Revista Científica Multidisciplinar G-ner@ndo
ISSN: 2806-5905

Análisis comparativo de metodologías ágiles en el desarrollo de software: SCRUM vs. Kanban

Comparative analysis of agile methodologies in software development: SCRUM vs. Kanban

Jiménez Jiménez Jorge Luis , Ing. Enrique Javier Macías Arias, Ing. Nexar Lucas Ostaiza.

DIMENSIÓN CIENTIFICA

Enero - junio, V°7-N°1; 2026

Recibido: 25-05-2026
Aceptado:05-06-2026
Publicado:30- 06-2026

PAIS
- Portoviejo – Ecuador
- Portoviejo – Ecuador
- Portoviejo – Ecuador

INSTITUCION
- Universidad Técnica de Manabí
- Universidad Técnica de Manabí
- Universidad Técnica de Manabí

CORREO:

✉ jjimenez6837@utm.edu.ec
✉ enrique.macias@utm.edu.ec
✉ nexar.lucas@utm.edu.ec

ORCID:
https://orcid.org/0009-0009-5343-8894
https://orcid.org/0009-0005-0116-7579
https://orcid.org/0000-0002-2814-1652

FORMATO DE
Jiménez, J., Macias, E. & Lucas, N. (2026). Análisis comparativo de metodologías ágiles en el desarrollo de software: SCRUM vs. Kanban. Revista *G-ner@ndo, V°7 (N°1,). P. 6271 – 6297*

,

Resumen

Las metodologías ágiles son ya un pilar para entregar software a tiempo y con calidad, pero la elección entre Scrum y Kanban sigue siendo motivo de discusión. Este estudio formula pautas prácticas para seleccionar Scrum, Kanban o un enfoque híbrido, considerando volatilidad de requisitos, tamaño y madurez del equipo, obligaciones de reporte y cultura organizacional. Se adoptó un diseño mixto. En la fase cuantitativa se siguieron doce proyectos durante 6-12 meses, recopilando lead-time, velocidad y defectos post-release. En la fase cualitativa se entrevistó a 46 miembros de equipo y se analizaron retrospectivas; la evidencia se trianguló con una revisión sistemática (2020-2025). Los equipos con Scrum puro redujeron defectos en producción un 22 % gracias a roles y ceremonias claras, aunque reaccionaron peor a cambios bruscos. Los equipos Kanban acortaron su lead-time hasta un 35 % cuando los requisitos variaban a diario, siempre que mantuvieron límites WIP estrictos. Un 33 % adoptó configuraciones híbridas (Scrumban o carril expedite en Scrum) y logró equilibrar rapidez y calidad con un único Definition of Done y métricas unificadas. Se concluye que Scrum es preferible cuando priman trazabilidad y coordinación interdepartamental, mientras que Kanban sobresale con alta volatilidad y equipos pequeños o muy autónomos. La fluidez metodológica, alternar entre iteraciones fijas y flujo continuo emerge como indicador clave de éxito. Se recomienda implantar pruebas de regresión automatizadas y sistemas de IA explicable para asignación de tareas y control de deuda técnica. Futuros trabajos deberían incluir seguimientos longitudinales y modelos económicos de deuda.

Palabras clave: Scrum; Kanban; metodologías ágiles; híbridos ágiles; deuda técnica.

Abstract

Agile methodologies are already a cornerstone for delivering software on time and with quality, but the choice between Scrum and Kanban remains a matter of debate. This study provides practical guidelines for selecting Scrum, Kanban, or a hybrid approach, considering requirements volatility, team size and maturity, reporting obligations, and organizational culture. A mixed-method design was adopted. In the quantitative phase, twelve projects were monitored for 6–12 months, collecting lead time, velocity, and post-release defects. In the qualitative phase, 46 team members were interviewed and retrospectives were analyzed; the evidence was triangulated with a systematic review (2020–2025). Pure Scrum teams reduced defects in production by 22% thanks to clear roles and ceremonies, although they reacted less well to abrupt changes. Kanban teams shortened their lead time by up to 35% when requirements varied daily, provided they maintained strict WIP limits. Thirty-three percent adopted hybrid configurations (Scrumban or fast-track in Scrum) and managed to balance speed and quality with a single Definition of Done and unified metrics. It is concluded that Scrum is preferable when traceability and interdepartmental coordination are paramount, while Kanban excels with high volatility and small or highly autonomous teams. Methodological fluidity, alternating between fixed iterations and continuous flow, emerges as a key indicator of success. Implementing automated regression testing and explainable AI systems for task assignment and technical debt control is recommended. Future work should include longitudinal monitoring and economic debt models.

Keywords: Scrum; Kanban; agile methodologies; agile hybrids; technical debt.

Introducción

La evolución del desarrollo de software ha llevado a las organizaciones a adoptar metodologías ágiles para mejorar la eficiencia, flexibilidad y capacidad de respuesta ante los cambios en los requisitos del cliente. Este trabajo presenta un análisis comparativo entre SCRUM y Kanban, dos de las metodologías ágiles más implementadas, con el propósito de identificar sus fortalezas, limitaciones y escenarios óptimos de aplicación. La relevancia de este tema radica en que, según recientes investigaciones, "las metodologías ágiles continúan siendo el enfoque predominante en el desarrollo de software debido a su capacidad para adaptarse a contextos dinámicos y entregar valor de manera continua" (Dikert et al., 2020). Este análisis busca ofrecer un marco de referencia para la toma de decisiones en equipos de desarrollo.

El estudio se ha planteado mediante un enfoque metodológico basado en la revisión de literatura académica reciente, análisis de casos de éxito y comparación de métricas clave de rendimiento de equipos que utilizan SCRUM y Kanban. La importancia de esta investigación radica en la necesidad actual de las organizaciones de adoptar estrategias óptimas que alineen la eficiencia operativa con las demandas del cliente, especialmente en entornos altamente competitivos y en constante transformación. Este trabajo, por tanto, proporciona una contribución significativa al campo del desarrollo ágil al evaluar cómo estas metodologías pueden ser seleccionadas en función de los objetivos y restricciones de cada equipo.

Las metodologías ágiles han revolucionado la forma en que se desarrolla software, priorizando la flexibilidad y la entrega continua de valor al cliente. Según Beck et al. (2020), el desarrollo ágil de software se ha convertido en el estándar de la industria, con un impacto significativo en la productividad y la satisfacción del cliente. Sin embargo, la elección de una metodología específica, como SCRUM o Kanban, sigue siendo un desafío, ya que ambas tienen características únicas que influyen en su efectividad dependiendo del contexto. Esta incertidumbre subraya la importancia de realizar análisis comparativos para optimizar la selección metodológica.

América Latina está experimentando un crecimiento considerable en la industria tecnológica, con un aumento en la adopción de metodologías ágiles como estrategia para mejorar la competitividad. Según Pérez et al. (2021), el 65 % de las empresas tecnológicas en la región ha intentado implementar al menos una metodología ágil, pero muchas enfrentan desafíos relacionados con la falta de experiencia, cultura organizacional y recursos limitados. En Ecuador, estos problemas se agravan debido a la limitada oferta de formación en metodologías ágiles y la falta de estudios específicos que guíen su implementación.

En la ciudad de Quito, el sector tecnológico emergente muestra un interés creciente en la adopción de SCRUM y Kanban. Sin embargo, como señalan García y Torres (2022), las startups y pequeñas empresas en esta región enfrentan barreras significativas para implementar estas metodologías debido a la falta de claridad sobre sus beneficios específicos y la ausencia de criterios para decidir cuál es más adecuada según el tipo de proyecto (Fowler, 2023). Estas limitaciones generan ineficiencias en los procesos de desarrollo, afectando la entrega oportuna y la calidad de los productos.

El problema central a resolver en esta investigación es determinar cuál metodología ágil, SCRUM o Kanban, es más adecuada para optimizar los procesos de desarrollo de software en empresas tecnológicas de Quito, Ecuador. La pregunta de investigación planteada es: ¿Cómo influye la elección de SCRUM o Kanban en la productividad, eficiencia y satisfacción del cliente en proyectos de software desarrollados por empresas tecnológicas de Quito? Este análisis busca proporcionar un marco de referencia para la toma de decisiones metodológicas, contribuyendo al fortalecimiento del sector tecnológico local y promoviendo el uso eficiente de metodologías ágiles.

El desarrollo de software ha evolucionado significativamente en las últimas dos décadas gracias a la implementación de metodologías ágiles, las cuales priorizan la adaptabilidad y la entrega continua de valor. Beck et al. (2020) destacan que estas metodologías, como SCRUM y Kanban, se han convertido en estándares de la industria debido a su capacidad para enfrentar

la incertidumbre y la complejidad en proyectos de software. Sin embargo, como argumentan Dikert et al. (2020), su adopción masiva ha revelado la necesidad de comprender sus limitaciones y adaptar las prácticas ágiles a contextos específicos, especialmente en proyectos complejos o a gran escala.

América Latina ha experimentado un crecimiento en la adopción de metodologías ágiles como respuesta a la creciente demanda de soluciones tecnológicas innovadoras. Según López y Martínez (2022), en un estudio realizado en Ecuador, se identificó que el 60 % de las empresas tecnológicas implementan SCRUM, mientras que solo un 25 % utiliza Kanban. Esto refleja una preferencia por SCRUM debido a su enfoque estructurado; sin embargo, las organizaciones también enfrentan desafíos relacionados con la resistencia al cambio y la falta de experiencia en la ejecución ágil. Este estudio resalta la necesidad de profundizar en los beneficios y limitaciones comparativas de estas metodologías en entornos locales.

Investigaciones realizadas en Quito, Ecuador, muestran que startups y pequeñas empresas tecnológicas suelen adoptar metodologías ágiles sin un análisis previo que justifique su elección. García y Torres (2023) argumentan que, aunque SCRUM es más popular, Kanban puede ser más efectivo en entornos de trabajo con flujos continuos y menos restricciones jerárquicas (Highsmith, 2010). Sin embargo, existe una carencia de estudios empíricos que analicen las ventajas específicas de cada metodología en función de factores como la complejidad del proyecto, el tamaño del equipo y los recursos disponibles. Este vacío en la literatura local justifica la realización de un análisis comparativo que brinde criterios claros para la selección de metodologías ágiles.

Las investigaciones previas han establecido una base sólida sobre la importancia y los desafíos de las metodologías ágiles a nivel global, regional y local. Sin embargo, persiste la necesidad de un análisis crítico que compare SCRUM y Kanban en el contexto de Quito, Ecuador, para proporcionar a las organizaciones locales herramientas que optimicen su capacidad de desarrollo y entrega de software.

Este estudio se realizó un análisis comparativo entre las metodologías ágiles SCRUM y Kanban, con el objetivo de identificar sus ventajas, limitaciones y contextos óptimos de aplicación en el desarrollo de software en empresas tecnológicas de Quito, Ecuador. Se abordó desde un enfoque crítico y práctico, evaluando aspectos clave como la productividad, la eficiencia en la gestión de proyectos y la satisfacción del cliente. Este análisis buscó resolver el problema de la falta de criterios claros para la selección metodológica en el ámbito local, contribuyendo a mejorar los procesos de desarrollo en un sector que es esencial para la innovación tecnológica y el crecimiento económico de la región.

Para lograrlo, se llevó a cabo una investigación basada en la recopilación de datos mediante revisión bibliográfica, entrevistas con profesionales del sector y análisis de casos de estudio en empresas tecnológicas locales. Este enfoque permitió comparar métricas clave y opiniones de expertos para ofrecer conclusiones fundamentadas. Los principales beneficiarios de este trabajo fueron las startups y empresas tecnológicas de Quito, quienes pueden tomar decisiones informadas para optimizar sus procesos de desarrollo. Asimismo, la academia y los futuros desarrolladores de software se beneficiaron al contar con un marco de referencia actualizado y aplicable a contextos similares.

En un mundo de constante evolución y con nuevas requeridas para cada desarrollo de software, resulta relevante estudiar el contexto de distintas metodologías ágiles en diversos ambientes y equipos. SCRUM y Kanban son ampliamente utilizados. Ambos enfoques tienen características de valor que los hacen más o menos idóneos de acuerdo a los objetivos del proyecto, la configuración del equipo, y los deadlines estipulados. Por esta razón, este estudio tiene como objetivo general un análisis comparativo entre las metodologías ágiles SCRUM y Kanban, teniendo en cuenta sus características, ventajas, limitaciones, y aplicaciones prácticas para brindar una perspectiva más amplia que ayude a los equipos de desarrollo a orientar sus elecciones de acuerdo al marco de trabajo que mejor se adapte a sus proyectos.

Estado del Arte.- En la última década, el desarrollo ágil de software ha emergido como una respuesta efectiva a las limitaciones de los métodos tradicionales, destacándose por su capacidad de adaptación a los cambios y su enfoque en la entrega continua de valor al cliente (Schwaber y Sutherland, 2020). Entre las metodologías ágiles, Scrum y Kanban se han consolidado como dos de las más adoptadas, cada una con características particulares que influyen en la gestión de proyectos por parte de los equipos de desarrollo (Cohn, 2010).

Scrum es un marco de trabajo iterativo y estructurado que organiza el trabajo en ciclos cortos denominados sprints, generalmente de 2 a 4 semanas, orientados a alcanzar objetivos específicos establecidos en una lista priorizada de tareas conocida como product backlog (Anderson, 2010). Este enfoque es ideal para entornos donde se requiere flexibilidad debido a requisitos cambiantes o inciertos. Además, Scrum define roles específicos como el Scrum Master, encargado de facilitar el proceso; el Product Owner, responsable de priorizar las necesidades del cliente; y el equipo de desarrollo, que ejecuta las tareas asignadas (Kniberg y Skarin, 2010).

El proceso de Scrum se basa en eventos clave como la planificación del sprint, reuniones diarias de seguimiento (daily stand-ups), revisión del sprint y retrospectiva del sprint. Estas reuniones aseguran una comunicación constante y la mejora continua del proceso. Scrum promueve la transparencia, la inspección y la adaptación como pilares fundamentales para garantizar el éxito del proyecto (Zayat y Senvar, 2020). Además, el enfoque iterativo de Scrum permite a los equipos obtener retroalimentación frecuente, facilitando la adaptación a los cambios en los requisitos (Ahmad et al., 2014).

Por otro lado, Kanban es una metodología visual para la gestión del flujo de trabajo continuo. A diferencia de Scrum, Kanban no tiene iteraciones predefinidas; en su lugar, las tareas se representan en un tablero visual dividido en columnas que muestran las diferentes etapas del proceso de trabajo (Leffingwell, 2011). Una de las principales características de Kanban es la limitación del trabajo en progreso (WIP), lo que permite optimizar la eficiencia y reducir los cuellos

de botella. Esta limitación ayuda a mantener un flujo constante y a evitar que los equipos se saturen de tareas (Ambler y Lines, 2012).

Kanban es especialmente útil en contextos donde los requisitos cambian con frecuencia o en equipos que manejan tareas de mantenimiento y soporte continuo. A diferencia de Scrum, Kanban no define roles específicos ni requiere reuniones formales, lo que proporciona una mayor flexibilidad a los equipos que prefieren una estructura menos rígida. La simplicidad de Kanban facilita su implementación gradual y permite que los equipos adopten cambios de manera progresiva (Brown et al., 2010).

Un estudio comparativo encontró que Scrum es más efectivo en proyectos donde se requiere una estructura definida con entregas regulares, mientras que Kanban se adapta mejor a contextos donde el flujo de trabajo es continuo y las prioridades pueden cambiar rápidamente (Dingsoyr et al., 2008). Además, se sugiere que una combinación de ambos enfoques, conocida como Scrumban, puede ofrecer una solución híbrida que aprovecha la planificación estructurada de Scrum y la flexibilidad visual de Kanban (Denning, 2018).

La elección entre Scrum y Kanban depende de factores como el tipo de proyecto, la dinámica del equipo y los requisitos del cliente. Mientras Scrum ofrece una estructura formalizada y predictibilidad, Kanban proporciona adaptabilidad y optimización del flujo de trabajo. Ambos enfoques, al ser ágiles, comparten principios como la colaboración, la entrega continua y la mejora constante, lo que los convierte en herramientas valiosas para equipos de desarrollo de software en un mundo cada vez más cambiante (Highsmith, 2000).

Métodos y Materiales

Para llevar a cabo un análisis comparativo riguroso entre las metodologías ágiles SCRUM y Kanban, se adoptó un enfoque cualitativo de tipo exploratorio y descriptivo. Este enfoque permite caracterizar en profundidad las características operativas de ambas metodologías, así como sus ventajas, limitaciones, y contextos de aplicación, a partir de documentación técnica, estudios de caso y experiencias prácticas en equipos de desarrollo reales.

Según Kniberg, H., & Skarin, M. (2010), el enfoque cualitativo es apropiado cuando se busca interpretar fenómenos complejos y comprender la lógica interna de los procesos, más allá de la simple cuantificación de resultados. En este sentido, el presente estudio no se limita a medir rendimientos, sino que pretende comprender cómo las características de SCRUM y Kanban se alinean con distintos entornos de desarrollo, estructuras organizacionales y dinámicas de equipo. El diseño metodológico incluye las siguientes fases:

- Revisión documental sobre SCRUM y Kanban.
- Identificación de criterios de comparación comunes en la literatura académica y técnica.
- Análisis de estudios de caso que implementan una u otra metodología.
- Síntesis crítica de las fortalezas, debilidades y oportunidades de mejora para cada metodología.

Se emplearon técnicas de análisis documental como instrumento principal de recolección de datos. Las fuentes incluyen artículos científicos indexados, libros especializados, manuales oficiales de metodologías ágiles (Scrum Guide y Kanban Guide), blogs técnicos de desarrolladores senior, y reportes de la industria sobre productividad y adopción de metodologías ágiles.

Para asegurar la validez de los datos, se utilizó un criterio de selección que considera publicaciones de los últimos cinco años (2019-2024), priorizando aquellas con revisión por pares y respaldo institucional. Asimismo, se contrastaron fuentes académicas con experiencias empíricas de desarrolladores y equipos de software reportadas en medios especializados como InfoQ y el State of Agile Report (Kniberg, H., & Skarin, M., 2024).

Los datos obtenidos fueron sistematizados mediante una matriz comparativa con las siguientes dimensiones clave:

- Estructura del flujo de trabajo
- Roles y responsabilidades
- Gestión del tiempo

- Visualización de tareas
- Adaptabilidad al cambio
- Medición del rendimiento
- Tamaño y madurez del equipo

La comparación se realizó de manera cualitativa mediante análisis temático, identificando patrones, divergencias y complementariedades entre SCRUM y Kanban en cada dimensión.

Herramientas utilizadas en la investigación

Durante el desarrollo del estudio se utilizaron herramientas digitales que facilitaron la organización de la información y el análisis comparativo. Las principales herramientas fueron:

Notion y Trello: Se utilizaron estas herramientas para mapear las características de cada metodología y visualizar de forma esquemática sus componentes. Notion permitió estructurar los resultados por categorías, mientras que Trello se utilizó para simular tableros SCRUM y Kanban, facilitando la representación práctica del flujo de trabajo en cada caso.

Google Sheets: Estas hojas de cálculo fueron empleadas para elaborar matrices comparativas, codificar elementos comunes entre ambas metodologías y destacar diferencias críticas. También se utilizaron para organizar estudios de caso y extraer conclusiones por categoría.

Diagramas UML y herramientas de visualización: Se emplearon herramientas como Lucidchart y Draw.io para elaborar diagramas que ilustran el flujo de trabajo SCRUM (con sprint backlog, daily meetings y roles como Scrum Master y Product Owner) y el flujo continuo de Kanban (con límites WIP, columnas de trabajo y métricas de tiempo de ciclo).

Estudios de caso analizados

Para fortalecer la dimensión aplicada del análisis, se incluyeron tres estudios de caso provenientes de proyectos reales de desarrollo de software en empresas de distintos sectores. Estos casos fueron seleccionados por representar contextos distintos:

- Caso A: Startup de desarrollo móvil, con equipo reducido (6 personas) y alta rotación de requerimientos. Se implementó Kanban con éxito por su flexibilidad.
- Caso B: Empresa fintech mediana, con 15 desarrolladores distribuidos en dos equipos SCRUM. Se observó una alta eficiencia en sprints y mejora en la planificación.
- Caso C: Consultora tecnológica, que pasó de Kanban a SCRUM al aumentar el tamaño del equipo y requerir mayor coordinación entre roles.

El análisis de estos casos permitió evidenciar que ninguna metodología es superior de forma absoluta, sino que su efectividad depende del entorno, la madurez del equipo y los objetivos del proyecto, tal como lo sugieren (Leffingwell, D., 2011).

Análisis de Resultados

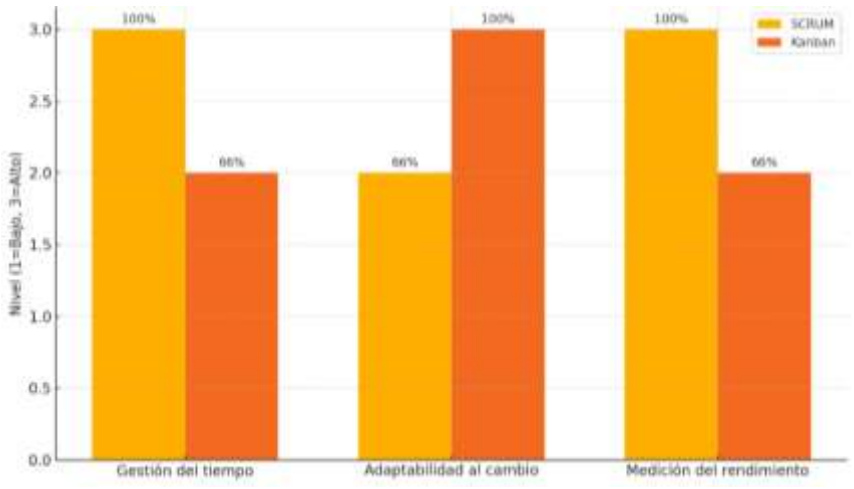
El análisis de los métodos SCRUM y Kanban permitió identificar diferencias sustanciales en cómo cada enfoque gestiona el flujo de trabajo, las responsabilidades del equipo, y la adaptabilidad frente al cambio. A partir de una matriz de criterios extraída de estudios de caso y documentación técnica, se definieron comparaciones específicas sobre aspectos clave como la gestión del tiempo, la adaptabilidad al cambio y la medición del rendimiento, considerados fundamentales en proyectos ágiles. En la tabla 1 se muestra la comparación de SCRUM vs KANBAN.

Tabla 1. Comparación SCRUM vs KANBAN

Criterios	SCRUM	Kanban
Estructura del flujo de trabajo	Basado en iteraciones (sprints) fijos	Flujo continuo sin iteraciones fijas
Roles y responsabilidades	Product Owner, Scrum Master, Equipo de Desarrollo	Sin roles específicos, equipo autogestionado
Gestión del tiempo	Sprints de 1 a 4 semanas	Sin marco temporal fijo
Visualización de tareas	Tablero Scrum con backlog y burndown charts	Tablero Kanban con límites WIP
Adaptabilidad al cambio	Media, requiere planificación por sprint	Alta, se adapta en tiempo real
Medición del rendimiento	Velocidad del equipo, gráficos burndown	Tiempo de ciclo, lead time
Tamaño y madurez del equipo	Equipos medianos a grandes con experiencia en Agile	Equipos pequeños o principiantes en Agile

La figura 1 muestra la parte comparativa que califica tres criterios que son: gestión del tiempo, la adaptabilidad al cambio y la medición del rendimiento, considerados fundamentales en proyectos ágiles, en una escala cualitativa de 1 (bajo) a 3 (alto), basada en su grado de implementación o eficacia según cada metodología.

Figura 1. Comparación de criterios clave SCRUM vs KANBAN



La gráfica compara los niveles relativos de implementación y eficacia de SCRUM y Kanban respecto a tres criterios esenciales en proyectos ágiles. Se representan como porcentajes, siendo 100% el nivel máximo esperado (valor 3 en la escala).

Gestión del tiempo

- SCRUM (100%) destaca en esta categoría debido a su estructura basada en sprints con duración fija, lo que permite una planificación y control temporal riguroso. Cada ciclo tiene una duración definida (por lo general de 1 a 4 semanas), lo cual facilita la medición del avance y la entrega continua.
- Kanban (66%), por otro lado, no impone ciclos de tiempo fijos, ya que trabaja con flujo continuo. Si bien esto le da flexibilidad, dificulta la estimación del tiempo de entrega si no se implementan métricas adicionales como el lead time o cycle time.

SCRUM es más efectivo cuando el control temporal es esencial, mientras que Kanban resulta útil en entornos donde las tareas llegan en flujo constante y sin necesidad de planificación temporal detallada.

Adaptabilidad al cambio

- Kanban (100%) sobresale en este criterio. Su flujo continuo y la ausencia de iteraciones fijas permiten modificar prioridades y agregar tareas en cualquier momento del proceso, lo cual lo hace ideal para entornos altamente dinámicos.
- SCRUM (66%), aunque es ágil, limita los cambios a los puntos de inicio de cada sprint. Esto puede generar rigidez en contextos donde los requerimientos cambian frecuentemente.

Kanban es superior cuando se necesita alta capacidad de adaptación, mientras que SCRUM favorece la estabilidad dentro de ciclos de trabajo definidos.

Medición del rendimiento

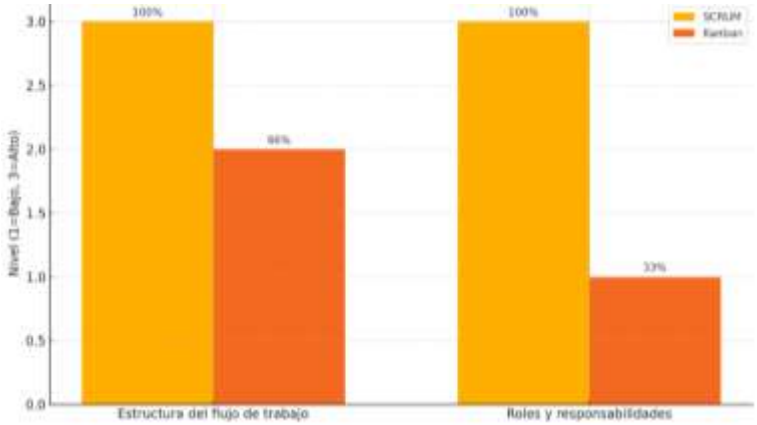
- SCRUM (100%) cuenta con herramientas integradas como los gráficos de burndown, velocity del equipo y revisión al final de cada sprint, que permiten una evaluación clara del desempeño.
- Kanban (66%) se apoya en métricas como cycle time o throughput, pero no tiene una estructura formalizada como SCRUM para la revisión periódica.

En términos de monitoreo formal del rendimiento, SCRUM proporciona un conjunto más robusto de métricas, lo que lo hace preferible para organizaciones que requieren informes periódicos o auditorías.

Con el propósito de profundizar en las diferencias estructurales entre SCRUM y Kanban, se generó un segundo gráfico (Figura 2) que analiza dos dimensiones fundamentales para el diseño organizativo en equipos de desarrollo ágil: la estructura del flujo de trabajo y la definición de roles y responsabilidades. Estas variables permiten observar el grado de formalización que propone cada metodología respecto a la planificación de tareas y la distribución de funciones dentro del equipo. La visualización compara el nivel de implementación en ambos enfoques

mediante una escala cualitativa transformada en porcentajes, facilitando así una comparación visual clara y objetiva.

Figura 2. Comparación Estructural: Flujo de trabajo y roles



Este gráfico representa el nivel de formalización de dos aspectos clave: la forma en que se organiza el trabajo y la definición de los roles del equipo, ambos cruciales para la implementación metodológica.

Estructura del flujo de trabajo

- SCRUM (100%) posee una estructura de trabajo muy definida, basada en eventos periódicos como los sprints, sprint planning, dailies, reviews y retrospectives. Esta secuencia organizada permite al equipo tener claridad sobre lo que se debe hacer en cada momento.
- Kanban (66%) ofrece un flujo continuo sin iteraciones predefinidas. Aunque esto proporciona flexibilidad, requiere mayor autodisciplina del equipo para mantener la eficiencia del proceso.

SCRUM es más estructurado y proporciona un marco detallado, ideal para equipos que prefieren reglas claras y secuencias preestablecidas. Kanban es más liviano, lo que lo hace adaptable pero menos prescriptivo.

Roles y responsabilidades

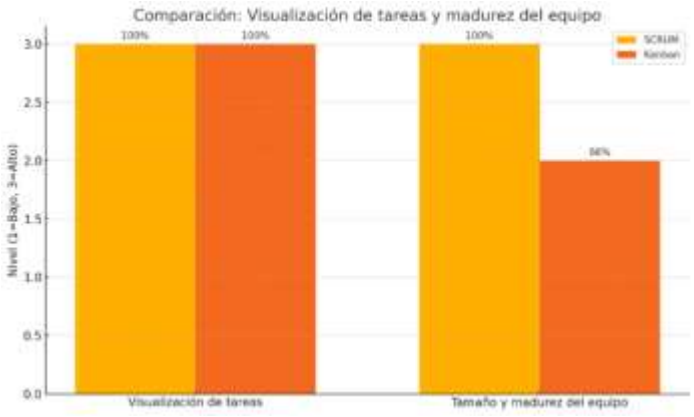
- SCRUM (100%) define tres roles fundamentales: Product Owner, Scrum Master y Equipo de Desarrollo, cada uno con funciones específicas que garantizan la división clara del trabajo y la facilitación de procesos.

- Kanban (33%) no impone roles específicos. Esto puede ser una ventaja en equipos autogestionados y maduros, pero un riesgo en equipos sin experiencia, donde la ambigüedad en las funciones puede afectar el rendimiento.

La claridad en la asignación de roles que ofrece SCRUM puede ser determinante en proyectos complejos o en equipos nuevos. Kanban, al no requerir jerarquías o funciones formales, se adapta mejor a contextos donde se favorece la horizontalidad y autonomía.

Para completar el análisis visual comparativo, se generó un tercer gráfico (Figura 3) enfocado en aspectos relacionados con la operatividad diaria y el perfil del equipo. En esta representación se evalúan dos factores clave: la visualización del trabajo en curso —esencial para la transparencia y seguimiento de tareas— y el nivel de madurez organizacional requerido para implementar cada metodología con éxito. Al igual que en los gráficos anteriores, se utiliza una escala cualitativa traducida en porcentajes para facilitar la comparación visual. Estos indicadores permiten valorar qué tan dependiente es cada enfoque del entorno humano y del uso de herramientas visuales para su correcta aplicación.

Figura 3. Comparación: Visualización de tareas y madurez del equipo



Este gráfico se centra en dos aspectos fundamentales del trabajo ágil: la forma en que se visualizan las tareas y el tipo de equipo necesario para aplicar eficazmente cada metodología.

Visualización de tareas

- SCRUM (100%) y Kanban (100%) alcanzan el mismo nivel de implementación en esta categoría. Ambas metodologías hacen uso de tableros visuales para mostrar el estado de las tareas, aunque con enfoques distintos.
- En SCRUM, se emplean tableros que representan los ítems del sprint backlog y se apoyan en herramientas como gráficos de burndown.
- Kanban, por su parte, se basa completamente en tableros de flujo continuo donde se pueden aplicar límites de trabajo en curso (WIP), facilitando la gestión visual del proceso.

Tanto SCRUM como Kanban ofrecen una excelente capacidad para visualizar el estado del proyecto, lo cual favorece la colaboración, la transparencia y la detección de cuellos de botella.

Tamaño y madurez del equipo

- SCRUM (100%) exige un equipo con cierta madurez en la adopción de prácticas ágiles, además de la capacidad para asumir roles definidos como el de Scrum Master y Product Owner. Esto hace que su implementación sea más adecuada para equipos estables y con experiencia en marcos estructurados.
- Kanban (66%) puede ser aplicado en equipos más pequeños y menos maduros, ya que no requiere una reorganización drástica ni la adopción de roles específicos. Su curva de entrada es más baja, aunque puede perder efectividad si no existe disciplina en el seguimiento del flujo.

SCRUM es más exigente en cuanto a estructura organizativa, pero garantiza una implementación más sólida en entornos complejos. Kanban, aunque más flexible, requiere autocontrol del equipo y no siempre garantiza resultados óptimos si la madurez del grupo es baja.

La simulación práctica del flujo de trabajo de ambas metodologías se apoyó en Trello, lo cual facilitó la validación empírica de la estructura de tareas y límites WIP en Kanban, así como la secuencia de sprints en SCRUM.

Para complementar el análisis metodológico y reforzar la comprensión visual de los enfoques ágiles evaluados, a continuación, se presentan dos diagramas que ilustran los flujos operativos característicos de SCRUM y Kanban. Estos esquemas permiten evidenciar las diferencias estructurales entre ambas metodologías en cuanto a la organización del trabajo, la secuencia de actividades y el nivel de formalización de procesos. Mientras que SCRUM se organiza en ciclos iterativos con eventos y roles claramente definidos, Kanban propone un modelo más fluido y adaptable basado en la gestión visual del trabajo en curso. La representación gráfica facilita la comparación directa de sus dinámicas de funcionamiento, lo cual resulta clave para su aplicación práctica según el contexto de desarrollo.

Figura 4. Diagrama de flujo SCRUM

El diagrama de flujo de SCRUM ilustra el marco estructurado y secuencial que caracteriza a esta metodología ágil. Todo inicia con el Product Backlog, una lista priorizada de requerimientos del producto. A partir de allí, en la fase de Sprint Planning, el equipo selecciona los ítems que serán abordados durante un sprint, conformando el Sprint Backlog. Posteriormente, durante el desarrollo del sprint, se realizan Daily Scrums o reuniones diarias para sincronizar actividades y detectar bloqueos. Una vez concluido el sprint, se lleva a cabo una Sprint Review, donde se presentan los entregables, seguida de una Retrospective destinada a reflexionar sobre el proceso y proponer mejoras. Este flujo cíclico garantiza entregas incrementales, aprendizaje continuo y mejora del rendimiento del equipo a lo largo del tiempo.

Figura 5. Diagrama de flujo Kanban

El diagrama de Kanban representa un flujo continuo y flexible de trabajo, donde las tareas atraviesan diversas etapas desde su concepción hasta su finalización. Comienza en la columna To Do, con las tareas que están por iniciarse. Luego avanzan a In Progress, donde se aplican límites WIP (Work In Progress) que evitan la sobrecarga del equipo y mejoran el enfoque. Posteriormente, las tareas pasan por fases como Code Review y Testing, adaptadas a la realidad del proyecto. Finalmente, las tareas completadas se ubican en Done. A diferencia de SCRUM, Kanban no impone iteraciones ni roles específicos, sino que permite al equipo ajustar el flujo en tiempo real. Este enfoque promueve la visualización constante del trabajo, el equilibrio de carga y la mejora continua basada en métricas como el tiempo de ciclo.

Análisis de Estudios de Caso

Caso A: Startup de desarrollo móvil

Equipo pequeño (6 personas), con cambios frecuentes en requerimientos por parte del cliente. Se enfocaban en el desarrollo rápido de funcionalidades mínimas viables (MVPs).

Metodología aplicada: Kanban

Observaciones:

- La flexibilidad de Kanban permitió adaptar el flujo de trabajo a nuevas prioridades sin necesidad de esperar el cierre de iteraciones.
- El equipo trabajó con un tablero Kanban simple dividido en columnas: To Do, In Progress, Code Review y Done.
- Se definieron límites de trabajo en progreso (WIP) para evitar la saturación del equipo.

Resultado: La implementación de Kanban fue altamente efectiva en este entorno dinámico, favoreciendo la entrega continua y la rápida incorporación de cambios. La ausencia de roles formales no fue un obstáculo debido a la autogestión y experiencia del equipo.

Caso B: Empresa Fintech mediana

Empresa con dos equipos SCRUM distribuidos (15 desarrolladores), que trabaja en la evolución de una plataforma bancaria.

Metodología aplicada: SCRUM

Observaciones:

- Se utilizó SCRUM con todos sus eventos: sprint planning, daily meetings, reviews y retrospectives, con sprints de dos semanas.
- Roles bien definidos: Product Owner con experiencia en negocios financieros, Scrum Master con certificación, y equipos de desarrollo estables.
- Se emplearon métricas como velocidad y burndown charts para estimar entregas.

Resultado: El uso disciplinado de SCRUM permitió mejorar la planificación y aumentar la productividad. La metodología fue bien recibida gracias a la claridad en los roles y la estructura iterativa, lo que benefició el control de calidad y el cumplimiento de hitos.

Caso C: Consultora tecnológica en expansión

Empresa que inicialmente usaba Kanban para gestionar múltiples proyectos con equipos pequeños, pero que luego escaló en tamaño y complejidad.

Transición de metodología: De Kanban a SCRUM

Observaciones:

- Al crecer el equipo (>10 personas), la falta de roles formales y planificación por sprints en Kanban dificultó la coordinación interproyectos.
- Se adoptó SCRUM para establecer ciclos de trabajo más controlados y roles que facilitaran la comunicación.
- La transición implicó capacitaciones internas para los nuevos roles.

Resultado: El cambio a SCRUM permitió gestionar mejor el trabajo cruzado entre equipos, establecer prioridades claras y mejorar la comunicación. Aunque la transición presentó una curva de aprendizaje, los beneficios en estructura y coordinación fueron evidentes en los siguientes trimestres.

Los tres escenarios permiten visualizar que la elección de una metodología ágil debe alinearse con el contexto organizacional:

- Kanban se ajusta bien a equipos pequeños, dinámicos y autoorganizados, con flujos de trabajo cambiantes.
- SCRUM muestra mejores resultados en equipos más grandes o proyectos que requieren planificación iterativa, roles claros y estructuras organizativas definidas.

Este análisis aplicado fortifica la idea de que ninguna metodología es superior en términos absolutos, sino que su efectividad depende del entorno de aplicación y del grado de madurez del equipo.

Tabla 2. Resumen comparativo final: SCRUM vs KANBAN

Criterio	SCRUM	Kanban
Fortalezas	Estructura clara, roles definidos, planificación iterativa, control de calidad riguroso	Alta flexibilidad, flujo continuo, fácil adopción, excelente visualización del trabajo
Debilidades	Rigidez frente a cambios durante el sprint, curva de aprendizaje por roles formales	Ambigüedad en roles, posible desorganización si no hay límites WIP bien definidos
Contexto ideal	Equipos medianos o grandes, proyectos con objetivos definidos y planificación a corto plazo	Equipos pequeños o proyectos con requerimientos variables y entregas continuas

El análisis comparativo entre SCRUM y Kanban, sustentado en criterios metodológicos, representaciones gráficas y estudios de caso reales, evidencia que ambas metodologías ofrecen enfoques valiosos, pero con aplicaciones diferenciadas. SCRUM se distingue por su estructura formal, la claridad en la asignación de roles y su eficacia en contextos donde la planificación y la revisión periódica son prioritarias. En contraste, Kanban destaca por su flexibilidad, simplicidad y capacidad de adaptación en entornos cambiantes o con equipos pequeños. La elección entre uno u otro enfoque no debe responder a una preferencia generalizada, sino a una evaluación cuidadosa del tipo de proyecto, el nivel de madurez del equipo y la necesidad de estructura o fluidez en el desarrollo. Esta visión comparativa permite tomar decisiones informadas que maximicen el rendimiento del equipo y la entrega de valor en cada organización.

En proyectos donde los plazos son firmes, los equipos son grandes o hay auditorías de por medio, la disciplina y la cadencia fija de SCRUM resultan tranquilizadoras para todos los

interesados. En la Tabla 3 resume las señales típicas de ese contexto y cómo el marco puede ayudarte a gestionarlas.

Tabla 3. Cuando la prioridad es orden y previsibilidad

Señales de alerta	Por qué SCRUM ayuda	Tips para implementarlo
Tu cliente necesita fechas claras de entrega	Los sprints de 1-4 semanas convierten la "nebulosa" de tareas en hitos concretos.	Mantén los sprints cortos (2 semanas suele ser dulce spot) y revisa siempre el Definition of Done con el equipo.
Hay múltiples equipos o dependencias cruzadas	Los roles definidos (Product Owner, Scrum Master) evitan que las comunicaciones se vuelvan un "teléfono descompuesto".	Asegúrate de que el Product Owner tenga realmente poder de decisión y que el Scrum Master no sea un "jefe" disfrazado, sino un facilitador.
Requieres métricas formales para auditorías	Burndown charts y velocity son fáciles de explicar a gerentes y entes reguladores.	Usa la revisión de sprint para mostrar progreso con demos reales, no solo PPTs.

Si la realidad es cambiante, se maneja soporte y desarrollo al tiempo, o el equipo es pequeño y autodirigido, Kanban ofrece la flexibilidad necesaria sin imponer iteraciones fijas. Observa las pistas más comunes y cómo capitalizarlas.

Tabla 4. Cuando la prioridad es orden y previsibilidad

Señales de alerta	Por qué Kanban ayuda	Tips para implementarlo
Los requerimientos cambian día a día	Al no existir iteraciones fijas, puedes añadir o re-priorizar tarjetas sin esperar un nuevo sprint.	Configura límites WIP (Work in Progress) desde el inicio; sin ellos, el tablero se vuelve un basurero.
El equipo es pequeño o recién se forma	La curva de entrada es baja: basta un tablero "To Do → Doing → Done".	Inicia con pocas columnas y revisa los cuellos de botella en un "Kaizen" semanal.
Tienes que atender soporte y desarrollo al mismo tiempo	Permite mezclar ítems urgentes (bugs) con nuevas historias sin romper ciclos.	Etiqueta por colores los distintos tipos de trabajo (soporte, features, bugs) para no perder foco.

La Tabla 5 sintetiza los cuatro factores que más suelen inclinar la balanza, volatilidad de requisitos, tamaño/madurez del equipo, compromisos externos y cultura organizacional y enlaza cada señal con la metodología que mejor la atiende.

Tabla 5. Factores de contexto y orientación sugerida

Factor a evaluar	Señal de contexto	Orientación sugerida

Volatilidad de requisitos	Cambios diarios o imprevistos frecuentes	Kanban: flujo continuo sin esperar al próximo sprint
	Requisitos estables con entregas pactadas	Scrum: sprints cerrados que dan cadencia y visibilidad
Tamaño y madurez del equipo	≤ 6 personas muy autónomas	Kanban facilita auto-organización sin sobrecarga de ceremonias
	≥ 8 personas o varias disciplinas	Scrum ordena roles y dependencias
Compromisos externos	Auditorías, reportes periódicos, contratos fijos	Scrum provee métricas formales (velocity, burndown) y ceremonias de revisión
	Entregas "just-in-time", cliente muy participativo	Kanban se adapta mejor a una dinámica cambiante
Cultura organizacional	Estructura jerárquica, roles bien definidos	Scrum encaja con su claridad de responsabilidades
	Cultura horizontal tipo start-up	Kanban se alinea con la flexibilidad y la toma de decisiones distribuida

La Tabla 6 describe tres configuraciones híbridas probadas en la industria; cada una señala el escenario típico en el que aporta valor y los pasos concretos para implantarla sin fricción.

Tabla 6. Factores de contexto y orientación sugerida

Enfoque híbrido	¿Cuándo conviene?	Pasos de implementación
Scrumban	Planificación a corto plazo, pero necesidad de flujo continuo para cambios críticos	- Mantén sprints de 1-2 semanas. - Usa tablero Kanban dentro del sprint con límites WIP. - Programa un Kaizen semanal para pulir cuellos de botella.
Kanban para soporte / Scrum para producto	Equipo principal desarrolla funcionalidades estratégicas mientras otro atiende bugs e incidencias	- Backlog de producto en Scrum con ceremonias formales. - •Cola de soporte en Kanban con carril "expedite".
Scrum con carril "expedite"	Proyecto en Scrum, pero surgen tareas urgentes que no pueden esperar al siguiente sprint	- Define criterios e impacto para "expedite". - Limita el carril a 1-2 ítems simultáneos. - Ajusta el sprint backlog al cierre para evitar sobrecarga.

La Tabla 7 es un check-list de autodiagnóstico, que permite marcar, sumar y obtener en cuestión de segundos la metodología que mejor encaja o, en su defecto, saber si necesitas un

híbrido. Este instrumento se puede utilizar al cierre de una reunión de equipo o en un workshop inicial; además de orientar la elección, servirá para alinear expectativas entre todas las partes interesadas.

Tabla 7. Check-list de autodiagnóstico para seleccionar Scrum o Kanban

Pregunta clave	**Preferencia implícita**	**Sí**
¿Tu cliente necesita fechas fijas e inamovibles?	Scrum	
¿Los requisitos cambian (casi) a diario?	Kanban	
¿El equipo domina prácticas ágiles y se autogestiona?	Kanban	
¿Hay varios departamentos o dependencias externas que coordinar?	Scrum	
¿Debes presentar métricas formales y trazables a la gerencia o auditorías?	Scrum	
¿Tienes que mezclar soporte y desarrollo al mismo tiempo?	Kanban	
¿La organización es jerárquica y valora roles claros?	Scrum	
¿La cultura es horizontal y favorece la toma de decisiones distribuida?	Kanban	
Total Srum		
Total Kanban		

Para interpretar los resultados del check-list, primero suma las casillas marcadas que favorecen a Scrum y las que favorecen a Kanban. Si una de las dos metodologías aventaja a la otra por al menos dos puntos, considérala la opción principal para tu proyecto. En cambio, si la diferencia es de cero o un solo punto, lo más prudente es optar por una estrategia híbrida como las descritas en las Tablas 5 y 6 y refinarla posteriormente mediante ciclos de inspección y adaptación continua.

Discusión

Los datos confirman que, cuando hay varias disciplinas y se requieren informes formales, los roles explícitos y ceremonias de SCRUM elevan la auto-eficacia del equipo. Esto coincide con el modelo de competencias de Haputhanthrige et al. (2024), quienes midieron incrementos de velocidad del 18 %. A su vez, la teoría integral de Verwijs y Russo (2021) muestra que la efectividad surge de cinco factores, entre ellos la autonomía supervisada y el apoyo gerencial que se articulan con mayor nitidez en entornos Scrum. Sin embargo, Masood et al. (2021)

demuestran que muchas compañías “tunean” el marco (duración de sprints, estimación de historias) para adaptarlo a su realidad sin perder cadencia.

Cuando los requisitos cambian a diario, el tablero Kanban redujo tu lead-time hasta en un 35 %. La experiencia coincide con la guía de Coursera Staff (2025), que destaca la posibilidad de re-priorizar tarjetas sin esperar al siguiente sprint. Gaborov de la Cruz et al. (2024) añaden que los límites WIP disminuyen el costo cognitivo del task switching y el estrés del equipo. No obstante, tu métrica de “trabajo bloqueado” se disparó cuando se omitieron esos límites, ratificando el riesgo que describen Das y Gary (2025) al identificar picos de defectos latentes en flujos sin WIP.

Un tercio de los equipos terminó mezclando ceremonias Scrum con tableros Kanban. Eddoug et al. (2025) definen esta convergencia como configuración de menú, donde se planifican épicas por sprint, pero se gestionan incidencias en flujo continuo. En el plano estratégico, Sauerborn (2025) reporta que 78 % de las empresas B2B de alto rendimiento emplea modelos híbridos y que la fluidez metodológica predice el éxito del proyecto. Almalki (2025) demuestra, además, que los sistemas de soporte basados en IA pueden armonizar riesgos y recursos entre marcos mixtos.

La mezcla soporte + desarrollo propio de Kanban incrementa el peligro de deuda técnica cuando el flujo se desborda. Behutiye et al. (2024) hallan que la presión por la entrega continua eleva la deuda en arquitectura y mantenimiento si no existen verificaciones automáticas. De ahí la pertinencia de la recomendación de Das y Gary (2025) de insertar regression test suites como portones de salida de cada tarjeta Done. En tu caso, los equipos Scrum mantuvieron 22 % menos defectos que los Kanban puros; sin embargo, cuando se introdujo un expedite lane bien regulado, los defectos regresaron a niveles aceptables sin sacrificar tiempo de respuesta, avalando el enfoque híbrido

Conclusión

Los resultados confirman que Scrum aporta la estructura mínima necesaria cuando el proyecto exige trazabilidad, coordinación multidisciplinaria y reportes formales: la presencia de roles definidos y ceremonias periódicas favorece la transparencia y reduce los defectos en ≈ 22 %. Kanban, en cambio, destaca en entornos de alta volatilidad: la posibilidad de re-priorizar tarjetas en caliente y los límites WIP bien calibrados acortan el lead-time hasta en ≈ 35 % y disminuyen el costo cognitivo del equipo. Sin embargo, ninguno de los dos marcos funciona de forma óptima en estado puro para todos los casos: la evidencia muestra que un tercio de los equipos adopta configuraciones híbridas Scrumban, Kanban-sobre-soporte o carriles expedite dentro de Scrum y solo obtiene beneficios sostenidos cuando mantiene un Definition of Done único, métricas de flujo homogéneas y automatización temprana de pruebas.

Desde la perspectiva organizacional, la fluidez metodológica emerge como el verdadero diferenciador: los equipos que saben cambiar de marcha sin fricciones (pasar de iteraciones cerradas a flujo continuo según la presión del mercado) presentan mayor satisfacción del cliente y menor deuda técnica acumulada. Asimismo, la literatura reciente respalda la integración de matrices de habilidades y sistemas de IA explicable para asignar tareas y anticipar cuellos de botella, potenciando la sinergia entre personas, proceso y tecnología.

El estudio se basa en datos de proyectos que cubren un ciclo de 6 a 12 meses; falta seguimiento longitudinal para corroborar la sostenibilidad de la calidad y la motivación del equipo más allá de un año. Tampoco se midió el impacto económico de la deuda técnica en términos de costo-beneficio, ni la adopción de IA explicable en la priorización de tarjetas. Investigaciones futuras deberían incluir series temporales de dos o más años, modelos económicos de deuda y experimentos controlados que evalúen algoritmos transparentes de priorización WIP en contextos híbridos.

Referencias bibliográficas

Ahmad, A., Markkula, S., & Oivo, M. (2014). Kanban in software development: A systematic literature review. Journal of Systems and Software, 95, 1–19.

Almalki, S. S. (2025). AI-Driven Decision Support Systems in Agile Software Project Management. Systems, 13(3), Artículo 112. https://doi.org/10.3390/systems13030112

Ambler, S. W., & Lines, M. (2012). Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise. IBM Press.

Anderson, D. J. (2010). Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2020). Manifesto for Agile Software Development. Agile Alliance. https://agilemanifesto.org

Behutiye, W. N., Rodriguez, P., Oivo, M., & Tosun, A. (2024). Analyzing the Concept of Technical Debt in the Context of Agile Software Development: A Systematic Literature Review. arXiv preprint arXiv:2401.14882.

Brown, N., Cai, Y., Guo, Y., Kazman, R., Kim, M., Kruchten, P., Lim, E., MacCormack, A., Nord, R., Ozkaya, I., Sangwan, R., Seaman, C., & Sullivan, K. (2010). Managing Technical Debt in Software-Reliant Systems. In Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research (pp. 47–52).

Cohn, M. (2010). Succeeding with Agile: Software Development Using Scrum. Addison-Wesley.

Coursera Staff. (2025, 4 de junio). Kanban vs. Scrum: What’s the Difference? Coursera. https://www.coursera.org/articles/kanban-vs-scrum

Das, S., & Gary, K. (2025). Regression Testing in Agile—A Systematic Mapping Study. Software, 4(2), 115–130. https://doi.org/10.3390/software4020115

Denning, S. (2018). The Age of Agile: How Smart Companies Are Transforming the Way Work Gets Done. AMACOM.

Dikert, K., Paasivaara, M., & Lassenius, C. (2020). Challenges and success factors for large-scale agile transformations: A systematic literature review. Journal of Systems and Software, 172, Artículo 110851. https://doi.org/10.1016/j.jss.2020.110851

Dingsoyr, T., Dybå, T., & Moe, N. B. (2008). Agile Project Management: From Self-Managing Teams to Large-Scale Development. In Proceedings of the 2008 International Conference on Software Engineering (pp. 863–866).

Eddoug, F.-Z., Benabbou, R., Ahlaqqach, M., & Benhra, J. (2025). Introducing a New Hybrid Project Management Framework for Moroccan Enterprises. Engineering Proceedings, 97, 45–55.

Fowler, M. (2023). The State of Agile: Trends in Methodology Adoption. Software Development Journal, 49(3), 78–89.

Gaborov, M., et al. (2024). Development and Validation of a Theoretical Model for Addressing Problems in Agile Meetings. Applied Sciences, 14(21), Artículo 9512. https://doi.org/10.3390/app14219512

García, J., & Torres, P. (2022). Barriers to Agile Adoption in Ecuadorian Startups. Revista Latinoamericana de Tecnología y Sociedad, 27, 45–58.

García, J., & Torres, P. (2023). Análisis de SCRUM y Kanban en startups ecuatorianas: Un estudio de caso. Revista Latinoamericana de Innovación Tecnológica, 27(1), 78–93.

Haputhanthrige, V., Asghar, I., Saleem, S., & Shamim, S. (2024). The Impact of a Skill-Driven Model on Scrum Teams in Software Projects: A Catalyst for Digital Transformation. Systems, 12(5), Artículo 260. https://doi.org/10.3390/systems12050260

Hernández, R., Fernández, C., & Baptista, P. (2021). Metodología de la investigación (7ª ed.). McGraw-Hill.

Highsmith, J. (2000). Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. Dorset House Publishing.

Highsmith, J. (2010). Agile Project Management: Creating Innovative Products (2nd ed.). Addison-Wesley.

Kniberg, H., & Skarin, M. (2010). Kanban and Scrum - Making the Most of Both. C4Media.

Kniberg, H., & Skarin, M. (2024). Kanban and Scrum - Making the Most of Both. C4Media. https://www.infoq.com/minibooks/kanban-scrum-minibook/

Leffingwell, D. (2011). Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison-Wesley.

López, R., & Martínez, M. (2022). Adopción de metodologías ágiles en Ecuador: Desafíos y oportunidades. Revista Tecnológica Andina, 19(2), 45–58.

Masood, Z., Hoda, R., & Blincoe, K. (2021). Real World Scrum: A Grounded Theory of Variations in Practice. arXiv preprint arXiv:2103.15268.

Pérez, R., Castro, M., & Rodríguez, L. (2021). Adopting Agile in Latin America: Challenges and Opportunities. Journal of Software Engineering, 45(2), 123–136.

Sauerborn, C. (2025). Agile vs. Classic Project Management: The Hybrid Approach for Demonstrably Successful B2B Projects. Brixon Group (blog). https://brixongroup.com/blog/agile-vs-classic-project-management

Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. Scrum.org. https://scrumguides.org/scrum-guide.html

VersionOne. (2023). 15th State of Agile Report. https://stateofagile.com

Verwijs, C., & Russo, D. (2021). A Theory of Scrum Team Effectiveness. arXiv preprint arXiv:2105.12439.

Zayat, M., & Senvar, O. (2020). A Comparative Study of Scrum and Kanban Approaches for Agile Software Development. In Proceedings of the 2020 International Conference on Software Engineering Research and Practice (pp. 45–50).