Los Desafíos de las Metodologías Ágiles en Equipos Distribuidos

Stephanie Fernández Salinas y Octavio Frutos Angelucci, *Facultad de Ingeniería, Universidad de Palermo*

***Abstract* – A partir de la introducción de las metodologías ágiles en proyectos basadas del Manifesto Agile, han comenzado a implementarse numerosos marcos de trabajo, cada uno buscando resolver una problemática en particular.**

En este paper se propone un marco de trabajo, definiendo técnicas y procedimientos específicos para satisfacer exclusivamente las necesidades de los equipos distribuidos. Se planteará cuáles son los desafíos de este tipo de equipos y de qué manera se superarán las dificultades a través de la implementación de este nuevo marco basado en el Manifesto Agile.

Index terms – equipos distribuidos, metodologías ágiles.

I. NOMENCLATURA

SD Software Development
QA Quality Assurance

II. INTRODUCCIÓN

En la actualidad las grandes empresas implementan técnicas y procesos para desarrollar software siguiendo los lineamientos de las metodologías ágiles. Esto les permite adoptar formas de trabajo adecuadas a las condiciones del proyecto ofreciendo inmediatez y flexibilidad a la hora de adaptarse a las circunstancias del entorno. [1]

Las formas de trabajo han ido cambiando a lo largo de los años. Desde el año 2019 en adelante, hemos visto que el contexto de trabajo migró radicalmente hacia el "teletrabajo", sobre todo en empresas de desarrollo de software donde no es exclusivamente necesaria la presencialidad [2]. De esta manera, las empresas han adquirido recursos de todas partes del mundo conformando equipos virtuales.

Para poder seguir adoptando procesos y técnicas siguiendo las metodologías ágiles, las grandes empresas deben enfrentarse al desafío de intentar adaptar un marco de trabajo existente, pero en un entorno virtual, donde los equipos se encuentran distribuidos. No siempre resulta ser exitoso, ya que no hay un marco de trabajo definido exclusivamente para este tipo de equipos.

En este contexto, se investigará la efectividad de diferentes marcos de trabajos actuales que cumplan los lineamientos de las metodologías ágiles con el fin de indicar su efectividad en equipos distribuidos.

III. CASO DE ESTUDIO

La organización necesita implementar una solución para un proyecto radicado en Inglaterra con el objetivo de desarrollar una plataforma e-commerce web de productos para el hogar.

En el desarrollo de la dinámica de trabajo, el proyecto tiene como principales inconvenientes: la diversidad cultural y la brecha de zonas horarias entre los recursos que lo componen.

Analizando dichos recursos, encontramos la participación de 11 personas con distintos perfiles, distribuidas geográficamente y además con husos horarios distintos. Entre ellos se describen perfiles técnicos y otros orientados al análisis del negocio. El cliente nos provee de 3 recursos adicionales, quienes lo representarán en la toma de decisiones a partir del conocimiento sólido sobre las necesidades del negocio.

Los recursos se cuantifican y se encuentran distribuidos de la siguiente manera:

- 5 desarrolladores full-stack que se encuentran distribuidos 3 en Inglaterra (Londres) y 2 en Argentina (Buenos Aires).
- 3 stakeholders se encuentran en Inglaterra (Londres).
- 2 business analysts se encuentran en Argentina (Buenos Aires).
- 1 Tester QA se encuentra en Estados Unidos (New York).

Con lo cual, los husos horarios de las ciudades involucradas son:

- Argentina - Buenos Aires: (GMT-3)
- Inglaterra - Londres: (GMT+1)
- Estados Unidos - Nueva York: (GMT-4)

Basándonos en la metodología ágil, no encontramos un marco de trabajo que explícitamente brinde las directivas necesarias para la administración y el soporte del desarrollo de este proyecto.

Dada la situación actual es necesario proponer una nueva metodología que se adapte y soporte la administración de recursos distribuidos para el desarrollo de esta plataforma.

IV. MARCO TEÓRICO

A. Manifesto Agile

El Manifiesto Agile fue creado en el año 2001 por 17 expertos en programación, con el propósito de definir técnicas y procesos para desarrollar software. [3]
A lo largo de las últimas dos décadas desde la aparición del Manifiesto Ágil, han surgido diferentes marcos de trabajo de los cuales se han basado en sus principios para su desarrollo e implementación. [4]

A pesar de que uno de los principios indica que *"El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.",* la esencia de este tipo de metodologías es la flexibilidad y la adaptación al cambio. Aunque algunos marcos de trabajo actuales han logrado adaptarse en favor de equipos distribuidos, no siempre logran contemplar en profundidad la problemática planteada. [5]

B. *Marcos de Trabajo populares*

Estudiando los distintos marcos de trabajo existentes más populares, tales como Scrum, Kanban, Lean, etc. encontramos que, si bien tienen la posibilidad de adaptarse a equipos distribuidos, ninguno está diseñado exclusivamente para ellos cubriendo las características más distintivas de los mismos.

Scrum, por un lado, es uno de los marcos de trabajo más populares y utilizados por su gran capacidad de adaptabilidad. Scrum es un marco de trabajo basado en procesos iterativos para desarrollar y entregar un producto de manera incremental. Reduce la complejidad del desarrollo de un producto, particionando en pequeñas porciones entregables [6].

Kanban también es un marco de trabajo popular y trabaja con procesos de trabajos iterativos y entregas incrementales, tal como lo hace Scrum. Sin embargo, Kanban se diferencia al centrarse en la mejora de la visibilidad del flujo de trabajo utilizando un tablero especialmente para este marco y limitando el trabajo en curso. [7]

Por último, hemos investigado otro marco de trabajo llamado Lean SD, el cual ofrece una opción mucho más flexible a la proporcionada por Scrum. Su objetivo radica en la generación de valor al cliente a través de la eliminación de residuos y entregas rápidas. [8]

C. *Herramientas para gestión de proyectos*

Todo proyecto de desarrollo de software, independientemente del marco de trabajo que se decida implementar, necesitará de herramientas de gestión de proyectos.

Estudiaremos la factibilidad de utilizar una herramienta de gestión de proyectos como por ejemplo las de Atlassian [9], una compañía que nos ofrece soluciones para: realizar seguimiento de tareas e incidencias (con Jira Software [10]), gestionar la documentación de manera colaborativa (con Confluence [11]), gestionar el código del desarrollo del producto de manera colaborativa (con Bitbucket [12]). Para administrar la comunicación entre los integrantes del proyecto estudiamos una plataforma llamada G-Suit [13] que nos permite desarrollar, entre otras cosas, la comunicación formal mediante mails y conversaciones informales mediante chats.

V. DESARROLLO

A. *Análisis de marcos propuestos*

Para desembocar en una solución con fundamentos ágiles y que resuelva nuestra problemática de manera precisa, nos propusimos investigar los marcos de trabajo mencionados previamente. El propósito de esta investigación es poder hacer un análisis crítico de cada uno de ellos con el fin de poder identificar el marco que se logre adaptar a la necesidad de nuestro caso de estudio propuesto, ponderando las distintas características y atributos de cada uno de los marcos.

Para empezar, Scrum es una de las primeras opciones con las que nos encontramos al momento de buscar una metodología ágil para implementar en un proyecto. Scrum se caracteriza por ser poco complejo, y poseer atributos como roles, artefactos y eventos. Entre estos podemos resaltar los siguientes:

- Roles: Scrum Master, Product Owner y Scrum Team.
- Artefactos: Product Backlog (lista de todas aquellas funcionalidades a implementar como defectos a resolver y que es mantenida por el Product Owner), Sprint Backlog (lista con todas aquellas features y bugs a desarrollar durante el sprint actual) y Potentially Releasable Product Increment.
- Reuniones: Sprint Planning Meeting, Daily Scrum Meeting, Sprint Review Meeting, Sprint Retrospective Meeting. [14]

A partir de esto, observamos que Scrum podría ser un potencial candidato para adaptarse a las necesidades de la gran mayoría de los proyectos, incluyendo el nuestro. Sin embargo, observamos que Scrum demanda mucho tiempo en reuniones, lo cual puede ser un inconveniente para equipos muy distribuidos como en nuestro caso. A esto hay que sumarle que Scrum tiene ciertas dificultades a la hora de implementar equipos en múltiples locaciones, lo cual implica una gran tarea para poder sentar las bases de los equipos y suponiendo que vamos a contar con un equipo lo suficientemente grande para poder dividirlo, aun tratando de proponer la posibilidad de trabajar con equipos virtuales. [15] [16]

Kanban se ofrece como otra metodología ágil, la cual se centra en un tablero especialmente diseñado para esta metodología en la que se muestra el trabajo del equipo.

Dentro de esta metodología se puede señalar que no se encuentran definidos roles formales, artefactos o tiempos de iteración. Debido a esto, el flujo de trabajo se realiza de forma continua, en donde las historias, a medida que avanza y finaliza su desarrollo, son implementadas directamente, o a discreción del equipo, permitiendo una mayor fluidez de trabajo. [17]

A pesar de la propuesta, Kanban se centra mucho en los tiempos de desarrollo de las historias, desde su creación hasta su implementación, como la volatilidad de cambios dentro de las mismas, debido a la ausencia de control por parte del cliente, sumado a la ausencia de roles definidos, lo cual en un grupo distribuido virtualmente puede traer aparejado cierto desorden en la distribución de responsabilidades. [18] [19]

La metodología ágil Lean SD tiene como base los principios de Lean, a los cuales se le incorporan las prácticas de ágil. Entre los principios de Lean que podemos resaltar se encuentra el eliminar los restos, el cual consiste en reducir todo aquello que no genere valor al producto, sea porque no es algo necesario o porque se puede obtener el resultado deseado sin necesidad del mismo, como también el poder adaptarse a la necesidad del cliente a partir de cada entrega realizada y de forma rápida. [20]

Lean SD es una metodología bastante útil a la hora de desarrollar trabajo eficiente y conciso, pero conlleva la desventaja de que depende fuertemente del equipo en el que se implementa, lo cual puede ser un impedimento para un equipo distribuido en varias locaciones. [21]

B. *Bridge Ágil*

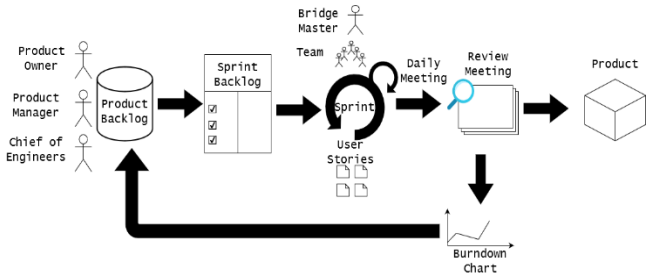


Fig 1. Dinámica de nueva metodología propuesta

A partir de la problemática planteada en el caso de estudio propuesto, hemos decidido proponer un nuevo marco de trabajo basado en la metodología ágil. Este nuevo marco de trabajo atenderá exclusivamente las necesidades de un equipo distribuido, tal como el propuesto.

Para el diseño del mismo, se incorporarán diferentes técnicas, artefactos, roles y eventos de marcos existentes.

Vamos a considerar algunos de los roles, reuniones y artefactos de Scrum, sacando todo lo que nos parezca redundante o nos haga desperdiciar tiempo. También consideraremos la fluidez de Kanban en el seguimiento de tareas y, por otro lado, la identificación y reducción de todo aquello que no genere valor como en Lean SD.

A esto también se le incorporarán todas aquellas técnicas que creemos fundamentales para dedicar este marco exclusivamente a las características de un equipo distribuido.

1. Roles

Dentro de nuestra metodología ágil se propondrán diversos roles. En ella se encuentra el "*Team*", al igual que en metodología Scrum, se trata de una colección de personas trabajando para entregar lo solicitado y comprometido en incrementos de producto similar a la de un equipo Scrum. Para trabajar de forma eficaz es importante que sigan un objetivo común, se adhieran a las mismas normas y reglas.

Por su parte estará el "*Bridge Master"* quien tiene como principal tarea cumplir con la regla "*No-gaps"* esto debe asegurar que los miembros del proyecto puedan trabajar como si entre ellos no hubiera distancias de ningún tipo. Para esto se ayudará de reuniones que le demuestren cuál es el estado de cada integrante con respecto a sus tareas. Adicionalmente, se encargará de facilitar la fluidez entre la comunicación y coordinación del equipo, de forma similar a un Scrum Master, en el marco Scrum, en la cual tratará de quitar todas aquellas disrupciones que puedan surgir a lo largo del proyecto.

Por parte del cliente se contará con la participación de un "*Product Manager*", el cual será responsable de la estrategia, planificación, ejecución y lanzamiento del Producto. Su tarea consistirá en coordinar todas las tareas implicadas en el desarrollo del Producto, desde su conceptualización hasta su lanzamiento. Además de darle seguimiento a las ventas del producto, gestión de las incidencias, pedidos de cambios o mejoras que haya que hacer en el mismo, será quien dé respuestas y brinde conocimiento sobre lo que se está produciendo y desarrollando.

Como intermediario entre "*Product Manager*" y el resto de los integrantes del proyecto, contaremos con un "*Product Owner*". Este rol será el propietario del Producto y la voz del cliente, buscando representar las necesidades del mismo y transformando sus requerimientos en "*User Stories*" medibles y desarrollables. Adicionalmente, será el responsable de aclarar cualquier detalle con respecto a la solución, apoyándose en el *"Product Manager*" y es también responsable de mantener priorizada una lista de ítems de trabajo que el equipo implementará para entregar la solución. Similarmente al Scrum "*Product Owner*", será el responsable del "*Product Backlog*" y el "*Sprint Backlog*".

Adicionalmente, se propondrá la figura del "*Chief of Engineers*", el cual será el representante técnico del equipo y el propietario de la arquitectura, tal como en el marco Lean. Funcionará como intermediario entre el equipo de desarrollo y el "*Product Owner*", pudiendo utilizar sus skills técnicos para cumplir con los requerimientos funcionales. Este rol debe tener experiencia técnica y conocimiento sólido en el dominio del negocio pudiendo apoyarse en la figura del "*Product Owner*".

2. Artefactos

Para que el equipo pueda desarrollar sus actividades diarias se contarán con diversos artefactos, los cuales están basados en los artefactos propuestos por Scrum.

En un principio se contará con un "*Product Backlog*", en el cual se encontrarán definidas todas aquellas tareas que se deben realizar dentro del proyecto. El dueño de este artefacto será el "*Product Owner*", el cual contará con la contribución del "*Product Manager*" y el "*Chief of Engineers*". El "*Product Owner*" junto al "*Product Manager*" y el "*Chief of Engineers*" van a resolver los detalles tanto funcionales como técnicos de cada "*User Story*" que compone el "*Product Backlog*", y van a definir el orden de prioridades de las mismos.

Los elementos que describen las tareas necesarias para el desarrollo del producto los llamaremos *"User Stories*", tal como Scrum. Será el nivel más elemental y pequeño de artefacto que vamos a tener. Una "*User Story*" tendrá el detalle del requerimiento funcional con una descripción a nivel técnica. Se centrarán en el usuario final. El "*Product Owner*" junto al "*Product Manager*" resolverán los detalles funcionales mientras que el "*Chief of Engineers*" se encargará de los detalles más técnicos. Es importante que la "*User Story*" esté lo más completa posible para que pueda ser desarrollada con la menor cantidad de impedimentos. Adicionalmente cada *"User Story*" tendrá una sección para ir completando a medida que es desarrollada con los impedimentos que se recaban de las reuniones de estado diarias. Esto significa que la *"User Story*" representará un documento vivo que durará todo un sprint y del cuál se puede obtener un aprendizaje al conocer qué impedimentos tuvo a lo largo de ese período de tiempo.

A medida que las "*User Stories*", sean priorizadas para desarrollarse se irán incorporando dentro del artefacto que denominamos como "*Sprint Backlog*". Este artefacto contendrá una lista de "*User Stories*" que se realizará en un período de tiempo pactado, la cual llamaremos "*Sprint*". Cada una de estas "*User Stories*" deberán ser estimadas en una base de horas-persona con el fin de rastrear progreso y esfuerzos restantes a lo largo del "*Sprint*".

Es necesario poder medir los esfuerzos realizados por el "*Team*" para desarrollar tareas. Esto nos permite conocer la capacidad y la velocidad del equipo. Para esto tendremos un último artefacto llamado "*Burndown Chart*". Este artefacto tendrá las métricas de desarrollo de por parte del *"Team*" en un determinado período de tiempo. Contará con información recopilada de varios Sprints con el fin de ayudar al "*Product Manager*", "*Chief of Engineers*", "*Product Owner*" y "*Bridge Master*" a conocer cuál es la velocidad del equipo, cuál es su capacidad en un Sprint y tomar decisiones al respecto.

3. Reuniones

Para poder coordinar varios de los artefactos definidos se contará con diversas reuniones, las cuales se basan en Scrum.

En un principio, se contará con una reunión con frecuencia diaria, la "*Status Meeting*", la cual será llevada adelante por el

"*Bridge Master*" y participarán todos los integrantes del proyecto. Cada miembro deberá informar el estado de sus tareas: el progreso realizado hasta ese momento, los objetivos para el resto del día, impedimentos e información necesaria para completarlas.

Esta información es el "input" del trabajo del *"Bridge Master*", ya que pondrá su foco y dedicación exclusivamente en remover los impedimentos que se hayan presentado y conseguir la información necesaria para que cada uno pueda completar sus tareas. Además, el "*Bridge Master*" tendrá conocimiento sobre las dificultades que se hayan presentado durante la realización de las tareas con el fin de dejar un precedente para la estimación de próximas tareas.

Se contará con otra reunión al comienzo de cada iteración, la "*Strategy Planning*", en la cual se discutirán aquellas "*User Stories*" que se encuentran en el "*Product Backlog*" y que el "*Product Owner*" las haya priorizado para esta iteración. Llevada adelante por el "*Product Owner*" y el "*Chief of Engineer*", explicarán en detalle (tanto funcional como técnico) cada una de las "*User Stories*" al "*Team*". Cada una de las "*User Stories*" discutidas deberá ser puntuada a partir del conocimiento y experiencia del "*Team*", con la finalidad de poder aproximar el esfuerzo requerido para la realización de la misma.

La principal estrategia para llevar a cabo una planificación basada en estimación precisa es hacer una retrospectiva de cada "*User Story*" de la iteración finalizada utilizando la información relacionada a los impedimentos escritos sobre este artefacto. Esto nos permite estimar las próximas "*User Stories*" aprendiendo sobre experiencias anteriores para buscar soluciones más precisas.

Una vez finalizado el "*Sprint*", se contará con la "*Review Meeting*", en la cual se planifican las próximas "*User Stories*" de la próxima iteración. Esta meeting será llevada adelante por el "*Product Owner*". Durante esta reunión, se establecerá la prioridad de las "*User Stories*" a desarrollar en la próxima iteración.

4. Reglas

Hemos definido cuáles serán los componentes que conforman nuestro marco de trabajo. Ahora es necesario definir cómo serán las interacciones entre los mismos.

- La primera regla indica que el proyecto se dividirá en entregas funcionales e iterativas hasta alcanzar el fin del desarrollo del producto. Estas iteraciones denominadas "*Sprints*", al igual que en el marco Scrum [22], serán una porción de trabajo funcional en un determinado período de tiempo. El proyecto se dividirá en una determinada cantidad de "*Sprints*".
- La segunda regla indica que todas las *Reuniones* tendrán uno de los siguientes objetivos: ser el medio por el cual los distintos *Roles* interactúan con los distintos *Artefactos (*por ejemplo, en la reunión *"Strategy Planning*" se decidirá qué tareas pasarán del *"Product Backlog*" al "*Sprint Backlog*" para ser realizadas en esa iteración), o bien ser el canal por el cual los integrantes del "*Team*" comuniquen formalmente sus avances e impedimentos o realicen las demostraciones del trabajo realizado.
- La tercera regla y más importante la llamaremos *"No-Gaps"* y nos asegurará la disponibilidad, en todo momento, de todos los recursos o información necesaria para que cada uno de los integrantes del proyecto pueda desarrollar sus tareas sin interrupciones. Para que esto se cumpla debemos considerar que el "*Bridge Master*" cumplirá un rol clave ya que será quien vele por la disponibilidad de los recursos para todos los integrantes del proyecto para un trabajo sin interrupciones.

5. Herramientas de Gestión

Es necesario contar con una o varias herramientas para la gestión de un proyecto dentro de nuestro marco propuesto. Con estas herramientas debemos poder gestionar:

- El código para el desarrollo.
- Las tareas e incidentes, así como también el seguimiento de las mismas.
- La documentación del proyecto.
- La comunicación (formal e informal) entre los miembros.

Utilizaremos como ejemplo las herramientas que pertenecen al paquete de Atlassian: para la gestión del código podríamos utilizar Bitbucket, para la administración de tareas e incidentes podríamos utilizar Jira. En cuanto a la documentación un ejemplo podría ser Confluence. Para la comunicación interna podríamos contar con un workspace de Google, G Suite.

VI. CONCLUSIONES

Dentro de los marcos de trabajo actualmente implementados no se encuentra definido un marco que logre contemplar las necesidades de los equipos distribuidos.

A partir del planteo de un marco de trabajo en la que se busca contemplar la existencia de equipos totalmente distribuidos, una realidad que hoy en día sigue en auge, se ha indagado en una posible solución a partir de adaptar criterios previamente establecidos como establecer nuevos procesos y técnicas que logren derribar las dificultades que conllevan la distancia geográfica, horaria y cultural entre este tipo de equipos.

Dentro del marco desarrollado, se logra acertar en aquellas técnicas existentes en diversos marcos previamente realizados y en nuevas herramientas necesarias para efectivizar la comunicación y capacidad de trabajo en equipos distribuidos. Incluso, es posible adaptar marcos ya existentes con algunas herramientas descritas con la finalidad de generar un híbrido entre su metodología y la planteada.

VII. LÍNEAS FUTURAS DE INVESTIGACIÓN

- Metodologías ágiles orientadas a equipos grandes.
- Metodologías ágiles orientadas a prácticas de desarrollo software.

VIII. REFERENCIAS

[1] E. Gómez Guitiérrez, M. M. Marcillo Guevara y N. Ramírez López, «Metodologías Ágiles para el Desarrollo de Proyectos,» Fundación Universitaria Católica Lumen Gentium, Santiago de Cali, 2020.

[2] L. Osio Havriluk, «El Teletrabajo: Una opción en la era digital,» *Observatorio Laboral Revista Venezolana,* vol. 3, nº 5, pp. 93-109, 2010.

[3] W. Cunningham, «Manifesto for Agile Software Development,» 2001. [En línea]. Available: https://agilemanifesto.org/.

[4] W. Cunningham, «Principles behind the Agile Manifesto,» 2001. [En línea]. Available: https://agilemanifesto.org/principles.html.

[5] R. C. Martin, «Whole Team,» de *Clean Agile: Back to Basics*, Pearson, 2019, pp. 95-96.

[6] International Scrum Institute, «What is Scrum?,» de *Scrum Revealed Training Book*, 2020, p. 8.

[7] M. Berejo, M. Florit y R. Sedó, «El Kanban,» de *Introducción a las metodologías ágiles en la gestión de proyectos*, Barcelona, Oberta UOC Publishing SL, 2016, p. 25.

[8] J. Gaete, R. Villarroe, I. Figueroa, H. Cornide-Reyes y R. Muñoz, «Enfoque de Aplicación Ágil con Serum, Lean y Kanban,» *Ingeniare*, vol. 29, nº 1, pp. 141-157, Agosto 2021.

[9] Atlassian, «Herramientas para Desarrollo de Software y Gestión de Proyectos,» [En línea]. Available: https://www.atlassian.com/.

[10] Atlassian, «Jira - Software de Seguimiento de Proyectos e Incidencias,» [En línea]. Available: https://www.atlassian.com/es/software/jira.

[11] Atlassian, «Confluence - El espacio de Trabajo Remoto para tu Equipo,» [En línea]. Available: https://www.atlassian.com/es/software/confluence.

[12] Atlassian, «Bitbucket: Herramienta de Gestion de Código Git,» [En línea]. Available: https://www.atlassian.com/es/software/bitbucket.

[13] Google, «G-Suite,» [En línea]. Available: https://workspace.google.com/intl/es-419/features/.

[14] International Scrum Institute, Scrum Revealed Training Book, 2020.

[15] International Scrum Institute, «Distributed & Large Scrum Projects,» de *Scrum Revealed Training Book*, 2020, pp. 43-44.

[16] Y. Khmelevsky, X. Li y S. Madnick, «Software development using agile and scrum in distributed teams,» de *2017 Annual IEEE International Systems Conference (SysCon)*, 2017.

[17] P. Diebold, S. Theobald, J. Wahl y Y. Rausch, «Stepwise transition to agile: From three agile practices to Kanban adaptation,» *Journal of Software: Evolution and Process*, vol. 31, nº 5, 2019.

[18] H. Alairdaros, M. Omar y R. Romli, «Identification of criteria affecting software project monitoring task of Agile Kanban method,» de *AIP Conference Proceedings 2016*, 2018.

[19] H. Alairdaros, M. Omar y R. Romli, «Towards an Improved Software Project Monitoring Task Model of Agile Kanban Method,» *International Journal of Supply Chain Management*, vol. 7, nº 3, pp. 118-125, 2018.

[20] J. Highsmith y K. Schwaber, Lean Software Development: An Agile Toolkit, Addison-Wesley, 2003.

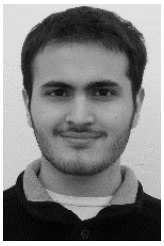
[21] C. Ebert, P. Abrahamsson y N. Oza, «Lean Software Development,» *IEEE Computer Architecture Letters*, vol. 29, nº 5, pp. 22-25, 2012.

[22] International Scrum Institute, «What is a Sprint?,» de *Scrum Revealed Training Book*, 2020, p. 32.

IX. Biografía



Stephanie Fernández Salinas, nacida en New Jersey, Estados Unidos, el 8 de Julio de 1993, estudiante de la carrera de Ingeniería en informática de la Universidad de Palermo.



Octavio Fruto Angelucci, nacido en Buenos Aires, Argentina, el 7 de Junio de 1994, estudiante de la carrera de Ingeniería en Informática en la Universidad de Palermo.