

Estudio comparativo del proceso software y los modelos industriales aplicando agilidad en los procesos

Alicia Mon, Marcelo Estayno

Universidad Nacional de La Matanza,
Departamento de Ingeniería e Investigaciones Tecnológicas
Florencio Varela 1903 – San Justo – Código Postal 1754,
Provincia de Buenos Aires, Argentina
aliciamon@sion.com
mestayno@fibertel.com.ar

RESUMEN

En el presente artículo se describen los ejes centrales de una investigación orientada al desarrollo de un Modelo de Proceso Software que incorpore las transformaciones y mejoras organizacionales introducidas en los diferentes modelos de desarrollo industrial, especialmente las innovaciones del Modelo Japonés, que aparecen con poca sistematización en novedosos modelos de proceso, a fin de proponer posibles mejoras en la definición del Proceso.

El desarrollo conceptual del Proceso Software no es ajeno a la evolución de las formas de organización del trabajo en la producción industrial en general. El surgimiento de Metodologías Ágiles en el desarrollo de software da cuenta de ello, al introducir formas de organización del trabajo provenientes del modelo japonés, que se encuentran inmersas en los ejes organizacionales de la producción industrial en la actualidad.

La investigación en curso que aquí se expone brevemente, se propone analizar comparativamente los criterios específicos de organización del trabajo que ordenan el Proceso Software, los diferentes modelos industriales y las innovaciones que introducen los nuevos métodos ágiles, que incorporan conceptos de desarrollo modular, flexibilidad en los procesos, importancia en los conocimientos de la gente o programación por pares, provenientes del modelo japonés de organización del proceso productivo, tendencias todas a quebrar las fronteras en los compartimentos estancos de los equipos de desarrollo.

1. Introducción

El área de estudio sobre proceso software se reconoce actualmente como un factor crucial en el desarrollo de software, ya que su propósito es proveer el conjunto de actividades que permiten, a partir de la expresión de una necesidad, derivar un sistema software que la satisfaga. La importancia del Proceso en el contexto de la ingeniería del software está respaldada por el gran número de trabajos de investigación que se han realizado hasta la fecha. Los resultados de dichos trabajos de investigación han sido numerosos estándares de proceso [7]; [8]; [9].

En la producción industrial, los diferentes modelos de organización se han ido desarrollando en la medida que la ciencia y la tecnología ha ido evolucionando, aunque implícitamente, se han sucedido cambios en las formas organizacionales que no responden exclusivamente a un tipo de desarrollo tecnológico, sino a unas formas particulares de organización del trabajo en el ámbito de las fábricas y las empresas.

El estudio descriptivo-comparativo de las diversas formas que ha ido adoptando la organización del trabajo en el desarrollo industrial, nos permite encontrar que algunas de las formas del desarrollo de la producción industrial son adoptadas por las formas de organización en el proceso software, y que las mejoras incorporadas a ciertas limitaciones en dichos modelos han sido encontradas e implementadas en diferentes áreas, permitiendo la mejora del proceso de desarrollo de la producción en general.

La definición de la organización del proceso de producción comienza, inicialmente, con las formas de producción artesanal, que se valía de las capacidades individuales para desarrollar un producto. A medida que las formas han ido mejorando, la organización industrial de principio del siglo XX incorpora las primeras técnicas de organización científica del trabajo y medición de tiempos en el taller, seguido por el modelo fordista de producción industrial en serie que introduce la cadena de montaje en la fábrica con tiempos en el desarrollo marcados por la organización que impone la máquina y la producción de grandes cantidades de mercancías, llegando como último modelo al toyotismo o modelo japonés que incorpora la idea de Proceso en la organización del desarrollo productivo y los conceptos de organización por módulos, trabajadores pluriespecializados y polifuncionales, y las técnicas del Total Quality Management.

Si bien, el proceso de desarrollo software contiene formas particulares de organización del trabajo dados por la especificidad del producto que desarrolla, no está exento de los criterios organizacionales de los diferentes modelos y estándares de desarrollo industrial.

En este documento se presenta una breve reseña de los Modelos de organización en la producción industrial desarrollados a través del tiempo, para realizar un análisis de aproximación acerca de cuales de los elementos contenidos en la organización industrial se conjugan en las formas de organización del proceso software, considerando los aspectos relevantes de cada modelo y que aspectos novedosos incorporan los Métodos Ágiles.

Para poder evaluar las limitaciones y posibilidades de la organización en el proceso software en el estado actual, se analiza en que medida se representan los elementos de cada modelo industrial en los modelos y estándares del proceso utilizados como parámetros de desarrollo software.

No se pretende realizar una evaluación exhaustiva de los modelos y estándares, ni dar por probada la vinculación existente entre cada uno de los modelos de organización industrial y los modelos de desarrollo de proceso software, sino brindar algunas líneas de discusión acerca del conjunto de elementos que están en juego en el ordenamiento del proceso software que ayuden a comprender alguna alternativas existentes en la organización industrial que podrían aportar mejoras en la definición del proceso caracterizado.

2. Modelos de Organización Industrial

En la historia de la fabricación de mercancía la producción artesanal es la primera forma de transformación de productos, que a partir de unas materias básica se elaboran productos que pueden ser vendibles en el mercado e intercambiadas, no ya por otro producto, sino por dinero como valor de cambio.

Con el comienzo de la producción industrial emerge un primer modelo de organización del trabajo hacia finales del siglo XIX, conocido como “Taylorismo”, que responde a la lógica de organización de la industria incipiente, en oposición a la producción artesanal, ejerciendo una primer separación técnica del trabajo al interior de la fábrica y en las pautas sociales de consumo de productos [2].

A partir de este modelo y como transformación del mismo se desarrolló el modelo de organización implementado por H.Ford en su fábrica automotriz, y se conoció como “Modelo Fordista” o “Fordismo”. Este modelo de organización industrial, surgió hacia los años 1920/1930, y experimentó su mayor apogeo en la etapa posterior a la segunda guerra mundial, con la producción en serie de productos estandarizados y la organización del trabajo en torno a la “cadena de montaje” o “cinta transportadora” [2].

Finalmente, el modelo introducido por la fabrica Toyota, que comienza a perfilarse en Japón hacia los años 1950/1960 se ha conocido como “Toyotismo”. Este modelo incorpora en el mercado una nueva forma de organización del trabajo al interior de la fábrica y se expande como modelo de organización que se opone a los principales postulados de los modelos anteriores de producción en serie, en las relaciones entre empresas proveedoras, así como nuevas pautas de consumo, adaptadas a las nuevas necesidades de producción diferenciada, la producción por módulos, la noción de calidad diferenciada de los diversos productos, y la inclusión de conceptos y esquemas novedosos que bajan sustancialmente los tiempos y costes de producción.

En el estado actual de la producción industrial, marcado por las constantes innovaciones tecnológicas para la fabricación de productos, conviven diversos modelos de organización que responden en forma genérica o específica a estos modelos mencionados. Las innovaciones a nivel tecnológico no están sujetas directamente a las innovaciones en las formas de organización del trabajo, puesto que esa lógica responde a las áreas de conocimiento desarrolladas en cada sector de

producción, cada rama de desarrollo industrial, así como a las condiciones particulares de cada país o región que se estudie.

El desarrollo de productos software, como emergente del avance científico, responde necesariamente a variables determinadas por las formas de organización del trabajo, desarrolladas para la organización de la producción en general. Es por ello que, en el siguiente apartado expondremos brevemente las características generales de cada uno de los modelos de organización del trabajo en la producción, tales como el Modelo Artesanal, el Modelo Taylorista, el Modelo Fordista y el Modelo Toyotista o Japonés, para analizar cuales de los elementos en ellos desarrollados se aplican y se combinan en la forma de organización del Proceso Software y realizar un análisis comparativo con las innovaciones planteadas por los agilistas.

3. Características específicas de los Modelos

Todos los Modelos de organización de la producción comprenden una serie de características genéricas a partir de las cuales organizan el proceso de desarrollo de sus productos. Para realizar un trabajo comparativo entre ellos, se han definido una serie de criterios de análisis que se reproducen en cada Modelo industrial y en el proceso software, adoptando diferentes formas de acuerdo a las particularidades específicas en las que han sido desarrolladas. A continuación se exponen muy brevemente los criterios definidos a fin de establecer una comparación entre las diversas formas de aplicarlos.

3.1. Tipo de Organización del trabajo

Este concepto permite analizar las formas de organización general del trabajo al interior de cada industria, empresa o equipo de trabajo.

3.2. Métodos de control de producción

Los métodos de control de la producción aplicados por cada Modelo permiten analizar las diferencias respecto al control de tiempos de cada actividad en el desarrollo de un producto.

3.3. División técnica del trabajo

El análisis de la división técnica hace referencia a las formas de distribución del trabajo técnico de cada una de las actividades productivas de la organización en su conjunto.

3.4. Conocimientos

El concepto de Conocimientos de los individuos involucrados en el proceso de trabajo, permite analizar de que manera cada Modelo en su organización concibe la transmisión, división y aplicación de los conocimientos necesarios para la producción al interior de la misma.

3.5. Planificación de la Producción

La noción de planificación es un concepto que permite analizar la estrategia de planeamiento en la distribución de las actividades específicas de producción o desarrollo utilizadas por los diferentes Modelos.

3.6. Calidad

Los conceptos de Calidad son introducidos en el análisis como expresión de la forma de ordenamiento de la organización en cada uno de los Modelos.

Para poder avanzar en el análisis comparativo entre los distintos modelos, el siguiente cuadro expone una asignación de los seis criterios definidos para cada uno de los cuatro Modelos de organización de la producción, describiendo las formas que adoptan en cada uno de ellos: Artesanal, Taylorismo, Fordismo y Toyotismo.

Cuadro 1. Características comparativas de los Modelos de Organización

	Artesanado	Taylorismo	Fordismo	Toyotismo
1.Tipo de Organización del Trabajo	Por oficio	Organización científica del trabajo	Cinta transportadora o cadena de montaje	Modulos flexibles y transfuncionales
2.Métodos de Control de Producción	Sin control sistemático del tiempo de producción	Asignación de tiempos de producción <i>Tiempos asignados</i>	Imposición de tiempos de producción <i>Tiempos impuestos</i>	Revisión constante de tiempos y tareas <i>Tiempos Compartidos</i>
3.División Técnica del Trabajo	Trabajo de desarrollo completo de producción	Fraccionamiento de tareas Repetición de gestos parcelarios	Parcelación Máximo Fraccionamiento de tareas Especialización	Polifunción Desespecialización con Pluriespecialización
4.Conocimientos	Dominan los conocimientos técnicos individuales	Reducción de conocimientos técnicos individuales	Especialización de conocimientos fraccionados Separación entre concepción y ejecución del trabajo	Ampliación de conocimientos, Pluriespecialización
5.Planificación de la Producción	Sin Planificación	Por tareas	Por Puestos	Por módulos polifuncionales
6. Calidad	Basada en el conocimiento y capacidad individual	Basada en la rapidez de producción	Basada en la rapidez y cantidad de producción	Basada en la organización del Proceso

4. Caracterización del Proceso Software

El software, como producto de desarrollo posee un conjunto de características que lo diferencian fuertemente del resto de los productos industriales. Esta diferenciación está marcada por lo intrínseco del producto, en tanto que es un producto lógico, abstracto, no físico; como producto se desarrolla, no se fabrica; la complejidad en su definición está dada por la volatilidad de los requisitos y la incorporación de nuevas funcionalidades a partir de lo desarrollado [10].

A pesar de estas particulares diferencias del producto software frente a la mayoría de los productos industriales, pareciera que el ordenamiento del proceso de desarrollo software responde a cierta combinación de las características mencionadas,

conjugando elementos diversos que responden en forma no lineal a ciertos aspectos organizacionales de los modelos de desarrollo industrial expuestos.

En este sentido, el criterio *1. Tipo de Organización del Trabajo* en el proceso software pareciera responder de alguna manera y en forma combinada al modelo taylorista de Organización Científica del Trabajo con la medición de tiempos de desarrollo y al modelo Toyotista en cuanto a la organización modular del desarrollo, regida, sin embargo por la predominancia de los tiempos de desarrollo y las capacidades individuales.

Respecto a los *2. Métodos de Control de producción* que se aplican en los diferentes modelos de desarrollo software, podría encontrarse que se utiliza un análisis de los tiempos asignados a cada tarea de desarrollo, y a partir de esa información se asignan los tiempos para cada actividad, tomando esta concepción del modelo taylorista de producción.

En cuanto a la *3.División técnica del trabajo*, el fraccionamiento de tareas y la especialización aplicadas en el modelo fordista parecieran ser los vectores que rigen la organización del proceso software. En tanto que los *4.Conocimientos* de la gente que integra los equipos de desarrollo parecerían quedar parcelarios con la separación existente entre concepción y ejecución del trabajo, marcado por la diferencia entre la dirección del proyecto, el analista y el programador en sus diferentes funciones, de la misma manera que se representa con la especialización de conocimientos fraccionados aplicados por el modelo Fordista.

En el desarrollo software la *5. Planificación de la producción* se realiza por tareas, tal como lo ha definido el modelo Taylorista.

Como último de los criterios analizados, los conceptos de *6.Calidad* que se utilizan en el desarrollo software, si bien incorporan elementos novedosos del modelo Toyotista en cuanto a la concepción de la “Calidad del proceso”, pareciera no responder tan claramente a ninguno de los modelos descriptos, aunque combina elementos de las diversas concepciones organizacionales. El Modelo Toyotista incorpora las técnicas de Total Quality Management basadas fundamentalmente en la detención del proceso productivo en cada módulo ante la presencia de fallos, elementos que no parecen ser incorporados directamente en el desarrollo de productos software, sino únicamente como conceptos, sin técnica directa, a partir de la medición de la madurez del proceso de desarrollo de la organización

A continuación se exponen los criterios definidos para los modelos industriales y las características que adquieren en el Proceso Software:

Tipo de organización del trabajo

Organización científica del trabajo

Métodos de control de producción

Asignación de tiempos de producción. Tiempos asignados y Tiempos impuestos

División técnica del trabajo

Parcelación. Máximo Fraccionamiento de tareas. Especialización.

Conocimientos

Especialización de conocimientos fraccionados. Separación entre concepción y ejecución del trabajo

Planificación de la Producción

Por tareas

Conceptos de Calidad

Combina elementos de los diversos modelos. No incorpora técnicas concretas de Calidad Total en el propio proceso de producción.

Frente a estas breves ideas expuestas, que se encuentran aún en plena investigación, podría pensarse que el proceso de desarrollo software, si bien requiere de un modelo propio de organización, no está exento de los modelos de organizativos que rigen para la producción en general y que debería tener en cuenta algunas mejoras sobre los criterios descriptos que han sido incorporadas por los diferentes modelos de la producción industrial.

Sin embargo, la especificidad del proceso software requiere que se diferencie de estos, incorporando innovaciones que provengan de la experiencia y la experimentación, contemplando las particularidades en la organización que genera la especificidad del producto que desarrolla.

5. Variante de Métodos Ágiles en el Proceso Software

Entre las diversas posibilidades de desarrollo software, existe en la actualidad un extenso abanico de tipos de procesos o modelos de desarrollo que plantean sus especificidades en cuanto a las formas de organización del proceso. Entre todos los modelos disponibles el más convencional es el modelo en cascada o lineal-secuencial, que convive con el modelo "V", el modelo de construcción de prototipos, el modelo de desarrollo rápido o RAD, el modelo incremental, el modelo en espiral básico, el espiral win-win, el modelo de desarrollo concurrente y un conjunto de modelos iterativos o evolutivos. No obstante, a finales de la década de 1990 los Métodos Ágiles irrumpen en las prácticas de la ingeniería de software, constituyendo un movimiento que se opone a estos modelos ya consagrados.

El surgimiento de los Métodos Ágiles se promueve como respuesta a la prolongada crisis del software y la responsabilidad que en ellas tienen las metodologías y los estándares tradicionales generadas por organismos internacionales y aplicadas en diversos ámbitos de desarrollo, tales como CMM, Spice, BootStrap o derivaciones de ISO9000 [15] [16], que, de acuerdo a la visión de los agilistas no han logrado resolver el problema crítico que se ha planteado hasta ahora.

Dentro de la múltiple oferta de métodos, metodologías de evaluación o estimación de conformidad, estándares para metodologías o meta-modelos existentes en la actualidad, que se ordenan de acuerdo a los modelos existentes, podemos establecer que, aunque difieren bastante en su conformación y naturaleza, la totalidad de ellos coinciden en la necesidad de la planificación y el establecimiento de normas para lograr un desarrollo acorde a las necesidades planteadas.

Todos los métodos y metodologías aplicadas hasta ahora parten de la elicitación y el análisis completo de los requerimientos del usuario, a partir del cual se puede establecer una serie de características, y definir los requisitos funcionales y no funcionales. La información ordenada, necesariamente, debe ser documentada como especificaciones para la siguiente etapa de diseño, en el que se genera la arquitectura del sistema, para derivar luego en la programación, la etapa de pruebas y la puesta en marcha del sistema.

Como oposición a este ordenamiento organizacional del desarrollo, los métodos ágiles se plantean estrategias que promueven prácticas adaptativas, en las cuales el desarrollo está centrado en la gente y en los equipos, son iterativas, se orientan a las prestaciones y la entrega del producto, y promueven la comunicación intensiva como medio de transmisión de los conocimientos.

Dentro de los métodos ágiles más conocidos y aplicados se encuentran Extreme Programming, Scrum, Crystal Methods, RUP, Dynamic Systems Development Method, Evo, Adaptive Software Development, Feature Driven Development y Agile Modeling. En general y hasta el momento, carecen de especificación formal y se expresan a través de manifiestos declarativos, lo que genera fuertes críticas y posiciones de rechazo desde los defensores de los modelos más tradicionales.

Si bien, sus definiciones están lejos de la sistematización del desarrollo, comparten un modelo organizativo incremental, basado en pequeñas entregas con ciclos rápidos, el trabajo debe ser cooperativo donde los desarrolladores y los usuarios trabajan juntos en estrecha comunicación, el método es simple, fácil de aprender y adaptativo en la medida que resulte capaz de incorporar los constantes cambios.

El eje que ordena la organización de los métodos ágiles es la velocidad y la simplicidad del desarrollo. De acuerdo con ello, los equipos de trabajo se concentran en obtener lo antes posible una pieza útil que implemente sólo lo que sea más urgente, de inmediato requieren feedback de lo que han desarrollado y a partir de allí siguen avanzando con ciclos igualmente cortos, desarrollando de manera incremental.

El Manifiesto Ágil de Desarrollo de Software se presenta como una alternativa a los métodos tradicionales basados en una gestión burocrática y una documentación exhaustiva y se expresa con los siguientes postulados [11]:

- Estamos poniendo al descubierto formas mejores de desarrollo de software, haciéndolo y ayudando a otros a que lo hagan. A través de este trabajo hemos llegado a valorar:
- Los individuos y la interacción por encima de los procesos y herramientas.
- El software que funciona por encima de la documentación abarcadora.
- La colaboración con el cliente por encima de la negociación contractual.
- La respuesta al cambio por encima del seguimiento de un plan.

Aunque hay valor en los elementos a la derecha, valorizan más los de la izquierda.

La definición de estos postulados agilistas se ha especificado en los siguientes principios [12]:

1. Nuestra prioridad más alta es satisfacer al cliente a través de la entrega temprana y continua de software valioso.
2. Los requisitos cambiantes son bienvenidos, incluso cuando llegan tarde en el desarrollo. Los procesos ágiles asumen el cambio como una ventaja competitiva para el cliente.
3. Entregar con frecuencia software que funcione, desde un par de semanas hasta un par de meses, con preferencia por las escalas de tiempo más breves.
4. La gente de negocios y los desarrolladores deben trabajar juntos cotidianamente a través de todo el proyecto.
5. Construir proyectos en torno de individuos motivados. Darles la oportunidad y el respaldo que necesitan y procurarles confianza para que realicen la tarea.
6. La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
7. El software que funciona es la medida primaria de progreso.
8. Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante indefinidamente.
9. La atención continua a la excelencia técnica enaltece la agilidad.
10. La simplicidad (el arte de maximizar la cantidad de trabajo que no se hace) es esencial.
11. Las mejores arquitecturas, requerimientos y diseños emergen de equipos que se auto-organizan.
12. A intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo, y ajusta su conducta en consecuencia.

Muchos de estos elementos indicados como postulados o principios de los Métodos Ágiles parecerían estar inspirados, sistemáticamente o no, en las mejoras introducidas por el modelo industrial japonés o Toyotista, aplicado en la producción automotriz primero y en la manufactura en general después.

El modelo Toyotista introduce una serie de innovaciones en las formas organizacionales de la industria basadas en la eliminación de tiempos muertos, la desaparición del stock de mercancías con la incorporación del concepto de Just in Time que indica la producción de lo pedido, exclusivamente. Además, desarrolla las técnicas innovadoras del Total Quality Management basadas en la detección del proceso de producción ante la presencia de fallos y a través de la mejora continua, haciendo que el producto fluya a instancias del cliente para hacerlo lo más perfecto posible.

Otros elementos esenciales del modelo lo constituye la pluriespecialización de los trabajadores y la relación participativa con el empleado en la toma de decisiones, y en el conocimiento de todas las instancias de la producción, así como la especificación de

principios y métodos iterativos, auto-organizativos e interdependientes en un patrón de ciclos de corta duración

A la luz de algunos de los elementos que, tendencialmente confluirían entre los postulados agilistas y el modelo japonés, podríamos establecer cierta línea de continuidad en la transformación constante de los elementos organizativos del proceso software, de manera que permita organizar y sistematizar algunas de las innovaciones incorporadas por los métodos ágiles y desechar aquellas que solo constituyan postulados declarativos.

6. Conclusiones preliminares

De acuerdo a la investigación en curso, el proceso software podría resultar mas eficiente si incorporara mejoras provenientes especialmente del toyotismo, tales como módulos flexibles, tiempos compartidos en el control de la producción basados en el trabajo polifuncional desespecializado de los desarrolladores, conocimientos pluriespecializados, planificación estimada por módulos y calidad basada en la detención del proceso de desarrollo ante la presencia de errores.

Esta tendencia a incorporar tales características, puede vislumbrarse en algunas metodologías ágiles, con postulados de desarrollo modular, flexibilidad en los procesos, la importancia en los conocimientos de los integrantes del equipo de desarrollo o la programación por pares, tendencias todas a minimizar las fronteras entre los diferentes sectores que componen un equipo de desarrollo, en la distribución de tareas y en la transmisión de los conocimientos del equipo de trabajo.

Más allá de la búsqueda de coincidencias entre los diferentes modelos de organización del desarrollo en la industria y los modelos clásico y ágiles de desarrollo software queda aún por realizar un trabajo comparativo que pueda analizar sistemáticamente cuáles son los elementos que definen la complementariedad y cuáles son los elementos que confluyen en uno u otro Modelo de organización industrial.

La intención de este trabajo ha sido introducir una primera aproximación sobre la naturaleza del proceso software, proponiendo que el desarrollo software no es ajeno a las formas organizativas que se han ido sucediendo en la industria en general, lo que permitiría pensar la adecuación de mejoras introducidas en los modelos de desarrollo industrial al modelo de proceso software, de manera tal que tome en cuenta los cambios organizacionales acaecidos hasta la actualidad.

No obstante, lo que parecería ser una tendencia irrevocable, debe estar acompañado de construcción teórica del modelo y experimentación del mismo para dar respuesta a una industria que se encuentra entre las más innovadoras, pero inmersa en el conjunto de formas organizativas de la producción en general.

Referencias

- [1]. Coriat, B. :*El Taller y el Robot*. Siglo XXI, 1992.
- [2]. Coriat, B.: *Pensar al Revés: Trabajo y Organización en la Empresa Japonesa*. Siglo XXI, 1992.

- [3]. Finkelstein, A.; Kramer, J.; Nuseibeh, B.: *Software Process Modelling and Technology*. Research Studies Press, 1994.
- [4]. Fruin. M.: *Las Fábricas del Conocimiento, la Administración del Capital Intelectual en Toshiba*. Ed. Oxford, 2000.
- [5]. Fuggetta, A.: "Software process: A roadmap". In *The Future of Software Engineering*, Ed. A. Finkelstein. pp 27-34. ACM Press, 2000.
- [6]. Hinley, D.S. : "Software evolution management: A process-oriented perspective". *Information and Software Technology* 38, 11 pp. 723-730. November 1996.
- [7]. *IEEE Standard for Developing Software Life Cycle Processes, IEEE Standard 1074-1991*. IEEE, 1991.
- [8]. ISO/IEC. *International Standard: Information Technology. Software Life Cycle Processes, ISO/IEC Standard 12207-1995*. ISO/IEC, 1995.
- [9]. ISO/IEC. *ISO/IEC TR 15504. Information Technology – Software process assessment*. International Organization for Standardization, International Electrotechnical Commission, 1998. <http://www.sel.iit.nrc.ca/spice>.
- [10]. Jacobson, I.; Booch, G.; Rumbaugh, J.: *The unified Software Development Process*, Addison Wesley 1999.
- [11]. Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland y Dave Thomas. "Agile Manifesto". <http://agilemanifesto.org/>, 2001.
- [12]. Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland y Dave Thomas. "Principles behind the Agile Manifesto". <http://agilemanifesto.org/principles.html>, 2001.
- [13]. Kent Beck. "Interview with Kent Beck and Martin Fowler". Addison-Wesley, 2001.
- [14]. Kent Beck y Martin Fowler. *Planning Extreme Programming*. Reading, Addison Wesley, 2000.
- [15]. Karl Wiegers. "Read my lips: No new models!". *IEEE Software*. 1998.
- [16]. Karl Wiegers. "Molding the CMM to your organization". *Software Development*, 1998.
- [17]. ISO/IEC. *International Standard: Information Technology. Software engineering -- Product quality -- Part 1: Quality model, ISO/IEC Standard 9126-1:2001*, ISO/IEC, 2001.
- [18]. ISO/IEC. *International Standard: Information Technology. Software engineering -- Product quality – Part 2: External metrics, ISO/IEC Standard 9126-2:2003*, ISO/IEC, 2003.
- [19]. ISO/IEC. *International Standard: Information Technology. Software engineering -- Product quality – Part 3: External metrics, ISO/IEC Standard 9126-3:2004*, ISO/IEC, 2004.
- [20]. ISO/IEC. *International Standard: Information Technology. Software engineering -- Product quality – Part 4: Quality in use metrics, ISO/IEC Standard 9126-4:2004*, ISO/IEC, 2004.