Recibido: 08/09/2021
Aceptado: 24/11/2021
Publicado: 15/01/2022

Universidad & Ciencia
Vol. 11, No. 1, enero-abril (2022)
ISSN: 2227-2690 RNPS: 2450
http://revistas.unica.cu/uciencia

SISTEMATIZACIÓN DE LA LITERATURA EN METODOLOGÍAS ÁGILES DE DESARROLLO DE SOFTWARE

SYSTEMATIZATION OF SCIENTIFIC LITERATURE OF SOFTWARE DEVELOPMENT METHODOLOGY

Autores: Aivys Benitez Lavastida
Juan Miguel Sandó Lopetey
Institución: Universidad de Ciego de Ávila Máximo Gómez Báez, Cuba
Correo electrónico: aivys@unica.cu
juanms@sma.unica.cu

RESUMEN

El trabajo muestra los resultados de una sistematización de la literatura científica sobre metodologías ágiles de desarrollo de software. Se llevó a cabo con el objetivo general de seleccionar una metodología ágil para el desarrollo del Sistema informático para la gestión colaborativa de expertos y especialistas por áreas de conocimiento. Como resultado de la labor realizada se pudo constatar que existe una gran heterogeneidad de propuestas, no obstante, varios estudios coinciden en destacar que las metodologías ágiles *XP* y *Scrum* son las más utilizadas para obtener productos de software en el menor tiempo posible y elaborar la documentación necesaria. Con el propósito de aprovechar las potencialidades de ambas metodologías, se decide seleccionar la metodología SXP, en tanto combina las fases más ágiles de XP con las prácticas más ágiles de Scrum.

Palabras clave: Ingeniería de Software, Metodologías Ágiles, Proceso de Desarrollo de Software.

ABSTRACT

The work shows the results of a systematization of specialized scientific literature on agile methodologies of software development. It was focus with the general objective of selecting an agile methodology for the development of the "Computer System for the collaborative management of experts and specialists by areas of knowledge". As a result of the work carried out, it was found that there is a great heterogeneity of proposals, however, several studies agree in emphasizing that XP and Scrum Agile methodologies are the most used to obtain software products in the shortest possible time and elaborate the necessary documentation. In order to take advantage of the

Recibido: 08/09/2021
Aceptado: 24/11/2021
Publicado: 15/01/2022

potential of both methodologies, the author adopted the decision to select the SXP methodology, while this methodology combines the more agile phases of XP with the more agile practices of Scrum.

Keywords: Agile Methodologies, Software Engineering, Software Process Development.

INTRODUCCIÓN

Un elemento de vital importancia en el proceso de desarrollo de software lo constituye la selección de la metodología que se va a utilizar, porque la misma proporciona una serie de procedimientos, técnicas, herramientas y soporte documental que guiarán a los desarrolladores en la creación del nuevo producto.

La sistematización de la literatura científica revela la existencia de una gran variedad de metodologías de desarrollo de *software,* las cuales pueden ser clasificadas en dos grandes grupos: las llamadas *metodologías tradicionales/pesadas*, las cuales están orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y las notaciones que se usarán; y las llamadas *metodologías ágiles/ ligeras*, orientadas al desarrollo incremental del *software* y la interacción con el cliente, mostrándole versiones parcialmente funcionales del *software*, en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto en la medida en que se va desarrollando (Amaya, 2013; Chamba, 2020; González, Calero y Loaiza, 2019; Velásquez, Vahos y Gómez, 2019).

Como parte del Proyecto de Investigación titulado “Competencias digitales para la investigación y la formación continua de los profesionales cubanos”, que se desarrolla en la Universidad de Ciego de Ávila, se ha planteado como exigencia el desarrollo de un Sistema informático para la gestión colaborativa de expertos y especialistas por áreas de conocimiento. De ahí la necesidad de seleccionar una metodología de desarrollo de *software* ágil, que responda a las exigencias y especificidades del sistema informático que se pretende crear y al contexto propio de la investigación que se desarrolla.

En correspondencia con lo antes expuesto, en el presente artículo se plantea como **objetivo general**: seleccionar una metodología ágil para el desarrollo del Sistema

Recibido: 08/09/2021
Aceptado: 24/11/2021
Publicado: 15/01/2022

informático para la gestión colaborativa de expertos y especialistas por áreas de conocimiento a partir de la sistematización de la literatura científica.

DESARROLLO

En el estudio fue necesario el uso de diferentes métodos:

*Análisis y síntesis***:** para el estudio de la bibliografía científica especializada y la determinación de las características esenciales que identifican las metodologías de desarrollo de *software* ágiles*.*

Método histórico-lógico: para precisar los antecedentes y tendencias de las metodologías de desarrollo de *software* en general y de las ágiles en particular.

Método inductivo-deductivo: para avanzar de lo particular a lo general y viceversa, en la determinación de las características de las metodologías ágiles y elaborar conclusiones.

El *análisis documental y la sistematización de información*, para la organización, categorización, almacenamiento y procesamiento de la información obtenida de la literatura científica especializada, con el propósito de reflexionar sobre la misma y seleccionar una metodología ágil para el desarrollo del “Sistema informático para la gestión colaborativa de expertos y especialistas por áreas de conocimiento”.

La sistematización de la literatura científica especializada revela que son múltiples las investigaciones en las que se analizan las numerosas metodologías ágiles de desarrollo de *software* existentes a nivel internacional. Estudios analizados (Britto, 2016; González, Calero y Loaiza, 2019; Molina, Vite y Dávila, 2018; Pérez, 2016) coinciden en apuntar que las metodologías ágiles más comúnmente usadas son: *Scrum*, Programación Extrema (XP), *Cristal Methodologies*, *Kanban*, *Dynamic Systems Development Method* (DSDM), *Adaptive Software Development* (ASD) y *Feature-Driven Development* (FDD).

Teniendo en cuenta los criterios de diferentes autores, a continuación, se describen las metodologías anteriormente mencionadas:

Metodología Scrum

Esta metodología se estructuró para desarrollar productos en Japón. No se trata de un concepto nuevo, sino que fue basada en una estrategia utilizada en rugby en la que todos los integrantes del equipo actúan juntos para avanzar la pelota y ganar el partido. Escogieron dicho nombre por las similitudes que consideraban que existían

Recibido: 08/09/2021
Aceptado: 24/11/2021
Publicado: 15/01/2022

entre el juego del rugby y el tipo de proceso que proponían: adaptable, rápido y con pocos descansos (Takeuchi y Nonaka, 1986, Triviño, Angarita y Rojas, 2020).

Uno de sus principios es lograr la simplicidad y escalabilidad, ya que no establece prácticas de ingeniería del software, sino que se aplica o combina, fácilmente, con otras prácticas ingenieriles, metodologías de desarrollo o estándares ya existentes en la organización. Se concentra, principalmente, a nivel de las personas y equipo de desarrollo que construye el producto. Su objetivo es lograr que los miembros del equipo trabajen juntos y de forma eficiente, obteniendo productos complejos y sofisticados (Montoya Suarez, Sepúlveda Castaño, y Jiménez Ramos, 2016).

Indicada para proyectos con alto número de cambio de requerimientos, su principal característica es la definición de *sprints* (iteraciones), cada una de las iteraciones del proceso tiene una duración máxima de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente. Otra característica importante son las reuniones diarias que se llevan a cabo a lo largo del proyecto. Estas reuniones no tardan más de quince minutos y su objetivo son la coordinación e integración del producto a entregar (MendesCalo, Estevez, y Fillotrani, s.f.)

Metodología Programación Extrema (XP)

Es una metodología ágil centrada en potenciar el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, propiciando un buen clima y se basa en lo siguiente (Montoya Suarez, Sepúlveda Castaño, y Jiménez Ramos, 2016):

- XP es implementado adecuadamente para proyectos con requisitos imprecisos y requisitos muy cambiantes, y donde puede existir un alto riesgo en la parte técnica.
- XP tiene como principio la retroalimentación continua entre el cliente y el equipo de desarrollo, partiendo de una comunicación fluida entre todos los participantes, manejo de simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.
- XP trabaja y estructura una serie de valores y principios que deben tenerse en cuenta y practicarlos durante el tiempo de desarrollo que dure el proyecto.
- XP se considera una disciplina, la cual está sostenida por valores y principios propios de las metodologías.

Recibido: 08/09/2021
Aceptado: 24/11/2021
Publicado: 15/01/2022

Define un proceso iterativo e incremental con pruebas unitarias continuas y entregas frecuentes. El cliente es integrado al equipo de desarrollo. Recomienda que el desarrollo de las funciones del producto sea realizado por dos personas en el mismo puesto (programación por pares). Antes de incorporar una nueva funcionalidad, se debe corregir todos los defectos encontrados. Constantemente, se llevan a cabo pruebas de regresión, a fin de detectar los posibles errores (Beck, 2000).

Metodología Crystal

Se trata de un conjunto de metodologías para el desarrollo de software, caracterizadas por la valoración de las personas que componen el equipo de trabajo y la reducción al máximo del número de artefactos producidos. Potencia las habilidades de los integrantes y define políticas de trabajo en equipo.

Las políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo, *Crystal Clear* corresponde a equipos de tres a ocho integrantes y *Crystal Orange* en equipos de 25 a 50 integrantes (Montoya Suarez, Sepúlveda Castaño, y Jiménez Ramos, 2016)

Esta metodología tiene siete procesos que deben establecerse para cada proyecto, sólo tres son obligatorios; los otros cuatro es cuestión de prevención y seguridad. *Crystal* comparte con la metodología XP en cuanto a la orientación humana, pero esta centralización en las personas se hace de una manera diferente (Montoya Suarez, Sepúlveda Castaño, y Jiménez Ramos, 2016)

Los siete procesos son:

1. Entrega frecuente: Hacer pruebas del desarrollo que se definen en las etapas de entregas, los usuarios serán capaces de ofrecer información sobre los requisitos implementados, el cliente verá el progreso.
2. Mejora continua: Dar tiempo para que el equipo plantee dificultades y sobreponerse a los retos trazados para mejorar las cosas que no funcionan.
3. Comunicación: Tener todo el equipo tan junto (si es posible en el mismo puesto) con el fin de dar respuestas a las inquietudes y dificultades al instante.
4. Seguridad personal: Hacer que las personas que conforman el equipo se sientan seguras de hablar sin temor a represalias, que puedan dar críticas constructivas sobre el trabajo de otras personas y admitir sus propios errores, lo que lleva a la honestidad y, en última instancia, a la confianza.

Recibido: 08/09/2021
Aceptado: 24/11/2021
Publicado: 15/01/2022

5. Enfoque: Si todo el equipo tiene tiempo para centrarse en sus objetivos prioritarios para dos horas al día, durante dos días consecutivos cada semana, sin ningún tipo de distracciones sin perder el norte (como reuniones u otro trabajo), el equipo estará más enfocado en el trabajo.
6. Fácil acceso a usuarios expertos: Si los usuarios expertos están disponibles para el equipo, pueden responder preguntas y ofrecer retroalimentación sobre la calidad y el diseño de las decisiones a tomar.
7. Medio Ambiente, técnica de prueba automatizada, gestión e integración frecuente: Un entorno técnico adecuado, donde las tareas sean controladas por las pruebas y gestión de la configuración (hacer copias de seguridad y la fusión de los cambios).

Metodología Kanban

Es una técnica creada por Toyota, basada en una palabra japonesa que hace referencia a “tarjetas visuales”, y se utilizó para controlar el avance del trabajo, en el contexto de línea de producción. Se ha convertido en un sinónimo de la aplicación justo a tiempo (JIT), que está diseñada para el control de inventario y reducir los tiempos (Ordysiński, 2014; Torres, 2020).

Kanban tiene como objetivo principal gestionar de manera general, cómo se van completando las actividades. Aunque *Kanban* no fue creada para la gestión de proyectos de *software*, en la actualidad se usa esta metodología dentro de dicha área (Anderson, 2010; Torres, 2020).

Esta metodología de desarrollo recoge la experiencia del *Kanban* de Toyota y la lleva al mundo del desarrollo tecnológico, siendo su expresión más común un tablero dividido en columnas, las que señalan los estados de un flujo de trabajo y tarjetas que van señalizando cómo fluyen los requerimientos dentro de un equipo de *software*.

Se basa en las siguientes propiedades fundamentales (Kniberg …[et al.], 2010):

- Visualizar el flujo de trabajo.
- Limitar el trabajo en progreso.
- Medir y manejar el flujo de trabajo.
- Explicitar las políticas del proceso.
- Usar modelos para incrementar oportunidades de mejora.

Recibido: 08/09/2021
Aceptado: 24/11/2021
Publicado: 15/01/2022

Según Pérez (2016), algunas características claves que hacen destacar a *Kanban* son:

- Requiere poco aprendizaje para generar valor: La capacitación de una persona en el uso de *Kanban* se reduce a enseñarle los códigos propios del equipo. Como en *Kanban* las políticas del proceso son explícitas, el acceso y la comprensión de estas se simplifica y disminuye los tiempos necesarios para que la persona entre en régimen.
- Simple construcción y evolución: Al ser un tablero distribuido por zonas, armar un *Kanban* puede tomar menos de cinco minutos en su versión más simple. Cambiar la distribución del tablero es sencillo y permite que esta evolucione ajustándose a las necesidades que el equipo desee implementar en su flujo de trabajo. El tablero *Kanban* físico posee la flexibilidad necesaria para adaptarse a las necesidades del equipo que lo utilizará.
- Baja inversión inicial y operacional: Construir un tablero *Kanban* resulta ser muy barato, porque puede ser construido con materiales que se encuentran en cualquier oficina. Por ejemplo, una versión sencilla puede construirse sobre una muralla con cinta adhesiva de color para crear la distribución de zonas. Para operar un *Kanban* se necesita un lápiz y papeles de colores (*post-it*), que son de bajo costo en el mercado.
- Orientación a la generación de valor: La filosofía tras el *Kanban* sostiene que el flujo de las tareas debe ser sobre la base de halar de un estado a otro, concepto que se conoce como modelo *Pull*. Este modelo, junto a la priorización de las tareas basada en la cantidad de valor generado al cliente, garantiza que sólo se construya lo que se necesite, evitando la acumulación de tareas y el desperdicio de tiempo.

Dynamic Systems Development Method (DSDM)

El Método de Desarrollo de Sistemas Dinámicos (en inglés *Dynamic Systems Development Method*) es un método que provee un marco de trabajo para el desarrollo ágil de software. Cumple con las características generales de definir un proceso iterativo e incremental. Propone cinco etapas de desarrollo: estudio de viabilidad, estudio del negocio, modelado funcional, diseño y construcción, e

Recibido: 08/09/2021
Aceptado: 24/11/2021
Publicado: 15/01/2022

ISSN: 2227-2690 RNPS: 2450
http://revistas.unica.cu/uciencia

implementación. La iteración se produce en las tres últimas etapas, sin embargo, prevé retroalimentación en todas (Stapleton, 1997).

Adaptive Software Development (ASD):

Traducido al español Desarrollo Adaptable de Software, es un proceso iterativo, tolerante a cambios y orientado a las funcionalidades que el programa va a tener. Define tres etapas para el ciclo de vida: a) Especulación – se inicia el proyecto y se planifican las características del software; b) Colaboración – se desarrolla el producto; y c) Aprendizaje – se revisa la calidad del producto y se entrega al cliente. La revisión tiene como objetivo aprender de los errores cometidos y volver a iniciar el ciclo de desarrollo (Highsmith y Orr, 2000).

Feature-Driven Development (FDD):

Por su traducción al español Desarrollo Basado en Funciones es una metodología que define un proceso iterativo, con iteraciones cortas de dos semanas como máximo. El ciclo de vida consta de cinco pasos: a) Desarrollo de un modelo global, b) Construcción de una lista de funcionalidades, c) Planeación por funcionalidad, d) Diseño por funcionalidad y e) Construcción por funcionalidad (*Feature Driven Development*, s.f.).

Como puede notarse, aunque todas las metodologías ágiles antes descritas pretenden cumplir los postulados y principios del Manifiesto Ágil (Beck et al., 2001), existe una gran heterogeneidad entre las mismas, lo que provoca que la selección de una metodología por sobre las otras se convierta en un proceso sumamente engorroso; no se trata, por tanto, de juzgar cuál metodología es mejor o cuál es peor, el propósito esencial es determinar, cuál de las metodologías ágiles se adapta mejor a las exigencias y especificidades del contexto de investigación-desarrollo.

Teniendo en cuenta lo antes planteado, como parte de la sistematización de la literatura científica especializada se puede determinar que varios estudios, entre los que se destacan Amaya, 2013; Brito, 2016; Mendes, Estevez, y Fillottrani, s.f; Montoya, Sepúlveda y Jiménez, 2016 y Pérez, 2016, coinciden en reconocer que, entre las metodologías ágiles *Scrum* y Programación Extrema (XP) son las más utilizadas para desarrollar *software* de forma ágil. Es por ello que, a continuación, se presenta una comparación entre ambas metodologías, teniendo en cuenta los

Recibido: 08/09/2021
Aceptado: 24/11/2021
Publicado: 15/01/2022

resultados de varios estudios en los que se utilizan enfoques cualitativos y cuantitativos para la interpretación del análisis.

Al sistematizar la literatura científica especializada, se encontraron resultados de estudios que comparan las metodologías *Scrum* y XP, que pueden servir de base a la hora de adoptar una decisión sobre cuál utilizar en el desarrollo del Sistema informático para la gestión colaborativa de expertos y especialistas por áreas de conocimiento.

En uno de los resultados estudiados se proponen un marco de trabajo de evaluación, para medir de qué manera las metodologías *Scrum* y XP cumplen con los postulados del Manifiesto Ágil. Como resultado del estudio realizado, se plantea que:

En el primer postulado (P1): “Valorar al individuo y a las interacciones del equipo de desarrollo por encima del proceso y las herramientas” (Beck et al., 2001, p. 7), los autores consideran que, aunque “ambas metodologías valoran al individuo, definen roles y responsabilidades y reconocen la importancia y promueven la capacitación de los integrantes del equipo (Beck et al., 2001, p. 7), XP satisface P1 mejor que *Scrum*, ya que XP solo define actividades y entregables, mientras que *Scrum* define actividades, entregables y herramientas de desarrollo.

En el segundo postulado (P2): “Valorar el desarrollo de software que funcione por sobre una documentación exhaustiva” (Beck et al., 2001, p. 7), los autores plantean que “XP satisface P2 mejor que *Scrum*, ya que requiere que los incrementos entregados sean integrados en forma continua con el resto de las funciones” (Beck et al., 2001, p. 7), es decir, considera la integración parcial del software al finalizar cada iteración.

En el tercer postulado (P3): Valorar la colaboración con el cliente por sobre la negociación contractual (Beck et al., 2001), se coincide con los autores en que ambas metodologías satisfacen P3 de manera óptima, ya que “consideran al cliente como un miembro del equipo, que colabora desde la planificación de las iteraciones hasta la escritura de requerimientos y pruebas funcionales” (Beck et al., 2001, p. 9) y no utilizan la relación contractual para agregar valor al producto.

En el cuarto postulado (P4): Valorar la respuesta al cambio por sobre el seguimiento de un plan (Beck et al., 2001), los autores plantean que XP satisface P4 mejor que

Recibido: 08/09/2021
Aceptado: 24/11/2021
Publicado: 15/01/2022

Scrum, ya que XP ofrece mayor flexibilidad, al permitir incorporar cambios durante la iteración, mientras que en *Scrum*, aunque se permite introducir cambios, no se recomiendan en el sprint en curso. “De ser prioritario el cambio en el sprint en curso, se debe estimar nuevamente el esfuerzo requerido y si es necesario, quitar tareas del sprint ya planificado” (Beck et al., 2001, p.9).

En síntesis, los resultados del estudio de Mendes, Estevez, y Fillottrani, s.f, evidencian que “XP satisface los postulados ágiles mejor que *Scrum*” (Mendes, Estevez, y Fillottrani, s.f, p.9).

Otros dos estudios que ofrecen información relevante son los desarrollados por Ávila y Meneses (2013) y Qumer y Henderson (2006), los que, utilizando el método 4-DAT (4-Dimensional Analytical Tool) propuesto por estos dos últimos autores, realizan comparaciones entre las metodologías *Scrum* y XP. Es importante explicar que el método 4-DAT, facilita el examen de las metodologías ágiles desde cuatro perspectivas o dimensiones: **1)** alcance de la metodología, **2)** caracterización de la agilidad, **3)** valores ágiles (Manifiesto Ágil), **4)** caracterización del proceso de software.

Utilizando el método 4-DAT y basados en análisis cuantitativos y cualitativos para la comparación entre las metodologías *Scrum* y XP, el hallazgo fundamental de Qumer y Henderson (2006), es que, aunque XP tiene fases más ágiles que *Scrum*, esta última metodología tiene prácticas más ágiles que XP. En ello también coinciden Peñalver, Meneses y García (2010). Teniendo en cuenta lo anterior, se considera que pudiera ser más provechoso para el desarrollo del Sistema informático para la gestión colaborativa de expertos y especialistas por áreas de conocimiento. si se pudiera aprovechar las potencialidades de cada una de las metodologías objeto de análisis.

A una conclusión similar arribaron Peñalver, Meneses y García (2010) al proponer SXP, un híbrido cubano de metodologías ágiles que tiene como base las metodologías SCRUM y XP. A partir del estudio realizado sobre la metodología SXP, se consideró en el proyecto de investigación que esta metodología es adecuada para responder a las exigencias y especificidades de la investigación. A continuación, se exponen los argumentos que justifican la decisión de utilizar SXP

Recibido: 08/09/2021
Aceptado: 24/11/2021
Publicado: 15/01/2022

para el desarrollo del Sistema informático para la gestión colaborativa de expertos y especialistas por áreas de conocimiento.

Una de las razones fundamentales por las cuales se selecciona la metodología SXP es porque esta metodología combina las fases más ágiles de XP con las prácticas más ágiles de Scrum. Ello permite una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al cliente, pues es uno de los requisitos para llevar a vía de éxitos el proyecto.

Esta exigencia de SXP fue positivamente valorada en el proyecto de investigación en tanto ello facilita la activa participación del cliente en cada una de las etapas o iteraciones planificadas, lo que posibilita realizar la corrección adecuada y las transformaciones necesarias a cada funcionalidad con vista a lograr un producto mejor terminado y con la calidad requerida, provocándole el mínimo de insatisfacciones al cliente.

SXP está basada completamente en los valores y principios de las metodologías ágiles, expuestos en el Manifiesto Ágil. Consta de cuatro fases principales:

- Planificación-Definición: donde se establece la visión, se fija las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- Desarrollo: donde se realiza la implementación del sistema hasta que esté listo para ser entregado.
- Entrega: puesta en marcha.
- Mantenimiento: donde se realiza el soporte para el cliente.

De cada una de ellas se despliegan siete flujos de trabajo: *concepción inicial, captura de requisitos, diseño con metáforas, implantación, prueba, entrega de la documentación, soporte e investigación*, el cual se utiliza por el equipo de desarrollo cuando sea necesario, es decir, es un flujo que se puede mover y utilizarlo en cualquier parte del ciclo de vida del proyecto (Peñalver, Meneses y García, 2010).

> De estos flujos se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la lista de reserva del producto, definición de las historias de usuario, diseño, implementación, planificación de las iteraciones y las actividades que se van a realizar para lograr el producto, pruebas, además de las tareas necesarias para realizar las investigaciones para documentar todo el proceso (Peñalver, Meneses y García, 2010, p.338).

Recibido: 08/09/2021
Aceptado: 24/11/2021
Publicado: 15/01/2022

CONCLUSIONES

La sistematización de la literatura científica especializada muestra que entre la gran variedad de propuestas de metodologías ágiles de desarrollo de software, XP y Scrum son dos de las metodologías más utilizadas a nivel internacional. Como síntesis del trabajo realizado, se consideró conveniente aprovechar las potencialidades de ambas metodologías, por lo que se seleccionó SXP como metodología ágil para el desarrollo del Sistema informático para la gestión colaborativa de expertos y especialistas por áreas de conocimiento.

REFERENCIAS BIBLIOGRÁFICAS

AMAYA BALAGUERA, Y. D. (2013). Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual. *Revista de Tecnología/Journal Technology,* Vol. 12, No.2, pp.111-124.

ANDERSON, D. J. (2010). Kanban: successful evolutionary change for your technology. *Blue Hole Press*.

ÁVILA DOMENECH, E., y MENESES ABAD, A. (2013). Delfdroid y su comparación evaluativa con XP y Scrum mediante el método 4-DAT. *Revista Cubana de Ciencias Informáticas,* Vol. 7, No. 1, pp.16-23.

BECK, K. (2000). Extreme Programming Explained. Embrace Change. *Addison-Wesley*.

BECK, K., …[et al.] (2001). The Agile Manifesto. Disponible en: http://agilemanifesto.org Consultado: 15 de febrero de 2021.

BRITTO MONTOYA, J. A. (2016). Comparación de metodologías ágiles y procesos de desarrollo de software mediante un instrumento basado en CMMI. *Scientia et Technica,* Vol. 21, No. 2, pp.150-155.

CANÓS, J. H., LETELIER, P. y PENADÉS, M. C. (2003). Metodologías ágiles en el desarrollo de software. *Universidad Politécnica de Valencia*.

CHAMBA LÓPEZ, J. F. (2020). Estudio comparativo de metodologías ágiles para el desarrollo de aplicaciones SAAS. Tesis de Grado, Universidad Técnica de Machala, Machala.

Feature Driven Development. (s.f.). Obtenido de http://www.featuredrivendevelopment.com.

Universidad & Ciencia
Vol. 11, No. 1, enero-abril (2022)
ISSN: 2227-2690 RNPS: 2450
http://revistas.unica.cu/uciencia

GONZÁLEZ GONZÁLEZ, F., CALERO CASTAÑEDA, S. L. y Loaiza Buitrago, D. F. (2019). Comparación de las metodologías cascada y ágil para el aumento de la productividad en el desarrollo de software. *Ingeniería Industrial.* Repositorio institucional de la Universidad de Santiago de Cali, Colombia, pp. 1-11. Disponible en https://repository.usc.edu.co/handle/20.500.12421/1208 Consultado: 22 de mayo de 2021.

HIGHSMITH, J. y ORR, K. (2000). Adaptive Software Development. A Collaborative Approach to Managing Complex Systems. *Dorset House*.

KNIBERG, H. …[et al.] (2010). Kanban y Scrum–obteniendo lo mejor de ambos. *Prólogo de Mary Poppendieck & David Anderson. ESTADOS UNIDOS DE AMÉRICA*: C4Media Inc.

MENDES CALO, K., ESTEVEZ, E., y FILLOTTRANI, P. (s.f.). Un Framework para Evaluación de Metodologías Ágiles. *Universidad Nacional del Sur*.

MOLINA MONTERO, B., VITE CEVALLOS, H. y DÁVILA CUESTA, J. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espirales*, Vol. 2, No.17, pp. 114-121.

MONTOYA SUAREZ, L. M., SEPÚLVEDA CASTAÑO, J. M., y JIMÉNEZ RAMOS, L. M. (2016). Análisis comparativo de las metodologías ágiles en el desarrollo de software aplicadas en Colombia. *Gestión del Talento Humano: Enfoques y Modelos*, Vol. 3, No.19 pp. 450-464.

ORDYSIŃSKI, T. (2014). System Kanban w administracji publicznej. *Roczniki Kolegium Analiz Ekonomicznych/Szkola Glówna Handlowa,* No.33, pp. 421-435.

PEÑALVER, G., MENESES, A., y GARCÍA, S. (2010). SXP, SXP, metodología ágil para el desarrollo de software. *1er Congr. Iberoaméricano Ing. Proy*, pp. 333-344.

PÉREZ PÉREZ, M. J. (2016). *Guía Comparativa de Metodologías Ágiles.* (Tesis de grado), Universidad de Valladolid.

QUMER, A., y HENDERSON-SELLERS, B. (2006). Comparative evaluation of XP and Scrum using the 4D Analytical Tool (4-DAT). *European and Mediterranean Conference on Information Systems (EMCIS)*, (págs. 1-8). Costa Blanca, Alicante, Spain.

Recibido: 08/09/2021
Aceptado: 24/11/2021
Publicado: 15/01/2022

STAPLETON, J. (1997). DSDM Dynamic Systems Development Method: The Method in Practice. *Addison-Wesley*.

TAKEUCHI, H., y NONAKA, I. (1986). The new new product development game. *Harvard Business Review*, Vol. 64, No. 1, pp. 137-146.

TORRES VALENCIA, I. (2020). Estudio comparativo entre metodologías tradicionales y metodologías ágiles aplicadas a proyectos IT en entornoindustrial. Tesis de Maestría, Universidad Pública de Navarra, Pamplona, España.

TRIVIÑO PRECIADO, J. C., ANGARITA ROJAS, L. I., y ROJAS RUBIO, D. (2020). Diagnóstico de la aplicación de la metodología SCRUM en el área de tecnología de Terpel. Seminario de investigación especialización Gerencia de Tecnología, Universidad Escuela de Administración de Negocios, Bogotá.

VELÁSQUEZ RESTREPO, S. M., VAHOS MONTOYA, J. D., y GÓMEZ ADASME, M. E. (2019). Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software. *Cintex*, Vol. 24, No. 2, pp. 13-23.