Revista Cubana de Ciencias Informáticas
Vol. 16, No. 4, Octubre-Diciembre, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 18-34

Tipo de artículo: Artículo original
Temática: Ingeniería y gestión de software
Recibido: 18/08/2022 | Aceptado: 06/09/2022

Adopción de una cultura Ágil en el desarrollo de proyectos de aplicaciones para Internet en Cuba

Adoption of an Agile culture in the development of Internet application projects in Cuba

Allan Pierra Fuentes [1*] https://orcid.org/0000-0003-4375-1034
Haniel Cáceres Navarro [1] https://orcid.org/0000-0002-0163-4852
Yadier Perdomo Cuevas [1] https://orcid.org/0000-0003-2360-036X
Yanio Hernández Heredia [1] https://orcid.org/0000-0001-9433-5511
Raydel Montesinos Perurena[1] https://orcid.org/0000-0003-4747-3166

[1] Universidad de las Ciencias Informáticas, Cuba. Carretera San Antonio km 2 ½. {apierra,hcaceres,ypc, yhernandezh,raydel}@uci.cu

*Autor para la correspondencia. (apierra@uci.cu)

RESUMEN

Las metodologías ágiles han ganado una amplia aceptación y aplicación en el desarrollo de software, las mismas fueron creadas como una alternativa a las metodologías tradicionales con el fin de minimizar la carga burocrática que éstas imponen en el desarrollo de proyectos de pequeña y mediana escala. Las metodologías ágiles, a diferencia de las tradicionales, son adaptativas para cada tipo de proceso y están orientadas a las personas y no a los procesos. Esta transformación requiere, más que la adopción de nuevas metodologías, un

cambio de comportamiento y mentalidad en toda la organización, esto implica que la misma necesita cambiar su cultura para volverse más ágil. En este trabajo, se explora cómo un equipo de desarrollo de software pequeño, en el entorno universitario de la Universidad de las Ciencias Informáticas, adopta un enfoque colaborativo para desarrollar una cultura ágil basada en un conjunto de herramientas para ayudar a su transformación, con vistas al trabajo con importantes clientes empresariales. Scrum, Kanban y el Modelo de ingeniería de Spotify han sido las técnicas empleadas. Con base en el estudio de caso, se extraen las estrategias de soluciones genéricas para adaptar el método.

Palabras clave: Metodologías ágiles; Scrum; Kanban; modelo de Spotify; autoorganización; equipos de software.

ABSTRACT

Agile methodologies have gained wide acceptance and application in software development, they were created as an alternative to traditional methodologies in order to minimize the bureaucratic burden that they impose in the development of small and medium-scale projects. Agile methodologies, unlike traditional ones, are adaptive for each type of process and are oriented to people and not to processes. This transformation requires, more than the adoption of new methodologies, a change in behavior and mentality throughout the organization, this implies that it needs to change its culture to become more agile. In this paper, it is explored how a small software development team, in the university environment of the University of Informatics Sciences, adopts a collaborative approach to develop an agile culture based on a set of tools to help its transformation, for work with major business clients. Scrum, Kanban and the Spotify Engineering Model have been the techniques used. Based on the case study, generic solution strategies are extracted to adapt the method.

Keywords: Agile methods; Scrum; Kanban; Spotify model; Self-management; Software teams.

Revista Cubana de Ciencias Informáticas
Vol. 16, No. 4, Octubre-Diciembre, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 18-34

Introducción

En la planificación estratégica 2017-2021 de la Universidad de Ciencias Informáticas (UCI), se implementó como área de resultado clave «Desarrollo y comercialización de aplicaciones y servicios informáticos y académicos», la cual contemplaba entre sus objetivos:

1. Comercializar soluciones de impacto, desarrollando tecnologías disruptivas a nivel nacional e internacional.
2. Consolidar el proceso de desarrollo logrando aplicaciones y servicios competitivos y de alto valor agregado.

Entre las acciones para dar respuesta a estos desafíos estuvo la creación del grupo de desarrollo de software conocido como Z-17, para trabajar en el marco de la alianza estratégica establecida entre la UCI y la Empresa Telecomunicaciones de Cuba (ETECSA), con el propósito de desarrollar soluciones informáticas para los proyectos de aplicaciones de Internet en Cuba.

Estas soluciones comenzaron a desarrollarse en el año 2017, realizándose el lanzamiento en junio de 2018, de la plataforma de mensajería instantánea y colaborativa *(toDus®)* y el Centro de Aplicaciones Android *(Apklis®)*, posteriormente se lanzó la plataforma cubana de contenidos audiovisuales *(Picta®)*.

En nuestro contexto, los temas relacionados con la innovación y la gestión de proyectos, han sido abordados de forma aislada y aunque hay intentos, su desarrollo teórico y conceptual es aún incipiente.

La UCI desde su creación, se ha dedicado a la producción y comercialización de software, para ello cuenta con una red de 14 centros de desarrollo de tecnologías de la información con intereses específicos. A partir del 2008 se inició un programa de mejora de procesos, con la contratación de los servicios de consultoría del «Software Industry Excellence Center de México S.C.», una compañía spin-off del Tecnológico de Monterrey, "partner" de líderes mundiales como el Software Engineering Institute (SEI) y el Capability Maturity Model Integration (CMMI) Institute. Con la ejecución de un programa de mejoras en el 2015 se logró la certificación con el Nivel 2 del CMMI, la UCI se convirtió en la primera institución cubana con este reconocimiento (Sospedra y otros, 2017).

Con la implementación de los procesos correspondientes al Nivel 2 del CMMI la organización ha logrado desarrollar un proceso de planificación y monitoreo continuo a lo largo del ciclo de vida de los proyectos y

Editorial "Ediciones Futuro"
Universidad de las Ciencias Informáticas. La Habana, Cuba
rcci@uci.cu

se han automatizado la mayor parte de las actividades mediante la suite de gestión de proyectos Xedro-GESPRO que tiene como base el estándar PMBOK, la ISO 21500 y los estándares asociados al sistema para la Planificación de Recursos Empresariales (ERP) (Sosa González y otros, 2016).

No obstante, estos logros, se presentan insuficiencias en la gestión de proyectos, que afectan el equilibrio de los factores tiempo, costo y resultado satisfactorio. Así mismo se cuenta con un alto volumen de información sobre modelos y estándares para la mejora de procesos (ISO, CMMI, PMBOK), pero escaso conocimiento en la base sobre cómo aplicarlos de forma práctica (Lugo-García y otros, 2014), una cuestión importante es que tradicionalmente se han gestionado los proyectos de software con metodologías diseñadas para grandes entornos empresariales, las cuales, se ha podido comprobar que en la mayoría de los casos no son las idóneas para este tipo particular de entorno de desarrollo de proyectos.

Esta problemática nos lleva a proponer como objetivo la búsqueda de una alternativa viable para la gestión de proyectos basada en el aprendizaje y la colaboración como determinantes del éxito.

Métodos o Metodología Computacional

La presente investigación se propone explorar en el campo de las metodologías ágiles, para la gestión exitosa de proyectos de desarrollo de software. Uno de los factores principales a la hora de incorporar una metodología Agile a una organización es el cambio cultural, que esta nueva forma de trabajo requiere. El aspecto humano y el empoderamiento de los equipos pasa a definir el tipo de liderazgo necesario para adoptar una cultura Agile.

Metodologías Ágiles

Le presentamos unas definiciones resumidas de las herramientas adoptadas, para ejemplificar cómo son, cómo se utilizan, que proporcionan y que rasgos generales contienen. Las mismas tienen su valor añadido en el camino recorrido antes del lanzamiento del producto final, en la cadena de ensayo y error, los prototipos e iteraciones, el trabajo conjunto con el cliente y la comunicación tanto entre miembros del equipo como con

el cliente. En definitiva, en los procesos de aprendizaje (learning by doing), de iteración, de adaptación y de mejora continua.

Marco de desarrollo Scrum

Es un marco para el desarrollo ágil de proyectos, surgido inicialmente en la industria del software, en el cual el proceso de desarrollo de un proyecto es concebido como una sucesión de ciclos cortos de trabajo denominados *sprints*, obteniendo de cada uno de ellos un producto funcional que se va completando de forma iterativa. Al desarrollar productos innovadores donde los requisitos del producto no están claros desde el principio, es una buena opción debido a que cada iteración se basa en los hallazgos empíricos de la anterior, los requisitos del producto serán más claros a medida que avanza el proyecto (Schwaber y Sutherland, 2017). Esta forma de concebir la gestión de proyectos es radicalmente diferente al modo secuencial *(Waterfall)* tradicionalmente utilizado, que los divide en etapas sucesivas especializadas, planificadas a priori en forma detallada, se propone en cambio un desarrollo en forma iterativa, como trabajo de un equipo *(scrum team)* sobre una versión completa del producto, centrada en el cliente (Estrada-Velasco y otros, 2021).

Uno de los roles clave de Scrum es el denominado propietario del producto *(product owner)*, que participa activamente en el proceso de desarrollo, facilitando la comprensión por parte del equipo de los aspectos prioritarios del resultado esperado. Representa al cliente, y tiene una continua interacción con el equipo, facilitando desde el inicio una clara visión del producto. Al mismo tiempo, adquiere una comprensión de las posibilidades y dificultades a partir de la comunicación con ellos (Wonohardjo y otros, 2019).

Se define otro rol relevante, el facilitador *(scrum master)*, quien es responsable de orientar al equipo en la aplicación de las prácticas adecuadas para lograr los beneficios esperados, al mismo tiempo que se encarga de eliminar impedimentos y reducir las fricciones que la dinámica de trabajo pueda producir.

Schwaber y Sutherland (2017), quienes formularon las versiones iniciales del marco Scrum, explican en su guía los tres pilares que sustentan el control de procesos en base a hallazgos empíricos, ellos son la transparencia, la fiscalización y la adecuación. Esto exige en etapas tempranas del proyecto lograr una visión

enfocada del objetivo a corto plazo, pues se apunta a obtener en un lapso de tiempo reducido una versión "viable" del producto, aunque restringida a sus características esenciales.

Esto induce a concentrarse en los aspectos de mayor relevancia, se espera de cada iteración o *sprint*, un entregable denominado "incremento" que es considerado un producto potencialmente completo. El *sprint* exige al equipo un alto grado de interacción para concretar el objetivo, favoreciendo que surjan tempranamente las dificultades, que serán afrontadas diariamente a través de reuniones de seguimiento, breves y enfocadas, tradicionalmente realizadas de pie *(daily stand-up meetings)*, que permiten realizar un control del avance, coordinar esfuerzos, compartir estrategias y mejorar la cohesión del equipo.

Cada *sprint* se desarrolla en tres fases: una reunión de planificación, un período de trabajo a lo largo del cual se realizan las reuniones diarias de seguimiento, al final se realiza la reunión de revisión del producto desarrollado en el *sprint*, posteriormente se hace la reunión de evaluación del proceso de trabajo con vista a su mejora continua, que se denomina "retrospectiva". Un proyecto completo entonces será visto como una sucesión de *sprints* a través de los cuales se irá perfeccionando el producto hasta que el *product owner* considere que se ha alcanzado el estado deseado. Este proceso se apoya en tres instrumentos: 1) La lista de los requisitos del producto denominada *Product Backlog*, representa el punto de vista del *product owner* de las funcionalidades del producto. 2) La lista producida antes de iniciar cada *sprint*, donde se define qué *Storyteller* serán satisfechas en ese ciclo de trabajo, denominada *Sprint Backlog*, que es preparada por el *squad* y el *product owner*. 3) El gráfico de *burndown*, es una herramienta que permite visualizar la cantidad de trabajo remanente (Morandini y otros, 2021)

Agile es una cultura y Scrum es un proceso formalizado. La premisa original de Scrum era acelerar la entrega del producto. Sin embargo, los equipos de Scrum se reúnen regularmente para la planificación de iteraciones, actualizaciones de estado diarias y revisiones de sprint. Esto puede distraer fácilmente a los equipos y hacer que pasen más tiempo informando lo que hacen, en lugar de ejecutarlo, seguir religiosamente una metodología basada en sprints como Scrum, en última instancia, puede obstaculizar la agilidad general del equipo. Aún así, los beneficios de Scrum generalmente constituyen un gran ahorro de tiempo en las especificaciones y

menos traspasos debido a los equipos multifuncionales. Finalmente, Scrum ofrece más flexibilidad en la planificación debido a los sprints cortos.

Tablero Kanban

Este método fue utilizado por primera vez por la firma japonesa Toyota, se trata de un panel, físico o virtual, el cual se divide en varias partes y cada una de estas se identifica la situación en la que se encuentran las tareas a desarrollar. De esta manera, se puede ver e identificar de una forma rápida el punto del proceso en el que se encuentran las diferentes tareas, facilitando información sobre los cuellos de botella en la ejecución, mejora la comunicación y ayuda a la toma de decisiones, la perspectiva basada en el flujo de Kanban puede mejorar y complementar el marco Scrum y su implementación (Scrum.org, 2018).

En principio Kanban es una estrategia para optimizar el flujo de valor de las partes interesadas a través de un proceso que utiliza un sistema de visualización del trabajo en progreso (Zayat y Senvar, 2020).

Scrum se basa en la teoría de control de procesos empíricos, tres pilares sostienen toda su implementación: transparencia, inspección y adaptación. Scrum exige que el *Sprint Backlog* sea transparente, pero proporciona una guía limitada sobre cómo lograrlo. Tampoco define cómo lograr una transparencia explícita en el flujo de trabajo desde el *Product Backlog* hasta el *Sprint Backlog*, y lo que suceda con el trabajo después de que se convierte en un incremento "Terminado". Aquí es donde Kanban puede ayudar, al visualizar el trabajo de nuevas formas (Lei y otros, 2017).

El modelo de ingeniería de Spotify[i]

El modelo de ingeniería de Spotify, como método ágil, atrae cada vez más a los profesionales. La idea principal del método es crear *squads* autónomos pero colaborativos. Sin embargo, un desafío observado de este método es, cómo lograr el equilibrio correcto entre la autonomía de los *squads* y la colaboración efectiva entre ellos (Salameh y Bass, 2020).

El objetivo es implementar lo que Kamer (2018) llama “burocracia mínima viable” para lograrlo se parte de la mejora continua, el desarrollo iterativo, la simplicidad, la confianza y el liderazgo de servicio.

Los componentes de este modelo de ingeniería según Kendis Team (2018) son:
1. *Squads:* Similar a los equipos de Scrum, Spotify tiene *squads*. En una organización, puede haber varios equipos formados por 6 a 12 personas, cada uno dedicado a trabajar en un área funcional. Un *squad* es autónomo, auto-organizado y auto-gestionado. Cada *squad* tiene una misión que cumplir y es libre de elegir la metodología para llevar a cabo sus funciones, para lograr entregas rápidas y frecuentes, los *squad* aplican la técnica del Producto Mínimo Viable (MVP).

Los equipos cuentan con un *Agile coach* que ayuda a mejorar su forma de trabajar. Hay un *Product Owner* que define la visión de las funciones. El *Agile Coach* realiza retrospectivas, mientras que las reuniones de planificación de *sprints* se mantienen como opcionales. Los equipos tienen contacto directo con el cliente.

Dev y *ops* funcionan de tal manera que las operaciones no interfieran con el trabajo de los desarrolladores. Más bien, los operadores allanan el camino para que los desarrolladores publiquen el trabajo por sí mismos. Esto se hace mediante la creación de un entorno como la construcción de una infraestructura y el soporte para permitir que los desarrolladores inicien su trabajo.

2. *Tribes*: Múltiples escuadras que trabajan en un área característica relacionada forman una *tribe*. Una *tribe* puede constar de 40 a 100 personas. Una *tribe* tiene un líder, que es responsable de crear un entorno productivo e innovador para los *squads*, el líder de la *tribe* también puede ser parte de un *squad*.
3. *Chapter*: Es un nivel horizontal de la organización, consta de individuos de diferentes *squads* que se agrupan por su especialización o rol. Un líder de *chapter*. que también puede ser un gerente, apoya a los miembros en su crecimiento personal y desafíos específicos. El líder del *chapter* tiene reuniones semanales en las que todos los miembros del mismo presentan los problemas y discuten su solución.

4. *Guilt*: Un grupo informal constituido por personas que tienen un interés común, forman un *guilt*. Una persona de cualquier *squad*, *chapter* o *tribe* puede ser parte de un *guilt* .

En principio el propósito de un *chapter* y un *guilt* es el mismo; asegurar la transparencia, resolver problemas y mantener a los equipos alineados y enfocados. Por ejemplo, hay un desarrollador *backend* del equipo A que tiene un problema. Hay otro desarrollador *backend* en el equipo B que ha resuelto fácilmente el mismo problema, si ambos están en el mismo *chapter*, pueden compartir su problema y encontrar una solución. Si están en diferentes *chapters*, tienen entonces la opción de pertenecer a un *guilt* de desarrolladores *backend*, donde pueden compartir sus conocimientos y ayudarse.

5. *Trio*: Un trío se forma cuando para cada *tribe* hay líderes de producto, diseño y *tribe*.
6. *Alliance*: Una combinación de tres *trios* hace una *alliance*. Tiene líderes de producto, diseño y *tribe*.
7. *Chief architect*: Por último, un miembro crucial de la organización es el *chief architect* que define la visión arquitectónica, guía en los diseños y se ocupa de los problemas de la arquitectura del sistema. Para realizar un seguimiento de los problemas de arquitectura, cada *squad* tiene un *product owner* que maneja las cuestiones relacionadas con la arquitectura y controla la entrega frecuente de las versiones de cada producto.

La base de las *tribes* de Spotify es la autonomía y la confianza. Cuando hay confianza, hay sentido de propiedad y responsabilidad por el trabajo realizado. La confianza ayuda a crear un entorno en el que el fracaso se toma como una oportunidad para aprender, innovar y cambiar en consecuencia. Esto también eleva la moral y el crecimiento individual (Mankins y Garton, 2017).

El modelo de ingeniería de Spotify es un modelo de desarrollo adecuado para nuestros fines, se tomará de él la base para su posterior adaptación, se puede confiar en el mismo para superar algunas de nuestras debilidades, además de nuestras fortalezas para la elaboración de nuestro propio modelo.

Resultados y discusión

Se defiende el concepto de metodología ágil pragmática. Se han estado usando casi intuitivamente principios basados en ella, desde muy temprano en proyectos para el Centro Nacional de Tecnologías de Información (CNTI) en la República Bolivariana de Venezuela y en el proyecto para el desarrollo del Sistema para el registro y control de los procesos de la vida interna para el Comité Central del Partido.

Se ha visto que muchas organizaciones adoptan ciegamente las metodologías ágiles sin comprender por qué lo hacen o sin comprender los problemas que intentan solucionar, en nuestro caso se llevó a cabo una evaluación crítica de las metodologías, con criterio contexto-céntrico, antes de proceder a su adopción o adaptación, siempre con enfoque de innovación.

Para nosotros es mucho más importante mantener el enfoque ágil, que poseer el control que ofrecen las herramientas tradicionales. Básicamente, se apuesta por la innovación y no por la planificación estricta. No obstante, esta apuesta exige un continuo aprendizaje como método de mejora y progreso alineado y promovido por la autonomía de sus integrantes para resolver problemas y consecuentemente, agregar valor al producto. Así pues, Z-17 se agrupa en equipos de 6 y 7 personas auto-organizados, con plenas capacidades para tomar decisiones correctas y autónomas.

Las oficinas de Z-17 están diseñadas para fomentar la colaboración entre los distintos miembros de los equipos, los cuales trabajan estrechamente y se reúnen en un salón de estar para comentar las diferentes retrospectivas, que suelen escribirlas en una pizarra para darles mayor visibilidad.

Revista Cubana de Ciencias Informáticas
Vol. 16, No. 4, Octubre-Diciembre, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 18-34

Fig. 1- Espacios de trabajo colectivo en Z-17.

En Z-17 se usan los conceptos de *squad*, *chapter* y *tribe*, como adaptación creativa del modelo de ingeniería de Spotify, donde más que adaptar una estructura, se trata de crear una cultura ingenieril. Se cuenta con una alta dirección que valora y apoya el marco ágil y comprende la importancia de la implementación de los conceptos de *squad*, *sprints*, retrospectiva, cohesión del equipo, etc.

Todos los *squads* son consistentes, flexibles y están alineados entre sí, relacionado con esto está la noción de responsabilidad, que podríamos definir como "autoorganización". Los *product owner* desempeñan un papel de puente entre las partes interesadas y el equipo de desarrollo.

Se han establecido tres *squads*, uno para la plataforma de mensajería instantánea y colaborativa *(toDus®)* otro para el Centro de Aplicaciones Android *(Apklis®)* y otro para la plataforma cubana de contenidos audiovisuales *(Picta®)*, los cuales de conjunto con los *tribe leads* conforman la *tribe Z*-17.

Editorial "Ediciones Futuro"
Universidad de las Ciencias Informáticas. La Habana, Cuba
rcci@uci.cu

Revista Cubana de Ciencias Informáticas
Vol. 16, No. 4, Octubre-Diciembre, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 18-34

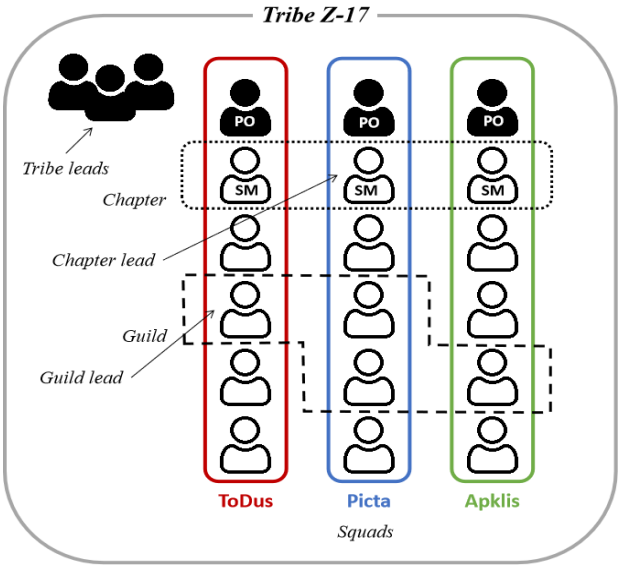

Fig. 2- Adaptación del modelo *Squad-Chapter-Tribe* de Spotify en Z-17.

Fuente: Elaboración propia a partir de (Kendis Team, 2018)

Se ha notado que, en comparación con otras organizaciones, aún no se tienen definidos todos los *scrum master* y *product owner*. En Z-17 se nota que el papel de *product owner* es más común, pero *scrum master* no, puede ser porque uno de los primeros *product owner* tenía habilidades de *scrum master* y asumió el rol, se hace necesario potenciar el rol de *scrum maste*r, aunque no como *scrum master* de tiempo completo.

Para mantener la cohesión se necesita tener una gran habilidad de comunicación, la forma en que se habla con ingenieros, diseñadores y empresarios es diferente, se necesita tiempo para cohesionarlos, cada equipo tiene sus características propias. Los desafíos son mantenerlos sólidos, responsables y con sentido de pertenencia, mientras se mantienen alineados y enfocados, para ello se debe comunicar y divertirse, al mismo tiempo entregar el valor necesario en cada sprint.

Editorial "Ediciones Futuro"
Universidad de las Ciencias Informáticas. La Habana, Cuba
rcci@uci.cu

Revista Cubana de Ciencias Informáticas
Vol. 16, No. 4, Octubre-Diciembre, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 18-34

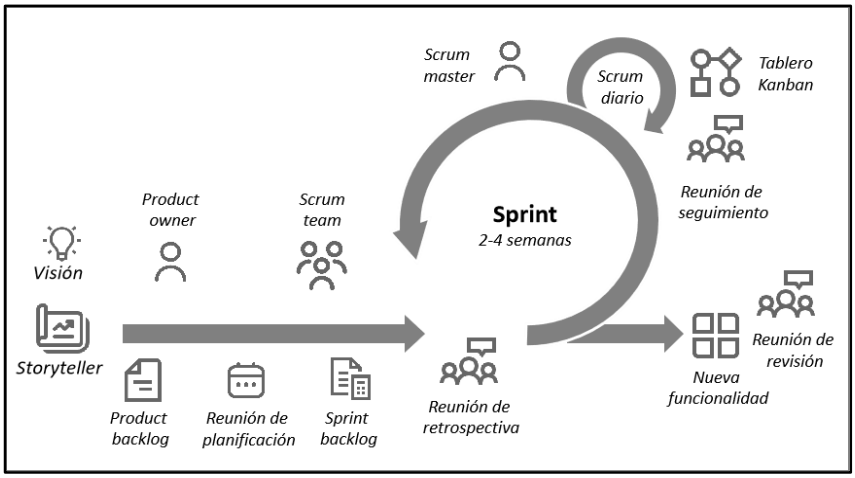

Fig. 3- Adaptación del Proceso Scrum en Z-17.

Fuente: Elaboración propia a partir de (Schwaber y Sutherland, 2017)

En Z-17 se ha sustituido el habitual gráfico de *burndown*, una herramienta que permite visualizar la cantidad de trabajo remanente en los *sprints*, por un tablero Kanban personalizado.

Fig. 4- Tablero Kanban en Z-17.

Se encuentra a menudo que es difícil desarrollar un "sentido de pertinencia", las cuestiones que han contribuido en ese sentido en Z-17 son: Brindar al equipo un trabajo emocionante y de alta responsabilidad; incluir al equipo en las decisiones importantes; incluir los *squads* en el diseño de los *sprints*; depositar

Editorial "Ediciones Futuro"
Universidad de las Ciencias Informáticas. La Habana, Cuba
rcci@uci.cu

confianza en ellos; compartir la situación actualizada de las negociaciones, para que el equipo se sienta confiable e involucrado. Es importante compartir y divertirse con los miembros del equipo, para entender su mundo, de esa manera no verán la figura de un jefe o gerente, se transmite la imagen de un líder servidor. Se les ayuda tanto como se pueda. Cosas simples como adquirir por iniciativa colectiva medios para mejorar las condiciones de trabajo, hacen que aumente la cohesión y la productividad.

Esta experiencia es susceptible de generalizarse para equipos pequeños de desarrollo de software que actúan en el entorno de las relaciones universidad-empresa o vinculados a una organización de interfaz.

Conclusiones

Se llevó a cabo una evaluación crítica de las tecnologías y se adaptaron para crear un modelo propio de equipo de desarrollo, en el contexto de la organización (UCI), de la cual se toman, su cultura, objetivos y muchos otros factores que determinan su proyección, se investigó y tomaron prestadas ideas de organizaciones como Spotify, Google, Scrum.org y otras, pero solo como punto de partida.

Se definió un modelo propio que fomenta la gestión ágil, en el mismo se logran integrar algunos conceptos importantes, aunque no es en sí mismo un modelo pensado para el diseño organizacional, constituye una importante innovación organizacional de tipo incremental, por su aporte a la cultura ingenieril de los equipos de desarrollo de software en el entorno productivo de la UCI.

Se realizó el empoderamiento de los equipos de desarrollo inspirado en metodologías ágiles como Scrum y Kanban. El Modelo de ingeniería de Spotify, sin llegar a implementarse totalmente, aportó principalmente su cultura y el principio de la autoorganización de los equipos.

El trabajo realizado constituye una prueba realizada sobre una muestra de los equipos de desarrollo que actualmente se desempeñan en el entorno productivo de la UCI, para garantizar que la solución sea viable, antes de implementarla en toda la organización, este proceso debe repetirse en otros estudios de casos.

Referencias

Estrada-Velasco, Marco Vinicio, Et Al. Revisión Sistemática De La Metodología Scrum Para El Desarrollo De Software. Dominio De Las Ciencias, 2021, Vol. 7, No 4, P. 434-447.

Kamer, J. How To Build Your Own "Spotify Model". [En Línea]. Medium 2018 Disponible En: Https://Medium.Com/The-Ready/How-To-Build-Your-Own-Spotify-Model-Dce98025d32f

Kendis Team. Exploring Key Elements Of Spotify's Agile Scaling Model. [En Línea]. Medium, 2018. Disponible En: Https://Medium.Com/Scaled-Agile-Framework/Exploring-Key-Elements-Of-Spotifys-Agile-Scaling-Model-471d2a23d7ea

Kniberg, Henrik, Et Al. Kanban Y Scrum–Obteniendo Lo Mejor De Ambos. C4media Inc, 2010.

Lei, Howard, Et Al. A Statistical Analysis Of The Effects Of Scrum And Kanban On Software Development Projects. Robotics And Computer-Integrated Manufacturing, 2017, Vol. 43, P. 59-67.

Lugo-García, José Alejandro, Et Al. Proceso Para La Planificación Y Control De Proyectos De Software Utilizando Xedro-Gespro. Revista Cubana De Ciencias Informáticas, 2014, Vol. 8, No 2, P. 144-161.

Mankins, Michael; Garton, Eric. How Spotify Balances Employee Autonomy And Accountability. Harvard Business Review, 2017, Vol. 95, No 1.

Morandini, Marcelo, Et Al. Considerations About The Efficiency And Sufficiency Of The Utilization Of The Scrum Methodology: A Survey For Analyzing Results For Development Teams. Computer Science Review, 2021, Vol. 39, P. 100314.

Salameh, Abdallah; Bass, Julian M. Heterogeneous Tailoring Approach Using The Spotify Model. En Proceedings Of The Evaluation And Assessment In Software Engineering. 2020. P. 293-298.

Schwaber, Ken; Sutherland, Jeff. The Scrum Guide. [En Línea]. 2017. 2017, P. 6-17. Disponible En: Https://Www.Scrumguides.Org/Docs/Scrumguide/V2017/2017-Scrum-Guide-Us.Pdf#Zoom=100

Scrum.Org. The Kanban Guide For Scrum Teams. 2018. [En Línea]. Disponible En: Https://Scrumorg-Website-Prod.S3.Amazonaws.Com/Drupal/2018/4/2018%20kanban%20guide%20for%20scrum%20teams_0.Pdf

Sosa González, Rosel, Et Al. Ecosistema De Software Gespro-16.05 Para La Gestión De Proyectos. Revista Cubana De Ciencias Informáticas, 2016, Vol. 10, P. 239-251.

Sospedra López, Diannet; Ramírez Pérez, José Felipe; Feria, Gutiérrez Luz María . La Integración En La Gestión De Proyectos: Diagnóstico Y Buenas Prácticas A Implementar En La Uci. Serie Científica De La Universidad De Las Ciencias Informáticas, 2017, Vol. 10, No 3.

Wonohardjo, Eduard Pangestu; Sunaryo, Rizky Febriyanto; Sudiyono, Yusuf. A Systematic Review Of Scrum In Software Development. Joiv, 2019, Vol. 3, No 2, P. 108-112.

Zayat, Wael; Senvar, Ozlem. Framework Study For Agile Software Development Via Scrum And Kanban. International Journal Of Innovation And Technology Management, 2020, Vol. 17, No 04, P. 2030002.

Conflicto de interés

El autor autoriza la distribución y uso de su artículo.

Contribuciones de los autores

Conceptualización: Allan Pierra Fuentes

Curación de datos: Haniel Cáceres Navarro

Análisis formal: Yadier Perdomo Cuevas

Adquisición de fondos: Yanio Hernández Heredia

Investigación: Allan Pierra Fuentes

Metodología: Allan Pierra Fuentes

Administración del proyecto: Allan Pierra Fuentes

Recursos: Yanio Hernández Heredia

Software: Yadier Perdomo Cueva

Supervisión: Raydel Montesinos Perurena

Validación: Yanio Hernández Heredia

Visualización: Haniel Cáceres Navarro

Redacción – borrador original: Allan Pierra Fuentes

Redacción – revisión y edición: Raydel Montesinos Perurena

Revista Cubana de Ciencias Informáticas
Vol. 16, No. 4, Octubre-Diciembre, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 18-34

[i] Spotify Technology S.A., es una compañía proveedora de transmisión de música en todo el mundo que presta un servicio de música digital a través de una plataforma accesible desde una computadora, teléfono inteligente u otros dispositivos electrónicos. Spotify opera en más de sesenta países ofreciendo una variedad de estilos de música a cualquier hora y en cualquier lugar a sus más de 140 millones de usuarios.