

CALIDAD TOTAL EN UN MODELO INTEGRADO DE PROCESO SOFTWARE Y CICLO DE VIDA

Alicia Mon¹ y Javier Garzás²

¹Grupo GIS-UNLaM

Universidad Nacional de La Matanza, Argentina

aliciamon@fibertel.com.ar

²Kybele Research, Dpto. de Lenguajes II;

Universidad Rey Juan Carlos; Madrid, España.

Javier.Garzas@urjc.es

CONTEXTO

La presente línea de investigación se desarrolla en el contexto de I/D del Grupo de Ingeniería de Software GIS de la Universidad Nacional de La Matanza que trabaja en proyectos sobre proceso software y modelos de calidad para la industria del software. Asimismo, la investigación forma parte de una tesis doctoral que se está desarrollando en el Depto. de Lenguajes II de la Universidad Rey Juan Carlos de Madrid, participante de numerosos proyectos de I/D sobre Fábricas de Software y mejora de procesos.

RESUMEN

La modelización del Proceso Software constituye un marco de referencia para la organización de las actividades que involucran todas las etapas del desarrollo. La representación del ciclo de vida define los estados por los que pasa un producto software y la representación del proceso software define el conjunto de actividades esenciales no ordenadas en el tiempo que requiere el desarrollo de software.

En el presente artículo se exponen los avances de una investigación que propone una modelización integrada del Proceso Software y del Ciclo de Vida del producto en la que se diferencian actividades y productos con incorporación de mejoras inspiradas en los Modelos de Proceso industriales, tales como la Calidad Total.

Palabras Clave: Proceso Software, Modelos de Proceso, Ciclos de Vida Software

1. INTRODUCCION

La modelización de los procesos para la industria en general, requiere de la Ingeniería de Procesos como un factor clave para brindar niveles de calidad predecibles y escalables, basándose en la clara definición de las actividades de los procesos. La forma de aumentar la eficiencia en este sentido, consiste en centrar la producción en procesos y mejorar la capacidad de éstos.

El diseño, la medición y la mejora de los procesos en ciclos se han revelado como la clave para mejorar de forma continua la eficiencia y la calidad productiva. En esta irrupción de la relevancia del proceso, algunos modelos de proceso industrial configuran una ecuación en la que, las personas ayudadas por la tecnología actúan como recursos para ejecutarlos.

En los aspectos organizativos centrales del proceso en la industria del software, la materia prima que posibilita la construcción del producto software es el conocimiento, constituido en el único elemento capaz de generar valor, a través del capital intelectual y la capacidad de transformarlo en producto.

La modelización del proceso software debe considerar los productos del software que va construyendo, tanto como la relación entre los

procesos específicos y los entornos socio-organizacionales en los cuales se desarrolla que se determinan por esa capacidad intelectual generada por el conocimiento.

Los diferentes estándares, Modelos de Proceso [IEEE,1997]; [ISO/IEC,2002] o Modelos de Madurez [ISO,1998]; [CMMI,2006][CMMI,2008] y Modelos de Ciclos de Vida [Boehm, 1988]; [Boehm,1994]; [Alexander,1991]; [Jacobson, 1999] comparten la concepción de que una buena definición del proceso software facilita el aseguramiento de que cada elemento de trabajo se asigna y se desarrolla adecuadamente, lo que indica que la calidad del proceso determina la calidad del producto.

Un análisis global de estos marcos de referencia determina la falta de consenso en la terminología empleada para caracterizar los enfoques de modelización del proceso software y de los modelos de ciclos de vida. En este campo se trabaja esta perspectiva diferencial entre el producto y el proceso, no obstante, en la definición y descripción de la mayoría de los modelos de ciclos de vida propuestos en la actualidad, hacen referencia al ciclo de vida del producto software a través del conjunto de actividades del proceso software. Por lo tanto, esta separación entre producto y proceso no queda claramente diferenciada en el desarrollo de software.

El desarrollo conceptual del proceso software no es ajeno a la evolución de las formas de organización en la producción industrial en general. El surgimiento de Metodologías Ágiles [Beck, 2001]; [Fowler,2001]; [Larman,2004] en el desarrollo de software da cuenta de ello, al introducir formas de organización provenientes de la mejora de los modelos de proceso industriales en la actualidad [Coriat,1992a]; [Coriat,1992b], tales como el capital intelectual, la importancia de los conocimientos el desarrollo modular, la flexibilidad en los procesos, o la programación por pares [Beck, 1999], todas ellas tendentes a quebrar las fronteras en los compartimentos estancos de los equipos de desarrollo.

En la producción industrial, los modelos de proceso se han desarrollado en la medida que la ciencia y la tecnología han ido evolucionando. Los cambios sucedidos en los modelos de proceso industriales no responden exclusivamente a los avances tecnológicos, sino a las formas de organización del proceso de trabajo en el ámbito de las fábricas y las empresas.

Si bien el producto software no puede ser comparado con los productos ingenieriles o industriales clásicos, es posible establecer similitudes entre el proceso de desarrollo o de construcción del software, en términos de organización, y los elementos de los modelos de proceso industriales.

Las innovaciones provenientes de los procesos industriales, podrían incorporarse al desarrollo de software con el fin de obtener mejoras semejantes a las ocurridas con su introducción en la producción industrial.

2. LINEAS DE INVESTIGACION Y DESARROLLO

En la presente investigación, se propone un Modelo de Proceso Software, que representa un enfoque dirigido por el desarrollo del producto en cuanto al ciclo de vida del software y por el proceso software en cuanto al conjunto de actividades sistematizadas.

El modelo propuesto diferencia conceptualmente el proceso software con una definición clara de actividades y el ciclo de vida de producto, incorporando ambos conceptos en un único modelo. Esta diferenciación, permite separar las actividades del proceso, de las características técnicas de los productos que construye, y comparar el proceso software con los procesos industriales para incorporar al desarrollo de software las mejoras generadas en otros campos de la producción.

El modelo propuesto está centrado en el conocimiento de los recursos humanos que requiere el desarrollo de software, como capital intelectual intrínseco de la producción. Representa

la construcción del producto software en una serie de versiones, conformando cada una de ellas un ciclo de desarrollo indeterminado en el tiempo, en una secuencia no ordenada ni predefinida de actividades y en una secuencia indeterminada de productos construidos, que incorpora los conceptos de calidad total al producto[Coriat,1992b], garantizado por lo particular de la organización de las actividades del proceso basadas en el conocimiento y en la detención del proceso ante la presencia de fallos.

3. RESULTADOS OBTENIDOS

3.1. Características del Modelo propuesto

El Modelo de Proceso Software desarrollado, representa un enfoque dirigido por el desarrollo del ciclo de vida de la construcción del producto software y por el proceso software con el conjunto de actividades sistematizadas, repetibles, no ordenadas en el tiempo como algo diferente del ciclo de vida, es decir que las actividades serán repetibles e independientes del ciclo de vida que adopte un producto software específico.

Dirigido por la gestión para el análisis y estructuración del proceso software, se propone un modelo de desarrollo evolutivo que acompaña la naturaleza iterativa y concurrente de las actividades de construcción del producto en su ciclo de vida software, con los aspectos controlados y sistemáticos del proceso software.

El Modelo propone la construcción del producto software en una serie de versiones, conformando cada una de ellas un ciclo de desarrollo indeterminado en el tiempo, en la secuencia de actividades y en la secuencia de productos construidos. Las actividades del proceso son acciones que realiza un equipo de desarrollo, que están definidas por los estándares de proceso y el ciclo de vida del producto constituye los estados de transformación por los que va pasando el producto software a lo largo de las actividades del proceso software.

Esta diferenciación, permite separar las actividades del proceso, de las características de los productos que construye, y de esta manera poder analizar genéricamente el proceso software comparativamente con los procesos industriales, por una parte, y los productos software, por otra parte.

3.2. Conocimientos del equipo

El conocimiento es considerado como una materia prima inherente al producto que debe desarrollarse. La planificación de la producción debe ser consensuada y coordinada. Debe realizarse en cada puesto o módulo funcional y para construir la planificación global coordinada.

La pluriespecialización de los integrantes de un equipo de trabajo, implica que el amplio conocimiento y la alta cualificación generan una relación participativa en la toma de decisiones, y en el conocimiento de todas las instancias de la producción o desarrollo.

El modelo propone incorporar prácticas que implican un mayor acercamiento en el conjunto de tareas, así como un vínculo más estrecho entre la concepción y la ejecución de tareas, lo que se logra a partir del conocimiento como capital e insumo de la producción.

3.2. Calidad

El Modelo propuesto incorpora los conceptos de calidad total basados en la detención del proceso de producción ante la presencia de fallos, en la que cada instancia del producto, desarrollada en una fase del Modelo, puede pasar a otra fase del desarrollo, cualquiera que sea, solo si tiene una aceptación de que todo lo necesario ha sido completado, y si el producto que la requiere para iniciar su ejecución cuenta con los conocimientos necesarios para comenzar dicha instancia.

Si se detecta un fallo, el producto no pasa de fase y se detiene el proceso hasta que el problema está resuelto. Es condición, la existencia de cero defectos para que una parte del producto pase a la siguiente fase de su desarrollo.

De este modo, cada fase continúa trabajando en forma concurrente en cualquier instancia del producto si es que tiene lo estrictamente necesario para avanzar, provisto por los procesos anteriores, sin ninguna organización temporal y/o secuencial determinada a priori.

La Figura 1 expone una representación gráfica del modelo, en el cual las líneas paralelas representan las actividades del proceso software y las líneas tipo meridianos, representan los estados del producto software o sub productos por los que pasa a través del ciclo de vida, que es finalizado al llegar a cada uno de los polos del círculo.

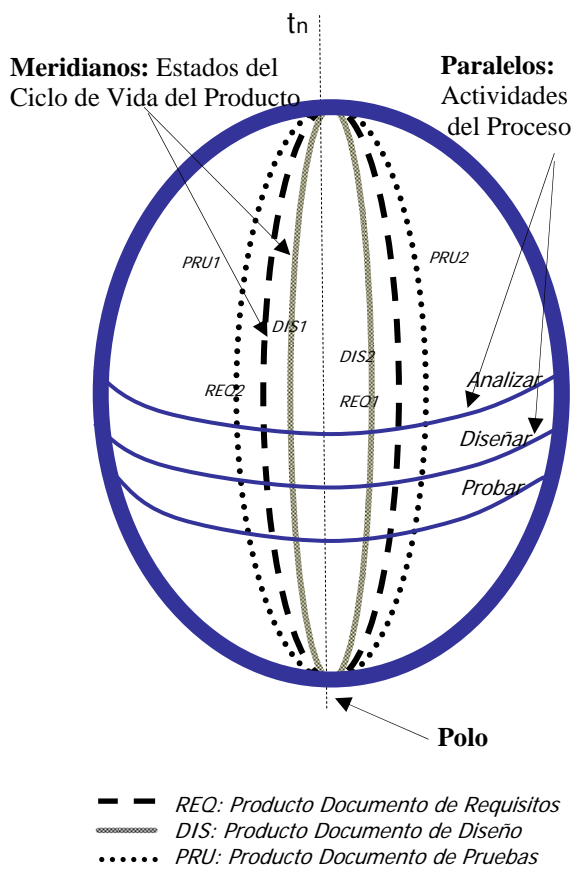


Figura 1. Representación del Modelo

3.4. Iteratividad versus Secuencialidad

En el Modelo desarrollado, la construcción de los productos se inicia a partir de la entrega de un producto en un polo, requiere, como condición esencial el conocimiento aportado por otro producto anterior. Esta entrega de producto

conlleva el conocimiento necesario para que se dispare el inicio de otro producto.

La construcción de los productos es realizada por las actividades del proceso, en tanto que, la transmisión de la condición de inicio entre los productos determina la falta de secuencialidad en las actividades del proceso.

La iteratividad y sucesión no secuencial de actividades es generada por las características de cada producto y por la indeterminación del tiempo de desarrollo de cada una de las actividades del proceso.

Las condiciones para que cada producto sea entregado en un polo, la brindan las diferentes técnicas aplicadas en el proceso para cada una de las actividades que garantizan la calidad en el producto, como por ejemplo, las técnicas de requisitos, las técnicas de validación de diseño, las técnicas de verificación del código, etc.

Las técnicas utilizadas para cada instancia del producto accionan para probar el conocimiento acerca del producto, que puede ser transmitido a los demás productos. En este sentido, cada producto entregable en un polo está conformado por el conocimiento sobre sí mismo.

3.5. Estados del ciclo de vida

Cada producto, en el estado en que se encuentre en el ciclo de vida, se comporta como un dispositivo receptor y transmisor de conocimiento sobre cómo deben seguir los otros productos. De este modo, cada instancia del producto, desarrollado en una fase del ciclo de vida puede pasar a otra fase, cualquiera que sea, solo si tiene una aceptación de que todo lo necesario para esa parte ha sido completado y si el producto que lo requiere para iniciar su ejecución cuenta con los conocimientos necesarios para comenzar dicha instancia.

El desarrollo concurrente y paralelo de diversos productos en una misma fase del ciclo de vida, no se encuentra determinado en el orden de sucesión de instancias, sino que se va determinando de acuerdo a las condiciones para que cada producto,

se construya, se detenga o se entregue en un polo. Cada actividad comienza a desarrollar un producto cuando tiene las condiciones para hacerlo.

Los productos que se construyen no tienen una secuencia definida en el ciclo de vida, sino que la consecución del producto está definida por la indeterminación de lo que cada producto va construyendo y la indeterminación del tiempo para realizar las actividades de su construcción.

4. FORMACION DE RECURSOS HUMANOS

El presente artículo expone las líneas generales de una investigación que se está llevando a cabo en el Departamento de Lenguajes II de la Universidad Rey Juan Carlos de Madrid, España, en el marco de la tesis doctoral “Incorporación de la calidad total de la producción industrial en un modelo integrado de proceso software y ciclo de vida” de una investigadora del Grupo GIS de la Universidad Nacional de La Matanza.

En la Tesis se desarrolla una propuesta de modelo de proceso que integra la ingeniería de proceso con la ingeniería de producto para la industria del software, incorporando de manera definida y sistematizada las mejoras provenientes de los modelos de procesos industriales, tales como la detención del proceso ante la presencia de fallos y la transferencia de conocimiento, como elementos organizativos de la Calidad Total del producto.

5. BIBLIOGRAFIA

[Alexander, 1991] Alexander, L. and Davis, A. “Criteria for selecting software process models”. Proceedings of COMPSAC’91. 521-528. 1991.

[Beck, 1999] Beck, K. Extreme Programming Explained: Embrace Change. (Addison Wesley, 1999).

[Beck, 2001] Beck, K; Beedle, M; Cockburn, A; Cunningham, W; Fowler, M; Agile Manifesto. web site. (2001). <http://agilemanifesto.org>.

[Boehm, 1988] Boehm, B. A Spiral Model of Software Development and Enhancement Computer, pp. 61-78, May 1988.

[Boehm, 1994] Boehm, B. & R. Ross A Collaborative Spiral Software Process Model Based on Theory W. Proceedings, ICSP3. IEEE, Reston, VA, Octubre 1994.

[CMMI, 2006] Capability Maturity Model Integration for Development (CMMI-DEV), Version 1.2. Software Engineering Institute, Carnegie Mellon University. 2006.

[CMMI, 2008] Chrissis, M.B., Konrad, M., Shrum, S. CMMI: Guidelines for Process Integration and Product Improvement (Sei Series in Software Engineering). Amazon, 2008.

[Coriat, 1992a] B. Coriat. El Taller y el Robot. Siglo XXI, 1992.

[Coriat, 1992b] B. Coriat. Pensar al Revés: Trabajo y Organización en la Empresa Japonesa. Siglo XXI, 1992.

[Fowler, 2001] M. Fowler. “Is design dead?” Proceedings XP2000. Web site (2001). <http://www.martinfowler.com/articles/designDead.html>. <http://www.refactoring.com/>

[IEEE, 1997] IEEE. IEEE Standard for Developing Software Life Cycle Processes, IEEE Standard 1074-1997. IEEE, 1997.

[ISO/IEC, 2002] ISO/IEC. International Standard: Information Technology. Software Life Cycle Processes, ISO/IEC Standard 12207-1995/Amd. 1-2002.

[ISO/IEC, 1998] 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>

[Jacobson, 1999] Jacobson, I; Booch, G; Rumbaugh, J. El Proceso Unificado de Desarrollo de Software, Addison Wesley, 1999.

[Larman, 2004] C. Larman. Agile & Iterative Development. A Manager’s Guide. (Addison Wesley, 2004).