

**INDIVIDUAL WORK DEVELOPMENT OF SOFTWARE PROJECTS:  
A REALITY WITHOUT METHOD****EL DESARROLLO INDIVIDUAL DE PROYECTOS DE SOFTWARE:  
UNA REALIDAD SIN MÉTODO****MSc(c). Javier Mogollón Afanador, MSc. Luis Alberto Esteban Villamizar****Universidad de Pamplona**

Grupo de Investigación en Ciencias Computacionales -CICOM.  
Ciudadela Universitaria. Pamplona, Norte de Santander, Colombia.  
Tel.: 57-7-5685303, Fax: 57-7-5685303, Ext. 158  
E-mail: jmogollon.a@gmail.com, lesteban@unipamplona.edu.co

**Abstract:** This paper presents a summary of the study of the evolution of software development processes showing the tendency of team work processes and the need of a well-developed guide to be followed to adapt development processes of software, which were designed to be used as a procedure in software processes team work also to be used as a guide to individual work development process.

**Keywords:** Software engineering, software development process, adaptation of software process

**Resumen:** Este artículo presenta una síntesis del estudio de la evolución de los procesos de desarrollo de software, en el cual se observa la tendencia de procesos de trabajo en grupo, y plantea la necesidad de un procedimiento para la adaptación de los procesos de desarrollo de software, que fueron diseñados para trabajo en grupo, en procesos de desarrollo individual.

**Palabras clave:** Ingeniería del software, proceso de software, adaptación de procesos de software.

**1. INTRODUCCIÓN**

Los proyectos de desarrollo de software de manera individual, es decir abordados por una sola persona, son frecuentes en contextos particulares, donde se requiere automatizar pequeñas tareas, y para lo cual no se cuenta con grandes presupuestos. Esta informalidad de contratación de proyectos de desarrollo de software, exige unos requisitos mínimos de calidad, por lo cual es ideal contar con una metodología de desarrollo de software.

En la actualidad existen innumerables metodologías de desarrollo de software, (Shenome, 2004; Hurtado, 2005; Vijayarathy,

2008; Meda, 2005; Canós, 2004; Ginestà, 2005; Cabrera, 2009) que promueven cada una de ellas sus propias buenas prácticas de desarrollo, pero el común denominador de dichas metodologías es la necesidad de interactuar entre diferentes personas para lograr obtener un producto software y esto implica tener más de un miembro en el equipo de desarrollo.

Como parte de un proyecto de investigación (Mogollón, 2010) en la Maestría en Gestión de Proyectos Informáticos, de la Universidad de Pamplona, se realizó una encuesta para determinar los tipos de proyectos de desarrollo de software que comúnmente son solicitados a desarrolladores

y de la cual se encontró que el 82% de los encuestados han tenido experiencia con el desarrollo individual.

Este artículo plantea la necesidad de adaptar las existentes metodologías<sup>1</sup> a procesos secuenciales y simplificados, que puedan ser desarrollados por una sola persona, en lugar de construir nuevas metodologías para el desarrollo individual.

La primera parte del artículo presenta una síntesis de la evolución de los procesos de desarrollo de software. La segunda parte presenta los resultados de la encuesta realizada, haciendo énfasis en la justificación de un procedimiento de adaptación de metodologías ágiles al desarrollo individual. Finalmente la tercera parte describe las características del proceso de adaptación que al final de la investigación se obtendrá.

## 2. EVOLUCIÓN DEL PROCESO DE DESARROLLO DE SOFTWARE

Actualmente existen diferentes metodologías para el desarrollo de software, las cuales evolucionaron a la par y como respuesta, a los avances tecnológicos a nivel industrial desde la década de los 50 del siglo 20 hasta nuestros días (Ginestà, 2005).

Todas estas metodologías, desde el método en cascada (Ginestà, 2005), hasta la aparición a finales de los noventa de la gran mayoría de los métodos ágiles (Canós, 2004; Meda, 2005), tienen una característica en común, son diseñadas para un contexto de trabajo en equipo, planteando la interacción de diferentes tipos de roles, con funciones diferentes dentro del proyecto.

Esto implica que estas metodologías han dejando de lado el contexto de trabajo de desarrollo individual en el cual todo un proyecto es realizado por una sola persona, al cual en este artículo, se denomina “desarrollo individual”.

Dentro de la evolución de los procesos de desarrollo de software, se encuentra un aporte cercano al desarrollo individual, desarrollado en 1995 por Watts Humphrey (Humphrey, 2000; Cokbaine, 2002), quien propone un conjunto de practicas disciplinadas denominadas PSP<sup>2</sup> (Humphrey, 2000; Cokbaine, 2002; Pelaez, 2003; Teodorovicz, 2002; Zulueta, 2007), con el fin de servir de guía personal para la gestión del tiempo y mejoras en la productividad de ingenieros Juniors de empresas que utilicen como modelos de procesos el CMMI (Pelaez, 2003), donde su producto de trabajo forma parte del conjunto de elementos que necesita la empresa para la producción de artefactos software, siendo el primer circulo de calidad de la empresa, seguido del TSP<sup>3</sup> (Humphrey,2000a; Cokbaine, 2002) que provee el esquema de trabajo de procesos definidos para equipos de gerentes e ingenieros y que toman el trabajo individual del programador y lo integran en un circulo de calidad mayor, ayudando a garantizar un sistema de desarrollo de aplicativos software de mayor calidad y de gran tamaño.

Sin embargo el PSP puede ser utilizado en proyectos donde participen varias personas, cada una con responsabilidades específicas, a las cuales el ingeniero aplica los principios del PSP y que formaran parte en la integración del proyecto final. Esto implica que el desarrollo individual sigue siendo un área inexplorada, pese a los aportes del PSP.

---

1 Estas metodologías están diseñadas típicamente para trabajo en equipo

---

2 PSP del Ingles (Personal Software Process), o Proceso de software personal en Español

3 TSP el Inglés Team Software Process, o Proceso de software en equipo

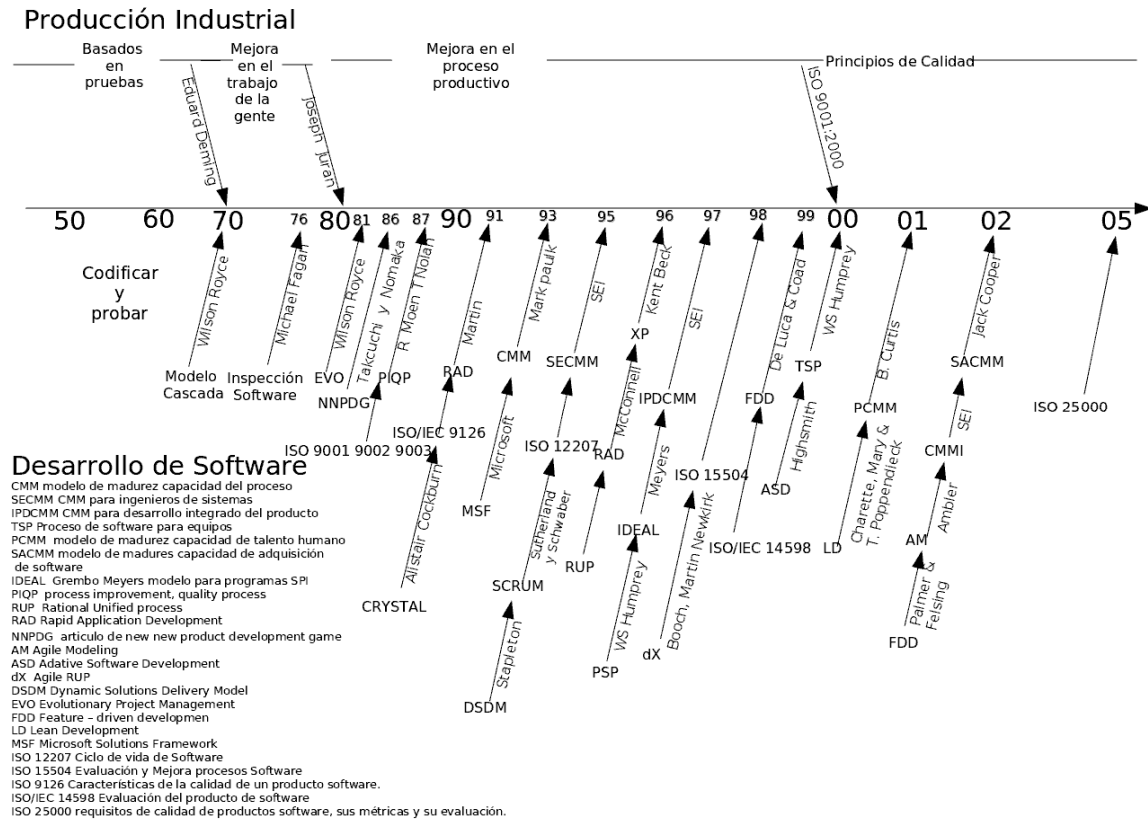


Fig. 1: Evolución de los procesos de desarrollo de software

La figura 1 muestra en forma general, como los momentos históricos a nivel de la producción industrial<sup>4</sup> (Ruiz, 2003), han influido directamente en el desarrollo y evolución de la ingeniería del software, marcando los momentos en los cuales surgen y evolucionan las diversas metodologías, conceptos, herramientas y estándares que hasta nuestros días orientan la forma como se desarrollan los productos software (Meda, 2005; Gilb, 2002; Ginestà, 2005; Cabrera, 2009; Booch, 1998; Shenomne, 2004; Arboleda, 2005; Pressman, 2002). Como punto particular se observa en la figura, como algunas de las metodologías tradicionales aun vigentes de desarrollo de software, así como la gran mayoría de las metodología ágiles (Canós, 2004; Hurtado, 2005), surgen en la década de los 90s, respondiendo directamente a las necesidades del mercado a nivel industrial, buscando la mejora en los procesos de desarrollo, las herramientas, artefactos y los productos software. En esa misma época, las

tecnologías emergentes<sup>5</sup> y la expansión de los sistemas informático a los contextos de las medianas y pequeñas empresas, han brindado la posibilidad de desarrollar productos software cada vez más complejos, en los más diversos campos de aplicación y con requerimientos de usuarios cada vez más exigentes

### 3. EL DESARROLLO INDIVIDUAL

Dentro del desarrollo de la investigación “Adaptación de procesos ágiles de desarrollo de software al desarrollo individual de aplicativos pequeños y de bajo presupuesto” (Mogollón, 2010), como proyecto de grado en la Maestría en Gestión de Proyectos Informáticos, de la Universidad de Pamplona, se planteó como una de las cuatro hipótesis de investigación, “El desarrollo individual de software, es una práctica frecuente en contextos de desarrollo de proyectos pequeños y de bajo presupuesto” y para soportar esta hipótesis se

4 Que a su vez representan un desarrollo en las herramientas y en la tecnología que se utilizan

5 Como la aparición de los sistemas operacionales de manejo en interfase gráfica, el Internet, y la masificación de sistemas distribuidos, entre otras.

realizó una encuesta a programadores de la región<sup>6</sup>. En este contexto el presente artículo, presenta los resultados de dicha encuesta relacionados con esta hipótesis del trabajo de investigación.

El objetivo principal de la encuesta es: recolectar la información relacionada con los procesos que utilizan los desarrolladores de software. A su vez, se recolecta información que permite conocer, los tamaños de los proyectos mayoritariamente desarrollados y las metodologías ágiles más conocidas para la elaboración de diferentes aplicativos software.

La encuesta se publica vía web y contiene un total de 20 preguntas, de las cuales, este artículo analiza las siguientes.

*¿Tiene experiencia en desarrollo de aplicativos software en forma individual?*

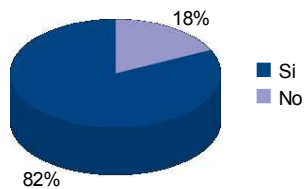


Fig. 2: Experiencia en desarrollo.

Del total de encuestados<sup>7</sup>, un 82% manifiestan que han realizado aplicativos en forma individual y un 18% han realizado trabajos única y exclusivamente en grupos de desarrollo, demostrando este resultado, la pertinencia en la realización de la adaptación metodológica para este tipo de desarrollo.

La pregunta: Los proyectos de desarrollo de software en los que ha participado, *¿mayoritariamente de que tamaño son?*

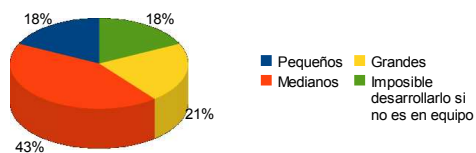


Fig. 3: Tamaños de proyectos

Claramente se observa que se desarrollan en la región mayoritariamente proyectos de tamaño grandes y medianos, pero el porcentaje correspondiente al desarrollo de aplicativos pequeños, es significativamente alto, demostrando así que existen programadores, cuya actividad se centra en el desarrollo de aplicativos de tamaño pequeño y de bajo presupuesto

A la pregunta, *¿Mayoritariamente que tipo de software ha desarrollado usted?*

Para esta pregunta se unificaron los siguientes conceptos:

**Software intensivo en interfase:** Software que se desarrolla con predominancia en diseños de pantallas y presentaciones finales al usuario. Este tipo de software requiere gran cantidad de prototipos de interfaces, definición de reglas para interacción del usuario, amplia investigación en ambientes y usabilidad. Normalmente requieren de herramientas de diseño y desarrollo multimedial, incluyendo actividades como: Edición de audio, edición de vídeo, edición gráfica y creación de imágenes de todo tipo.

**Software intensivo en procesamiento de información:** Producto software que requiere gran cantidad de actividades fundamentalmente para la creación de código que asegure el manejo, aplicación y realización de cálculos intensivos con los datos suministrados al programa.

**Software intensivo en captura, almacenamiento y recuperación de datos:** Desarrollos donde mayoritariamente el código se dedica a interactuar con motores de bases de datos, capturando, procesando, almacenando y recuperando datos de diferentes tipos y respondiendo a diferentes necesidades.

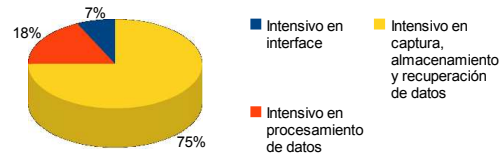


Fig. 4. Tipos de proyectos

Como se observa en los resultados, en el contexto, es muy común el desarrollo de aplicativos intensivos en captura, almacenamiento y recuperación de datos, lo que indica que en la mayoría de los aplicativos desarrollados, el código se dedica a interactuar con motores de bases de

6 La encuesta publicada en línea, entre mayo y noviembre de 2010, en la URL <http://encuesta.neblinux.org>, fue diligenciada por 30 personas de manera anónima, sin embargo, se realizó la invitación a ser diligenciada a programadores de la región de los Santanderes y la frontera, por tal motivo se considera este el contexto regional

7 Datos a la fecha de realización del artículo

datos, capturando, procesando, almacenando y recuperando datos de diferentes tipos y respondiendo a diferentes necesidades.

Es de resaltar que en nuestro medio el promedio mas bajo de aplicativos desarrollados corresponde a tipo de aplicativos intensivos en interfaces gráficas, mostrando esto que los programadores se dedican mas a la generación de código, que a el diseño de interfaces y ambientes, tal vez, por que este campo en la actualidad es desempeñado por diseñadores gráficos o el porcentaje de programadores con estas habilidades es muy bajo. Otra posible explicación es dada por el tipo de clientes que encargan estos desarrollos.

*¿Que tipo de artefactos almacena como historial de un proyecto de desarrollo de software?*

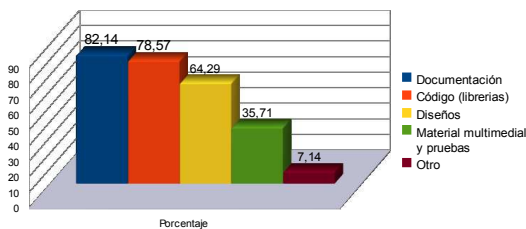


Fig. 5. Artefactos producidos

La pregunta permite la selección múltiple de artefactos generados por parte del encuestado. Los resultados obtenidos, demuestran que los programadores en la actualidad, almacenan de sus desarrollos, todo tipo de artefactos, pero mayoritariamente los más útiles para ellos, como lo son: la documentación, el código y los diseños, indicando esto que aunque existen en los desarrollos la generación de los mismos, no en todos los proyectos se producen en forma ordenada. La adaptación de los procesos de las metodología ágiles permitirá que el programador estandarice la generación de dichos artefactos y el almacenamiento y seguimiento a los mismos, proyecto tras proyecto.

*¿Cual de las metodologías ágiles para el desarrollo del software conoce?*

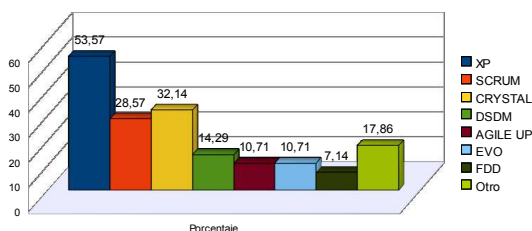


Fig. 6. Metodologías ágiles conocidas

La pregunta permite la selección múltiple de metodologías por parte del encuestado. Los resultados demuestran que en nuestro entorno, se conocen todas las metodologías ágiles, ya sea que estas se utilicen en los desarrollos o no, pero prima el conocimiento sobre la metodología Extreme Programming XP (Meda, 2005) (Hurtado, 2005) (Ginestà, 2005), tal vez por la difusión que se hace en los entornos educativos, o por la gran disposición de información en la Internet. Estos resultados avalan el hecho, que al desarrollar una adaptación a dichas metodologías de desarrollo ágil, podrá ser utilizada por un gran número de personas en la elaboración de aplicativos software.

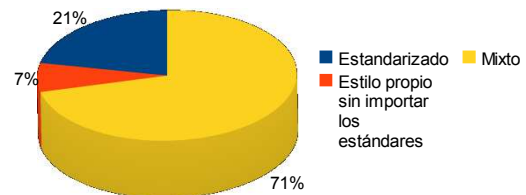


Fig. 7. Estilos de programación

*¿Como define su estilo de programación?*

Los resultados a la pregunta demuestran como cada vez mas los programadores optan por utilizar algún tipo de estándar en su estilo de programación, sea este o no definido por una metodología de desarrollo particular, o por exigencias del proyecto o de la empresa donde se labora, combinándolo con su estilo de desarrollo propio. Esto podría garantizar que al desarrollar un modelo de adaptación de los procesos de las metodologías ágiles, con características de simplicidad y flexibilidad, podrá tener buena receptividad por parte de los programadores para la aplicación de los mismos, en cualquier entorno de trabajo, y en cualquier situación especifica de desarrollo.

#### 4. ADAPTACIÓN DE PROCESOS

La guía del SWEBOK<sup>8</sup> (SWEBOK, 2004), define el área de conocimiento “Proceso de Ingeniería de Software”, en la cual se considera temas que aplican a cualquier parte de la gestión del proceso de ciclo de vida del software, donde se están introduciendo cambios procedimentales o técnicos para la mejora de los procesos.

<sup>8</sup> La guía del SWEBOK dentro de su propósito intenta estructurar el conocimiento comúnmente aceptado sobre Ingeniería del software, dividiendo este conocimiento en 10 áreas.



El desarrollo de software individual no es común para proyectos grandes, pero es muy común en el desarrollo de pequeñas y medianas aplicaciones a la medida.

Para la adaptación de métodos ágiles al desarrollo individual de aplicaciones, se requiere de reglas sencillas que prioricen los principios metodológicos, los roles, actividades del proceso, prácticas y herramientas, con el fin de descartar todas aquellas consideradas poco relevantes de acuerdo a la clasificación de los proyectos software.

Dada la gran cantidad de metodologías existentes para el desarrollo de software, es mucho más práctico buscar un método de adaptación, que generar un método nuevo enfocado, única y exclusivamente al desarrollo individual de aplicativos software.

El trabajo de investigación aun no terminado<sup>9</sup>, pretende generar dicho procedimiento de adaptación de los procesos de trabajo en grupo a procesos de trabajo individual.

## REFERENCIAS

- Arboleda Jiménez, Hugo F. (2005). Modelos de ciclo de vida en desarrollo de software. Revista ACIS. Edición No. 93, Julio - Septiembre.
- Booch, Grady, Martin, Robert C, Newkirk, James. (1998). *The Process. Object Oriented Analysis and Design with Applications*, 2d. ed. Addison Wesley Longman, Inc.
- Cabrera, Armando, Solano, Raquel, Montalván, Mayra. (2009). *Procesos de Ingeniería del Software*. Universidad Técnica Particular de Loja, Ecuador.
- Canós, José H. Letelier, Patricio. Penadés, Ma Carmen. (2004). *Métodologías Ágiles en el Desarrollo de Software*. DSIC - Universidad Politécnica de Valencia. España.
- Cockburn, Alistair. Highsmith, Jim. (2001). *Agile Software Development: The People Factor*. Computer, Innovative technology for computer professionals, IEEE.
- Cockbaine, Juan Carlos, Espinoza Eduardo E. (2002). *Guía del Proceso de Gerencia de Verificación y Validación*. Fabrica de software. Portal de ingeniería del software, Chile. Disponible en internet: [http://www.fabricadesoftware.cl/fabrica\\_documentos.php](http://www.fabricadesoftware.cl/fabrica_documentos.php). Última visita 25 abril de 2010.
- Gacitúa Bustos, Ricardo A. (2003). *Métodos de desarrollo de software: El desafío pendiente de la estandarización*. Universidad del Bío-Bío, Chile.
- Gilb, Tom. (2002). *Evolutionary Project Management (Evo)*.
- Ginestà, Marc Gibert. Hernández Matías, Martín, Peña González, Álvaro, Aycart Pérez, David. (2005). *Ingeniería del software en entornos de SL*. Primera edición. Fundació per a la Universitat Oberta de Catalunya. Barcelona, España.
- Humphrey, W. S. (2000). *The Personal Software Process PSP*. Technical report, CMU/SEI-2000-TR-022. Software Engineering Institute. Carnegie Mellon.
- Humphrey, W. S. (2000). *The Team Software Process TSP*. Technical report, CMU/SEI-2000-TR-023. Software Engineering Institute. Carnegie Mellon.
- Hurtado, Julio Ariel, Bastarrica, Cecilia. (2005). *Hacia una línea de procesos ágiles Agile SpsL*. Proyecto SIMEP-SW. FIET Universidad del Cauca.
- Meda Rodolfo, Ierache Jorge. (2005). *Una Propuesta de Conjunción de Elementos Metodológicos en común dentro de los Enfoques ágiles para el Desarrollo de Software*. Facultad de Informática Ciencias de la Comunicación y Técnicas Especiales. Instituto de Sistemas Inteligentes y Enseñanza experimental de la Robótica. Universidad de Morón. Buenos Aires, Argentina.
- Mogollón Afanador, Javier O. (2010). *Adaptación de procesos ágiles de desarrollo de software al desarrollo individual de aplicativos pequeños y de bajo presupuesto*. Tesis de grado. Maestría Gestión de Proyectos Informáticos, Universidad de Pamplona, Colombia.
- Pelaez Ramírez, Juan José. (2003). *El modelo de capacidad de madurez y su enfoque al proceso personal de software (PSP)*. tesis. Escuela de Ingeniería. Departamento de Ingeniería en Sistemas Computacionales. Universidad de las Américas Puebla, Puebla, México.
- Pressman, Roger S. (2002). *Ingeniería del Software un enfoque práctico*. 5a edición. McGraw Hill interamericana de. España.

<sup>9</sup> A la fecha de presentación para evaluación del presente artículo se encuentra en desarrollo los procedimientos de adaptación que constituyen la tesis de investigación, soportada por cuatro hipótesis, de las cuales este artículo aborda solo una.

- Ruiz N, Joaquín. (2003). Breve análisis de la evolución, innovación y mejores prácticas de los procesos industriales. Boletín IIE, Internet: <http://www.iie.org.mx/>. México julio-septiembre.
- Shenomne Marcelo Hernan. (2004). Diseño de una metodología ágil de desarrollo de software. Tesis de grado en ingeniería informática, Fiuba Facultad de ingeniería Universidad de Buenos Aires, 2004.
- Guide to the Software Engineering Body of Knowledge (2004). Version SWEBOK. IEEE Computer Society (2001-2003). <http://computer.org>, Los Alamitos, California.
- Teodorovicz, José Carlos, Dergint, Dario E. (2002). Proceso Personal de Trabajo. Centro Federal de Educación Tecnológica de Paraná, Brazil.
- Vijayasathay, Leo R. Agile. (2008). Software development: a survey of early adopters. Journal of Information Technology Management. Volume XIX, Number 2.
- Zulueta Véliz, Yeleny. (2007). Introducción de técnicas del Personal Software Process desde los primeros años en la formación del ingeniero informático. Revista Ingeniería Informática, edición 14, mayo.