# Uso de los modelos tradicionales y las metodologías ágiles aplicadas en la industria de software colombiano

Manuel Alejandro Ortega (1)
manuel.ortega00@usc.edu.co

Edgar David Camacho (2)
edgar.camacho00@usc.edu.co

**Universidad Santiago de Cali, Facultad de Ingeniería, Programa de tecnología en sistemas de información 1**
**Universidad Santiago de Cali, Facultad de Ingeniería, Programa de tecnología en sistemas de información 2**

***Resumen***

El mundo de la tecnología informática ha vivido una constante evolución en cuanto a la concepción de sus proyectos. Gracias al "boom" del internet se ha incrementado la demanda de aplicaciones en todos los sectores de la industria. Hoy hablamos de modelos tradicionales y metodologías ágiles como bases de desarrollo aplicados en proyectos, con sus ventajas y desventajas.

Los modelos tradicionales están basados en los primeros paradigmas de desarrollo existentes en el campo tecnológico. Estos optan por el paradigma de un análisis fundamental a priori, que garantiza un desarrollo enriquecido y logran cerrar la brecha de problemas y complicaciones que puedan generarse durante la etapa de desarrollo, disminuyendo finalmente tiempo y recursos. Por otra parte, las metodologías ágiles proponen una forma de trabajo flexible, cuya planificación se actualiza continuamente, siendo su objetivo principal poner en funcionamiento el software lo antes posible. El definir correctamente la metodología a implementar en un proyecto nuevo puede ser el sustento a largo plazo del éxito o fracaso de una aplicación. Cada metodología tiene su orientación, el definir y validar correctamente esto, puede llevar un buen tiempo del total de la planeación previa a la implementación de un proyecto, pero es necesario ya que esto a la larga definirá, incluso, la rentabilidad del mismo.

El presente artículo, de tipo monográfico, aporta un panorama general acerca del uso de algunos modelos y metodologías de desarrollo y su uso a nivel de Colombia, tomando como base algunos de los modelos más representativos de enfoque tradicional como también algunas metodologías ágiles.

*Palabras Clave:* Ingeniería de software, Desarrollo de Software, Modelos tradicionales, Metodologías ágiles, SCRUM.

***Abstract***

The world of computer technology has been constantly evolving in terms of the design of its projects. Thanks to the "boom" of the internet, the demand for applications in all sectors of the industry has increased. Today we talk about traditional models and agile methodologies as development bases applied in projects, with their advantages and disadvantages.

Traditional models are based on the first existing development models in the technological field. They opt for the paradigm of a fundamental a priori, analysis that guarantees enriched development and manage to close the gap of problems and complications that may arise during the development stage, ultimately decreasing time and resources. Moreover, agile methodologies propose a flexible form of work, whose planning is continuously updated, being its main objective to get the software up and running as soon as possible. Correctly defining the methodology to be implemented in a new project can be the long-term livelihood of an application's success or failure. Each methodology has its orientation, correctly defining and validating this can take a good time out of the total pre-implementation planning of a project, but it is necessary because this will eventually even define the profitability of it.

This monographic article provide an overview of the use of some development models and methodologies and their use at the Colombian level, based on some of the most representative models of traditional approach as well as agile methodologies.

*Keywords:* Software engineering, Software Development, Traditional models, Agile methodologies, SCRUM.

## 1. INTRODUCCION

En el presente artículo se propone una revisión de algunos modelos de desarrollo de software tradicional y de

metodologías de desarrollo de software ágiles desde el contexto del uso de estas en la industria colombiana de software.

La industria del software en Colombia ha reportado un crecimiento sostenible durante los últimos años en términos de ingresos, exportaciones y participación de mercado. De acuerdo a la entidad ProColombia (s,f), el sector de la industria del software y servicios en tecnología de información, exportó durante 2015, 208.1 millones de dólares. El servicio que sobresale en la oferta nacional es el desarrollo de aplicaciones empresariales que se adecuan a dominios de negocio específicos. El diseño, desarrollo y mantenimiento de este tipo de aplicaciones empresariales es una labor compleja. Martín Fowler define este tipo de aplicaciones por medio de las siguientes características:

Administran grandes volúmenes de datos persistentes; los datos persistentes son accedidos y manipulados concurrentemente; están compuestas por una gran variedad y cantidad de interfaces de usuario; tienen necesidades de integración con otras aplicaciones empresariales; implementan reglas de negocio complejas; y tienen que ser mantenidas en el tiempo. (Fowler, 2002).

Desde un alto nivel de abstracción, actualmente dividimos los modelos de desarrollo de aplicaciones empresariales en dos grandes categorías: modelos tradicionales y metodologías ágiles. Los modelos de desarrollo tradicional como Cascada, Incremental o Espiral, se caracterizan por imponer una disciplina de trabajo sobre el desarrollo de software haciendo énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado a través de documentación bien definida (Sommerville I. , 2011).

En el contexto del desarrollo de aplicaciones empresariales, los modelos tradicionales han demostrado pobres resultados en términos de planificación de presupuestos, tiempos de entrega y verificación de requisitos (Ahlemann, 2012). Entre las causas que generan estas debilidades están la volatilidad de los requisitos causada principalmente por la naturaleza dinámica del negocio en el que se desarrollan las aplicaciones empresariales, la falta de entendimiento de sus propias necesidades por parte de los interesados y la falta de habilidades y/o vocabulario técnico para comunicarlas (Stober, 2010).

Para afrontar estos desafíos, las metodologías de desarrollo ágiles, como SCRUM, Programación Extrema (XP) o Kanban, proponen interacciones continuas e informales con los usuarios finales siguiendo iteraciones de desarrollo cortas (Stober, 2010). Sin embargo, actualmente, las metodologías ágiles resultan de difícil aplicación en el desarrollo de aplicaciones empresariales, ya que éstas requieren de diseño arquitectónico y documentación bien definidos que aseguren su efectividad. La integración continua sin la documentación adecuada puede fácilmente degradar la arquitectura (Sommerville I. , 2011).

Por lo tanto, el sentido de este artículo, es poder brindar información de utilidad que cuente con datos de interés, mostrando los puntos a favor y en contra entre algunos modelos tradicionales y algunas metodologías ágiles. Además, brindar una referencia actual de las tecnologías usadas al día de hoy en el sector de software colombiano.

Esto se hará primeramente dando una corta explicación de los modelos y metodologías seleccionados, los cuales fueron elegidos de acuerdo a su nivel de documentación y popularidad en la industria del desarrollo, para posteriormente ser llevados a una comparativa general. Finalmente, estos datos serán contextualizados en la industria de software colombiano, con estadísticas y cifras que lo validen y así concluir con los resultados del estudio monográfico.

## 2. CONTENIDO / DESARROLLO

### 2.1 **Modelos de desarrollo tradicionales**

#### 2.1.1 Modelo de desarrollo en Cascada

Este es considerado el modelo más básico debido a su sencillez de comprensión y antigüedad, pero a su vez ha sido la base para el desarrollo de nuevas metodologías mejorando su estructura, su ejecución y la distribución de tareas para cada secuencia del ciclo del desarrollo del software. La versión original de este modelo fue propuesta por Winston W. Royce en 1970, posteriormente fue revisada por Barry Boehm y Ian Sommerville.

Las cinco fases que componen el modelo en cascada son las siguientes: Análisis de requisitos del software, diseño,

codificación, pruebas, mantenimiento **(PRESSMAN, 2010)**.

El hecho de contar con requisitos bien definidos, estables, unidos a un planteamiento efectivo del diseño y desarrollo, garantiza la generación de aplicaciones no solamente funcionales, sino también estructuralmente bien definidas, con excelente documentación y con una base sólida para futuras implementaciones u actualizaciones.

### 2.1.2 Modelo de desarrollo Incremental

Se fundamenta en la idea de proyectar una implementación inicial, exponiéndolo a las críticas de los clientes y mejorándolo a través de diferentes versiones, hasta que un sistema adecuado sea elaborado. Las labores de especificación, desarrollo y validación están mezcladas en lugar de separadas, con una pronta retroalimentación entre las tareas (Pressman, 2010)

Desarrollar software de modo incremental es más económico, por otro lado, realizar modificaciones en el software es más sencillo a medida que se va elaborando. Cada mejora o versión del sistema integra determinadas funciones que precisa el usuario. Usualmente, las primeras mejoras del sistema contienen la funcionalidad más significativa o solicitada con mayor premura. Esto quiere decir que el usuario puede valorar el sistema en una fase relativamente prematura en la elaboración para ver si la entrega cumple con lo requerido.

Los dilemas del desarrollo incremental se tornan propiamente intensos para sistemas enormes, complicados y de extensa durabilidad, en los que diferentes grupos elaboran distintas piezas del sistema. Esto debido a que cada incremento debe ser limitado (Menos de 20.000 líneas) para reducir riesgos de integración posterior (Sommerville, 2016). Los sistemas considerables pueden manejar un modelado incremental e implementar una metodología en cascada, éstas combinadas requerirán un ámbito o estructura duradera y los compromisos de los diferentes conjuntos que laboran en fracciones del sistema, tienen que fijarse abiertamente con relación a esa arquitectura.

La entrega e implementación incremental manifiesta que el software se emplea en procedimientos auténticos y operacionales (Figura 1). Esto no siempre se puede hacer, puesto que la comprobación con un nuevo software puede conllevar la interrupción de los procedimientos comerciales habituales.

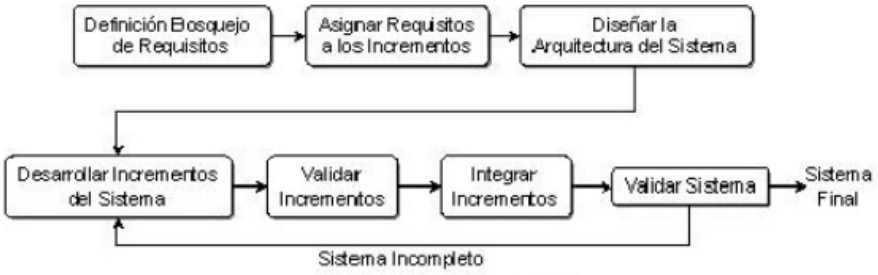


**Figura 1:** Modelo de diseño iterativo incremental **Fuente:** (Steller, 2012)

### 2.1.3 Modelo de desarrollo en espiral

Fue propuesto por primera vez por Barry Boehm, este modelo es evolutivo del proceso de software que utiliza prototipos como apoyo. Consiste regularmente en presentar entregables (prototipos) donde se evidencie implementaciones solicitadas por el cliente y que en la medida conforme se vaya revisando cada "iteración" (repetición) se vaya puliendo el producto final, yendo y revisando cada vez que sea necesario el planteamiento y los requerimientos solicitados para irse acercando al producto final, teniendo el potencial para hacer un desarrollo rápido de versiones cada vez más completas (Gamboa, 2018).

El modelo de desarrollo en espiral tuvo varias modificaciones, entre ellas encontramos: Modelo Original de Boehm, Modelo Típico de Seis Regiones y Modelo WINWIN.

Un ciclo común de desarrollo de este modelo se divide en cuatro fases: Definición de objetivos, evaluación y reducción de riesgos, desarrollo y validación, planificación (Muñoz, 2015).

El producto inicial podría definirse como un prototipo de "concepto", el cual irá evolucionando a lo largo de múltiples iteraciones que irán dando forma al proyecto pensado o mejor aún, optimizado para lo que fue solicitado (Pressman, 2010)

El prototipo del modelo en espiral para la ingeniería de software es en la actualidad el enfoque más realista para el desarrollo de software y de sistemas a gran escala (Sommerville I. , 2011).

Este es un modelo difícil de vender (sobre todo bajo contrato) al cliente, ya que es un modelo costoso de implementar, demanda mucha experiencia en la evaluación de riesgos y convencer de que los riesgos serán controlados es complicado. "No hay duda de que habrá problemas si algún riesgo importante no se descubre y administra" (Pressman, 2010).

## 2.2 **Metodologías de desarrollo ágiles**

"Las metodologías ágiles son procesos ligeros que utilizan ciclos iterativos cortos. Involucra a los usuarios con entusiasmo para establecer, priorizar y garantizar que se atiendan las necesidades" (Boehm, 2005), esto quiere decir que estas metodologías son aplicadas en el desarrollo de software con el fin de brindar resultados óptimos en corto plazo y así mejorar la satisfacción del cliente.

### 2.2.1 Metodología de desarrollo SCRUM

Introducida por Ken Swabe en el año 1995 (Butt., 2016), esta metodología es diseñada para atender a los cambios repentinos y repetitivos que pueda generarse en el desarrollo de un proyecto. SCRUM se fundamenta en la fácil documentación del desarrollo y en la relación directa entre el cliente y el grupo de desarrollo; es aquí donde se presentan los sprints, que no es más que un periodo corto de tiempo donde se obtienen resultados sobre tareas planeadas desde un inicio. En la siguiente figura, se define el funcionamiento de SCRUM y se explican los objetos en la Tabla 1.

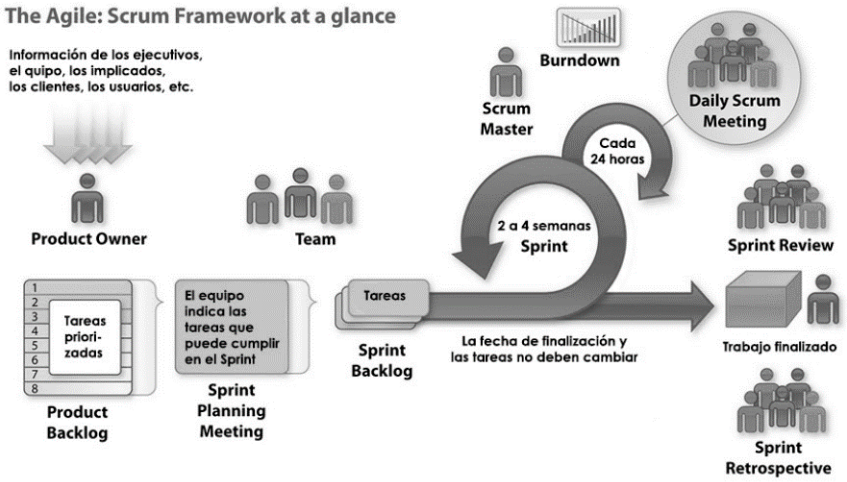


**Figura 2:** Funcionamiento de SCRUM. **Fuente:** (García, 2019)

**Tabla 1:** Características primordiales de la Figura 3 de SCRUM

| Objeto | Descripción |
|---|---|
| Cuadro de incendio | Este gráfico muestra el trabajo día a día sobre el progreso del sprint |
| Product Backlog | Es una lista completa de todas las tareas que se ejecutarán en el proyecto |
| Sprint backlog | Es una lista de tareas que se ejecutará y que no están finalizadas |

**Fuente:** (Butt., 2016)

### 2.2.2 Metodología de desarrollo Programación extrema (XP)

Introducida por Ken Beck en el año 2000 (Nawrocki, 2012), es una metodología que ayuda al desarrollo ágil y ligero, el objetivo principal es mejorar la calidad del software y la capacidad de respuesta al cambio de requerimiento del cliente, según la siguiente figura, la programación extrema se basa en cinco etapas de desarrollo, que son: Planificación, diseño, codificación, pruebas, lanzamiento. (Ver Figura 3).

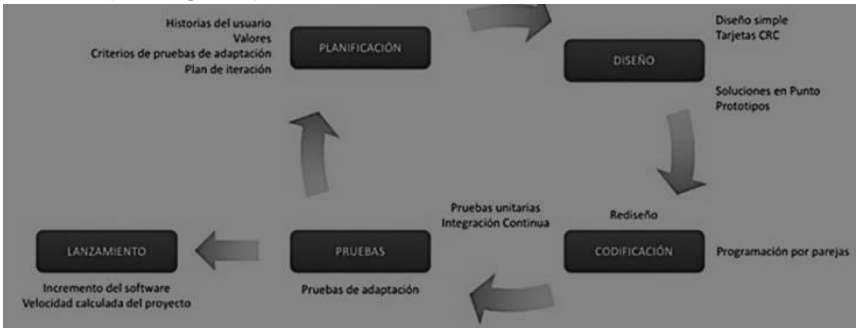


**Figura 3:** Funcionamiento de XP. **Fuente:** (Calvo, 2018)

La idea de esta metodología es que exista una retroalimentación en cada paso, entre el desarrollador y el cliente, así mismo fortalecer las relaciones entre todo el equipo de trabajo, ya que es necesaria la comunicación continua y oportuna.

## 2.3 Modelos de desarrollo tradicionales vs Metodologías de desarrollo ágiles

Para hacer una comparativa entre metodologías tradicionales y ágiles, es necesario abordar primero el concepto de metodología según Balanguera (2015):

> "Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo" (p, 122)

Ya teniendo claro el concepto de metodología, es más fácil detectar las diferencias entre los modelos tradicionales y las metodologías ágiles para el desarrollo de software. Por esto, la tabla 2 señala sus diferencias a grandes rasgos:

**Tabla 2:** Diferencias entre modelos tradicionales vs metodologías ágiles

| Modelos Tradicionales | Metodologías Agiles |
|---|---|
| Basados en normas provenientes de estándares seguidos por el entorno de desarrollo | Basadas en heurísticas provenientes de prácticas de producción de código |
| Cierta resistencia a los cambios | Especialmente preparados para cambios durante el proyecto |
| Impuestos externamente | Impuestas internamente (por el equipo) |
| Proceso mucho más controlado, con numerosas políticas/normas | Proceso menos controlado, con pocos principios. |
| El cliente interactúa con el equipo de desarrollo mediante reuniones | El cliente es parte del equipo de desarrollo |
| Más artefactos | Pocos artefactos |
| Más roles | Pocos roles |
| Grupos grandes y posiblemente distribuidos | Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio |
| La arquitectura del software es esencial y se expresa mediante modelos | Menos énfasis en la arquitectura del software |
| Existe un contrato prefijado | No existe contrato tradicional o al menos es bastante flexible |

**Fuente:** (Cadavid, 2013).

### 2.3.1 Uso de metodologías y modelos en la industria de software colombiana

Las empresas colombianas del sector de la tecnología de desarrollo de software han incrementado en los últimos años según

el MinTIC. De acuerdo con el "Censo MinTIC 2014", el sector TI cuenta con 4.016 empresas a nivel nacional. (MinTIC, 2015). Adicional a esto, sabemos según cifras del Observatorio TIC del 2016, la industria TI representa el 1,2% del PIB. (MinTIC, 2016).

Este crecimiento se ha visto impulsado en gran medida por el surgimiento de pequeñas y medianas empresas (MiPyMES) que requieren de sistemas de software de calidad y a precios no tan elevados. El hecho de hacer uso de metodologías ágiles permite el mejoramiento de tiempos, calidad, productividad y costos asociados.

En épocas pasadas, los proyectos de software eran abordados por modelos que hacían énfasis en el control de los procesos mediante una rigurosa definición de roles, actividades, recursos, artefactos, modelos de diseño debidamente detallados. Esto iba acompañado por un equipo humano "sénior" que se reunía por semanas, incluso meses para documentar con "exactitud" el tipo de software a desarrollar.

Por esto se plantea muchas veces la teoría de que al día de hoy las metodologías ágiles han ido reemplazando a las tradicionales, ya que la dinámica del mundo de hoy exige resultados en tiempos muy reducidos, esto sin perder la calidad del producto (Gamboa, 2018).

En Colombia se hace reconocimiento especial por el MinTIC a empresas que hacen uso de las metodologías ágiles, estas siendo avaladas por distintas certificaciones internacionales como IT Mark y CMMI-DEV, apuntando con esto a aumentar la competitividad del sector a nivel internacional (Figura 4).

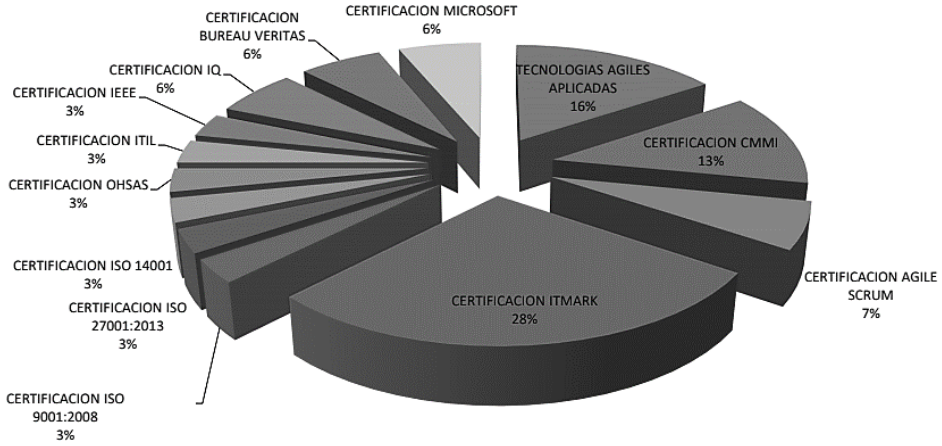


**Figura 4:** Tipos de certificaciones por empresas en Colombia.
**Fuente:** (GONZÁLEZ, 2017)

Según una encuesta de la Universidad Francisco de Paula Santander del año 2016, evidenció que de los encuestados (104 sobre una población de 1.234 egresados) usan o conocen algunas de las metodologías ágiles existentes, siendo SCRUM la más usada con el 43.42% y XP con 10,50 (Parada, 2016)

Por medio de la siguiente figura (figura 5) evidencian también las respuestas del por qué los encuestados toman la decisión de hacer uso de estas metodologías ágiles por sobre las tradicionales.

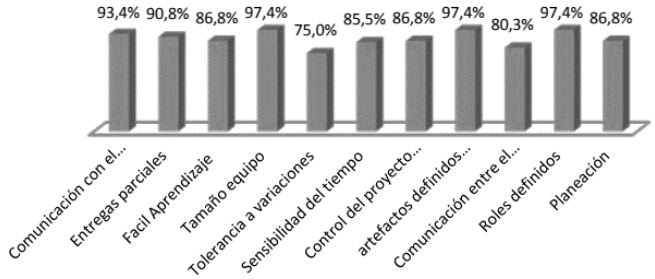


**Figura 5:** Características identificables en una metodología ágil
**Fuente:** (Parada, 2016)

A térmi ón, mientras que las metodología radicional posee

líneas base de tiempo, coste y alcance, mientras que la ágil es adaptable al cambio, su restricción principal es la calidez del producto y su rapidez de producción (Moya, 2017)

En Colombia, según estudios, el uso de metodologías ágiles ha ido incrementando, al igual que el número de empresas con certificaciones internacionales sobre el uso de estas metodologías, lo que incrementa su competitividad a nivel regional e internacional. Según informes de la Universidad Libre con sede en Barranquilla (Pedroza, 2013), se estima que a nivel mundial un 60% de los proyectos ejecutados dentro de la empresa o del departamento de desarrollo son ágiles, pero aún existe resistencia al uso de estas metodologías, lo cual dificulta su implementación en algunos proyectos que se lleven a cabo.

Las metodologías de desarrollo tradicionales nos brindan software sólido, bien estructurado y con posibilidades de futuras actualizaciones, pero esto si el cliente desde un principio tiene claro sus requerimientos, esto muy rara vez sucede (Pelaez Valencia, 2015).

No existe una metodología universal que se adapte a todo proyecto de software, pero cada metodología existente tiene sus características particulares para ser aplicadas al contexto que se quiere analizar. Por lo tanto, es preciso tener conocimiento y documentarse sobre el proceso a utilizar, para así acceder a una metodología o modelo que aporte facilidades a la hora de llevar implementaciones, futuras actualizaciones y mejor se adapte a las necesidades de nuestro proyecto (Orjuela Duarte, 2008).

## 3 CONCLUSIONES

El interés puntual este artículo consiste en presentar un recopilatorio monográfico, justificando la selección de los modelos o metodologías por lo que optamos para el estudio, y a partir de estos, brindar un panorama actual sobre el uso de estos en el desarrollo de software a nivel nacional.

Es claro que, a pesar de los grandes avances en investigación y desarrollo, gran parte de los proyectos no concluyen satisfactoriamente, esto debido a ciertos factores, entre ellos, las malas prácticas a la hora de elegir e implementar una metodología de desarrollo, ya sea ágil o tradicional.

Este tipo de artículos son importantes, ya que nos dan un marco de referencia sobre el cual se puede tener una idea de decantación, sobre qué modelo o metodología debemos usar para abordar un proyecto de desarrollo en específico. La tendencia, a día de hoy, es hacer uso de dos o más métodos combinados que nos permitan ir abarcando los objetivos trazados desde la planificación, y así poder integrar de manera más sencilla los avances realizados (Henández, 2015).

Pero concluimos, a términos generales, que las metodologías ágiles han tomado cierta delantera por sobre los modelos tradicionales, esto debido a su flexibilidad a los cambios y nuevas integraciones. Las metodologías ágiles manejan procesos independientes y modulares en sus esquemas de trabajo, esto permite que las modificaciones por replanteamiento de requisitos del usuario sean más fáciles y económicas de implementar.

## 4 REFERENCIAS

Ahlemann, F. e. (2012). Strategic enterprise architecture management: challenges, best practices, and future developments. Springer Science & Business Media.

Balaguera, Y. D. (2015). Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Revista de Tecnología, 12.

Boehm, B. T. (2005, Septiembre-Ocubre). IEEE. Retrieved from Management challenges to implementing agile processes in traditional development organizations: https://ieeexplore.ieee.org/abstract/document/1504661.

Butt., S. (2016). Study of agile methodology with the cloud. Pacific Science Review B: Humanities and Social Sciences, 22-28.

Cadavid, A. N. (2013). Revisión de metodologías ágiles para el desarrollo de software. Retrieved from Universidad Autónoma del Caribe:

https://uac.edu.co/images/stories/publicaciones/revistas_cientificas/prospectiva/volumen-11-no-2/4_articulo_vol_11_2.pdf

Calvo, D. (2018, Abril 7). Diego Calvo. Retrieved from Metodología XP Programación Extrema (Metodología ágil): http://www.diegocalvo.es/metodologia-xp-programacion-extrema-metodologia-agil/

Fowler, M. (2002). Patterns of enterprise application architecture. Boston: Addison-Wesley.

Gamboa, J. Z. (2018). Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software. INNOVA Research Journal.

García, F. J. (2019). Metodologías de Ingeniería de Software. Salamanca: Universidad de Salamanca.

GONZÁLEZ, H. R. (2017). Universidad Nacional de Colombia. Retrieved from Análisis y diseño de un modelo con: http://bdigital.unal.edu.co/62425/1/tesisentregafinalmodelodefinitivo.pdf

Henández, G. (2015). Metodología adaptativa basada en Scrum: Caso empresas. Revista Tecnológica ESPOL.

MinTIC. (2015). Caracterización del sector de teleinformática, software y TI en Colombia 2015.

MinTIC. (2016, Agosto 11). Ministerio de Tecnologías de la Información y las Comunicaciones. Retrieved from https://www.mintic.gov.co/portal/604/w3-article-16015.html

Moya, J. (2017, Febrero 20). PMI Madrid . Retrieved from https://pmi-mad.org/index.php/socios/articulos-direccion-proyectos/1288-metodologia-agil-vs-metodologia-tradicional

Muñoz, J. (2015). Estudio de las fases de ciclo de desarrollo de software educativo con empleo de metodología ágil SCRUM. Machala: Universidad Técnica de Machala.

Nawrocki, J. e. (2012, Septiembre 13). Extreme programming modified: embrace requirements engineering practices. Retrieved from IEEExplore Digital Library: https://ieeexplore.ieee.org/abstract/document/1048543.

Orjuela Duarte, A. (2008). Las metodologías de desarrollo ágil como una oportunidad para la Ingeniería de Software Educativo. Avances en Sistemas e Informática- UNAL.

Parada, C. J. (2016). Caracterización de las metodológias agiles para el desarrollo de apliaciones moviles. Retrieved from Universidad Francisco de Paula Santander. Facultad de Ingenierías. Colombia: http://service.udes.edu.co/eisi/memorias/ponencias/ep17.pdf

Pedroza, P. (2013). Elección de una metodología de desarrollo a partir de las ventajas de una metodología ágil y un modelo robusto como CMMI-DEV 1.3. Ingeniare.

Pelaez Valencia, L. E. (2015). Caracterización del proceso de desarrollo de software en Colombia-Una mirada desde las PYMES productoras. Revista académica e institucional de la UCPR.

Pressman, R. S. (2010). Ingeniería del software. Un enfoque práctico. Mexico: The McGraw-Hill.

ProColombia. (n.d.). ProColombia. Retrieved from EXPLORE OPORTUNIDADES DE NEGOCIOS: http://www.procolombia.co/compradores/es/explore-oportunidades/software-0

Sommerville. (2016). Software Engineering, 10th Ed. London: Pearson Education Limited.

Sommerville, I. (2011). Ingenieria de Software. Ciudad de México: Pearson Educación de México, S.A. de C.V.

Steller, V. (2012). Proceso de desarrollo de software y materiales educativos computarizados. Revista de Tecnología de Información y Comunicación en Educació, 93.

Stober, T. H. (2010). Agile Software Development: Best Practices for Large Software Development Projects. Springer Science & Business Media.