



Casa abierta al tiempo

**UNIVERSIDAD AUTÓNOMA METROPOLITANA  
UNIDAD XOCHIMILCO**

---

DIVISIÓN DE CIENCIAS SOCIALES Y HUMANIDADES

**DOCTORADO EN CIENCIAS SOCIALES**  
ESPECIALIDAD: ECONOMÍA Y GESTIÓN DE LA INNOVACIÓN

---

*Capacidad de Absorción de Información y  
Conocimiento a partir de la Creación de  
Interfases en la Industria de Software: El Caso de  
Empresas Mexicanas.*

Tesis para optar al Grado de Doctor en Ciencias Sociales,  
con especialidad en Economía y Gestión de la Innovación.

Presenta:

**José Luis Sampedro Hernández.**

Tutor de Tesis:

**Dr. José Alexandre Oliveira Vera-Cruz.**

México, D. F., agosto de 2008.



## **DEDICATORIA**

**A quienes me dieron la vida, Teresa y Lázaro,  
A quienes han velado por ella, Griselda y Alberto,  
A mis hermanas y hermanos, Xochitl, Crystela, Alejandro,  
Ulises, Christopher y Jonathan,  
A quien apareció en mi vida y me ha acompañado en el  
proceso de vivirla, Ada.**



## **AGRADECIMIENTOS**

Expreso mi profunda gratitud a mi tutor de tesis José Alexandre O. Vera-Cruz por su apoyo y enseñanzas a lo largo del proceso de formación doctoral. Sus comentarios, críticas constructivas y consejos me ayudaron a madurar el proceso de investigación que será siempre parte de mi evolución profesional.

Agradezco a los líderes de proyecto y desarrolladores de software que me brindaron la oportunidad de llevar a cabo esta tarea. Especial agradecimiento a Armando Gallegos y Luis Vinicio por compartir conmigo información y experiencias que fueron de crucial importancia para comprender una parte de la diversa y compleja realidad mexicana. Las omisiones y errores en esta tesis son de mi entera responsabilidad.

Especial agradecimiento a los profesores del área de especialización del Doctorado -Economía y Gestión de la Innovación-, quienes con sus enseñanzas y experiencias contribuyeron a mi formación: Dra. Gabriela Dutrénit, Dr. Daniel Villavicencio y Dr. Arturo Lara.

Siempre en agradecimiento con mis colegas y amigas(os) con quienes he compartido experiencias y de quienes siempre aprendo: Marina Sánchez, Enrique Kato, Ma. Antonieta Barrón, Manuel Hernández, Aryenis Arias, Lourdes Ávila, Salvador Cisneros y Claudia de Fuentes.

Finalmente, y no por ello menos importante, también agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) por su apoyo financiero y a la Universidad Autónoma Metropolitana por haberme dado la oportunidad de participar en uno de sus posgrados de excelencia.

## LISTADO DE CUADROS Y FIGURAS

	Pág.
<b>Capítulo Uno</b>	
Cuadro 1.1. Actividades y roles en la empresa de software	15
Cuadro 1.2. Características evolutivas de las tecnologías de hardware, 1940-1990	17
Cuadro 1.3. Características evolutivas de las tecnologías de software, 1940's-1990's	20
Cuadro 1.4. Configuración social y tecnológica del software de fuente abierta, 1960-2000	24
Cuadro 1.5. Características de los modelos de producción en masa vs SIPC	28
 <b>Capítulo Dos</b>	
Cuadro 2.1. Estructura de la industria de software en India, Irlanda e Israel, 2002	40
Cuadro 2.2. Estructura de la industria de software en Brasil y Argentina, 2002	43
Cuadro 2.3. Tecnologías de Información y Comunicación en México. Millones de dólares	45
Cuadro 2.4. Clasificación de las actividades de software	46
Cuadro 2.5. Participación por segmento de software, y sectores de demanda, 2005	47
 <b>Capítulo Tres</b>	
Cuadro 3.1. Características de los dos principios en el diseño de interfases	80
 <b>Capítulo Cuatro</b>	
Figura 4.1. Estructura conceptual	118
Cuadro 4.1. Tipo de interfase y efectos esperados en la capacidad de absorción y generación de nuevo conocimiento	126
 <b>Capítulo Cinco</b>	
Cuadro 5.1. Tipo de estudio de caso	131
Cuadro 5.2. Empresas incluidas en el estudio exploratorio	138
Cuadro 5.3. Empresas estudiadas y su ubicación geográfica	138
Cuadro 5.4. Fuentes de información utilizadas en el trabajo de campo	139

Cuadro 5.5. Entrevistas incluidas en la investigación	140
Cuadro 5.6. Operacionalización de los conceptos	141
Cuadro 5.7. Resumen de las diferencias y similitudes entre los estudios de caso	148

### **Capítulo Seis**

Cuadro 6.1. Operacionalización de la interfase	159
Cuadro 6.2. Características de una interfase simple	167
Cuadro 6.3. Características de una interfase semi-compleja	178
Figura 6.1. Dos modelos diferentes de generación de recursos humanos para la industria del software	188
Cuadro 6.4. Características de una interfase compleja	191
Cuadro 6.5. Tipo de interfase asociada a las empresas de software	192
Cuadro 6.6. Características de las interfases	193

### **Capítulo Siete**

Cuadro 7.1. Operacionalización de la Capacidad de Absorción	200
Cuadro 7.2. Principales fuentes y actividades para incrementar la Capacidad de Absorción	225
Cuadro 7.3. Interfase y nivel Capacidad de Absorción	227

### **Capítulo Ocho**

Cuadro 8.1. Creación de conocimiento e información a partir de diferentes proyectos	251
Cuadro 8.2. Caracterización de la innovación en el segmento industrial de software hecho a la medida	253

## ÍNDICE

	<b>Pág.</b>
Portada	i
Dedicatoria	ii
Agradecimientos	iii
Listado de cuadros y figuras	iv
Índice	vi

### **INTRODUCCIÓN**

1. Planteamiento	1
2. Objetivo y preguntas de investigación	6
3. Estructura de la investigación	8

## **PARTE I**

### **EVOLUCIÓN TECNOLÓGICA DE LA INDUSTRIA DE SOFTWARE A NIVEL INTERNACIONAL Y SU CARACTERIZACIÓN EN MÉXICO**

#### **CAPÍTULO UNO**

##### **EVOLUCIÓN DE LA INDUSTRIA DEL SOFTWARE**

Introducción	11
1.1 Definición	11
1.2 Tipología y actividades de software	13
1.3 El software como factor de discontinuidad en la industria de hardware	16
1.3.1 Evolución de la industria de computadoras, 1940-1990	17
1.3.2 Del software empaquetado al software de fuente abierta: seis etapas evolutivas	19
1.4 Dos formas de caracterizar la tecnología de software	26
1.4.1 El desarrollo de software desde el enfoque de sistemas de productos complejos	26
1.4.2 El desarrollo de software desde el enfoque de software incorporado (embedded software)	31
1.5 Racionalización del desarrollo de software	32
1.6 Conclusiones del capítulo	36

#### **CAPÍTULO DOS**

##### **JERARQUÍA DE PROBLEMAS EN LA INDUSTRIA MEXICANA DE SOFTWARE**

Introducción	39
2.1 La industria de software en un contexto internacional	40
2.2 La industria del software en México	44
2.2.1 Software propietario	48
2.2.2 Software de fuente abierta	50
2.2.3 Características de la oferta y demanda de recursos humanos en la industria mexicana de software	51
2.3 Estrategia de desarrollo nacional y regional para la industria de software	54
2.3.1 Estrategia sectorial	54
2.3.2 Estrategias locales	56
2.4 Conclusiones	60



## **PARTE II**

### **INTERFASE, CAPACIDAD DE ABSORCIÓN Y GENERACIÓN DE CONOCIMIENTO: REVISIÓN DE LA LITERATURA**

#### **CAPÍTULO TRES**

##### **INTERFASE COMO MECANISMO CRUCIAL PARA IDENTIFICAR INFORMACIÓN Y CONOCIMIENTO EXISTENTES EN EL ENTORNO**

Introducción	64
3.1 Interfase y paradigma tecno-económico (PTE) con un enfoque micro	65
3.1.1 De la invención a la revolución tecnológica	65
3.1.2 Insumos clave y cambio de PTE	67
3.2 La interacción proveedor-usuario como antecedente de la interfase	69
3.2.1 Interacción proveedor-usuario y sus diferentes dimensiones	70
3.3 Interfase en el esquema de PTE	73
3.3.1 Dos principios de interfase: ‘estabilidad de la interfase’ y ‘aprendizaje interactivo’	76
3.3.2 Operacionalizando el concepto de interfase	82
3.4 El papel de los usuarios y las comunidades en los procesos de innovación	83
3.5 Evaluación crítica del diseño de interfaces	90

#### **CAPÍTULO CUATRO**

##### **CAPACIDAD DE ABSORCIÓN Y GENERACIÓN DE CONOCIMIENTO**

Introducción	93
4.1 Capacidades de absorción	94
4.1.1 Dimensiones conceptuales y mecanismos de la capacidad de absorción	96
4.1.2 El problema de la internalización de la información y conocimiento	103
4.1.3 Evaluación crítica de la literatura sobre capacidades de absorción	104
4.2 Información y conocimiento como insumos clave en el proceso de innovación	106
4.2.1 Información versus conocimiento	106
4.2.2 Conocimiento tácito y conocimiento codificado	108
4.2.3 El concepto de conocimiento en operación	112
4.2.4 Evaluación crítica del concepto de información y conocimiento	113
4.3 Estructura conceptual y categorías analíticas	115
4.3.1 Interfase	119
4.3.2 Capacidad de absorción	121
4.3.3 Conocimiento y flujos de información	124
4.4 Conclusiones del capítulo	126

#### **CAPÍTULO CINCO**

##### **METODOLOGÍA DE LA INVESTIGACIÓN**

Introducción	127
5.1 Metodología en las ciencias sociales	128
5.1.1 Investigación cualitativa	128
5.1.2 Investigación de estudio de caso	131
5.2 Diseño de la investigación	132
5.2.1 Preguntas de investigación	133
5.2.2 Propositiones teóricas	134
5.2.3 Unidad de análisis	136

5.2.4	Proposiciones teóricas y criterio de evaluación	136
5.3	Operacionalización de la investigación	136
5.3.1	Selección de escenarios	137
5.3.2	Fuentes de información	139
5.3.3	Recolección de datos e información	140
5.3.4	Formas de interpretación (análisis de la información)	140
5.4	Perfil de los estudios de caso	142
5.4.1	Empresas ubicadas en un contexto de software de fuente abierta	143
5.4.2	Empresas ubicadas en un contexto de software propietario	144
5.4.3	Diferencias y similitudes entre los caso	146
5.4.4	Algunos criterios a considerar en la configuración de interfases	150

### **PARTE III**

## **DISEÑO DE INTERFASES, CAPACIDAD DE ABSORCIÓN Y GENERACIÓN DE CONOCIMIENTO: EVIDENCIA EMPÍRICA**

### **CAPÍTULO SEIS**

#### **CREACIÓN DE INTERFASES SIMPLES, SEMI-COMPLEJAS Y COMPLEJAS ENTRE LAS EMPRESAS Y OTROS AGENTES**

	Introducción	155
6.1	Enfoque y método	155
6.2	Creación de interfases simples	160
6.2.1	Tipos de agente en la construcción de la interfase simple	162
6.2.2	Tipos de interacción y estabilidad de la tecnología en la construcción de la interfase simple	164
6.2.3	Intensidad y formalidad en la interfase simple	165
6.3	Creación de interfases semi-complejas	168
6.3.1	Tipos de agente en la construcción de la interfase semi-complejas	169
6.3.2	Tipos de interacción y estabilidad de la tecnología en la construcción de la interfase semi-compleja	171
6.3.3	Intensidad y formalidad en la interfase semi-compleja	176
6.4	Creación de interfases complejas	179
6.4.1	Tipos de agente en la construcción de la interfase compleja	180
6.4.2	Tipos de interacción y estabilidad de la tecnología en la construcción de la interfase compleja	183
6.4.3	Intensidad y formalidad en la interfase compleja	189
6.5	Resumen y conclusiones del capítulo	192

### **CAPÍTULO SIETE**

#### **DIFERENTES NIVELES DE CAPACIDAD DE ABSORCIÓN A PARTIR DE LAS INTERFASES CREADAS ENTRE LAS EMPRESAS Y OTROS AGENTES**

	Introducción	197
7.1	Enfoque y método	198
7.2	Bajo incremento de la capacidad de absorción a partir de una interfase simple	201
7.2.1	Conocimiento previo	201
7.2.2	Flujos de información y conocimiento al interior de la empresa.	202
7.2.3	Actividades de aprendizaje tecnológico	205
7.3	Moderado incremento de la capacidad de absorción a partir de una interfase semi-compleja	208

7.3.1 Conocimiento previo	209
7.3.2 Flujos de información y conocimiento al interior de la empresa	210
7.3.3 Actividades de aprendizaje tecnológico	211
7.4 Significativo incremento de la capacidad de absorción a partir de una interfase compleja	215
7.4.1 Conocimiento previo	215
7.4.2 Flujos de información y conocimiento al interior de la empresa	217
7.4.3 Actividades de aprendizaje tecnológico	220
7.5 Resumen y conclusiones del capítulo	224

## **CAPÍTULO OCHO**

### **CREACIÓN DE INFORMACIÓN Y CONOCIMIENTO A PARTIR DEL INCREMENTO EN LA CAPACIDAD DE ABSORCIÓN**

Introducción	229
8.1 Interfase simple y baja capacidad de absorción como determinantes de la ‘replica de información y conocimiento de propósito general’	231
8.1.1 Identificación de información y conocimiento a partir de la interfase simple	231
8.1.2 Internalización de información y conocimiento a partir de la interfase simple	232
8.1.3 Creación de información y conocimiento en un entorno de interfases simples y baja capacidad de absorción	233
8.2 Interfase semi-compleja y moderada capacidad de absorción como determinantes de la ‘mejora incremental y desarrollo de nuevos productos de software’	235
8.2.1 Identificación de información y conocimiento a partir de la interfase semi-compleja	235
8.2.2 Internalización de información y conocimiento a partir de la interfase semi-compleja	237
8.2.3 Creación de información y conocimiento en un entorno de interfases semi-complejas y moderada capacidad de absorción	238
8.3 Interfase compleja y significativa capacidad de absorción como determinantes de la ‘creación de habilidades y conocimiento específicos’	241
8.3.1 Identificación de información y conocimiento a partir de la interfase compleja	241
8.3.2 Internalización de información y conocimiento a partir de la interfase compleja	242
8.3.3 Creación de información y conocimiento en un entorno de interfases complejas y significativa capacidad de absorción	244
8.4 Conclusiones del capítulo	250

## **CONCLUSIONES GENERALES**

Introducción	255
1. Creación de interfases en un entorno de rápido cambio tecnológico	257
2. Absorción de información y conocimiento	260
3. Transformación de información en conocimiento en nuevos productos a partir de las interfases	262
4. Aportaciones teóricas y empíricas de la tesis	265
5. Temas a considerar para futuras investigaciones	268
6. Recomendaciones para la política de fomento de la industria de software	270

<b>BIBLIOGRAFÍA</b>	274
---------------------	-----

<b>ANEXO UNO</b>	287
------------------	-----

## INTRODUCCIÓN

### 1. PLANTEAMIENTO

La industria de software se encuentra en una etapa de gran dinamismo tecnológico. Las tecnologías de software (aplicaciones, herramientas, entre otras) y sus usos cambian y evolucionan rápidamente creando entornos dinámicos, donde se propician procesos de generación de una importante cantidad de información y conocimiento, a los cuales las empresas pueden acceder. En dichos entornos se abren ‘ventanas de oportunidad’<sup>1</sup> formadas por la evolución de las tecnologías de software, que pueden ser aprovechadas por las empresas dependiendo de su habilidad para identificar, valorar, absorber, modificar y explotar información y conocimiento. En esos entornos tecnológicos y dinámicos algunas empresas se actualizan constantemente a través de una adecuada identificación y absorción de información y conocimiento, pero otras tienen limitaciones para ello debido a que no poseen mecanismos adecuados para identificar información y conocimiento útiles a sus necesidades e incorporarlos de manera coherente y creativa en sus procesos.

La información y conocimiento son los insumos principales para el diseño y desarrollo de software. En los inicios de la industria estos insumos eran suministrados principalmente por los usuarios. Durante la década de los 40 y 50, el diseño y desarrollo de software estuvieron asociados a proyectos militares y científicos financiados, principalmente, por los gobiernos de Estados Unidos e Inglaterra, siendo usuarios las agencias de gobierno, universidades y centros de investigación. A finales de los 50 se empieza a producir software comercial, pero los requerimientos para el diseño y desarrollo estaban poco sistematizados y estandarizados, de forma que el usuario participaba en los proyectos de software con los fabricantes de computadoras<sup>2</sup> proporcionando información acerca de sus requerimientos técnicos, por tanto, se convirtió en la principal fuente de información. En este sentido, el

---

<sup>1</sup> Las ‘ventanas de oportunidad’ son denominadas por Perez (2002) como oportunidades creadas por la evolución de la tecnología en un entorno de sucesivas revoluciones tecnológicas.

<sup>2</sup> Hasta finales de la década de los 60, el software era diseñado y desarrollado por empresas que manufacturaban las diferentes plataformas de hardware, i.e., computadoras.

software desarrollado era específico a cada tipo de computadora y usuario, por tanto, podría ser considerado software hecho a la medida (*customized software* en inglés).

Posteriormente, algunos usuarios desarrollaron su propio software, pero conforme los lenguajes de programación y las herramientas de software maduraban y se estandarizaban, en la década de los 60, un tipo de software podía funcionar en diferentes plataformas de hardware propiciando que fuera usado por diferente tipo de usuario. Esto originó que, a partir de la década siguiente, el software desarrollado por o para usuarios específicos (software ‘hecho a la medida’) fuera desarrollado en ‘paquete’ por empresas independientes. Con la introducción de la computadora personal (PC, por sus siglas en inglés) en la década de los 80 una gran cantidad de programas de software se estandarizaron y produjeron a gran escala, los requerimientos para el diseño y desarrollo de esos programas también se estandarizaron y disminuyó relativamente la participación del usuario. Esta evolución de la industria y la participación de los usuarios en la misma han sido estudiados por autores tales como Quintas (1995), Mowery (1996), Chandler y Cortada (2002).

Desde finales de los 90 las empresas usuarias de software con requerimientos específicos han tomado nuevamente una crucial importancia en el diseño y desarrollo de software (von Hippel, 1998 y 2000; Yarime y Baba, 2005). Una de las razones es que las aplicaciones de software estandarizadas no cubren los requerimientos específicos de usuarios cada vez más demandantes de productos ajustados a sus necesidades productivas y organizacionales, por tanto, el usuario interactúa con el productor en el diseño y desarrollo de aplicaciones.

Las empresas líderes como MICROSOFT, IBM, NOVELL, ORACLE, entre otras grandes empresas, estandarizaron y empaquetaron una gran cantidad de aplicaciones de software para mercados masivos, los usuarios (empresas e individuos) tuvieron que aceptar los productos aun sin estar completamente ajustados a sus necesidades. Pero también hay una gran cantidad de micro, pequeñas y medianas empresas desarrolladoras de software hecho a la medida para las empresas usuarias con requerimientos específicos. Las pequeñas y medianas empresas que desarrollan este tipo de software han sido parte importante en la

consolidación de la industria de países como la India, Irlanda e Israel (Delgado y Hernán, 2005, Shapiro, 2006). En México, predomina la micro, pequeña y mediana empresa de software, con una representación del 87%.<sup>3</sup> Si bien en México hay empresas que desarrollan o comercializan software empaquetado, el cual representa cerca del 80% del total de ventas en el país (aunque casi todo es importado), el segmento de software hecho a la medida representa poco más del 20%.<sup>4</sup>

La estructura de la industria mexicana de software se encuentra en una etapa relativamente joven, no obstante, para las empresas ubicadas en el segmento de software hecho a la medida hay una oportunidad de desarrollo económico y tecnológico, ya que, como han mostrado Mowery (1996), Chandler y Cortada (2002) y von Hippel (1998, 2000 y 2005), este segmento ha definido una parte importante de la evolución de la industria a nivel internacional. Al mismo tiempo, esta estructura expone a las empresas mexicanas de software a diferentes tipos de interacción con las empresas usuarias en el diseño y desarrollo de nuevos productos. Por esta razón, la presente investigación se centra en el estudio de las diferentes interacciones entre las empresas proveedoras y usuarias de software hecho a la medida en México.

Como se ha señalado en párrafos anteriores, desde el inicio de la industria de software, las interacciones entre proveedores y usuarios han sido de fundamental importancia para el desarrollo de aplicaciones de software, ya que sólo a través de ellas se solucionaban problemas particulares de los usuarios. Con la estandarización de las aplicaciones se generalizaron las soluciones a problemas asociados a procesos productivos y organizacionales, pero no cubrían las necesidades tecnológicas específicas de los usuarios, los cuales demandan productos para solucionar problemas individuales que cambian constantemente pues están cada vez más vinculados al cambio vertiginoso de las Tecnologías de Información y Comunicación (TIC). Para lograr cubrir ésta demanda, y ante el gran dinamismo de la industria, las empresas de software deben identificar, internalizar y

---

<sup>3</sup> Con base en estudios realizados por AMITI (2001) y González (2006) a empresas que desarrollan software.

<sup>4</sup> Estos porcentajes no consideran el software producido y al mismo tiempo consumido por instituciones de gobierno y empresas de manufactura (software de autoconsumo). En el capítulo 2 se explica la participación de cada segmento de la industria de software.

hacer un uso adecuado de la información y conocimiento asociado a esas necesidades para desarrollar aplicaciones confiables y de calidad. Esta evolución de la demanda crea una alta incertidumbre que puede reducirse a través de la interacción entre productores y usuarios de software.

Algunos autores como Lundvall (1985 y 1992), Andersen (1991 y 1996), Fagenberg (1993) y Cohen y Levithal (1990) argumentan que las interacciones entre diversos agentes del entorno son un mecanismo que mejora la capacidad de las empresas para identificar información y conocimiento generados en entornos dinámicos e incorporarlos en sus procesos. Las formas de interacción entre agentes han sido estudiadas usando diferentes perspectivas. Por ejemplo, bajo la perspectiva de Sistema Nacional de Innovación, Lundvall (1985 y 1992) utiliza el concepto de relación proveedor-usuario para analizar las características de los procesos de aprendizaje que se dan por la interacción e intercambio de experiencias entre agentes participantes en un proceso de innovación, aportando información y conocimiento desarrollados desde diferentes ángulos. Bajo la perspectiva de Paradigma Tecno-económico, Andersen (1991) define el concepto de interfase para referirse a la interacción entre dos agentes a través de la cual fluyen diferentes tipos de información utilizada en la generación de nuevo conocimiento. Otros autores como Cohen y Levithal (1990) y Van den Bosch, *et al.* (1999 y 2002) argumentan que las interacciones entre las áreas funcionales formalmente de la empresa, y entre la empresa y su entorno, son mecanismos importantes a través de los cuales la empresa puede internalizar información y conocimiento; las primeras interacciones, son denominadas por estos autores como interfases internas (*interface cross-function* en inglés), y las segundas son denominadas en esta tesis como interfases externas. En esta tesis la interfase se asocia a diferentes interacciones entre la empresa y diversos agentes.

En la literatura sobre interfase se destacan los flujos de información que se generan entre proveedor y usuario para el desarrollo de nuevos productos, y se ha centrado en los flujos de información externos para explicar el proceso de generación de nuevo conocimiento, sin embargo, ha puesto poca atención a la forma en cómo las empresas internalizan la información y conocimiento para lograr ese proceso.

Una de las principales dificultades que enfrenta la micro y pequeña empresa de software es su falta de capacidad para identificar y absorber información y conocimiento de acceso libre en el entorno, y que son generados principalmente a partir de las interacciones entre los diversos agentes. La industria a nivel internacional inicia con estos problemas y, como se mostrará en esta tesis, existen también en las empresas de software en México. Esa situación ha sido estudiada por varios autores, tales como Cohen y Levithal (1990), Van den Bosch, *et al.* (1999 y 2002), Lane, *et al.* (2002), Lenox y King (2003), y Lund Vinding (2004). En sus estudios han mostrado que a nivel organizacional es importante absorber información y conocimiento y, para ello, es necesario contar con procesos de socialización de conocimiento, de forma que éste no se pierda ni se olvide (Vera-Cruz, 2004). La absorción implica procesos de valoración, asimilación y uso adecuado de información y conocimiento para expresarlos en nuevos productos, procesos y servicios. Cohen y Levithal (1990) y Van den Bosch, *et al.* (1999 y 2002) muestran que el uso de conocimiento previo asociado a las actividades particulares de la empresa, la capacitación, las actividades de aprendizaje, así como las interfases internas son importantes para esos propósitos. Pero en un entorno de rápido cambio tecnológico, como es el caso de la industria de software, las empresas muestran dificultades para absorber adecuadamente la información y conocimiento pues no desarrollan sistemáticamente mecanismos para este fin. Es por eso que en esta tesis se hace énfasis en los mecanismos a través de los cuales las empresas de software incrementan su capacidad de absorción.

La preocupación de esta literatura se ha centrado, principalmente, en medianas y grandes empresas que cuentan con áreas funcionales formalmente, en particular con la de Investigación y Desarrollo (I+D), a partir de la cual la empresa puede identificar e internalizar información y conocimiento del entorno transfiriéndolos al resto de las áreas. Por lo cual, dicha literatura ha analizado con mayor profundidad los flujos internos y ha identificado patrones de comportamiento generales para estos tipos de empresas, sin embargo, ha profundizado poco en el estudio de los flujos externos en micro y pequeñas empresas carentes de áreas de I+D. Aunque esa literatura ha sido desarrollada a partir de otras realidades que no corresponden a la de las empresas mexicanas de software, no



obstante, aporta los elementos importantes para estudiar la realidad de la industria de software en México.

Con base en estos cuerpos de literatura se ha realizado trabajo empírico y se ha documentado, por un lado, las formas de interacción proveedor-usuario y, por otro, la forma en cómo las empresas absorben información y conocimiento. Con estos estudios se ha buscado demostrar la existencia de las interfases entre proveedores y usuarios, así como los incrementos en la capacidad de absorción. No obstante, en el segmento de software hecho a la medida, que se caracteriza por la existencia de una gran cantidad de micro y pequeñas empresas, no hay espacios para profundizar en los flujos internos, y el desarrollo de flujos de información depende en buena medida de la interacción con diversos agentes externos. Lo anterior es parte del problema para ese tipo de empresas, y no había sido parte de las prioridades de investigación de esos cuerpos de literatura. Por tal razón, a partir de esos cuerpos de literatura, esta tesis aporta empírica y teóricamente al estudio de las interfases externas como mecanismo de identificación de información y conocimiento existentes en el entorno (y generados a partir de las interfases), así como su importancia en los procesos de aprendizaje y en el incremento de la capacidad de absorción de las empresas ubicadas en el segmento de software hecho a la medida.

## **2. OBJETIVO Y PREGUNTAS DE INVESTIGACIÓN**

El objetivo de esta tesis es describir y analizar el proceso de construcción de interfases entre empresas mexicanas que desarrollan software hecho a la medida y diversos agentes de su entorno, tales como empresas, universidades, gobiernos y comunidades de desarrolladores de software de fuente abierta. Asimismo, busca explicar cómo a partir de diferentes interfases las empresas identifican información y conocimiento existentes en el entorno, y cómo al ser incorporados en las empresas influyen en su capacidad de absorción. Un nivel diferente de capacidad de absorción permite a las empresas hacer réplica de productos ya existentes o generar nuevo conocimiento expresado en el diseño y desarrollo de nuevos productos, procesos y servicios.

Las preguntas de investigación que guían esta tesis son las siguientes:

1. ¿Cómo y por qué se construyen interfases entre las empresas de software y los diversos agentes del entorno? ¿Qué tipos de interfase se construyen y cuáles son sus características?
2. ¿En qué medida las interfases permiten a las empresas identificar información y conocimiento e incrementar su capacidad de absorción?
3. ¿Cómo se expresa un incremento de la capacidad de absorción de las empresas? ¿Qué diferencias existen entre las empresas con distinta capacidad de absorción para identificar e internalizar información y conocimiento y materializarlos en aplicaciones específicas?

El argumento central de esta tesis es que en el segmento de software hecho a la medida hay múltiples interacciones entre los agentes que intercambian información y conocimiento, los cuales son fragmentarios pues cada agente posee una parte de éstos y son importantes para el diseño y desarrollo de software. A través de las interfases las empresas identifican partes importantes de información y conocimiento que no poseen. Por esta razón, la construcción de interfases constituye un mecanismo crucial de aprendizaje mediante el cual las empresas pueden incrementar su capacidad de absorción. Se parte del supuesto de que las empresas cuentan con un nivel básico de capacidad de absorción que les permite incrementarla a lo largo del tiempo. Si las empresas logran identificar e internalizar información y conocimiento de propósito general, sólo podrán replicar y, en algunos casos, hacer mejoras incrementales a productos ya existentes en el mercado; pero si logran identificar e internalizar información y conocimiento específicos, podrán diseñar y desarrollar nuevos productos, procesos y servicios.

Esta tesis contribuye empírica y teóricamente al estudio de las interfases, y muestra su importancia en los procesos de aprendizaje y en el incremento de la capacidad de absorción de las empresas ubicadas en entornos de rápido cambio tecnológico. Una de las aportaciones de esta tesis consiste en mostrar que las interfases externas permiten identificar y articular información y conocimiento fragmentarios que cada agente posee y, por tanto, son un mecanismo importante de aprendizaje a partir del cual las empresas de software incrementan su capacidad de absorción y generan nuevo conocimiento. Esta causalidad de efectos estará presente a lo largo de esta investigación.

El estudio se centra en empresas mexicanas que desarrollan software hecho a la medida. La estrategia de investigación utilizada es el estudio de caso múltiple-exploratorio. Los casos son empresas de software de fuente abierta y empresas de software propietario, en donde la unidad de análisis es la interfase. Los resultados se sustentan en entrevistas a profundidad realizadas a líderes de proyecto y desarrolladores de software ubicados en 6 empresas, tres de software de fuente abierta y tres de software propietario. El trabajo de campo se realizó entre octubre de 2004 y febrero de 2006, en las ciudades de México, Guadalajara, León y Monterrey, donde se encuentran las empresas consideradas. Previo a éste trabajo se realizó un estudio exploratorio entre agosto de 2003 y hasta finales de 2004, a través del cual se identificó y seleccionó la muestra.

### **3. ESTRUCTURA DE LA INVESTIGACIÓN**

Después de esta introducción, la tesis se estructura en tres partes comprendidas en 8 capítulos. La primera parte titulada “Evolución tecnológica de la industria de software a nivel internacional y su caracterización en México”, se compone de los capítulos 1 y 2, en los cuales se presentan los aspectos fundamentales que permiten entender la evolución de la industria, la relevancia de la interacción entre empresas y usuarios poseedores de información y conocimiento fragmentarios, y la problemática de la racionalización del proceso de desarrollo de software. En el capítulo 2, específicamente, se describe la estructura industrial y tecnológica del software en México. Se presentan datos acerca de la demanda y oferta de recursos humanos en la industria, acerca de su capacidad de producción, así como las estrategias sectoriales y regionales planteadas en algunos estados del país.

La segunda parte titulada “Interfase, capacidad de absorción y generación de conocimiento: revisión de la literatura”, se compone de los capítulos 3, 4 y 5. En los dos primeros, 3 y 4, se hace una revisión de los cuerpos de literatura sobre los que se sustenta esta tesis, los cuales hacen referencia a la creación de interfases, la capacidad de absorción y la generación de conocimiento. A partir de esos cuerpos de literatura se estructura el marco

conceptual de la investigación. Dentro del capítulo 5 se describe el diseño y la estrategia de investigación, al final de este capítulo se describen los *casos* (empresas estudiadas).

La tercera parte titulada “Diseño de interfases, capacidad de absorción y generación de conocimiento: evidencia empírica”, se compone de los capítulos 6, 7 y 8. En el capítulo 6 se presenta y analiza la evidencia empírica sobre cómo las empresas que desarrollan software hecho a la medida crean diferentes tipos de interfase con diferentes agentes en entornos tecnológicos complejos y estables. En el capítulo 7 se presenta y analiza la evidencia empírica sobre cómo las empresas de software absorben información y conocimiento existentes en el entorno y captados a partir de las interfases. En el capítulo 8 se discute cómo, a partir de las diferentes interfases creadas y del incremento en la capacidad de absorción, las empresas de software generan nuevo conocimiento expresado en nuevos productos, servicios y procesos a nivel de empresa e industria.

Finalmente, se presentan las conclusiones generales y las aportaciones teóricas y empíricas derivadas de esta investigación, así como algunas sugerencias para futuras investigaciones y recomendaciones para el diseño de políticas orientadas a fortalecer la creación de interfases que permitan un mayor incremento de la capacidad de absorción de las empresas de software.

**PARTE I**

**EVOLUCIÓN TECNOLÓGICA DE LA INDUSTRIA DE SOFTWARE  
A NIVEL INTERNACIONAL Y SU CARACTERIZACIÓN EN  
MÉXICO**

Esta tesis tiene por objeto el estudio de los procesos de construcción de interfases entre empresas desarrolladoras de software hecho a la medida y los diversos agentes del entorno. Se argumenta que las interfases permiten identificar y articular información y conocimiento fragmentarios que existen en el entorno, al mismo tiempo influyen en el incremento de la capacidad de absorción de las empresas. En la Parte I se presentan los antecedentes de la investigación, atendiendo en el capítulo 1 cuatro aspectos fundamentales. Primero, la independencia de la industria de software respecto a la industria de hardware. Segundo, el papel de los usuarios en los procesos de innovación en la industria de software, en particular en el segmento de software hecho a la medida. Tercero, las diferentes interacciones entre múltiples agentes que intercambian información y conocimiento fragmentarios para el desarrollo de software hecho a la medida. Se dice fragmentarios ya que cada agente es poseedor de una parte importante de la información y conocimiento necesarios para el desarrollo de un producto o sistema de software. Cuarto, la racionalización de los procesos de software y su problemática en los procesos de internalización de información y conocimiento.

En el capítulo 2 se expone la estructura industrial y tecnológica del software en México, además de los datos acerca de la demanda y oferta de recursos humanos en la industria, acerca de su capacidad de producción, así como las estrategias sectoriales y regionales planteadas en algunos estados del país. Al mismo tiempo, se plantean los principales problemas que limitan el desarrollo y crecimiento de la industria.

# CAPÍTULO UNO

## EVOLUCIÓN DE LA INDUSTRIA DEL SOFTWARE

### INTRODUCCIÓN

En el presente capítulo se describen los elementos más importantes en la evolución de la industria de software a través de cinco apartados. Dentro del **primero**, se describe el concepto de la tecnología de software, para que, en el **segundo**, se muestren la tipología y las actividades centrales en el desarrollo de software. Así, en el **tercero**, se presenta la evolución de las tecnologías de hardware (computadoras) y su relación con la tecnología de software, así como el rol que han jugado los diferentes agentes involucrados en su desarrollo, es decir, la importancia que tuvieron los productores de computadoras, los usuarios y los fabricantes independientes de software. En el **cuarto** apartado se describen dos enfoques: i) Sistemas de Productos Complejos (SIPC), y ii) Software Incorporado (SWI) para mostrar las múltiples interacciones entre agentes en el desarrollo de software. En el **quinto**, se hace referencia a los problemas que emergen al tratar de ‘*racionalizar*’ el proceso de desarrollo de software, lo cuales no han sido superados plenamente a pesar de los esfuerzos por establecer al interior de las empresas modelos de calidad como CMM<sup>1</sup> o MoPROSOFT<sup>2</sup>. Por último, se presentan las conclusiones del capítulo, en donde el argumento central **explica** que en la industria de software hay múltiples interacciones entre los agentes que intercambian información y conocimiento (I+C). La I+C son fragmentarios ya que cada agente posee una parte importante de éstos para el desarrollo de un producto o sistema de software.

### 1.1 DEFINICIÓN

Las computadoras se forman de dos elementos básicos, el hardware y el software. El primero es la parte visible y tangible de una computadora. El segundo está formado por diferentes instrucciones que controlan el hardware. Desde el punto de vista de la ingeniería de software, éste consiste en secuencias de notación que, al instalarse en componentes de

---

<sup>1</sup> Modelo de Maduración de Capacidades, por sus siglas en inglés, CMM, Capability Maturity Model.

<sup>2</sup> Modelo de Procesos para la Industria de Software, norma mexicana vigente a partir de 2005.

hardware, crea un sistema funcional con el cual podemos interactuar o comunicar. En otras palabras, el software se refiere a los programas o instrucciones que controlan y operan computadoras y sistemas, los cuales proveen las ‘aplicaciones’ o funciones que corren sobre un sistema (Quintas, 1995:3).

Quintas (1995:6) argumenta que el software difiere de las tecnologías convencionales en varios sentidos. Primero, el software tiene la característica económica de la información más que de los artefactos físicos, y las leyes de derechos de propiedad intelectual lo consideran semejante a los trabajos literarios. Segundo, el software es más que símbolos o códigos pues cuando se instala en un hardware, se convierte en parte integral de un sistema y representa procesos que controlan directamente las funciones físicas.

Las diferencias conceptuales del software, y en general de la industria de Tecnologías de Información y Comunicación (TIC<sup>3</sup>), tienen connotaciones de carácter técnico y sociológico. Técnicamente, la definición más simple de TIC se puede reducir a *algoritmo*. Según Eischen (2002a), en esencia todas las TIC definen una función lógico-algebraica y binaria que produce resultados consistentes para procesos específicos. Los algoritmos existen en los procesos naturales y sociales, crean patrones que forman un puente entre la tecnología y la sociedad. El software es fundamentalmente un ejercicio de traslado de algoritmos existentes en la naturaleza o en las prácticas sociales hacia formas digitales.

Desde esta perspectiva tecnológica, se señala que la raíz del software, al igual que de las TIC, es el *código*. Todos los códigos tienen reglas de sintaxis y semántica, pero hay tantos lenguajes de software como múltiples formas de programar. Una vez que el código toma forma, el programa es compilado, lo que significa traducir cualquier lenguaje de programación en ‘códigos máquina’ -o en la forma binaria compuesta de 0 y 1- que las computadoras pueden entender. Solamente en este punto se puede decir que es software y no grandes códigos, es un *programa* disponible para ser replicado y distribuido como un producto completo (Eischen, 2002a).

---

<sup>3</sup> Las TIC la conforman las industrias de microelectrónica, computación, informática [software] y telecomunicaciones (OECD, 1988).

Desde el punto de vista económico y social, las TIC están conformadas por procesos que requieren capital, trabajo calificado y conocimiento. En especial, el software tiene como insumo principal el conocimiento que está incorporado en individuos, empresas y regiones, es un proceso único que traslada ese conocimiento a una estructura social. Gran parte de este conocimiento es tácito, indefinido y no siempre está explícito entre los agentes involucrados en ese proceso. El conocimiento es dinámico, constantemente en evolución y transformación (Eischen, 2002a).

Desde el punto de vista de la sociología, el software es un elemento que transforma las formas de producción y gobernanza, es una actividad única que combina creatividad, habilidades y métodos disciplinarios, traduciendo la I+C del entorno en formas binarias o algoritmos. Aunque algunos procesos de desarrollo de software son automatizados, están basados en las capacidades humanas. De tal forma que el proceso de desarrollo está estructurado en aspectos de automatización e ingeniería, pero es fundamentalmente una “labor intensiva en capacidades intelectuales y una actividad costosa, en la cual una buena administración y comunicación es mucho más valiosa que la tecnología” (Eischen, 2002a).

Según este autor, el software se mueve de altos a bajos niveles de abstracción. Cuando se traducen los procesos sociales a algoritmos, gran parte del conocimiento dominante es tácito, indefinido, no codificado, desarrollado a lo largo de un periodo de tiempo y a menudo no explícito a los individuos implicados en el proceso social. El conocimiento social y las prácticas son dinámicas, en constante evolución y transformación.

## **1.2 TIPOLOGÍA Y ACTIVIDADES DE SOFTWARE**

Aunque no hay una tipología única del software, éste se puede clasificar en tres tipos:

- a) *Software de sistema*: software básico que controla las operaciones de una computadora y la hace funcionar, incluye sistemas operativos, controladores de dispositivo, entre otros. Este software permite la interacción entre todas las partes del hardware y con cualquier equipo conectado a ella, desde dispositivos de almacenamiento de información hasta redes de telecomunicaciones.



- b) *Software de programación (o herramientas de aplicación)*: proporciona herramientas para ayudar al programador a escribir programas informáticos y usar diferentes lenguajes de programación<sup>4</sup> de forma práctica. Incluye editores de texto, compiladores, depuradores, entre otros.
- c) *Software de aplicación*: son un conjunto de instrucciones de software que permiten a la computadora desempeñar tareas específicas de interés para el usuario final, tales como procesamiento de palabras y de datos, contabilidad de negocios, etcétera. Estas soluciones son también denominadas programas de software o aplicaciones de software. Cabe mencionar que se considera aplicaciones de software horizontales a los procesadores de palabras y datos; aplicaciones de software verticales al software para bancos, transporte, etc., y aplicaciones para usuarios específicos a los programas de software hecho a la medida.

Desde la década de los 50, el *software de aplicación* se ha comercializado y diversificado de acuerdo a la evolución de las plataformas de hardware y a las necesidades de los usuarios. De manera general, se identifican tres variedades de software de aplicación:

- El *software empaquetado* se refiere a una aplicación o programa que es comercializado a través de distribuidores. Los programas están dirigidos a grupos de empresas o sectores (usuarios en masa), ofreciendo una solución común a un gran número de usuarios. Incluye los programas comercialmente disponibles para venta o arrendamiento. Este tipo de software es instalado por el usuario y está listo para ser usado, por ejemplo, los procesadores de palabras como WORD o WORKS, o procesadores de datos como EXCEL, entre muchos otros.
- El *software hecho a la medida* se refiere a una aplicación o programa hecho con base en las necesidades y requerimientos del usuario, como una solución específica para un usuario o pequeño grupo de usuarios.
- El *software incorporado* (embedded software, por sus siglas en inglés) es una combinación de hardware y software de sistemas para lograr una aplicación que otorga funcionalidad a un dispositivo que no cuenta con una computadora tradicional. Esta

---

<sup>4</sup> En Estados Unidos se han inventado más de 1000 lenguajes de programación desde la década de los 50, entre los más utilizados actualmente están HTML, JAVA, C++, PERL.

variedad de software se incorpora en productos de manufactura tales como máquinas-herramienta, bienes electrónicos de consumo, como PDA<sup>5</sup> y teléfonos móviles, aparatos médicos y domésticos, componentes para automóvil, entre otros.

La empresa de software tiende a jerarquizar las actividades de desarrollo de software, las cuales están basadas en diferentes actividades de trabajo. En el siguiente cuadro se muestra la estructura de roles y actividades más importantes en una empresa de software.

**Cuadro 1.1**  
**Actividades y roles en la empresa de software**

Roles	Actividades
Gerente Administrativo de proyecto	Administra el desarrollo del proyecto
Arquitecto de software	Análisis y diseño integral de todos los módulos del sistema
Líder técnico de proyecto	Análisis y diseño particular de algunos módulos del proyecto, coordinación de desarrolladores
Desarrolladores	Programación, pruebas unitarias de pequeños componentes del software
Testers	Pruebas de integración de módulo que componen el sistema completo
Personal de operación	- - -
Personal de soporte	- - -

Fuente: Con base en UAM, 2004; y entrevistas a gerentes y líderes de proyecto.

Algunas actividades son intensivas en trabajo, por ejemplo, el desarrollo de software o programación y las pruebas de software (testing). Las empresas que no cuentan con áreas funcionales formalmente, sobre todo la micro y pequeña empresa, generalmente trabajan en un esquema de actividades multitareas, de forma que la organización del trabajo se basa en un esquema multitareas y no en áreas estructuradas.

Entre las actividades centrales está el proceso de desarrollo de software, el cual intensivo en trabajo y requiere de habilidades especializadas, entrenamiento en lenguajes (HTML, JAVA, C++, PERL, entre otros) y conceptos de desarrollo de software (arquitectura de software, estructura matemática, entre otros). Además de las habilidades en programación, son importantes las de comunicación (que corresponden a nuevas demandas de comunicación en todos los niveles del proceso de desarrollo), diseño y calidad, aspectos del proceso que

<sup>5</sup> Por sus siglas en inglés: Personal Digital Assistants.

requieren toma de decisiones y conocimiento tácito logrado a través del *aprender haciendo* (learning by-doing).

### **1.3 EL SOFTWARE COMO FACTOR DE DISCONTINUIDAD EN LA INDUSTRIA DE HARDWARE**

A partir del marco conceptual de paradigma tecno-económico (PTE) presentado por Freeman y Perez (1988) y Perez (1986, 2001, 2002, 2003), es posible conocer el inicio, madurez y declinación de cada una de las revoluciones tecnológicas logradas a lo largo de los dos últimos siglos.<sup>6</sup> Cada revolución tiene diferentes vectores de insumos, infraestructura y agentes involucrados en su difusión. En particular, la última revolución tecnológica aún en desarrollo, la de TIC, revela diferentes combinaciones de esos vectores y diferentes estados de desarrollo para cada uno de los países anclados a ella.

Según estos autores, la difusión de cada revolución tecnológica es guiada por un PTE que afecta la estructura y las condiciones de producción y distribución en casi todas las ramas de la economía. El lanzamiento del microprocesador INTEL<sup>7</sup>, a principios de la década de los 70 marcó un nuevo periodo de transición en el que se articuló un nuevo PTE integrando progresivamente las TIC. El nuevo PTE surge de las restricciones que impone el modo de producción y distribución predominante sobre el crecimiento de la productividad, rentabilidad y los mercados, adoptando nuevas nociones de eficiencia, reglas y hábitos por parte de ingenieros, diseñadores, empresarios y gerentes (Freeman, 1988).

Las TIC, definidas como la combinación de innovaciones radicales basadas en la computación, la informática [software], la microelectrónica y las telecomunicaciones (OECD, 1998) y como los medios para adquirir, almacenar, procesar y transmitir la información (incluyendo software y hardware) (Steinmueller, 1996:16), son el PTE más reciente y difundido. Han aparecido otros paradigmas como producto del incremento del conocimiento científico y tecnológico tales como la biotecnología, nuevos materiales, la

---

<sup>6</sup> Este marco conceptual se presenta en el apartado 3.1 del capítulo 3 de esta tesis.

<sup>7</sup> En noviembre de 1971, en Santa Clara, California, Bob Noyce y Gordon Moore dieron a conocer el primer procesador Intel. Evento considerado como uno de los más significativos en el área de la computación y telecomunicación digital.

tecnología espacial y nuclear, pero se encuentran en un estado de menor desarrollo comparado con las TIC (OECD, 1998).

El software, como un bien intangible asociado directamente al conocimiento, es un aspecto trascendente en la era de la información, la forma en cómo llegó a existir y cómo se desplegó en toda la economía representa la primera discontinuidad relevante dentro de las TIC (Chandler y Cortada, 2002:363).

### 1.3.1 Evolución de la industria de computadoras, 1940-1990

El desarrollo de la tecnología de software está ampliamente relacionado con la evolución de las tecnologías de hardware. Como se muestra en el cuadro 1.2 las plataformas de hardware han marcado una trayectoria tecnológica en la industria de computadoras.

**Cuadro 1.2**  
**Características evolutivas de las tecnologías de hardware, 1940-1990**

<b>Tecnologías de hardware (plataformas)</b>	<b>Características</b>
Computadora central (mainframes)	<ul style="list-style-type: none"> <li>- Primeros desarrollos en la década de los 40</li> <li>- Uso científico y militar</li> <li>- I+D financiado por el gobierno</li> <li>- IBM principal productor en los años 50</li> <li>- Uso industrial en los 50</li> <li>- En 1964 IBM introduce computadoras compatibles modulares, la IBM/360</li> </ul>
Mini-computadora	<ul style="list-style-type: none"> <li>- En 1965 la empresa DEC introduce la mini-computadora para uso industrial</li> <li>- Uso de esta plataforma en pequeñas empresas</li> <li>- En los 70 esta plataforma tuvo un crecimiento intensivo (mayor uso dentro de las empresas) y extensivo (incremento de nuevas empresas usuarias)</li> </ul>
Micro-computadora	<ul style="list-style-type: none"> <li>- En los 70 se introduce el micro- procesador que da origen a la computadora personal (PC)</li> <li>- Esta plataforma incide en el desarrollo de sw de empresas independientes de las de hw</li> <li>- En los 80, la PC acelera las ventas de sw</li> <li>- En 1993 el sw para micro-computadoras representó el 75% respecto del total de ventas de sw</li> </ul>
Estaciones de trabajo (work stations) (década de los 80)	<ul style="list-style-type: none"> <li>- En los 80 las empresas Apolo y Sun Microsystems introducen las estaciones de trabajo</li> </ul>
PDA (finale de los años 90)	<ul style="list-style-type: none"> <li>- Personal digital assistant, PDA</li> <li>- Personal mobile assistant (embedded software)</li> </ul>

sw: Software. hw: Hardware.

Fuente: Elaboración propia, con base en varios estudios (Borja, 1995; Malerba y Orsenigo, 1996; Mowery, 1996; Langlois y Mowery, 1996; Chandler y Cortada, 2002).

La industria de computadoras tiene sus antecedentes en la década de los años 40 con la introducción de las primeras computadoras centrales (mainframes), fabricadas, primero con bulbos y luego con transistores, para ser utilizadas en proyectos científicos y militares. Uno de los principales agentes en la industria de estas computadoras fue la empresa International Business Machines (IBM), quien desarrolló productos importantes en la década de los 50 dando oficialmente origen a la industria de la computadora comercial,<sup>8</sup> dominando la venta de computadoras centrales y las actividades de procesamiento de datos de sus clientes.

IBM introduce en 1964 la IBM S/360, con la cual la estandarización de componentes y software permitió tanto la explotación de economías de escala en la producción de componentes como también la consecuente integración vertical de empresas fabricantes de computadoras (Malerba y Orsenigo, 1996; Borja, 1995). Ésto permitía ofrecer un grupo de máquinas de diferentes tamaños, en las cuales el software que funcionaba en una máquina lo podía hacer en otra.

En 1965, la empresa Digital Equipment Corporation (DEC) introduce una innovación radical, la mini-computadora PDP-8, interactiva en tiempo real, originando una nueva trayectoria tecnológica e industrial. Esta computadora tuvo un impacto a pequeña escala alterando la distribución de las actividades de desarrollo de software y el uso de computadoras dentro de las organizaciones. También hizo posible que las pequeñas empresas empezaran a usar sus propias computadoras y dejar el esquema de renta de las computadoras centrales (mainframes).

La rápida difusión de la tecnología de computación, principalmente a través de las estaciones de trabajo y las microcomputadoras, creó grandes oportunidades para generar aplicaciones de solución particulares. La base instalada de micro-computadoras excedió rápidamente la base instalada de computadoras centrales y mini-computadoras, siendo uno

---

<sup>8</sup> Las primeras computadoras ENIAC (Electronic Numerical Integrator and Computer) y SEAC (Standards Electronic Automatic Computer) creadas en 1946 y 1950, respectivamente, tuvieron aplicaciones militar y científica. Después es creada la IAS por von Neuman y el Institute of Advanced Study en 1951, UNIVAC (Universal Automatic Computer) y la IBM 701 fueron creadas en 1953 con aplicaciones más comerciales. (Véase Langlois y Mowery, 1996).

de los principales factores que incidieron en el desarrollo de software independiente, incentivando además al mercado de aplicaciones estándar para microcomputadoras.

Por su parte, las estaciones de trabajo tuvieron su capacidad gráfica como uno de los mejores atractivos. Las aplicaciones intensivas en gráficos implicaron el ‘diseño asistido por computadora’ (CAD, por sus siglas en inglés: computer-aided design) y la ‘ingeniería asistida por computadora’ (CAE, por sus siglas en inglés: computer-aided engineering), que habían sido desarrolladas para mini-computadoras, pero sufrieron de limitaciones en la capacidad de despliegue de gráficos y de la fragmentación en el mercado de software para mini-computadoras.

La diversidad de plataformas de hardware representó un paso importante en la ‘incorporación’ de procesos, permitió la inter-cambiabilidad, conectividad e interoperabilidad entre hardware y software. En los años 90 la industria de computadoras se convirtió en una industria bastante articulada y compleja, lejos de la estructura concentrada mostrada en las décadas anteriores. El desarrollo de nuevas computadoras requirió la integración de arquitecturas complejas, sistemas de software avanzado e integración de componentes a gran escala (Malerba y Orsenigo, 1996).

### **1.3.2 Del software empaquetado al software de fuente abierta: seis etapas evolutivas**

Paralelo al desarrollo de la industria de computadoras inició el de la industria del software. Chandler y Cortada (2002) plantearon 4 etapas de desarrollo de la industria de software, sin embargo, con base en otros estudios realizados por Quintas (1995), Malerba y Orsenigo (1996), Steinmueller (1996), Eischen (2002a y 2002b), Lerner y Tirole (2002), Von Hippel (2005), entre otros, se pueden identificar seis etapas evolutivas de esta industria, tal como se muestra en el cuadro 1.3.

***Primera etapa: gestación de la industria.*** Durante la década de los 40 y 50 se descubrieron la mayoría de los conceptos básicos usados en el diseño de software en la época de 1960 a

1980.<sup>9</sup> El software era desarrollado y dominado por los fabricantes de hardware, quienes incentivados por el complemento económico que representaba 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.

**Cuadro 1.3**  
**Características evolutivas de las tecnologías de software, 1940-1990**

Etapa	Características
1ª Gestación 1940-1950	<ul style="list-style-type: none"> <li>- En las décadas de los 40 y 50 el sw es desarrollado por los productores de hw</li> <li>- El sw es específico para cada usuario, es artesanal con aplicaciones científicas y militares</li> <li>- El uso industrial de hw estimuló el desarrollo de sw</li> <li>- Los usuarios cooperaban en el desarrollo de mejores aplicaciones</li> </ul>
2ª Nacimiento 1960	<ul style="list-style-type: none"> <li>- Con la IBM/360 introducida en 1964, el sw (sistema operativo único) se hace compatible en diferentes tamaño de computadora</li> <li>- El sw es desarrollado por empresas independientes de la producción de hw</li> <li>- En 1969 IBM separa los precios de software del hardware</li> </ul>
3ª Crecimiento 1970	<ul style="list-style-type: none"> <li>- A principios de la década de los 70 había en Estados Unidos entre 1500 y 2800 empresas</li> <li>- El sw empacado se hace masivo para las PC</li> <li>- La PC acelera el crecimiento de las ventas de sw</li> <li>- En 1971 IBM introduce el hard disk que permite almacenar grandes cantidades de información, mejora la arquitectura del hw y diversifica las actividades de sw</li> <li>- Problemas de calidad, administración y medición</li> </ul>
4ª Consolidación 1980	<ul style="list-style-type: none"> <li>- Se consolidan las empresas independientes de sw: MICROSOFT, LOTUS, WORDPERFECT, ASHTON-TATE, BORLAND, entre otras.</li> <li>- Bajas barreras a la entrada</li> <li>- Diseño dominante: PC</li> <li>- Homogenización de sistemas operativos y aplicaciones para PC</li> </ul>
5ª Uso de 'redes' Finales de los 80	<ul style="list-style-type: none"> <li>- En la década de los 80 surge Intranet e Internet basados en la plataforma PC</li> </ul>
6ª Sistemas abiertos 1990-?	<ul style="list-style-type: none"> <li>- En la década de los 90 inicia el desarrollo de sw de fuente abierta en comunidades virtuales de desarrolladores</li> <li>- Códigos abiertos</li> </ul>

sw: Software. hw: Hardware.

Fuente: Elaboración propia, con base en varios estudios (Quintas, 1995; Malerba y Orsenigo, 1996; Steinmueller, 1996; Mowery, 1996; Eischen, 2002a y 2002b; Lerner y Tirole, 2002; Chandler y Cortada, 2002; Von Hippel, 2005).

El software era específico a cada tipo de máquina y a cada usuario, de forma que el escalamiento de una computadora hacia otra con mejor desempeño exigía reescribir todo el software. Pero a finales de la década de los 50 los usuarios de computadoras empezaron a desarrollar aplicaciones para su propio uso, cooperaron con los fabricantes de

<sup>9</sup> En 1957 se introduce el lenguaje FORTRAN (Formula Traslator) y en 1960 se introduce COBOL (Common Business-Oriented Language), dos de los primeros y más importantes lenguajes desarrollados.

computadoras en el intercambio de rutinas de programación y métodos de organización con el fin de mejorar las actividades de desarrollo de software.

***Segunda etapa: el nacimiento de una industria.*** En 1964 IBM introduce la familia de computadoras S/360, que unifica el sistema operativo de la línea de productos de esta empresa. 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. Lo anterior origina una rápida estandarización tecnológica tanto del hardware como del software, en consecuencia, el software funcionaba en más de una máquina.<sup>10</sup>

La disponibilidad de sistemas compatibles fue un detonante para la creación de la industria del software, la cual se componía de empresas creadas por antiguos programadores de las empresas de computadoras que tenían una experiencia previa en el desarrollo de software, o de proveedores de hardware que se habían independizado y especializado en el procesamiento de datos. Entonces, se pasó del software desarrollado por los usuarios al software desarrollado por empresas independientes.

***Tercera etapa: crecimiento de la industria.*** Con la existencia de empresas de software independientes, la industria experimentó una extensiva división del trabajo y una fuerte especialización. IBM introdujo el modelo de computadoras S/370 en 1971 con una nueva tecnología llamada *hard disk* (disco duro), lo que hizo posible para los usuarios, por primera vez en la historia de la industria, almacenar centralizadamente un excedente de información. Esto mejoró el tiempo de respuesta de los sistemas (arquitectura), aumentando significativamente su desempeño y permitiendo el crecimiento de empresas de servicios como nueva fuente de las actividades de desarrollo de software.

---

<sup>10</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.



En la década de los 70 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 mini-computadoras. 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.

***Cuarta etapa: consolidación de la industria.*** En la década de los 80 la industria de software parecía llegar a su madurez, surgieron empresas independientes que rápidamente se consolidaron en el mercado con diseños dominantes usando una nueva plataforma de hardware.<sup>11</sup> La PC IBM, introducida al 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, según Mowery (1996), se logró: (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.

Las nuevas empresas de software pudieron, entonces, competir en el mercado de aplicaciones de software sin la necesidad de producir hardware; la dimensión clave para la innovación fue el desarrollo de habilidades para administrar complementariedades de tecnologías relacionadas o vinculadas, de usuarios y aplicaciones de software. Las aplicaciones y servicios de software, así como el hecho a la medida requirieron una base de conocimiento diferente para la innovación.

---

<sup>11</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.

Las empresas debían entender las necesidades del mercado, identificar las dimensiones relevantes de las aplicaciones específicas del mismo, focalizar sus productos a esas dimensiones e interactuar continuamente con los usuarios. De forma que, dada la complejidad de la base de conocimiento y la heterogeneidad de las competencias requeridas para la actividad innovativa, la cooperación entre empresas y usuarios, así como la estrategia de redes, llegaron a ser cruciales ya que se habían creado experiencias y especialización respecto al uso del hardware o la base operativa de software (Malerba y Orsenigo, 1996).

**Quinta etapa: las ‘redes’ entran en juego.** 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*; más tarde, a mediados de esa década, las empresas utilizaban Internet<sup>12</sup> para enlazar las redes internas (Intranet) y las redes privadas entre sí. El software para lograr éso provenía de proyectos patrocinados por el gobierno, como APARNET, o de proyectos independientes de software como Novell, Oracle y Lotus. En este sentido, el desarrollo de software en los 90 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.

**Sexta etapa: el software de fuente abierta (open-source) como nuevo paradigma.** El software de fuente abierta (SWFA) tiene una característica que lo diferencia del software propietario: la difusión del código fuente<sup>13</sup>, misma que transforma los procesos de desarrollo del software. Mientras que el software propietario se desarrolla y diseña en espacios y tiempos delimitados, el SWFA es desarrollado especialmente por comunidades de desarrolladores que trabajan interconectados a través de una red (Internet) y correo electrónico. Como se muestra en el cuadro 1.4, su diseño y desarrollo se centra en las

---

<sup>12</sup> Internet se creó a partir del desarrollo de APARNET por parte del gobierno estadounidense, en la década de los años 60 a los 80.

<sup>13</sup> En términos simples, el código fuente es el libro de instrucciones informáticos donde reside el secreto de programación.

comunidades de desarrolladores conectados a través de Internet, de cualquier parte del mundo, a sitios WEB específicos.<sup>14</sup>

**Cuadro 1.4**  
**Configuración social y tecnológica del software de fuente abierta, 1960-2000**

Etapa	Características
Antelación: 1960-1970	<ul style="list-style-type: none"> <li>- Desarrollado de Internet en la Universidad de Berkeley y MIT, en los centros de I+D de Bell Laboratories y de Palo Alto de Xerox</li> <li>- En los 70 se desarrolla el sistema operativo UNIX basado en estándares abiertos</li> <li>- Los programadores escribían código fuente usando lenguajes como BASIC, C, JAVA, entre los más importantes</li> </ul>
Nacimiento: 1980	<ul style="list-style-type: none"> <li>- Uso intensivo de UNIX en las estaciones de trabajo</li> <li>- Richard Stallman inicia la 'Fundación del Software Libre' en 1983</li> <li>- La 'Fundación' desarrolla y disemina una gran variedad de sw sin costo</li> <li>- Se crean licencias que impiden la imposición de derechos de patente respecto al desarrollo cooperativo de sw, como la GPL (General Public Licence)</li> </ul>
Expansión: 1990-	<ul style="list-style-type: none"> <li>- En 1991 Linus Torvalds desarrolla LINUX, un sistema operativo basado en UNIX</li> <li>- Se crean comunidades de desarrolladores con usuarios 'conscientes', tales como APACHE, DEBIAN, etc.</li> <li>- Bajo impacto en el desarrollo de aplicaciones de 'escritorio'</li> </ul>

Fuente: Elaboración propia con base en varios estudios (Quintas, 1995; Malerba y Orsenigo, 1996; Steinmueller, 1996; Mowery, 1996; Eischen, 2002a y 2002b; Lerner y Tirole, 2002; Chandler y Cortada, 2002; Von Hippel, 2005).

Dos eventos importantes que antecedieron al nacimiento del SWFA fueron el desarrollo del sistema operativo UNIX, basado en estándares abiertos, y después la creación de Internet. El uso intensivo de UNIX en las estaciones de trabajo y la 'Fundación del Software Libre' a principios de los años 80, comandada por Richard Stallman<sup>15</sup>, dieron origen al movimiento de SWFA. Una importante innovación introducida por la 'Fundación' fue el proceso de licenciamiento formal, que impide la imposición de derechos de patente respecto al desarrollo cooperativo de software (como se cree que AT&T hizo en el caso de UNIX). Como parte de la licencia GPL<sup>16</sup> (General Public License) conocida como 'copylefting', el usuario podía mejorar un programa sin imponer restricciones de licencia a otros usuarios.<sup>17</sup>

<sup>14</sup> La corriente de software libre defiende que los programas deben ser libremente creados, modificados y distribuidos.

<sup>15</sup> Richard Stallman fue programador en el Massachusetts Institute Technology (MIT), EU, y ante un problema técnico de impresión, cuyo código no pudo modificar porque el software era 'cerrado' (propietario), decidió abanderar el movimiento de software libre, en un principio al interior del MIT.

<sup>16</sup> Es un conjunto específico de términos de distribución para proteger con *copyleft* a un programa. Un software protegido con *copyleft* es software libre cuyos términos de distribución no permiten a los distribuidores agregar alguna restricción adicional cuando éstos redistribuyen o modifican el software. Lo

La amplia difusión del Internet a principios de los 90 aceleró el movimiento de SWFA. En este periodo nacen nuevos proyectos de fuente abierta como LINUX, cuya difusión se acelera a través de las redes creadas en Internet. También surge una nueva relación entre las empresas comerciales y la comunidad de SWFA; las primeras empiezan a promover el desarrollo de este tipo de software a través de proyectos de inversión, de tal forma que algunas empresas como IBM empiezan a desarrollar programas basados en SWFA.

Existen millones de programadores y usuarios ‘conscientes’ en todo el mundo, quienes contribuyen al movimiento de SWFA, que crean y modifican ese tipo de software, administradores de sistemas computacionales, administradores de bases, programadores y otros técnicos e ingenieros de la computación. El Internet, prácticamente, ha sido un facilitador de comunicación y descarga de programas entre estos programadores.

El precio, flexibilidad e independencia son tres de las ventajas más visibles de este tipo de software. Primero, aunque el precio de la implantación de software depende de la complejidad del proyecto, no se paga por licencias. Segundo, este software es flexible ya que se diseña de acuerdo a las necesidades y requerimientos del usuario, gran parte del SWFA que se desarrolla es ‘hecho a la medida’. Tercero, no hay un organismo único que controle el SWFA (Marrero, 2004<sup>18</sup>).

En la actualidad, el movimiento de SWFA ha sufrido cambios, entre ellos la ‘bifurcación’ de proyectos (variación de programas para distinguirse de la competencia) y el desarrollo de productos para el usuario final sofisticado (Marrero, 2004).

---

anterior significa que aún cuando el software ha sido modificado, no deja de ser libre (Free Software Foundation, 2003).

<sup>17</sup> Hay una segunda licencia conocida como BSD que, a diferencia de la GPL, permite al desarrollador quedarse con sus descubrimientos.

<sup>18</sup> Presidente de la Asociación Mexicana de Software Libre ([www.amesol.org.mx](http://www.amesol.org.mx)). Entrevista realizada en noviembre de 2004.

## **1.4 DOS FORMAS DE CARACTERIZAR LA TECNOLOGÍA DE SOFTWARE**

En este apartado se muestra que en la industria de software hay múltiples interacciones entre los agentes. Al mismo tiempo, se plantea que la I+C son fragmentarios pues cada agente posee una parte importante de éstos para el desarrollo de un producto o sistema de software. Dicha temática se explicará considerando dos enfoques de desarrollo de software: primero, desde la óptica de Sistemas de Productos Complejos (SIPC<sup>19</sup>) y, segundo, bajo la óptica de Software Incorporado (SWI). Como se mostró en el apartado anterior, el usuario surge como un agente importante en el segmento de software hecho a la medida. Este usuario posee I+C cruciales para el desarrollo de software.

### **1.4.1 El desarrollo de software desde el enfoque de sistemas de productos complejos**

Autores como Hobday (1996 y 1998), Hobday y Brady (1996 y 1997), Brady, (1997), Markose (1996) y Quintas (1995) caracterizan el proceso de innovación de software bajo el concepto de Sistemas de Productos Complejos (SIPC), definidos como productos o sistemas hechos a la medida con actividades intensivas en ingeniería y diseño, interacciones entre proveedor-usuario y altos costos de producción. Las investigaciones recientes de estos autores se han situado en: i) las mejores prácticas a nivel internacional en la administración de software, ii) los procesos de generación, administración y aplicación de software a través de redes de productos complejos, iii) la funcionalidad de la ingeniería de software, la integración de sistemas en el diseño, desarrollo y manufactura de productos complejos específicos, y iv) los métodos para mejorar la ingeniería de grandes proyectos.

En muchos SIPC el desarrollo de software ha sido de fundamental importancia, tanto para el producto como para el proceso de administración (herramientas de administración de proyectos). En estos sistemas complejos, el papel de los usuarios (algunas veces el propio operador del sistema) es central en el proceso de innovación, ya que pueden tener algún tipo de participación en las actividades de investigación y desarrollo (I+D), diseño,

---

<sup>19</sup> Sus siglas en inglés: CoPS, Complex Product Sytems.

integración hardware-software, mantenimiento y modificación ex-post (Hobday y Brady, 1997).

Estos sistemas complejos son intensivos en el uso de software en múltiples dimensiones: (i) sus características son determinadas por el software incorporado en ellos, lo cual provee las funciones del sistema; (ii) el uso de herramientas de software, tal como CAD/CAM<sup>20</sup> en el diseño y manufactura de componentes y sistemas; (iii) el software es el medio a través del cual se integran todos los componentes y sub-sistemas que componen la totalidad de un sistema complejo; (iv) el proceso de diseño y desarrollo de nuevos sistemas complejos implica la interacción entre oferentes, integradores de sistemas, usuarios y diferentes organizaciones nacionales o extranjeras; (v) el software provee al usuario las interfases por las cuales éste interactúa con el sistema; y (vi) el software permite el continuo desarrollo evolutivo de los SIPC, aun después de ser transferido al usuario, es decir, los sistemas están en continuo desarrollo (Quintas, 1995:3).

En este cuerpo de literatura se contrastan los sistemas complejos (modelo de SIPC) con los bienes producidos en masa (modelo tradicional), como se muestra en el cuadro 1.5, basados en componentes estándares con procesos codificados, por ejemplo, automóviles, televisores, circuitos integrados, transistores, calculadoras, máquinas de escribir, etc. (Hobday, 1996). A diferencia del modelo tradicional, el modelo SIPC se centra en las necesidades de usuarios sofisticados que se involucran en la creación y desarrollo del producto. Esto hace que los sistemas<sup>21</sup> complejos estén basados en proyectos que incorporan la participación de más de una empresa o institución en su producción, ya que los proyectos son inciertos y el riesgo es alto (Hobday y Brady, 1996:1).

---

<sup>20</sup> Diseño asistido por computadora (CAD: computer-aided design) y Manufactura asistida por computadora (CAM: computer-aided engineering).

<sup>21</sup> Un sistema se define por tres características: componentes, estructuras de red y mecanismos de control. Un sistema es una colección de sistemas o sub-sistemas distintos pero interrelacionados, cada uno desempeñando distintas tareas en forma organizada para lograr una meta común (por ejemplo, los aeropuertos consisten de la aviación, terminales, pistas de despegue-aterrizaje, control de tráfico aéreo, sistema de reclamo de equipaje, entre otros) (Hobday, 1998:5).

**Cuadro 1.5**  
**Características de los modelos de producción en masa vs SIPC**

<b>Categoría</b>	<b>Productos Simples/Producción en Masa*</b>	<b>SIPC/Proyectos</b>
Características de producto	Basados en el ensamble Interfases simples Bajos costos Ciclos del vida del producto cortos Insumos de baja tecnología Componentes estandarizados Intensivos en hardware Funciones simples Propiedades lineales predecibles Bienes de capital aguas abajo	Jerárquico/sistemático Interfases complejas Altos costos Ciclos de vida del producto largos Insumos multi-tecnológicos Componentes no estandarizados Intensivos en software Multi-funcionales, complejos Propiedades no lineales Bienes de capital aguas arriba
Características de producción	Alto volumen, grandes series de producción Diseño de manufactura Intensivos en manufactura Procesos incrementales, minimización de costos	Series pequeñas de proyectos Integración de sistemas Intensivos en ingeniería de software Intensivos en escala, producción en masa no relevante
Procesos de innovación	Dirigido por los oferentes  De los negocios para el consumidor Codificado, formalizado Innovación y difusión separados Sendero de innovación mediado por la selección de mercado Know-how incorporado en la maquinaria	Dirigida por la relación productor-usuario  De los negocios para los negocios Alta flexibilidad, basada en oficios Innovación y difusión plegables Sendero de innovación en acuerdo ex-ante entre los oferentes y usuarios Conocimiento incorporado en la gente
Estrategia competitiva y gestión de la innovación	Focalizado en economías de escala y minimización de costos Convencional Competencias de producción en volumen Focalizado en empresas individuales (i.e. producción esbelta)	Focalizado en el diseño y desarrollo de productos Orgánico Competencias integradas en sistemas  Alianzas multi-empresa en proyectos temporales
Organización, evolución y estructura industrial	Grandes empresas/estructura de cadenas de oferentes Empresas individuales, producción en masa Alianzas poco comunes, sólo para intercambio de actividades e I+D Diseño dominante	Instituciones complejas  Alianzas multi-empresa basadas en proyectos Alianzas multi-empresa temporales para la innovación y producción Estabilidad a largo plazo a pesar de cambio técnico radical
Características de mercado	Muchos compradores y vendedores Gran número de transacciones Mecanismos de regulación de mercado Comercial Mínima regulación Precios de mercado Alta competencia	Estructura duopólica Pocas transacciones Mecanismo de no-mercado vitales Institucional/política Regulado/controlado Negociación de precios A menudo no concursado

\*La especialización flexible es sólo una forma avanzada de la producción en masa, más que una categoría analítica distinta.

Fuente: Hobday, 1996, 1998.

De este cuadro se destacan las características más importantes en el desarrollo de SIPC, que difieren del modelo tradicional:

- a) La participación del usuario juega un papel importante en la interconexión entre sub-sistemas y componentes.<sup>22</sup>
- b) Los productos complejos son hechos a través de muchas interfaces<sup>23</sup> complejas entre usuarios, clientes, sub-sistemas y componentes (Hobday, 1996:10).
- c) Algunos sistemas y componentes son hechos a la medida: las partes son organizadas jerárquicamente y adaptadas a clientes o mercados específicos, la exigencia de calidad en el diseño y producción requiere continua retroalimentación de los usuarios, ingenieros, una interacción cercana y compleja entre compradores y vendedores.
- d) La naturaleza de los SIPC indica que el desarrollo, diseño e interacción son mucho más importantes que la propia manufactura. En el modelo tradicional, en las primeras etapas del ciclo de vida de la innovación, hay una actividad intensiva en diseño y un alto grado de interacción entre proveedor-usuario; sin embargo, el conocimiento se estabiliza (es decir se vuelve de tácito a formal), los mercados se expanden y los componentes se estandarizan. Posteriormente, la interacción proveedor-usuario es mediada por el mercado, como en el caso de los automóviles, microcomputadoras o electrónica de consumo. Mientras que en el modelo de SIPC, los sistemas complejos no llegan a las últimas etapas del volumen de producción del modelo tradicional (producción en masa), la ventaja competitiva y los incentivos para la innovación son centrales, la interacción proveedor-usuario es compleja y dinámica en varias etapas del ciclo de vida de la innovación (Hobday, 1998).
- e) Hay una alta interacción entre proveedor-usuario. Una de las razones se debe a que las necesidades de los usuarios son una fuente importante de información, se involucran en las actividades de I+D, producción, mantenimiento, escalamiento y rediseño. Otra razón es que de ellos depende el crecimiento de sus negocios y, en muchas ocasiones, son los responsables del mantenimiento, escalamiento y

---

<sup>22</sup> Desde la óptica de la producción modular (véase Hughes, 1989; Arora *et al.*, 1997), un componente siempre es parte de un sistema.

<sup>23</sup> Aunque en este cuerpo de literatura no se define interfase, ésta se conceptualiza en un contexto social-espacial como el lugar de la interacción, el espacio donde se desarrollan los intercambios. En el capítulo 3 se define el concepto de interfase, pero la idea fundamental parte de esta conceptualización.



modificaciones en el desempeño del producto y de la información para futuras innovaciones (Hobday, 1998).

- f) A diferencia de los usuarios ubicados en mercados de producción en masa, los usuarios que participan en SIPC aprenden e internalizan los sistemas tecnológicos para hacer productivos sus propios negocios, y llegan a acumular altos niveles de capacidades tecnológicas. Según Hobday (1996:13), a través de los usuarios el entorno alimenta directamente la trayectoria de innovación de los SIPC. Gran parte del conocimiento requerido está incorporado en los usuarios y no puede ser codificado al mismo grado, como sucede en el modelo de producción en masa.
- g) La complejidad de los SIPC también involucra instituciones complejas. La complejidad organizacional puede ser muy elaborada debido a la necesidad de sincronizar acciones y asegurar la participación conjunta de los agentes.
- h) La incorporación de software en productos ha transformado a los SIPC e impactado en la investigación, el diseño y la producción (Hobday, 1998).
- i) Dentro de las redes de SIPC usuarios, productores, sub-contratistas, reguladores e instituciones de estandarización integran sus capacidades y recursos a través del tiempo. Adicional a esto, el avance tecnológico en los SIPC depende en gran medida de la contribución de pequeñas y medianas empresas, las cuales han creado interfases con operadores, reguladores y otros agentes involucrados en el proceso de innovación.

A partir de la diferenciación que Hobday (1996 y 1998) hace entre ambos modelos, se puede resaltar la importancia de los siguientes elementos válidos para la industria de software. Primero, una alta participación del usuario debido a que es una fuente importante de información para el desarrollo de proyectos y acumula una parte importante del conocimiento que no puede ser fácilmente codificado. En este sentido, las necesidades y requerimientos de los usuarios son una fuente importante de I+C para el desarrollo de los proyectos y, por tanto, elementos cruciales para el proceso de innovación. Segundo, la relación unívoca proveedor-usuario no es suficiente, se requiere la participación de otros agentes como instituciones de gobierno, universidades, entre otros; es decir, hay múltiples interacciones entre los agentes o interfases como señala Hobday (1996 y 1998). Tercero, las

interacciones entre los agentes tienen un rol fundamental en el proceso de desarrollo de los proyectos debido a la integración de I+C que se ubican en diferentes poseedores, en este sentido, son I+C que se encuentran fragmentados y que, a través de la interacción, se articulan y complementan para un propósito. Finalmente, el software incorporado (SWI) en diversos productos complejos es un elemento que da originalidad a la tecnología de software, pues la vuelve transversal hacia otras industrias.

#### **1.4.2 El desarrollo de software desde el enfoque de software incorporado (embedded software)**

Como se mencionó en el apartado 1.2, el SWI es una combinación de hardware y software de sistemas para lograr una aplicación que otorga funcionalidad a un dispositivo carente de una computadora tradicional. Como se mostró en el punto anterior, los SIPC, tales como los sistemas de telecomunicaciones y de transporte público, así como productos de manufactura, como máquinas-herramienta, bienes electrónicos de consumo como PDA y teléfonos móviles, automóviles, aparatos médicos y aparatos domésticos, entre muchos otros, están incorporando software.<sup>24</sup>

Al igual que en los SIPC, el desarrollo e incorporación de software en diferentes dispositivos implica la participación de varios agentes. Por ejemplo, anteriormente los simuladores de vuelo y las centrales de telecomunicación eran construidos electromecánicamente por grandes fabricantes individuales, ahora, una gran parte de esas tareas de producción involucran el desarrollo de software y muchos de los sub-sistemas y componentes son subcontratados a oferentes especializados (Hobday, 1998:6).

SWI ha transformado la composición física y la funcionalidad de los sistemas y la forma en que los sistemas complejos son administrados, lo cual ha requerido la participación de oferentes y usuarios en la adquisición de riesgos y oportunidades. Asimismo, ha transformado la forma en cómo funciona la relación proveedor-usuario en un entorno

---

<sup>24</sup> Se estima que el mercado mundial de SWI (embedded software) es de 21 billones de dólares. Las industrias demandantes de SWI son, principalmente, la automotriz, electrónica de consumo y semiconductores (FUMEC, 2004).

complejo. A pesar de su importancia, SWI ha recibido poca atención, comparado con las áreas tradicionales de procesamiento de datos, sistemas de información y software empacado (Hobday, 1996:23). Actualmente, las TIC y, en particular, el SWI están cambiando la naturaleza, estructura y dinámica de la innovación en muchos SIPC.

Dada la complejidad que representa por sí mismo el diseño y desarrollo de software, la incorporación del software en sistemas complejos requiere de la integración de diferentes áreas de conocimiento y habilidades, y por tanto la participación de varios agentes de forma que se pueda unificar I+C fragmentarios.

A lo anterior se añade otro elemento, en los últimos años ha aumentado el número de empresas que utilizan sistemas operativos libres como LINUX, principalmente en servidores, y muchas otras lo han empezado a utilizar en sistema incorporados. Grandes empresas como Sony, Sharp y Matsushita Electric han utilizado LINUX para desarrollar productos como videograbadoras, televisión digital, juegos, PDA's. Sin embargo, los desarrolladores involucrados en sistemas incorporados y los fabricantes de productos comerciales se enfrentan con la realidad de que los sistemas de fuente abierta presentan problemas en términos de mantenimiento, soporte y calidad, asociados a la falta de mecanismos de racionalización de los procesos de desarrollo de software y de sistematización de la I+C utilizados en ese proceso, como se verá en el próximo apartado.

## **1.5 RACIONALIZACIÓN DEL DESARROLLO DE SOFTWARE**

El desarrollo de software ha sido en gran medida una actividad artesanal, es decir, en muchos aspectos es una tarea difícil de medir, maximizar y administrar, y en buena parte está en función de las habilidades y creatividad de los profesionales que lo desarrollan. Ello tiene sus implicaciones en la calidad, eficiencia y costos. A principios de la década de los 90 la tecnología de software inició una transformación importante, las actividades de desarrollo empezaron a cambiar del software de *oficio (o artesanal)* al software de *fabricación* (Tapscott y Caston, 1995), proceso que actualmente sigue teniendo relevancia.

La evolución en el desarrollo de software en los últimos años ha estado marcada por la necesidad de una mejor calidad y eficiencia en el trabajo, y por los límites de comunicación en cada estado de producción de software, así como por el aumento en la demanda de productos de software nuevos y más complejos (Eischen, 2002a).

Según este autor, en el desarrollo de software, la productividad ha estado asociada al aumento de las habilidades de trabajo restando importancia a la calidad y complejidad de las redes de comunicación interna,<sup>25</sup> lo cual es uno de los problemas de la racionalización: i) el desarrollo de software se ha resistido a la creación de un proceso repetible y cuantificable, ii) el proceso se puede administrar pero no racionalizar o sistematizar como en las grandes industrias, iii) racionalizar implica un proceso de medición, de maximización de la eficiencia a través de la división del trabajo, con insumos y productos definidos, administrados por una estructura organizacional con reglas delimitadas, iv) de modo que la calidad y productividad del software continúan siendo impredecibles e incuantificables, y los nuevos métodos organizacionales han fracasado para subsanar los límites de comunicación (Eischen, 2002a y 2003).

En los últimos dos siglos han habido industrias que, en un principio, parecían imposibles de racionalizar, basadas más en actividades de oficio que de manufactura, pero al madurar fueron racionalizando el proceso de manufactura y la forma de producir en masa. Los resultados de esa racionalización fueron el aumento en eficiencia y calidad de los productos. Los patrones básicos industriales de las actividades productivas que, un principio fueron de oficio o artesanales, estuvieron marcados por la división del trabajo, economías de escala, procesos e ingeniería de manufactura, patrones que aún no han dominado la industria de software. Se espera que el proceso de desarrollo de software pueda llegar a ser racionalizado (Eischen, 2003).

También, como apunta este autor, una simple razón de esa deficiencia es el ‘límite de comunicación’, lo cual no es nuevo, la comunicación siempre es difícil, ya sea mediada por

---

<sup>25</sup> El movimiento de software de fuente abierta (en proyectos como los de LINUX o APACHE, por ejemplo) está basado en redes de cooperación para escribir, escalar y depurar el código, guiados por una ideología de libertad y comunidad y no por una compensación económica.

códigos, normas, cultura o percepción, los cuales son siempre contextuales. Lo nuevo es que el software tiene la característica de otros medios de comunicación. Desde la década de los 70 se han enfatizado los límites del desarrollo de software, no como la escasez de herramientas o de estructuras organizacionales, sino como los límites en la calidad de la comunicación.

Siguiendo a Eischen (2002a y 2002b), el desarrollo de software y la construcción de los requerimientos básicos son un proceso de comunicación de conocimiento tácito, y trasladar el conocimiento de un contexto a otro, como trasladar cualquier lenguaje, implica no sólo reglas básicas de gramática y sintaxis, sino también el problema de los diferentes contextos en que se mueve. Los programadores hacen uso de reglas de sintaxis y semántica pero, como todos los lenguajes, la estructura y el significado varían en el tiempo y espacio, de la misma forma que entre programadores en un proyecto. No hay traducción automática de diseño o requerimientos dentro de un lenguaje de programación. Como en todos los lenguajes, hay múltiples vías de comunicación, conceptos y resultados. En esta interpretación, el proceso de oficio o artesanal<sup>26</sup> (contrario al proceso de ingeniería) significa que no hay leyes físicas universales que definan todo el desarrollo de software, y por tanto, no hay métodos directos que cuantifiquen el proceso de software y de los productos a estándares universales de calidad y eficiencia. Este aspecto restrictivo de comunicación limita la racionalización del desarrollo de software.

Eso ayuda a definir la importancia de la comunicación en el desarrollo de software, entre programadores o entre los diversos flujos de comunicación donde intercambian conocimiento tácito u organizacional. Ello explica por qué el desarrollo de software es más exitoso cuando se encapsula en un solo individuo (Eischen, 2002a).

La comunicación entre usuarios finales y programadores debe ser detallada e interactiva a todos los niveles del proceso de producción a fin de hacer específico, y a veces dominante,

---

<sup>26</sup> Las alternativas en la producción de software se han centrado sobre el conocimiento dominante tácito, así como en procesos artísticos descritos como oficios. A diferencia de la 'ingeniería de software', que pretende hacer del desarrollo de software un proceso racionalizable, con necesidades de manufactura y de lineamientos o patrones industriales.

el conocimiento tácito. El diseño en cualquier industria es siempre un reto intelectual, aun más en el software, porque es esencialmente un proceso de diseño y desarrollo. El método artesanal viene de las universidades y de las comunidades de programadores, donde la solución a problemas evoluciona colectivamente a través del tiempo. Por tanto, una racionalización del proceso de desarrollo de software podría crear una división del trabajo que permitiría estratificar el desarrollo en aspectos cualitativos, es decir, en el desarrollo de habilidades.

### ***Calidad y confiabilidad***

Según Yarime y Baba (2005a y 2005b), la calidad es difícil de medir y la confiabilidad es difícil de pronosticar. Precisamente, algunas de las características del software son, en primer lugar, la calidad, de la cual se derivan la confiabilidad del sistema y el desempeño operacional. Cada una tiene diferentes formas de ser medidas y evaluadas, por ejemplo, a través de los modelos de calidad. Entre éstos, uno de los más difundidos a nivel mundial es el Modelo de Maduración de Capacidades, CMM (Capability Maturity Model), desarrollado por la Universidad Carnegie-Mellon (Estados Unidos) para el Software Engineering Institute (SEI<sup>27</sup>), con el fin de evaluar las áreas de proceso en una empresa de software. Según este modelo, la madurez se alcanza cuando la empresa alcanza el quinto nivel:

**1°. Inicial.** Las empresas en este nivel no disponen de un ambiente estable para el desarrollo y mantenimiento de software. Aunque se utilicen técnicas correctas de ingeniería, los esfuerzos se ven minados por falta de planificación. El éxito de los proyectos se basa, la mayoría de las veces, en el esfuerzo personal aunque frecuentemente se producen fracasos y, casi siempre, retrasos con altos costos. El resultado de los proyectos es impredecible.

**2°. Repetible.** En esta etapa las empresas disponen de prácticas institucionalizadas de gestión de proyectos, existen métricas básicas y un seguimiento de la calidad. La relación con subcontratistas y clientes se gestiona sistemáticamente.

---

<sup>27</sup> Centro de Investigación y Desarrollo patrocinado por el Departamento de Defensa de los Estados Unidos y gestionado por la Universidad Carnegie-Mellon.

**3°. Definido.** Además de una buena gestión de proyectos, a este nivel las empresas disponen de procedimientos adecuados de coordinación entre grupos, formación del personal, técnicas de ingeniería más detalladas y un nivel más avanzado de métricas en los procesos. Se implementan técnicas de revisión.

**4°. Gestionado.** A este grado las empresas disponen de un conjunto de métricas de calidad y productividad que se usan de modo sistemático para la toma de decisiones y la gestión de riesgos. El software resultante es de alta calidad.

**5°. Optimizado.** Cuando la empresa obtiene el último nivel, los procesos están en mejora continua. Se hace uso intensivo de las métricas y se gestiona el proceso de innovación.

Como se ha mencionado en párrafos anteriores, los problemas para administrar los esfuerzos de programación interno y los altos costos de mantenimiento, así como la calidad y los problemas de medición del desempeño en las actividades de desarrollo han sido los grandes retos para la industria, y modelos como el CMM está orientado a cubrirlos. Sin embargo, este modelo no contempla procesos para internalizar en la organización la I+C existentes en el entorno (o bien I+C generados al interior de la empresa y a partir de las interacciones con otros agentes).

## **1.6 CONCLUSIONES DEL CAPÍTULO**

En este capítulo se han mostrado los siguientes aspectos fundamentales que definen la evolución de la industria de software: i) la separación de los precios del software respecto al hardware, evento que origina la independencia en el desarrollo de software respecto de a producción de hardware, ii) la organización de la industria de software en competencia oligopólica, iii) la producción de software ligada, en un principio, a plataformas de hardware, pero después se diversifica e incorpora software a otras plataformas de diversos sistemas y productos, iv) el nacimiento del software de fuente abierta, y v) la persistencia del problema de la racionalización del proceso de desarrollo de software.

Hay otros aspectos relevantes que se consideran en esta tesis. Primero, el desarrollo de software hecho a la medida de los usuarios y la participación de éstos en su desarrollo. Segundo, la participación del usuario como poseedor y fuente de una parte importante de

I+C, pues es importante por la asociación con sus experiencias. Tercero, la existencia de múltiples interacciones entre los agentes debido a la integración de diversas disciplinas, habilidades y conocimientos que cada agente posee para el desarrollo de software. En este sentido, las interacciones entre los agentes tienen un rol fundamental en el proceso de desarrollo de los proyectos debido a la integración de I+C que se ubican en diferentes poseedores, en este sentido, es I+C que se encuentran fragmentados y que, a través de la interacción, se articulan y complementan para un propósito.

En las primeras etapas evolutivas, el usuario tomó un rol crucial en la industria de software, pero conforme se estandarizaron los sistemas operativos y aplicaciones su participación fue decreciendo, sin embargo, en el segmento de software hecho a la medida el usuario ha seguido jugando un rol importante en la creación de nuevos productos de software. A pesar del dominio de las aplicaciones empaquetadas, el software hecho a la medida tiene una participación importante en el mercado total, en aproximadamente del 20% en México, ya que las necesidades y requerimientos de los usuarios siguen evolucionando y no todas sus necesidades se estandarizan, de forma que los usuarios seguirán teniendo un rol importante en el segmento de software hecho a la medida.

En suma, los puntos a rescatar de este capítulo y que sirven como antecedentes para esta tesis son los siguientes:

- i) El usuario se vuelve más importante en la generación de nuevos productos de software, pues, aunque su participación decreció cuando se estandarizaron los sistemas operativos y las aplicaciones, ha iniciado una nueva etapa en el segmento de software hecho a la medida y en el de software de fuente abierta. Ante ello, los usuarios son cada vez más *conscientes* de sus necesidades y requerimientos y, en muchas ocasiones, cuentan con conocimientos y habilidades relacionados con la tecnología de software. En esta tesis las interacciones entre empresas de software y usuarios son de fundamental importancia para el diseño y desarrollo de aplicaciones de software.



- ii) Los sistemas de software son hechos a través de múltiples interacciones entre las empresas de software, usuarios y otros agentes. Lo cual significa que la I+C son fragmentarios en un entorno complejo y dinámico como la industria de software.
- iii) Los productos son adaptados a clientes o mercados específicos, por lo que algunos componentes y sistemas de software son generalmente hechos a la medida.
- iv) El problema de la baja racionalización o sistematización del proceso de desarrollo de software se asocia a varias causas, entre ellas, la dificultad de comunicación interna o de intercambio de conocimiento tácito a nivel organizacional (como bien lo señala Eischen, 2002a y 2002b), la dificultad para identificar e internalizar la I+C existentes en el entorno y la I+C generados a partir de las interacciones con diversos agentes.

Además, hay otros elementos importantes que han sido poco señalados; por un lado, no es clara la forma en cómo las empresas de software identifican I+C existentes en el entorno y generados a partir de las múltiples interacciones con usuarios y otros agentes, por otro, cómo internalizan esa I+C y los explotan para fines comerciales. Estos problemas han sido poco abordados en los estudios acerca de la industria de software y, como veremos en los siguientes capítulos, son dos de las limitaciones más importantes en las empresas de software, sobre todo para el caso de la micro y pequeña empresa mexicana.

## CAPÍTULO DOS

### JERARQUÍA DE PROBLEMAS EN LA INDUSTRIA MEXICANA DE SOFTWARE

#### INTRODUCCIÓN

La industria mexicana de software no ha sido estudiada ni caracterizada a profundidad, de hecho, se conoce poco acerca de sus características industriales y tecnológicas. Sin embargo, recientemente se han hecho algunos esfuerzos en esa dirección, por ejemplo, se ha tratado de identificar: i) las características de la oferta y demanda de recursos humanos (UAM, 2004), ii) las acciones y decisiones fundamentales que permitan a esta industria alcanzar un nivel competitivo y sostenido en el mercado local e internacional (ITAM, 2005<sup>1</sup>), iii) los impactos que la industria tendría sobre el crecimiento económico (Sallstrom y Damuth, 2003), iv) los lineamientos de política industrial que debieran imperar para que esta industria alcance niveles de competitividad local basado en el mercado interno y competitividad internacional basado en las exportaciones (SE, 2005<sup>2</sup>), y v) los factores críticos de éxito en la industria de software (González, 2006).

Con base en los estudios existentes, la importancia de este capítulo radica en presentar algunos de los problemas más relevantes en la industria de software. En el primer apartado se describe la estructura de los principales países emergentes que cuentan con una industria de software. En el segundo se describe la estructura de la industria de software en México, así como también las características de la oferta y demanda de recursos humanos. En el siguiente se describen las estrategias sectorial y regional más importantes para el fortalecimiento de la industria de software. Finalmente, se presentan las conclusiones tanto del capítulo como de la Parte I.

---

<sup>1</sup> Informe elaborado en el Centro de Estudios de Competitividad del Instituto Tecnológico Autónomo de México (ITAM) y coordinado por el Dr. Ignacio Ania B.

<sup>2</sup> Informe elaborado por la Secretaría de Economía: *Programa para el Desarrollo de la Industria del Software* (PROSOFT), México.

## 2.1 LA INDUSTRIA DE SOFTWARE EN UN CONTEXTO INTERNACIONAL

En 2005, la OECD (2007) estimó a nivel mundial el gasto en Tecnologías de Información y Comunicación (TIC<sup>3</sup>) en 2,964 billones de dólares, de los cuales el 10% correspondió a gastos en software (296.4 billones de dólares). Estados Unidos es el principal consumidor de TIC, pues en el 2005 consumió el 37% del total de TIC y el 47% del total de software producido.

Los principales países productores y exportadores son, en orden de importancia, Estados Unidos, Alemania, Japón, Inglaterra y Francia, quienes dominan ciertos segmentos de la producción de software, particularmente los de mayor tamaño y uniformidad de requerimientos funcionales. Algunas naciones emergentes han desarrollado capacidades de diseño y producción de software y se han incorporado en la competencia mundial de esta tecnología, tales como la India que tiene una alta capacidad exportadora de software respecto a sus ventas totales (76%), asimismo Irlanda (74%) e Israel (73%) (cuadro 2.1).

**Cuadro 2.1**  
**Estructura de la industria de software en India, Irlanda e Israel, 2002**

País	Ventas (mdd*)	Exportaciones (% de ventas)	Empresas	Empleo	Capital de riesgo (mdd)	Política Pública orientada a:
India	12,500	76%	6,000	400,000	1,100	IED**, Capital de riesgo, Incubadoras de empresas, Parques tecnológicos
Irlanda	14,000	74%	1,000	30,000	150	IED, Capital de riesgo, Incentivos fiscales, Joint-Ventures
Israel	4,100	73%	400	20,000	1,500***	IED, Capital de riesgo, Incentivos fiscales, Subsidios

\*mdd: millones de dólares.

\*\*IED: Inversión Extranjera Directa.

\*\*\*Dato para el año 2005.

Fuente: OECD (2003 y 2007), Delgado y Hernán (2005), Shapiro (2006).

Como señalan Piore, Schrak y Ruiz Durán (2005), estos países han forjado sus propias rutas de desarrollo y creado nichos de mercado exitosos diferentes para cada uno. La India es un buen ejemplo del grado de desarrollo de esta industria, pues en 2002 exportó 76% del total de software, proporción que se distribuyó entre Estados Unidos (63%), Europa (26%)

<sup>3</sup> Las TI la conforman las industrias de hardware, software y servicios relacionados. Mientras que las TIC la conforman las industrias de microelectrónica, computación, informática [software] y telecomunicaciones.

y Asia (11%) (OECD, 2003). Se ha cuestionado el modelo de la industria de software de la India, se dice que está basado en la venta masiva de recursos humanos para tareas de bajo valor agregado (*off share*), ya que la subcontratación se ha presentado de manera masiva en los procesos más simples y estandarizados (Leamer y Storper, 2001<sup>4</sup>). Sin embargo, han empezado a diseñar y desarrollar grandes proyectos de TIC (OECD, 2007).

En la India hay empresas grandes como IBM, MICROSOFT, CISCO, GE, ORACLE, NOVELL, SUN MICROSYSTEMS, entre otras, pero la industria de software presenta un alto grado de atonicidad: el 55% son pequeñas empresas, el resto lo componen medianas y grandes empresas (OECD, 2007).

La industria de software de Irlanda presenta una estructura interesante. El modelo de la industria está basado en una estrategia de exportación de productos y servicios de software con un alto valor agregado, principalmente a Europa. A diferencia de la India, los inicios de la industria estuvieron marcados por un gran número de proyectos que implicaron su concepción, diseño, desarrollo, implementación y soporte a los clientes. El 50% de las empresas ubicadas en Irlanda son extranjeras, este fenómeno se da por ser un país de habla inglesa, cuenta con recursos humanos calificados y con menores costos relativos a otros países de la Unión Europea (Delgado y Hernán, 2005).

Por su parte, la industria de software de Israel ha dado saltos cuantitativos considerables, por ejemplo, en 1990 exportó 90 millones de dólares (mdd) y en 1999 esa cifra alcanzó los 2,600 mdd, actividad que hasta la fecha ha tenido un crecimiento medio anual del 25%; entre sus principales compradores están Inglaterra, Alemania y Francia (Shapiro, 2006). El desarrollo de software en Israel se ha posicionado en los mercados europeo y asiático por su alta confiabilidad tecnológica creada a partir de los desarrollos hechos para la industria militar (basado en una intensa actividad de Investigación y Desarrollo -I+D-); además de una alta flexibilidad para contratos de outsourcing. Es de los pocos países que ha

---

<sup>4</sup> Citado por Hualde y Gomis (2007:198).

desarrollado una ventaja competitiva basada en actividades de I+D, y el único que cuenta con centros de I+D de empresas globales como CISCO y MICROSOFT<sup>5</sup> (Shapiro, 2006).

Siguiendo a este autor, la mayoría de empresas locales de Israel son medianas, lo cual permite un mejor acercamiento y flexibilidad, así como atacar nichos de mercado en los cuales no compiten con las grandes empresas globales para satisfacer necesidades desatendidas por éstas.

En términos de política pública, aunque cada país tiene sus particularidades, el apoyo a la industria de software se rige por algunos patrones similares. Los gobiernos de India, Irlanda e Israel, por ejemplo, han creado esquemas para atraer Inversión Extranjera Directa (IED), uno de ellos a través de fomentar el capital de riesgo y otorgar incentivos fiscales a la producción o a la exportación. Aunque la política pública de estos países está orientada a fortalecer la estructura exportadora de esta industria, el fortalecimiento del mercado interno también ha sido un factor importante.

En el contexto latinoamericano, algunos países como Brasil, México, Argentina, Uruguay y Costa Rica han mostrado un desempeño significativo en la industria de software, cada uno con estrategias de desarrollo diferentes. Según Mochi (2006:116), países como Brasil y Argentina están basados en el desarrollo del mercado interno, otros como Uruguay y Costa Rica fundamentan el desarrollo del mercado externo. La política sectorial para el desarrollo de la industria de software en México tiene una doble orientación: una estrategia de exportación basada en el desarrollo del mercado interno, como veremos en el siguiente apartado.

En el cuadro 2.2 se muestran algunos indicadores de la industria de software de Brasil y Argentina. En él se ilustra una pobre actividad exportadora, comparada con la de los países citados en párrafos anteriores, debido a que, en términos de política pública, no hubo una

---

<sup>5</sup> Otras empresas globales importantes ubicadas en ese país son IBM, MICROSOFT, HP, MOTOROLA, INTEL, AOL, entre otras.

orientación de apoyo a esa actividad y las empresas no basaron su crecimiento en la exportación.

**Cuadro 2.2**  
**Estructura de la industria de software en Brasil y Argentina, 2002**

País	Ventas (mdd)	Exportaciones (% de ventas)	Empresas	Empleo	Política Pública orientada a:
Brasil	7,700	1.3%	3500*	158,000*	Fomento a la inversión, Joint-Ventures, Uso de software libre, IED.
Argentina	1,100	16%	300	14,500	Incentivos fiscales, Fomento a la inversión, Capital de riesgo.

\*Dato para 2003

Fuente: Delgado y Hernán (2005), Mochi (2006), Jenkins (2006).

La industria de software de Brasil tiene una amplia diversidad de especializaciones y productos que destinan a la industria de manufacturas, telecomunicaciones, banca y finanzas, al sector gobierno y educación, entre otros. Mientras que la industria argentina de software se caracteriza por proveer a grandes empresas de manufactura y servicios (tanto locales como extranjeras) ubicadas al interior del país.

La estructura de la industria en ambos países es: en Brasil alrededor del 43% son micro y pequeña, 22% mediana y el resto lo constituyen empresas grandes; en tanto que en Argentina el 68% son microempresas (Delgado y Hernán, 2005).

Aunque en ambos países la industria de software creció sin la protección estatal y sin una política sectorial oficial, actualmente Brasil cuenta con programas de apoyos a la industria, tales como el Programa Nacional de Software para Exportación y el Programa para la Industria Nacional de Software y Servicios Relacionados; mientras que Argentina promulgó la Ley de Promoción de la Industria del Software en 2004, bajo la cual las empresas exportadoras gozarán de incentivos fiscales y estabilidad fiscal (Jenkins, 2006).

Se observa que no hay un país latinoamericano con una industria de software con la capacidad de producción y exportación comparada, con la de India, Irlanda o Israel. Lo anterior se puede agudizar con la entrada de China al mercado, ya que este país se ha focalizado en diferentes nichos de mercado, entre los más importantes en el desarrollo de

software anti-virus, aplicaciones para el sector financiero, educación, gobierno y procesos de manufactura, con una alta participación de las universidades locales en la formación de recursos humanos y capacitación. El gobierno ha jugado un rol proactivo desde la década de los noventa en la formulación e implementación de políticas para el desarrollo de las TIC, ha creado parques de alta tecnología en 90 por ciento de las provincias (Piore, Schrak y Ruiz Durán, 2005).

Baskaran y Muchie (2006) analizaron el sistema nacional de innovación y el sub-sistema de Tecnologías de Información -entre éste el de software- en la India, China, Sudáfrica, Tailandia y Brasil. La evidencia mostró que en todos estos países el sub-sistema tiene problemas por resolver, tales como: (i) la desarticulación y falta de vínculos entre las instituciones de gobierno, empresas, instituciones de I+D, organizaciones públicas y privadas, organizaciones multi-nacionales, etc., y (ii) la falta de oferentes locales que permitan una mayor integración de las cadenas de valor locales. Además, la relación proveedor-usuario es pobre, el aprendizaje tecnológico es insuficiente y el mercado estrecho.

## **2.2 LA INDUSTRIA DE SOFTWARE EN MÉXICO**

En México, la industria del software aún está en proceso de desarrollo y crecimiento, tiene un nivel de gasto en TIC de 1.4% respecto al producto interno bruto (PIB), ubicándose en el lugar 19 a nivel mundial; comparado con 4.3% del promedio mundial de los países de la OCDE, y 5.5% el de Estados Unidos (SE, 2005<sup>6</sup>). Este rezago es aún mayor en términos de gasto en software, ya que respecto al PIB, la industria mexicana del software representa apenas el 0.1% (SE, 2005), esta participación es 6 veces inferior al promedio mundial y 9 veces menor que el de Estados Unidos (SE, 2006). En América Latina, México ocupa el segundo lugar en todos los sectores de TI en cifras absolutas, después de Brasil, pero en el periodo 2001-2005 creció menos que este país y otros latinoamericanos (Mochi, 2006).

---

<sup>6</sup> Secretaría de Economía (SE) y su portal [www.Software.net.mx](http://www.Software.net.mx), sitio oficial de la industria de software en México.

Durante el periodo 1992-2003 la industria de software tuvo una participación promedio anual de 7.7% respecto al total de la industria de TI en México, una participación pequeña comparada con los mayores niveles de participación de la industria de Servicios de TI (22% promedio anual) y la industria de hardware (40.7%) (cuadro 2.3).

**Cuadro 2.3**  
**Tecnologías de Información y Comunicación en México. Millones de dólares**

Año	TI hardware	TI software (manufactura)	TI servicios	Interno *	Otro equipo de oficina	Total TI (A)	Telecomunicaciones (B)	Total A+B
1992	1,675	302	818	1,467	178	4,440	N. D.	4,440
1993	1,695	370	1,016	1,448	177	4,706	9,068	13,774
1994	1,873	401	1,165	1,454	205	5,098	9,897	14,995
1995	1,098	234	636	1,343	104	3,415	7,205	10,620
1996	1,825	336	758	1,909	165	4,993	7,873	12,866
1997	2,267	428	1,025	1,943	205	5,868	8,389	14,257
1998	2,377	492	1,299	2,097	215	6,480	8,052	14,532
1999	2,751	533	1,564	2,171	249	7,268	8,930	16,198
2000	3,340	610	1,780	2,188	302	8,220	10,166	18,386
2001	3,316	597	1,865	2,326	300	8,404	10,806	19,210
2002**	3,600	631	1,955	N. D.	N. D.	6,186	20,743	26,929
2003	3,773	637	2,100	N. D.	N. D.	6,510	22,923	29,433
2004***	N. D.	N. D.	N. D.	N. D.	N. D.	7,418	N. D.	-
2005	N. D.	817	2,311	N. D.	N. D.	8,249	N. D.	-
2006	N. D.	N. D.	N. D.	N. D.	N. D.	9,824	N. D.	-

\*Gasto interno: mientras que el gasto externo incluye, casi siempre, la porción tangible del mercado de TI, el gasto interno se compone de la porción del 'efecto de rizo' del gasto externo. Esos gastos no pueden atribuirse a ningún vendedor y por consiguiente se denomina gasto interno. De esta forma, el gasto interno incluye la porción interna de presupuesto operativo del departamento de Sistemas Internos, desarrollo interno de software, la depreciación del capital y cualquier otro gasto relacionado con TI que no pueda ser atribuido directamente a un vendedor. N.D.=No disponible.

Fuente: Digital Planet. *The Global Information Economy*. WITSA, ITU, World Bank, Consensus Forecast, International Data Corporation (Citado por Maroto y Zavala, 2002). \*\*A partir de 2002 con datos de INEGI/Select-IDC. \*\*\*A partir de 2004 con datos de PROSOFT (SE, 2005).

Si bien la participación de la industria manufacturera de software en las TIC es baja, hay que considerar dos aspectos. Por un lado, la industria manufacturera de software sólo considera el software empaquetado y no el hecho a la medida, éste se registra en el segmento de Servicios de TI. La razón de registrarlo en este segmento se debe, en parte, a que es considerado una actividad específica a las necesidades del usuario y no se produce en grandes cantidades como el software empaquetado. Según Mochi (2006), en el año 2005 el software hecho a la medida alcanzó los 221 mdd.



Por otro lado, como se aprecia en el cuadro 2.3, los valores del segmento de Servicios de TI son mayores al de software porque registra, además del hecho a la medida, diferentes servicios que van desde la consultoría de software hasta servicios de capacitación, como se aprecia en el cuadro 2.4. Hay que considerar también, que una gran parte del segmento de Servicios de TI (alrededor del 90%) se realiza en México, mientras que una gran parte del software de manufactura es importado, sobre todo las aplicaciones empaquetadas cuya importación es alrededor del 90%, principalmente de Estados Unidos (González, 2006).

**Cuadro 2.4**  
**Clasificación de las actividades de software**

<ul style="list-style-type: none"> <li>• Desarrollo de software empaquetado</li> <li>• Desarrollo de software de sistema y herramientas para desarrollo de software aplicativo</li> <li>• Desarrollo de software aplicativo</li> <li>• Servicios de consultoría de software</li> <li>• Servicios de mantenimiento y soporte de sistemas computacionales</li> <li>• Servicios de análisis de sistemas computacionales</li> <li>• Servicios de diseño de sistemas computacionales</li> <li>• Servicios de programación de sistemas computacionales</li> <li>• Servicios de procesamiento de datos</li> <li>• Servicios de diseño, desarrollo y administración de bases de datos</li> <li>• Servicios de implantación y pruebas de sistemas computacionales</li> <li>• Servicios de integración de sistemas computacionales</li> <li>• Servicios de mantenimiento de sistemas computacionales y procesamiento de datos</li> <li>• Servicios de seguridad de sistemas computacionales y procesamiento de datos</li> <li>• Servicios de análisis y gestión de riesgos de sistemas computacionales y procesamiento de datos</li> <li>• Procesos de negocio relacionados con sistemas computacionales y comunicaciones</li> <li>• Servicios de valor agregado de análisis, diseño, desarrollo, administración, mantenimiento, pruebas, seguridad, implantación, mantenimiento y soporte de sistemas computacionales, procesamiento de datos y procesos de negocio</li> <li>• Servicios de capacitación, consultoría y evaluación para el mejoramiento de la capacidad humana, aseguramiento de la calidad y de procesos de las empresas del sector de TI.</li> </ul>
---

Fuente: SE, 2005.

Como se aprecia en el cuadro 2.4, la industria de software se puede separar en diversas actividades, sin embargo, en forma agregada son importantes dos: las aplicaciones empaquetadas y las que son hechas a la medida. Mochi (2006) identifica cuatro grandes segmentos de la industria mexicana de software:

- a) Desarrollo de software y servicios informáticos: orientada a la producción nacional.
- b) Autoconsumo: producción, desarrollo o adaptación hecha por los departamentos de sistemas de empresas y gobierno.

- c) Software empaquetado: distribución para consumo nacional que realizan las empresas trasnacionales, acompañado de actividades de soporte técnico y asistencia a las empresas.
- d) Subcontratación: grandes empresas exportadoras de productos electrónicos producen y subcontratan ‘software incorporado’ como parte de sus actividades de producción.

En esta tesis se han considerado sólo empresas (estudios de caso) desarrolladoras de software hecho a la medida. Una de las razones se debe a que una gran parte de las aplicaciones empaquetadas son importadas y no reflejan las capacidades tecnológicas de desarrollo de software reales de las empresas nacionales.

Considerando el cuadro 2.5, se observa que del total de producción de software en México, el empaquetado representa el 29.4 % del total, el segmento de software hecho a la medida representa sólo el 7.9%, mientras que el de producción y autoconsumo (segmento C) representa el 62.6%. Si la producción de autoconsumo del gobierno y las industrias de manufactura se contratara a empresas independientes, la demanda de software hecho a la medida representaría alrededor del 70.5% del total de la industria. Esta demanda potencial puede ser aprovechada por las empresas de software si cuentan con las capacidades tecnológicas para ofrecer soluciones a problemas específicos del sector gobierno y de las industrias de manufactura, pero además si el gobierno y las empresas deciden dejar de producirlo por sí mismos y contratarlo a empresas independientes.

**Cuadro 2.5**  
**Participación por segmento de software, y sectores de demanda, 2005**

<b>Segmento de la industria</b>	<b>Producción/ Desarrollo (mdd)</b>	<b>Porcentaje de participación</b>	<b>Sectores de demanda (por orden de importancia)</b>
(A) Software empaquetado	817	29.4%	Servicios, Gobierno, Finanzas, Comercio, y Manufactura.
(B) Software hecho a la medida	221	8.0%	Servicios (finanzas, seguros, educación, transporte, salud, cultura), Gobierno, Industria (construcción, electricidad, agua, gas, manufactura y minería), y Comercio.
(C) Software de producción y consumo interno	1,738	62.6%	Gobierno y Manufactura.
<b>Total</b>	<b>2,776</b>	<b>100%</b>	-

Fuente: Elaboración propia con base en Mochi (2006).

Si consideramos sólo los segmentos A y B del cuadro 2.5 y excluimos el segmento C, que corresponde a la producción y autoconsumo interno de parte del gobierno y las empresas de manufactura, las aplicaciones empaquetadas representan el 78.7% de la industria, mientras que el software hecho a la medida representa el 21.3%, participación real del segmento B en la industria de software en México.

En 2005, las exportaciones de software hecho a la medida fueron de 164 mdd, lo cual representa el 74% de este segmento, pero considerando la suma de los segmentos A y B sólo representan el 16%.

### **2.2.1 Software propietario**

El número de empresas de software en México no es exacto, se especula que hay entre 1000 y 1500. Tampoco hay datos reales de la proporción de empresas desarrolladoras de software, ya que muchas de ellas ofrecen servicios de consultoría o comercializan productos empaquetados. Sin embargo, se han hecho estudios que permiten deducir en la estructura que podría presentar la industria nacional. En el año 2001, la Asociación Mexicana de la Industria de Tecnologías de la Información (AMITI) realizó una encuesta a 206 empresas desarrolladoras de software y, en 2006, González realizó un estudio a 68 empresas, en ambos estudios se encontró que la industria presenta una estructura atomística: alrededor del 87% de las empresas son micro y pequeña, 7% son mediana empresa, 5% son gran empresa, y solamente una empresa que cuenta con alrededor de 1500 empleados se considera empresa corporativa.

Si las empresas nacionales quisieran incursionar en el mercado de exportación, el escenario se complica. El tamaño mínimo de entrada a este mercado es de 250 empleados y debe crecer a 1,000 empleados para ser competitiva. Las empresas de clase mundial cuentan con más de 5 mil empleados, además, de estar regidas por los estándares de calidad internacional como CMM,<sup>7</sup> ISO 15504 u otro (AMITI, 2001).

---

<sup>7</sup> Este modelo de calidad en software tiene 5 niveles, los cuales se describieron en el capítulo 1.

Los criterios internacionales de calidad de software indican que las empresas deben estar certificadas en sus procesos de desarrollo, pero pocas empresas nacionales son las que han accedido a implementar los procesos de calidad. Según la Secretaría de Economía, SE, (2006), durante 2005 en México había alrededor de 47 empresas en proceso de certificación (13 en CMM, 27 en CMMI y 7 en MoPROSOFT<sup>8</sup>), pero un gran porcentaje de ellas son filiales de empresas extranjeras.<sup>9</sup> Las ventajas que se obtienen al implementar programas de calidad en la mejora de procesos de software se orientan en tres direcciones:

- a) La Mejora de Procesos de Software (MPSW) incrementa el nivel de productividad y permite la estandarización y optimización de procesos y recursos,
- b) Los Modelos de MPSW son la norma para determinar la madurez de los procesos de desarrollo de software y sirven de estándar para evaluar la capacidad de las empresas.
- c) La Calidad del Software proporciona una mejor y más sólida posición competitiva tanto en el mercado mundial como en el local.

A nivel de empresa son pocos los estudios que se han hecho en esta industria. En un estudio realizado a 68 empresas, González (2006) encontró que los factores de éxito de las empresas de software son los recursos humanos, la calidad, especialización, negociación con los clientes y la innovación. En otro estudio realizado a 30 empresas de Baja California, Hualde y Gomis (2007) encontraron que las características técnico-productivas de las distintas actividades que comprenden la industria del software dan lugar a diferentes relaciones entre empresas, y en particular, el segmento de software hecho a la medida se desarrolla mediante una estrecha y constante interacción con el cliente. A esta misma afirmación llegó Mochi (2006). En cuanto a la organización de la empresa, ésta tiene una estructura interna flexible: ‘una diferenciación formal poco clara de los puestos de trabajo’, sobre todo en las pequeñas y medianas empresas (Hualde y Gomis, 2007).

---

<sup>8</sup> El Modelo de Procesos para la Industria de Software (MoPROSOFT) es una norma mexicana reconocida a nivel internacional que certifica los procesos de desarrollo e ingeniería de software, la cual se hizo oficial en 2005. Este modelo permitirá que la pequeña y mediana empresa de software tenga acceso a las prácticas de Ingeniería de Software de clase mundial y así alcanzar ‘niveles internacionales en capacidad de procesos’ (SE, 2006). Para un mayor detalle véase [www.Software.net.mx](http://www.Software.net.mx)

<sup>9</sup> La India es un parámetro de referencia. En 2002 había 48 empresas de software certificadas en CMM nivel 5, y en 2003 la cifra llegó a 65. Pero si se consideran a las empresas de software que han obtenido un certificado de calidad de procesos, la cifra asciende a 275 (Baskaran y Muchie, 2006:125).

### 2.2.2 Software de fuente abierta

Respecto al software de fuente abierta (SWFA), éste tiene aproximadamente 15 años de desarrollarse en el país. En México y otros países hay empresas de software comercial o propietario que utilizan el SWFA como parte de una estrategia mayor (dominación de su nicho de mercado, venta de servicios, soporte y capacitación, venta de "agregados" propietarios) y por lo tanto sus mercados son muy variados.

Se especula que en México existen alrededor de 100 empresas que desarrollan SWFA, de las cuales, casi un 90% son micro empresas con menos de 15 empleados cada una.<sup>10</sup> El desarrollo de este tipo de software está regido por las 'comunidades de desarrolladores',<sup>11</sup> cuya cifra en 2004 consideró que había 1,000 desarrolladores y cerca de 10 mil usuarios a nivel nacional trabajando en diversas comunidades.<sup>12</sup>

A pesar de su incipiente desarrollo, los programas hechos en SWFA han aumentado en los últimos años, sobretudo en los servidores.<sup>13</sup> En el 2002, por ejemplo, la participación del SWFA respecto al software total nacional se consideró en 7.9% y en 9.2% para el 2004. A nivel nacional, el número de servidores con programas de SWFA fueron 14,960 en el 2003 y 20 mil en el año 2004.<sup>14</sup> Las cifras muestran un rápido crecimiento de participación de programas hechos en SWFA respecto del total de software hecho en el país.

Por el contrario, el SWFA no ha podido penetrar en las microcomputadoras, tales como computadores de 'escritorio' o laptops, es decir, en el mercado más grande de consumo. El SWFA se ha centrado en las computadoras centrales o mainframes, en servidores y estaciones de trabajo, sin embargo, se prevé que en un corto plazo empezarán a llegar programas hechos en SWFA para microcomputadoras. Sin bien ya existen programas que

---

<sup>10</sup> Con base en entrevistas realizadas a desarrolladores de SWFA.

<sup>11</sup> Las comunidades de desarrolladores de SWFA está formada por dos agentes: 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 en estas comunidades.

<sup>12</sup> Con base en entrevistas realizadas a desarrolladores de SWFA.

<sup>13</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 SWFA ha penetrado mayormente en el segmento de servidores.

<sup>14</sup> Con base en entrevistas realizadas a desarrolladores de SWFA.

pueden correr en sistemas operativos propietarios, esos programas aún no son lo suficientemente maduros para consolidarse en el mercado de masas.

También se especula que las ventas de SWFA crecen 10 veces más que las de software comercial, pues los programas son económicos, flexibles, confiables e independientes. Algunos gobiernos como el español, el chino, el alemán, el noruego, han usado SWFA como parte de su estrategia para el desarrollo de su industria de software.

### **2.2.3 Características de la oferta y demanda de recursos humanos en la industria mexicana de software**

Uno de los insumos más importantes en la industria de TIC, y en particular en la de software, es el conocimiento incorporado en las personas y en las organizaciones. Una parte del conocimiento requerido por la industria es generado, principalmente, en universidades, institutos y centros de capacitación. Aunque estos agentes han formado una masa crítica importante de recursos humanos con diferentes conocimientos y habilidades, son insuficientes y no hay una correspondencia con la demanda real de la industria.

Según el Censo Económico 2004, en México había 3 millones 5 mil 157 unidades económicas en 2003, que realizaron alguna actividad económica objeto de censo (INEGI, 2004<sup>15</sup>), de éstas el 37% son usuarias de tecnologías de información (TI)<sup>16</sup>. Un estudio realizado por la UAM (2004)<sup>17</sup> muestra que el 62% de las empresas con uso de un departamento o área de sistemas emplean entre 1 y 5 profesionales, a diferencia del 38% restante con más de 6 profesionales, y aproximadamente el 30% de los empleados de sistemas son contratados bajo el esquema de outsourcing.

---

<sup>15</sup> El 97.2% de las unidades se ubican en los sectores de servicios, comercio y manufacturas, y emplean al 88.8% del total del personal ocupado (en total había 16.2 millones de personas ocupadas). Véase “Censos Económicos 2004” en [www.inegi.gob.mx](http://www.inegi.gob.mx)

<sup>16</sup> Con base en el Censo Económico 2004, una empresa es considerada usuaria de TI si utiliza Internet en su relación con clientes y proveedores, o si cuenta con equipo informático para sus procesos administrativos o para el desarrollo de programas orientados a mejorar sus procesos de producción.

<sup>17</sup> La muestra estadística de este estudio es de 800 empresas, de las cuales 750 tienen entre 101 y 1000 empleados, mientras que las 50 restantes tienen más de 1000 empleados. Entre los sectores industriales donde se ubican las empresas de la muestra están: Minería, Electricidad y Construcción, Industrias Manufactureras, Comercio y Servicios, Transporte, Servicios Financieros y Actividades de Gobierno. El estudio fue elaborado por un grupo de investigadores de la Universidad Autónoma Metropolitana (UAM).

Por el lado de la oferta de recursos humanos, a finales de la década pasada había 334,971 profesionales de TI trabajando en departamentos de sistemas, cerca del 67% desempeñaba actividades relacionadas con el software en departamentos de sistemas de las empresas y gobiernos y en empresas de software (UAM, 2004:20). De este porcentaje, los profesionales en software con estudios principalmente de licenciatura en computación e informática y egresados de instituciones públicas, trabajaban en empresas de comercio y servicios (53%), manufactureras (18%), actividades de gobierno (9%), servicios financieros (5%), electricidad y construcción (3%), transporte (3%), y actividades agropecuarias (1%), concentrados, particularmente el 50.1%, en el Estado de México, Distrito Federal, Nuevo León y Jalisco (UAM, 2004:13).

En 2004 se estimó que los profesionales de TI ascendieron a 525,741, de ellos el 61% tienen grado técnico y el 39% de licenciatura y posgrado. Cerca de 289 mil personas se encuentran laborando en áreas o departamentos de software, principalmente en desarrollo y pruebas de software. De éstos, poco más de la mitad (53.6%) cuenta con estudios adicionales, tales como cursos de capacitación, diplomados y maestrías (UAM, 2004:13, 77, 84).

Las empresas argumentan que, en el corto plazo, los profesionales deberán tener conocimiento y habilidades en bases de datos con interfaces XML y aquellas orientadas a objetos; mientras que las herramientas y lenguajes que se demandarán son C#, JAVA, JAVASCRIPT, HTML/XML, ORACLE DEVELOPER, POWER BUILDER, VISUAL Y BASIC, C, y FORTRAN (UAM, 2004:74). También hay una marcada tendencia a la realización de proyectos según las necesidades (software hecho a la medida) con una integración de plataformas, programas y aplicaciones.

Aunque no se conoce la demanda cuantitativa real de recursos humanos para la industria de software, en mayor medida se conoce la demanda cualitativa a través de la demanda de habilidades de los departamentos de sistemas de empresas ubicadas en diversas industrias. Según el estudio realizado por la UAM (2004), hay diferencias entre las características cualitativas que tienen los empleados ubicados en áreas relacionadas con el software y las

requeridas por las empresas que los contratan, es decir, se han encontrado asimetrías entre la oferta y la demanda de recursos humanos.

Estas empresas solicitan personal con licenciatura en computación (41.1%) e informática (25.6%), seguidas por técnicos en programación y otras licenciaturas y carreras técnicas (12.8%). Las empresas prefieren las certificaciones, los cursos y especialidades (71.9%), mientras que para el 79.9% de los profesionales es más importante realizar cursos, diplomados y maestrías. Por otro lado, las empresas prefieren la capacitación en mantenimiento y bases de datos, en contraste con los profesionales quienes se inclinan por la capacitación en computación e inglés. A pesar de esas diferencias, oferta y demanda están de acuerdo en la importancia de los cursos de programación/desarrollo y redes.

En cuanto a la tendencia futura de los conocimientos y habilidades requeridas en tres años, empresas y profesionales coinciden: éstos últimos deben contar con habilidades de razonamiento y solución a problemas específicos. Sin embargo, difieren en la importancia de otros elementos. Las empresas consideran más importante las habilidades en bases de datos, programación de sistema y en herramientas de programación, en tanto, los profesionales consideran relevante que los empleados cuenten con habilidades en comunicación oral y escrita, en sistemas operativos, inglés y en consultoría de negocios.

Respecto al uso de tecnologías de software, las empresas prefieren la especialización en WINDOWS (29.5%), AS/400<sup>18</sup> (27.4%), ASP (19.2%) y ACCESS (13.6%), por el contrario, los profesionales consideran importante la especialización en WINDOWS (41.2%), LINUX (34.1%) y UNIX (19.1%). Según el estudio, las empresas suelen dar mayor importancia a la generación de valor en el corto plazo, por lo que los profesionales deben ser capaces de resolver problemas prácticos en corto tiempo para ser rentables (como mantenimiento de sistemas y bases de datos).

---

<sup>18</sup> La importancia de que los profesionales se especialicen en AS/400 se debe a que hay una gran base de servidores instalados en México con esta tecnología.



## **2.3 ESTRATEGIA DE DESARROLLO NACIONAL Y REGIONAL PARA LA INDUSTRIA DE SOFTWARE**

### **2.3.1 Estrategia sectorial**

La industria de software se encuentra en una etapa incipiente de desarrollo. Sus problemas más importantes se acentúan en una limitada formación de recursos humanos especializados (como se mostró en el apartado 2.2.3), infraestructura inadecuada, ausencia de un marco legal, modelos de evaluación y certificación de procesos, entre otros. Ante ello, empresarios, instituciones de educación y del gobierno federal plantearon en conjunto las iniciativas que darían origen al Programa para el Desarrollo de la Industria del Software (PROSOFT<sup>19</sup>). Paralelo a estas iniciativas, una de las estrategias del gobierno federal en materia de TIC es posicionar a México como un país competitivo a nivel mundial en el sector de software, estrategia que se ha sistematizado a través de ese Programa.

El PROSOFT plantea los lineamientos de política industrial que debieran imperar para que la industria de software alcance niveles de competitividad local basada en el mercado interno, y competitividad internacional basada en las exportaciones. Los apoyos públicos a la industria de software están en esa dirección, principalmente a través del PROSOFT, en el cual se plantean dos estrategias generales. La primera es consolidar el mercado interno mediante la formación de recursos humanos calificados, crear un marco regulatorio y promotor de la industria, mejorar la infraestructura en materia de TIC y fomentar la conversión digital de procesos en diferentes industrias, así como incentivar la demanda de software a partir de las compras del sector público. La segunda estrategia consiste en aumentar la capacidad de exportación que en la actualidad es poco significativa (como se mostró en el apartado 2.2), es decir, para 2013 se pretenden alcanzar un monto de 5 mdd anuales entre exportaciones y ventas internas, equivalente al 1.5 % de la producción anual de software mundial, además de crear 100 mil empleos bien remunerados para profesionistas y técnicos (SE, 2005).

---

<sup>19</sup> El PROSOFT fue lanzado oficialmente el 9 de octubre de 2002.

El mercado de consumo más importante es el estadounidense y como objetivo se pretende cubrir una parte de su demanda de productos, tales como multimedia (animación 2D y 3D), gráficos, contenidos de Internet (trabajo en redes internas y externas), servicios de computo (prueba de software, por ejemplo), entre otros. Debido a la ubicación geográfica de México se considera llevar a cabo esta estrategia a través del esquema de *nearshore*.<sup>20</sup>

Se ha cuestionado el impacto que la IED ha tenido como estrategia de desarrollo en términos de la generación de capacidades tecnológicas locales, encadenamientos productivos hacia adelante y hacia atrás, desarrollo de proveedores, entre otros, por lo que no es claro el efecto que tendrá la IED sobre el sector de software mexicano.

Adicional a estas estrategias también se contempla: i) la educación y formación de personal competente en el desarrollo de software, en cantidad y calidad convenientes, ii) contar con un marco legal promotor de la industria, iii) desarrollar el mercado interno y fortalecer la industria local, iv) alcanzar niveles internacionales en capacidad de procesos, y v) promover la construcción de infraestructura básica y de telecomunicaciones.

Para lograr el cumplimiento de las estrategias, el PROSOFT ha obtenido apoyo de otros agentes tales como los Estados en los cuales opera el Programa, las empresas y la academia. Con base en Datos del Fondo PROSOFT, al 27 de noviembre de 2006,<sup>21</sup> el Fondo PROSOFT asignó 12.8 millones de dólares (mdd) a 10 entidades y 3 organismos de apoyo en 2004, para el siguiente año se incrementó a 17.7 mdd en 20 entidades y 3 organismos, y en 2006 la suma fue de 42 mdd para 25 entidades y 4 organismos. En el primer año de asignación de recursos del PROSOFT (2004), las aportaciones del Fondo representaron el 56% del total de la inversión, para el 2005 sólo el 26% y en el año 2006 representaron el 30%. Lo anterior se traduce en una amplia participación del resto de los agentes en la inversión total, la cual se concentró en el último año en proyectos productivos (31%),

---

<sup>20</sup> *Nearshore* es un esquema en el que una empresa contrata los servicios de software fuera del país donde se ubica, en un país geográficamente cercano. Algunas de las ventajas de este esquema es el trabajo en tiempo real y mismo uso-horario. El esquema opuesto es el de *offshore*, en el cual, por ejemplo, empresas transnacionales estadounidenses contratan esos servicios a empresas locales de la India. El principal consumidor de servicios offshore es Estados Unidos, con promedio anual de 9,150 millones de dólares (SE, 2006).

<sup>21</sup> <http://www.software.net.mx/anuario/>

innovación y desarrollo tecnológico (25%), fortalecimiento de la capacidad regional y empresarial (12%), capital humano (10%), y otros<sup>22</sup> (22%).

### 2.3.2 Estrategias locales

Paralelo a la estrategia sectorial se están realizando esfuerzos regionales con objetivos similares, pero fines distintos. Ruiz Durán (2004) describió y analizó los factores que potencian a las entidades para el desarrollo de núcleos de ‘economía digital’. Los estados de Nuevo León y Jalisco cuentan con programas formales focalizados en outsourcing<sup>23</sup> y en la creación de capacidades locales, respectivamente. Aguascalientes y Morelos son entidades que han promovido la creación de clusters de software. Por su lado, Puebla e Hidalgo han elaborado importantes esfuerzos con la industria vinculándola al programa *e-gobierno*. Otras Guanajuato, Sinaloa, Campeche y Yucatán han creado programas para acelerar el desarrollo de esta industria a través de empresas locales así como la formación de recursos humanos. Un instrumento importante para el desarrollo del sector en las regiones es la formación de *empresas integradoras*,<sup>24</sup> cuyo objetivo en el mediano plazo es, entre otros, ayudar a consolidar a la industria a nivel nacional, además de alcanzar niveles de certificación internacionales en capacidad de procesos y así buscar nichos de mercado externos, en particular el de Estados Unidos.

Algunas entidades como Jalisco, Nuevo León, Aguascalientes o Morelos han seguido, como estrategia para el desarrollo de software, el ‘user driven’, que se basa en el uso de SWFA como herramienta para fortalecer la infraestructura económica de tal manera que soporte el desarrollo de otros sectores de la economía. Una segunda estrategia llamada ‘supplier creating’, considera a la industria de software como sector prioritario, por la

---

<sup>22</sup> Otros incluye: i) calidad y capacidades de procesos, ii) acceso al financiamiento, iii) desarrollo de capacidades empresariales y de estrategia, iv) promoción y comercialización, y v) desarrollo de masa crítica en el sector.

<sup>23</sup> Se estima que el mercado potencial mundial de outsourcing actual es de 308 mil millones de dólares (SE, 2006).

<sup>24</sup> Una empresa integradora se define como una organización empresarial que asocia a micro, pequeña y medianas empresas formalmente constituidas con el objeto de prestar servicios especializados a sus socios tales como: (i) gestión de financiamiento, (ii) compra de manera conjunta de materias primas e insumos, (iii) vender de manera consolidada la producción. <http://www.contactopyme.gob.mx/integradoras>

generación de nuevos empleos y desarrollo y crecimiento industrial basado en la creación de oferta de software de exportación.

Jalisco y Nuevo León fueron los primeros estados en desarrollar programas formales para impulsar la industria de software, aunque los programas se implementaron cuando la industria ya estaba en funcionamiento, es decir, ya existía una serie de articulaciones económicas y productivas en torno a esta industria. Sin embargo, Guanajuato y el Distrito Federal a pesar de tener programas de apoyo a esta industria, aún están en proceso de consolidación.

Estas últimas cuatro entidades son importantes para el presente estudio por dos razones: primero, porque los estudios de caso de esta tesis se ubican en dichos estados y, segundo, porque tres de esas entidades son las más representativas en términos de existencia de empresas (D. F., Nuevo León y Jalisco concentran casi el 50% de las empresas de software respecto al total nacional<sup>25</sup>) y asignación de recursos del PROSOFT (en 2006, en el D. F., Nuevo León, Jalisco y Guanajuato se ubicó el 40.5% de la inversión total a la industria de software<sup>26</sup>).

### ***Jalisco***

Jalisco es una de las entidades federativas que combina ambas estrategias, user driven y supplier creating, para desarrollar su industria de software. El SWFA lo usa como herramienta para fortalecer la infraestructura económica, que será reemplazado gradualmente por la estrategia de desarrollo de software para exportación, pero basada en productos y no en maquila de procesos (Ruiz Durán, 2004).

---

<sup>25</sup> Con base en la encuesta realizada por ESANE Consultores, 2004 (citado por González, 2006).

<sup>26</sup> Con base en Datos del Fondo PROSOFT (SE, 2006), al 27 de noviembre de 2006, la inversión total fue de 1,408 millones de pesos. Este total incluye el monto asignado por la Secretaría de Economía, la aportación de cada una de las entidades con operación del PROSOFT, y las aportaciones de la Inversión Privada y de la Academia. En ese año, el D. F. es la primera vez que recibe apoyo del Fondo PROSOFT, las otras tres entidades lo reciben desde el primer año de asignación en 2004.

En el año 2001, tan sólo en Jalisco se ubicaban 151 micro, pequeñas y medianas empresas desarrolladoras de software, además de la empresa global IBM.<sup>27</sup> El promedio de empleados en cada empresa es de 3.3, mientras que IBM ocupó a 540 desarrolladores ese mismo año. La mayoría de las empresas en esa entidad son virtuales, solamente el 39.7 % de ellas contaban con establecimiento fijo (PECYTJAL, 2001-2007).

La industria de software en Jalisco se concentra en servicios de cómputo tradicionales como integración de sistemas, programación, entrenamiento, administración y mantenimiento de sistemas, renta de equipo de cómputo, *leasing* y *web hosting*. El procesamiento de datos, los servicios de red y los servicios electrónicos de información se encuentran en una etapa inicial (PECYTJAL, 2001-2007).

Además, esta entidad se posiciona en 4 grandes áreas de desarrollo de software. La primer área, ‘software empacado’, incluye: a) software de infraestructura de sistemas (*middleware –legacy systems*), b) paquetes de aplicación (software para empresas y negocios -vertical y horizontal-, software de consumo -multimedia<sup>28</sup>- y videojuegos -en plataforma de Internet -), c) herramientas de desarrollo (sistemas de administración de bases de datos orientadas a objetos y lenguajes de programación utilizados para escribir código para programas de software, incluyendo lenguajes clásicos como BASIC, C y PASCAL), y d) herramientas de desarrollo orientadas a objetos como C++ y JAVA.

Como segunda área de desarrollo figuran los ‘servicios de computo’, entre ellos el *software testing*. El mercado nacional de pruebas de software es del orden de 60 millones de dólares al año, concentrado básicamente en el sector financiero. Se estima que el mercado norteamericano atendido por empresas irlandesas es alrededor de 160 millones de dólares y podría ser transferido a México en el corto plazo a precios competitivos para irlandeses y mexicanos. Una empresa típica de software testing tiene alrededor de 100 testers, cada uno de éstos recibe en promedio \$10.00 dólares por hora, mientras que la hora hombre se

---

<sup>27</sup> A mediados de la década de los noventa, esta empresa desarrolló una fábrica de software en su planta ubicada en Salto, Jalisco, empleando a 500 desarrolladores e ingenieros de software (Ruiz Durán, 2004:39).

<sup>28</sup> La multimedia implica animación 2D y 3D, así como efectos especiales. La clave del desarrollo de multimedia está en el contenido, más que en las técnicas de programación.

factura en \$18.00 dólares. En China la hora hombre es de \$16.00 dólares pero no tienen la misma calidad que los desarrolladores mexicanos, y en la India se factura a \$18.00 dólares la hora- hombre pero no tienen la cercanía geográfica con que cuenta México (PECYTJAL, 2001-2007).

La tercera gran área son ‘servicios de información de negocios’, conocido como inteligencia de negocios. La cuarta área consiste en las ‘aplicaciones de Internet’ como *Business to Business* (administración de cadenas de valor y logística), área en la que el software a nivel internacional está migrando a través del trabajo en redes internas y externas.

### ***Nuevo León***

La industria de software de Nuevo León inició como resultado de las necesidades de empresas grandes tales como CONSORCIO FEMSA y de los grupos ALFA, CEMENTOS MEXICANOS, VITRO, BANORTE y SERFIN, entre otros. A las actividades de estas empresas se adhirió una estrategia local con el fin de aumentar el nivel de ocupación y certificar, con CMMI nivel 3, a la mayoría de las empresas de software de esta entidad. Tan sólo una de las empresas más importantes de México creada en Nuevo León, Softtek, emplea poco más de 2 mil personas (Ruiz Durán, 2004:34).

Siguiendo a este autor, en 2005 se estimaba entre 30 y 40 empresas de software en la entidad, y se considera que para el año 2010 haya más de 150 empresas y se generen más de 40 mil empleos directos.

### ***Guanajuato***

Esta entidad no tiene un programa formal para impulsar la industria de software, aún se encuentra en proceso de consolidación. La estrategia del gobierno de Guanajuato es consolidar a esta industria a través del fortalecimiento de las empresas locales y la formación de capital humano con conocimiento específico en determinadas tecnologías de software, además de fomentar la IED. En la entidad se encuentran empresas importantes

como MICROSOFT,<sup>29</sup> y el Centro de Investigación en Matemáticas (CIMAT). Se pretende que el Consejo de Ciencia y Tecnología del Estado de Guanajuato (CONCYTEG) sea el vinculador entre MICROSOFT y las universidades y empresas de software.

### *Distrito Federal*

A diferencia de las tres entidades anteriores, el Distrito Federal (D. F.) es el mercado más importante de TIC en el país, representa el 40 % del total nacional, ya que se estiman 700 empresas, es decir, una importante concentración de empresas de software en la entidad (Ruiz Durán, 2004:60). Varias empresas han conformado la Asociación Mexicana de la Industria de Tecnologías de Información (AMITI) y otras, como EMPEIRA, pretenden crear una integradora para competir con las empresas transnacionales por grandes proyectos nacionales e internacionales.

El gobierno del D. F. puso en marcha un parque tecnológico (inaugurado en noviembre de 2006) en el que se incluye al sector privado y de educación, basado en un esquema de incentivos fiscales, con el cual se pretende atraer un importante número de empresas de software. Aunque se define como un cluster de tecnología de información, el parque tecnológico es uno de los primeros resultados del Programa de Fomento para el Desarrollo de la Industria de Software en el D. F. con una participación inicial de 16 empresas de la industria. Además, el gobierno del D. F., en colaboración con universidades como la UNAM, planea la instalación de un Parque de Desarrollo Científico y Tecnológico para el año 2009, considerando el diseño y desarrollo de software como uno de los objetivos rectores.

## **2.4 CONCLUSIONES**

En este capítulo se han presentado los elementos más importantes de la estructura de la industria de software de los principales países emergentes exportadores a nivel internacional (India, Irlanda e Israel), así como la importancia de esta industria en naciones

---

<sup>29</sup> Esta empresa se estableció en Guanajuato a mediados de la década de los noventa. Entre los convenios que ha establecido con algunas universidades de la entidad se encuentra la entrega de licencias de software para su uso académico.

latinoamericanas como Brasil y Argentina. Comparado con los primeros países, México no ha alcanzado un nivel de desarrollo de la actividad exportadora, tampoco ha logrado un desarrollado del mercado interno como Brasil o Argentina. Aunque en México se han implementado políticas públicas de apoyo a la industria, el gobierno actúa relativamente tarde comparado con los primeros países mencionados, los cuales han desarrollado una política e instrumentos de apoyo a la industria desde hace más de dos décadas. En este sentido, la industria mexicana de software está en una etapa de desarrollo.

Durante el proceso de maduración esta industria se ha enfrentado a problemas de diversa naturaleza, entre los que se pueden resumir los siguientes puntos más importantes con base en diferentes estudios (AMITI, 2001; Sallstrom y Damuth, 2003; UAM, 2004; ITAM, 2005; González, 2006): i) problemas de financiamiento, fomento y apoyo, capacitación, gestión y marketing, también de vinculación e interacción con otros agentes tales como universidades, centros de investigación y capacitación; ii) problemas técnicos en relación a la sistematización, racionalización y administración del proceso de desarrollo de software, así como dificultades en la medición del desempeño y mejoramiento de la calidad de las actividades de desarrollo;<sup>30</sup> iii) falta de articulación entre la oferta y demanda de recursos humanos calificados de acuerdo a las especificidades de la industria; iv) baja penetración en otros sectores industriales, pues es débil la incorporación de la tecnología de software en productos de manufactura (embedded software), es decir, no es extensivo el uso y aplicación de productos y servicios de software en otras industrias diferentes a las de tecnologías de información y comunicación (TIC) y v) rápido cambio técnico en la industria, en particular las herramientas y aplicaciones de software.

Pero hay otros elementos de la industria mexicana de software que han sido poco estudiados, específicamente, la estructura divisional y funcional de las empresas, las dificultades de comunicación interna y externa, así como las fuentes de información y formas de explotación por parte de las empresas. Los estudios mencionados en este apartado no analizan estos problemas a nivel de empresa, los cuales son parte fundamental

---

<sup>30</sup> Estos problemas técnicos no son exclusivos de la industria de software nacional sino también internacional; véase Eischen (2002a y 2002b).



en esta tesis ya que se pretende analizar las diferentes interacciones entre las empresas de software y los agentes del entorno, y cómo éstas identifican e internalizan la información y conocimiento generados en el entorno para el desarrollo de nuevos productos.

Estos problemas han sido poco atendidos, y en su conjunto representan limitaciones al desarrollo y crecimiento de la industria de software nacional, en particular cuando una de las estrategias de política para el desarrollo de la industria considera la expansión del mercado interno y el aumento de las exportaciones. Cabe mencionar que para penetrar en el mercado internacional, es necesario que las empresas tengan certificados de calidad en sus procesos de desarrollo de software como CMMi, MoPROSOFT u otro, es decir, deben contar con mecanismos de sistematización e internalización de la información y conocimiento, lo cual les permitirá diseñar y desarrollar productos competitivos en precio, calidad y tiempos de entrega que el mercado internacional demanda.

## **PARTE II**

### **INTERFASE, CAPACIDAD DE ABSORCIÓN Y GENERACIÓN DE CONOCIMIENTO: REVISIÓN DE LA LITERATURA**

El objetivo de esta tesis es describir y analizar el proceso de construcción de interfaces entre empresas mexicanas que desarrollan software hecho a la medida y diversos agentes de su entorno, tales como empresas, universidades, gobiernos y comunidades de desarrolladores de software de fuente abierta. Se explicará cómo a partir de diferentes tipos de interfase las empresas identifican información y conocimiento existentes en el entorno, los cuales al ser incorporados en las empresas les permite reforzar sus capacidades de absorción y expresarlos en el diseño y desarrollo de nuevos productos, procesos y servicios.

En la Parte II se hace una revisión de la literatura y posteriormente se presenta el diseño de investigación. En los capítulos 3 y 4 se revisan tres cuerpos de literatura con los que se construye el marco conceptual. Específicamente en el capítulo 3 se revisa la literatura sobre el diseño de interfaces entre proveedores y usuarios de tecnología. En el 4 se muestra, por un lado, la literatura sobre capacidad de absorción y las fuentes y mecanismos a través de los cuales se incrementa, por otro, se examina la literatura referente a la generación de conocimiento. Al final del capítulo se presenta la estructura conceptual de esta tesis, para después, en el capítulo 5, describir el diseño y la estrategia de investigación y, finalmente, exponer los 6 estudios de casos (empresas estudiadas).

## **CAPÍTULO TRES**

### **INTERFASE COMO MECANISMO CRUCIAL PARA IDENTIFICAR INFORMACIÓN Y CONOCIMIENTO EXISTENTES EN EL ENTORNO**

#### **INTRODUCCIÓN**

En este capítulo se revisa la literatura referente al diseño de interfases entre proveedores y usuarios de tecnología en un esquema de paradigma tecno-económico. Una interfase se conceptualiza como “una relación entre dos agentes a través de la cual fluyen diferentes tipos de información” (Andersen, 1996:45). En un entorno con tecnologías estables y estandarizadas se crea conocimiento rutinario y esto origina interfases simples, en cambio, en un entorno de rápido cambio tecnológico con tecnologías nuevas emergentes, la información y conocimiento es abundante y origina interfases complejas, donde predominan las actividades de aprendizaje que dan origen a innovaciones y nuevo conocimiento.

El argumento de este capítulo es que la interfase entre proveedores y usuarios es un mecanismo importante para solucionar los problemas relacionados a las asimetrías cognitivas para el desarrollo de nuevos productos. En su primer apartado se hace referencia a las interacciones entre proveedores y usuarios de tecnología en el esquema de paradigma tecno-económico (PTE) con un enfoque micro. En el segundo, se muestran las dimensiones más importantes en las que ocurre la interacción proveedor-usuario (P-U). Este apartado sirve de antecedente para el siguiente, en el cual se define y se presentan dos principios para el diseño de interfases. En el cuarto apartado se describe el rol crucial de los usuarios como poseedores de una parte de información y conocimiento que son importantes en los procesos de innovación. Al final, se presenta una evaluación crítica del diseño de interfases y las conclusiones del capítulo.

### 3.1 INTERFASE Y PARADIGMA TECNO-ECONÓMICO (PTE) CON UN ENFOQUE MICRO

En el contexto de cambio tecnológico, un paradigma tecnológico es visto principalmente como un medio de coordinación entre productores de conocimiento científico dentro de una comunidad científica (Dosi, 1982). Mientras que un paradigma tecno-económico (PTE) significa coordinación entre los productores y usuarios de artefactos funcionalmente definidos que se intercambia en el mercado. El PTE muestra una definición de commodity<sup>1</sup>, un diseño o una interfase entre dos agentes (Andersen, 1991:121).<sup>2</sup>

Cuando Perez (1986, 2002 y 2003) habla de PTE se refiere a un cambio de ‘conciencia colectiva’ que se convierte en el ‘sentido común’ de ingenieros, gerentes e inversionistas para lograr la máxima eficiencia y la óptima práctica productiva. En un PTE las categorías que están en juego son, en un primer plano, la invención, la innovación y la difusión, en un segundo plano, las innovaciones incrementales y las innovaciones radicales y, en un tercer plano, los sistemas tecnológicos, las revoluciones tecnológicas y el cambio estructural.

#### 3.1.1 De la invención a la revolución tecnológica

En la base del PTE y de su consecuente revolución tecnológica están las categorías de invención, innovación y difusión. La *invención* de un producto o proceso ocurre en la esfera científico-técnica y puede permanecer allí para siempre. Mientras que la *innovación* es un hecho económico: la primera introducción comercial de una invención se traslada a la esfera técnico-económica como un hecho aislado (según el grado de apropiabilidad y el impacto que tenga sobre la competencia), donde su futuro estará determinado por el mercado. Lo importante es el proceso de adopción masiva, es decir, la *difusión* que transforma una invención en un fenómeno económico-social (Perez, 1986 y 2002).

---

<sup>1</sup> Un commodity es una mercancía o producto genérico, básico y sin gran diferenciación entre sus variedades. Su fabricación, disponibilidad y demanda es mundial, tiene un rango de precios internacional y no requiere de procesos tecnológicos complejos para su fabricación.

<sup>2</sup> El concepto de PTE difiere del concepto de Paradigma Tecnológico, éste 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 determinada por la coordinación entre los productores de conocimiento científico.

La innovación como parte esencial del cambio tecnológico tiene dos características fundamentales, las innovaciones incrementales e innovaciones radicales. Las primeras son mejoras sucesivas realizadas a los productos y procesos existentes. Desde el punto de vista económico, originan el aumento general de la productividad. Los aumentos en la eficiencia técnica, la productividad y la precisión de los procesos, los cambios en los productos para elevar su calidad o reducir su costo y ampliar la gama de sus posibles usos, caracterizan la dinámica evolutiva de toda tecnología. Esta lógica ha sido llamada por Dosi (1982) como ‘paradigma tecnológico’.

Por otro lado, las innovaciones radicales consisten en la introducción de un producto o proceso nuevo que dan vida a toda una industria y pueden significar una ruptura capaz de iniciar una nueva trayectoria tecnológica. Estas rupturas crecen en la medida en que la trayectoria precedente está llegando a su agotamiento, y su aparición puede ocurrir en cualquier momento y acortar el ciclo de vida de productos o procesos que sustituye (Perez, 2003:17-18).

La combinación de innovaciones radicales e incrementales, junto con las innovaciones organizacionales y administrativas que afectan a más de una empresa, pueden constituir un *sistema tecnológico*, el cual se define como constelaciones de innovaciones interrelacionadas técnica y económicamente que afectan a varias ramas del aparato productivo.<sup>3</sup> Una vez que se establece la lógica del sistema, se puede predecir una sucesión de nuevos productos y procesos, que individualmente pueden ser innovaciones radicales, pero en el conjunto del sistema se pueden considerar como un cambio incremental. Cada sistema tecnológico conjuga innovaciones en insumos, productos y procesos con innovaciones organizativas y gerenciales (Freeman y Perez, 1988).

---

<sup>3</sup> Para Carlsson y Stankiewicz (1991:93), un sistema tecnológico está definido como una red dinámica de agentes interactuando en un área económica-industrial específica, bajo una infraestructura institucional particular implicados en la generación, difusión y utilización de tecnologías. El sistema tecnológico está definido en términos de flujos de conocimiento más que en flujos de bienes y servicios ordinarios. Estos flujos consisten de redes dinámicas de conocimiento y competencias. Según estos autores, los elementos constitutivos del sistema tecnológico son a) redes de conocimiento y competencia y b) redes industriales y bloques de desarrollo.

Algunos cambios en el sistema tecnológico son tan extensos es sus efectos que logran tener una mayor influencia en el comportamiento de toda la economía. Un cambio de esta clase conlleva a muchos *clusters* de innovaciones radicales e incrementales y pueden incorporar eventualmente un número de sistemas tecnológicos. Al conjunto de sistemas tecnológicos que gradualmente crean las condiciones necesarias para la aparición de nuevos sistemas, los cuales siguen principios similares y cuentan con los mismos factores externos, se denomina *revolución tecnológica*. Ella es el núcleo generador de cambios masivos y fundamentales en el comportamiento de los agentes económicos, tiene la capacidad de transformar todo el aparato productivo; por lo que es, en realidad, un conjunto de sistemas tecnológicos con una dinámica común. Esta revolución conduce a *cambios estructurales* y está en la raíz de cada gran auge de la economía mundial (Perez, 2001:123).<sup>4</sup>

La difusión de cada revolución tecnológica es guiada por un PTE arraigado en la conciencia colectiva. Pero, para que una revolución tecnológica se difunda de una rama a otra y a escala mundial, se requiere algo más allá de la constatación de un nuevo potencial técnico. Esta autora sugiere como elemento ‘organizador del mecanismo selectivo y estructurador’ de cada PTE, un *insumo* o *vector de insumos* capaz de ejercer una influencia determinante en el comportamiento de la estructura de costos relativos (Perez, 1986). Este sería el vector de incorporación del nuevo paradigma al sentido común de ingenieros y gerentes.

### **3.1.2 Insumos clave y cambio de PTE**

A mediados del siglo XIX, el carbón y el transporte fueron los insumos clave, pero el acero los sustituyó a finales del mismo, ya que a partir de éste insumo crecieron las industrias de ingeniería pesada, mecánica, eléctrica y química. Posteriormente, el petróleo se convirtió en la materia prima más importante del paradigma tecnológico, que sustentó el modelo de producción en masa, desplegado plenamente después de la Segunda Guerra Mundial.

Siguiendo a Perez (1986, 2002 y 2003), aunque esos insumos ya existían, sus aplicaciones no. Lo relevante es la ruptura que se produce cuando se entrelazan lo técnico y lo

---

<sup>4</sup> Dos grandes revoluciones tecnológicas son la producción en serie que se cristalizó alrededor de 1910 y que llegó a su madurez en la década de los 60 y 70; y la informática que se ha estado difundiendo desde la década de los setenta (véase Freeman, 1988; Perez 1986, 2001 y 2003; Chandler y Cortada, 2002).

económico a través de una reducción del costo relativo del insumo o conjunto de insumos clave. Cuando el conjunto de tecnologías basadas en el uso del insumo clave han agotado su potencial para contribuir al aumento de la productividad, puede originar saltos tecnológicos debido a la incorporación de nuevas materias o de innovaciones radicales. En cada nuevo PTE, un insumo particular o grupo de insumos pueden ser descritos como ‘factores clave’ de ese paradigma.

A partir de la década de los 70 se empezó a conformar y difundir un nuevo PTE, es decir, ya no parece del sentido común usar intensivamente el insumo tradicional (petróleo) y los materiales derivados. Ahora, la organización ideal de producción es la que fusiona la administración, la producción y la comercialización en un solo sistema integrado, para producir de manera flexible un conjunto cambiante y variado de bienes y servicios intensivos en información. A partir de este nuevo paradigma, las ramas motrices del crecimiento serían el sector electrónico y de informática, impulsando una red de infraestructura de telecomunicaciones (Perez, 1986; Mowery, 1996; Cortada, 2002).

En este nuevo PTE, la electrónica es la tecnología que sirve como vector a la cual están subordinados la biotecnología, nuevos materiales, energía y otras.<sup>5</sup> El rasgo más relevante del nuevo paradigma es la tendencia a aumentar el contenido de ‘información’ en los productos más que contenidos de energéticos o de materiales. Esto surge del cambio radical en la estructura de costos relativos hacia el abaratamiento constante del potencial de manejo y transmisión de información. Un impacto inmediato de la penetración de la microelectrónica barata es su capacidad de insertarse en los productos tradicionales agregando una nueva trayectoria incremental con nuevos conceptos guía.

Así, un nuevo PTE emerge gradualmente como un nuevo ideal de organización productiva, tomando suficiente ventaja de los factores clave, los cuales llegan a ser más visibles en la estructura de costos relativos. Pero la constelación completa va más allá de los factores

---

<sup>5</sup> Las ramas motrices de la economía, entonces, son diferentes en cada auge económico, de forma que los polos de acumulación donde se concentran los mayores capitales son aquellos donde se produce el factor clave y aquellos donde se logra el mejor aprovechamiento de las ventajas que éste brinda para lograr un salto cuántico en la productividad (Perez, 1986; Freeman y Perez, 1988).

clave y del cambio técnico, de tal forma que brinda una reestructuración del sistema productivo global. Con base en Freeman y Perez (1988), la difusión de un recién PTE implica:

- a) Una emergente forma de ‘mejores prácticas’ en la organización de la empresa y a nivel de planta,
- b) Nuevas habilidades en la fuerza de trabajo que afectan directamente la calidad,
- c) Nuevos productos y negocios,
- d) Diferentes patrones en la localización de inversiones nacionales e internacionales, resultado del cambio en la estructura de costos relativos que transforma las ventajas comparativas,
- e) Inversiones en infraestructura diseñada para proveer apropiadas externalidades a través del sistema y facilitar el uso de nuevos productos y procesos,
- f) Una tendencia de las pequeñas empresas a ser innovadoras,
- g) Nuevos patrones de consumo de bienes y servicios y nuevos tipos de distribución y comportamiento del consumidor.

A estos puntos se agregan nuevas y múltiples formas de interacción entre los agentes, en particular entre productores de conocimiento (que se expresa en productos, procesos y servicios) y usuarios (empresas, individuos o comunidades de usuarios) del mismo, así como diversas dimensiones en la que ocurre la interacción, como veremos en el siguiente apartado.

### **3.2 LA INTERACCIÓN PROVEEDOR-USUARIO COMO ANTECEDENTE DE LA INTERFASE**

Andersen (1991 y 1996) enfatiza la coordinación entre productores y usuarios de conocimiento tecnológico. En este sentido, propone que el paradigma tecnológico no debe considerarse sólo como un medio de coordinación entre los productores de conocimiento científico sino también como medio de coordinación entre grupos de productores y usuarios de diferentes tipos de artefactos específicos, es decir, como especificación de interfases entre las dos partes, a esto último prefiere llamarlo PTE.



### 3.2.1 Interacción proveedor-usuario y sus diferentes dimensiones

El concepto de interfase no está muy distante de aquel sobre la interacción proveedor-usuario (P-U) definido por Lundvall (1985, 1988 y 1992). Este concepto resalta la importancia del aprendizaje por interacción de los agentes. Los proveedores aprenden por la práctica<sup>6</sup> y los usuarios por el uso<sup>7</sup>. La interacción P-U permite la retroalimentación de sus aprendizajes<sup>8</sup>. Según Lundvall, un proveedor se define como una empresa dedicada a la fabricación de bienes tangibles (maquinaria y equipo, insumos intermedios o productos finales) o intangibles (servicios, software), y usuario se considera a una empresa o consumidor individual que espera beneficiarse del uso de un producto o servicio. Von Hippel (2005:22) menciona que una empresa o individuo pueden tener diferentes relaciones para diferentes productos o innovaciones.

Siguiendo a estos autores, las empresas se vinculan por un conjunto de transacciones que establecen entre sí. Por ejemplo, la compra y venta de bienes de capital o de insumos intermedios supone relaciones entre las empresas e intercambio de algún tipo de información sobre precios y cantidades que se comercian,<sup>9</sup> pero lo relevante es el intercambio de información tecnológica, conocimientos, habilidades o *know how* específico sobre productos, procesos y patrones de organización productiva. Esta interacción supone aprendizaje por interacción entre las empresas, lo que aumenta el potencial innovativo y competitivo de ambas.

En una interacción de ese tipo, proveedores y usuarios pueden obtener beneficios. Los primeros se favorecen de las innovaciones de producto realizadas por el usuario, detectan las demandas potenciales de sus clientes y se apropian del conocimiento técnico adquirido por el usuario mediante el aprendizaje. Los segundos, por su parte, se benefician del trabajo en conjunto con su proveedor para mejorar la especificación de un producto y obtener mejoras en los procesos productivos, logran asesoramiento técnico, participan en la

---

<sup>6</sup> Por sus siglas en inglés: *learning by doing*.

<sup>7</sup> Por sus siglas en inglés: *learning by using*.

<sup>8</sup> Por sus siglas en inglés: *Learning by interacting*.

<sup>9</sup> Esta relación se configura como una de las primeras formas de interacción P-U.

solución de problemas durante el proceso productivo y mejoran la calidad y tiempos de entrega.

Para Lundvall (1988 y 1992), la interacción P-U es un prerequisite necesario para la innovación de producto, aunque no todas estas interacciones promueven la innovación en un mismo grado. La interacción P-U puede convertirse en un círculo virtuoso, pero si no es satisfactoria, es débil o poco frecuente puede volverse frágil y dar resultados negativos obstaculizando el aprendizaje y la innovación. En este sentido, el costo de vinculación puede ser alto o bajo.

Con base en este autor, la interacción P-U se basa en distintos tipos de relaciones que dependen de diversos espacios (económicos, culturales) o contextos nacionales e institucionales. Según él, las dimensiones más importantes en las cuales ocurre la interacción P-U son: i) las relaciones simétricas y asimétricas, ii) la estandarización, frecuencia y duración del intercambio, iii) el espacio económico y cultural, iv) el entorno organizacional, y v) los flujos de información.

***Relaciones simétricas y asimétricas.*** Una interacción *simétrica* se presenta cuando las empresas que interactúan presentan habilidades semejantes en cuanto a:

- a) Capacidad de acceder a fuentes de información técnica dentro y fuera de la empresa,
- b) Habilidades para establecer redes con otras empresas o instituciones,
- c) Grado de especialización y de dominio tecnológico sobre sus procesos y productos,
- d) Posición en la cadena de valor agregado, y
- e) Grado de internacionalización y cultura organizacional.

Cuando hay grandes diferencias, la colaboración es asimétrica y uno de los agentes domina la interacción, generando posiciones oportunistas que limitan la regularidad en el largo plazo (Corona y Hernández, 2000:761).

***Estandarización, frecuencia y duración del intercambio.*** Cuando el equipo ha alcanzado la madurez tecnológica, está muy estandarizado o es de muy baja complejidad tecnológica,

la interacción puede ser muy limitada, pues el mercado bastaría para proporcionarles la información requerida. En este caso, la frecuencia de la interacción (cantidad de información y contactos formales e informales) puede ser baja e incidir poco en la innovación. Por el contrario, si el producto es complejo y su diseño cambia rápidamente, la interacción puede ser de alta frecuencia y regularidad, lo que permite identificar las potencialidades del producto y reducir el esfuerzo de innovación.

**Espacio económico y cultural.** Las interacciones P-U selectivas implicarían unidades más o menos distantes entre ellas en un espacio geográfico y cultural. La importancia de la distancia variará con el tipo de actividades de innovación. Cuando la tecnología es estandarizada y relativamente estable, el intercambio de información puede ser trasladado en códigos estándares y la transmisión de información en grandes distancias tendrá bajos costos. Por tanto, la interacción P-U puede crearse entre unidades lejanas y ser efectivas. Pero si la tecnología es compleja y cambiante, la proximidad geográfica puede ser importante. Aquí los códigos de información pueden ser flexibles y complejos, y una experiencia cultural común puede ser importante para establecer códigos tácitos de conducta y facilitar la decodificación del intercambio de mensajes complejos. La necesidad de distancias cortas será reforzada cuando las necesidades de los usuarios sean complejas y cambiantes.

Un proveedor o un usuario localizados en espacios económicos diferentes, separados por barreras económicas, culturales o grandes distancias geográficas, podrían desarrollar habilidades distintas y beneficiar a la interacción. Según Corona y Hernández (2000:761), las tecnologías de información y comunicación (TIC) acortan las distancias permitiendo mayores flujos de información en forma de datos, voz, textos, diseños y diagramas.

**Dimensión organizacional.** El precio, la calidad, los tiempos de entrega y el servicio son factores cruciales en la interacción P-U. Esto sólo se logra si la organización es flexible y tiene capacidad de respuesta rápida ante las exigencias del cliente o proveedor. El cambio organizacional en el sistema productivo y administrativo afecta los procesos de aprendizaje

tecnológico, los flujos de información y la capacidad para producir a tiempo y con calidad en la empresa.

***Flujos de información.*** La innovación es la materialización de la información, la experiencia, el conocimiento y las habilidades que la empresa ha acumulado a lo largo del tiempo. El acceso a fuentes de información tecnológica por parte de las empresas incrementa su acervo de conocimiento y permite reducir el efecto negativo de riesgo e incertidumbre.

Las fuentes de información se clasifican en dos: internas y externas. Las internas, a su vez, se dividen en dos bloques: las que se originan en el trabajo incluyendo las opiniones de ingenieros y técnicos, y las que provienen de la experiencia y habilidades de los obreros. Las externas emanan de filiales, proveedores, clientes, de los centros de investigación públicos y privados, entre otros (Corona y Hernández, 2000; Dutrénit, *et al.*, 2002).

El flujo de información puede llevarse a cabo sólo si existen *canales de información* a través de los cuales pueden transitar los mensajes, es necesario un *código de información* a fin de lograr una efectiva transmisión de los mensajes. Además, el desarrollo de un código común requiere tiempo e implica *aprendizaje* entre los agentes implicados, de forma que el aprendizaje por interacción aumenta la efectividad de una serie de canales y códigos de información.

Como se ha descrito, Lundvall (1985, 1988 y 1992) resalta la importancia del aprendizaje por interacción y plantea las dimensiones más importantes en las cuales ocurre. A partir de estos elementos, Adersen (1991 y 1996) enfatiza la interacción entre productores y usuarios de conocimiento tecnológico bajo el esquema de PTE, como veremos en el siguiente apartado.

### **3.3 INTERFASE EN EL ESQUEMA DE PTE**

Andersen (1996:45) define interfase como una interacción entre dos agentes a través de la cual fluyen diferentes tipos de información. Si una interfase es simple 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 consiguiente, interfases complejas y no simples.

En Andersen (1991, 1996), el juego entre productores y usuarios tiene un rol fundamental en la composición de la definición de PTE.<sup>10</sup> Según este autor, un PTE es una especificación aceptada y estable de la interfase entre productores y usuarios de artefactos tecnológicos. Es decir, señala que “el PTE es visto como una definición agregada de la ‘interfase P-U’, la cual toma la forma de especificaciones aceptadas y estables entre productores y usuarios de artefactos tecnológicos. Las especificaciones entre productor y usuario de un producto pueden ser generalizadas y crear estructuras estables sobre las que se basa la relación de los agentes y en algún grado determina la mejora en el desempeño tecnológico de productores y usuarios...Los cambios radicales en las estructuras pueden romper con las especificaciones básicas y crear nuevas condiciones para los productores y usuarios.”

La noción de PTE cambia la perspectiva de una comunidad de científicos e ingenieros involucrados en la perfección y administración de una tecnología particular. El PTE se focaliza en la comunidad de productores y usuarios de un producto dado (*commodity/artefacto*) y su influencia en la tecnología relacionada, y no en la comunidad de científicos o ingenieros involucrados en el proceso de invención. La razón, dice este autor, es que los ingenieros ubicados en las empresas de los productores y de los usuarios no forman una comunidad homogénea, por eso prefiere hablar de paradigma tecno-económico en lugar de paradigma tecnológico.

El PTE puede ser considerado como medio para delimitar las necesidades de los flujos de información entre los agentes, por eso Andersen (1991 y 1996) lo define como una especificación estable y aceptada de las interfases. En este sentido, uno de los elementos de la interfase es la estabilidad que no se encuentra en el concepto de paradigma tecnológico acuñado por Dosi (1982). La estabilidad, dice Andersen (1996:47-48), se puede ver en

---

<sup>10</sup> Desde la perspectiva de este autor, el paradigma tecnológico, como lo acuñó Dosi (1982), debe tener como cambio central la inclusión explícita de la interacción entre productores y usuarios de mercancías.

términos del diseño básico de la mercancía a la cual la heurística positiva y negativa se adapta, y cambiar la interfase es difícil porque está arraigada en las normas y en la organización de las dos partes que componen la interfase. ¿Cómo y por qué la empresa y su entorno local llegan a estructurarse en un sentido relativamente sencillo reduciendo alguno de los problemas de información? Una de las respuestas de este autor es que el mundo se caracteriza por un nivel relativamente alto de reproducibilidad, de conocimiento rutinario y de interfases estables, lo cual hace posible que los agentes acumulen experiencia que puede funcionar como conocimiento re-utilizable. De este modo, los agentes pueden organizar el mundo tecno-económico en un sentido cognitivo.

A diferencia del concepto de PTE a nivel macro acuñado por Perez (1986, 2002 y 2003) y Freeman y Perez (1988), Andersen (1991, 1996) lo ve como un ‘PTE a nivel micro’ ya que enfatiza los aspectos inter-organizacionales. Desde esta óptica hace dos consideraciones:

- a) Primero, se pregunta cómo estudiar y organizar la información obtenida que es usada para la toma de decisiones de las empresas individuales en un mundo rico en información.
- b) Segundo, menciona que el PTE es visto como un medio para delimitar el flujo de información a través de la interfase.

Dadas estas consideraciones, Andersen (1991, 1996) discute dos principios relevantes en el diseño de interfases. Primero, el principio de ‘*estabilidad de la interfase*’ en la construcción de sistemas tecno-económicos, los cuales minimizan las necesidades de flujos de información en el sistema. Este principio, llamado originalmente *commodity abstraction*, es nombrado así ya que su traducción al español no da la idea sintética de lo que implica.<sup>11</sup> Por eso se traduce como ‘*estabilidad de la interfase*’. La idea fundamental de este principio es que las mercancías o commodities originan interfases simples y estables, con un alto grado de estabilidad en el diseño y precios. Segundo, el principio de ‘*aprendizaje por interacción*’<sup>12</sup> entre productores y usuarios, una forma de interacción necesaria en la

---

<sup>11</sup> Su traducción literal sería ‘abstracción de la mercancía’.

<sup>12</sup> Por sus siglas en inglés: *interactive learning*.

creación de nuevos productos que, al mismo tiempo, incrementa las necesidades de flujos de información.

### **3.3.1 Dos principios de interfase: ‘*estabilidad de la interfase*’ y ‘*aprendizaje por interacción*’**

En un sistema tecnológico complejo, cuando una gran parte de la fuerza de trabajo industrial está empleada en la acumulación y uso de conocimiento tecnológico, la *estabilidad* y las *interrelaciones* son importantes (Andersen, 1991, 1996), elementos considerados en los dos principios que discute este autor.

#### ***El principio de estabilidad de la interfase***

Bajo este principio, la validación y uso del conocimiento de una empresa depende en buena medida de un cierto grado de estabilidad del diseño y precio de los insumos y productos. Es decir, Andersen (1991 y 1996) menciona que el conocimiento de la empresa es parte de un sistema tecno-económico basado en una extensa división del trabajo entre diferentes unidades de conocimiento, un sistema donde no hay mucho lugar para las idiosincrasias y posibles innovaciones. Por eso las interfases simples y relativamente estables son de fundamental importancia para la funcionalidad del sistema económico, ya que bajo este principio se minimiza la necesidad de flujos de información.<sup>13</sup>

El supuesto del que parte este autor es que, en un mundo rico en información, los usuarios asumen que su conocimiento acumulado acerca de las características de un tipo de producto será útil para conocer los próximos productos. Este supuesto se formula como un principio de *estabilidad de la interfase*, el cual es fundamental para construir ciertos tipos de interfase entre diferentes partes del sistema: las fallas y problemas originados en el entorno pueden ser solucionados sólo a través de la interfase, y los cambios de una parte específica pueden ser logrados siempre y cuando no influyan en las características del producto definido por la interfase (Andersen, 1996:50).

---

<sup>13</sup> En palabras de Andersen (1991, 1996), el rompimiento de este principio es lo que caracteriza al empresario shumpeteriano.

Tan pronto como la interfase está organizada de acuerdo a este fundamento, es difícil hacer cambios. Primero, porque las rutinas de ambos agentes involucrados en la interfase son desarrolladas bajo el supuesto de una interfase estable, y hay pocas posibilidades de un acuerdo sobre una estrategia de cambio. Segundo, los flujos de información se minimizan y no son lo suficientemente largos para hacer cambios en la interfase, ya que la interacción entre los agentes está delimitada por un diseño estable del producto (Andersen, 1996:51).

Siguiendo a este autor, una tecnología puede ser madura y estandarizada, en tal caso una interfase puede ser creada, fijada y permanecer por mucho tiempo, como sucedió con las máquinas de escribir. Todos los diseños rivales incorporados al mercado fueron producidos años después con economías de escala, con pocas posibilidades de superar los efectos del aprendizaje logrado por las primeras máquinas. En este sentido, se crea un sendero cuasi-irreversible de mantenimiento y evolución. El problema no es, ciertamente, una cuestión de inflexibilidad técnica, pero sí muestra el lock-in de un diseño y un sendero seleccionado.<sup>14</sup>

Esas máquinas incluyen algunas especificaciones de su funcionalidad pero no del diseño. El usuario puede concebir la información sobre los materiales con los que está hecha la máquina de escribir, como una forma de aproximarse a las características de su diseño. Estas características explícitas definen un paradigma de acuerdo al cual las *partes* concretas de la máquina de escribir, así como los teclados de computadoras, son producidos y comprados sin mucha dificultad de comunicación. Pero ellos abstraen muchas de las características de cada *parte* única de las máquinas. Pocas de estas características pueden componer una interfase que le permita al usuario determinar fácilmente ciertas cuestiones generalizables. Esas cuestiones y las características en general presuponen un mundo estable organizado y estructurado de acuerdo al principio de *estabilidad de la interfase*, donde cualquiera puede considerar que un producto es bueno o irrelevante una vez que tiene información de sus características generales (Andersen, 1996:52).

---

<sup>14</sup> Ante un problema de inflexibilidad el mercado por sí mismo puede crear un *lock in* a una tecnología inferior. Arthur (1989) señala este problema en un escenario de largo plazo con diferentes formas de 'retorno' entre dos tecnologías posibles; menciona que los eventos poco significativos pueden llegar a ocasionar reacciones importantes en el mercado e inducir la selección de una tecnología inferior que presenta altas tasas de retorno. Las interfases 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.



La estabilidad, según Andersen (1991), 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 entorno donde las relaciones se simplifican. Por ello, este autor menciona que el mundo se caracteriza por un nivel alto de reproducibilidad de rutinas y de interfases estables, lo cual permite que los agentes acumulen experiencia convertible en conocimiento re-utilizable.

Según Fagenberg (1993), la importancia de interacciones estables puede variar a través de las industrias, sobre todo en aquellas caracterizadas por tecnologías complejas y específicas, donde la necesidad de comunicación cercana y el costo de establecer nuevas interacciones aumenta. Así, una interacción P-U estable puede ser interpretada como un mecanismo que reduce los costos de transacción y permite incrementar el aprendizaje, al mismo tiempo de internalizar los efectos externos positivos.

Como señala Hobday, (1996:15), en algunos casos los sistemas integrados muestran un alto grado de estabilidad a largo plazo (por ejemplo en simuladores de vuelo, telecomunicaciones y sistemas militares) debido a la emergencia de diseños estándares, pero en otros la estabilidad es lograda porque el ajuste industrial ocurre entre las empresas que integran la red de oferentes.

### ***El principio de aprendizaje por interacción***

El principio de *aprendizaje por interacción* entre productores y usuarios es una forma de interacción necesaria en la creación de nuevos productos, la cual, al mismo tiempo, incrementa las necesidades de flujos de información. Andersen (1996:53) menciona que en un sistema tecno-económico organizado de acuerdo al principio de *estabilidad de la interfase* hay poco lugar para la innovación. La razón: una tecnología estandarizada y pobre en información genera pocas posibilidades de innovación. Por eso considera a los nuevos productos como ‘pre-mercancías’, los cuales pueden ser transformados en mercancías reales, que no determina una solución estable y bien definida sino su construcción, lo cual requiere de un proceso complejo de depuración. Este proceso toma la forma de un sendero de desarrollo que puede ser altamente dependiente de algunas idiosincrasias en el diseño de interfases.

El segundo principio puede ser llamado *aprendizaje por interacción* en la innovación de producto y en la modificación del mismo. Como señala Andersen (1991), este fundamento puede ser visto como una generalización del ‘aprendizaje por práctica’ de Arrow (1962) o del ‘aprendizaje por uso’ de Rosenberg (1982). Hay un tipo de aprendizaje y acumulación de conocimiento logrado a través de prueba y error en el uso del producto, mismos que pueden ser encontrados en los usuarios y sus resultados pueden ser incorporados en los productos. En este sentido, los usuarios pueden constituir un sector oferente de conocimiento o de posibilidades tecnológicas para mejorarlos. Por esta razón una interfase exitosa entre P-U es una condición necesaria para que se logre el cambio técnico incorporado en la mejora de los productos (Anderser, 1996:54).

Bajo este principio la interfase no puede ser estabilizada como en el primero, ya que el *aprendizaje por interacción* contiene fuertes *vínculos y efectos* que no son del todo localizados en el intercambio económico de productos con diseños estables, al menos que se incluya información en común (Andersen, 1991). Por eso las interfases tienden a ser distintas y complejas.

En esa lógica, no todas las interfases son igual de importantes para el desarrollo del producto. Algunos usuarios toman un rol activo en los nuevos productos, otros no participan en el proceso de innovación pero recogen los beneficios de las mejoras sin jugar un rol activo en el proceso de innovación. Otros tienen que aceptar los productos estandarizados en el mercado aun si ellos no son adecuados para sus necesidades, entonces estos usuarios aceptan el principio de *estabilidad de la interfase* (Andersen, 1996:55).<sup>15</sup>

*Estabilidad de la interfase y aprendizaje por interacción* son dos principios opuestos para diseñar interfases P-U. La formulación de una estrategia tecnológica para una empresa depende de cuál de todas las interfases de insumo o producto en la empresa están funcionando de acuerdo a uno u otro principio (Andersen, 1991). El cuadro 3.1 resume las características de los dos principios y el tipo de interfase que se genera; se muestra cuan

---

<sup>15</sup> En el siguiente punto se describirá en detalle el papel que los usuarios han tomado como innovadores en un entorno de interfases estables y complejas.

importante es analizar los flujos de información a través de las interfases (los cuales dependen del grado de estandarización de la tecnología), los vínculos y efectos que generan esos flujos, así como el tipo de usuario que participa, ya que en algunos casos constituye un ‘sector oferente de conocimiento’ e información.

**Cuadro 3.1**  
**Características de los *dos principios* en el diseño de interfases**

<b>Principio</b>	<b>Características</b>	<b>Interfase generada</b>
<b>Estabilidad de la interfase</b>	a) Estabilidad del diseño b) Estabilidad en precios c) Se minimizan las necesidades de flujos de información (cortos) d) Rutinas e) Tecnologías maduras y estandarizadas f) Dificultad al cambio g) Conocimiento basado en una extensa división del trabajo h) Conocimiento re-usable i) Usuarios pasivos	<b><i>Interfase simple</i></b>
<b>Aprendizaje por interacción</b>	a) Diseños no-estables b) Precios oscilantes c) Se incrementa las necesidades de flujos de información (largos) d) Fuertes vínculos y efectos e) Tecnologías nuevas, emergentes f) Flexibilidad al cambio g) Conocimiento idiosincrásico h) Generación de nuevos productos i) Usuarios activos - Los usuarios constituyen un sector oferente de conocimiento k) Procesos de aprendizaje	<b><i>Interfase compleja</i></b>

Fuente: Elaboración propia, con base en Andersen (1991 y 1996).

Por otro lado, Andersen (1996:56) menciona que únicamente una parte de las relaciones potenciales entre las empresas puede desarrollarse a través de canales de información bien establecidos, y sólo una pequeña parte de los canales desarrollados son centrales para los procesos de innovación, es decir, pocos canales son eficientes para el intercambio de información en los procesos de innovación. De esto depende que la información sea transformada en ‘proyectos’ importantes para el desarrollo tecnológico de la empresa. De esta forma, el grado de estandarización de las interfases define si los canales de información establecidos son relevantes para cambiar la trayectoria de desarrollo de productos y procesos. Algunos estudios del ciclo de vida del producto pueden ser reinterpretados en esta perspectiva y mostrar cómo y por qué la importancia de los flujos de información entre productores y usuarios está cambiando en el tiempo (Andersen, 1991).

Las empresas que cuentan con canales de información bien establecidos tienen una ventaja comparativa en los procesos de innovación. Sin embargo, dice Andersen (1996:57), muchos usuarios nuevos no tienen la posibilidad ni la motivación para entender todas las consideraciones que están detrás del diseño de un producto. Para ellos, dice este autor, el producto tiende a ser una abstracción, un objeto con atributos definidos representados por un nombre, un precio y alguna información extra proporcionada por los vendedores (especificaciones del producto), en otras palabras, el producto tiende a ser un commodity. En esta situación, señala dicho autor, es posible establecer un diseño de producto dominante sobre un paradigma económico, lo cual implica estandarizar la interfase entre proveedores y usuarios, y al mismo tiempo se simplifican los canales de información desarrollados.

Es decir, los usuarios no influyen en la creación de nueva información para el diseño y desarrollo de nuevos productos, y el tipo de interfase será simple o estandarizado porque no se intercambia información diferente a la que proporciona el mercado, la que se intercambia se simplifica. Pero en un entorno rico en información para proveedores y usuarios, el mercado no es suficiente, habrá usuarios que requieren productos no estandarizados y por tanto se involucrarán en su diseño, de forma que la información requerirá canales diferentes para su intercambio y su correcto reconocimiento o asimilación por parte de los proveedores, originando interfases complejas. En este esquema, la información de los agentes, y en particular de las empresas usuarias, contiene elementos idiosincrásicos que permite al sistema basarse en asimetrías cognitivas. Es otras palabras, los procesos a través de los cuales se explica la conducta de los agentes serán diferentes para proveedores y usuarios, ya que tendrán formas diferentes de identificar, almacenar, recuperar, comprender y organizar la información recibida.

Entonces, el problema estará asociado a la forma en cómo los proveedores identifican y asimilan la información y el conocimiento de los usuarios. El conocimiento de la empresa es sólo una parte limitada de lo que existe en su medio ambiente, es sólo un componente del sistema de conocimiento tecno-económico basado en una división del trabajo entre diferentes poseedores de conocimiento. Por eso la importancia de las interfases.

Hasta aquí se ha descrito teóricamente el concepto de interfase entre proveedor y usuario de artefactos tecnológicos estables y complejos, así como los principios bajo los cuales se pueden diseñar. Estos tienen implicaciones en la forma de organizar la información y el conocimiento obtenidos (identificados) en un entorno rico en información y conocimiento. Como veremos en el próximo punto, un aspecto relevante para entender el concepto de interfase es conocer cómo se operacionaliza.

### **3.3.2 Operacionalizando el concepto de interfase**

En diversos estudios se ha intentado operacionalizar la idea de interfase a través de la interacción, por ejemplo, entre desarrolladores de proyecto (proveedores de sistemas) y operadores (usuarios de sistemas) (Geyer y Davies, 2000), entre diferentes sistemas tecnológicos o comunidades tecnológicas (Taminiau, 2006), o bien entre los aspectos sociales y ambientales (OECD, 2006).<sup>16</sup>

Geyer y Davies, (2000) analizaron dos tipos de interfase. La primera entre el *proyecto* de ferrocarriles y los *operadores* de ferrocarril (usuarios). En esta interfase los operadores se incorporaron en el proyecto en actividades como la definición y administración del proyecto, distribución de capacidades e integración dinámica de sistemas. La segunda interfase es entre el *sistema operacional* del ferrocarril y los *desarrolladores de proyectos* (oferentes). En esta interfase los desarrolladores del proyecto se vincularon con el lado de la operación del sistema de ferrocarril y se incorporaron en actividades como integración de capacidades de mantenimiento e integración de otras capacidades relacionadas a la red de ferrocarril. Este estudio reveló que, en la interfase, los desarrolladores de proyectos (oferentes) construyeron nuevas capacidades en el diseño, integración de sistemas y mantenimiento que fueron externalizadas por los usuarios (Geyer y Davies, 2000:1010).

---

<sup>16</sup> Desde la perspectiva de desarrollo sustentable, la OECD (2006) señala que la necesidad de analizar la interfase social-medioambiente es una de las prioridades de la organización para la primera década del siglo XXI. Sin definir claramente el concepto, señala que se deben entender los vínculos entre los aspectos sociales y ambientales, los efectos de la política ambiental sobre las comunidades más pobres, sobre la salud y el empleo, y las relaciones entre la política social y la política ambiental: cómo los países pueden asegurar que los costos y beneficios de las políticas ambientales sean compartidas y distribuidas rápidamente entre la sociedad.

Por su parte, Taminiou (2006) analizó las interfases entre dos comunidades tecnológicas (relacionadas a dos sistemas tecnológicos con dos tecnologías centrales), y cómo fue afectada por factores externos. Una de las comunidades está compuesta por ingenieros y técnicos de la industria del automóvil, la otra por ingenieros de la industria del petróleo. Por criterios de política ambiental europea, el objetivo era mejorar la calidad del aire en el año 2010 con tecnologías basadas en las normas europeas de 1996. Algunos de los logros fueron nuevas investigaciones sobre convertidores catalíticos para gasolina y diesel y mejoras en la calidad del combustible, pero sin llegar a una satisfacción completa de las demandas de ambas comunidades. La razón es que la interfase originó una incertidumbre tecnológica, impactada por la influencia de diferentes sistemas tecnológicos (automotriz-petróleo).

En este estudio, dice Taminiou (2006:255), las dos industrias nacieron por separado y llegaron a conectarse en el tiempo, debido principalmente a la intervención gubernamental y su relación al tema de salud y medio ambiente, más la dependencia tecnológica entre ellas no es igual. La industria automotriz depende mucho más de la calidad de los combustibles que provee la industria del petróleo, mientras que ésta no depende realmente de la industria automotriz para su proceso de refinación de los combustibles sino de la satisfacción de sus clientes.

En estos estudios se ilustran las interfases entre sistemas o comunidades tecnológicas y se muestra que, por un lado, hay una relativa autonomía entre los sistemas o agentes que intervienen y, por el otro, que los cambios pueden ser estimulados por fuerzas externas a la interfase (como la regulación gubernamental o las nuevas estructuras de mercado). Sin embargo, la interacción continúa como un proceso de adaptación tecnológica.

### **3.4 EL PAPEL DE LOS USUARIOS Y LAS COMUNIDADES EN LOS PROCESOS DE INNOVACIÓN**

Como se ha visto en los puntos anteriores, el concepto de interfase se asocia a una interacción entre proveedores y usuarios de artefactos tecnológicos (físicos e intangibles), a través de la cual se establecen códigos de comunicación y flujos de información. Como se

verá en este apartado, el usuario ha tomado un papel fundamental en este proceso. El trabajo seminal de Schmookler (1972) planteó que la demanda juega un rol importante en determinar la dirección y magnitud de la actividad innovativa. En su estudio encontró que las empresas usuarias de bienes de capital dirigían en buena medida la invención y la innovación de la industria de bienes de capital. Las premisas fueron, por un lado, que las habilidades para realizar invenciones son amplias, flexibles y responden a las oportunidades de obtener beneficios, y por otro, que los mercados actuales o potenciales son aquellos donde los beneficios derivados de la invención se incrementan con el tamaño de mercado. Para que los usuarios sean innovadores, menciona este autor, es importante que las industrias (como ocurrió con la química o electrónica) tengan acceso a una base de conocimiento científico y tecnológico en crecimiento en los campos de especialización. Así, la base de conocimiento y sus aplicaciones son adaptables a las necesidades de la demanda. Esta es la hipótesis de las ‘innovaciones originas por la demanda’ -*demand-pull*- (Scherer, 1982).

Hay varios estudios empíricos que han tratado de probar esa hipótesis (Scherer, 1982; Van den Ende y Dolfsma, 2004; Stadler, 2005; Von Hippel, 1988, 2000 y 2005; entre otros). Este último autor ha encontrado que un usuario líder puede innovar de la misma forma que el productor, y que el usuario es innovador tanto en mercados donde la tecnología y el producto son estandarizados (y producidos en masa), así como en mercados donde la tecnología y el producto son relativamente nuevos (y no producidos en masa).

Von Hippel (2005:13) señala que la habilidad de los usuarios para innovar está mejorando rápida y radicalmente como resultado de las constantes mejoras en la calidad de las computadoras, de los programas de software y en el acceso a herramientas de fácil uso y componentes para la innovación. Hoy en día, las empresas y usuarios individuales tienen acceso a herramientas de programación sofisticadas de software y a herramientas sofisticadas de diseño (CAD-CAM<sup>17</sup> por ejemplo).

---

<sup>17</sup> Diseño asistido por computadora (CAD: computer-aided design), y Manufactura asistida por computadora (CAM: computer-aided engineering)

Este autor ha estudiado las comunidades de desarrolladores de software de fuente abierta. Señala que los proyectos de software de fuente abierta (open-source) son lecciones que muestran a usuarios con habilidades para crear, producir, difundir, proveer apoyo a otros usuarios, actualizar y usar productos complejos por y para ellos mismos. Sin embargo, las economías de escala asociados con la producción y distribución de productos físicos da a los productores una ventaja sobre los usuarios.

En palabras de Von Hippel (2005:1, 33), los usuarios de productos y servicios (empresas y usuarios individuales) son capaces de innovar por sí mismas. Los procesos de innovación centrados en el usuario ofrecen grandes ventajas sobre aquellos centrados en los fabricantes, lo cual -desde el enfoque de cambio técnico- ha sido importante en términos teóricos y prácticos. Los usuarios que innovan pueden desarrollar lo que ellos desean y no depender de los fabricantes, además, los usuarios individuales no están obligados a crear todo lo que ellos necesitan, pueden beneficiarse de las innovaciones de otros usuarios y que son *libremente compartidas*. Los productores en masa tienden a seguir la estrategia de desarrollar productos que son diseñados para satisfacer las necesidades de un segmento de mercado grande y suficiente para inducir la compra y capturar beneficios significativos de un gran número de clientes. Pero, cuando las necesidades de los usuarios son diversas, esta estrategia de ‘pequeño tamaño ajustado a todos’ dejará a algunos usuarios insatisfechos con los productos ofrecidos.

Los procesos de innovación centrados en el usuario ilustran el contraste con el modelo tradicional. En éste modelo los productos y servicios son desarrollados por los fabricantes en una forma cerrada, usan patentes o copyright para evitar imitadores y proteger la inversión realizada. En este modelo tradicional el papel del usuario se limita a tener necesidades que los fabricantes identifican y cubren con nuevos diseños. No obstante, se han hecho varios trabajos empíricos donde se muestra que los usuarios son los primeros en desarrollar productos industriales y de consumo, actividad que aumenta como resultado de los continuos avances en la computación y comunicación (Von Hippel, 2005:2).



En ese sentido, el cambio en el proceso de innovación está orientándose a los usuarios como innovadores. Esto tiene algunas implicaciones directas en el corto plazo: facilita obtener precisamente lo que los usuarios necesitan, ajustado a sus necesidades y requerimientos, como en la industria de software.

### ***Usuarios en actividades de diseño y desarrollo de producto***

En los estudios realizados por Von Hippel (1988, 2000 y 2005) y Lakhani y Von Hippel (2000) se muestra que los usuarios ubicados en diferentes sectores se incorporan en el desarrollo o modificación de productos. Por ejemplo, en una de sus investigaciones Von Hippel (2005:4, 22) encontró que entre el 10 y 40% de los desarrollos son hechos por usuarios 'líderes'. Esto significa que los usuarios innovadores son el principal agente de un grupo de usuarios en un mercado determinado y esperan obtener altos beneficios por otorgar una solución a las necesidades que ellos han encontrado. Al mismo tiempo, muchas de las innovaciones reportadas por usuarios líderes son comercialmente atractivas, las cuales han sido comercializadas por los productores.

Pero, ¿por qué los usuarios modifican o desarrollan productos para su propio uso?. Este autor señala tres razones: i) porque los usuarios pueden innovar si desean algo que no está disponible en el mercado, ii) porque tienen las capacidades necesarias para lograrlo, y iii) porque tienen los recursos financieros para hacerlo.

### ***Decisión de los usuarios: comprar o innovar***

Los fabricantes se especializan en desarrollar productos para un grupo de usuarios homogéneos. Dado que estas empresas son especialistas en su producto, es posible que puedan diseñar y construir productos para usuarios individuales o empresas usuarias en menor tiempo, con calidad o más baratos que si los usuarios lo hicieran por sí mismos. Pero, varios factores pueden inducir a los usuarios a innovar y a no comprar (Von Hippel, 2005:6). El usuario puede pagar al productor por desarrollar un producto, en este caso el usuario es el principal agente y el productor el desarrollador, pero si los intereses de ambos son diferentes incurrirán en costos de gestión:

- a) Costos por monitorear al productor para asegurar que seguirá los intereses del usuario,
- b) Costos en los que incurre el productor si los intereses del usuario no se realizan de acuerdo a las especificaciones de éste, y
- c) Costos asociados con el resultado que podría o no servir a los intereses del usuario.

El usuario desea obtener precisamente lo que necesita, al grado que él mismo se puede esforzar por desarrollar un producto, por otro lado, el productor desea reducir los costos de desarrollo al incorporar soluciones que ya alguna vez ha usado o que predictivamente otros usuarios podrían usar en el futuro. Este dilema se convierte en un modelo de elección: innovar o comprar es la decisión que deberá tomar el usuario. En este modelo, los usuarios con necesidades únicas cuentan con una mejor visión para desarrollar el producto por ellos mismos, mientras que para los productores, el desarrollo puede ser más económico siempre y cuando el número de usuarios sea considerable y que éstos deseen el mismo producto en un futuro cercano.

Si tomamos en cuenta a la industria de software, en particular al segmento de software hecho a la medida, el argumento anterior pareciera ser contradictorio. En el capítulo 6 se mostrará que, por un lado, en este segmento hay usuarios con necesidades particulares, algunos de ellos a veces no pueden expresarlas adecuadamente a los desarrolladores de software, pero otros pueden tener conocimiento de cómo elaborarlo pero no lo hacen por varias consideraciones (económicas, de tiempo, por estrategia productiva). Por otro lado, los productores, aún con una baja expectativa acerca del número de usuarios que puedan usar el mismo producto, se involucran en el proyecto. Mientras que el usuario obtiene lo que necesita, el productor, en función de las necesidades particulares de cada usuario, puede diversificar sus productos buscando la rentabilidad en la 'reutilización de ciertos procesos de desarrollo de software' (como se retoma en el capítulo 6).

### ***Innovación a partir de las necesidades de los usuarios***

Usuarios y productores tienden a desarrollar diferentes productos y servicios debido, en parte, a las asimetrías y a que ambos tienden a conocer diferentes cosas. Los

desarrolladores de producto necesitan dos tipos de información para hacer exitoso su trabajo: características de las necesidades de los usuarios e información de contexto (generado por los usuarios), e información de soluciones genéricas (generado en ocasiones por productores especialistas). Generalmente los usuarios tienen una mayor certeza y detalle de sus necesidades, sin embargo, los productores cuentan con un mejor modelo de soluciones y se especializan en este campo (Von Hippel, 2005:8).

Una consecuencia de las asimetrías en la información entre productores y usuarios consiste en que éstos últimos tienden a desarrollar productos funcionalmente novedosos, y para ello requieren una gran asociación de la información acerca de sus necesidades e información de contexto para su desarrollo. Por su lado, los productores tienden a realizar innovaciones a partir de las necesidades de los usuarios, las cuales son conocidas y requieren una clara sistematización de la información acerca de las soluciones para su desarrollo (Von Hippel, 2005:70); en consecuencia, requieren de un proceso adecuado de identificación e internalización de esa información para poder cubrir las necesidades de los usuarios.

### ***Innovación hecha por los usuarios y acceso libre a la información***

La eficiencia social de un sistema en el cual las innovaciones individuales son desarrolladas por usuarios individuales se incrementa si éstos difunden lo que han creado. Los productores logran esto parcialmente cuando venden el producto o servicio en el mercado abierto (difunden el producto en el cual se incorpora la innovación, pero no toda la información de modo que otros puedan replicarlo). Si los usuarios no llegasen a difundir lo que desarrollan, otros con las mismas necesidades están obligados a desarrollar los productos independientemente y la innovación sería muy semejante.

Von Hippel (1988, 2000, 2005) ha mostrado que los usuarios revelan libremente lo desarrollado por ellos. Esto significa que todos los derechos de propiedad intelectual son otorgados por el innovador, y todas las partes interesadas podrán acceder a esa información y conocimiento, convirtiéndolos en un bien público. La práctica más visible de esto es el

desarrollo de software de fuente abierta, aun haciendo desarrollos con recursos privados (Lakhani y Von Hippel, 2000; Von Hippel, 2005).<sup>18</sup>

### ***Comunidades de innovación***

Taminiau (2006:260) define una *comunidad tecnológica* como ‘una serie de organizaciones que se involucran en una tecnología en particular’, comparten conocimiento común respecto a una interfase establecida y se pueden integrar otros agentes tratando de influenciar el desarrollo de la tecnología. Para Von Hippel (2005:11,93,96), en las comunidades de innovación hay una *cooperación organizada* a través de redes que proveen estructuras útiles y herramientas para la interacción y distribución de las innovaciones, a través de ellas se pueden incrementar la velocidad y efectividad con la cual los usuarios y productores desarrollan, prueban y difunden sus innovaciones.

Siguiendo a este autor, las *comunidades de desarrolladores* se definen como “nodos que consisten de individuos o empresas interconectadas a través de vínculos donde hay transferencia de información, puede implicar comunicación cara a cara, electrónica o de otro tipo”. La comunidad a veces se entiende como una red de interacciones personales que proveen sociabilidad, apoyo, información, un sentido de pertenencia e identidad social (Wellman *et. al*, 2002:4<sup>19</sup>). Según Davenport y Prusak (1998:31), en las comunidades hay al menos cuatro factores claves de trabajo: reciprocidad, reputación, altruismo y confianza, donde ésta última es el corazón del intercambio de conocimiento.

En la industria de software, las comunidades de innovación pueden tener usuarios o desarrolladores como miembros. Pueden florecer, según Von Hippel (2005:96), cuando una innovación es revelada libremente y cuando otros encuentran la información interesante. En consecuencia, estas comunidades parecen ser eficientes en la difusión de soluciones a problemas específicos de los agentes de la comunidad.

---

<sup>18</sup> En un modelo de inversión privado, la sociedad financia y protege a los innovadores mediante patentes, copyrights y leyes de secretos industriales, lo que les permite obtener retornos privados con base en las inversiones hechas. En cierta forma se crea un mecanismo de control sobre la información y conocimiento. Por el contrario, en una comunidad de ‘acción colectiva’ se minimiza ese control y se evita la pérdida social relacionada al conocimiento.

<sup>19</sup> Citado por Von Hippel, 2005:96.

El INTERNET ha fomentado la creación de comunidades de desarrolladores de software y, al mismo tiempo, el desarrollo de proyectos basados en software de fuente abierta. Los innovadores pueden difundir, a través del correo electrónico, información acerca de sus programas de software nuevos, incluso los programas a costos mínimos. Por su parte, los adoptadores de esos programas pueden acceder a la información a bajo precio debido a que se transmite en forma legible a través del INTERNET.

### **3.5 EVALUACIÓN CRÍTICA DEL DISEÑO DE INTERFASES**

La literatura sobre interfase provee una estructura analítica para entender el diseño de interfases entre proveedores y usuarios de productos en entornos tecnológicos estables y dinámicos, en particular para el segmento industrial de software hecho a la medida. La pertinencia de esta literatura radica en que el concepto de interfase es envolvente de otros como el de proveedor-usuario. Las principales contribuciones de esta literatura son:

- a) Se ha mencionado que en la raíz de cada PTE se encuentra la invención, la innovación y la difusión, y en cada una de estas categorías hay diferentes formas de interacción entre diversos agentes. Mientras que en la invención la interacción principal se da entre los productores de conocimiento científico dentro de comunidades científicas, la innovación, sólo en la esfera técnico-económica y a través de la difusión que sirve como facilitador, puede contener diferentes formas de interacción que no están exclusivamente dentro del ámbito científico sino en el económico-social.
- b) Como se ha mostrado, la innovación es un proceso interactivo que puede presentar problemas teóricos y empíricos. Lundvall (1985, 1988 y 1992) estudia este proceso interactivo con diferentes niveles de agregación, primero a nivel microeconómico, y sobre estos fundamentos explica conceptualmente el Sistema Nacional de Innovación. Andersen (1991 y 1996) conceptualiza la interacción P-U bajo un esquema de PTE a nivel micro, a la cual le llama interfase y la define como una interacción entre dos agentes a través de la cual fluyen diferentes tipos de información.
- c) Las interfases son de diversa naturaleza y su diseño depende del grado de madurez del sistema tecnológico al que se asocie. En un entorno de industrias maduras y

tecnologías estandarizadas se genera conocimiento rutinario y la interfase tiende a ser simple, de modo que la información para cada uno de los agentes será delimitada. Por el contrario, cuando el entorno es dinámico, de rápido cambio tecnológico y las tecnologías no han alcanzado su madurez o estandarización, el proceso innovativo presupone una interacción rica en información y, por ello, presupondría interfases complejas.

- d) El usuario ha tomado un papel importante en los procesos de innovación en industrias donde la información y conocimiento son el insumo principal (como en la industria de software), pero también donde las tecnologías son estandarizadas y la información y conocimiento acerca de su fabricación son del ‘sentido común’. Los usuarios pueden no aceptar las interfases establecidas cuando sus necesidades ya no son cubiertas por los productos que el mercado proporciona. En este sentido, el usuario se vuelve re-activo ante la insuficiencia de la interfase para proporcionar información adecuada acerca de nuevos productos y usos del mismo.
- e) Diversos estudios empíricos (Von Hippel; 1988, 2000 y 2005; y Lakhani y Von Piel, 2000) han mostrado que en algunas industrias los usuarios no actúan aislados, sino en ‘comunidades de innovación’, en principio, porque los usuarios individuales carecen de mecanismos para evaluar, asimilar y utilizar adecuadamente la información y el conocimiento útil del entorno. Las comunidades son una especie de interfase que sirven como filtro de la información y conocimiento del entorno, y la constante interacción les permite depurar estos insumos hasta crear nuevo conocimiento. En algunas comunidades el usuario se vuelve activo y puede llegar a definir una trayectoria tecnológica, como sucede en las ‘comunidades de desarrolladores de software de fuente abierta’.

Esta tesis se desarrolla sobre las contribuciones mencionadas en estos incisos. Sin embargo, como se mostrará en los capítulos 6-8, en la industria mexicana de software las interfases son cruciales para identificar, valorar la información y conocimiento existentes en el entorno y que son generados a través de las interfases, en particular para la micro y pequeña empresa que no cuenta con departamentos funcionales formalmente. A pesar de la pertinencia del concepto de interfase, como se define en esta literatura, es necesario

considerar otros aspectos para interpretar una de las realidades que muestra la industria mexicana de software. Por eso es relevante considerar lo siguiente:

- Primero, es necesario considerar diferentes tipos de interacción entre múltiples agentes, pues las empresas de software interactúan con usuarios y al mismo tiempo con universidades o centros de capacitación para el desarrollo de productos, procesos y servicios.
- Segundo, la interfase debe ser vista como un mecanismo que permite la combinación de diferentes tipos de información y conocimiento fragmentarios que posee cada agente. En el segmento de software hecho a la medida cada usuario tiene necesidades únicas que sólo se pueden expresar a través de la interfase y, dado que es hecho a la medida, el productor sólo podrá tener información y conocimiento de las especificidades del usuario a través de la interfase.
- Tercero, la interfase debe ser vista también como un mecanismo que permite identificar y valorar información y conocimiento útiles a las necesidades de las empresas. Para productos estandarizados el entorno codificado<sup>20</sup> podrá ser la fuente de información y conocimiento, pero para productos con un alto grado de especificidad de los usuarios, la interfase es el mecanismo crucial para identificarlos. Por esta razón la interfase entre proveedores y usuarios es un espacio donde se puede resolver los problemas relacionados a las asimetrías cognitivas para el desarrollo de nuevos productos.
- Finalmente, como veremos en el próximo capítulo, la interfase se distingue en internas y externas, ya sea porque se dan al interior de la empresa o en el entorno. La interfase externa es un mecanismo crucial para que las empresas puedan identificar la información y conocimiento generados en el entorno, y en consecuencia para aumentar su capacidad de absorción.

---

<sup>20</sup> Un *entorno codificado* es un entorno en el que se identifica información y conocimiento en forma de códigos que son transformados en lenguajes convenidos y preestablecidos para su comunicación.

## CAPÍTULO CUATRO

### CAPACIDAD DE ABSORCIÓN Y GENERACIÓN DE CONOCIMIENTO

#### INTRODUCCIÓN

En este capítulo se revisa el cuerpo de literatura referente a la capacidad de absorción y su interrelación con el concepto de interfase. La capacidad de absorción se define como aquella capacidad de la empresa para evaluar información y conocimiento útil del entorno y, a su vez, asimilar y expresarlos en nuevos productos, servicios o procesos. Autores como Cohen y Levithal (1990), Van den Bosch, et al (1999, 2002), Lane, *et al.* (2002), Lenox y King (2003), Lund Vinding (2004), argumentan que algunas fuentes y mecanismos por los cuales las empresas pueden aumentar su capacidad de absorción son el conocimiento previo, asociado a las actividades particulares de la empresa, la capacitación, las actividades de aprendizaje, las interacciones entre las diferentes áreas de la empresa, denominadas interfases internas<sup>1</sup>, y las interacciones entre la empresa y su entorno (denominadas interfases externas en esta tesis). Como se mostrará en el capítulo 7, la realidad de la industria mexicana de software revela que las interfases externas son un mecanismo crucial de aprendizaje que influye en el incremento de la capacidad de absorción, ya que la estructura de la micro y pequeña empresa no permite establecer espacios para profundizar en los flujos internos y establecer interfases internas.

Dentro del primer apartado se examina la literatura de capacidad de absorción y se analizan los trabajos que abordan la relación entre la creación de nuevo conocimiento y los procesos de aprendizaje. La literatura sobre aprendizaje ha mostrado que éste ocurre a nivel individual y a nivel organizacional. A este último nivel es importante internalizar la información y conocimiento, necesario contar con procesos de socialización de forma que el conocimiento generado no se pierda ni se olvide (Dodgson, 1993; Vera-Cruz, 2004). Cohen y Levithal (1990) y Van den Bosch, *et al.* (1999 y 2002) señalan que las interfases internas son importantes para este propósito. Pero en un entorno de rápido cambio

---

<sup>1</sup> El término utilizado en inglés es *Interface Cross-Function*.



tecnológico, como es el caso de la industria de software, las empresas deben acelerar sus procesos de internalización, por tanto requieren de mecanismos adecuados para internalizar información y conocimiento existentes en el entorno y aquellos que se generan a partir de las interfases con los diversos agentes.

En el segundo apartado se explora la literatura que estudia la relevancia de los diferentes tipos de información y conocimiento. Autores como Senker y Faulkner (1996) han analizado los tipos de conocimiento (tácito y codificado) involucrados en los procesos innovativos, donde el conocimiento tácito es relevante y su apropiación se puede ver como un proceso de innovación colectiva. En relación a esto, Andersen (1991) argumenta que la innovación está condicionada por el diseño de *interfases* entre proveedores y usuarios de tecnología, en donde los diferentes tipos de información se pueden simplificar y estandarizar. En este apartado se discutirá la utilidad de ambos conceptos (información y conocimiento) en el proceso de innovación de la industria de software, así como su intrínseca relación con la creación de interfases y la capacidad de absorción.

En el tercer apartado se presenta la estructura conceptual derivada de las preguntas de investigación así como de la revisión de literatura hecha en los capítulos 3 y 4. Finalmente, se presentan las conclusiones.

#### **4.1 CAPACIDADES DE ABSORCIÓN**

Las empresas absorben y generan conocimiento que lo expresan en nuevos productos, servicios o procesos para mantenerse en el mercado, ser innovadores y estar adelante de sus competidores o para mover la frontera tecnológica. En un ambiente tecnológico dinámico las empresas deben generar capacidades para identificar e internalizar información y conocimiento existentes en el entorno y que son generados a partir de las interfases con los diversos agentes. Los primeros autores en explicar cómo las empresas absorben información son Cohen y Levinthal (1990), quienes señalan que las empresas deben contar con conocimiento previo para generar la ‘habilidad de evaluar la información y conocimiento del entorno, asimilarlos y explotarlos con fines comerciales’.

El concepto de capacidad de absorción (CA) tiene varios niveles de análisis. En el primero, a nivel individual, tienen relevancia el aprendizaje individual y la diversidad del conocimiento que facilita el proceso de innovación habilitando a personas para hacer actividades novedosas y vinculaciones con otros sujetos. El segundo es de tipo organizacional (Cohen y Levinthal, 1990:131), entre otros relevantes como son a nivel industrial, cluster, región y país (Van den Bosch, *et al.*, 2002:7). Actualmente algunos estudios han relacionado diferentes niveles de análisis, por ejemplo, De Fuentes (2007) identificó las relaciones entre las derramas de conocimiento de empresas grandes y las capacidades de absorción de las pequeñas y medianas empresas de maquinados industriales en una localidad específica. Esta autora encontró que la CA está determinada por: la formación y experiencia del propietario y empleados, la tecnología incorporada a los equipos, las capacidades organizacionales, las actividades de innovación y aprendizaje, y los vínculos establecidos con otros agentes de la localidad. En esta tesis, el concepto de CA se analizará particularmente a nivel organizacional.

La capacidad de absorción puede ser incrementada por las empresas a través de la acumulación de experiencias (conocimiento específico a la empresa). Esto indica que las empresas deberán tener capacidades internas para asimilar y utilizar efectivamente la información y conocimiento del entorno. La literatura sobre aprendizaje y acumulación de capacidades tecnológicas ha mostrado que los procesos de aprendizaje y el desarrollo de capacidades internas permite a las empresas mejorar su productividad e innovar a nivel de producto, proceso y organización a lo largo del tiempo (Maxwell, 1981; Bell, 1984; Bell y Pavitt, 1993 y 1995; Lall, 1992 y 2000; Hobday, 1995, 2000, 2001; Dodgson, 1993).

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 (Dutrénit, 2000; Dutrénit, *et al.*, 2002; Vera-Cruz, 2004).

Los procesos de acumulación tecnológica y cambio técnico difieren fundamentalmente entre sectores. El cambio técnico adopta procesos tecnológicos representados en un modelo lineal de innovación-difusión. Sin embargo, la difusión implica más que la adquisición de maquinaria o diseño de productos o la asimilación de know how, también implica cambio técnico por el cual la innovación original es, en primer término, adaptada a condiciones particulares y, en segundo, mejorada para obtener un mayor desempeño del originalmente alcanzado (Bell y Pavitt, 1993).

Siguiendo a estos autores, el cambio técnico tiende a ser visto como dos actividades principales: primero, como el desarrollo y comercialización inicial de innovaciones, y segundo, como la aplicación de éstas en un proceso descrito como difusión. La acumulación de capacidades implica, entonces, más que el cúmulo 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.

#### **4.1.1 Dimensiones conceptuales y mecanismos de la capacidad de absorción**

Cohen y Levinthal (1990:128) señalan que las empresas deben tener la habilidad para explotar el conocimiento externo, pues es un componente crítico para las capacidades de innovación. Uno de los componentes a través del cual pueden lograrlo es el *conocimiento previo* asociado a las actividades particulares de la empresa, a partir de él, ésta desarrolla las habilidades y permite reconocer el valor de nueva información, asimilar y explotarla con fines comerciales.

##### ***Conocimiento previo***

El *conocimiento previo* es uno de los componentes por los cuales las empresas pueden aumentar su CA, permite tanto la asimilación de nuevo conocimiento como su utilización creativa para desarrollar uno nuevo. Siguiendo a Cohen y Levinthal (1990:136), el conocimiento previo tiene dos implicaciones en la CA que afectan el desempeño innovativo

en un entorno en evolución. Primero, la acumulación de CA en un periodo permite otra más eficaz en el siguiente y explotar cualquier conocimiento externo que pueda estar disponible. Segundo, el cúmulo de experiencias permitirá a la empresa entender y evaluar la importancia de los avances tecnológicos y formarse expectativas para el desarrollo de nuevos productos. Las expectativas pueden condicionar el incentivo a invertir en CA, es decir, si la empresa cesa de invertir en CA en un campo que se mueve rápidamente, no podrá ni asimilar ni explotar nueva información, y cuando aparecen las oportunidades no podrá apreciarlas adecuadamente.

Van den Bosch, *et al.* (1999:553,558) consideran dos entornos de conocimiento con implicaciones en la CA, uno estable y otro turbulento. En un entorno de conocimiento estable, por ejemplo en una industria madura, hay empresas que estarán más enfocadas en la *explotación* de la base de conocimiento y tenderán a ser más *re-activas*, probablemente no requerirán un alto nivel de CA, y serán menos *pro-activas* para explorar oportunidades fuera del entorno de conocimiento existente. Si la información y el conocimiento externo están relacionados a las actividades actuales de la empresa, podrán ser asimilados sin muchas dificultades y no necesitarán de sistemas complejos de procesamiento. Mientras que en un entorno de conocimiento turbulento, es probable que las empresas dediquen una buena proporción de recursos para aumentar su CA, la cual puede estar focalizada en la *exploración* de nuevo conocimiento, es decir, serán más *pro-activas* para buscar nuevas oportunidades y, dado que la información y el conocimiento no están relacionados con las actividades actuales de la empresa, necesitarán de conocimiento previo para poder *internalizarlos* y procesarlos.

### ***Aprendizaje inter-organizacional***

Bajo el supuesto de que una empresa aprende de otra, Lane y Lubatkin (1998) y Lane, *et al.* (2002) muestran que la habilidad de una empresa (estudiante) para aprender de otra (maestra) está determinada por las características relativamente comunes entre ambas. La empresa estudiante debe tener la habilidad para evaluar, asimilar y aplicar nuevo conocimiento de una alianza de aprendizaje que depende de:

- a) Tipo específico de nuevo conocimiento ofrecido por la empresa maestra,

- b) Las similitudes de las prácticas de compensación entre la empresa estudiante y la empresa maestra y la estructura organizacional, y
- c) La familiaridad de la empresa estudiante con la serie de problemas organizacionales de la empresa maestra.

Una vez que la empresa estudiante tiene conocimiento previo necesario para reconocer el conocimiento externo, el siguiente reto es cómo internalizarlo. Lane y Lubatkin (1998:465) argumentan que una forma de hacerlo es a través del aprendizaje inter-organizacional, lo cual implica que la empresa estudiante debe contar con un sistema de procesamiento de conocimiento similar al de la empresa maestra. Si las estructuras organizacionales de ambas empresas son muy diferentes, la empresa estudiante tendrá problemas para asimilar el conocimiento de la empresa maestra.

A la dimensión de aprendizaje inter-organizacional se asocia un problema interno: el de aprendizaje intra-organizacional, discutido por otros autores que veremos a continuación.

***Aprendizaje intra-organizacional: formas organizacionales y combinación de capacidades***

Lane y Lubatkin (1998) y Lane, *et al.* (2002) exploran la utilidad de la CA a nivel inter-organizacional, mientras que Van den Bosch, *et al.* (1999) lo exploran a nivel intra-organizacional. El argumento de los últimos autores es que al interior de una empresa, en un departamento de producción, administración, marketing u otro, puede haber conocimiento externo respecto de otro, de forma que cada una de las áreas debe contar con CA para internalizar el ‘conocimiento interno’ generado en cada una de ellas.

Van den Bosch, *et al.* (1999:551) se preguntan cuáles son los determinantes organizacionales de la CA. Explica que la CA de las empresas depende del conocimiento previo acumulado, tal como lo señalan Cohen y Levitnhal (1990), pero también depende de las formas organizacionales y de la combinación de capacidades, como lo señalan Lane y Lubatkin (1998) y Lane, *et al.* (2002). Los mecanismos internos que influyen en la CA de la empresa son el conocimiento relacional, las habilidades básicas y métodos de solución de

problemas, experiencias de aprendizaje a priori y un lenguaje compartido (Van den Bosch, *et al.*, 2002:8).

Según Van den Bosch, *et al.* (1999:554-555), una forma organizacional se puede ver como un tipo de infraestructura que permite la evaluación, asimilación, integración y utilización del conocimiento. Existen tres formas básicas de organización: funcional, divisional y matriz, que son comparadas en términos de la eficiencia, alcance y flexibilidad de la absorción de conocimiento.<sup>2</sup>

Uno de los principales argumentos de este enfoque es que las formas organizacionales son determinantes en el aumento de la CA, ya que la forma de organización de una empresa está relacionada con sus actividades de procesamiento de conocimiento, es decir, con los mecanismos internos que ayudan a aumentar la CA. Según este enfoque, los mecanismos más importantes para aumentar la CA son: i) la transferencia de conocimiento a través y al interior de los departamentos, ii) la estructura de comunicación entre el entorno externo y la empresa, iii) las redes de relaciones internas y externas, y iv) las relaciones entre departamentos funcionales y contactos personales.

El segundo argumento importante se refiere a la combinación de capacidades que, de acuerdo a estos autores, existen tres tipos. Primero, un 'sistema de capacidades' a través del cual la empresa puede integrar conocimiento explícito. Segundo, la 'coordinación de capacidades' que se acumulan en una empresa como resultado de la capacitación, la rotación de empleados entre los departamentos, los enlaces naturales y la participación. Tercero, la 'socialización de capacidades' relacionada a la cultura de la empresa y a una ideología con valores compartidos que permite a los empleados una identidad e interpretación colectiva de la realidad. Estos autores señalan que, de los tres tipos de combinación, la 'coordinación de capacidades' tiene un mayor impacto positivo en la CA, los métodos de coordinación pueden ser explícitamente diseñados aunque también pueden emerger desde un proceso de interacción. También muestran que, en un entorno de

---

<sup>2</sup> Para un detalle de las formas organizacionales y las combinaciones posibles con los términos de eficiencia, alcance y flexibilidad, véase Van den Bosch, *et al.*, 1999 y 2002.

conocimiento turbulento, las empresas son proclives a incrementar su CA a través del desarrollo de formas de organización y combinación de capacidades que conducen a un aumento y flexibilidad de absorción de conocimiento.

Los problemas organizacionales se presentan cuando la empresa crece en términos de sus estructuras organizacionales, ya que se vuelve más complejo tanto compartir el conocimiento como la integración entre los departamentos (Lane, *et al.*, 2002:3), además, el stock de conocimiento es diferente entre cada uno de los departamentos (Lenox y King, 2003:3).

Lenox y King (2003) han explorado si los administradores pueden desarrollar CA a través del suministro de información interna, si los gerentes pueden tomar decisiones de adopción tecnológica mediante la transferencia de datos práctico-específicos desde un almacén de conocimiento (i. e, departamento de I+D) hacia el resto de los departamentos. Estos autores explican que el suministro de información ocurre a través de seminarios internos, demostraciones, boletines promocionales, entre otros, los cuales son mecanismos importantes para el aumento de CA. Esto significa que la empresa debe contar, además, con la habilidad para transferir conocimiento e información entre los diferentes departamentos de la empresa y no sólo la habilidad para absorberlos.

Pero algunos departamentos de la empresa son más importantes en la transferencia y absorción de conocimiento que otros, en particular, en esta literatura el departamento de I+D puede absorber información y conocimiento de nuevas prácticas más eficientes, o evaluar en el mercado (benchmark) las diferentes prácticas y su relación costo-beneficio, en parte porque ellos mantienen vínculos importantes con fuentes externas de información, por ejemplo con universidades u oferentes (Lenox y King, 2003:6). El departamento en donde se genere la mejor información y conocimiento no es necesariamente el mismo donde mejor se utilice, por eso es importante contar con suministros de información y conocimiento para transferirlos al interior de la empresa.

### ***Movilización de recursos organizacionales en pequeñas empresas***

Jones y Crave (2001) analizaron cómo pequeñas empresas absorben información y conocimiento del entorno. Asumen que la literatura sobre CA ha enfatizado en que el gasto que realizan las empresas en I+D representa una aproximación de la adquisición de conocimiento y de CA, pero muchas son pequeñas o medianas empresas sin un área de I+D o departamentos funcionales formalmente, sin embargo, dicen estos autores, son empresas que adquieren CA por otros medios.

Estos investigadores argumentan que la habilidad de un administrador para extender las capacidades existentes depende de la movilización de recursos organizacionales en ese tipo de empresas. Por otro lado, muestran que es esencial identificar aquellas rutinas organizacionales centrales en la traducción de conocimiento tácito a conocimiento codificado para que se logre diseminar a través de la empresa.

Macpherson y Holt (2007) realizaron un análisis a 459 artículos derivados de “investigaciones empíricas sobre conocimiento o aprendizaje y crecimiento en pequeñas empresas”, en el cual encontraron que los sistemas de gestión de conocimiento son necesarios para que la empresa potencie sus habilidades, durante el crecimiento de la empresa esos sistemas son ajustados y las redes de conocimiento llegan a ser más importantes para absorber el conocimiento externo y distribuirlo internamente.

### ***Interfase y CA***

Cohen y Levinthal (1990) y Van den Bosch, *et al.* (1999) muestran que las interfases (*interface cross-function*) afectan la CA y el desempeño innovativo. Cohen y Levinthal, (1990:131-132) argumentan que la CA, además de referirse a la adquisición o asimilación de la información y conocimiento por parte de una empresa, también se refiere a la habilidad para explotarla. Por eso, según estos autores, la CA de la empresa depende de la transferencia de conocimiento a través y dentro de las áreas y no sólo de la interfase entre la empresa y el entorno. De forma que, para entender las fuentes y mecanismos de CA, la literatura se focaliza en las estructuras de comunicación entre las diferentes áreas o departamentos de la empresa (interfases internas, particularmente a partir de las relaciones



entre el departamento de I+D y el departamento de diseño, manufactura o marketing), entre la empresa y el entorno (interfases externas), así como sobre las formas de distribución de las experiencias al interior de la organización.

De acuerdo con Van den Bosch, *et al.* (2002:16), en un entorno de explotación de conocimiento (entorno de conocimiento estable), la función de interfase (la cual consiste en capturar la estructura de comunicación entre el entorno y la empresa y las áreas o departamentos, y además es uno de los medios que anteceden a la CA) tenderá a ser centralizada, lo que permitirá incrementar la eficiencia de la comunicación interna. La razón de la centralización se deriva de una estructura de conocimiento compartido bien desarrollada y un lenguaje interno, creando una CA interna, y donde las empresas tienden a ser más reactivas. Por otro lado, argumentan que en un entorno de exploración de conocimiento (entorno de conocimiento turbulento), la función de interfase será más descentralizada, lo cual se refleja en un aumento de la diversidad o amplitud de las estructuras de conocimiento y una mayor importancia de las interacciones al interior de la empresa.

En la literatura sobre CA se menciona la importancia de la transferencia de conocimiento a través de las áreas o departamentos de la empresa para el aumento de la CA, es decir, ésta depende en mayor medida de las interfases internas y en menor medida de las interfases entre la empresa y el entorno.

En esta tesis se argumenta que el concepto de interfase implica relaciones entre la empresa con el entorno, siendo un mecanismo de aprendizaje que influye en el incremento de la CA, y no sólo relaciones entre departamentos funcionales formalmente, contactos personales o roles de enlace entre cada departamento de la empresa. Esto es relevante dado que, en el segmento de software hecho a la medida, una de las formas más importantes para identificar información y conocimiento útiles para el desarrollo de productos y procesos es a través de la interfase externa.

#### **4.1.2 El problema de la internalización de la información y conocimiento**

La literatura sobre CA ha puesto énfasis en que las empresas deben tener la habilidad para transferir y asimilar la información y conocimiento entre las áreas de la empresa, y entre ésta y el entorno. Cohen y Levintahl (1990:131) mencionan que la habilidad para asimilar la información es una función de la riqueza de la estructura de conocimiento pre-existente, ya que el aprendizaje es acumulativo y su desempeño es alto cuando se relaciona a lo que ya es conocido.

En sentido estricto, la CA de la empresa depende de los individuos, quienes actúan en cualquier tipo de interfase (interna o externa) y, como se ha mencionado en párrafos anteriores, la función de la interfase puede ser difundida entre los individuos, o bien ser centralizada. Cuando la experiencia de muchas personas dentro de la empresa difiere considerablemente de un actor externo (quien puede proveer información útil), algunos miembros del grupo probablemente asumirán roles centralizados. Pero, si la información externa está de alguna manera relacionada a las actividades actuales de la empresa, será inmediatamente asimilada y no requerirá grandes esfuerzos para traducirla. La dificultad emerge bajo condiciones de rápido cambio técnico, aun si la función de interfase está centralizada (Cohen y Levintahl, 1990:132), es decir, en un entorno turbulento se requerirán mayores habilidades para internalizar y asimilar la información y conocimiento del entorno.

Como señalan Cohen y Levintahl (1990:135), la discusión de la literatura sobre CA se ha centrado en los mecanismos internos que influyen en ella, pero hay algunas especificidades a la empresa que no se pueden absorber tan fácilmente o integrar rápidamente, como la información específica asociada a productos y procesos de innovación. Para integrar exitosamente ciertos tipos de conocimiento complejo y sofisticado dentro de las actividades de la empresa, se requiere de un staff de científicos y tecnólogos competentes en sus campos, que estén familiarizados con las necesidades idiosincrásicas de la empresa, procedimientos organizacionales, rutinas, capacidades complementarias e interacciones externas.

#### 4.1.3 Evaluación crítica de la literatura sobre capacidades de absorción

La literatura sobre CA proporciona una estructura analítica para entender cómo las empresas pueden asimilar, transferir (entre las áreas de la empresa) y explotar la información y conocimiento del entorno para aumentar su capacidad de absorción. Las principales contribuciones de esta literatura, y que al mismo tiempo son relevantes para esta tesis, son las siguientes:

- a) Cohen y Levithal (1990), Van den Bosch, *et al.* (1999 y 2002), Lane, *et al.* (2002), Lenox y King (2003), Lund Vinding (2004) y otros autores han hecho énfasis en la importancia de la CA de la empresa para valorar la información y conocimiento del entorno, asimilarlos y explotarlos con fines comerciales. Según estos autores, hay varias fuentes y mecanismos mediante los cuales las empresas pueden lograrlo, tales como el conocimiento previo, la capacitación, actividades de aprendizaje, así como a través de las interfases entre las diferentes áreas de la empresa (interfases internas), e interfases entre la empresa y su entorno [interfases externas]. Aunque estos dos últimos mecanismos han sido menos desarrolladas por dichos autores, las interfases internas son relativamente más importantes en el proceso de asimilación de la información y conocimiento.
- b) Estos investigadores han mostrado que la CA es un concepto multidimensional. Entre las dimensiones más importantes se encuentran: a) las habilidades para reconocer, evaluar y asimilar o adquirir nuevo conocimiento externo, b) las habilidades para transferir la información y conocimiento entre las diferentes áreas de la empresa, y c) las habilidades para explotar y comercializar nuevo conocimiento externo. El esquema de asimilación-transferencia-explotación implica examinar las estructuras de comunicación entre las áreas de la empresa y entre ésta y su entorno.
- c) Un alto nivel de CA influirá positivamente en la formación de nuevas expectativas de la empresa, y el sendero explotación/exploración tenderá a cambiar hacia el de exploración-adaptación (Van den Bosch, *et al.*, 1999:559).

Si bien esta literatura ha contribuido al entendimiento de las diferentes dimensiones y mecanismos por los cuales las empresas pueden incrementar su CA, los estudios se han

centrado particularmente en medianas y grandes empresas que están formalmente estructuradas y cuentan con un área de I+D, la cual les permite identificar e internalizar información y conocimiento del entorno y transferirlos al resto de las áreas. A partir del análisis de los flujos internos, esta literatura ha identificado patrones de comportamiento que se generalizan para este tipo de empresas, pero ha mostrado poco interés en analizar las diferencias entre los comportamientos de micro y pequeñas empresas poco estructuradas y carentes del área de I+D.

En ese sentido, la literatura de CA se ha desarrollado a partir de otras realidades en las que existen medianas y grandes con estructuras organizacionales y una división del trabajo, en particular en las actividades de I+D. La realidad cambia para las micro y pequeñas empresas mexicanas que se ubican en el segmento de software hecho a la medida. Sin embargo, esa literatura aporta los elementos importantes para estudiar este segmento industrial en México.

Por otra parte, aunque las interfases han sido menos analizadas como mecanismo que influye en la CA, se otorga una mayor relevancia a las interfases internas que a las externas. No obstante, como se mostrará en el capítulo 7, la realidad de la industria mexicana de software revela que las interfases externas son un mecanismo crucial de aprendizaje y, por tanto, influyen en el incremento de la CA, pues la estructura organizacional de la micro y pequeña empresa no permite establecer interfases internas (con las características requeridas que señalan estos autores) debido en buena medida, a la ausencia de áreas funcionales formalmente, particularmente con la de I+D.

En este apartado se ha mostrado la importancia y utilidad de las fuentes y mecanismos internos y externos de las empresas para incrementar su CA, y así poder generar nuevo conocimiento expresado en nuevos productos, servicios y procesos. En el siguiente se discuten los conceptos de información y conocimiento, y se mostrará la utilidad de ambos en el proceso de innovación en la industria de software.

## **4.2 INFORMACIÓN Y CONOCIMIENTO COMO INSUMOS CLAVE EN EL PROCESO DE INNOVACIÓN**

En el esquema de PTE, los insumos clave de la revolución tecnológica actual son la información y el conocimiento, por eso, en este apartado veremos las diferencias conceptuales de estos conceptos y su importancia y relación con las interfases y la capacidad de absorción.

### **4.2.1 Información vs conocimiento**

Dos de los primeros autores en tratar el concepto de información y conocimiento en el contexto de empresas que aprenden han sido Nonaka y Takeuchi (1994). Para ellos conocimiento implica creencias y compromisos, a diferencia de la información que es acción, y de la misma manera que la información el conocimiento contiene significados que dependen de contextos específicos.

Según estos autores, la información permite interpretar eventos u objetos desde un punto de vista distinto, es un medio material necesario para extraer y construir conocimiento. Se considera de dos formas:

- a) Sintácticamente (por el volumen que tiene): se refiere al volumen de la información sin cuidar sus significados.
- b) Semánticamente (por el significado que posee): el aspecto semántico es más importante para la creación de conocimiento porque se centra en el significado expresado.

El concepto de información antecede al de conocimiento. Siguiendo a Nonaka y Takeuchi (1994:64), la *información* es un flujo de mensajes compuesto de datos manipulados para la toma de decisiones, mientras que el conocimiento está relacionado con la acción humana y es creado a partir de ese flujo de información, anclado en las creencias y el compromiso de su poseedor.

El conocimiento como la información son elementos de contexto específico y son relacionales, ya que dependen de la situación y se crean dinámicamente durante la

interacción social. Berger y Luckmann (1996<sup>3</sup>) señalan que al interactuar en cierto contexto histórico y social, las personas comparten información con la cual construyen un conocimiento social que conforma una realidad influyente en sus juicios, su comportamiento y su actitud. Si la gente no tiene un contexto para la información o para entender cómo usarla, no tiene valor y, por tanto, no puede ser transformada en conocimiento (O'Dell, *et al*, 2000:1).

### ***Datos-información-conocimiento***

Con base en Davenport y Prusak (1998:4), el conocimiento no es ni datos ni información. Datos es una serie de eventos discretos y de hechos objetivos registrados ordenadamente. En un contexto organizacional, los datos son utilizados para describir un evento estructurado de transacciones. Los datos no tienen significado inherente, describen solamente una parte del suceso, pero son importantes para la organización ya que constituyen la materia prima de la información.

Siguiendo a este autor, de igual forma que en la transmisión de un mensaje, la información tiene un hablante y un receptor. La información está tratando de cambiar el sentido que el receptor tiene o percibe de algo, impactando sobre su juicio y comportamiento. La información se mueve alrededor de los datos a través de redes duras y blandas. Una red dura tiene una estructura visible y definida tales como una instalación eléctrica, equipo de entrega, antenas de satélite, oficinas, correos. Una red blanda es menos formal y visible.

A diferencia de los datos, la información tiene significado y contiene juicios, es decir, los datos se convierten en información cuando su creador agrega significado y valor: contextualizando, categorizando, calculando, corrigiendo y condensando (Davenport y Prusak, 1998:4).

La distinción entre conocimiento e información consiste en que ésta ha sido definida como datos en proceso de formación, esto es, datos que han sido clasificados, analizados, demostrados y comunicados a través del lenguaje hablado, gráficamente o en tablas

---

<sup>3</sup> Citado por Nonaja y Takeuchi, 1994:64.

numéricas. Por su parte, el conocimiento es definido como el vínculo de significados que la gente hace en sus mentes entre la información y su aplicación en acción en un escenario específico. Vincular el conocimiento a la acción es útil para diferenciarlo de la información, pero la diferencia es clara cuando se aplica a situaciones organizacionales. Para Dixon (2000:13) el conocimiento común está siempre vinculado a la acción.

Entonces, ‘el conocimiento es una combinación fluida de experiencias, valores, información contextual e ideas expertas que proveen una estructura para evaluar e incorporar nuevas experiencias e información, por ello se origina y es aplicado en la mente de los conocedores. En las organizaciones a menudo se incorpora en rutinas organizacionales, procesos, prácticas y normas, y no sólo en documentos’ (Davenport y Prusak, 1998:5).

Además, conocimiento implica entendimiento, pues su adquisición es un proceso cognitivo, puramente perceptible. En principio, todo conocimiento puede ser transferido por la interacción personal, único canal por el cual el conocimiento tácito puede ser transferido, en cambio, el conocimiento codificado puede ser transferido a través de la escritura (Senker y Faulkner, 1996), e imágenes, sonido y planos (Grimaldi y Torrisi, 2001).

En este sentido, el conocimiento proviene de la información como ésta de los datos. El conocimiento también se puede mover hacia abajo de la cadena de valor, retornando a información y datos, sobretodo cuando el conocimiento ha sido codificado.

A diferencia de la información, el conocimiento tácito no puede ser transmitido o imitado, por tal razón, las empresas seguidoras o los países en desarrollo tienen que acumular capacidades de absorción con el propósito de aprovechar las externalidades producidas por los innovadores (Cohen y Levinthal, 1990).

#### **4.2.2 Conocimiento tácito y conocimiento codificado**

Rosenberg (1982) describió al conocimiento tácito desde una perspectiva científica y tecnológica, asume que este conocimiento es heurístico, subjetivo e interno, no es fácil de

comunicar y es aprendido a través de ejemplos prácticos, de la experiencia y de la práctica misma.

Desde una perspectiva organizacional, Nonaka y Takeuchi (1996:65) señalan que el conocimiento de una empresa incluye elementos tácitos y codificados que se encuentran distribuidos en diversas áreas. El conocimiento tácito se sitúa en el plano de la ontología, es decir, sólo es creado por los individuos, de forma que es personal y de contexto específico, difícil de formalizar y comunicar. Este conocimiento contiene elementos cognoscitivos y técnicos. Los elementos cognoscitivos se centran en los modelos mentales: los humanos crean modelos activos del mundo haciendo y manipulando analogías en su mente. Los modelos mentales, como esquemas, paradigmas, perspectivas, creencias y puntos de vista ayudan a los individuos a percibir y a definir su mundo. El elemento técnico del conocimiento tácito contiene *know how*, oficios y habilidades concretos que se aplican a contextos específicos.

En tanto el conocimiento explícito o ‘codificado’ se puede transmitir utilizando el lenguaje formal y sistemático, como argumentan Senker y Faulkner (1996), por ejemplo, a través de principios generales y leyes reconocidas por la comunidad de científicos e ingenieros que están escritos en manuales y libros de texto, trabajos técnicos y científicos, en especificaciones técnicas de materiales o componentes, y en manuales de operación para procesos industriales y de comercialización. El conocimiento tácito está incorporado en las habilidades, mismas que pueden ser copiadas. Pero también hay conocimiento tácito que no puede ser demostrado y por tanto es difícil de transferir. Cohendet y Steinmueller (2000) argumentan que codificar el conocimiento tácito puede ser un proceso costoso, sobretodo cuando el conocimiento es profundamente contextual y está incorporado en la experiencia de los individuos.

Estudios de innovación acerca de la transferencia y difusión tecnológica han identificado al conocimiento tácito como un elemento importante en el proceso de innovación. Si bien no se ha dado una definición a detalle de él y su rol en la innovación, tampoco en cómo puede diferir de acuerdo al sector industrial o tecnológico. Los estudios se han centrado más en



cómo las empresas pueden adquirirlo (Senker Faulkner, 1996:76). En este sentido, los insumos científicos y tecnológicos para la innovación incorporan un componente considerablemente tácito que solamente puede ser adquirido por experiencia y práctica, es decir, a través del aprendizaje.

La diferencia fundamental entre el conocimiento tácito y codificado se da cuando el primero permite a la empresa solucionar problemas, concretamente cuando no hay un conocimiento previo detrás de esos problemas o no hay métodos óptimos de solución (Grimaldi y Torrisi, 2001:1427).

### ***El conocimiento como determinante de una trayectoria tecnológica***

La base de conocimiento utilizada en las actividades de una empresa puede ser idiosincrásica a nivel de la empresa, y por tanto ser distinta entre una y otra. Siendo así, el conocimiento no se difunde automática ni fortuitamente entre las empresas, pero puede ser adquirido por éstas través de sus diferentes habilidades desarrolladas y acumuladas en el tiempo.

Con base en Malerba y Orsenigo (1999), el conocimiento tiene diferentes dimensiones en las actividades innovativas de un sector. La primera dimensión hace referencia a los diferentes grados de *accesibilidad*, i.e., la oportunidad de acceder al conocimiento que es externo al sector. Si el conocimiento es interno al sector existen mayores posibilidades de favorecer la imitación, por el contrario, si es externo al mismo puede afectar el acceso a las oportunidades tecnológicas de las nuevas empresas. La accesibilidad del conocimiento decrece en la medida en la que existe una mayor concentración industrial.

Cuando los competidores cuentan con una accesibilidad interna, pueden obtener conocimiento de nuevos procesos y productos y tener la posibilidad de imitarlos, lo que significa menor apropiabilidad. La accesibilidad del conocimiento también puede ser *externa* al sector y estar relacionada con las oportunidades tecnológicas y científicas, es decir, el entorno puede afectar a las empresas a través del conocimiento científico y tecnológico incorporado en el capital humano que se desarrolla en otros sectores y

organizaciones, tales como universidades o centros de I+D. En este sentido, las fuentes de oportunidades tecnológicas difieren entre sectores, y algunas empresas las pueden encontrar en las universidades, en los departamentos de I+D, y en la interacción entre oferentes y usuarios.

La segunda dimensión hace referencia al *grado de acumulación* del conocimiento. Hay tres fuentes de acumulación: la primera es el proceso de aprendizaje y el incremento dinámico a nivel tecnológico; la segunda está relacionada a las capacidades organizacionales que son específicas a la empresa y son mejoradas y renovadas a través del tiempo; la tercera es la retroalimentación que se establece con el mercado.

La accesibilidad y la acumulatividad son dos dimensiones de conocimiento que se asocian con la noción de régimen tecnológico, el cual difiere entre sectores. Según Malerba y Orsenigo (1996), ese régimen se compone de condiciones de oportunidad y apropiabilidad, grados de acumulatividad del conocimiento tecnológico y características relevantes de la base de conocimiento.<sup>4</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 importantes los medios formales de transmisión del conocimiento tales como las publicaciones internas para difundir los avances de la empresa, manuales de proceso y producto, patentes, etc. Con ello, la proximidad geográfica no es fundamental para facilitar la transmisión del conocimiento (Malerba y Orsenigo, 1990, 1995 y 1996), y el conocimiento puede ser

---

<sup>4</sup> A manera de paréntesis, las industrias de microelectrónica moderna, de software y microcomputadoras -en las cuales hay una cantidad importante de innovadores geográficamente concentrados con límites de conocimiento local y global- están asociadas a altas condiciones de oportunidad y a una amplia variedad de potencialidades tecnológicas.

codificado en un contexto, pero tácito en otro (Senker y Faulkner, 1996), o bien complementarios (Ancori, *et al.*, 2000). Aunque los diferentes tipos de conocimiento coexisten en la organización, la relación complementaria entre ellos no ocurre de manera automática ya que está ligada a la existencia de reglas y modos de regulación que la favorecen o la obstaculizan (Villavicencio y Salinas, 2002).

#### **4.2.3 El concepto de conocimiento en operación**

Cusumano (1988), Hobday y Brady (1996 y 1997), Grimaldi y Torrissi (2001) y otros autores, han explicado que la industria de software está caracterizada por realizar actividades ‘artesanales’ o ‘creativas’, lo cual implica actividades intensivas en conocimiento tácito más que explícito, pero hay elementos externos a la empresa que estimulan la codificación del conocimiento tácito. Por su parte, Grimaldi y Torrissi (2001) mencionan que en la industria del software la codificación está influenciada por los altos costos del trabajo debido a la competencia entre las empresas por acceder a recursos humanos especializados, y por la internalización del mercado de software, lo cual incide en adoptar técnicas de producción eficiente y a reestructurar las empresas.

La experiencia adquirida a través de arreglos de colaboración entre las áreas de la empresa induce a un alto nivel de codificación del conocimiento, o al menos estimula el desarrollo de interfases organizacionales, lo cual hace transparente algunas partes del conocimiento y, por lo tanto lo hace transferible entre las empresas. Esta noción de hacer codificable el conocimiento tácito permite una mayor integración entre las áreas de la empresa ya que permite construir un lenguaje común y rutinas organizacionales que incrementan la productividad de la empresa y la calidad de los productos (Grimaldi y Torrissi, 2001:1438).

Estos autores encontraron que algunas de las actividades que pueden ser codificables son las de comercialización, de administración de proyectos y consultoría, lo cual permite una mejor circulación del conocimiento y la codificación de las mejores prácticas entre las diferentes áreas de la empresa. Mientras que es tácito y difícil de codificar el conocimiento generado en las actividades de análisis de las necesidades y requerimientos de los usuarios y del diseño del sistema derivado de los requerimientos. A esto se asocian los problemas de

transmisión del conocimiento tácito entre los departamentos de la empresa y, desde el enfoque de CA, de asimilación de nuevo conocimiento generado al interior de la empresa y existente en el entorno.

La evidencia encontrada por Grimaldi y Torrisi (2001) indica que a través de los vínculos externos, las empresas de software acceden a conocimiento general científico o tecnológico (relacionado a nuevas tecnologías, plataformas y estándares de calidad), así como a conocimiento de contexto-específico (relacionado a mercados particulares, usuarios y aplicaciones).

Otros autores como Assimakopoulos y Yan (2006) encontraron que las fuentes principales de conocimiento para los ingenieros de software son los libros y artículos por Internet, que proveen conocimiento acerca de técnicas específicas en un sentido abstracto. La abstracción de las técnicas requiere de la colaboración de colegas internos o externos sin formar una red formal, o bien del Internet pues se puede acceder a conocimiento mediante los foros y comunidades de desarrolladores.

Como se señala en la literatura sobre CA, el conocimiento crítico no sólo incluye conocimiento técnico sino también la conciencia de en dónde reside la experiencia complementaria útil dentro y fuera de la organización. Esto permite saber quién conoce qué, quién puede ayudar con qué problema, o quién puede explotar nueva información (Cohen y Levinthal, 1990:133).

#### **4.2.4 Evaluación crítica del concepto de información y conocimiento**

La literatura sobre conocimiento desde una perspectiva organizacional ofrece un esquema para entender las diferencias entre información y conocimiento. Entre sus principales argumentos se encuentran:

- a) La información es un flujo de mensajes compuesto de datos manipulados para la toma de decisiones (Nonaka y Takeuchi, 1994:64).
- b) El conocimiento es creado por ese flujo de información, anclado en las creencias y el compromiso de su poseedor (Nonaka y Takeuchi (1994:64), ‘es una combinación

fluida de experiencias, valores, información contextual e ideas expertas que proveen una estructura para evaluar e incorporar nuevas experiencias e información...En las organizaciones se incorpora en rutinas organizacionales, procesos, prácticas y normas, y no sólo en documentos' (Davenport y Prusak, 1998:5).

- c) A partir de esto, se puede diferenciar entre conocimiento general y conocimiento específico. El primero se obtiene a través de la educación formal, y el segundo por experiencia en las actividades productivas, por acceso a información técnica de nuevos productos o procesos, entre otros (Dahl y Pederson, 2004:1680).

La importancia de hacer la diferenciación entre esos conceptos se debe a la 'delgada línea' que existe entre conocimiento tácito y codificado en la industria de software. Nuevas líneas de código (en el campo de la programación de software) no son sinónimo de conocimiento codificado, detrás de ese nuevo código hay conocimiento tácito incorporado en el desarrollador. Es decir, no son un 'flujo de mensajes compuesto de datos manipulados', tampoco es información, sino conocimiento (tácito) que se crea a partir de "una combinación fluida de experiencias, valores, información contextual e ideas expertas que proveen una estructura para evaluar e incorporar nuevas experiencias e información" (Davenport y Prusak, 1998:5). Aunque es posible trasladar ese conocimiento tácito a nuevas líneas de código, esto no significa 'codificarlo', sólo se hace cuando alguna parte del programa (un módulo o proceso, por ejemplo) tiende a la estandarización.

Ambos conceptos son contextuales pero el conocimiento tácito sólo se adquiere a través de la práctica, del contacto personal, por observación e imitación. Las empresas pueden generar y absorber conocimiento tácito y codificado, general y específico, y deberán identificar e internalizarlos de acuerdo a sus necesidades tecnológicas. En este sentido, uno de los elementos que interesan en esta tesis es cómo las empresas internalizan la información y conocimiento existentes en el entorno y creados a partir de las interfases entre los agentes, es decir, mediante qué mecanismos pueden incrementar su CA y generar nuevo conocimiento expresado en nuevos productos, procesos y servicios.

### 4.3 ESTRUCTURA CONCEPTUAL Y CATEGORÍAS ANALÍTICAS

A partir de la anterior revisión de literatura y las preguntas de investigación es posible estructurar los conceptos guía que sustentarán teóricamente el presente trabajo. La estructura conceptual se ilustra en la figura 4.1, la cual incluye los tres conceptos guía y las relaciones entre ellos.

**Interfase.** Se define como una interacción entre proveedores y usuarios (y otros agentes del entorno) de artefactos tecnológicos que, a través de mecanismos socio-técnicos, transforman flujos de información y conocimiento existente en el entorno en nuevos productos, procesos y servicios.<sup>5</sup> Se distingue interfase interna de la interfase externa. La primera se da entre las áreas de la empresa, la segunda entre la empresa y los diversos agentes del entorno. La interfase externa se considera de mayor importancia en esta tesis.

**Capacidad de absorción (CA).** Con base en Cohen y Levithal (1990), Van den Bosch, *et al.* (1999 y 2002), Lane y Lubatkin (1998 y 2002), Lenox y King (2003), Lund Vinding (2004), CA se define como “la habilidad de la empresa para identificar la información y conocimiento existentes en el entorno [y que son generados a través de las interfases], asimilarlos y explotarlos adecuadamente con fines comerciales [en el desarrollo de nuevos productos, procesos y servicios].”

**Conocimiento.** Se utiliza la definición de Davenport y Prusak (1998): es “una combinación fluida de experiencias, valores, información contextual e ideas expertas que proveen una estructura para evaluar e incorporar nuevas experiencias e información...En las empresas usualmente se incorpora en rutinas organizacionales, procesos, prácticas y normas, y no sólo en documentos.”

La relación entre los tres conceptos puede mostrar un proceso de circularidad, pero el proceso que se trata de expresar es el de causalidad y retroalimentación. Las empresas requieren identificar información y conocimiento útiles para el desarrollo de productos y, dado que el producto es hecho a la medida con especificidades únicas, gran parte de éstos

---

<sup>5</sup> El conocimiento generado puede ser *nuevo* para la empresa o la industria nacional.

se identifica sólo a través de las interfases creadas con los usuarios. Las empresas deben valorar, asimilar e internalizar organizacionalmente la información y conocimiento que ha sido capturada previa y adecuadamente. Si la empresa logra eso, en consecuencia, podrá incrementar su CA y hacer mejoras incrementales o desarrollar nuevos productos, procesos y servicios.

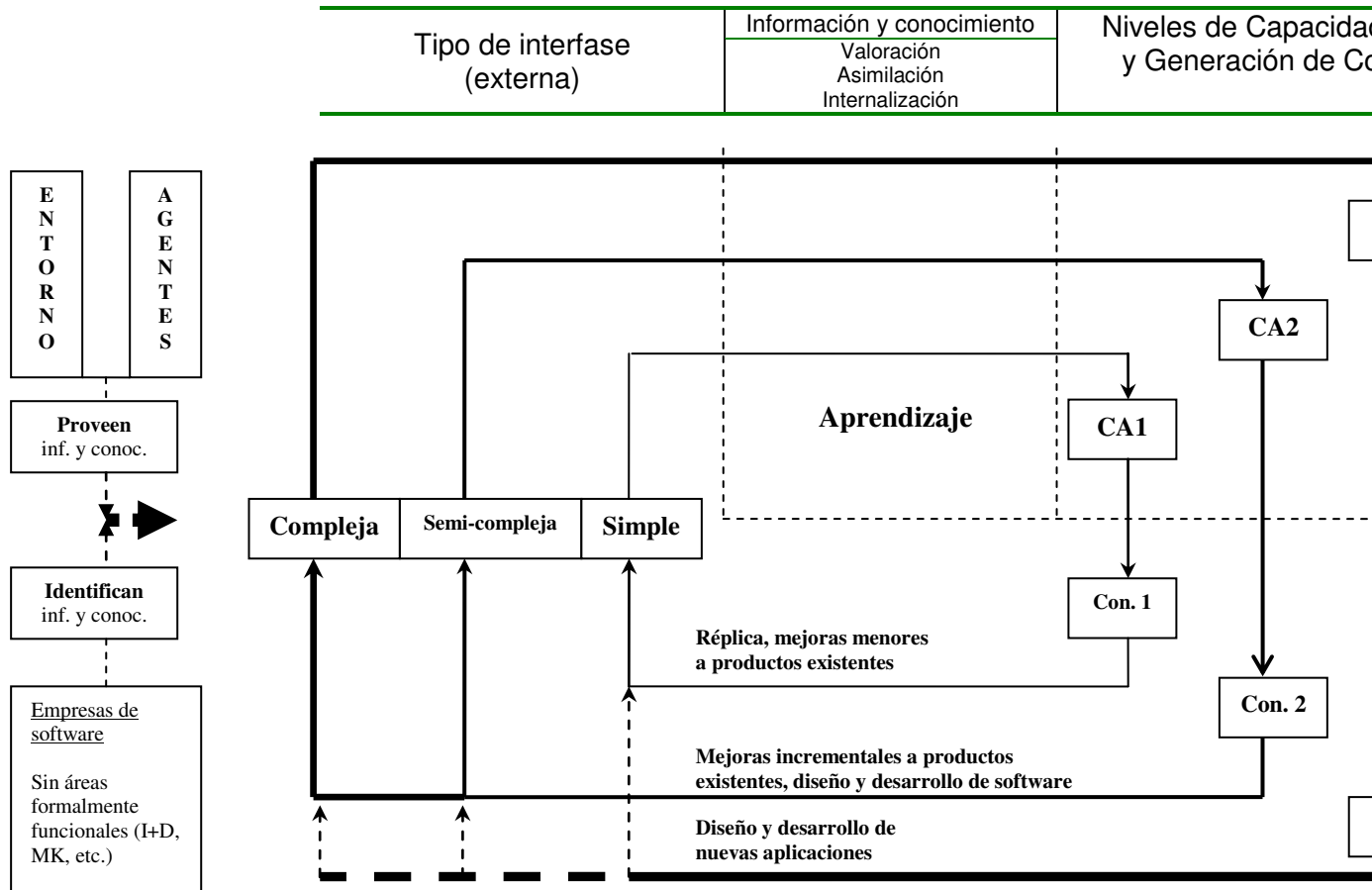
En la medida en que las empresas de software desarrollen su CA y se esfuercen por desarrollar proyectos de mayor valor, podrán pasar de una interfase simple a crear interfases más complejas. Como se muestra en la figura 4.1, una interfase simple puede crear un sendero poco permisible para explotar nuevas oportunidades tecnológicas. Una de las causas se da cuando la empresa desarrolladora de proyectos/productos estandarizados, que son al mismo tiempo rentables económicamente, y éstos otorgan a la empresa un nicho de mercado estable que no tendrá un incentivo para aprender nuevas tecnologías de software y desarrollar proyectos/productos de mayor valor. Pero el entorno es dinámico, las necesidades de los usuarios evolucionan y algunas empresas se adaptan a éstas, se esfuerzan por desarrollar proyectos de mayor valor y por aprender del usuario, pierden aversión al riesgo y, por tanto, rompen con el esquema vicioso que se puede crear en una interfase simple. Bajo este nuevo esquema, los usuarios son más sofisticados, lo cual induce al diseño de interfases más complejas, la empresa puede incrementar su CA, dependiendo del aprendizaje logrado a lo largo del proyecto y la interfase, y en consecuencia, se harán mejoras incrementales o nuevos productos de software.

El argumento central de esta tesis es que las interfases entre las empresas de software y los diferentes agentes del entorno (interfases externas) son un mecanismo crucial de aprendizaje y por tanto influye en la CA. En un ambiente de rápida evolución tecnológica de software, las empresas deben actualizarse constantemente en términos tecnológicos para mantenerse en el mercado; una forma de hacerlo es identificando y absorbiendo rápida y adecuadamente la información y conocimiento específicos de los usuarios. Las empresas deben ser capaces de valorar la información y el conocimiento que se genera constantemente en el entorno, asimilar aquellos que son útiles a sus necesidades y explotarlos apropiadamente para generar nuevos productos, procesos y servicios.

Los tres conceptos guía tienen diferentes formas de aproximación o variables, las cuales se tratan en los siguientes puntos. La forma de operacionalizar estos conceptos se explica en el último apartado del siguiente, correspondiente a la metodología de investigación, y dentro del apartado de 'enfoque y método' de los capítulos 6 y 7, se detallan las formas de identificación -indicadores-.



**Figura 4.1**  
**Estructura conceptual**



Fuente: Elaboración propia. Línea delgada implica menor interacción entre los agentes que la línea gruesa.

### 4.3.1 Interfase

La literatura revisada en el capítulo 3 muestra una serie de elementos importantes para esta investigación, pero también limitaciones en la conceptualización de la interfase. Por un lado, en el apartado 4.2 se argumentó que el conocimiento es generado a partir de procesos cognitivos y se transfiere a través de la interacción social entre individuos en determinados contextos. En esta tesis se plantea que la interfase es un articulador del conocimiento particular a cada agente. En un entorno rico en información, las empresas pueden identificar la información y conocimiento útil del entorno y adaptarlo a sus necesidades a través de la interfase. Entonces, la interfase se convierte en un mecanismo crucial en el proceso de creación de nuevo conocimiento.

Por otro lado, a partir del trabajo seminal de Andersen (1991 y 1996), algunos autores como Cohen y Levithal (1990), Van den Bosch, *et al.* (1999 y 2002), Geyer y Davies (2000), Padilla y Martín (2003), Taminiau (2006) y OECD (2006) han señalado indistintamente una interfase como una relación, interacción o vinculación. Bajo el esquema de PTE, Andersen define el concepto de interfase como una interacción entre dos agentes a través de la cual fluyen diferentes tipos de información. Como se argumentó en el capítulo 3, este concepto es importante para esta tesis ya que destaca los flujos de información que se generan entre proveedores y usuarios con el propósito de desarrollar nuevos productos, pero ha puesto poca atención a la forma en cómo las empresas internalizan la información y conocimiento para lograr ese propósito. Por otra parte, esta literatura no considera: i) la interacción entre múltiples agentes, es decir, la interfase como un concepto envolvente de otras interacciones como la de proveedor-usuario o la de vinculación universidad-empresa, ii) la interfase como un mecanismo que permite la combinación de diferentes tipos de información y conocimiento -fragmentarios- que posee cada agente, iii) la interfase como un mecanismo que permite identificar información y conocimiento útiles a las necesidades de las empresas, y iv) la interfase externa como mecanismo crucial de aprendizaje que influye en la CA.

Con base en lo anterior, en esta tesis se redefine el concepto de interfase propuesto por Andersen (1991) como *una interacción entre proveedores y usuarios (y otros agentes del*

*entorno) de artefactos tecnológicos que, a través de mecanismos socio-técnicos, transforman flujos de información y conocimiento existente en el entorno en nuevos productos, procesos y servicios.*<sup>6</sup> Los mecanismos sociales más importantes de la interfase que son utilizados en esta tesis son la interacción proveedor-usuario, la vinculación universidad-empresa y las comunidades de desarrolladores de software de fuente abierta. Los mecanismos 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 desplegados en un monitor, etcétera). Los contenidos principales y habituales en cada uno de los mecanismos son la información y conocimiento. Estos mecanismos técnicos son un habilitador en la creación de nuevo conocimiento, pero en esta tesis son más relevantes los mecanismos sociales.

El concepto de interfase contiene elementos de contexto y específicos a la empresa. Los elementos de contexto son, primero, el gran dinamismo de la industria de software que implica cambios constantes en la creación y uso de nuevas aplicaciones y herramientas (básicamente) para diferentes sectores industriales, en estos términos induce un cambio de paradigma tecno-económico a nivel micro. Segundo, la segmentación de nichos de mercado local e internacional.

Los elementos específicos a la empresa son los que permiten aproximarse al concepto de interfase. Considerando la literatura y la definición de interfase que se usa en esta tesis, es posible identificar una interfase por medio de su simplicidad o complejidad en términos de la interacción de una empresa con diferentes agentes del entorno. Más específicamente, la interfase se identifica a través de:

- a) Los principales agentes con los que la empresa de software establece algún tipo de interfase: (i) usuarios, (ii) universidades y centros de capacitación, (iii) instituciones de gobierno, y iv) comunidades de desarrolladores de software de fuente abierta.

---

<sup>6</sup> El conocimiento generado (expresado a través de nuevos productos, procesos o servicios) puede ser *nuevo* para la empresa o la industria.

- b) La complejidad de la interacción por tipo de proyecto: relación compra-venta, adaptabilidad o escalamiento de software ya instalado, diseño y desarrollo de nuevo software, implantación, mantenimiento, integración de sistemas, formación de recursos humanos especializados y capacitación.
- c) La estabilidad de la tecnología: grado de estandarización de las herramientas de software utilizadas en el desarrollo de un programa o aplicación; la cual es valorada cualitativamente como alta, media, baja.
- d) La intensidad de la interacción:
  - Días, semanas o meses.
  - Cantidad de información intercambiada (alta, media, baja)
- e) La formalidad de la interacción implica un proceso de sistematización del intercambio de la información y conocimiento, por ejemplo, la documentación y sistematización de las reuniones de trabajo relacionadas con el análisis de requerimientos y diseño de un proyecto de software. La informalidad implica la ausencia de esto. Esta categoría analítica no es planteada por los cuerpos de literatura revisados en esta tesis, pero es importante ya que es un componente central en los diferentes proyectos realizados por las empresas de software; se deduce a partir del trabajo empírico llevado a cabo en esta investigación.

Las empresas de software han creado diferentes interfases con los diferentes agentes. Éstas son: i) simples, ii) semi-complejas, y iii) complejas.

#### **4.3.2 Capacidad de absorción**

La creación de nuevo conocimiento requiere de diversos mecanismos tales como el aprendizaje pero, en entornos tecnológicos como el de la industria de software, las empresas necesitan mecanismos para identificar, evaluar, asimilar y utilizar adecuadamente la información y el conocimiento que constantemente se crea en ese entorno. Cohen y Levithal (1990), Van den Bosch, *et al.* (1999, 2002), Lane, *et al.* (2002), Lenox y King (2003), Lund Vinding (2004), entre otros, hicieron énfasis en la importancia de la CA de la empresa para “valorar la información y conocimiento del entorno, asimilarlos y explotarlos con fines comerciales”. Según estos autores hay varias fuentes y mecanismos mediante los

cuales las empresas pueden lograrlo, tales como el conocimiento previo asociado a sus actividades particulares, la capacitación, las actividades de aprendizaje, así como a través de las interfases entre las diferentes áreas de la misma, denominadas interfases internas, e interfases entre la empresa y su entorno, denominadas interfases externas en esta tesis.

Como se mencionó en el punto 4.1, la literatura que aborda conceptual y empíricamente la CA señala la importancia organizacional y su estructura formalmente funcional. Una gran cantidad de estudios sobre CA se han focalizado en empresas medianas y grandes que cuentan con áreas funcionales formalmente, en particular con la de I+D, a partir de la cual la empresa puede identificar e internalizar información y conocimiento del entorno y transferirlos al resto de las áreas. Pero han profundizado poco en el estudio de los flujos externos en micro y pequeñas empresas que, por su estructura y naturaleza, no cuentan con esas áreas estructuradas formalmente, es decir, las actividades de administración, contabilidad o manufactura se centralizan en un departamento, en el peor de los escenarios en una sola persona. Esta atomicidad es característica de las empresas mexicanas que desarrollan software hecho a la medida.<sup>7</sup> No obstante, esta literatura aporta elementos importantes para estudiar una parte de la realidad de la industria de software en México: el segmento de software hecho a la medida.

Teóricamente esto podría presentar un problema porque los casos en estudio de esta tesis son micro y pequeña empresa que no llegan a contar con departamentos formalmente funcionales, y difícilmente cuentan con un departamento de I+D, el cual ha sido en varios estudios el pilar para entender el proceso de CA de las empresas. Sin embargo, esa limitación presenta una oportunidad de estudio ya que, como menciona Jones y Craven

---

<sup>7</sup> Segelod y Jordan (2004), en un estudio realizado en 19 países (15 países europeos, Estados Unidos, Canadá, Perú y Australia) a pequeñas y medianas empresas de software, encontraron que una característica de la pequeña y mediana empresa de software es que no hay una división del trabajo. Hualde y Gomis (2007:206) hallaron que las empresas de software en Baja California tienen una estructura interna flexible con una diferenciación poco clara del trabajo, y que esto se debe principalmente a la dificultad de los empresarios para responder de forma organizada a la actividad productiva, en especial cuando se acumula el trabajo. En esta tesis se muestra (en el capítulo 8) que, para cada una de las etapas de desarrollo de software, se requiere de personal especializado, sin embargo, unas cuantas personas (en ocasiones una) realizan todas las actividades requeridas en cada una de las etapas, de forma que es difícil registrar cada actividad (por ejemplo llenar bitácoras), en este sentido la organización del trabajo tiende a ser aún 'artesanal'.

(2001:19), las habilidades de los empleados para extender las capacidades existentes dependen de la movilización de los recursos con que cuenta la pequeña o mediana empresa.

La forma de aproximación al concepto de CA en esta tesis es a través de:

a) El conocimiento previo

- Habilidades básicas con que cuentan los empleados de las empresas de software, y
- Contratación de nuevo personal con conocimiento general (educación formal).

b) Flujos de información y conocimiento al interior de la empresa

- Sistemas de procesamiento de conocimiento observable (sistemas de procesamiento, bitácoras)
- Transferencia de conocimiento entre las áreas (foros de discusión, grupos de trabajo)

c) Las actividades de aprendizaje tecnológico

- Aprendizaje por reutilización
- Aprendizaje por capacitación
- Aprendizaje por actividades multitareas al interior de la empresa
- Aprendizaje por I+D

En el contexto de la industria de TIC, el paradigma tecno-económico está cambiando constantemente. Por un lado, refleja el grado y formas de interacción entre empresas de software y agentes del entorno. Las formas de interacción han sido estudiadas a profundidad en sectores industriales donde el producto y los componentes son físicos, pero en menor grado donde éstos son intangibles, como el caso del software.

Por otro lado, hay un nuevo paradigma tecno-económico que surge de la transformación interna de la industria de software, como se mostró en el capítulo anterior. Por eso es importante mencionar los dos grandes contextos bajo los cuales se desarrolla esta industria: i) un contexto de software de fuente abierta, y ii) un contexto de software propietario. En el primero, la empresa genera códigos y aplicaciones que son liberados, no se pagan licencias. El segundo contexto es radicalmente diferente, todo el conocimiento que se genera a través de nuevas aplicaciones o códigos, es secreto y su difusión tiene un costo.

### **4.3.3 Conocimiento y flujos de información**

El conocimiento es el tercer concepto guía de esta investigación, se define como “una combinación fluida de experiencias, valores, información contextual e ideas expertas que proveen una estructura para evaluar e incorporar nuevas experiencias e información...En las empresas usualmente se incorpora en rutinas organizacionales, procesos, prácticas y normas, y no sólo en documentos” (Davenport y Prusak, 1998:5).

A pesar del entorno rico en información, pero las empresas deben tener la habilidad para distinguir aquella de valor para sus procesos tecnológicos, productivos y organizacionales. Un mismo tipo de información no tiene la misma utilidad para dos empresas diferentes, aunque éstas realicen el mismo producto o servicio, ya que sus necesidades y la de sus clientes o usuarios serán diferentes, en otras palabras, el valor de la información y conocimiento depende de los usos y aplicaciones.

Entonces las empresas deben identificar el valor de la información y conocimiento existentes en el entorno, lo cual no es fácil cuando éste es de rápido cambio tecnológico. En esta tesis se argumenta que un mecanismo importante para lograrlo es a través de las interfases externas.

La empresa debe ser capaz de identificar información y conocimiento general y específico, ubicar sus fuentes principales y comprender los medios por los cuales fluye. La forma de identificar los diferentes tipos de información y conocimiento es a través de:

- a) Información general: identificada por medio de revistas especializadas, Internet, seminarios, consultoría.
- b) Información específica: identificada con ayuda de la capacitación interna y externa, y usuarios.
- c) Conocimiento general: identificado mediante los empleados nuevos egresados de universidades o centros técnicos de capacitación.
- d) Conocimiento específico: identificado por medio de los años de experiencia de los empleados, capacitación interna y externa, y usuarios.

El tipo de proyecto es determinante en la generación de conocimiento. En el cuadro 4.1 se muestra la causalidad de efectos a partir de las diferentes interfases, los cuales se validarán en esta tesis en los capítulos 6, 7 y 8. A partir de una interfase simple, en la que predominan proyectos de software que implican mantenimiento, adaptación menor y compra-venta, las empresas identifican e internalizan información y conocimiento de propósito general existentes en el entorno, lo que incide en un bajo incremento de su capacidad de absorción. En este proceso, las empresas sólo son capaces de crear más información de propósito general y replicar o hacer adaptaciones menores a productos ya existentes.

A partir de una interfase semi-compleja, en la que predominan proyectos de escalamiento, prueba y en algunas ocasiones de diseño y desarrollo, las empresas identifican e internalizan información y conocimiento de propósito general existentes en el entorno, pero también de tipo específico identificados a través de los usuarios, lo que incide en un incremento moderado de su capacidad de absorción. En este proceso, las empresas pueden hacer mejoras incrementales a los productos de software ya creados en el mercado, pero también desarrollar nuevos productos.

**Cuadro 4.1**  
**Tipo de interfase y efectos esperados en la capacidad de absorción y generación de nuevo conocimiento**

Tipo de interfase	Tipo de proyecto	Categoría de información o conocimiento	Efecto en la CA	Efecto en la generación de nuevo conocimiento
Simple	Mantenimiento, adaptación menor y compra-venta	Las empresas identifican e internalizan información y conocimiento de propósito general existentes en el entorno	Bajo incremento de su capacidad de absorción	Se crea más información de propósito general y se replican o hacen adaptaciones menores a productos ya existentes
Semi-compleja	Escalamiento, diseño y desarrollo, y pruebas de software	Las empresas identifican e internalizan información y conocimiento de propósito general existentes en el entorno, así como información y conocimiento específico identificados a través de los usuarios	Moderado incremento de su capacidad de absorción	Mejoras incrementales a productos de software ya existentes en el mercado, también diseño y desarrollo de nuevas aplicaciones
Compleja	Diseño y desarrollo de nuevas aplicaciones, y formación de recursos humanos especializados	Las empresas identifican e internalizan información y conocimiento específico	Significativo incremento de su capacidad de absorción	Formación de recursos humanos con habilidades específicas de acuerdo a los requerimientos de un segmento de la industria de software; diseño y desarrollo de nuevas aplicaciones

Fuente: Elaboración propia.

A partir de una interfase compleja, en la que predominan la vinculación universidad-empresa y la interacción entre empresas de software y comunidades de desarrolladores de



software de fuente abierta, son importantes los proyectos de diseño y desarrollo, así como la formación de recursos humanos especializados. Las empresas identifican e internalizan información y conocimiento específico, lo que recae en un alto incremento de su capacidad de absorción. Las empresas pueden desarrollar nuevo conocimiento mediante la formación de recursos humanos con habilidades específicas de acuerdo a los requerimientos de un segmento de la industria de software.

#### **4.4 CONCLUSIONES**

En los capítulos 3 y 4 se revisaron los cuerpos de literatura sobre los cuales se sustenta la estructura conceptual de esta investigación. Por un lado, se ha documentado las formas de interacción proveedor-usuario y, por otro, la forma en cómo las empresas absorben información y conocimiento. Estos estudios han buscado demostrar la existencia de las interfases entre proveedores y usuarios, así como los incrementos en la capacidad de absorción. Sin embargo, las empresas ubicadas en el segmento de software hecho a la medida no cuentan con espacios para profundizar en los flujos internos, pues su desarrollo depende en buena medida de la interacción con diversos agente externos poseedores de diferentes tipos de información y conocimiento fragmentarios. Esto es parte del problema para micro y pequeñas empresas, pues no habían sido consideradas en prioridad de investigación como lo eran las medianas y grandes empresas con áreas funcionales formalmente, en particular con la de I+D. Por esta razón, a partir de esos cuerpos de literatura, esta tesis aporta empírica y teóricamente al estudio de las interfases externas (como mecanismo de identificación de la información y conocimiento) y a su importancia en los procesos de aprendizaje y en el incremento de la capacidad de absorción de las empresas ubicadas en el segmento de software hecho a la medida.

En este sentido, se adecuan los cuerpos de literatura en los cuales se apoya esta tesis, ya que proveen los elementos para explicar lo que viene sucediendo en la industria mexicana de software.

## CAPÍTULO CINCO

### METODOLOGÍA DE LA INVESTIGACIÓN

#### INTRODUCCIÓN

Este capítulo tiene como propósito explicar la manera en cómo se desarrolla esta tesis. Considerando la literatura revisada en los capítulos anteriores y las preguntas de investigación, esta tesis se basa en un *estudio de caso múltiple-exploratorio*. Los *casos* son empresas que desarrollan software hecho a la medida, tanto propietario como de fuente abierta. La *unidad de análisis* es la interfase creada entre las empresas de software y los agentes entre los años 2002-2004.

El tipo de información analizada es cualitativa y los resultados presentados se sustentan en información recopilada de los proyectos de software. La principal fuente de información consiste en entrevistas a profundidad realizadas a líderes de proyecto y desarrolladores de software ubicados en 3 empresas de software propietario y 3 empresas de software de fuente abierta. Las empresas consideradas en el trabajo de campo se ubican en las ciudades de Guadalajara, León, Monterrey y Distrito Federal. El trabajo de campo se realizó entre marzo de 2004 y enero de 2006.

En el primer apartado se describen conceptualmente la investigación cualitativa y el estudio de caso múltiple como elementos centrales de la metodología de investigación, también se presentan los elementos que permiten elegir como metodología un estudio de caso múltiple. En el segundo apartado se presenta el diseño de investigación. Todos los tipos de investigación empírica tienen implícita o explícitamente un diseño de investigación. En su definición más elemental, un diseño de investigación es la secuencia lógica que enlaza los datos empíricos obtenidos con las preguntas de investigación iniciales (Yin, 1994:19). Una vez establecidas estas interrogaciones, el siguiente paso es la obtención de datos e información y su análisis. La lógica del diseño, entonces, es vincular los datos y la información obtenida con las preguntas iniciales del estudio para después establecer las conclusiones y responder a esas preguntas. En el tercer apartado se describe la

operacionalización de los datos que sigue esta investigación: se expone la selección de escenarios, se presenta un resumen de las fuentes de información y de las formas de obtención de datos, y se explica la forma de análisis de la información recabada. En el cuarto apartado se presenta el perfil de los casos de estudio.

## **5.1 METODOLOGÍA EN LAS CIENCIAS SOCIALES**

El término metodología designa el modo en que enfocamos los problemas y buscamos las respuestas. En las ciencias sociales se aplica a la manera en que se realiza la investigación desde dos perspectivas teóricas: positivista y fenomenológica. La primera busca los *hechos o causas* de los fenómenos sociales con independencia de los estados subjetivos de los individuos. La segunda quiere entender los fenómenos sociales desde la perspectiva del actor, es decir, examina la manera en que se experimenta el mundo (Taylor y Bogdan, 1992:15).

Ambas perspectivas teóricas abordan distintos problemas y buscan diferentes respuestas, de tal forma que las investigaciones requieren distintas metodologías. El positivista, por ejemplo, busca las causas a través de métodos tales como cuestionarios, inventarios y estudios demográficos que producen datos susceptibles de análisis estadístico. En cambio, el fenomenólogo busca comprender la realidad por medio de métodos cualitativos tales como la observación participante, la entrevista a profundidad, historias de vida, entre otros, que generan datos descriptivos (Taylor y Bogdan, 1992:15-16). Esta investigación se inscribe en el segundo tipo, es decir, al tipo fenomenológico basado en la aplicación de entrevistas a profundidad con diferentes agentes.

### **5.1.1 Investigación cualitativa**

En las ciencias sociales la investigación cuantitativa o cualitativa tiene dos objetivos, *describir* y *explicar*. La relación entre éstos es interactiva y ambos dependen de las reglas de la inferencia científica. La *descripción* tiene varios aspectos fundamentales, uno de ellos conlleva un proceso inferencial, es decir, describir consiste en inferir información sobre los hechos no observados a partir de aquellos que sí se han contemplado. Otro aspecto es su

capacidad para distinguir lo que tienen de sistemático y de no sistemático los hechos observados (King, *et al.*, 1990:45).

*Explicar* -relacionar causas y efectos- es el objetivo último, sin embargo, en toda explicación es determinante describir, cuya actividad es fundamental en sí misma. Lo que distingue los estudios científicos de otros tipos de investigación no es el enfrentamiento entre descripción y explicación, sino el hecho de que se hagan inferencias sistemáticas según procedimientos válidos. La inferencia, sea descriptiva o causal, cuantitativa o cualitativa, es el objetivo último de toda ciencia social de calidad (King, *et al.*, 1990:46). El fin de la investigación ‘es obtener generalizaciones válidas, pues es la forma más eficaz de aplicar la comprensión a una amplia variedad de situaciones pertinentes’. La mejor manera de generalizar es hacerlo mediante el desarrollo de la teoría explicativa: ‘es la aplicación de la teoría la que convierte la mera recopilación de información en investigación’ (Phillips y Pugh, 2001:69).

La interpretación, al igual que la ciencia, depende de la elaboración de cuidadosas descripciones, de una comprensión profunda del mundo, de que se planten buenas preguntas, de la formulación de hipótesis falsables a partir de teorías más generales y de la obtención de pruebas necesarias para evaluar tales hipótesis. La aportación primordial de la ciencia es un conjunto de procedimientos con los que se pueden dar *respuestas* a preguntas descriptivas y causales elaboradas apropiadamente (King, *et al.*, 1990:48-49).

En este capítulo no se profundiza en las diferencias entre investigación cualitativa e investigación cuantitativa, sin embargo, se hacen tres observaciones. Primero, la investigación cualitativa se ha considerado como la base para la creación de teoría, es inductiva, subjetiva y no positivista; mientras que la investigación cuantitativa se ha considerado como medio para probar una teoría, es deductiva, objetiva y positivista (Lee, 1999:9). Este autor señala que la investigación cualitativa puede sumar ambas, la generación y prueba de teoría, mientras que Yin (1994), como veremos más adelante, argumenta lo mismo al considerar la *investigación de estudio de caso* como una de las diversas estrategias de investigación.

Segundo, la metodología cualitativa se refiere ‘a la investigación que produce datos descriptivos: las propias palabras de las personas, habladas o escritas y la conducta observable’ (Taylor y Bogdan, 1992:20). Es un modo de encarar el mundo empírico desde la perspectiva fenomenológica de la vida social, es decir, se intenta ver las cosas desde el punto de vista de otras personas.

Tercero, la investigación cualitativa es inductiva, los investigadores siguen un diseño de investigación flexible pues ven el escenario y a las personas con una perspectiva holística (donde las personas, los escenarios o los grupos no son reducidos a variables, sino considerados como un todo, pero dando importancia a la interdependencia de los elementos). Al mismo tiempo, los investigadores cualitativos subrayan la validez, mientras que los investigadores cuantitativos hacen hincapié en la confiabilidad y reproducibilidad de la investigación, aunque esto no significa que a los primeros no les preocupe la precisión de los datos.

Cuantificar produce precisión, pero no necesariamente favorece la exactitud, pues los índices cuantitativos que no se relacionen directamente con los conceptos o acontecimientos que se pretenden medir pueden conducir a graves errores de medida y a problemas en las inferencias causales. Esta es una de las razones por las que la investigación cualitativa, y en particular los estudios de caso, son esenciales para la inferencia descriptiva y, por tanto, para las ciencias sociales (King, *et al.*, 1990:55).

En el estudio particular de esta tesis, por ejemplo, la capacidad de absorción de una empresa puede ser identificada a través de metodologías de investigación cualitativas. Un enfoque cuantitativo puede proveer alguna evidencia del mejoramiento en el desempeño organizacional pero no puede demostrar el sentido en el cual las organizaciones absorben y utilizan nuevo conocimiento (Jones y Crave, 2001:19).

### 5.1.2 Investigación de estudio de caso

El estudio de caso es sólo una de varias estrategias de investigación en las ciencias sociales.<sup>1</sup> La selección para un estudio de caso está condicionada por las preguntas de investigación y la unidad de análisis. La metodología del estudio de caso es más apropiada donde las preguntas del ‘cómo’ y ‘por qué’ se responden por un fenómeno actual con múltiples fuentes que dan evidencia de ello (Yin, 1994:1). Las preguntas del ‘qué’ requiere una cuidadosa definición de términos, recopilación de información libre de parcialidades, tratamiento estadístico adecuado y un cuidadoso resumen a fin de obtener una descripción equilibrada que nos proporcione ‘un cuadro verdadero y equitativo’ (Phillips y Pugh, 2001:66), sin embargo, en el estudio de caso los investigadores no requieren control y manipulación de las variables (Lee, 1999:54).

Pero la investigación trasciende la descripción y exige el análisis, de tal forma que busca explicaciones, relaciones, comparaciones, predicciones, generalizaciones y teorías. Éstas se inscriben en el campo de las preguntas del ‘por qué’. Sin duda estas interrogantes exigen la búsqueda de información, pero ésta es usada para desarrollar la comprensión (comparándola, relacionándola, teorizando a partir de ella y sometiendo a prueba las teorías) (Phillips y Pugh, 2001:67).

Yin (1993 y 2003) muestra la importancia de usar el ‘estudio de caso’ como una estrategia de investigación y, de la misma manera que Phillips y Pugh (2001), clasifica la investigación en diferentes tipos que se pueden aplicar tanto a la investigación cuantitativa como cualitativa.

**Cuadro 5.1**  
**Tipo de estudio de caso**

Exploratoria °*
Probatoria *
Resolución de problemas*
Explicativa °
Descriptivo °

Fuente: Elaboración propia con base en ° Yin (1993 y 1994) y \* Phillips y Pugh (2001)

<sup>1</sup> Otras estrategias de investigación son la *experimentación*, *historias de vida*, *encuestas*, entre otros (Yin, 2003).

El tipo de investigación *exploratoria* aborda un problema, una cuestión o un tema nuevo de los cuales se sabe poco, se examinan qué teorías y conceptos son los más apropiados, se elaboran nuevos conceptos y, de ser necesario, nuevas metodologías, lo que implica expandir la frontera del conocimiento con la ‘esperanza’ de averiguar algo (Yin, 1994:5; Phillips y Pugh, 2001:70). A través de la investigación *probatoria* se busca descubrir los límites de las generalizaciones previamente propuestas. Las pruebas necesarias son interminables y permanentes, pues sólo así se mejoran las generalizaciones que contribuyen al desarrollo de la disciplina. Por su parte, el tipo de investigación para la *resolución de problemas* inicia con un problema concreto del mundo real y se adhieren todos los recursos intelectuales que puedan contribuir a su solución. Esto implica crear o identificar las soluciones de los problemas originales en cada etapa del camino (Phillips y Pugh, 2001:70-71). A un estudio de caso *explicativo* se asocian las preguntas ‘cómo’ y ‘por qué’, mientras que en un estudio de caso de tipo *descriptivo* se trazan secuencias personales en el tiempo, describiendo fenómenos que raramente han sido tópicos de estudios anteriores (Yin, 1994:6).

El estudio de caso presenta variaciones, pueden ser casos individuales o múltiples y cada opción depende del tipo de preguntas de investigación. La estrategia metodológica que se propone en esta tesis consiste en el estudio de caso múltiple-exploratorio.

## **5.2 DISEÑO DE LA INVESTIGACIÓN**

En el apartado anterior se describió la validez del estudio cualitativo en las ciencias sociales y las formas de llevar a cabo una investigación cualitativa. También vimos que el estudio de caso puede ser una de las estrategias de investigación cuando las preguntas están en el plano del cómo y por qué. Este es el caso de la presente tesis, por eso se utiliza la estrategia del estudio de caso múltiple-exploratorio. La fortaleza de este tipo de estrategia de investigación radica en su capacidad de incluir datos de una gran variedad de fuentes de evidencia, como observaciones directas, entrevistas sistemáticas, datos de archivo y documentación (Vera-Cruz, 2004:101).

Una vez identificada la estrategia de investigación, el siguiente paso es diseñar el estudio de caso. Todos los estudios tienen implícita o explícitamente un diseño de investigación (Yin, 1994:19), a través del cual se vinculan los datos coleccionados (y las conclusiones obtenidas) con las preguntas de investigación iniciales.

Yin (1994) y Lee (1999) muestran los principales componentes del estudio de caso, los cuales se presentan a continuación:

- a) Las preguntas de investigación: como en la investigación experimental, el estudio de caso tiende a focalizarse sobre *cómo* y *por qué* ocurren los fenómenos.
- b) Proposiciones teóricas: como en la investigación cuantitativa en general, y experimentos en particular, la probación de la teoría guía el diseño del estudio y su ejecución. En una manera deductiva, ésta debería clarificar las preguntas de investigación específicas, las variables y la evaluación interactiva y la naturaleza del análisis.
- c) Unidades de análisis: la interfase.
- d) Una vinculación lógica de los datos con las proposiciones teóricas, y
- e) El criterio de evaluación de esas proposiciones.

A diferencia de la mayor parte de los métodos, en los cuales la hipótesis y los procedimientos de los investigadores están determinados a priori, el diseño de la investigación en estudios de caso permanece flexible, tanto antes como durante el proceso real. *‘Hasta que no entramos en el campo, no sabemos qué preguntas hacer ni cómo hacerlas’*, permanecer fuera puede ocasionar que la imagen preconcebida que tenemos del fenómeno a estudiar pueda ser ingenua, engañosa o falsa (Taylor y Bogdan, 1992:32). En esta tesis, la fase piloto constituyó un elemento crucial para re-definir las preguntas de investigación, las proposiciones teóricas y la selección de los casos.

### **5.2.1 Preguntas de investigación**

El objetivo de esta tesis es describir y analizar el proceso de construcción de interfases entre empresas mexicanas desarrolladoras de software hecho a la medida y diversos agentes de su entorno, tales como empresas, universidades, gobiernos y comunidades de



desarrolladores de software de fuente abierta, y explicar cómo a partir de diferentes tipos de interfase las empresas identifican información y conocimiento existentes en el entorno y cómo influyen en el incremento de su capacidad de absorción, con el fin de generar nuevo conocimiento expresado en el diseño y desarrollo de nuevos productos, procesos y servicios. Las empresas de software construyen tres tipos de interfase: simples, semi-complejas y complejas, con cuatro agentes en particular: empresas usuarias, universidades, instituciones de gobierno y comunidades de desarrolladores. Las empresas que son capaces de identificar, evaluar, asimilar y explotar adecuadamente flujos de información y conocimiento específicos logran incrementar su capacidad de absorción, de manera que pueden generar nuevos productos.

Las preguntas de investigación que guiarán esta tesis son las siguientes:

1. ¿Cómo y por qué se construyen interfases entre las empresas de software y los diversos agentes del entorno? ¿Qué tipos de interfase se construyen y cuáles son sus características?
2. ¿En qué medida las interfases permiten a las empresas identificar información y conocimiento e incrementar su capacidad de absorción?
3. ¿Cómo se expresa un incremento de la capacidad de absorción de las empresas? ¿Qué diferencias existen entre las empresas con distinta capacidad de absorción para identificar e internalizar información y conocimiento y materializarlos en aplicaciones específicas?

### **5.2.2 Proposiciones teóricas**

En el segmento de software hecho a la medida, los sistemas de software son desarrollados a través de múltiples interacciones entre las empresas de software, usuarios y otros agentes. En este segmento industrial, el desarrollo de nuevas aplicaciones requiere de la integración de información y conocimiento que radican en diferentes agentes. A partir de esto, se formula la primera proposición:

*P1. El diseño de interfases es importante ya que la información y conocimiento son fragmentarios en un entorno complejo y dinámico como la industria de software. Son fragmentarios ya que cada agente es poseedor de una parte importante de la información y conocimiento necesarios para el desarrollo de un producto o sistema de software.*

Las empresas de software se ubican en un entorno de rápido cambio tecnológico, por tanto, deben actualizarse constantemente en términos tecnológicos para mantenerse en el mercado y una forma de hacerlo es a través de las interfases. Deben ser capaces de identificar y valorar la información y conocimiento que se genera constantemente en el entorno, internalizar lo útil a sus necesidades y explotarlos adecuadamente para generar nuevo conocimiento. Con ello, se formula la segunda proposición:

*P2. A través de las interfases externas las empresas de software pueden identificar y valorar tanto información como conocimiento específicos y adecuados para el desarrollo de nuevos productos, procesos y servicios.*

Por otro lado, la organización del trabajo de la micro y pequeña empresa mexicana de software está basada en un esquema multitareas y no en áreas funcionales formalmente como en las medianas y grandes empresas. La capacidad de absorción de las empresas de software depende de los mecanismos que desarrolle para una adecuada asimilación de la información y conocimiento existentes en el entorno y captados a través de las interfases. Con base en esto, se formula la tercera proposición:

*P3. Las empresas que desarrollan mecanismos adecuados de internalización de la información y conocimiento existentes en el entorno y captados a partir de las interfases podrán incrementar en mayor grado su capacidad de absorción y desarrollar nuevos productos, procesos y servicios.*

Como se mostró en el capítulo anterior, hay una causalidad de efectos a partir de diferentes interfases, la cual se validará en esta tesis. A partir de un tipo de interfase y de su efecto en el aumento de la capacidad de absorción de la empresa, ésta podrá crear más información de propósito general y replicar o hacer adaptaciones menores a productos ya existentes; o bien hacer mejoras incrementales a productos de software ya existentes en el mercado, diseñar y desarrollar nuevas aplicaciones, así como formar recursos humanos con habilidades específicas de acuerdo a los requerimientos de un segmento de la industria de software.

### **5.2.3 Unidad de análisis**

Como se mencionó anteriormente la estrategia de investigación a seguir en esta tesis es el *estudio de caso múltiple-exploratorio*, los *casos* son empresas que desarrollan software hecho a la medida de ambos tipos, propietario y de fuente abierta, y la *unidad de análisis* es la interfase.

La naturaleza de las preguntas de investigación y la unidad de análisis implican estudiar múltiples casos, y no uno individual, para entender el fenómeno de estudio. Ni uno ni otro caso es susceptible de generalizaciones. Sin embargo, el estudio de casos múltiples aumenta la validación externa y ayuda a prevenir interpretaciones distorsionadas en la observación (Vera-Cruz, 2004:102). Esta es la razón por la cual se eligió el estudio de caso múltiple-exploratorio como estrategia de investigación.

### **5.2.4 Proposiciones teóricas y criterio de evaluación**

La vinculación de los datos con las proposiciones teóricas y el criterio de evaluación de éstas, en sentido estricto, requieren que las hipótesis o proposiciones teóricas estén conectadas con los datos e información obtenidos en el trabajo de campo. En el último apartado de este capítulo se detallarán las formas de interpretación más usuales y la que se utilizará en esta investigación.

## **5.3 OPERACIONALIZACIÓN DE LA INVESTIGACIÓN**

De acuerdo con las definiciones de los conceptos guía proporcionadas en los capítulos 3 y 4, y en la estructura conceptual desarrollada, en este apartado se plantea la forma en cómo se harán operativos los conceptos a través de sus diferentes aproximaciones.

La estrategia de investigación ‘estudio de caso’ requiere de un programa que logre alcanzar los puntos cruciales señalados en el diseño de investigación. Los puntos considerados en la operacionalización de este trabajo son: i) la selección de escenarios, ii) las fuentes de información, iii) la obtención de datos, y iv) variables y formas de análisis de la

información. Cada uno de los conceptos principales tiene una aproximación empírica, la cual se detalla en punto 5.3.4.

### **5.3.1 Selección de escenarios**

El escenario ideal para la investigación es aquel en el cual el observador obtiene fácil acceso, establece una buena relación inmediata con los informantes y recoge datos directamente vinculado con los intereses de la investigación. Un buen estudio cualitativo combina una comprensión en profundidad del escenario particular en estudio con interacciones teóricas generales, que trascienden ese tipo de escenario (Taylor y Bogdan, 1992:36).

Siguiendo a este autor, quien está ligado a una tradición teórica particular debe estar preparado para cambiar un escenario por otro. Recomienda no aferrarse a ningún interés teórico, sino explorar los fenómenos tal como ellos emergen durante la observación. Todos los escenarios son intrínsecamente interesantes y suscitan importantes cuestiones teóricas.

#### ***Estudio exploratorio***

El trabajo de campo estuvo precedido por un estudio exploratorio, el cual fue necesario dado que el fenómeno y la forma de abordarlo son relativamente recientes. El estudio exploratorio se realizó a partir del mes de agosto de 2003 y hasta finales del 2004. Durante el proceso se entrevistaron 13 empresas y 1 asociación. En el cuadro 5.2 se especifica el nombre de la empresa/institución y su ubicación geográfica.

A través del estudio exploratorio se trató de entender el fenómeno y los dos contextos en los que las empresas mexicanas desarrolladoras de software cohabitan, permitiendo redefinir el instrumento de aproximación (guía de entrevista) conforme a los temas arrojados por la literatura revisada. En el cuadro 5.3 se muestran los casos seleccionados a partir del estudio exploratorio.

**Cuadro 5.2**  
**Empresas incluidas en el estudio exploratorio**

<b>Empresa/Institución</b>	<b>Ubicación geográfica</b>
Mi Ciudad.Com	Guadalajara, Jal.
Dawcons	Guadalajara, Jal.
E-Quality	Guadalajara, Jal.
Innox	Guadalajara, Jal.
System Haus	León, Gto.
Kernel	Monterrey, N. L.
Cyber Works	Monterrey, N. L.
Soluciones Tecnológicas	México, D. F.
Avantare Consultores	México, D. F.
PTC	México, D. F.
Kiven, S. C.	México, D. F.
Seguridata	México, D. F.
Ximian	México, D. F.
AMESOL*	México, D. F.

\*Asociación Mexicana de Software Libre.

### ***Casos seleccionados***

Los casos fueron seleccionados bajo diversos criterios. Primero, las empresas otorgaron abundante información, con la cual se enriqueció cada uno de los casos seleccionados y se obtuvo información adecuada para la descripción de los procesos de creación de interfases y de las fuentes y mecanismos para incrementar la capacidad de absorción. Segundo, el fácil acceso a estas empresas y una buena relación con los informantes permitió la recopilación de datos relacionados con los intereses de la investigación.

**Cuadro 5.3**  
**Empresas estudiadas y su ubicación geográfica**

	<b>Empresas*</b>	<b>Ubicación</b>
Empresas ubicadas en el contexto de software de fuente abierta	Empresa Manet	México D. F.
	Empresa Renoir	Guadalajara, Jal.
	Empresa Degas	León, Gto.
Empresas ubicadas en el contexto de software propietario	Empresa Goya	Monterrey, N. L.
	Empresa Constable	Guadalajara, Jal.
	Empresa Velásquez	Guadalajara, Jal.

\*Por carácter de confidencialidad, el nombre real de cada empresa ha sido cambiado.

Cabe mencionar que estos casos seleccionados son ‘ejemplares’ pues ilustran los tipos de interfase existentes. Adicionalmente, otro criterio de selección fue la diferencia en el tipo de mercado al cual destina cada empresa sus productos (como se verá en el apartado 5.4),

los diferentes contextos en los que se ubican (software propietario vs software de fuente abierta), el tamaño de las empresas (5 son micro y 1 mediana), y el tiempo de permanencia en el mercado (cuatro empresas tienen entre 4 y 5 años de creación y las otras dos cuentan con más de 10 años).

### 5.3.2 Fuentes de información

La información utilizada en esta tesis es cualitativa, obteniéndose en dos fases. La primera bajo el trabajo exploratorio, mediante el cual se trató de entender el fenómeno y la forma de abordarlo. La segunda mediante el trabajo de campo formal, que se caracteriza principalmente por la aplicación de entrevistas a profundidad a líderes de proyecto y desarrolladores de software. El objetivo consistió en recabar la información y responder a las preguntas de investigación planteadas.

Las fuentes de información más importantes utilizadas en esta investigación son de tres tipos, como se muestra en el cuadro 5.4.

**Cuadro 5.4**  
**Fuentes de información utilizadas en el trabajo de campo**

<b>Fuentes de información</b>	<b>Informantes</b>	<b>Detalle</b>
(1) Entrevistas abiertas	-Líderes de proyecto -Desarrolladores de software	Obtener información sobre: los proyectos realizados, la organización de la empresa; la forma en que interactúa con usuarios, clientes, universidades, centros de capacitación, entre otros; los canales mediante los cuales fluye información y conocimiento; identificar actividades relevantes acerca de la forma en que internalizan la información y conocimiento externo.
(2) Observación directa no participante	-Líderes de proyecto	Recopilar información acerca de las formas de interacción con usuarios, universidades y comunidades.
(3) Documentos internos y externos a la empresa	-Artículos de prensa -Reportes industriales y tecnológicos -Libros, revistas, entre otros	Obtener información adicional para complementar los argumentos sobre el proceso de creación de interfases y capacidad de absorción.

De estas tres, la fuente crucial de evidencia es la entrevista abierta, misma que servirá como ‘el cauce principal para llegar a realidades múltiples’ (Stake, 1998:63).

### 5.3.3 Recolección de datos e información

Como apunta Stake (1998:51), no existe un momento determinado para iniciar la obtención de datos, puede comenzar antes de que se haga el estudio formalmente, y aparecer como primeras impresiones o recolecciones de datos de modo informal a través de los primeros contactos que tiene el investigador con el caso. Posteriormente, las primeras impresiones serán perfeccionadas o sustituidas.

La recolección de datos e información se hizo básicamente a través de 10 entrevistas a profundidad, como se muestra en el cuadro 5.5, tanto a líderes de proyecto como desarrolladores de software. A cada uno de ellos se le aplicó más de una entrevista.

**Cuadro 5.5**  
**Entrevistas incluidas en la investigación\***

	<b>Empresas de software</b>
Líderes de proyecto	6
Desarrolladores	4
Total	10

### 5.3.4 Formas de interpretación (análisis de la información)

Al final del capítulo 4 se definieron los tres conceptos guía de esta tesis y la forma de aproximación a cada uno de ellos. En el cuadro 5.6 se presenta cómo estos conceptos guía son operacionalizados. En la segunda columna se presenta la forma de aproximación -variables- a cada uno de ellos, y en la tercera columna se presentan los medios de identificación -indicadores- (las cuales se detallan en el apartado de ‘enfoque y método’ de los capítulos 6 y 7).

Como se mencionó en el capítulo anterior, y como se muestra en el cuadro 5.6, en esta investigación se asume que las variables mantienen un grado de interacción entre ellas, lo

cual implica observar algún grado de correlación entre las mismas, aunque sin presentar los problemas observados en estudios cuantitativos.

El análisis sólo puede iniciar cuando se tienen datos que han sido obtenidos en el trabajo de campo aunque, como señala Stake (1998:67), no hay un momento determinado para hacerlo, puede comenzar a partir de la primera obtención de datos e información en el estudio exploratorio, con el propósito de dar sentido a las primeras impresiones.

**Cuadro 5.6**  
**Operacionalización de los conceptos**

<b>Concepto</b>	<b>Variable</b> <b>(Aproximación al concepto)</b>	<b>Indicador</b>
<b>Interfase</b>	Interacción con diversos agentes	Se entiende como agentes a empresas, usuarios, universidades, centros de capacitación, instituciones de gobiernos y comunidades de desarrolladores.
	Tipos y complejidad de la interacción	La complejidad de la interacción se identifica a través del tipo de proyecto, que pueden ser: a) Una simple relación de compra-venta b) Adaptabilidad de un software ya instalado c) Escalamiento, diseño y desarrollo de nuevo software, implantación del software, pruebas y mantenimiento, integración de sistemas d) Formación de recursos humanos, capacitación
	Estabilidad de la tecnología	Se identifica por el grado de estandarización de las herramientas de software utilizadas en el desarrollo de un programa o aplicación.
	Intensidad de la interacción	a) Duración de un proyecto: días, semanas, meses b) Cantidad de información intercambiada (alta, media, baja)
	Formalidad e informalidad de la interacción	a) Reuniones de trabajo documentadas y sistematizadas b) Reuniones de trabajo no documentadas ni sistematizadas
<b>Capacidad de Absorción</b>	Conocimiento previo	a) Habilidades básicas de los empleados b) Contratación de nuevo personal con educación formal
	Flujos de información y conocimiento al interior de la empresa	a) Bitácoras y sistema de procesamiento b) Foros de discusión internos c) Grupos de trabajo
	Actividades de aprendizaje tecnológico	a) Aprendizaje por reutilización b) Aprendizaje por capacitación c) Aprendizaje por actividades multitareas al interior de la empresa d) Aprendizaje por investigación y desarrollo (I+D)
<b>Conocimiento</b>	Información general	a) Revistas especializadas, internet b) Seminarios, consultoría
	Información específica	a) La capacitación interna y externa b) Usuarios
	Conocimiento general	a) Empleados contratados en el último año que han egresado de universidades o centros técnicos, pero que no han sido capacitados
	Conocimiento específico	a) Año de experiencia de los empleados b) Capacitación interna y externa c) Usuarios



En palabras de Lee (1999:32-34), la investigación cualitativa debe iniciar con el diseño del estudio y con una clara noción de cómo se perciben los datos. Los componentes del análisis en un estudio cualitativo pueden clasificarse de acuerdo a la forma de obtener los datos, ya sea *literal, interpretativa o reflexiva*.

La *interpretación literal de los datos* implica que el significado de observaciones, comentarios verbales, la percepción de los participantes y la presión contextual de la experiencia de los participantes son tomadas literalmente. La *interpretación de datos* significa que el investigador infiere lo que el participante ‘realmente quiere decir’; se basa en otras observaciones, comentarios verbales, la percepción de los participantes y la presión contextual, pero sobretodo en lo que el investigador interpreta de los significados. Finalmente, obtener *datos reflexivos* sucede cuando el investigador debe emplear más inducción e interpretación sensitiva para determinar sus significados (Lee, 1999:33-34).

Con base en los datos e información obtenida a través de las tres fuentes mencionadas arriba, en esta tesis se pretende dar sentido a determinadas observaciones de los casos a través de una interpretación reflexiva a partir de la descripción de los casos.

#### **5.4 PERFIL DE LOS ESTUDIOS DE CASO**

A partir de entrevistas realizadas a líderes de proyecto y desarrolladores de software de las 6 empresas elegidas, se obtuvo la información para responder a las preguntas de investigación de esta tesis se obtuvo.

Como se mencionó anteriormente, la naturaleza tecnológica de cada de una las empresas es distinta: Manet, Renoir y Degas se ubican en un contexto de software de fuente abierta, mientras que Goya, Constable y Velásquez se ubican en un contexto de software propietario. Algunas características organizacionales y productivas son similares entre ellas, por eso tienen patrones comunes de organización y administración. En los siguientes párrafos se describe el perfil de los estudios de caso. Cabe mencionar que, por carácter de confidencialidad, el nombre de cada una ha sido cambiado por seudónimos.

#### **5.4.1 Empresas ubicadas en un contexto de software de fuente abierta**

##### ***Manet***

La empresa Manet surge en 1997 en la ciudad de México como empresa de desarrollo de software de fuente abierta con capital propio. Cuenta con 4 socios, su fundador principal es ingeniero químico y anteriormente había trabajado en una empresa desarrollando aplicaciones de software para el sector de servicios.

Manet usa sistemas operativos como LINUX o GNOME<sup>2</sup> y se especializa en el uso de tecnología WEB e interfases gráficas.

El nicho de mercado al que destina sus actividades de desarrollo es variado. Las aplicaciones se han dirigido a pequeñas y medianas empresas ubicadas en el sector de servicios (farmacias, despachos contables) y en menor medida a empresas de manufactura (textil). Los proyectos llevados a cabo por Manet son desarrollo de aplicaciones contables y administrativos y páginas WEB, tanto para el sector de servicios como para el de manufactura.

##### ***Renoir***

La empresa Renoir inicia operaciones en 2002 en la ciudad de Guadalajara, Jal. La empresa cuenta con 9 personas, en su mayoría ingenieros industriales e ingenieros en sistemas, además de programadores. El capital con que inicia la empresa proviene de los ahorros personales de los tres socios fundadores. El rango de facturación anual en 2004 fue de 1.5 millones de pesos.

El lenguaje de programación más usado por la empresa es PHP, un lenguaje de programación utilizado por desarrolladores de software de fuente abierta. Renoir se enfoca a realizar proyectos de software hecho a la medida y a implementar sistemas de código abierto, principalmente con ayuda de la tecnología WEB con elementos de código abierto.

---

<sup>2</sup> Sistema operativo de fuente abierta desarrollado a finales de la década de los noventa en México por un ingeniero mexicano.

Esta empresa se enfoca a una tecnología y no a un nicho de mercado, es decir, las industrias de destino de sus productos y servicio son variados: la industria de TIC, manufactura, gobierno y servicios. Por su especialización, es de las pocas que desarrollan en plataformas de código abierto en Guadalajara.

### ***Degas***

A partir de 2002 la empresa Degas comienza su labor en la Ciudad de León, Guanajuato, bajo capital mexicano y brasileño. Cuenta con 4 trabajadores, tres ingenieros y un programador. Su rango de facturación en 2004 se ubico alrededor de 1.8 millones de pesos.

Esta empresa utiliza tecnología JAVA y ORACLE y sistema operativo LINUX. La empresa no tiene como estrategia tecnológica usar exclusivamente herramientas de software de fuente abierta, para algunos proyectos utiliza software propietario como ORACLE (específicamente bases de datos). Es decir, hay una combinación en el uso de plataformas. Los proyectos de software que realiza son de dos tipos: hechos a la medida y empaquetados. El software empaquetado es de uso general, pero específico a un sector, es desarrollado en Brasil y modificado o adaptado a las empresas usuarias mexicanas.

El nicho de mercado al que destina sus productos es el de manufactura, en particular el sector de tenerías, sobre el cual el personal de Degas tiene un amplio conocimiento acerca de los procesos industriales. Dos de sus ingenieros anteriormente trabajaron en empresas de tenería, llevando a cabo actividades relacionadas con el proceso de curtido, después se especializaron en software y ahora participan en Degas desarrollando proyectos de software para tenerías. La particularidad de esta empresa es la especialización en proyectos de desarrollo de software específico para empresas de tenería.

## **5.4.2 Empresas ubicadas en un contexto de software propietario**

### ***Goya***

La empresa Goya abre sus servicios de TIC en 1989 en Monterrey, N. L., cuenta con 150 empleados, de los cuales el 30% son ingenieros en sistemas. El monto de facturación en 2004 fue de 5 millones de dólares.

Esta empresa trabaja con tecnología JAVA y los desarrollos que realiza son hechos a la medida para diversos sectores industriales, pero principalmente proyectos de 'punto de venta' para el sector servicios.

Goya tiene un socio tecnológico que no aporta capital físico sino conocimiento en tecnologías específicas para el desarrollo de proyectos. Los primeros 7 años los dedicó a desarrollo de software para servicios, lo que representaba entre el 80 y 90% de las ventas de la empresa en esos años. En 1999 se abrieron unidades nuevas y actualmente el 30% de las ventas de la empresa provienen del desarrollo de software, el resto se debe a las aplicaciones de tipo ingenieril (herramientas de Borlan, entre otras). Estas aplicaciones no son masivas, son específicas ya que solamente lo utilizan entre 5 y 6 usuarios.

### ***Constable***

Surge en 2001 en la ciudad de Guadalajara, Jal., con recursos personales de su fundador. Hasta 2004 contaba con 9 empleados, de los cuales 6 eran ingenieros en electrónica y sistemas, 1 diseñador gráfico y 2 programadores. El monto de facturación en ese año fue de 1.5 millones de pesos.

El tipo de software que realiza es hecho a la medida, y la tecnología que usa en el desarrollo de proyectos es PUNTO NET para la realización de páginas WEB. El destino de sus productos es variado, puede ser cualquier empresa de todo sector necesitado de tecnologías WEB.

### ***Velásquez***

Esta empresa surge en 2002 en la ciudad de Guadalajara, Jal., con capital derivado de ahorros personales de los socios fundadores. Cuenta con 10 empleados, de los cuales 5 son ingenieros (uno con doctorado) y el resto son programadores especializados en prueba de software. En 2004 y 2005 facturó alrededor de 1.5 millones de pesos, el 80% corresponde a pruebas de software y el resto a servicios de consultoría y administración de proyectos.

La empresa se especializa en probar programas de software hechos por otras empresas. Es decir, Velásquez no desarrolla un programa o aplicación para un uso específico sino examinar a través de procesos sistematizados si el programa a utilizar o que está utilizando una empresa cuenta con los requerimientos mínimos de confiabilidad y eficiencia, de tal forma que el destino de sus servicios son empresas de TIC, en particular aquellas desarrolladoras y usuarias de software ubicadas en diferentes sectores industriales y tecnológicos.

Velásquez es una de las pocas empresas especializadas en prueba de software en México. Algunas de las empresas desarrolladoras de software tienen un área de prueba de software al interior de la misma, sin embargo, las empresas usuarias no cuentan con esa área. Por eso, Velásquez otorga el servicio a empresas usuarias de software o a aquellas que lo desarrollan y que no tienen un área de prueba o que hace outsourcing para esta actividad.

#### **5.4.3 Diferencias y similitudes entre los casos**

En el cuadro 5.7 se muestran las diferencias y similitudes entre los 6 casos de estudio. Una de las características más significativas entre las empresas es el uso de plataformas tecnológicas de código abierto versus propietarias. Las primeras tres empresas (Manet, Renoir y Degas) se caracterizan por usar tecnologías basadas en software de fuente abierta (SWFA), además usan empresas usan sistemas operativos como LINUX, GNOME o SAMBA,<sup>3</sup> así como herramientas basadas en software de fuente abierta tales como PHP o JAVA.

Técnicamente, la primera gran diferencia es el uso y difusión de los códigos fuente. A partir del conocimiento de codificación y de su acceso abierto, los desarrolladores de software pueden hacer modificaciones menores o radicales para su uso particular o colectivo.<sup>4</sup> El código, una vez creado, queda liberado a los usuarios para hacer uso de él. No hay un criterio de propiedad sobre los códigos, circulan libremente y no pueden ser sujetos de

---

<sup>3</sup> Sistema Operativo de origen brasileño.

<sup>4</sup> En el próximo capítulo se describe y explica una ‘comunidad de desarrolladores de software libre’ y la forma de trabajo colectivo.

transacción económica. Los desarrolladores o empresas que trabajan con códigos abiertos o software de fuente abierta no comercializan el código sino el proyecto o programa que desarrollan a partir de ellos. A la empresa usuaria le venden el programa o aplicación con código abierto, de tal forma que puede modificarlo con base en sus necesidades, pero la empresa usuaria requerirá de personal con conocimientos y habilidades en el uso de lenguajes y herramientas de software para esa actividad.

Las otras tres empresas (Constable, Velásquez y Goya) usan plataformas o sistemas operativos propietarios, es decir, el código fuente no está 'disponible' y por tanto, los usuarios no pueden acceder al código fuente y hacer modificaciones por sí mismos. Los sistemas propietarios mantienen sus códigos fuente cerrados, los cuales tienen un costo ya sea por arrendamiento o compra.

Trabajar con una u otra plataforma tecnológica plantea diferentes tipos de estrategia de negocios, productiva y tecnológica. En el capítulo 1 se mencionó que hasta la década de los ochenta las empresas trabajaban exclusivamente con sistemas operativos y herramientas propietarias, pero a partir de los años noventa el uso de sistemas operativos libres y las herramientas basadas en código abierto empezaron a tomar un gran auge, en particular el uso de LINUX, desarrollado una década antes en la academia.

El uso comercial del software de fuente abierta es reciente y empieza a crecer entre los sectores productivos tanto tradicionales como de alta tecnología, a nivel global y local. Es parte de la estrategia tecnológica de algunas empresas para posicionarse en el mercado, incluso de la estrategia inicial de desarrollo de la industria de software de algunas entidades como Jalisco.

Algunas empresas ubicadas en un contexto de software de fuente abierta tienen como estrategia tecnológica usar sistemas operativos y herramientas basadas en código abierto y avanzar en el uso de plataformas y herramientas propietarias, como lo ha hecho la empresa Degas. Ésta, en sus primeras etapas de crecimiento, realizó proyectos usando LINUX (sistema operativo basado en SWFA) pero después empezó a combinarlo con el uso de

herramientas basadas en software propietario (SWP), tales como ORACLE ya que tiene mayor grado de estandarización y cuenta con soporte técnico. Sin duda, al usar herramientas propietarias entra en juego el factor costo, ya sea por arrendamiento o compra. Sin embargo, Degas es una empresa que tiene como estrategia utilizar plataformas y herramientas basadas en software de fuente abierta, pero combinando proyectos que requieren ambos tipos de tecnología.

**Cuadro 5.7**  
**Resumen de las diferencias y similitudes entre los estudios de caso**

Característica		Empresas de SWFA	Empresas de SWP	Grado de diferencia o similitud
Tecnología	Sistemas operativos	GNOME, LINUX	WINDOWS	Alta
	Herramientas	PHP, JAVA	JAVA, ORACLE	Media
Tipo de software desarrollado		Hecho a la medida	Hecho a la medida/empaquetado	Baja
Certificación de procesos		CMM3 (1 empresa)	CMM3 (3 empresas)	Alta
Tamaño de la empresa		Pequeñas	Pequeñas y mediana	Media
Origen de capital		Ahorros personales	Ahorros personales	No hay
Nicho de mercado		Servicios, manufactura, gobierno	Servicios, manufactura, educación	Baja

Fuente: Elaboración propia con base en entrevistas.

Otra característica de las empresas en estudio es el tipo de proyectos de software que realizan. En contraste con la primera característica, en esta segunda hay pocas diferencias en términos del tipo de software que desarrollan. Las empresas en estudio muestran una tendencia creciente hacia el desarrollo de software hecho a la medida con ambos tipos de plataforma tecnológica: de fuente abierta y propietario.

Las empresas en estudio no desarrollan software de propósito general o empaquetado, salvo Degas que lo hace para propósito general pero dirigido al sector de tenerías y siempre adaptado a los requerimientos de la empresa usuaria.

En relación al origen de capital, todas las empresas en estudio iniciaron actividades con recursos propios y ninguna de ellas ha accedido a capital de riesgo o préstamos de otro tipo. Esto llama la atención particularmente porque implica que no hay créditos para empresas donde el principal insumo es la información y el conocimiento, a diferencia de los créditos

tradicionales basados en garantías físicas o financieras. Pero la evidencia ha mostrado la posibilidad de las empresas por crecer productivamente con recursos financieros propios, y asociado a otros factores (como se verá en los próximos capítulos). Aunque tres de ellas (Constable, Velásquez y Renoir) han obtenido apoyos económicos e institucionales para la certificación de sus procesos en CMMI<sup>5</sup> nivel 3.

Cuando las empresas son pequeñas no cuentan con una estructura organizacional semejante al de las empresas medianas o grandes, es decir, no tienen áreas funcionales formalmente. Las empresas de software en México tienen esta particularidad, en las cuales el propietario es al mismo tiempo líder de proyecto, diseñador, programador, ejecutivo de ventas, entre muchas otras actividades a realizar. A diferencia de las empresas medianas como Goya, que cuenta con un área para cada una de las actividades mencionadas que le permite profundizar en la curva de aprendizaje, sobre todo en áreas técnicas tales como programación, prueba de software, entre otros. Sin embargo, las pequeñas empresas han podido mantenerse en el mercado por diversas razones, dos de las cuales son importantes: (i) porque el nicho de mercado al que destinan sus productos y servicios es relativamente nuevo (para el caso Velásquez, por ejemplo) y hay pocos competidores (para el caso de Degas, por ejemplo), y ii) porque las empresas han acumulado cierto grado de capacidad de absorción en cada uno de los nichos de mercado específicos de su conveniencia, lo que les ha permitido absorber nueva información y conocimiento del entorno y transformarlo en nuevos productos, servicios y procesos, tal como se mostrará en los capítulos siguientes.

Velásquez se dedica a una actividad poco común como unidad de negocio: ‘prueba de software’. Esta empresa tiene como estrategia productiva cubrir este segmento casi virgen en México, aunque, al mismo tiempo, presenta grandes retos. El problema al que se enfrenta es el de un mercado por ahora estrecho, ya que las empresas desarrolladoras hacen por sí mismas pruebas de software (por lo regular con poca confiabilidad, pues no es su especialidad) y las empresas usuarias de software no tienen los mecanismos adecuados para verificar que el programa o aplicación de software que están adquiriendo sea confiable.

---

<sup>5</sup> Por sus siglas en inglés: Capability Maturity Model Integration (Modelo Integral de Maduración de Capacidades).



Por su parte, la empresa Degas tiene como usuarios a empresas ubicadas en un sector tradicional, especialmente el de tenerías, cuyos procesos de producción han migrado al uso de TIC, en particular al uso de software; esta empresa se especializa en la implantación de sistemas para este sector. La empresa logró esta capacidad debido a su trayectoria de aprendizaje sobre dichos procesos ya que, como se mencionó en párrafos anteriores, el personal que ahora desarrolla e implanta esos sistemas trabajó anteriormente por varios años en las actividades productivas de tenerías.

Otra de las características, igual de importante que se toma en consideración, es la certificación de procesos de desarrollo de software. No todas las empresas, sobre todo las pequeñas, tienen la posibilidad de ser certificadas en sus procesos. Sin embargo, en un contexto de SWP las 3 empresas cuentan con certificación de procesos, mientras que en el contexto de SWFA sólo hay una empresa con certificación, Renoir. La certificación que obtuvieron es CMMI nivel 3, la cual les llevó 3 años alcanzarla y les costó entre 25 mil y 30 mil dólares a cada empresa.<sup>6</sup>

Renoir, Costable y Velásquez terminaron el proceso de certificación a finales de 2005. En este sentido, hay una gran diferencia entre las empresas en términos del acceso a recursos para obtener certificación en sus procesos. La razón se debe a que las empresas que están en proceso de certificación tienen como objetivo de mediano plazo insertarse en el mercado de exportación (principalmente en el mercado estadounidense) y uno de los primeros requisitos de competencia es tener certificado de calidad en los procesos en CMMI nivel 3 o superior.<sup>7</sup>

#### **5.4.4 Algunos criterios a considerar en la configuración de interfases**

Al interior de la industria mexicana de software existe una diversidad importante en términos de la tecnología utilizada en el desarrollo de software. Como se mencionó en el

---

<sup>6</sup> Entrevista con líderes de proyecto, empresas Constable, Velásquez y Goya, Renoir.

<sup>7</sup> El nivel máximo de certificación que pueden alcanzar las empresas de software es CCM nivel 5. Los sistemas de calidad han pasado de ser simples mecanismos que aseguran la repetición eficiente de operaciones a ser plataformas sobre las cuales se construyen sistemas de administración de la tecnología. Esto ha permitido a las empresas desarrollar sistemas de 'cero' defectos y ha originado un cambio en sus nichos de mercado.

capítulo anterior, hay al menos dos contextos tecnológicos sobre los que se basa el desarrollo de software hecho a la medida: software de fuente abierta (SWFA) y software propietario (SWP). La diversidad tecnológica no sólo es parte de la preferencia tecnológica de los ingenieros y desarrolladores por usar cualquiera de ellas, también puede ser parte de una estrategia técnica, de costos o competitiva.

A pesar de un incremento en el uso de SWFA por parte de las instituciones públicas y privadas, varias de las empresas que desarrollan este tipo de software se enfrentan a un mercado aún estrecho para el uso de aplicaciones comerciales basadas en fuente abierta. El nicho de mercado más rentable en la actualidad es el de servidores o estaciones de trabajo utilizados por medianas y grandes empresas, al cual han destinado una gran parte de sus esfuerzos, conocimientos y recursos, quedando atrás, casi virgen, un campo de acción masivo: aplicaciones para las computadoras de escritorio (microcomputadoras) o PDA's, por ejemplo, en donde el software propietario es dominante en aplicaciones para estas plataformas.

En el capítulo 1 se mencionó que el SWFA ha tenido una importante aplicación en el 'software incorporado' (embedded software), es decir, en el desarrollo de aplicaciones o programas para productos de manufactura. Como veremos en los siguientes capítulos, Degas y Renoir han elaborado programas para industrias de manufactura y servicios con la finalidad de mejorar el proceso de producción y el producto (en el sector de tenerías, para el primer caso) o el servicio público (creación de un sistema de información que agiliza la toma de decisiones en relación a problemas forestales, para el segundo caso). Esto muestra que las empresas privadas e instituciones públicas están incorporando software (embedded software) en productos y servicios.

Por otro lado, el SWFA se enfrenta a un marco institucional aún mucho más joven que el que presenta el contexto de SWP. Es decir, no hay una infraestructura institucional de regulación adecuada o lo suficientemente madura para fomentar el desarrollo y crecimiento de las empresas de SWFA en términos de créditos o programas de fomento y financiamiento, entre otros.

A pesar de estas limitaciones, las pocas empresas de SWFA que existen en el país se han posicionado en nichos de mercado específico confrontado la estrategia de las empresas ubicadas en el contexto de SWP. En los próximos capítulos se mostrará que empresas como Renoir, Degas y Manet han elaborado proyectos de software específico para industrias de manufactura y el sector gobierno.

Aunque el mercado de software es relativamente joven en México, las empresas de software ubicadas en un contexto de SWP, como Constable, Velásquez o Goya, encuentran mejores posibilidades de desarrollo y una mejor infraestructura de soporte técnico que las empresas de SWFA. En este sentido, la diferencia fundamental es que la industria de software propietario a nivel nacional e internacional tiene una infraestructura de soporte mucho más articulada y madura para sistemas operativos, herramientas y aplicaciones. MICROSOFT, por ejemplo, tiene una división que da soporte y mantenimiento al sistema operativo WINDOWS, diversas herramientas y aplicaciones, las 24 horas del día y los 365 días del año, aparentemente sin costo alguno (el costo de este servicio está incorporado en la venta del producto), lo mismo pasa con las bases de datos de la empresa ORACLE y con cientos de herramientas o aplicaciones desarrolladas que tienen detrás una infraestructura de soporte técnico. Las empresas de SWFA como Renoir, Degas y Manet tienen que dar por sí mismas ese soporte, ya que no cuenta con la misma infraestructura que tiene WINDOWS, ORACLE o IBM, de tal forma que el costo de ese soporte técnico es alto y reduce el uso masivo de aplicaciones basados en SWFA.

Bajo esta visión, el mercado del SWP parece estar más organizado que el de SWFA. A esto se añade que las empresas usuarias de software tienen un mayor referente de los sistemas operativos, herramientas y aplicaciones propietarias, debido en gran medida a las campañas de mercadotecnia que pueden financiar las grandes empresas.

Además, hay empresas que buscan certificarse en calidad de procesos como parte de su estrategia para enfrentar al mercado; algunas con apoyo del gobierno local (Jalisco), otras con recursos propios, pero todas se esfuerzan por alcanzar niveles de calidad, confiabilidad, productividad y competitividad para penetrar en los mercados de exportación

(principalmente el mercado estadounidense). Esta conducta no se observa en las empresas ubicadas en un contexto de SWFA, salvo Renoir que está en el mismo proceso de certificación.

### **PARTE III**

## **DISEÑO DE INTERFASES, CAPACIDAD DE ABSORCIÓN Y GENERACIÓN DE CONOCIMIENTO: EVIDENCIA EMPÍRICA**

La Parte III se compone de los capítulos 6, 7 y 8. En el capítulo 6 se presenta evidencia empírica y análisis acerca de cómo las empresas que desarrollan software hecho a la medida crean diferentes tipos de interfase con distintos agentes ubicados en entornos tecnológicos dinámicos y estables. En el capítulo 7 se presenta evidencia empírica y análisis acerca de cómo las empresas de software absorben información y conocimiento existentes en el entorno y generados a partir de las diferentes interfases. En el capítulo 8 se presenta evidencia empírica y análisis acerca de cómo, a partir de las diferentes interfases creadas en un entorno de rápido cambio tecnológico y de su efecto en la capacidad de absorción, las empresas de software generan nuevo conocimiento expresado en recientes productos, servicios y procesos a nivel de empresa e industria. Esta causalidad de efectos se valida en los 3 capítulos mencionados.

## **CAPÍTULO SEIS**

### **CREACIÓN DE INTERFASES SIMPLES, SEMI-COMPLEJAS Y COMPLEJAS ENTRE LAS EMPRESAS Y OTROS AGENTES**

#### **INTRODUCCIÓN**

En este capítulo se presenta evidencia empírica y análisis acerca de cómo las empresas que desarrollan software hecho a la medida crean diferentes tipos de interfase. Dentro del primer apartado, se muestra el enfoque y método utilizado para caracterizar el proceso de construcción de interfases. En cada uno de los siguientes tres apartados se expone evidencia empírica y el análisis de la construcción de interfases: en el segundo apartado se aborda la construcción de interfases simples, en el siguiente se aborda la construcción de interfases semi-complejas, y en el cuarto apartado se aborda la construcción de interfases complejas. Las categorías que se usarán para analizar las diferentes interfases son las siguientes: i) tipos de agente, ii) tipos y complejidad de la interacción, iii) estabilidad de la tecnología utilizada en un proyecto, iv) intensidad de la interacción, y v) formalidad vs informalidad de la interacción. En este capítulo, además, se mostrará que las seis empresas sobre las cuales se basa el estudio interactúan con los siguientes agentes: empresas usuarias, universidades, gobiernos estatales y comunidades de desarrolladores de software de fuente abierta. Las empresas en estudio han establecido algún tipo de interfase con al menos uno de estos agentes. Los mecanismos más importantes en la construcción de interfases son: a) la interacción proveedor-usuario, b) la vinculación universidad-empresa y c) la interacción entre las empresas de software y las comunidades de desarrolladores de software de fuente abierta. Finalmente, se presenta el resumen y las conclusiones del capítulo.

#### **6.1 ENFOQUE Y MÉTODO**

A continuación se presenta el enfoque y método utilizado para caracterizar el proceso de construcción de interfases entre las empresas de software y diversos agentes del entorno. El concepto principal utilizado en este capítulo es interfase y que, en esta tesis, tiene una

connotación más amplia que aquel concepto definido por Andersen (1991 y 1996), lo cual se discutió en el apartado 3.3 y se redefinió en el 4.3.

La información acerca de la construcción de interfases fue obtenida principalmente a través de entrevistas abiertas con líderes de proyecto y desarrolladores de software. La compilación de información cualitativa permitió la reconstrucción de eventos acerca del proceso de creación de interfases, particularmente sobre la interacción proveedor-usuario, la vinculación universidad-empresa y la interacción entre empresas de software y comunidades de desarrolladores de software de fuente abierta, asociadas a la elaboración de proyectos de desarrollo de software hecho a la medida. Estas descripciones son la base para analizar el proceso de creación de interfases. Las categorías analíticas consideradas para su creación son las siguientes:

### ***Tipos de agente***

Esta categoría se refiere al tipo de agente con el cual la empresa de software interactúa. Se entiende como agente a “empresas usuarias, universidades, gobiernos y comunidades de desarrolladores”. Estos son los agentes más importantes, aunque las empresas de software han establecido algún tipo de interacción con otros más, tales como la Empresa Integradora, centros de capacitación, consejos de ciencia y tecnología estatales y usuarios individuales. Según el tipo de proyecto, en cada interacción la empresa de software desarrollará actividades particulares. Se identifica el tipo de agente que participó en los proyectos más representativos para la empresa de software durante el periodo 2002-2004.

### ***Tipos y complejidad de la interacción***

El tipo de interacción se identifica principalmente a través de las diferentes actividades realizadas en cada proyecto de desarrollo de software. Por ejemplo, en un primer nivel, un proyecto simple puede tener como actividad principal la instalación de un programa de software que ya existe en el mercado, con pequeñas modificaciones, lo cual se traduce en una actividad de ‘compra-venta’. En un segundo nivel la interacción tiene un grado de complejidad ya que la actividad principal puede consistir en la ‘adaptación o escalamiento’

de un programa o aplicación de software ya existente en el mercado, implica mejoras incrementales, así como la capacitación a usuarios. En un tercer nivel las actividades principales consisten en el “análisis, diseño y desarrollo de nuevo software”, “implantación y despliegue del programa en todas las áreas de la empresa usuaria”,<sup>1</sup> actividades que hacen de la interacción aún más compleja.

En otro nivel de complejidad, las actividades se asocian a la vinculación entre la empresa y universidades, centros de capacitación y Empresa Integradora; las actividades particulares en esta vinculación están asociadas a la “formación de recursos humanos”, programas de “capacitación” y fomento de desarrollo de la industria, respectivamente.

La interacción entre las empresas de software y las comunidades de desarrolladores de software libre es otro nivel de complejidad. Las actividades particulares de este tipo de interacción están asociadas al desarrollo de nuevos algoritmos de programación o nuevos códigos, nuevos módulos, mejoramiento o escalamiento, entre otras actividades.

En cada tipo de proyecto se identifican *necesidades y requerimientos* de los agentes, los cuales son *simples* si el proyecto implica una relación de compra-venta o adaptabilidad de software ya instalado, *complejos* si el proyecto incluye escalamiento, diseño y desarrollo, pruebas o formación de recursos humanos especializados, *codificados* si están sistematizados en diagramas, documentos, etcétera, y *no codificados* si están incorporados en los agentes y no han sido sistematizados.

### ***Estabilidad de la tecnología***

La estabilidad de la tecnología se identifica por el “grado de estandarización de las herramientas de software utilizadas en el desarrollo de un programa o aplicación”. La estabilidad de la tecnología de software utilizada en un proyecto es valorada cualitativamente de la siguiente manera: *alta*, cuando las tecnologías de software son estables y difundidas entre las empresas de la industria; *media*, cuando aquellas tienen un

---

<sup>1</sup> Como se muestra en el Anexo Uno, estas actividades son parte del modelo de desarrollo de software hecho a la medida.



bajo nivel de estabilidad pero son difundidas; y *baja* cuando las tecnologías de software tienen un bajo nivel de estabilidad y no son ampliamente difundidas.

Las categorías ‘tipos y complejidad de la interacción’ y ‘estabilidad de la tecnología’ están relacionadas por dos razones. Primero, porque el tipo de proyecto implica diferentes actividades con diferentes agentes y la utilización de distintos tipos de tecnología de software. Segundo, el aumento en la complejidad del tipo de proyecto implica interacciones más complejas entre los agentes.

### ***Intensidad de la interacción***

La intensidad de la interacción se identifica a través de la “duración del proyecto, medido por días, semanas o meses” y por la “cantidad de información intercambiada”. Esta categoría es valorada cualitativamente de la siguiente manera: *alta*, cuando la interacción entre la empresa de software y los diversos agentes del entorno es intensa y de larga duración (en términos de número de meses o años); *media*, cuando la interacción es intensa pero de corta duración (en términos de número de semanas o meses); y *baja* cuando es débil y de corta duración (en términos de número de días o semanas, en algunos casos pocos meses). Una interacción alta es indicativa de una constante participación de la empresa de software y del usuario (u otro agente) en las actividades de análisis y diseño de los proyectos de software, y en la capacitación del usuario para el uso del producto/sistema desarrollado. Una interacción media significa una inconstante pero intensa participación entre los agentes, la empresa de software capacita a los usuarios para el uso del producto/sistema. Una interacción baja muestra una inusual participación del usuario en las actividades propias del proyecto de software, pocas veces requiere ser capacitado.

### ***Formalidad vs informalidad de la interacción***

La formalidad de la relación se identifica por el número de reuniones de trabajo documentadas y sistematizadas durante el proyecto, es decir, la formalidad de la interacción implica un proceso de documentación y sistematización del intercambio de la información y conocimiento, a través de la bitácora de trabajo, por ejemplo, mientras que la

informalidad implica la ausencia de ello. Las categorías “intensidad de la interacción” y “formalidad vs informalidad de la interacción” se describen y analizan en un mismo apartado.

La forma de aproximación al concepto y las categorías de medición se resumen en el cuadro 6.1.

**Cuadro 6.1**  
**Operacionalización de la interfase**

Variable (Aproximación al concepto)	Indicador	Categoría de medición
Interacción con diversos agentes	Se entiende como agentes a empresas, usuarios, universidades, centros de capacitación, instituciones de gobiernos y comunidades de desarrolladores.	Se identifica el tipo de <i>agente</i> que participa en los proyectos más representativos para la empresa de software durante el periodo 2002-2004. Pasivos: no se involucran en el diseño, desarrollo de proyectos. Pro-activos y propositivos: participan en el diseño y desarrollo del proyecto Activos y conscientes: participan en el diseño y desarrollo del proyecto y además tienen habilidades en el manejo de herramientas de software
Tipos y complejidad de la interacción	La complejidad de la interacción se identifica a través del tipo de proyecto, estos pueden ser: a) Una simple relación de compra-venta b) Adaptabilidad de un software ya instalado c) Escalamiento, diseño y desarrollo de nuevo software, implantación del software, pruebas y mantenimiento, integración de sistemas d) Formación de recursos humanos y capacitación	En cada proyecto se identifican las <i>necesidades</i> y <i>requerimientos</i> de los agentes que pueden ser: Simples: cuando el proyecto implica una relación de compra-venta o adaptabilidad de software ya instalado Complejos: cuando el proyecto implica diseño y desarrollo, pruebas o formación de recursos humanos especializados Codificados: cuando están sistematizados en diagramas, documentos, etcétera No codificados: están incorporados en los agentes y no han sido sistematizados
Estabilidad de la tecnología	Se identifica por el grado de estandarización de las herramientas de software utilizadas en el desarrollo de un programa o aplicación.	Alta: si las tecnologías de software son estables y difundidas entre las empresas de la industria Media: si las tecnologías de software tienen un bajo nivel de estabilidad pero son difundidas Baja: si las tecnologías de software tienen un bajo nivel de estabilidad y no son ampliamente difundidas
Intensidad de la interacción	a) Duración de un proyecto: días, semanas, meses b) Cantidad de información intercambiada (alta, media, baja)	Alta: si la interacción entre la empresa de software y los diversos agentes del entorno es intensa y de larga duración (en términos de número de meses o años) Media: si la interacción es intensa pero de corta duración (en términos de número de semanas o meses) Baja: si la interacción es débil y de corta duración (en términos de número de días o semanas, en algunos casos pocos meses)
Formalidad e informalidad de la interacción	Proceso de documentación y sistematización del intercambio de la información y conocimiento	Formal: implica un proceso de documentación y sistematización del intercambio de información y conocimiento utilizada en el proyecto Informal: ausencia de procesos de documentación y sistematización del intercambio de información y conocimiento utilizados en el proyecto

## 6.2 CREACIÓN DE INTERFASES SIMPLES

El primer tipo de interfase que se describe y analiza en esta tesis es una interfase simple, cuya interfase permite a las empresas de software identificar y obtener información del entorno acerca de procesos tecnológicos estandarizados y estables.

“Cuando un proyecto consiste en vender e instalar una aplicación o programa de software, la empresa usuaria mantiene una actitud receptora de la información que la empresa proveedora de software pueda proporcionarle acerca del programa. El usuario da por hecho que el proveedor sabe todo acerca de los procesos, por ejemplo, de la recepción de insumos, inventarios, ventas, etcétera, pero, aunque el proveedor se ha especializado en programas que permiten la optimización de esos procesos, siempre es necesario entender qué es lo que exactamente requiere el cliente. La empresa proveedora puede conocer a priori en cierta medida lo que el usuario requiere, pero es imprescindible conocer las particularidades ya que aun siendo un programa que lo usa más de una empresa, como los Sistemas Integrados de Gestión [conocidos como ERP, por sus siglas en inglés]<sup>2</sup>, en un mismo tipo de empresa el usuario puede requerir cosas diferentes, aunque esto solamente implique modificar ciertos iconos.”<sup>3</sup>

La información que la empresa de software identifica y obtiene por medio de una interfase simple es acerca de procesos estandarizados, sobre todo organizacionales, de empresas ubicadas en diversos sectores industriales. Por ejemplo, una empresa de servicios como las farmacias o una que produce calzado o que confecciona ropa tienen una estructura organizacional que se ha establecido como patrón entre una gran cantidad de empresas dentro de su industria. Siendo así, la información y conocimiento generados alrededor de esa estructura organizacional es de uso común entre los gerentes de las empresas y se difunden a través de medios codificados tales como revistas, folletos, libros de texto, seminarios, etc.

---

<sup>2</sup> Enterprise Resource Planning (ERP) es un Sistema Integrado de Gestión, tiene el objetivo de facilitar la gestión de todos los recursos de la empresa a través de la integración de la información de los diferentes departamentos y áreas funcionales (finanzas, proyectos, recursos humanos, nómina y mantenimiento, distribución y manufactura).

<sup>3</sup> Entrevista con desarrollador de software, empresa Manet.

En ese sentido, en una interfase simple se utiliza información y conocimiento acerca de productos, servicios y procesos que son de uso común entre las empresas de una industria, de forma que una empresa de software puede identificar información acerca de las estructuras organizacionales estandarizadas de empresas ubicadas en diversas industrias. Esa información y conocimiento son transformados al interior de la empresa de software en productos tales como aplicaciones o programas de administración o gestión de recursos.

Por otro lado, algunas empresas como Constable, Velásquez y Renoir mantienen una forma de comunicación con el entorno a través de diferentes agentes, además del usuario, con la denominada Empresa Integradora. Una Empresa Integradora se define como una organización empresarial que asocia a micro, pequeña y medianas empresas formalmente constituidas con el objeto de prestar servicios especializados a sus socios tales como: i) gestión de financiamiento, ii) compra de manera conjunta de materias primas e insumos para reducir costos, y iii) vender de manera integrada la producción.<sup>4</sup> Sin embargo, el objetivo de mediano plazo es especializarse en el desarrollo de software y exportar, principalmente, al mercado estadounidense.

En algunas entidades como Jalisco, el Consejo Estatal de Ciencia y Tecnología (COECYTJAL) fue el principal promotor para formar la Empresa Integradora con el objetivo de reducir los costos de certificación CMM nivel 3. En otras entidades como Nuevo León la Empresa Integradora se formó por iniciativa de las empresas de software y tiene la finalidad de consolidar a la industria local, aunque, como señala el gerente de proyectos de la empresa Goya, el modelo de la Integradora no es productivo sino de comercialización de los productos y servicios producidos al interior de la empresa integradora:

“Los miembros de la Empresa Integradora pueden desarrollar software para las empresas del grupo, es decir desarrollan para los integrados, pero no se puede hacer para otras empresas fuera del grupo.”<sup>5</sup>

---

<sup>4</sup> <http://www.contactopyme.gob.mx/integratoras>

<sup>5</sup> Entrevista con líder de proyecto, empresa Goya.

### 6.2.1 Tipos de agente en la construcción de la interfase simple

De los seis estudios de caso, cuatro han establecido un proyecto de desarrollo de software para procesos administrativos y organizacionales. Las empresas Constable, Goya, Renoir y Manet han tomado en consideración la información que en gran medida les proporciona el entorno codificado.<sup>6</sup> Esta información es acerca de las necesidades y requerimientos de los usuarios para el desarrollo de aplicaciones o programas de software, tales como programas de nómina, programas contables, sistemas de flujos de trabajo, administración de documentos, aplicaciones o herramientas para ERP, o bien servicios de consultoría y adaptación menor a programas ya existentes en el mercado.

Los usuarios se ubican en diferentes industrias. Por ejemplo, los clientes de Constable se ubican en el sector de manufactura y servicios, así como en empresas de la misma industria de software; los clientes de Goya son empresas medianas, básicamente de servicios; los de Renoir y Manet son pequeñas y medianas empresas de manufactura y servicios. A pesar de que los usuarios están ubicados en diversos sectores industriales, los productos de software para procesos administrativos y organizacionales no son muy diferentes, de hecho, son difundidos entre los gerentes de las empresas usuarias.

Todos los usuarios tienen necesidades y plantean requerimientos en un proyecto de desarrollo de software:

“Algunas necesidades son más sofisticadas que otras. Cuando la empresa usuaria tiene procesos productivos o administrativos simples, es relativamente fácil acceder a esas necesidades y requerimientos a través de medios como el Internet, o las bitácoras de la empresa que registra parte de las actividades de los proyectos. Si los procesos productivos y administrativos son simples, la información de los procesos se difunde entre las empresas del sector. De esta forma, las empresas de software pueden acceder a ese tipo de información sin necesidad de interactuar directamente con el usuario, la relación entre la empresa y el usuario se limita a contratos de compra-venta, de revisión de requerimientos e implantación, pero no se profundiza en la relación.”<sup>7</sup>

---

<sup>6</sup> Un *entorno codificado* es un entorno en el que se identifica información y conocimiento en forma de códigos que son transformados en lenguajes convenidos y preestablecidos para su comunicación.

<sup>7</sup> Entrevista con líder de proyecto, empresa Manet.

En este sentido, un entorno codificado proporciona una gran cantidad de la información necesaria para el desarrollo de aplicaciones o programas con base en las necesidades y requerimientos estandarizados de los usuarios.

Pero, como se mencionó en párrafos anteriores, hay otro agente a través del cual las empresas de software pueden relacionarse con el entorno. La figura de la Empresa Integradora es un medio de difusión y comercialización de los productos y servicios que un grupo de empresas ofrece tanto al mercado nacional como extranjero. Como lo indican algunos líderes de proyecto de la empresa Constable:

“La creación de una Empresa Integradora es parte de la estrategia colectiva para posicionarse en el mercado local y tratar de incorporarse a los mercados extranjeros... así como especializarse en el desarrollo de software, reducir costos y gestionar financiamiento.”

La Empresa Integradora implica formas de cooperación comercial, con una limitación productiva:

“Por definición una Empresa Integradora no puede vender más del 10% fuera del grupo integrado, pues es un servicio que se proporciona para los miembros del grupo. Cabe mencionar que las empresas que componen la Integradora no pueden competir entre sí, los productos y servicios deben ser complementarios entre las empresas de la Integradora.”<sup>8</sup>

La cooperación tiene sentido en la medida en que hay complementariedades para lograr un objetivo común: exportar los productos. De esta forma, se crea una interfase simple entre la empresa de software y la Empresa Integradora, ya que la relación se caracteriza por ser básicamente de compra-venta.

En resumen, el tipo de interfase entre las empresas Constable, Goya, Renoir y Manet con los agentes es clasificado como simple, ya que está basado en proyectos que implican una relación básicamente de compra-venta, en la cual los usuarios son pasivos.

---

<sup>8</sup> Entrevista con líder de proyecto, empresa Goya.

### **6.2.2 Tipos de interacción y estabilidad de la tecnología en la construcción de la interfase simple**

Como veremos en este punto, cuando el tipo de proyecto implica la instalación o implementación de aplicaciones que ya existen en el mercado, escalamiento o pequeñas modificaciones, el grado de interacción derivado de ese proyecto es limitado y el usuario con el cual la empresa de software interactúa es pasivo. En algunas ocasiones la implantación del software en la empresa usuaria requerirá de algún grado mínimo de interacción, pero:

“Hay proyectos que no lo ameritan, sobre todo cuando son sólo de adecuación o de escalamiento.”<sup>9</sup>

Las aplicaciones o programas desarrollados más comunes son acerca de procesos administrativos y organizacionales, como programas de nómina, contables, de gestión o administración de recursos (ERP, CRM)<sup>10</sup>, entre otros. Los usuarios que utilizan este tipo de programas se ubican en sectores tradicionales y no-tradicionales, aunque en cada proyecto hay un nivel de adaptabilidad a los programas ya existentes:

“Aunque los programas que desarrollan las empresas de software sean de uso general, siempre hay un rango de adaptabilidad a cada empresa. Cuando se trata de replicar algún programa no se requiere de un análisis y diseño a detalle y sólo se adapta a las necesidades de los usuarios. En algunos casos se requiere capacitar a los usuarios para usar el sistema.”<sup>11</sup>

Siguiendo el argumento de los párrafos anteriores, el tipo de información sobre los requerimientos y necesidades del usuario se puede identificar en el entorno codificado, en consecuencia, hay una pobre interacción entre la empresa desarrolladora y el usuario, y las formas de interacción se remiten a contratos de compra-venta, a la implantación o a la

---

<sup>9</sup> Entrevista con líder de proyecto, empresa Constable.

<sup>10</sup> Customer Relationship Management (CRM) también es parte de un Sistema Integrado de Gestión (ventas, servicio, mercadotecnia y centros de interacción –call centers-). Un ERP y un CRM son productos desarrollados para propósitos generales, pero cuando los sistemas se implantan o instalan es necesario hacer cambios o adaptarlos a las necesidades de la empresa. En México hay empresas que desarrollan ERP y CRM, como es el caso Goya. Para un detalle de este y otros sistemas, véase Gómez y Suárez (2004).

<sup>11</sup> Entrevista con líder de proyecto, empresa Manet.

adecuación de un programa ya existente en el mercado. En algunos casos la interacción implica algún tipo de capacitación elemental a los usuarios para usar el sistema de software:

“La interacción se remite al contrato de compra-venta y en algunas ocasiones a capacitar al personal (la capacitación dura poco tiempo, 6 a 10 horas en total). Los encuentros entre el proveedor y usuario, si los llega a haber, son informales. La formalidad se da en el contrato de compra-venta.”<sup>12</sup>

Otro aspecto importante es el tipo de tecnología de software utilizada en el desarrollo de aplicaciones. La tecnología de software es estable y difundida y muchas ocasiones la competitividad se centra en los precios y en el capital humano o en el uso de software de fuente abierta (SWFA) como es el caso de las empresas Renoir y Manet:

“Esto significa que la interfase técnica entre el programa y el usuario está bien realizada y no requiere mucho tiempo en aprender a usar el programa. Esta interfase técnica es simple, ya que la forma de operar el programa puede lograrse con sólo hacer uno o dos clic en la computadora.”<sup>13</sup>

La evidencia sugiere que, respecto a la categoría *tipo de interacción* los proyectos están basados en una compra-venta y a la implantación o adecuación de un programa ya existente en el mercado, por tanto las necesidades y requerimientos de los usuarios son simples y codificados. Respecto a la categoría *estabilidad de la tecnología*, ésta es ‘alta’ ya que las tecnologías de software son estables y difundidas entre las empresas de la industria.

### **6.2.3 Intensidad y formalidad en la interfase simple**

El tipo de proyectos descritos en párrafos anteriores y realizados por las empresas Constable, Goya, Renoir y Manet tienen una duración de 1 o 2 meses, con trabajo semanal de 20 horas en promedio por empleado.<sup>14</sup> Sin duda esto varía de acuerdo al tamaño de proyecto. Por lo cual, la evaluación de la categoría *intensidad de la interacción* es ‘baja’, pues son proyectos de corta duración.

---

<sup>12</sup> Entrevista con líder de proyecto, empresa Constable.

<sup>13</sup> Entrevista con líder de proyecto, empresa Manet.

<sup>14</sup> Entrevista con líderes y administradores de proyecto.



En algunos proyectos la relación entre la empresa de software y los usuarios son a través del correo electrónico, teléfono y, en menor medida, la comunicación cara a cara. El tipo de proyecto implica sistematizar el intercambio de información, el cual está relacionado al contrato de compra-venta. Por tanto, esta categoría se evalúa como *formal*.

En el cuadro 6.2 se resumen las características de una interfase simple. Por filas, se muestra las categorías utilizadas para caracterizar la construcción de la interfase, tal como se explicó en el primer apartado de este capítulo. La columna de la derecha muestra la característica asociada a cada categoría.

Como se observa en el cuadro, las empresas interactúan básicamente con usuarios pasivos, y en algunos casos con una empresa integradora. El grado de interacción es *bajo* ya que el tipo de proyecto se asocia a la compra-venta, a la implantación o instalación de programas/sistemas que ya existen en el mercado con pequeñas modificaciones, y en algunas ocasiones a la capacitación de corta duración para el uso adecuado de la aplicación/sistema de software instalado en la empresa usuaria; el proyecto implica rutinas de revisión de requerimientos y de implantación, lo cual constituye un espacio de interacción, pero de baja intensidad. En estas circunstancias, por lo regular las tecnologías de software, en particular las herramientas de software, son estables o estandarizadas y ampliamente difundidas para el desarrollo de ese tipo de proyectos.

Al mismo tiempo, se determina el tiempo de duración de la interacción entre el proveedor y el usuario. Como se ha mostrado en este apartado, la intensidad de la interacción es baja, medida relativamente en pocos meses. Finalmente, el tipo de proyecto asegura la formalidad de la interacción a través de la compra-venta, y como señala Andersen (1991), en una interfase simple donde predominan los productos estandarizados se minimizan los flujos de información.

**Cuadro 6.2**  
**Características de una interfase simple**

<b>Categoría de medición</b>	<b>Característica</b>
[1] Tipos de agentes con los cuales la empresa interactúa	Usuarios pasivos, empresa integradora
[2] Tipos y complejidad de la interacción*  <i>Información y conocimiento acerca de las necesidades y requerimientos de clientes-usuarios</i> <i>Acceso a la información acerca de las necesidades y requerimientos de los clientes-usuarios</i>	Relación compra venta, adaptabilidad de software ya instalado (el programa o aplicación no parte de cero) <i>Simples y codificados</i>  <i>A través de medios impresos, INTERNET</i>
[3] Estabilidad de la tecnología de software utilizada en un proyecto***	Alta, tecnologías de software difundidas entre las empresas de la industria
[4] Intensidad de la interacción** <i>Cantidad de información intercambiada***</i>	Baja, proyectos de corta duración (1-2 meses) <i>Baja</i>
[5] Formalidad versus informalidad de la interacción****	Formal, basado en contratos de compra-venta

\* (i) Relación compra-venta, (ii) escalamiento de software ya instalado, (iii) diseño y desarrollo de nuevo software, (iv) implantación de software, (v) mantenimiento, (vi) integración de sistemas, (vii) capacitación.

\*\* Semanas, meses, años.

\*\*\* Alta, media, baja.

\*\*\*\* Documentación y sistematización del intercambio de información y conocimiento utilizados en el proyecto.

Fuente: Elaboración propia con base en entrevistas.

Como se señaló en el capítulo 3 de esta tesis, Andersen (1991 y 1996) conceptualiza una interfase simple como una relación entre dos agentes a través de la cual fluyen diferentes tipo de información, con ella la información necesaria para cada uno de los agentes es delimitada. Bajo esta premisa, y con base en la evidencia presentada en este apartado, se puede argumentar que las especificaciones del proyecto de software pueden ser generalizadas y crear estructuras estables sobre las que se basa la relación de los agentes y en algún grado determina la mejora en el desempeño tecnológico de productores y usuarios.

En el caso de una interfase simple, y en un entorno en el cual la información y conocimiento se asocian a procesos productivos, organizacionales y tecnológicos estandarizados y estables ampliamente difundidos, las empresas que cuentan con ciertas habilidades técnicas pueden identificar y acceder a la información a través de medios impresos o electrónicos como el Internet, sin necesidad de la vinculación o la interacción.

Al respecto se pueden señalar dos puntos. El primero tiene que ver con el entorno en el cual se crean las interfases simples. Las empresas de software se ubican en un entorno dinámico, pero los productos que desarrollan son utilizados por empresas usuarias ubicadas en un

entorno estable. En este caso, las aplicaciones o programas de software son administrativos, de nómina, entre otros, los cuales no requieren de conocimiento especializado para poder desarrollarlos. En este sentido, y como se ha descrito en el capítulo 3, en un entorno de industrias maduras con tecnologías estandarizadas el conocimiento es difundido por medios codificados y se minimizan los flujos de información.

Segundo, en un entorno relativamente estable (en el cual se ubican los usuarios) las interfases difícilmente pueden cambiar porque están basadas en conocimiento rutinario y se han establecido códigos comunes de comunicación entre los agentes, las posibles modificaciones a la interfase estarán determinadas por la actividad reactiva y proactiva de los usuarios de tecnología (Andersen, 1991), pero, como hemos visto en este apartado, la actitud pasiva del usuario limita el dinamismo de la interacción con la empresa de software.

Finalmente, y como ha señalado Andersen (1991 y 1996), los cambios radicales en las estructuras pueden romper con las especificaciones básicas y crear nuevas condiciones para los productores y usuarios, dando indicios de otro tipo de interfase. Las empresas de software reconocen que en el entorno también fluye información y conocimiento acerca de tecnologías tanto nuevas como emergentes y que los usuarios tienen necesidades específicas, pero saben que no es fácil acceder a estas necesidades, sino sólo a través de la interacción proveedor-usuario o de la vinculación, o bien a través de las patentes, la capacitación externa, entre otros. Esto se mostrará en los siguientes dos apartados.

### **6.3 CREACIÓN DE INTERFASES SEMI-COMPLEJAS**

El segundo tipo de interfase que se describe y analiza es una interfase semi-compleja, la cual, como se mostrará en este apartado, tiene dos características fundamentales. La primera es que las empresas de software identifican y acceden a información y conocimiento acerca de procesos tecnológicos a través de los usuarios, ya que esa información y conocimiento no se encuentran completamente en el entorno codificado.

La segunda es que el tipo de usuario tiene procesos productivos, administrativos y organizacionales estandarizados, pero sus requerimientos son de mayor especificidad. Por tanto, la empresa de software requiere de una interacción más cercana con la empresa usuaria para conocer sus requerimientos y necesidades específicos para el desarrollo de una aplicación/sistema de software. En palabras del líder de proyecto de la empresa Velásquez:

“Las necesidades básicas o estandarizadas de los usuarios se pueden encontrar, por ejemplo, a través de Internet [entorno codificado], las medulares no, requieren de un alto grado de interacción entre la empresa de software y el usuario.”

El tipo de proyectos de software que permiten la construcción de interfases semi-complejas consisten en el diseño y desarrollo de programas o aplicaciones para procesos de manufactura, o sistemas de software que interactúan al mismo tiempo con los procesos productivos, de distribución, administración y ventas. Para identificar la información específica y desarrollar este tipo de proyectos, se requiere de la interacción con el usuario, ya que la información no se puede encontrar fácilmente en el entorno codificado o a través de medios como el Internet, medios impresos o seminarios, sino que está incorporada en la empresa usuaria, por tanto la interacción es crucial.

Estas son algunas razones por las cuales una interfase semi-compleja es diferente de la interfase simple. La evidencia que se presenta en este apartado es acerca de proyectos de ‘prueba de software’ y proyectos de ‘desarrollo de aplicaciones para empresas de manufactura’ (en particular para empresas de tenería).

### **6.3.1 Tipos de agente en la construcción de la interfase semi-compleja**

De los seis *casos*, dos han logrado crear interfases semi-complejas, Velásquez y Degas. Con base en entrevistas realizadas al gerente de proyecto de la empresa Velásquez, se puede constatar que las actividades asociadas al proyecto de prueba de software consistieron básicamente “en verificar la confiabilidad de las aplicaciones, su calidad y consistencia.” El proyecto más representativo se llevó a cabo con una empresa mediana de software ubicada en Jalisco. El gerente de proyectos menciona que la empresa Velásquez es “pionera en el servicio que ofrece a las empresas de la industria, o bien a empresas que lo

requieran...La realización de ‘pruebas de software’ como servicio que ofrece la empresa, la hace ser pionera en este nicho de mercado.”

Por su parte, la empresa Degas ha desarrollado proyectos de software para pequeñas y medianas empresas de tenería, siendo también pionera en su nicho de mercado. Las actividades centrales en este proyecto comprenden las 5 etapas del modelo de desarrollo descritas en el Anexo Uno. El proyecto representativo sobre el cual se recogió evidencia fue “la Implementación de un Sistema Informático (ISI)” en la empresa de tenería LEFARC ubicada en León, Guanajuato.

Ambas empresas de software, Velásquez y Degas, ofrecen a sus clientes productos y servicios especializados, lo cual implica la realización de actividades más complejas, como veremos a continuación.

Una parte de la información y conocimiento acerca de los procesos productivos, administrativos y organizacionales están codificados y se difunden entre las empresas del sector en donde se ubican los usuarios. Por ejemplo, para el caso de Velásquez:

“Los lenguajes de programación son conocidos por los ingenieros y por quienes desarrollan software, dependiendo del tipo de proyecto y de las habilidades de los desarrolladores utilizarán uno u otro tipo de lenguaje y diversas herramienta de software, y por tanto nosotros que hacemos ‘pruebas de software’ debemos saber eso, conocemos con qué lenguaje y herramientas se hicieron las aplicaciones, es decir, conocemos el programa como un todo, su arquitectura, su estructura general.”<sup>15</sup>

Los medios de difusión de ese tipo de procesos son a través de medios especializados como “seminarios, talleres, experiencias propias de los empleados”. El entorno proporciona una parte de la información necesaria para el desarrollo de aplicaciones o sistemas de software con base en las necesidades y requerimientos de los usuarios, pero hay otra parte de esa información necesaria para el desarrollo del proyecto que no está en el entorno codificado, está incorporada tácitamente en los empleados de la empresa usuaria y por ello es necesaria

---

<sup>15</sup> Entrevista con el director de proyectos de la empresa Velásquez.

la interacción, lo mismo para que una aplicación o sistema sea “eficiente, de calidad y funcional”, de otra forma, como señala el director de proyectos de la empresa Velásquez, “la aplicación será ineficiente y poco funcional para la empresa usuaria.” Siguiendo el mismo argumento de la cita anterior:

“...Si bien nosotros como empresa de prueba de software conocemos el programa como un todo, no conocemos los propósitos del programa o sistema para el que fue hecho, es decir, no conocemos las especificidades de los requerimientos de los usuarios que finalmente va a utilizar el programa. Esto es importante porque la confiabilidad, calidad y consistencia del programa depende de ello, y no sólo de las herramientas y lenguajes con que fue hecho. Entonces, para saber esas especificidades, es necesario interactuar o bien con la empresa de software que desarrolló el sistema de software, o con la empresa y usuarios finales que van a utilizar esa aplicación.”

Para la empresa Degas también es importante la interacción con el usuario ya que éste es la principal fuente de información y conocimiento acerca de los procesos productivos y administrativos, aunque el tipo de proyecto que realiza esta empresa es diferente a los proyectos de Velásquez.

De esta forma, se evalúa que el tipo de agentes con los que interactúan Velásquez y Degas para la realización de proyectos son propositivos.

### **6.3.2 Tipos de interacción y estabilidad de la tecnología en la construcción de la interfase semi-compleja**

Las aplicaciones o programas desarrollados más comunes en la industria mexicana de software son de nómina, contables, punto de venta, entre otros.<sup>16</sup> Los usuarios que utilizan este tipo de programas se ubican en industrias tradicionales y no-tradicionales, sobre todo donde los procesos administrativos y organizacionales han sido estandarizados. Cuando los programas de software que son de uso general y estandarizados no cubren por completo las necesidades de los usuarios, éstos requerirán de aplicaciones ajustadas a sus necesidades y,

---

<sup>16</sup> Entrevistas con líderes de proyecto de las empresas de software citadas en este capítulo.

por tanto, requerirá de una mayor interacción entre la empresa proveedora y la empresa usuaria.

En el caso de la empresa Velásquez, dado que su actividad más importante es ‘prueba de software’, la información que obtiene a través de la interacción no se encuentra por completo en el entorno codificado, y las experiencias logradas a través de la interacción permiten que la empresa desarrolladora tenga más elementos para hacer una buena aplicación o programa. En este sentido, la interacción entre la empresa de software y el usuario se definen mediante una ‘planeación estratégica’:

“Hay un grado de complejidad cuando dos empresas [proveedor y usuario] buscan soluciones conjuntas a un problema...pues generan conocimiento, y una gran parte del conocimiento lo proporcionan los clientes a través de la asesoría técnica. Esto requiere de planeación estratégica del proyecto, que permita definir tiempos, costos y alcances.”<sup>17</sup>

A pesar que esta empresa tiene un nicho de mercado relativamente novedoso y que hay un grado de interacción importante con los usuarios de la aplicación, la tecnología de software es estable y difundida, según el gerente de proyectos de la empresa Velásquez, “la competitividad se centra en el tipo de servicio especializado y novedoso que ofrece la empresa en la industria [mexicana] de software.”

Por otro lado, como se ha mencionado, la empresa Degas desarrolla aplicaciones para un sector de manufactura tradicional. Sus usuarios son empresas de tenería (curtido de cuero) con procesos de producción, distribución y organizacionales estandarizados. El líder de proyecto de la empresa Degas menciona que:

“La información y el conocimiento de los procesos de tenería son conocidos en la industria, de tal forma que la empresa de software puede conocer a priori los procesos básicos de la empresa usuaria [una empresa de tenería] y con base en ello desarrollar una aplicación específica para las empresas de curtido de cuero.”

Sin embargo, hay un grado de especificidad a cada empresa usuaria. Al respecto, el líder de proyecto argumentó:

---

<sup>17</sup> Entrevista con líder de proyecto, empresa Velásquez.

“Aunque el programa es hecho para empresas de tenería en general, cada empresa tiene necesidades y requerimientos muy particulares, por eso el software es hecho a la medida y no empaquetado. Por ejemplo, el personal de la empresa LEFARC [empresa usuaria] enfatizó que no quería software comercial sino uno ‘hecho a la medida’ que pudiera adaptarse a su equipo...El usuario requería sistematizar todas las áreas de la empresa, desde la compra de la piel, pasando por los procesos de curtido, hasta la venta del cuero tratando de satisfacer las necesidades de sus clientes.”

El proyecto se denominó Implantación de Sistema Informático (ISI), el cual permite a las empresas de tenería mejorar su productividad y eficiencia en las áreas de recepción, producción, venta y distribución, así como organizacional. Una parte de la información acerca de las necesidades y requerimientos básicos del usuario los conoce Degas, es decir, la obtiene en el entorno codificado ya que una buena parte de los procesos son estandarizados y difundidos, pero las necesidades particulares no están codificadas. Por otro lado, como lo ha mencionado el líder de proyectos de la empresa Degas:

“La implantación del programa al interior de una empresa usuaria siempre tendrá algún grado de adaptabilidad, es decir, al final la aplicación o el programa es hecho a la medida con base en requerimientos específicos de cada empresa usuaria.”

Estos proyectos requieren de un alto grado de adaptación y desarrollo hecho a la medida y no se basan sólo en la implantación del programa o sistema de software, como sucede en las interfases simples. El proyecto ISI implica pasar por todas las fases del modelo de desarrollo descritas en el Anexo Uno. El líder de proyecto describe este proceso:

“En la fase de análisis se diagnosticó cómo se encontraba la empresa, en qué condiciones estaba y qué necesitaba para mejorar. Se estudió si el personal que tenía LEFARC era suficiente y sólo necesitaba reubicarse, o por el contrario si tenía que contratar personal adicional para el sistema a implementar. También se estudió si necesitaba servidores o infraestructura de sistema (hardware)...se encontró que no tenían gerente de sistemas y un área de sistemas formal, por lo que se les sugirió sensibilizar a la gente y capacitarla para que estuviera preparada ante la instalación del sistema.”

Las particularidades de este proyecto las describe como sigue:



“La empresa usuaria requería conocer en principio cómo se hacen las compras de la materia prima, cómo se eligen los proveedores de los diferentes tipos de piel, ya que varían de acuerdo al tipo de vaca o novillo, el peso de éstos, tamaño, fibra, grosor, raza, etcétera...y con base en las necesidades de los clientes de LEFARC se requerían ciertos tipos de piel y por tanto diferentes proveedores...entonces, en función de las necesidades del cliente se ubica al proveedor adecuado a través del ISI.”

Pero esta particularidad de la empresa usuaria requería de una adecuada coordinación entre las áreas de la empresa. Tal como lo expresa el líder de proyecto:

“Después viene la parte de la venta, la cual implica tomar el pedido pero no confirmarlo. Esto, según la empresa LEFARC, es una diferencia respecto a su competencia, pues a través del ISI se valida que haya en existencia el tipo y la cantidad exacta de cuero. La persona ubicada en el área de ventas registra en el ISI la información de los clientes y se simulan los tipos de piel y cantidades existentes. Si no hay piel, el sistema estima para cuándo llegará y para cuándo se podrá surtir el pedido. Esto asegura la fecha de entrega (a diferencia de la competencia que no cumple con las fechas estimadas de entrega, siendo este un gran problema en el sector de tenerías. Con base en esto se toman decisiones reales de entrega del pedido, y con base en el pedido, el sistema da instrucciones al área de lavado, pintura, estilizado, etc., de la piel. El sistema permite hacer un balance de los pedidos confirmados por día o por semana: el ISI balancea las cargas para no saturar un área y así se evita que la empresa usuaria caiga en incumplimiento. El sistema también mide las capacidades reales de producción en cada área de la empresa ya que todas las áreas tienen restricciones de capacidad...Todo esto permite saber la eficiencia de las áreas, si están cumpliendo los tiempos de proceso, etcétera.”

Como se muestra en las entrevistas, el tipo de proyectos llevado a cabo por empresas como Velásquez y Degas profundizan en la interacción con el cliente, sobre todo en las etapas de análisis (diagnóstico) y diseño:

“Un buen diseño es reflejo de una buena y suficiente interacción con el usuario, implica que al momento en que se entrega el sistema funcionando no se piden hacer cambios radicales. Un buen diseño del proyecto es calibrado por la interacción entre el proveedor y usuario.”<sup>18</sup>

---

<sup>18</sup> Estas empresas se ha esforzado por profundizar en la etapa de análisis y diseño, ya que en los proyectos anteriores perdían tiempo, dinero y recursos. Degas, por ejemplo, requirió de un aprendizaje a lo largo de 7

El proveedor o la empresa de software conoce las herramientas, lenguajes, etcétera, que va a utilizar, pero el usuario es quien conoce su proceso productivo y organizacional y sus necesidades particulares que requiere de la aplicación o sistema de software, y sólo mediante una constante interacción al inicio del proyecto es posible empalmar la oferta y la demanda de habilidades.

Una vez que el sistema de software queda ajustado a las necesidades del cliente y es implantado al interior de la empresa usuaria, el siguiente paso es capacitar a los empleados de ésta. La capacitación está en función de las habilidades que los empleados tienen con el uso de sistemas informáticos. En particular para el proyecto implantado en la empresa LEFARC:

“Aunque los empleados de la empresa ya conocen los procesos de tenería, no conocen el uso del programa de software, por tanto, la capacitación y los servicios de consultoría dan valor agregado y son importantes después de la instalación.”<sup>19</sup>

En este sentido, hay un grado de complejidad en la información que las empresas de software deben identificar para poder dar soluciones adecuadas a las necesidades y requerimientos del usuario. Sin duda, como lo señala el líder de la empresa Velásquez:

“La complejidad de la interacción depende del tamaño del proyecto, es otras palabras, del tamaño del programa o aplicación de software...aunque también hay una complejidad estructural cuando la empresa usuaria es grande.”

Por lo regular, las tecnologías de software utilizadas en estos proyectos son estandarizadas, sin embargo, en el caso de la empresa Degas, las tecnologías de software (herramientas, sistemas operativos) utilizadas están basadas en software de fuente abierta y, como veremos en el capítulo 8, son tecnologías que constantemente están en evolución y escalamiento.

De esta forma, la evidencia sugiere que, respecto a la categoría *tipo de interacción*, los proyectos están basados en el análisis, diseño, desarrollo y pruebas de software, así como

---

años en sistematizar el proceso interno de análisis y diseño, ahora el tiempo destinado es menor, impactando en los costos de operación y reduciendo el tiempo de desarrollo de software.

<sup>19</sup> Entrevista con líder de proyecto de la empresa Degas.

en la capacitación, en donde una gran parte de las necesidades y requerimientos de los usuarios son tácitos. Respecto a la categoría *estabilidad de la tecnología*, ésta es ‘media’ ya que tienen un bajo nivel de estabilidad pero son difundidas entre las empresas de la industria.

### **6.3.3 Intensidad y formalidad en la interfase semi-compleja**

Las empresas Velásquez y Degas han realizado proyectos que implican un grado de interacción más intensa que lo que requieren proyectos como los descritos en una interfase simple. Los proyectos de estas empresas han requerido de actividades que constituyen el espacio de la interacción: análisis, diseño (diagnóstico), desarrollo (programación), revisión y capacitación; las cuales nutren a la adaptación, escalamiento e implementación del programa/sistema.

La intensidad de la interacción también es mayor en una interfase semi-compleja. Por ejemplo, como señala el líder de proyecto de Degas, para el proyecto ISI:

“El número de reuniones formales a la semana fue, en promedio, de 2 a 3, es intensa al principio, en especial en las etapas de análisis y diseño, y disminuye la intensidad de la interacción en las etapas de desarrollo (programación), revisión e implantación, en éstas dos últimas las relaciones llegar a ser poco frecuentes y la mayoría de ellas informales. El número de horas de un proyecto para una empresa mediana como LEFARC es en promedio de 1000 horas, con un tiempo promedio de realización de 8 a 16 meses. Una vez instalado el programa o el sistema en la empresa usuaria, una buena parte de esas horas se ocupa en capacitar al personal que lo va a usar. Al principio hay una mayor formalidad de la interacción ya que se establece a partir de contratos y reuniones formales, y conforme el proyecto avanza *se establecen vínculos de confianza y la interacción se vuelve informal.*”

Por lo cual, la evaluación de la categoría *intensidad de la interacción* es ‘alta’, ya que los proyectos son de larga duración. Si bien la categoría de *formalidad* está asociada a la sistematización del intercambio de información (el cual está relacionado al contrato de compra-venta), también hay un grado de *informalidad* ya que en las últimas etapas del proyecto se establecen vínculos de confianza.

El cuadro 6.3 lista las características descritas de una interfase semi-compleja. Las empresas interactúan con usuarios pro-activos y propositivos. El grado de interacción es alto ya que, con base en los requerimientos y necesidades del usuario, el tipo de proyecto está asociado al análisis y diseño, a la implementación, prueba de software e implantación del programa o aplicación de software. Adicionalmente, la capacitación que la empresa proveedora da a los usuarios es fundamental para el uso adecuado del sistema de software instalado en la empresa.

Las tecnologías de software utilizadas en estos proyectos son estandarizadas, aunque en el caso de la empresa Degas las tecnologías de software utilizadas están basadas en software de fuente abierta, constantemente en evolución.

El tipo de proyecto asegura la formalidad de la interacción a través de la compra-venta, como se ha mostrado en el apartado anterior, pero a medida que el proyecto avanza se crean vínculos de confianza y la interacción se vuelve informal. Andersen (1991) señaló que en una interfase donde predominan los productos estandarizados se minimizan los flujos de información pero, como se ha mostrado en este apartado, en una interfase semi-compleja los productos no son estandarizados ya que son hechos a la medida. Por tanto, hay intercambio de grandes montos de información y conocimiento en la interfase.

A diferencia de Andersen (1991 y 1996) que establece dos tipos de interfase, simple y compleja, en esta tesis se plantea una interfase intermedia denominada interfase semi-compleja. La diferencia con respecto a la interfase simple es que las especificaciones del proyecto de software no pueden ser generalizadas para todas las empresas, incluso para aquellas de una misma industria, como en el caso de la industria de tenería, de tal forma que las estructuras tecnológicas sobre las que se basa la interacción entre los agentes, y sobre las cuales se realizan los proyectos, no son completamente estables, ya que son productos de software hechos a la medida de cada empresa usuaria.

En este sentido, y como señala Andersen (1991 y 1996), los usuarios “no aceptan las especificaciones del producto ya hecho y probado por otras empresas y toman un rol pro-

activo en la configuración de nuevos productos”, en este sentido el usuario se vuelve proactivo y “es la principal fuente en el diseño de interfases [y de nuevos productos de software].”

**Cuadro 6.3**  
**Características de una interfase semi-compleja**

<b>Categoría de medición</b>	<b>Característica</b>
[1] Tipos de agentes con los cuales la empresa interactúa	Usuarios pro-activos y propositivos
[2] Tipos y complejidad de la interacción*  <i>Información y conocimiento acerca de las necesidades y requerimientos de clientes-usuarios</i>  <i>Acceso a la información acerca de las necesidades y requerimientos de los clientes-usuarios</i>	Análisis, diseño, desarrollo y pruebas de software; capacitación <i>Tácitos y codificados</i>  <i>A través de INTERNET, correo electrónico, interacción cara a cara con el usuario</i>
[3] Estabilidad de la tecnología de software utilizada en un proyecto***	Media, tecnologías de software con bajo nivel de estabilidad pero difundidas entre las empresas de la industria
[4] Intensidad de la interacción** <i>Cantidad de información intercambiada***</i>	Alta, proyectos de larga duración (8-16 meses) <i>Alta</i>
[5] Formalidad versus informalidad de la interacción****	Formal, basado en contratos de compra-venta y capacitación. Informal en las últimas etapas del proyecto

\* (i) Relación compra-venta, (ii) escalamiento de software ya instalado, (iii) diseño y desarrollo de nuevo software, (iv) implantación de software, (v) mantenimiento, (vi) integración de sistemas, (vii) capacitación.

\*\* Semanas, meses, años.

\*\*\* Alta, media, baja.

\*\*\*\* Documentación y sistematización del intercambio de información y conocimiento utilizados en el proyecto.

Fuente: Elaboración propia con base en entrevistas.

Adicionalmente, las empresas de software se ubican en un entorno tecnológico dinámico, pero algunas empresas usuarias, como el caso de LEFARC, se ubican en un entorno tecnológico estable, en el cual las interfases difícilmente pueden cambiar porque están basadas en conocimiento rutinario y se han establecido códigos comunes de comunicación entre los agentes y “las modificaciones a la interfase estarán determinadas por la actividad reactiva y proactiva de los usuarios de tecnología” (Andersen, 1991). Lo relevante en una interfase semi-compleja es que la empresa usuaria que se ubica en un entorno tecnológico estable está influyendo en buena medida en este tipo de interfase, ya que sus necesidades y requerimientos obedecen a una transformación y modernización de sus procesos, no sólo administrativos (bastaría usar programas de nómina, contables, de gestión de recursos humanos, entre otros) sino de toda la cadena de valor de la empresa, desde la compra de

materia prima hasta la venta final, considerando las particularidades de sus proveedores y las necesidades particulares de sus clientes, como es el caso de LEFARC.

Andersen (1991 y 1996) argumenta que los cambios en las estructuras pueden romper con las especificaciones básicas y crear nuevas condiciones para los productores y usuarios, originando otro tipo de interfase. Esto se ha mostrado en el presente apartado, a partir de la conducta proactiva del usuario. Pero, al respecto surgen preguntas por contestar, por ejemplo, ¿Cuáles son los factores que están incidiendo en las empresas usuarias para modernizar sus procesos? Sobre todo de aquellas que se ubican en entornos tecnológicos estables. ¿Obedecen a una reacción ante la competencia, o a la demanda de los mercados internacionales? ¿O son una reacción natural ante las ventanas de oportunidad que ofrecen las nuevas tecnologías de información (TI), en particular el software? Aunque estas preguntas no son el foco de esta investigación, es importante plantearlas ya que en función de las respuestas pueden abrirse nuevas ventanas de oportunidad para las empresas mexicanas de software.

#### **6.4 CREACIÓN DE INTERFASES COMPLEJAS**

La categoría más importante en la construcción de una interfase compleja es, además del tipo de proyecto, el tipo de interacción con los diferentes agentes del entorno, sobre todo con usuarios activos, usuarios ‘conscientes’ y universidades.

En este apartado se mostrará que la interacción con cada uno de estos agentes requiere de intercambio de información y conocimiento específicos para realizar proyectos de software, información que no se puede identificar fácilmente en el entorno codificado, sino a través de los agentes. Los proyectos comprenden análisis, diseño y desarrollo de nuevas aplicaciones, integración de sistemas, así como la creación de fábricas de software y la formación de recursos humanos especializados. Esto aumenta el grado y la complejidad en la interacción entre la empresa de software y los diferentes agentes del entorno.

#### 6.4.1 Tipos de agente en la construcción de la interfase compleja

En este punto se mostrará que en una interfase compleja los agentes, con los cuales la empresa de software interactúa, son distintos a los de las interfases anteriores. Por ejemplo, uno de los proyectos más representativos para la empresa Renoir fue el desarrollo de un programa para detectar problemas de sanidad en bosques, realizado para la Comisión Nacional Forestal de Jalisco.

“El objetivo del proyecto es que el personal de la dependencia pueda entrar al campo y detectar problemas relacionados con plagas de una manera sistemática y adecuada, sin necesidad de que asista un especialista...De forma que el diagnóstico se pueda hacer lo más rápido y adecuadamente posible.”<sup>20</sup>

Este usuario tenía necesidades y requerimientos específicos que fueron cubiertos por Renoir. Como señala el líder de proyecto de esta empresa, durante el proyecto el usuario tomó un rol activo en el diseño del sistema de software:

“En la interacción los usuarios participan activamente en la configuración del producto o servicio y sus observaciones, criterios y recomendaciones son los que dan origen a las innovaciones: en última instancia, lo que influye es la presión del cliente, lo cual nos orilla a innovar o mejorar ciertas aplicaciones.”

La naturaleza del proyecto fue compleja, ya que ninguno de los 9 ingenieros que trabajan en Renoir son especialistas en plagas o en áreas relacionadas con la conservación de ecosistemas. Sin embargo, la empresa logró desarrollar el programa tal como la dependencia estatal lo solicitó, pero, ¿cuáles son los factores decisivos para desarrollar exitosamente un proyecto de esta naturaleza? En palabras del líder de proyecto de la empresa:

“Las capacidades de la empresa residen en que las personas que trabajan en Renoir tienen habilidades para sistematizar en un *lenguaje común* lo que los usuarios necesitan. En relación al proyecto para detectar problemas de sanidad, se logró traducir las necesidades y requerimientos de los usuarios en un sistema con las características deseadas. Pero también fue de gran utilidad que los usuarios tuvieran conocimiento acerca de las tecnologías de software, de forma que hubo una retroalimentación acerca de qué tecnología de software

---

<sup>20</sup> Entrevista con líder de proyecto, empresa Renoir.

sería la más idónea para el sistema. Por otro lado, fue importante traducir lo que realmente necesitaba el usuario, además de la funcionalidad la simplicidad, ya que cualquier persona podría utilizar el sistema sin necesidad de ser ingeniero agrónomo o especialista en el área.”

En este sentido, el usuario tienen conocimiento general acerca de las herramientas que usa la empresa de software, hecho que le permite interactuar dinámicamente con esta empresa y mejorar la aplicación o programa, es decir, el usuario es activo pues sugiere la forma en que debería ser desarrollado el sistema. Como señala el líder de proyecto: “en la industria de software los usuarios son los que empujan a la innovación.”

Un segundo actor es el que se denomina ‘usuario consciente’, el cual además de conocer las herramientas de software tiene habilidades en el uso de ellas, es decir, es un usuario que puede realizar actividades de programación de software. Como veremos en el siguiente punto, este tipo de usuario se ubica en las comunidades de desarrolladores de software de fuente abierta (SWFA).<sup>21</sup> La función de un usuario consciente es diferente al usuario que participa en los proyectos descritos anteriormente:

“El propósito del usuario consciente es mejorar una aplicación hasta cierto grado y en determinado momento hacer uso de ella, aunque no necesariamente usa comercialmente la aplicación o programa que el grupo de desarrolladores o la ‘comunidad’ desarrolla. Esto sucede particularmente en el contexto de software de fuente abierta, ya que una gran cantidad de las aplicaciones son libres y pueden ser mejoradas a través de las ‘comunidades’, de las cuales los usuarios conscientes son parte.”<sup>22</sup>

Como mencionan algunos líderes de proyecto de empresas basadas en SWFA, este tipo de usuario es diferente de aquel que se encuentra en un contexto de software propietario (SWP). Hay varias razones, como se muestra a continuación:

“[Primero,] porque es un usuario consciente de las necesidades y requerimientos de sus proyectos, es decir, sabe lo que necesita y tiene una idea general y particular de cómo

<sup>21</sup> En un contexto de software propietario también existe un tipo de usuario consciente, sobre todo en un esquema de subcontratación entre las mismas empresas de software. Por ejemplo, sea un usuario A que contrata a una empresa de software B para realizar un proyecto, y B a su vez subcontrata a otra empresa de software (C) para realizar una parte del proyecto que por razones de tiempo, costo, etcétera, no puede realizar. En este esquema la empresa B se vuelve un usuario consciente para la empresa C pero, generalmente, las empresas de tipo B subcontratan procesos que no son medulares sino de bajo contenido tecnológico.

<sup>22</sup> Entrevista con desarrollador y líder de proyecto, empresa Manet.



hacerlo. [Segundo,] tiene un cierto grado de experiencia en el manejo de algún tipo de herramienta o tecnología, lo cual lo hace más interactivo con los desarrolladores o comunidades. [Tercero,] el medio más utilizado por el cual los usuarios conscientes interactúan es a través de Internet, específicamente por medio del correo electrónico, aunque también son importantes los foros de discusión virtual, entre otros.”<sup>23</sup>

Esto no sucede en el contexto de software propietario, donde los usuarios no mantienen relación entre sí, son aislados y actúan independientemente. Esta forma de interacción hace que el usuario consciente sea poseedor de una parte importante de las experiencias que se acumulan entre las empresas de software y en las comunidades. Empresas como Manet han elaborado proyectos donde el usuario es consciente, existe una retroalimentación que les permite aprender nuevas herramientas de software y crear nuevos códigos o programas.

En ese sentido, los usuarios activos y conscientes, así como las comunidades de desarrolladores de SWFA, son una fuente importante de información y conocimiento para el desarrollo de sistemas de software, este tipo de usuarios acumulan experiencias de los proyectos en que participan y que comparten entre ellos y con los desarrolladores.

Un tercer actor con el cual las empresas de software interactúan es la universidad. En esta interfase es de particular importancia el proyecto de vinculación universidad-empresa que se ha establecido entre la empresa Goya y UNIVER.<sup>24</sup> Como se mostrará en el siguiente punto, la universidad es un agente fundamental en la creación y formación de recursos humanos especializados que demanda la empresa y un segmento de la industria mexicana de software.

De esta forma, se evalúa que el tipo de agentes con los que interactúan Renoir, Manet y Goya en la realización de proyectos son activos, ya que participan en el diseño y desarrollo del proyecto, y en algunos casos son conscientes ya que además tienen habilidades en el manejo de herramientas de software.

---

<sup>23</sup> *Ídem.*

<sup>24</sup> Universidad privada ubicada en el Edo. de México.

#### **6.4.2 Tipos de interacción y estabilidad de la tecnología en la construcción de la interfase compleja**

Como se menciona en párrafos anteriores, en la configuración de una interfase compleja el tipo de usuario con el cual las empresas de software interactúan es activo, ello implica que juega un rol dinámico en la creación de nuevos productos y servicios. En la interfase compleja hay una mayor interacción entre la empresa de software y los usuarios en las etapas de análisis y diseño, y también posteriormente de la instalación del sistema de software a través de la capacitación a usuarios.

Para el caso de Renoir, el usuario activo fue la Comisión Nacional Forestal de Jalisco, a la cual se le desarrolló un sistema para sanidad. Como se mencionó en el punto anterior, el propósito del proyecto era determinar rápida y adecuadamente, a partir de un sistema de software, la magnitud del problema asociado a las plagas, sin necesidad de especialistas.

Este proyecto se inserta en un área tecnológica compleja, como lo indica su líder:

“Tratar de identificar todas las variables que influyen y se toman en cuenta para incorporarlas en el sistema y que éste haga un diagnóstico sin necesidad del especialista es una tarea por sí misma compleja. Pero hubo intercambio de ideas e información con los ingenieros de la Comisión para identificar adecuadamente esas variables. Esto es fundamental porque si en la parte de diseño no se especifican adecuadamente las variables centrales, por muy buen desarrollo que se haga, el sistema no será útil para los propósitos del usuario.”

Los requerimientos de la Comisión (usuario) no están estandarizados, de tal forma que hay un conocimiento incorporado en los ingenieros de la Comisión que se ha trasladado a especificaciones técnicas para la elaboración del sistema, de esta forma, el proyecto requirió desde el análisis y diseño, hasta la implementación del programa en computadoras personales, además de la capacitación en su operación.

#### ***Comunidades de desarrolladores de software de fuente abierta como interfase social***

Las comunidades de desarrolladores de SWFA están tomando un rol cada vez más importante en una industria donde el entorno es rico en información y conocimiento.

“La actividad principal de una comunidad es desarrollar una herramienta y escalarla permanentemente; sólo en algunos casos esas herramientas tendrán alguna aplicación comercial.”<sup>25</sup>

Esto no significa que las herramientas tengan nula utilidad comercial, por el contrario, son herramientas que pueden tener infinidad de aplicaciones en el mercado, pero el objetivo de una gran parte de las comunidades no es crear aplicaciones con fines comerciales. La importancia de las comunidades es que constituyen el mecanismo por el cual se innovan nuevas herramientas. Con base en entrevista realizada al líder de proyectos de la empresa Manet:

“Una comunidad de desarrolladores de software de fuente abierta puede tener 3 o más desarrolladores y 100 usuarios conscientes, todos relacionados en una misma lista de correo, a través del cual intercambian opiniones, criterios de desarrollo, módulo de un sistema, herramientas, entre otros. Los 3 desarrolladores no necesariamente son quienes generan el conocimiento (éstos conocen cómo funciona por dentro un sistema) ya que gran parte del conocimiento reside en los 100 usuarios. Una comunidad de desarrolladores sin usuarios ‘conscientes’ no tendría la posibilidad de dirigir adecuadamente su desarrollo.”

En ese sentido, el conocimiento está centralizado en la comunidad que la componen desarrolladores y usuarios, y no en una persona. Estas comunidades son una interfase que permite a los desarrolladores transformar un mundo rico en información y conocimiento que se encuentra ‘incorporado’ en los usuarios conscientes en nuevas herramientas o aplicaciones.

La empresa Manet ha desarrollado diversos proyectos basados en tecnologías de SWFA. Algunos han sido desarrollados con herramientas de SWFA que no han sido estandarizadas en el mercado, son poco utilizadas por los desarrolladores mexicanos y extranjeros. Uno de los líderes de esta empresa es parte de una comunidad de desarrolladores de software libre, aunque en ésta se desarrollan herramientas y aplicaciones que no siempre tienen alguna estrategia de comercialización. Sin embargo, ser parte de alguna comunidad obedece a dos razones fundamentales:

---

<sup>25</sup> Entrevista con líder de proyecto, empresa Manet.

“Por un lado, la actualización es permanente ya que existen miles de desarrolladores en todo el mundo que escalan herramientas y programas y uno puede acceder a ese conocimiento nuevo siendo miembro de la comunidad, por otro lado, un módulo [una parte de una herramienta o aplicación] de uso general le puede servir a más de una persona, con funcionalidades extras que le puede hacer otra persona...lo que incide directamente sobre la reducción de costos de programación.”<sup>26</sup>

Esto implica que la comunidad otorga beneficios indirectos a una empresa que basa su estrategia tecnológica en SWFA.

### ***Vinculación universidad-empresa como interfase social***

Goya es la única empresa mediana de este estudio, y según el gerente de proyectos:

“La empresa tuvo como uno de los proyectos más representativos la creación de una fábrica de software a través de la vinculación universidad-empresa. El proyecto tuvo como objetivo general desarrollar programadores de bajo costo pero especializados en herramientas que la empresa y una parte del sector necesitan. Goya se especializa en tecnología JAVA y SQL y necesita personal especializado en éstas y otras tecnologías de software.”

Asimismo, continúa diciendo, el proyecto de vinculación responde a uno de los problemas que en la industria de software nacional prevalecen: “la falta de programadores especializados”, problemática que es compartida por otros líderes de proyecto entrevistados. La preocupación de este gerente es que:

“Los ingenieros y licenciados que egresan de las carreras de informática y afines necesitan capacitación para el uso de herramientas que utiliza Goya, pero esa capacitación es costosa para la empresa. [Por un lado,] cuando la empresa contrata profesionales en informática los capacita y entrena para que obtengan las habilidades en las tecnologías de software. Pero en la industria de software la tasa de rotación de personal es alta y es común que después de pocos meses la persona a la que se le invirtió tiempo y recursos en la generación de las habilidades específicas se va a otra empresa: se lleva conocimiento que a la otra empresa no le costo desarrollar. [Por otro lado,] una vez que la persona que ha recibido capacitación se va de la empresa, es necesario contratar nuevo personal y capacitarlo nuevamente.”

---

<sup>26</sup> *Ídem.*

Esto implica un costo económico para la empresa y tiempo para la creación de las habilidades específicas requeridas. El líder del proyecto de la fábrica de software menciona:

“El problema de fondo es que los licenciados o ingenieros en las áreas de informática se gradúan con conocimientos muy generales, pero no cuentan con las habilidades específicas para desempeñarse en el mercado de trabajo.”

En este sentido, y como lo señala el gerente de proyectos de la empresa Goya, el objetivo del proyecto de vinculación fue:

“...Tener un ‘semillero de estudiantes’ con habilidades específicas en el manejo de plataformas tales como JAVA, SQL, DBIL, entre otras. Lo que Goya necesita son ‘obreros de programación’ que puedan permanecer tiempo suficiente en la empresa...por eso el proyecto de vinculación.”

A través del proyecto de vinculación, el propósito de Goya fue formar personas con habilidades al interior de la universidad, y una vez que los estudiantes egresen contarán con las habilidades específicas que el mercado, y en particular Goya, demanda:

“Este proyecto de vinculación es una innovación en el sector de software. El modelo actual en el caso general de la industria está cerrado a que la empresa contrata y capacita al personal...y no se van más ‘atrás’, a la formación de recursos humanos al interior de la universidad.”

En términos de costo del proyecto, este se reduce debido a dos razones. Primero, a medida que el proyecto de vinculación avanzaba y generaba certidumbre, se asociaron otras empresas como IBM (quien donó el software utilizado en el proyecto) e INEVO (quien dio las herramientas y el know-how acumulado sin costo alguno). Segundo, se aprovechan instalaciones adecuadas de la universidad, en particular las aulas. De esta forma Goya y las otras dos empresas asociadas dieron forma a la fábrica de software.

Bajo la misma línea, la formación de personal con habilidades específicas no se haría al interior de la empresa sino en la universidad con la que se realizó el proyecto de vinculación. Esto requirió de una intensa interacción formal entre Goya y UNIVER:

“...Sobre todo en la adecuación de los planes de estudio, ya que debían estar basados en las necesidades reales de la empresa. El proyecto requería una adecuada selección de las personas adecuadas para entrenar a los estudiantes, las cuales fueron proporcionados por la empresa, y también hubo una selección de los estudiantes adecuados para el manejo de las herramientas específicas.”

Los estudiantes egresados bajo este esquema estarán capacitados en herramientas específicas, se incorporarán al mercado de trabajo con habilidades ya formadas, y al menos en esas tecnologías que Goya y otras empresas requieren no necesitarán capacitación.

En la figura 6.1 se muestra el modelo agregado de la cadena de valor del insumo principal, el conocimiento. La parte superior del gráfico constituye el modelo tradicional representado por U1. En éste se forma a los estudiantes en una universidad a lo largo de 4 años que en promedio dura una carrera de ingeniería o licenciatura relacionada con la informática. Al finalizar los estudios, la persona que elige trabajar en alguna empresa de software requiere de algún tipo de capacitación o entrenamiento, los cuales pueden ser otorgados por la empresa que los contrata. Esto conlleva costos y tiempo para formar las habilidades necesarias en las personas de nueva contratación y recién egresadas de las universidades (del tipo U1).

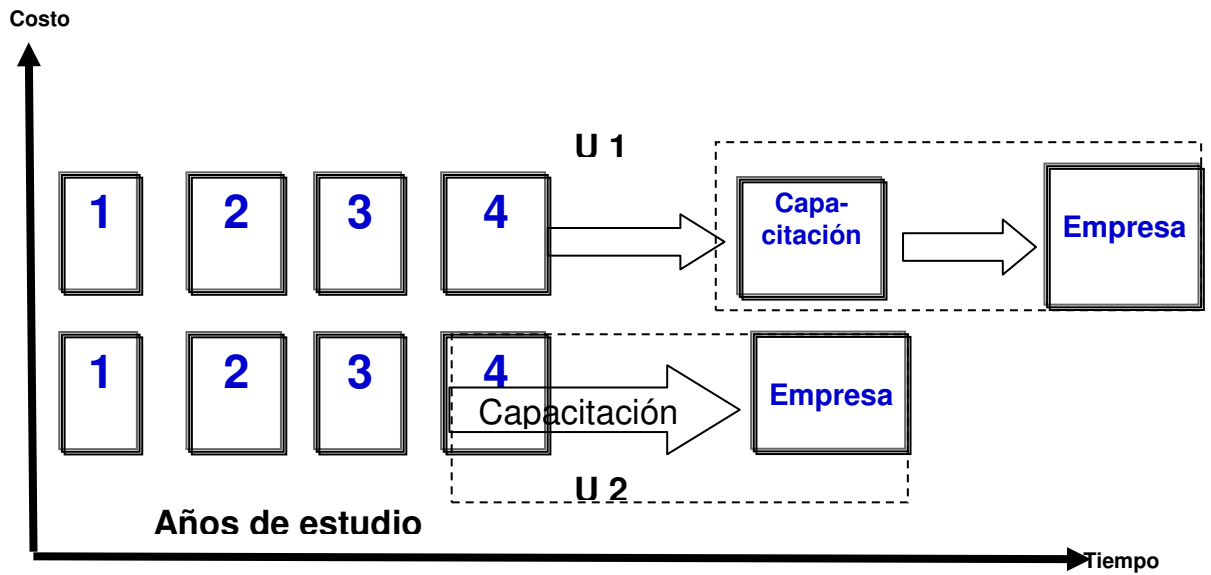
Si la empresa quisiera contratar a alguien con experiencia tendría que pagar mayores salarios pero:

“La empresa no necesita líderes de proyecto o ingenieros de alta especialización, lo que requiere son programadores de calidad y especializados en las tecnologías de software de la empresa, de forma que es relativamente menos costoso contratar técnicos superiores (programadores) que ingenieros de alta especialidad (se considera que estos ingenieros son mano de obra de altos salarios porque además de desarrollar software han sido líderes de proyecto, tiene habilidades administrativas y gerenciales, de análisis y consultoría, entre otras)...además, llega un momento en que a estos ingenieros y licenciados ya no les interesa el desarrollo o programación de software, y por eso buscan mejores opciones de salario realizando otras actividades diferentes a la programación.”<sup>27</sup>

---

<sup>27</sup> Entrevista con el gerente de proyecto, empresa Goya.

**Figura 6.1**  
**Dos modelos diferentes de generación de recursos humanos para la industria de software**



U1: Universidad tipo 1. U2: Universidad tipo 2.  
Fuente: Elaboración propia con base en entrevistas.

Por el contrario, el segundo modelo representado por U2 es generado por el proyecto de vinculación entre Goya y UNIVER. En él, el estudiante cursa los 4 años escolares, pero antes de terminar el plan de estudios formal, recibe la capacitación en las tecnologías que Goya y otras empresas de la industria requieren, de forma que las habilidades específicas del estudiante se forman paralelamente con el conocimiento general. Cuando el estudiante se incorpora al mercado de trabajo ya no será necesario capacitarlo o entrenarlo. Así, en consecuencia, se reduce los costos de capacitación y entrenamiento y el tiempo para la formación de habilidades específicas.

En síntesis, para la empresa Renoir la complejidad radicó en trasladar las necesidades de la Comisión en una aplicación, ya que ningún empleado de la empresa tenía conocimiento en plagas, incendios, deforestación, etcétera, y gran parte del proyecto consistía en sistematizar estos conocimientos en un sistema de software. Para el caso de Manet la complejidad radica en dónde, cómo y con quiénes se desarrollan herramientas y aplicaciones: en las comunidades de desarrolladores de SWFA, en las cuales interactúan con usuarios conscientes. Para Goya la complejidad se determina por el tipo de proyecto

realizado: crear una masa crítica especializada con habilidades en tecnologías específicas de software.

De esta forma, respecto a la categoría *tipo de interacción* los proyectos están basados en el diseño y desarrollo de nuevas aplicaciones de software, así como en la creación de recursos humanos especializados, las necesidades y requerimientos de los usuarios están incorporados en los agentes y no han sido del todo sistematizados o codificados. Respecto a la categoría *estabilidad de la tecnología*, ésta es ‘baja’ ya que las tecnologías de software tienen un bajo nivel de estabilidad y son poco difundidas en la industria.

### **6.4.3 Intensidad y formalidad en la interfase compleja**

Para el caso del sistema de software de sanidad desarrollado por Renoir, el proyecto duró “5 meses”.

“En el primer mes se definió el proyecto de software a detalle, los requerimientos y la forma en que se desarrollaría el sistema. En las primeras etapas del proyecto (análisis y diseño) los líderes de proyecto de la empresa proveedora y de la Comisión Nacional Forestal de Jalisco se reunían 2 veces por semana; en la etapa intermedia (programación) la interacción no fue muy constante; y al final del proyecto se realizaron 2 o 3 reuniones por mes. Las reuniones por lo general tenían un programa formal de actividades.”

Para el caso de Manet la interacción con los usuarios es intensa al inicio del proyecto, pero disminuye a medida que el proyecto madura. La duración de un proyecto para Manet es de entre “3 y 4 meses, donde la mayoría de las veces la interacción es informal”, no hay procesos de documentación o sistematización de las reuniones de trabajo.

Por otro lado, cuando Manet interactúa con usuarios conscientes en una comunidad de SWFA la intensidad de la interacción es de gran dinamismo:

“La forma de relación es por lo regular informal y los proyectos que desarrollan son de larga duración; un proyecto puede durar hasta dos años y seguir en escalamiento. [Pero,]



dependiendo de la complejidad de la aplicación, el proyecto durará más o menos, ya que algunas aplicaciones son menos complejas y duran menos tiempo, seis meses o un año.”<sup>28</sup>

Para los proyectos desarrollados en las comunidades de SWFA, el correo electrónico es el medio de comunicación idóneo.

En el proyecto de “vinculación universidad-empresa la intensidad de la relación fue por dos años aproximadamente”. El líder de este proyecto acudía al principio del proyecto “al menos 2 veces por mes a UNIVER, la comunicación por teléfono y correo electrónico fue intensa”. Además de él, Goya asignó a UNIVER un ingeniero de software de tiempo completo.

A pesar de la importancia del proyecto para la generación de habilidades específicas, la documentación del proyecto de vinculación fue poca, salvo las experiencias obtenidas.

“Uno de los problemas asociados a la falta de documentación del proyecto es debido a la falta de recursos económicos, dada la magnitud del proyecto de vinculación se requirió de un staff que no se pudo desarrollar en UNIVER por falta de esos recursos.”

Por lo cual, la evaluación de la categoría *intensidad de la interacción* es ‘alta’, ya que son proyectos de larga duración. Además, si bien la categoría de *formalidad* está asociada a la sistematización del intercambio de información, también hay un grado de *informalidad* ya que en las últimas etapas del proyecto se establecen vínculos de confianza.

En el cuadro 6.4 se resumen las características de la interfase compleja. En ella, las empresas de software interactúan con usuarios que tienden a ser activos y conscientes, ya que no toman como tal las especificaciones establecidas de los productos y servicios que se generan en el mercado, pues requerirán algún tipo de modificación y adaptación para cubrir sus necesidades. Ante esto, los usuarios se involucran dinámicamente en la interacción con el proveedor, aportando ideas, criterios y, en algunos casos, propuestas en el uso de tecnologías de software. La participación activa del usuario y su conocimiento en ciertas

---

<sup>28</sup> Entrevistas con líder de proyecto, empresa Manet.

herramientas y aplicaciones de software lo hacen activo en la interacción, y ésta es una de las diferencias fundamentales entre una interfase compleja y una semi-compleja o simple.

A diferencia del concepto de interfase compleja acuñado por Andersen (1991), donde los proveedores interactúan con los usuarios para conocer los requerimientos de éstos y diseñar nuevos productos, pero donde los usuarios no se involucran activamente en este proceso, en esta tesis se muestra que en la construcción de una interfase compleja los agentes del entorno demandan productos de acuerdo a sus necesidades específicas y éstos se involucran activamente en el diseño y desarrollo de aplicaciones de software. Esta es una de las razones por las cuales el proyecto es hecho a la medida de los usuarios.

**Cuadro 6.4**  
**Características de una interfase compleja**

<b>Categoría de medición</b>	<b>Característica</b>
[1] Tipos de agentes con los cuales la empresa interactúa	Usuarios activos, usuarios conscientes, universidades, instituciones de gobierno
[2] Tipos y complejidad de la interacción*  <i>Información y conocimiento acerca de las necesidades y requerimientos de clientes-usuarios</i>  <i>Acceso a la información acerca de las necesidades y requerimientos de los clientes-usuarios</i>	Diseño y desarrollo de nuevo software, creación de recursos humanos especializados <i>Tácitos y codificados</i>  <i>A través de la interacción cara a cara, correo electrónico</i>
[3] Estabilidad de la tecnología de software utilizada en un proyecto***	Media-Baja, tecnologías de software con bajo nivel de estabilidad y no son ampliamente difundidas en la industria de software
[4] Intensidad de la interacción** <i>Cantidad de información intercambiada***</i>	Alta, proyectos de larga duración (8-16 meses) <i>Alta</i>
[5] Formalidad vs informalidad de la interacción****	Formal al inicio del proyecto, basado en contratos de compra-venta. Informal en las últimas etapas del proyecto.

\* (i) Relación compra-venta, (ii) escalamiento de software ya instalado, (iii) diseño y desarrollo de nuevo software, (iv) implantación de software, (v) mantenimiento, (vi) integración de sistemas, (vii) capacitación.

\*\* Semanas, meses, años.

\*\*\* Alta, media, baja.

\*\*\*\* Documentación y sistematización del intercambio de información y conocimiento utilizados en el proyecto.

Fuente: Elaboración propia con base en entrevistas.

Por otro lado, sin ser nuevo el estudio de la vinculación universidad-empresa, ésta tiene particularidades en la industria de software. En este apartado se ha mostrado que es posible configurar planes de estudio y por tanto crear una masa crítica que se ajuste a las necesidades y requerimientos que cambian constantemente en la industria de software. Dicha necesidad puede ser cubierta por universidades medianas, en especial las privadas,

con una estructura normativa flexible en el corto plazo. Esto es importantes porque, como menciona el líder de proyecto de la fábrica de software:

“UNIVER tiene una estructura organizativa más flexible y delgada en comparación con las grades universidades, además de la flexibilidad que muestran en la normatividad, ya que es un factor en el ajuste a los planes de estudio.”

En ese sentido, el entorno se caracteriza por la madurez de los agentes y no sólo por ser rico en información y conocimiento, y la interfase compleja se caracteriza por la interacción con múltiples agentes y por el aprendizaje interactivo con usuarios activos y conscientes.

## 6.5 RESUMEN Y CONCLUSIONES DEL CAPÍTULO

Los estudios de caso sobre los cuales se basa esta tesis, han creado algún tipo de interfase. En el cuadro 6.5 se ilustran los estudios de caso y el tipo de interfase asociada a cada uno de ellos. Constable se asocia a una interfase simple, Velásquez y Degas a una interfase semi-compleja. Manet, Renoir y Goya se asocian a una interfase compleja, aunque también brindan información para corroborar y abundar sobre las características de una interfase simple.

**Cuadro 6.5**  
**Tipo de interfase asociada a las empresas de software**

Empresa / Interfase		Simple	Semi-compleja	Compleja
SWFA*	SWP**			
	Constable	√		
	Velásquez		√	
	Degas		√	
	Manet	√		√
	Renoir	√		√
	Goya	√		√

\*SWFA: software de fuente abierta. SWP: software propietario.  
Fuente: Elaboración propia con base en entrevistas.

Los 6 estudios de caso ilustran cómo, por qué y a través de qué mecanismos se diseñan diferentes interfases entre las empresas de software y diversos agentes del entorno. En el cuadro 6.6 se resumen las características más importantes de cada una de las interfases. Los tres tipos de interfases se diseñan a partir del tipo de agente que interviene en el proyecto (usuarios, universidades, instituciones de gobierno), de la complejidad, intensidad y formalidad o informalidad de la interacción, y del tipo de tecnologías de software utilizadas en los proyectos.

**Cuadro 6.6**  
**Características de las interfases**

<b>Categoría / Interfase</b>	<b>Simple</b>	<b>Semi-compleja</b>	<b>Compleja</b>
Tipos de agente en la interacción	Usuarios pasivos, empresa integradora	Usuarios pro-activos y propositivos	Usuarios activos y conscientes, universidades, comunidades de desarrolladores de SWFA
Tipos y complejidad de la interacción	Instalación de aplicaciones que ya existen en el mercado, pequeñas modificaciones, relación compra-venta	Mejoras incrementales; análisis, diseño y desarrollo de nuevas aplicaciones; pruebas de software, capacitación	Análisis, diseño y desarrollo de nuevas aplicaciones; pruebas de software; capacitación y formación de recursos humanos
Estabilidad de la tecnología	Alta, TSW* ampliamente difundidas en la industria de software	Media, TSW* con bajo nivel de estabilidad pero difundidas en la industria de software	Media-Baja, TSW* con bajo nivel de estabilidad y poco difundidas en la industria de software
Intensidad de la interacción	Baja, proyectos de corta duración (1-2 meses)	Alta, proyectos de larga duración (8-16 meses)	Alta, proyectos de larga duración (5-24 meses)
Formalidad vs informalidad de la interacción	Formal, basada en contratos de compra-venta	Formal al inicio del proyecto, informal en las últimas etapas	Formal al inicio del proyecto, informal en las últimas etapas

\*TSW: tecnologías de software.

Fuente: Elaboración propia con base en entrevistas.

En una interfase simple predominan los proyectos que implican instalación de aplicaciones que ya existen en el mercado con pequeñas modificaciones (proyectos basados prácticamente en una relación de compra-venta) y donde los usuarios son pasivos, no se involucran en el diseño del proyecto y no requieren de una capacitación exhaustiva para el uso de la aplicación/sistema de software instalada en su empresa. Por esta razón, la intensidad de la interacción es baja y está usualmente formalizada en el contrato de compra-venta. Adicionalmente, las tecnologías de software utilizadas para el desarrollo de los

proyectos son ampliamente difundidas entre las empresas de la industria y están estandarizadas, lo que se refleja en la simplicidad del proyecto y de la interfase.

Con estas características, la empresa de software puede identificar en el entorno codificado gran parte de los requerimientos de información para el desarrollo de estos proyectos, de forma que se simplifican las necesidades de los usuarios y la necesidad de interacción con éstos.

En una interfase semi-compleja predominan los proyectos que implican mejoras incrementales, análisis, diseño y desarrollo de nuevas aplicaciones, pruebas de software y capacitación. Los usuarios son pro-activos ya que se involucran en el diseño del proyecto aportando ideas, y dada la naturaleza del proyecto, los usuarios requieren de capacitación para el uso de la aplicación/sistema de software instalada en su empresa. Por esta razón, la intensidad de la interacción es alta y, de la misma forma que en la interfase simple, usualmente formalizada en el contrato de compra-venta. Las tecnologías de software utilizadas para el desarrollo de los proyectos no son completamente estandarizadas pero son difundidas entre las empresas de la industria, esto se refleja en la complejidad del proyecto y de la interfase.

Para el desarrollo de ese tipo de proyectos la empresa proveedora identifica una gran parte de la información y conocimiento a través del usuario, ya que no se encuentran en el entorno codificado sino incorporado en los usuarios. El usuario tiene acumulada una parte importante de la información y conocimiento necesarios para el adecuado desarrollo del proyecto, es pro-activo y coopera en la medida de sus capacidades. A diferencia del usuario pasivo que se ubica en una interfase simple, un usuario pro-activo no acepta las especificaciones de los programas de software existentes en el mercado, requiere de un alto grado de adaptabilidad a sus necesidades y requerimientos, por eso la interacción es crucial y de mayor intensidad en esta interfase. Estas son las diferencias fundamentales entre una interfase simple y una semi-compleja.

En una interfase compleja predominan los proyectos que implican análisis, diseño y desarrollo de nuevas aplicaciones, pruebas de software, capacitación y formación de recursos humanos especializados. Los usuarios son activos ya que se involucran en el diseño del proyecto aportando ideas, información y conocimiento, en algunos casos tienen habilidades en el uso de tecnologías de software (herramientas, sistemas operativos, aplicaciones, etcétera). Por esta razón, la intensidad de la interacción es alta y las tecnologías de software utilizadas para el desarrollo de los proyectos son poco estandarizadas y no son ampliamente difundidas entre las empresas de la industria, esto se refleja en una mayor complejidad del proyecto y de la interfase.

Pero también hay otros agentes que intervienen, lo cual hace relevante y diferente a esta interfase respecto a las dos anteriores. La universidad y las comunidades de desarrolladores de SWFA son dos agentes con un rol aun más activo en la configuración de la interfase. La formación de recursos humanos especializados a través de la vinculación universidad-empresa implica una constante e intensa participación en la reformulación de los planes de estudio y en el modelo educativo. En las comunidades de SWFA, el usuario es consciente y toma un rol igual o más activo que la empresa desarrolladora, tiene know-how acerca de las tecnologías de software y se involucra en el desarrollo de herramientas y aplicaciones, al mismo tiempo que toma conciencia de la madurez de la aplicación o programa en proceso de desarrollo; y en particular en el contexto de SWFA, el papel activo del usuario consciente depende de la madurez de las comunidades de desarrolladores de SWFA.

Conforme las empresas crean interfaces complejas, la interacción entre proveedor-usuario y la vinculación universidad-empresa se intensifican y la cooperación tecnológica tiende al desarrollo de otro tipo de proyectos, como la fábrica de software.

Cabe mencionar que no todas las interacciones entre proveedor-usuario son igualmente importantes para el desarrollo de un producto. Algunos usuarios son pasivos y tienden a aceptar los bienes estandarizados en el mercado, aun si esos bienes no son completamente adecuados para sus necesidades. Otros usuarios recogen los beneficios del mejoramiento de los productos y tienen un rol pro-activo en el proceso de mejoramientos de los programas o

aplicaciones de software. No obstante, otros juegan un rol activo, establecen nuevas formas de interacción y son la fuente central para la creación de nuevos productos. Estos tres tipos de conducta de los usuarios, respectivamente, se muestran en una interfase simple, semi-compleja y compleja.

El argumento final de este capítulo es que las interfases externas se diseñan y son de crucial importancia porque en el segmento de software hecho a la medida, los proyectos siempre requerirán de algún grado de interacción entre las empresas y los diversos agentes para identificar adecuadamente las necesidades y requerimientos. Es decir, el desarrollo de nuevas aplicaciones de software requiere de la integración de diferentes áreas de conocimiento y habilidades, y a través de la interfase externa se puede identificar y unificar información y conocimiento fragmentarios que posee cada agente.

En algunos casos la interfase va más allá de la interacción proveedor-usuario para el desarrollo de aplicaciones/sistemas de software, y requerirá de la vinculación con universidades para la configuración de entornos de conocimiento donde se desarrollen habilidades específicas para segmentos particulares de la industria.

Uno de los problemas asociados al entorno dinámico en el cual se ubican las empresas de software es que tienen limitaciones para evaluar información y conocimiento del entorno, asimilarlo y expresarlo en nuevo conocimiento. La información y conocimiento (tácito y codificado) específicos que circulan en el entorno deben ser útiles de acuerdo a las necesidades de las empresas y deben ser asimilados adecuadamente. En el próximo capítulo se argumentará que las empresas que diseñan interfases semi-complejas y complejas tendrán una mejor posibilidad de internalizar una gran parte de la información y conocimiento existentes en el entorno y generados a partir de los proyectos de desarrollo de software. Es decir, a partir de las diferentes interfases creadas, las empresas de software pueden incrementar en diferente grado su capacidad de absorción.

## CAPÍTULO SIETE

### DIFERENTES NIVELES DE CAPACIDAD DE ABSORCIÓN A PARTIR DE LAS INTERFASES CREADAS ENTRE LAS EMPRESAS Y OTROS AGENTES

#### INTRODUCCIÓN

En el capítulo 6 se mostró que cada empresa está asociada a un tipo de interfase, y en cada una de ellas, se identifica información y conocimiento (I+C) existentes en el entorno que las empresas utilizan en el desarrollo de proyectos. En el presente capítulo se describen los mecanismos que permiten internalizar los flujos de I+C generados en el entorno, y se analizará cómo operan en cada tipo de empresa. Se mostrarán los efectos de causalidad de los diferentes tipos de interfase en la capacidad de absorción, es decir, las interfases creadas influyen de forma diferente en el incremento de la capacidad de absorción (CA) de las empresas de software.

En el primer apartado se presenta el enfoque y método utilizado para caracterizar la CA. Las categorías que se utilizan en este apartado para analizar la CA son: i) conocimiento previo, ii) flujos de información y conocimiento al interior de la empresa, y iii) actividades de aprendizaje tecnológico. Cada categoría tiene diferentes variables de aproximación.

En el segundo apartado se muestra que algunas empresas, con fuentes y mecanismos para internalizar I+C específicos, sólo incrementan su CA a un *bajo* nivel. En el siguiente apartado se discute que algunas empresas basadas en su conocimiento previo y que desarrollan actividades de aprendizaje como la capacitación, incrementan su CA *moderadamente*. En el cuarto apartado se ilustra cómo algunas empresas actualizan su fuente de conocimiento previo, desarrollan sistemáticamente actividades de aprendizaje como la capacitación y actividades de Investigación y Desarrollo (I+D), y realizan una constante búsqueda por aprender de los proyectos previos para el desarrollo de otros de mayor valor agregado. Estas empresas son capaces de incrementar su CA *significativamente*. Finalmente, se presenta el resumen y las conclusiones del capítulo.



## 7.1 ENFOQUE Y MÉTODO

El concepto principal utilizado en este capítulo es el de capacidad de absorción (CA), el cual se definió en la sección 4.3 como la capacidad de la empresa para valorar la I+C del entorno, asimilarlos y explotarlos con fines comerciales. La información para conocer los mecanismos más importantes por los cuales las empresas mexicanas de software incrementan su CA fue obtenida, principalmente, a través de entrevistas abiertas con líderes de proyecto y desarrolladores de software. La compilación de información cualitativa asociada a la elaboración de proyectos de desarrollo de software hecho a la medida permitió la reconstrucción de eventos acerca de esos mecanismos. Las categorías analíticas asociadas a las fuentes y mecanismos de absorción de información y conocimiento se explican a continuación:

### *Conocimiento previo*

El conocimiento previo asociado a las actividades propias de la empresa es la base para asimilar y utilizar creativamente la I+C para el desarrollo de nuevo conocimiento. Esta categoría se identifica a través de: a) las habilidades básicas de los empleados de las empresas de software, y b) la contratación de nuevo personal con conocimiento general (educación formal). La primera se evalúa a través del uso de técnicas de desarrollo de software asociadas a los requerimientos de la empresa, la segunda a través de la contratación de nuevo personal egresados de institutos de capacitación o universidades. Estos egresados cuentan con conocimiento general pero no específico, ya que no tienen la experiencia o la capacitación para el uso de tecnologías de software particulares que requieren las empresas.

### *Flujos de información y conocimiento al interior de la empresa*

Esta categoría está asociada a la transferencia de I+C *prácticos y específicos*, desde una unidad central de conocimiento hacia el resto de las áreas de la empresa, una adecuada transferencia entre las áreas impacta en un incremento de la CA (Lenox y King, 2003). En la literatura sobre CA, en medianas o grandes empresas, esa unidad es tradicionalmente la

de I+D, pero la micro y pequeña empresa de software generalmente no tiene una estructura organizacional funcional y, en particular, carecen de la unidad de I+D. Por tanto, en esta tesis, las formas de almacenamiento, procesamiento y transferencia de conocimiento observable al interior de la empresa son las bitácoras, los foros de discusión y los grupos de trabajo en la empresa. La primera se evalúa por la existencia de bitácoras y su uso sistemático y periódico durante los proyectos, esta actividad de procesamiento de conocimiento observable ‘permite integrar el conocimiento explícito’ (Van den Bosch, *et al*, 1999). La segunda y la tercera se evalúan por la organización de foros de discusión y grupos de trabajo de manera constante al interior de la empresa.

### ***Actividades de aprendizaje tecnológico***

Las actividades de aprendizaje son cruciales para el aumento de la capacidad de absorción de la empresa, ya que implican el desarrollo de habilidades para asimilar y utilizar adecuadamente la información y conocimiento existentes en el entorno y captados a través de las interfases externas. En la literatura sobre acumulación de capacidades tecnológicas, el concepto de aprendizaje está referido a los procesos por los cuales las empresas adquieren habilidades y conocimiento que les permiten acumular capacidades tecnológicas, sin embargo, como señala Hobday (1995:32-33), el conocimiento “es difícil de absorber y medir con exactitud o distinguir de otras actividades de manufactura.”

En esta tesis la categoría para evaluar los procesos de aprendizaje está referida a las *actividades de aprendizaje*. Se muestra que las actividades más importantes por las cuales las empresas mexicanas de software aprenden son: a) aprendizaje por reutilización, en particular, de los módulos de software utilizados en el proceso de desarrollo de software, b) aprendizaje por capacitación, c) aprendizaje por actividades multitareas, esquema en el cual un profesional participa en diferentes actividades de trabajo al interior de la empresa, y d) aprendizaje por investigación y desarrollo (I+D). Como señala Vera-Cruz (2004), el que una actividad en particular pueda convertirse en una de aprendizaje dependerá de los esfuerzos que la empresa realice para aprender de esa actividad en particular, de tal forma que las actividades de aprendizaje son específicas a cada empresa y pueden ser orientadas a diferentes objetivos.

El aprendizaje por reutilización se evalúa a través del uso de uno o más módulos en diferentes proyectos. El aprendizaje por capacitación se mide a través de la capacitación de forma sistemática del personal al interior y fuera de la empresa. El aprendizaje por actividades multitareas, asociado a la participación del personal en diferentes etapas del proyecto y en otras funciones relacionadas con el desempeño de la empresa (comercialización, contabilidad y administración), se evalúa por la frecuencia de colaboración de un empleado en cada parte del proyecto. La actividad de I+D se valora a través de la existencia de un laboratorio para actividades de ‘prueba y error’, en el cual se prueban algoritmos nuevos y se observa su funcionamiento, según los resultados obtenidos se hacen ajustes.

En la literatura sobre capacidad de absorción, tanto el aprendizaje por capacitación como el aprendizaje por I+D son ampliamente utilizados como actividades de aprendizaje organizacional y como medio para adquirir capacidades de absorción. En la industria mexicana de software ambas actividades de aprendizaje son importantes, pero la evidencia muestra que también lo son el aprendizaje por reutilización de los módulos de software y el aprendizaje por actividades multitareas. La forma de aproximación al concepto y las categorías de medición se resumen en el cuadro 7.1.

**Cuadro 7.1**  
**Operacionalización de la Capacidad de Absorción**

<b>Variable</b> (Aproximación al concepto)	<b>Indicador</b>	<b>Categoría de medición</b>
Conocimiento previo	a) Habilidades básicas de los empleados	Uso de técnicas de desarrollo de software asociadas a los requerimientos de la empresa
	b) Contratación de nuevo personal con educación formal (educación formal)	Personal egresados de institutos de capacitación o universidades con conocimiento general
Flujos de información y conocimiento al interior de la empresa	a) Bitácoras y sistemas de procesamiento	Uso sistemático y periódico durante los proyectos
	b) Foros de discusión internos	Organización de foros de discusión de manera constante al interior de la empresa
	c) Grupos de trabajo	Organización de grupos de trabajo al interior de la empresa
Actividades de aprendizaje tecnológico	a) Aprendizaje por reutilización	Uso de uno o más módulos de software en diferentes proyectos
	b) Aprendizaje por capacitación	Uso sistemático de esta actividad al interior de la empresa y fuera de ella.
	c) Aprendizaje por actividades multitareas	Frecuencia con que un empleado participa en diferentes fases de un proyecto de software
	d) Aprendizaje por investigación y desarrollo (I+D)	Existencia de laboratorio para actividades de ‘prueba y error’

## **7.2 BAJO INCREMENTO DE LA CAPACIDAD DE ABSORCIÓN A PARTIR DE UNA INTERFASE SIMPLE**

En este apartado se describen las fuentes y mecanismos por los cuales las empresas de software absorben I+C generados en el entorno. Se argumenta que a partir de una interfase simple, las empresas que hacen uso de esas fuentes y mecanismos tienen un bajo incremento en su CA. Esto se ilustra a través del análisis de las fuentes y mecanismos que la empresa Constable ha usado para internalizar I+C. El estudio de caso es relevante porque a pesar de que la empresa Constable tiene importantes fuentes de conocimiento previo, sistemas de suministro de información y conocimiento al interior de la empresa, además de realizar actividades de aprendizaje, el incremento en su CA es poco significativo, pues no ha realizado proyectos de mayor valor agregado.

### **7.2.1 Conocimiento previo**

La literatura sobre CA ha puesto énfasis en que las empresas deben contar con un grado de conocimiento previo para desarrollar habilidades y evaluar I+C del entorno, asimilar y explotarlos comercialmente, dado que las habilidades desarrolladas están basadas en el conocimiento previo. Dos formas mediante las cuales la empresa ha obtenido éste conocimiento son las habilidades básicas con las que cuentan los empleados, sin ellas, la empresa no tendría oportunidad de mantenerse en su correspondiente nicho de mercado, y la contratación de nuevo personal con educación formal.

#### ***Habilidades básicas de los empleados***

Con base en entrevistas realizadas a gerentes y líderes de proyecto de los 6 casos en estudio, las habilidades básicas de los empleados de la empresa están asociadas a técnicas de desarrollo de software tales como programación, prueba de software, de compilación, mantenimiento, además del dominio del idioma inglés, entre otras. Pero también depende del nicho de mercado de cada una. Constable, por ejemplo, realiza proyectos basados en tecnología WEB y las habilidades más específicas están relacionadas al diseño de gráficos y programación con PUNTO NET. Según el gerente de proyecto de esta empresa:

“Los ingenieros involucrados en los proyectos cuentan con las habilidades básicas, las otras [habilidades más específicas] se van formando en la empresa.”

### ***Contratación de nuevo personal con conocimiento general***

Hasta el momento de las entrevistas, Constable había contratado 2 profesionistas con educación formal egresados de universidades. Pero, según el gerente de proyecto:

“La mayoría de los egresados cuentan con conocimiento general pero no específico. Tienen un amplio conocimiento acerca de lenguajes de programación, herramientas, técnicas, etcétera, pero no tienen las habilidades específicas requeridas por la industria. Muchas veces hay que dedicar una buena cantidad de recursos humanos y financieros para capacitarlos de acuerdo a las necesidades tecnológicas de la empresa. En un segundo momento, requerimos de otro tipo de capacitación, la cual está relacionada con la administración de la empresa, la gestión de recursos y proyectos, la toma de decisiones, o bien cursos de actualización del idioma inglés.”

Esto refleja que el perfil de los profesionales es limitado comparado con las características que debieran cubrir un desarrollador o ingeniero de software. Si bien la empresa cuenta con las habilidades básicas, el mercado está demandando otras habilidades: administrativas y de gestión, de evaluación de proyectos, marketing, entre otras.

La categoría de conocimiento previo es evaluada por el uso de técnicas de desarrollo de software asociadas a los requerimientos de la empresa y por la contratación de profesionistas con educación formal. Dado que esta categoría es una condición necesaria, no es única para que las empresas puedan absorber I+C del entorno y generados a partir de las interfases externas. Como veremos en el siguiente apartado, la CA depende también de los sistemas de procesamiento de la información y conocimiento, así como de la forma en que se suministran o transfieren al interior de la empresa.

### **7.2.2 Flujos de información y conocimiento al interior de la empresa**

Como se señaló en el apartado 7.1, la categoría *flujos de información y conocimiento al interior de la empresa* está asociada a la transferencia de I+C desde una unidad central de

conocimiento hacia el resto de las áreas de la empresa. Sin embargo, como veremos en esta sección, Constable sólo ha logrado desarrollar mecanismos de procesamiento de I+C.

### ***Procesamiento de conocimiento observable***

Una vez que la empresa obtiene conocimiento previo necesario para reconocer el conocimiento externo, el siguiente reto es cómo asimilarlo y transferirlo al interior de la empresa. Lane y Lubatkin (1998:465) argumentan que, para que una empresa pueda asimilar la I+C del entorno, debe contar con un *sistema de procesamiento* de información y conocimiento.

Constable ha implementado el Modelo de Maduración de Capacidades (CMM, por sus siglas en inglés, Capability Maturity Model) y obtendrá el nivel 3. Como se mencionó en el capítulo 1, este modelo, además de evaluar la calidad, confiabilidad y desempeño de los procesos de desarrollo de software, implica la implementación de sistemas de procesamiento de la información y conocimiento, de forma que el proceso de certificación requiere que se documenten las actividades desarrolladas por la empresa. Sin embargo, como dice el gerente de proyecto de esta empresa:

“La implementación de estos sistemas permite procesar una cantidad importante de información y conocimiento que se genera a lo largo de los proyectos. De esta forma se salvaguardan las experiencias. Si alguien detecta alguna falla técnica en un módulo, se revisa el sistema donde se salvaguardan estas experiencias y se categoriza la solución de problemas. Los conocimientos externos a la empresa también quedan registrados en el sistema...pero es más difícil procesar en un sistema las experiencias particulares [o muy específicas] de los desarrolladores.”

Como se observa, el sistema permite asimilar una cantidad importante de información genérica y de conocimiento codificado, pero la asimilación de conocimiento tácito tiene otras particularidades, como se verá más adelante. De acuerdo a algunos líderes de proyecto, “una de las razones por la que es importante contar con un sistema de procesamiento es que permite a las empresas saber cómo actuar ante eventos similares”.

La empresa Constable ha adoptado las *bitácoras* como mecanismo para procesar I+C que se genera a lo largo de los proyectos. A través de las bitácoras los desarrolladores pueden ubicar procesos, problemas y soluciones. Este medio es común en la empresa:

“Todos los eventos nuevos que se presentan en el desarrollo de software se registran en bitácoras, las cuales pueden hacer referencia a documentos o a código. Si hay un documento en particular que define el problema y lo soluciona, entonces la bitácora hace referencia a ese documento. La empresa llena bitácoras diariamente, genera respaldos y se configuran.”

Las bitácoras y el monitoreo, así como el control de los procesos son una de las formas de salvaguardar la información y conocimiento generado a partir de los proyectos de desarrollo de software. Pero, como bien señala el gerente de proyecto:

“Una parte del conocimiento que se genera a través de los grandes proyectos es codificable, la otra parte es tácita, es decir, resulta difícil procesar todas las experiencias.”

Según lo mencionado en el capítulo 2, las empresas que están en procesos de certificación lo hacen con apoyo de la Empresa Integradora a la que están asociadas, como es el caso de Constable. El gerente de proyecto menciona que “de no haber contado con el apoyo hubiera sido difícil acceder al proceso de certificación por el costo que representa”.

Las empresas de software que cuentan con un sistema de procesamiento de información y conocimiento podrían transferirlo al interior de la empresa. Así lo comenta el líder de proyecto de la empresa Constable:

“Una vez que se salvaguardan las experiencias en un sistema, es relativamente más fácil categorizar el tipo de solución a determinados problemas técnicos, los desarrolladores pueden acceder a ese sistema y revisar procesos, problemas y soluciones específicas.”

Sin embargo, no hay evidencia clara de que las bitácoras sean un mecanismo de transferencia de la I+C al interior de Constable, es decir, las bitácoras no operan como una unidad central de conocimiento y de transferencia hacia el resto de las áreas de la empresa.

En suma, la categoría de *flujos de información y conocimiento al interior de la empresa* se evalúa por el uso sistemático y constante de las bitácoras de trabajo durante el proyecto,

pero Constable no muestra claridad en la forma en cómo, a partir de las bitácoras, se transfieren al interior de la empresa.

### **7.2.3 Actividades de aprendizaje tecnológico**

Como se mencionó en el apartado 7.1, son cuatro las actividades más importantes por las cuales las empresas de software aprenden. La única actividad de aprendizaje que la empresa Constable no ha desarrollado es la de I+D.

#### ***Aprendizaje por reutilización***

Una de las principales actividades de aprendizaje utilizadas por la empresa Constable es la ‘reutilización de módulos o partes de una aplicación’. En un proyecto de software hay aplicaciones grandes o pequeñas y una parte de ellas puede ser reutilizable en otros proyectos. Con base en entrevistas al gerente de proyecto de la empresa Constable:

“El desarrollo de software se puede realizar siguiendo diferentes modelos y dependiendo del tamaño del proyecto se utilizará uno u otro. Aun con diferentes modelos, e independientemente de si un proyecto es pequeño, el proyecto siempre tendrá un módulo de *seguridad* y un módulo *funcional*. Al menos estos dos tienen una aplicación. A lo largo de un proyecto se pueden crear diferentes módulos, algunos de ellos podrán ser reutilizables, otros no, depende de la naturaleza de los proyectos, pero cuando se reutilizan hay un grado de adaptación al nuevo proyecto. La reutilización implica, además del ahorro de tiempo en la codificación, el aprendizaje.”

Esto permite que en algunas partes del desarrollo del proyecto haya un grado de especialización en el uso de ciertos módulos de una herramienta o aplicación de software. Para Constable, “el tener especialización implica no subcontratar nada”.

#### ***Aprendizaje por capacitación***

Otra de las actividades más relevantes para Constable es la capacitación externa. Respecto a ésta, no todos la toman, sin embargo, el gerente de proyecto menciona que:

“Al menos cada trimestre alguien está en capacitación externa. En ocasiones se envía a capacitar a una o dos personas, la razón de que no se envíe a capacitar a más personal es el



costo (en algunas ocasiones la capacitación es en el extranjero, acerca de modelado, utilización de herramientas especializadas, entre otras). Quienes toman la capacitación tratan de compartir el conocimiento adquirido a los demás...es una forma de reducir el costo. La capacitación es una de las estrategias más importantes para la empresa ya que está asociada al proceso de certificación del modelo CMM. Hay gente que tiene mayor experiencia en ciertas actividades propias del diseño o desarrollo, y la forma de que otros aprendan es por medio de la capacitación, la cual es de gran ayuda para el resto del personal, sobre todo cuando éste es nuevo.”

Esta empresa capacita externamente a una parte de su personal debido al proceso de certificación en el que se involucró, el cual requiere de una constante capacitación, pero es poco frecuente para otros propósitos diferentes de los que requiere ese proceso, por ejemplo, en pruebas de software, modelado, etc. Por otra parte, si bien los empleados de mayor experiencia capacitan a los de menor experiencia, al no haber una capacitación externa en esas tecnologías que constantemente evolucionan, la empresa limita su frontera de desarrollo y limita su desarrollo de proyectos de mayor valor agregado.

### ***Aprendizaje por actividades multitareas***

En el apartado 7.1, se mencionó que la movilización de recursos se refiere a difundir las capacidades existentes al interior de la empresa. Una actividad importante para lograrlo son las actividades multitareas, la cual se identifica por la frecuencia con que un empleado participa en diferentes fases de un proyecto de software. Si bien las empresas de software se refieren a su organización de trabajo como una forma de ‘rotación de roles’, en realidad su concepto se parece a un ‘esquema multitareas’, en el cual el profesional debe desempeñar simultáneamente varias actividades.

Una de las características organizacionales y productivas de las pequeñas empresas de software es la participación de los profesionales en actividades multitareas. Un proyecto que requiera de varias etapas (como se describe en el Anexo Uno) implicará la participación de un grupo de personas, básicamente desarrolladores y líder(es) de proyecto:

“Lo común en la industria de software [sobre todo en la micro y pequeña empresa], aunque no es una regla, es que una persona tenga que cubrir varios roles. Por ejemplo, el líder de

proyecto puede ser al mismo tiempo desarrollador, agente de relaciones públicas, publicista, administrador, entre otras múltiples actividades, mientras que un desarrollador puede realizar las tareas de programación, testing, compilación, escalamiento, entre otras.”<sup>1</sup>

Esto se expresa mejor en la siguiente cita:

“Es muy frecuente que una persona tenga varios roles en un proyecto. Por ejemplo, para realizar un *sistema de flujos de trabajo*, el líder o gerente de proyecto tiene que dirigir el rumbo del proyecto, tiene que monitorear los posibles problemas y soluciones que no pueda dar el desarrollador. Pero, además, tiene que estar pendiente de tareas administrativas, de comercialización, y hacer gestión de negocios ya que muchas veces el líder o gerente de proyecto es al mismo tiempo dueño o socio de la empresa. Pero esto tiene que dejar de ser así, sobre todo cuando la empresa crece ya que se divide en áreas para cada actividad.”

La razón de por qué los líderes y desarrolladores tienen tareas múltiples se debe a que la micro y pequeña empresa no cuenta con áreas funcionales formalmente, como se mencionó en el capítulo 1. Si bien puede crecer el tipo y número de proyectos (hacia proyectos más complejos y de mayor duración) no crece el tamaño de la empresa o no se desarrollan esas áreas.

Esta categoría se evalúa por el uso de uno o más módulos de software en diferentes proyectos, por el uso sistemático de la capacitación al interior de la empresa y fuera de ella, y por la frecuencia con que un empleado participa en diferentes fases de un proyecto de software.

Hasta aquí se ha mostrado que Constable, asociada a una interfase simple, ha realizado significativos esfuerzos internos. Primero, tiene importantes fuentes de conocimiento previo: el personal cuenta con las habilidades básicas para el desarrollo de proyectos basados en tecnologías WEB, y además ha contratado nuevo personal con educación formal que ha sido capacitado en esas tecnologías específicas. Segundo, la empresa ha instalado sistemas de suministro de información al interior de la empresa: el proceso de certificación de calidad que ha implementado (CMM) implica contar con sistemas de procesamiento de

---

<sup>1</sup> Entrevista con el gerente de proyecto, empresa Constable.

conocimiento observable, lo cual lo logra a través del desarrollo de bitácoras. Tercero, realiza actividades de aprendizaje tales como: la reutilización de módulos y por tanto de procesos, la capacitación externa y al interior de la empresa, y está en un esquema multitareas de trabajo, como una gran cantidad de micro y pequeñas empresas de la industria.

No obstante, la pregunta es por qué sólo se mantiene en un esquema de replicar y hacer mejoras y escalamiento menores a sistemas de software que ya existen en el mercado, y no desarrolla nuevos productos o diversifica su nicho de mercado. Constable puede identificar I+C de propósito general existentes en el entorno (como se mostró en el capítulo anterior), tiene los elementos para internalizarlos adecuadamente (como se ha mostrado en este capítulo), pero los resultados de su explotación son menores a los que podría alcanzar. En este sentido, tiene las capacidades para identificar y absorber I+C específicos de los usuarios, pero no lo hace, identifica e internaliza solamente los de propósito general; tiene la capacidad para desarrollar proyectos de valor superior, establecer una mayor interacción con los agentes y crear otro tipo de interfase que implique la aplicación del aprendizaje logrado a lo largo de los proyectos. Por esta razón, a pesar de los esfuerzos que la empresa ha realizado internamente, el incremento en su CA es ‘bajo’, está sub-utilizando su CA.

El hecho de que su nicho de mercado le represente alrededor de 1.5 millones de pesos anuales es un elemento que le da ‘comodidad’, y no parece generar incentivos para desarrollar proyectos de mayor valor agregado o diversificar su mercado. Esto indica estar acompañado de una aversión al riesgo por cambiar el sendero tecnológico que le otorga cómodos márgenes de ganancia.

### **7.3 MODERADO INCREMENTO DE LA CAPACIDAD DE ABSORCIÓN A PARTIR DE UNA INTERFASE SEMI-COMPLEJA**

En este apartado se argumenta cómo, a partir de una interfase semi-compleja, dos empresas (Velásquez y Degas) han incrementado moderadamente su CA.

### 7.3.1 Conocimiento previo

#### *Habilidades básicas de los empleados*

Con base en las entrevistas realizadas a gerentes y líderes de proyecto de las empresas Velásquez y Degas, las habilidades básicas de los empleados de la empresa están asociadas a técnicas de desarrollo de software, tales como programación, prueba de software, compilación<sup>2</sup>, mantenimiento, además del dominio del idioma inglés, entre otras. Pero también depende del nicho de mercado de cada una. Velásquez se especializa en realizar pruebas de software (testing) y requiere de “programadores con habilidades básicas en pruebas de software”.<sup>3</sup> Como se mencionó en el capítulo anterior, hay varias empresas de software que realizan esta actividad como parte de sus etapas de desarrollo, pero no se especializan en ella, de forma que al entregar el producto o sistema de software tienen fallas debido a la baja calidad. La especialización de Velásquez radica en realizar pruebas de software, y para ello requiere de profesionales con estas habilidades. Degas se especializa en programas de software para empresas de tenería y requiere “que los ingenieros de software tengan conocimiento en áreas relacionadas con el proceso de tenería y habilidades básicas de programación en JAVA, ORACLE y LINUX”.<sup>4</sup>

#### *Contratación de nuevo personal con conocimiento general y específico*

Hasta el momento de las entrevistas, Degas había contratado un profesionalista con conocimiento general y específico. Tuvo educación formal en una universidad de León y aprendió una de las tecnologías centrales sobre las cuales se basa el desarrollo de productos de la empresa. Según el líder de proyecto de esta empresa:

“La empresa requiere de personal que cuente con los conocimientos básicos (lenguajes de programación, herramientas, técnicas, etcétera), pero deben contar con conocimiento específico en JAVA y ORACLE. Sin embargo, algo que es medular para la empresa es el desarrollo de productos para la industria de tenería, y es difícil que los profesionales de software o informática adquieran los conocimientos acerca de los procesos de tenería en la escuela, esos conocimientos los adquieren en la empresa.”

---

<sup>2</sup> En el ámbito de la programación, compilación es un proceso por el cual se traducen programas en *código fuente* a programa en *código objeto*. El programa que realiza esta traducción se llama compilador.

<sup>3</sup> Entrevista con líder de proyecto, empresa Velásquez.

<sup>4</sup> Entrevista con líder de proyecto, empresa Degas.

Como se mencionó en el capítulo anterior, los ingenieros que ahora son parte de Degas habían trabajado en empresas de tenería, se involucraron en las tecnologías de software y al cabo de unos años lograron el desarrollo de productos de software que no existían en el mercado nacional. La empresa inició con el conocimiento previo para el nicho de mercado al que destina sus productos, la industria de tenería, pero ese conocimiento previo fue adquirido al interior de las propias empresas de tenería. En palabras del líder de proyecto, “capacitar a un profesionalista para el dominio de las tecnologías y productos que realiza la empresa requiere alrededor de 2 años”. La duración se debe a la dificultad y complementariedad de conocimientos: los que son propios de la programación y los que son particulares a los procesos de tenería.

En párrafos anteriores se indicó que cada una de las empresas tiene nichos específicos de mercado, y la contratación de nuevo personal y su especialización tecnológica depende de su mercado.

La categoría *conocimiento previo* es evaluada por el uso de técnicas especializadas de desarrollo y prueba de software de parte de los profesionales. Esas técnicas están asociadas a los requerimientos de la empresa, la cual requiere complementariedades de conocimiento, por tanto la contratación de profesionistas es más específica.

### **7.3.2 Flujos de información y conocimiento al interior de la empresa**

#### ***Procesamiento de conocimiento observable***

Velásquez estuvo en proceso de certificación, por tanto implementó el Modelo de Maduración de Capacidades (CMM) para obtener el nivel 3. Las actividades propias del modelo requieren la implementación de un sistema de procesamiento de la información y conocimiento, ya que la empresa debe documentar las actividades desarrolladas durante los proyectos. Como se mencionó en párrafos anteriores, este tipo de sistemas de procesamiento de información y conocimiento permiten que se “salvaguarden las experiencias” y que puedan acudir quienes requieran conocer algún tipo de solución a problemas específicos.

Con base en las entrevistas realizadas a gerentes y líderes de proyecto, en la industria mexicana de software hay pocas empresas, sobre todo las micro y pequeñas, que cuentan con sistemas de procesamiento de información y conocimiento, y algunas que lo llegan a desarrollar es porque están en algún modelo de certificación de procesos de desarrollo de software:

“Una gran cantidad de empresas mexicanas de software no desarrollan software con una metodología de procesos, lo hacen de manera muy *artesanal*, además, cada empresa lleva un modo distinto... y la mayoría desarrolla software por prueba y error. Lo que esto provoca es que los sistemas se desarrollan con poca calidad que el cliente solicita, que se entreguen fuera de tiempo y que los proyectos rebasen el presupuesto original.”

En suma, la categoría *flujos de información y conocimiento al interior de la empresa* se evalúa por el uso sistemático y constante de un ‘sistema de procesamiento de información’ derivado de la implementación del modelo CMM, en particular para la empresa Velásquez.

### **7.3.3 Actividades de aprendizaje tecnológico**

A partir de los proyectos desarrollados por Velásquez y Degas, sobre los cuales se presentó evidencia en el capítulo anterior, se deducen las actividades de aprendizaje más importantes que realizan: capacitación y actividades multitareas al interior de la empresa.

#### ***Aprendizaje por capacitación***

Para el caso de Velásquez, la capacitación es predominantemente externa, en cambio, es interna para Degas. De la misma forma que para otras empresas, para Velásquez “la capacitación es una de las estrategias más importantes para la empresa ya que está asociada al proceso de certificación del modelo CMM”. El personal está en constante capacitación debido a que una de las áreas en proceso de integración es la de I+D, Velásquez es de las pocas empresas mexicanas de software que cuenta con un laboratorio de I+D, y como menciona el líder de proyecto:

“...Aunque en México hay gente que hace pruebas de software, no la hay en calidad para hacer actividades de I+D en *testing* como lo hace la empresa. Por eso es necesario la capacitación externa, la cual normalmente se toma fuera de México ya que no hay quién la

dé en el país; también es muy importante la asistencia a seminarios y congresos especializados.”

La capacitación externa es constante, por el tipo de proyectos que realiza, pero representa un costo muy alto para la empresa:

“Siempre hay un temor de que el personal capacitado interna o externamente deje la empresa, ya que la capacitación implica costos que absorbe la empresa y es constante que los ingenieros busquen otras opciones. Pero esto es común en la industria, el problema es que constantemente tenemos que estar capacitando a alguien y no sabemos si permanecerá por un tiempo en el que podamos decir que se ha amortizado ese costo.”

La expresión anterior es una preocupación particular para Velásquez, pero en realidad lo es también para las 6 empresas. No hay forma de evitar la alta rotación de personal en la industria de software. Pero, como se mostró en el capítulo anterior, algunas empresas como Goya han tratado de minimizar este costo a través de la vinculación universidad-empresa, formando recursos especializados al interior de la universidad.

Para el caso de Degas, la capacitación toma otro nivel. Si bien considera que es importante la capacitación externa, el líder de proyecto menciona que:

“El personal no sólo debe conocer las tecnologías de software sino también debe conocer la estructura de costos o los procesos de ingeniería específicos para tenería. La gente que ahora trabaja en Degas requiere de al menos dos años de capacitación, pero no es una capacitación convencional como la conocemos en el sentido de asistir a talleres o foros sino que la capacitación se obtiene a través de las actividades desarrolladas en empresas propiamente de tenería. Los ingenieros de la empresa deben conocer, en particular, los procesos de curtido, teñido, etcétera. La capacitación interna es más importante ya que los conocimientos acerca de los procesos de tenería sólo se pueden aprender por las experiencias adquiridas a lo largo de los proyectos, a través del contacto con los ingenieros que tienen el conocimiento integrado, [conocimientos de programación y de los procesos de tenería].”

En este sentido, a pesar de no haber expresión de una capacitación interna formal, el contacto con los ingenieros de mayor experiencia permite el aprendizaje, aunque, como

menciona el líder de proyecto, “adquirir ese aprendizaje integrado podría tardar hasta 2 años”. Sobre todo porque el ingeniero de nueva contratación no ha trabajado en empresas de tenería, como sí lo hicieron los ingenieros anteriores.

### *Aprendizaje por actividades multitareas*

Con base en párrafos anteriores, es usual el esquema multitareas entre las empresas de software. Un argumento es que la empresa no cuenta con áreas funcionales formalmente, la organización del trabajo está basada en actividades multitareas, por tanto el líder de proyecto puede ser desarrollador, agente de relaciones públicas, publicista, administrador, entre otras múltiples actividades; mientras que un desarrollador puede realizar las tareas de programación, testing, compilación, escalamiento, entre otras.<sup>5</sup>

Los líderes y gerentes de proyecto de las empresas entrevistadas coinciden en que eso no debe continuar así, pero no es clara la forma en cómo se pueda solucionar. Mencionan que algunas empresas que han crecido en tamaño lo han solucionado, pero esta solución es casi natural pues en la medida que la empresa crece, hay una tendencia hacia la división del trabajo y, ante esto, una forma de organización del trabajo es estructurar a la empresa en áreas funcionales formalmente.

La categoría *actividades de aprendizaje* se evalúa por el uso sistemático de la capacitación al interior de la empresa (para el caso de Degas) y fuera de la empresa (para el caso de Velásquez), también se evalúa por la alta frecuencia del uso del esquema multitareas al interior de los proyectos.

En el capítulo 6 se señaló que Velásquez y Degas se asocian a una interfase semi-compleja. En este apartado se ha mostrado que ambas han realizado esfuerzos importantes a nivel organizacional, aunque ambas han desarrollado menos actividades (respecto al caso del apartado anterior) con el objeto de aumentar su CA. Ambas empresas concentran gran parte de sus esfuerzos internos en la capacitación externa (Velásquez) e interna (Degas) para

---

<sup>5</sup> Esta cita se deriva de entrevistas realizadas a gerentes y líderes de proyecto de las empresas en estudio. Todos convergen en la misma cita primeramente señalada por el gerente de proyecto de la empresa Constable.



aumentar su CA. Velásquez no ha contratado nuevo personal, a pesar de que en términos de proyectos ha crecido, ni tampoco tiene un área formalmente funcional de I+D<sup>6</sup>, aunque el líder de proyecto menciona que la empresa es en sí ‘un laboratorio de I+D’ por el tipo de actividad que desarrolla la empresa. Asimismo esta firma no usa con frecuencia la reutilización de módulos de herramientas o aplicaciones de software. Por su parte, Degas sí ha contratado nuevo personal con conocimiento previo, pero no cuenta con sistemas de procesamiento de información, ni con un área de I+D, tampoco hace uso frecuente de la reutilización de módulos.

Sin embargo, estas empresas han sido capaces de establecer nuevos proyectos que implican mejoras incrementales, análisis, diseño y desarrollo de nuevas aplicaciones, así como pruebas de software y capacitación a usuarios. Para lograr el desarrollo de este tipo de proyectos, las empresas han mostrado capacidad para identificar la información y conocimiento de propósito general en el entorno, pero también aquellos que son específicos a través de los usuarios. La internalización de esta información y conocimiento se ha logrado básicamente por la capacitación.

Por tanto, estas empresas han incrementado moderadamente su CA. Velásquez pudo haberse quedado en un esquema de subcontratación y depender de la demanda de empresas que desarrollan productos de software, pero, dado que es pionera en su nicho de mercado, trata de penetrar en otros mercados con un servicio cada vez de mayor calidad, además, uno de sus mayores recursos es la capacitación de su personal. Por su parte, el mayor logro de Degas ha sido la internalización de I+C específicos de los usuarios y el subsecuente desarrollo de productos para empresas de tenería. Éstas empresas tienen procesos estandarizados y estables y conocimiento de uso común para ellas, pero no explícitos para Degas, de forma que una tarea importante de dicha empresa fue traducir en un ‘lenguaje común’ las necesidades específicas de los usuarios. De esta forma la complementariedad de conocimientos básicos y específicos es importante y, en conjunto con las actividades de aprendizaje, ayuda a la empresa a incrementar moderadamente su CA.

---

<sup>6</sup> Según el líder de proyecto está en proceso de consolidación una área de I+D, la cual contará con computadoras y un servidor, las actividades del área estarán basadas en procesos de “prueba y error”.

## **7.4 SIGNIFICATIVO INCREMENTO DE LA CAPACIDAD DE ABSORCIÓN A PARTIR DE UNA INTERFASE COMPLEJA**

En este apartado se discutirá que, a partir de una interfase compleja, 3 empresas (Manet, Renoir y Goya) han incrementado significativamente su CA. Estos estudios de caso han desarrollado interfases complejas, aunque también proporcionaron información para deducir que desarrollaron interfases simples, como se mostró en el capítulo 6. La evidencia presentada en este apartado permite argumentar que a partir de una interfase compleja, en combinación con los esfuerzos internos, las empresas pueden aumentar significativamente su CA.

### **7.4.1 Conocimiento previo**

#### ***Habilidades básicas de los empleados***

Manet y Renoir se especializan en tecnología WEB y las habilidades básicas están relacionadas al diseño de gráficos y programación con PHP y LINUX. Estas empresas utilizan herramientas basadas en software de fuente abierta. Goya desarrolla diferentes actividades y cada una de ellas demanda diferente tipo de conocimiento y habilidades. Una de las más importantes es el desarrollo de software, para lo cual requiere que los profesionales tengan habilidades básicas en JAVA y en herramientas de MICROSOFT, entre otras.

#### ***Contratación de nuevo personal con conocimiento general***

Hasta el momento de las entrevistas, Renoir y Goya habían contratado profesionistas con conocimiento general, es decir, egresados que fueron formados con educación formal en universidades o centros técnicos de capacitación. Los gerentes de proyecto de estas empresas coinciden en que:

“Los egresados cuentan con conocimiento general pero no específico. Tienen conocimiento acerca de lenguajes de programación, herramientas, técnicas, etcétera, pero no habilidades específicas requeridas por la industria. De forma que se dedica una buena cantidad de recursos humanos y financieros para capacitarlos de acuerdo a las necesidades tecnológicas de la empresa.”

Como se mencionó en párrafos anteriores, cada una de las empresas en estudio tiene nichos específicos de mercado y la contratación de nuevo personal y su especialización tecnológica depende de éste. Renoir, por ejemplo, se especializa en tecnología WEB y requiere de diseñadores gráficos, además de programadores e ingenieros de software.

A diferencia de las empresas anteriores, Goya es una empresa mediana. Las actividades que lleva a cabo son diversas, exigiendo cada una de ellas distintos tipos de conocimiento y habilidades.

“Por ejemplo, una parte importante de las actividades de la empresa es la consultoría, otra la venta de aplicaciones empaquetadas o el desarrollo de software hecho a la medida, y también es importante la gestión de proyectos como el de vinculación universidad-empresa o la participación en la creación de ‘Integradoras’. Sin embargo, una de las actividades más relevantes es el desarrollo de software y para ello se requieren programadores e ingenieros.”

Manet se ha especializado en tecnologías WEB y en aplicaciones contables y administrativas. Esta empresa fue formada por 4 desarrolladores independientes y todos habían hecho desarrollos a la medida para firmas de manufactura y servicios. Esta empresa no ha contratado profesionales con conocimiento general como lo han hecho las anteriores, pero el conocimiento que ellos han adquirido ha sido a través del trabajo independiente de cada uno, es decir, ya contaban con el conocimiento previo necesario y lo adquirieron individualmente.

La categoría *conocimiento previo* es evaluada por el conocimiento de los profesionales en el uso de técnicas genéricas de desarrollo de software (para el caso de Renoir y Goya), las cuales están asociadas a los nichos específicos de las empresas. En el caso de Manet, los integrantes de la empresa tenían conocimiento previo que adquirieron como profesionales independientes antes de formar la empresa. También ha sido evaluada por la contratación de profesionales (excepto para el caso de Manet).

Cada empresa requiere de diferente tipo de conocimiento previo, el cual dependerá del nicho de especialización de mercado en el que se encuentren. Una vez adquirido este conocimiento, las empresas tendrán la posibilidad de generar habilidades para asimilar

nueva información y conocimiento y así, le darán un uso creativo para desarrollar nuevo conocimiento.

En el primer apartado de este capítulo se mencionó que las habilidades de administración, gestión, entre otras, también son requeridas para integrar de manera eficiente todo el conjunto de actividades organizacionales y productivas en un proyecto. Estas habilidades son importantes para la industria de software, porque los programadores e ingenieros no tienen una formación en esas áreas, y sólo pueden obtenerla a través de la capacitación y la experiencia lograda al interior de la empresa.

#### **7.4.2 Flujos de información y conocimiento al interior de la empresa**

##### ***Procesamiento de conocimiento observable***

De la misma forma que Constable y Velásquez, Renoir y Goya han implementado el Modelo CMM y obtendrán también el nivel 3. Se ha mencionado en párrafos anteriores que este modelo implica la implementación de sistemas de procesamiento de información y conocimiento, de forma que el proceso de certificación requiere de documentar las actividades desarrolladas.

La realidad de varias empresas, sobre todo micro y pequeña, es la falta de recursos para certificar sus procesos de desarrollo en modelos como CMM. Como ya se ha mencionado, las empresas Constable, Velásquez, Renoir y Goya han podido lograr incorporarse a un proceso de certificación porque cuentan con el apoyo de una Empresa Integradora o con apoyos estatales.

En el apartado 7.3.2 se citó que pocas empresas cuentan con sistemas de procesamiento de información y conocimiento, no cuentan con metodologías de procesos, lo cual ocasiona que los proyectos se entreguen fuera de tiempo y rebasen los costos originales. Las empresas que logran rebasar/superar estos problemas son porque están en algún modelo de certificación de procesos.

Empresas como Goya han desarrollado actividades alternas como las *bitácoras* para documentar la información y conocimiento generados a lo largo de los proyectos, y que al mismo tiempo sirve como medio de transferencia de la información y conocimientos, en particular en Manet, Renoir y Goya. Como menciona el gerente de proyectos de la empresa Goya, “a partir de las bitácoras los líderes de proyecto y desarrolladores pueden ubicar procesos, problemas y soluciones”, este medio es “común en la empresa”.

Según Lenox y King (2003), los suministros de información son una de las actividades más importantes que impactan en el aumento de la CA de la empresa, ya que están ligados a la toma de decisiones, y ésta depende del suministro de información interna, es decir, de la información que fluyen de un área específica hacia el resto de la empresa. Esto implica que si las empresas tienen sistemas de procesamiento de información y conocimiento, resulta relativamente fácil ubicar un área suministradora de otras áreas. Pero se ha mencionado que las empresas en estudio carecen de áreas funcionales formalmente y los suministros de información se representan por un *sistema de procesamiento* informático (bitácoras de trabajo) y no por áreas.

### ***Transferencia de conocimiento entre las áreas***

En los proyectos realizados por las empresas Manet, Renoir y Goya hay conocimiento tácito que no se puede procesar mediante un sistema informático o mediante las bitácoras, debido a que “los programadores e ingenieros tienen dificultades para sistematizar lo que aprenden a lo largo de los proyectos”, por tanto, no se puede transferir de manera sistemática. Pero hay otros medios por los cuales se puede asimilar y transferir conocimiento tácito. Por ejemplo, resultan particularmente importantes en estas empresas los *foros de discusión* al interior de la empresa, y como bien lo expresa el líder de proyecto de la empresa Renoir:

“El desarrollo de software es una actividad que requiere el uso de conocimiento que no está codificado por completo, ya que los desarrolladores tienen conocimiento que han adquirido a lo largo de su trayectoria profesional y que no documentan por dos razones: porque es difícil trasladar ese conocimiento a papel o a un sistema [informático], y porque a veces los desarrolladores no tienen establecidas las rutinas para transferir ese conocimiento a medios codificados y además no están acostumbrados a hacerlo. La experiencia no siempre es

codificable, pero se puede difundir a través de un foro. Por ejemplo, cuando hay un problema particular en algún proceso del proyecto se discute en el foro. En función de las experiencias compartidas entre los integrantes siempre se trata de llegar a una solución, la cual se almacena en un sistema [cuando es posible trasladar la solución a medios codificados], lo cual permite que la solución al problema se quede en la empresa, pero no siempre es posible almacenarla. Los foros de discusión se hacen una vez a la semana, y sólo si hay soluciones a un problema se difunde el conocimiento [por medios codificados].”

En palabras del líder de proyecto de la empresa Renoir, ‘la solución a un problema es reactiva’ y periódica, ya que al menos un día a la semana se reúnen ‘para comentar las experiencias’:

“A través de los foros semanales se comparten experiencias, y a través del trabajo en grupo se integran tareas individuales especializadas.”

Los foros de discusión permiten que cierto conocimiento generado a lo largo de los proyectos se acumule en más de una persona. De forma que, cuando un empleado decide dejar la empresa, una parte del conocimiento adquirido por éste está salvaguardado en otro empleado. Dado que en la industria de software hay una alta rotación de personal, es frecuente que un profesional decida cambiar de lugar de trabajo,<sup>7</sup> de tal forma que los foros de discusión y el trabajo en grupo permiten conservar una parte del conocimiento generado a través de los proyectos.

Otro de los elementos que las empresas de software consideran como medio de transferencia de información y conocimiento es la interacción con otros agentes del entorno. Algunas empresas como Renoir han establecido alianzas con otras sociedades de la misma industria y las consideran como “un medio para compartir las experiencias logradas en los proyectos”. También la Empresa Integradora es otro medio de transferencia, pero con ciertas limitaciones como las descritas en el capítulo 6.

---

<sup>7</sup> Aunque en 2004, según los entrevistados, la rotación de personal en la industria de software fue baja. Esta es una de las causas por las cuales la categoría de empleados temporales se ha vuelto común.

En suma, la categoría de *flujos de información y conocimiento al interior de la empresa* se evalúa por el uso sistemático y constante de un ‘sistema de procesamiento de información’ derivado de la implementación del modelo CMM, así como de las ‘bitácoras’ de trabajo. En esta categoría resultan particularmente importantes los ‘foros de discusión’ como medio de transferencia de I+C, mecanismos que se llevan a cabo de manera sistemática y constante al interior de las empresas.

### **7.4.3 Actividades de aprendizaje tecnológico**

#### ***Aprendizaje por reutilización***

En el primer apartado se mencionó que una de las principales actividades de aprendizaje utilizadas por algunas empresas de software como Constable y Manet es la ‘reutilización de módulos o partes de una herramienta o aplicación’. Como bien lo expresa el gerente de proyecto de la empresa Constable, “lo que implica la reutilización, además del ahorro de tiempo en la codificación, es el aprendizaje”.

La reutilización impacta directamente en las etapas de desarrollo y diseño. Puede haber cierto grado de retroalimentación en cada una de las etapas, sin embargo, no todos los desarrolladores son eficientes en el diseño y lo que hacen es entregar el programa en *etapas*, entrando en un ‘círculo vicioso de revisión-rediseño-reprogramación’. El líder de la empresa Manet expresa que:

“Cuando los módulos son muy pequeños el 80% de los módulos pasan ‘limpios’ y el 20% se rediseñan, esto es costeable. Pero cuando el diseño no es modular sino monolítico, en cada revisión se rediseña el producto completo y se reprograma completamente, lo cual eleva los costos, entonces se crea un círculo vicioso. Además, si el diseño es monolítico no es posible reutilizarlo, ya que si se entra en un círculo vicioso esto implica rediseñar todo y eleva más los costos.”

Por eso la empresa trabaja con módulos, ya que en términos de costo y tiempo es más rentable:

“Para un desarrollador independiente que hace una aplicación sólo por gusto, le es mucho mejor trabajar con módulos. El desarrollador profesional que prefiere hacer un sistema

monolítico difícilmente lo podrá terminar, aunque el desarrollador que lo hace por gusto tendrá más posibilidades de terminarlo, después lo liberará, ya que no tiene la presión del costo.”

La ventaja de esta actividad es que a lo largo del tiempo se incrementa la eficiencia de esa parte de la aplicación y se incrementa el conocimiento acerca del tipo de utilidad práctica de la misma:

“Las partes de una aplicación son más reutilizables cuando el proyecto es modular que cuando es monolítico. La reutilización es importante porque nuestros productos salieron de proyectos que hemos realizado y cuyos módulos pudimos integrar a un producto. Se trata de no tirar ni tiempo ni código a la basura, por eso es importante que sean reutilizables.”

### ***Aprendizaje por investigación y desarrollo***

Una actividad que no han hecho sistemáticamente los casos anteriores y que realiza de manera regular la empresa Renoir es la de investigación y desarrollo (I+D). A diferencia de los centros de I+D ubicados en empresas de manufactura, en las empresas de software un centro de I+D se denomina como:

“...Aquella área física que cuenta con computadoras, servidor(es), redes y personal dedicado a la búsqueda de nuevas aplicaciones de software y su utilidad práctica. La especificidad de esta actividad está determinada por *prueba y error*.”

En el proyecto de la empresa Renoir, por ejemplo, el gerente de proyecto menciona que parte de las actividades de I+D consisten en:

“Cómo producir aplicaciones más rápido, esto es, para desarrollar o programar líneas de código más rápido y de esta forma aumentar el ciclo de desarrollo de las aplicaciones. En este sentido, la investigación se orienta a cómo trabajar menos y hacerlo más rápido ya que ‘el mejor programador es el que programa menos, pues es el más eficiente’. Esto conlleva a hacer aplicaciones robustas y eficientes.”

Según los desarrolladores, un rasgo importante de los centros de I+D en la industria de software es que no se basan en la creación o desarrollo de nuevos lenguajes de



programación, “éstos ya están establecidos en el mercado y son bastante maduros”, sin embargo:

“Nuevas herramientas son creadas todos los días para cubrir nuevas necesidades que surgen de necesidades anteriores, o de las necesidades de los usuarios que son cada vez más específicas.”

Esta es una actividad que llevan a cabo pocas empresas, pero esta actividad ha resultado ser importante para la generación de nuevo conocimiento, como se mostrará en el capítulo 8.

### ***Aprendizaje por capacitación***

La capacitación es una de las estrategias más importantes para las empresas de software. En algunos casos está orientada a lenguajes informáticos, herramientas, bases de datos, y en otros a la gestión, administración, inglés, entre otras. Para algunos casos es externa, para otros es interna. Sin embargo, al igual que en casos anteriores, para las empresas Renoir, Manet y Goya, el temor es que el personal capacitado deje la empresa y se lleve el conocimiento adquirido.

En ocasiones, el conocimiento obtenido mediante la capacitación se traslada a un ‘marco de trabajo’. Como señala el líder de proyecto de la empresa Renoir:

“Estos marcos recogen una gran parte de la información y conocimiento que se generan a partir de la solución a problemas específicos, de lo aprendido en la interacción con los agentes del entorno, de la experiencia lograda en los proyectos y en la capacitación.”

Renoir ha estado trabajando con este marco desde que se creó la empresa, el cual es un programa hecho en software de fuente-abierta que administra la información y el conocimiento generado en la empresa. En palabras del líder de proyecto:

“Todas las aplicaciones que se hacen dependen de ese marco y esto es lo que da mucha más velocidad y robustez porque el marco se está actualizando y mejorando con cada aplicación.”

En este sentido, algunas aplicaciones que están relacionadas a ese marco de trabajo -cuando se hace alguna mejora a éste- se benefician de él:

“De una mejora realizada al marco de trabajo se benefician todas las aplicaciones sobre las que trabajamos y las cuales están relacionadas con ese marco.”

Como bien señala el líder de proyecto de la empresa Manet:

“Todos los sistemas hechos a la medida necesitan de algún tipo de capacitación, ya sea para el usuario, el cliente o el programador, ya que *nunca son tan intuitivos.*”

Por otro lado, si bien las empresas tienen actividades de capacitación interna y externa, estas actividades tienen un costo mayor ya que las empresas invierten en ello en un entorno donde la rotación de personal entre las empresas es alta. Como se mostró en el capítulo 6, Goya ha podido disminuir este costo a través del proyecto de vinculación con UNIVER, ya que la capacitación se genera al interior de la universidad y no en la empresa, reduciendo los costos y tiempos de la misma. Pero este es un mecanismo que requiere de la sinergia de otros agentes como las universidades.

#### ***Aprendizaje por actividades multitareas***

La movilización de recursos se refiere a ampliar las capacidades existentes al interior de la empresa. Una de las actividades más importante para lograrlo es el esquema de trabajo multitareas. Es una actividad usual en los proyectos realizados por Manet y Renoir, y en general entre micro y pequeñas empresas, como se ha mostrado en los apartados anteriores. El problema es que este esquema minimiza la capacidad de especialización en actividades específicas, por ejemplo, en pruebas de software (testing), compilación, entre otras.

La categoría *actividades de aprendizaje* se evalúa por el uso sistemático de la ‘reutilización de uno o más módulos’ de software en diferentes proyectos, por la ‘actividad de I+D’ (en particular para el caso de Renoir), por la ‘capacitación’ y por las ‘actividades multitareas’ entre los proyectos. Con estos elementos, las empresas tienen la capacidad para identificar e internalizar I+C de propósito general, pero también específicos a través de los usuarios, como se mostró en el capítulo anterior, lo cual es crucial para el desarrollo de nuevos productos, como se verá en el siguiente capítulo.

Manet, Renoir y Goya están asociadas a una interfase compleja. En este apartado se ha mostrado que han realizado significativos esfuerzos a nivel organizacional, lo que ha influido en el proceso de internalización de I+C específicos, y no sólo de propósito general, como sucede en una interfase simple. Las tres empresas tienen importantes fuentes de conocimiento previo: el personal cuenta con las habilidades básicas para el desarrollo de proyectos y, a excepción de Manet, han contratado nuevo personal con educación formal que ha sido capacitado interna o externamente. Las tres empresas han instalado sistemas de procesamiento de información al interior de la empresa, a partir del proceso de certificación de calidad (CMM), en particular Renoir y Goya. Esto implica contar con sistemas de procesamiento de conocimiento observable, y para ello se apoyan con bitácoras, además, logran la transferencia e internalización de conocimiento al interior de la empresa a través de los foros de discusión internos, lo cual ha sido crucial para incrementar su CA. Manet ha puesto énfasis en las actividades de aprendizaje, en particular en la reutilización de módulos y en las actividades multitareas y, como se mostró en el capítulo anterior, también ha sido importante para la empresa la interacción con las comunidades de desarrolladores, de las cuales ha podido absorber conocimiento que después ha explotado comercialmente. Para Renoir son importantes las actividades de aprendizaje por I+D, la capacitación y las actividades multitareas; en cambio, para Goya la capacitación ha sido de mayor relevancia.

Los resultados de la combinación de esos elementos han permitido a Manet el diseño y desarrollo de productos basados en software de fuente abierta, a Goya el diseño y desarrollo de nuevos productos y la formación de recursos humanos especializados (como se mostró en el capítulo anterior), y a Renoir le ha permitido diseñar y desarrollar productos específicos en un área de poca exploración para las empresas de software (programas para actividades forestales). Por tanto, con base en la evidencia presentada en este apartado, se infiere que estas empresas han incrementado significativamente su CA.

## **7.5 RESUMEN Y CONCLUSIONES DEL CAPÍTULO**

Con base en el capítulo 6 se puede recapitular que las interfases externas son un mecanismo crucial para que las empresas puedan identificar I+C del entorno, y en consecuencia

incrementar su CA. A lo largo de este capítulo se ha mostrado que las empresas internalizan I+C generados en el entorno a partir del conocimiento previo, de los flujos de información y conocimiento al interior de la empresa, y de las actividades de aprendizaje. Esto se resume en el cuadro 7.2.

**Cuadro 7.2**  
**Principales mecanismos para incrementar la Capacidad de Absorción**

Empresa	Conocimiento previo		Flujos de I+C al interior de la empresa		Actividades aprendizaje			
	Habilidades básicas	Contratación de nuevo personal	Procesamiento de conocimiento observable	Transferencia de I+C al interior de la empresa	Reutilización	I+D	Capacitación	Actividades multitareas
Constable	√	√	√		√		√	√
Velásquez	√		√				√	√
Degas	√	√					√	√
Manet	√			√	√			√
Renoir	√	√	√	√		√	√	√
Goya	√	√	√	√			√	

I+C: Información y conocimiento

Fuente: Elaboración propia con base en entrevistas.

Los puntos más importantes del resumen se mencionan a continuación:

- a) *Conocimiento previo*. Se muestra que el proceso de internalización de información y conocimiento tiene diversos mecanismos. Si bien todas las empresas cuentan con las habilidades básicas asociadas a los nichos de mercado de su especialidad, no todas contratan nuevo personal con el fin de aumentar su base de conocimiento, Velásquez y Manet no lo han hecho.
- b) *Flujos de conocimiento al interior de la empresa*. Se muestra que pocas empresas tienen sistemas de procesamiento de la información y del conocimiento observable. Aquellas que han desarrollado éstos se encuentran en un sistema de certificación de sus procesos de desarrollo de software (CMM en particular), pero las que no están en ese proceso no los han desarrollado, como es el caso de Degas y Manet. La transferencia de conocimiento entre las áreas ha resultado ser una actividad crucial, en particular los foros de discusión, ya que a través de éstos se comparte conocimiento y experiencias asociados al desarrollo de proyectos y a la capacitación externa. En particular, los foros

de discusión permiten, en gran medida, internalizar información y conocimiento generados en el entorno, sobre todo aquellos de mayor especificidad.

- c) *Actividades de aprendizaje*. La evidencia muestra que sólo Constable y Manet hacen uso frecuente de la reutilización de módulos de herramientas o aplicaciones de software como actividad de aprendizaje. Únicamente Renoir realiza actividades de I+D, aunque Velásquez menciona que la empresa es una especie de unidad de I+D por el tipo de proyectos que realiza (pruebas de software). Pero la mayoría de las empresas tienen como actividad de aprendizaje común la capacitación (a excepción de Manet) y las actividades multitareas al interior de la empresa (a excepción de Goya).
- d) Los flujos de conocimiento al interior de la empresa -en particular los foros de discusión- y las actividades de aprendizaje - como la capacitación- son actividades que en gran medida contribuyen a incrementar la CA de las empresas.

Por otro lado, como se mencionó en el capítulo 6, cada empresa está asociada a un tipo de interfase, y hay un efecto de causalidad en la CA. En el cuadro 7.3 se ilustra el tipo de interfase, el tipo de información y conocimiento que han logrado identificar e internalizar cada empresa, y su efecto en la CA.

Una interfase simple permite sólo un bajo incremento en la CA. El tipo de interfase es el origen de la causalidad, pero hay otros elementos que inciden en ello. Como se ilustra en el cuadro 7.2, la empresa Constable cuenta con mecanismos para internalizar I+C del entorno, pero se ha mostrado que aún contando con esos mecanismos sólo ha logrado un bajo incremento de su CA. En principio, esta limitación está asociada al tipo de proyecto que la empresa desarrolla. A lo largo de este capítulo se mostró que Constable ha desarrollado proyectos que satisfacen necesidades genéricas de los usuarios, para lo cual requiere I+C de propósito general, de forma que su internalización no representa un obstáculo para la empresa. Sin embargo, aunque tiene las capacidades para desarrollar proyectos de mayor valor agregado, incorporando I+C específicos, no los genera. Al parecer, uno de los problemas es que no cuenta con mecanismos de transferencia de I+C al interior de la empresa, en particular con los foros de discusión internos, lo que limita su absorción de I+C específicos.

La diferencia fundamental por la cual una interfase semi-compleja permite un mayor incremento de la CA es que, empresas como Velásquez y Degas, han podido identificar I+C de propósito general (en el entorno) y específicos (a través de la interacción con los usuarios), y también internalizar esa información y conocimiento específicos de los usuarios básicamente por la capacitación externa e interna.

A diferencia de las interfases anteriores, una interfase compleja permite un significativo incremento de la CA. En esta interfase, las empresas identifican I+C específicos, lo que implica una intensa interacción entre las empresas de software y diversos agentes del entorno. Los foros de discusión interno y la capacitación interna y externa son mecanismos importantes por los cuales absorben I+C específicos, es decir, por medio de los cuales incrementan su CA.

**Cuadro 7.3**  
**Interfase y nivel de Capacidad de Absorción**

<b>Empresa</b>	<b>Tipo de Interfase</b>	<b>Internalización (absorción) de información y conocimiento</b>	<b>Nivel de Capacidad de Absorción</b>
Constable	Simple	De propósito general	Bajo
Velásquez	Semi-compleja	De propósito general y específico	Moderado
Degas	Semi-compleja	De propósito general y específico	Moderado
Manet	Compleja	Específicos (y de propósito general*)	Significativo
Renoir	Compleja	Específicos (y de propósito general*)	Significativo
Goya	Compleja	Específicos (y de propósito general*)	Significativo

\*Si bien la empresa identifica *información y conocimiento específicos* a través de los diversos agentes y los internaliza, también pueden internalizar aquellos que son de propósito general, pero estos son de menor relevancia para los propósitos del tipo de proyecto que realizan.

Fuente: Cuadro 6.5 y cuadro 7.2.

Uno de los elementos que inciden en la limitación de las empresas para incrementar significativamente su CA es el esquema de actividades multitareas en la empresa. Este esquema asegura la acumulación de I+C en los desarrolladores, lo que incide en la internalización a nivel individual pero no a nivel organizacional. Esto influye en lo que las empresas logren internalizar, en buena medida, I+C de propósito general pero no específicos, lo cual se traduce en replica o escalamiento de productos ya existentes en el

entorno y limita la generación de nuevos productos, procesos y servicios, tal como se mostrará en el próximo capítulo.

## **CAPÍTULO OCHO**

### **CREACIÓN DE INFORMACIÓN Y CONOCIMIENTO A PARTIR DEL INCREMENTO EN LA CAPACIDAD DE ABSORCIÓN**

#### **INTRODUCCIÓN**

En el capítulo 6 se analizó la evidencia empírica acerca de cómo las empresas que desarrollan software hecho a la medida crean diferentes tipos de interfases externas. Se mostró que cada empresa está asociada a un tipo de interfase. En una interfase simple la empresa de software puede identificar en el entorno codificado gran parte de los requerimientos de información para el desarrollo de los proyectos, de forma que las necesidades de los usuarios y la interacción proveedor-usuario están estandarizadas. En una interfase semi-compleja, la empresa identifica una parte de la información en el entorno codificado, y otra parte a través de la interacción con los usuarios. El usuario tiene acumulada un importante fragmento de la información y conocimiento (I+C) necesarios para el desarrollo del proyecto, es pro-activo y coopera en la medida de sus capacidades. En una interfase compleja la empresa identifica la I+C básicamente a través de la interacción con los diversos agentes, pues no es suficiente la I+C existentes en el entorno.

En el capítulo 7 se argumentó que cada tipo de interfase tiene efectos en la capacidad de absorción (CA) de las empresas. Las empresas que diseñan interfases simples tienen limitaciones para incrementar su capacidad de absorción. Aun cuando tienen mecanismos por los cuales pueden internalizar I+C específicos, se mantienen en un esquema de absorción de I+C de propósito general. Por su parte, las empresas que diseñan interfases semi-complejas han logrado absorber I+C de propósito general y específicos a través de las actividades de aprendizaje, como la capacitación interna y externa. Mientras que las empresas que diseñan interfases complejas han logrado absorber I+C específicos, en particular, a través de los foros de discusión internos y la capacitación interna y externa.



Este capítulo tiene un doble propósito. Por un lado, se hace un análisis en conjunto de cómo, a partir de la construcción de diferentes interfases, las empresas mexicanas de software identifican I+C existentes en el entorno (como se analizó en el capítulo 6), y cómo, una vez que las empresas los identifican, los internalizan mediante diversos mecanismos de manera que incrementan en diferente grado su CA (como se analizó en el capítulo 7). Por otro lado, una vez que se explica cómo las empresas identifican e internalizan la I+C, en este capítulo se discute cómo las empresas transforman esa I+C en nuevas aplicaciones.

La justificación de la estructura de este capítulo se sustenta en la causalidad de los efectos observados. En un primer momento, las interfases permiten que las empresas identifiquen adecuadamente información y conocimiento existentes en el entorno y captados a través de las interfases. En un segundo momento, deben tener mecanismos que les permita internalizarlos, es decir, para incrementar su CA. En un tercer momento, si el incremento de la CA es *bajo* sólo podrán replicar o hacer adaptaciones menores a los productos ya existentes en el mercado, pero si el incremento de la CA es *moderado*, podrán hacer mejoras incrementales a productos ya existentes y diseñar y desarrollar nuevas aplicaciones, y si es *alto*, podrán crear recursos humanos especializados y también diseñar y desarrollar nuevas aplicaciones de software.

La estructura del capítulo es la siguiente. Después de esta introducción, en el primer apartado se analiza cómo a partir de una interfase simple las empresas identifican e internalizan I+C de propósito general existentes en el entorno, lo que incide en un bajo incremento de su CA. En este proceso, las empresas sólo son capaces de crear más información de propósito general y replicar o hacer adaptaciones menores a productos ya existentes.

En el segundo apartado se analiza cómo a partir de una interfase semi-compleja las empresas identifican e internaliza I+C de propósito general y específicos, lo que incide en un incremento moderado de su CA. En este proceso, las empresas pueden hacer mejoras incrementales a los productos de software ya existentes en el mercado, pero también desarrollar nuevos productos, hacer pruebas de software y capacitar a los usuarios.

En el tercer apartado se analiza cómo a partir de una interfase compleja, en la que predominan la vinculación universidad-empresa y la interacción entre empresas de software y comunidades de desarrolladores de software de fuente abierta (SWFA), las empresas de software identifican e internalizan I+C específicos a través de los usuarios activos, lo que incide en un alto incremento de su CA. En este proceso, las empresas pueden desarrollar nuevo conocimiento mediante la formación de recursos humanos con habilidades específicas de acuerdo a los requerimientos de un segmento de la industria de software y desarrollar nuevos productos. Finalmente, se presentan las conclusiones del capítulo.

## **8.1 INTERFASE *SIMPLE* Y BAJA CAPACIDAD DE ABSORCIÓN COMO DETERMINANTES DE LA ‘RÉPLICA DE INFORMACIÓN Y CONOCIMIENTO DE PROPÓSITO GENERAL’**

En este apartado se analiza cómo a partir de una interfase simple las empresas identifican e internalizan I+C de propósito general existentes en el entorno, lo que incide en un bajo incremento de su CA. Las fuentes y mecanismos por los cuales las empresas los internalizan son ‘el conocimiento previo asociado a las actividades particulares de la empresa’, los ‘flujos de información y conocimiento al interior de la empresa’ y las ‘actividades de aprendizaje’. En este proceso, las empresas sólo son capaces de crear I+C de propósito general y replicar o hacer adaptaciones menores a productos ya existentes.

### **8.1.1 Identificación de información y conocimiento a partir de la interfase simple**

En el capítulo 6 se mostró que las empresas de software desarrollan diferentes tipos de proyecto. Por ejemplo, el proyecto puede consistir en instalar en empresas usuarias una aplicación que ya existe en el mercado con pequeñas modificaciones, actualizar o escalar una aplicación ya instalada en la empresa usuaria, dar mantenimiento, o bien, se puede diseñar y desarrollar una nueva aplicación, hacerle pruebas, además de capacitar a los usuarios para el uso adecuado del sistema.

Con base en información presentada en el apartado 6.2, es posible señalar que en una interfase simple predominan los proyectos de baja complejidad. Estos proyectos consisten

en instalar en la empresa usuaria un programa o aplicación que ya existe en el mercado con pequeñas modificaciones de acuerdo a los requerimientos del usuario. Una gran parte del conocimiento para realizar las diversas actividades del proyecto está codificado y se ha convertido en información de propósito general. Es decir, es información y conocimiento acerca de productos que son de uso común entre las empresas usuarias, de forma que una empresa de software puede acceder a esa información y conocimiento a través de medios codificados. En este sentido, un entorno codificado proporciona una gran cantidad de la información necesaria para el desarrollo de aplicaciones o programas para procesos organizacionales y tecnológicos estandarizados y estables.

Las empresas que han establecido interfases simples como Constable, Manet, Renoir y Goya han declarado realizar programas o aplicaciones de ‘productos similares ya existentes en el mercado,’ es decir, son productos que ya están estandarizados y son tecnológicamente ‘maduros y estables’. Este tipo de productos son, básicamente, programas de nómina, contables y administrativos, los cuales son usados por empresas ubicadas en industrias tradicionales y no tradicionales.

### **8.1.2 Internalización de información y conocimiento a partir de la interfase simple**

Como se indicó en el punto anterior, en una interfase simple se identifica I+C de propósito general a través de medios codificados, como el Internet, en mucho menor medida a través del usuario.

Las empresas como Constable, Manet, Renoir y Goya han mostrado una amplia capacidad para identificar I+C de propósito general. Como se mostró en el capítulo 7, una forma de internalizarlos es a través de la contratación de profesionales que han sido formalmente educados en universidades o centros técnicos de capacitación. Estos profesionales cuentan con conocimiento previo para la realización de diversas actividades en la empresa, y este conocimiento es, al mismo tiempo, la base para internalizar la I+C existentes en el entorno.

Los profesionales de reciente contratación cuentan con conocimientos generales pero no específicos, sin embargo, para realizar las actividades de un proyecto que implica instalar

en la empresa usuaria un programa o aplicación que existente en el mercado con pequeñas modificaciones, no es necesario que el personal cuente con habilidades específicas para internalizar la I+C que previamente se identifican a través de la interfase simple. La razón es que las “pequeñas particularidades de los usuarios son relativamente fáciles de asimilar por los desarrolladores o ingenieros de software, éstos no requieren de una capacitación específica por medio de los cuales se pueda asimilar esas particularidades.”<sup>1</sup> En este sentido, basta con que la empresa cuente con conocimiento previo para que pueda internalizar la I+C de propósito general, pero esto incide sólo en un bajo incremento de su capacidad de absorción.

### **8.1.3 Creación de información y conocimiento en un entorno de interfases simples y baja capacidad de absorción**

Los procesos de identificación e internalización de I+C descritos en los capítulos 6 y 7, tienen un efecto causal en el tipo de aplicaciones que las empresas desarrollan.

Las empresas como Constable, Goya, y Renoir han optado por instalar aplicaciones de software que ya existen en el mercado con pequeñas modificaciones o adaptaciones de acuerdo a las necesidades del usuario. Como se mostró en el capítulo 6, las empresas de software siempre hacen algún tipo de modificación al momento de instalar un sistema en la empresa usuaria, pero no hacen el desarrollo completo. Por lo cual utilizan información muy general acerca de una aplicación que ya existe en el mercado:

“...Información acerca de los componentes o módulos<sup>2</sup> del sistema, acerca del lenguaje con el que fue escrito o de la forma en que fue compilado, entre otros. Esta información es general en la industria de software. Pero hay otro tipo de información que es particular y se conoce cuando la empresa de software establece relación con la empresa usuaria para conocer los detalles de sus requerimientos y necesidades informáticas.”<sup>3</sup>

---

<sup>1</sup> Entrevista con líder de proyecto, empresa Constable.

<sup>2</sup> Recientemente, los desarrolladores en ciencias de la computación y lenguajes de programación (en particular en el desarrollo de programación orientada a objetos) han fomentado la modularización y puede facilitar los proyectos futuros basados en tecnologías de software de fuente abierta. Entrevista con líder de proyecto, empresa Manet.

<sup>3</sup> Entrevista con líder de proyecto, empresa Renoir.

Cuando el proyecto consiste sólo en la compra-venta de una aplicación de software con pequeñas modificaciones, el resultado de esta interfase simple es más información de propósito general. Las razones son las siguientes. Primero, la relación no involucra al usuario más que para determinar cuáles son las necesidades y requerimientos de informática, si el proyecto sólo consiste en instalar un programa que ya existe en el mercado con pequeñas modificaciones, la relación se reduce a una transacción de compra-venta. Segundo, en una interfase simple se simplifica el tipo de información utilizada en el proyecto. Las empresas de software saben las especificaciones del producto que ya existe en el mercado y, por tanto, pueden discriminar los grandes montos de información contenida en esa aplicación. De esta forma se simplifica la información, al igual que la relación con el usuario. Tercero, de esta relación “lo que se aprende es poco comparado con la información que se vuelve abundante y repetitiva.”<sup>4</sup> Cuarto, la información utilizada es difundida, pues se encuentra en el mercado a través de las aplicaciones ya existentes. El conocimiento utilizado es de propósito general porque las aplicaciones han sido ampliamente difundidas, de forma que el resultado de un proyecto simple, como el que se desarrolla en una interfase simple, es nuevamente I+C de propósito general.

Por último, algunas empresas sólo son capaces de crear más información de propósito general y replicar o hacer adaptaciones menores a productos ya existentes. Como se discutió en el capítulo 7, Constable se mantiene en un esquema de replicar, con pequeñas modificaciones, productos ya existentes. Si bien esto le genera una rentabilidad económica sólida, pierde oportunidades tecnológicas para desarrollar proyectos de mayor valor agregado pues cuenta con mecanismos importantes por los cuales podría identificar I+C específicos y migrar a un tipo de interfase más complejo. El círculo en el que se encuentra esta empresa se torna vicioso en la medida en que la empresa se mantiene en una interfase simple y no expande sus expectativas tecnológicas, aunque puede ser ventajoso en el corto plazo cuando el nicho de mercado en el cual se ubica es rentable económicamente.

---

<sup>4</sup> *Ídem.*

## **8.2 INTERFASE SEMI-COMPLEJA Y MODERADA CAPACIDAD DE ABSORCIÓN COMO DETERMINANTES DE LA ‘MEJORA INCREMENTAL Y DESARROLLO DE NUEVOS PRODUCTOS DE SOFTWARE’**

En este apartado se explica cómo a partir de una interfase semi-compleja las empresas identifican información y conocimiento de propósito general a través de medios codificados, pero también información y conocimiento específicos a través de la interacción con los usuarios. Los resultados que logran las empresas a partir de la identificación e internalización de I+C son mejoras incrementales y el desarrollo de nuevos productos de software.

### **8.2.1 Identificación de información y conocimiento a partir de la interfase semi-compleja**

Las empresas que han establecido interfases semi-complejas como Velásquez y Degas identifican información y conocimiento de propósito general a través de diferentes medios codificados, pero también específicos a través del usuario. El líder de proyecto de la empresa Degas menciona que:

“El tipo de producto que desarrolla Degas es para empresas que se ubican en el sector de tenería, son empresas que tienen procesos productivos estandarizados. Dado que Degas cuenta con ingenieros que conocen esos procesos [varios de ellos trabajaron en empresas de tenería], la empresa conoce en gran medida las necesidades de las empresas mexicanas de tenería...Pero hay particularidades que no se pueden conocer del todo, se requiere del usuario. Aun siendo usuarios similares [empresas de la misma industria], hay especificidades que Degas no conoce ya que cada usuario tiene necesidades y requerimientos específicos.”

Si bien las empresas pueden identificar I+C necesarios para el desarrollo del proyecto, el entorno codificado no es el único ni el más importante para ese propósito, se requiere del usuario. El gerente de proyecto de la empresa Velásquez menciona que:

“Las revistas y libros especializados son una fuente importante para acceder al conocimiento que se requiere para ciertos procesos del proyecto, pero en el segmento de software hecho a la medida, sobre todo cuando el programa o el sistema es específico a cada

empresa usuaria, el contacto con el usuario es imprescindible en las primeras etapas del proyecto. Suponiendo que la empresa usuaria pertenece a un mismo sector [como en el caso de los clientes de Degas] y que tiene las mismas características productivas y organizacionales que el resto, la empresa proveedora podría usar su experiencia para desarrollar un proyecto que un nuevo cliente requiere sin mucha interacción con éste, pero dar por hecho que la empresa requerirá lo mismo que otra es un riesgo muy alto, aún tratándose de un proyecto muy similar.”

Como se muestra en las citas anteriores, las empresas de software reconocen que deben interactuar con el usuario para acceder a información y conocimiento acerca de sus necesidades y requerimientos, ya que no todas las necesidades del usuario se encuentran expresadas en el entorno codificado. Si bien esos usuarios pueden tener procesos productivos, administrativos y organizacionales estandarizados, cada empresa tiene un grado de especificidad.

Los proyectos de software desarrollados por estas empresas han consistido en el desarrollo de programas o aplicaciones para procesos de manufactura, o sistemas de software que interactúan al mismo tiempo con los procesos productivos, de distribución, administración y ventas, y no sólo aplicaciones o programas contables, punto de venta, o una simple relación de compra-venta.

La I+C que las empresas de software pueden identificar a partir de una interfase semi-compleja tiene un alto grado de especificidad que no es usual en una interfase simple. La razón principal es que, al contrario de lo descrito en la interfase simple, en la interfase semi-compleja el usuario toma un rol activo en la configuración de los productos o aplicaciones, como se mostró en el capítulo 6, y sus necesidades y requerimientos son de mayor especificidad.

Recordemos que, en el caso de Degas, los fundadores de la empresa adquirieron conocimiento específico acerca de los procesos de tenería en empresas del mismo giro, es decir, anteriormente trabajaron en éstas desarrollando actividades de ingeniería relacionadas con el curtido de piel. Para estos ingenieros, una de las condiciones principales

en la adquisición de conocimiento específico es que los profesionales se hayan involucrado en el desarrollo del proceso de curtido de piel. Es decir, los ingenieros que han trabajado en empresas de tenería tienen el conocimiento específico que una empresa como Degas requiere, porque conocen el proceso de lavado, de curtido, etcétera. Dado que el principal producto de Degas es una aplicación para este tipo de empresas, el conocimiento específico se deriva de las actividades relacionadas con toda la cadena del proceso de curtido de piel. Este conocimiento no es nuevo para la industria de tenería, de hecho, se ha estandarizado en esta industria, pero sí lo es para la industria de software. Esto ha facilitado a Degas identificar e internalizar la I+C necesarios. Incluso se afirma con la siguiente cita hecha por el líder de proyecto de esta empresa:

“Tenemos información acerca de empresas extranjeras que han querido incursionar en el mercado [específicamente en la industria de tenería de Guanajuato] pero no han podido porque el personal de aquellas empresas no cuentan con la experiencia que nuestros ingenieros han acumulado a lo largo de su trayectoria al interior de las propias empresas de tenería, y ahora como parte de una empresa de software se involucran en procesos de desarrollo de software.”

Como se ha mostrado, la información que las empresas de software pueden identificar en una interfase semi-compleja es específica y no sólo de propósito general.

### **8.2.2 Internalización de información y conocimiento a partir de la interfase semi-compleja**

Las empresas como Velásquez y Degas han diseñado una interfase semi-compleja, en la cual identifican I+C específicos y los internalizan, particularmente, a través de la capacitación interna y externa.

Con el propósito de internalizar I+C, para Velásquez ha sido importante el proceso de certificación CMM, ya que para alcanzar el nivel 3 requirió documentar y sistematizar una gran parte de sus procesos internos. Esta es una forma de sistematizar al interior de la empresa la I+C que se identifica a través de las interfases, es decir, permite sistematizar conocimiento observable, pero se requiere transferirlos al interior de la empresa. Un



mecanismo importante para lograrlo son los ‘foros de discusión internos’. En el capítulo 7 se mostró que la experiencia no siempre es codificable, pero se puede difundir internamente a través de los foros de discusión, sin embargo, estos no han sido implementados sistemáticamente en Velásquez y Degas.

Otros mecanismos por los cuales Velásquez y Degas han podido internalizar la información y conocimiento específicos son las actividades de aprendizaje. Velásquez está tratando de instalar un laboratorio de investigación y desarrollo (I+D). Como se mencionó en el capítulo 7, el aprendizaje por I+D está basado en actividades de ‘prueba y error’, y a través de estas actividades la empresa aprende a utilizar diversas técnicas asociadas a la prueba de software. Ambas empresas cuentan con capacitación interna y externa, las cuales son sus actividades de aprendizaje más importantes y por medio de las cuales internalizan información y conocimiento específicos.

En consecuencia, a pesar de que las empresas Velásquez y Degas tienen fuentes de conocimiento previo e importantes actividades de aprendizaje como la capacitación interna y externa, como se mostró en el capítulo 7, pueden identificar información y conocimiento específicos, pero tienen limitaciones para transferirlos al interior de la empresa, lo que incide en un moderado incremento de su capacidad de absorción.

### **8.2.3 Creación de información y conocimiento en un entorno de interfases semi-complejas y *moderada* capacidad de absorción**

Una interfase semi-compleja se caracteriza por la interacción proveedor-usuario con un nivel de intensidad tal que ambas empresas obtienen algún tipo de aprendizaje. Lundvall (1985, 1988 y 1992) menciona que el concepto proveedor-usuario es importante por la interacción de sus aprendizajes, pero hay diversas consideraciones para lograr esta interacción.

Los procesos descritos en los puntos anteriores tienen un efecto en el tipo de I+C que las empresas usan para sus fines comerciales. En este proceso, las empresas pueden hacer

mejoras incrementales a los productos de software ya existentes en el mercado, pero también desarrollar nuevos productos, hacer pruebas de software y capacitar a los usuarios.

En una interacción proveedor-usuario con proyectos de ‘escalamiento, actualización, o prueba de software’, la empresa que tiene la habilidad para identificar información y conocimiento específicos, que cuenta con mecanismos para transferirlos al interior de la empresa y con ciertas actividades internas de aprendizaje, tiene la capacidad para hacer mejoras incrementales y desarrollar nuevos programas de software y no sólo replicarlos (como sucede a partir de una interfase simple).

Por ejemplo, para el caso de Degas, después de haber identificado e internalizado información y conocimiento de propósito general y específicos, ha podido generar productos para empresas de tenería ubicadas en otros países. Si bien la inserción a mercados como el español o el italiano aún es baja, la empresa tiene como estrategia posicionarse en el mercado internacional:

“Si bien nuestro mercado se ha ubicado en México y Brasil, nuestros productos han sido reconocidos en mercados como el español o el italiano. Tenemos un cliente en España y a partir de 2007 pretendemos incorporarnos al mercado italiano. Aún siendo una industria con procesos estandarizados y en la que predominan las pequeñas y medianas empresas, las empresas de tenería en cada país tiene sus particularidades...Pero posicionarnos en otros mercados sólo ha sido posible por el aprendizaje que nos ha dado el trabajar en empresas de tenería [como ingeniero de procesos de tenería].”<sup>5</sup>

Además, Degas ha podido migrar de una plataforma de software de fuente abierta a una propietaria:

“Hoy tenemos el producto principal en plataforma JAVA, pero ‘mañana’ lo tendremos en ORACLE. Trabajar en software de fuente abierta tiene sus ventajas y sus desventajas. Una de las ventajas es que podemos interactuar con comunidades de desarrolladores y aprender de ellos y ellos de nosotros, es una comunidad de aprendizaje constante, la desventaja es que no ‘hay quien se haga responsable’, por ejemplo, respecto de las bases de datos (aunque son más livianas) nosotros mismos tenemos que actualizarlas y darles soporte. A diferencia

---

<sup>5</sup> Entrevista con el gerente de proyecto, empresa Degas.

de ORACLE, que da soporte y actualización de las bases de datos, la desventaja de trabajar con ORACLE es el precio, ya que por usuario/año cobra alrededor de 3,500 dólares. Sabemos que tenemos la capacidad para migrar de una plataforma tecnológica a otra, podemos elegir una u otra estrategia.”<sup>6</sup>

Esto nos muestra que la empresa Degas ha podido hacer mejoras incrementales a su producto principal, utilizado por empresas de tenería, y genera conocimiento específico para la industria. Esta es una de las diferencias respecto de empresas que se involucran en proyectos de menor valor agregado.

Es importante mencionar que el usuario juega un rol importante en el desarrollo de la industria de software. Como menciona el gerente de proyectos de la empresa Degas:

“A inicios de 2002 LEFARC [la empresa usuaria] inició un programa de competitividad donde se buscaba fortalecer la empresa en muchas áreas y se dieron cuenta que faltaba algo para hacerla más eficiente. A través del programa de competitividad definieron que era necesario fortalecer las áreas de servicios al cliente, de producción, de calidad, etcétera, y tenían que tomar decisiones rápidas y eficientes. Este fue uno de los principales elementos que incidieron en la Implementación del Sistema Informático (ISI), y por el cual nos contrataron.”

En un contexto de conversión tecnológica los usuarios juegan un rol importante, ya que algunos segmentos de la industria de software dependerán de la velocidad con que las empresas usuarias, ubicadas en industrias maduras, empiecen a migrar a una versión de modernidad en sus procesos productivos y organizacionales, y a la incorporación de software en sus procesos, lo cual, como se muestra en las citas anteriores, tiene efectos de gran relevancia para la industria mexicana de software. Lo que se ha mostrado es que el software ha impactado y revitalizado empresas ubicadas en sectores tradicionales, en este caso a empresas ubicadas en la industria de tenería.

---

<sup>6</sup> *Ídem.*

### **8.3 INTERFASE COMPLEJA Y SIGNIFICATIVA CAPACIDAD DE ABSORCIÓN COMO DETERMINANTES DE LA ‘CREACIÓN DE HABILIDADES Y CONOCIMIENTO ESPECÍFICOS’**

En este apartado se analiza cómo a partir de una interfase compleja, en la que predominan la vinculación universidad-empresa y la interacción entre empresas de software y comunidades de desarrolladores de SWFA, las empresas identifican I+C específico a través de los usuarios, los internalizan y los aprovechan para crear recursos humanos especializados, así como nuevas aplicaciones de software.

#### **8.3.1 Identificación de información y conocimiento a partir de la interfase compleja**

Las empresas que han establecido interfases complejas como Manet, Renoir y Goya han identificado información y conocimiento a través de los usuarios activos y ‘conscientes’ de las redes virtuales y de las universidades. En la medida en que los proyectos implican una mayor complejidad de desarrollo, los agentes tienen una mayor participación en los mismos. Como bien señala el gerente de proyectos de la empresa Velásquez, en una interfase compleja puede haber información específica:

“En una interfase compleja se administra información más específica, pero montos menores de información, es decir, los flujos de información son cortos pero complejos. A diferencia de las interfases simples [y semi-complejas] donde gran parte de [la información es de propósito general y] los flujos de información son largos pero simples.”

Como mostró en el capítulo 6, los usuarios activos que configuran una interfase compleja tienen información y conocimiento más específicos acerca de sus necesidades y requerimientos, así como habilidades en el uso de herramientas y aplicaciones que usa la empresa de software.

Para el caso de Manet y Renoir, las fuentes importantes de I+C son los usuarios, pero también las comunidades de desarrollo de SWFA. Según el líder de proyecto de la empresa Manet:

“Quienes generan el conocimiento son los ‘usuarios conscientes’ ubicados en las comunidades de desarrolladores de software de fuente abierta, son estos quienes tienen el

conocimiento de cómo instalar una aplicación, cómo se usa y qué soluciones otorga, y no necesariamente los desarrolladores. Cada línea de código útil que se programa es generación de conocimiento, cada línea de documentación que se escribe también. Las empresas que participan en las comunidades pueden identificar hacia dónde se están moviendo ciertas tecnologías que después podemos utilizar para el desarrollo de aplicaciones comerciales.”

En el caso de Goya, una gran parte de la información y conocimiento específicos los identifica a través de los usuarios, también las áreas de marketing o de servicios al cliente son una importante fuente de información interna, a través de las cuales identifican el comportamiento de las necesidades y requerimientos de los usuarios. Pero, para esta empresa, la mejor forma de identificar y obtener conocimiento específico es a través de la interacción con usuarios y universidades:

“Una de las formas de obtener conocimiento específico es a través de la interacción que se da entre el personal especializado de Goya y personal especializado de la empresa usuaria... Pero también la universidad es una importante fuente de conocimiento, aunque no siempre coincide con las necesidades de la industria. Por eso el proyecto de vinculación que hemos llevado a cabo con UNIVER ha sido importante para la generación de recursos humanos especializados.”<sup>7</sup>

### **8.3.2 Internalización de información y conocimiento a partir de la interfase compleja**

Como se señaló en el punto anterior, en una interfase compleja las empresas identifican información y conocimiento específico a través de los usuarios activos y conscientes, a través de la interacción entre empresas y comunidades de desarrolladores de SWFA, y mediante la vinculación universidad-empresa. Es importante la forma en cómo internalizan y transfieren al interior de la empresa la información y conocimiento específicos (tal como se mostró en el capítulo 7). Para este propósito resultan importantes los foros de discusión y la capacitación:

“El personal adquiere conocimiento a través de la interacción interna y de otra fuente de conocimiento: de un plan de capacitación formal externa e interna, así como de los foros de

---

<sup>7</sup> Entrevista con líder de proyecto, empresa Goya.

discusión interna...A partir de los cuales se obtienen soluciones a problemas específicos. Cuando se encuentra la solución a algún problema se difunde el conocimiento al interior de la empresa.”<sup>8</sup>

Aunque algunas empresas como Goya mencionan que:

“Los resultados de las interacciones que hacemos con el cliente o con otro agente lo tratamos de incorporar a la empresa, pero no hay un mecanismo formal.”<sup>9</sup>

En el capítulo 7 también se mencionó que en algunas empresas como Renoir, el conocimiento adquirido a través de los diferentes proyectos y actividades se salvaguarda a través del monitoreo y control de las actividades de los proyectos:

“Nos aseguramos de que se llenen las bitácoras a diario, se generen respaldos y se realice toda la parte correspondiente a la configuración...La parte del conocimiento adquirido se queda en bitácoras y en código.”

Pero sabemos que una gran parte del conocimiento no se puede codificar fácilmente, y por tanto no puede llevarse a bitácoras. Este es un problema para las empresas de software en el cual algunos líderes y gerentes de proyecto coinciden:

“Uno de los problemas es cómo plasmar en papel el conocimiento específico creado en cada desarrollador, el cual origina la innovación.”<sup>10</sup>

Un punto que dificulta plasmar y modificar este conocimiento es el exceso de documentación derivada de los proyectos, ello también es un problema para muchas empresas que desarrollan software:

“La documentación de un proyecto de desarrollo de software es muy importante, sin embargo en el desarrollo de software es muy fácil que esa documentación se vuelva demasiada, mas de la necesaria, o poca e insuficiente; por otro lado, es extremadamente difícil mantener la documentación consistente con lo que se tiene, esto es, actualizada.”<sup>11</sup>

---

<sup>8</sup> *Ídem.*

<sup>9</sup> *Ídem.*

<sup>10</sup> Entrevista con líder de proyecto, empresa Renoir.

<sup>11</sup> Entrevista con líder de proyecto, empresa Goya.

La evidencia muestra que el conocimiento específico se puede identificar a través de la interacción con los usuarios y se internaliza a través de los sistemas de procesamiento de conocimiento observable (bitácoras), de los foros de discusión interna, y de la capacitación interna, tal como se mostró en el capítulo 7. Este conocimiento es el principal insumo que permite a las empresas de software generar nuevos productos en el segmento de software hecho a la medida.

Con el uso del conocimiento de propósito general, las empresas pueden mantenerse en su nicho de mercado (como es el caso de la empresa Constable), pero crean un círculo de dependencia que las mantiene en la réplica de ciertos productos que, dada la rápida evolución de la industria, se vuelven de bajo valor agregado, y crean un círculo vicioso que no les permite incorporar mejoras incrementales en función de las necesidades de los usuarios.

La identificación e internalización de I+C específicos permite a las empresas crear productos con mayor valor agregado, creando un círculo virtuoso de mejoras incrementales ya que, ante la dinámica de la industria, los usuarios demandarán productos de mayor especificidad ajustados a sus necesidades.

### **8.3.3 Creación de información y conocimiento en un entorno de interfases complejas y *significativa* capacidad de absorción**

En esta sección muestra que, a partir de las interfases complejas, las empresas pueden identificar e internalizar I+C específicos para el desarrollo de nuevo productos, así como la formación de recursos humanos con habilidades específicas de acuerdo a los necesidades de un segmento de la industria de software. En los siguientes puntos se ilustra el tipo de I+C que se generan a partir de una interfase compleja y un significativo incremento en la CA.

#### ***Interacción proveedor-usuario***

Manet realizó un proyecto de escalamiento, actualización y mantenimiento de aplicaciones de software para una pequeña empresa:

“La aplicación había sido desarrollada e instalada por otra empresa, pero estaba hecha ‘con poca funcionalidad’, de forma que la empresa usuaria contrató a Manet para modificar y escalar esa aplicación. Este tipo de aplicaciones la hemos desarrollado para diferentes empresas...El conocimiento que logramos es genérico en cuanto al uso de la tecnología de software (herramientas, aplicaciones, entre otros), pero también específico porque cada cliente tiene necesidades particulares, aún siendo proyectos muy similares.”

En el proyecto particular realizado por Manet, la empresa usuaria no requería por completo el diseño del portal, sino sólo una parte del mismo ya que el anterior no cumplía con la funcionalidad requerida. El proyecto duró 8 meses y el líder de proyecto de esta empresa mencionó que:

“La interacción con el cliente fue poco frecuente pero muy específica, sabían exactamente lo que requería y nosotros entendimos sus necesidades rápidamente. En algunas ocasiones los líderes de proyecto de la empresa de software han tenido que hacer el papel de líder al interior de la empresa usuaria en alguna de las fases del desarrollo del sistema. Esto dio como resultado un producto factible, con funcionalidad, confiable y no requerirá rehacerlo.”

Este proyecto muestra que la información fue específica y la empresa fue capaz de asimilar esas necesidades rápidamente. El resultado fue óptimo tanto para la empresa usuaria como el proveedor. Cabe resaltar que, como se señala en la cita anterior, la empresa de software asignó un líder de proyecto al interior de la empresa usuaria para solucionar los problemas derivados del escalamiento y actualización del sistema, lo que implicó una mayor participación del personal de la empresa usuaria en el proyecto.

Bajo este tipo de proyectos, las empresas establecen una interacción proveedor-usuario caracterizada por un alto contenido de información específica (acerca del proyecto ya diseñado e instalado por otra empresa de software) y conocimiento general (logrado por la realización de proyectos similares), con resultados distintos, como menciona el líder de proyecto de la empresa Manet:

“Se adquirió información específica acerca de *nuevos problemas* relacionados con el *desarrollo de portales*, así como conocimiento general y codificado acerca de la solución a problemas relacionados con portales desarrollados por otras empresas.”



### ***Vinculación universidad-empresa***

En el capítulo 6 se explicó el proyecto de vinculación entre Goya y la universidad UNIVER. El objetivo del proyecto fue formar recursos humanos especializados en ciertas tecnologías de software que la empresa y una parte de la industria requieren.

Como se explicó en ese capítulo, al proyecto de vinculación se adhirieron otras dos empresas, IBM e INEVO, quienes proveyeron las herramientas de software. El primer resultado del proyecto de vinculación fue la fábrica de software, la cual es llamada Centro de Excelencia para la Ingeniería de Software (CEIS) ubicada al interior de UNIVER. En una entrevista con el líder del proyecto de la empresa Goya, se menciona que:

“El CEIS provee los planes de especialización e instructores, de tal forma que el estudiante sale con un alto grado de especialización en tecnologías que la empresa requiere. En estos CEIS los estudiantes seleccionados se especializan en tecnologías de software, algunos de ellos se encuentran trabajando y practicando con plataformas JAVA, SQL, DBIL, IBM, entre otras. En este proyecto se han incorporado materias como lógica de programación, metrología, calidad para el desarrollo, etcétera, de forma que el estudiante que egrese bajo este esquema será bastante competitivo, y si no se queda con nosotros pues ya no perdimos porque nos recuperamos en el proyecto...Respecto a la competencia, esos estudiantes nos van a dar mayor valor agregado.”

Como se muestra en la cita anterior, los recursos humanos especializados que requiere Goya y un segmento de la industria de software son uno de los resultados más importantes del proyecto de vinculación, los cuales se obtienen mediante la combinación de educación formal y la capacitación (antes otorgada al interior de Goya, ahora otorgada en los CEIS que se ubican en UNIVER). Pero hay otro resultado también importante:

“Al mismo tiempo, con cada socio tecnológico que se acerca al proyecto de vinculación se desarrollan Centros Especializados en el Desarrollo de Software (CEDES), el cual es una especie de célula productiva con especialización en ORACLE, MICROSOFT, TESTING, PUNTO-NET, entre otros. Cada CEDES especializado le podrá dar servicios a otras empresas además de Goya. A lo largo del tiempo estos CEDES pueden convertirse en empresas independientes, ahora son una especie de incubadora de empresas...Este es el objetivo de los CEDES a mediano plazo.”

Como se muestra, los resultados del proyecto de vinculación son la creación de una fábrica de software, una incubadora de empresas al interior de UNIVER, y recursos humanos especializados en las tecnologías centrales para Goya y otras empresas del sector. Para Goya, uno de los segmentos de mercado importantes es el desarrollo de aplicaciones o programas de software, y la calidad (además del costo) de esos productos depende de la formación de recursos humanos especializados. Este modelo de vinculación ‘es una innovación en la industria’.

### ***Comunidades de desarrolladores de software***

Las empresas como Manet y Renoir interactúan con comunidades de desarrolladores de software de fuente abierta. Según el líder de proyecto de la empresa Manet, la estructura de trabajo de las comunidades es como se describe a continuación:

“Una comunidad de desarrolladores de software de fuente abierta se compone de dos agentes principales. El primero se denomina desarrollador y es el encargado de crear, desarrollar y escalar algoritmos o aplicaciones. El segundo se denomina usuario ‘consciente’, y es quien tiene conocimiento específico de los algoritmos, aplicaciones o herramientas sobre las cuales los desarrolladores están trabajando. En una comunidad pueden haber 3 desarrolladores y 200 usuarios conscientes asociados a una misma lista de correo.”

Siguiendo la cita anterior, estos agentes trabajan en la comunidad de la siguiente manera:

“Los desarrolladores son quienes suben a una red (a través del correo electrónico) algún proyecto que les interesa desarrollar o escalar, conocen el funcionamiento del sistema (de uno o más módulos), hacen las mejoras subsecuentes y soportan todo el desarrollo. A través del correo electrónico más de un desarrollador establece comunicación con otro desarrollador interesado en líneas de código o la aplicación. Estos desarrolladores se especializan en ciertos algoritmos, aplicaciones, etcétera, y trabajan con lenguajes y herramientas de programación específicos. A través de la interacción por correo electrónico fluye el conocimiento. Quienes generan el conocimiento son los 200 usuarios, ellos tienen el conocimiento de cómo se instala, cómo se usa, cómo resolvieron un problema determinado con la instalación, conocen las fallas esperadas, saben qué deben reportar a los desarrolladores y lo que no les interesa reparar del sistema [ya sea porque creen que no es

útil o simplemente porque no les interesa]; mientras que los 3 desarrolladores conocen cómo funciona el sistema por dentro.”

La interacción entre los desarrolladores y los usuarios es de extrema importancia:

“Mientras que los primeros saben hacia dónde se dirige el escalamiento o el desarrollo de las herramientas o aplicaciones, los segundos saben si ese algoritmo tendrá alguna utilidad en el mercado, si el desarrollo es adecuado, si se puede mejorar o en qué medida se puede escalar o modularizar<sup>12</sup>. En la comunidad se puede intercambiar opiniones, criterios de desarrollo, módulos de un sistema, aplicaciones, entre otros.”<sup>13</sup>

En ocasiones, los usuarios conscientes tienen una relación estrecha con las necesidades del mercado o con usuarios finales. Los desarrolladores, a través de los usuarios conscientes, establecen una conexión con las necesidades del mercado o de los usuarios finales. Pero, como menciona el líder de proyecto de la empresa Manet, no todo lo que se genera en las comunidades tiene aplicaciones comerciales:

“En algunas comunidades de desarrollo algunos miembros de la comunidad tienen una actividad comercial y requieren de madurar o escalar alguna herramienta, aplicación, módulo, etcétera, pero en muchas ocasiones a los desarrolladores o usuarios conscientes no les interesa reparar o escalar aunque les pagues. Ese algoritmo puede tener aplicaciones comerciales, ya que un usuario identificó su potencialidad comercial, pero a veces a la comunidad no le interesa.”

En el contexto de fuente abierta hay muchas comunidades asociadas a una lista de correo, así por ejemplo podemos encontrar la comunidad de desarrolladores de PHP (una de las herramientas más importantes en el contexto de SWFA), de Apache (que se especializa por trabajar con base en módulos pequeños<sup>14</sup>), entre otras.<sup>15</sup>

---

<sup>12</sup> El término de modularización se refiere a que el sistema (aplicación o programa) puede ser dividido en pequeños módulos, se trabajan por separado (división de tareas) y al final pueden ser integrados independientemente. De esta forma, al momento de la implantación, si la aplicación falla o el usuario requiere hacer modificaciones de último momento, sólo se re-diseña y re-programa el módulo correspondiente y no todo el sistema.

<sup>13</sup> Entrevista con líder de proyecto, empresa Manet.

<sup>14</sup> Von Hippel (1998, 2000, 2005) ha elaborado trabajos empíricos sobre el rol que el usuario tiene en los procesos de innovación y sobre todo la forma en que trabajan en una comunidad de desarrolladores de software basados en fuente abierta. La comunidad Apache ha sido ampliamente explorada por este autor.

Algunos líderes de proyecto, en particular de las empresas Manet, Renoir y Degas, mencionan que una comunidad de desarrolladores de SWFA:

“Puede ser parte de una estrategia de conveniencia para los desarrolladores independientes o para las mismas empresas. Por una parte, porque no pagan por el desarrollo o escalamiento de los algoritmos que usan para fines comerciales, y por otra, porque ser parte de una comunidad permite aprender de otros sin costo económico. El problema es que no hay forma de conocer todo lo que se está generando al interior de las comunidades sino hasta el momento en que se libera alguna parte de la aplicación (algún módulo), o hasta que una parte de la aplicación se utiliza con fines comerciales.”

Un punto adicional a considerar es la estructura de gobernanza de los proyectos de fuente-abierta, ya que son diferentes a los de software propietario:

“En un contexto de software propietario existe una fuerte centralización de autoridad de los que desarrollan el código inicial. En otros casos, como en la comunidad Apache, los problemas se resuelven a través de un comité por votación y consenso, y la toma de decisiones es un proceso relativamente más transparente. Los líderes de una comunidad no necesariamente son los que desarrollan el código inicial, pero pueden ser quienes hagan una contribución importante en las etapas iniciales de un proyecto. Además, esos líderes no tienen una ‘autoridad formal’, es una autoridad moral que van ganando en la medida en que hacen contribuciones sustanciales a un proyecto, desde recomendaciones y señalización de criterios, hasta la escritura de nuevos códigos. Sin embargo, la naturaleza casi anárquica de las comunidades puede generar conflictos de intereses respecto a cómo evoluciona la tecnología, hacia dónde dirigirla, en qué momento parar, en qué momento liberar un módulo [hacerlo propietario o comercial], etcétera.”

A partir de la evidencia se puede deducir que hay una gran cantidad de conocimiento que se está generando en las comunidades de desarrollo de SWFA, de una forma no controlada pero sí muy significativa, aunque muchas de las aplicaciones que desarrollan o escalan no están incentivadas directamente por el mercado, sino por una conducta ‘ideológica’ de desarrollar en forma ‘libre’ lo que es propietario. No hay mecanismos claros de

---

<sup>15</sup> Los programadores estudian el código fuente, publican sus descubrimientos (agujeros de seguridad, vulnerabilidades a virus informáticos, mejoras, etcétera) y permiten a otros beneficiarse del conocimiento, el cual se construye sobre los descubrimientos de otros. Algunas de las comunidades más importantes y grandes son PHP, Linux, Apache, Perl y Sendmail.

comercializar gran parte de lo que se genera al interior de la comunidad. En el caso de Manet, la empresa ha identificado herramientas desarrolladas en la comunidad que explota comercialmente, pero como menciona el líder de proyecto de esta empresa, “las comunidades son abstractas, carecen de una estrategia comercial”.

Los desarrolladores independientes y los usuarios conscientes que trabajan bajo este esquema tienen una labor importante para reflejar hacia el interior de las comunidades lo que el mercado puede estar demandando, el problema es que las comunidades no tienen la capacidad para identificar e internalizar adecuadamente esa I+C. Sistematizar este proceso tal vez pueda romper la estructura sobre la cual se ha estado desarrollando una cantidad importante de conocimiento. Pero, tal vez sea pertinente definir mecanismos, como los *foros* entre empresas y comunidades, para aprovechar lo que se produce al interior de las comunidades, respetando su estructura dinámica y eficiente en la producción de conocimiento.

#### **8.4 CONCLUSIONES DEL CAPÍTULO**

A lo largo del capítulo 6 se explicó cómo y por qué las empresas mexicanas de software diseñan diferentes interfases, a través de las cuales identifican I+C de propósito general existentes en el entorno codificado, y específicos mediante la interacción con los usuarios y otros agentes. En el capítulo 7 se explicó cómo y bajo qué mecanismos las empresas de software internalizan esa I+C. Esta causalidad tiene diferentes efectos que se explican en este capítulo.

Como se muestra en el cuadro 8.1, a partir de una interfase simple la empresa identifica I+C de propósito general, los internaliza relativamente sin muchos problemas, pero lo que genera es información que se vuelve abundante, repetitiva y de propósito general. El conocimiento se expresa en la réplica de productos o aplicaciones ya existentes en el mercado.

A partir de una interfase semi-compleja las empresas pueden identificar e internalizar un alto contenido de información general y específica acerca de las necesidades y requerimientos del usuario. Los resultados son la generación de productos o aplicaciones con mejoras incrementales y nuevas aplicaciones.

A partir de una interfase compleja, las empresas identifican e internalizan I+C específicos. La información general utilizada es menor, de forma que los flujos de información se vuelven cortos, pero específicos y complejos. Los resultados son la generación de nuevo conocimiento, el diseño y desarrollo de nuevos productos, así como la formación de recursos humanos especializados.

**Cuadro 8.1**  
**Creación de conocimiento e información a partir de diferentes proyectos**

<i>Interfase</i>	<i>Insumos relevantes identificados e internalizados</i>	<i>Proyecto</i>	<i>Resultado</i>
Interfase simple	-Información y conocimiento de propósito general	-Réplica, adaptación y mejoras menores de aplicaciones ya existentes en el mercado -Relación compra-venta	-Información abundante y repetitiva, y conocimiento de propósito general
Interfase Semi-compleja	-Información y conocimiento de propósito general y específico	-Escalamiento y actualización de aplicaciones ya instaladas en la empresa usuaria	-Conocimiento general y codificado -Información específica
Interfase compleja	-Conocimiento e información específicos	-Diseño y desarrollo de nuevos productos, procesos y servicios	-Conocimiento específico y tácito -Información específica -Recursos humanos especializados

Fuente: Elaboración propia con base en entrevistas.

La vinculación universidad-empresa es uno de los mecanismos que caracterizan a una interfase compleja, es un tipo de proyecto distinto pero con un propósito de corto y mediano plazo necesario para las empresas de software. La fábrica de software como resultado de este proyecto es por sí mismo una muestra del contenido de conocimiento que se genera y del cual las empresas involucradas, y el sector en general, se benefician. El problema de este tipo de proyecto es la continuidad y el compromiso que deben tener los agentes involucrados, lo cual no ha sido fácil sostener en un entorno que cambia rápidamente y en donde deben moverse a la misma velocidad para que el conocimiento

generado pueda ser explotado y haya un margen de apropiabilidad del mismo, de otra forma, no habrá incentivos a seguir desarrollando este tipo de proyectos.

Si bien se ha estudiado a la industria de software, lo que permite conocer sus características productivas y de organización, no hay una definición ampliamente consensada de innovación en esta industria. Retomando a Segelod y Jordan (2004), en esta tesis se considera a la conversión de la I+C en el diseño y desarrollo de nuevas aplicaciones como una aproximación a la creación de nuevo conocimiento en la industria de software.

En el cuadro 8.2 se muestran los diferentes tipos de innovación en el segmento de software hecho a la medida. La innovación se distingue en incrementales y radicales para los tres tipos de interfase, simple, semi-compleja y compleja.

En una interfase simple, la innovación incremental implica mejoras y escalamiento a productos ya existentes en el mercado y que han alcanzado un alto grado de estandarización. En una interfase semi-compleja, la innovación implica rediseño y escalamiento de programas que ya existen en el mercado, así como diseño y desarrollo de nuevas aplicaciones para sectores particulares, como en el caso de la industria de tenería. En una interfase compleja, una innovación incremental consiste en escalar y rediseñar productos que bien pueden ser estandarizados pero también de productos que han sido hechos a la medida por otras empresas; la innovación radical implica el diseño y desarrollo de nuevas aplicaciones con base en las necesidades y requerimientos del usuario. Si bien las empresas, ubicadas en el segmento industrial de software hecho a la medida, realizan innovaciones a nivel de aplicaciones, una innovación importante es la creación de conocimiento especializado a través de proyectos de vinculación universidad-empresa.

**Cuadro 8.2**  
**Caracterización de la innovación en el segmento industrial de software hecho a la medida**

Tipo de interfase	Innovación	
	Incremental	Radical
Simple	Mejoras y escalamiento a programas de software ya existentes en el mercado y que han sido estandarizados.	No se realiza.
Semi-compleja	Rediseño y escalamiento de programas de software ya existentes en el mercado y que han sido estandarizados.	Diseño y desarrollo de nuevas aplicaciones hechas a la medida.
Compleja	Rediseño y escalamiento de programas de software hechos a la medida de los usuarios.	Diseño y desarrollo de nuevas aplicaciones hechas a la medida. Creación de conocimiento a través de proyectos como el de vinculación universidad-empresa.

Fuente: Elaboración propia.

Respecto a las comunidades de desarrolladores de SWFA, parece haber una gran producción de conocimiento que carece de explotación comercial. Los flujos de conocimiento son mayores que los de información y los resultados son la generación de conocimiento específico para las comunidades y para la empresa de software vinculada a esas comunidades. Una de las fortalezas de estas comunidades es su trabajo en redes a través de la cooperación personal que, aunque están organizadas tácitamente, tienen una gran capacidad de mejorar, escalar e innovar en nuevas herramientas y aplicaciones.

Al parecer, las comunidades pueden llegar a ser uno de los actores sociales de gran relevancia para la creación de capacidades de innovación en la industria de software. La pregunta que surge es: ¿Cómo absorber todo ese conocimiento generado al interior de las comunidades, de forma que puedan beneficiarse tanto los desarrolladores independientes (empresas basadas en SWFA) y usuarios conscientes, como el resto de los agentes (empresas usuarias, universidades, individuos)? La respuesta tiene un mosaico diverso de respuestas y pocas de ellas son contundentes, pero responderla será un reto para desarrolladores de software, usuarios conscientes, empresas, universidades e instituciones de gobierno.



Por otro lado, ¿Por qué no se replica en un contexto de software propietario? Una de las razones es que los propietarios del código no son capaces de liberarlo sin arriesgar los derechos de propiedad intelectual. El único lugar donde los desarrolladores de software propietario pueden emular al desarrollo de fuente abierta es al interior de la empresa. Muchas empresas de software propietario comercial han entendido la estrategia de capitalizar el movimiento de fuente abierta, al menos al interior de las empresas, debido a las derramas internas de conocimiento que se generan. Mientras que las mejoras en el SWFA no sean *apropiables*, las empresas propietarias se pueden beneficiar indirectamente de los desarrollos hechos por las comunidades de software de fuente abierta.

Si bien hay redes de trabajo al interior de las empresas de software, éstas no tienen la misma libertad ni alcance que las comunidades, y tratar de competir con una comunidad de desarrolladores de SWFA implica, entre otras cosas: i) una filosofía de trabajo radicalmente diferente, ii) una lógica de negocio distinta a la que actualmente prevalece, iii) ajustarse a un comportamiento tecnológico en constante y rápida evolución, y iv) una responsabilidad económica, social y tecnológica aún mayor.

## CONCLUSIONES GENERALES

### INTRODUCCIÓN

En esta tesis se ha descrito y analizado el proceso de construcción de interfases entre empresas mexicanas que desarrollan software hecho a la medida y diversos agentes de su entorno. Asimismo, se ha explicado cómo a partir de la construcción de diferentes interfases las empresas identifican información y conocimiento (I+C) existentes en el entorno, los cuales al ser valorados, asimilados y explotados adecuadamente por las empresas refuerzan su capacidad de absorción (CA) y sus capacidades tecnológicas. Un aumento de la CA se expresa en el diseño y desarrollo de nuevos productos, procesos y servicios. Estos temas se han abordado parcialmente en dos cuerpos de literatura.

Por un lado, la literatura sobre interfase destaca los flujos de información que se generan entre proveedor y usuario para el desarrollo de nuevos productos. Esta literatura ha centrado su atención en los flujos de información externos para explicar el proceso de generación de nuevo conocimiento, pero ha puesto poca atención a la forma en cómo las empresas identifican e internalizan la información y conocimiento para lograr ese proceso. Por otro lado, la literatura sobre CA se ha preocupado por cómo las empresas internalizan y explotan comercialmente la I+C del entorno. Con este enfoque, en algunos estudios se han analizado medianas y grandes empresas que cuentan con áreas funcionales formalmente, en particular con la de Investigación y Desarrollo (I+D), a partir de la cual la empresa puede identificar e internalizar información y conocimiento del entorno para luego transferirlos al resto de las áreas, es decir, se ha analizado en mayor profundidad los flujos internos y su influencia en la CA e identificado patrones de comportamiento generales para medianas y grandes empresas.

El primer cuerpo de literatura ha ayudado a documentar las formas de interacción proveedor-usuario y, el segundo, la forma en cómo medianas y grandes empresas absorben información y conocimiento. Sin embargo, en México el segmento de software hecho a la medida se caracteriza por la existencia de una gran cantidad de micro y pequeñas empresas en las cuales no hay espacios para profundizar en los flujos internos, pues carecen de áreas

funcionales formalmente, por esta razón, el desarrollo de flujos de información depende en buena medida de la interacción con diversos agentes externos. En este tipo de empresas la interfase les permite identificar I+C que pueden ser internalizados a través de mecanismos como la capacitación, los foros de discusión interna, el conocimiento previo relacionado a sus funciones, los cuales son actividades de aprendizaje que influyen en su CA y en el diseño y desarrollo de nuevos productos. Estos problemas particulares no habían sido parte de las prioridades de investigación de esos cuerpos de literatura ya que su contexto de estudio estaba formado por otro tipo de empresas.

A partir de esos cuerpos de literatura se construye la estructura conceptual que conduce a la formulación y respuestas de las siguientes preguntas de investigación:

1. ¿Cómo y por qué se construyen interfases entre las empresas de software y los diversos agentes del entorno? ¿Qué tipos de interfase se construyen y cuáles son sus características?
2. ¿En qué medida las interfases permiten a las empresas identificar información y conocimiento e incrementar su capacidad de absorción?
3. ¿Cómo se expresa un incremento de la capacidad de absorción de las empresas? ¿Qué diferencias existen entre las empresas con distinta capacidad de absorción para identificar e internalizar información y conocimiento y materializarlos en aplicaciones específicas?

Las preguntas de investigación han sido abordadas a través del estudio de empresas mexicanas que desarrollan software hecho a la medida, mediante la estrategia de investigación conocida como estudio de caso múltiple-exploratorio. Los casos son empresas de software de fuente abierta y empresas de software propietario, considerando a la interfase como la unidad de análisis.

En los siguientes tres apartados se encuentran las conclusiones generales derivadas de las respuestas a estas preguntas. En el cuarto apartado se presentan las aportaciones teóricas y empíricas de la tesis. En el quinto apartado se exponen los temas a considerar para futuras investigaciones. Finalmente, se reflexiona acerca de las recomendaciones de política orientadas a fomentar el desarrollo de la industria de software.

## 1. CREACIÓN DE INTERFASES EN UN ENTORNO DE RÁPIDO CAMBIO TECNOLÓGICO

Ante la primera pregunta de investigación ¿Cómo y por qué se construyen interfases entre las empresas de software y los diversos agentes del entorno? y la pregunta subsidiaria ¿Qué tipos de interfase se construyen y cuáles son sus características? se mostró, en el capítulo 6, que las empresas construyen interfases simples, semi-complejas o complejas, las cuales dependen del tipo de agente involucrado y de las características del proyecto. Asimismo, los diferentes tipos de interfase se caracterizaron en función del grado de complejidad, intensidad y formalidad de la interacción con los agentes participantes.

Durante el proceso de consolidación de la industria mexicana de software, en el segmento de software hecho a la medida las empresas enfrentan problemas de incertidumbre tecnológica debido a la evolución constante tanto de las tecnologías de software como de las necesidades de los usuarios. Una forma de reducir esa incertidumbre es a través de las interfases externas, ya que a través de éstas las empresas pueden intercambiar información y de esta forma conocer, al menos de manera aproximada, la evolución de esas tecnologías y mercados. Sin embargo, no todas las empresas pueden identificar con la misma precisión y velocidad la I+C del entorno. Aquellas que logran identificarlos e internalizarlos adecuadamente incrementan sus capacidades técnicas y esto, a su vez, les permite reforzar su CA y evolucionar en la medida en que lo hace el mercado. Una parte de la información y conocimiento requeridos para el diseño y desarrollo de los productos de software que son demandados por los usuarios se encuentran en un entorno no codificado, por tanto, para acceder a ella se requiere de la interacción entre los diferentes agentes, lo cual propicia la creación de interfases.

En la investigación se encontró que, en el segmento de software hecho a la medida, una de las tareas más importantes para las empresas de software en el desarrollo de aplicaciones consiste en *identificar* adecuadamente la I+C acerca de los requerimientos de las empresas usuarias. Estas necesidades son específicas a cada cliente y los proveedores las llegan a conocer sólo mediante la interacción. Cada empresa usuaria posee solamente un fragmento de dicha I+C, los cuales son importantes para el diseño y desarrollo de software, otros

fragmentos los poseen los proveedores, y sólo toman coherencia cuando se articulan entre sí a través de las interfases.

El desarrollo de un producto confiable y de calidad requiere de un cúmulo de información y conocimiento diversos, algunos de ellos específicos a la empresa usuaria y sólo pueden ser proporcionados por ésta, pero hay otro cúmulo de información y conocimiento que son del dominio de la empresa de software. La articulación de estos dos cúmulos de I+C sólo se puede dar a partir de la interacción entre las empresas de software con las empresas usuarias. Si la I+C que posee la empresa usuaria son altamente específicos y no están codificados, la empresa de software requerirá de una intensa interacción con el usuario para identificarlos y esto propicia un tipo de interfase, la cual será diferente cuando la I+C que la empresa de software requiere de la empresa usuaria sean de menor especificidad o se encuentren codificados. Esta es una de las razones por las cuales las empresas de software deben establecer más de un tipo de interfase.

Por ejemplo, los estudios de caso muestran que las interfases simples están asociadas con la identificación e internalización de I+C de propósito general, por tanto las empresas participantes en esas interfases, como Constable, realizan sólo réplica, escalamiento o mejoras menores a productos ya existentes en el mercado. Esta limitación obedece a diversos elementos causales: por un lado, hay una falta de capacidad de la empresa para identificar a empresas que requieran software específico y no sólo de propósito general, lo cual constituiría para esa empresa nuevos nichos de mercado y realizar proyectos de mayor valor agregado. Por otro, existe un segmento amplio de micro, pequeñas y medianas empresas usuarias que se encuentran tecnológicamente atrasadas, situación que las hace poco demandantes de tecnologías específicas, pero demandan aplicaciones de menor complejidad (en algunos casos réplica de productos ya existentes) para sobrevivir en el mercado, y esto representa buenos rendimientos económicos para empresas como Constable. De esta forma, parece que ésta empresa encuentra rentable económicamente mantenerse en el círculo vicioso *interfases simples–réplica de conocimiento–interfases simples*, un esquema donde prevalece la réplica con adaptaciones menores a productos que se destinan a nichos de mercado consolidados; esta circularidad ocasiona un patrón de

comportamiento asociado a la ausencia de búsqueda, por parte de los proveedores, de soluciones a problemas específicos de las empresas usuarias, y de éstas por modernizar sus procesos productivos y organizacionales.

Para poder hacer esa búsqueda, se requiere de una capacidad para identificar I+C de propósito general y específicos, como es el caso de las empresas que están asociadas con una interfase semi-compleja. Degas, como ejemplo de este tipo de interfase ha identificado un nicho importante de desarrollo: la industria de tenería. Esta empresa ha desarrollado proyectos que implican mejoras incrementales, análisis, diseño y desarrollo de nuevas aplicaciones, así como la capacitación a usuarios. Gran parte de la I+C requeridas para el desarrollo de proyectos los ha obtenido a través del usuario, ya que no se encuentran en el entorno codificado. Lo anterior le ha permitido a la empresa diseñar y desarrollar aplicaciones inexistentes en una industria con procesos estandarizados: la industria de tenería, la cual parecía haber llegado a su límite tecnológico, sin embargo, la tecnología de software permeó los procesos organizacionales y productivos de las empresas dedicadas a este ramo. Pero esto se logró debido a que Degas tuvo la capacidad para identificar adecuadamente los requerimientos específicos de la empresa usuaria.

En lo que se refiere a las interfases complejas, la evidencia muestra que éstas implican un alto nivel de identificación e internalización de I+C específicos y, por consiguiente, el diseño y desarrollo de nuevos programas y, en algunos casos, la formación de recursos humanos especializados, como sucede en Goya. Además de esta empresa, Renoir o Manet, asociadas a una interfase compleja, han obtenido un mejor desempeño, han hecho grandes esfuerzos por emprender proyectos de mayor valor agregado y han atendido a la demanda de los usuarios de productos de mayor especificidad debido a que han logrado desarrollar actividades de aprendizaje a lo largo de los proyectos previos, actividades que inciden en el rompimiento del círculo vicioso mencionado en el caso de Constable.

De esta forma, se muestra que las interfases externas se construyen y son de suma importancia porque en el segmento de software hecho a la medida cada agente posee una parte crucial de I+C que se articulan a través de la interfase, es decir, se conjugan diferentes

unidades de I+C fragmentarios. En algunos casos la interfase va más allá de la interacción proveedor-usuario para el desarrollo de aplicaciones/sistemas de software, y requerirá de la vinculación con universidades para la configuración de entornos de conocimiento donde se desarrollen habilidades específicas para segmentos particulares de la industria, de forma que el concepto de interfase se vuelve un concepto envolvente de los conceptos proveedor-usuario y vinculación universidad-empresa.

## **2. ABSORCIÓN DE INFORMACIÓN Y CONOCIMIENTO**

Ante la segunda pregunta de investigación formulada ¿En qué medida las interfases permiten a las empresas identificar información y conocimiento e incrementar su capacidad de absorción?, en el capítulo 7 se argumentó que las empresas en estudio pueden identificar I+C de propósito general y específico, si bien pueden absorber aquellos de propósito general, tienen limitaciones para absorber los específicos. Los mecanismos más importantes para lograr incrementar la CA son la contratación de profesionales con educación formal, los sistemas de procesamiento de conocimiento observable (en particular las bitácoras), la transferencia de conocimiento entre las áreas (en particular los foros de discusión) y otras actividades de aprendizaje (en particular la reutilización de módulos y la capacitación).

Una de las principales conclusiones del capítulo 7 radica en que el esquema multitareas, bajo el cual los profesionales deben desempeñar simultáneamente varias actividades, permite a estos aprender e internalizar conocimientos a nivel individual, pero con barreras o límites importantes para expresarse a nivel organizacional debido a que ese esquema reduce el tiempo de socialización de lo aprendido entre una actividad y otra. Los casos estudiados, a excepción de Goya, tienen una organización del trabajo basada en dicho esquema.

Otro fenómeno observado es que muchas veces la obtención de conocimientos se da a través de la capacitación, sin embargo, cuando el profesional de software participa en varias actividades en un mismo proyecto frecuentemente no cuenta con el tiempo para aplicar los conocimientos adquiridos y tendrá bajo impacto en la empresa, pues, debido al uso de ese esquema multitareas, se involucrará en la solución de problemas no planeados y no

explotará los conocimientos obtenidos mediante la capacitación. Esta característica es común en la micro y pequeña empresa de software, ya que por escases de personal este no puede especializarse y, además, generalmente carecen de áreas organizacionales funcionales formalmente.

No obstante, un aspecto positivo de ese esquema de trabajo es que los profesionales adquieren habilidades en áreas administrativas tales como gestión o marketing, actividades de las cuales las empresas de software carecen profundamente.

Otra de las conclusiones es que, en un contexto de modernización tecnológica en el cual las empresas usuarias buscan mejorar su desempeño organizacional y productivo a través de la incorporación de software en sus procesos, el dinamismo del segmento de software hecho a la medida dependerá de la capacidad de las empresas de software para identificar e internalizar I+C específicos que les permita desarrollar las aplicaciones que requieren las empresas usuarias, así como de la velocidad con la cual éstas, sobre todo aquellas ubicadas en industrias maduras, empiecen a modernizarse y a incorporar software en sus procesos organizacionales y productivos.

Por ejemplo, Degas identificó las necesidades específicas de una empresa ubicada en la industria de tenería y desarrolló aplicaciones para empresas de esta industria, gracias a ellos logró posicionarse en el mercado. Esta empresa y otras como Manet, Renoir y Goya han mostrado una alta capacidad para incursionar en proyectos que implican diseño y desarrollo de software en sus múltiples etapas, los cuales representan un alto valor agregado para la empresa, y no sólo participar en proyectos de instalación, adaptación o modificación menores, que normalmente representan un bajo valor agregado para la empresa. En este sentido, dada la relevancia que toma la transferencia de I+C de carácter específicos, mismos que sólo pueden ser proporcionados por el usuario, la CA de las empresas es fundamental para el desarrollo de las aplicaciones específicas de software.

Finalmente, se ha mostrado la existencia de una interdependencia en la que un mayor aprovechamiento de los flujos de información y conocimiento en la interfase contribuye a



reforzar la CA de las empresas, y un incremento en la CA contribuye a la creación de una interfase de mayor complejidad. Esta circularidad tiene efectos distintos en la creación de nuevos productos, los cuales dependen del tipo de interfase creada y del nivel de CA lograda por las empresas de software, como veremos en el próximo apartado.

### **3. TRANSFORMACIÓN DE INFORMACIÓN Y CONOCIMIENTO EN NUEVOS PRODUCTOS A PARTIR DE LAS INTERFASES**

La tercera y última línea de investigación de esta tesis se formuló con base en las siguientes interrogantes: ¿Cómo se expresa un incremento de la capacidad de absorción de las empresas? ¿Qué diferencias existen entre las empresas con distinta capacidad de absorción para identificar e internalizar información y conocimiento y materializarlos en aplicaciones específicas? Para responder a ello, como se expone en el capítulo 8, se analizó el cómo a partir de construir diferentes interfases, las empresas estudiadas identifican I+C existentes en el entorno, y cómo mediante diversos mecanismos de aprendizaje internalizan dicha I+C, incrementan su capacidad de absorción y los transforman en aplicaciones nuevas y específicas.

La I+C que las empresas de software utilizan para el desarrollo de aplicaciones los pueden identificar de dos fuentes fundamentales: (i) del entorno codificado, por ejemplo a través del internet, bitácoras, libros y revistas especializados, etc., y (ii) de los clientes o usuarios de las aplicaciones que desarrollan. Si las empresas de software logran internalizar dicha I+C, en principio, estos deben traducirse en líneas de código, módulos y aplicaciones comerciales. Las empresas requieren de conocimientos y habilidades de programación para lograr transformar la I+C en aplicaciones, las cuales pueden ser réplicas de las ya existentes en el mercado o bien ser nuevas. Retomando a Segelod y Jordan (2004), en esta tesis se considera a la conversión de la I+C en el diseño y desarrollo de nuevas aplicaciones como una aproximación a la creación de nuevo conocimiento en la industria de software. Así, la importancia de la interfase radica en el logro del aprendizaje y en el aumento de la CA, pero también en la generación de conocimiento específico y en su materialización en aplicaciones nuevas y específicas.

En esta tesis se ha mostrado que en las interfases simples circula fundamentalmente I+C de propósito general, en las cuales generalmente no participan usuarios sofisticados con necesidades específicas y complejas. En ese sentido, las empresas de software que crean estas interfases sólo son capaces de identificar I+C de propósito general, y si bien los pueden internalizar de manera relativamente fácil, tienen poca capacidad para generar conocimiento específico. Lo anterior está altamente asociado al tipo de proyecto que desarrollan, pues son simples y enfocados a procesos contables, de nómina, etc., que en mayor medida requerirán del uso de información que se puede encontrar en el entorno codificado para el desarrollo de esas aplicaciones.

En las interfases semi-complejas circula tanto I+C de propósito general como específico, ya que en ellas participan usuarios sofisticados con necesidades específicas y complejas. Las empresas de software participantes en este tipo de interfases son capaces de identificar I+C de propósito general y específicos a través de procesos moderados de interacción entre las empresas de software y las empresas usuarias; éstas empresas tienen necesidades específicas, las cuales son resueltas si se articulan los conocimientos de ambas empresas.

Degas es representativa de las empresas de software que participan en una interfase semi-compleja. Uno de los resultados a partir de ese tipo de interfase y de su moderado incremento en su CA ha sido la generación de nuevo conocimiento (aplicaciones de software) tanto para la industria de tenería como para la de software. La I+C utilizada para el desarrollo de esas aplicaciones están asociados a procesos organizacionales y productivos estandarizados, es decir, la industria de tenería tiene procesos tecnológicos maduros y estandarizados; si bien una parte de esa I+C están codificados, otra parte son específicos a la empresa usuaria y sólo se adquieren a través de la interacción con el usuario. En esta tesis se muestra que la I+C codificados en industrias maduras como la de tenería, no lo están en la de software, esto significa que traducirlos a aplicaciones ha sido uno de los mayores logros de la empresa Degas, pues las aplicaciones cumplen con una nueva función tecnológica, económica y social antes no conocida en México ni por las empresas de tenería ni las de software.

En las interfases complejas circula fundamentalmente I+C específicos, en las cuales el usuario juega un papel aún más activo y se involucra en el desarrollo del proyecto, en consecuencia, aumenta la interacción y el intercambio de I+C específicos. Empresas como Goya, Manet y Renoir han mostrado la capacidad para internalizar una buena parte de esa I+C específicos de los usuarios, los resultados son nuevas aplicaciones generalmente de mayor valor agregado, así como la formación de recursos humanos especializados, como sucedió en el caso del proyecto de vinculación de la empresa Goya con UNIVER.

De lo anterior se desprende que la generación de aplicaciones nuevas, específicas y de mayor valor agregado depende del tipo de interfase creada y del tipo de usuario involucrado en la misma. Asimismo, se muestra que una gran cantidad de esas aplicaciones diseñadas y desarrolladas en la industria de software son derivadas de una correcta identificación e internalización de las necesidades específicas de los usuarios. La descripción de los casos en el capítulo 6 (apartado 6.2, 6.3 y 6.4) muestra que las aplicaciones de mayor calidad o más ‘puras’ se han logrado cuando incorporan de la mejor manera las necesidades específicas de los usuarios, quienes se vuelven cada vez más importantes en el segmento de software hecho a la medida.

Por otra parte, el software incorporado en procesos organizacionales y productivos, como en la industria de tenería, es una muestra de la transversalidad del software y de la oportunidad de crear nuevos nichos de mercado para la industria de software. Aprovecharlos requiere el uso de un conjunto de factores, a saber: i) las capacidades tecnológicas acumuladas, ii) la capacidad de identificar y evaluar la I+C que se ha generado en esas industrias, y iii) la capacidad para absorberlos y explotarlos con fines comerciales. Uno de los resultados más importantes derivado de esto consiste en que, las industrias que creían haber llegado a su límite tecnológico, han podido reconfigurar sus procesos productivos y organizacionales, es decir, la tecnología de software ha revitalizado industrias ya maduras y ha conformado tejidos y competencias distintos. Sin embargo, existirá una limitación estructural de la industria de software mientras no haya fomento a la modernización de las industrias que potencialmente pueden incorporar software en sus productos, procesos o servicios. Por tanto, al no modernizarse las empresas usuarias, éstas tampoco presentan nuevas demandas

a las empresas de software. Lo anterior, a su vez, puede ser una limitación para que las empresas de software, ubicadas en una interfase simple, migren a interfaces más complejas.

Finalmente, otra conclusión del capítulo 8 consiste en que las empresas ubicadas en un contexto de software propietario (SWP) tienden a generar conocimiento a nivel de empresa, al interior o a través de la relación proveedor-usuario, mientras que en el contexto de software de fuente abierta (SWFA) tienden a generar grandes montos de conocimiento a través del trabajo en redes informales de cooperación personal, es decir, a través de las comunidades de desarrolladores de SWFA. La ausencia de mecanismos para lograr que ese conocimiento generado tenga aplicaciones comerciales se ha convertido en uno de los problemas de estas comunidades.

#### **4. APORTACIONES TEÓRICAS Y EMPÍRICAS**

En los capítulos 6, 7 y 8 se presentó y analizó la evidencia empírica de los estudios de caso. Los hallazgos mostrados en esos capítulos ilustran cómo se construyen las diferentes interfaces y la forma en que éstas influyen en la capacidad de absorción de las empresas. Como resultado de estos capítulos, a continuación se expresan las siguientes aportaciones.

Primero, Andersen (1991) menciona que las empresas crean diferentes interfaces dependiendo del nivel de estabilidad y estandarización de la tecnología vigente, de forma que se crean interfaces simples cuando se ha establecido conocimiento rutinario, un lenguaje común y conocimiento fácil de codificar; y complejas en cambio, cuando el entorno es rico en información, la tecnología es emergente, es difícil establecer patrones comunes de comunicación y predomina el conocimiento tácito. La evidencia empírica presentada en esta tesis muestra que los diferentes tipos de interacción entre las empresas de software y otros agentes son cruciales para que estas empresas identifiquen I+C relacionados con las necesidades de los usuarios y así desarrollar proyectos de software. Dichas necesidades tienden a ser específicas a cada agente, y éste, a su vez, posee un fragmento de dicha I+C. Estos fragmentos sólo toman coherencia cuando se articulan entre sí a través de las interfaces. De esta forma, en este estudio se consideran tres aspectos no

tratados en la literatura sobre interfase: i) la interacción entre múltiples agentes, ii) la interfase externa como un mecanismo articulador de I+C fragmentarios y específicos, y iii) la interfase como un concepto envolvente de los conceptos proveedor-usuario y vinculación universidad-empresa.

Por otra parte, la literatura sobre interfase ha puesto mayor énfasis en la causalidad de los efectos que van del proveedor hacia el usuario, pero la realidad de la industria de software muestra que este hecho, al menos en las interfaces semi-complejas y complejas, sucede al inverso, del usuario hacia el proveedor. En la medida que buena parte del conocimiento necesario para desarrollar nuevas aplicaciones proviene del usuario, éste está dirigiendo, también en gran parte, el diseño y desarrollo de las nuevas aplicaciones.

Segundo, la creación de nuevo conocimiento requiere de diversos procesos tales como el aprendizaje. Cohen y Levithal (1990) y Van den Bosch, *et al.* (1999 y 2002) señalan que las interfaces internas son importantes para este propósito. Aunque la categoría de interfase ha sido menos considerada en la literatura sobre CA, desde éste enfoque han sido estudiados en mayor profundidad las interfaces internas que las externas. Sin embargo, la realidad de la industria mexicana de software muestra que las interfaces externas son un mecanismo crucial de aprendizaje que influye en el incremento de la CA de las empresas de software. Se ha mostrado que un mayor aprovechamiento de los flujos de I+C en la interfase contribuye a reforzar la CA de las empresas, y un incremento en la CA favorece la creación de una interfase de mayor complejidad. En este sentido, dependiendo del tipo de interfase creada y del nivel de CA lograda por las empresas, esa circularidad tiene efectos distintos en la creación de nuevas aplicaciones. La limitación para la creación de interfaces internas (con las características que señalan estos autores) se debe a la estructura organizacional y productiva de la micro y pequeña empresa carente de áreas funcionales formalmente, en consecuencia, limita la creación de flujos internos entre las áreas. Lo anterior no había sido parte de las prioridades de investigación para esta literatura, ya que su contexto de estudio estaba formado por medianas y grandes empresas con áreas funcionales formalmente, específicamente con la de I+D.

Tercero, ante la falta de estudios que presenten una caracterización de la industria de software a nivel nacional, esta tesis intenta llenar este vacío mostrando la caracterización de empresas ubicadas en el segmento de software hecho a la medida en México, concretamente de la micro y pequeña empresa:

- a) las empresas analizadas tienen entre 4 y 10 empleados y son parte de una industria atomística donde casi el 87% son micro y pequeña empresa,
- b) el destino de sus productos es nacional, pero con una búsqueda de inserción en el mercado internacional, concretamente el estadounidense,
- c) algunas requieren certificación de calidad en sus procesos para aumentar su participación de mercado en el exterior, pero obtener esos certificados es costoso,
- d) el uso de tecnologías (herramientas, aplicaciones) es variado y en constante cambio, lo cual hace a la industria altamente dinámica y en constante transformación y evolución,
- e) tienen una estructura de organización del trabajo poco funcional y es informal, ésta organización se basa en un 'esquema multitareas',
- f) la capacitación de los empleados es una de las actividades más importantes para su desarrollo y crecimiento, y
- g) las empresas inician actividades productivas con recursos financieros personales, pues hay una carencia de mecanismos financieros para otorgar créditos ajustados a sus necesidades.

Al mismo tiempo, las empresas enfrentan limitaciones y logros importantes. Una de las limitaciones es la falta de oportunidades y capacidades para el desarrollo de productos de mayor valor agregado. Por un lado, las empresas de software carecen de rutinas de búsqueda o exploración de nuevos nichos de mercado sobre los cuales pudieran expandirse, por otro, como se mencionó en párrafos anteriores, hay una escasa demanda de las empresas usuarias por modernizarse, particularmente a partir de la incorporación de software en sus procesos organizacionales y productivos. Sin embargo, uno de los mayores logros es el proceso de articulación de I+C fragmentarios que cada agente posee a través de las interfases. Por eso las interfases son importantes para esas empresas, ya que una gran cantidad de I+C acerca de los procesos productivos y organizacionales de los usuarios son

específicos y no están codificados, pero no por ser completamente tácitos sino porque las empresas usuarias han realizado pocos esfuerzos para codificarlos.<sup>1</sup>

Finalmente, bajo la perspectiva de la economía de la innovación, en un mundo que tiende al uso del conocimiento como insumo y al mismo tiempo a generarlo como producto (sobre todo en la industria de tecnologías de la información y comunicación -TIC-), las interfases creadas permiten un mejor flujo y transferencia de conocimiento generado al interior de las empresas, en particular en las de software, de tal forma que la configuración de interfases externas entre la empresa y los diversos agentes del entorno aumenta la complementariedad de capacidades y reduce la incertidumbre del riesgo a invertir en la exploración de nuevo conocimiento. Andersen (1991 y 1996) hizo parcialmente esta afirmación al decir que las interfases simples y estables reducen el riesgo a invertir en nuevo conocimiento. Por su parte, Lane y Lubatkin (1998), Lane, *et al.* (2002), y Van den Bosch, *et al.* (1999 y 2002) argumentan que “en un entorno de conocimiento turbulento...la complementariedad de capacidades conduce a un incremento y flexibilidad en la absorción de conocimiento”.

## **5. TEMAS A CONSIDERAR PARA FUTURAS INVESTIGACIONES**

En esta tesis se investiga cómo y por qué las empresas construyen diferentes interfases a través de las cuales identifican I+C, y cómo éstos son internalizados para expresarlos en nuevos productos. El estudio se limita a cinco micro empresas y una mediana empresa ubicadas en el segmento de software hecho a la medida.

En el análisis se incluyeron empresas ubicadas en cuatro de las entidades más dinámicas en el desarrollo de software en México, no se incluyeron empresas de otras entidades como Tijuana o Morelos, también dinámicas y con un importante número de empresas en el contexto nacional. Por ejemplo, las empresas de software ubicadas en Baja California Norte interactúan con empresas transnacionales (maquiladoras), las cuales tienen requerimientos diferentes de las empresas no maquiladoras y, por tanto, se podría eventualmente suponer la

---

<sup>1</sup> Resultados similares asociados a la falta de esfuerzos por codificar conocimientos tácitos, que pueden ser de relativamente fácil codificación, fueron reportados por Dutrénit (2002) en su estudio sobre la industria de vidrio en México.

existencia de algún tipo de interfase con características diferentes. En el segmento de software hecho a la medida hay una gran diversidad de nichos de mercado a los que las empresas destinan sus productos, ya que cada una de ellas atiende a usuarios con necesidades distintas y, en su conjunto, cada entidad y nicho de mercado expondría a las empresas de software a diferentes comportamientos y alcances de interacción y de CA.

En ese sentido, en esta tesis se mostró el tipo de interfase y el grado de incremento en la CA de las empresas de software, así como sus efectos en el desarrollo de nuevas aplicaciones, pero sólo para nichos de mercado tales como el de aplicaciones para tenería, administración de negocios, educación, actividades forestales, páginas WEB y prueba de software. Sin embargo, ante la falta de estudios que presenten una caracterización de la industria de software a nivel nacional, y por ende la falta de información de los nichos de mercado a los cuales las empresas de software destinan sus productos, delimitó la ‘elección de los casos de estudio considerando criterios como: información abundante relacionada con los propósitos de la investigación, fácil acceso a las empresas y una buena relación con los informantes’.

En esta investigación se encontró que la industria de software tiene un impacto positivo sobre otras industrias diferentes de las TIC, como la de tenería. Se mostró que la relación entre ambas industrias permitió, por un lado, un avance en el desarrollo de nuevas aplicaciones y, por otro, revitalizar a empresas tradicionales. Aunque sólo se cuenta con la evidencia de un caso, ésta sugiere que la tecnología de software puede ser importante para revitalizar industrias que creían haber llegado a su tope tecnológico, queda por saber si esto se cumple para otras diferentes a la de tenería.

En ese sentido, un tema para futuras investigaciones será profundizar en cómo la industria de software se articula con otras industrias de alta, media o baja tecnología. No hay claridad de que la articulación con industrias de alta tecnología (como la de hardware) sea la mejor forma para lograr el desarrollo de la industria de software, tal vez las industrias maduras que empiezan a incorporar software en sus procesos productivos y organizacionales sean más importantes para el desarrollo y crecimiento de la industria nacional de software y, al



mismo tiempo, una fuente importante de I+C para migrar a procesos más complejos. Lo anterior lo muestra la industria de tenería, la cual está incorporando software en sus procesos organizacionales y productivos, ejemplo de la importancia de la transversalidad del software hacia industrias maduras que creían haber llegado a su límite tecnológico y que pueden ser revitalizadas.

Por otro lado, hay otros temas que quedan pendientes de abordar. Por ejemplo, se encontró que al interior de las comunidades de desarrolladores de SWFA se crea un cúmulo importante de conocimiento que bien podría ponerse a disposición de las empresas para la solución de problemas específicos, pero una gran parte de ese conocimiento no tiene aplicaciones comerciales debido a la falta de mecanismos para explotarlos en el mercado. En este sentido, otro tema importante por estudiar con mayor profundidad es cómo las comunidades de desarrolladores de SWFA contribuyen a la generación de conocimiento y, por tanto, a la creación de nuevos productos.

## **6. RECOMENDACIONES PARA LA POLÍTICA DE FOMENTO DE LA INDUSTRIA DE SOFTWARE**

Esta tesis se hizo con fines de responder a preguntas empíricas considerando aspectos teóricos, y no expresamente para delimitar las acciones de política para el desarrollo de la industria de software. No obstante, a partir de los argumentos presentados en los apartados anteriores, se pueden elaborar recomendaciones de política para el fomento de apoyo a esta industria.

De acuerdo con la evidencia presentada en esta tesis, se ha mostrado que el segmento de software hecho a la medida depende en gran parte de la demanda presentada por las empresas usuarias, pero éstas presentan una baja demanda, sobre todo la micro y pequeña empresa. El desarrollo de este segmento de la industria depende de que las empresas usuarias inicien un proceso de modernización a través de la incorporación de software en sus procesos productivos, organizacionales y de comercialización. Por ello, la política debe orientarse a crear vías de inversión en ciertas industrias que puedan modernizarse tecnológicamente a través de la incorporación de esa tecnología, en especial resultan

importantes las empresas ubicadas en industrias tradicionales. De esta forma se estaría favoreciendo el crecimiento de la industria de software. Al mismo tiempo, se debe diferenciar y focalizar el tipo de apoyo según el segmento de mercado que se quiera fomentar y apoyar.

En el capítulo 2, sección 2.2.1, se mostró que en México no hay un padrón de empresas de software a través del cual se puedan conocer sus necesidades de inversión, sus características productivas, tecnológicas y de desempeño, el destino de sus productos o la proporción de empresas desarrolladoras de software. Por eso, se deben diseñar medidas de política para conocer sus necesidades y características y así orientar adecuadamente el fomento a la inversión y los apoyos con recursos financieros.

Cada segmento de la industria requerirá de diversos tipos de apoyo. Algunas empresas, por ejemplo las micro y pequeñas que muestran un incremento significativo de su CA, podrán requerir recursos financieros para realizar y fortalecer sus actividades de I+D, vinculación con las universidades, bechmarcking, así como para adquirir certificados de calidad internacional con el objeto de competir en el mercado mundial. Otras, en cambio, requerirán recursos para la compra y actualización de equipo, capacitación, contratación de personal, entre otros.

También se ha mostrado que la vinculación universidad-empresa permite lograr el desarrollo de habilidades específicas requeridas en un segmento de la industria de software. El proyecto de vinculación entre Goya y UNIVER ha mostrado la importancia de esa vinculación por las externalidades positivas que genera, como lo es, por ejemplo, la creación de una fábrica de software y de una empresa incubadora al interior de la universidad. De esta forma, se deben fortalecer y promover los vínculos entre las empresas y las universidades con una proyección de corto, mediano y largo plazo. El diseño de programas de estudio en conjunto con empresas y universidades propicia una simetría entre demanda y oferta de recursos humanos especializados. Además, una de las actividades más importantes que se puede lograr a través de la vinculación es la capacitación (la cual es una actividad importante y constante en la industria), con beneficio mutuo tanto para las

empresas como para las universidades; es decir, las primeras se benefician por la reducción de costos y obtención de recursos humanos especializados, y las segundas tanto por la reformulación de los planes de estudio ajustados a las necesidades reales de la industria como por la generación de externalidades positivas mencionadas arriba.

Por otra parte, en el capítulo 6 y 7 se mostró que las empresas como Renoir, Constable, Velázquez y Goya se involucraron en la certificación de sus procesos con el fin de lograr un mejor desempeño de sus actividades y exportar sus productos a Estados Unidos. Conseguir la certificación de procesos como CMMi representa altos costos para las empresas; ante esto, MoPROSOFT ofrece a las empresas una alternativa viable a menor costo y a menor tiempo, en particular para aquellas que tienen la capacidad de exportar. México es de los pocos países en el mundo que tienen registrado a nivel internacional un certificado de calidad de los procesos, pero aunque resulta relativamente más económico las micro y pequeñas empresas no pueden acceder a él por la falta de recursos financieros. Por tanto, debe haber un fomento a la inversión para que las empresas puedan obtener certificados de calidad como el de MoPROSOFT y después migrar a otros como el de CMMi.

Finalmente, como se mostró en el capítulo 2, del total de la demanda de software en México, el segmento de software hecho a la medida representa sólo el 7.9%, (ver cuadro 2.5), mientras que el segmento de producción y autoconsumo realizado por instituciones de gobierno y empresas grandes manufactureras se ubica en 62.6%, el resto lo ocupa el software empaquetado. Si la producción de autoconsumo del gobierno y las industrias de manufactura se contratara a empresas independientes, la demanda de software hecho a la medida representaría alrededor del 70.5% del total de la industria. Si bien es posible que las empresas de software no accedan por completo a éste porcentaje, deben buscar la forma de cubrir una cuota considerable del mercado potencial. Sin duda, esta demanda potencial sólo puede ser aprovechada por las empresas si cuentan con las capacidades tecnológicas para ofrecer soluciones a problemas específicos del sector gobierno y de las industrias de manufactura, y si gobierno y empresas deciden dejar de producirlo por sí mismos para contratarlo a empresas independientes. Sin negar la necesidad de que gobierno y grandes empresas cuenten con grupos internos de desarrollo de software, se deben buscar

mecanismos para que las empresas independientes puedan desarrollar parte del software de autoproducción y autoconsumo. En este sentido, debe haber un fomento a la subcontratación de empresas especializadas en el desarrollo de software. Este esquema de subcontratación es una medida que ha sido aplicada por países como Brasil e Israel y es un elemento importante para el desarrollo de la industria mexicana de software, con efectos inmediatos en la generación de empleos calificados.

## BIBLIOGRAFÍA

- Abernathy, W, Utterback J. (1978).** “Patterns of industrial Innovation”, in *Technological Review*, Num. 9: 2-9.
- AMITI (2001).** *Esquema de Apoyo Gubernamental a la Industria del Software*. AMITI, México.
- Ancori, B., Antoine Bureth, and Patrick Cohendet (2000).** “The economics of knowledge: The debate about codification and tacit knowledge.” *Industrial and Corporate Change*, Vol. 9, Num. 2. pp. 255-287.
- Andersen, E. S. (1991).** “Techno-economic Paradigms as Typical Interfaces between Producers and Users”, in *Evolutionary Economics*, Vol. 1, Num. 2. USA:119-144.
- ..... (1994).** *Innovative linkages and systems of innovation*. Working Paper 994-6, ISSN 0908-9152. Department of Business Studies, University of Aalborg.
- ..... (1996).** *Evolutionary Economics. Post-Shumpetian Contributions*. Biddles Ltd. Guildtord and King’s Lynn, London.
- Andersen, E. S. and B-A. Lundvall (1988).** *Small Nacional Systems of Innovation Facing Technological Revolutions: An Analytical Framework*. Pinter Publishers Limited, London.
- Arora, A. et al. (2001).** “The Indian Software Services Industry”. *Research Policy*, Num 30, pp. 1267-1287.
- Arthur, W. B. (1989).** “Competing Technologies, Increasing Returns, and Lock-In by Historical Events”, in *The Economic Journal*, Vol. 99, Num. 394, March, pp 116-131.
- Arrow, K. (1962).** “The Economic Implications of Learning by Doing”, *Review of Economic Studies*, Vol. 29.
- Assimakopoulos, D. y Jie Yan (2006).** “Sources of knowledge acquisition for Chinese software engineers”. *Research and Developments Managements*, Vol. 36, Num. 1. pp. 97-106.
- Baskaran, A. and M. Muchie (edts) (2006).** *Bridging the Digital Divide. Innovation System for ICT in Brazil, China, India, Thailand and Southern Africa*. Adonis & Abbey Publishers LTD, London.

- Bajpai, N. and V. Shastri (1998).** “Software Industry in India: a Case Study”, in *Development Discussion Paper*, Num. 667, Harvard University.
- 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 Pavitt, K. (1993).** “Technological Accumulation and Industrial Growth: Contrasts Between Developed and Developing Countries”, in *Science Policy Research*. Oxford University Press, UK: 157-209.
- ..... (1995). “The Development of Technological Capabilities”, in: Haque, I. (ed.) *Trade, Technology and International Competitiveness*. The World Bank, Washington: 69-101.
- Borja, A. (1995).** “El cambio tecnológico de las computadoras: Patrones de la intervención estatal”, en *El Estado y el Desarrollo Industrial. La Política Mexicana de Computo en una Perspectiva Comparada*. CIDE, México.
- Brady, T. (1997).** Technology Management in Complex Product Systems: Who do you turn to when your technology supplies let you down? In *CoPS*, Num. 20, SPRU, UK.
- Breschi, S. y F. Malerba (1996).** ‘Sectoral Innovation Systems: Technological Regimes, Shumpeterian Dynamics, and Spatial Boundaries’, en *Systems of Innovation I*. Harvard Business Review, Nov-Dec, 1998.
- Breznitz, D. (2003).** *Software Tooling: The Evolution of the Israeli Software Industry*. Documento de trabajo. MIT, USA.
- Brusoni, S. and A. Prencipe (1999).** “Modularity in Complex Product Systems: Managing the Knowledge Dimension”, in *CoPS*, Num 57, SPRU, UK.
- Carlsson, B. and R. Stankiewicz (1991).** “On the Nature, Function and Composition of Technology Systems”, in *Evolutionary Economics*, Vol. 1, Num. 2. USA: 93-118.
- Chandler, A. D. Jr. y James W. Cortada (2002).** “La era de la información. Continuidades y diferencias”, en Alfred Chandler Jr. y James W. Cortada (eds.), *Una Nación Transformada por la Información. Cómo la Información ha Modelado a EUA Desde la Época de la Colonia hasta la Actualidad*. Oxford University Press, México.
- Clarke, T. and S. Clegg (1998).** *Changing Paradigms. The Transformation of Management Knowledge for the 21<sup>st</sup> Century*. Harper Collins Business, London.

- Cohen, W. M. and D. A. Levinthal (1990).** “Absorptive Capacity: A New Perspective on Learning and Innovation”, in *Administrative Science and Quarterly*, Vol. 35, Num. 1, March, pp. 128-152.
- Cohendet, P. and W. E. Steinmueller (2000).** “The codification of knowledge: A conceptual and Empirical exploration”. *Industrial and Corporate Change*, Vol. 9, Num. 2, pp. 195-209.
- Coombs, Rod and Richard Hull (1998).** “‘Knowledge Management Practices’ and Path-Dependency in Innovation”, in *Research Policy* 27, USA: 237-253.
- Cortada, J. W. (2002).** “Progenitores de la era de la información. El desarrollo de chips y computadoras”, en Alfred Chandler Jr. y James W. Cortada (eds.), *Una Nación Transformada por la Información. Cómo la Información ha Modelado a EUA Desde la Época de la Colonia hasta la Actualidad*. Oxford, México.
- Corona, J. M. y C. A. Hernández (2000).** “Relación Proveedor-Usuario y flujos de Información Tecnológica en la Industria Mexicana”, en *Comercio Exterior*, Vol. 50, Núm. 9, sept., México.
- Cruz, A. R. (2005).** “La India: Avance Tecnológico y Pobreza Extrema”, en *Comercio Exterior*, Vol. 55, Núm. 1, México: 58-66.
- Cusumano, M. (1988).** *Software Technology Management: ‘Worst Problems and ‘Best’ Solution*. MIT, USA.
- Clarke, T. and S. Clegg (1998).** *Changing Paradigms. The Transformation of Management Knowledge for the 21<sup>st</sup> Century*. Harper Collins Business, London.
- Dahl, M. S. and C. O. Pederson (2004),** “Knowledge Flows through Informal Contacts in Industrial Clusters: Myth or Reality”, in *Research Policy*, Num. 33, pp. 1673-1686.
- Dahlander, L. and M. G. Magnusson (2005).** “Relationships between open source software companies and communities: Observations from Nordic firms”, in *Research Policy*, Num. 34, p. 481-493.
- Davenport, T. H. and L. Prusak (1998).** *Working Knowledge. How Organizations Manage what They Know*. Harvard Business Scholl Press, USA.
- De Fuentes, C. (2007).** *Derramas de conocimiento y capacidades de absorción: El caso de las pymes de maquinados industriales en Querétaro*. Tesis Doctoral, UAM-X.

- Delgado, J. y G. Hernán (2005).** *Políticas de Promoción de la Industria de Software. Análisis Comparativo de Argentina con las 3I (India, Irlanda e Israel)*. Documento de Trabajo. Universidad del CEMA, Arg.
- Dieter, E. (1986).** “La Automatización Basada en el Uso de Computadoras y la Internacionalización de la Industria Electrónica. Implicaciones Estratégicas para los Países en Desarrollo”, en Mintan (coord.) *Industria y Nuevas Estrategias de Desarrollo en América Latina*, CIDE, México.
- ..... (1998). *Catching-Up, Crisis and Industrial Upgrading, Evolutionary Aspects of Technological Learning in Korea’s Electronics Industry*. DRUID.
- Dixon, N. M. (2000).** *Common Knowledge*. Harvard Business Scholl Press, USA.
- Dosi, G. (1982).** “Technological Paradigms and Technological Trajectories”, in *Research Policy*, Num 11.
- Dodgson, M. (1993).** “Organizational Learning: A Review of Some Literatures”. In *Organizational Studies*. Vol. 14, Num. 3. UK. pp. 375-394.
- Dutrénit, G. (2000).** *Learning and Knowledge Management in the Firm: From Knowledge Accumulation to Strategic Capabilities*, Cheltenham: Edward Elgar, U. K.
- Dutrénit, Vera-Cruz, Arias, Avendaño, Gil, Sampedro y Uriostegui (2002).** “Marco analítico para el análisis de los procesos de acumulación de capacidades tecnológicas”, documento de trabajo, proyecto “Aprendizaje tecnológico y escalamiento industrial: Generación de capacidades de innovación en la industria maquiladora de México”, COLEF/FLACSO/UAM, UAM.
- Eischen, K. (2002a).** *The Social Impact of Informational Production: Software Development as an Informational Practice*. Working Paper Series 2002-1 University of California, USA.
- ..... (2002b). *Software Development: A View from the Outside*. Working Paper Series 2002-3. University of California, USA.
- ..... (2003). ‘Development Architectures: Software Practice, Industry Organization and Industry Evolution in India’. Prepared for: *The Software Industry in the Developing World Workshop*. Yale University.
- ..... (2005). “Los servicios de tecnología de información en la India: protagonistas, lugares y prácticas”, en *Comercio Exterior*, Vol. 55, Núm. 9, sep. México.



- Fagerberg, J. (1993).** “User-Producer Interaction, Learning and Comparative Advantage”. Paper presented at the conference *Technology Collaboration: Networks, Institutions and States*, Manchester, April 21-23.
- Freeman, C. and Carlota Perez (1988).** “Structural Crises of Adjustment: Business Cycles and Investment Behaviour”, in Dosi *et al.*, *Technical Change and Economic Theory*. Pinter Publishers, London, UK.
- Freeman, C. (1988).** “Japan: a New National System of Innovation?”, in Dosi *et al.*, *Technical Change and Economic Theory*. Pinter Publishers, London, UK.
- FUMEC (2004).** “The Development of Embedded Software in Micro, Small and Medium sized Enterprises”, Vol. 1, Núm. 7, Octubre, México. [www.fumec.org](http://www.fumec.org)
- Geyer, A. and A. Davies (2000).** “Managing project–system interfaces: case studies of railway projects in restructured UK and German markets”, in *Research Policy*, Num. 29. p. 991-1013.
- Gold, J. (1994).** “La empresa basada en los conocimientos”, en Clarke, K. y E. Monkhouse (edts.), *Replantarse la Empresa*. Ed. Folio, España.
- Graaf, B. et al (2003).** “Embedded Software Engineering: The State of the Practice”, IEEE SOFTWARE, Novembe-December, USA. <http://computer.org/software>
- Gómez, A y C. Suárez (2004).** *Sistema de Información, Herramientas para la Gestión Empresarial*. Alfaomega, México.
- González, D. (2006).** *Estudio Exploratorio de los Factores Críticos de Éxito de la Industria Mexicana del Software y su Relación con la Orientación Estratégica de Negocio*. Informe de trabajo de investigación, ITIO, Universidad Politécnica de Valencia, Esp.
- Grimaldi, R and S. Torrasi (2001).** “Codified-tacit and general-specific knowledge in the division of labour among firms: A study of the software industry”. *Research Policy*, Num. 30. pp. 1425-1442.
- Hernández, C., y L. M. Sánchez (2003).** “Aprendizaje Tecnológico y Dinámica Industrial”, en J. Aboites y G. Dutrénit (edts.), *Innovación, Aprendizaje y Creación de Capacidades Tecnológicas*, UAM, México.
- Hobday, M. (1995).** *Innovation in East Asia*. Edward Elgar, UK.

- ..... (1996). “Complex System vs Mass Production Industries: A new Innovation Research Agenda”. *CoPS*, Num. 5. SPRU, UK.
- ..... (1998). “Product Complexity. Innovation and Industrial Organisation”. *CoPS*, Num. 52. SPRU, UK.
- ..... (2000). “East Versus Southeast Asian Innovation Systems: Comparing OEM and TNC led Growth in Electronics”, in Kim y Nelson, *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 (edts), *Sistema Nacional de Innovación: Temas Para el Debate en México*, UAM, México.
- Hobday, M. and Tim Brady (1996).** “Software Processes and Practices in Complex Product Systems: An Exploration of the Flight Simulation Domain”. *CoPS*, Num. 7. SPRU, UK: 1-3.
- ..... (1997). “A fast Method for Analysing and Improving Complex Software Processes”. *CoPS*, Num. 13. SPRU, UK.
- Horowitz, B., et al (2005).** *Embedded Software Design and System Integration for Rotorcraft UAV using Platforms*. EECS, University of California, Berkeley, USA.
- Hualde, A. y R. Gomis (2007).** “Pyme de software en la frontera norte de México: Desarrollo empresarial y construcción institucional de un cluster”, en *Problemas del Desarrollo*. Revista Latinoamericana de Economía. Vol. 38, Núm. 150, julio-septiembre.
- INEGI (2004).** *Censos Económicos 2004*. INEGI, México.
- INEGI-SELECT-IDC (2003).** “Mercado Mexicano de TIC 1998-2003”, *SELECT*. México.
- ITAM (2005).** *La Industria de Servicios de Software en México: Diagnóstico, Prospectiva y Estrategia*. Centro de Estudios de Competitividad, ITAM; Select. México.
- Jenkins, M. (2007).** “Comparación de las iniciativas latinoamericanas para mejorar la industria del software”, presentación en el *XXXIII Conferencia Latinoamericana de Informática*, 9-12 oct., Costa Rica.

- Jones, O. and M. Craven (2001).** “Absorptive Capacity and New Organizational Capabilities: A TCS Case Study”. Working Paper. *The Business School of the Manchester Metropolitan University*, UK.
- Jordan, G. and E. Segelod (2006).** “Software innovativeness: outcomes on project performance, knowledge enhancement, and external linkages”, in *R&D Management*, Num 36, 2. p. 127-142.
- King, G., R. O. Keohane y S. Verba (1990).** *El Diseño de la Investigación Social. La Inferencia Científica en los Estudios Cualitativos*. Alianza Editorial, México.
- Lall, S. (1992).** "Technological Capabilities and Industrialization", en *World Development*, Vol. 20, Num. 2, pp. 165-186.
- ..... **(2000).** “Technological Change and Industrialization in the Asian Newly Industrializing Economies: Achievements and Challenges”, en Kim y Nelson, *Technology, Learning, & Innovation Experiences of Newly Industrializing Economies*, Cambridge.
- Lane y Lubatkin (1998).** “Relative absorptive capacity and interorganizational learning”, in *Strategic Management Journal*, Vol. 19, Num. 5. pp. 461-477.
- Lane, P. J., B. Koka y S. Pathak (2002).** “A Thematic Analysis and Critical Assessment of Absorptive Capacity Research”, in *Academy of Management Proceedings*, Arizona State University, USA.
- Langlois, R. N., and David C. Mowery (1996)** “The Federal Government Role in the Development of the American Software Industry: An Assessment,” in David C. Mowery, ed., *The International Computer Software Industry: A Comparative Study of Industrial Evolution and Structure*. Oxford University Press, USA.
- Lakhani, K. and E. von Hippel (2000).** *How Open Source software works: “Free” user-to-user assistance*. MIT Sloan School of Management, Working Paper #4117, US.
- Leamer, E. and Michael Storper (2001).** “The Economy Geography of the Internet Age”, NBER Working Paper, Num. W8450. *National Bureau of Economic Research*. August. ([www.nber.org/papers/w8450](http://www.nber.org/papers/w8450))
- Lee, T. W. (1999).** *Using Qualitative Methods in Organizational Research*. Sage Publications, USA.

- Lenox, M. y A. King (2003).** “Prospects for Developing Absorptive Capacity through Internal Information Provision” in *Strategic Management Journal*, August.
- Lerner, J. and Jean Tirole (2002).** ‘Some Simple Economics of Open Source’. *The Journal of Industrial Economics*. Volume L, No. 2. June. USA.
- Lundvall, B. (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.
- Lund Vinding, A. (2004).** “Interaction between firms and Knowledge institutions”, in *Research on Technological Innovation and Management Policy*, Volume 8. p. 257–283.
- Malerba, F. (1993).** “Technological Regimes and Firm Behavior”, in *Industrial and Corporate Change*, Vol. 2, Num. 1, Oxford University Press, USA.
- ..... **(1999).** *Sectoral Systems of Innovation and Production*. CESPRI-Bocconi University, Milan, Italy.
- Malerba, F. y Luigi Orsenigo (1990).** “Technological Regimes and Patterns Innovation: a Theoretical and Empirical Investigation of the Italian Case”, in Heertje A. (ed.), *Envolving 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.
- Markose, S. M. (1996).** “*The New Evolutionary Computational Paradigm of Complex Adaptive Systems*”. CoPS, SPRU, UK.
- Maroto, C. y Jorge Zavala (2002).** “La Industria del Software en México”. *El Mercado de Valores*, Núm. 12, Diciembre. México.

- Maxwell, (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.
- Monge, J. (2002).** *Industrial Upgrading en el Sector Software en Costa Rica.* Fundación Ford-FLACSO.
- Mowery, D. (1996).** *The International Computer Software Industry. A Comparative Study of Industry Evolution and Structure.* Oxford University Press, USA.
- Mochi, P. (2006).** *La industria del software en México en el contexto internacional y latinoamericano.* UNAM. México.
- Mochi, P. y A. Hualde (2008).** “Oportunidades y desafíos de la industria del software en México”, en *Oportunidades y Desafíos de la Industria de Software en América Latina*, CEPAL, Santiago, Chile.
- National Science and Technology Council (1999).** *Information Technology for the Twenty. First Century: a Bold Investment in America’s Future.* USA.
- Nonaka, I. y H. Takeuchi (1994).** *La Organización creadora de Conocimiento. Cómo las Compañías Japonesas crean la Dinámica de la Innovación.* Oxford, Londres.
- OECD (1993).** “El software”, en *El Manual de Frascati*, México: 215-219.
- ..... (1998). *The Software Sector: a Statistical Profile for Selected OECD Countries.* Paris, Fr.
- ..... (2003). *Report of the OECD task force on software measurement in the national accounts*, March.
- ..... (2004). *Information Technology Outlook 2003.*
- ..... (2006). “The Social Dimension of Environmental Policy”, *Policy Briefs*. Paris.
- ..... (2007). *Information Technology Outlook 2006.*
- O’Dell, C., Susan Elliott and Cindy Hubert (2000).** *Knowledge Management. A Guide for your Journey to Best-Practice Processes.* American Productivity & Quality Center, USA.
- Padilla y Martín (2003).** “Tremec-Chrysler: Una experiencia exitosa de innovación”, en *Espacios*, Vol. 24, Núm 3.
- Pavitt, K. (1984),** “Sectorial Patterns of Technological Change. Towards a Taxonomy and Theory”, *Research Policy*, Num. 13: 343-374.

- Perez, C. (1986).** “Las Nuevas Tecnologías: Una Visión de Conjunto”, en Ominami, C., (ed.) *La Tercera Revolución Industrial. Impactos Internacionales del Actual Viraje Tecnológico*, GEL, Argentina.
- ..... (2001). “Cambio Tecnológico y Oportunidades de Desarrollo como Blanco Móvil”, en *Revista de la CEPAL*, Núm. 75, Chile: 115-136
- ..... (2002). *Technological Revolutions and Financial Capital*. Edward Elgar, UK.
- ..... (2003). “Revoluciones Tecnológicas, Cambios de Paradigma y Marco Socioinstitucional”, en J. Aboites y G. Dutrénit (edts.), *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.
- Piore, M. (2003).** *The limits of the Division of Labor in Design and the Prospects for Off-Shore Software Development in Mexico*. Mimeo, Department of Economics, MIT, USA.
- Piore, M., A. Schrak y C. Ruiz Durán (2005).** “Los retos para el desarrollo de la industria del software”, en *Comercio Exterior*, Vol. 55, Núm. 9, sep. México.
- Prencipe, A. (1998).** “Modular Design and Complex Product Systems: Facts, Promises, and Questions”, in *CoPS*, Num 47, SPRU, UK.
- Programa Especial de Ciencia y Tecnología del Estado de Jalisco, PECYTJAL (2001-2007).**
- Quintas, P. (1995).** “Software Innovation in the Context of Complex Product Systems”. *CoPS*, Num. 1. SPRU, UK.
- Rosenberg, N. (1982).** *Inside the Black Box: Technology and Economics*. Cambridge University Press, Cambridge.
- Ruiz Durán, C. (2004).** *Potencialidades de las Entidades Federativas para Desarrollar Núcleos de Economía Digital*. Facultad de Economía, UNAM, México.
- Sallstrom, L y R. Damuth (2003).** *El Papel Fundamental de la Industria del Software en el Crecimiento Económico. Foco: México*. Secretaría de Economía, México.
- Sanchez, R. y J. T. Mahoney (1996).** “Modularity, Flexibility, and Knowledge Management in Product and Organization Design”, in *Strategic Management Journal*, Vol. 17 (Winter Special Issue), pp. 63-76.

- Secretaría de Economía (2005).** *Programa para el Desarrollo de la Industria del Software, v1.2.* PROSOFT. México. ([www.Software.net.mx](http://www.Software.net.mx))
- ..... (2006). *Programa para el Desarrollo de la Industria del Software, v1.3.* PROSOFT. México. ([www.Software.net.mx](http://www.Software.net.mx))
- Scherer, F. M. (1982).** “Demand Pull and Technological invention: Schmoockler revisted”, in *The Journal of Industrial Economics*, Vol. XXX, Num. 3. March.
- Schmoockler, J. (1972).** *Patents, Invention, and Economic Change: Data and Selected Essays.* Harvard University Press, Cambridge.
- Segelod, E. and G. Jordan (2004).** “The Use and Importance of External Sources of Knowledge in the Software Development Process”, in *R&D Management*, 34, 3, 239–252.
- Senker, J. y W. Faulkner (1996).** “Networks, Tacit Knowledge and Innovation”, in Coombs, R., Al Richards, P. Saviotti y V. Walsh (eds.), *Technological Collaboration. The Dynamics of Cooperation in Industrial Innovation*, Edward Elgar, UK: 76-97.
- Shapiro, C. (2006).** “Israel: Un nicho de oportunidad”. Presentación en *El Seminario: Israel: Oportunidades de Negocio en el Marco del TLC*, Banco de Comercio Exterior de México, 27 de sept. México.
- Shy, Oz (2001).** *The Economics of Network Industries.* University of Cambridge, UK.
- Stadler, M. (2005).** *Demand pull and technology push effects in the quality ladder model.* Research Paper, University of Tübingen, Germany.
- Stake, R. E. (1998).** *Investigación con Estudios de Casos.* Ediciones Morata, S. L. España.
- 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.
- Taminiau, Y. (2006).** “Beyond known uncertainties: Interventions at the fuel–engine interface”, in *Research Policy*, Num. 35. p. 247-265.
- Taylor, S. J. y R. Bogdan (1992).** *Introducción a los Métodos Cualitativos de Investigación. La Búsqueda de Significados.* Ediciones Paidós, España.
- Tapscott, D. y Art Caston (1995).** *Cambio de Paradigmas Empresariales.* Mc Graw Hill, Colombia.

- Torrise, S. (1996).** “Firm Specialization and Growth: A Study of the European Software Industry”, in *Luic Papers*, Núm. 35, Serie Economía, Fra.
- UAM (2004).** *Estudio para Determinar la Cantidad y Calidad de Recursos Humanos Necesarios para el Desarrollo de la Industria de Software en México*. SE, México.
- Van den Bosch, F., H. Volberda and M. de Boer (1999).** “Coevolution of Firm Absorptive Capacity and Knowledge Environment: Organizational Forms and Combinative Capabilities”, in *Organization Science*, Vol. 10, Num. 5, September-October. Pp. 551-568.
- Van de Bosch, F., R. Van Wijk and H. Volberda (2002).** *Absorptive Capacity: Antecedents, Models and Outcomes*, Erasmus University Rotterdam, Net.
- Van den Ende, J. and W. Dolfsma (2004).** “Technology-push, demand-pull and the shaping of technological paradigms - Patterns in the development of computing technology”, in *Journal of Evolutionary Economics*, Vol. 15, Num 1. March. pp. 83-99.
- Vera-Cruz, A. O. (2004).** *Cultura de la Empresa y Comportamiento Tecnológico*. Porrúa-UAM, Méx.
- Villavicencio, D. y Mario Salinas (2002).** “La gestión del conocimiento productivo: las normas ISO y los sistemas de aseguramiento de calidad”, en *Comercio Exterior*, Vol. 52, Núm.6, junio, México, pp. 508-520.
- Villavicencio, D. y Rigas Arvanitis (1994).** “Transferencia de tecnología y aprendizaje tecnológico”, en *El Trimestre Económico*, Vol. LXI, Núm. 242, abril-junio, México, FCE, pp. 257-279.
- Von Hippel, E. (1988).** *The Sources of Innovation*. Oxford University Press, US.
- ..... (2000). *Horizontal innovation networks-by and for users*. MIT Sloan School of Management, Working Paper #4366-02, June, US.
- ..... (2005). *Democratizing Innovation*. MIT Press, USA.
- Wenger, E. (2000).** “Communities of Practice: The key to Knowledge Strategy”, in Lesser, et al., *Knowledge and Communities*. Butterworth Heinemann, USA.
- Yarime, M. and Y. Baba (2005a).** *Dynamics of Embedded Software Development: Coevolution of OS Standards and Community Networks in Japan*. Workshop on International Computer Software Industry, Montreal, C.



- Yarime, M. y Y. Baba (2005b).** “El desarrollo de software incorporado en Japón: co-evolución de normas en sistemas operativos y redes comunitarias”, en *Comercio Exterior*, Vol. 55, Núm. 9, sep. México.
- Yin, R. K. (1993).** *Applications of Case Study Research*. Sage Publications, USA.
- ..... **(1994).** *Case Study Research: Design and Methods*. SAGE Publications, USA.
- ..... **(2003).** *Case Study Research: Design and Methods*. 2th Edition, SAGE Publications, USA.
- Yoguel, G., M. Novick, D. Milesi, S. Roitter y J. Borillo (2004).** “Información y Conocimiento: la Difusión de las Tecnologías de Información y Comunicación en la Industria Manufacturera Argentina”, en *Revista de la CEPAL*, Núm. 82, Chile: 139-156.
- Zanfei, A. (1994).** “Technological Alliance Between Weak and Strong Firms: Cooperative Ventures wit Asymetric Competences”, in *Revue D'Économie Industrielle*, Num. 67, France.

## ANEXO I

### EL MODELO DE DESARROLLO DE SOFTWARE EN EL SEGMENTO 'HECHO A LA MEDIDA'

Se han hecho algunos esfuerzos para definir modelos de innovación en la industria de software tomando como base los modelos tradicionales de innovación de producto.<sup>1</sup> Segelod y Jordan (2004) presentan un modelo lineal basado en cuatro etapas: idea, decisión, desarrollo y comercialización.<sup>2</sup> Este modelo tiene como referencia a los modelos de desarrollo de software, uno de los más difundidos es el modelo de desarrollo y diseño de software descrito por Sommerville (1992), quien presenta 5 etapas: definición de requerimientos, diseño, implementación y prueba, integración, y operación y mantenimiento. Tomando como referencia estos modelos y con base en la evidencia recopilada a través de los estudios de caso, en esta tesis se describe el modelo de desarrollo para el segmento de software hecho a la medida.

Como se mencionó en capítulos anteriores, estas empresas no cuentan con departamentos o áreas funcionales formalmente, en particular con el área de Investigación y Desarrollo (I+D). La organización del trabajo está basada en grupos de trabajo y en un esquema multitareas. Por otro lado, el desarrollo de nuevos programas de software (aplicaciones) depende en gran medida de dos agentes: los usuarios y las comunidades de desarrolladores de software de fuente abierta, en las cuales son importantes los usuarios 'conscientes'.<sup>3</sup>

---

<sup>1</sup> La innovación está acotada al desarrollo de *nuevos programas* de software comercial (soluciones de software hechos a la medida), procesos organizacionales o servicios, y no a *lenguajes de programación* o *herramientas de software*.

<sup>2</sup> Estos autores mostraron que, en la industria de software, la fuente externa más importantes en el proceso de innovación es el usuario. El estudio se hizo en micro, pequeñas y medianas empresas que desarrollan software hecho a la medida ubicadas en varios países de Europa.

<sup>3</sup> Un 'usuario consciente' es uno de los agentes que integran una comunidad de desarrolladores de software de fuente abierta (además del desarrollador), generalmente tiene habilidades en programación y conocimiento de lenguajes y herramientas de programación. En la comunidad aporta ideas, conocimiento y en algunos casos define la viabilidad técnica de un proyecto de software (desarrollo de alguna solución, aplicación o herramienta).

El modelo sobre el cual las empresas en estudio basan sus actividades de desarrollo de software hecho a la medida consiste de 5 etapas: i) análisis, ii) diseño, iii) programación, iv) revisión, y v) implantación y despliegue.<sup>4</sup> La primera está asociada al análisis de los requerimientos y necesidades del cliente. La segunda consiste en saber qué programar a partir de las necesidades y requerimientos del cliente; el diseño comprende módulos, procesos, etc. Un buen diseño (si hay buena y suficiente interacción con el cliente) implica que, al momento de entregar el sistema funcionando no hay cambios sustanciales en éste. De forma que esta actividad es fundamental para las etapas subsiguientes, pues si no se hace un buen diseño el ciclo de “revisión-rediseño-reprogramación se vuelve vicioso”.<sup>5</sup> Por esta razón, en estas dos etapas la frecuencia de la interacción con los usuarios es alta.

La tercera etapa es exclusiva de las empresas desarrolladoras, implica el uso de lenguajes de programación, la reutilización y creación de códigos y herramientas de software, entre otros. El usuario no se involucra en esta actividad. La etapa de revisión implica asegurar la confiabilidad del sistema. Finalmente, la etapa de implantación consiste en pegar los módulos a cada proceso y hacer funcionar el sistema completo, mientras que el despliegue significa instalar el sistema en todos los ‘rincones’ de la empresa y capacitar a todos aquellos que usarán el sistema. Esta capacitación debe llevarse a cabo en corto tiempo pues es un indicativo de la efectividad de la *interfase técnica*.<sup>6</sup>

El modelo de desarrollo de software parte de una *necesidad* para llegar a consolidar una *solución*, en la cual se incluye el desarrollo, la estructura de costos y los tiempos.

En las diferentes etapas del modelo de desarrollo hay diferentes grados de interacción con el usuario. Por ejemplo, en las primeras dos etapas (análisis y diseño), el usuario es la fuente principal de información y en algunos casos también de conocimiento para el

---

<sup>4</sup> En cada una hay sub-etapas o procesos particulares como el de control de calidad, sobre todo cuando la empresa se encuentra en algún proceso de certificación como CMMI (Capability Maturity Model Integration), Personal Software Process (PSP), Test Software Process (TSP), ISO 15504, y recientemente el Modelo de Procesos para la Industria de Software (MOPROSOTF).

<sup>5</sup> Entrevista con líderes de proyecto y desarrolladores de software.

<sup>6</sup> Según líderes de proyecto de las empresas Manet y Velásquez, los parámetros de eficiencia de una interfase técnica son el tiempo en que se aprende a usar el sistema, una buena interfase técnica debe ser aprendida en el menor tiempo.

desarrollo de nuevos programas de software. Dependiendo de la complejidad del proyecto, la interacción podrá de mayor o menor intensidad. Si el proyecto es, en cierta medida, una replica de algún programa que ya existe en el mercado, el programa ya existente se adapta a la empresa usuaria y la intensidad será baja. Pero si el programa a desarrollar es nuevo, la interacción será intensa en las primeras etapas de desarrollo, sobre todo para sistematizar los requerimientos y necesidades del cliente.

En la tercera y cuarta etapa la intensidad de la interacción disminuye, ya que son actividades propias de la empresa proveedora. Pero en la última etapa la interacción también es alta pues está asociada a la implantación del sistema al interior de la empresa usuaria, y la actividad principal es la capacitación a los usuarios del sistema, la cual es mediada por el tipo de proyecto y usuario.<sup>7</sup> La capacitación es importante pues los sistemas son hechos a la medida y “nunca son tan intuitivos”.<sup>8</sup>

Como mencionan algunos de los líderes de proyecto entrevistados, ‘entre más interacción haya con el usuario más pura es la aplicación, ya que el usuario tiene el conocimiento de para qué se va usar’.<sup>9</sup> Por eso, el mayor grado de interacción entre la empresa de software y el usuario se da en la etapa de análisis y diseño.

---

<sup>7</sup> Por eso es importante la interacción con la empresa usuaria en la etapa de diseño, ya que en ésta se define quién va a operar cada proceso y qué rol tendrá cada actor en cada proceso. Las interfases se definen generalmente en el diseño y comprende ventanas, web, etcétera. Con base en entrevistas a líder de proyecto “no es posible diseñar 25 interfases distintas en un sistema, es decir, el usuario u operador debe aprender en un mínimo de tiempo, si es así, es una buena interfase...y cuando se cambia de usuario u operador, la capacitación será ‘rápida’ y ‘eficiente’, pero esto depende de la calidad y eficiencia de las interfases”.

<sup>8</sup> Entrevista con líderes de proyecto y desarrolladores de software.

<sup>9</sup> Al cliente se le entrega la *implementación* ya lista para hacer el *despliegue* en el momento de la entrega. Un buen *diseño* (si hay una buena y suficiente interacción con el cliente) implica que en el momento en que se entrega el *sistema* funcionando, el usuario no pide hacer cambios al sistema.