

Pautas para optar por una metodología ágil para proyectos de software

Lucy Nohemy Medina Velandia ^a & Daniel Andrés Gutiérrez ^b

^a Facultad de Ingeniería y Ciencias Básicas de la Universidad Los Libertadores, Bogotá, Colombia.

^b Grupo LPA, Santiago, Chile.

lunome@gmail.com, dgutierrez@grupolpa.com

Resumen— El dinamismo en la industria del software ha transformado la forma en que se desarrollan las aplicaciones. La ingeniería de software ahora se centra en satisfacer las necesidades de manera rápida, flexible y con alta calidad para mejorar la productividad y la satisfacción del cliente. A pesar de la variedad de metodologías de desarrollo disponibles, seleccionar la adecuada para un proyecto específico puede ser un desafío. Este estudio ofrece pautas para la selección de una metodología ágil apropiada para el desarrollo de software, basadas en un análisis de 50 empresas de software, tanto pequeñas como medianas, ubicadas en Bogotá, Colombia. Utilizando un enfoque cuantitativo con alcance descriptivo y explicativo, se recopiló datos a través de encuestas en línea y entrevistas. Los resultados indican que el 44% de los desarrolladores prefieren la metodología Scrum, mientras que el 46% de los líderes de proyectos optan por una combinación de Scrum y Kanban. Estas metodologías ágiles se han convertido en fundamentales en el desarrollo de software actual.

Palabras Clave— innovación; proyectos de software; metodología de desarrollo ágil; pautas; ingeniería de software.

Recibido: 30 de octubre de 2023. Revisado: 23 de enero de 2024. Aceptado: 16 de enero de 2024.

Guidelines for Choosing an Agile Methodology for Software Projects

Abstract— The dynamism in the software industry has transformed the way applications are developed. Software engineering now focuses on addressing needs quickly, flexibly, and with high quality to enhance productivity and customer satisfaction. Despite the variety of development methodologies available, selecting the right one for a specific project can be challenging. This study provides guidelines for selecting an appropriate agile methodology for software development, based on an analysis of 50 software factories, both small and medium-sized, located in Bogotá, Colombia. Using a quantitative approach with a descriptive and explanatory scope, data was collected through online surveys and interviews. The results indicate that 44% of developers prefer the Scrum methodology, while 46% of project leaders opt for a combination of Scrum and Kanban. These agile methodologies have become essential in current software development.

Keywords— innovation; software projects; agile development methodology; guidelines; software engineering.

1 Introducción

El software es una herramienta esencial en la vida cotidiana y se actualiza constantemente para satisfacer las necesidades de los usuarios. La Ingeniería de Software se define como la aplicación sistemática y disciplinada de principios y técnicas para crear y mantener software que cumpla con los requisitos de los usuarios [1], [25].

Las actividades básicas en el desarrollo de software, independientemente de la metodología utilizada, incluyen

requerimientos, análisis, diseño, implementación, pruebas y mantenimiento. Para representar el problema a resolver, se requiere una metodología o modelo de desarrollo de software [1].

La elección de la metodología adecuada depende de las necesidades del proyecto, y puede ser tradicional, orientada a objetos o ágil [2] - [3]. Este trabajo contribuye proporcionando pautas para la selección de la metodología adecuada en la planificación del proyecto de software. Beneficia a empresas de desarrollo, sin importar su tamaño, así como a estudiantes que se encuentran en los últimos semestres o están desarrollando proyectos finales. Las fuentes citadas respaldan la información presentada en este resumen. La estructura del documento incluye una revisión de literatura, el procedimiento de recopilación de datos, resultados y discusión, así como conclusiones y recomendaciones que servirán como base para futuras investigaciones. Cabe destacar que el presente proyecto de investigación titulado “Pautas para optar por una metodología ágil para proyectos de software”, tuvo dos fuentes de financiación, de una parte, la Universidad Los Libertadores y de otra, la empresa LPA Corporate Solutions, de Santiago de Chile.

2 Revisión de Literatura

El desarrollo de software va más allá de la mera codificación; implica practicar de manera organizada y detallada, priorizando la calidad, el orden y las buenas prácticas, desde la definición de requisitos hasta la documentación. Las metodologías para el desarrollo de software han evolucionado significativamente, pasando de enfoques rígidos a metodologías ágiles, iniciadas en 2001 con el Manifiesto Ágil, liderado por Kent Beck y otros 16 colaboradores.

Este manifiesto promueve la adaptabilidad y la colaboración, centrando la atención en las personas como el activo más valioso de las empresas.

Los cuatro valores fundamentales del Manifiesto Ágil se resumen en la tabla 1, según [4], y se orientan hacia la creación de modelos organizativos que priorizan la colaboración y la concentración en las personas en el desarrollo de software.

En resumen, los valores esenciales del Manifiesto Ágil se centran en destacar la importancia de la persona en un proyecto, la necesidad de que el software cumpla con los requisitos y funcione correctamente, la adaptabilidad ante el cambio y la colaboración entre desarrolladores y usuarios.

Tabla 1
Valores en los que se fundamenten las metodologías ágiles

Valores	Descripción
Individuos e interacciones sobre procesos y herramientas	Las personas están por encima de los procesos y las herramientas
	Los procesos y las herramientas son apoyo para que los humanos cumplan objetivos
	Los procesos han de adaptarse a las empresas, los equipos y las personas
	Los resultados no se logran por sí solos, sino que las herramientas incrementan la eficiencia
	Las personas deberán proponer cambios y adaptarse a ellos rápidamente
Software funcionando sobre documentación extensiva	El software debe cumplir con todas las funcionalidades previstas
	Los usuarios deberán interactuar con el prototipo y retroalimentar al desarrollador
	La documentación deberá ser únicamente la necesaria, que servirá como soporte
Colaboración con el cliente sobre negociación contractual	Como los entornos cambian constantemente, se debe crear el producto en compañía del usuario y con su retroalimentación constante
	El usuario o cliente son parte del equipo de trabajo
Respuesta ante el cambio sobre seguir un plan	Los desarrollos de software han de dar respuesta a las necesidades y adaptarse a los cambios Las metodologías ágiles se deben anticipar, adaptar y aprovechar los cambios

Fuente: Adaptado de Palacio [4], p.15

Los 12 principios del desarrollo de software Ágil según [4] subrayan la satisfacción del cliente, la flexibilidad en los requisitos, la entrega frecuente de software funcional, la colaboración continua y la mejora constante del equipo. Estos valores y principios no se limitan al desarrollo de software, sino que también se aplican en otros campos y han impulsado la adopción de enfoques ágiles en la gestión de proyectos a nivel global, es el caso de la asociación mundial Project Management Institute (PMI) [18] en su informe anual Pulse of the profession (2021), realiza anualmente una encuesta a nivel mundial sobre la dirección de proyectos, asegura que por medio de las empresas que hoy se denominan gimnásticas y ágiles, han acatado la ventaja que ofrece la tecnología (Fig. 1), teniendo como premisa que estas metodologías apoyan un flujo de trabajo estable y permite adaptar los procesos a sus propias necesidades, en vez de adoptar marcos altamente prescriptivos [25].

El uso de empresas ágiles que integran tecnología y enfoques ágiles en proyectos está en aumento. Aunque la elección de una metodología para el desarrollo de software no es sencilla, según lo afirmado por [15], las soluciones en la nube experimentan actualmente un crecimiento del 11 % en ventas en comparación con los métodos tradicionales. Las empresas que implementan IoT y metodologías ágiles han registrado un aumento del 8 %, mientras que aquellas que incorporan Inteligencia Artificial e Internet móvil 5G han experimentado un crecimiento del 5 %.

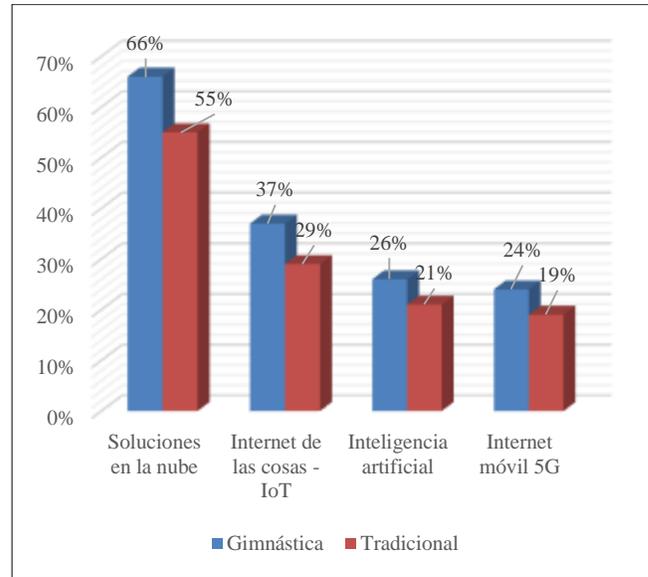


Figura 1. Ventajas de la tecnología en las empresas
Fuente: Adaptado de PMI, [18], p. 11.

En contraste, las metodologías tradicionales en el desarrollo de proyectos de software requieren definiciones completas desde el principio y no son adaptables a cambios [20]. Por otro lado, las metodologías ágiles permiten la flexibilidad para adaptarse a cambios durante el proyecto, lo que se refleja en una planificación a corto plazo, a menudo de pocas semanas, con posibilidad de adaptación en periodos posteriores.

Entre las metodologías ágiles reconocidas se encuentran Scrum, Kanban, Extreme Programming (XP), Crystal Methodologies, Dynamic Systems, Development Method (DSDM), Adaptive Software Development (ASD), Feature-Driven Development (FDD), y Lean Development (LD). Las más utilizadas son Scrum, XP y Kanban, y estas se analizarán en detalle [5], [22], [23].

Scrum, originada en 1986 en Japón, es ampliamente adoptada en la construcción de software, destacando su énfasis en la velocidad, flexibilidad y calidad. Los autores de [6] fueron promotores clave de Scrum Edge (2019), que organiza a los equipos para lograr objetivos eficientes y entregas de alta calidad. Scrum incluye roles como el equipo de desarrollo, el *product owner* y el *scrum master*, así como artefactos que ayudan a mantener la transparencia y supervisión del proceso [7].

Scrum es un marco de trabajo que se enfoca en la gestión de proyectos. Se basa en eventos, que permiten inspeccionar y analizar el proceso de principio a fin. Los cinco eventos de Scrum son:

- Sprint: ciclos ideales de una semana a cuatro semanas con objetivos y entregables incrementales.
- Sprint Planning: define el objetivo del *sprint* y revisa los elementos a desarrollar.
- Daily Scrum: reuniones diarias del equipo para compartir logros, tareas del día y obstáculos.

- Sprint Review: revisión del incremento del proyecto, que puede llevar a modificaciones en el *backlog*.
- Retrospectiva de Sprint: reunión para mejorar relaciones del equipo, procesos y herramientas utilizadas [4], [19], [21].
- Conocimiento de la metodología Kanban: el equipo se familiariza con Kanban.
- Puesta en marcha inicial: se comienza con componentes que presentan problemas y se resuelven.

Extreme Programming (XP). XP es otra metodología ágil centrada en el desarrollo de software. Kent Beck la desarrolló y enfatiza la planificación, el seguimiento y la adaptabilidad. Las características clave de XP incluyen:

- Énfasis en lo técnico: enfatiza la parte técnica del desarrollo de software.
- Alta calidad del software: orienta a equipos medianos y pequeños para producir software de alta calidad.
- Valores de XP: comunicación, sencillez, retroalimentación, valor y respeto [8], [22], [24].

Valores de XP según Beck (2012):

- Comunicación: necesaria para compartir ideas y conocimientos entre los miembros del equipo.
- Sencillez: animar a hacer las cosas de la manera más simple posible sin perder el objetivo.
- Retroalimentación: fundamental para monitorear procesos, corregir rumbos tempranamente y realizarse entre parejas o miembros del equipo.
- Valor: acción eficaz frente al miedo, mostrando coraje en adversidades para recibir y entregar críticas o información.
- Respeto: debe existir preocupación y respeto por los compañeros de proyecto, los clientes, el proyecto y los usuarios en general.
- Así, estas dos metodologías ágiles, Scrum y XP, se centran en la gestión de proyectos y el desarrollo de software, cada una con sus propias características y valores clave.

Kanban es una metodología que se originó en Japón, inicialmente relacionada con la gestión de negocios y letteros para comunicar servicios de forma clara. Posteriormente, en la década de 1950, Toyota la adoptó como parte de su estrategia de producción para mejorar la eficiencia y control. Kanban se popularizó en la industria del software a partir del año 2000 [13].

La esencia de Kanban se refleja en un tablero con tres columnas: (a) Trabajo Pendiente, (b) Trabajo en Progreso y (c) Trabajo Terminado. Cada tarea se representa con tarjetas de colores y se mueve a través de las columnas a medida que progresa. Kanban utiliza un enfoque incremental en lugar de iteraciones. No se inicia una nueva tarea hasta que la actual se haya completado. Esto fomenta la mejora continua.

Aunque la propuesta original de Kanban incluía tres columnas, algunos grupos, como Team Asana [14], han adaptado el sistema con cuatro columnas: (a) Por Hacer, (b) En Proceso, (c) Pendiente de Revisión y (d) Hecho. De ahí que para que Kanban funcione, se deben seguir cuatro principios: (a) Garantizar la calidad, (b) Reducir el desperdicio, (c) Fomentar la mejora continua y (d) Mantener la flexibilidad. Estos principios facilitan la supervisión del progreso y la gestión eficiente de las tareas [9].

Es así como Kanban se implementa en varias fases: [10]

- Implementación en el resto de los componentes: una vez resueltos los problemas principales, se aplica Kanban al resto.
- Revisión continua: se revisan y reajustan las fases según sea necesario, manteniendo la secuencia de tareas.

Finalmente, Kanban es una metodología de gestión de tareas que se originó en la industria automotriz y se adaptó exitosamente a la industria del software. Su enfoque en la visualización del trabajo, la mejora continua y la flexibilidad lo hace valioso para equipos y proyectos de diversos tipos [9], [14].

3 Metodología

En esta investigación se empleó un enfoque cuantitativo con un alcance descriptivo y explicativo. El enfoque cuantitativo según [11] se basa en la recolección de datos para probar hipótesis mediante mediciones numéricas y análisis estadístico. Se utilizó el alcance descriptivo para definir las propiedades y particularidades de las empresas desarrolladoras de software en términos de sus metodologías de desarrollo. El alcance explicativo se aplicó para comprender las causas y el contexto de la situación en las empresas de la muestra. [12]

Para recopilar datos, se emplearon encuestas y entrevistas. Las encuestas se realizaron utilizando Google Forms R® y se centraron en explicar y describir el uso de las metodologías de desarrollo de software. Se utilizaron preguntas cerradas y se analizaron los datos con el software SPSS R®. Estas encuestas fueron transversales y se llevaron a cabo en un período específico para conocer las metodologías utilizadas en los proyectos de software.

4 Resultados y discusión

En la investigación de empresas de desarrollo de software en Bogotá, Colombia, se establecieron criterios de selección que incluyeron empresas ubicadas en Bogotá, con tamaños que se ajustaran a las categorías de pequeñas (11-50 empleados) y medianas (51-200 empleados) [16]. De 1.014 resultados de empresas registradas ante la Cámara de Comercio, 483 estaban en Bogotá, y se enfocaron en aquellas relacionadas con el desarrollo de software. Se encontraron 241 pequeñas, 19 medianas, 5 grandes y 13 corporativas, con líneas de negocio como desarrolladoras de sistemas informáticos (39.10 %), consultoría TI (29.60 %), procesamiento y hosting de datos (5.20 %), edición de software (2.70 %) y diseño y desarrollo web (2.3 %) [17].

La muestra final estuvo compuesta por 50 empresas pequeñas y medianas dedicadas al desarrollo de software, seleccionadas en respuesta a una solicitud realizada a la población. Estas empresas respondieron de manera positiva, lo que permitió llevar a cabo la investigación con su participación.

Para recopilar datos, se emplearon encuestas cerradas y entrevistas. Los participantes expresaron sus preferencias entre diversas metodologías ágiles (Fig. 2). En orden de preferencia, Scrum, Extreme Programming, Kanban y Design Sprint se destacaron como las metodologías más apreciadas.

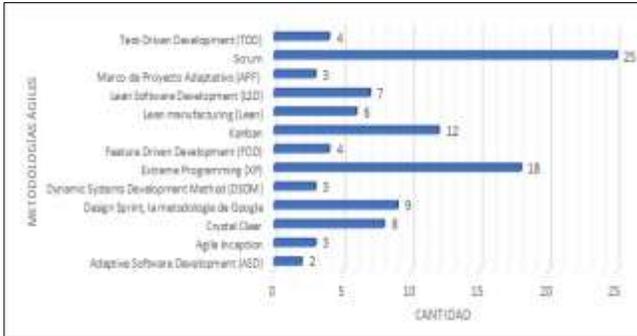


Figura 2. Preferencia de las metodologías ágiles para desarrollar software
Fuente: Elaboración propia

Inicialmente, se llevaron a cabo encuestas y entrevistas con desarrolladores y líderes de proyectos antes de que conocieran las pautas para seleccionar una metodología ágil para el desarrollo de proyectos de software. El objetivo era identificar las metodologías más conocidas entre ellos, no necesariamente las que consideraban mejores. Estas metodologías se utilizaron para crear las pautas que luego se aplicarían en proyectos futuros.

Unos de los resultados importantes fueron los elementos generales que se tuvieron en cuenta para escribir la guía, los cuales surgieron del trabajo que se tuvo con las empresas de la muestra antes y después de aplicar la guía construida y que entrega pautas para seleccionar la metodología ágil, estas fueron, entre otros:

- El tipo y tamaño de proyecto
- Los entregables del proyecto
- El presupuesto destinado para el desarrollo del proyecto
- El tiempo estimado del proyecto
- La cantidad de personas que intervienen en el proyecto
- La voluntad del usuario final en acompañar al equipo de desarrolladores
- Los coach son fundamentales en las metodologías ágiles para asesorar, orientar y guiar al equipo, los clientes y el proyecto en general.

4.1 Pautas para optar por una metodología ágil para proyectos de software utilizando Scrum

- Si su proyecto es tecnológicamente complejo, considere utilizar Scrum, ya que le permite realizar pruebas constantes y trabajar de manera gradual.
- Para proyectos de larga duración, el uso de Scrum es beneficioso, ya que le permite advertir y planificar el proyecto de manera efectiva.

- Si la implementación de su proyecto implica incertidumbre, especialmente en proyectos de innovación, considere utilizar Scrum en tramos evaluables periódicamente.
- Si en su proyecto surgen muchas ideas del equipo desarrollador, implementarlas paso a paso con Scrum es más eficiente y menos costoso que hacerlo simultáneamente.
- Si el cliente desea involucrarse en las fases del proyecto, Scrum es una opción adecuada.
- Si necesita obtener resultados rápidos, Scrum es una metodología específica que puede ayudar.
- Scrum es útil cuando el proyecto enfrenta situaciones problemáticas y tareas no completamente definidas.
- Para proyectos altamente especializados, Scrum puede ser la elección adecuada.
- Si su proyecto enfrenta dificultades para entregar partes del proyecto a tiempo, Scrum puede ayudar a mantener el enfoque y la entrega incremental.
- Scrum es apropiado cuando el alcance del proyecto cambia con frecuencia y surgen nuevos requisitos regularmente.
- Si los miembros directivos de la empresa no tienen acuerdos claros sobre la dirección del proyecto, Scrum puede facilitar la toma de decisiones.
- Scrum es ideal para equipos de trabajo pequeños.
- Si el cliente no requiere una documentación extensa, Scrum es una opción eficiente.
- Si su proyecto implica procesos que requieren interacción constante con el cliente y programación extrema, Scrum puede ser la metodología adecuada.
- Si su objetivo principal es administrar y organizar el proyecto, así como gestionar los objetivos, Scrum es una elección adecuada.
- Scrum es apropiado cuando las interacciones para la entrega de trabajo se realizan en ciclos de 1 a 4 semanas.
- Si, al finalizar un Sprint, los resultados no requieren ajustes significativos, Scrum es la metodología recomendada.
- Si anticipa dificultades en la implementación de reglas y prácticas debido al alto compromiso del personal técnico, Scrum puede ayudar a simplificar el proceso.
- Scrum es apropiado si su proyecto necesita cambiar el desarrollo de las tareas según el orden de prioridad.

Estas pautas ayudan a determinar cuándo Scrum es la metodología ágil más apropiada para proyectos de desarrollo de software, teniendo en cuenta diversos factores como la complejidad, la duración, la incertidumbre y las necesidades del cliente, así mismo, se adapta mejor, en la gestión del proyecto, ciclos de entrega cortos y estabilidad en los resultados de los Sprints, entre otros factores relevantes.

4.2 Pautas para optar por una metodología ágil para proyectos de software utilizando Extreme Programming (XP)

- Si busca calidad óptima en su producto final, XP es la elección adecuada para proyectos que priorizan la calidad del producto final.

- Utilice XP cuando su proyecto implica cambios constantes que pueden gestionarse de manera dinámica.
- Si su proyecto permite una planificación flexible del cronograma de actividades, considere utilizar XP.
- XP es una opción exitosa cuando se valora el trabajo en equipo y los requerimientos del cliente como aspectos cruciales.
- Utilice XP si planea aprovechar lecciones aprendidas y enriquecer su proyecto con la reutilización de código.
- En proyectos donde la comunicación constante entre el *tester* y el cliente es esencial, XP puede ser la elección adecuada.
- Si requiere un *mánager* que coordine las comunicaciones, gestione recursos y esté al tanto del estado del proyecto en todo momento, considere XP.
- XP es beneficioso cuando se busca equidad en las horas de trabajo y tiempos muertos del equipo.
- Utilice XP cuando los ciclos de trabajo son de 40 horas y el equipo desarrollador consta de hasta 12 personas.
- Si desea que gran parte del desarrollo se enfoque en pruebas unitarias, XP es una opción recomendada.
- XP se adapta cuando se pretende integrar funcionalidades probadas unitariamente en la versión actual para identificar problemas y realizar correcciones.
- XP es adecuado para equipos con experiencia y dominio de aspectos técnicos.
- Utilice XP cuando su proyecto se centre en la programación o creación del producto.
- XP es eficaz en proyectos con interacciones de entrega de trabajo que van de 1 a 3 semanas.
- Si las tareas pueden ser modificadas después de su aprobación, XP es una opción que facilita este proceso.
- XP es apropiado cuando el cliente ha definido las prioridades de las tareas y no se pueden modificar.

Estas pautas ayudan a identificar cuándo Extreme Programming (XP) es la metodología ágil más adecuada para proyectos de desarrollo de software, considerando diversos factores como la calidad, la gestión dinámica, la planificación, la comunicación y las necesidades del equipo.

4.3 Pautas para optar por una metodología ágil para proyectos de software utilizando Kanban

- Si se desea priorizar las actividades de acuerdo con la planificación del calendario, utilice tableros Kanban.
- Si desea gestionar su proyecto y tener una vista clara de la lista de tareas, comentarios, notas de información y el estado de cada una de ellas, utilice Kanban.
- Kanban es apropiado cuando busca un proceso productivo, organizado y eficiente en su proyecto.
- Si su proyecto se centra en la comunicación dentro del equipo, la generación de métricas y la mejora continua de los procesos, Kanban es la elección adecuada.
- Si su objetivo principal es proporcionar más valor al cliente sin aumentar los costos, utilice Kanban.

- Si su proyecto tiene como objetivo contribuir a que su negocio sea más eficiente, Kanban es una opción beneficiosa.
- Si necesita encontrar el mejor nivel de productividad con un enfoque continuo y minimizar las fallas, utilice Kanban.
- Kanban es adecuado cuando necesita que los procesos de su proyecto se ejecuten eficientemente, asignando responsabilidades y plazos para cada tarea.
- Si su proyecto no se ajusta a un tipo específico y desea gestionar tanto el proyecto como el trabajo individual de los miembros del equipo, Kanban es una elección versátil.
- Kanban es útil cuando tiene un conjunto de proyectos que requieren cambios, tareas desglosadas y no cuenta con un responsable u organizador.
- Si necesita desarrollar páginas web, la elección entre Kanban o Scrum dependerá de la empresa en la que trabaje.

Estas pautas ayudan a determinar cuándo la metodología Kanban es la más apropiada para proyectos de desarrollo de software, considerando diversos aspectos como la organización, la comunicación, la eficiencia y la gestión de proyectos.

4.4 Resultados después de aplicar las Pautas para optar por el tipo de metodología a utilizar para desarrollar un proyecto de software

Con las mismas 50 empresas con las que se trabajó desde el inicio del proyecto, donde se identificaron tres metodologías de desarrollo utilizadas periódicamente en sus proyectos de software (Scrum, XP y Kanban), se entregó a los coordinadores o gestores de proyectos de cada empresa un compendio de pautas elaborado por los autores de este escrito en un documento para su estudio. Una vez que iniciaron un nuevo proyecto de desarrollo, se llevaron a cabo nuevamente encuestas y entrevistas con los gestores o coordinadores de proyectos, con los siguientes resultados (Fig. 3).

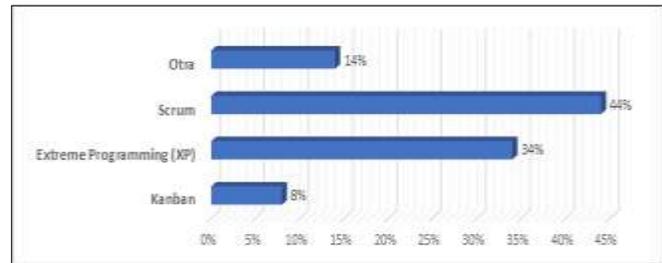


Figura 3. Resultado de las metodologías escogidas por los desarrolladores luego de la presentación de las pautas
Fuente: Elaboración propia

En la gráfica anterior se confirma que las metodologías más preferidas por los desarrolladores de software son, en su orden, Scrum (44 %) y XP (34 %). La diferencia con lo que se mostró en la gráfica 4 es que ahora surge una nueva opción: "Otra" (14 %), y finalmente, Kanban (8 %).

Cuando se les preguntó acerca de la opción "Otra", los desarrolladores indicaron que les gusta utilizar dos

metodologías al mismo tiempo en sus proyectos. Estas combinaciones incluyen Scrum y XP, Scrum y Kanban, y Kanban y XP.

Otro resultado significativo provino de las entrevistas realizadas a los coordinadores, gestores o líderes de proyectos (Fig. 4).

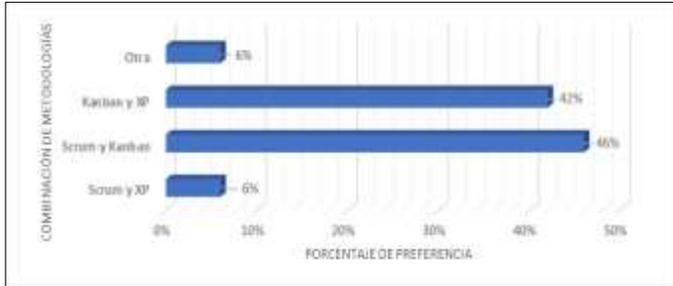


Figura 4. Resultado de las metodologías escogidas por coordinadores, gestores o líderes de proyectos luego de la presentación de las pautas

Fuente: Elaboración propia

En la gráfica anterior, se presentan los resultados después de presentar las pautas diseñadas para seleccionar una metodología ágil para el desarrollo de proyectos de software a los líderes, coordinadores o gestores de proyectos de software de las 50 empresas que formaron parte de nuestro estudio. La combinación preferida por la mayoría de ellos fue Scrum y Kanban (46 %). Esta elección se debe a que, como líderes, les permite tener un control inmediato del proyecto a través de las tarjetas de colores y los carriles en los que se ubican. Esta combinación se muestra altamente eficiente para el seguimiento sin necesidad de solicitar informes constantes o llevar a cabo reuniones frecuentes con el equipo de trabajo para revisar los avances.

Un número significativo de líderes optó por la combinación de Kanban y XP (42 %). La diferencia principal entre esta elección y la de Scrum y Kanban radica en que Scrum representa una estrategia de desarrollo incremental que permite una planificación y ejecución completa del producto, mientras que XP ofrece una gestión de proyectos más eficaz, flexible y controlada.

Una opción diferente que emergió fue la combinación de Design Sprint y Crystal (6 %). Los líderes prefieren Design Sprint porque es una metodología completa que les permite idear, crear prototipos, realizar pruebas y refinar procesos, además de tomar decisiones sobre la marcha del proyecto. Por otro lado, optan por Crystal debido a su capacidad para establecer conexiones estrechas entre las personas que forman parte de su equipo y el núcleo del proceso de desarrollo.

La mayoría de los proyectos de desarrollo de software en la actualidad optan por el uso de metodologías ágiles. Estas metodologías se dividen en dos categorías: aquellas que facilitan la coordinación y gestión integral del proyecto y aquellas que integran al personal con los procesos, la toma de decisiones y el tiempo necesario para el desarrollo de un proyecto.

Antes de aplicar las pautas para optar por el tipo de

metodología a utilizar para desarrollar un proyecto de software, los líderes preferían combinaciones de metodologías entre las que se destacaron: Scrum-Kanban, Kanban-XP, Scrum-XP, y la opción "otra". En esta última, la preferencia principal fue la combinación Design Sprint-Crystal. Pero, después de proporcionar los documentos con las pautas a líderes y desarrolladores y aplicar una segunda encuesta al iniciar nuevos proyectos en las empresas de la muestra, se observó una preferencia generalizada por el marco Scrum.

Es así como, al comparar los resultados de las primeras encuestas y entrevistas con los de las segundas, aplicadas después de que los participantes conocieran las pautas escritas para la selección de una metodología, se confirmó el orden de preferencias: Scrum, seguido de XP y Kanban. Además, se reafirmó la tendencia de los líderes a preferir combinaciones de metodologías.

Los resultados obtenidos demuestran que las metodologías ágiles son ampliamente utilizadas en la mayoría de las factorías de software debido a su capacidad para adaptarse a los cambios, la dinámica de los mercados, las constantes innovaciones en diversos campos y la evolución constante de la tecnología.

En síntesis, luego de trabajar con las empresas de la muestra, las pautas para escoger una metodología ágil para proyectos de software sirvieron entre otras, para:

- Optimizaran la elección de la metodología, pues esta les ayudó a la toma de decisiones, les proporcionó criterios claros y fundamentados sobre la selección de la metodología adaptada a las necesidades propias.
- Mejorar la eficiencia, en la gestión de los proyectos, pues con las directrices específicas y claras, les ayuda a agilizar los procesos reducir tiempos y optimizar recursos.
- Promover el éxito del proyecto, pues este se alinea con la metodología seleccionada, así como con los objetivos del proyecto, de esta forma, se mejora la calidad y la entrega en tiempo y forma.
- Facilita la comunicación entre los miembros del equipo del proyecto, permitiendo una comprensión compartida de la metodología a seguir.

Con los puntos expuestos anteriormente, se clarificaron los procesos en las empresas de la muestra, lo que ayudó para que las empresas desarrolladoras de software tuvieran una ayuda extra en la consecución de la metodología adecuada para el desarrollo de sus proyectos, pues las pautas les sirvieron como herramientas estratégicas para guiar a los gestores hacia la elección y aplicación efectiva de metodologías ágiles, contribuyendo al éxito y eficiencia en la ejecución de proyectos de software.

5 Conclusiones

Los objetivos planteados en el proyecto se han alcanzado de manera integral al lograr la construcción de pautas para la selección de metodologías ágiles en proyectos de software. Estas pautas se desarrollaron considerando las necesidades identificadas entre los desarrolladores y gestores de proyectos participantes, las cuales se demostraron a través de las técnicas

de recolección de información y sus respectivos instrumentos asociados.

Los desarrolladores de software expresaron una clara preferencia por las metodologías ágiles, fundamentada en diversas razones que convergían con los principios fundamentales de estas metodologías. Entre estos motivos, se destacó la flexibilidad y adaptabilidad que ofrecían, permitiendo ajustes ágiles a cambios en los requisitos del proyecto, lo cual fue altamente valorado por los desarrolladores en función de las necesidades cambiantes del cliente o del mercado.

La aplicación de las pautas propuestas evidenció beneficios significativos, como el fomento de una colaboración estrecha y una comunicación continua entre los miembros del equipo y los interesados. Asimismo, se observó la capacidad de realizar entregas incrementales en ciclos cortos, facilitando la presentación rápida del progreso y la recepción de retroalimentación temprana.

Un aspecto crucial que se destacó fue el enfoque hacia el cliente promovido por las metodologías ágiles, hecho que se materializó al aplicar las pautas construidas. La participación continua del cliente durante el desarrollo posibilitó ajustar el producto de manera eficaz para satisfacer sus necesidades y expectativas.

La mejora continua se consolidó como un logro destacado al emplear las metodologías ágiles junto con las pautas proporcionadas. La reflexión constante y la posibilidad de ajustar prácticas en cada iteración condujeron a una mejora gradual en la eficiencia y calidad del trabajo del equipo de desarrollo.

El mayor control y visibilidad obtenidos por los equipos de desarrollo, gracias a la implementación de las pautas, fortalecieron la gestión del trabajo. Además, las entregas incrementales, con su enfoque en la planificación a corto plazo y la retroalimentación frecuente, contribuyeron a mitigar riesgos al identificar y abordar problemas de manera temprana en el proceso de desarrollo.

Después de proporcionar los documentos con las pautas a líderes y desarrolladores y aplicar una segunda encuesta al iniciar nuevos proyectos en las empresas de la muestra, se observó una preferencia generalizada por el marco Scrum.

Al comparar los resultados de las primeras encuestas y entrevistas con los de las segundas, aplicadas después de que los participantes conocieran las pautas escritas para la selección de una metodología, se confirmó el orden de preferencias: Scrum, seguido de XP y Kanban. Además, se reafirmó la tendencia de los líderes a preferir combinaciones de metodologías.

La guía con las pautas para optar por una metodología ágil para proyectos de software se construyó a partir de las necesidades que se presentaron en las encuestas y las entrevistas con los gestores de proyectos, así como de la experiencia de los autores de este proyecto en la gestión de proyectos, en el desarrollo de software, en la construcción de varias metodologías para desarrollar software y por supuesto, en el análisis de documentos especializados.

Los resultados obtenidos demuestran que las metodologías

ágiles son ampliamente utilizadas en la mayoría de las factorías de software debido a su capacidad para adaptarse a los cambios, la dinámica de los mercados, las constantes innovaciones en diversos campos y la evolución constante de la tecnología.

Referencias

- [1] E. Bahit, "Scrum & Extreme Programming para programadores," Buenos Aires, Argentina.
- [2] T. Granollers, J. Lorés y J. Cañas, *Diseño de sistemas interactivos centrados en el usuario*, Ed. UOC, Barcelona, España, 2015, p. 98.
- [3] G. Pantaleo y L. Rinaudo, *Ingeniería de Software*, Alfaomega Colombiana S.A., Bogotá, Colombia, 2016.
- [4] M. Palacio, "Scrum Master. Temario troncal 1. Versión 3.0," Scrum Manager. Scrum Manager®, 2022. [En línea]. Disponible: https://www.scrummanager.com/files/scrum_master.pdf
- [5] R. McCarthy, *El Método Agile: Lo que Necesita Saber Sobre Gestión de Proyectos Agile, el Proceso Kanban, Pensamiento Lean, y Scrum*, Primasta, México, D.F., 2021.
- [6] K. Schwaber and J. Sutherland, "La Guía Scrum," nov. 2020. [En línea]. Disponible: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>
- [7] W. Clark, *Metodología Ágil: Una Guía Para Principiantes Sobre el Método y los principios ágiles*, Independently Published, Estados Unidos, 2020.
- [8] K. Beck, *Extreme Programming Explained: Embrace Change*, 2a ed., Addison Wesley, Estados Unidos de América, 2012.
- [9] H. Kniberg y M. Skarin, *Kanban y Scrum – obteniendo lo mejor de ambos*, Projectalis, Estados Unidos de América, 2010.
- [10] M. Skarin, *Real World Kanban*, Pragmatic Bookshelf, Estados Unidos, 2019.
- [11] S. Hernández, C. Fernández y P. Baptista, *Metodología de la investigación*, McGraw Hill, México, D.F., 2014.
- [12] R. Galarza, "Los alcances de una investigación," *CienciAmérica*, vol. 9, no. 3, 2020. DOI: <http://dx.doi.org/10.33210/ca.v9i3.336>
- [13] Team Asana, "¿Cuál es la diferencia entre Waterfall, Agile, Kanban y Scrum?," [En línea]. Disponible: <https://asana.com/es/resources/waterfall-agile-kanban-scrum>
- [14] L. N. Medina y W. M. López López, "Escoger una metodología para desarrollar software, difícil decisión," *Revista Educación En Ingeniería*, vol. 10, no. 20, pp. 98–109, 2015. [En línea]. Disponible: <https://educacioneningenieria.org/index.php/edi/article/view/579>
- [15] Ministerio del Comercio, "Decreto 957. Criterio de clasificación de las micro, pequeñas y grandes empresas," 2019. [En línea]. Disponible: <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=94550>
- [16] Fedesoft, 2019. [En línea]. Disponible: <https://fedesoft.org/>
- [17] Project Management Institute-PMI, "Pulse of the Profession Beyond Agility," *Pulse of the Profession®*, 2021. [En línea]. Disponible: <https://tinyurl.com/mrdzvje9>
- [18] E. Bahit, "Scrum & Extreme Programming para programadores," Buenos Aires, Argentina. [En línea]. Disponible: <https://www.atlassian.com/>
- [19] Davis, *Software Requirements: Analysis and Specification*, Prentice Hall, México, D.F., 1989.
- [20] R. Maccarthy, *Agile y Scrum: Descubra el Poder de la Gestión de Proyectos Agile*, Primasta, México, D.F., 2021.
- [21] D. Pérez y J. Sepúlveda, *Extreme Programming (XP)*, Academia Española, Madrid, España, 2019.
- [22] Garzás, J., "Metodologías ágiles: Los artículos más destacados". [En línea]. Disponible: <https://www.javiergarzas.com/metodologias-agiles-los-articulos-mas-destacados>.
- [23] F. Flores-Cerna, V. M. Sanhueza-Salazar, H. M. Valdés-González, y L. Reyes-Bozo, "Metodologías ágiles: un análisis de los desafíos organizacionales para su implementación," *Revista Científica*, vol. 43, no. 1, pp. 38-49, 2022. <https://doi.org/10.14483/23448350.18332>
- [24] M. A. Fonseca Vargas, E. I. Obregón Hernández, y L. D. Espinoza Jaen, "Metodologías ágiles de desarrollo de software. La programación extrema aplicada al desarrollo del sistema de información web para la gestión de presupuestos en CEM Comunicaciones S.A.," Tesis de maestría, Dept. Computación, Univ. Nacional Autónoma de Nicaragua,

Managua, Nicaragua, 2024. [En línea]. Disponible:

<https://repositorio.unan.edu.ni/6197/1/15911.pdf>.

- [25] N. T. Quesada Reyes, "Estudio sobre Metodologías Ágiles en los Proyectos Software. Propuesta de Plan de Implantación para PYMES," Proyecto Fin de Máster, Univ. de Sevilla, Sevilla, España, 2020. [En línea]. Disponible: <https://biblus.us.es/bibing/proyectos/abreproy/71934/fichero/TFM-1934+QUESADA+REYES%2C+NORA.pdf>.

L. N. Medina-Velandia es Ingeniera de Sistemas y Magister en Ingeniería de Sistemas de la Universidad Nacional de Colombia, Diploma de Estudios Avanzados del doctorado en Ingeniería de Software e Inteligencia Artificial en la Universidad de Salamanca en España, título de Especialista en Pedagogía de la Universidad Pedagógica Nacional en 2014, doctorando en Proyectos TIC (actual), Doctorado en Ciencias de la Educación (actual) de la Universidad de Cuahtémoc (México). Profesora de tiempo completo desde 1994 en diferentes universidades. Investigadora reconocida por el Ministerio de Ciencia Tecnología e Innovación en donde sus intereses investigativos incluyen:

inteligencia artificial, ingeniería de software y la influencia de las TIC en la Educación.

<http://orcid.org/0000-0002-2871-4991>

D. A. Gutiérrez Medina es Ingeniero de Sistemas de la Universidad Politécnica Gancolombiano. Empresario y Director de LPA de Chile, con más de 17 años de experiencia en implementación de proyectos relacionados con Sistemas de Gestión Documental, gestión de procesos y Case Management. Tiene las siguientes certificaciones internacionales: FCP P8 Certified: CM Administrator – FileNet, FCP P8 Certified: BPM Administrator – FileNet, FCP Content Manager-Administrator – FileNet, FCP Forms Manager-Administrator – FileNet, FCP Content Manager-Administrator – FileNet, FCP Forms Manager-Administrator – FileNet, FCP Content Manager-Administrator – FileNet, FCP Business Process Manager-Administrator – FileNet, FCP Records Manager-Administrator – FileNet, IBM Certified Specialist FileNet Content Manager – IBM, IBM Certified Specialist FileNet Business Process Manager – IBM, IBM Certified Specialist FileNet Content Manager – IBM, IBM Enterprise Content Management Sales Profesional.