

# El uso de las metodologías ágiles y su importancia para el desarrollo de software

## The use of dynamic methodologies and their importance for software development

Dora López Mora<sup>1\*</sup>, Marco Villamar Coloma<sup>1</sup>, Ángela Bravo Pino<sup>1</sup> y Evelin Lozano Rodríguez<sup>2</sup>

Instituto Tecnológico Superior Babahoyo<sup>1</sup>

Universidad Técnica de Babahoyo<sup>2</sup>

\*dlopez@itsb.edu.ec

DOI: [https://doi.org/10.26871/killkana\\_tecnica.v3i1.473](https://doi.org/10.26871/killkana_tecnica.v3i1.473)

### Resumen

Desarrollar un software sin duda no es trabajo fácil, por lo que existen diversas metodologías para facilitar el proceso en el desarrollo de proyectos, en este caso de Ingeniería de software. Las Metodologías de desarrollo ágil permiten mejores resultados, que facilitan el mejoramiento continuo en el desarrollo del producto basado en el aprendizaje, innovación y cambios continuos con el fin de mejorar la calidad, fiabilidad y seguridad del software. Por otra parte, se tienen aquellas propuestas más tradicionales que se centran en especial en el control del proceso, donde se establecen de forma rigurosa las actividades involucradas, los efectos que se deben producir, las herramientas y notaciones que se usarán, siendo muy efectivas en unos casos, pero también, presentando problemas en muchos otros. Las metodologías ágiles están revolucionando la manera de producir software, y a la vez generan un amplio debate entre sus seguidores y quienes por escepticismo o convencimiento no las ven como alternativas antes las metodologías tradicionales. En este artículo a través de una revisión bibliográfica se pretende evidenciar el impacto de las metodologías ágiles en la actualidad al momento de desarrollar un software, en específico Scrum y Programación Extrema (eXtreme Programming, XP) a la que se considera la metodología ágil más popular en la actualidad.

**Palabras clave:** Desarrollo, Software, calidad, Metodologías Ágiles, Programación Extrema (XP).

### Abstract

*Developing a software program is certainly not an easy job. Therefore, there are several methodologies that, in the case of software engineering, facilitate the development of the projects' process. Dynamic development methodologies allow better results, facilitating the continuous improvement in product development, which is based on learning, innovation, and continuous changes, in order to improve software quality, reliability, and security. On the other hand, there are more traditional proposals that focus especially on the control of processes. Here, the activities involved, the effects that must be produced, the tools, and the notations that will be used, are rigorously established, and are very effective in some cases, but present difficulties in others. Dynamic methodologies are transforming the way software is being produced. At the same time, they are generating a broad debate among their followers and those who, because of their skepticism or conviction, do not see them as alternatives to traditional methodologies. Through a bibliographic review, this article intends to highlight the impact that dynamic methodologies currently have when developing software, specifically Scrum and Extreme Programming (eXtreme Programming, XP), which is nowadays considered the most popular dynamic methodology.*

**Key words:** Development, Software, quality, Dynamic Methodologies, Extreme Programming (XP).

### I. INTRODUCCIÓN

El desarrollo de software y la gran necesidad de que los proyectos logren al éxito y sobre todo obtener un producto de gran valor para los clientes, generan grandes cambios en las metodologías adoptadas por los equipos para cumplir sus objetivos, puesto que unas se adaptan mejor que otras, al contexto del proyecto brindando mejores ventajas.

En las décadas de los noventa nacen las metodologías ágiles como respuesta a los problemas que se presentan en las tradicionales; es decir, metodologías robustas que ajustadas en un equipo cumpla con sus metas, y satisfaga más allá de las necesidades definidas al inicio del proyecto.

Este artículo a través de una revisión bibliográfica de varias fuentes busca demostrar el impacto de las meto-

dologías ágiles al momento de desarrollar un software, donde se comparan las diferencias entre las metodologías tradicionales y las ágiles, así como las etapas y enfoque metodológicos y aspectos fundamentales como son las características del proyecto y la curva de aprendizaje.

## II. METODOLOGÍAS TRADICIONALES

Las metodologías tradicionales de desarrollo de software son orientadas por planeación. Inician el desarrollo de un proyecto con un riguroso proceso de elicitación de requerimientos, previo a etapas de análisis y diseño. Con esto tratan de asegurar resultados con alta calidad circunscritos a un calendario. [1]

Las metodologías tradicionales (formales) se focalizan en documentación, planificación y procesos. (Plantillas, técnicas de administración, revisiones, etc.). A continuación, se detalla Rational Unified Process (RUP) uno de los métodos más usados dentro de los métodos tradicionales

### 1. RATIONAL UNIFIED PROCESS (RUP)

RUP es un proceso formal: Provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Fue desarrollado por Rational Software, y está integrado con toda la suite Rational de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte. (Customización). Es guiado por casos de uso y centrado en la arquitectura, y utiliza UML como lenguaje de notación.

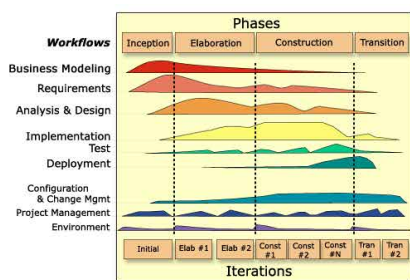


Fig. 1. Proceso Unificado Rational

## III. METODOLOGÍAS ÁGILES

Las metodologías ágiles son flexibles, pueden ser modificadas para que se ajusten a la realidad de cada equipo y proyecto. Los proyectos ágiles se subdividen en proyectos más pequeños mediante una lista ordenada de características. [1]

Cada proyecto es tratado de manera independiente y desarrolla un subconjunto de características durante un periodo de tiempo corto, de entre dos y seis semanas. La comunicación con el cliente es constante al punto de requerir un representante de él durante el desarrollo. Los

proyectos son altamente colaborativos y se adaptan mejor a los cambios; de hecho, el cambio en los requerimientos es una característica esperada y deseada, al igual que las entregas constantes al cliente y la retroalimentación por parte de él. Tanto el producto como el proceso son mejorados frecuentemente. [2]

Hablar de metodologías ágiles implica hacer referencia a las metodologías de desarrollo de software tradicionales ya que las primeras surgieron como una reacción a las segundas. Sus características principales son antagónicas y su uso ideal aplica en contextos diferentes. [1]

En tiempos recientes, han surgido las metodologías ágiles como una alternativa, una reacción a las metodologías tradicionales y principalmente a su burocracia. [3]

Diversos autores coinciden en señalar algunos requisitos que deben tener las metodologías de desarrollo:

- Visión del producto.
- Vinculación con el cliente.
- Establecer un modelo de ciclo de vida.
- Gestión de los requisitos.
- Plan de desarrollo.
- Integración del proyecto.
- Medidas de progreso del proyecto.
- Métricas para evaluar la calidad.
- Maneras de medir el riesgo.
- Como gestionar los cambios.
- Establecer una línea de meta.

### 1. EXTREME PROGRAMMING (XP)

Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. [4]



Fig. 2. XP Practices

### A. SCRUM

Scrum es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. El desarrollo

se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora nueva funcionalidad. Las iteraciones en general tienen una duración entre 2 y 4 semanas. Scrum se utiliza como marco para otras prácticas de ingeniería de software como RUP o Extreme Programming.

Scrum se focaliza en priorizar el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión. Está diseñado en especial para adaptarse a los cambios en los requerimientos, por ejemplo, en un mercado de alta competitividad. Los requerimientos y las prioridades se revisan y ajustan durante el proyecto en intervalos muy cortos y regulares. De esta forma se puede adaptar en tiempo real el producto que se está construyendo a las necesidades del cliente. Se busca entregar un software que en realidad resuelva las necesidades, aumentando la satisfacción del cliente. [4]

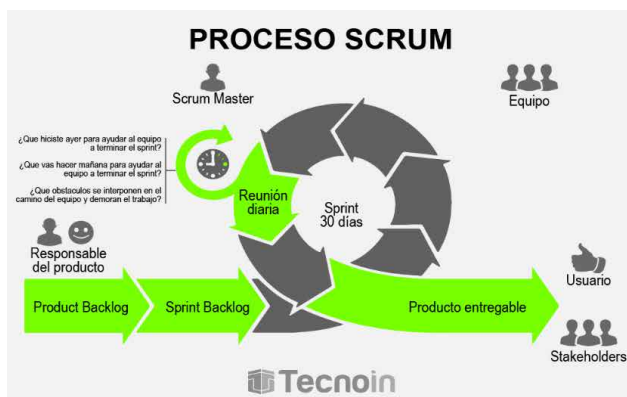


Fig. 3. Metodología SCRUM

#### IV. MATERIALES Y MÉTODO

Con la finalidad de demostrar el impacto de las metodologías ágiles al momento de desarrollar un software, fue necesario realizar un análisis de las ventajas y desventajas de la metodología ágiles tradicionales de varios artículos publicados como: *Revisión de metodologías ágiles para el desarrollo de software* del autor Andrés Navarro [1], a través del método de comparación, que permita validar la importancia de estas y así concluir cuál será el impacto que tendrán.

En primer lugar, se llevó a cabo una búsqueda en Google Académico de documentos publicados por diferentes autores, tales como *Metodologías de desarrollo de software. ¿Cuál es el camino?* Del autor Ing. Erly Delgado Exposito [5].

Respecto a las revisiones sistemáticas y los estudios científicos se aplicó como criterio de inclusión que los trabajos realizados sobre la importancia del uso de Metodologías Ágiles, y su importancia en el desarrollo de software, así como documento denominado *Manifiesto por el desarrollo ágil*.

En 2001 se crea el *Manifiesto por el desarrollo ágil de software*, documento en el que se acuerdan cuatro principios básicos para el desarrollo de software. Establece prioridades y marca diferencias de fondo frente a los sistemas tradicionales: individuos e interacciones, por encima de procesos y herramientas; software funcionando por encima de documentación extensiva; colaboración con el cliente, por encima de negociación contractual; y respuesta ante el cambio, por encima de seguir un plan. [6]

Las metodologías ágiles son flexibles, sus proyectos son subdivididos en proyectos más pequeños, incluyen comunicación constante con el cliente, y son altamente colaborativos y se adaptan mejor a los cambios. De hecho, el cambio en los requerimientos es una característica esperada al igual que las entregas constantes al cliente y la retroalimentación por parte de él. Tanto el producto como el proceso son mejorados frecuentemente. [7]

#### V. RESULTADOS

##### Comparación entre metodologías

La tabla I muestra aspectos relevantes de las metodologías de desarrollo tradicional contrastándolas con los aspectos relevantes de las metodologías de desarrollo ágil.

TABLA I. Diferencias entre metodología tradicionales y ágiles

Metodologías Tradicionales	Metodologías ágiles
Basadas 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
Impuestas 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: Metodologías Ágiles[4]

En la tabla 2 se puede observar un cuadro comparativo entre las etapas más comunes del desarrollo de software y los enfoques de metodologías revisados.

TABLA II. Diferencias por etapas y enfoque metodológico

Modelos Rigurosos	Etapas	Modelos Ágiles
Planificación predictiva y "aislada"	Análisis de requerimientos	Planificación adaptativa: Entregas frecuentes + colaboración del cliente
	Planificación	
Diseño flexible y Extensible + modelos + Documentación exhaustiva	Diseño	Diseño Simple: Documentación Mínima + Focalizado en la comunicación

TABLA II. Diferencias por etapas y enfoque metodológico ...continuación

Modelos Rigurosos	Etapas	Modelos Ágiles
Desarrollo individual con Roles y responsabilidades estrictas	Codificación	Transferencia de conocimiento: Programación en pares + conocimiento colectivo
Actividades de control: Orientado a los hitos + Gestión miniproyectos	Pruebas	Liderazgo-Colaboración: empoderamiento +auto-organización
	Puesta en Producción	

Fuente: Metodologías Tradicionales Vs. Metodologías Ágiles [2]

También se puede encontrar un cuadro comparativo por los dos aspectos analizados:

TABLA III. Por las características del proyecto

Modelo de Proceso	Tamaño del Proceso	Tamaño del Equipo	Complejidad del Problema
RUP	Medio / Extenso	Medio / Extenso	Medio / Alto
ICONIX	Pequeño / Medio	Pequeño / Medio	Pequeño / Medio
XP	Pequeño / Medio	Pequeño	Medio / Alto
SCRUM	Pequeño / Medio	Pequeño	Medio / Alto

Fuente: Metodologías Tradicionales Vs. Metodologías Ágiles[2]

TABLA IV. Por la curva de aprendizaje

Modelo de Proceso	Curva de aprendizaje	Herramienta de integración	Soporte Externo
RUP	Lenta	Alto Soporte	Alto Soporte
ICONIX	Rápida	Algún Soporte Disponible	Algún Soporte Disponible
XP	Rápida	No mencionado	Algún Soporte Disponible
SCRUM	Rápida	No mencionado	Algún Soporte Disponible

Fuente: Metodologías Tradicionales Vs. Metodologías Ágiles[2]

## VI. DISCUSIÓN DE RESULTADOS

Como se puede observar en la tabla I las metodologías ágiles permiten disminuir costos y brindar flexibilidad a los proyectos de software, además los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados. Con el uso de estas metodologías es prioridad crear un producto software que funcione más que escribir documentación exhaustiva.

Con estas metodologías temas como la colaboración del cliente, la cual debe prevalecer sobre la negociación de contratos, donde la capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan.

En lo que se denomina diferencia por etapas y enfoque metodológico, se destacan temas relevantes como el análisis y la planeación. Se pueden fusionar logrando un enfoque tan importante como es la comunicación y la colaboración del cliente, enfoque que se manifiesta de gran importancia en el diseño del producto. Además, se resalta en la etapa de

la codificación la necesidad de un trabajo en equipo para si lograr una puesta en marcha donde se demuestre liderazgo, colaboración y empoderamiento.

En los aspectos analizados las características del proyecto se muestra una gran ventaja como es el tamaño del proceso, del equipo y la complejidad del problema para cada uno de los modelos. Se resaltar que: con un pequeño equipo de desarrollo se pueden realizar grandes proyectos, de alta complejidad; es el caso de XP y SCRUM.

En el análisis realizado sobre otro de los aspectos como lo es la curva de aprendizaje, las metodologías ágiles ofrecen una gran ventaja con ciertas limitaciones, debido a que no han sido explotadas a gran escala como lo es RUP que posee alto soporte y herramientas integrales, facilitando aplicar con mayor efectividad esta metodología, y permitirán aprovecharla al máximo.

Esto ayudaría a evidenciar el impacto de las metodologías ágiles en la actualidad al momento de desarrollar un software.

## VII. CONCLUSIONES

La historia indica que las metodologías tradicionales abordan la mayor cantidad de situaciones de contexto del proyecto, marcando exigencia y un esfuerzo para ser adaptadas, más que todo en proyectos pequeños y con requisitos muy cambiantes. Las metodologías ágiles ofrecen una solución de proyectos que tienen estas características. Entre las cualidades principales en una metodología ágil es su sencillez, tanto en su aprendizaje como en su aplicación, minimizando costos de implantación en un equipo de desarrollo.

La importancia de las metodologías ágiles radica que funcionan dentro de un contexto específico conformado por pequeños equipos de desarrollo, trabajando en el mismo sitio, con clientes que pueden tomar decisiones acerca de los requerimientos y evolución de los mismo, requerimientos que pueden cambiar con una frecuencia semanal y/o mensual, logrando un proyecto de presupuesto variable.

## REFERENCIAS

- [1] F. M. J. D. Navarro cadavid Andres Morales Velez Jonathan, "Revision de metodologias agiles para el desarrollo de software," *Redalyc*, vol. 11, no. 2, pp. 30–39, 2013.
- [2] R. G. Figueroa, "metodologias Tradicionales Vs. Metodologias agiles," *www.academia.edu*, 2008.
- [3] I. O. T. Gómez, "Criterios de selección de metodologías de desarrollo de software," *www.redalyc.org*, p. 7, 2010.
- [4] P. L. Jose H. Canos Maria del Carmen Penades, "Metodologia Agiles para el desarrollo de software," *Dialnet*, vol. 5, no. 26, p. 26, 2015.
- [5] I. E. D. Exposito, "Metodologías de desarrollo de software ¿Cuál es el camino?," *www.redalyc.org*, vol. 2, no. 3, p. 7, 2008.

- [6] D. Thomas, “Manifiesto for Software Agile Development,” 2001.
- [7] S. Ghosh, “www.umd.edu,” 2013.



***Recibido:*** 6 de enero de 2019

***Aceptado:*** 10 de marzo de 2019

