

Las fábricas de software en España. Organización y división del trabajo: El trabajo fluido en la sociedad de la información

Juan José Castillo*

Dossier

1. El trabajo del conocimiento en la sociedad de la información: los desarrolladores de software como analizador¹

Tanto en la investigación sociológica como en las ideas hechas sobre el futuro de las sociedades contemporáneas, uno de los tópicos más repetidos es el de que nos encaminamos hacia una ‘sociedad de la información’. Sociedad que es descrita, en más de una ocasión, como algo por venir, una tendencia emergente o imponente, que como una realidad consolidada. Y, sin embargo, el imaginario sobre esta sociedad de la información produce, cada vez más, estudios, propuestas, investigaciones sociales, y políticas de producción de la sociedad, tanto, en lo que más cerca nos concierne, para los territorios, cada vez más amplios, de la Unión Europea, como de nuestro país, España, y de las formaciones sociales que lo configuran y traman. Por otro lado, en el ámbito de las ciencias sociales, desde la sociología hasta la geografía, pasando, claro está

* Catedrático de Sociología y profesor de Sociología de la Universidad Complutense de Madrid Departamento de Sociología III (Estructura social). Facultad de Ciencias Políticas y Sociología. Universidad Complutense de Madrid. Endereço electrónico: jjcastillo@cps.ucm.es.

1 Este estudio de caso es una versión reducida y reelaborada de uno de los nueve estudios de caso incluidos en el Programa Nacional de Investigación, financiado por el Ministerio de Educación y Ciencia, cuya referencia es SEJ2004-04780/SOCI, diciembre 2004-diciembre de 2007: “Escenarios de vida y trabajo en la sociedad de la información”, del que es investigador principal el autor. La versión completa se ha publicado como libro: *El trabajo fluido en la sociedad de la información*, Buenos Aires y Madrid, Miño y Dávila, 2007.

por la economía o la psicología, el énfasis en la preeminencia del ‘trabajo inmaterial’, o, volviendo a un seminal concepto marxiano, el *general intellect*, ocupa miles y miles de páginas, de reflexión y de investigación de primera línea (SCARBROUGH, 1999; PYÖRIÄ, 2003; SELEIM & ASHOUR, 2004; LINDKVIST, 2005; SORENSON et alii, 2006; entre otros muchos). La promesa de mucha de esta investigación y de estas propuestas de políticas, especialmente en la Unión Europea, ocupa, al menos en las declaraciones de los responsables, un lugar estratégico. Tal es el caso de la Agenda de Lisboa que se propone convertir las economías europeas en las más competitivas del mundo. Más de una investigadora, ha propuesto una crítica de estos “mundos felices”, que llevan consigo un estudio y puesta en evidencia de la verdadera realidad actual, y de las tendencias que pueden llevarnos a predecir lo porvenir. Pero, además, han sugerido que este discurso *embellecido*, al que tanto han contribuido algunos *gurús* sociológicos, lo tomen los sindicatos y trabajadores al pie de la letra.

Por decirlo coloquialmente, que se les coja por la palabra, como una posibilidad más de la acción de los trabajadores: “los discursos sobre la economía del aprendizaje pueden ser estratégicamente utilizados por los sindicatos, los formadores de los trabajadores y otros actores del lugar de trabajo, para una revitalización de la regulación sociocultural del trabajo” (CASEY, 2004, p. 607)². Un argumento este que se apoya en esa promoción especular de un “trabajador ideal”, que también ha sido desplegado, en la misma dirección, tanto crítica como de aprovechamiento discursivo, por Ilona Kovács, como por nosotros mismos (CASTILLO, 1998, 2005).

Ahora bien, en este contexto, una realidad se impone como primer punto de partida para este estudio de caso: “el sector de la producción de bienes inmateriales”, para decirlo en una formulación certera del maestro Arnaldo Bagnasco, sector en el que incluye, claro está, la producción de software, es emblemático de las tendencias actuales, tanto de la sociedad como del trabajo:

2 “Learning economy discourses may be strategically utilized by trade unions, worker educators and other workplace actors in a revitalization of the socio-cultural regulation of work”. Dentro de un epígrafe sobre *Possibilities for worker action*.

“una empresa que produce bienes inmateriales puede conseguir ser mucho más elástica, capaz de adaptarse y de adherirse con mayor facilidad a los mercados móviles de la ganancia a corto plazo típicos de la era de la globalización” (BAGNASCO, 1999, p. 142 e 143). Y, con ella, una cuestión estratégica que ha articulado y organizado nuestra investigación y reflexión: ¿cuál es la realidad y el futuro de estos trabajadores del conocimiento, emblemáticamente aquí representados por los desarrolladores de software, por la producción de programas informáticos?

¿Van estos trabajadores, ahora identificados como un colectivo disperso en localizaciones a veces distantes entre sí cientos o miles de kilómetros, a sufrir – como brillantemente argumenta Christopher May (2000) – los mismos efectos que sufrieron con anterioridad otros trabajadores de la manufactura con bajas calificaciones? Por supuesto, esta es una primera formulación que, como se verá en lo que sigue, debe, necesariamente, hacerse mucho más compleja, tanto en los términos teóricos que la sustentan, como en su reciente evolución histórica.

Una muestra de esa complejidad necesaria es el planteamiento de Harry Scarbrough, quien muestra en su trabajo como las tendencias sociales de fondo actuantes en nuestras sociedades han estimulado, también en el trabajo intelectual, en el trabajo del conocimiento, la codificación de los saberes tácitos y su mercantilización, es decir su transformación en mercancía, *commodity* (SCARBROUGH, 1999, p. 14). Lo que, como veremos al estudiar el papel que ha jugado la estandarización de procedimientos, o la introducción de requerimientos y métricas tales como las normas ISO o las certificaciones CMMI, tiene una especial relevancia para el análisis de la constitución y funcionamiento del trabajador colectivo de la producción de software.

2. Nuestro perfil epistemológico, nuestro abordaje

El estudio de las tendencias en la evolución de la *fabbrica del software*, como la denominaron tempranamente los investigadores italianos, han venido analizándose en la literatura sociológica desde hace más de tres décadas, con un énfasis especial en las

formas que adoptaba la organización del trabajo, la división de la inteligencia aplicada a la producción, la reorganización empresarial (MANACORDA, 1976, 1984; KRAFT, 1979; GALLINO, 1983; PERRING, 1983; BOLOGNANI & CORTI, 1984; GAMELLA, 1985; PERULLI, 1988). Y esas investigaciones, que se llevaron a cabo en contextos que han mudado sustancialmente, tanto en el entorno social como en el tecnológico, al igual que en la capacidad de organización de la producción, nos pueden, sin embargo, aportar, en cuanto al enfoque de fondo de su estudio iluminaciones de interés.

Nosotros mismos, ya a finales de la década de los ochenta y principios de los noventa del pasado siglo, hemos dedicado alguna atención al sector, aunque nos centráramos, sobre todo, en el uso de la informatización en muy distintos tipos de empresas, sectores y procesos. Desde los supermercados hasta los estudios de arquitectura, pasando por el diseño y corte asistido por ordenador en el sector de la confección, entre otros (CASTILLO, 1989). Ahora bien, el abordaje del presente caso de estudio, se apoya en los planteamientos y resultados de la investigación llevada a cabo en los últimos años dentro de la red TRABIN (CASTILLO, 2005).

Por ello trata, necesariamente, un conjunto de problemas nuevos, junto con el planteamiento de la necesidad de renovar las herramientas conceptuales e interpretativas de las ciencias sociales del trabajo, para poder abordar, a la altura de los *tiempos modernos*, la “industria del software”, y sus trabajadoras y trabajadores, hoy en día. Para enmarcarlos y situarlos, hemos creído conveniente destacar en este apartado algunos rasgos fundamentales que caracterizan nuestro abordaje.

2.1. Estudiar lo realmente existente, lo visible y lo invisible

El primero de ellos, es el de tratar de llevar a cabo un estudio de terreno, teóricamente orientado, capaz de separar lo que *debe ser de lo que es*. Dicho en los términos ya acuñados y probados de la ergonomía, y de la antropotecnología de Alain Wisner, se trata de mostrar no sólo el trabajo y la organización del mismo *teórica* o *prescrita*, sino la actividad y la organización *real*.

En la investigación sobre el desarrollo de software, la construcción de programas, el ciclo de vida, como se le llama en la profesión en España, que comprende desde la toma de requisitos del “cliente”, el diseño, la arquitectura, el análisis funcional, las pruebas parciales y de conjunto, la prueba, la aplicación y el mantenimiento, es especialmente indicado esta manera de mirar. Porque, en numerosas ocasiones, como han analizado con brillantez los investigadores daneses Hansen, Rose y Tjornehoj (2004), para un conjunto de 322 investigaciones sobre métodos de mejora de los procesos de desarrollo de software, lo que predomina es más la prescripción, que la descripción o la reflexión³. Su conclusión no puede ser más esclarecedora: la inmensa mayoría de los artículos u obras analizadas dicen como deben ser las cosas, pero no necesariamente como son. Incluso, para destacar su argumento, en la versión en documento de trabajo, los autores juegan con los tipos de imprenta para darle un tamaño gigante a la “prescription” frente a la “description”, reduciendo a un tamaño minúsculo la “reflection”.

Por ejemplo, estudiar los “problemas reales de los equipos virtuales”, es una condición –a nuestro juicio- indispensable para poder dar cuenta de los cambios reales que están teniendo lugar en estos procesos productivos y en la sociedad. Aunque ello haga, naturalmente, más complejo tanto el abordaje teórico, como los temas abordados. Haciendo aparecer a los actores sociales en formas invisibles ante la mirada apresurada, de gestión de los sistemas de producción, o de formas inesperadas (para ese abordaje apresurado) de resistencia o sometimiento en el trabajo. O de creatividad (HUGHES *et alii*, 2001, p. 53)⁴.

Esta voluntad de reconstruir “las situaciones reales de trabajo”, lo realmente existente, es una necesaria marca epistemológica en el caso del software, puesto que es más que habitual, no sólo la generalización, con escaso fundamento, respecto al propio ‘sector’, sino su transferencia a los cambios globales de la sociedad como un

3 El campo de investigación, dicen, esta dominado por un enfoque, el capability maturity model (CMM). Sobre esto volvemos más adelante.

4 La investigación concreta, en este caso, muestra que los principales asuntos de gestión permanecen, como siempre, que no desaparecen con los ‘virtual teams’ ni con la ‘virtual organisation’.

todo. Y si no, véase lo que dicen dos catedráticos, uno de Harvard y otro de Berlín⁵, en un elegíaco prologo al libro éxito de ventas *Secrets of software success*:

“en el centro de este libro, sin embargo [en relación a otros problemas de gestión, según el sector: el software empaquetado, las soluciones empresariales y los servicios profesionales], está un muy diferente enfoque que demanda esta industria a la gestión de recursos humanos. Las jerarquías rígidas de la era industrial; los caminos de carreras largas, y así sucesivamente, no funcionan aquí. Este es, genuinamente, un mundo diferente” (HOCH, 2000, p. 6)⁶.

2.2. La reconstrucción de los procesos completos de producción

Para poder situar el estudio de los procesos actuales de trabajo en el desarrollo del software, de la construcción de programas informáticos, hemos continuado, reelaborado y adecuado, una metodología de trabajo que trata de colocar cada proceso productivo en su contexto más amplio, en la misma línea de abordaje que lo que ha llamado Burawoy (2004), con gran acierto, “the extended case method”. Colocando así los estudios empíricos, artesanos, minuciosos y detallados en un marco explicativo que les da sentido y profundidad.

Nuestro punto de partida, desde luego, habrá de ser el que hemos venido planteando en distintos estudios, desde los primeros años noventa, y que se plasma en un marco teórico, fundado en muy distintas investigaciones empíricas⁷. Marco que incluye la consideración de las *policy options*, las opciones de políticas razonables y razonadas, de crear entornos donde los círculos virtuosos de sinergias y recursos públicos y privados, puedan dar origen a distritos,

5 Los catedráticos son F. Warren McFarlan y Hermann Krallmann.

6 “At the center of the book, however [...] is a very different approach demanded by this industry to human resource management. Rigid hierarchies of the industrial age, long career paths and so on, don't work here. It is a genuinely different world”.

7 Un balance de los mismos puede consultarse en los trabajos recogidos en Castillo (1994) y en Castillo (1998)

clusters, desarrollos locales endógenos, que permitan garantizar una opción de desarrollo tanto personal como institucional y regional sostenible, y que transite por la *via alta* del desarrollo económico y social. Punto de partida que entronca con el *mainstream*, el marco de análisis actual en nuestra comunidad científica, que se apoya en trabajos muy semejantes a los que hemos desarrollado en nuestro equipo: como la división del trabajo entre empresas (GRIMALDI & TORRISI, 2001); los problemas de gobierno estratégico de las redes de empresas, (STURGEON, 2004; GEREFFI *et alii*, 2005); la evolución de la división del trabajo (CUSUMANO, 1992; BEIME, RAMSAY & PANTELI, 1998; CAPPELLI, 2001) etc. Esta literatura fundamenta el hecho de que aquello que formaba el núcleo central de la “nueva división internacional del trabajo” (FRÖBEL *et alii*, 1980), basado en la externalización de trabajo descualificado, se dobla, hoy en día con la posibilidad, y la realidad, desde luego, en este sector de la producción de software, de la externalización de trabajo cualificado, de trabajo inmaterial, de tareas que antes se consideraban sólo realizables en los países centrales (MAY, 2000; MIR, 2000; RIAN, 2000, 2004; ARORA *et alii*, 2001; BARRET, 2001; NICHOLSON & SAHAY, 2001; ILAVARASAN & SHARMA, 2003; PARTHASARATHY, 2004).

El ‘sector’ de la producción de software es, en este sentido, un terreno especialmente adecuado para analizar el contexto, las fuerzas que lo mueven, las transformaciones y las consecuencias para el trabajo, que están, según muestra la investigación social, mudando cada día tanto en la conformación de las empresas, como en la vinculación entre ellas. Para ello, nuestro enfoque, basado en la reconstrucción de los procesos completos de producción, es especialmente esclarecedor. Un punto de mira que ilumina aspectos descuidados desde otras perspectivas. Un abordaje semejante es el realizado por Miriam Glucksmann (2004), con el objetivo de situar los centros de atención de llamadas, los *call centres*, en lo que llama “total social organisation of labour”(p.798), equivalente a nuestro proceso completo de producción, elaborando un marco analítico, gracias al cual se pone en cuestión la inserción misma de estos centros (como podría hacerse con los centros de desarrollo de software, o las ‘fábricas de software’, en su caso), en un sector específico.

Antes al contrario, lo que Glucksman está proponiendo es un enfoque que no considera al centro de atención de llamadas como una unidad aislada, sino que pretende integrarla en el proceso de producción y de trabajo del que forma parte desprendida, externalizada, o subcontratada. Su minuciosa y sugerente discusión, evita el uso de términos como ‘supply chain’, para quedarse con un “‘overall processes’ of ‘provision and consumption’”, que le permite elaborar una serie de “call configurations”, que sirven para separar tipos, aunque ella no acepta llamarlos así, de centros de atención. En su conclusión se coloca al lado de nuestra posición metodológica, aún reconociendo las dificultades que implica: “los centros de atención de llamadas forman una parte, mas que un todo, de un conjunto organizacional, que lleva a cabo una etapa en una serie compleja de actividades interconectadas”.

3. La organización y división del trabajo en la producción de software: el foco de nuestro estudio

3.1. Programas informáticos: unas mercancías particulares...

Michael Cusumano (2004) en un espléndido libro orientado a “directores, programadores o emprendedores, o que quieren serlo”, comienza por destacar que producir software no es como cualquier otro negocio, como la fabricación de otros muchos bienes o servicios. Porque una vez creado, tanto cuesta hacer una copia, como un millón. Porque es un tipo de empresa donde el beneficio sobre ventas puede llegar al 99 por ciento. Porque es un negocio que puede cambiar, sin más, de fabricar productos a fabricar servicios.

Retengamos aquí, para lo que nos importa ahora, en el diseño de nuestra investigación, es decir, para aportar luz sobre el trabajador colectivo de la producción de software, sobre sus condiciones de trabajo, sobre sus posibilidades, sobre su eventual futuro, algunas distinciones producto-servicio que nos van a ser útiles. El desarrollo de software puede ser de productos estándar, de productos personalizados y únicos para una empresa, de servicios a distancia... La variabilidad y la diferencia de complejidad,

bajo el mismo rótulo, es abrumadora. No hay un producto software. Sino múltiples y variados.

Basta con asomarse a las páginas web de las muchas empresas que hemos consultado para comprender que, por otro lado, productos dedicados a un mercado restringido, y con la apariencia de ser un producto ‘empaquetado’, estándar, que puede ir dirigido, casi en exclusiva a un mercado restringido, a un colectivo profesional, a un tipo de diseño asistido por ordenador, a una específica gestión de personal..., nos dará complejidades y problemas muy distintos a la hora de constituir un colectivo de trabajadores, de analistas, de programadores, de jefes de aplicaciones... de posibles *fábricas de software*. De lo que el mismo Cusumano, un autor de referencia y solidez, llama la “actividad técnica más fundamental de las empresas de software: el desarrollo de software”. Muchos investigadores han llamado la atención hacia esta riqueza de figuras productivas y de vivencias y expectativas de trabajo, e incluso hacia las repercusiones en la vida privada y la organización del tiempo. Con un énfasis especial, precisamente, en los trabajadores del software cuyos puestos de trabajo se mueven entre “la rutina y los puestos del mayor nivel” (HYMAN, SCHOLARIS & BALDRY, 2005, p. 708)⁸.

3.2. ...en una nueva división internacional del trabajo

Por otro lado, la división internacional del trabajo, la fragmentación de los procesos de creación y desarrollo de programas informáticos, no es sólo una necesidad metodológica, sino que, por otra parte, es el punto de partida de una reflexión de más vasto alcance sobre el papel que juega la deslocalización de actividades y servicios en la actual configuración económica mundial. Para poder abordar el sentido y las tendencias de la propia organización del trabajo, de sus formas y características, tenemos que comenzar por visitar, en nuestro caso, las propuestas de la más reciente investigación. En ella la pregunta fundamental, desde el punto de vista de

8 Y no sólo como prototipos del trabajador del conocimiento. Su foco está centrado en la relación trabajo, casa, vida, y en como el largo brazo del trabajo modela esta última.

los países receptores de trabajo cualificado, y especialmente, de la fabricación de software, tal y como la formula con agudeza Prasad (1998) es: los trabajos cualificados, con perspectivas de carrera, con posible incidencia en el desarrollo local, que se pierden en el centro para los trabajadores, ¿se ganan en la periferia?

Para tratar esta cuestión, desde el punto de vista de los países desde donde estos trabajos ‘emigran’, varios programas de investigación han querido averiguar (por ejemplo, para la economía y la sociedad norteamericana) qué efecto tiene el desplazamiento de muchos servicios, fuera de sus fronteras, la pérdida de empleos que puede suponer. Uno de esos programas, el más desarrollado, es el llevado a cabo por el MIT, el Instituto Tecnológico de Massachusetts (GEREFFI & STURGEON, 2004; STURGEON & LEVY, 2005). Preguntándose, para poder poner en marcha políticas adecuadas, qué sucede con la emigración electrónica del trabajo del conocimiento, incluyendo, por supuesto la programación, comparando salarios entre origen y destino, v.g., Estados Unidos y la India, en relación al salario mínimo en ambos países. Y no debe olvidarse que, como ha señalado Hellander (2004) en un trabajo de referencia, la opción clásica entre el *make or buy*, entre hacer dentro o mandar hacer fuera, se dobla en el sector de software con una tercera opción: conectarse (HELLANDER, 2004)⁹.

E igualmente, el abordaje en términos de división del trabajo entre empresas, de distritos industriales, de *clusters*, ha puesto un gran énfasis en las perspectivas de desarrollo local y de *vías altas*, o *upgrading*. Una obra emblemática y destacada es, sin duda, *Local enterprises in the global economy. Issues for governance and upgrading*, editada por Hubert Schmitz en 2004. Sobre la base de investigaciones empíricas de largo alcance, y como presentación de programas de investigación de gran calado, se presenta una sistematización de

9 Este libro, que, como decimos en las primeras páginas, es un ejemplo excelente de investigación, presentado como tesis doctoral en Finlandia hace un recorrido epistemológico, real, impresionante: primero elabora un modelo teórico sobre la creación de valor; luego aplica el mismo a la realización de un estudio empírico exhaustivo; y, finalmente, rectifica y enriquece el modelo con los resultados de la investigación. Algo muy semejante a la *grounded theory*. De hecho su terminología coincide con ese enfoque.

las distintas posibilidades de organización de los sistemas locales de empresas, incluidas las de software, en una tipología que va desde las *redes* hasta la *jerarquía*, en función de la mayor o menor posibilidad de desarrollo autónomo o dependencia en la división del trabajo. Una forma no muy distante del continuo que nosotros identificamos como *empresas cabeza* y *empresas mano*.

Volver a estos esquemas de investigación y revitalizarlos permite a los investigadores, como veremos más abajo en detalle con el ejemplo de Bangalore en la India, el recurso a un conjunto de interpretaciones de gran complejidad, como es el papel institucional y de los gobiernos en el fomento del desarrollo de estos conglomerados locales virtuosos. El papel de la confianza y la negociación. El rol reservado a los propios trabajadores, a la formación y la Universidad. Y, además, permiten comparaciones de carácter estratégico que pueden poner en relación los modelos de desarrollo más exitosos, ya sea en Silicon Valley, Irlanda, Brasil o México¹⁰. Y, por supuesto, España.

La respuesta a algunas de las preguntas, de las grandes preguntas, de estos abordajes, enmarcan y dan sentido y alcance a la interpretación que parte de los procesos de producción, reconstruye la forma en que éstos marcan y condicionan la vida de las personas, y las tramas y expectativas posibles de las sociedades, locales, regionales o nacionales. ¿Existen *regiones adherentes*, como las ha calificado un investigador, ricas en conocimientos, en saberes, en experiencia, en *confianza*, en infraestructuras, en redes, en potencialidades?. ¿Pueden crearse por la intervención política planificada, uniendo recursos locales, iniciativas privadas, demandas sociales, voluntad de fabricar *trabajo decente* para la mayoría?.

¿Y puede hacerse todo ello (y cómo) tomando, precisamente, el sector del software y los servicios informáticos como referente? Para retornar a la pregunta de Prasad, que iniciaba este epígrafe, ¿los buenos trabajos que se pueden deslocalizar de los países centrales,

10 Ver el importante trabajo de Rasmus Lema (2005), sobre el papel de la eficiencia colectiva en la producción de software en Bangalore. Y el balance de Humphrey y Schmitz, "Chain governance and upgrading: taking stock", incluido en Schmitz (2004, pp. 349-381). Dayansindhu (2001) utiliza conceptos más amplios como *embeddedness*, para abordar la industria del software en contexto.

se mantienen y estabilizan como buenos trabajos en la ‘periferia’?¹¹. ¿En qué medida se pueden potenciar, en el entorno de una nación y sus diversas formaciones sociales territoriales, España en nuestro caso, recursos para el desarrollo local, a través de la implantación de *fábricas de software* en distintas localizaciones?¹² Este el marco de posibilidades analíticas que nos abre el estudio de los procesos completos de producción, dentro de la división internacional del trabajo, en su anclaje territorial y social, para el abordaje del estudio de la “industria del software”¹³.

3.3. *La India como terreno de reflexión y de problemas de investigación*

El desarrollo de la industria del software en la India, y otros servicios tecnológicamente avanzados, ha sido objeto de análisis y estudio sistemático, hasta el punto de que podríamos clasificar la evolución de las investigaciones y de sus preocupaciones fundamentales, como un magnífico espejo de la evolución de los problemas y evolución de la industria misma del software, e incluso de los avances de la acelerada dispersión internacional de la producción basada en el conocimiento¹⁴. Así, la tesis de la exportación de

11 Una mirada especial, desde América Latina, merecen los casos de distintos países, especialmente México y Argentina (Chudnowsky, López y Melitsko (2001); Novick y Miravalles (2002); CEPAL, 2003; López (2003); Foro de Software y Servicios Informáticos (2004); Carrera (2005); Ruiz Durán, Piore y Schrank (2005)).

12 Casos de este tipo son los que hemos tomado, tras un muestreo estratégico, como los estudios de caso sobre el terreno, que forman la parte central de esta investigación y de los que aquí recogemos una muestra significativa.

13 No podemos extendernos aquí en las referencias, pero hay que destacar la importancia de la obra de O’Riain (2004, especialmente), para Irlanda, porque, precisamente, partiendo de la industria del software ha reconstruido un análisis general, del desarrollo económico y social de este país. Ver también Hellander (2004), para Finlandia; Isaksen (2004), para Noruega; y Cumbers y McKinnon (2004).

14 A lo largo de todo el trabajo hemos ido, e iremos mencionando muy diversos estudios que, aunque puedan tener como centro la industria del software en la India, tienen, en muchas ocasiones un alcance mucho más general, estudiando problemas muy específicos, como la relación trabajo-vida, las formas de gestión de la información, la confianza como criterio fundamental en el recurso a las fuentes documentales, etc. Por ello se recomienda una consulta de las referencias incluidas en la bibliografía.

tareas descualificadas, también en el desarrollo de programas, lo mismo que en la fabricación de componentes electrónicos, podía ser sostenida sin grandes debates a principios de los años noventa¹⁵, lo que indica un alto grado de consenso, al menos en la interpretación... Estos argumentos, que seguían la estela del clásico libro de Fröbel *et allí* (1980), se doblaban con un argumento complementario: la exportación de trabajadores, estos sí cualificados al centro de la producción, ya fuera el Reino Unido o los Estados Unidos: el *bodyshopping*. Esta “nueva división internacional del trabajo”, se “caracterizaba por una fragmentación de los procesos manufactureros que se dispersan globalmente”, buscando, o “motivados por la necesidad de trabajo barato”, descalificado o semicalificado. Lo que se buscaba era disponer de vastos recursos humanos a bajo precio (LAKHA, 1994, pp. 383).

Pero la pregunta que adelantábamos en el epígrafe anterior, formulada por Monica Prasad (1998), de si lo que pierden los trabajadores del centro cuando el trabajo emigra, lo ganan los trabajadores de la periferia, a los que llega ese trabajo, despliega un conjunto de análisis que acaban en el centro de la división del trabajo. Los años noventa del pasado siglo contemplaron una serie de políticas locales que llevaron a una gran implantación local de multinacionales, al igual que a la creación de un ‘ambiente local’ que convierte a Bangalore, y a lo que se denominará ‘Silicon Plateau’, en una región *adherente*, codiciada por sus ventajas comparativas.

Entre 1990 y 1994, mil subdivisiones de empresas multinacionales en la India solicitaron y obtuvieron la certificación ISO. Para Prasad esta certificación, sirviera o no para estandarizar la calidad, para lo que si servía era como marca o garantía para poder vender a compradores lejanos, como podía ser entonces el esquivo mercado de la Comunidad Europea. Su argumento, central y muy matizado, es que estas normas, impuestas por el mercado y por la globalización de la producción y el consumo, trajeron como consecuencia una descualificación de los procesos de trabajo que se llevaban a cabo en la India. Llevaron a una “descualificación invisible”, que –afirma

15 El mejor ejemplo es sin duda Salim Lakha, 1994, “The new international division of labour and the Indian computer software industry”.

tras citar a empresarios entrevistados- podían no ser el objetivo final de los mismos, pero estuvieron entre las consecuencias no queridas, pero bien recibidas de la misma.

La norma de calidad ISO (y más aún, hay que adelantar) el modelo CMMI, que hoy es norma imprescindible para poder ser proveedor de software, tanto a las empresas, como a las instituciones, o a los llamados *system integrators*), ha contribuido –argumenta- a una taylorización del trabajo de programación y a una pérdida de control sobre el trabajo individual. La internacionalización de la producción, afirma, hace necesario para el comprador una estandarización que haga irrelevante el lugar en el mundo de su fabricación, lo que “lleva a la creación de estándares internacionales y normas, y, la introducción obligatoria de estas normas, reintroduce una dinámica de descualificación” (PRASAD, 1998, p. 431)¹⁶. Con estas técnicas de documentación y *good programming*, los puestos de trabajo quedan liberados del trabajador concreto, hace necesario menos trabajo, en términos cualitativos, y, por ende, produce más paro.

Prasad se apoya en los estudios que en la industria y en otras áreas productivas más tarde, tuvieron lugar con la introducción del TQM, Total Quality Management, o de las normas ISO, como requisito inexcusable para poder aspirar, las empresas subcontratadas, a ser proveedor de los grandes constructores del automóvil, por ejemplo. Y, como ha mostrado ampliamente la investigación, las normas de calidad han terminado, en efecto, por ser utilizadas más como una forma de control, que como una querida documentación y trazabilidad de los productos o servicios en aras de la calidad ofrecida al cliente¹⁷.

Ahora bien, esta interpretación, pocos años después, en 2003, es ya puesta en cuestión, identificando el tipo de productos y el avance o superación de la situación más baja en la “cadena de valor”, o, en nuestros propios términos, situándose las empresas indias más lejos de las *empresas mano* y más próximas a las *empresas*

16 Prasad, 1998, p. 431: “leads to the creation of international standards and norms, and mandatory adherence to these standards reintroduces a workplace dynamic of deskilling”. A una conclusión semejante llegan Beirne *et alii* (1998).

17 El trabajo más influyente y orientador de este análisis es el de Segrestin, 1997.

cabeza. Ocupándose de proyectos más complejos, que comprendan también la toma de requisitos en el cliente, el diseño, la arquitectura, etc. Esto es, en la clásica cascada con que se representa el proceso de producción de software, donde más arriba es más independiente, más cualificado como conjunto productivo: “en los dos últimos años hay evidencia suficiente para asegurar que las empresas indias están madurando y creciendo en su capacidad para ejecutar proyectos más amplios más complejos; al igual que están ejecutando partes con más alto valor añadido de tales proyectos (como la toma de requisitos y el diseño de alto nivel)” (ARORA *et alii*, 2001; HELLANDER, 2004).

Ese es el argumento central de la discusión que enfrentan Ilavarasan y Sharma (2003), preguntándose “Is software work routinized?”. Para responder a esta pregunta revisan una por una estas cuestiones que bien pueden aplicarse a cualquier estudio sobre la producción de software, para averiguar en qué medida esta fragmentada o descualificada:

- a) “Los trabajadores del software están claramente divididos en trabajadores de la concepción y de la ejecución, tales como diseñadores, codificadores o probadores”.
- b) “Los trabajadores de ejecución no participan en la ejecución del proyecto”.
- c) “Los trabajadores implicados en un módulo, no tienen conocimiento de los restantes módulos en el mismo proyecto”.
- d) “Los requerimientos de formación son diferentes para las distintas categorías de trabajadores”.
- e) “Las oportunidades de carrera están restringidas para los trabajadores de ejecución”.
- f) “Los procedimientos de certificación potencian el control directivo” (ILAVARASAN & SHARMA, 2003).

A cada una de estas cuestiones, los autores responden, por medio de estudios empíricos por la negativa. Y en su conclusión destacan el hallazgo de una división camaleónica del trabajo; un

trabajo en equipo, incluso en equipos virtuales; los trabajadores gozan de una información simétrica; el mismo nivel de formación se da a todos; no hay barreras para las carreras; y el control está distribuido. Con un panorama tan idílico, pueden terminar afirmando que ni el trabajo del software está “rutinizado”, ni parece poder serlo, ni lo será en el futuro. Algo que, desde luego, muchas otras investigaciones matizan, e incluso ponen radicalmente en cuestión¹⁸. Mientras que la investigación más reciente apuesta, de nuevo, por la constatación de un desarrollo que abandona, en Bangalore, los servicios de software basados en su baja localización en la cadena de valor, afirmando que “estos desarrollos pueden muy bien marcar [la aparición] de un naciente distrito tecnológico en red” (PARTHASARATHY & AOYAMA, 2006)¹⁹.

3.4. Los estándares como mecanismos de organización del trabajo

La introducción y el uso de estándares organizativos, tales como las normas ISO, SPICE, o la más conocida y promovida por el SEI, Software Engineering Institute, CMM (Capability Maturity Model), o la posterior CMMI, ha sido estudiado, en la literatura sobre la organización del trabajo del software como un instrumento privilegiado para conocer las tendencias del desarrollo de software hacia lo que ha venido llamándose *fábricas de software*²⁰.

Un caso ejemplar de este abordaje es el llevado a cabo por Paul Adler, en un minucioso y complejo estudio de caso llevado a cabo en una gran empresa norteamericana, tomando como terreno de estudio cuatro programas distintos, que se sitúan, dos de ellos

18 Una tesis más compleja e internacionalmente más explicativa en Mir *et alii*, 2000, “The codes of migration. Contours of the global software market”, sobre la polarización de calificaciones en y a través de las naciones-estados. Véase, igualmente, el artículo ya citado de Arora *et alii*, 2001, “The indian software services industry”.

19 Estos autores matizan bien que su investigación de campo justifica esta afirmación para la India, pero es dudoso que pueda generalizarse a otros contextos, dado el particular y específico caso de Bangalore.

20 Una referencia fundamental es Watts Humphrey (2002). En la literatura sociológica el análisis pionero e iluminador es de Segrestin (1997).

en el nivel 5 del modelo CMM, y los otros dos en el nivel 3. Adler ha interpretado su detallado trabajo de campo con una batería de problemas teóricos que ha resumido en distinta forma y modos, intentando, por un lado dar cuenta de la influencia que la introducción de estos estándares han tenido en la posible simplificación del trabajo, o, dicho más precisamente, en su ‘rutinización’, o taylorización, como dirá expresamente²¹. En una primera formulación, Adler toma como analizador la introducción del modelo CMM, con el fin de plantearse como pregunta central, “cómo cambia la disciplina del proceso el trabajo del software, esto es, la naturaleza de este trabajo, su organización, y, sobre todo, la experiencia del mismo” (ADLER, 2002, p. 24)²². Una pregunta que –como ya se ha manifestado- está también en el centro de nuestras preocupaciones.

Las conclusiones a las que llega Adler son matizadas, y su esfuerzo por mostrar los aspectos de cambio y pérdida de la autonomía de los desarrolladores, basándose en documentación muy sólida, y, sobre todo en una compleja trama de entrevistas transcritas y presentadas al lector, no deja, por otro lado, de manifestar tendencias contrarias, que implican una mejora que no se compadece con la versión simplista de la taylorización del trabajo, y de la descolectivización del trabajo de producción de software²³.

Ante las tendencias, y vivencias, contradictorias que aparecen en su estudio, Adler propone un recurso a la teoría para interpretar y dar sentido a las tendencias de fondo. Y, para ello, recurrirá a distintos paradigmas, que, finalmente interpreta, según lo que él mismo denomina una renovación de las teorías del *labour process*.

21 Conviene decir que hemos podido consultar este estudio y las reflexiones del autor, sus énfasis y articulaciones, con un detalle y minucia poco habitual en la investigación sociológica en distintas formulaciones, ya sea en textos publicados o en documentos no publicados, cuyas referencias se incluyen en la bibliografía.

22 “The central research question of this study is: how does process discipline change software work –the nature of that work, its organization, and above all, the experience of it”.

23 Por supuesto, los énfasis varían, y si en 2002, p.76, las conclusiones comienzan con una referencia, y cita expresa a la “revolución mental” taylorista, le siguen, ya en ese texto una larga serie de argumentos que matizan esa visión. Para acabar dando un visión de los resultados de la introducción del CMM, matizada y contradictoria.

Volviendo a Marx, y a una visión que la segunda generación de estudios de esta importante escuela ha dejado en la sombra. A esa visión la llama ‘paleo marxista’. Y la resume diciendo que deben contemplarse dos tendencias contradictorias, que son las que hacen emerger, en el caso del software un objeto poliédrico y, a veces, confuso. La primera tendencia, en su visión, reside en el hecho de que las fuerzas productivas tienden a hacer el trabajo y los requerimientos de calificación más complejos. También en el software, con la modernización que supone la introducción del modelo CMM, y los cambios que supone. Ello tiene consecuencias para los desarrolladores, los constructores, ampliando su objeto de trabajo, entre otros rasgos, que incluirá en lo que llamará ‘socialización de la producción’. Mejorarán así las tareas, se ampliarán, y no serán sólo ya “tirar código”, sino que abarcarán el ciclo de vida completo de la producción [lo que, por cierto, no hemos encontrado en nuestro trabajo de campo], entre otros aspectos.

Pero, una segunda tendencia puede incluso arruinar por completo estas beneficiosas transformaciones. Y estas contradicciones están ancladas en el proceso de valorización del capital: más ganancias, más mercado, más competencia. Las relaciones de producción, en suma. Así, en su estudio de caso, las empresas que adoptan el CMM, uno, mejoran costos, calidad y tiempos de entrega de los proyectos. Pero, dos, las contratendencias son los márgenes estrechos y a corto plazo de los beneficios y la competencia entre plantas²⁴.

Para nuestra investigación hemos tenido en cuenta algunos argumentos que suponen un reto en el análisis de la información recogida sobre el terreno. Una mirada que obliga a incluir en un lado u otro de las tendencias de la organización del trabajo, puede basarse en su concepto de ‘socialización’, que enumera en un conjunto de índices:

- a) la ampliación del objeto de trabajo. Consistiría en la no separación drástica entre ‘ejecutantes’ y ‘diseñadores’. Es más, como revelan los distintos títulos, *abstracts*, y énfasis de sus elabora-

24 Un buen resumen, en Adler (2004, pp. 253-254 y 255-256), en cada caso. Y con todo detalle, en Adler (2005).

ciones, un argumento importante de Adler es la afirmación de que el *objeto* de trabajo ha cambiado sustancialmente con la ‘modernización’ de los procesos.

- b) La ampliación del trabajador colectivo, especialmente fundado en la creación de equipos, la necesidad de trabajo en grupo, la consulta a los pares, etc.
- c) La profundización en la interdependencia en la colaboración.
- d) La socialización de las herramientas utilizadas en el trabajo.
- e) La socialización del desarrollo de las reglas y herramientas.
- f) La socialización de los procesos de formación y de cualificación²⁵.

En nuestra investigación, tanto desde la perspectiva de los técnicos y expertos, como de las asociaciones empresariales, de la Administración pública, e igualmente, obviamente, de las empresas, la adecuación de la producción a los modelos mencionados, tanto ISO, como CMM (Capability Maturity Model) o SPICE (Software Process Improvement and Capability dEtermination), es ya, hoy en día, un punto imprescindible de partida. Un condicionante. Un modelo, tendencialmente, cuya aplicación parece inevitable²⁶. Tomar los estándares como analizadores de la realidad del trabajo en el desarrollo de software permite abrir perspectivas, y descubrir aspectos de la realidad de la vida de trabajo difícilmente perceptibles

25 Una visión rica y matizada de este argumento, y de las contradicciones halladas en la realidad, en Adler, 2005, p. 421: “In sum, the CMM deepened rather than resolved the contradiction between use value and exchange value”, y el desarrollo en las pp. 422-426, “Bureaucracy: simultaneously mock, enabling, and coercive”.

26 Con esta afirmación pretendemos ahora, únicamente, llamar la atención hacia la similitud entre los argumentos desplegados no ya sólo en el lejano precedente del taylorismo o el fordismo, sino en el más reciente y comparable de la *producción ligera*, la *lean production*. Veáse, por todos, nuestro trabajo “Nuevos modelos productivos...”, recogido en Castillo (1998) *A la búsqueda del trabajo perdido*, pp. 2-84. CMMI, Capability Maturity Model Integration es un despliegue algo distinto del CMM original. En España, el apoyo público para la “Mejora de la calidad del software”, de la Secretaría de Estado de Telecomunicaciones y para la Sociedad de la Información, privilegia la implantación de CMMI y SPICE, sobre otras normas ISO. Veáse la Resolución de 5 de abril de 2006, BOE, n. 85, 10 de abril de 2006, p. 13910.

con los viejos arquetipos y conceptos ya usados, y gastados, de las ciencias sociales. Desde luego, uno de ellos, las múltiples formas que adquieren las resistencias del trabajo, de los trabajadores frente a los métodos de control.

Baste ahora el ejemplificarlo con un estudio, brillante y revelador, llevado a cabo por Nicholson y Sahay, y publicado en 2001, con un enfoque de investigación que sentimos como muy próximo, tanto por la estrategia, como por el énfasis en la construcción del modelo teórico a medida que la propia realidad investigada, la discusión con los interlocutores sociales, y la investigación publicada, lo funda y sostiene (NICHOLSON & SAHAY, 2001). El estudio trabaja sobre dos empresas de la misma propiedad, una en Inglaterra y la otra en la India. La externalización a la India lleva consigo la formación y aplicación de normas ISO estrictas que consiguen un colectivo de programadores indios muy disciplinados. Cuando la empresa ‘madre’ quiere hacer lo mismo en su casa central, y, ante las resistencias de los trabajadores cualificados de la misma contra lo que sienten como la ‘degradación’ de su trabajo, serán los trabajadores indios, ‘importados’ a Inglaterra los que sirvan de ariete que intente quebrar la resistencia de los programadores británicos.

Este es el aspecto más interesante en este contexto (el artículo es ejemplar en muchos otros). El utilizar a los indios de ‘Eron’, ya disciplinados, por así decir, en la *software factory*. Que están habituados a las normas ISO, a colocarse en el ‘ciclo de vida’ del producto que la organización les asigne. Que piden todo por escrito, y que documentan cuanto hacen. La empresa ‘Gowing’ intenta así cambiar las actitudes; o a los propios desarrolladores, “renovarlos”; o, en última instancia buscar que se vayan de la empresa. Estos programadores, que provenían de tres distintas empresas, con culturas muy distintas de la que la empresa quiere hacer ‘dominante’, se pueden ejemplificar con los “creativos” de RDC, que se destacan de sus compañeros, según el director de la empresa, hasta en el vestir, y que no dudan en demostrar su arrogancia. Gowing importó la “Eron Quality Methodology” y, con una larga historia de cambio, que los investigadores siguen con detalle, participando incluso en las reuniones y debates. El final

es una “mecanización y división del trabajo que trajo consigo el proceso [y que] fue rechazado por la contra-organización de los programadores de RDC” (*idem*, p. 38). Los autores, a través de la exploración de la compleja transferencia de normas y estándares no dudan en firmar, conectando con la mejor tradición de la investigación sobre el *labour process*, que la disciplina llevaba a una “producción fordista y de cadena de montaje” (*idem*, p. 39)²⁷.

3.5. Trabajo en equipo: colectivos reales y virtuales

Una de las piedras de toque en el análisis de la organización del trabajo de la producción de software es la constitución de los equipos o grupos de trabajo que, desde mediados de los años setenta, formaron parte de la panoplia de ‘nuevos métodos de organización del trabajo’. Por otro lado, el trabajo del conocimiento, el tratamiento de información, el carácter inmaterial de la materia prima que se utiliza en este proceso, obliga a considerar, y a plantearse, esas nuevas formas de organización, en contextos de alta tecnología, que permiten la circulación, la puesta en común, el compartir una intervención sobre un programa, en algo que ya no está condicionado por barreras físicas, geográficas, nacionales u otras. La literatura especializada se ha producido con abundancia sobre los llamados ‘equipos virtuales’, comunidades de práctica, o colectividades de práctica, que pueden estar a miles de kilómetros de distancia física, y, a veces, casi tanto en distancia cultural o de estilo organizativo.

Ahora bien, si esta forma de organización del trabajo tuvo una gran repercusión en los años setenta, no cabe olvidar que es necesaria una contextualización que devuelva las razones, los fines, los actores que intervinieron en aquella gran mutación organizativa que se pudo condensar en el lema, y en las instituciones correspondientes, de “mejora de las condiciones de trabajo”. La capacidad de

27 “In an organisation like Gowing where the main ‘product’ is software, the Eron methodology helped to disassociate the Gowing management from the RDC methods of development (described by Jones as ‘design on the back of a cigarette packet’) and towards the fordist production or assembly-line approach”.

contestación obrera en los centros de trabajo, la puesta en cuestión de la organización, llamada “científica”, del trabajo, el taylorismo y el fordismo, fue el motor principal de aquellos cambios. Y los ‘grupos semiautónomos de producción’ la forma más consolidada. El trabajo en grupo volvió a la palestra de la organización del trabajo, de forma masiva, en los años noventa, en un contexto notablemente diferente. Esta vez, por iniciativa empresarial, sin grandes negociaciones, y como colofón a lo que se llamó, de nuevo, el *one best way* de la época: la *producción ligera*, la *lean production*. Aún así, hoy en día, y para la constitución de una plantilla de análisis de las nuevas organizaciones de la producción, en nuestro caso, en el software, se hace imprescindible el revisar y poner al día el marco de análisis y las realidades de los equipos de trabajo, los pequeños colectivos de trabajadores, que pueden tener delegados en ellos, más tareas, más capacidad de organización, de iniciativa, de ruptura de la división del trabajo. Y verificar su existencia y condiciones de funcionamiento en la realidad de nuestras actuales *fábricas de software*. Porque, tras la misma denominación pueden encontrarse grandes novedades organizativas, o puras adecuaciones a la moda empresarial de turno. Esto es, que los problemas reales, se oculten tras presuntas realidades virtuales (HUGHES *et alii*, 2001; HAMDE, 2002; PRUJIT, 2003; MARTINS *et alii*, 2004).

En la realidad, y en la literatura, de las fábricas de software, los equipos virtuales son un reto, una posibilidad, no sin dificultades, como nos han transmitido las personas que hemos entrevistado en nuestra investigación. En una revisión de la literatura sobre el tema Martins y otros (2004) sustentan que son equipos virtuales aquellos en los “que los miembros usan la tecnología para interactuar entre sí, a través de fronteras geográficas, organizacionales, y otras, y que se están convirtiendo en lugar común en las organizaciones” (p. 805)²⁸. Un poco más sociológica, y menos mediada por la tecnología, es

28 Virtual teams, in which members use technology to interact with one another across geographic, organisational, and other boundaries, are becoming common place in organisations”. Ver también p. 807. Su *parti pris* les lleva a despreciar la comparación con los equipos *face to face* “puesto que estos últimos desaparecen.... Y en p. 823, “with rare exceptions all organisational teams are virtual to some extent”.

la definición que da Pyöriä (2003, p.167): los equipos virtuales se definen. 1) por el uso de las tecnologías de la información; 2) por el diseño independiente de partes importantes de su trabajo; y 3) por una formación profesional alta (PYÖRYÄ, 2003, p. 167)²⁹. El aspecto anti-taylorista, no individualizador, no rutinizable, también estará siempre presente, aunque puedan darse aspectos de intensificación del trabajo, de traslado de la vigilancia y disciplina al interior del grupo así constituido, y, por tanto a los propios programadores³⁰. Para nuestro estudio, en todo caso, quizá sea muy importante el encajar claramente el tipo de trabajo del conocimiento que es la producción de software, y las características más arquetípicas de los trabajadores que componen el colectivo³¹. Así, por ejemplo, una forma de aproximarse más a la verdadera realidad del trabajo, lejos de formulaciones ideales, que sólo son útiles para la reflexión fuera del mundo y del tiempo, es la hecha por Lindkvist (2005) entre “Comunidades de conocimiento y colectividades de conocimiento”. La diferencia es notable, aunque parezca de matiz: una colectividad es algo menos estable, más circunstancial, más contingente..., mucho más parecido a las realidades que hemos encontrado en nuestro trabajo (LINDKVIST, 2005, p. 1205).

29 Que puede completarse con la aportación de Kraan (2006), que busca una definición de equipo que permita un trabajo de “high performance and low stress”, como un grupo que, por supuesto usando las tecnologías de la información para superar barreras geográficas, etc., sea un “group of people who cooperate to attain a common goal”(p.4).

30 El argumento lo despliega coherentemente Pruijt (2003, p. 89), entre otros lugares. Hamde (2002), por su parte, ha reflexionado, con un recurso excelente a la obra de Simmel, sobre como una forma de organizar el trabajo que se impone como moda, inevitable, acaba siendo una institución cuando se toma por la única manera de hacer las cosas, sin discusión.

31 Sorenson *et alii* (2006) distingue las posibilidades de redes y transferencias en función de la complejidad del conocimiento de que se trate. En las “industrias basadas en el conocimiento moderadamente complejo”, entre las que parece incluir el software, puede ser necesaria la proximidad social. “Social proximity here refers to the distance between two parties in a social network”, y pone como ejemplo aquellas redes en que los integrantes nunca se vieron la cara, frente a los que tienen relación directa. Ver Leamer y Storper (2001); Coe y Bunnell, (2003); Scott y Storper (2003); Forsman y Solitander, (2003); Sturgeon, (2004); Sherer, (2005).

4. Las factorías de software en España

4.1. Introducción. Los datos generales: ¿un sector o una parte de diversos procesos de producción de bienes y servicios?

Aplicando nuestro propio enfoque interpretativo³², y en discusión con la literatura en este terreno, hemos de partir de algo evidente que nos obliga a construir nuestro objeto ‘material’ de estudio a la vez que lo definimos. Más precisamente: la fabricación, aplicación y mantenimiento de programas informáticos está sometida a una acelerada externalización, que era *antes* una función desarrollada internamente en las grandes empresas, y, por tanto, los datos que obtenemos de los registros estadísticos más solventes (la *Encuesta del Sector Servicios 2003* del INE, 2005, por ejemplo) podríamos decir que son un mínimo: aún quedan muchos servicios informáticos sin externalizar, o que se llevan a cabo por personal de las propias empresas, y que, por tanto figurarán en los respectivos sectores³³.

En la *Encuesta del Sector Servicios 2003*, la categoría 72 de la CNAE-93rev³⁴ recoge una serie de datos enormemente significativos, y que hemos explotado en una primera aproximación, lo que nos permite conocer datos generales *aproximados e ilustrativos* tanto de ocupación, como de distribución por categorías, sexo, tamaño de las empresas, etc. Por ejemplo, que de las 180.102 personas ocupadas en “Servicios Informáticos”, el 47,8 %, o sea, 86.400 son técnicos de software. Y de ellos sólo el 27,46 mujeres. Que de las 23.265 empresas registradas, 14.817, es decir el 63,7% tienen “menos de 2 trabajadores”.

32 Véase, más arriba, el apartado “La reconstrucción de los procesos completos de producción”.

33 De las entrevistas mantenidas con informantes privilegiados del sector, se colige que, en la propia definición del sector y de su representación empresarial, la definición sectorial es una baza en juego en la actualidad. El Ministerio de Industria español ha abierto una serie de mesas de reflexión sectorial, ‘Observatorios’, uno de los cuales, el *Observatorio Industrial de Electrónica, Tecnologías de la Información y Telecomunicaciones* presentó sus primeros resultados en una jornada de trabajo celebrada el 1 de diciembre de 2006. www.mityc.es/Observatorios/Observatorios/SectorElectronica.

34 Clasificación Nacional de Actividades Económicas 1993, revisada.

En un estudio de referencia (AETIC, 2005), y en los debates públicos mantenidos con ocasión de la jornada de difusión de los resultados del Observatorio Industrial de Electrónica, Tecnologías de la Información y Telecomunicaciones, en diciembre de 2006, precisamente, uno de los debates (y aportaciones singulares) fue un análisis detallado, y una propuesta de nueva clasificación estadística. Así se expresaba en las conclusiones de la reunión y en la nota de prensa publicada con ocasión de la misma: “la evolución reciente de las actividades del sector, hace difícil su encuadramiento en la actual Clasificación Nacional de Actividades Económicas, por lo que resulta necesario ampliar el marco clasificatorio del sector, para facilitar su evaluación y seguimiento”³⁵. Esto nos obliga a introducir una reflexión sobre el proceso mismo de creación del ‘sector’. Y nada puede mostrar, siquiera sea de forma sumaria, las líneas fundamentales de esta transformación en un caso verdaderamente significativo, tanto por la importancia económica, como por la relevancia, en la propia constitución de los actores de la “industria del software”. Se trata de la evolución del papel de la informática en el sector, o para el sector, bancario. Una investigación que queda ahora pendiente, por su envergadura misma. Un estudio de caso que debe comprender el proceso por el cual se externaliza de un conjunto de grandes empresas, la forma y el modo en que se lleva a cabo, las consecuencias para la recomposición del obrero colectivo del software, las transformaciones en el propio proceso de trabajo y en la constitución del trabajo fluido³⁶.

4.2. Cómo hemos construido nuestro abordaje cualitativo

Tras cuanto hemos venido argumentando en los apartados anteriores, y en la introducción a éste, podemos presentar de manera más sucinta, haciéndonos eco de lo dicho hasta ahora, las dos

35 Hay que destacar que incluso estos trabajos subestiman, a nuestro juicio, en gran medida las personas ocupadas en lo que llaman “hipersektor” de las TIC: 214.000 personas, según las conclusiones de los estudios, consultadas en enero y marzo de 2007, <http://www.mityc.es/NR/rdonlyres/01428F2C-0F92-4C0C-9F51-2470C89DCF9B/0/Conclusiones.pdf>.

36 Ver, por ejemplo, la excelente discusión de Panteli y otros, 2001, p. 3, nota 1.

perspectivas complementarias que fundamentan nuestro punto de partida en el trabajo de campo que se presenta a continuación. Para identificar los diversos trabajadores colectivos de la producción de software nos proponemos:

A) Seguir al producto-programa desde el diseño, y, sucesivamente, luego, todos los trabajos que sobre él se van ejecutando hasta su puesta en el mercado final, sea este una empresa individual o un conjunto, posible, de ellas. Incluidos, obviamente, los distintos métodos de prueba y error utilizados. Muchos de esos trabajos consisten en darlo a hacer a terceros. Los cuales hacen trabajos, muy calificados, sobre el producto, le añaden valor, y lo devuelven a quien lo encargó. Procesos que se parecen, en la forma, claro está, a una maquila: dar un producto a terceros que hacen trabajos (muy cualificados, insistimos) sobre el producto, le añaden valor y lo devuelven a quien lo encargó. Ahora bien, conviene subrayarlo, el hecho distintivo es que lo que se consideraba, en una forma anterior de la división entre empresas, como un *handicap*, es ahora la baza y característica más destacada: el trabajo del conocimiento es el rasgo central, el que más se destaca en la actual transformación de las sociedades, incluida la sociedad europea, por recordar un reto inmediatamente presente, con su ‘programa de Lisboa’.

En las clasificaciones y tipologías que florecieron en los años dorados de los *distritos industriales*, entre 1982 y 1992, aquellas empresas que sólo hacían *lavori*, para usar la mejor tradición italiana de la investigación, eran los últimos eslabones de la cadena... de valor. De la división del trabajo entre empresas ‘cabeza’ y empresas ‘mano’. A veces simples piezas destacadas de una cadena de montaje en el territorio.

En este proceso, actualmente, que puede hacerse a pocos, cientos, o miles de kilómetros de distancia, se crea o se transforma el programa, se pone en él elaboración, se invierte saber y conocimiento, tácito y explícito. Si se trata de un servicio: el producto se puede cambiar o transformar en la misma sede de la empresa que lo utiliza, su destinatario final. Sin la presencia física del trabajador, sin migración física. Hace tiempo ya que, en la literatura especializada se los llama “inmigrantes electrónicos”. Frente al *bodyshopping* que prevaleció con anterioridad: programadores

‘importados’ en condiciones de trabajo muy degradadas, a quienes no se les reconocían derechos laborales.

Para identificar el trabajo global o total que piensa, construye y desarrolla los *programas*, el proceso completo no sólo de trabajo, sino de producción, el trabajador colectivo, proponemos, además, o simultáneamente,

B) Identificar los procesos de producción, incluyendo, por supuesto el diseño, la preparación, el ensayo, la puesta a punto, la aplicación. Ya sea siguiendo algunos de los métodos que, tradicionalmente, se usan en el sector (Hellander, 2004), ya sea identificando desde la demanda de un cliente, y del producto hasta su elaboración y puesta a punto, y la atención de servicio subsiguiente. Una de las claves explicativas que hemos encontrado en muy distintos estudios de casos, como ya hemos puesto de manifiesto, ha sido el tratar de analizar: si es posible, hasta qué punto, para qué productos, y con qué herramientas informáticas de apoyo, el *rutinizar* procedimientos, fragmentar la producción, fragmentar el trabajo creativo que, al menos en el imaginario social existe sobre el trabajo informático, hasta poder construir *fábricas de software*.

Este es un argumento discutido y contradictorio, pero, obviamente, fundamental en el abordaje en términos de proceso de trabajo, *labour process*. Como ya recordamos en la introducción, desde hace treinta años es un argumento que figura en las agendas de investigación, y en las mejores interpretaciones sobre el fenómeno. Hoy en día disponemos de investigación abundante sobre el asunto, con perspectivas también históricas, que muestran la evolución y la maduración del ‘sector’ en distintos contextos nacionales y regionales, así como la transformación de las reglas del juego de la llamada “economía informacional” y su división del trabajo en la actualidad.

4.3. Primera ilustración de las ‘cadenas’ de software: trabajo y división del trabajo entre empresas en España

En el modelo de los grandes proyectos, por ejemplo de Telefónica, en el que pueden participar tres o cuatro grandes em-

presas, siempre hay algo equivalente a lo que existe en el sector de la construcción: esto es, una oficina técnica de coordinación, que “trabaja con las especificaciones, con el control sobre todo de las especificaciones y demás, y después realiza un diseño preliminar, donde sabe ya que es lo que le encarga a uno o a otro, como tiene que enganchar los fragmentos, como tienen que ser los mecanismos de interconexión, y lo que le tiene que exigir a cada uno de ellos, para la integración y demás”³⁷.

Por supuesto, las personas que forman este comité han de ser del máximo nivel de cualificación, y, además, tener mucha experiencia. Forman un equipo en el que predominan los *seniors*, “seniors con, además, con mucha experiencia, con experiencia, con mucha experiencia. Además, sobre todo, experiencia no solo en general, sino en lo que previsiblemente o preferentemente incluso concierne a ese tipo de proyectos, en ese tipo de entorno. Porque claro, hay veces, de unos proyectos a otros es muy difícil o muy complicado y son, tienen que ser, gente muy cualificada, ahí es donde el tema tiene que ser muy, muy importante.

A medida que después ya vas [descendiendo] de esto, sí es cierto que para uno de los partners también su cabecera, su liderazgo, tiene que ser cualificado. Pero después ahí hay una fuerza de trabajo mucho más amplia, que es el programador más de base. ¿Que qué tiene que tener?: Siempre tiene que tener cualificación lógicamente, y pasión y necesidad. Pero ya admite otro tipo de perfiles”.

Tomemos un ejemplo conocido: France Telecom. “France Telecom, tiene un núcleo de personas de plantilla, muy cualificadas, muy bien pagadas también porque tienen un nivel muy, muy amplio, y prácticamente aunque los veas allí *en sus instalaciones* [cursiva jjc], todo el resto de gente es subcontratación. Pero este núcleo es el que diseña, incluso toma las especificaciones. No se fía, no se fía. Por ejemplo, proyectos pues para el manejo de todo un tema crítico den-

37 Este modelo general de la división del trabajo entre empresas que preside, que enmarca, la forma en que se plantea la organización del trabajo del desarrollo de software, nos sirve aquí de ilustración primera y general. Los entrecorridos pertenecen a la entrevista de un representante de ATI, la Asociación de Técnicos de Informática. En este caso se han tomado informaciones de contexto de otras experiencias para que la utilidad de este esquema general sea más válido.

tro del marketing, todos los datos de facturación, interrelacionarlo para que vean si cuando hacen una campaña eso da resultado. Eso es clave para la empresa porque si no, si el dinero no va, el resto no va. La especificación la hace este núcleo de France Telecom que es de su plantilla y demás. Y después ya la parte de desarrollo más puro etc., la encargan a cualquiera de sus proveedores. Y este núcleo hace la supervisión de las entregas, los métodos de trabajo y demás de los proveedores; y además tiene que ser durísimo (...), tienen además que ser muy, muy duros con los proveedores. Cuando entregan les hacen una serie de auditorias, con una serie de reglas y demás, etc. Y este es el esquema. Porque, además, la mayoría de las empresas están considerando que el trabajo de fuerzas de trabajo muy amplias para el desarrollo de proyectos (claro que los proyectos siempre son temporales), no es el *core business*, como se dice ¿no?. Y, entonces eso, se tiende a subcontratar. Lo que se quedan son núcleos duros de profesionales muy cualificados de la casa que controlan todo el tema de proveedores, pero incluso controlan la fase inicial y final del proyecto, la de especificación y la de entregas y demás, controles finales y demás. Y eso es la tendencia pues prácticamente en casi todos, casi todas las empresas”.

Y, ¿cómo se llevan a cabo esos controles, esas especificaciones, esas exigencias?. La manera más habitual es el utilizar formatos concretos, métricas, “toda una batería de técnicas que se suelen agrupar, así, técnicamente, en lo que se llama aseguramiento de la calidad y demás. Y es una batería muy, muy amplia, de todos los ámbitos, y que incide tanto en esa parte de control y tal, como en elementos del propio proceso, y de cómo se tiene que generar el trabajo”. La división del trabajo se muestra así con toda nitidez, si tomamos como referencia el proceso completo que se inicia en la recogida de requerimientos y requisitos, en la demanda de la empresa, y termina en los programadores de las empresas subcontratadas ‘tirando código’. En este ejemplo de una gran empresa, la parte más importante, el núcleo del diseño y el armazón y espina dorsal del sistema queda dentro de la empresa, en manos de una parte del trabajador colectivo muy cualificada, bien pagada, con gran experiencia, fiel a la empresa e incluso adscrita a la alta dirección de la misma. Continúa con la fragmentación de los grandes

proyectos-procesos en procesos de menor alcance, pero que aún mantienen una alta complejidad, y que reproducen en los *prime contractors*, las grandes integradoras (los ejemplos más notables pueden ser Soluziona, Azertia, Capgemini, Atos-Origin, Coritel, etc.), otra división del trabajo que permite la desagregación de cada uno de los proyectos en proyectos más pequeños que pueden, a su vez, ser subcontratados, dentro del propio integrador, o a otra empresa distinta o lejana.

En cada escalón, quien capta estos trabajos suele ser un comercial (a veces asistido por un analista) que vuelve a recoger la demanda, los requisitos, que luego, un analista funcional ha de convertir en paquetes de trabajo, en unidades que puedan tener sentido por sí mismas, que se ordenen según una secuencia de ‘armado’ o construcción, que ha de pasar un conjunto de pruebas unitarias, etc., hasta su entrega como producto parcial integrado en este proyecto intermedio, que, a su vez, habrá de serlo en el proyecto global, controlado y supervisado desde el centro, desde la cabeza del proceso completo de producción.

En cada empresa, pues, y a distinto nivel, y con distinto sentido, tenemos una estructura que incluye: los jefes de proyecto y los analistas, “que en general suelen ser más cualificados”. El camino suele ser pasar de programador junior a programador senior y, con experiencia, con mayor capacidad, pasar a ser analista. Pero –basándose en la amplia experiencia personal propia, a la que se refiere nuestro entrevistado- “es cierto que [luego] hay muchos programadores. Porque esta es una actividad muy intensiva en mano de obra, porque los programadores tienen un límite razonable, por cuestión de calidad y productividad, de instrucciones y línea de código que pueden generar por día. Porque estamos hablando de estas grandes cantidades, de estas inmensidades de miles de líneas de código. Aunque algunas se reaprovechen de otros proyectos, siguen siendo una cantidad muy amplia. Y, claro, necesitas mucha gente. Simplemente. Que si no, no tienes suficiente gente como para escribir todo lo que necesitas...”

La imagen de la gran fábrica está presente en la propia definición de un trabajo tan necesitado de fuerza de trabajo directa. Y, el problema de la rutinización, de la estandarización, impuesta por la

propia forma de organizar el trabajo es una consecuencia obligada. A esos trabajadores hay que darles instrucciones lo más precisas posibles de lo que han de hacer: “en efecto, tienes que darles especificaciones muy claras. Porque ellos no pueden tomar decisiones, no porque no tengan capacidad, sino porque no deben tomar decisiones, porque si toman una decisión independiente, puede que lo que hayan hecho es que eso sea incompatible con el resto. Y hay que darles especificaciones y guías muy claras. Especificaciones, porque tienen que saber muy bien que es lo que tienen que hacer Y, lamentablemente, esto ha fallado muchísimo. Los analistas, por una razón o por otra, daban especificaciones muy poco elaboradas, detalladas. En algún caso, incluso, comunicación meramente oral, con lo cual se presta a muchos malentendidos, problemas y demás.

Y, además, tienen que dar otra serie de guías de cómo, además, quieren que se desarrolle. O sea, no solo qué, si no algunas ideas de cómo. Que eso puede estar con una normativa interna o con otros temas. Por ejemplo, hay toda una serie de herramientas que verifican si un programador por ejemplo, está cumpliendo una serie de reglas, incluso para su propio auto control. Hay muchas propuestas, por ejemplo, de reglas de estilo de programación; eso significa que el código, si uno lo intenta mirar así directamente, es crudísimo ies crudísimo!...

Entonces ¿que se suele hacer? Poner comentarios, documentación insertada como líneas, pues para que, bueno, si hay que retocar, si hay una rotación y un programador se te va, pues quien entra tiene que tener una guía porque si no la mayoría tiende a desechar ese código y volverlo a escribir otra vez, lo cual es antieconómico³⁸. O incluso determinadas formas de hacer una serie de operaciones, que según cómo las hagas puedes incurrir en cuestiones de riesgo. Es decir, determinadas instrucciones son *per se*, en cierto modo, poco recomendables, porque bajo ciertas condiciones pueden provocar un fallo; entonces lo que se suele hacer es decir: “por favor, estas no se utilicen nunca”.

38 La documentación permite, según nuestro entrevistado, la posibilidad de la rotación de puestos y la sustitución. Lo que es una norma habitual en el uso de los estándares, como el CMMI, como mostraremos en otros lugares de este informe.

Luego hay herramientas que utilizan los jefes de proyecto, lo analistas para decir: “bueno, tú me estás entregando esto, dices que has acabado, espérate, vamos a pasar los chequeos..., sí vale, por lo menos en cuanto a esto está bien, ahora vamos a pasar las pruebas, etc., etc.”. Y esa es la labor de recepción, clave, para que la labor de todas esas personas encaje y sea y tenga un mínimo de calidad”.

4.4. 'SAL': Un modelo de la secuencia de división y organización del trabajo en una empresa tipo en Castilla-León

Las empresas pequeñas de software, en España, que pueden tener entre 100 y 150 trabajadores son, obviamente, muy distintas entre sí. Ya sea por la especialización en una herramienta de programación; por su mayor habilidad para trabajar en determinados sectores (ya sean la sanidad, la administración pública, el sector bancario u otros); o por su vinculación con un “gran integrador”. Pero todas esas circunstancias, como hemos constatado en nuestra investigación, dejan marco suficiente para tratar de identificar una tipología de problemas enfrentados, que son comunes a la mayoría de las empresas. Siempre que se presenten, más tarde las matizaciones y precisiones convenientes.

Por ello nos ha parecido conveniente mostrar cómo se recibe, en una empresa de este tipo, un proyecto, una propuesta de trabajo. Cuáles son las características de la división del trabajo entre, generalmente, el ‘gran integrador’, quien ha contratado el proyecto completo, y esta fábrica de software, SAL, a quien se le encarga realizar una parte, más o menos importante, significativa, o cualificada, del proyecto global. Al hilo de la presentación, esquemática, basada en un caso real, se van identificando los problemas, las dependencias de la fábrica respecto del contratante. La imposición o negociación de los ritmos, el control de la calidad, las condiciones en que se ejercerán el trabajo y su organización³⁹.

39 Todo el texto siguiente se basa en las entrevistas realizadas en una empresa a la que llamaremos SAL, a distintos responsables, y llevada a cabo en octubre de 2006. Los textos entrecomillados corresponden a la transcripción textual

“Para nosotros un proyecto factoría es: uno que nos contrata un integrador, de un proyecto concreto y, sólo, nos contrata esa parte. Ya nos dan el análisis hecho, el análisis funcional, y nosotros hacemos el diseño técnico, la construcción y las pruebas. Luego se lo pasamos a ellos [al integrador], y ellos hacen la implantación al cliente. Y bueno, si hay algún error, se hace un mantenimiento correctivo hasta que el producto está [funcionando correctamente]. Y luego, ya, vendría un mantenimiento evolutivo. Esto es un proyecto factoría.

[Como veis son proyectos] que proceden de un cliente que es un gran integrador, no es el cliente final del desarrollo sino un integrador, como puede ser INDRA, AZERTIA, SOLUZIONA. Son proyectos, suelen ser, de millones de euros, proyectos de dos, tres años de desarrollo, implantación, y que luego tienen un coste de mantenimiento. Suelen ser equipos de hasta 20-25 FTEs, personas a tiempo completo, en un año. Y, esos proyectos, cubren todo el ciclo de vida [del producto], desde la consultoría en el cliente de lo que necesita, las especificaciones, el análisis, tal, hasta la implantación. Y nosotros, un proyecto factoría, cubrimos la etapa del diseño técnico hasta las pruebas.

Cuando uno de nuestros comerciales, tiene a un cliente, que en este caso son integradores, que necesita de nuestros servicios para cubrir un proyecto de factoría, lo primero que hacemos es: nos pasan el análisis funcional, con los casos de uso, todo definido. (...) Son documentos funcionales, pues, con los casos de uso que tienen las personas o, los procesos que interactúan y cómo se relacionan entre sí dentro de la aplicación que se vaya a desarrollar. Con esos casos de uso, el director de producción, determina que, bueno, normalmente en estos la tecnología viene dada, nosotros no tenemos que elegir la técnica, te dicen que es en Java, con esta arquitectura, con esta especialidad, con... o sea, que ya tienes la tecnología.

Y el director de producción lo que hace es, primero, identificar al equipo que va a estimar este trabajo a hacer. Porque a nosotros el integrador nos da eso y tenemos que devolverle un PES, que es un Plan de Especificación del Servicio que vamos a realizar. Es decir, una estimación en horas, en tiempo, una planificación

de las entrevistas. Especialmente, en este caso, corresponden a la Directora de Planificación y Control.

y una estimación económica. Y con los perfiles y los curricula, incluso de los perfiles que van a, que vamos a poner a trabajar en ese proyecto. Entonces, el director de producción con ese análisis funcional, determina un equipo, el equipo idóneo para establecer esa estimación, ese PES. Es un poco el *point*, la fecha, el proyecto, el trabajo que se va a realizar aquí, en SAL.

La planificación, las horas-hombre, el precio por perfil, los curricula, todas las particularidades que se vayan a cerrar. Es como el contrato previo que se cerraría con un cliente (llave en mano) pero, aquí, en este caso, para el proyecto en total, pues esta parte, esta parte que te va a llevar este tiempo y este coste. Y, los entregables, todo lo que vamos a entregar, en qué fecha... Una vez que tenemos ese PES, se lo mandamos al integrador.

En este caso como os voy a hablar del proyecto TRI, de X, que es una aplicación para tributos. Este proyecto todavía está vivo, todavía tenemos trabajo, y llevamos ya tres años en él. Si, es un proyecto grande y que está bastante estabilizado. Bueno, una vez que tenemos el PES se pasa a X y el jefe de proyecto, de todo el proyecto, puesto que nosotros sólo hacemos una parte, lo encaja en la planificación global del proyecto, ve si se va en costes, si se va en plazo, si encaja todo. Todos los parámetros que hemos tenido en cuenta ahí. Si hay que cambiar algo pues va y vuelve, vamos ajustando hasta que se, sale el definitivo. De ahí, eso es como si tuviéramos el contrato firmado de: <nosotros vamos a hacer este trabajo con estas condiciones>, y ahí, empieza ya, el proyecto pasa a la cadena de producción.

Normalmente, estos proyectos factoría, nosotros no hacemos la parte gráfica, la parte de interface, de diseñar las pantallas sino que ya vienen diseñadas del análisis y de X, y, nuestra parte es desarrollo, hacer el diseño técnico y desarrollar. Semanalmente o periódicamente, dependiendo de la criticidad del proyecto, de los riesgos, de algunos parámetros, de cómo se hace ese proyecto de X, porque, a veces, no se fían mucho en la distancia, hasta que no van cogiendo el ritmo, no están tranquilos de tener un proyecto que se está desarrollando, no ahí, al lado de tu mesa y que no puedes entrar en cualquier momento. Entonces, dependiendo de muchas variables, pues, se hacen una serie de seguimientos semanales, diarios, mensuales. Dependiendo un poco, como te digo, pues del

orden del proyecto, la criticidad; de la visibilidad dentro de la empresa, porque muchas veces cuando es un proyecto para directivos pues estás más pendiente de que todo vaya en regla. A lo mejor, casi, diariamente se hace un seguimiento.

[Bueno. Ya tenemos el proyecto. Vamos a ver ahora el] equipo que desarrolla aquí el proyecto. Después del contrato, se mete en la línea de producción y aquí se genera una pirámide que, aunque no es el proyecto completo, para ese proyecto, para ese trozo que vamos a acometer, hay una jerarquía, hay un coordinador, dentro de ese equipo, que es el que hace de interlocutor con el jefe de proyecto en X. Todas las comunicaciones se unifican en esa persona. En los seguimientos diarios es él el que da los informes, los reportes, si ve que hay alguna variación, porque, pues, no una vez que se cierra el PES todo es inamovible y tal, sino que normalmente te encuentras con algo que no estaba así previsto. Hay que replanificar, hay que reestimar. O nos piden acortar plazo porque ellos tenían que mover otra cosa. Se van haciendo ajustes sobre la marcha. Y ese papel lo hace el coordinador del equipo, que sería como el jefe de proyecto allí pero a más bajo nivel.

[-Para planificar la carga de trabajo del conjunto de la factoría, el acertar en el cálculo de los PES es fundamental, ¿no?]

- Para que mi planificación de toda la factoría, que son todos los proyectos, son todos, todos los que tenemos abiertos en la factoría; para que todo cuadre, pues tengo que fiarme de que esos coordinadores han afinado bastante en el tema. Como podrás comprender, no somos máquinas, no somos perfectos y esto, sólo es a base de prueba y error. Y ni siquiera así porque cada proyecto es un mundo. Y, alguien que tenga mucha experiencia, se supone que se va a equivocar menos pero, viene un proyecto que no tiene nada que ver y se puede ir en horas, en..., pero bueno, se trata de la pericia y de la experiencia.

De todas maneras, lo de las factorías de horas-hombre, uno estima y, pues, es una estimación sin todas las variables que hay que tener en cuenta y cuando se hace en el proyecto pues (...) surgen desvíos y tal que hay que ir corrigiendo sobre la marcha. Con el *handicap* de que cuando tu cierras un coste económico con el proyecto, con el integrador, eh, muy rara vez puedes volver a abrirlo

y decirle que ahora es el doble. (...) Cerrar un PES es, para el resto del proyecto, es muy, muy, muy importante.

[Y, además, está la presión de tiempo]. Normalmente, dependiendo del proyecto, pero, desde que te piden una estimación hasta que se lo mandas, no suelen pasar más de dos, tres días. Entonces, y cuando te lo piden, pues a lo mejor la persona ideal para estimarlo está en proyecto y con una sobrecarga de trabajo, una carga de trabajo que no lo puedes sacar. Y es muy difícil, la verdad es que se trata de tener una ocupación constante y tal pero, siempre hay picos, a lo mejor una semana necesitas que se hagan muchas horas extras y la semana siguiente tienes a la mitad de la factoría sin carga de trabajo. Entonces, es lo que hay que evitar. Se suele hacer I+D, ir preparando para un proyecto que sabes que va a entrar, pero vamos, es, es bastante difícil equilibrar la carga.

[Volviendo al caso concreto que nos sirve de ejemplo preciso, TRI, ese es un tipo de proyectos que] nos gustan, porque cuando duran tanto en el tiempo se estabiliza mucho. Se hace más monótono y es como, este proyecto en particular está tan rodado, que se hacen PES, de un día para otro se hacen PES. Claro. [Porque] la parte de trabajo que a nosotros nos mandan, no se mete sólo en un PES, se divide en UTs, Unidades de Trabajo, en módulos. Nosotros, a lo mejor tenemos que hacer, muchas aplicaciones, todo el desarrollo de esa aplicación. Pero tiene varios módulos. Por ejemplo, si tuviera más módulos de pagos, de facturación, de recursos humanos, de tal, pues se van partiendo en UTs y cada UT es un PES, se suele acortar.

[Cada uno de estos módulos podrían ser las partes que es necesario secuenciar, del mismo modo que se secuencian en la organización tradicional de una cadena de montaje]: las partes que estén relacionadas pues tienes que hacerlas a la vez para poder probar. Entonces, en cuanto ves el proyecto, y ves las relaciones, esto viene en el análisis funcional. [El técnico] que está estimando, sabe que esta parte tiene que ir con esta y que no la puede dejar para dentro de seis meses. Porque si no la otra parte no la puede probar. Entonces, no lo puedes saber hasta que no ha empezado el siguiente, entonces, es un poco de lógica. De todas maneras, no solemos hacerlo nosotros, casi vienen hecho de X, porque ya el análisis funcional está dividido en, en, como en paquetes, en módulos.

[Para constituir los equipos de trabajo para cada módulo], normalmente, nosotros lo que hacemos es agrupar a la gente, en la sala por proyectos. Inicialmente no lo hacíamos así pero al empezar el proyecto factoría de larga duración, vimos que era mucho mejor tener a gente en la sala, del mismo proyecto.

Antes los teníamos a todos agrupados por tecnologías. Estaban diseñadores gráficos, los de interface de maqueta, animación, luego la parte de desarrollo, luego de base datos, luego sistemas... Como por bloques, con los mismos perfiles. Pero en un proyecto, un proyecto se compone de varios perfiles. Un equipo de personas pero de distinto perfil. Entonces, tenías a gente trabajando en toda la sala de producción, en el mismo proyecto, por todas partes. Entonces, todo el rato levantándose, al teléfono, más ruido, menos...

Porque el proyecto, depende mucho, también, de lo implicados, lo relacionados que esté la gente. Porque una misma (Unidad de Trabajo) la pueden estar desarrollando cuatro programadores. Una UT puede tener, 20 pantallas, que de una se pasa a otra. Que de lo que yo estoy haciendo, paso a la tuya. Entonces, si están los cuatro en la misma mesa y, se puede hablar, es mucho menos ruido, o sea, es todo mucho más fácil. Entonces, nosotros optamos, con estos proyectos largos, por agrupar a la gente por proyecto. Entonces tenemos, en dos mesas, a lo mejor, está toda la gente de ese proyecto y así hablan lo que necesiten. Y eso es facilitar el trabajo y la coordinación porque el coordinador necesita agruparlos en un momento dado van a, a la gota [un espacio redondo en extremo de cada mesa] que tenemos y no hace falta que esté todo el mundo por la sala, en la sala de reuniones.

Es, un poco, optimizar todo porque, al fin y al cabo, son horas-hombre, aunque no queramos reconocerlo y, en la medida en que optimicemos todos esos tiempos y que estén más implicados, o sea que si uno tiene una duda se pueda levantar y decir <oye, esta parte la hiciste así, ¿cómo...?>. Entonces eso, son pequeñas cositas que ayudan. Sí, porque vas viendo que eso va a facilitar las cosas⁴⁰.

40 El edificio de esta fábrica, situado en un polígono industrial, forma parte del diseño mismo de la organización del trabajo. Tal y como se expresa el director de la empresa: "diseñamos este edificio; este edificio que es nuestro, lo teníamos pensado para evitar los problemas que veníamos sufriendo [en el centro

Dentro de un equipo de proyecto hay un coordinador, hay, diseñadores técnicos, hay programadores senior, hay programadores junior, auxiliar programador, y luego, finalmente, pueden entrar perfiles creativos como diseñador gráfico o maqueta, cuando los proyectos requieren que hagamos la parte visual también. Lo que es el estilo, la parte gráfica del proyecto. Pero cada uno tiene su perfil y, sí que uno aprende de otro, por supuesto. Cuando estás trabajando en un proyecto, cuando ya llevas tiempo programando y ya, piensa que un diseñador técnico, el siguiente paso de tu propia carrera lo tiene al lado.

Aunque, en ocasiones, hay gente que se quiere quedar programando. No les gusta o, pues eso, meterse a hacer PES, porque le pesa más la responsabilidad. Claro, si tú haces mal un PES y te has equivocado en dimensionar un trabajo, pues puede que eso repercute en que el equipo tenga que hacer horas extra, cosas que... Son responsabilidades que hay gente que no quiere tomar, él quiere que le des trabajo y desarrollar; y hasta ahí”.

5. ‘ALAR’: un centro mixto de desarrollo, Universidad-Empresa en Castilla-La Mancha⁴¹

5.1.1. Origen de la empresa, constitución y localización.

Las primeras noticias de esta empresa, localizada en una ciudad intermedia de Castilla-La Mancha, y que llamaremos ‘Alar’, las encontramos en 1998, cuando se constituye, “Y”, como una empresa independiente, especializada en desarrollo de software,

de la ciudad. Para facilitar la comunicación, para la integración de los equipos, para la flexibilidad de incorporación de personas de los clientes, para todo lo que necesita tener una factoría. Entonces, luego, cuando lo veais, vais a ver que sólo está pensado para el desarrollo de software en el modelo factoría”. Entrevista realizada el 17 de octubre de 2006. Director SAL.

41 Presentamos aquí, en detalle, por razones de espacio en esta comunicación, tan sólo uno de los tres casos analizados en profundidad. Un estudio de caso, especialmente exitoso.

a partir de una gran empresa del sector eléctrico⁴². Sus orígenes se remiten su instalación en esta ciudad castellano-manchega en 1999, aprovechando las oportunidades ofrecidas para su ubicación por los Centros Europeos de Empresas e Innovación, que habían surgido en 1984, como una iniciativa de política de desarrollo regional propiciada desde las Comunidades Europeas, y apoyada por un conjunto de instituciones, entre ellas las Cámaras de Comercio. En 1999, con sólo 14 personas “Alar” se instala en el parque empresarial de la ciudad. Así lo narra el Director actual de la empresa: “nos encontramos con un lugar, con todo, con la electrónica. Para llegar, enchufar los equipos, las cabezas, y a trabajar. Y así empezamos, en octubre de 1999, con 14 personas. Y lo primero que hacemos es establecer contacto con la Universidad, en concreto con la Escuela de Informática [de reciente creación, como veremos, jjc]. Y rápidamente establecemos un nivel de relación: “oye, que vamos a necesitar informáticos”⁴³. Y este rasgo es el que más va a caracterizar la situación y las perspectivas de esta empresa, como más tarde veremos en detalle, y que acaba constituyéndose en un raro ejemplar de colaboración entre la Universidad Pública y una empresa concreta con consecuencias muy importantes para ambas instituciones. Y, desde luego, para la explicación del funcionamiento de la misma. Pero, sobre esto volveremos en detalle más abajo.

Decíamos que la empresa se instala a finales de 1999, y todo parece indicar que hay una estrategia muy clara y definida para su evolución futura. La prensa económica destacará que “esta iniciativa se enmarca en el proyecto de deslocalización o desarrollo de software de la firma de servicios profesionales, [Y], fuera de los centros

42 ‘Alar’, que es el nombre supuesto que damos a esta empresa en la localización que estudiamos, forma parte de esa gran empresa de desarrollo, a la que en esta investigación llamaremos “Y”. Al igual que en los otros casos estudiados, hemos manejado una documentación de primera mano sobre ambas, Alar e Y, incluida su situación actual, sometida a grandes transformaciones y adquisiciones. La mayor parte de esa documentación es interna a la empresa, facilitada públicamente, en su página web, en la prensa especializada, o en sus propias publicaciones. Utilizamos aquí tan sólo los datos imprescindibles de situación que nos permiten una comprensión de los problemas que abordamos, manteniendo la necesaria discreción sobre la empresa concreta.

43 Entrevista a DIR-ALAR, 24 de febrero de 2006.

tradicionales, ubicados en su mayoría en Madrid o Barcelona, con el objetivo de obtener ventajas, no solamente en costes, sino también en calidad, productividad, eficacia y formación de profesionales, con vistas a competir con los líderes internacionales del sector”⁴⁴.

Pronto llegarán a un acuerdo con la Universidad, por el que promueven un Centro de Desarrollo Mixto, en terrenos de la Universidad, cuya dirección general la ostenta un Catedrático de la misma, y, dentro del cual está la empresa. Los acuerdos, firmados en el año 2001, prevén, mediante un convenio específico firmado por el Rector de la Universidad y la alta dirección de Y, que se implante lo que, en la prensa local y en los boletines de la empresa, se considera “la mayor iniciativa de impulso tecnológico para la región, en el ámbito del desarrollo de software y de las nuevas tecnologías informáticas”. En un plazo de cuarenta años los terrenos, de la propia Universidad, revertan a la misma. La Empresa, por su parte, invierte, según los datos aireados por la prensa, ocho millones de euros en unas instalaciones modélicas que podrán llegar a albergar hasta 500 desarrolladores.

Cuando se inaugure el CMID, Centro Mixto de Investigación y Desarrollo de Software, en abril de 2003, son ya 80 los empleados por Alar. Y, en el momento de nuestro trabajo de campo, visitas a la empresa, y entrevistas, el total de ocupados se acerca ya a las 250 personas, que trabajan con una organización estandarizada, el nivel 3 de CMMI.

El Director de Alar cree que “realmente, cuando podemos decir que somos una fábrica de software, con todo lo que eso implica, no hace más de tres años. El tiempo que llevamos aquí, 3 ó 4 años. Porque tienes que tener muy consolidadas la metodologías y todo el entorno productivo, la reutilización... Todo eso que hablamos de Fábrica de Software, pues no es fácil llegar allí. Porque como todo el mundo, nos hemos llegado a denominar fábrica de software aunque seamos un taller artesanal. Entonces claro, llamarte fábrica de software es muy fácil porque tú entras en Internet y cualquiera se llama fábrica de software. Pero de un taller artesanal a una fábrica de software hay mucho recorrido. Tienes que tener unas capacidades, unos procesos, unas métricas...

44 *Finanzas.com*, 29 de abril de 2005.

5.1.2. Universidad, empresa y mercado de trabajo

El 83 por ciento de los desarrolladores ocupados en Alar, en 2006, según los datos proporcionados por la empresa, proviene de la propia Universidad local. Este dato llamativo es un buen indicador de las especiales características de la empresa que nos ocupa. Para el Director de la Escuela de Informática, que nos recuerda que la Escuela había comenzado su andadura, precisamente, en octubre de 1998: “Entonces, era una época en la que era una vorágine, era, era tremendo. De hecho, venían las empresas, porque en aquella época venían muchas a hacer presentaciones y tal, y nos decían: ‘nos da igual superiores que técnicos’.

Entonces, claro hablaban con ellos y yo decía: ‘Hombre, no me digáis eso porque claro me desmotiváis [a la gente]’. Porque nosotros si tenemos claro lo que es un técnico y un superior. Otra cosa que las empresas estéis buscando gente como locas y os dé igual.

Pero no digáis que da igual un perfil profesional de un técnico o un superior porque no es cierto. O si no, algo estamos haciendo mal, no. Entonces, pues esa percepción se transmitía a los alumnos. Se matriculaban en las técnicas y decían: ‘luego si quiero me matriculo en la superior’. Pero cuando acababan la técnica o sin acabar la técnica empezaban a trabajar”⁴⁵.

La observación que conviene hacer, es constatar el hecho de que se está llamando la atención hacia el tipo de trabajos ofrecidos a estos jóvenes, por cierto, en el 98 por ciento de los casos, chicos, con una llamativa ausencia de mujeres. “El problema –continúa el Director de la Escuela- es que tampoco eran trabajos muchas veces muy cualificados. Eran trabajos de informáticos, pero eran fundamentalmente programadores. Entonces, evidentemente esa gente no era la que gestionaba los proyectos, diseñaba los proyec-

45 Entrevista al Director de la Escuela de Informática, 24 de febrero de 2006. Esta tendencia hacia el recurso a la menor de las cualificaciones que ofrece la Universidad es un indicador del tipo de profesionales demandados, al menos en las escalas más bajas del trabajo en Alar, como becarios. Que luego se integrarán como programadores junior. En todo caso, la baja edad media de los programadores ocupados en 2006, 27 años, es muy llamativa, y requiere de un estudio detallado de la rotación y abandono del trabajo.

tos. Era la que hacía temas de mantenimiento, programación... [Porque] llega un momento que las empresas tienen metodologías muy automatizadas para hacer las cosas. Entonces, lo que necesitan son gente que machaque, que piquen código, que se dice normalmente, no. Pues bueno, había gente que nos decía: 'bueno, nos habéis dicho que el ingeniero superior hacía unas cosas y luego las empresas lo que quieren es esto' Era un momento, en que hacía falta, en que había mucha demanda de personas pero fundamentalmente para ese tipo de cosas”.

[A la empresa] digamos que le ha gustado la gente que salía de aquí, el perfil profesional que tenía. Y nos ha ido demandando; y en las peores épocas a lo mejor demandaba 10 ó 15. Y, en las mejores, ahora, está demandando del orden de 30 ó 35 al año. En prácticas en empresa. Pero bueno, nuestra función de sacar personas al mercado laboral, yo creo que nadie se puede quejar [...]. Yo creo que ahora producimos una cantidad importante y más o menos lo absorbe la empresa local, hay gente que se lanza a Madrid. En fin, hay unas encuestas que hicimos hace poco del tiempo que tardaba una persona en salir de aquí y encontrar trabajo y estaba en el mes y medio, mes, mes y medio. No nos podemos quejar”⁴⁶.

Otro catedrático de la Escuela de Informática, que ocupa el cargo de Director del CMID, nos precisa en este sentido: “En la escuela tenemos tres perfiles. Uno orientado a la multimedia, otro orientado más al hardware, redes, que es la parte que lleva el Director, y otro a sistemas de información, que es la que llevo yo. Ya se hizo la escuela con un perfil, orientado un poco al desarrollo del software. Con lo cual, salen muy bien preparados. Pero vamos, no porque lo digamos nosotros, que es que esta orientada a eso. Si me dices en otra cosa, pues te digo, en otros campos a lo mejor no salen tan bien preparados. Pero, en este campo, sí salen muy bien preparados. Y entonces, pues ya los tienen... Los cogen de becarios, tienen un programa de becarios con la escuela, y luego se incorporan”⁴⁷.

46 Entrevista citada al Director de la Escuela de Informática, 24 de febrero de 2006.

47 Entrevista al Director del CMID, 5 de julio de 2006.

5.1.3. Sinergias y refuerzos: la investigación universitaria y Alar

La Universidad, con la llegada de Alar, ha podido atraer a profesores e investigadores de gran valía, hasta el punto de haberse constituido un grupo de investigación con presencia internacional de primer plano, y que participa en numerosos proyectos de investigación. El Director de este grupo, Catedrático de Universidad, es, también, el Director del CMID, y tiene en su haber una larga experiencia profesional, así como de publicaciones y proyectos en curso. Para él, Alar no es una empresa “representativa de lo normal” en España. Es una excepción notable, “es de lo más avanzado que hay”, con CMMI nivel 3 implantado, y un edificio modélico, como decíamos en pleno centro de la Universidad, junto a otros centros de investigación de primera línea. Su grupo de investigación trabaja “en un solo tema realmente. Aunque nosotros somos catorce doctores y unos cincuenta doctorandos en total. Trabajamos en calidad de sistemas de información”. “Dentro de calidad de sistemas de información, trabajamos en la parte de software, y ahí, lo que hacemos es abordar distintos aspectos. Por ejemplo, por un lado el aspecto metodológico, desarrollamos metodologías para el desarrollo de software. El aspecto metrológico: nosotros estamos muy metidos en la parte de métricas, para la calidad y la mejora de la calidad. Pues ya sabéis que CMMI y todo eso son procesos para mejorar la calidad del desarrollo de software. También la gestión del conocimiento: porque también es una parte importante de la gestión de la calidad y de la mejora de la calidad. Y luego, tratamos diferentes aspectos de procesos, etc.[y tenemos en marcha varios proyectos de investigación en estos campos]”⁴⁸.

Entre esos proyectos internacionales se incluye “una especie de CMMI, pero iberoamericano. Porque CMMI al final es estado-unidense, y hay un problema de cultura. Tú no puedes trasladar la cultura de Estados Unidos a una cultura latina-española, no funciona. Aparte que, aunque lo traslades, es carísimo. Porque tú tienes que pagar unos royalties al Software Engineering Institute, y entonces

⁴⁸ Entrevista al Director del CMID, 5 de julio de 2006.

no tiene sentido que éstas empresas que son pequeñitas y en éstos países, paguen esos royalties”. [Porque] una cosa es lo que nosotros hacemos para la gente de Iberoamérica, para las PYMES, o PYMES españolas; y otra cosa es, el tema de aquí de Alar, que es un nivel de lujo, claro, porque es la fábrica de software. Es una de las pocas que está certificada a nivel tres en España (se calcula que habrá unas 10 ó 14 como mucho entre más de tres, de tres a cinco), y va a llegar a nivel cuatro en el 2007; y claro tienen personas de calidad, personas responsables de procesos, tienen infraestructuras, tienen todo un proceso bien definido, tienen... otras cuestiones”.

Igualmente, la Universidad trabaja en la elaboración de nuevas metodologías, que vayan por delante de la práctica actual, que puedan adelantarse a los cambios radicales que se presentan o van a presentarse. Un ejemplo de ello es “el Model Driven Architecture Engineering”, es decir generar códigos a partir de modelos automáticamente. Y eso va a cambiar toda la forma de trabajar, y ya está cambiando. Lo que pasa que las empresas no se lo creen todavía, pero ya existe. En informática donde se pone mayor énfasis, desgraciadamente, al final es, en codificar y programar. Y se olvida, o se le dedica poco tiempo, a las fases de análisis o diseño. En algunos sitios como más avanzados, pues se dedica más tiempo a análisis y diseño. Pero es que además con estas herramientas y esta tecnología, una vez que te he hecho el modelo, le das al botón y se genera automáticamente la aplicación. Es la informatización de la informática. Pero, esta situación –argumenta a continuación el Director del CMID- “a nivel de empresas, todavía no está muy implantado”. “O sea, la realidad todavía, es que hay mucho de codificación, que hay que codificar mucho y que bueno, que hay que pasarle especificaciones a la gente para que codifique. Y de las [empresas] que yo conozco, tanto en Iberoamérica sobre todo, como aquí en España, pues al final se codifica mucho. Y hay mucho énfasis en las pruebas, en la verificación de que el código esté bien y tal; pero todavía sigue siendo mucho la codificación. Es la realidad. Porque en informática hay un problema: la diferencia entre el estado del arte y el estado de la práctica. Y hay un desfase muy grande [...]”⁴⁹.

49 Entrevista al Director del CMID, 5 de julio de 2006.

5.1.4. Alar en acción: muchos y diversos proyectos, una sola metodología

En Alar se estaban llevando a cabo muy distintos proyectos cuando realizamos una visita detallada y guiada por su Director, deteniéndonos en los detalles de cada uno de ellos (15 a 20), para conocer la diversidad de los problemas planteados, según sea un proyecto a tres años, a dos años, a uno, de varios meses.... Según las arquitecturas utilizadas y los equipos y personal comprometido en el mismo. Ahora bien, como insistirá nuestro guía e interlocutor, “el entorno metodológico es el mismo; la metodología es una cosa, y luego la técnica, la herramienta que utilices es la que quieras”⁵⁰.

La distribución del espacio, “en pradera”, agrupa a distintos equipos, según el volumen de trabajo, pero todos próximos, y, a la vez, perfectamente delimitados. El diseño del espacio de trabajo ha sido muy cuidado y, se nos hace notar, es el mismo en los distintos centros de desarrollo que la empresa tiene, lo mismo en Madrid que en Bratislava o Panamá.

Indicaremos ahora, ejemplarmente, algunos de esos múltiples proyectos, con las limitaciones que impone la confidencialidad y la deontología profesional, pero que transmitan al lector una imagen mínima del entorno en el que, como trataremos en detalle en el apartado 5, se ubica el proyecto de Sanidad que hemos tomado como centro de nuestra atención.

Un primer equipo trabaja para el Servicio Regional de Empleo en un proyecto de desarrollo incremental para la gestión de la formación continua. Es un grupo pequeño de cinco personas. En otro lugar se trabaja en la construcción de un módulo aduanero, de comercio, para un gobierno de América Latina. En otro se desarrolla un programa para un canal de televisión. Más adelante se está llevando a cabo un proyecto corto para el portal de consumidores de una Comunidad Autónoma cercana. En este caso, el módulo sirve también como formación para un becario recién incorporado. Más adelante se termina un complejo sistema de gestión de armas y explosivos, que ha durado un año y medio, y que va a tener su con-

50 Entrevista al Director de Alar, 24 de febrero de 2006.

tinuación en un proyecto paneuropeo. La Sanidad, “el área de más futuro”, ocupa un lugar privilegiado, y se desarrollan proyectos para distintas Comunidades Autónomas, y distintos proyectos o módulos (ver el apartado siguiente). Sesenta personas se dedican en un ala del edificio a hacer software bancario para ISBAN. “Nos pasan grandes lotes de desarrollo. Aquí la única diferencia es que trabajamos con la metodología de ellos. Ellos aportan su metodología, como es lógico, y nosotros componemos piezas que luego ellos encajan en *puzzle*. Estas piezas quizás por sí solas no funcionan. Ellos tienen el mosaico y nosotros diseñamos piezas para ellos”. Un gran grupo ocupa una superficie notablemente superior diseñando el sistema fiscal de Palestina, trabajando en consorcio con la Agencia Tributaria, y financiado por la Unión Europea.

Muchos otros proyectos se llevan a cabo en el centro, pero, en todos ellos, la secuencia de pasos a desplegar la marca la metodología de Alar. Claro está, como resulta obvio para proyectos de complejidad y tramas tan distintas, en ciertos proyectos no se precisa ser tan exhaustivos como en otros: “La metodología tiene que ser un aspecto práctico, y que de valor al trabajo, y que sea útil. Una metodología, la nuestra, con trazos muy vivos”, particularizada y propia de la empresa, que se basa en CMMI nivel 3.

*5.1.5. Una fabricación particular o la recuperación de la división del trabajo. Concepción más ejecución en un programa complejo*⁵¹

“Antes no era así [la riqueza y diversidad de la programación]. Los programadores antes, me refiero a los tiempos en que solamente las grandes corporaciones tenían ordenadores, grandes máquinas, en esa época, un programador era un mero traductor de lo que se llamaba cuadernos de carga, que emitían los analistas. Los analistas

51 En este apartado utilizamos las informaciones recogidas sobre el terreno, las notas de campo, y de las entrevistas realizadas en “Alar”, con ocasión de dos visitas de trabajo, los días 24 de febrero y 5 de julio de 2007, para las consideraciones más generales. Las citas entrecomilladas, mientras no se indique lo contrario, corresponden a la segunda entrevista al Gerente del Área de Sanidad, 5 de julio de 2007.

hacían todo el trabajo de cabeza, y el programador se limitaba a, traducir lo que venía perfectamente especificado en el cuaderno de carga. Eso ha ido cambiando muchísimo a lo largo del tiempo, y desde luego la percepción que pueda tener un programador en la actualidad de su puesto de trabajo, o del trabajo que realiza, es muy distinta y mucho más motivante. Porque al final la motivación que pueda tener una persona que se dedica al desarrollo del software, pues está, como la de cualquier otra persona en función de, de tres grados de libertad: “qué, cuándo y el cómo”.

Si a una persona le dices, “qué es lo que tiene que hacer, cómo lo tiene que hacer y cuándo tiene que estar hecho”, lo matas. No, no, le impides que sea creativo y lo conviertes en una especie de autómatas o en un robot, que, que es muchas veces lo que se pretende, ¿no?, con este tipo de actuación. [...]. Pero lo que quiero decir, es que la gente, en mi opinión, la gente que se dedica a programación, tiene algo de creativos, es decir son, son artistas, de alguna forma, porque crean algo. Yo no creo que transformen, yo creo que tienen la suficiente capacidad como para crear. En el caso de las fábricas de software se centran evidentemente mucho más en la tarea de producción que en la tarea completa. Que la tarea completa es la de creación ¿no?: la divides en partes, la parte final, una de las partes casi finales, es la de producción.

Pero en las fábricas de software lo que también se valora mucho, es poder atacar desde el principio el proceso, y eso es lo que hacemos en software. Como los requerimientos, al final, que los médicos van dictando, es decir lo que ellos quieren hacer, es bastante complejo de explicar o de formalizar en un documento, nosotros lo que empleamos es una técnica que se llama ‘Incremental con Maqueta’. Significa que vamos cerrando requerimientos muy funcionales, muy acotados, lo maquetamos en muy poquito tiempo, para que lo que nos cuentan los médicos, lo vean inmediatamente reflejado en una maqueta de cartón piedra en el ordenador, y que vayan viendo ellos qué efectos, tienen las conversaciones que vamos teniendo.

En el momento en que se cierra, es un paquete de funcional, que tenga sentido, ya tenemos el análisis ahí, y eso ya pasa a fábrica. Entonces esa técnica se emplea cuando en realidad los clientes, que son los que te van a dictar qué es lo que quieren, no tienen muy

claro qué es lo que quieren. Entonces esto requiere de una labor pequeñita de consultoría inicial, en la cual tú vas acotando con ellos exactamente cuáles son los requerimientos. Es muy distinto a trabajar en un banco. Un banco tiene perfectamente catalogados todos sus procesos, al milímetro. Es un trabajo, evidentemente, muy distinto, mucho más volumen. Multitud de procedimientos, pero de creatividad, poca. O sea hay poco que aportar. Todo está perfectamente entendido. A grandes rasgos es lo que hacemos. Luego ya en la fase de producción, es donde digo que, a mi entender, no hay una única forma de hacer las cosas, no existe [...]. Ahora mismo estamos haciendo un proceso de ese estilo.

5.2. Estamos diseñando una aplicación de atención primaria, y lo estamos haciendo así

Hay una comisión de médicos. O sea esto se hizo de la siguiente forma. [Lo vemos] desde el principio. El ‘Servicio de Salud’⁵² decide que quiere hacer una aplicación de atención primaria, y lo primero que nos propone es que, nosotros propongamos una aplicación, o sea que nosotros hagamos la lista de requerimientos; o sea, tú fíjate que en teoría [deberían ser ellos].

‘Proponnos un análisis, una toma de requerimientos de la aplicación de atención primaria. Es decir, descríbeme en un documento y en una maqueta, que es lo que debería ser a tu entender, una aplicación de atención primera’. Contratamos a un médico, estuvimos hablando con él, estuvimos trabajando con uno que tiene experiencia ya en utilizar otros productos de atención primaria. Vamos recopilando experiencia. Incluido el software libre. O sea, que ya no hay ningún tipo de..., o sea, no hay una distinción entre la procedencia del software, no, no.

Voy componiendo ese tipo de documento. Entonces, una vez que se tiene eso, lo que yo pido entonces al ‘Servicio de Salud’, es que monte una comisión médica. Es decir, una reunión de cuatro, cinco, diez médicos de atención primaria, y de distinto perfil, pues

52 Se trata del Servicio Público de Salud de una Comunidad Autónoma.

pediatras, enfermeros, médicos generales, etc. Se les distribuye la maqueta y el documento, y se va dirigiendo la reunión. [...] Y se ponen, los deberes, entre comillas, de que estudien ese documento, vean si lo entienden, si sí o si no, y qué problemas le pueden poner. Entonces a partir de ahí se inicia el trabajo con la comisión médica. Cada reunión, se tienen reuniones semanales, los comentarios que ellos hacen o deshacen de la maqueta, se ven en la siguiente semana [...]. Entonces, en el momento en que ellos están conformes, con esa maqueta, eh, digamos, ahí paras para tomar más requerimientos. Ellos aceptan que eso es lo que quieren. Ahí paramos.

Y eso es la documentación que entra en el circuito de producción, de fábrica. Son como los planos, es como una fábrica de coches, en el momento que tienes los planos, la fábrica hace perfectamente el coche, si el plano está correcto. O sea que aquí, lo que hemos hecho es dividir, en dos partes, uno es el trabajo de confeccionar los planos, y el otro es ejecución del sistema.

Hay pruebas, hay puesta en producción, hay formación, hay más etapas en el proyecto; pero lo que compete a fábrica de software es: por supuesto la producción, y *en algunos casos como éste*, la confección de los propios planos, de los propios requerimientos. Esta es más o menos, esta es una técnica. Esta se llama, 'Incremental con Maqueta'. La hay en cascada, con o sin maqueta, la hay... Esta es un poquito más cara, resulta un poquito más cara, pero tiene la ventaja de que vas dando pasos seguros. Es decir, los usuarios van viendo el trabajo, poco a poco, y no hay ninguna sorpresa.

Hay otras metodologías que lo que dicen es: "Bueno, recoge todo, todos los requerimientos y hazlo todo a la vez". Con lo cual, cuando vas a ponerlo en producción y se los enseñas a los usuarios, pueden surgir cincuenta mil problemas. Entonces cuando no están muy claros, los requerimientos iniciales de usuarios, la técnica a emplear es esta; y si esto sale un poquito más caro, tampoco mucho más caro, desde luego, es mucho más seguro.[...]

5.3. "Y esto entra en producción": la concepción

[¿Cómo es la organización de la producción?]

-Eso es otro tema. Eso también aquí, también cada maestrillo tiene su librillo.[...] En cuanto a la organización de la producción en sí, vamos a ver, aquí hay distintas fases [en la ‘concepción’], distintos perfiles. Hay un analista de nivel alto, que es el que está tomando requerimientos, con el cliente. A estas reuniones también suelo llevar a personal técnico, pero no porque aporte nada funcional, si no porque muchas veces cuando yo estoy enseñando la maqueta y a alguien se le ocurre: -Oye, ¿y si hago esto, de esta forma y tal?--; a lo mejor el técnico, si levanta la mano: ¡problemas!. [Porque no es tan fácil].

O sea que hay perfiles: analista, y un técnico de apoyo. Simplemente para que vaya validando que lo que están pidiendo se puede hacer razonablemente.

Una vez que se toman los requerimientos, pasa a lo que yo llamo: diseño. Es decir, aquí es donde ya se trocean, aquí ya entra la metodología, digamos interna, es el primer paso que se da dentro de fábrica. Hay un diseñador y un arquitecto. Normalmente el arquitecto es una persona de perfil muy alto, que trabaja una vez, montando la arquitectura, técnica, del producto. O sea, qué servidores va a haber, qué bases de datos, cómo están montados, o sea toda la parafernalia ‘arquitectura’.

Y el diseñador que es el que al final trocea la aplicación, el caso de uso, va generando la documentación, y los esqueletos, para que ya, cada uno de los equipos que se van a hacer cargo de cada parte de la funcional, se haga cargo de ello. Entonces estos dos perfiles son muy importantes, el primero es el que, una persona que va dictando cómo se va a comportar físicamente, técnicamente el producto; el otro es el que dice cómo se va a comportar funcionalmente. Y lo que hacen es distribuir trabajo. Pueden ser los dos la misma persona, incluso. Si tiene capacidad funcional el arquitecto.

A mí me gusta mucho que los perfiles sean mixtos, en ese sentido, o sea, no, no me gusta, aunque entiendo que es necesario, gentes absolutamente técnicas, pero sin ningún barniz funcional. Porque es muy difícil hablar con ellos, no puedes. Y lo contrario tampoco, o sea un perfil absolutamente funcional y que no tenga ni idea de la tecnología, porque suelen ser kamikazes. Creen que

se puede hacer todo... Hay un, un equilibrio que se puede, y que hay que tener entre el aspecto funcional y el aspecto técnico, que desde luego para mí es ideal. O sea una persona que sabe lo que se puede o no se puede hacer, y que cuando está hablando con un médico, pues puede seguir su forma de hablar, en fin, hablar su mismo [lenguaje]. Entonces éstos, esta persona, una o dos, es la que se encarga a continuación de dar trabajo, ya dependiendo del tamaño del proyecto, a los equipos.

5.4. Organización del trabajo, la ejecución: los equipos y su composición, metodología, estándares

Estos equipos están formados por, normalmente, cuatro programadores y un jefe de equipo. En el caso del ‘Servicio de Salud’, el de atención primaria, pues ahí hay tres equipos de cuatro personas, cada uno, cada uno con su jefe de equipo. Y la organización interna del equipo y la responsabilidad de cada equipo, pues entronca ya con la metodología de Alar [...]. Aquí cada equipo, la responsabilidad que tiene cada equipo, es de, realizar el trabajo que le manda el diseñador, dentro del entorno, casos de uso, y se hace responsable de su rama de desarrollo en el control de versiones.

Luego hay un jefe de proyecto, también. Es una persona importante también aquí, que se encarga de hacer, el “merger” o la unión de todas las ramas de desarrollo de cada uno de los equipos. Nosotros lo que hacemos, cuando damos de alta un proyecto es, de acuerdo con el cliente, atacamos las fases de la metodología que tengan sentido para ese proyecto. Es decir, para los proyectos de ciclo corto, o sea que son, tienen un tamaño de menos de doce meses hombre, más o menos, lo que hacemos es, utilizamos una metodología simplificada.

Y luego para proyectos superiores, lo que hacemos es, de la metodología general de Alar, vamos aplicando exclusivamente aquello que tenga sentido para ese proyecto concreto. Porque evidentemente no toda la metodología tiene sentido, eso no hay, yo no he visto ningún proyecto en que toda la metodología tenga,

sentido completamente⁵³. Normalmente también lo que ocurre, es que la metodología es algo que encarece el desarrollo. Es un índice de calidad muy bueno, porque te da una seguridad de que todo va a ir bien, pero es que lamentablemente, hay veces que no te pagan lo que cuesta hacerlo así. Y en ocasiones, pues bueno, tenemos ciertos problemas, porque claro, nosotros tenemos que tener, o mantener un equilibrio entre hacerlo o querer hacer el proyecto [...], y hacerlo en unas condiciones en el que un proyecto ya sea mínimamente asumible [...]”.

Recorremos, de la mano de nuestro interlocutor, lo que hemos llamado proceso completo de producción, en el que se incluye, como también hemos detallado en otros estudios de caso, en esta misma investigación, todo el camino, detallado, cruzado, de la planificación, del análisis de riesgo, de viabilidad, del cálculo, con métricas ya homologadas y probadas, de los posibles costes (a los que ha precedido un minucioso estudio de los posibles riesgos). E incluso de la posterior función de mantenimiento de nivel tres que se asigna a la empresa. Y junto a ello la posible externalización –y encargo a una empresa diferente- de algunas de las pruebas, tanto unitarias como de integración, que son –creemos- uno de los problemas centrales en la recomposición del *puzzle*, como algunos interlocutores lo definen, de los fragmentos que han sido atribuidos a equipos distintos, tanto en esta fábrica de software, en esta localización concreta, como a otras fábricas que pueden estar localizadas en otros lugares, más o menos distantes. Aquí se nos hablará, en más de una ocasión del centro de Madrid, o del de Bratislava, Panamá, o en tiempo pasado de Buenos Aires.

53 Hemos consultado directamente y en detalle, tanto la metodología abreviada, como la completa, recorriendo todos los requisitos, los requerimientos, los instrumentos de recogida de la información, etc., con el fin de comprender la forma de funcionamiento de la organización del trabajo de los proyectos en esta empresa. Para ello pudimos acceder, por gentileza de la empresa, a la Intranet de la misma, consultar e imprimir para su estudio la compleja trama que compone esta metodología, que, como nos recordaba nuestros interlocutores, tanto el director de la empresa, como este gestor, es un mecanismo pesado que subsume, y amplía Métrica Tres, y, desde luego, como se verá a continuación en el texto, también la certificación adquirida por Alar en CMMI nivel 3. Conste aquí nuestro agradecimiento de nuevo. JJC.

Al hilo de las pruebas unitarias, previstas en la metodología, se nos recuerda, que “a esto nos ha obligado CMMI 3”. “Pero es que CMMI tres, lo que te obliga, es a documentar que has hecho la prueba. Es decir, ¿qué datos tenías de entrada?; ¿Qué datos eran los esperados?; ¿Y qué datos has obtenido de salida?. Eso, hacerlo en un proyecto, medianamente grande, yo he calculado que aproximadamente, el sobrecoste de documentaciones sería un 20%. Una quinta parte del tiempo de documentación se la llevarían las Pruebas Unitarias. Pero, bueno, ¡es lo que hay!.[...] La gente de calidad se encarga de que cada uno de los proyectos que yo ataque, tenga un plan de calidad, o un plan de proyecto y un plan de calidad, que sea conforme a ISO, a CMMI o a cualquier otra certificación que tenga. Normalmente nadie te pide que documentes las Pruebas Unitarias, porque además ten en cuenta que una Prueba Unitaria, pues un proyecto de doce meses hombre, puede tener, aproximadamente 500 puntos función. Pues un punto función requerirá aproximadamente unas 5 Pruebas Unitarias, aproximadamente. Es decir, para el trabajo de una persona, al año, tendría que documentar ¡2.500 pruebas!.

[-¿Y los clientes?]

-Saben que tiene disponible nuestra metodología. Saben que en un momento determinado, si lo han pactado así, lo pueden requerir. Pero eso, no es una información útil para el cliente, a él que más le da la prueba unitaria que le hayas hecho y tal. El cliente, sobre todo, está interesado en las pruebas de integración, en las pruebas de volumen. Está muy, muy, muy interesado en las pruebas de volumen, porque no es lo mismo probar una aplicación con 2 usuarios que con 2000, entonces las pruebas de volumen normalmente le son mucho más importantes. Y también hay pruebas de estrés, que es llevar al límite, un poquito [el sistema], sobrecargarlo, para ver cómo se comporta, sobre todo para ver, saber si tiene, normalmente efectos laterales en otras aplicaciones que están funcionando en el mismo servidor”.

6. Los equipos, sin perder la “cabeza”: el proyecto de atención primaria en sanidad

Para nuestro interlocutor, con más de de catorce años de experiencia en la gestión y construcción informática, de ellos seis en Alar,

donde ha constituido un equipo que perdura a lo largo del tiempo, y que no dudará en presentar como su mejor ‘baza’, la preferencia por partir, como hace en el caso del Servicio de Atención Primaria que nos sirve de ejemplo concreto, por no perder la parte del proceso que se puede resumir bajo el epígrafe de la toma de requisitos, es fundamental. Esta distancia, esta separación, esta pérdida en la división del proceso de producción, es una primera construcción de una separación que, en los términos clásicos de la sociología del trabajo, se pueden identificar como la separación entre concepción y ejecución. Más tarde vendrán otras divisiones, más ancladas ya en el mismo proceso de trabajo, incluida la pérdida del control o de la vigilancia sobre su propio trabajo y creatividad. Pero, esta es desde luego la primera, la separación entre concepción y ejecución.

Recogemos, en primer lugar, su ideal (aplicado en la práctica, no hablamos de un ideal que no se *aplique*) de lo que puede ser una organización que, en su decir, y en los datos que hemos podido recoger, suponen un camino en el que las carreras profesionales, el despliegue de conocimientos acumulados, no sólo en la propia técnica de resolución de los problemas de codificación, sino también del funcionamiento real del área en que trabajan, la sanidad, llevan a una especialización que es, en el sentido que la palabra se usa en italiano para la organización del trabajo, a una mayor cualificación y despliegue profesional. Y, por cierto, a un mayor éxito empresarial.

“Este para mí es –nos dice- lo ideal. Aquí el tamaño del grupo] está en función del tamaño del proyecto, o sea el número de equipos. Los equipos tienen, para mí, cuatro personas, un jefe de equipo, o sea en realidad son cinco. La responsabilidad es por equipo. Hay un jefe de proyecto; hay un arquitecto, un analista. Estos suelen ser transversales a los equipos. Hay una tarea de diseño que o bien lo hace el analista o bien está especializado, pero son tareas separadas, aparte. Y la comunicación va entre el diseñador y los equipos. Una vez que entra en producción [el proyecto], esta persona se queda de consulta, cuando hay alguna duda, se queda de consulta. Y ya el trabajo lo realiza el diseñador, que a mí me gusta también, que sea el jefe de proyecto. Porque este es el que está al cargo, es el responsable de todo”.

El gerente, nuestro entrevistado, es responsable de varios proyectos. En el proyecto de atención primaria hay tres equipos, es decir, quince personas, organizados con la estructura ya indicada, y bajo la responsabilidad de un jefe de proyecto. La metodología utilizada, “incremental con maqueta”, lleva a una forma de trabajo en la cual, el vaivén entre toma de requisitos, análisis, presentación de la maqueta al cliente final, discusión y puesta en común, y vuelta a la fábrica es reiterativo. Y se prevé que tenga lugar en tres fases distintas, de las cuales, la primera ya ha terminado cuando realizamos nuestro trabajo de campo.

“En este momento hay disponible una maqueta que refleja exactamente que es lo que quiere el cliente. En eso es en lo que van a trabajar el analista, (que además en este caso ha venido conmigo a las reuniones para tener información de primera mano), con el diseñador. [Tardarán unas] dos semanas a lo sumo, en trocear el desarrollo, y de ahí empezarán a dar trabajo a los equipos. Estas personas se encargan de montar el entorno de desarrollo, es decir habilitar el espacio en el controlador de las versiones, dónde van a ir todos los ficheros. Establecer qué jerarquía, qué estructura de directorios, va a tener el proyecto. Van a asignar los permisos correspondientes a las personas, para que puedan escribir cada uno en su sitio. Van a organizar el proyecto desde el punto de vista de sistemas. Y esta gente ya empieza a trabajar, con el trabajo que le proporciona el diseñador, que va, simplemente, con la metodología: casos de usos, escenarios y demás. Que es lo que recibe cada jefe de equipo [...].”

Este tipo de organización de la producción de software permite estabilizar mucho más a la gente, generando confianza y saber sobre un área como la sanidad, donde, según nuestro entrevistado, se reúne calificación técnica y conocimiento del “negocio”, saberes funcionales. La duración de este proyecto de sanidad se estima que será de un año para el desarrollo completo, más el mantenimiento evolutivo o correctivo. Para lo cual, parte del equipo que ha participado es una baza fundamental.

Se trata, en los casos de jefes de equipo, de ingenieros informáticos senior que tienen mucha experiencia, tres años al menos, que se combinan en los equipos con personas con menos experien-

cia, un año, para componer la unidad de trabajo de referencia. Que “tienen don de gentes, saben cuándo de hecho, hay que pedirle un esfuerzo a la gente, pues lo hacen, y lo hacen de una forma, pues, razonable, y la gente le entra ien fin!. Es distinto ¿no?. Y suelen, a lo mejor trabajan muy bien con paneles de documentación, saben leerla, saben leer, no les cuesta leer documentación y producirla, no les cuesta trabajo monitorizarlo dentro del área”.

Un paso más en contra de la parcelización y rutinización del trabajo es el dotar a cada equipo de un trabajo con sentido, una unidad funcional, o un caso de uso, coherente: “el trabajo que recibe éste equipo no es un trabajo inconexo, es un bloque funcional que tiene sentido en si mismo; yo le puedo cercenar el proyecto, pero esto que tiene aquí, tiene sentido, o sea no le doy [algo] decapitado ¿de acuerdo?”. “Él tiene que ser capaz de entender, que esto [su trabajo], lo va a usar este circuito funcional [...], pero esto es un circuito funcional completo. Es decir, es un procedimiento que tiene sentido por si mismo, que tiene su inicio como un evento, detectable sencillamente, y que tiene que tener un desarrollo completo.

Entonces, a mí me gusta que los equipos trabajen de esta manera. Luego las tareas de integración son infinitamente más sencillas, evidentemente. Si yo troceara de otra forma la funcionalidad, tendría muchos problemas luego a la hora de juntarlas Y, sobre todo, no tendría a nadie, que tenga una visión del conjunto. Entonces yo, lo que requiero es, que esta persona sepa leer muy bien la documentación, sepa leer un análisis y un diseño funcional perfectamente”.

“Aquí el barniz funcional, nuevamente es importante. Normalmente si esta persona, jefe de equipo, ha estado trabajando en varios proyectos de salud, ya tiene mucho aprendido del concepto, de los conceptos que se trabajan en salud, de tal forma que él se hace responsable de todo esto, del paquete funcional.

Y por supuesto que tiene que entrar en el programa. Porque él es el encargado de revisar el código de éstos [los programadores de su equipo]. A lo mejor él, no termina picando demasiado código; qué también ieh!. Ahora,ieso sí!, para revisar código hace falta tener experiencia, haber visto mucho código. Para detectar qué problemas...”

7. Conclusión

7.1. Tendencias claras, pero matizadas

Al cerrar esta investigación, y como el lector ha podido comprobar, las tendencias puestas en evidencia, en el despliegue en España de las *fábricas de software*, son, matizadamente, muy semejantes a las que se detectan en la literatura y en la *realidad* internacional. En efecto, como planteamos en las primeras líneas de este trabajo, una de nuestras preocupaciones fundamentales en la investigación era, y es, el acercarnos a lo que *realmente* sucede. Al cómo se desarrollan las nuevas organizaciones productivas en la fabricación de software, para así poder identificar, aunque fuera sumariamente, las grandes líneas de tendencia del *destino*, del presente y del futuro que espera a los trabajadores del sector del software. Unos trabajadores y trabajadoras que resultan ser emblemática representación de cuanto se discute actualmente sobre el porvenir del trabajo en la sociedad de la información.

Nos planteamos en la introducción a este estudio una serie de preguntas. Eran también atalayas desde las que miramos a esta realidad del trabajo fluido en la sociedad de la información. Pero, asimismo queríamos dar indicaciones precisas al lector o lectora de los intereses intelectuales y de política de aplicación de los resultados científicos a la realidad que nos rodea, que han movido nuestro trabajo. Esas preguntas han surgido de la investigación que hemos llevado a cabo, sí. Pero también, claro está, de las preguntas que otros investigadores e investigadoras han planteado, y en algún modo, han conseguido responder, aunque fuera parcialmente, con sus estudios.

Su núcleo fundamental está vinculado a la exploración de los tipos de trabajo *trasladados*, de las posibilidades para los lugares donde se desplazan o crean estas nuevas *factorías*, del papel que pueden jugar en el fomento o la creación misma de círculos virtuosos de creación de riqueza y de trabajo decente y cualificado. De posibilidades para los *territorios sociales* en los que se implantan. De los eventuales futuros de esperanza para los miles de jóvenes que ponen en esos puestos de trabajo sus ilusiones y sus saberes. Lo que hemos puesto en evidencia, matizada, como decimos, es

que la tendencia a separar concepción de ejecución, con una reiteración renovada de la división del trabajo entre empresas (o entre centros de trabajo de la misma empresa), es una marca fuerte de los desarrollos en curso. La parte más ‘noble’ –y que condiciona el resto del proceso de trabajo–, la toma de requisitos, el análisis, el contacto directo con el cliente final queda en un lado. En el otro, en las *factorías*, tendencialmente, acaba llevándose a cabo tan sólo (i) el “desarrollo puro y duro”, como se nos ha dicho. El ‘picar código’. Con contrastes y tensiones como el último caso desarrollado en profundidad, la Sanidad en ALAR, que no hace sino mostrar las contradicciones, límites y problemas que *también* plantea la división extrema del trabajo.

A la vez una tendencia absolutamente imperante es la de que los estándares, que parecen implantarse como una forma de organización del trabajo *inevitable*, marcan las pautas de la división del trabajo en las empresas o centros de trabajo concretos. Como un paradigma que limita las posibilidades de ‘nuevas formas de organización’. Pero también, como han destacado otros autores, aunque nosotros no hayamos encontrado trazas consistentes en nuestros estudios de caso, como posibilidad de ampliación del objeto de trabajo, de mayor socialización de los trabajadores, de mayor interrelación entre ellos, etc. La tendencia hacia la simplificación del trabajo que se traslada a las factorías, que se desplaza de los grandes centros parece así afirmarse, incluso en las tendencias a contratar como *programadores junior* en estas empresas a técnicos medios, más que a superiores. O a personas con Formación Profesional frente a Diplomados o Licenciados universitarios. Y como un reflejo de esas tendencias hacia la separación del trabajo de concepción del de ejecución, y hacia la rutinización y simplificación, en un contexto de organización estandarizada y muy formalizada, las empresas se acaban enfrentando a los problemas de rotación y abandono del trabajo. Para algunas de las empresas que hemos estudiado en profundidad, las políticas de gestión de recursos humanos tienen como preocupación fundamental el ser capaces de construir un itinerario profesional para sus trabajadores, de modo que las altas tasas de rotación se reduzcan. Es más, como se nos ha explicado en detalle, las empresas se plantean, también, el llevar a sus fábricas de software productos y líneas de

trabajo más sofisticadas que permitan construir esos itinerarios de carrera con fundamentos reales.

También hemos dado cuenta en nuestro estudio de algunas reflexiones de más hondo calado, y que abren problemáticas de fondo en la sociología del trabajo. Aunque sólo hayamos podido ahora esbozar argumentos. Abrir interrogantes para futuras investigaciones. En efecto, en este estudio de los procesos de creación, producción, fabricación, o desarrollo del software hemos abordado, necesariamente, esto es, porque lo impone el objeto teórico, el trabajo inmaterial, cualificado, y el referente empírico, algunos de los problemas clásicos de la organización del trabajo, aunque sea sólo de forma tentativa en esta primera aproximación: la implicación de los trabajadores; las formas de colaboración y constitución del trabajador colectivo y de su coordinación; la constitución de equipos de trabajo, o del trabajo en equipo y de la vinculación entre ellos; los llamados “equipos virtuales” y los problemas de coordinación e integración en sistemas de trabajo complejo.

En la localización de los estudios de casos hemos buscado aquellos lugares que, como ya ha mostrado la investigación, y en casos en los que la comparación es un buen punto de partida, tenemos los elementos fundamentales interpretativos para enmarcar los procesos fragmentados de la producción de software. Las empresas que hemos estudiado están localizadas en zonas industriales, lo que refuerza su aparente identificación con una *fábrica*: un polo de desarrollo; un polígono industrial; un edificio de oficinas...

Los edificios pueden ser de nueva planta, diseñados específicamente para organizar el desarrollo de software ‘en cadena’, como se dirá repetidamente en las entrevistas que hemos realizado. Aunque la metáfora no sea una descripción muy adecuada de la disposición y encadenamiento de la producción en cada caso, si cabe decir que, en todos los casos, la disposición espacial, e incluso el diseño arquitectónico han sido pensados *ex profeso* con mayor rigor, conciencia y propósito de lo que es habitual en otros sectores productivos. Por ello, sin excepción, en cada caso se enfatizará que se diseñó específicamente para ese fin. Que son una estructura especialmente pensada para una determinada organización y división del trabajo, que *tiende* hacia la cadena productiva, hacia la fragmentación.

Como se ha dicho en más de una ocasión, en referencia a los *greenfields*, localizaciones de este tipo sirven, además, para aislar la planta, el proceso productivo desgajado de la producción global, de las otras plantas productivas de la misma empresa, o de la misma ‘industria’ o sector⁵⁴. Pero, sobre todo, estas deslocalizaciones internas, dentro del territorio español, que ya gustan de llamar los responsables *nearshore*, atrae nuestra atención hacia la influencia que han podido tener en la ubicación espacial, y social, de las mismas, las facilidades otorgadas por los gobiernos, nacionales, regionales, o locales. O todos simultáneamente: los incentivos fiscales o las subvenciones directas o indirectas a la contratación. Las políticas de fijación de los y las jóvenes cualificados a su territorio de origen. La (mayor) baratura del coste de la mano de obra y la falta de experiencia reivindicativa y sindical. La existencia de infraestructuras tecnológicas avanzadas, en nuestro caso, de fácil uso y ofrecidas a precios muy por debajo de los de mercado. Auténticas *autopistas públicas de la información* sin peaje. Instituciones de formación prácticamente gratuitas. Institutos tecnológicos o de desarrollo económico prontos a solventar estudios de viabilidad o de marketing. Un conjunto, en fin, de tramas sociales que construyen las posibilidades de un territorio en los términos que se han venido utilizando, en la ciencia social, en la economía política y en las políticas económicas, desde los ya lejanos tiempos de los distritos industriales, los medios innovadores, o los sistemas locales de desarrollo endógeno.

7.2. El trabajo fluido en la sociedad de la información

Mucho de lo que hoy se presenta como *nueva* interpretación en las ciencias sociales está anclado, limitado y potenciado, a la vez, por lo que fue la corriente dominante, la *mainstream* de las ciencias sociales. Con maquillajes más preocupados por hacerse con una marca, un *label*, en los casilleros de las ciencias sociales respetables que de interpretar la realidad.

Así, se dirá, como en los manuales irrelevantes, que las empresas no utilizan la “división internacional del trabajo”, sino que

54 Véase Helen Newell, “Training in greenfield sites”, en Helen Rainbird, 2000, pp. 101-125.

se han hecho “empresas manufactureras sin manufacturas”: detentan únicamente la marca, y todo lo que antes se llamaba trabajo productivo, ahora se hace en lugares recónditos, allende los mares, en China, o en algún perdido polígono industrial de una zona no desarrollada de Europa.

Se le llame como se le llame, los cambios de palabras sólo pueden ocultar la realidad. No se avanza mucho cambiando el nombre del trabajo precario por ‘contingente’. Y, bajo esa *nueva* división internacional del trabajo, bajo esa nueva forma fluida que toma el trabajo cualificado, lo que importa hoy en día, a las ciencias sociales y a los ciudadanos y más aún a los responsables de tomar decisiones y orientar la política de desarrollo, es el ser capaces de discernir las tendencias en curso.

Porque hoy tenemos que constatar que –aún perpetuándose la tendencia a la dispersión y externalización de los trabajos ‘tradicionales’, industriales o de servicios- también los trabajos muy cualificados, y en muchos casos, los más emblemáticos, como puede ser la fabricación de programas informáticos, se desplaza también. Se dispersa, por todo el ancho mundo, cambiando procesos de trabajo, situaciones sociales locales, composiciones del trabajador colectivo, desregulaciones, nuevas formas de control y de posibilidades del trabajo y de los trabajadores y trabajadoras. No descubrimos una novedad, pues, como muestra la literatura científica, especialmente en este punto, para el objetivo de nuestra investigación, esta no es una tendencia surgida de pronto, sino que lleva muchos años gestándose. Un amplio conjunto de países, hoy integrados en la cadena mundial de la producción de programas informáticos, de servicios a las empresas, de atención a distancia, etc., o aspirantes a ello, dan una idea precisa de la gran cantidad de trabajadores cualificados que *pueden*, gracias, es cierto, a los desarrollos tecnológicos y de comunicación a distancia, integrarse en el *trabajador colectivo de la producción de software*. Y, desde luego, no parece que los sueños de la *sociedad de la información* sean el horizonte más probable de ese nuevo *mundo feliz* de la economía de los servicios.

Recebido em 7 de fevereiro de 2008

Aprovado em 10 de junho de 2008

Referências

ABRAMO, L. (ed.). **Trabajo decente y equidad de género en América Latina**. Santiago: Oficina Internacional del Trabajo, 2006.

ADLER, P. **The Discipline of Process: the Transformation of Software development**. confidential Draft. Marshall School of Business, University of Southern California, 2002.

_____. **Practice and Process: the Socialization of Software Development**". Draft paper. University of Southern California, 2003.

_____. **Skills Trends under Capitalism and the Socialisation of Production**. In: WARHURST, C.; GRUGULIS I. & KEEP E. (eds.). **The Skills that matter**. Houndmills: P. Macmillan, 2004.

_____. **The evolving Object of Software development**. **Organization**, Los Angeles, v. 12, n. 3, p. 401-435, 2005.

AETIC y DMR CONSULTING. **Las tecnologías de la sociedad de la información en la empresa española 2004**. Madrid: Cyan S.A, 2005.

AETIC. **Análisis y propuestas de delimitación del sector de la electrónica y de las tecnologías de la información y las telecomunicaciones**. Observatorio Industrial de Electrónica, Tecnologías de la Información y Telecomunicaciones. Madrid: Ministerio de Industria, Turismo y Comercio, 2005. Disponible en: www.mityc.es/NR/rdonlyres/2F431DB2-FACA-4C76-9295-C8F086C2FD95/0/03AETIC_AnalisisDelimitacion.pdf.

ANEESH, A. **Skill saturation: Rationalization and post-industrial Work**. **Theory and Society**, New York, v. 30, n. 3, p. 363-396, June, 2001.

ARORA, A.; ARUNACHALAM, V.; ASUNDI, J. ; FERNANDES, R. **The indian Software Services Industry**. **Research Policy**, London, v. 30, n. 8, p. 1267-1287, October, London, 2001.

BAGNASCO, A. **De la sociología del trabajo a la sociedad**. In: CASTILLO, J. (ed.). **El trabajo del futuro**. Madrid: Complutense, 1999.

BALDRY, C.; BAIN, P. & TAYLOR, P. **Bright satanic Offices: intensification, Control and team Taylorism**. In: THOMPSON P. & WARHURST, C. (eds.). **Workplaces of the future**. London: Macmillan, 1998.

BARRET, R. Laboring under an Illusion. The Labor Process of software-development in the Australian Information Industry. **New Technology, Work and Employment**, Oxford, v. 16, n. 1, p. 18-34, 2001.

_____. Working at Webboyz: An Analysis of Control over the Software Development Labour Process. **Sociology**, London, v. 38, n. 4, p. 777-794, october, 2004.

BARRY, F. & CURRAN, D. Enlargement and the European Geography of the Information Technology Sector. **The World Economy**, Dublin, v. 27, n. 6, p. 901-922, march, 2004.

BEIRNE, M.; RAMSAY, H. & PANTELI, A. Developments in computing Work: Control and Contradiction in the Software Labour Process. *In*: Thompson, P. & Warhurst, C. **Workplaces of the future**. Houndmills: Macmillan, 1998.

BENJAMIN, W. **Sens unique**. Paris: Les Lettres Nouvelles, 1978.

BOLOGNANI, M. & CORTI, E. **La fabbrica del software**. Milán: F. Angeli, 1984.

BOLTANSKI, L. & CHIAPELLO, È. **El nuevo espíritu del capitalismo**. Madrid: Akal, 2002.

BOLTON, S. 2005. **Emotion management in the workplace**. Houndsmill: P. Macmillan, 2005.

_____. Una tipología de la emoción en el lugar de trabajo. **Sociología del trabajo**, Madrid, nueva época, n. 58, p. 3-29, Primavera, 2006.

BONAZZI, G. Il cambiamento del paradigma organizzativo nel 20^o secolo: alcune ripercussioni sulle convinzioni profonde. **Sociología del Lavoro**, Bolonia, n. 100, IV trimestre, 2005.

BONO, A. Deslocalización extraterritorial de empleos del sector servicios. Sentidos y transformaciones del trabajo. **Sociología del trabajo**, Madrid, nueva época, n. 56, p. 3-32, Invierno, 2006.

BOURDIEU, P. **Science de la science et réflexivité**. Paris: Raisons d'Agir, 2001.

_____. L'objectivation participante. **Actes de la recherche en Sciences Sociales**, Paris, n. 150, p. 43-57, diciembre, 2003.

BOWRING, F. Post-fordism and the end of work. **Futures**, Cardiff, v. 34, p. 159-172, march, 2002.

BURAWOY, M. The extended case method. **Sociological Theory**, New York, v. 16, n.1, p. 4-33, march, 1998.

_____. Revisits: an outline of a theory of reflexive ethnography. **American Sociological Review**, v. 68, n. 5, october, p. 645-679. Washington: American Sociological Association, 2003.

_____. Public Sociologies: a Symposium from Boston College. **Social Problems**, New York, v. 51, n. 1, p. 103-130, september, 2004.

_____. Public sociologies: contradictions, dilemmas and possibilities. **Social Forces**, Chapel Hill, v. 82, n. 4, p. 1603-1618, 2004.

_____. For public Sociology. 2004 Presidential Address. **American Sociological Review**, Washington, v. 70, april, p. 4-28, 2004.

_____. Por una sociología pública. **Política y Sociedad**, Valencia, v. 42, n. 1, p. 197-225, 2005.

BUSTOS, R. Métodos de desarrollo del software: el desafío pendiente de la estandarización. **Teoría, ciencia, arte y humanidades**, Concepción, v. 12, p. 12-23.

CALDERON, J. Repensar la cuestión de la resistencia en el trabajo, o buscando al trabajador perdido: un estudio de caso en el sector del telemarketing. **Sociología del trabajo**, Madrid, nueva época, n. 56, p. 33-73, primavera, 2006.

CAPECCHI, V. La crisis del 'modelo emiliano': el aumento de los trabajos atípicos y de riesgo", **Sociología del trabajo**, Madrid, nueva época, n. 48, outono, p. 17-44, 2003.

CAPPELLI, P. Why is it so hard to find information technology workers? **Organizational Dynamics**, v. 30, n. 2, p. 87-99, 2001.

CARRERA, S. El Prosoft y la industria del software en México, **Comercio Exterior**, México D.F., v. 55, n. 9, p. 754-763, 2005.

CASEY, C. Knowledge-based economies, organizations and the sociocultural regulation of work. **Economic and Industrial Democracy**, v. 25, n. 4, p. 607-627, 2004.

_____. **Informatización, trabajo y empleo en las pequeñas empresas españolas.** Madrid: Ministerio de Trabajo - Dirección General V de la Comisión Europea, 1989.

CASTILLO, J. **A la búsqueda del trabajo perdido.** Madrid: Tecnos, 1998.

_____. **En la jungla de lo social.** Reflexión y oficio de sociólogo. Madrid-Buenos Aires: Miño y Dávila, 2003.

_____. **Los estragos de la subcontratación.** La organización del trabajo como factor de riesgo laboral. Madrid: Dirección General de Trabajo de la CAM y UGT-MADRID, 2003.

_____. **Contra los estragos de la subcontratación: trabajo decente. Sociología del Trabajo,** Madrid, nueva época, n. 54, p. 3-37, Primavera, 2005.

_____. **El trabajo recobrado.** Una evaluación del trabajo realmente existente en España. Buenos Aires-Madrid: Miño y Dávila, 2005.

CASTILLO, J. & CALLE, P. **Los obreros del polo: una cadena de montaje en el territorio.** Madrid: Complutense, 2003.

CEPAL. **Software and information services industry.** Study in selected productive clusters in the Argentine Republic. Japan International Cooperation Agency - Cepal Buenos Aires, [Equipo de investigación: G. Anlló, G. Bezchinsky, A. López, A. Ramos, A. Sacroisky], 2003.

CHIARVESIO, M.; DI MARIA, L. & MICELLI, S. From local Networks of SMEs to virtual Districts? Evidence from recent Trends in Italy. **Research Policy**, London, v. 33, p. 1509-1528, march, 2004.

CHUDNOVSKY, D.; LOPEZ, A. & MELITSKO, S. **El sector del software servicios informáticos (SSI) en la Argentina:** situación actual y perspectivas de desarrollo. Documento de Trabajo, 2001.

COE, N. Internationalisation, Diversification and spatial restructuring in transnational Computer Service Firms: Case Studies from the U.K. Market. **Geoforum**, Vancouver, v. 28, n. 3-4, p. 253-270, august, 1997.

COE, N. & BUNNELL, T. Spatializing' knowledge communities: towards a conceptualization of transnacional innovation networks. **Global Networks**, Manchester, v. 3, n. 4, p. 437-456, 2003.

CORNFIELD, D. Tendencias mundiales recientes en la Sociología del Trabajo. *In*: TOLEDO, E. (coord.). **Tratado latinoamericano de Sociología**. Barcelona: Anthropos, 2006.

CUMBERS, A. & MACKINNON, D. 2004. Introduction: clusters in urban and regional development. **Urban Studies**, Glasgow, v. 41, ns. 5/6, p. 959-969, may, 2004.

CUSUMANO, M. Sifting economies: from craft production to flexible systems and software factories. **Research Policy**, London, v. 21, n. 5, p. 453-480, 1992.

_____. **The Business of Software**. New York: Free, 2004.

DAYASINDHU, N. Embeddedness, knowledge transfer, industry clusters and global competitiveness: a case study of the indian software industry. **Technovation**, Bangalore, v. 22, n. 9, p. 551-550, september, 2002.

FIDANZA, E. La jaula de hierro cien años después: consideración acerca de una metáfora perdurable. **Estudios Sociológicos**, México D.F. v. XXIII, n. 69, setiembre, p.845-855, 2005.

FORO DE SOFTWARE Y SERVICIOS INFORMÁTICOS. **Software y servicios informáticos. Libro azul y blanco Plan Estratégico de SSI 2004-2014**. Plan de Acción 2004-2007. Buenos Aires: Ministerio de Economía y Producción, 2004.

FORSMAN, M. & SOLITANDER, N. Knowledge transfer in clusters and networks. An interdisciplinary conceptual analysis. **Journal of International Business Studies**, pp. 1-23, 2003.

FOUNTAIN, J. Constructing the information society: women, information technology, and design. **Technology in Society**, Cambridge, n. 22, p. 45-62, 2000.

FRÖBEL, F.; HEINRICHS, J. & KREYE, O. **La nueva división internacional del trabajo**. Madrid: Siglo XXI, 1980.

GALLINO, L. **Informatica e qualità del lavoro**. Turín: Giulio Einaudi, 1983.

GAMELLA, M. (ed.). **La tecnología del software: Temática y situación en España**. Madrid: Fundesco, 1985.

_____. **Tratado latinoamericano de Sociología**. Barcelona: Anthropos, 2006.

GARCÍA, J. Trabajo profesional y renuncia a la universalidad fáustica. Goethe en *La ética protestante y el espíritu del capitalismo*. In: RODRIGUEZ, J. (Ed.). *En el centenario de La ética protestante y el espíritu del capitalismo*. Madrid: Centro de Investigaciones Sociológicas, 2005.

GEISLER, E. & RUBENSTEIN, A. The successful implementation of application Software in new production Systems. *Interfaces*, Cincinnati, v. 17, n. 3, p. 18-24, 1987.

GEREFFI, G.; HUMPHREY, J.; STURGEON, T. The governance of global Value Chains. *Review of International Political Economy*, London, v. 12, n. 1, p. 78-104, february, 2005.

GEREFFI, G. & STURGEON, T. **Globalization, Employment, and economic Development**: a briefing Paper. Sloan workshop Series in Industry Studies, Rockport, Massachusetts Institute of Technology, Industrial Performance Center, Working Paper Series, Massachusetts, 14-16 de junio de 2004.

GLOVER, L.; NOON, M. "Shop-floor workers" responses to quality management initiatives: broadening the disciplined worker thesis. *Work, Employment and Society*, Glasgow, v. 19, n. 4, p. 727-745, march, 2005.

GLUCKSMANN, M. Call configurations: varieties of call centre and divisions of labour. *Work, Employment and Society*, Glasgow, v. 18, n. 4, p. 795-811, march, 2004.

GOETHE, J. *Los años de aprendizaje de Wilhem Meister*. Madrid: Cátedra, 2000.

GRABHER, G. (ed.). **The embedded firm**. On the socioeconomics of industrial Networks. New York: Routledge, 1993.

GRIMALDI, R. & TORRISI, S. Codified-tacit and general-specific knowledge in the division of labour among firms. A study of the software industry. *Research Policy*, London, v. 30, n. 9, pp. 1425-1442, December, 2001.

HAMDE, K. Teamwork: fashion or institution?. *Economic and Industrial Democracy*, Uppsala, v. 23, n. 3, p. 389-420, august, 2002.

HAMPSON, I. & JUNOR, A. Invisible work, invisible skills: interactive customer service as articulation work. *New Technology, Work and Employment*, Oxford, v. 20, n. 2, p. 166-181, october, 2005.

HANSEN, B.; ROSE, J. & TJOMEHOJ, G. Prescription, description, reflection: the shape of the software process improvement field. **International Journal of Information Management**, Cambridge, v. 24, n. 6, p. 457-472, 2004.

HARDILL, I. & GREEN, A. Remote working: altering the spatial contours of work and home in the new economy. **New Technology, Work and Employment**, v. 18, n. 3, p. 212-222, march, 2003.

HARRISON, B. **Lean and mean**. The changing Landscape of corporate Power in the Age of flexibility. New York: Basic Books, 1994.

HARVEY, D. **The condition of postmodernity**. An inquiry into the origins of cultural change. Oxford and Cambridge: Mass, 1989.

HEATH, C.; KNOBLAUCH, H. & LUFF, P. Technology and social interaction: the emergence of “workplace studies”. **British Journal of Sociology**, London, v. 51, n. 2, p. 299-320, june, 2000.

HELLANDER, N. **Value-creating networks**: an analysis of the software component business. Tesis presentada en diciembre de 2004 en la Faculty of Economics and Business Administration. Oulu: University of Oulu, 2004.

HOCH, D.; ROEDING, C.; PURKERT, G.; LINDNER, S. & MÜLLER, R. **Secrets of Software Success**. Management Insights from 100 Software Firms around the World. Boston: Harvard Business School, 2000.

HOCHSCHILD, A. Emotion work, feeling rules, and social structure. **American Journal of Sociology**, Chicago, v. 85, n. 3, p. 551-575, november, 1979.

HUGHES, J.; O'BRIEN, J. ; RANDALL, D.; ROUNCEFIELD, M. & TOLMIE, P. Some “real” Problems of “virtual” Organisation. **New Technology, Work and Employment**, Oxford, v. 16, n.1, 2001, p. 49-64, october, 2001.

HUMPHREY, W. Three Process Perspectives: Organizations, Teams, and People. **Annals of Software Engineering**, v. 14, ns. 1-4, p. 39-72, december, 2002.

HYMAN, J.; BALDRY, C.; SCHOLARIOS, D. & BUNZEL, D. Work-life Imbalance in Call Centres and Software Development. **British Journal of Industrial Relations**, London, v. 41, n. 2, p. 215-239, june, 2003.

HYMAN, J.; SCHOLARIOS, D. & BALDRY, C. Getting on or getting by? Employee Flexibility and coping Strategies for Home and Work. **Work, Employment and Society**, Glasgow, v. 19, n. 4, p. 705-725, 2005.

ILAVARASAN, P. & SHARMA, A. Is Software work routinized? Some empirical Observations from Indian Software Industry. **The Journal of Systems and Software**, New York, n. 66, p. 1-6, april, 2003.

INE (Instituto Nacional de Estadística). **Encuesta del Sector Servicios 2003**. 2005. (Disponible en la red: www.ine.es)

ISAKSEN, A. Knowledge-based Clusters and urban Location: the clustering of Software Consultancy in Oslo. **Urban Studies**, Glasgow, v. 41, n. 5/6, p. 1157-1174, mayo, 2004.

KEMPLE, T. Instrumentum vocale. A note on Max Weber's value-free polemics and sociological Aesthetics. **Theory, Culture and Society**, British Columbia, v. 22, n. 4, p. 1-22, 2005.

KRAAN, K. **Virtual work**: new Insights into the high Road. Paper to be presented at RC30 Sociology of Work, Session 01 "Working Conditions in a globalized World", XVI World Congress of Sociology, Durban, South Africa, 23-29, july 2006.

KRAFT, P. The industrialization of Computer Programming: from Programming to "Software Production". ZIMBALIST, A. **Case studies on the labor process**. NewYork: Monthly Review, 1979.

KRAFT, P. & DUBNOFF, S. Job Content, Fragmentation, and Control in Computer Software Work. **Industrial Relations**, Berkeley, v. 25, n. 2, p. 184-196, Primavera, 1986.

LAKHA, S. The new International Division of Labour and the Indian Computer Software Industry, **Modern Asian Studies**, Cambridge, v. 28, n. 2, p. 381-408, mayo, 1994.

LAUTIER, B. Mondialisation, travail et genre: une dialectique qui s'épuise. **Cahiers du Genre**, Paris, n. 40, p. 39-65, 2006.

LEAMER, E. & STORPER, M. 2001. The economic Geography of the Internet Age. **Journal of International Business Studies**, v. 32, n. 4, p. 641-665. In: www.jibs.net.

LEMA, R. **The role of collective efficiency in Bangalore's software-export success**. Draft paper presentado a la DRUID Academy Winter

Conference 2005, “Industrial Evolution and Dynamics”, Skorning, Dinamarca, 27-29 de enero de 2005.

LEÓN SERRANO, G.; BERNARDOS, A.; BURILLO, V.; DUEÑAS, J.; MATÍAS, C. & SÁEZ, F. **Evolución de los perfiles profesionales TIC en la sociedad del conocimiento**. Madrid: ANIEL y COIT, Ministerio de Ciencia y Tecnología, 2003.

LINDKVIST, L. Knowledge Communities and Knowledge collectivities: a Typology of Knowledge Work. **Journal of Management Studies**, Leeds, v. 42, n. 6, p. 1189-1210, septiembre, 2005.

LINHART, D. Los asalariados y la mundialización. El caso francés. **Sociología del Trabajo**, Madrid, nueva época, n. 45, p. 53-68, marzo, 2002.

_____. Ayer solidarios, hoy adversarios. Salarios amenazados y derechos sociales atacados. **Le Monde Diplomatique**. Edición española, año X, n. 125, marzo, pp. 16-17, más encarte en p. 18 “Todo comenzó en 1972...”. [Dossier “El futuro del trabajo: precariedad para todos”, pp. 16 a 26]. 2006.

LÓPEZ CALLE, P. & CASTILLO, J. **Los hijos de las reformas laborales**. Vivienda, formación y empleo de los jóvenes en la Comunidad de Madrid. Madrid: UGT-Madrid, 2004.

LÓPEZ, A. El sector de software y servicios informáticos en la Argentina: ¿es posible una inserción exportadora sostenible?. *In*: BOSCHERINI, F. & NOVICK, M. & YOGUEL, G. (comp.). **Nuevas tecnologías de información y comunicación**. Los límites en la economía del conocimiento. Buenos Aires: Universidad Nacional General Sarmiento, 2003.

MARCHINGTON, M. 2000. Teamworking and employee involvement: terminology, evaluation and context. *In*: PROCTER, S & MUELLER, F. (ed.). **Teamworking**. London: Macmillan, 2000.

MANACORDA, P. **Il calcolatore del capitale**. Una analisi marxista dell'informatica. Milán: Feltrinelli, 1976.

MANACORDA, P. **Lavoro e intelligenza nell'età microelettronica**. Milán: Feltrinelli, 1984.

MARTINS, L.; WILSON, L. & MAYNARD, M. Virtual Teams: what do we know and where do we go from here?. **Journal of Management**, Portland, v. 30, n. 6, p. 805-835, 2004.

MAY, C. Information Society, Task mobility and the End of Work". **Futures**, Cardiff, v. 32, n. 5, p. 399-416, June, 2000.

MICHELI, J. El trabajo en la sociedad de la información. El caso ilustrativo del telemercado. **Estudios Sociológicos**, México D.F. v. XXIV, n. 70, p. 197-220, enero-abril, 2006.

MIR, A; MATHEW, B. & MIR, R. The Codes of Migration: Contours of the global Software Labor Market. **Cultural Dynamics**, Austin, n.12 (1), p. 5-33, 2000.

MULHOLLAND, K. **Workplace resistance in an Irish call centre: slammn', scammin' smokin' an' leavin'**. **Work, Employment and Society**, Glasgow, v. 18, n. 4, p. 709-724, december, 2004.

NADHAKUMAR, J. Managing time in a software factory: temporal and spatial organization of IS development activities. **The Information Society**, Bloomington, v. 18, p. 251-262, 2002.

NICHOLSON, B. & SAHAY, S. Some political and cultural issues in the globalisation of software development: case experience from Britain and India". **Information and Organization**, Atlanta, vol. 11, n. 1, p. 25-43, January, 2001.

NOVICK, M. & MIRAVALLES, M. **La dinámica de oferta y demanda de competencias en un sector basado en el conocimiento en Argentina**. Documento de Trabajo LITTEC, 2002.

Ó RIAIN, S. The flexible developmental State: Globalization, information Technology, and the 'Celtic Tiger'. **Politics and Society**, v. 28, n. 2, p. 157-193, June, 2000.

_____. **The politics of high-tech growth**. Developmental Network States in the global Economy. Cambridge: Cambridge University, 2004.

_____. The Politics of mobility in technology-driven Commodity Chains: developmental Coalitions in the Irish Software Industry", **International Journal of Urban and Regional Research**, Toronto, v. 28.3, p. 642-663, September, 2004.

PANTELI, N.; STACK, J. & RAMSAY, H. Gendered patterns in computing Work in the late 1990s. **New Technology, Work and Employment**, Oxford, v. 16, n. 1, p. 3-17, 2001.

PARTHASARATHY, B. India's Silicon Valley or Silicon Valley's India? Socially embedding the Computer Software Industry in Bangalore. **International Journal of Urban and Regional Research**, Toronto, v. 28.3, p. 664-685, september, 2004.

PERLOW, L. Time to coordinate. Toward an understanding of work-time standards and norms in a multicountry study of software engineers. **Work and Occupations**, Nashville, v. 28, n. 1, p. 91-111, february, 2001.

PERRING, I. **Information Technology and Job creation: Software and the Software Industry**. Bruselas, Comisión Europea, Fast paper series, n. 13, 1983.

PERULLI, P. Organizzazione del lavoro nella produzione di software. **Quaderni di Sociología**, Torino, v. XXXIV, n. 11, p.27-46, 1988.

PIATTINI, M.; CALVO-MANZANO, J.; CERVERA, J. & FERNÁNDEZ SANZ, L. **Análisis y diseño de aplicaciones informáticas de gestión**. Madrid: RA-MA, 2004.

PORTER, M. **The competitive advantage of nations**. New York: Free, 1990.

PRASAD, M. International Capital on 'Silicon Plateau': Work and Control in India's computer Industry. **Social Forces**, Chapel Hill, v. 77, n. 2, p. 429-52, december, 1998.

PRASHANTHAM, S. & YOUNG, S. The Internet and the internationalisation of small knowledge-intensive Firms: Promises, Problems and Prospects. **International Journal of Entrepreneurship and Small Business**, v. 1, n. 1/2, p. 153-175, 2004.

PRUIJT, H. Teams between neo-taylorism and anti-taylorism. **Economic and Industrial Democracy**, Uppsala, v. 24, n. 1, p. 77-101, 2003.

PYÖRIÄ, P. Knowledge Work in distributed Environments: Issues and Illusions. **New Technology, Work and Employment**, Oxford, v. 18, n. 3, p. 166-180, 2003.

RUIZ DURÁN, C.; PIORE, M. & SCHRANK, A. 2005. Los retos para el desarrollo de la industria del software. **Comercio Exterior**, México D.F., v. 55, n. 9, p. 744-753, 2005.

SANZ, L & GARCÍA, J. El factor humano en la ingeniería del software. **Novática**, Madrid, n.179, p. 48-54, 2006.

SAYER, A. & WALKER, R. 1992. **The new social economy. Reworking the division of labor.** Oxford: Blackwell, 1992.

SCARBROUGH, H. Knowledge as Work: conflicts in the Management of Knowledge Workers. **Technology Analysis and Strategic Management**, v. 11, n. 1, p. 5-16, 1999.

SCHMITZ, H. (ed.). **Local enterprises in the global economy.** Issues for Governance and Upgrading. Cheltenham: E. Elgar, 2004.

SCOTT, A. & STORPER, M. Regions, Globalization, Development. **Regional Studies**, v. 37, n. 6 y 7, p. 579-593, 2003.

SEDISI. 1989. El sector informático en España. Madrid y Barcelona: SEDISI-Asociación Española de Empresas de Informática, 1988.

_____. **Directorio 2001**, [Consultado en papel y Cd-ROM].

SEGRESTIN, D. L'entreprise à l'épreuve des normes de marché. Les paradoxes des nouveaux Standard de gestion dans l'industrie. **Revue Française de Sociologie**, Paris, v. XXXVIII, n. 3, p. 553-585, 1997.

SELEIM, A.; ASHOUR, A. & BONTIS, N. Intellectual capital in egyptian software firms. **The Learning Organization**, v. 11, n. 4/5, p. 332-346, 2004.

SHERER, S. From supply-chain management to value network advocacy: implications for e-supply chains. **Supply Chain Management: an International Journal**, v. 10, n. 2, p. 77-83, 2005.

SORENSEN, O.; RIVKIN, J. & FLEMING, L. Complexity, networks and knowledge flow, **Research Policy**, artículo en prensa, 2006.

STURGEON, T. How we define Value Chains and production Networks?. MIT, Background paper prepared for the Bellagio Value Chains Workshop, 25 de septiembre-1 de octubre de 2000, Rockefeller Conference Center, Bellagio, Italia, 2000.

STURGEON, T. What really goes on in Silicon Valley? Spatial clustering and dispersal in modular production Networks”, **Journal of Economic Geography**, v. 3, pp.199-225, 2003.

STURGEON, T. & LEVY, F. Measuring the offshoring of Service Work and its Impact on the United States. A working Group. Propuesta de constitución, MIT, Industrial Performance Center, 2005.

SULLIVAN, C. What's in a name? Definitions and conceptualisations of Teleworking and Homeworking. **New Technology, Work and Employment**, Oxford, v. 18, n. 3, p. 158-165, 2003.

SUPERVIELLE, M & QUIÑONES, M. 2003. La incorporación del trabajador al trabajo: gestión y auto-gestión de los conocimientos en la sociedad del control (La perspectiva de la sociología del trabajo). **Revista Latinoamericana de Estudios del Trabajo**, año 8, n. 16, p. 77-116, 2003.

TAYLOR, P. & BAIN, P. India calling to the far away Towns: the Call Centre Labour Process and Globalization. **Work, Employment and Society**, Glasgow, v. 19, n. 2, p. 261-282, 2005.

TAYLOR, R. Extending conceptual Boundaries: Work, voluntary Work and Employment. **Work, Employment and Society**, Glasgow, v. 18, n. 1, p. 29-49, 2004.

TELLA, T.; BRAMS, L. ; REYNAUD, J. & TOURAINE, A. **Huachipato et Lota. Étude sur la conscience ouvrière dans deux entreprises chiliennes**. Paris : CNRS, 1966.

TOLEDO, E. (coord.). **Teorías sociales y estudios del trabajo: nuevos enfoques**. Barcelona: Anthropos - Universidad Autónoma Metropolitana de México, 2006.

VAAST, E. & WALSHAM, G. Representations and Actions: the Transformation of Work Practices with IT use. **Information and Organization**, Atlanta, v. 15, p. 65-89, 2005.

WARHURST, C.; GRUGULIS, I. & KEEP, E. (eds.). 2004. **The skills that matter**. Houndmills: Palgrave Macmillan, 2004.

WEBER, M. Remarks on technology and culture. **Theory, Culture and Society**, Nottingham, v. 22, n. 4, p. 23-38 [intervención de 1910, editada y anotada por Thomas M. Kemple], 2005.

WEBER, M. **La ética protestante y el espíritu del capitalismo**. Madrid: Alianza, 2001.

WISNER, A. Contenido de las tareas y carga de trabajo. **Sociología del Trabajo**, n. 1, p. 129-160, 1979.