

Concepto y Evolución de las Fábricas Software^{1 y 2}

Javier Garzás, Mario Piattini

www.kybeleconsulting.com
<http://kybeleconsulting.blogspot.com/>

A lo largo de la historia de la Ingeniería del Software ha aparecido repetidamente el concepto de fábrica con diferentes matices que se han ido adoptando de acuerdo a la propia evolución de la tecnología y los procesos software. En la actualidad el término ha vuelto a tomar relevancia en el sector de la industria del software, debido a las especiales condiciones socio económicas, tecnológicas y de madurez de la ingeniería del software; no obstante no debemos olvidar que, como se muestra en la Tabla 1, el concepto de fábrica software goza de una gran madurez y antigüedad.

1968	Aparece por primera vez el término “fábrica de software”
1969	Primera fábrica de software: Hitachi Software Works
1975	Fábrica software de la Systems Development Corporation
1976	Fábrica software de NEC
1977	Fábrica software de Toshiba
1979	Fábrica software de Fujitsu
1985	Fábrica software de Hitachi y de NTT
1987	Fábrica software de Mitsubishi

Tabla1. Primeros hitos en la historia de las fábricas de software

En este artículo haremos un recorrido por las principales etapas y conceptos que han ido marcando el término fábrica software, así como las principales empresas en implementar dichas estrategias, de lo cual puede obtenerse una importante visión y comprensión a la hora de constituir y evolucionar tanto fábricas como departamentos de desarrollo en la época actual.

¹ Esta obra está bajo una licencia de Creative Commons: Reconocimiento-No comercial-Compartir bajo la misma licencia 2.5 España (ver <http://creativecommons.org/licenses/by-nc-sa/2.5/es/>)

² Este artículo es un extracto de Garzás, J., & Piattini, M. (2007). *Factorías de software: Experiencias, tecnologías y organización*: Ra-ma.

1 AÑOS 70 Y 80: ORIGEN DE LAS FÁBRICAS DE SOFTWARE

El término se acuñó en el año 1968, a la vez que otros tan famosos como el término reutilización (propuesto por McIlroy de AT&T en la famosa conferencia de ese año de la OTAN sobre Ingeniería de Software). En efecto, la primera vez que se cita “fábrica de software” es en un *position paper* presentado en el congreso IFIP (International Federation of Information Processing) del año 1968 por Bemer, quien afirmaba que los gestores de software no disponían de entornos adecuados: Bemer señalaba también que es imposible que los programadores hagan buen software simplemente bajo supervisión humana, mientras que “*una fábrica, sin embargo, tiene más que supervisión humana. Mide y controla la productividad y calidad. Se mantienen registros financieros para coste y planificación*”.

Fue Hitachi la primera empresa que utilizó el término “fábrica” en 1969 cuando fundó *Hitachi Software Works*.

Por otra parte en EEUU, la *Systems Development Corporation* (que formaba parte de Rand Corporation) estableció la segunda fábrica de software entre 1975-1976, llegando a registrar esta denominación.

Durante los años setenta y ochenta en Japón se siguieron instalando fábricas de software: NEC en 1976, Toshiba en 1977, Fujitsu en 1979 y 1983, Hitachi en 1985, NTT en 1985 y Mitsubishi en 1987.

2 AÑOS 90: CASE, REUTILIZACIÓN Y PROCESOS

Durante los noventa surgen diferentes aproximaciones a las fábricas de software.

2.1 Fábricas basadas en Entornos de Desarrollo Integrados

A finales de los ochenta y principios de los noventa se implantó la primera generación de herramientas CASE (M Piattini & Daryanani, 1995), y los denominados Entornos Integrados de Desarrollo de Software (conocidos por sus siglas inglesas ISDE, *Integrated Software Development Environments*), y los Entornos de Ingeniería del Software orientados al Proceso (*PSEE, Process-centered Software Engineering Environment*).

En este caso, el contexto lo constituyen grandes empresas europeas, fabricantes de ordenadores, desarrolladoras de software y universidades.

El objetivo que se persigue es producir una arquitectura y un marco de trabajo para los ISDE. La estrategia utilizada es la de adaptar el entorno de soporte, creando una instancia de la fábrica en la organización de desarrollo. El modelado de procesos se pretenden estandarizar y soportar mediante herramientas automáticas.

2.2 Fábrica de componentes basadas en experiencia

Esta es la experiencia desarrollada en el SEL (Software Engineering Laboratory) de la NASA por Basili (V. R. Basili, 1989, , 1993; V. R. Basili, Caldiera, & Cantone, 1992), con el fin de experimentar con nuevas tecnologías en entornos de producción. Nace con el triple objetivo de mejorar la eficacia del proceso, reducir la cantidad de re-proceso y reutilizar los productos de ciclo de vida.

Ejemplos reales de factorías de experiencia son el SEL (*Software Engineering Laboratory*) del Goddard Space Flight Center de la NASA, el SEC (*Software Experience Center*) de DaimlerChrysler, o el EPIK (*Engineering Process Improvement and Knowledge Sharing*) de ICL.

2.3 Fábrica de software basada en la madurez de procesos

El contexto de esta aproximación lo constituye el modelo CMM, patrocinado por el Departamento de Defensa de EEUU con el fin de evaluar a los subcontratistas. El objetivo es crear un marco para la mejora de procesos software que permitan conseguir un proceso predecible, fiable y auto-mejorable que produzca software de alta calidad.

2.4 Fábrica de software basada en la reutilización

(Griss, 1993) señala que una reutilización efectiva requería más que tecnología para bibliotecas y código, y que utilizar sólo la metáfora de la biblioteca limitaba los resultados de la reutilización, la solución pasaba por familias de soluciones relacionadas. Este experto propone combinar la noción de fábrica de software de los años anteriores con la idea de los sistemas de fabricación flexible para dar lugar a la “fábrica de software flexible” en las que se construyen las partes para trabajar juntas y además se optimiza la producción de componentes y el ensamblado de productos con el fin de decrementar el reproceso de ingeniería. Enfatiza en prestar atención a los estándares de construcción, certificación y pruebas, haciendo trabajar de manera conjunta las guías de diseño y los procesos cuidadosamente afinados.

2.5 Fábricas de renovación de software

Al acercarse el final de la década de los noventa se agudizaron aún más los clásicos problemas del mantenimiento de software (Polo, Piattini, & Ruiz, 2003), sobre todo por las conversiones de los programas existentes debido al “problema” del año 2000 y la introducción del euro. Surgen entonces otras “fábricas” denominadas “fábricas de renovación de software”, en las que entran los programas en una especie de línea de ensamblado, pasando por una secuencia de herramientas de transformación (Brunekreef (Brunekreef & Diertens, 2002).

En (van den Brand, Sellink, & Verhoef, 2000) se presenta incluso la generación de componentes para la fábrica de renovación de software: transformadores de código, re-generadores, re-estructuradores, migradores, etc.

2.6 Fábricas enfocadas a otras técnicas de gestión de la calidad

(Swanson, Kent, McComb, & Dave, 1991) destacan la aplicación de TQM (Gestión de Calidad Total) y reutilización, así como generadores de código y herramientas CASE, buscando la flexibilidad de las fábricas de software.

También en los noventa en Japón se trasladaron métodos de la fabricación de automóviles a las fábricas de software, como el proceso de desarrollo concurrente (Aoyama, 1996) que integra conceptos convencionales de proceso-producción con los sistemas de producción “esbeltos” (*lean*) y otras técnicas de gestión basadas en el tiempo. Estas técnicas *lean* persiguen la eliminación del desperdicio dentro de una organización, combinando la planificación y los sistemas de producción.

3 AÑOS 2000: COMPONENTES, MODELOS Y LÍNEAS DE PRODUCTOS

En los años 2000 se siguió perfeccionando las técnicas de las décadas anteriores, afianzándose la ingeniería basada en modelos, el desarrollo basado en componentes, las líneas de producto y los modelos de madurez de procesos.

Así en (Li, Li, & Li, 2001) se puede encontrar una propuesta más reciente de modelo de fábrica de software para organizaciones chinas, en las que (como se puede ver en la figura), se considera que una fábrica de software se expresa como (Correa, Werner & Zaverucha, 2000):

$$\text{Fábrica Software} = (\text{Especificaciones de Gestión, Líneas de producto}) \times (\text{Procesos, Personas, Técnicas})$$

Ya que se combina, desde el punto de vista directivo, la gestión de la calidad orientada a procesos, con el punto de vista técnico, de las líneas de producto basadas en tecnologías de componentes. En esta propuesta se integran ISO9000, CMM y PSP/TSP (véase (M. Piattini, Caballero, & García, 2006)).

(Greenfield, Short, Cook, Kent, & Crupi, 2004) de Microsoft vuelven a poner de moda a nivel internacional el concepto de fábrica de software como enfoque de desarrollo de aplicaciones en el que confluyen el desarrollo basado en componentes, el desarrollo dirigido por modelos y las líneas de producto software. Lenguajes Específicos de Dominio (DSL), patrones, armazones (*frameworks*), y herramientas (incluido código y metadatos) que permiten implementar el esquema para construir un miembro de la familia de productos.

4 REFERENCIAS

Briand, L., S. Morasca, et al. (1999). "Defining and Validating Measures for Object-Based high-level design." IEEE Transactions on Software Engineering25(5): 722-743.

Chrissis, M. B., M. Konrad, et al. (2003). CMMI: Guidelines for Process Integration and Product Improvement, Addison Wesley Professional.

Fenton, N. and S. Pfleeger (1997). Software Metrics: A Rigorous Approach. Londres, Chapman & Hall.

Fowler, M. (1999). Refactoring. Improving the Design of Existing Code, Addison-Wesley.

Giles, A. E. and G. T. Daich (1995). "Universal MetricsTools." CrossTalk(February).

ISO/IEC (2002). "ISO/IEC 15939 International Standard Software engineering - Software measurement process."

Kitchenham, B. A., S. L. Pfleeger, et al. (2002). "Preliminary Guidelines for Empirical Research in Software Engineering." IEEE Transactions on Software Engineering28(8): 721-734.

Lavazza, L. (2000). "Providing Automated Support for the GQM Measurement Process." IEEE Software17(3).

Pérez, J. (2005). "Integración Continua utilizando herramientas OpenSource." Agile-spain, from http://javahispano.net/frs/download.php/126/PonenciaIntegracionContinua_AgileSpain.pdf.

Wong, B., B. Boehm, et al. (2004). Second Workshop on Software Quality. 26th international conference on Software engineering, St. Louis, MO, USA, ACM Press

Wong, B., J. Verner, et al. (2005). Third Workshop on Software Quality. 27th international conference on Software engineering, St. Louis, MO, USA, ACM Press