



<http://jates.org>

Revista de aplicaciones  
Ciencias Técnicas y de la Educación

jates

ISSN 2560-5429



## Análisis comparativo de metodologías ágiles y tradicionales en la gestión de proyectos TI

Maja Gaborova Dijana Karuović<sup>a</sup>, Mila Kavalić<sup>b</sup>, Dragica Radosavč<sup>c</sup>, Dragana Milosavljević<sup>d</sup>, Sanja Stanisavljević<sup>e</sup>, Jozef Bushatiđ<sup>f</sup>

<sup>a</sup>Universidad de Novi Sad, Facultad técnica "Mihajlo Pupin" Zrenjanin, Serbia, [maja.gaborov@tfzr.rs](mailto:maja.gaborov@tfzr.rs)

<sup>b</sup>Universidad de Novi Sad, Facultad técnica "Mihajlo Pupin" Zrenjanin, Serbia, [dijanakaruoovic@gmail.com](mailto:dijanakaruoovic@gmail.com)

<sup>c</sup>Universidad de Novi Sad, Facultad de Ciencias Técnicas, Novi Sad, Serbia, [milakavalic@gmail.com](mailto:milakavalic@gmail.com)  
(Correspondiente)

<sup>d</sup>Universidad de Novi Sad, Facultad técnica "Mihajlo Pupin" Zrenjanin, Serbia, [aronradosav@gmail.com](mailto:aronradosav@gmail.com)

<sup>e</sup>Universidad de Novi Sad, Facultad técnica "Mihajlo Pupin" Zrenjanin, Serbia, [dragana.milosavljev@tfzr.rs](mailto:dragana.milosavljev@tfzr.rs)

<sup>f</sup>Universidad de Novi Sad, Facultad técnica "Mihajlo Pupin" Zrenjanin, Serbia, [sanja84stanisavljev@tfzr.rs](mailto:sanja84stanisavljev@tfzr.rs)

<sup>g</sup>Universidad de Shkodra "Luigj Gurakuqi", Albania, [jozefbushati@gmail.com](mailto:jozefbushati@gmail.com)

### Abstracto

Se sabe que inicialmente, durante el desarrollo de software, la metodología tradicional se utilizaba para todos los proyectos de TI y que estos a menudo fracasaban debido al rápido crecimiento de la industria de TI. Fue entonces cuando comenzaron a desarrollarse las metodologías ágiles. Los métodos tradicionales tienen algunas ventajas sobre las metodologías ágiles, y principalmente la metodología tradicional más común es la cascada. Dadas sus limitaciones en el manejo de problemas, como código no estructurado, moral del equipo, poca visibilidad, falta de comunicación entre las partes interesadas y la frecuente priorización de los requisitos del usuario, no sería malo utilizar metodologías ágiles, que se centran en el trabajo con los usuarios, las pruebas continuas, la refactorización y el desarrollo incremental. Este documento aborda la comparación de metodologías en la gestión de proyectos de TI con base en otras investigaciones científicas. Algunas de las metodologías mencionadas son Scrum, Kanban, cascada, etc. Se concluyó que la metodología ágil Scrum se utiliza principalmente en empresas de TI, y que a menudo surge la combinación de varias metodologías, debido a la necesidad de los proyectos, para eliminar las deficiencias de cada metodología.

Palabras clave: ágil; tradicional; proyecto TI; gestión; Scrum; cascada; Kanban;

## 1. Introducción

Para que los proyectos de ingeniería de software tengan éxito, es necesaria una buena organización.

Los proyectos son más complejos, por lo que es necesario conocer algún proceso formal de gestión de proyectos.

Es importante gestionar el proyecto para utilizar los recursos de manera eficiente y tener un control uniforme.

división del trabajo, para formar un plan que se implemente a tiempo, con una forma predecible de

ejecución y resultados buenos y esperados.

Los proyectos se utilizan en diversas industrias, por ejemplo en construcción, tecnología, materiales y

La gestión de proyectos es importante en la industria de la tecnología de la información, donde las nuevas...

Los requisitos de los usuarios cambian muy rápidamente. Para que los proyectos tengan éxito y se gestionen correctamente

De la manera correcta, lo más importante es cuidar que los proyectos sean lo más eficientes posible.

posible, que se implementen en un periodo de tiempo determinado con los recursos disponibles y

Dentro de un presupuesto determinado. Las metodologías en la gestión de proyectos de TI pueden ser tradicionales y ágiles.

y la elección de algunos de ellos depende de las características y naturaleza de cada proyecto en particular.

(Vresk, 2020). Los métodos de gestión de proyectos ágiles se han utilizado durante varios años, principalmente en

La profesión de TI. Al decidir qué método de gestión de proyectos utilizar, las necesidades de...

las partes interesadas, los riesgos asociados al proyecto, el tamaño del proyecto, los costos y, por supuesto,

Por supuesto, hay que tener en cuenta la complejidad del proyecto (Vladimirovich Orlov, et al.,

2021). Este artículo trata sobre la comparación de enfoques tradicionales y ágiles basados en

literatura científica y otros estudios y también llega a la conclusión de qué metodología es

más utilizado en la práctica.

## 2. Investigación de metodología

### 2.1 Objeto y problema de la investigación

Dado que las metodologías son importantes en la gestión de proyectos, es muy importante decidir sobre la

metodología apropiada y por tanto el tema de la investigación es considerar metodologías en

Gestión de proyectos de TI y selección de la opción más adecuada. El problema de esta investigación es...

Elección de la metodología que mejor se adapte a la naturaleza de un proyecto en particular. Hay muchas

Razones a tener en cuenta al elegir, como la duración del proyecto, los requisitos del cliente, el presupuesto,

etc. Los gerentes deben conocer las metodologías y conocer los beneficios de cada metodología.

La elección de la metodología garantiza el éxito del proyecto. Una revisión sistemática de la literatura...

Puede ayudar a los directores de proyectos a tomar esta decisión.

### 2.2 Preguntas de investigación y objetivo de la investigación

El objetivo de esta investigación es determinar qué metodología es la más aplicada en la práctica, qué

son las razones y qué combinaciones de metodologías existentes se utilizan. Se puede presentar

mediante las siguientes preguntas de investigación:

- RQ1: ¿Cuáles son las razones para elegir metodologías ágiles y tradicionales?
- RQ2: ¿Qué metodología es la más utilizada?

- RQ3: Si se utilizan combinaciones de múltiples metodologías, ¿cuáles son aquellas?

Combinaciones y ¿cuáles son las razones de ello?

- RQ4: ¿Cuáles son las diferencias entre Scrum, XP y Kanban?

### 3. Consideración teórica

"La gestión de proyectos es la aplicación de herramientas, habilidades, técnicas y conocimientos en el proyecto. actividades para cumplir los objetivos del proyecto o los requisitos del cliente y su alcance predeterminado" (Vresk, 2020) Existen varias metodologías de gestión de proyectos y se presentarán algunas de ellas. en esta sección.

#### 3.1 Metodología tradicional

Los métodos tradicionales se aplican en organizaciones en las que se gestionan proyectos completamente definidos. Se implementan y se preparan planes con antelación que tienen como objetivo cumplir con los objetivos de tiempo, presupuesto y proyecto. (Vresk, 2020; Shenhar y Dvir, 2007). La gestión en sí se basa en órdenes y controles, La comunicación es formal, el modelo de desarrollo se basa en un modelo de ciclo de vida. Es una cuestión de planificación completa, y el examen de los resultados viene al final (Vresk, 2020; Engelhardt, 2019). En el método tradicional, los requisitos del usuario están claramente definidos en el comienzo del proyecto y los clientes establecen sus requisitos desde el comienzo del proyecto. Los equipos son grandes. Los proyectos toman mucho tiempo (Vresk, 2020).

#### 3.2 Método de cascada

El modelo de cascada fue introducido por Royce en 1970 (Cocco, et al., 2011). Es el más Método común de los métodos tradicionales. Las fases son las siguientes: especificación de requisitos, diseño, implementación/development, pruebas y mantenimiento (McCormick, 2012). El El equipo de TI del proyecto dedica mucho tiempo a la fase de especificación de requisitos, es decir, a la planificación y diseño, y cuando comienza la fase de implementación/development, no hay requisitos adicionales Requisitos, dudas o partes indefinidas. Proyecto o resultado final. Por esta razón, a largo plazo Los proyectos son adecuados para el acceso a la cascada (Vresk, 2020; McCormick, 2012; Bhavsar, et al., 2020; Andrei, et al., 2019).

#### 3.3 Príncipe2

Prince2 es un método tradicional llamado método "impulsado por procesos" que responde a preguntas tipo "qué". y el «por qué», y en pequeñas cantidades a la pregunta del «cómo». PRINCE2, desarrollado por el Reino Unido.

gobierno y más recientemente son los derechos a la metodología en una empresa público-privada (Vresk, 2020).

### 3.4 Método de la Ruta Crítica (CPM)

"El método de la ruta crítica (CPM) se basa en un modelo lógico y matemático del proyecto, el cuya base es el tiempo óptimo requerido para un proceso individual en el proyecto y la explotación simultánea de los recursos más económicos disponibles" (Antill y Woodhead, 1991). Su mayor ventaja es determinar la secuencia que no tiene línea de tiempo. reserva (ruta crítica) y en reconocer aquellas secuencias de actividades que las tienen (Vresk, 2020).

### 3.5 Metodología ágil

Los métodos de gestión ágil de proyectos de software son guías para la planificación y el control de los mismos. (Parker & del Monte, 2014). Algunas de las características de los proyectos de TI han condicionado la Necesidad de crear nuevas metodologías para trabajar eficientemente en programación y software. desarrollo. Los enfoques tradicionales no siempre han sido apropiados, principalmente porque estos Son proyectos que suelen comenzar sin especificaciones firmes e inmutables debido a frecuentes Por lo general, se requieren cambios, lo que también requiere un enfoque diferente y más flexible. El cliente a menudo no está seguro de qué espera exactamente como resultado y, a través de solicitudes de cambio, va hacia su objetivo. El cliente a veces quiere que el proceso vuelva a la etapa anterior. y ciertos cambios por realizar. El equipo que trabaja en el desarrollo de software debe estar listo para cambios constantes y una estrecha colaboración con el cliente (Islam, 2013). Cooperación en equipo, Buena comunicación entre los miembros del equipo, así como buena comunicación con el cliente. son importantes para armonizar los deseos de los clientes y las posibilidades de lograrlos. resultados (Ahmad, et al., 2016). Las metodologías ágiles de desarrollo de software surgieron a finales de siglo XX, donde el trabajo en equipo es clave (Ahmad, et al., 2016; Milošević, 2018; Islam, 2013). El término "ágil" significa la capacidad de un método para responder a cambios frecuentes en los requisitos. Como el cliente tiene una idea del prototipo, puede definir los requisitos y explicarlos. Al equipo cuáles son sus verdaderos requisitos. Con el método ágil, el énfasis está en los cambios. que puede ser constante y así aumentar el éxito del proyecto. Es mejor hacer cambios. Periódicamente según sea necesario porque es más barato que hacer cambios cuando el proyecto ya está en marcha. completado (Ahmad, et al., 2016; Milošević, 2018; Islam, 2013; Vresk, 2020). en ágil metodologías, el proyecto puede avanzar a partir de una idea y al final dar como resultado algo diferente de la idea original (Vresk, 2020; Islam, 2013). Agile/Scrum y Extreme

La programación (XP) se utiliza ampliamente en las empresas para lograr el desarrollo de software. proyectos (Barash, 2013; García, et al., 2020; Venkatachalam, et al., 2017)

### 3.5.1 Scrum

Scrum es un proceso iterativo-incremental y Scrum es el método ágil más común (Mahnič, 2015; Schwaber, 2004; Cocco, et al., 2011), enfoque (Stellman & Greene, 2017; Lei, et al., 2017; Granulo y Tanović, 2019; Ingason, et al., 2013). El proceso contiene un conjunto de medidas de gestión. recomendaciones, pero no define las actividades del proceso de desarrollo en sí. A menudo es utilizado en combinación con otros procesos de desarrollo de software. Scrum mide el resultado de El sistema futuro después de cada iteración.

Los roles en la metodología Scrum son propietario del producto, equipo de desarrollo, scrum master, gerente, cliente (Pantelić, et al., 2020; Schwaber, & Beedle 2001; Laanti, 2013; Albarqi & Qureshi, 2018; Fustik, 2017). El propietario del producto (Terlecka, 2012; Venkatachalam, et al., 2017), Recopila información de clientes, usuarios finales y miembros del equipo de desarrollo y luego la convierte. Entiende los requisitos y los evalúa en términos de prioridades de desarrollo. Es responsable de Desarrollo y entrega de productos según los requisitos del cliente. El equipo de desarrollo... Generalmente se compone de cinco a diez miembros, algunos de los cuales son analistas, desarrolladores, diseñadores y probadores. El equipo tiene autonomía en la toma de decisiones, así como la libertad de proporcionar la Propietario del producto con ideas para mejorar el producto. El Scrum Master debe establecer mediaciones entre los propietarios de productos y los miembros del equipo de desarrollo. Responsable de el desarrollo exitoso del producto final y es responsable del éxito Implementación del método Scrum en el proyecto, brindando asistencia continua y Apoyo a los miembros del equipo de desarrollo. El gerente es responsable de la decisión final. y participa en el proceso de establecimiento de objetivos y definición de requisitos. El cliente participa en el proceso de generar solicitudes y definir la funcionalidad que debe tener el futuro sistema Tiene y participa en el proceso de verificación de los resultados y funcionalidades obtenidos, proporcionando retroalimentación al equipo de desarrollo.

El trabajo de desarrollo de software se lleva a cabo en ciclos más cortos llamados sprints. Después de eso, es Es necesario consultar continuamente con el cliente y, después de un cierto ciclo, realizar un análisis y Se realiza una revisión, así como cualquier cambio deseado y necesario. Las reuniones son obligatorias. antes y después de cada sprint para determinar si todo se ha hecho como lo solicita el cliente y para determinar si es necesario cambiar algo. El sprint puede durar hasta un mes (Ferrão y Canedo, 2015; Brezočnik & Majer, 2016), y el resultado es un producto ejecutivo (Pantelić, et al., 2020;

Schwaber y Sutherland, 2020; Fustik, 2017). Al planificar un sprint, se especifica qué necesidades que debe realizarse dentro del sprint (Bhavsar, et al., 2020), es decir, cuál será el resultado del producto y cómo se logrará ese resultado. Es necesario realizar reuniones diarias y pueden durar aproximadamente Quince minutos. Cada miembro del equipo está obligado a entregar un informe diario de mañana. reunión sobre los problemas que notó, sobre el trabajo realizado el día anterior y sobre los resultados que obtuvo. planes a lograr (Pantelić, et al., 2020; Abrahamsson, et al., 2002). El término Backlog es una parte indispensable de Scrum. Existen el Product Backlog y el Sprint Backlog. El Product El backlog es una lista de todas las características del resultado final que el equipo necesita desarrollar. Para que el resultado sea satisfactorio, el Backlog del Sprint es un conjunto de todas las características proporcionadas por El propietario del producto junto con el equipo que decidió desarrollar en el Sprint que sigue. El Las características mismas se toman del backlog del producto, del cual se extraen las funciones. asumido por la prioridad (Vresk, 2020; Stellman y Greene, 2017). Después de un par de años, la El segundo autor (gerente de operaciones de I+D en SI) y el tercer autor (CTO) consideraron que Scrum era demasiado rígido, no escalable y no adecuado para tareas de mantenimiento (Sjøberg, et al., 2012).

### 3.5.2 Kanban

Kanban entró en el desarrollo de software en 2004. Kanban es el segundo método ágil más famoso. (Moonden, 2011). Kanban es un método basado en el término «justo a tiempo» (Ahmad, et al., 2013). La idea de este método surgió de tres ingenieros de Toyota (Ahmad, et al., 2018; Andrei, et al., 2019), Kichiro Toyota, Eiji Toyoda y Taiichi Ohno en la década de 1940 (Womack, et al., 1990; Stellman & Greene, 2017; Moonden, 2011). "Kanban tiene cinco principios básicos: visualizar flujo de trabajo, limitar el flujo de trabajo, medir y gestionar el flujo de trabajo, hacer explícitas las políticas de proceso y utilizar modelos para identificar mejoras y oportunidades" (Ahmad, et al., 2016). Sjøberg, Johnsen y Solberg (2012) compararon el uso de Scrum y Kanban en una empresa de tamaño mediano Durante un período de dos años, sus hallazgos muestran que con Kanban, el número de errores ponderados... se redujo en un 10% y un 11% respectivamente, y la productividad mejoró en un 21%. Funciones implementadas. Kanban se centra en la comunicación, la colaboración y la integración. entre desarrolladores de software, probadores y equipos de soporte, lo que da como resultado una rápida implementación del software. desarrollo y entrega continua al cliente. Kanban en el desarrollo de software aporta visibilidad del trabajo y mejora la eficiencia y el rendimiento del trabajo. La práctica industrial ha demostrado que Kanban puede reducir los costos de mantenimiento y desarrollo (Ahmad, et al., 2016). El básico La idea es que cada organización implemente los principios básicos de Kanban a su manera, a través de aprendizaje continuo a partir de sus propias experiencias. Por esta razón, no existen pruebas sólidas Pautas para su implementación efectiva. La filosofía Kanban se centra en reducir

costos generales, mejorando la calidad general y la calidad de los productos entregados a los clientes, acortando los plazos de entrega de los productos y aumentando la satisfacción del cliente. Una de las más importantes ideas del método Kanban es eliminar el exceso. Esto se logra mediante el uso de Kanban. Tarjetas y tableros Kanban para visualizar cómo se mueven los recursos a través del ciclo de producción. Esto permite que todos los participantes se involucren plenamente en el proceso y ayuda a los gerentes a lograr un excedente o déficit de producción. Kanban permite a las organizaciones comenzar con sus productos existentes. flujo de trabajo, introduciendo cambios lentamente a lo largo del tiempo para alcanzar el nivel deseado y lograr el resultado deseado. Esto también se puede lograr limitando el trabajo en curso (Trabajo en curso - WIP). Las actividades fomentadas son visualización, trabajo en progreso, gestión del flujo de trabajo, reglas y retroalimentación claramente definidas. Los sistemas Kanban utilizan mecanismos como el tablero Kanban. Visualizar el trabajo y el proceso por el que pasa. Varios indicadores de efectividad se han definido los puntos de visualización. Algunos de ellos son el punto de obligación y el punto de entrega. El punto de obligación surge cuando el equipo de trabajo se compromete a realizar un determinado trabajo. tarea (Pantelić, et al., 2020). El punto de entrega es cuando el equipo entrega el elemento de trabajo al cliente. Entonces, la visualización efectiva se refleja en políticas que determinan qué trabajo debe estar en una etapa particular, y también en restricciones sobre el trabajo en curso. Cuando los límites en la cantidad de trabajo en progreso se determina mediante un sistema y se utilizan esos límites como si supiera cuándo. Para iniciar un nuevo proceso, se puede reducir el flujo de trabajo y el tiempo, mejorar la calidad y entregar con más frecuencia. El flujo de trabajo debe maximizar la entrega de valor, minimizar el tiempo perdido y ser lo más predecible posible. Los equipos utilizan el control empírico a través de la transparencia, la inspección y adaptación para equilibrar objetivos potencialmente conflictivos y cómo no llegarían en absoluto. Claro las reglas ayudan a explicar todo el proceso y las diferentes etapas del mismo. Las reglas deben ser simple, bien definido, visible, siempre aplicable y que pueda modificarse fácilmente. La retroalimentación es un elemento esencial en cualquier sistema que busque un cambio evolutivo (Kniberg y Skarin, 2010; Cocco, et al., 2011; Senapathi y Drury-Grogan, 2021; Santos, et al., 2018).

### 3.5.3 Diferencias entre Scrum y Kanban

Tanto Scrum como Kanban pertenecen a los métodos Agile y Lean, donde el foco es responder rápidamente a las solicitudes de los clientes. Ambos son altamente adaptables y se basan en un enfoque altamente colaborativo y equipos autogestionados. Kniberg y Skarin (2010) afirmaron que Scrum es más prescriptivo que Kanban. Scrum prescribe el uso de iteraciones de tiempo, Kanban no. Existe una diferencia entre las placas Scrum y Kanban. Scrum tiene dos modos: modo de programación y modo operativo. El tablero Scrum es utilizado por los equipos para planificar el trabajo antes de que comience. En la planificación En este modo, los equipos comparten su trabajo en sprints y asignan puntos de historia a las historias de usuario para ayudar a planificar qué

La historia se desarrolla en cada sprint. Por el contrario, Kanban no tiene un plan. Kanban utiliza el mismo Interfaz basada en columnas como Scrum para rastrear el estado del elemento sin organizar el trabajo en sprints (Ahmad, et al., 2016). Sahota explica que Kanban puede gestionar una gran cantidad de interrupciones. En el proyecto, personal de apoyo con roles especializados y diferentes habilidades. Kanban funciona. Bueno para equipos más grandes porque los costos de comunicación y planificación son menores. Scrum es Mejor en proyectos que requieren colaboración profunda e innovación y trabaja mejor con pequeñas empresas. equipos multifuncionales y alienta a los expertos generales (Lei, et al., 2017; Ingason, et al., 2013; Ahmad y otros, 2014; Nikitina y Kajko-Mattsson, 2011; Al Baik y Miller, 2015; Sjøberg, et al., 2012; Ahmad, et al., 2016; Fagarasan, 2021). Muchas empresas que anteriormente utilizaban Scrum están Adoptar Kanban debido a sus características, incluida la transparencia y el trabajo en progreso limitado (Ahmad, et al., 2014; Nikitina y Kajko-Mattsson, 2011; Al Baik y Miller, 2015; Sjøberg, et al., 2012; Ahmad, et al., 2016; Shafiq e Inayat, 2017).

### 3.5.4 Programación extrema (XP)

Con el método de programación extrema, el enfoque está en producir software de mejor calidad y Sobre el trabajo más productivo del equipo de desarrollo. Fue desarrollado por Kent Beck y Ward. Channing a mediados de la década de 1980. Beck agregó nuevos principios y formas de aplicación a proyectos con la aplicación de los elementos más importantes, a saber, la comunicación, la simplicidad, la retroalimentación y Coraje (Pap, 2008). Es específico porque se utiliza específicamente para el desarrollo de software. (Beck & Andres, 2004). El principio de este método es que el cliente es miembro del equipo. y define los objetivos y prioridades dentro de las unidades usuarias, en constante cooperación con el miembros del equipo para que el equipo de desarrollo comprenda mejor los requisitos del usuario. El desarrollo pasa por ciclos cortos que incluyen el plan de iteración actual. El éxito de El proceso de desarrollo se mide observando el progreso. Todos los detalles de las unidades de usuario son documentados en forma de pruebas, para facilitar el seguimiento del progreso y su implementación. Requisitos. El código productivo debe ser escrito por dos programadores en una unidad de desarrollo. Un miembro escribe el código mientras otros lo revisan y así siguen el proceso de implementación. Es deseable que los roles cambien frecuentemente para mantener una relación de calidad entre Miembros. El código productivo se escribe para satisfacer el código de prueba. El software desarrollado es el propiedad colectiva de los miembros del equipo. La esencia de la planificación está en la división de responsabilidades entre el usuario y el equipo de desarrollo. El usuario decide qué funcionalidades que el software debe implementar y su costo. El sistema debe ser Diseñado lo más simple posible con refactorización frecuente. La mayor ventaja de XP es que... permite a las empresas de desarrollo de software ahorrar costos, frustraciones y tiempo al eliminar



actividades improductivas. Busca principalmente reducir los riesgos asociados al fracaso del proyecto.

Y esto permite a los desarrolladores centrarse en la codificación. La construcción de la solución de software XP

El proceso se realiza a través de varias fases de desarrollo: investigación, planificación, iteración

implementación, producción, mantenimiento y finalización (Marić, 2015; Tadić, 2005; Malik, et

otros, 2019; Papanicolaou, 2008; Subih, et al., 2019).

Según el autor (Malik, et al., 2019), la flexibilidad de los métodos ágiles es la más

característica importante, ya sea Scrum, programación extrema u otros métodos. Ellos

afirman que uno de los problemas de los métodos ágiles es que los proyectos complejos, partes de los métodos ágiles,

como reuniones o llamadas telefónicas, son necesarias y pueden ser un problema en el ámbito internacional.

Equipos con diferentes zonas horarias. La programación extrema se centra más en el software.

desarrollo o las mejores prácticas para su desarrollo, dejando de lado, o dando menor prioridad, a las mejores

Prácticas sobre cómo implementar todo el proyecto dentro de un presupuesto determinado y dentro de plazos definidos.

plazos (Javanmard y Alian, 2015). Para facilitar esta aplicación en proyectos más grandes de

complejidad, se recomienda utilizar un software de gestión de proyectos donde, según la

autores (Fabac, et al., 2009), divisiones en tres categorías de software de gestión de proyectos,

Son posibles los programas de gestión de procesos y de seguimiento del tiempo (Vresk, 2020).

## 4. Análisis empírico y discusión

Con base en una investigación realizada en una empresa de TI, se identificó que los encuestados eligen el modelo tradicional

métodos cuando saben exactamente lo que hay que hacer durante el proyecto, mientras que los métodos ágiles

Se eligen con mayor frecuencia cuando solo se conocen requisitos y objetivos "aproximados". Tradicional

Se eligen métodos incluso cuando el proyecto no se puede dividir en partes más pequeñas, mientras que los métodos ágiles

Se eligen metodologías cuando se necesita flexibilidad. Es importante destacar que

Los métodos tradicionales se eligen cuando el cliente del proyecto no cuenta con el personal técnico necesario.

verifica ciertas partes y solo le interesa el producto final, mientras que se eligen métodos ágiles

cuando se esperan cambios durante el proyecto y son comunes (Vresk, 2020). Livermore

La investigación (Livermore, 2008) muestra que no hubo correlación significativa entre el tamaño del equipo

y el éxito de la implementación de métodos en relación con ellos. Afirma que esto es un

Resultado inesperado, considerando que existe un gran número de investigaciones sobre el tema mencionado.

Concluyen que los métodos ágiles tienen menos éxito en equipos más grandes. Estos resultados respaldan el hecho

que los métodos, especialmente los ágiles, son adaptables a diferentes equipos y situaciones. Su

La flexibilidad no sólo se aplica aquí en términos de respuesta rápida a solicitudes adicionales, sino que también

También son flexibles para adaptarse a un equipo o proyecto: tamaño del equipo, rol del equipo, duración del sprint (Vresk,

2020). Las metodologías ágiles son mejores que las tradicionales porque pueden ahorrar dinero, tiempo y entregar productos de calidad a tiempo (Venkatachalam, et al., 2017). Según los autores (Merzouk, et al., 2018), y su comparación de métodos ágiles, al elegir el método ágil correcto. Método para un proyecto, es necesario observar el tamaño del equipo, del proyecto (Vresk, 2020). En las metodologías ágiles los requisitos del usuario no están claramente definidos, sólo se define el resultado final. Los equipos son más pequeños que con la metodología tradicional. El cliente participa en todo el proyecto. De principio a fin. La clave es contar con un equipo que trabaje unido. La comunicación es informal. Las metodologías ágiles son un modelo iterativo (Vresk, 2020). En 2010, Forrester (Womack, et al., 1990), publicaron los resultados de su Encuesta Global de Tecnografía para Desarrolladores, que reveló que El 35% de los encuestados utilizó un proceso de desarrollo ágil, el 21% uno iterativo y el 13% uno en cascada. Mientras que el 31% no utilizó una metodología de proceso formal (Mahnič, 2014).

Según Rising y Janoff (2000) y Schwaber (2004), la metodología ágil más extendida El método es Scrum. Se informa que el 54% utiliza Scrum puro, el 11% utiliza Scrum/XP híbrido y el 11%... Scrumban por el 7% de los encuestados. Sin embargo, una encuesta también reveló un rápido crecimiento de la Número de usuarios de Kanban. En comparación con 2011, Kanban y sus variantes (Anderson, 2010; Ladas, 2008), casi se duplicó en 2012, principalmente debido a un aumento en el uso de Scrumban (Mahnič, 2014).

Como cada método tiene sus propias características y desventajas, un marco independiente de Scrum o Kanban o Waterfall no pueden brindar soluciones completas a todos los desafíos. Los marcos ágiles son... Muy adecuado para proyectos de software de gran o mediano alcance, donde el requisito del proyecto también es Su viabilidad a menudo cambia con el tiempo, mientras que el método en cascada es adecuado para proyectos a corto plazo y Un requisito claro en la fase inicial del proyecto permite predecir los requisitos del proyecto. no puede cambiar durante su ciclo de vida hasta la entrega final del proyecto. La integración de Scrum y Kanban con Waterfall aportan una gran fortaleza a la Ingeniería de Software

Prácticas de gestión de proyectos (SEM) en forma de Scrumbanfall. Los artefactos de Scrumbanfall son la base de El marco en forma de valor para el cliente, documentación, gestión del flujo de trabajo, transparencia, aumento y oportunidad para el equipo de Scrumbanfall y otras partes interesadas (Bhavsar, et al., 2020). Cocco et al (2011), desarrollaron un modelo de dinámica de sistemas para la comparación de medias de técnicas de simulación para el enfoque prescriptivo utilizando Scrum y Kanban en Cascada en 2011, y evaluó que cada uno de ellos tiene sus propias fortalezas y debilidades y propuso Se realizarán más investigaciones sobre la combinación de todos ellos para resolver el desarrollo ágil de software. problema (Bhavsar, et al., 2020). En 2019, Mohan, Devisree y Kumar combinaron las reglas de Scrum con flexibilidad Kanban en forma de Scrumban e implementada en el sector gubernamental Desarrollo de aplicaciones de software que redujeron el estrés del desarrollo general y

Aumento de la eficiencia del software. Scrum, Kanban y Waterfall en Scrumbanfall, que tiene un gran Resistencia en comparación con un marco independiente y capacidades para responder a los desafíos de Prácticas de desarrollo y gestión de software como la participación directa de personal externo.

las partes interesadas en la documentación de análisis y requisitos del proyecto; planificación y estimación del proyecto y rastreador. Una estimación inexacta genera riesgos en el proyecto y una visión poco clara del software.

producto, lo que resulta en un costo inesperado en el proyecto para los propietarios del proyecto de software o organizaciones de desarrollo de software que asumen el costo del desarrollo del proyecto (Bhavsar, et al., 2020). En un artículo, los autores analizaron el comportamiento dinámico de la adopción de Kanban y Scrum, en comparación con un proceso de desarrollo de software tradicional como el enfoque en cascada.

utilizar un modelo de dinámica de sistemas, basado en las relaciones entre las variables del sistema, para evaluar Los beneficios relativos de los enfoques estudiados. El flujo de trabajo Kanban se gestionó mediante un Mecanismo de control eficaz para limitar el trabajo en curso y minimizar el plazo de entrega. Uno de Las ventajas de este enfoque es que el trabajo está mejor controlado, por lo que los efectos de los errores se mantiene limitado. Por el contrario, en el caso de la cascada, a menudo los proyectos pueden no completarse debido a a la dificultad de corregir errores, incluidos los errores en los requisitos (Cocco, et al., 2011). En En segundo lugar, el proyecto de muro Kanban puede facilitar la revisión y ejecución del proyecto y hacer Comunicación entre las partes interesadas más eficiente y eficaz. El uso combinado de Scrum Y Kanban ha sido impecable en este proyecto y el nuevo método ha tenido éxito desde su inicio.

inicio hasta ahora (Ingason, et al., 2013). Un estudio mostró que el tamaño del grupo más adaptable en Kanban y XP que en Scrum, mientras que el tamaño del WIP (Limiting Work in Progress) Scrum es uno, dos o un mes en el sprint, pero la evaluación de Kanban y XP WIP es relativamente bajo (puede ser tan pequeño como un grupo de una hora) (Saleh, et al., 2019; Cohn, 2007). En este Respecto a los permisos Kanban, se organizan las condiciones previas día a día y los permisos XP se organizan. necesidades continuamente, mientras que las condiciones previas deben organizarse en función de la duración de la ejecución. Durante la adopción de Scrum (Kanwal, et al., 2010; Beck y Fowler, 2000). Además, en Kanban El volumen del componente es pequeño en comparación con Scrum y XP, pero una industria se reduce a la mitad cuando se pasa de Scrum a Kanban en los lapsos de tiempo entre proponer el segundo elemento y crear solicitudes y enviarlas al sitio web de su usuario (Shalloway, 2011; Raman, 2014). Kanban y XP también están más centrados en el software de mejora de la calidad (Verweij y Maassen, 2011; Sjøberg, et al., 2012; Ramón, 2014).

Las investigaciones muestran que aproximadamente la mitad de las empresas todavía utilizan el modelo en cascada, mientras que el resto La mitad utiliza enfoques ágiles e iterativos (Holz, 2019). Las empresas que utilizan métodos ágiles, Según datos de la décima encuesta anual VersionOne, la mayoría opta por Scrum y

Scrum + XP (70%), Scrumban (7%) y Kanban (5%) (Sutherland, 2010). De nuestra selección

De los métodos ágiles, eliminamos la Programación Extrema (XP), porque sus principios a menudo son utilizado en combinación con otros métodos (Scrum, Kanban) (Brezočnik y Majer, 2016). Uno

El estudio demostró que Scrum ciertamente funciona mejor en empresas maduras que han experimentado equipos que trabajan en el producto o proyecto durante más de un año. Para empresas con

producción continua que necesitan una respuesta rápida a los cambios y equipos de producto que están

Para trabajar en el soporte y mantenimiento del producto, sugerimos usar Kanban. Scrumban es

Ideal para empresas jóvenes y pequeñas, ya que contiene la flexibilidad de Kanban y las características básicas. de Scrum. Pero sin duda, los métodos ágiles incluyen un fuerte componente de flexibilidad. Los equipos podrían,

Independientemente del método elegido, adáptelo de manera que cumpla su propósito: trabajo eficaz.

Organización y desarrollo de productos de calidad (Brezočnik y Majer, 2016).

Según Alaidaros, et al. (2018), Scrum es el método más seguido y el 58% de los

Los encuestados lo practican entre otros métodos ágiles, mientras que más del 39% de los encuestados Practicar el método Kanban en su organización. Scrum es el método ágil más popular.

Las actividades cotidianas se basan en las experiencias pasadas y no sólo en aspectos teóricos (Mircea, 2019). La metodología más popular utilizada por los candidatos en la segunda investigación fue Scrum, con un 62,5% de ellos utilizando Scrum al menos una vez. El 28,1% de ellos utilizó Waterfall y solo el 25% Usaron Kanban (Andrei, et al., 2019). Los encuestados que usaron Scrum informaron que eran altamente...

motivados durante el desarrollo y les gustó el hecho de que estaban en buena sincronización

Con el resto del equipo. Sin embargo, algunos no disfrutaron del gran número de reuniones.

y el hecho de que inducían una rutina. Los usuarios de Kanban disfrutaban de la simplicidad y flexibilidad de

La metodología al informar problemas con la priorización de tareas y la cantidad de tiempo que

Se dedicó a actualizar las tarjetas. Los usuarios de Waterfall disfrutaron del desarrollo basado en planes que este

La metodología fomenta y los estrictos requisitos impuestos desde el principio pero tuvieron problemas medición del progreso (Andrei, et al., 2019). Según las estadísticas presentadas en el Informe Anual Informe State of Agile (Engelhardt, 2019), desarrollado por Digital.ai, el 58% de los proyectos de software que se implementaron utilizando las metodologías ágiles también eligieron Scrum como

Método de implementación. En contraste, solo el 7% del total de proyectos de software se implementaron.

Usando el método de desarrollo Kanban puro, esta estadística no debería influir en un proyecto.

decisión de los gerentes al elegir el método de implementación adecuado (Fagarasan, et al., 2021). Diversos estudios informaron que el método Kanban, actualmente, es el contendiente entre los métodos ágiles.

métodos porque tiene numerosas ventajas que hacen que funcione mejor que Scrum y otros

Métodos ágiles en términos de tener experiencia mayor consistencia en la gestión de software

proyectos de ingeniería (SE) (Flora, & Chande, 2014; Lei, et al., 2017; Karunanithi, 2016).

Sin embargo, el método Kanban presenta deficiencias significativas en la tarea de seguimiento del progreso durante proceso de desarrollo de proyectos de software. Este problema afecta negativamente el éxito de proyectos de software debido a retrasos en la programación de los proyectos que conducen a entregas tardías (Skinner, et otros, 2015; Al Baik y Miller, 2015; Kirovska y Koceski, 2015).

Como se puede ver en la Tabla 1, las metodologías ágiles se utilizan solo cuando hay requisitos aproximados.

y un objetivo, cuándo son necesarios cambios durante el proyecto y dónde son posibles los cambios

durante todo el proyecto. Esto ayuda a ahorrar dinero y a entregar un producto de calidad a tiempo. Tradicional

Se utilizan metodologías cuando todos los requisitos del cliente se conocen exactamente desde el principio.

Utilizado por el cliente del proyecto, sabe exactamente qué quiere que sea su producto final. Según...

En la literatura científica, la metodología ágil más utilizada es SCRUM, y la tradicional es

Cascada.

Tabla 1. Una revisión de las razones para el uso de metodologías ágiles y tradicionales y una

Revisión de la metodología más utilizada

Metodologías en TI	METODOLOGÍA ÁGIL TRADICIONAL	
Gestión de proyectos	METODOLOGÍA	
Selección de la metodología	Se utilizan cuando sólo se conocen requisitos y objetivos aproximados y se espera la flexibilidad y los cambios necesarios durante el proyecto, ya que pueden ahorrar dinero, tiempo y entregar productos a tiempo.	Se utilizan cuando se sabe exactamente lo que hay que hacer a lo largo del proyecto y cuando el proyecto no se puede dividir en partes más pequeñas y cuando el cliente del proyecto sólo está interesado en el producto final y la comunicación en el equipo es informal.
Mayormente aplicado	Scrum (Mircea, 2019)	Cascada (Mircea, 2019; (Andrei, et al., 2019).)

Dado que cada metodología tiene algunas deficiencias, es necesario combinar varias

metodologías en algunos casos cuando la naturaleza del proyecto lo requiere. Scrum y Kanban son

Adecuado para proyectos de software de gran o mediana escala donde los requisitos cambian a lo largo de la vida del proyecto.

El proyecto. La cascada es adecuada para proyectos a corto plazo y se conocen los requisitos.

desde el inicio del proyecto. Scrum ciertamente funciona mejor en empresas maduras que tienen

equipos experimentados que llevan más de un año trabajando en el producto o proyecto. Para

Empresas con producción continua que necesitan una respuesta rápida a los cambios y a los equipos de producto.

que están trabajando en soporte y mantenimiento del producto, es necesario utilizar Kanban.

Scrumban es mejor para empresas jóvenes y pequeñas, ya que contiene la flexibilidad de Kanban y los conceptos básicos.

Características de Scrum. Scrumban se puede utilizar como una combinación de Scrum y Kanban y La ventaja es que el trabajo está mejor controlado. Puede ser mejor que usar Cascada porque En Waterfalls a menudo los proyectos no se pueden completar debido a la dificultad de corregir errores. La combinación de Scrum, Kanban y Waterfall puede facilitar la comunicación entre las partes interesadas sean más eficientes y eficaces. La combinación de estos tres mencionados Las metodologías se denominan Scrumbanfall. Scrumbanfall tiene una gran capacidad para responder a las necesidades de software. desafíos de desarrollo en las prácticas de gestión, como la participación directa de actores externos partes interesadas en los requisitos del proyecto y la inclusión de la documentación de análisis, así como del proyecto. planificación, evaluación y seguimiento. Dado que las metodologías ágiles son actuales, sería No estaría mal ver cuándo es necesario elegir el adecuado. Según la ciencia En la literatura, se puede observar que Scrum es el más utilizado, sin embargo el tamaño del grupo es más adaptable en Kanban y XP que en Scrum, mientras que el tamaño del WIP de Scrum es de uno, dos o un mes en el sprint, Pero la estimación de Kanban y XP WIP es relativamente baja. El permiso Kanban organizado condiciones previas del día a día y XP permite organizar las necesidades de forma continua, mientras Las condiciones previas deben organizarse en función de la duración de la ejecución durante la adopción de Scrum. Kanban y XP también están más centrados en el software de mejora de la calidad.

## 5. Conclusión

En el desarrollo de software, el papel del cliente y la comunicación constante con él es Muy importante para cumplir con sus requisitos y llegar al resultado final deseado. Presentando y comparar estos métodos nunca funciona por una razón: para encontrar el mejor método, sino para Se exploraron las diferencias entre ellos y se encontraron posibles razones que explican por qué y en qué situaciones elegir un método en particular o una combinación de ellos. No existe una única Receta para la gestión y el liderazgo de proyectos debido a la amplia gama de problemas y requisitos que surgen. No existe una mejor metodología. Dependiendo de la naturaleza de la problema, se elegirá la metodología adecuada. Si el proyecto tiene claramente definido requisitos y objetivos, se utilizarán algunas de las metodologías tradicionales, pero para los proyectos Con requisitos inestables, siempre se utilizarán metodologías ágiles o combinaciones, Porque son adaptables.

## Referencias

- Abrahamsson, P., Salo, O., Ronkainen, J. y Warsta, J. (2002). Desarrollo de software ágil  
Métodos: Revisión y análisis. Centro de Investigación Técnica VTT de Finlandia. Publicaciones VTT.  
N.º 478. <http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf>
- Ahmad, MO, Dennehy, D., Conboy, K. y Oivo, M. (2018). Kanban en software  
Ingeniería: Un estudio de mapeo sistemático. Revista de Sistemas y Software, 137, 96-113.  
<https://doi.org/10.1016/j.jss.2017.11.045>
- Ahmad, MO, Markkula, J. y Oivo, M. (2013). Kanban en el desarrollo de software: una  
Revisión sistemática de la literatura. Actas de la 39.ª Conferencia Euromicro de 2013 sobre  
Ingeniería de Software y Aplicaciones Avanzadas, 9-16.  
<https://doi.org/10.1109/SEAA.2013.28>
- Ahmad, MO, Kuvaja, P., Oivo, M. y Markkula, J. (2016). Transición de software  
Equipos de mantenimiento de Scrum a Kanban. 49.ª Conferencia Internacional de Hawái de 2016 sobre  
Ciencias de Sistemas (HICSS), 5427-5436. <https://doi.org/10.1109/HICSS.2016.670>
- Ahmad, MO, Markkula, J., Oivo, M. y Kuvaja, P. (2014). Uso de Kanban en software  
Empresas: Un estudio empírico sobre motivación, beneficios y desafíos. Actas del...  
Novena Conferencia Internacional sobre Avances en Ingeniería de Software (ICSEA 2014), 150-155.  
ISBN: 978-1-61208-367-4
- Alaidaros, H., Omar, M., y Romli, R. (2018). Hacia un proyecto de software mejorado  
Modelo de Monitoreo de Tareas del Método Ágil Kanban. Revista Internacional de la Cadena de Suministro  
Gestión (IJSCM), 7(3), 118-125. <https://core.ac.uk/download/pdf/230745993.pdf>
- Al-Baik, O., y Miller, J. (2015). El enfoque kanban, entre agilidad y leanness: una  
Revisión sistemática. Ingeniería de software empírica, 20(6), 1861-1897.  
<https://doi.org/10.1007/s10664-014-9340-x>
- Albarqi, AA y Qureshi, R. (2018). La metodología L-Scrumban propuesta para mejorar  
La eficiencia del desarrollo ágil de software. Revista Internacional de Información  
Ingeniería y Negocios Electrónicos, 3, 23-35. <https://doi.org/10.5815/ijieeb.2018.03.04>
- Anderson, DJ (2010). Kanban: Un cambio evolutivo exitoso para su tecnología  
Negocios. Blue Hole Press. ISBN: 0984521402
- Andrei, BA., Casu-Pop, AC., Gheorghe, SC. y Boiangiu, CA. (2019). Un estudio sobre el uso  
Métodos ágiles y en cascada en la gestión de proyectos de software. Revista de Sistemas de Información

& Gestión de Operaciones (JISOM), 13(1), 125-135.

<http://jisom.rau.ro/Vol.13%20No.1%20-%202019/JISOM-SU19-A12.pdf>

Antill, MJ y Woodhead, WR (1991). Métodos de la ruta crítica en la práctica de la construcción (4.ª Edición). John Wiley e hijos. ISBN: 978-0-471-62057-0

Barash, I. (2013). Uso de metodologías ágiles con XP y Kanban en el mismo proyecto. PM Revista Mundial, II(X). <https://pmworldlibrary.net/wp-content/uploads/2013/10/pmwj15-oct2013-barash-uso-de-agile-xp-kanban-mismo-proyecto-segunda-edición.pdf>

Beck, K., y Andres, C. (2004). Programación extrema explicada: Aceptar el cambio (2.ª Edición). Addison-Wesley Professional. ISBN: 0321278658.  
<https://ptgmedia.pearsoncmg.com/images/9780321278654/samplepages/9780321278654.pdf>

Beck, K. y Fowler, M. (2000). Planificación de programación extrema. Addison-Wesley Longman Compañía Editorial, Inc. ISBN:978-0-201-71091-5

Bhavsar, K., Shah, V. y Gopalan, S. (2020). Scrumbanfall: una integración ágil de Scrum y Kanban con Cascada en Ingeniería de Software. Revista Internacional de Ingeniería Innovadora Tecnología e Ingeniería de Exploración (IJITEE), 9(4), 2075-2084.  
<https://10.35940/ijitee.D1437.029420>

Brezočnik, L. y Majer, Č. (2016). Comparación de métodos ágiles: Scrum, Kanban y Scrumban. Actas de la XIX Multiconferencia Internacional sobre la Sociedad de la Información (SI) 2016: Colaboración, software y servicios en la sociedad de la información, 30-33.  
[http://library.ijs.si/Stacks/Proceedings/InformationSociety/2016/IS2016\\_Volume\\_C%20-%20CSS.pdf](http://library.ijs.si/Stacks/Proceedings/InformationSociety/2016/IS2016_Volume_C%20-%20CSS.pdf)

Cocco L., Mannaro K., Concas G., Marchesi M. (2011). Simulando Kanban y Scrum vs. Cascada con dinámica de sistemas. Procesos en ingeniería de software y situaciones extremas. Programación. XP 2011. Apuntes de clase sobre procesamiento de información empresarial, 117-131.  
[https://doi.org/10.1007/978-3-642-20677-1\\_9](https://doi.org/10.1007/978-3-642-20677-1_9)

Cohn, M. (6 de abril de 2007). Diferencias entre Scrum y Programación Extrema. Mountain Goat Software. <https://www.mountaingoatsoftware.com/blog/differences-between-scrum-and-programación-extrema/comentarios>

dos Santos, PSM, Beltrão, AC, de Souza, BP y Travassos, GH (2018). en el Beneficios y desafíos del uso de Kanban en ingeniería de software: un estudio de síntesis estructurado.



Revista de investigación y desarrollo en ingeniería de software, 6(1).

<https://doi.org/10.1186/s40411-018-0057-1>

Engelhardt, N. (2019). Comparación entre la gestión de proyectos ágil y tradicional: Simulación Modelos de procesos. *Acta académica karvinienisa*, 19(2), 15-27. <https://10.25142/aak.2019.011>

Fabac, R., Pihir, I., y Radošević, D. (2009). La aplicación de metodologías modernas y Las TIC en la gestión de proyectos en empresas croatas. *Actas del 1.er Conferencia Internacional sobre Sociedad de la Información y Tecnologías de la Información, ISIT 2009*. <https://ssrn.com/abstract=2505743>

Fagarasan, C., Popa, O., Pislă, A. y Cristea, C. (2021). Ágil, en cascada e iterativo Enfoque en proyectos de tecnologías de la información. *Actas de la serie de conferencias del IOP: Ciencia e Ingeniería de Materiales*, 1169(1). <https://10.1088/1757-899X/1169/1/012025>

Ferrão, S.É. R., y Canedo, ED (2015). Un estudio de la aplicabilidad de una metodología ágil Scrum aliado al método Kanban. *Actas de la X Conferencia Ibérica sobre Sistemas y tecnologías de la información (CISTI)*. <http://dx.doi.org/10.1109/CISTI.2015.7170382>

Flora, HK, y Chande, SV (2014). Un estudio sistemático sobre el desarrollo ágil de software. *Metodologías y Prácticas. Revista Internacional de Ciencias de la Computación e Información Tecnologías (IJCSIT)*, 5(3), 3626-3637. [https://www.researchgate.net/publication/266742256\\_Un\\_Estudio\\_Sistemático\\_sobre\\_Software\\_Agil\\_re\\_Metodologías\\_y\\_prácticas\\_de\\_desarrollo](https://www.researchgate.net/publication/266742256_Un_Estudio_Sistemático_sobre_Software_Agil_re_Metodologías_y_prácticas_de_desarrollo)

Fustik, V. (2017). Metodología Scrum comparada con otras metodologías aplicadas en el Proyectos de desarrollo de software. *Actas de la Conferencia Internacional sobre Información Tecnologías (InfoTech-2017)*. <http://infotech-bg.com/sites/default/files/2017/A-VF.pdf>

García, AL, da Rocha Miguel, I., Eugênio, JB, da Silva Vilela, M. y Marcondes, GAB (2020). Aplicación basada en Scrum para la gestión ágil de proyectos. *Journal of Software*, 15(4), 106-113. <https://10.17706/jsw.15.4.106-113>

Granulo, A., y Tanović, A. (2019). Comparación de SCRUM y KANBAN en el aprendizaje. *Proceso de implementación del Sistema de Gestión. Actas de la 27.ª Conferencia de Telecomunicaciones. foro TELFOR 2019*, 636-639. <https://doi.org/10.1109/TELFOR48224.2019.8971201>

- Holz, B. (2019). Resumen de resultados: Agile en la empresa. Gartner.  
[https://circle.gartner.com/Portals/2/Resources/pdf/Agile%20in%20the%20Enterprise%202019%20-%20Resumen%20de%20resultados%20\(actualizado\).pdf](https://circle.gartner.com/Portals/2/Resources/pdf/Agile%20in%20the%20Enterprise%202019%20-%20Resumen%20de%20resultados%20(actualizado).pdf)<https://doi.org/10.1002/hfm.4530040310>
- Ingason, HT, Gestsson, E. y Jonasson, HI (2013). El Proyecto Kanban Wall: Combinando Kanban y Scrum para la coordinación de proyectos de software. PM World Journal, II(VIII).  
<https://pmworldlibrary.net/wp-content/uploads/2013/08/pmwj13-agosto2013-Ingasson-Gestsson-Jonasson-Project-Kanban-Wall-FeaturedPaper.pdf>
- Islam, KA (2013). Metodología ágil para el desarrollo y la medición del aprendizaje: Formación Desarrollo para el mundo actual. Autor House. ISBN: 9781491823880
- Javanmard, M. y Alian, M. (2015). Comparación entre software ágil y tradicional. Metodologías de desarrollo. Revista Cumhuriyet Science, 36(3), 1386-1394.  
<https://dergipark.org.tr/en/pub/cumuscij/issue/45132/564438>
- Kanwal, F., Junaid, K. y Fehiem, MA (2010). Evaluación de una arquitectura de software híbrida. Método para FDD: un modelo de proceso ágil. Actas de la Conferencia Internacional sobre Inteligencia Computacional e Ingeniería de Software (CiSE), 1-5.  
<https://doi.org/10.1109/CISE.2010.5676863>
- Karunanithi, K. (2016). Métricas en Agile y Kanban , Técnicas de medición de software. Proyecto: Medición de software. Universidad Estatal de California.  
[https://www.researchgate.net/publication/305613018\\_Métricas\\_en\\_Software\\_Agile\\_y\\_Kanban\\_re\\_Técnicas\\_de\\_Medición](https://www.researchgate.net/publication/305613018_Métricas_en_Software_Agile_y_Kanban_re_Técnicas_de_Medición)
- Kirovska, N., y Koceski, S. (2015). Uso de la metodología Kanban en el desarrollo de software. equipos. Revista de Economía Aplicada y Negocios, 3(3), 25-34.  
<http://www.aebjournal.org/articles/0303/030302.pdf>
- Kniberg, H., y Skarin, M. (2010). Kanban y Scrum: Aprovechar al máximo ambos. C4Media Inc. [http://www.agileinnovation.eu/wordpress/wp-content/uploads/2010/09/KanbanAndScrum\\_AprovechandoAlMáximoDeAmbos.pdf](http://www.agileinnovation.eu/wordpress/wp-content/uploads/2010/09/KanbanAndScrum_AprovechandoAlMáximoDeAmbos.pdf)
- Laanti, M. (2013). Ágil y bienestar: Estrés, empoderamiento y rendimiento en Scrum y equipos Kanban. Actas de la 46.ª Conferencia Internacional de Hawái sobre Sistemas Ciencias, 4761-4770. <https://doi.org/10.1109/HICSS.2013.74>
- Ladas, C. (2008). Scrumban - Ensayos sobre sistemas Kanban para el desarrollo de software Lean. Prensa Modus Cooperandi.

Lei, H., Ganjezadeh, F., Jayachandran, PK y Ozcan, P. (2017). Un análisis estadístico de la Efectos de Scrum y Kanban en proyectos de desarrollo de software. Robótica e informática. Manufactura Integrada, 43, 59-67. <https://doi.org/10.1016/j.rcim.2015.12.001>

Livermore, AJ (2008). Factores que impactan significativamente la implementación de un enfoque ágil. Metodología de desarrollo de software. Revista de Software, 3(4), 31-36. <http://www.jssoftware.us/vol3/jsw0304-04.pdf>

Mahnič, V. (2014). Mejora del desarrollo de software mediante la combinación de Scrum y Kanban. Actas de la 8.ª Conferencia Internacional WSEAS sobre Ingeniería Informática. y Aplicaciones (CEA '14), 281-288. <http://www.wseas.us/e-biblioteca/conferencias/2014/Tenerife/INFORM/INFORM-40.pdf>

Mahnič, V. (2015). De Scrum a Kanban: introduciendo los principios Lean a un software Curso final de ingeniería. Revista Internacional de Educación en Ingeniería, 31(4), 1106-1116

Malik, RS, Ahmad, SS y Hussain, MTH (2019). Una revisión de la metodología ágil en TI. Proyectos. Actas de la 2ª Conferencia Internacional sobre Computación Avanzada y Ingeniería de Software (ICACSE-2019), 655-656. <http://dx.doi.org/10.2139/ssrn.3351064>

Marić, M. (2015). Metodološko-radni okvir za razvoj softverske arhitekture poslovnog softvera u agilnim procesima [Un marco metodológico para el desarrollo del software] Arquitectura de software empresarial en procesos ágiles]. [Tesis doctoral, Universidad de Novi Sad, Facultad de Economía, Subotica]. PQDT Abierto. [https://hdl.handle.net/21.15107/rcub\\_nardus\\_5151](https://hdl.handle.net/21.15107/rcub_nardus_5151)

McCormick, M. (2012). Metodología en cascada vs. metodología ágil. MPCS, Inc. [http://mccormickpcs.com/images/Cascada\\_vs\\_Metodologia\\_Agil.pdf](http://mccormickpcs.com/images/Cascada_vs_Metodologia_Agil.pdf)

Merzouk, S., Elhadi, S., Cherkaoui, A., Marzak, A. y Sael, N. (2018). Software ágil Desarrollo: Estudio comparativo. Actas de la Segunda Conferencia Internacional sobre Aplicación inteligente y análisis de datos para ciudades inteligentes (SADASC'18). <http://dx.doi.org/10.2139/ssrn.3186323>

Milošević, D. (2 de diciembre de 2018). Agilna metodologija [Metodología ágil]. Analítica i vizuelizacija podataka [Análisis y visualización de datos]. <https://dusanmilosevic.com/agilna-metodologija/>

- Mircea, E. (2019). Gestión de proyectos con marcos ágiles. *Economía Informática*, 19(1), 34-44. <https://10.12948/ei2019.01.04>
- Mohan, A., Devisree, AS, y Kumar, CVP (2019). Scrumban racionalizado basado en Metodología para mejorar el sistema de pensiones de los sectores de empleados en la India. *Revista Internacional de Tecnología e Ingeniería Reciente (IJRTE)*, 8(1), 1525-1550. <https://www.ijrte.org/wp-content/uploads/papers/v8i1/A9236058119.pdf>
- Moonden, Y. (2011). Sistema de Producción Toyota: Un Enfoque Integrado para el Justo a Tiempo (4.ª edición). CRC Press, Taylor & Francis Group. <https://doi.org/10.1201/b11731>
- Nikitina, N., y Kajko-Mattsson, M. (2011). Transición del proceso Big Bang impulsada por el desarrollador. De Scrum a Kanban. *Actas del 2.º Taller de Ingeniería de Software para Aplicaciones de redes de sensores - SESENA '11*, 159-168. <https://doi.org/10.1145/1987875.1987901>
- Pantelić, O., Nikolić, U. y Krstović, S. (2020). Razvoj softvera za upravljanje projektima zasnovan na različitim agilnim metodologijama [Desarrollo de software de gestión de proyectos basado en diferentes metodologías ágiles]. *Časopis za informacionu tehnologiju i sisteme multimedia*, (71), 38-43 [Revista de tecnología de la información y multimedia] sistemas, (71), 38-43].
- Pap, J. (2008). Ekstremno programiranje kao metoda agilnog razvoja softvera [Extremo programación como método de desarrollo ágil de software]. [Tesis de posgrado, Universidad de Novi Sad, Facultad de Ciencias]. PQDT Abierto. [http://www.robertpapp.info/docs/tesis/PappRobert\\_XP\\_BSc-tesis.pdf](http://www.robertpapp.info/docs/tesis/PappRobert_XP_BSc-tesis.pdf)
- Parker, MEF y del Monte, YF (2014). La gestión ágil de proyectos de desarrollo de software que combina Scrum, Kanban y consultoría experta. *Actas del 10.º Conferencia Internacional IFIP sobre Sistemas de Código Abierto (OSS)*, 176-180. [https://10.1007/978-3-642-55128-4\\_25](https://10.1007/978-3-642-55128-4_25)
- Raman, S. (2 de abril de 2014). Programación Extrema La Metodología. InfoQ. <https://www.infoq.com/articles/implementando-xp-methodology/>
- Rising, L., Janoff, NS (2000). El proceso de desarrollo de software Scrum para equipos pequeños. *IEEE Software*, 17(4), 26-32. <https://10.1109/52.854065>
- Saleh, SM, Huq, SM y Rahman, MA (2019). Estudio comparativo dentro de Scrum. Kanban, XP enfocado en sus prácticas. *Actas de la Conferencia Internacional sobre*

Ingeniería Eléctrica, Computación y Comunicaciones (ECCE), 1-6.

<https://doi.org/10.1109/ECACE.2019.8679334>

Schwaber, K. (2004). Gestión ágil de proyectos con Scrum. Microsoft Press. ISBN: 978-0-7356-1993-7

Schwaber, K., y Sutherland, J. (2020). La Guía de Scrum. La guía definitiva de Scrum: Las reglas del juego. Scrum.org. [https://scrumguides.org/docs/scrumguide/v2020/2020-Guía Scrum de EE. UU. \(pdf\)](https://scrumguides.org/docs/scrumguide/v2020/2020-Guía%20Scrum%20de%20EE.%20UU.%20(pdf))

Schwaber, K., Beedle, M. (2002). Desarrollo ágil de software con Scrum. Upper Saddle Río, Nueva Jersey: Prentice Hall. ISBN: 0130676349

Senapathi, M., y DruryGrogan, ML (2021). Enfoque de pensamiento sistémico para la implementación Kanban: Un estudio de caso. Revista de Software: Evolución y Proceso, 33(4). <https://doi.org/10.1002/smr.2322>

Shafiq, S., e Inayat, I. (2017). Hacia el estudio de los patrones de comunicación de Kanban Equipos: Un diseño de investigación. Actas de los Requisitos Internacionales del 25.º Congreso IEEE de 2017 Talleres de la Conferencia de Ingeniería (REW), 303-306. <https://doi.org/10.1109/REW.2017.34>

Shalloway, A. (2011). Desmitificando Kanban. Cutter IT Journal, 24(3), 12-17.

Shenhar, JA, y Dvir, D. (2007). Reinventando la gestión de proyectos: El enfoque del diamante Hacia un Crecimiento Exitoso y la Innovación. Harvard Business Publishing

Sjøberg, DIK, Johnsen, A., Solberg, J. (2012). Cuantificar el efecto del uso de Kanban versus Scrum: Un estudio de caso. IEEE Software, 29(5), 47-53. <https://doi.org/10.1109/MS.2012.110>

Skinner, R., Land, L., Chin, W. y Nelson, RR (2015). Revisando el pasado para un futuro mejor. Futuro: Reevaluación de la retrospectiva de proyectos de TI. Actas de la 10.ª Conferencia Internacional Taller de investigación sobre gestión de proyectos de tecnología de la información (IRWITPM), 110-119. <https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1008&context=irwitpm2015>

Stellman, A., y Greene, J. (2017). Head First Agile: Una guía práctica. O'Reilly Media, Inc. ISBN: 9781449314330

Subih, MA, Malik, BH, Mazhar, I., ul-Hassan, I., Sabir, U., Wakeel, T., Ali, W., Yousaf, A., bin-Ijaz, B., y Nawaz, H. (2019). Comparación del método ágil y el método Scrum con Factores que afectan la calidad del software. Revista Internacional de Ciencias de la Computación Avanzadas y Aplicaciones (IJACSA), 10(5), 531-535.

[https://thesai.org/Descargas/Volumen10No5/Documento\\_69](https://thesai.org/Descargas/Volumen10No5/Documento_69)

Comparación entre el método ágil y el método Scrum.pdf

Sutherland, J. (2010). Manual de Scrum de Jeff Sutherland. Instituto de Formación Scrum.

[https://www.researchgate.net/publication/301685699\\_Manual\\_Scrum\\_de\\_Jeff\\_Sutherland](https://www.researchgate.net/publication/301685699_Manual_Scrum_de_Jeff_Sutherland)

Tadić, B. (2005). Ekstremno programiranje i primjena na Balkanu [Programación extrema y aplicación en los Balcanes]. Actas del 4. Naučno-stručni skup sa međunarodnim učešćem "KVALITET 2005" [4º Congreso científico-profesional con organizaciones internacionales participación "CALIDAD 2005"], 237-246.

<http://www.quality.unze.ba/zbornici/CALIDAD%202005/034-Q05-028.pdf>

Terlecka, K. (2012). Combinando Kanban y Scrum: Lecciones de un equipo de administradores de sistemas.

Actas de la Conferencia Agile (AGILE), 99-102.

<http://dx.doi.org/10.1109/Agile.2012.20>

Venkatachalam, A., Rajkumar, S., Selvanathan, N. y Karthick, K. (2017). SCRUM ágil –

La revolución en la industria de las TI. Actas de la Conferencia Internacional sobre Inteligencia Artificial.

Sistemas Computacionales (ICICS 2017 – 15-16 de diciembre de 2017), 228-238.

<http://dx.doi.org/10.2139/ssrn.3122169>

Verweij, D., y Massen, O. (2011). Kanban en una compañía de seguros en los Países Bajos.

Revista Cutter IT, 24(3), 18-22.

Vladimirovich Orlov, E., Mikhailovna Rogulenko, T., Alexandrovich Smolyakov, O.,

Vladimirovna Oshovskaya, N., Ivanovna Zvorykina, T., Grigorevich Rostanets. V., y

Petrovna Dyundik, E. (2021). Análisis comparativo del uso de Kanban y Scrum

Metodologías en proyectos de TI. Revista Universal de Contabilidad y Finanzas, 9(4), 693-700.

<https://10.13189/ujaf.2021.090415>

Vresk, A. (2020). Agilne metode za upravljanje projektima [Tesis de posgrado, Universidad de

Zagreb, Facultad de Organización e Informática]. PQDT Abierto.

<https://urn.nsk.hr/urn:nbn:hr:211:105099>

Womack, J., Jones, D. y Roos, D. (1990). La máquina que cambió el mundo. Rawson Asociados.

---

#### Acerca de los autores

Maja GABOROV es actualmente candidata a doctorado en la Facultad de Ciencias Técnicas, Universidad de Novi Sad. Trabaja como profesora asistente en el área de Información. Tecnología en la Facultad Técnica "Mihajlo Pupin" de Zrenjanin. Obtuvo su maestría. Licenciatura en la Facultad Técnica "Mihajlo Pupin" de la Universidad de Novi Sad en 2020. Su investigación Sus intereses incluyen el campo de la tecnología de la información.

Dijana KARUOVIĆ es doctora en ciencias de la información. Obtuvo su doctorado en Ingeniería Técnica. Ciencias en la Universidad de Novi Sad, Facultad Técnica "Mihajlo Pupin" en Zrenjanin, Serbia en 2010. Su principal campo de estudio es la interacción humano-computadora en el sistema de gestión del aprendizaje. Desde noviembre de 2000 trabaja en la Universidad de Novi Sad, Serbia, en el Departamento Técnico. Profesor asociado "Mihajlo Pupin" en Zrenjanin.

Mila KAVALIĆ defendió su tesis doctoral en la Facultad Técnica "Mihajlo Pupin" en Zrenjanin, Universidad de Novi Sad, Serbia en 2020. Actualmente es candidata a doctorado en la Facultad de Ciencias Técnicas de Novi Sad. Obtuvo su maestría en la Facultad Técnica. "Mihajlo Pupin" en 2014 y en la Facultad de Ciencias Técnicas de la Universidad de Novi Sad en 2017. Sus intereses de investigación incluyen la gestión del conocimiento, los recursos humanos y la organización. Comportamiento e Investigación Pedagógica.

Dragica RADOSAV es profesora titular (Departamento de Tecnología de la Información) y decana en la Facultad Técnica "Mihajlo Pupin" en Zrenjanin, Universidad de Novi Sad, Serbia. Ella defendió su tesis doctoral en la Facultad de Ciencias Técnicas de la Universidad de Novi Sad, Serbia, en 1994. Empleada en la Facultad Técnica de Zrenjanin desde 1983. Su investigación Los intereses incluyen Metodología de la Investigación Pedagógica, Metodología de la Investigación Científica, Ingeniería de Software, Diseño de Sistemas de Información y Sistemas de Información Geográfica.

Dragana MILOSAVLJEV es estudiante de doctorado en la Facultad Técnica "Mihajlo Pupin" en Zrenjanin, Universidad de Novi Sad. Anteriormente, completó una maestría en la Universidad Técnica Facultad "Mihajlo Pupin" en 2019 y Facultad de Ciencias Técnicas de Novi Sad en 2020. Trabaja como profesora asistente en la Facultad Técnica "Mihajlo Pupin", en la El estrecho campo científico de la gestión. Sus intereses de investigación incluyen la gestión del conocimiento, Cultura Organizacional, Relaciones Públicas y Gestión de Riesgos.

Sanja STANISAVLJEV recibió su doctorado en 2017 en la Facultad de Ciencias Técnicas de Novi Sad. Trabaja en la Universidad de Novi Sad, en la Facultad Técnica "Mihajlo". Pupin" en Zrenjanin como profesora del Departamento de Administración. Sus áreas de interés son

gestión, organización del trabajo, producción y gestión, gestión de la tecnología desarrollo.

Jozef BUSHATI es profesor titular en la Universidad de Shkodra "Luigj Gurakuqi", Albania. sostiene un Doctor en Tecnologías de la Información y la Comunicación y MBA (Facultad de Economía, Universidad de Tirana, Albania y Universidad de Nebraska Lincoln, Estados Unidos) y Licenciatura en Ciencias en la Universidad Instituto Pedagógico de Shkodra /hoy Universidad de Shkodra "Luigj Gurakuqi" en la Facultad de Ciencias Naturales en la rama de Matemáticas y Física. Sus publicaciones en International Los congresos y revistas se realizan en el área de aplicaciones de la Información y la Comunicación. Tecnología en Educación, Economía y Medio Ambiente, Aprendizaje Digital, etc.