

**UNIVERSIDAD DEL CEMA
Buenos Aires
Argentina**

Serie
DOCUMENTOS DE TRABAJO

Área: Negocios

**METODOLOGÍAS ÁGILES EN EQUIPOS DE OPERACIONES
DEL ÁREA DE TECNOLOGÍA DE LA INFORMACIÓN (TI)**

Pablo Cababie y Fernando Troilo

**Marzo 2021
Nro. 783**

**www.cema.edu.ar/publicaciones/doc_trabajo.html
UCEMA: Av. Córdoba 374, C1054AAP Buenos Aires, Argentina
ISSN 1668-4575 (impreso), ISSN 1668-4583 (en línea)
Editor: Jorge M. Streb; asistente editorial: Valeria Dowding jae@cema.edu.ar**

Metodologías ágiles en equipos de operaciones del área de tecnología de la información (TI) *

Mgter. Pablo Cababie ¹

Dr. Fernando Troilo ²

Resumen

El propósito del presente trabajo consistió en mostrar el recorrido de una implementación ágil en un equipo que no efectuaba tareas de desarrollo de software como el que cita la metodología de Scrum en su forma más primitiva. Inicialmente, se brindó un marco teórico para partir de estudios y experiencias anteriores. La bibliografía citada cubrió una amplia gama de publicaciones y autores, desde los creadores de los principios ágiles hasta investigadores que propusieron nuevas ideas para optimizar los tiempos de este cambio radical en la forma de trabajo. A lo largo del presente trabajo, se propusieron diferentes actividades utilizadas para la transformación de un equipo de soporte de tecnología en el área de TI de una corporación multinacional. Para cada experiencia o ejercicio, se efectuó un análisis previo y una evaluación posterior a fin de medir su efectividad. Asimismo, el trabajo enmarcó las métricas como pieza fundamental dentro de las transformaciones ágiles, ya que permiten censar el estado de estas y definir las áreas que faltan desarrollar o mejorar.

Palabras clave: Agilidad organizacional, Scrum, transformación cultural.

* Los puntos de vista son de los autores y no representan la posición de la Universidad.

¹ Magíster en Dirección de Empresas, UCEMA.

² Director del Posgrado en Recursos Humanos, Profesor del MBA y carreras de grado de UCEMA. Consultor en Capital Humano.

INTRODUCCIÓN

El presente trabajo analiza la implementación de metodologías ágiles en un equipo del área de tecnología de una empresa multinacional del sector energético y describe los vaivenes del proceso de conformación de dicho equipo hasta convertirse en una unidad productiva que entrega valor con confianza dentro del marco corporativo.

En el origen de las metodologías ágiles, se encuentra el desarrollo de software. La literatura señala casos concretos aplicados a este tipo de actividad y ejemplos relacionados. En este caso particular, el equipo efectúa tareas de soporte de tecnologías específicas en sistemas operativos Windows y de implementación de nuevas soluciones, por lo tanto, es difícil encontrar ejemplos similares.

El desafío planteado parte de una implementación “más libre”, según las actividades y los objetivos particulares del equipo. Por eso, no existe una receta o un procedimiento estricto y riguroso que asegure su éxito, como sí ocurre con otro tipo de cambios organizacionales. Dado que la literatura no cubre casos tan específicos ni situaciones aplicadas a las tareas y a la organización de una industria de tal magnitud corporativa, el caso requiere metodologías creativas e innovadoras alineadas con la cultura de la empresa, la metodología en sí y el tipo de tareas realizadas.

El desarrollo de este trabajo presenta las estrategias que resultaron efectivas y los ejercicios de implementación de dichas estrategias, así como las técnicas que, en otros equipos, fueron catalizadores del cambio y, en este caso particular, no tuvieron el impacto esperado.

El objetivo principal del presente trabajo es describir el proceso de transformación ágil de un equipo de tecnología de inicio a fin con sus aciertos y desaciertos. Asimismo, se plantean dos objetivos secundarios: brindar una referencia para futuras implementaciones tanto dentro como fuera de la compañía y listar las prácticas y los experimentos realizados para facilitar la transformación efectiva y asegurar la integración del equipo hasta convertirse en un equipo ágil.

MARCO TEÓRICO

Las metodologías ágiles

Las metodologías ágiles toman fuerte impulso y popularidad desde la publicación del Manifiesto Ágil (2001). Un grupo de consultores y expertos que trabajaban en proyectos de desarrollo de software se reunió para definir y acordar las bases de la guía fundamental de recomendaciones, valores y principios comunes para las metodologías ágiles.

En el Manifiesto Ágil (2001), se enumeran los siguientes valores:

- Individuos e interacciones sobre procesos y herramientas.
- Software de trabajo sobre documentación integral.
- Colaboración del cliente sobre negociación de contratos.
- Responder al cambio sobre seguir un plan.

Estas metodologías están regidas, además, por doce principios que ayudan a que el proceso de desarrollo se vuelva menos complejo y a responder de manera oportuna ante los cambios que surgen a lo largo de este, siempre considerando el punto de vista del cliente.

Las metodologías ágiles abarcan métodos, técnicas y herramientas de gestión de proyectos que buscan mejorar continuamente los productos, garantizar la eficiencia de la finalización del proyecto, el rendimiento de los equipos autoorganizados y la entrega de productos de calidad.

“A diferencia de las metodologías tradicionales, las metodologías ágiles abordan la imprevisibilidad confiando en las personas y su creatividad, más que en los procesos” (Cockburn, 2001, pp. 131-133.). Esta afirmación deja en evidencia la debilidad de las metodologías tradicionales, ya que, para ellas, la imprevisibilidad es un obstáculo; en cambio, las metodologías ágiles la incorporan como parte del mismo proceso de trabajo.

Según Yordanova (2019), el empleo de metodologías ágiles en la gestión ayuda a los gerentes a alcanzar los objetivos rápidamente, reducir los riesgos y crear un entorno de trabajo y gestión ágil y eficaz, en el que los equipos puedan responder a la imprevisibilidad mediante iteraciones y comentarios de los clientes. Por lo tanto, se utilizan metodologías ágiles para desarrollar productos de software cuando los elementos y funciones de un producto no son

familiares al comienzo del proyecto, y se alienta a los clientes a participar en el proceso junto con los desarrolladores de software.

De este modo, el principal enfoque de los métodos ágiles es la rapidez de la entrega de valor (Makoto, 2017), que se incrementa en cada nueva iteración y asegura la calidad del producto final.

La metodología Scrum

Las dos metodologías ágiles más populares de gestión de proyectos que utilizan las empresas de tecnología de la información (TI) para mejorar sus procesos de gestión cuando desarrollan productos y servicios son Scrum y Kanban.

De acuerdo con Demigha (2019), Scrum es una metodología ágil de entrega de productos iterativa e incremental, que utiliza retroalimentación frecuente y toma de decisiones colaborativa. Si bien, a simple vista, todas las metodologías ágiles parecen similares, Scrum es la más popular. Relativamente fácil de implementar, puede usarse para resolver numerosos problemas que enfrentan los desarrolladores de software, como ciclos de producción complejos, planificación ágil de proyectos, producción retrasada de un artículo, etc. Permite desarrollar y probar software rápidamente en pequeños equipos de proyecto.

El objetivo de esta metodología, como mencionan Schwaber y Beedle (2001), consiste en elaborar uno o varios productos que cumplan las necesidades solicitadas por el cliente. A la hora de desarrollar este producto, es muy importante que se descomponga en pequeñas partes que tengan entidad propia y se correspondan con una historia de usuario, es decir, una funcionalidad que aporte un valor concreto al usuario. Sobre estos fragmentos, se deben poder ejecutar todos los pasos de un ciclo ágil o Sprint; en caso contrario, es necesario revisar su tamaño (Palacios, 2014).

Jeff Sutherland, creador de la metodología Scrum, basó su desarrollo en el trabajo de Takeuchi y Nonaka (1986). De este modo, Sutherland define una forma de desarrollar software que se orienta a obtener un producto más eficaz, brindando autonomía y propósito a los equipos, y donde los managers actúen sin decir qué hacer, sino eliminando impedimentos (Schwaber y Sutherland, 2013).

Scrum se funda sobre los cimientos de un concepto acuñado por Edward Deming y que desarrollaron los japoneses, el ciclo PDCA (Deming, 1982), que mejora cualquier proceso significativamente en unos pocos ciclos.

Es entendible que, para quien estuvo acostumbrado a desarrollar software durante mucho tiempo con el modelo tradicional (cascada), pueda resultar un desafío importante este cambio en la forma de encarar un proyecto de desarrollo de software. Por eso, la bibliografía y los expertos sugieren aprender Scrum como se aprenden artes marciales: implementando el concepto de Shu-Ha-Ri (Capobianco, 2016; Vaasanthi, 2017):

- Shu: Repetir movimientos (forma) para interiorizarlos.
- Ha: Introducir variaciones propias.
- Ri: Abandonar la forma, se ha interiorizado la esencia.

Características de la metodología

Los artefactos de Scrum son aquellos elementos de la metodología que brindan transparencia y oportunidades para la inspección y la adaptación. Están diseñados para maximizar la transparencia de la información clave a fin de que todos tengan la misma comprensión sobre el producto.

En esta instancia, es interesante destacar que las porciones de trabajo son las que más adelante construyen un mínimo producto viable (MVP en inglés), con todas las historias de usuarios principales que haya solicitado el cliente. Es muy importante que cada una de las partes o historias de usuario que componen un producto sean validadas por el cliente durante la demostración del Sprint presentada por el equipo que las ha trabajado.

Una vez finalizado el mínimo producto viable, el cliente comienza a trabajar en su funcionalidad básica. Se continúa el ciclo ágil a partir de las historias de usuario adicionales, que permiten una mejora incremental del producto.

A continuación, se listan los elementos indispensables para implementar la metodología Scrum. Estos componentes son los que se utilizan con más frecuencia, pero pueden existir variaciones, adaptaciones y cambios según la implementación.

Backlog del producto

El backlog del producto (lista de requisitos priorizada para entregar el producto) describe los requisitos y los elementos del producto, y enumera las funciones deseadas, clasificándolas según su prioridad. Está disponible para todos los interesados que puedan agregarle información. El dueño del producto es responsable de organizar el contenido de la cartera de productos.

La lista de requisitos priorizada representa las expectativas del cliente respecto a los objetivos y entregas del producto o proyecto. El cliente es el responsable de crear y gestionar la lista (con la ayuda del Facilitador/Scrum Master) y del equipo, que proporciona el esfuerzo estimado para completar cada requisito. Al reflejar las expectativas del cliente, esta lista permite involucrarlo en la dirección de los resultados del producto o proyecto. En otras palabras, el backlog contiene los requisitos de alto nivel del producto o proyecto. Para cada requisito, se indica el valor que aporta el cliente y el costo estimado para completarlo. En la lista, se prioriza cada requisito según el valor que aporta al negocio frente al tiempo estimado de su desarrollo.

En la lista, también se indican las posibles iteraciones y las entregas esperadas por el cliente (los puntos en los cuales desea que le entreguen los requisitos completados hasta ese momento) en función de la velocidad de desarrollo del equipo que trabaja en el proyecto. Dentro de esta, se deben considerar los riesgos del proyecto e incluir los requisitos o tareas necesarios para mitigarlos.

Antes de iniciar la primera iteración o ciclo, el cliente debe tener la meta del producto o proyecto definida, y la lista de requisitos creada. No es necesario que la lista sea completa ni que todos los requisitos estén detallados al mismo nivel. Basta con que los requisitos más prioritarios con los que el equipo debe empezar a trabajar estén identificados con suficiente detalle. Los requisitos de iteraciones futuras pueden ser mucho más amplios y generales. La incertidumbre y la complejidad propia de un proyecto hacen conveniente no detallar todos los requisitos hasta que su desarrollo esté próximo. De esta manera, el esfuerzo de recoger, detallar y desarrollar el resto de los requisitos (menos prioritarios) se reparte durante el período de ejecución del proyecto.

Las ventajas:

- Se evita caer en parálisis por análisis al inicio del proyecto, de manera que se inicie antes el desarrollo y el cliente pueda empezar a obtener resultados útiles.
- Se evita analizar en detalle requisitos no prioritarios que pueden cambiar durante el transcurso del proyecto, cuando se conozca mejor cuál ha de ser el resultado, o bien porque pueden ser reemplazados por otros.
- Se puede llegar a un punto del proyecto en que no valga la pena analizar ni desarrollar los requisitos restantes. En el caso del desarrollo de un producto, la lista va evolucionando

durante toda la vida de este. En el caso de un proyecto, conforme avance, aparecerán los requisitos menos prioritarios que falten. El cliente y el equipo tienen que acordar la definición de “completado” de los requisitos si el equipo considera que el producto está preparado para ser entregado al cliente al finalizar cada iteración, de manera que no haya tareas pendientes que impidan utilizar los resultados del proyecto lo antes posible. De este modo, el cliente puede tomar decisiones correctas cuando, al final de cada iteración, el equipo le hace una demostración de los requisitos completados (por ejemplo, solicitar una entrega del producto). Cuando el cliente solicita una entrega de los requisitos completados hasta ese momento, el equipo puede necesitar añadir una iteración de entrega más corta que las habituales, donde realiza alguna tarea no necesaria o posible hasta el momento de la entrega final.

Backlog del Sprint (lista priorizada de requisitos para una iteración)

En el backlog del Sprint (lista priorizada de requisitos para una iteración), se enumeran los requisitos y el contenido de cada Sprint. Además, constan los elementos seleccionados del backlog del producto que el equipo y el dueño del producto acordaron durante la reunión de planificación del Sprint. Cualquier cambio que pueda poner en peligro el logro de los objetivos es inaceptable durante el Sprint. Los objetivos principales son establecidos por el equipo durante dicha reunión.

Incrementos de funcionalidad del producto

Los incrementos de funcionalidad del producto son las versiones utilizables y factibles después del final de cada iteración. Se inspeccionan en cada reunión para supervisar el trabajo realizado durante el período.

Roles

Los tres roles principales de la metodología Scrum dividen de manera clara y simple las perspectivas y las responsabilidades.

La forma más simple de entender esta división es partir de la diferencia entre táctica y estrategia: táctica es la mejor manera que encontramos de ganar una batalla, y estrategia es la mejor elección de batallas que decidimos en pos de ganar la guerra. Es decir, en términos de un proyecto, la estrategia depende de qué características tenga el producto; y la táctica, de cómo se desarrollen dichas características.

Equipo de desarrollo

El equipo de desarrollo (o equipo a secas) es un grupo conformado, usualmente, con entre cinco y nueve personas que poseen las habilidades necesarias para construir un producto que cumpla con la visión definida por el dueño. Por ejemplo, para un desarrollo tradicional de software, se necesita que sus miembros tengan conocimientos sobre programación, testeo e incluso diseño gráfico y usabilidad, con lo cual el equipo debe ser multidisciplinario.

El equipo tiene la responsabilidad y la perspectiva de la tarea (el cómo de la construcción del producto), y es responsable de planificar un Sprint a la vez con el dueño del producto.

Cabe destacar en esta instancia que el mismo equipo es independiente en términos de decidir cómo desarrollar el incremento, pero funciona en un entorno altamente colaborativo. La metodología evita desplazar personas o dividirlos en equipos. Y los equipos tienen más éxito cuando se encuentran en una sala de equipo; en especial, durante los primeros Sprints.

Además, los miembros del equipo se encargan directamente de la mayor parte de la asignación de tareas, la gestión detallada diaria, los informes de progreso y el control de calidad del producto. Los equipos de proyecto deben adoptar diferentes reglas y aplicar diferentes prácticas. Estos argumentos conducen a la necesidad de una nueva organización del equipo. Por ejemplo, el caso de personas que trabajan en pareja y usan la misma computadora para resolver problemas difíciles (programación extrema).

Muchos equipos comienzan usando hojas de cálculo para administrar el trabajo atrasado del producto y listas de tareas para cambiar el estado de estas durante el Sprint actual; frecuentemente, utilizan una pizarra y notas adhesivas. Este enfoque tiende a funcionar bien en equipos pequeños y ubicados de manera conjunta. Sin embargo, a medida que aumenta la acumulación de requerimientos y los miembros remotos exigen la visibilidad del proyecto, muchas organizaciones implementan una herramienta más sofisticada que permite administrar proyectos de manera centralizada y la colaboración entre equipos.

Scrum Master

El Scrum Master es responsable de implementar la metodología y garantizar que los objetivos y los principios de Scrum sean entendidos y adoptados por los miembros del equipo. En otras palabras, es el encargado de asegurar que el equipo adopte las teorías, prácticas y reglas de la

metodología Scrum. Debe crear un entorno favorable para el equipo autoorganizado, proteger al equipo contra influencias externas o distracciones, promover la mejora de las prácticas de ingeniería. Aunque no está autorizado a administrar el equipo, ayuda a eliminar cualquier impedimento para el desarrollo del producto (Fustik, 2017). Dentro de este rol de facilitador, debe encargarse de que el equipo pueda cumplir con su compromiso y no merme su productividad, protegiéndolo de interrupciones externas que puedan afectar su compromiso o su productividad.

Product Owner

El Product Owner (dueño del producto) es la persona responsable de transmitir al equipo la visión del producto que se desea crear, aportando la perspectiva de negocio. Además, es el responsable de maximizar el retorno de la inversión (Fustik, 2017) y de la visión del producto. Asimismo, en términos de gestión, este líder constantemente vuelve a priorizar el backlog del producto al ajustar las expectativas a largo plazo. También este rol implica actuar como árbitro final en asuntos de requisitos y decidir si un producto está listo para ser lanzado. Además, el dueño del producto se encarga de establecer los objetivos del proyecto, manejar la compensación del cronograma versus el alcance, adaptarse a los requisitos cambiantes del proyecto y establecer prioridades para las características del producto.

Stakeholders

Los Stakeholders son el conjunto de personas que no forman parte del proceso de desarrollo de manera directa, pero deben ser tenidas en cuenta porque están interesadas en este, tales como directores o gerentes comerciales.

El cliente

En la mayoría de los casos, los clientes son los mismos usuarios. Al igual que los Stakeholders, no forman parte del proceso de creación de manera directa, pero pueden estar en la fase de revisión de entregables si se considera necesario. Son los destinatarios finales de la aplicación desarrollada, el público objetivo.

Cadencia

En la metodología Scrum, un proyecto se ejecuta en bloques temporales (iteraciones/ Sprints) de un mes natural (pueden ser de dos o tres semanas, si se necesita). Cada iteración debe

proporcionar un resultado completo, un incremento del producto susceptible de ser entregado con el mínimo esfuerzo cuando el cliente lo solicite.

La iteración/Sprint es el ritmo de los ciclos de Scrum. Está delimitado por la reunión de planificación del Sprint y la reunión retrospectiva. Una vez fijada la duración del Sprint, es inamovible. La mayoría de los equipos eligen dos, tres o cuatro semanas de duración. A diario, durante el Sprint, el equipo realiza una reunión de seguimiento muy breve. Al final del Sprint, se entrega el producto al cliente con un incremento de la funcionalidad respecto del inicio del Sprint. El proceso parte de la lista de requisitos priorizada del producto, que actúa como plan del proyecto. En dicha lista, se da prioridad a los requisitos del cliente según el valor que le aportan en relación con su esfuerzo, y se distribuyen en iteraciones y entregas.

Eventos de la metodología Scrum

Los eventos de la metodología Scrum se utilizan para garantizar la regularidad y minimizar la necesidad de reuniones no definidas en Scrum. Una vez comenzado un Sprint, su duración es fija: no se puede acortar ni alargar. Los eventos pueden finalizar siempre que se haya logrado su propósito y que pase una cantidad de tiempo adecuada sin ningún desperdicio en el proceso.

Son fundamentales los siguientes eventos de Scrum: planificación del Sprint, Scrum diario, reunión de revisión de Sprint, reunión retrospectiva de Sprint, reuniones de refinamiento del backlog.

Planificación del Sprint

Antes del comienzo de cada Sprint, se celebra una reunión de planificación de Sprint entre el dueño del producto y el equipo para identificar los elementos del producto que se desarrollarán (backlog del producto). El dueño del producto especifica qué elementos son importantes para el negocio, mientras que el equipo de desarrollo elige cuáles se desarrollarán durante el Sprint. Por lo tanto, se crea el backlog de iteración para describir los requisitos del Sprint y los elementos seleccionados para su implementación.

El cliente le presenta al equipo la lista de requisitos priorizados del producto o proyecto, le pone un nombre a la meta de la iteración (de manera que ayude a tomar decisiones durante su ejecución) y propone los requisitos más prioritarios para desarrollar.

Durante este evento, el equipo examina la lista, le pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita. Luego, se planifica la iteración, en base a las prioridades, organizando el trabajo de forma óptima. Por lo tanto, en esta ceremonia, el equipo define las tareas necesarias para completar cada requisito comprometido y efectúa una estimación del esfuerzo para llevarlas a cabo. Cabe aclarar que esta es más fiable, dado que tiene en cuenta los diferentes conocimientos, experiencia y habilidades de los integrantes del equipo. Por último, se asignan las historias de usuario a cada miembro del equipo.

Las ventajas de esta dinámica:

- Empoderamiento del equipo para organizar el trabajo, ya que es quien mejor conoce cómo realizarlo.
- El equipo define las tareas necesarias para completar cada requisito, creando la lista de historias de usuario de la iteración.
- Realiza una estimación conjunta del esfuerzo necesario para realizar cada tarea, potenciando el compromiso de cada miembro con el equipo
- Cada uno de los miembros se compromete a realizar las tareas asignadas.

Scrum diario (daily standup)

El objetivo de esta reunión es facilitar la transferencia de información y la colaboración entre los miembros del equipo para aumentar su productividad. Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencia entre tareas, progreso hacia el objetivo de la iteración, obstáculos para alcanzar el objetivo). Al finalizar la reunión, se hacen las adaptaciones necesarias para cumplir con el compromiso conjunto que el equipo adquirió para la iteración (en la reunión de planificación de la iteración).

Cada miembro del equipo debe responder en un intervalo máximo de 15 minutos las siguientes preguntas:

1. ¿Qué he hecho desde la última reunión de sincronización? ¿Pude hacer todo lo que tenía planeado? ¿Cuál fue el problema?
2. ¿Qué voy a hacer a partir de este momento?

3. ¿Qué impedimentos tengo o voy a tener para cumplir mis compromisos con esta iteración y con el proyecto?

Como apoyo, el equipo cuenta con la lista de tareas de la iteración, donde se actualiza el estado y el esfuerzo pendiente para cada tarea, así como con el gráfico de horas pendientes de la iteración.

Los problemas que requieren mayor consideración solo pueden discutirse después de que cada miembro del equipo haya presentado sus informes. El equipo puede usar una lista de tareas actuales. Durante el Sprint, se identifican las nuevas tareas necesarias para lograr el objetivo del Sprint. El objetivo del marco Scrum es eliminar los anteriores hábitos de trabajo independiente, por lo tanto, los miembros del equipo deben tener mucho cuidado de no permitir ninguna instancia con un enfoque similar.

Reunión de revisión de Sprint

Esta ceremonia es una reunión informal donde el equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento del producto preparado para ser entregado con el mínimo esfuerzo, haciendo un recorrido lo más real y cercano posible al objetivo que se pretende cubrir. De este modo, los equipos celebran una reunión al final de cada Sprint para revisar el progreso: los asistentes colaboran con lo que se ha hecho, y el dueño del producto revisa los elementos seleccionados en la reunión de planificación de Sprint para evaluar cuáles se han logrado. Los elementos no logrados se incluyen en la nueva lista para ser priorizados en futuros Sprints.

En este evento, el rol del Scrum Master es ayudar al dueño del producto y a los usuarios finales del producto a transformar sus comentarios en nuevos requisitos, que luego serán priorizados por el dueño del producto. Con frecuencia, los nuevos requisitos pueden ir más allá del alcance del proyecto. Cuando esto ocurre, el nuevo alcance del proyecto puede reemplazar totalmente al anterior si el dueño del producto considera que el nuevo alcance es más importante que las expectativas iniciales. A una reunión de revisión de Sprint pueden asistir participantes externos e incluso usuarios finales.

Las ventajas de esta dinámica:

- El cliente puede evaluar de manera objetiva cómo se desarrollaron los requisitos que proporcionó y si se cumplen sus expectativas, entender qué más necesita y tomar mejores decisiones respecto al proyecto.

- El equipo puede validar si realmente entendió los requisitos del cliente y en qué puntos mejorar la comunicación entre ambos.
- El equipo se siente más satisfecho cuando puede ir mostrando los resultados que va obteniendo. No trabaja durante meses sin poder exhibir su obra.

Reunión retrospectiva de Sprint

Cada Sprint finaliza con una reunión retrospectiva de Sprint en la que el equipo revisa su desempeño y analiza cómo ha sido su manera de trabajar durante la iteración, qué ha funcionado bien, qué hay que mejorar, qué hay que probar en la siguiente iteración, qué se ha aprendido y cuáles son los problemas que podrían impedirle progresar adecuadamente, con el objetivo de mejorar de manera continua su productividad.

Una buena reunión retrospectiva requiere un entorno psicológico favorable, que excluya cualquier elemento de hostilidad o crítica destructiva. En la mayoría de las organizaciones, sin embargo, es difícil garantizar dicha atmósfera. Uno de los principales impedimentos para garantizar la transparencia total en las reuniones retrospectivas de Sprint es la presencia de personas que evalúen la implementación del proyecto.

Resulta fundamental el rol del Scrum Master para facilitar el establecimiento de una atmósfera positiva y constructiva dentro de la organización. Puede lograrlo mediante el empleo de diversas técnicas. El objetivo general de una reunión retrospectiva de Sprint es garantizar una conciencia común sobre numerosos puntos de vista e iniciar acciones que ayuden al equipo con el cumplimiento de las tareas. Este evento es clave dentro de la metodología, ya que permite incrementar la productividad y el aprendizaje del equipo de manera sistemática, iteración a iteración, con resultados a corto plazo.

Reuniones de refinamiento del backlog

Durante cada Sprint, los equipos de Scrum dedican algo de tiempo a refinar el backlog del producto. A la reunión asisten el equipo y el dueño del producto, quien lo aprueba y actualiza.

Los cambios en la lista de requisitos pueden deberse a las siguientes causas:

- Modificaciones que el cliente solicita tras la demostración que el equipo realiza al final de cada iteración sobre los resultados obtenidos, luego de que el cliente entiende mejor el producto o proyecto.

- Cambios en el contexto del proyecto (sacar al mercado un producto antes que un competidor, hacer frente a urgencias o nuevas peticiones de clientes, etc.).
- Nuevos requisitos o tareas como resultado de nuevos riesgos en el proyecto. Para realizar esta tarea, el cliente colabora con el equipo y obtiene la estimación de costos de desarrollo para completar cada requisito. El equipo ajusta el factor de complejidad, el coste para completar los requisitos y la velocidad de desarrollo en función de la experiencia adquirida hasta ese momento en el proyecto. Cabe notar que el equipo sigue trabajando con los requisitos de la iteración en curso (de hecho, los más prioritarios al iniciar la iteración). No es posible cambiar los requisitos que se desarrollan durante la iteración. En la reunión de planificación de la iteración, el cliente presenta la nueva lista de requisitos para que sea desarrollada.

La principal ventaja (iteración a iteración) se encuentra en que el cliente puede tomar decisiones con tiempo respecto al progreso del proyecto y posibles desviaciones, permitiendo:

- Replanificar el proyecto para obtener un nuevo calendario de entregas que cumpla con las necesidades actuales.
- Incorporar nuevos recursos.
- Cancelar el proyecto con los requisitos completados hasta el momento plenamente operativos si el beneficio pendiente es menor que el coste de desarrollo.

El plan de proyecto se actualiza con la velocidad de desarrollo del equipo, entonces, se evitan sorpresas de última hora.

Factores que influyen la transformación ágil

Si bien el éxito de una transformación exitosa puede atribuirse a varios factores, los principales pueden agruparse dentro de estas dos categorías:

1. Los individuos/miembros del equipo que se responsabilizan de la adopción de la metodología y de que cada ceremonia tenga sentido, cumpla su propósito y sea efectiva.
2. El liderazgo, el cual asume la responsabilidad de su papel en la transformación, lidera con el ejemplo y brinda las herramientas para llevar a cabo la transformación.

Visión y objetivos claros

La mayoría de las transformaciones fallidas se deben a la falta de visión y objetivos. Las organizaciones no responden bien algunas preguntas. ¿Por qué queremos transformarnos en ágiles? ¿Qué queremos cambiar? ¿Por qué queremos cambiar?

La transformación no es un proceso individual, sino un esfuerzo conjunto del equipo. Por ejemplo, todos los evaluadores deben trabajar junto con los desarrolladores; todos los miembros del equipo deben estar familiarizados con la forma en que el cliente usa su producto y saber cómo implementarlo y brindarle soporte.

Aquí es necesario distinguir la adopción ágil de la transformación ágil. Si bien las personas y los equipos pueden iniciar la adopción mediante la práctica de principios ágiles, el liderazgo cumple el papel importante de mantenerlo, escalarlo y difundirlo en toda la organización. En otras palabras, los individuos de un equipo pueden comenzar una adopción ágil, mientras que es el liderazgo el que permite la transformación ágil a nivel de la organización. Por lo tanto, los líderes deben realizar las acciones correctas en el momento adecuado para facilitar la transformación.

En general, los movimientos ágiles exitosos pueden iniciarse como esfuerzos ascendentes y limitarse solo al desarrollo de software. Muchas consultoras de transformaciones ágiles han tomado medidas para deshacerse de la mayoría de los gerentes como gastos generales, para reemplazarlos por un mayor número de dueños de productos y Scrum Masters, frecuentemente como contratistas temporales. Esto ha provocado efectos negativos en las iniciativas ágiles, lo que resulta en fallas en la transformación. Sin saber qué hacer, los gerentes pueden incluso convertirse en obstáculos, en lugar de eliminar obstáculos.

Los Stakeholders deben trabajar en estrecha colaboración y continua comunicación. Los equipos y los clientes deben crear circuitos de retroalimentación donde los productos se puedan desarrollar e implementar rápido con la retroalimentación proporcionada directamente al equipo. Esto se crea con agilidad y ciclos de desarrollo más efectivos. Si una organización etiqueta ágilmente otro "proceso" y reemplaza la metodología de cascada con la nueva, no se obtiene ninguna ventaja. El liderazgo debe tener muy claro el objetivo de abrazar lo ágil.

Metodologías ágiles desde el liderazgo

Transformar una organización desde una estructura mecanicista, jerárquica, funcional y formalizada con un sistema burocrático hacia un sistema orgánico, plano y descentralizado es

un cambio estratégico importante, que necesita la aceptación de la gerencia y del directorio (Walenta, 2017; Doz, 2019).

No tener un objetivo claro para adoptar manifiestos ágiles es un problema. Muchos líderes no entienden el cambio de comportamiento requerido para operar en un entorno ágil.

En general, el impulso inicial y el deseo de agilidad provienen del mismo equipo que ejecuta las tareas: individuos que impulsan la transformación desde abajo. Pero esto por sí solo no es suficiente.

La verdadera transformación ágil es posible solo cuando el liderazgo la promueve para que sea exitosa. Se pueden categorizar las responsabilidades de liderazgo de la siguiente forma:

- Crear la composición adecuada para equipos y ubicaciones.
- Organizar los equipos y las estructuras del programa.
- Establecer las expectativas correctas para el seguimiento de las actividades de lanzamiento.
- Gestionar el backlog.
- Mejores prácticas continuas de gestión de equipos.
- Liderazgo ágil: roles y responsabilidades generales.
- Promover y fomentar la transformación organizacional.

Planear la transformación ágil

Crear la estructura de equipo correcta es esencial incluso cuando se construye desde cero. Ya sea en un startup o en una gran empresa (en especial, cuando se buscan recursos calificados en áreas altamente técnicas), el trabajo en equipo ocupa un segundo lugar. Se tiende a contratar personas que estén disponibles sin tener en cuenta la colocación y la distribución geográfica. Si se espera que trabajen como individuos y no como un equipo Scrum, su trabajo se puede gestionar en un tablero Kanban. Si necesitan trabajar juntas como un equipo ágil de Scrum, entonces, deben estar juntas. En los casos en que no es posible la colocación geográfica, al menos, se debe considerar a las personas en un equipo trabajando juntas en zonas horarias lo suficientemente cercanas, con salas de equipo altamente conectadas y acuerdos comunes de tiempo de trabajo.

Las grandes organizaciones tienen mucha flexibilidad para reorganizar a los miembros en un equipo, ya que pueden tener una fuerza crítica en cada ubicación. En general, esto requiere la reorganización de las personas en los niveles más altos de liderazgo, puesto que los gerentes individuales pueden no tener suficientes personas para realizar tales cambios de equipo.

Este énfasis en la ubicación física en un mismo lugar (colocación) ocurre a nivel equipo. En esta economía global, no es realista esperar que toda la organización, o incluso un solo programa, se localice por completo en un mismo lugar geográfico. Esto significa que las comunicaciones y las reuniones entre equipos deben gestionarse con cuidado, en momentos adecuados para todos.

En realidad, es necesario analizar los talentos adecuados de las personas que se mapean a las tecnologías necesarias. Es indispensable identificar líderes clave para cada equipo y agregar a otras personas como jugadoras en el equipo. Tener una mezcla de miembros senior y junior hace un mejor equipo. En los equipos donde se necesitan contratistas temporales, se precisa un esfuerzo concertado para tener la fuerza crítica de los miembros más permanentes del equipo.

Capacidades de los equipos

Para que los equipos sean ágiles, no solo deben cumplir con las ceremonias y metodología, sino que también debe considerarse cómo se encuentran formados o cómo se enfoca el desarrollo de las habilidades de sus miembros. Si el equipo dispone de muchos especialistas, cada uno focalizado en su especialidad, el trabajo puede acumularse y afectar su efectividad. Por lo tanto, los equipos ágiles deben formarse con miembros expertos en forma de “T”. Si bien algunos pueden ser especialistas en ciertas áreas, también deben tener habilidades generales y estar dispuestos a trabajar en otras áreas. Es menester que los miembros del equipo puedan trabajar en diferentes áreas e involucrarse en temas fuera de su especialidad para colaborar con el equipo cuando sea necesario.

Se denomina en forma de “T”, dado que se espera que el individuo tenga conocimientos generales y básicos de las áreas en las que está involucrado el equipo (la parte de arriba de la “T”) y desarrolle una especialidad en la cual sea el experto y que conozca en profundidad (la parte el tronco de la letra “T”).

El empoderamiento: la clave de los equipos ágiles

En el trabajo en equipo tradicional, el gerente de proyecto controla todo: tiene la responsabilidad y la autoridad sobre todas las decisiones y planes.

En un entorno ágil, todos poseen autoridad y responsabilidad. El liderazgo y la gestión son muy diferentes. El liderazgo consiste en detectar un problema y luego reunir a las personas para resolverlo. Cualquier miembro del equipo que sepa lo suficiente como para prever un problema sabe lo suficiente como para liderar la solución. Es su responsabilidad hacerlo.

Aunque parezca lo contrario, un equipo que confía en su trabajo sin una sola persona como jefe es muy decisivo. Al no darle cierto control al equipo, los gerentes corren el riesgo de comprometer toda la transformación ágil y evitar que su equipo tenga un alto rendimiento.

El rol de desarrollador/miembro del equipo debe cubrir habilidades de programación, arquitectura, administración de bases de datos y más. Sus responsabilidades incluyen decidir cómo organizarse y qué hacer a continuación. Aceptan liderar y buscar soluciones potenciales, mientras se mitigan los riesgos. Esto es totalmente diferente de lo que se espera del rol de desarrollador en la metodología de trabajo tradicional (limitado a escribir líneas de código).

El Scrum Master tiene muchas responsabilidades nuevas. El mayor cambio es ver al equipo interactuar y estar allí en el medio. Su responsabilidad implica que, si las relaciones se rompen, debe facilitar la conversación para repararlas. Por otro lado, el cliente no brinda una visión consistente del producto, el Scrum Master debe ayudar a clarificar esta visión para prevenir que el equipo trabaje en tareas que no tengan sentido ni aporten valor.

Todos necesitan acostumbrarse al liderazgo distribuido. Las personas más involucradas con una decisión la asumen cuando la enfrentan. Si saben lo suficiente para ver que se debe tomar una decisión y creen que conocen la mayoría de las consecuencias, saben lo suficiente para tomarla.

Hay que tomar decisiones, no siempre son las correctas. Las decisiones de quien llega primero pueden afectar la armonía.

El origen del empoderamiento en equipos ágiles

La metodología ágil nació debido a la necesidad de cambiar las formas ineficientes de desarrollar productos de software. Además del proceso de trabajo en sí, el tratamiento de las personas como "activos" o "recursos" también tuvo que cambiar. Para formar equipos reales

dentro de los procesos ágiles, las colecciones de contribuyentes individuales requieren una atención seria.

En consecuencia, la idea de empoderamiento ha sido parte de la metodología desde su inicio. Los 17 creadores del Manifiesto Ágil (2001) entendieron que los individuos, en realidad, no necesitaban microgestión para ser productivos. De hecho, hicieron su mejor trabajo cuando se les permitió más libertades dentro del proceso de trabajo.

Esta buena práctica se convirtió en parte esencial del Manifiesto Ágil (2001). El quinto principio recomienda lo siguiente: “Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo” (Manifiesto por el Desarrollo Ágil de Software, 2001, párr. 5).

¿Cómo empoderar equipos?

Un factor que puede acelerar el empoderamiento del equipo es eliminar la culpa. Algunos miembros del equipo dudan en tomar decisiones importantes si sienten que alguien los puede intimidar más adelante. En algunas compañías, llegar tarde si hay alguien a quien culpar es tan bueno como llegar a tiempo. Eliminar la culpa como aceptable y dejar claro que nadie quiere escucharla es clave para el proceso.

Uno de los mayores malentendidos sobre el empoderamiento del equipo es que, simplemente, los equipos se empoderan cuando se anuncia que la metodología ágil ha llegado. El equipo comienza a seguir un conjunto de reglas diseñadas para establecer expectativas entre sus miembros. A la mayoría de la gente, no le gustan las reglas porque, en general, limitan. Un equipo ágil establece reglas para ser libre. Crea reglas simples enfocadas en definir las interfaces entre los miembros del equipo.

Con un único punto de control, necesita una sola persona completamente informada. Los miembros del equipo pueden pasar meses sin hablar entre ellos, siempre y cuando una persona lo sepa todo y lo controle todo. Desafortunadamente, incluso en proyectos más pequeños, este modelo de comando y control se descompone rápido.

El empoderamiento del equipo es fundamental. Permitir que los equipos identifiquen cómo mejorar, continuamente "inspeccionar y adaptar" permite que los equipos realmente empoderados tomen mejores decisiones que cualquier individuo.

La sabiduría colectiva brinda una serie de beneficios:

- Mayor aceptación y compromiso con los acuerdos cuando surgen de la toma de decisiones participativa.
- Decisiones más sólidas cuando se reúne información de todo el equipo.
- Recompensas a los miembros del equipo por trabajar juntos, dándoles el poder de asumir sus decisiones.
- Al involucrar a los miembros del equipo en las propias decisiones, son motivados para ejecutarlas con confianza y estándares altos.

Para empoderar a los equipos durante la transformación, se requiere lo siguiente:

1. Permitir que los equipos tomen la iniciativa ágil, en lugar de empujarla. Las adopciones ágiles impuestas a los miembros del equipo bloquean la aceptación de la iniciativa desde el principio.
2. La planificación realizada por los equipos ágiles no admite el “acoso” de los clientes para que el equipo asuma más trabajo del que puede absorber de manera segura. Los Stakeholders deben comprender el valor de los equipos empoderados.
3. Las transformaciones verdaderamente ágiles adoptan la disciplina de la planificación de todo el equipo.
4. Fomentar el empoderamiento del equipo a través de sus propios comportamientos.
5. Hacer responsables a los equipos. Los equipos empoderados se hacen cargo de sus compromisos.
6. Empoderar a los equipos significa eliminar la noción de un "héroe" como el medio para el éxito organizacional. La evaluación del desempeño individual crea adversarios accidentales entre los miembros del equipo. El grupo de trabajo no puede ser un verdadero equipo cuando los héroes son compensados por actuar como individuos.
7. Un contrato de equipo ágil es un conjunto de reglas generadas por el equipo para el equipo. Una receta común que el equipo acuerda y sigue para entregar con éxito valor a la organización. Al establecer un estatuto de equipo, pueden definir qué significa empoderamiento en la práctica. Estas reglas no son "dejadas" en el equipo por un gerente o líder del equipo.

8. Alineación en torno a objetivos comunes. La comunicación efectiva es vital para cualquier organización, y aún más cuando queremos que las personas tengan poder de decisión. Si los líderes no están seguros de que el equipo vea el panorama general, no pueden esperar razonablemente que tomen las decisiones correctas en el momento adecuado. Los equipos de ejecución, asimismo, deben comprender cómo sus acciones contribuyen al éxito estratégico de la organización.

Equipos autoorganizados

Boehm y Turner (2005) afirman que un proceso verdaderamente ágil también debe ser autoorganizado y emergente. “Autoorganizado” significa que los equipos toman decisiones a través de la comunicación informal y de reuniones breves y frecuentes, en lugar de confiar en un responsable para guiar el proyecto. En metodologías ágiles, “emergente” significa que los requisitos surgen durante el curso del proyecto, casi sin dedicar tiempo al diseño antes de que comience la codificación (Barlow, 2011).

Un equipo autoorganizado es un equipo empoderado. Cuando el gerente se baja del pedestal tradicional de jefe, la dinámica mejora.

La función del Scrum Master es la de facilitador y asesor del equipo, no la de quien toma las decisiones. Esto les permite a los miembros del equipo evolucionar en propietarios de procesos, que participan activamente en el proceso de planificación y en las reuniones retrospectivas.

Además, con el tiempo, comienzan a involucrar de manera proactiva a colegas de otros departamentos para maximizar el valor que producen y mejorar la eficiencia de su trabajo.

Propiedad del proceso

Los equipos, incluso si tienen poca influencia sobre qué trabajo necesitan hacer, deben poder controlar cómo lo hacen. La propiedad del cómo permite que los contribuyentes individuales se sientan en control del proceso de trabajo. Además, les permite enorgullecerse de su contribución en el logro de los objetivos de la empresa.

Cuando los miembros del equipo tienen la oportunidad de influir en cómo procesan su trabajo, pueden lograr más de lo imaginado. Por ejemplo, si los individuos deben construir un proceso eficiente para sus equipos, se sentirán motivados a perfeccionarlos eliminando los obstáculos que impiden ofrecer valor. No tendrán miedo de experimentar y buscar mejores

formas de efectuar el trabajo para aumentar el rendimiento del equipo. Esto se debe a que ellos son parte de la construcción de la solución.

Métricas en metodologías ágiles

Es difícil discutir el viejo dicho "No puedes mejorar lo que no puedes medir". Las métricas son esenciales para el seguimiento de las empresas y de los equipos. Son herramientas útiles para planificar, adaptar y comprender cómo progresan los proyectos, y cuán bien está trabajando un equipo. Las métricas permiten saber si es necesario hacer cambios o, simplemente, continuar con el buen trabajo.

Los indicadores clave de rendimiento (KPI: Key Performance Indicators, en inglés) son métricas que se utilizan para evaluar el éxito de la empresa, para alcanzar los objetivos comerciales. Ayudan a comprender si la empresa está en el camino correcto para el éxito, también arrojan luz sobre el progreso del proyecto y sobre las contribuciones individuales a dicho proyecto. El objetivo de los KPI es lograr mejoras e identificar áreas en las que centrar la atención. Los KPI se rastrean a lo largo del tiempo para observar tendencias y comparar con el plan del rendimiento esperado de la empresa.

En una metodología ágil como Scrum, las métricas son necesarias para el desarrollo exitoso del producto. Un equipo Scrum opera en Sprints, períodos de tiempo durante los cuales se deben completar tareas específicas. Los Sprints le dan al equipo la oportunidad de establecer una meta alcanzable y construir el producto correcto en el momento correcto.

Entonces, es útil contar con métricas que midan si el equipo ha mejorado o mantiene una eficiencia satisfactoria en su trabajo. Es bueno evaluar cada proyecto y aislar los problemas para mejorar la calidad del trabajo del equipo. Existen KPI de Scrum efectivos para mantener al equipo de software y al cliente en la misma página. Proporcionan todos los datos necesarios para controlar cada etapa del proyecto y, en consecuencia, permiten lanzar un producto de alta calidad sin riesgos. La implementación de métricas correctas es una excelente manera de evitar confusiones, plazos vencidos y fallas sistemáticas en el código.

Sin embargo, las buenas métricas no deben limitarse a gráficos e informes. Brindan un conocimiento cuantitativo del desempeño del equipo y proporcionan objetivos medibles para el equipo, pero igualmente importante es una buena comunicación entre los compañeros de trabajo y escuchar los comentarios del equipo y de los clientes. La combinación de

retroalimentación cuantitativa y cualitativa puede brindar efectos duraderos y mejorar el proceso de gestión del proyecto.

¿Qué son las métricas ágiles?

Las métricas ágiles ayudan a los equipos a medir el proceso de desarrollo, la productividad, la calidad del trabajo, la previsibilidad, la salud del equipo y los productos que se están desarrollando. Un enfoque clave de las métricas ágiles es el valor entregado a los clientes: en lugar de medir qué o cuánto se está haciendo, se mide cómo impacta en un cliente.

Cómo lograr que una métrica sea relevante en un entorno ágil

Los entornos ágiles requieren métricas que los equipos entiendan bien y puedan ayudar a aprender y mejorar los procesos.

Estas son algunas cualidades inspiradas en el trabajo de Tranter (2018) que hacen que una métrica sea poderosa, es decir, que ayudan a impulsar una mejora positiva en un equipo ágil:

- Es utilizada por el equipo: La gerencia no debe imponer o medir métricas ágiles, sino que deben ser utilizadas voluntariamente por equipos ágiles para aprender y mejorar.
- Está rodeada de conversación: Las métricas no solo deben ser números, sino el punto de partida de una conversación sobre procesos y obstáculos que afectan al equipo.
- Es parte de un experimento específico: Las métricas deben usarse para responder una pregunta específica sobre procesos ágiles, no solo por el bien de la medición.
- Se utiliza en un contexto junto con otras métricas: Una gran métrica, si se usa sola, puede conducir e incentivar a los equipos a maximizar esa métrica en particular, a expensas de todo lo demás. El uso integrado de varias métricas proporciona una imagen equilibrada de la actividad ágil.
- Es fácil de calcular y comprender: Las métricas demasiado complejas o que no se entienden completamente, incluso si proporcionan una buena información sobre el trabajo de un equipo, no son útiles para guiar las actividades cotidianas.

Las principales métricas ágiles

Rendimiento o quemado del Sprint (burndown)

La tabla de rendimiento del Sprint (Figura 1) visualiza cuántos puntos de la historia (medida de esfuerzo) se han completado y cuántos quedan, y ayuda a pronosticar si se completará a tiempo.

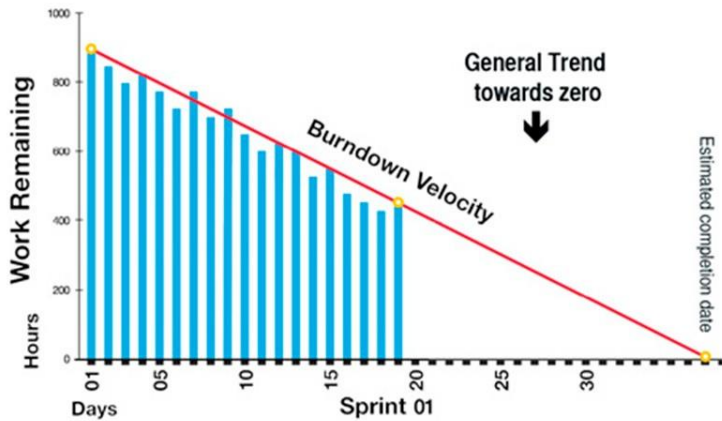


Figura 1. Diagrama de quemado (Linnanvuo, 2015)

Respecto del valor proporcionado, deja claro al instante cuánto valor ha brindado un Sprint y qué tan cerca se está de completar el compromiso con los clientes.

Velocidad ágil

En términos de agilidad, la velocidad mide cuántos puntos de historia (esfuerzo) completó un equipo, en promedio, en los últimos Sprints (Figura 2). Se puede usar para predecir la salida del equipo en las próximas iteraciones.

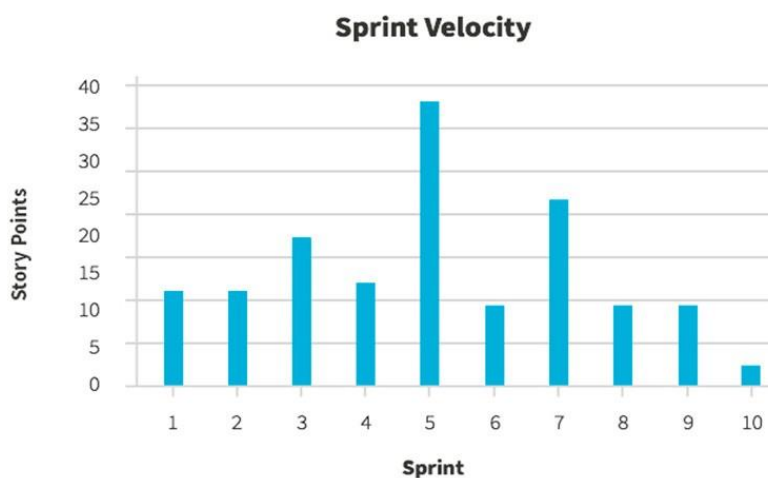


Figura 2. Gráfico de velocidad por iteración (Linnanvuo, 2015)

Respecto del valor proporcionado, la velocidad es importante porque es una métrica de resultados (cuánto valor se entregó realmente a los clientes en una serie de Sprints). Tener cuidado de no comparar la velocidad entre equipos porque los puntos de la historia y la definición de hecho pueden variar entre los equipos.

Plazo de ejecución

El tiempo de entrega mide el tiempo total desde el momento en que una historia ingresa al sistema (en el backlog) hasta que se completa como parte de un Sprint o se lanza a los clientes (Figura 3). Mide el tiempo total para que se cumpla un requisito y comience a ganar valor: la velocidad de su cadena de valor.

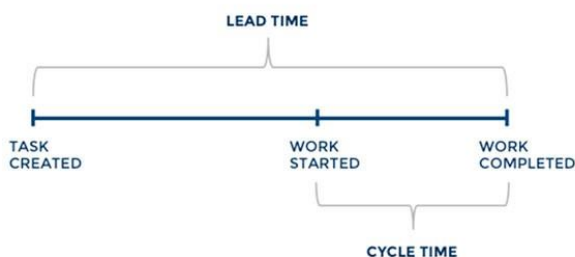


Figura 3. Tiempo de ejecución: desde principio a fin (Linnanvu, 2015)

Respecto del valor proporcionado, en cierto sentido, el tiempo de entrega es más importante que la velocidad porque mide todo el sistema ágil de extremo a extremo. Reducir el tiempo de entrega significa volver más eficiente el proceso de desarrollo.

Tiempo del ciclo

El tiempo del ciclo es un subconjunto del tiempo de entrega: mide el tiempo en que una tarea pasa de iniciada, o en progreso, a terminada. Normalmente, los tiempos de ciclo deben tener alrededor de la mitad de la longitud del Sprint. Si los tiempos de ciclo son más largos que un Sprint, los equipos no están completando el trabajo en el tiempo prometido.

Respecto del valor proporcionado, una métrica muy simple puede levantar una bandera roja cuando los elementos dentro de los Sprints en todo el sistema no avanzan.

Comunicación

La comunicación clara es uno de los aspectos más importantes del Scrum. Las conversaciones cara a cara son el corazón y el alma de los proyectos ágiles. Permiten que los clientes y los equipos de desarrollo trabajen más estrechamente y estén en la misma página.

Pero las métricas o los KPI no son suficientes cuando se trata de mejorar la eficiencia: debe haber apertura entre los miembros del equipo y sus clientes. Solo la comunicación abierta y frecuente brinda un mejor rendimiento del equipo y, por lo tanto, un mejor producto final.

Respecto del valor proporcionado, es crucial para entregar casi cualquier proyecto con éxito. La buena comunicación mantiene informados a los interesados del proyecto, y los equipos Scrum se centran en completar el trabajo comprometido para el Sprint.

Mejora retrospectiva de procesos

La retrospectiva es una parte esencial del Scrum y está dedicada a la mejora del proceso de software. Una reunión retrospectiva le da al equipo Scrum la oportunidad de discutir Sprints anteriores y decidir cómo aumentar la productividad en el próximo Sprint. Suele tener lugar al final del Sprint o de cualquier otro ciclo de desarrollo.

El objetivo de la retrospectiva es definir los cambios que deben implementarse para que el proceso sea más efectivo en el futuro.

Respecto del valor proporcionado, da una buena idea de los problemas encontrados por el equipo y ayuda a establecer un plan de acción de mejora.

Comprender los objetivos

Comprender los objetivos es una medida subjetiva de cuán bien el cliente, el equipo de producto y el equipo de desarrollo entienden las historias y el objetivo del Sprint. Cada miembro del equipo debe comprender el propósito de los Sprints y centrarse en lo que se debe lograr.

Respecto del valor proporcionado, tener al equipo alineado resulta fundamental para completar con éxito un proyecto.

Los involucrados en transformaciones ágiles pueden elegir una amplia gama de métricas para diferentes tipos de progreso. Ayudan a medir el trabajo realizado por los equipos y el valor entregado a los clientes, así como a introducir más claridad en el proceso de desarrollo. Gracias a las métricas, es más fácil cuantificar la satisfacción de las personas e identificar áreas de mejora con los clientes y con el propio equipo.

METODOLOGÍAS ÁGILES EN EQUIPOS DE OPERACIONES DEL ÁREA DE TECNOLOGÍA DE LA INFORMACIÓN EN UNA EMPRESA MULTINACIONAL DE LA INDUSTRIA ENERGÉTICA

La organización

El presente trabajo está enfocado en el área de sistemas de una empresa multinacional del sector energético. La compañía, se estableció en Argentina, en 2004; y desde entonces fue sumando posiciones y áreas, tales como contabilidad, impuestos, compras, recursos humanos y sistemas. Así, desde Buenos Aires, los casi 2000 empleados actuales brindan soporte a todas las unidades de negocio a nivel global.

La organización, líder en la industria dispone de sedes y operaciones en los cinco continentes, en más de 40 países. Su objetivo es mantenerse como la empresa número uno de la industria, alcanzar resultados operativos superiores respetando los más altos niveles de conducta empresarial.

A lo largo de su historia, ha sabido mantenerse con resultados financieros y operativos inigualables.

Cultura y valores

Desde sus comienzos, la corporación tuvo una cultura muy conservadora y una organización tradicional. Evitaba riesgos de cualquier tipo. Fiel a sus orígenes y a la cultura propia de la industria, su valor se basaba en la seguridad.

Un ejemplo claro de esta aversión al riesgo: diez años atrás, se trabajaba, por política de la compañía, con versiones de software anteriores a la última versión disponible en el mercado.

Sin embargo, en los últimos seis años, se experimentó una serie de cambios graduales que, en el último año, terminó de dar un vuelco vertiginoso. Los cambios comenzaron desde la estrategia de negocios. Se empezó a evidenciar mayor riesgo en los siguientes aspectos:

- Implementación de las últimas tecnologías.
- Fomento de la innovación.
- Tercerización del trabajo operativo.

- Cambio en la política de reclutamiento: El foco pasó de contratar a los mejores del mercado a reclutar candidatos junior con ambición y perspectivas de crecimiento a futuro. En la actualidad, interesa más en el potencial del candidato que su experiencia y sus credenciales.

La transformación ágil en la compañía

En los últimos seis años, se comenzó a escuchar cada vez más sobre la implementación de metodologías ágiles. Al principio, como programas piloto; después, como “capricho” de la alta dirección o para subirse a la ola de lo nuevo.

Luego de varios anuncios y gran promoción de estas metodologías y capacitación, varios equipos se animaron a adoptarlas, independientemente de la política o imposición desde arriba. Muchas de estas implementaciones terminaron en equipos tradicionales con terminología o título cool, o en ceremonias donde parecer lo que uno no es. En estos últimos casos, lamentablemente, el cambio provenía desde abajo. Los equipos cumplían con los requisitos de las metodologías ágiles, pero, a la hora de ser independientes, tenían a la alta dirección o a los supervisores tomando decisiones y guiando al equipo en cómo hacer, además de qué hacer.

Finalmente, para evitar tantos equipos en esta situación, se abrieron posiciones de Agile Coach. Este nuevo rol fue (y es) clave para que el cambio se consolide realmente. Ellos se encargaron de mostrarle a la alta dirección que los equipos deben tener cierta independencia y sentirse empoderados para actuar y tomar decisiones. Un gran cambio de paradigma con respecto a las expectativas de trabajo tradicional, donde se espera que el empleado acate órdenes y cumpla su labor sin cuestionamientos. Casi de repente, comenzaron a reconocer a quienes cuestionaban los procesos y a motivarlos a desafiar los procedimientos vigentes, promoviendo la innovación y el pensamiento lateral.

En sintonía con la implementación de metodologías como Scrum o Kanban, la asignación de posiciones comenzó a efectuarse de manera tal que, cuando era posible, el equipo se encontraba localizado físicamente en el mismo espacio físico (colocación) para aprovechar los beneficios de estar reunidos cara a cara, agilizar la toma de decisiones y mejorar la comunicación y la productividad dentro de los equipos.

Gestión de capacidades

Con la adopción de metodologías ágiles, la política de gestión de capacidades se está modificando, ya que surge el concepto de DevOps, lo cual implica que los equipos se

focalizan en entregar soluciones de principio a fin, en vez de desarrollar solo una porción. Esto implica que, a cada miembro del equipo, se le pueden presentar desafíos diferentes en el transcurso del tiempo (no siempre el mismo) a fin de fomentar el equipo T-shape y desalentar la especialización. De este modo, las capacidades se desarrollan más por demanda y necesidad que por inercia y lógica, como en la organización tradicional (en la cual el trabajo de un equipo se restringe a una tecnología o a una herramienta puntual).

Evaluación de desempeño en pleno cambio

Desde hace más de veinte años, la compañía adopta la evaluación de desempeño de personal para luego, de acuerdo con los aportes de cada empleado, ubicarlos en una tabla de rango percentil. Este es uno de los factores determinantes del aumento salarial anual.

Cabe destacar que la política de la compañía no estipula ningún tipo de bono anual de acuerdo con objetivos cumplidos/ganancias de la compañía.

La evaluación de desempeño consiste en acordar con el superior directo lo siguiente:

- Logros durante el período.
- Capacidades desarrolladas durante el período.
- Objetivos para el siguiente ciclo.
- Oportunidades para el siguiente ciclo.

Más tarde, un resumen de los puntos acordados es expuesto y debatido intensamente entre gerentes para llegar al ranking de empleados por área, que luego se mezcla con el de otras áreas, para llegar a uno global.

Durante años, el debate continuo se daba en la inconsistencia de promover el trabajo en equipo y la competencia individual. El ranking promueve el desarrollo individual, mientras que los directivos fomentan la colaboración y el trabajo en equipo.

En otras palabras, el management felicitaba al equipo por los logros, pero premiaba a algunos individuos por los logros y el progreso. De ese modo, una persona que mostraba lo que hacía el equipo como propio o recibía ayuda del equipo era mejor rankeada que alguien que activamente ayudaba, colaboraba y empujaba al equipo para que estuviera más integrado y para que el trabajo entregado fuera un producto conjunto.

Estado actual de la transformación cultural

Si bien la implementación de metodologías ágiles a nivel corporativo aún está en plena transición y algunos equipos no son lo suficientemente maduros como para llamarse “ágiles”, ya empiezan a percibirse cambios actitudinales de los empleados respecto del trabajo. Por ejemplo, innovación, iniciativas de mejora, entusiasmo y mayor trabajo en equipo (satisfacción del empleado).

Esta nueva cultura totalmente alineada con las metodologías ágiles cambió de manera drástica la forma de trabajo dentro de la organización de la TI. Inicialmente, este cambio fue percibido como desorganizado (caos). La resistencia al cambio desde la objeción psicológica se daba, principalmente, por el temor a lo desconocido; las objeciones lógicas se evidenciaban en cuestionamientos de la factibilidad técnica del cambio y de los costos de este expresadas por los empleados.

Sin embargo, dados los esfuerzos de comunicación y transparencia a lo largo del proceso, los estadios de finalización y zona neutral fueron más breves de lo esperado. Los empleados hoy son más conscientes del valor agregado que le generan a la compañía y del impacto de su trabajo. El incentivo va más allá del salario (parte de la nueva política de compensaciones). Se sienten más independientes y parte de la toma de decisiones, responsables y dueños (accountability) de los resultados, debido a la libertad de emprender y a la confianza que deposita la compañía en ellos (una mayor satisfacción de los empleados reduce el índice de desertión). Los equipos se dieron cuenta de que son parte del cambio y de que pueden autogestionarse como deseen mientras sean productivos.

“La cultura se come a la estrategia en el desayuno” (Drucker, 2000), sin duda. Siempre hablamos de estrategia de negocio, estrategia tecnológica, estrategia de capacitación de personas..., pero todas estas estrategias de transformación serán un fracaso si no hacemos una óptima gestión del cambio cultural. La implementación de nuevas metodologías debe basarse en una estrategia de cambio, en la que la cultura sea una pieza fundamental. Es preciso comunicar claramente los beneficios esperados y mostrar el propósito de los cambios para lograr una adopción real. Este cambio no es posible sin la conciencia y la alineación de todas las áreas y los procesos de la compañía. El rol de Recursos Humanos es fundamental para asegurar que estos cambios sean orquestados correctamente y la implementación no quede a mitad de camino.

A fin de evitar la resistencia a los cambios gestados y asegurar la correcta implementación de la nueva cultura, se necesita alineamiento continuo y adaptación de los procesos desde el

punto de vista funcional, estratégico y de Recursos Humanos para que esta sea profunda y definitiva.

De este modo, el gran desafío consiste en identificar rápidamente los procesos deficientes e incoherentes con la nueva realidad y efectuar los cambios necesarios para brindarles seguridad a los empleados.

Implementación inicial y primeros pasos

El equipo se conformó en el transcurso de seis meses, en los cuales fueron incorporándose integrantes reclutados por el Departamento de Recursos Humanos. El criterio de selección estuvo centrado en las capacidades tecnológicas y analíticas para resolver los desafíos técnicos que surjan como parte de las responsabilidades del equipo. De este modo, se formó un equipo de ocho personas, de las cuales únicamente el Scrum Master y el dueño de producto tenían conocimiento y experiencia en las metodologías ágiles.

Inmersión en la empresa y equipo

A medida que cada integrante ingresaba, se lo incluía en el proceso normal de inmersión de la compañía, en el cual aprenden terminologías, historia de la compañía, políticas, herramientas utilizadas, valores culturales, etc. Adicionalmente, los nuevos miembros tomaron cursos cortos online sobre Scrum y metodologías ágiles para obtener conocimientos básicos y luego poder asistir mejor preparados a cursos presenciales de la metodología Scrum. En los cursos presenciales, se llevó a cabo una simulación de la metodología de principio a fin, donde experimentaron cada una de las ceremonias, asumieron diferentes roles y entendieron las motivaciones de cada uno de los involucrados en el proceso.

En paralelo a este proceso, se los invitó a participar pasivamente de las ceremonias del equipo cuando fuera posible para empezar a familiarizarse con la dinámica.

Desafíos y malos hábitos

Como establece la misma metodología, el plan consistía en que el equipo comenzara a funcionar y poder efectuar ajustes y cambios, iteración a iteración, para mejorar y ser cada vez más ágiles y efectivos. Sin embargo, dados los antecedentes de los miembros del equipo, pudimos observar patrones de comportamiento heredados de sus experiencias anteriores en la forma de trabajo tradicional. A continuación, se detallan los más significativos.

Trabajo no planeado aceptado: pedidos espontáneos

Con el paso del tiempo, el equipo fue aprendiendo las tareas técnicas, la dinámica de las ceremonias y comenzó a tomar trabajo del backlog. Sin embargo, se experimentaron varios casos de trabajo fuera de la lista de tareas y compromisos del equipo.

El síntoma evidente fue que no llegaban a cumplir con las tareas y los objetivos planteados para la iteración. De este modo, se arrastraba un gran número de historias de usuario de un ciclo al siguiente. Luego de varias reuniones al respecto, se efectuó un análisis de los tiempos estimados para cada tarea con el fin de verificar que las estimaciones durante la planificación no fueran erradas.

Asimismo, comprometidos a encontrar una solución de inmediato, se acordó permitir que el Scrum Master hiciera un seguimiento más de cerca de las tareas de cada uno y de los avances en las reuniones diarias. La idea no era controlar al equipo, sino identificar el problema de raíz que generaba demoras al entregar valor.

Finalmente, se pudo comprobar que el problema se encontraba en pequeñas tareas. Los clientes, acostumbrados a las formas tradicionales de trabajo, contactaban a los miembros del equipo directamente y les efectuaban pedidos. En muchos casos, porque ellos mismos habían resuelto un problema similar con el mismo cliente; en otros, simplemente, porque buscaron a una persona disponible del equipo para resolver su problema de manera inmediata sin advertir que la metodología ágil establece un proceso diferente.

El equipo interpretaba que cada cliente que los contactaba debía ser satisfecho sin importar la prioridad acordada. Si el cliente lo contactaba, ya era condición suficiente para cumplir con su pedido y expectativas.

De este modo, el equipo trabajaba para satisfacer al cliente; no se basaba en las prioridades planificadas y comprometidas, sino en lo que iba surgiendo. Cabe destacar que este tipo de trabajo no aparecía en la lista de tareas, ya que, muchas veces, era insignificante o, al ser no planificado, los miembros del equipo no lo ingresaban en el backlog. Los clientes “espontáneos” estaban satisfechos, pero los que utilizaban los canales y procedimientos establecidos por la metodología se veían afectados por demoras en la entrega del trabajo. Cabe destacar que la metodología establece que los pedidos fuera del plan deben ser consensuados, priorizados y planificados, salvo excepciones o casos particulares previamente acordados.

Falla de planificación y trabajo escondido

Otra de las situaciones que generaban demoras y acarreo de compromisos de un Sprint a otro era la falla en la planificación. Los miembros del equipo planificaban el trabajo en historias de usuarios, pero luego, sobre la marcha, identificaban otras tareas necesarias para terminar el trabajo. En estos casos, quien lo necesitaba se remitía al backlog sin mencionar nada al resto a sus compañeros y creaba nuevas historias de usuarios dentro del mismo Sprint para trabajar inmediatamente. El dueño del producto no visibilizaba esta tarea extra hasta que era demasiado tarde. Como resultado de esto, el backlog excedía la capacidad del equipo para el ciclo, y no se llegaba a terminar el trabajo comprometido.

Objetivos no claros

Como define Scrum, el equipo toma los requerimientos y va estimando y planificando la ejecución del trabajo. En las reuniones de refinamiento, el especialista en cada tecnología tomaba las historias que más fáciles parecían y con las que más cómodos se sentían. De este modo, cada uno trabajaba en su área, sin compartir tareas ni conocimiento. Peor aún, empezaban a trabajar en todas las características en simultáneo, con lo cual no se podía terminar una, entregar valor y luego atacar la siguiente; sino que todas avanzaban al mismo ritmo con perspectiva de entregarlas al final del período.

Valor versus entregas

Cabe destacar en esta instancia que el objetivo es maximizar el valor entregado al cliente y entregarlo lo antes posible, en vez de maximizar entregas.

Moreira lo define como:

Una salida es la entrega de un producto. Un resultado es cuántos clientes compraron o usaron el producto. A menudo, las personas se centran en las salidas porque tienden a ser más fáciles de medir o son un remanente de una mentalidad más tradicional (Moreira, 2017, pp 21-27.).

Entender la diferencia entre una salida/ producto y un resultado es primordial para poder cambiar de mentalidad tradicional a mentalidad ágil.

En otras palabras, la salida es el "producto de un proceso", mientras que el resultado se define como un "resultado" o un "efecto visible" (Casanova, 2013). Scrum se basa en que el equipo se centre en el resultado y no en el producto.

Evidencia de la situación

Un ejemplo claro de la falta de entendimiento de esta diferencia conceptual y cambio de mentalidad fue al tratar el tema de los tickets u órdenes de trabajo en nuestra herramienta interna, en la cual los clientes reportan problemas o piden soluciones sobre la infraestructura actual.

La cola crecía, y los tickets se iban acumulando. En consecuencia, el dueño de producto planificó más dedicación y prioridad para este tipo de tareas. El equipo, intentando aclarar las expectativas, le consultó al dueño de producto cuántos el equipo debía cerrar por semana. Esta pregunta, que parece inocente, deja entrever que el equipo no estaba entendiendo el objetivo ni la diferencia entre entrega y resultado, y no se estaba manejando en forma empoderada.

El equipo tenía la libertad de efectuar un análisis y definir con cuántos cerrar para evitar la acumulación y asegurar la satisfacción de los clientes. En este caso, no es crítica la cantidad, sino la calidad: podían quedar tickets esperando sin cerrar, mientras los urgentes fueran atendidos, y los clientes no se quejaban y pudieran continuar su operación. La respuesta estaba en manos del equipo; llevó tiempo percatarse de ello y cambiar la forma de ver el problema. Ya no era el dueño de producto quien debía brindar un número, sino que ellos debían autoorganizarse para proponer una solución que maximizara el valor entregado al cliente. Esta situación desencadenó conversaciones sobre accountability y responsabilidades dentro de la metodología ágil para reforzar el cambio de paradigma necesario dentro del equipo. En definitiva, el equipo no debía esperar que un superior le enviara trabajo y le dijera qué hacer, sino que su papel fuera más protagónico al ser quien analizara el problema (y, a veces, lo anticipara) y propusiera la solución. Dada esta situación, el Scrum Master planteó un ejercicio interesante: proponer diferentes situaciones cotidianas del trabajo y discutir en equipo roles y responsabilidad usando una matriz (Anexo: Matriz de responsabilidades).

Supuestos no validados

Por otro lado, el equipo se remitía al backlog y asumía ciertas condiciones. Por ejemplo, que la información faltante para una implementación les sería provista cuando fuera necesaria. Para citar casos concretos, cuando clientes que necesitaban servidores específicos para sus soluciones eran contactados, dudaban o no estaban seguros de las especificaciones, y debían dirigirse a un tercero (en muchos casos, al proveedor de la aplicación que utilizaban) para confirmar la respuesta. Esto generó demoras en las entregas, ya que se asumían

especificaciones erróneamente o se buscaba la respuesta de los clientes sobre qué necesitaban realmente.

Experimentos y ejercicios

Como se ha descrito anteriormente, el equipo presentaba deficiencias, y era inminente tomar acciones para entender mejor la dinámica de la metodología. Si bien la expectativa era clara en la teoría, se notaba resistencia al cambio en la práctica o errores conceptuales en la aplicación de la metodología. Por ende, se coordinaron actividades para concientizar y discutir las diferentes situaciones, y que, poco a poco, cada miembro del equipo llegara a su momento “Ha” de “Shu-Ha-Ri” (en que entienden el circuito completo de la metodología).

Romper el hielo

La primera propuesta apuntó a generar mayor confianza dentro del equipo. Para ello, era indispensable que se conocieran los miembros más allá del trabajo.

El Scrum Master propuso dos ejercicios en diferentes momentos. El primero se puede llamar “Dos verdades y una mentira”. La actividad consistió en que cada miembro del equipo por turnos contara dos verdades sobre sí mismo y una mentira. El resto del equipo debía adivinar cuál era verdad y cuál mentira.

El segundo ejercicio, llamado “Constelación” (documentado en anagilemind.net), se llevó a cabo en un salón libre de mesas y sillas. El objetivo es descubrir de una forma diferente niveles similares de acuerdo y desacuerdo para que el equipo sea consciente de similitudes y diferencias. La actividad consiste en posicionar al equipo en ronda y colocar un objeto como “centro del universo” (en nuestro caso, un borrador de pizarrón). Luego el facilitador efectúa una declaración, y los miembros del equipo deben moverse según su nivel de acuerdo con esta. Si está totalmente de acuerdo, debe acercarse; si no está de acuerdo en absoluto, debe alejarse lo máximo posible del objeto. Ejemplos de las declaraciones son “entregamos productos de calidad en tiempo y forma” y “nuestros clientes están satisfechos con nuestro trabajo”. Al finalizar varias iteraciones, se pudieron discutir diferentes puntos de vista, y, en algunos casos, se acordaron cambios para mejorar.

Maximizar la entrega de valor: el juego de la moneda

Como se mencionó anteriormente, no era claro para el equipo el concepto de maximización de resultados. Por eso, se propuso un ejercicio para visualizarlo. Se dispuso al equipo

alrededor de una mesa con 10 monedas con el objetivo de que cada uno volteara las monedas y luego se las pasara al siguiente. Se efectuaron 3 iteraciones, y el facilitador tomó el tiempo en cada una.

En la primera iteración del ejercicio, cada jugador giraba todas las monedas y luego se las pasaba al siguiente.

En la segunda iteración, cada participante volteaba de a cinco monedas y luego se las pasaba al siguiente, hasta llegar al último. La tercera ronda era libre, y el equipo propuso que cada jugador girara de a una y la pasara. De modo que, en determinado momento, todo el equipo estaba trabajando con una moneda hasta llegar a pasar todas las monedas al último jugador.

Lo interesante ocurrió al terminar el ejercicio, cuando el equipo analizó los tiempos y las diferencias en las tres dinámicas. El equipo comprobó que, en la última iteración, hubo menos presión y mayor velocidad de entrega. Más aún, pudieron ver la analogía de las monedas y las tareas en su trabajo diario. Cabe destacar que el equipo pudo darse cuenta de que, en la última iteración, la primera moneda llegaba al final en menos tiempo.

Estructurar el trabajo con prototipos y mínimo producto viable (MVP)

Otro de los ejercicios propuestos fue el reto del malvavisco (Reeping, 2013). El ejercicio consiste en dividir al equipo en grupos de tres personas y se les suministra veinte fideos spaghetti, cinta adhesiva y un malvavisco. El objetivo es construir en 18 minutos una torre lo más alta posible que soporte el peso de un malvavisco en la punta (cabe aclarar que el equipo está formado por ingenieros y universitarios del área de sistemas).

Lo interesante y destacable de esta experiencia fue que los equipos planificaron cómo construirla (en algunos casos, hicieron un plano) y se dividieron las partes de la torre, pero solo uno de ellos lo logró luego de darse cuenta de que la clave estaba en construir una torre pequeña e ir agregando niveles de a poco. De este modo, lograron una torre robusta que pudo soportar la golosina en la cima.

Este juego permitió entender la importancia de efectuar prototipos, pruebas piloto antes de embarcarse a desarrollar un producto. La idea de empezar con algo de menor magnitud e ir incrementado su tamaño resultó más efectivo que lanzarse a construir la torre completa desde el primer momento. Por otro lado, esta experiencia abrió el debate al concepto de la bicicleta

ágil (Kniberg, 2016), en el cual se puede observar la importancia de entregar un producto con funcionalidad reducida y luego pequeñas mejoras o incrementos de valor en este, hasta que el cliente esté satisfecho.

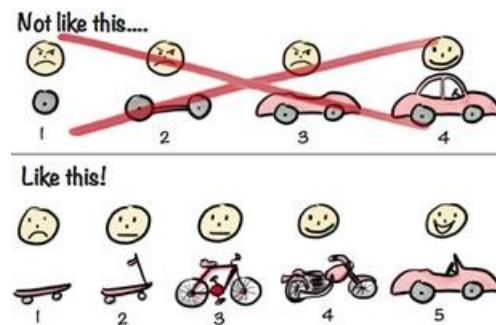


Figura 4. Entregar valor lo antes posible (Kniberg, 2016)

La Figura 4 ilustra el concepto de la bicicleta ágil: desde el momento 1, el cliente dispone de un producto que puede utilizar y decidir si desea continuar con su desarrollo o si es suficiente con la funcionalidad brindada. En otras palabras, el cliente puede evaluar si es suficiente con una patineta o si necesita que se continúe con el producto. Es posible que, antes de llegar a la iteración 5, en la cual se entrega el automóvil, el cliente ya encuentre satisfecha su necesidad, y no haya necesidad de continuar.

Adaptar la metodología a la realidad

Las metodologías ágiles establecen pautas, principios y buenas prácticas. No se debe perder de vista que la raíz de la iniciativa de cambiar la forma de trabajo surge en el desarrollo de software. Muchas de las herramientas, ceremonias y aspectos particulares de Scrum se encuentran orientados a este tipo de tareas. Por eso, no se debe perder de vista que es necesario ser flexibles y definir un entorno de trabajo acorde a las necesidades propias de la industria, la cultura y las características del trabajo en sí.

En el caso bajo estudio, las tareas no son de desarrollo de software, sino, más bien, están relacionadas con el soporte de servidores, el aprovisionamiento y su mantenimiento, incluso con el funcionamiento del sistema operativo y con las aplicaciones de seguridad dispuestas para asegurar su alineamiento con las políticas de seguridad corporativa de la compañía.

Por eso, fue necesario acordar cambios entre todos los involucrados para poder implementar Scrum en este caso particular.

Los clientes piden características entregables con sus historias de usuario correspondientes, pero también existen tareas cotidianas y rutinarias. Aplicar parches de seguridad, efectuar tareas de mantenimiento en servidores, reparación, cambios de configuración, actualización de hardware, reparar los servidores que fallan son algunas de las tareas que se incluyen en el alcance del equipo.

Para incorporar estas tareas en el backlog y que sea compatible con Scrum, se crearon cajones. En el backlog, aparece como una característica, pero no tiene un principio y un fin, y no hay muchos detalles al momento de planificar el período. Sin embargo, este mecanismo permite reservar tiempo de antemano y refleja el esfuerzo estimado y el trabajo comprometido para entregar. Por ejemplo, inicialmente, puede aparecer una historia de usuario ambigua o vacía que, simplemente, mencione la aplicación de parches. Más adelante, se incorpora el parche que se debe aplicar y en qué servidores. Resulta importante que los clientes puedan observar que está previsto efectuar este procedimiento y que el equipo asignó el tiempo para hacerlo.

Asimismo, cabe destacar que, si bien este tipo de tareas tiene un valor agregado para el negocio, no es fácil de mostrar en la ceremonia correspondiente, ya que no es algo novedoso, sino, simplemente, asegurar que el entorno se mantenga en buen estado y compatible con las políticas corporativas. Por lo tanto, la forma de demostrar el progreso es compartiendo las métricas: en el ejemplo anterior, cuántos servidores son compatibles, cuántos recibieron el parche y cuál es el porcentaje de conformidad (compliance) de la red corporativa.

Participación en las ceremonias

Resulta esencial que el equipo sea transparente, abierto y sincero durante las ceremonias para que estas sean efectivas. De nada sirve una reunión diaria si no se comparte lo que realmente se está trabajando y los obstáculos.

En el caso particular, varios de los integrantes no participaban de las ceremonias. Asistían, pero no hablaban. Al principio, fue tomado como timidez y falta de familiaridad con la dinámica y con la metodología, pero, con el paso del tiempo, se volvió un hábito que algunos hablaran y otros no, de modo que las decisiones o los temas de discusión estaban influenciados por unos pocos que siempre participaban.

Este último tema fue tratado varias veces en la reunión de retrospectiva y se acordaron algunas medidas, que se desarrollan a continuación.

Reuniones diarias efectivas

Durante el transcurso de la implementación, el Scrum Master identificó que las reuniones diarias se habían tornado reuniones de estatus y no reuniones diarias de equipo. Cada miembro compartía su porción sin prestar atención o interesarse por la parte del otro; este no es el propósito de dichas reuniones, ya que, si lo fuere, sería más eficiente hacerlo por e-mail. Por lo tanto, se comenzaron a realizar frente a una pizarra en la que figuraban los objetivos de la iteración; y solo hablaría quien tuviera un avance en pos de esos objetivos o un obstáculo que desbloquear. Desde luego, en esta nueva dinámica, no era esperable que compartieran todos los miembros todos los días, pero sí dejaba en evidencia a los tímidos, que no compartían nunca (se podía asumir que no tenían avances o que estaban trabajando fuera de los objetivos de la iteración).

Si bien este último cambio fue positivo, la ceremonia que debía demorar entre 10 y 15 minutos, a veces, duraba 1 hora. Con lo cual, se resolvió hacerla con todos los miembros parados (aun, en forma remota) sin el backlog enfrente y con un tiempo restringido de 1 o 2 minutos cada uno (para evitar que se generaran debates en el medio y que quienes hablan siempre acapararan todo el tiempo reservado); además, se reservó tiempo después de la reunión diaria para quienes tuvieran necesidad de discutir un tema o sobre un área particular.

Solo con el fin de experimentar y probar (y comprobar) que era posible efectuar estas ceremonias durante no más de 15 minutos, se acordó una sesión con el compromiso de que quien hablara debía estar parado en un pie.

Asimismo, con el propósito de mejorar el entendimiento de la dinámica de la reunión de todos los integrantes del equipo, se definió que, durante un mes, debía ser facilitada por alguien que no fuera el Scrum Master para que quien la estuviera liderando pudiera observar la ceremonia desde otro punto de vista.

Innovación en las reuniones de retrospectiva

En general, el facilitador efectuaba las preguntas “¿qué se hizo bien?”, “¿qué se hizo mal?” y “¿qué se debe cambiar de ahora en más?”, esperando que, voluntariamente, todos opinaran.

Por lo tanto, se experimentó con una encuesta anónima (Figura 5), para que todos pudieran opinar desinhibidamente. Esta contenía algunos aspectos y áreas sobre los cuales debatir, tales como rendimiento, valor entregado al cliente, diversión, innovación, aprendizaje, planificación, velocidad de entrega, proceso, trabajo en equipo.

1. ¿Que áreas vamos a discutir en esta sesión?

<input type="checkbox"/> Trabajo en equipo	<input type="checkbox"/> Innovación
<input type="checkbox"/> Procesos	<input type="checkbox"/> Diversión
<input type="checkbox"/> Velocidad de entrega	<input type="checkbox"/> Valor entregado al cliente
<input type="checkbox"/> Planificación y ejecución del trabajo	<input type="checkbox"/> Rendimiento
<input type="checkbox"/> Aprendizaje	

Figura 5. Encuesta para retrospectiva

Lamentablemente, no prosperó que este formato se utilizara con frecuencia, sino una vez por período (3 meses), ya que las opiniones eran muy diversas y la votación sobre las áreas en discusión era heterogénea la mayoría de las veces; por lo tanto, resultaba complicado sacar conclusiones de todas las opiniones en la misma sesión (no se llegaba a debatir todo lo que se proponía).

Finalmente, se optó por implementar un tablero con tres columnas: “¿qué se hizo bien?”, “¿qué se puede mejorar?” y “¿cuáles son las acciones de ahora en más?”.

El facilitador comparte el vínculo del tablero, y todos tienen 5 minutos para ingresar sus opiniones en formato de carta (una oración o una palabra). Luego se agrupan las cartas o ideas por tema, y el facilitador las comparte a todos para discutir. Si hay ítems en la columna “para mejorar”, se votan los más importantes. La idea central es presentar cada ítem y luego debatir el acuerdo o desacuerdo; respecto de cada una de las cartas de la columna “para mejorar”, se debe acordar y planear una acción concreta que alguien del equipo se compromete a completar.

Por ejemplo, si se identifica que se debe incrementar la comunicación con el cliente, un miembro del equipo se compromete a agendar reuniones periódicas con cada uno de ellos para mostrar el progreso de sus pedidos y recolectar retroalimentación a fin de mejorar el resultado final.

Mejoras implementadas de alto impacto

Por último, es necesario destacar las mejoras implementadas que brindaron buenos resultados y un cambio significativo en el equipo. Si bien algunas parecen triviales o forman parte de la

metodología desde el punto de vista teórico, al focalizarse el equipo en ellas, el rendimiento cambió drásticamente y la dinámica mejoró en forma considerable.

Acordar objetivos por iteración

En paralelo a estos cambios, se acordó que, antes de cada iteración, el dueño de producto compartiría por e-mail antes de comenzar la reunión de planificación los objetivos de esta a fin de quedar documentado y claro para todos lo que se esperaba al final del ciclo ágil. Esto facilitó que el equipo pudiera hacer foco en lo realmente planificado y acordado con el cliente. Todo lo que surgiera fuera de los objetivos debía ser acordado y priorizado con el dueño de producto.

Existieron casos en los cuales, en medio de la iteración, surgieron urgencias, y hubo que redefinir los objetivos de la iteración. Sin embargo, estos cambios fueron acordados con el cliente, y eran visibles en el backlog de inmediato, con lo cual el cliente podía saber dónde estaba enfocado el esfuerzo del equipo y estaba al tanto de que los entregables comprometidos al principio del Sprint no iban a poder ser entregados como se había acordado en un primero momento.

Demostración con los clientes

Si bien la metodología dispone que se deben efectuar demostraciones con los clientes frecuentemente, en este caso particular, no era clara la frecuencia ni las pautas para efectuarlas. El equipo asumía que se harían cuando fueran necesarias. Sin embargo, gracias a comentarios del cliente y a las discusiones en las reuniones de retrospectiva, se identificó la falta de comunicación con el cliente. No porque no estuviera al tanto de lo que hacía el equipo, sino por falta de involucramiento en el desarrollo de la solución o de la puesta en marcha del trabajo solicitado.

Las comunicaciones eran muy informales, y, en ciertas ocasiones, ambas partes daban por sentado cuestiones técnicas o de diseño erróneamente (las asumían). Por eso, el dueño de producto acordó con los clientes que, como mínimo, cada dos iteraciones habría una reunión corta formal de demostración de equipo y cliente, en la cual se mostrara el avance y se documentaran las discusiones a fin de evaluar su satisfacción o considerar algún cambio. Esta fecha límite imponía al equipo un ritmo y un compromiso diferente, ya que, al planificar el trabajo, el equipo debería asegurarse de que, cada dos iteraciones como mínimo, hubiera un avance considerable para compartir que evidenciara el valor brindado. De hecho, esta práctica

ayudó a que el equipo pudiera adoptar el concepto de “fallar rápido” (Shore, 2004) con el cual el grupo se asegura de fallar lo antes posible en etapas tempranas (y no cuando el trabajo está terminado) sin miedos ni culpas, tomándolo como algo natural y parte del proceso de desarrollo y aprendizaje.

Desarrollo de un equipo con forma de “T”

El objetivo desde el principio fue formar un equipo ágil con forma de “T”. Esto implica que cada miembro conozca cada una de las capacidades del equipo. Si bien los equipos en Scrum pueden ser multidisciplinarios, la misma metodología sugiere que el desarrollador debe saber sobre el proceso de pruebas, y viceversa. En este caso, a medida que el equipo fue adquiriendo conocimiento, cada integrante fue tomando tareas y desarrollando sus capacidades en las áreas trabajadas. Esto último era evidente, ya que todo un equipo no podría abarcar y aprender todas las áreas al mismo tiempo. Sin embargo, al pasar el tiempo, en las reuniones de retrospectiva, se identificó un círculo o patrón en el cual los miembros del equipo tomaban del backlog las tareas en las que más confiados se sentían, sin salir de su zona de confort y sin fomentar el desarrollo de otras en las diferentes tecnologías que el equipo soportaba. Por eso, se tomaron dos medidas sumamente útiles y efectivas. La primera consistió en imponer dentro de las historias de usuario la condición de que, para ser completada, deberían trabajar dos o más miembros del equipo sin excepción (condición incluida en el criterio de aceptación). Esta restricción implicaba que, si un miembro se enfermaba, se ausentaba o surgía un imprevisto, al menos, otro miembro del equipo podría ayudar a resolverlo porque estaría al tanto del estado específico del trabajo.

La segunda medida consistió en llevar a cabo un bingo de habilidades (skills bingo). Un ejercicio en el cual el equipo se reunía cada seis meses a revisar sus capacidades como equipo y a evaluar cuáles sería necesario desarrollar para ser más eficientes. En pocas palabras, una reunión en la cual se listaban en una tabla las capacidades que necesitaba el equipo para efectuar su trabajo y luego se listaba a cada miembro del equipo en columnas. Después, cada miembro se autoevaluó completando el casillero correspondiente de la capacidad debajo de su nombre con “0- no sé de qué se trata”, “1- lo puedo resolver con Google” o “2- soy un experto”.

De este modo, al final de la sesión, se obtenía un panorama general de las capacidades del equipo y se identificaban las que era necesario desarrollar.

Cabe mencionar que, una vez identificadas las capacidades para desarrollar, se puede definir un plan de acción; por ejemplo, involucrar al individuo A en el trabajo X, o acordar que cierta tarea no la tome más la misma persona (Tabla 1).

Habilidad	Miembro A	Miembro B
Desarrollo de scripts	1	2
Soporte de sistema operativo	2	0
Soporte de herramientas de seguridad	0	1

Tabla 1. Ejemplo de bingo de habilidades

Corte vertical del trabajo

Otro de los grandes problemas y dificultades observadas en las reuniones de retrospectiva y un desafío para demostrar que, realmente, un equipo es ágil es el concepto de división vertical de la historia de usuario. Un segmento vertical implica un corte transversal. Por ejemplo, en las tareas del equipo, si el pedido del cliente era entregar 10 servidores web, una historia que implicara “entregar un servidor web” es una historia bien dividida. Sin embargo, bajo la mentalidad de trabajo tradicional, la historia de usuario hubiese sido “planificar la construcción de 10 servidores web”, luego “construir 10 servidores” y después “configurar y entregar los 10 servidores”. Cabe aclarar que esta historia de usuario incluirá en su alcance las pruebas correspondientes para asegurarse de que sea entregado bajo las normas de seguridad/calidad y tal como fue acordado.

El corte vertical desafía al equipo a descomponer sus historias suficientemente pequeñas, al mismo tiempo que logran un corte vertical. Si bien en teoría todos los miembros lo entienden, en varias ocasiones, se vuelve a los viejos hábitos de crear historias de usuarios divididas horizontalmente. Por ejemplo, definir una historia para crear un servidor, una historia para crear la base de datos y una historia más para probar.

El corte vertical permite a los equipos demostrar un progreso incremental, aumentando así la transparencia.

En un primer momento, se pudo comprobar que el dueño de producto no estaba siendo eficaz; no por incompetencia, sino porque las historias de usuario no estaban divididas verticalmente.

El equipo estaba trabajando en una iniciativa que implicaba una complejidad técnica significativa.

Finalmente, luego de varios ejemplos y retrabajo, se logró reestructurar el trabajo en forma vertical y aprender cómo debía enfocarse el trabajo según la mentalidad ágil. El equipo vio y citó este concepto varias veces en forma práctica como el corte de una torta: nadie quiere probar solo la parte de arriba, sino que la porción debe incluir todas las capas para que, efectivamente, sea una porción de torta. El corte horizontal llevaba a demorar la entrega de valor en forma inmediata y a “entregar” solo una capa por vez al cliente, lo cual implicaba que debía esperar para probar el sabor final de la torta elaborada.

CONCLUSIONES

Desde el punto de vista del marco teórico, basado en los principios de las metodologías ágiles, la transformación representa un proceso planificado con anticipación y en continuo ajuste. Dentro del proceso de cambio, varias áreas de la compañía bajo estudio deben involucrarse y estar alineadas para alcanzar el éxito, y que los cambios lleguen a tiempo y de acuerdo con el estadio de la transformación.

Es indispensable que la alta dirección se encuentre conforme con la propuesta y que Recursos Humanos pueda ofrecer un soporte lo suficientemente flexible para facilitar y agilizar el cambio. La misma metodología degrada la noción de jerarquía, tan arraigada en las organizaciones. Si bien a nivel corporativo este concepto es muy difícil de eliminar por completo, la metodología propone mediante los conceptos de empoderamiento y autonomía que estas barreras culturales sean derribadas de algún modo, permitiendo un acercamiento o una intimidad particular del equipo con sus clientes y con la alta dirección. De nada sirve si el equipo se embarca en la transformación, pero la dirección no le permite tomar decisiones o ser independiente. Existen muchos casos en los cuales el equipo se dice ágil, pero dispone de un supervisor o de un jefe que define qué, quién y cómo se efectúa el trabajo. Por lo tanto, una condición indispensable para el éxito de la transformación es el nivel de confianza que la dirección deposita en el equipo y en cada miembro en particular.

A raíz del análisis realizado durante el desarrollo del trabajo que abarca el objetivo principal, en el cual se propone describir en detalle el proceso de transformación ágil de un equipo de tecnología de en el sector energético, se puede evidenciar que esta transformación demuestra que cambios de este tipo son un recorrido que cada equipo efectúa en busca de su propio equilibrio para que se produzca más y se genere más valor al cliente, lo cual se ve reflejado en los hábitos adoptados incorrectamente, en los ejercicios y prácticas llevados a cabo para disparar los cambios necesarios y en la conciencia de este nuevo método de trabajo.

El tiempo de duración depende de la cultura y de los valores previos al cambio, de la predisposición y de la adaptabilidad de todos para que este suceda. Por ende, cabe destacar que, antes de plantear una transformación ágil, deben considerarse precondiciones a fin de que el proceso pueda fluir sin demoras. La transformación no ocurre de un día para el otro, sino que el progreso debe ser gradual.

Al principio, puede ser frustrante, los miembros del equipo sienten que es incómodo, hasta una pérdida de tiempo y aburrido. Sin embargo, es cuestión de práctica, voluntad y tiempo para que la mente piense de un modo ágil; y para que la forma de encarar los desafíos y dividir el trabajo cambie de una mentalidad orientada a entregar valor lo antes posible a aceptar fallar y a buscar fallar lo antes posible; donde el error no sea algo negativo, sino esperado.

Cada equipo e implementación es diferente por la experiencia previa de los integrantes, por la cultura de la empresa y por el tipo de tareas que efectúa el equipo. Para que sea efectiva, es necesario planificar la transformación y ajustar el plan cuando sea necesario.

Algunos logran cambiar la mentalidad y virar rápidamente; otros se mantienen en una metodología híbrida hasta que comienzan a pulir malos hábitos (lo cual puede llevar años); otros nunca lo logran y dicen ser ágiles tan solo por llevar a cabo las ceremonias mencionadas por la metodología, pero su forma de desenvolverse no es ágil: su trabajo no genera un valor incremental en cada iteración, y, probablemente, las ceremonias no sean efectivas en absoluto.

El cambio de paradigma es indispensable para madurar dentro de la metodología. Por eso, es clave continuar identificando mejoras y experimentando hasta llegar a un equipo óptimo. La metodología establece que incluso se puede llegar a una instancia en la cual el equipo sea independiente y autoorganizado, al punto de no necesitar un Scrum Master; que del mismo equipo surja un voluntario espontáneo que facilite las reuniones y ayude al equipo a maximizar la entrega de valor.

En conclusión, el cambio puede compararse con el proceso de aprender a andar en bicicleta, en el cual lleva tiempo acostumbrarse a coordinar todas las partes del cuerpo: pedalear, mantener el equilibrio, darle dirección, etc., conscientes de todo lo que está ocurriendo y de cada movimiento. Pero una vez que el individuo aprende a hacerlo, ya puede andar de forma natural y sin pensarlo. A cada persona puede llevarle más o menos tiempo y practicar de una u otra forma, pero es cuestión de subirse e intentar pedalear para que empiece a incorporarse el proceso.

Según los diferentes autores citados, cada actividad propuesta es efectiva en un entorno y puede o no ser exitosa en otro contexto (como la planteada para este análisis). Dicho de otro modo, la actividad que funciona en un equipo puede no funcionar en otro.

Asimismo, en relación con uno de los objetivos secundarios, luego de listar las prácticas y los experimentos efectuados para facilitar la implementación, se puede concluir que la transformación se construye probando alternativas, permitiendo flexibilidad dentro de lo que la metodología admite y brindando al equipo el poder de definir cómo organizarse y cómo entregar mayores beneficios.

Sin embargo, en esta instancia, es necesario destacar que, desde el punto de vista del aprendizaje y de la adopción de la metodología, es altamente recomendable que el equipo en el inicio respete todas las pautas al pie de la letra para asegurarse de no perderse ningún paso o saltarse un proceso indispensable para el aprendizaje y la absorción de conocimiento. Es decir que, inicialmente, aunque exista resistencia, el equipo debe respetar todas las ceremonias propuestas por la metodología a fin de comprender el sentido de cada una.

Una vez que el equipo se siente afianzado y maduro, puede evaluar si, dadas las circunstancias, resulta más efectivo reducir la frecuencia o duración de alguna ceremonia en particular.

Esto último queda expuesto en varios casos citados, como por ejemplo el cambio de formato a un evento en las reuniones diarias donde quien habla debe estar parado sobre un pie. La idea puede ser descabellada, pero, a la vez, muy efectiva para reducir la duración; y dado que surge del mismo equipo y existe un acuerdo entre todos de llevarla a cabo de este modo, la propuesta es totalmente válida. En esta circunstancia, cabe aclarar que la ceremonia es en sí continua, el contenido es el que la misma metodología define, pero la dinámica es lo que cambia.

Por ende, el camino de la transformación puede ser visto como una serie de actividades y experimentos que lleven al equipo a adoptar su propia dinámica y a encontrar dentro de la metodología su propio sabor.

Asimismo, resulta de vital importancia que el equipo pueda habituarse a la cadencia, a trabajar con objetivos por iteración, comprometerse a cumplirlos y, a su vez, al entregar o terminar el trabajo pueda entender el valor que representa su aporte a la compañía. Esto último, además de promover la motivación, permite que, como equipo o individualmente, pueda visualizar el fruto del trabajo realizado con un principio y un fin.

Además, cabe destacar que uno de los primeros pasos para comenzar con la implementación consiste en asegurarse de que el equipo se conozca, se sienta cómodo y sus miembros tengan

confianza entre sí. Por más que el equipo sea heterogéneo desde el punto de vista generacional, desde las capacidades de cada uno, debe existir un acuerdo explícito o implícito de que se parte de buenas intenciones y de la confianza en el otro. De nada sirve plantear una reunión de retrospectiva en un equipo donde recién se conocen, ya que algunos pueden estar inhibidos, otros pueden ofenderse de lo que dice su compañero, y los miembros pueden tomar los comentarios en forma personal. La reunión será efectiva cuando entre todos exista una relación de confianza y entiendan que el sentido de los comentarios o lo que se discute es confidencial, y posee el objetivo de mejorar la forma de trabajo para ser más productivos.

Como se anticipó en uno de los objetivos secundarios, el trabajo representa una guía de referencia para futuras implementaciones, ya que brinda diferentes posibilidades y alternativas para abordar la transformación ágil de forma efectiva. Sin embargo, no existen recetas en este tipo de cambio cultural, puesto que involucra a personas, y cada persona puede reaccionar de forma diferente ante el mismo estímulo o situación. Queda a criterio de quienes estén planeando la transformación evaluar cuáles pueden ser las prácticas que den resultado y cuál es el momento adecuado para llevarlas a cabo de acuerdo con el nivel de madurez del equipo y con el nivel de adopción de la mentalidad ágil. El presente análisis promueve la necesidad de evaluar y analizar la implementación de forma continua para asegurarse de que el equipo va adoptando los diferentes conceptos propios de la metodología para mejorar su eficiencia de inmediato.

La transformación en sí debe adaptarse al equipo en el cual se está implementando. Ya fue evidenciado en el desarrollo del caso que, si bien la metodología tiene su origen en el desarrollo de software y está estructurada para este tipo de actividades, es posible flexibilizarla tanto como sea necesario y aplicarla más allá de la disciplina para la cual fue creada. Justamente, el desafío de cada implementación consiste en encontrar las prácticas, los hábitos y los procedimientos que permitan que las actividades se adapten a la metodología y cada miembro del equipo pueda estar focalizado en la entrega de valor incremental más allá de si el entregable es software, un reporte, un servicio o tareas de soporte para que el negocio pueda seguir funcionando con la infraestructura que necesita y en buen estado.

Por último, las métricas representan un punto interesante para analizar la performance y la madurez del equipo ágil, permitiendo concientizar sobre lo que se está haciendo bien y lo que se puede mejorar desde el punto de vista del proceso. Como ya fue mencionado, las métricas son un marco de referencia y constituyen sugerencias o recomendaciones para medir la

eficiencia del equipo en términos ágiles, pero la misma metodología no impone ninguna en particular. Que el equipo pueda analizar si, efectivamente, está planificando bien las historias de usuario, si está estimando correctamente el esfuerzo ayuda a plantear un panorama del nivel de agilidad del equipo al mismo tiempo que brinda las pautas de qué área o qué parte de la implementación se puede ajustar y mejorar.

REFERENCIAS

- 12 Principles Behind the Agile Manifesto | Agile Alliance,
<https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>
- Antonio Capobianco D. (2016) Shu-ha-ri: How to Break the Rules and Still Be Agile. In: Ciancarini P., Sillitti A., Succi G., Messina A. (eds) Proceedings of 4th International Conference in Software Engineering for Defence Applications. Advances in Intelligent Systems and Computing, vol 422. Springer, Cham. https://doi.org/10.1007/978-3-319-27896-4_19
- Barlow, J. B., Keith, M. J., Wilson, D. W., Schuetzler, R. M., Lowry, P. B., Vance, A., & Giboney, J. S. (2011). Overview and Guidance on Agile Development in Large Organizations. Communications of the Association for Information Systems, 29, 25–44. <https://doi.org/10.17705/1CAIS.02902>
- Boehm, B. and R. Turner (2005) "Management Challenges to Implementing Agile Processes in Traditional Development Organizations", IEEE Software (22)5, pp. 30–39.
- Casanova, P. (2013). Agile processes: a unifying approach for the future of projects. Paper presented at PMI® Global Congress 2013—EMEA, Istanbul, Turkey. Newtown Square, PA: Project Management Institute.
- Cockburn, A. and Highsmith, J. Agile software development 2: The people factor. IEEE Computer (Nov. 2001) pp. 131-133.
- Demigha, S., & Kharabsheh, R. (2019). Knowledge Sharing in an Agile Organization. Proceedings of the European Conference on Knowledge Management, 1, 267–277. <https://doi.org/10.34190/KM.19.102>
- Deming, W. Edwards (1982). Out of the Crisis, Center for Advanced Engineering Study, Massachusetts Institute of Technology, Cambridge, Massachusetts
- Doz, Y., & Guadalupe, M. (2019). Escaping the “S-Curve” - is the “Agile” Organization the Answer? INSEAD Working Papers Collection, 15, 1–30.
- Fustik, V. (2017). The Advantages of Agile Methodologies Applied in the Ict Development Projects. International Journal on Information Technologies & Security, 9(4), 51–62.
- Sjoberg (2019). Team Building Icebreaker – Constellation Exercise <http://anagemind.net/2019/02/20/team-building-icebreaker-constellation-exercise>
- Kniberg (2016). Making sense of MVP (Minimum Viable Product) <https://blog.crisp.se/2016/01/25/henrikkniberg/making-sense-of-mvp>
- K. Schwaber, J. Sutherland (2013). La guía de Scrum, <https://www.scrumguides.org/index.html>.
- Kniberg (2016). Making sense of MVP (Minimum Viable Product) - and why I prefer Earliest Testable/Usable/Lovable - Crisp's Blog
- Leon Tranter (2018). Agile Metrics: the Ultimate Guide - Extreme Uncertainty <https://www.extremeuncertainty.com/agile-metrics-ultimate-guide/>

Linnanvuo (2015). Agile & Lean Metrics: Cycle Time — Screenful
<https://screenful.com/blog/software-development-metrics-cycle-time>

Makoto Higuchi, M., & Noboru Nakano, D. (2017). Agile Design: A Combined Model Based on Design Thinking and Agile Methodologies for Digital Games Projects. *Revista de Gestão e Projetos*, 8(2), 109–126. <https://doi.org/10.5585/gep.v8i2.528>

Manifiesto por el Desarrollo Ágil de Software,
<https://agilemanifiesto.org/iso/es/manifiesto.html>

Moreira M.E. (2017). Achieving Better Business Outcomes. In: *The Agile Enterprise*. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-2391-8_3, pp 21-27

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5), 72-78.

Organized and disciplined, <https://www.ayberk.co/en/blog/organized-and-disciplined>

Palacios, V. Merchán (2014). “Guía de fundamentos para la dirección de proyectos de desarrollo de software con enfoque pmi y los métodos ágiles,” Ecuador.

Reeping D., Reid K, (2013). *The Marshmallow Metaphor—Iterative Design Tailored to 6th Graders*, Ohio Northern University, Ada, OH 45810

Schwaber K, Beedle M. (2001). *Agile software development with scrum*. Upper Saddle River (NJ): Prentice Hall.

Shore, "Fail fast [software debugging]," in *IEEE Software*, vol. 21, no. 5, pp. 21-25, Sept.-Oct. 2004, doi: 10.1109/MS.2004.1331296.

Takeuchi y Nonaka, *The New New Product Development Game* - Harvard Business Review, <https://hbr.org/1986/01/the-new-new-product-development-game>

Team Building Icebreaker - Constellation Exercise - An Agile Mind

Vaasanthi, R., Kumari, V.P., & Kingston, S.P. (2017). Adopting SHU-HA-RI Training Methodology with Scrum Framework in DevOps Space. *International Journal of Computer Applications*, 166, 28-32.

von Eva-Maria Ayberk (2020). Matriz de responsabilidades,
<https://www.ayberk.co/en/blog/organized-and-disciplined>

Walenta, T. (2017). Agile Transformation for Organisations and Projects. *PM World Journal*, 6(12), 1–5

ANEXOS

	Traditional	Agile
Fundamental Assumptions	Systems are fully specifiable, predictable, and can be built through meticulous and extensive planning.	High-quality, adaptive software can be developed by small teams using the principles of continuous design improvement and testing based on rapid feedback and change.
Control	Process centric	People centric
Management Style	Command-and-control	Leadership-and-collaboration
Knowledge Management	Explicit	Tacit
Role Assignment	Individual—favors specialization	Self-organizing teams—encourages role interchangeability
Communication	Formal	Informal
Customer's Role	Important	Critical
Project Cycle	Guided by tasks or activities	Guided by product features
Development Model	Life cycle model (Waterfall, Spiral, or some variation)	The evolutionary-delivery model
Desired Organizational Form/Structure	Mechanistic (bureaucratic with high formalization)	Organic (flexible and participative encouraging cooperative social action)
Technology	No restriction	Favors object-oriented technology

Tabla 2. Desarrollo de software tradicional versus agile (Nerur, 2005)

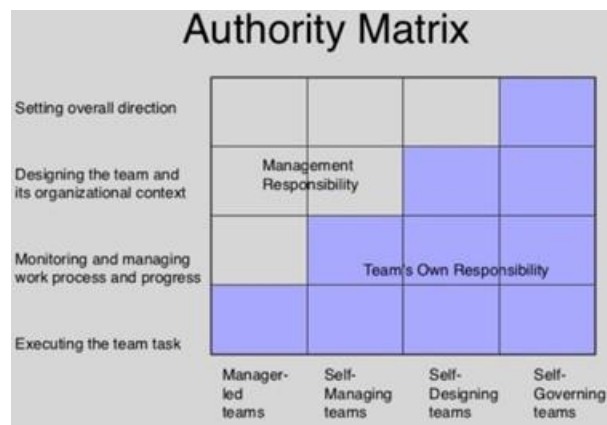


Tabla 3. Matriz de responsabilidades (von Eva-Maria Ayberk, 2020)