

## Enfoque de aplicación ágil con Scrum, Lean y Kanban

### *Agile application approach with Scrum, Lean and Kanban*

José Gaete<sup>1</sup>    Rodolfo Villarroel<sup>1\*</sup>    Ismael Figueroa<sup>2</sup>  
Héctor Cornide-Reyes<sup>3</sup>    Roberto Muñoz<sup>4</sup>

Recibido 5 de agosto de 2020, aceptado 16 de octubre de 2020

*Received: August 5, 2020    Accepted: October 16, 2020*

### RESUMEN

En este artículo se presenta la revisión de tres principales enfoques ágiles existentes: *Scrum*, *Lean Software Development* y *Kanban*, las cuales sirvieron de base para la elaboración de una propuesta para un nuevo enfoque en el desarrollo ágil.

Para lograrlo, se realiza una descripción de cada enfoque ágil para el desarrollo de software y una comparativa para detectar sus puntos fuertes y débiles. Posteriormente, se establece la nueva propuesta a partir de la integración de los tres enfoques, definiendo un conjunto de métricas y desarrollando un caso de estudio para evaluar la integración y obtener datos cuantitativos y cualitativos. Los resultados obtenidos a partir del caso de estudio resultaron ser bastantes positivos porque nos permitió evaluar de forma integral el nuevo enfoque. Estos resultados representan un prometedor inicio para continuar trabajando en este camino.

Palabras clave: Enfoque ágil, *Scrum*, *Lean Software Development*, *Kanban*, gestión de proyectos ágil.

### ABSTRACT

*This paper presents the review of three main existing agile approaches: Scrum, Lean Software Development and Kanban, which served as the basis for the development of a proposal for a new approach in Agile development.*

*To achieve this, a description of each agile approach to software development and a comparison of their strengths and weaknesses are made. Then, the new proposal is established based on the integration of the three approaches, defining a set of metrics and developing a case study to evaluate the integration and obtain quantitative and qualitative data. The results obtained from the case study proved to be quite positive because it allowed us to comprehensively evaluate the new approach. These results represent a promising start to continue working in this line of research.*

*Keywords: Agile approach, Scrum, Lean Software Development, Kanban, Agile project management.*

---

<sup>1</sup> Pontificia Universidad Católica de Valparaíso. Escuela de Ingeniería Informática. Valparaíso, Chile.

E-mail: joseigarp@gmail.com; rodolfo.villarroel@pucv.cl

<sup>2</sup> Universidad de Valparaíso. Ingeniería en Información y Control de Gestión. Escuela de Auditoría. Valparaíso, Chile. Pragmatics Lab <https://pragmatics-lab.com>. E-mail: ismael.figueroa@uv.cl

<sup>3</sup> Universidad de Atacama. Departamento de Ingeniería Informática y Ciencias de la Computación. Copiapó, Chile. E-mail: hector.cornide@uda.cl

<sup>4</sup> Universidad de Valparaíso. Escuela de Ingeniería Civil Informática. Valparaíso, Chile. E-mail: roberto.munoz@uv.cl

\* Autor de correspondencia: rodolfo.villarroel@pucv.cl

## INTRODUCCIÓN

En la actualidad, los métodos ágiles continúan ganando espacio e importancia en la disciplina de la Ingeniería del Software porque han demostrado ser mucho más eficientes en ambientes de alta incertidumbre y cambios en comparación a los métodos tradicionales de desarrollo de software [3, 31]. La definición moderna del desarrollo ágil apunta a que los requisitos y soluciones evolucionan en el tiempo según las necesidades del proyecto y en donde la colaboración de equipos de trabajo es esencial para el cumplimiento de objetivos [13]. En la literatura científica, existe abundante evidencia que hacen visibles los beneficios del uso de métodos ágiles tales como *Scrum* [15, 17, 26], *Lean Software Development* [19, 20, 24] y *Kanban* [7, 18, 21]. Estas positivas evidencias han provocado que muchas empresas inicien procesos de adopción que permitan incorporar la agilidad en sus empresas [27] para conseguir que sus equipos alcancen altos estándares de trabajo y, que, en cada iteración, los clientes perciban que se les está agregando valor a su producto [9].

Los enfoques ágiles están alineados con los principios y valores declarados en el manifiesto ágil [16], donde es posible resaltar características como el énfasis en el valor del WIP (*Work in Progress*, Trabajo en Progreso), la responsabilidad de cada individuo durante el desarrollo y la actitud positiva frente a los cambios en los requerimientos [24]. Uno de los enfoques ágiles más aceptados y usados en la actualidad es *Scrum* [10, 27], definido como un marco de trabajo por el cual las personas pueden abordar problemas complejos adaptativos, a la vez entregar productos del máximo valor posible productiva y creativamente [15, 17]. *Lean Software Development*, es otro enfoque ágil notoriamente más flexible que *Scrum*, el cual ofrece un conjunto de recomendaciones altamente adaptables enfocadas principalmente a la eliminación de desperdicios y al mejoramiento del flujo de trabajo [18, 20, 24]. *Kanban* por su parte, es otro enfoque que se basa en una idea muy simple: el trabajo en curso debería limitarse y sólo se debería comenzar con algo nuevo cuando el bloque de trabajo actual haya sido entregado o haya pasado a la siguiente fase del flujo de trabajo [22, 30].

Este artículo elabora una propuesta de integración para un nuevo enfoque de aplicación ágil,

considerando las características más importantes de los tres enfoques mencionados anteriormente. En la literatura es posible encontrar información sobre la integración de enfoques *Scrum* con *Kanban* [1, 5, 28], reconocido bajo el nombre de *Scrumban*. También es posible identificar iniciativas que incorporan recomendaciones *Lean* al marco de trabajo *Scrum* [6, 23, 29]. La diferencia de esta investigación respecto a los trabajos existentes en la literatura es que la integración se realizará considerando tres enfoques ágiles. El resto del documento se estructura como sigue: en la segunda sección se presentan las características de los tres enfoques ágiles a integrar. En la tercera sección se describe la propuesta de integración de los enfoques. La cuarta sección expone el caso de estudio desarrollado. Posteriormente, en la quinta sección se exponen los resultados obtenidos. Finalmente, en la sexta sección se presentan las conclusiones y un panorama de posibles trabajos futuros en el tema del artículo.

## ENFOQUES ÁGILES: SCRUM, LEAN Y KANBAN

### **Scrum**

*Scrum* tiene sus orígenes a mediados de la década de los 90's, cuando Jeff Sutherland y Ken Schwaber adaptaron un estudio [11] sobre nuevas prácticas de producción al proceso de desarrollo de software. *Scrum* es un marco de trabajo que proporciona una serie de reglas y tareas específicas que se deben realizar en cada una de las iteraciones de un proyecto de software para asegurar la correcta elaboración de este [15]. Esto da como resultado que *Scrum* sea un enfoque liviano, muy fácil de aprender, pero muy difícil de dominar [17].

El marco de trabajo *Scrum* está conformado principalmente por los Equipos *Scrum* y sus roles, los eventos, los artefactos y sus reglas asociadas. Cada componente dentro del marco de trabajo sirve para un propósito específico y es esencial para el éxito de *Scrum* y para su uso [19].

La guía de *Scrum* es el documento que explica en detalle el mecanismo para aplicar el marco de trabajo, sus eventos (*Sprint*, *Sprint Planning*, *Daily Scrum*, *Sprint Review* y *Sprint Retrospective*), los roles (*Scrum Master*, *Product Owner* y *Development Team*) y artefactos (*Product Backlog*, *Sprint Backlog*

e Incremento). *Scrum* permite la creación de equipos auto organizados impulsando la co-localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

*Scrum* es un enfoque que proporciona una serie de reglas y tareas específicas que se deben realizar en cada una de las iteraciones de un proyecto de software para asegurar la correcta elaboración de este. Esto da como resultado un enfoque bastante sencillo de aplicar, puesto que sus reglas son claras y no se prestan a la subjetividad, pero también puede suponer una menor capacidad de adaptación a la hora de aplicarla.

Al analizar la literatura y experiencias en *Scrum*, se señala como una debilidad la rigidez que establece el marco de trabajo, lo que refleja poca flexibilidad en la adaptación a proyectos de desarrollo de software comparado con otros enfoques ágiles [28, 31]. Esto se puede explicar porque muchos profesionales consideran que sus reglas en general son bastante estrictas y muchos tienden a seguirlas al pie de la letra. Lo anterior se debe a que asumen que es la mejor forma de obtener el máximo provecho a un enfoque de desarrollo ágil. Sin embargo, en ocasiones puede ocurrir que alguna de las reglas *Scrum* no sea realmente necesaria para el desarrollo óptimo de un proyecto específico, lo que podría acarrear deficiencias que retrasen innecesariamente la elaboración de este, justamente lo que se desea evitar. Un ejemplo claro de esto son las *Daily Scrum* que, si bien es indudable su utilidad, en algunas circunstancias y dependiendo de la situación particular de cada proyecto, podrían realizarse con menor frecuencia.

### **Lean Software Development**

*Lean Software Development* (LSD) o simplemente *Lean*, es un conjunto de principios definidos por la industria de fabricación de automóviles de Japón en la década de los 80. El ingeniero de calidad de Toyota, John Krafcik, acuñó el término mientras observaba los procesos y herramientas utilizados para la eliminación de residuos en la producción de los automóviles. No fue sino hasta 2003 que María y Tom Poppendieck introdujeron *Lean* como un proceso de desarrollo de software en su libro "*Lean Software Development: An Agile Toolkit*" [19].

Los principios fundamentales de este enfoque son: Eliminación de desperdicios, amplificar el aprendizaje, decidir lo más tarde posible, entregar lo más rápido posible, potenciar al equipo, establecer integridad y visualizar siempre todo el proceso.

*Lean* ofrece una opción mucho más flexible a la proporcionada por *Scrum*, en la cual se entrega una serie de recomendaciones altamente adaptables cuyo objetivo radica en la generación de valor al cliente a través de la eliminación de residuos y entregas rápidas [19, 24]. Como se trata de recomendaciones en lugar de reglas, presentan un alto grado de subjetividad y cada equipo de trabajo deberá decidir cómo aplicarlas. Esto supone una mayor dificultad en la aplicación de este enfoque, pero al mismo tiempo, una mayor adaptabilidad y versatilidad en los desarrollos. El mayor problema que presenta *Lean* está muy relacionado con su incesante afán por generar valor al cliente, ya que, muchas veces para lograrlo será necesario acelerar ciertas entregas o prescindir de ciertos elementos o tareas importantes con tal de proporcionarle un valor al cliente lo antes posible [6,7]. Esto no es del todo malo, ya que gracias a las entregas rápidas el cliente puede estar más involucrado en los procesos de desarrollo y el equipo de trabajo podrá recibir un mayor feedback de él, pero a la vez, podría resultar en una disminución en la calidad del producto final.

### **Kanban**

El término japonés *Kanban*, fue empleado por Taiichi Ohno (Toyota), para referirse al sistema de visualización empleado en los procesos de producción que coordinan en una cadena de montaje la entrega a tiempo de cada parte en el momento que se necesita, evitando sobreproducción y almacenamiento innecesario de producto. El término aplicado a la gestión ágil de proyectos se refiere a técnicas de representación visual de información para mejorar la eficiencia en la ejecución de las tareas de un proyecto. Las principales reglas de *Kanban* son las siguientes: visualizar el flujo de trabajo, determinar el límite de trabajo en curso y medir el tiempo en terminar una tarea [18].

*Kanban* cumple el rol tanto de enfoque ágil como de herramienta y su objetivo principal es cumplir un conjunto de reglas definidas. Las reglas de *Kanban* son: Visualizar el flujo de trabajo, limitar el WIP, definir políticas explícitas, medir el flujo de trabajo

y mejora continua [30]. Al poseer características propias de una herramienta, es un enfoque bastante fácil de implementar, sólo basta con entender su funcionamiento y asegurarse que todos los miembros del equipo de trabajo lo apliquen correctamente. Además, gracias a sus cualidades de herramienta, es muy sencillo de integrar a otros enfoques.

El principal problema que presenta *Kanban* tiene que ver con WIP, ya que al ser aplicado de la forma que lo plantea *Kanban*, pueden generar cuellos de botella en el desarrollo. Lo anterior se explica de la siguiente forma: al trabajar con un tablero de *Kanban*, en cada una de las columnas se asigna un número que representa la máxima cantidad de tareas que puede haber en dicha columna, y si en algún determinado momento la columna se llena, no se podrá recibir nuevas tareas hasta que se finalice alguna de las tareas en ejecución [1, 5, 18]. Lo anterior es un gran problema puesto que, si alguno de las columnas de la tabla se llena, la columna anterior no podrá entregarle sus tareas finalizadas, lo que podría resultar en tareas inactivas, que son una gran ineficiencia para el proyecto.

### Comparación de los enfoques ágiles

En base a todo lo anterior y a la información existente en la literatura, es posible observar que existen una serie de características que podrían integrarse para elaborar una nueva propuesta de enfoque ágil. Algunos de los elementos comunes que es posible reconocer en *Scrum*, *Lean* y *Kanban* son:

- Se requiere un equipo de trabajo muy cohesionado.
- Es esencial que haya al menos un grupo de miembros experimentados en los equipos de trabajo, de modo que puedan ayudar y capacitar a los miembros con menos experiencia.
- Los miembros de los equipos de trabajo deben ser muy disciplinados, orientados al trabajo en equipo y con una gran capacidad de auto-organización.
- Se requiere una constante comunicación con el cliente.
- El cliente debe estar dispuesto a tener un rol participativo en los proyectos.
- Los tres son enfoques ágiles por lo que comparten los principios básicos del desarrollo ágil.

La Tabla 1 presenta una comparativa en base a factores comparativos definidos.

En base a la comparación realizada y a los análisis efectuados se han detectado los siguientes puntos:

- *Scrum* proporciona un marco de trabajo completo y específico que no se presta a ambigüedades ni a errores de interpretación. Los equipos de trabajo saben exactamente cómo deben realizar cada iteración, cuáles son los pasos por seguir, qué reglas deben respetar, cual es el rango de tiempo que deberían asignar para la realización de cada actividad, etc. Sin embargo, el gran problema de este enfoque es que su estructura puede provocar una pérdida de flexibilidad.
- *Lean* por su parte, es un enfoque más complejo de implementar que *Scrum*, ya que sólo entrega recomendaciones, no instrucciones específicas. Esto da como resultado que se preste bastante a la subjetividad y a la interpretación de cada equipo de trabajo, pero lo compensa con una gran versatilidad, la posibilidad de crear software más económico y en menor tiempo y la opción de tener una relación mucho más cercana con el cliente gracias a las constantes entregas y generación de valor.
- Finalmente, *Kanban* se presenta como un enfoque que además de establecer una serie de propiedades, proporciona una herramienta con la que se puede mejorar notablemente la visualización del flujo de trabajo dentro de los proyectos. Esto permite que *Kanban*, al igual que *Scrum*, sea bastante sencillo de implementar y, además, bastante sencillo de integrar con otros enfoques. Un posible problema en la implantación de este enfoque radica en que *Kanban* puede producir cuellos de botella a la hora de ser aplicado, lo cual representa una gran deficiencia para el desarrollo.

### PROPUESTA DE INTEGRACIÓN

Para realizar la integración de los tres enfoques ágiles, se ha decidido elaborar una propuesta considerando las siguientes características:

- Se tomará el enfoque clásico de *Scrum* como base y se integrarán algunas características ofrecidas por *Lean* y *Kanban* con el objetivo de mejorarla. A este nuevo enfoque ágil le llamaremos “*Scrum* mejorado”.
- De *Lean* se utilizarán sus siete recomendaciones básicas, para mejorar la flexibilidad ofrecida por *Scrum*. De esta forma, a la hora de aplicar el nuevo enfoque, no se ejecutarán todas las

Tabla 1. Comparativa de los enfoques ágiles *Scrum*, *Lean* y *Kanban*.

Factores Comparativos	Scrum	Lean	Kanban
Nivel de interacción con el cliente	Alto.	Alto.	Medio.
Complejidad de uso	Media.	Alta.	Media.
Adaptabilidad	Media-Baja.	Alta.	Alta.
Características del equipo de trabajo	Altamente Cohesionado.	Altamente Cohesionado.	Altamente Cohesionado.
Cualidades necesarias para los miembros del equipo de trabajo	Disciplinados, autoorganizados, orientados al trabajo en equipo, altamente motivados.	Disciplinados, autoorganizados, orientados al trabajo en equipo, altamente motivados.	Disciplinados, autoorganizados, orientados al trabajo en equipo, altamente motivados.
Objetivo principal	Brindar un marco de trabajo que ayude a agilizar los procesos que conforman un proyecto de software, utilizando una serie de reglas y tareas específicas divididas en iteraciones.	Generar la mayor cantidad posible de valor al cliente a través de sugerencias enfocadas a la eliminación de residuos y entregas rápidas.	Proporcionar un conjunto de normas que permitan mejorar la visualización de las tareas y el flujo de trabajo a través de la utilización del tablero Kanban.
Método de aplicación	El proyecto se divide en Sprints. Los Sprints representan la iteración del proyecto y están conformados por un conjunto de tareas y reglas específicas.	El proyecto se divide en un conjunto de iteraciones cortas enfocadas a entregar rápidamente avances beneficiosos para el cliente y procurando eliminar todo lo que no sea útil para este.	Las tareas del proyecto se exponen en un tablero visible para todos los miembros del equipo de trabajo. El tablero se deberá mantener actualizado y tendrá reglas específicas.
Principal ventaja	Se trabaja en iteraciones cortas denominadas Sprint, lo que permite tener un alcance acotado y viable. Además, al finalizar cada Sprint, el cliente recibirá un incremento funcional del producto final.	La eliminación de los residuos conduce a la eficiencia global del proyecto, lo que se traduce en una reducción de costos y tiempo. Además, la entrega temprana permite que se genere valor al cliente lo antes posible.	Otorga información siempre actualizada sobre el estado actual del proyecto y permite saber en todo momento quién está trabajando en qué, lo que favorece la organización y distribución de las tareas a realizar.
Principal desventaja	Comparativamente a otros enfoques ágiles, Scrum presenta una gran rigidez relacionada principalmente con sus reglas, las cuales suelen ser bastante estrictas.	Para poder generar rápidamente valor al cliente, a veces será necesario prescindir de alguna tarea o adelantar alguna entrega, lo que puede resultar en una disminución en la calidad.	Debido a la propiedad de limitación del trabajo en curso, pueden producirse cuellos de botella durante las etapas de desarrollo.

reglas de *Scrum* de manera tan estricta, sino que se flexibilizará su aplicación utilizando los principios de *Lean*. Para que esto funcione, es esencial estar siempre al tanto de qué cosas son las que realmente le generan valor al cliente y cuáles pueden considerarse como residuo, de modo de descartar todo lo que no sea estrictamente necesario para cumplir los objetivos de cada Sprint. Además, se adherirán nuevas recomendaciones para complementar las recomendaciones básicas de *Lean* y así mejorar la integración de los enfoques.

- *Kanban* por su parte aportará a la propuesta en su rol de herramienta, por lo cual se utilizará un tablero *Kanban* para manejar todas las actividades que se estén ejecutando en cada Sprint. Esto último, sin embargo, no resuelve el gran inconveniente de cuello de botella de *Kanban*, por lo que, para solucionar este problema, se propone una estrategia de manejo dinámico de recursos. La Figura 1 detalla la Visión General de la Propuesta.

Con el objetivo de mejorar la integración de *Lean* dentro del nuevo enfoque propuesto, se plantean 3

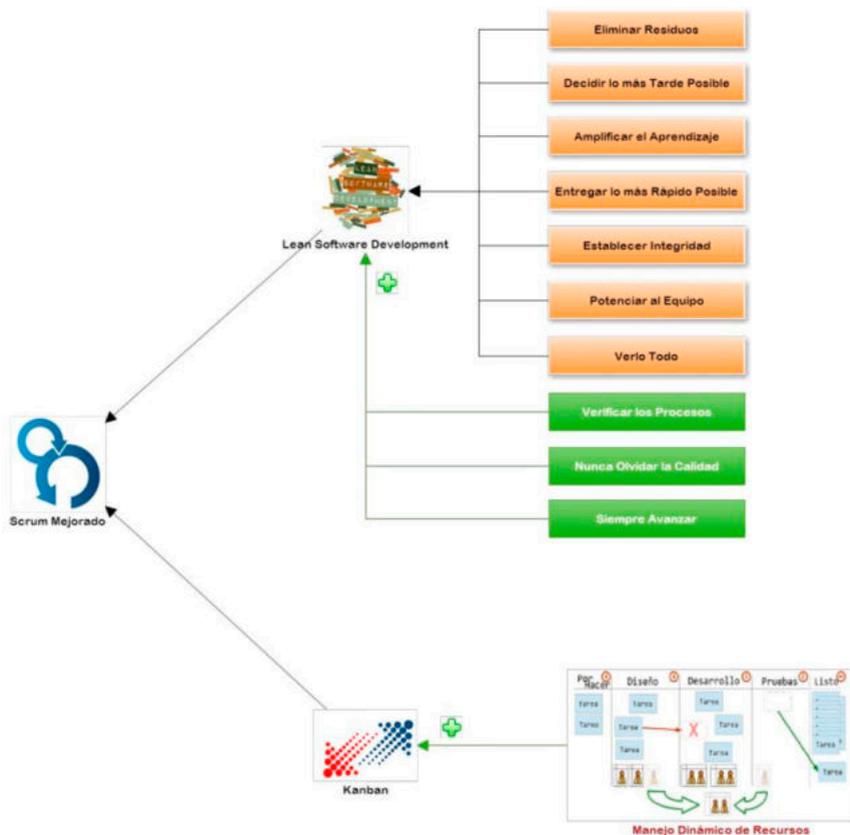


Figura 1. Visión General de la Propuesta.

nuevas recomendaciones que complementarán a las recomendaciones básicas de *Lean*:

- **Verificar los Procesos:** Para respetar el principio más importante de *Lean*, es decir, eliminar los residuos, se recomienda que ninguno de los procesos establecidos en el enfoque sea considerado de carácter obligatorio. Siempre se debe priorizar la generación de valor, y si se detecta que alguna de las actividades o tareas que conforman el enfoque interfiere con este objetivo, se recomienda eliminarla.
- **Nunca Olvidar la Calidad:** Siempre que se decida eliminar algún elemento que a simple vista no esté generando valor al cliente, antes de llevar a cabo la eliminación se recomienda verificar cómo repercutirá esta acción en la calidad final del producto y decidir si realmente vale la pena hacerlo.
- **Siempre Avanzar:** No se recomienda en ninguna circunstancia detener las actividades o tareas que se están realizando. Cualquier actividad ociosa o cuello de botella que se produzca,

puede suponer una notable disminución en la eficiencia del desarrollo, lo cual implica un enorme residuo para el proyecto.

Para combatir el problema de la generación de cuellos de botella que tiene *Kanban*, se ha propuesto una estrategia de manejo dinámico de recursos. La explicación completa de esta estrategia se expone a continuación: Para explicar la estrategia propuesta, se ha diseñado un escenario correspondiente a un proyecto de software cualquiera, en el que se expondrá un ejemplo explicativo que permita al lector entender de la mejor forma posible cómo se generan los cuellos de botella en *Kanban* y de qué forma se pretende solucionar.

En la Figura 2, se muestra el tablero *Kanban* de un proyecto en el que hay 3 personas que trabajan en diseño, 2 pares que trabajan en desarrollo y un tester. Las tareas sólo pueden pasar a la siguiente columna cuando se hayan completado. Si un diseñador termina una tarea, esta se mueve hacia

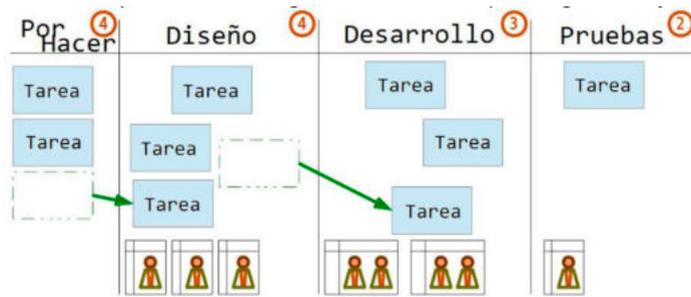


Figura 2. Propuesta Kanban - parte 1.

desarrollo y el diseñador coge una nueva tarea para seguir trabajando.

Este es el proceso normal, pero ¿qué pasa si se llega al máximo de tareas por columna? Si un diseñador termina una tarea, pero en desarrollo todavía no se han terminado, se tendrían 4 tareas en el carril de desarrollo: una situación no deseada. La Figura 3 muestra la situación expuesta.

En este caso lo que se debe hacer es reasignar los recursos. Lo primero que intentaría realizar el diseñador es entregar la tarea que acaba de terminar a Desarrollo, pero no puede hacerlo porque el carril está al máximo de su capacidad. Su única opción es trabajar en alguna de las tareas de desarrollo. Puede que no sea el mejor desarrollador, pero sus esfuerzos ayudarán a mantener el flujo del proceso. Bajo la misma idea, si el tester, que sólo tenía una única tarea, la termina, podría también ayudar a los desarrolladores en sus tareas. Cabe recalcar que para que esta solución dé resultado, hay que tener mucha precaución con respecto a las tareas que se le asignará a cada persona a la hora de ser cambiado de un carril a otro. La idea es que la persona que realice el cambio, lo haga sólo en

carácter de apoyo, no es recomendable que se le asignen tareas demasiado complejas en los otros carriles, sobre todo si la persona no es experta en la actividad en cuestión. La Figura 4 muestra el escenario descrito.

Finalmente, una vez que alguna tarea en la columna de desarrollo se haya completado, se puede regresar a la situación de normalidad, el tester volverá a su columna para poder probar la nueva tarea desarrollada y el diseñador comenzará a trabajar en una nueva tarea de diseño, como se observa en la Figura 5. Se debe tener en cuenta que esto no es una receta exacta, cada uno puede elegir qué hacer en estos casos.

### Requisitos para la utilización de la propuesta

Para que la propuesta pueda ser aplicada correctamente, deben cumplirse una serie de requisitos, los cuales se listan a continuación:

- Al igual que los enfoques que la conforman, la propuesta deberá cumplir con las mismas características explicadas en la sección “Comparación de los enfoques ágiles”.
- Para que la estrategia de manejo dinámico de recursos de Kanban funcione correctamente, es

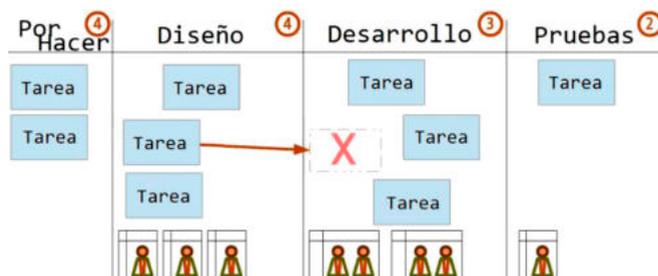


Figura 3. Propuesta Kanban - parte 2.

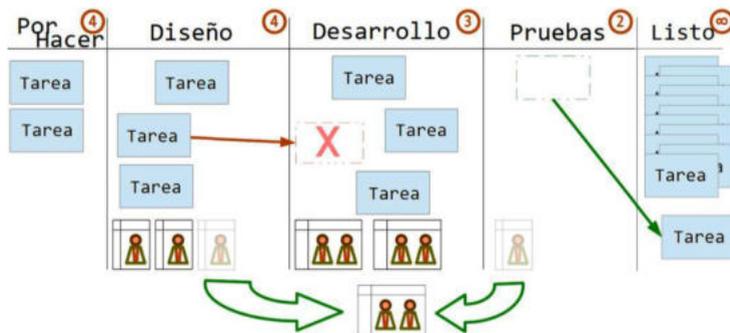


Figura 4. Propuesta Kanban - parte 3.

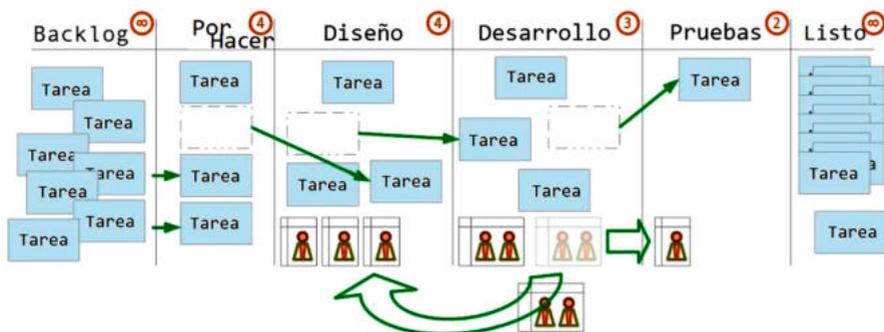


Figura 5. Propuesta Kanban - parte 4.

necesario que todos los miembros del equipo de trabajo tengan al menos un conocimiento básico de cada una de las fases que conforman el flujo de trabajo de un proyecto de software.

- Dentro del enfoque propuesto deberá existir un nuevo rol que se encargue de velar que las recomendaciones de *Lean* no se dejen de tomar en cuenta en ningún momento. Este nuevo rol será denominado *Lean Master* y deberá ser una persona de confianza para los miembros del equipo de trabajo, con un gran liderazgo y una gran capacidad para entender lo que el cliente necesita. Este nuevo rol deberá participar activamente en todos los procesos que conforman la elaboración de los proyectos, desde la definición del *Product Backlog* hasta la entrega final del producto.

**Definición de métricas**

Luego de todos los análisis efectuados y las deducciones obtenidas a partir de ellos, es esencial que se reúnan todos estos datos empíricos y se lleven a la práctica, con el fin de comprobar si la propuesta planteada representa realmente una

contribución para el campo del desarrollo ágil. Para esto, se han formulado una serie de métricas que permitirán recabar datos específicos enfocados al análisis de las principales características de la propuesta, de modo de verificar su comportamiento al momento de ser implementada. A continuación, se describen las métricas que se utilizaron para la medición y comprobación final de los resultados de esta investigación. Para seleccionar las métricas a utilizar, primero se analizaron las métricas más utilizadas en contextos de desarrollo ágil [6, 8, 12, 25], luego, se analizó cuáles de ellas ofrecen un beneficio real para los objetivos que se persiguen en este trabajo y, finalmente, se seleccionaron las más adecuadas para la investigación. Además, se ha diseñado una nueva métrica creada específicamente para la medición del manejo dinámico de recursos que plantea el nuevo enfoque. Las métricas cuantitativas definidas para la propuesta son:

1. **Lead Time:** Registra el tiempo que transcurre entre el momento en que se solicita el desarrollo de algún ítem de trabajo del *Product Backlog* hasta el momento en que finalmente dicho ítem de trabajo es entregado. El objetivo de esta

métrica será evaluar qué tanto debe esperar un cliente para poder obtener avances reales durante la ejecución de un proyecto (obtener valor), y para implementarla se utilizarán los días de trabajo como unidad de medida [6].

2. **Touch Time:** Registra el tiempo en el cual un ítem de trabajo es realmente trabajado (o tocado) por el equipo, es decir, mide el tiempo real que tarda el equipo de trabajo para finalizar algún ítem, descontando todos los posibles períodos de inactividad que puedan producirse en él a lo largo del proceso. La unidad de medida utilizada para esta métrica serán las horas de trabajo [14].
3. **Velocidad:** Permite visualizar la velocidad de avance del equipo de trabajo a través de la cantidad de requerimientos o ítems de trabajo completados en cada iteración. Esta métrica permite medir la productividad del equipo de trabajo, ya que mientras mayor sea la cantidad de ítems de trabajo que sean capaces de terminar por iteración, menor será el tiempo necesario para finalizar el proyecto [8].
4. **Requisitos no Completados:** Registra la cantidad de requerimientos que no se pudieron completar en cada una de las iteraciones del proyecto pese a haber sido establecidos como objetivo de la iteración. Esta métrica permitirá evaluar qué tan bien se están cumpliendo los objetivos establecidos al inicio de cada Sprint y, además, evaluar el desempeño del equipo de trabajo [8].
5. **Velocidad de Estabilización:** Registra los eventos de sobrecarga de trabajo que se hayan producido a lo largo de cada iteración. Esta métrica está diseñada específicamente para comprobar la funcionalidad del manejo dinámico de recursos que se plantea en la propuesta y su objetivo principal es medir la eficiencia del equipo de trabajo para sobreponerse a estos cuellos de botella.
6. **Eficiencia:** La eficiencia está relacionada con el cumplimiento de las tareas con el mínimo gasto de recursos que sean posibles. Para el caso en estudio, resulta principalmente relevante verificar el gasto de tiempo en relación con las tareas realizadas, para lo cual se pondrá especial énfasis a aquellas tareas que presentan características atípicas, ya que a través de ellas se podría llegar a obtener información interesante incluso fuera del ámbito de la eficiencia.

Derivado del análisis de la literatura realizado, también fue posible identificar e integrar algunas métricas cualitativas que ayudarán a la consecución de los objetivos definidos en esta investigación. Las métricas cualitativas consideradas en la propuesta son:

1. **Satisfacción del Equipo de Trabajo:** Esta métrica está ligada principalmente a los desarrolladores. Se espera obtener información importante respecto a lo útiles que les puedan resultar las herramientas que el enfoque les proporciona, el ambiente de trabajo que se genera, entre otros. Para recabar estos datos, se realizarán una serie de encuestas y/o entrevistas que deberán ser respondidas por los miembros de los equipos de trabajo.
2. **Facilidad de Adopción:** Esta métrica tiene como objetivo evaluar la facilidad con que el enfoque puede ser implementado, así como también la facilidad de aprendizaje y la facilidad de uso. Para evaluar estos conceptos, al igual que para la métrica anterior se utilizarán encuestas y/o entrevistas que permitan recabar los datos necesarios.
3. **Organización y Planificación:** Permite evaluar qué tan bien la metodología ayuda a organizar las distintas tareas que se deben realizar en cada iteración de un proyecto y qué tanto favorece el enfoque a la planificación. Esta métrica también deberá ser evaluada a través de encuestas y/o entrevistas.

## CASO DE ESTUDIO

### Descripción de la asignatura y alumnos participantes

Para validar los componentes de la propuesta de integración de enfoques ágiles, se desarrolló una experiencia con un grupo de 42 alumnos pertenecientes a la carrera Ingeniería Civil Informática que cursaban la asignatura de Ingeniería Web impartido por la Escuela de Ingeniería Informática de la Pontificia Universidad Católica de Valparaíso. Ingeniería Web es una asignatura de carácter obligatorio, correspondiente al área de Ingeniería Aplicada. En este curso, se les pidió a los alumnos desarrollar un proyecto semestral en grupos de 3 a 6 integrantes. Para esta actividad, se conformaron 7 grupos y los alumnos comenzaron utilizando el enfoque clásico de *Scrum* con el objetivo de que pudieran interiorizarse con las

características generales de un desarrollo ágil, puesto que muchos de ellos jamás habían trabajado con este tipo de enfoques. Posteriormente, se agregaron las características adicionales correspondientes a la propuesta explicada en este documento, de modo que los alumnos pudiesen tener una experiencia real al utilizarla.

Para llevar a cabo este estudio, el primer paso consistió en realizar una reunión con los alumnos del curso Ingeniería Web, en la que se explicó en detalle todos los aspectos de la propuesta diseñada, para que ellos implementaran sus respectivos proyectos. Además, se explicó el funcionamiento de la herramienta JIRA, de modo que fueran capaces de utilizarla sin problemas a la hora de trabajar con ella.

### **Recolección de datos**

Para reunir los datos necesarios para las métricas cuantitativas, se trabajó con una herramienta basada en web denominada JIRA, la cual se enfoca en el seguimiento de errores, incidencias y gestión operativa de proyectos. El objetivo era que los estudiantes utilizaran esta herramienta para organizar el flujo de trabajo de sus proyectos, de manera de poder utilizar los datos ingresados por ellos para construir cada una de las métricas requeridas. El motivo por el cual se escogió JIRA es porque, entre sus funciones, existe la posibilidad de generar diversos informes ágiles que permiten visualizar el estado de los proyectos, la eficiencia de los equipos de trabajo, la cantidad de tareas ejecutadas en cada iteración, entre otros; lo cual facilita enormemente la creación de las métricas. Además, para la organización del flujo de trabajo, JIRA ofrece una interfaz completa y amigable que denomina pizarras, las cuales son altamente configurables y adaptables, gracias a lo cual fue posible crear pizarras con características *Kanban* para que los estudiantes pudieran utilizarlas a lo largo del desarrollo de sus proyectos, permitiendo así la posterior comprobación del manejo dinámico de recursos.

Cabe señalar que JIRA no fue el único elemento utilizado para la captación de los datos cuantitativos, ya que, si bien es una herramienta muy útil, en algunas ocasiones la información proporcionada por esta puede resultar algo confusa, especialmente si los usuarios (en este caso, los alumnos) no la

utilizan adecuadamente. Por lo anterior, se tomó la precaución de solicitar a los estudiantes que a lo largo del semestre realizaran presentaciones orales (*Sprint Retrospective*) al finalizar cada iteración, en las cuales debían explicar la experiencia adquirida hasta ese punto, las dificultades que debieron enfrentar y los beneficios que les aportó el enfoque utilizado, así como los objetivos que perseguirán en la siguiente iteración. Además, dicha presentación debía estar acompañada de un breve video complementario en el que se explicarán los detalles respecto a las historias de usuario que hayan colocado en JIRA a lo largo de la iteración.

Para el caso de las métricas cualitativas, se utilizaron encuestas creadas a través de la aplicación web *Google Form*, ya que brinda una mayor libertad a los estudiantes a la hora de responder cada encuesta gracias a que pueden hacerlo desde cualquier dispositivo con acceso a internet. Además, *Google Form* permite generar automáticamente gráficos porcentuales con las respuestas de los encuestados, lo cual facilita bastante la posterior evaluación de los resultados. Se diseñaron dos encuestas, la primera se aplicó al inicio del semestre con el propósito de evaluar los conocimientos generales de los encuestados respecto a los enfoques ágiles, mientras que la segunda encuesta, se aplicó una vez finalizados los proyectos, y tuvo como objetivo evaluar sus experiencias y apreciaciones respecto al enfoque propuesto.

## **RESULTADOS OBTENIDOS**

Tal y como fue mencionado en la sección anterior, la actividad fue desarrollada por siete grupos de alumnos de la asignatura Ingeniería Web que desarrollaron un proyecto de desarrollo de software, utilizando la propuesta de integración desarrollada. A continuación, se analizarán, desde el punto de vista de las métricas cuantitativas, los valores obtenidos durante la recolección de datos.

1. **Análisis *Lead Time*:** Al observar los valores de *Lead Time* en las incidencias de los diferentes grupos, se pueden apreciar las siguientes características: Algunos grupos ingresaron la mayor parte de las incidencias de su proyecto al inicio de este y las fueron resolviendo poco a poco en cada iteración, lo que demuestra una buena planificación inicial. Otros grupos hicieron justo lo contrario, al principio sólo

crearon unas pocas incidencias y a medida que avanzaron en el proyecto fueron planteando nuevas, lo que demuestra que su estilo de planificación fue más progresivo. Gran parte de los grupos planificaron incidencias que no pudieron resolver en el tiempo asignado, lo que es un claro error de planificación. En la Tabla 2 es posible observar los datos para cada grupo de trabajo. El *Lead Time* promedio entre todos los grupos es de 45.7 días, lo cual corresponde aproximadamente al 68.2% del ciclo de vida de los proyectos, por lo tanto, desde el punto de vista del cliente, los requerimientos tardaron mucho en resolverse.

2. **Análisis *Touch Time*:** Los valores correspondientes al *Touch Time* de las incidencias presentan las siguientes características a destacar: Se aprecia una discrepancia bastante grande entre los valores de *Lead Time* y *Touch Time*, lo cual indica que, en muchas ocasiones, las incidencias permanecieron en el *Touch Time* por largos períodos, lo cual es una clara ineficiencia en el desarrollo. Se aprecia también una gran diferencia entre los valores del *Touch Time* de un grupo a otro, habiendo grupos que llegaron a ocupar más de 100 horas en la resolución de una incidencia y otros que sólo tardaron 5 horas como máximo. Esto es un claro indicio de que, a la hora de calcular las horas trabajadas, algunos grupos utilizaron sus propios criterios subjetivos, ya que es prácticamente imposible que se presente una desigualdad tan abismal entre grupos de trabajo con un nivel de conocimiento relativamente similar. En promedio, los grupos presentaron un *Touch Time* de 15.6 horas, es decir, menos de 1 día por incidencia, y considerando que en promedio los grupos registraron unas 13 incidencias, significa que en unos 8 días y medio podrían haberlas terminado todas, lo que al ser comparado con los 45.7 días de promedio del *Lead Time*, demuestran una gran cantidad de tiempo no aprovechado. En la Tabla 2, se aprecian los datos recopilados para el grupo 1 y los valores obtenidos para las métricas *Lead Time* y *Touch Time*.

3. **Análisis Velocidad y Requisitos no Completados:** Al observar los datos correspondientes a las métricas de velocidad y requisitos no completados, se observa lo siguiente: Pese a la capacitación, los alumnos no fueron capaces

Tabla 2. Cálculo *Lead Time*/*Touch Time* Grupo 1.

Nº Iteración	Nº Historia	Tiempo utilizado (Hrs.)	Lead Time (Días)	Touch Time (Hrs.)
1	1	40	38	40
	2	25	38	25
	3	50	29	50
	4	65	24	65
	5	30	17	30
	6	45	24	45
	7	20	8	20
2	7	20	51	20
	9	110	40	110
	10	70	48	70
3	11	60	17	60
	12	45	51	45
	13	45	1	45

de utilizar adecuadamente la herramienta web JIRA, dificultando bastante el análisis de los datos. En este caso, se tuvo que recurrir a la información obtenida a través de los *Sprint Retrospective* para determinar los verdaderos valores correspondientes a las métricas de velocidad y requisitos no completados. A nivel general, los grupos demuestran una gran eficacia a la hora de resolver sus incidencias (ver Tabla 3), sólo dos equipos tuvieron incidencias no resueltas, pero de estos, hubo uno que presentó una mayor cantidad de incidencias no resueltas en comparación a las que sí se resolvieron, lo que indica que dicho equipo no está cumpliendo los objetivos que tenían planificados. A través de la velocidad es posible determinar la productividad del equipo de trabajo, ya que, en situaciones normales, a medida que el tiempo avanza el equipo va adquiriendo una mayor experiencia y entendimiento del proyecto, lo que les permite resolver un mayor número de incidencias por iteración. En este caso y, como es posible observar en la Tabla 4, ninguno de los grupos de trabajo cumplió con esta premisa, lo que refleja una baja productividad.

4. **Análisis Velocidad de Estabilización:** Según los informes extraídos de JIRA, salvo por dos equipos, la mayor parte de los grupos presentaron sobre esfuerzo durante el transcurso de sus proyectos, algunos incluso más de una vez. Al observar los tiempos de

Tabla 3. Resumen Gráficos Incidencias por Iteración.

Grupo	1° Iteración	2° Iteración	3° Iteración	Ninguna	Total
1	17	13	4	9	43
2	5	8	2	0	15
3	4	2	3	1	10
4	4	4	3	0	11
5	5	5	4	2	16
6	1	2	4	14	21
7	25	14	12	0	51

Tabla 4. Cálculo de Velocidad y Requerimientos No Completados.

Grupo	1° Iteración		2° Iteración		3° Iteración		Total	
	C	N	C	N	C	N	C	N
1	17	0	13	0	4	0	34	0
2	5	0	8	0	2	0	15	0
3	3	1	0	2	0	3	3	6
4	4	0	4	0	0	3	8	3
5	5	0	5	0	4	0	14	0
6	5	0	5	0	8	0	18	0
7	25	0	14	0	12	0	51	0

C = Incidencias completadas; N = incidencias no completadas.

Tabla 5. Resumen Gráficos Velocidad de Estabilización.

Grupo	¿Hubo Sobre esfuerzo?	Fase	Tiempo de estabilización (Días)
1	Sí	Desarrollo	Menor a un día
2	Sí	Pruebas	Menor a un día
3	No	-	-
		Desarrollo	3
4	Sí	Pruebas (2 veces)	9
			7
		Análisis y Diseño	1
5	Sí	Desarrollo (2 veces)	Menor a un día
			Menor a un día
6	No	-	-
		Análisis y Diseño	7
7	Sí	(2 veces)	Menor a un día
		Desarrollo	7
		Pruebas	5

estabilización de estos sobre esfuerzos (ver Tabla 5), se observa que algunos equipos lo solventaron rápidamente (ver ejemplo del grupo 1 en la Figura 6), mientras que otros mantuvieron esta situación durante varios días, 9 días en el peor de los casos. Concretamente,

la situación de los grupos fue la siguiente: 2 grupos no presentaron sobre esfuerzo, 3 grupos presentaron una o más situaciones de sobre esfuerzo, pero las resolvieron en un día o menos y, finalmente, 2 grupos presentaron una o más situaciones de sobre esfuerzo y las

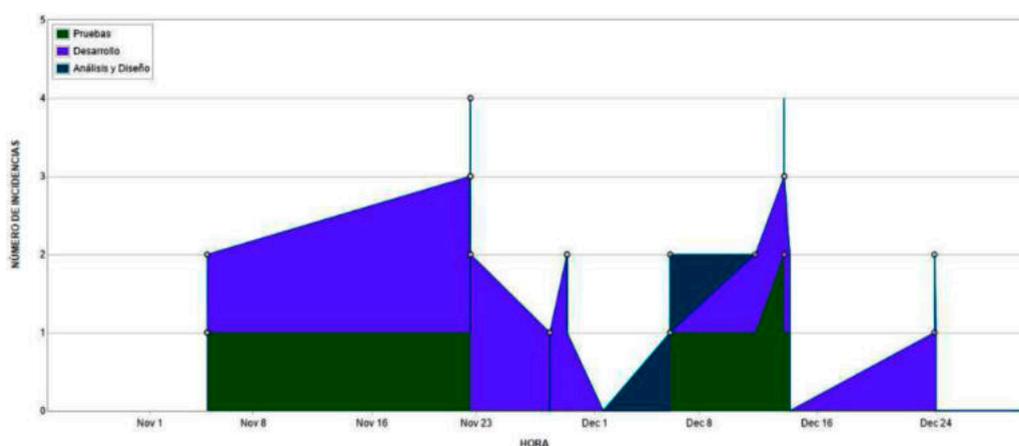


Figura 6. Gráfico Velocidad de Estabilización Grupo 1.

mantuvieron por varios días. Con esto, se puede concluir que la mayor parte de los grupos en estudio tuvo un comportamiento bastante positivo con respecto a los sobreesfuerzos, pudiendo solventarlos con rapidez y exhibiendo así una alta velocidad de estabilización.

- 5. Análisis Eficiencia:** La Tabla 6 presenta un resumen con la información más relevante obtenida. Al observar esta tabla se aprecia claramente que, en términos de eficiencia, hubo grupos con un alto nivel de eficiencia, y grupos que resultaron ser bastante ineficientes, de modo que es difícil encontrar un patrón claro al respecto. La predictibilidad por su parte demuestra que, en casi todos los casos, al observar el comportamiento pasado, se podría llegar a predecir las tendencias que se presentarían en el futuro, lo cual sería de gran utilidad a la hora de realizar una planificación. Finalmente, los valores atípicos y clústeres, por la cantidad y las características de estos, son un claro indicio de que los estudiantes presentaron diversas complicaciones durante su

desarrollo, lo cual se relaciona en gran medida con su inexperiencia con el enfoque y su falta de conocimientos respecto a las herramientas utilizadas, eso, sumado a que presumiblemente no utilizaron la aplicación JIRA como debían hacerlo, dio como resultado estos valores.

Respecto a los resultados cualitativos, en la Tabla 7 se presentarán todos los datos relevantes que se extrajeron a partir de los resultados obtenidos de las encuestas realizadas.

Respecto a las métricas cualitativas definidas en la propuesta de integración de enfoques ágiles, se describen los siguientes resultados:

- 1. Análisis Satisfacción del Equipo de Trabajo:** En base a la información mostrada en la tabla anterior, se observa que los estudiantes demostraron una gran satisfacción con el enfoque utilizado y las herramientas que esta les ofrece. Sus principales disgustos se relacionan con aspectos como la planificación de las tareas y el trabajo en equipo, lo cual, considerando que

Tabla 6. Resumen Datos Eficiencia.

Grupo	Eficiencia General	Predicibilidad General	Valores atípicos	Nº Clústeres
1	Alta	Alta	14	4
2	Baja	Alta	10	3
3	–	–	2	0
4	Alta	Alta	11	3
5	Baja	Baja	10	3
6	–	–	1	0
7	Baja	Alta	26	6

Tabla 7. Análisis de las encuestas aplicadas.

Aspecto analizado	Resultado
Características de los sujetos en estudio.	Corresponden a estudiantes universitarios con poca o nula experiencia en enfoques ágiles. Algunos de ellos poseen conocimientos teóricos, pero de un nivel muy básico.
Nivel de conocimiento de Scrum, Lean y Kanban.	Sólo un pequeño porcentaje de los estudiantes cuentan con un cierto nivel de conocimiento de Scrum. Kanban y Lean son desconocidos para ellos.
Dificultad de aprendizaje y adopción de la metodología.	Moderada. Los estudiantes coinciden en que el enfoque ofrece las herramientas necesarias para facilitar esta tarea.
Capacidad del enfoque de impulsar el trabajo en equipo.	Muy alta.
Relación entre la cohesión de un grupo de trabajo y su desempeño al momento de utilizar el enfoque.	Muy importante. Se percibe una notoria diferencia en las respuestas de aquellos alumnos que dicen haber pertenecido a equipos de trabajo sólidos con aquellos que no.
Utilidad de las Daily Scrum.	Para los sujetos en estudio, estas reuniones resultaron ser bastante útiles, aunque no se ha podido corroborar que efectivamente las hayan realizado todos los días.
Capacidad de resolver problemas para un equipo de trabajo regido por este enfoque.	Alta.
Importancia de los líderes dentro de los grupos de trabajo.	Muy importante. Incluso aquellos alumnos que mostraban un claro descontento con el grupo de trabajo al que pertenecían, destacaban la labor realizada por su líder.
Capacidad del enfoque de favorecer la visualización del flujo de trabajo.	Muy alta.
Capacidad del enfoque de incentivar la planificación.	Muy alta.
Dificultad en la división de proyectos en iteraciones para grupos de trabajo sin experiencia previa.	Baja.
Dificultad en la asignación y distribución de tareas para un grupo de trabajo sin experiencia previa.	Moderada.
Dificultad en la planificación de tiempos para un grupo de trabajo sin experiencia previa.	Alta.
Satisfacción de los sujetos en estudio con el enfoque utilizado.	Muy satisfechos.
Mayores dificultades presentadas por los sujetos en estudio.	Organización del trabajo en equipo, asignación de las tareas y distribución de los tiempos.

era su primer encuentro con un enfoque de este tipo, puede relacionarse perfectamente con su falta de experiencia.

- Análisis Facilidad de Adopción:** Los estudiantes coinciden en que el enfoque proporciona las herramientas y la información necesarias para facilitar su adopción, permitiendo que incluso un grupo inexperto pueda implementarlo con relativa facilidad.
- Análisis Organización y Planificación:** Sin duda este aspecto fue el más complejo para todos los estudiantes. Muchos tuvieron problemas a la hora de planificar, demorándose

muchas más horas de las presupuestadas para realizar ciertas tareas o en algunos casos, muchas menos. También les resultó bastante difícil planificar el número de tareas a realizar por iteración, lo que daba como resultado tareas que no se realizaban o que debían trasladarse a iteraciones siguientes. Respecto a la organización, muchos tuvieron problemas para coordinarse y distribuir las tareas, además, varios estudiantes coincidieron en que su grupo de trabajo no era el mejor, lo que repercutía directamente en su rendimiento y en el ambiente de trabajo.

## CONCLUSIONES

Los enfoques ágiles hoy en día son una alternativa muy efectiva para el desarrollo de software debido a la gran versatilidad que ofrece, que es ideal para el mercado cambiante de la era actual. Los proyectos en general poseen una gran incertidumbre y los requerimientos en muchos casos tienden a cambiar con el tiempo, lo que implica la realización de cambios que pueden significar grandes costos para las metodologías tradicionales. Lo anterior, sin embargo, no implica que los enfoques ágiles sean perfectos o aplicables para cualquier caso, depende mucho del contexto y de la situación particular de cada proyecto [10, 13, 15].

A través de la presente investigación, se estudiaron en detalle tres enfoques ágiles, y en todas ellas se han detectado deficiencias que deberían corregirse y también muchos beneficios. Esa fue la idea que dio origen a esta investigación, en la cual se pensó en la posibilidad de agrupar todos estos beneficios e intentar unificarlos a través de una propuesta que permitiera minimizar las deficiencias y ofrecer una nueva forma de aplicar estos enfoques.

Para lograr esto, el primer paso fue elaborar una extensa investigación que se dividió en 3 pasos fundamentales: seleccionar y estudiar los enfoques que conformarían la propuesta, elaborar la propuesta y, una vez planteada, comprobarla a través de un estudio, que consistió en la implementación del enfoque en un curso de la Pontificia Universidad Católica de Valparaíso, en donde los estudiantes debían desarrollar un proyecto semestral en grupos, lo que brindaba una ocasión perfecta para implementar el nuevo enfoque en sus proyectos.

Una vez finalizado el estudio y analizado los resultados obtenidos, las conclusiones que se pueden formular a partir de estos resultan bastante interesantes. En primer lugar, es importante identificar a los sujetos de estudio, es decir, a los estudiantes, y corroborar el nivel de conocimientos que estos poseían. El resultado es que los sujetos sometidos a este estudio corresponden a personas jóvenes con ciertos conocimientos de programación, pero sin ninguna experiencia en el campo de los enfoques ágiles para el desarrollo de software, además, muchos de ellos desconocían varias de las herramientas que debieron utilizar durante el

semestre, entre las que destaca JIRA, que cumplió un rol fundamental en la obtención de los datos cuantitativos.

Los proyectos fueron realizados en un plazo de unos 2 meses y medio aproximadamente, los cuales no estuvieron exentos de problemas. El estudio se realizó de la mejor manera posible y efectivamente se pudieron comprobar la mayor parte de las características de la propuesta planteada. Según la información proporcionada por los alumnos, el enfoque les pareció amigable, fácil de entender y muy útil a la hora de planificar, organizar, distribuir las tareas y resolver los problemas. Además, todos coincidieron en que este nuevo enfoque lograba fomentar la visibilidad del flujo de trabajo gracias a su pizarra *Kanban* y, pese a que algunos grupos no lograron una buena relación de trabajo, la mayoría concordó que este enfoque lograba fomentar el trabajo en equipo. Otro aspecto que destacar es la importancia de un buen equipo de trabajo, las opiniones de los estudiantes lo dejaron de manifiesto. Es de suma importancia que para cualquier equipo que decida incursionar en este enfoque, haya la suficiente confianza y cohesión entre sus integrantes, ya que de lo contrario la utilización de un enfoque ágil podría resultar incluso perjudicial.

Respecto a los datos cuantitativos recabados, se observó que los grupos, a nivel general, desempeñaron un trabajo bastante bueno a lo largo de todo el proyecto. Las principales ineficiencias se concentraron en las primeras semanas de trabajo, lo cual es comprensible considerando que los alumnos recién estaban en proceso de aprendizaje. En el ámbito de la planificación, fue bastante notoria la falta de experiencia de los equipos de trabajo, habiendo incidencias que nunca se resolvieron y otras que permanecieron por demasiado tiempo en el *Touch Time*, lo cual es un claro ejemplo de “residuo” para el proyecto. Por otra parte, desde el punto de vista del tiempo, los alumnos tuvieron grandes lagunas de tiempos muertos, existiendo períodos muy concretos de resolución de incidencias y otros en los que no se apreciaba ninguna actividad, aun así, los alumnos demostraron, en su mayoría, ser bastante eficaces en la resolución de sus incidencias, siendo capaces de resolver la mayor parte de estas pese a todo el tiempo desaprovechado. De lo anterior, se pueden deducir que los equipos de trabajo lograron cumplir con los objetivos que se habían planteado, pero a la

vez, su productividad fue bastante baja, ya que, al concentrar todo su trabajo en períodos cortos, no les daba tiempo de ir más allá e intentar mejorar la calidad de su trabajo o cuando menos, resolver una mayor cantidad de incidencias por iteración.

En referencia a los sobreesfuerzos o cuellos de botella de *Kanban*, los grupos en general tuvieron un buen comportamiento al respecto, pudiendo resolverlos con rapidez y logrando mantener un trabajo bien distribuidos entre todos los miembros. Asimismo, los grupos demostraron un nivel de eficiencia bastante normal durante el desarrollo, habiendo algunos equipos más eficientes y otros no tanto, lo cual se relaciona en gran medida con la cohesión que estos hayan tenido para organizarse de mejor o peor manera.

Este nuevo enfoque demostró ser capaz de ayudar a los estudiantes en la elaboración de sus proyectos, mejorando la visibilidad de sus flujos de trabajo y brindándoles herramientas para mejorar su planificación, su organización y su trabajo en equipo. Además, a través de *Kanban*, fueron capaces de mantener un adecuado orden en sus tareas y lograron sobreponerse sin dificultades a los sobreesfuerzos, evitando así que algún miembro del equipo se viera abrumado por una sobrecarga de trabajo. Por todo lo anterior y pese a todas las dificultades que tuvieron los grupos durante el desarrollo de los proyectos, el nuevo enfoque brindó resultados bastante positivos y sin duda podría llegar a ser un gran aporte para los desarrollos ágiles.

## REFERENCIAS

- [1] A. Reddy. “The *Scrumban*[r] evolution: getting the most out of Agile, *Scrum*, and *Lean Kanban*”. Addison-Wesley Professional. 2015.
- [2] A. Janes. “A guide to *Lean Software Development* in action”. In 2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW), pp. 1-2. IEEE. 2015.
- [3] A. Cockburn. “The heart of agile”. Humans and Technology, Salt Lake City, UT. 2018.
- [4] A. Cockburn and J. Highsmith. “Agile software development, the people factor”. Computer. Vol. 34, Issue 11, pp. 131-133. 2001.
- [5] C. Matthies. “Scrum2kanban: integrating kanban and scrum in a university software engineering capstone course”. In Proceedings of the 2nd International Workshop on Software Engineering Education for Millennials (pp. 48-55). 2018.
- [6] D. Anderson, G. Concas, M.I. Lunesu and M. Marchesi. “Studying *Lean-Kanban* approach using software process simulation”. In International Conference on agile software development, pp. 12-26. Springer, Berlin, Heidelberg. 2011.
- [7] E. Corona and F.E. Pani. “A review of *Lean-Kanban* approaches in the software development”. WSEAS Transactions on Information Science and Applications. Vol. 10, Issue 1, pp. 1-13. 2013.
- [8] E. Kupiainen, M.V. Mäntylä and J. Itkonen. “Using metrics in Agile and *Lean Software Development*” A systematic literature review of industrial studies”. Information and Software Technology. Vol. 62, pp. 143-163. 2015.
- [9] E. Valentin, J.R.H. Carvalho and R. Barreto. “Rapid improvement of students’ soft-skills based on an agile-process approach”. In 2015 IEEE Frontiers in Education Conference (FIE), pp. 1-9. IEEE. 2015.
- [10] H. Cornide-Reyes and R. Villarroel. “Agile methods in Chile: Identification of skills, learning processes, techniques and tools”. XIII Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento. Guanacaste, Costa Rica. 27-28 junio 2019.
- [11] H. Takeuchi and I. Nonaka. “The new product development game”. Harvard business review. Vol. 64 Issue 1, pp. 137-146. 1986.
- [12] J. Heidenberg and I. Porres. “Metrics functions for *Kanban* guards”. In 2010 17th IEEE International Conference and Workshops on Engineering of Computer Based Systems, pp. 306-310. IEEE. 2010.
- [13] J.H. Canós, P. Letelier y M.C. Penadés. “Metodologías ágiles en el desarrollo de software”. Taller realizado en el marco de las VIII Jornadas de Ingeniería del Software y Bases de Datos, JISBD 2003. Alicante.
- [14] J. Heidenberg and I. Porres. “Metrics functions for *Kanban* guards”. In 2010 17th IEEE International Conference and Workshops on Engineering of Computer Based Systems, pp. 306-310. IEEE. 2010.

- [15] K. Schwaber and M. Beedle. "Agile software development with *Scrum* (Vol. 1)". Upper Saddle River: Prentice Hall. 2002.
- [16] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler and J. Kern. "Manifesto for agile software development". 2001.
- [17] K. Schwaber and J. Sutherland. "La guía de *Scrum*". *Scrumguides. Org*, 1, 21. 2013.
- [18] N. Kirovska and S. Koceski. "Usage of *Kanban* methodology at software development teams". *Journal of Applied Economics and Business*. Vol. 3 Issue 3, pp. 25-34. 2015.
- [19] M. Poppendieck and T. Poppendieck. "*Lean Software Development: An Agile Toolkit*". Addison-Wesley. 2003.
- [20] M. Poppendieck and M.A. Cusumano. "*Lean Software Development: A tutorial*". *IEEE software*. Vol. 29 Issue 5, pp. 26-32. 2012.
- [21] M.O. Ahmad, D. Dennehy, K. Conboy and M. Oivo. "*Kanban* in software engineering: A systematic mapping study". *Journal of Systems and Software*. Vol. 137, pp. 96-113. 2018.
- [22] P.E. Colla. "Uso de opciones reales para evaluar la contribución de metodologías *Kanban* en desarrollo de software". In *Simposio Argentino de Ingeniería de Software (ASSE 2016)-JAIIO 45*. 2016.
- [23] R.J. Qureshi, M.O. Alassafi and H.M. Shahzad. "*Lean Agile Integration for the Development of Large Size Projects*". *International Journal of Modern Education and Computer Science*. Vol. 11 Issue 5, p. 24. 2019.
- [24] S. Dharmapal and K.T. Sikamani. "Applying *Lean* on agile *Scrum* development methodology". *Compusoft*. Vol. 3 Issue 3, p. 633. 2014.
- [25] S. Downey and J. Sutherland. "*Scrum* metrics for hyperproductive teams: how they fly like fighter aircraft". In *2013 46th hawaii international conference on system sciences*, pp. 4870-4878. IEEE. 2013.
- [26] S.I. Lozano, E. Suescún, P. Vallejo, R. Mazo y D. Correa. "Comparando dos estrategias de aprendizaje activo para enseñar *Scrum* en un curso introductorio de ingeniería de software". *Ingeniare. Revista chilena de ingeniería*. Vol. 28 N° 1, pp. 83-94. 2020.
- [27] VersionOne. "13th annual state of agile report". 2019. URL: <http://stateofagile.versionone.com>
- [28] V. Mahni'. "From *Scrum* to *Kanban*: introducing *Lean* principles to a software engineering capstone course". *Inter. J. of Eng. Educ*. Vol. 31 Issue 4, pp. 1106-1116. 2015.
- [29] V.E. Jyothi and K.N. Rao. "Effective implementation of agile practices-In coordination with *Lean Kanban*". *International Journal on Computer Science and Engineering*. Vol. 4 Issue 1, p. 87. 2012.
- [30] Y. Sugimori, K. Kusunoki, F. Cho and S. Uchikawa. "Toyota production system and *Kanban* system materialization of just-in-time and respect-for-human system". *The International Journal of Production Research*. Vol. 15 Issue 6, pp. 553-564. 1977.
- [31] Z. Masood, R. Hoda and K. Blincoc. "Adapting agile practices in university contexts". *Journal of Systems and Software*. Vol. 144, pp. 501-510. 2018.