exportaciones y ventas internas de software, equivalente al 1.5 por ciento de la producción anual de software mundial, y crear 100 mil empleos bien remunerados. Actualmente la demanda de TI nacional se concentra en los sectores electrónico, automotriz, comercio al detalle, gobierno, financiero y corporativo, pero dejando de lado a las micro, pequeñas y medianas empresas (MIPyME's).

## IV.2 El segmento de SFA en México

En México prácticamente no hay una industria de software de fuente abierta. Es una actividad que apenas tiene 10 años de ejercerse en el país y su desempeño ha crecido muy lentamente.<sup>19</sup> De igual manera que en el segmento propietario, no hay una caracterización real del segmento de SFA. Se estima que existen en el país alrededor de 100 empresas de SFA, de las cuales, casi 90 por ciento son micro empresas, con menos de 15 empleados cada una.<sup>20</sup> El desarrollo de este tipo de software está regido por las 'comunidades de desarrolladores',<sup>21</sup> en 2004 se estimó que había 1000 desarrolladores y cerca de 10 mil usuarios a nivel nacional trabajando en diferentes líneas de código y en diversas comunidades.<sup>22</sup>

---

<sup>17</sup> Si las empresas quisieran competir en los mercados internacionales el tamaño mínimo de entrada es de 250 empleados, y debe crecer a 1 mil empleados para mantenerse competitiva y ser de 'clase mundial' (AMITI, 2001).

<sup>18</sup> A cargo de la Secretaría de Economía.

<sup>19</sup> En otros países las empresas de software propietario utilizan el software de fuente abierta como parte de una estrategia de dominación de su nicho de mercado.

<sup>20</sup> Con base en entrevistas realizadas a desarrolladores de SFA.

<sup>21</sup> Las comunidades de desarrolladores de SFA está formada por dos actores: desarrolladores y usuarios que se pueden encontrar en cualquier parte del mundo, trabajan a través del correo electrónico con líneas de código de software particulares. Von Hippel (2000 y 2005) analiza el tipo de interacciones y formación de redes estas comunidades.

<sup>22</sup> Con base en entrevistas realizadas a desarrolladores de SFA.

A pesar de su incipiente desarrollo, los programas hechos en SFA han aumentado en los últimos años, sobre todo en los servidores.<sup>23</sup> El mercado aún es limitado, pero se muestra un rápido crecimiento de la participación de programas hechos en SFA. En el año 2002, por ejemplo, la participación del software de fuente abierta en el software total nacional se estimó en 7.9 por ciento, y para el 2004 en 9.2 por ciento. El número de servidores con programas basados en SFA en el 2003 fueron 14,960, y para el año 2004 se estimó en 20,000 a nivel nacional.

En donde no ha podido penetrar el SFA es en las microcomputadoras tales como computadores de 'escritorio' o laptops, es decir en el mercado más grandes de consumo. El SFA se ha centrado en las computadoras centrales o mainframes, sobre todo en servidores y estaciones de trabajo, sin embargo, en un corto plazo empezarán a llegar programas hechos en SFA para microcomputadoras. Sin bien ya existen programas que pueden correr en sistemas operativos propietarios, esos programas aún no son lo suficientemente maduros para consolidarse en el mercado de masas. A pesar de esto, el segmento de SFA se estima que crece 10 veces más que las ventas de software comercial ya que los programas son relativamente de menor costo, flexibles y fiables e independientes.<sup>24</sup>

## V. Presentación de resultados

El ambiente tradicional donde las empresas de TI desarrollan software es de tipo propietario, pero en los últimos años han emergido individuos y empresas que desarrollan software en un ambiente de fuente abierta. Las empresas que desarrollan software en un 'ambiente propietario' se ubican en un contexto de propiedad industrial donde el código fuente es propiedad de ellas. Mientras que las empresas que desarrollan software en un 'ambiente de fuente abierta' se ubican en un contexto donde el acceso al código fuente no tiene restricciones,

---

<sup>23</sup> Un servidor es una computadora que administra información a un grupo de PC's, su capacidad de procesamiento es mayor a la de éstas. En México, el SFA ha penetrado mayormente en el segmento de servidores.

<sup>24</sup> Sistemas operativos y programas en SFA ya empiezan a funcionar en grandes empresas como Merrill Lynch, 7-Eleven; y algunos gobiernos como el de España, China, Alemania y Noruega han optado por usar SFA como parte de su estrategia de desarrollo del sector.

cualquier individuo, grupo de individuos o empresas que desarrollen nuevo código pueden optar por hacerlo de dominio público. En ambos ambientes se crean diferentes tipos de interfases, 'simples, estandarizadas y complejas', pero en el ambiente de fuente abierta el esquema bajo el cual se desarrolla software está marcado por una alta interacción entre desarrolladores y 'usuarios conscientes', ubicados en un espacio virtual denominado 'comunidades de desarrolladores'. Como veremos en los siguientes apartados, esto tiene implicaciones directas en el proceso de creación de interfases, generación de conocimiento y acumulación de capacidades tecnológicas.

### **V.1 Creación de interfases**

Con base en el marco analítico-conceptual desarrollado en la primera sección y a la evidencia empírica, en este trabajo se ajusta el concepto de interfase. En este sentido, una interfase la redefino como una interacción entre proveedores y usuarios de artefactos tecnológicos que a través de dispositivos socio-técnicos transforman flujos de información en flujos de conocimiento. Los dispositivos sociales están caracterizados por la relación, vinculación e interacción entre diversos agentes sociales tales como individuos, empresas, organizaciones y gobiernos. Los dispositivos técnicos están caracterizados por artefactos tecnológicos que permiten a los diferentes agentes sociales transformar acciones comandadas en acciones realizadas, tales como el teclado de una computadora, los iconos de un programa de software, etcétera. El componente principal y común en cada uno de los dispositivos es el flujo y contenido de información. Los dispositivos son un habilitador en la transformación de flujos de información en flujos de conocimiento. En este trabajo se pone énfasis sobre los dispositivos sociales.

Las empresas de software dan muestra de la existencia y creación de diferentes interfases con diferentes agentes. Podemos encontrar interfases simples, estandarizadas y complejas. Las empresas interactúan con tres actores: (i) el mercado en el que cohabitan, (ii) instituciones públicas y privadas, y (iii) empresas nacionales e internacionales.

**Una interfase simple** implica que la empresa de software utiliza información (necesidades y requerimientos de los usuarios) que puede encontrar a través del mercado para el desarrollo de aplicaciones o programas de software

simples tales como programas de nómina, programas contables o administrativos, entre otros. Empresas ubicadas en ambos ambientes<sup>25</sup> han establecido al menos un proyecto de desarrollo tomando en consideración la información que el mercado les proporciona.

Si los procesos productivos y administrativos son simples, la información y el conocimiento acerca de ellos están codificados y se difunden entre las empresas del sector. Los medios de difusión de ese tipo de procesos son a través de Internet, correo electrónico, intranets o extranets. De esta forma, las empresas de software pueden acceder a ese tipo de información sin necesidad de interactuar directamente con el usuario, o al menos la interacción no es intensa, las formas de interacción se remiten a contrato de compra-venta y a la implantación del programa. Los encuentros entre el proveedor y usuario, si los llega a haber, son informales y se da en el contrato de compra-venta<sup>26</sup>. En este sentido, el mercado proporciona la información necesaria para el desarrollo de aplicaciones o programas con base en las necesidades y requerimientos estandarizados de los usuarios.

***Una interfase estandarizada*** implica que la empresa de software utiliza información (necesidades y requerimientos de los usuarios) que no puede encontrar fácilmente en el mercado sino que necesita del usuario para obtenerla. Las aplicaciones o programas no se remiten solamente a programas contables, de administración, punto de venta o páginas web, sino también a programas o sistemas que interactúan al mismo tiempo con procesos productivos, de distribución y administración de la empresa. En este sentido, la interacción con el usuario requiere de información más específica para llevar a cabo el proyecto de desarrollo de software, información que no se puede encontrar fácilmente en el mercado, por ejemplo a través de Internet.

La empresa usuaria tiene un cierto grado de complejidad que requiere de una relación cercana con la empresa desarrolladora para dar a conocer sus requerimientos y necesidades centrales o medulares para el proyecto. Las necesidades básicas se pueden encontrar en el mercado, las medulares no, requieren de esa interacción.

---

<sup>25</sup> De aquí en adelante cuando me refiero a 'ambos ambientes' debemos tomar en consideración el desarrollo de software en un ambiente propietario y en un ambiente de fuente abierta, tal como se explicó en la sección IV.

<sup>26</sup> Entrevista con líderes y administradores de proyecto.

En **una interfase compleja** la interacción con el usuario requiere de información más específica para llevar a cabo el proyecto de desarrollo de software, información que no se puede encontrar fácilmente en el mercado, sólo a través del usuario. Una gran cantidad de información y conocimiento acerca de los procesos productivos y administrativos no están codificados y no se difunden rápidamente entre las empresas del sector en donde se ubican los usuarios.

En una interfase compleja el usuario es diferente al que se puede encontrar en una interfase simple y estandarizada. En un ambiente de fuente abierta el usuario tiene información y conocimiento acerca de las herramientas que usan las empresas de software, lo que les permite interactuar con las empresas desarrolladoras y mejorar la aplicación o programa. Este usuario denominado 'usuario conciente' no necesariamente usa la aplicación o programa que la empresa desarrolla, el propósito del usuario conciente es mejorar la aplicación hasta cierto grado y en determinado momento hacer uso de ella. La forma de interacción es por lo regular informal, el medio de comunicación es a través del correo electrónico y los proyectos que desarrollan son de larga duración. Un proyecto puede durar hasta dos años y seguir escalándolo. El desarrollador de software y el usuario pueden interactuar hasta por 3 años, dependiendo de la complejidad de la aplicación, hay algunas aplicaciones menos complejas y duran menos tiempo, seis meses o un año.<sup>27</sup>

A través de este tipo de interacción, la empresa asimila una gran cantidad de conocimiento e información y aprende a gran velocidad. Al interior de las comunidades de desarrolladores de software, los desarrolladores tienen acumulado una gran cantidad de proyectos y experiencias que comparten entre ellos y con los usuarios concientes. Esto es posible sólo en un contexto de software de fuente abierta.

Pero la complejidad de la interfase no se caracteriza sólo por tener usuarios 'concientes', sino también por realizar proyectos complejos de vinculación. Algunas empresas han emprendido proyectos de vinculación con empresas con la finalidad de formar capital humano de bajo costo pero calificado, capacitándolo al interior de la universidad y no al interior de la empresa.

---

<sup>27</sup> Entrevistas con líderes y administradores de proyecto.



## V.2 Generación de conocimiento

El conocimiento que se genera a través de las diferentes interfases tiene grados de complejidad y dinamismo. En una interfase simple y estandarizada la relación y las interacciones entre P-U tienen mayores flujos de información y son predominantes sobre los flujos de conocimiento. El tipo de conocimiento predominante en este tipo de interfases es codificado y difundido a través de medios formales como la capacitación o publicaciones.

En cambio, en una interfase compleja hay una mayor interacción entre P-U, los flujos de información son largos y predomina el flujo de conocimiento. En un ambiente propietario el conocimiento se genera predominantemente al interior de la empresa, ayudado con la sistematización de procesos y administración del conocimiento. A pesar de los problemas en términos de racionalización del software,<sup>28</sup> las empresas propietarias tienden a sistematizar sus procesos, a medir la calidad y eficiencia, lo que en gran medida les permite administrar una mayor cantidad de información y conocimiento. Mientras que en un ambiente de fuente abierta, el conocimiento es generado a través de las comunidades de desarrollo de software, donde los usuarios tienen una mayor participación en la interacción, es decir, la interacción es frecuente y dinámica y entre más interacción haya con el usuario más puro será el producto/servicio generado.<sup>29</sup>

En un ambiente propietario los desarrolladores trabajan en equipo al interior de la empresa y de la relación P-U, y a diferencia del ambiente de fuente los desarrolladores se apoyan en las comunidades de desarrolladores que trabajan en redes.

## V.3 Acumulación de Capacidades tecnológicas

El *proceso de desarrollo* de SP y SFA tiene algunas particularidades. El SP usualmente se desarrolla dentro del marco conceptual del secreto comercial, en

---

<sup>28</sup> El problema de la industria del software aún no resuelve los problemas de sistematización, repetibilidad y cuantificación de los procesos de desarrollo.

<sup>29</sup> *Idem.*

donde el que pueda liberar una versión más poderosa en menor tiempo podrá tener una mejor oportunidad de dominar una parte del mercado. En este sentido la protección del código fuente de dicha aplicación es de la mayor importancia. También existen piezas de software propietario no-comercial, de tal forma que el componente financiero no es el único factor que se toma en cuenta para decidir si un software es propietario o de fuente abierta. El software se vuelve de fuente abierta o propietario en tanto su creador o propietario deciden hacer público, o no, el código fuente y garantizar el acceso y modificación del mismo para todos.

El SFA es desarrollado usualmente por un pequeño grupo de líderes, los 'administradores' del proyecto, y una gran cantidad de desarrolladores que aportan código, mejoras, ideas, etcétera. Este equipo de trabajo se comunica primordialmente por correo electrónico y en función de la complejidad utiliza herramientas más sofisticadas de desarrollo colaborativo.

El proceso de desarrollo de software es el que potencia más las capacidades de la empresa, no así el proceso de diseño. Las etapas del proceso de diseño parten de las especificaciones de requerimientos, validación interna de requerimiento, validación de requerimientos de parte del cliente, etapa de diseño de alto nivel (validación del cliente), etapa de diseño de bajo nivel basado en el anterior, finalmente el desarrollo.

Desarrollar software bajo un tipo de ambiente permite generar mayor o menor capacidad tecnológica. En un ambiente propietario es más dinámico ese proceso ya que en buena medida las empresas se empeñan en estandarizar los procesos y administrar el conocimiento generado. En un ambiente de fuente abierta el proceso de desarrollo de software sigue con la misma inercia artesanal de hace décadas, sin embargo con una mayor generación de conocimiento el aumento de sus capacidades tecnológicas dependerá de la resolución de los problemas relacionados con la sistematización de los procesos de desarrollo.

## **Conclusiones**

-La industria de software está cargada de un fuerte dinamismo donde las cosas cambian rápidamente y donde los tipos de interacción entre P-U son dinámicos. De esta forma las interfases se modifican constantemente ante la búsqueda de soluciones a problemas específicos.

-Ante la búsqueda de soluciones las interfases se vuelven dinámicas y cruciales en el proceso de generación de conocimiento y acumulación de capacidades tecnológicas. En este proceso el conocimiento tácito es un elemento relevante ya que algunas interfases están dominadas por este tipo de conocimiento. En este trabajo las interfases que se plantean son parte medular de un sistema mucho más amplio, un sistema donde se articulan diferentes unidades de conocimiento y no sólo de información.

-En las interfases estables lo que está en juego es la información y en menor grado el conocimiento. La base de conocimiento que se genera en este tipo de interfase está sobre los procesos, es decir en cómo mejorar los procesos de desarrollo de software, cómo hacer las cosas más rápido y con menor costo, etcétera, pero no sobre en cómo generar nuevos productos y servicios. Mientras que las interfases complejas tienen un alto contenido de conocimiento y largos flujos de información asociado a la generación de nuevos productos y servicios.

-En ese sentido, el tipo de interfase compleja parece tener una mayor posibilidad de generar conocimiento y no sólo información, como en las interfases simples o estandarizadas, pero eso depende en gran medida de la madurez de la tecnológica a la que se asocie la interfase. Cuando un sector es tecnológicamente maduro puede generar interacciones entre los actores con flujos de información mayores a los flujos de conocimiento, en buena medida porque el mercado puede proveer cierta información requerida en la interfase simple y estandarizada, pero cuando ese sector incorpora nuevas tecnologías como software en sus procesos y productos la información proporcionada por el mercado no es suficiente y por ello requerirá de un alto grado de interacción con diversos actores, es decir la interfase tiende a ser compleja y dinámica.

-Finalmente, el nicho de mercado que representa grandes oportunidades para las empresas desarrolladoras de software hecho a la medida se ubica en sectores tanto tradicionales como de alta tecnología que demanden software de manufactura (embedded software) y no sólo software administrativo (sobre el cual se han focalizado una gran cantidad de empresas mexicanas desarrolladoras de software hecho a la medida); en la búsqueda de esas oportunidades el software de fuente abierta ofrece la posibilidad de acceder a nuevos nichos de mercado a menor costo laboral y de desarrollo de software y con mayores márgenes de competitividad.

## Bibliografía

- AMITI (2001). *Esquema de Apoyo Gubernamental a la Industria del Software*. AMITI, México.
- Andersen, E. S. (1991). "Techno-economic Paradigms as Typical Interfaces between Producers and Users", in *Evolutionary Economics*, Vol. 1, Num. 2. USA:119-144.
- Andersen, E. S. (1996). *Evolutionary Economics. Post-Shumpetian Contributions*. Biddles Ltd. Guildtord and King's Lynn, London.
- Arthur, W. B. (1989). "Competing Technologies, Increasing Returns, and Lock-In by Historical Events", *The Economic Journal*, Vol. 99, Num. 394, March, pp 116-131.
- Bell, M. (1984). "Learning and the Accumulation of Industrial Technological Capacity in Developing Countries", in Fransman and K. King (eds), *Technological Capability in the Third World*. Macmillan, London: 187-209.
- Bell, M. y K. Pavitt (1995). "The Development of Technological Capabilities", in I. Haque (ed.) *Trade, Technology and International Competitiveness*. The World Bank, Washington: 69-101.
- Dodgson, M. (1993). "Organizational Learning: A Review of Some Literatures". In *Organizational Studies*. Vol. 14, Num. 3. UK: 375-394.
- Dosi, G. (1982). "Technological Paradigms and Technological Trajectories", *Research Policy*, Num 11.
- Dutrénit, G. (2000). *Learning and Knowledge Management in the Firm: From Knowledge Accumulation to Strategic Capabilities*, Cheltenham: Edward Elgar, U. K.
- Dutrénit, G. y A. O. Vera-Cruz (2001). "Aprendizaje, Conocimiento y Capacidades Tecnológicas", Monografía No 2 del proyecto *Aprendizaje Tecnológico y Escalamiento Industrial: Generación de Capacidades de Innovación en la Industria Maquiladora de México*, COLEF/FLACSO/UAM.
- Eischen, K. (2002). *The Social Impact of Informational Production: Software Development as an Informational Practice*. Working Paper Series 2002-1 University of California, USA.
- Freeman, C. and C. Perez (1988). "Structural Crises of Adjustment: Business Cycles and Investment Behavior", in Dosi *et al.*, (eds.), *Technical Change and Economic Theory*. Pinter Publishers, London, UK.
- Hobday, M. (1995). *Innovation in East Asia*. Edward Elgar, UK.
- ..... (2000). "East Versus Southeast Asian Innovation Systems: Comparing OEM and TNC led Growth in Electronics", in Kim and Nelson (eds.), *Technology, Learning, & Innovation Experiences of Newly Industrializing Economies*, Cambridge, UK.
- ..... (2001). "OEM vs TNC-led, Growth in Electronics: Comparing East and South East Asian Innovation System", en Dutrénit, Garrido y Valenti (eds.), *Sistema Nacional de Innovación: Temas Para el Debate en México*, UAM, México.

- Lall, S., (1987). "The Acquisition of Technological Capability by India". In *Learning to Industrialize*. Macmillan, London.
- .....(2000). "Technological Change and Industrialization in the Asian Newly Industrializing Economies: Achievements and Challenges", in Kim and Nelson (eds.), *Technology, Learning, & Innovation Experiences of Newly Industrializing Economies*, Cambridge, UK.
- Lundvall, B. A. (1985). *Product Innovation and User-Producer Interaction*. Aalborg University, Press Aalborg.
- ..... (1988). "Innovation as an Interactive Process: from User-Producer Interaction to the National System of Innovation", in Dosi, *et al.*, (eds.), *Technical Change and Economic Theory*, Pinter Publishers, UK.
- ..... (1992). *National System of Innovation. Towards a Theory of Innovation and Interactive Learning*. Pinter Publisher, London.
- Malerba, F. y L. Orsenigo (1990). "Technological Regimes and Patterns Innovation: a Theoretical and Empirical Investigation of the Italian Case", in Heertje A. (ed.), *Evolving Technology and Market Structure*, University of Michigan Press, USA.
- ..... (1995). "Shumpeterian Patterns of Innovation", in *Cambridge Journal of Economics*, Num. 19, USA: 47-65.
- ..... (1996). 'The Dynamics and Evolution of Industries', in *Industrial and Corporate Change*, Vol.5, Num. 1, Oxford University Press, USA.
- Maxwell, P. (1981). *Technological Policy and Firm Learning in Less Development Countries: a Case Study of the Experience of the Argentina Steel Firm Acindar SA*. D. Phill, SPRU, Sussex. Cap. 2.
- Nonaka, I. y H. Takeuchi (1994). *La Organización creadora de Conocimiento. Cómo las Compañías Japonesas crean la Dinámica de la Innovación*. Oxford, Londres.
- Pérez, C. (1986). "Las Nuevas Tecnologías: Una Visión de Conjunto", en C. Ominami (ed) *La Tercera Revolución Industrial. Impactos Internacionales del Actual Viraje Tecnológico*, GEL, Argentina.
- ..... (2002). *Technological Revolutions and Financial Capital*. Edward Elgar, UK.
- ..... (2003). "Revoluciones Tecnológicas, Cambios de Paradigma y de Marco Socioinstitucional", en J. Aboites y G. Dutrénit (eds), *Innovación, Aprendizaje y Creación de Capacidades Tecnológicas*, UAM, México.
- Phillips, E. y D. S. Pugh (2001). *Cómo Obtener un Doctorado. Manual para Estudiantes y Tutores*. Gedisa Edit., Barcelona.
- Steinmueller, E. W. (1996). "The U.S. Software Industry: An Analysis and Interpretative History", in D. Mowery (ed.), *The International Computer Software Industry*. Oxford Univ. Press, USA.

- Vera-Cruz, A. O. (2004). *Cultura de la Empresa y Comportamiento Tecnológico*. Porrúa-UAM, Méx.
- Von Hippel, E. (2000). "Horizontal innovation networks-by and for users". *Working Paper Num. 4366-02*, MIT Sloan School of Management, June, US.
- ..... (2005). *Democratizing Innovation*. MIT Press, USA.
- Yin, R. K. (1994). *Case Study Research: Design and Methods*. SAGE Publications, USA.