



MESONES - Herveo - Tolima

*El conflicto en el desarrollo ágil:
una perspectiva desde el SCRUM*

SÍNTESIS:

El artículo plantea una discusión sobre la importancia del conflicto como oportunidad para el mejoramiento continuo dentro de los equipos que trabajan bajo la metodología de desarrollo de *software*, conocida como SCRUM. Consiste en un estudio investigativo de tipo cualitativo, que presenta los resultados de una investigación documental, además de una serie de entrevistas realizadas a trece personas que llevan a cabo el rol de *Scrum Master*, dentro de una de las empresas de desarrollo de *software* más importantes de Colombia. La conclusión más importante es la valoración que hacen los entrevistados del conflicto, como una dinámica que propicia la maduración y el mejoramiento continuo en los equipos de trabajo que usan metodologías de desarrollo ágil.

PALABRAS CLAVE:

SCRUM, conflicto, equipos de trabajo, metodologías ágiles.

Clasificación JEL: C87, D74, J21, C8

ABSTRACT:

The present article seeks to raise a discussion about the importance of conflict as a tool for continuous improvement within teams that work under the software development methodology known as SCRUM. It consists of a research study of qualitative type, which presents the results of a documentary research, besides a series of interviews carried out among thirteen people who carry a Role of Scrum Master within one of the most important software development companies in Colombia. The most important conclusion is the interviewees' assessment of the conflict as a dynamic that encourages maturation and continuous improvement in work teams using agile development methodologies.

KEYWORDS:

SCRUM, conflict, work teams, agile methodologies.

Clasificación JEL: C87, D74, J21, C8

El conflicto en el desarrollo ágil: una perspectiva desde el SCRUM¹

C

Esteban Villegas Gómez²

Johan Miguel Ruiz Rodríguez³

Francisco López Gallego Ph. D.⁴

Conflict in the agile development: perspective from the SCRUM

Primera versión recibida el 2 julio de 2016. Versión final aprobada el 31 de Agosto de 2016.

Para citar este artículo: Villegas Gómez, Esteban, Ruiz Rodríguez, Johan M. López Gallego, Francisco (2016). El conflicto en el desarrollo ágil: una perspectiva desde el SCRUM. En: Revista Gestión y Región N° 21 (Enero-Junio de 2016); pp 121-149

La industria del desarrollo de *software* ha crecido rápidamente en las últimas décadas, a tal punto que se puede afirmar que la mayoría de las empresas del mundo se encuentran en proceso de implementar un producto de *software* o están considerando hacerlo en el corto plazo. Sin embargo, la mayoría de las empresas enfrentan complicaciones durante la ejecución de este tipo de proyectos, llegando al punto de cancelarlos sin haber obtenido beneficio alguno por la inversión realizada. De acuerdo con Standish Group (2015), en el año 2015 solo el 29% de los proyectos de desarrollo de *software* se consideraron exitosos, el 52% tuvieron algún tipo de complicación y el 19% fueron declarados como fracasos.

Los resultados en los proyectos de construcción de productos de *software* han sido generalmente preocupantes a través de los años, lo que ha llevado a la industria a estar en una búsqueda constante de una metodología de trabajo, que permita estandarizar la manera de hacer las cosas y reducir la cantidad de inconvenientes presentados durante la ejecución de los proyectos. Si bien es cierto que al analizar los proyectos se encuentra que estos tienen razones de fallas diferentes, de acuerdo con Jian (2005), entre las fallas principales en los proyectos de *software* se encuentran: el costeo inicial, la poca interacción con los posibles usuarios finales del mismo durante las etapas de definición, el poco entendimiento de los requerimientos y las necesidades reales, la falta de investigación para encontrar opciones y la incompreensión de los productos disponibles en el mercado.

¹ El artículo es el resultado de la investigación de tipo cualitativo y documental, además se complementa con entrevistas realizadas a trece personas que llevan a cabo el rol de Scrum Master. Investigación realizada en el programa de maestría en Administración de la Universidad EAFIT. Medellín.

² Estudiante en Maestría en Administración de la Universidad EAFIT. Contacto: ovinoslaberraquera@hotmail.com

³ Estudiante de maestría en Administración de la Universidad EAFIT. Contacto: jruiroz@eafit.edu.co

⁴ Profesor Escuela de administración Universidad EAFIT. Asesor del proyecto de investigación.

Contacto: flopez@eafit.edu.co

El conflicto ha estado presente a través de toda la historia de la humanidad, en todas las comunidades y sociedades; las organizaciones no son la excepción. Por eso es que el conflicto ha tenido especial interés para muchos expertos; a través del tiempo, se ha visto, dadas las diversas ideologías, como algo completamente negativo, por una parte, y por otra, como un elemento positivo. Esto es lo que exponen Lopez y González (2012), quienes al estudiar el tema han encontrado que para algunos es “una realidad indeseada”, y cuando se ve de esta manera se considera que eventualmente podría llegar a convertirse en un obstáculo; por ende, debe ser atacado inmediatamente en cuanto hace su aparición. Pero los autores antes mencionados también han establecido que existen personas que no ven el conflicto de una forma negativa; por el contrario, lo consideran un elemento que hace posible el movimiento social de las organizaciones, ayudando a que en ellas se dé el cambio. Con todo, se resalta la idea de que en el mundo existen diversas percepciones del conflicto.

Partiendo de que el conflicto es inherente al ser humano, es apenas normal que en los equipos de trabajo se presenten conflictos y, dado que “el software es desarrollado por personas para personas” (Michael, Maurer & Tessem, 2005, citado en Crawford, Soto, Leon de la Barra, Crawford y Olgúin, 2014), se puede deducir que los conflictos estarán siempre presentes en los proyectos de desarrollo de *software*, y que es importante que las personas que trabajan en dicha industria entiendan lo que implica y sepan administrarlos correctamente.

Este trabajo muestra indicios de que existe una relación positiva entre la aparición y administración de conflictos en los equipos de trabajo que usan SCRUM como metodología y su constante mejoramiento en términos de desempeño. El estudio realizado para la elaboración de este trabajo es de tipo descriptivo y fue llevado a cabo como se propone a continuación.

Inicialmente se realizó una investigación documental en dos frentes: en el primero se buscaba realizar una breve descripción histórica de la evolución de los métodos de construcción de *software* y presentar la metodología SCRUM teóricamente, para facilitar el entendimiento de los resultados de la investigación. El otro frente de investigación está relacionado con el conflicto; en este caso, se buscó encontrar algunas construcciones de conflicto realizadas desde la teoría por varios autores identificados previamente, además de presentar algunas de las caracterizaciones del conflicto realizadas a través de los años.

Una vez construido el marco teórico se realizó la preparación y elaboración de entrevistas, teniendo en cuenta los conceptos encontrados durante la investigación documental. Para la elaboración de las entrevistas se seleccionó una de las empresas de desarrollo de *software* más importantes de Colombia y luego se seleccionó a los posibles candidatos para ser entrevistados, teniendo en cuenta su experiencia, principalmente.

Las metodologías de desarrollo de software

Antes de que aparecieran los primeros lenguajes de programación en los años 50, la programación consistía principalmente en trabajar con interruptores eléctricos. Los computadores disponibles en la época tenían muchas limitantes físicas para la ejecución de programas, por lo que la principal preocupación de los programadores era lograr que los programas corrieran en las máquinas disponibles, mas no había interés en definir metodología de trabajo alguna para estandarizar el proceso (Gacitúa Bustos, 2003).

Metodologías tradicionales

En el año 1970, Winston W. Royce presentó las primeras definiciones de lo que luego se conocería como Metodología de desarrollo en *Cascada*, y que se convertiría, con el paso de los años, en la más popular de las metodologías de desarrollo tradicionales. En el método de Cascada, el desarrollo de un producto de *software* se dividía en fases, de manera que cada una debía terminar antes de que comenzara la siguiente, como se muestra en la Figura 1 (Royce, 1970).

Si bien Royce plantó las bases para la metodología de desarrollo en Cascada, el término (*Waterfall*) solo fue acuñado por Bell & Thayer (1976).

En 1986, Barry W. Boehm presentó una novedosa metodología de desarrollo de software llamada Desarrollo en espiral. Este modelo buscaba presentar una alternativa al modelo de Cascada, que tuviera en cuenta los riesgos que conllevaba el *software*; el modelo proponía entonces ciclos basado en prototipos, en los que se desarrollaba la porción del sistema que representara el menor riesgo para el proyecto. Antes de comenzar cada uno de estos ciclos, se debían identificar los objetivos de la porción del sistema a construir, las posibles alternativas que se tuvieran para alcanzar dicho objetivo y las restricciones que cada una de esas alternativas conllevaba (Boehm, 1988).

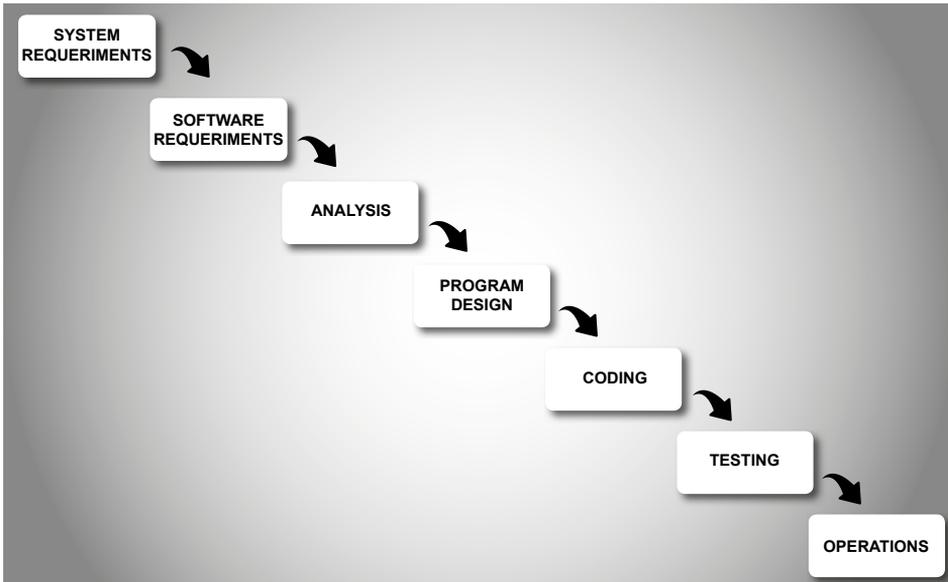


Figura 1. Metodología de desarrollo en Cascada (Royce, 1970)

El modelo propuesto por Boehm sentó las bases para el nacimiento de la metodología de desarrollo conocida como Proceso Unificado Racional (RUP, *Rational Unified Process*) en el año 1999, que luego se convertiría en la metodología más usada hasta nuestros tiempos. RUP propone una división del proyecto de desarrollo de *software* en 4 fases: Iniciación, Elaboración, Construcción y Transición. Dentro de cada una de estas fases, el equipo debía ejecutar actividades de todo el ciclo de desarrollo (Modelado de negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas y Despliegues). Aunque teniendo en cuenta que, dependiendo de la fase en la que se encuentre el proyecto, habrá más enfoque en algunas actividades que en otras. Así, por ejemplo, durante la fase de Construcción se realizan todas las actividades, pero la mayor parte del esfuerzo se enfoca en la construcción del producto (Rational Unified Process, 2001).

Metodologías ágiles

Las metodologías ágiles para el desarrollo de *software* surgieron como una alternativa a las metodologías de desarrollo de tradicionales ya presentadas, que eran consideradas como muy “pesadas”, tanto por los teóricos como por los desarrolladores. El esquema de desarrollo de *software* ágil propone un esquema de construcción basado en iteraciones cortas muy productivas y tiene como enfoque principal a las personas y la colaboración (Fowler & Highsmith, 2011), a diferencia del método tradicional, en el que se prestaba más importancia a la

metodología que al producto mismo, lo que causaba que los equipos de trabajo perdieran el foco de lo que realmente importaba, construir *software* que generara valor para el cliente.

Las metodologías tradicionales promovían el uso de equipos basados en roles y planes detallados, que cubrían todo el ciclo de vida de desarrollo de *software*. En ellas, el foco no está en los individuos y sus capacidades creativas, sino en los procesos mismos. Los métodos ágiles, por el contrario, valoran más a los individuos y sus interacciones que a los procesos. Los métodos de desarrollo tradicionales usan documentación rigurosa para administrar el conocimiento adquirido en todas las etapas del desarrollo, mientras que los métodos ágiles sugieren que la mayor parte de la documentación puede ser reemplazada por comunicaciones informales internas entre los miembros del equipo y entre estos y el cliente (Crawford, Soto, Leon de la Barra, Crawford, & Olguin, 2014).

Las metodologías ágiles comenzaron a surgir de manera experimental a principios de los años 90, pero el término surgió apenas en el año 2001, con la aparición del manifiesto ágil, resultado de una reunión en la que participaron sus representantes y otras personas interesadas, para tratar temas relativos al desarrollo de *software* y conformar lo que luego se conocería como Alianza Ágil (Agile Alliance, 2001).

En ese manifiesto, los firmantes presentaron las prioridades del desarrollo ágil y establecieron las bases para un pequeño marco de trabajo que buscaba, ante todo, mejorar la vida de las personas que trabajaban en la industria del *software* y el resultado de los proyectos para los clientes. A continuación se presenta el manifiesto, como se puede todavía encontrar en el sitio web (Beck, et al., 2001):

Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros.

A través de este trabajo hemos aprendido a valorar:

Individuos e interacciones sobre procesos y herramientas.

Software funcionando sobre documentación extensiva.

Colaboración con el cliente sobre negociación contractual.

Respuesta ante el cambio sobre seguir un plan. Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

Del manifiesto presentado se desprenden también los 12 principios del *software* ágil, que definen las características principales de un proceso de desarrollo ágil y enmarcan las principales diferencias entre esas metodologías y las tradicionales. Estos son los 12 principios, tal como aparecen en el sitio web de la Alianza Ágil (Agile Alliance, 2001):

- Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- 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.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El *software* funcionando es la medida principal de progreso.
- Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños, emergen de equipos auto-organizados.
- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Este trabajo se enfoca, principalmente, en la metodología de desarrollo ágil, conocida como SCRUM, que para el año 2016 es la más usada en el mundo, entre todas las metodologías de desarrollo ágil (VersionOne, 2016), pero es importante aclarar que existen otras metodologías en la misma área, algunas de las cuales son:

Joint Application Development: Chuck Morris y Tony Crawford desarrollaron esta metodología de trabajo a finales de los años 70, que luego publicaron y comenzaron a enseñar durante los años 80. Buscaba involucrar más a los usuarios en el diseño de los productos de software a través de talleres, para mejorar las especificaciones del proyecto, y agilizar de esta manera la labor de desarrollo del producto (Crock, 2008).

Rapid Application Development: James Martin propuso este proceso en 1990, el cual buscaba presentar un marco de trabajo para el desarrollo de sistemas de alta calidad, de manera rápida y económica. Se enfocaba en la velocidad de construcción del producto, utilizando técnicas de recolección de requerimientos, con grupos de enfoque, prototipos, reutilización de componentes de *software*, reducción de la formalidad en las revisiones y las comunicaciones del equipo de trabajo. (Beynon-Davies, Carne, Mackay, & Tudhope, 1999).

Adaptive Software Development: desarrollada por Jim Highsmith y Sam Bayer a comienzos de 1990, hace énfasis en la adaptación continua al proceso de trabajo. Su ciclo de vida está dado por tres pasos principales: especular, colaborar y aprender (Riehle, 2001).

Metodologías Cristal: Alistair Cockburn las presentó en 1992. Consistían en un conjunto de metodologías cuyo enfoque principal eran los integrantes del equipo y la convicción de mantener la cantidad de artefactos entregados al mínimo posible. En este enfoque, las metodologías estaban caracterizadas por colores que se asignaban de acuerdo con el tamaño del equipo (Grupo ISSI, 2003).

Dynamic Systems Development Method (DSDM): esta metodología fue presentada en 1994, por un consorcio de compañías y profesionales que buscaban un marco de trabajo para RAD (*Rapid Application Development*). La novedad en este caso correspondía a la elaboración de un estudio de viabilidad durante una etapa previa al desarrollo y un estudio de negocio, en el que se hacía el levantamiento de los requerimientos base del *software* (Paetsch, Eberlein, & Maurer, 2003).

Extreme Programming (XP): presentada por Kent Beck en 1999, en esta metodología se resalta la adaptabilidad, al considera que los cambios en las necesidades del negocio sobre la marcha son un aspecto natural, inevitable e incluso deseable en el desarrollo de proyectos de *software*. La metodología consiste en adaptar las mejores prácticas disponibles de acuerdo con las necesidades del proyecto que se vayan identificando en el camino.

SCRUM

SCRUM es un marco de trabajo desarrollado a principios de los años 90 por Jeff Sutherland y Ken Schwaber. Los resultados de los primeros trabajos con dicha metodología fueron presentados por primera vez en 1995 en la conferencia OOPSLA, en Austin, Texas (USA) (Scrum.Org, 2014). Este marco de trabajo está diseñado para que las personas solucionen los problemas de adaptación normales que se presentan en los proyectos de desarrollo de *software*, mientras entregan productos que generan valor de manera creativa y productiva.

El proceso Scrum no está pensado para entregar una forma de construir productos, sino que es más bien una forma más eficaz de gestionar los equipos de trabajo para que estos puedan mejorar constantemente mientras desarrollan el producto, apoyándose en otras técnicas o procesos que sean más específicos para desarrollar software.

Lo que Scrum propone, entonces, es una forma de organización de equipos por roles con responsabilidades específicas, ceremonias orientadas al mejoramiento continuo del equipo, algunos artefactos que ayudan a medir el rendimiento del equipo y unas pocas reglas de juego muy orientadas al manifiesto ágil (Schwaber & Sutherland, 2016).

Roles

En Scrum se tienen 3 roles principales: El *Product Owner* o Dueño de producto, el *Scrum Master*, y el *Team Member* o Miembro del equipo. A continuación se presenta una definición de cada uno de ellos, de acuerdo con Deemer, Benefield, Larman & Vodde (2012); ás adelante, cuando se analicen los resultados de la investigación realizada para este trabajo, se contrastarán algunas de esas responsabilidades con la realidad de las empresas que aplican Scrum en sus proyectos:

Dueño de Producto (PO): es la persona encargada de identificar las necesidades del negocio y convertirlas en una lista de posibles funcionalidades del producto, conocida como *Backlog de producto*. El PO es también el responsable de mantener priorizado dicho backlog y comunicar las necesidades y funcionalidades que lo componen a los demás miembros del equipo. Otra de las responsabilidades principales del PO es trabajar con los interesados del proyecto (*Stakeholders*) y revisar los resultados presentados por el equipo al final de cada iteración de trabajo o *Sprint*, para mantener controlado el valor generado y las variables importantes del proyecto (tiempo, costo y alcance).

Miembro del equipo: forma parte activa del equipo encargado de construir las funcionalidades identificadas por el Dueño de producto. Es importante aclarar que en Scrum el equipo debe tener todos los conocimientos y herramientas

para la construcción de las funcionalidades esperadas por el cliente, además de que se considera que el equipo es completamente autogestionado (equipos que no requieren una figura administrativa que tome las decisiones), lo que indica que cuenta con una gran autonomía; esto además conlleva una gran responsabilidad. La responsabilidad principal de los miembros del equipo es aprender constantemente, no solo para mejorar en sus fortalezas, sino también para apropiarse los conocimientos de sus compañeros y mejorar así las capacidades grupales del equipo.

Scrum Master: la principal labor del Scrum Master es acompañar al equipo, al Dueño de producto y la organización misma, para que aprendan a utilizar Scrum de la mejor manera, con el fin de que se convierta en una herramienta de generación de valor para el negocio. La labor del Scrum Master no está orientada tanto al control del equipo, sino más al entrenamiento y constante mejoramiento del mismo en el marco de trabajo, para que el equipo a su vez apropie las ceremonias y reglas de Scrum, en busca siempre de incrementar el rendimiento del equipo con cada iteración y la generación continua de valor al cliente.

Es importante agregar también que en Scrum no existe un rol que corresponda a un jefe de proyecto o jefe del equipo, dado que la principal responsabilidad de todas las labores recae sobre los roles mencionados anteriormente. El propósito es garantizar que el equipo sea autogestionado y pueda identificar por sí mismo las mejoras que debe implementar, con base en su propia experiencia.

La iteración (*Sprint*)

Esta es la iteración principal que tiene el equipo en un proyecto Scrum, es el espacio de tiempo concentrado en la implementación de las funcionalidades que ellos mismos se han comprometido a entregar desde el inicio del mismo. La metodología recomienda que tenga una duración máxima de un mes, pero el equipo tiene la potestad de definir cuál es la duración que mejor se acomoda a las necesidades. Durante el *sprint*, el equipo lleva a cabo un número de ceremonias y actividades necesarias para llevar el proceso a feliz término (Schwaber & Sutherland, 2016).

Las ceremonias en Scrum son una serie de sesiones de trabajo que la metodología recomienda, para reducir la cantidad de reuniones sin metas claras a las que asiste el equipo, de modo que se pueda garantizar la disponibilidad de sus miembros y construir *software*. A continuación se presentan las ceremonias prescritas por Scrum, con una breve descripción de cada una, de acuerdo con Schwaber & Sutherland (2016):

Planeación del *Sprint*: esta ceremonia tiene lugar al inicio del *Sprint*, con la participación de todo el equipo; en ella el equipo compromete el trabajo que se

va a realizar durante el *Sprint*, de acuerdo con su capacidad y las expectativas del cliente. El resultado de esta ceremonia se conoce como meta del *Sprint*.

Reunión diaria: es una reunión de 15 minutos que tiene lugar todos los días, en la que el equipo evalúa el avance en la construcción de la meta, con cada miembro del equipo respondiendo a tres preguntas recomendadas para permitir la visibilidad del avance de cada uno de los miembros y la identificación de posibles inconvenientes que deban ser atacados de inmediato, para que no se comprometa la meta del *Sprint*. Estas preguntas son:

- ¿Qué hice ayer?
- ¿Qué haré hoy?
- ¿Qué inconvenientes tengo para cumplir con la meta?

Revisión del *Sprint*: tiene lugar al final del *Sprint*; es en esta ceremonia cuando el equipo presenta los resultados al Dueño de producto. Además de esta presentación, el equipo también trabaja en la revisión del *Backlog* del producto, para identificar cuáles serían las funcionalidades a comprometer en el siguiente *Sprint*.

Retrospectiva del *Sprint*: Esta ceremonia es la oportunidad para que el equipo se revise a sí mismo y encuentre acciones de mejora, buscando resultados superiores en los siguientes *sprints*; tiene como meta clara la generación de valor al cliente. Se realiza después de la revisión del *Sprint* y la principal meta es identificar las cosas que se hicieron bien, para replicarlas, y aquellas cosas que no se hicieron bien, para definir un plan de acción que busque mejorar la forma de hacerlas.

ARTEFACTOS

Scrum propone también una serie de artefactos que ayudan a identificar elementos importantes dentro de un proceso de construcción, para hacerlos más palpables y medibles para el equipo. A continuación se presentan algunos de los artefactos principales, de acuerdo con Nazareno, Leone, & Gonnet (2013):

Producto: es la solución final que está siendo construida por el equipo a través de entregas incrementales en cada iteración; es la meta final del proyecto.

***Backlog* de producto:** contiene todas las funcionalidades identificadas por el Dueño de producto; juntas constituyen la necesidad completa del cliente. Debe estar siempre priorizado de acuerdo con las prioridades de negocio identificadas por el Dueño de producto en su trabajo y con los interesados del proyecto.

Elemento de *Backlog*: son las necesidades específicas de las cuales está compuesto el *Backlog* de producto. Cada uno de sus ítem corresponden a una

unidad funcional completa, que puede ser desarrollada y probada por el equipo. También son identificados como Historias de usuario, que a su vez pueden ser de tipo épica (agrupación de historias que representan un subproducto importante del proyecto; por ejemplo, un módulo completo de una aplicación).

Backlog del Sprint: todas las funcionalidades tomadas del *Backlog* de producto que el equipo se ha comprometido a desarrollar durante el tiempo que dura el *Sprint*.

Incremento de producto: corresponde al resultado final de la ejecución del *Sprint*. Es lo que se presenta en la ceremonia de revisión del *Sprint* y constituye la cantidad de funcionalidades nuevas que se agregan al producto final que se está construyendo para el cliente.

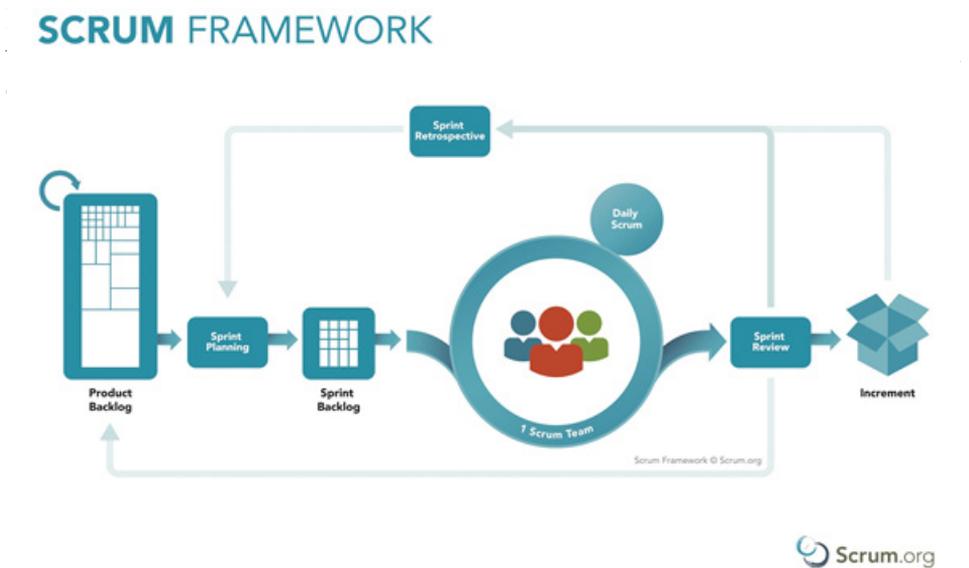


Figura 2. Resumen marco de trabajo Scrum (Scrum.org, 2016)

CONFLICTO

Se ha establecido que existen diferentes tipos de conflicto y, por ende, se han presentado a lo largo de la historia diferentes clasificaciones de ellos. Diversos autores han presentado clasificaciones del conflicto, para entenderlo y explicarlo de la mejor manera. Por ejemplo, Kunrtzberg y colaboradores (2005), citados por Kinnander (2011), establecieron tres clases principales de conflictos: los basados en el proceso de trabajo, los basados en las relaciones y el conflicto debido al descontento acerca de las tareas en el trabajo. Por su parte, Ohbuchib y colaboradores (2003) hicieron otra clasificación: conflicto de intereses, conflicto con el conocimiento y conflicto de valores.

No todos los conflictos son positivos o negativos en el mismo grado. Investigaciones realizadas por Martínez-Corts y sus colaboradores (2011) demostraron que, con respecto al conflicto por tareas, puede ser negativo o positivo y a veces no influye tanto en satisfacción por el trabajo. Concluyeron también que entre mejores sean las relaciones entre las personas que conforman la organización, mejor será el desempeño y la satisfacción en el trabajo, es decir, que tener unas buenas relaciones de trabajo es la mejor forma de lograr alto desempeño en las organizaciones. Para un buen trabajo en equipo se hace necesario que haya un ambiente de trabajo armónico, lo que permite y promueve la participación de los integrantes de los equipos, donde se aprovecha el desacuerdo para buscar una mejora en el desempeño.

De acuerdo con lo anterior, se espera entonces que los conflictos estén mucho más presentes en los equipos de trabajo que aplican metodologías de desarrollo ágil, como *Scrum*, ya que los principios ágiles han resaltado la importancia de la interacción y la comunicación entre personas en los proyectos. *Scrum* se ha convertido en la metodología más usada por su capacidad para incrementar la productividad a través del empoderamiento, un ambiente orientado al equipo y enfocándose en resultados y la transparencia (Crawford, Soto, Leon de la Barra, Crawford, & Olguin, 2014).

Una de las clasificaciones de conflictos más aceptada es aquella que se basa en la tricotomía conformada por conflictos basados en las tareas, conflictos basados en el proceso y conflictos basados en la relación. En el estudio de conflictos en equipos de trabajo, esta es una de las formas más comunes de clasificación de conflictos. A continuación se presenta una explicación de estos tipos de procesos, de acuerdo con Karn (2008):

- Conflictos de tareas: este tipo de conflictos está relacionado con una diferencia de puntos de vista o de entendimiento, ideas u opiniones. Generalmente no hay emociones negativas asociadas con este tipo de conflicto.
- Conflictos de proceso: este tipo de conflictos corresponde a desentendimientos en la forma de hacer las cosas dentro del equipo, o de las responsabilidades de cada uno de los individuos pertenecientes al equipo de trabajo.
- Conflictos de relación: este tipo de conflicto está más relacionado con las diferencias en actitudes, normas o valores, personalidad y creencias políticas. Se considera que es el más dañino de los tres tipos de conflicto mencionados, debido a que tiene una naturaleza emocional.

El conflicto en el desarrollo de *software*

El conflicto en los equipos de desarrollo de *software* no es necesariamente un factor negativo; de hecho, varias investigaciones han dado indicios de que juega un papel muy importante en el mejoramiento de equipos de trabajo. Una investigación llevada a cabo por Karn (2008) encontró que el 58% de los conflictos presentados en los equipos de su estudio fueron constructivos para el equipo.

Karn (2008) encontró, sin embargo, que no todos los tipos de conflicto son beneficiosos para el equipo. Sus resultados muestran que los conflictos de tarea en cantidades moderadas tienden a ser los más productivos para el equipo, hallazgo que concuerda con otras investigaciones que demuestran que los conflictos de tareas tienen un impacto positivo en la calidad de las ideas (West & Anderson, 1996), la presencia de más debates constructivos (Jehn, Northcraft, & Neale, 1999) y la aceptación de las decisiones del equipo (Turner & Pratkanis, 1994). Los conflictos de proceso y de relación presentan resultados un poco diferentes, pues en este caso, la mayoría de los procesos fueron destructivos para los equipos; el caso más preocupante fue el último.

Existe también evidencia de que en los equipos de alto rendimiento se presentaron más conflictos de proceso en etapas medianas y finales de los proyectos, que en los equipos con rendimientos bajos (Jehn & Mannix, 2001).

Los estudios mencionados anteriormente presentan evidencia de que los conflictos pueden ser beneficiosos para los equipos de trabajo, siempre y cuando sean administrados correctamente, ya sea por los integrantes del equipo o por aquellas personas encargadas de liderarlos. Este trabajo busca investigar la forma en que los equipos de trabajo que utilizan la metodología de trabajo *Scrum* suelen manejar los conflictos, y si el manejo que se da de los mismos hace que estos al final se traduzcan en beneficios para dichos equipos. Para ello, se realizaron entrevistas a integrantes de equipos de trabajo de una de las empresas de desarrollo de *software* más tradicionales de Colombia, buscando mostrar preferiblemente la perspectiva del *Scrum Master*.

Análisis de los resultados

Como ya se mencionó anteriormente, para la elaboración de este trabajo se realizaron entrevistas entre personas que trabajan en proyectos de desarrollo de *software* que aplican la metodología *Scrum*, principalmente en personas que ejercen el rol de *Scrum Master*, con al menos dos años de experiencia. En estas entrevistas se buscaba indagar principalmente por los aspectos que se presentan en la Tabla 1:

Tabla 1. Aspectos analizados y su descripción

Aspecto	Descripción
Responsabilidades del individuo	Percepción personal que tiene el individuo sobre cuáles son sus responsabilidades en los equipos de trabajo.
Funcionamiento del equipo	Funcionamiento de los equipos en aspectos específicos, que los autores consideran, pueden ser causantes de conflictos, con el fin de contrastar respuestas y notar si los individuos entrevistados encuentran alguna relación entre conflictos presentados y funcionamiento.
Administración de conflictos como equipo	Percepción que tienen los entrevistados sobre la capacidad de los equipos para identificar conflictos y tratarlos según el entendimiento que tienen de los mismos.
Diferenciación de conflictos	Capacidad de los individuos entrevistados para identificar y diferenciar los conflictos en los equipos a los que apoyan.
Disparadores de conflictos	Frecuencia con que los entrevistados consideran haber causado conflictos dentro de los equipos.
Responsabilidad en la resolución	Percepción del papel que los individuos consideran tener en el momento de solucionar un conflicto.
Papel del conflicto en el mejoramiento continuo	Busca encontrar la percepción que tienen las personas entrevistadas del papel que juegan los conflictos en el mejoramiento continuo de los equipos de trabajo.

Responsabilidades del individuo. Ante la pregunta: ¿Cuáles son sus responsabilidades dentro de los equipos de trabajo?, los entrevistados respondieron:

monitorear el avance (...) remover los bloqueos (...) asegurar que se lleven a cabo los planes de mejoramiento planteados por el equipo (...) dar visibilidad acerca del desempeño al equipo y a otros interesados (...) asegurar un crecimiento personal alineado con el crecimiento como equipo. Vigilar que se cumpla la metodología (...) adaptándola al equipo de forma que este logre un buen desempeño y ambiente laboral (...)

remover impedimentos (...) ayudar cuando el equipo necesita ponerse en contacto con otras áreas u otros equipos. Apoyar la mejora continua del equipo a nivel grupal e individual (...) definir los indicadores que considere necesarios para esta actividad (...) autoformación y aprendizaje continuo (...) apoyo en la facilitación de eventos (...) coaching al equipo y al PO (...) identificar cómo se siente el cliente con los incrementos de producto, con el equipo y con el marco de trabajo (...) seguimiento a prácticas técnicas (...) seguimiento al proceso (...) entender y adoptar valores, principios y prácticas ágiles.

Soy un facilitador que ayuda al grupo a alcanzar sus objetivos. Soportar y proteger el equipo, remover blockers, analizar riesgos, asegurar que no faltan recursos. Velar por el mejoramiento continuo (...) la correcta implementación de la metodología (...) soy un servidor del equipo por encima de cualquier cosa (...) identificar riesgos e impedimentos en el proyecto, para construir con el equipo planes de mitigación de los mismos. Todas aquellas que surjan en un sprint enfocadas a que el grupo aplique Scrum correctamente, mejoré continuamente, siga bien las prácticas de ingeniería (...) lograr los compromisos del sprint. Gestión de las ceremonias y solución de impedimentos (...) la principal tarea se da al motivar al equipo para que adquiera compromisos que lo llevan a la mejora constante tanto técnica, como personal, haciendo que cada vez sea menos necesario mi papel para alcanzar las metas planteadas. Hacer que el equipo con el que estoy trabajando sea mejor (...) (mejoramiento continuo).

Ser el principal canal de comunicación con el cliente (...) remover impedimentos que no puedan ser removidos por los miembros del equipo (...) identificar riesgos y proponer planes de mitigación (...) guiar al Product Owner (PO) (...) identificar y poner en marcha acciones que permitan mejorar el desempeño y normal trabajo del equipo y que permitan remover aspectos que nos estén impidiendo el normal funcionamiento (...) implementar aspectos de aproximaciones ágiles: XP, PR, estimaciones, CI, etc.

Al analizar estas respuestas, se encuentra que están alineadas a la definición del rol que da *Scrum* como metodología, por lo que no se evidencia en este aspecto un posible causante de conflictos dentro de los equipos de trabajo. La única variación real que parece existir entre las respuestas, depende principalmente de la madurez del equipo, puesto que en un equipo maduro, el papel del *Scrum Master* pierde mucho protagonismo, por el mismo hecho de que su principal labor es ser un guía metodológico para el equipo.

Funcionamiento del equipo. Ante la pregunta: ¿Quién define cómo se monitorea el progreso del proyecto y qué tanto se crean discusiones en el equipo

en el transcurso del proyecto relacionadas con este tema?, los entrevistados expresaron:

Usualmente el progreso del proyecto y discusiones de este tipo se dan entre el stakeholder y el Scrum master (...) en algunos casos es requerida la presencia de un BA para justificar el progreso y situaciones evidenciadas. Normalmente nosotros como proveedores (...) ofrecemos al cliente un conjunto de métricas para monitorear el desempeño (...) es el cliente el que define qué quiere ver (...) en los equipos en los que he participado generalmente el monitoreo no es una fuente de conflictos o discusiones en el equipo (...) el único tema que a veces es necesario aclarar es la definición de qué es un error y qué se espera tener descrito en las historias de usuario. Normalmente el cliente (...) define el control y monitoreo que debe hacerse sobre la ejecución del proyecto (...) es necesario tener un control estricto sobre los puntos de historia ejecutados en cada Sprint, la programación y fecha exacta de cada reléase (...) en otros escenarios no es tan relevante el control de la ejecución sino el cumplimiento de las metas o los objetivos en los plazos esperados.

Scrum ya nos define la principal forma de monitoreo del progreso del proyecto a través de la daily, donde se hace día a día validación del progreso del compromiso del sprint, lo cual nos permite ajustar el rumbo, de ser necesario, en el menor tiempo posible para alcanzar las metas. La gestión de proyectos está definida en los procesos de la compañía (...) no quiere decir que el grupo no pueda sugerir cambios/mejoras (...) si el grupo considera que algo del proceso no aplica para el proyecto, entonces se hacen los cambios correspondientes.

El equipo y el cliente. La definición del monitoreo en el avance del proyecto es una construcción colaborativa del equipo (...) de acuerdo a las necesidades del proyecto, del cliente, la metodología y las recomendaciones generales dadas por la empresa. Desde la planeación se definen (...) las variables del proyecto en términos de presupuesto, (...) ejecución funcional del backlog, (...) calidad, se definen en la planeación y luego se monitorean por los mecanismos que se hallan definido en el monitoreo.

Lo define el mismo marco de trabajo, Scrum, el grupo diariamente tiene daily para inspeccionar (...) avances de cada uno cómo va camino a lograr las metas y tomar correctivos, si no se está logrando, (...) hay discusión permanentes (...) cuando algunos temas requieren profundidad en el daily se acuerda quiénes revisan en profundidad. El Product Owner es el encargado de validar el progreso sobre el backlog que se tenga (...) las discusiones en este tema se dan porque el Product Owner siente que el avance no es el esperado y que el tiempo (...) no va a ser suficiente para

realizar las historias que generan mayor valor (...) deben ser manejadas con mucho cuidado para que no se generen conflictos que no aportan al crecimiento del equipo. El progreso lo monitorea el product owner y las discusiones se dan casi siempre durante las sesiones de planeación, de grooming y en las retrospectivas (...) durante las *dailies*, cuando realmente nos damos cuenta si está pasando algo que impida el progreso de alguien en particular.

En términos de funcionamiento de los equipos, más específicamente el monitoreo del avance del proyecto, sí se encuentran diferencias, ya que casi la mitad de ellos respondió que este monitoreo se realiza de acuerdo con lo requerido por el cliente, mientras que los demás dieron respuestas con las que se concluye que dicha definición la debe realizar el mismo equipo. En este caso, se considera que los primeros pueden sufrir de más conflictos relacionados con el funcionamiento del equipo que los segundos, ya que una de las principales características de *Scrum* es que el equipo es autogestionado y, como tal, debe ser autónomo al momento de definir la forma de monitorear el avance del proyecto. Sin embargo, en las respuestas a las entrevistas no se identifica una conexión directa entre los conflictos identificados por las personas que respondieron a las preguntas y la definición del aspecto de monitoreo.

Administración de conflictos como equipo.

Ante las preguntas: ¿Qué actitud toma el equipo cuando alguien no está rindiendo al ritmo del equipo? y ¿qué actitud toma usted como cuando sucede esto?, los entrevistados reconocieron:

La idea nunca es buscar un culpable (...) si la situación es de actitud (...) la idea es encontrar el inconveniente al menor plazo posible (...) las situaciones deben ser expuestas en las reuniones diarias y solucionar lo que pase entre todo el equipo apoyándose en el Scrum-Master. He evidenciado que por la forma de trabajo el equipo se da cuenta del avance de cada miembro y cuando uno está por debajo de la media, se siente un malestar en el equipo dado que el trabajo que no desarrolla este miembro termina siendo desarrollado por el miembro más eficiente, creando en algunos casos una sobrecarga y malestar en el equipo.

Normalmente se le van asignando menos tareas hasta que tenga el conocimiento de acuerdo al nivel esperado, si luego de buscar opciones para mejorar su desempeño, aún no lo logra, se habla con la persona acerca de sus habilidades y capacidades para el proyecto (...) los equipos siempre tratan de ayudar al rezagado. Normalmente el equipo lo protege por temor a generar una confrontación directa del tema, no se menciona el bajo rendimiento en las retrospectivas sino que se comenta entre los demás integrantes de forma más discreta. Normalmente, en los equipos maduros, cuando alguien del equipo no da lo que el equipo espera, se lo expresa abiertamente a la persona y entre todos

buscan como mejorar la situación con ayuda del Scrum Master (...) si no se logra la mejora tras intentarlo todo, el mismo equipo pide que la persona salga (...) los equipos que apenas inician (...) suelen hacer comentarios de pasillos o comentarios anónimos en las retrospectivas.

Si el rendimiento de alguien está por debajo de las expectativas, le ofrecemos la oportunidad de mejorar. El equipo busca la manera de mejorar el desempeño, elegir historias de usuario mas adecuados para la persona, asignar un coach. La actitud ante esta situación generalmente depende mucho de la madurez del equipo, cuando los equipos no son muy maduros, estas situaciones tienden a ser ocultadas por los demás miembros del equipo ante los responsables del proyecto y el cliente, pero tienden a ser discutidas en situaciones por fuera de lo laboral y a espaldas del implicado. En los equipos maduros, por otra parte, los demás miembros del equipo identifican el tema y lo proponen como algo que se debe solucionar, para dar la oportunidad a la persona de mejorar, con el apoyo del equipo o de la empresa, según la situación.

Cuando esta situación no funciona, se propone que la persona sea asignada a otro proyecto, en el que tal vez le pueda ir mejor. Por la experiencia (...) hemos tenido que en el primer punto es en los *dailies* que uno identifica en dónde está la falencia, como las metas son en equipo, entonces el equipo trata de resolver esa falencia, sin embargo, se retroalimenta directamente en la retrospectiva. Cuando se detectan esos comportamientos se pregunta por las razones que hace que no esté rindiendo para tomar acciones, aunque aún falta madurez dentro del grupo para este tema, algunos detectan que algo pasa, pero no lo dicen (...) alguien no entregó sus compromisos y tampoco apareció al sprint review, el grupo decidió que esa persona no continuaba en el proyecto, para otros casos hemos visto bajo rendimiento y entramos a entender por qué para ayudar a mejorar.

Normalmente, esto se toma como un impedimento interno y es abordado como tal (...) durante las retrospectivas (...) si la persona no encaja bien dentro del equipo (...) es mejor validar si se puede realizar una reestructuración del equipo y conseguir una persona diferente, que se acople al estilo de trabajo y ritmo del equipo. La actitud es encarar la dificultad reuniéndose con la persona que no esté rindiendo, decirle lo que está pasando, exponerle alternativas para mejorar (...) el peor de los casos es retirar a la persona del equipo.

Las respuestas encontradas en este aspecto dan a entender que la capacidad de administración de los conflictos de un equipo está ligada directamente con su madurez del mismo, mientras que los equipos inmaduros tratan de ocultar la situación o no discutirla, los equipos maduros tienden a hacer un manejo más efectivo de situaciones conflictivas. Esta correlación encontrada apoya la tesis, presentada en este trabajo, de que existe una conexión directa entre la

ocurrencia y administración de conflictos y el mejoramiento de los equipos: mejores equipos administran los conflictos de maneras más efectivas.

Diferenciación de conflictos.

Ante la pregunta: ¿Cómo diferencia usted un conflicto constructivo de uno destructivo y como los maneja?, los entrevistados mencionaron:

Un Conflicto constructivo ayuda a un equipo a retarse, a mejorar y aprender del mismo, este tipo de situaciones se exponen y se resaltan (...), uno destructivo se trata más en confianza con los implicados y se soluciona según el tema en compañía del equipo y de la manera más rápida posible para que no crezca y no genere incomodidad en el equipo, su armonía, velocidad y compromiso.

Uno constructivo usualmente propende por el bienestar del equipo y del producto, mientras que en uno destructivo puede verse como bienestar común o conflictos entre personalidades de los miembros. Cuando la persona que señala al otro lo hace sin dar opciones de cómo mejorar, usando palabras ofensivas y no se ofrece para ayudar al compañero, es una crítica destructiva. Esta está muy difícil (...), para diferenciarlos creo que hay que ser buen observador de la forma en cómo se está recibiendo la información por los implicados y sus reacciones (...) intento que las discusiones (...) se salgan de lo personal y se tomen como temas de equipo, resaltando que la idea es mejorar y que las responsabilidades individuales son también responsabilidades de equipo. Es difícil (...) los conflictos destructivos son los que se centran en la personalidad de alguien o en el ataque a la persona en sí, estos tipos de conflictos se deben cortar reorientándolos a temas de equipo (...) mientras que los conflictos que son de trabajo en equipo si se manejan correctamente son constructivos y se resuelven con varias técnicas que implican la conversación abierta y creativa.

En lugar de conflicto yo hablaría de obstáculo (...) algo que impide que el grupo logre sus objetivos (...) los obstáculos más difíciles de remover son aquellos que involucran a personas, al ser (...) todo obstáculo requiere atención inmediata (...), que una persona no comunique claramente sus ideas es un obstáculo (...) podría verse como un conflicto destructivo. Si todas las partes en el conflicto escuchan los razones de la otra, tratando de entender la otra parte, si es un discusión respetuosa, es un buen conflicto (...) si no, debe manejarse tratando a hablar con las partes separado, y después juntos.

Un conflicto constructivo es aquel que bien manejado puede dejar buenos resultados para el equipo en el que se presenta, mi percepción personal

es que todos los conflictos tienen el potencial para ser constructivos si se administran correctamente, por lo que creo que lo importante no es tanto diferenciarlos, sino más bien, tratarlos a tiempo. Yo pienso que la construcción o la destrucción va frente al proyecto porque una parte de la buena voluntad de las personas, entonces si se presenta un conflicto destructivo (...) se hace el debido proceso y con la buena retroalimentación constructiva hacia la persona.

Para mí los destructivos son cuando se enfocan en la persona y no en los problemas del proceso, para manejarlos siempre me pregunto, por qué X está afectado por la situación, X, puedo ser yo, el grupo, otro, la respuesta a esa pregunta siempre me da luces sobre cómo proceder. Considero que los conflictos que se llevan a niveles personales (...) y convierten a los equipos de trabajo en personas que solo los une un nombre y que no miran juntos al objetivo que deben lograr (...) los conflictos sobre la implementación de una solución, son constructivos y deben ser alentados dentro de los equipos de trabajo, ya que generan que las soluciones que salgan de dichos conflictos sean acordadas por varios miembros del equipo, haciendo que haya un compromiso mayor a la hora de ejecutarlas. Al reunir la información (...) me doy cuenta del tipo de conflicto, el manejo es igual: reunirnos, ser totalmente abierto a dar y recibir feedback y muy transparente con lo que digo, nunca oculto ni me guardo nada.

Las respuestas a esta pregunta evidencian que, en la mayoría de los casos, los entrevistados saben identificar los conflictos a tiempo y diferenciar aquellos que son constructivos de los destructivos, lo que les brinda una herramienta importante al momento de llevar al equipo a afrontar cada uno de los conflictos que se presentan. Estas personas también comprenden la importancia de gestionar los conflictos correctamente y a tiempo; se buscan como alternativas principales la comunicación, transparencia y honestidad entre los miembros del equipo.

En los casos en los que los *Scrum Master* dan a entender que no saben diferenciar los conflictos constructivos de los destructivos, también está presente un entendimiento erróneo del conflicto, llegando en algunos casos a percibirlo como algo malo para los equipos. Se encuentra, sin embargo, que estas personas tienen habilidades básicas de solución de conflictos y que son capaces de manejarlos correctamente una vez que ya se han manifestado.

Disparadores de conflictos.

Ante la pregunta: ¿Ha sido usted causante de algún conflicto?, los entrevistados respondieron así:

En parte sí, me considero amante de la calidad en el desarrollo (...) pero intentar aplicar esto es difícil y en un principio se dio el tema y fue beneficioso porque al final de los proyectos fue de gran ayuda y ahorro de tiempo en estas prácticas. No, aunque si me he visto afectado por conflictos. En un proyecto se generó una disparidad de criterios con el cliente, con los equipos no ha habido conflictos recordables. Sí, de forma regular; cuestionando las prácticas técnicas, arquitecturas, soluciones, rendimiento e interés por el aprendizaje.

Sí (...) en ocasiones con toda la intensidad, especialmente cuando el equipo está en un estado muerto donde no ve el potencial que tiene, entonces es necesario sacarlos de su estado de confort para que mejoren y eso implica llevarlos a un estado de caos para que luego puedan encontrar nuevas mejoras. Sí, he tenido problemas de comunicación por no expresar correctamente mis opiniones (...) y estoy en proceso de mejora para expresar sin carga sentimental lo que quiero decir. Nunca es solo un lado que causa un conflicto.

Sí, es una parte normal y común en el trabajo en equipo. Sí (...) yo también estuve en un proceso de construcción y crecimiento (...) muchas veces entra uno a hacer juicios por que una persona no está rindiendo, porque el equipo no logro los objetivos entonces hacer juicios de valor. Si, en algunas ocasiones, por no informar temas sensibles al grupo a tiempo. Sí, las interacciones humanas son sumamente complicadas y en algunas situaciones es difícil dejar que el equipo resuelva todos los impedimentos, haciendo necesaria la intervención directa dentro del equipo, causando conflictos que deben ser manejados correctamente para que no escalen más allá de lo laboral.

Muchas personas entrevistadas ven el conflicto como algo normal e inherente al trabajo en equipo; aceptan haber sido parte detonante dentro de sus equipos de trabajo, y los relacionan como una herramienta para causar movimiento en los equipos o como parte de su crecimiento personal.

Responsabilidad en la resolución.

Ante la pregunta: ¿Considera usted que es trabajo del Scrum Master resolver los conflictos?, las respuestas preguntas obtenidas fueron:

Resolverlos no, pero si servir de facilitador y mediar si se puede, así como proponer espacios, ambientes y formas de interacción para que esto no se dé (...) mediar en las situaciones que puedan retrasar al equipo o generar malos ambientes, lo que le da la obligación de intervenir cuando el equipo

solo no sea capaz de solucionar el conflicto. El Scrum Master (...) debe recordar la cultura de trabajo y hacer respetar los roles de cada uno, el ayudar a resolver conflictos en el equipo es parte de sus responsabilidades (...) es quien con su ejemplo debe intentar mediar entre las partes de forma sana e imparcial.

Si, es responsabilidad del Scrum Master mantener el equipo cohesionado. Entendiendo la palabra conflicto por cualquier factor que afecte al equipo y su desempeño, la respuesta es sí. Sí, y lo puede resolver directa o indirectamente según sea el caso, ya que hay conflictos donde la mejor estrategia es propiciar los espacios y las discusiones correctas para que el mismo equipo lo resuelva. El trabajo del SM es ayudar al grupo a identificar (...) y orientarlo en cómo resolver el conflicto (...) orientar al grupo haciendo preguntas y proponiendo acciones para resolver el conflicto (...) es el grupo quien tiene la decisión de tomar las acciones de mejora. A facilitar la resolución, sí.

No, si bien es un rol importante en la resolución de los conflictos, la responsabilidad de la resolución del conflicto recae sobre los principales implicados, con la guía y el apoyo de todos los demás integrantes del equipo, incluido el Scrum master. Sí, no tanto por el rol sino por ser miembro de ese equipo, ósea lo positivo del equipo es del equipo pero las cosas que les están faltando al proyecto son del Scrum Master en primera cabeza, porque una de las funciones principales del scrum master es aceptar esa máquina.

Creo que es trabajo (...) apoyar para que se resuelvan lo más rápido posible, algunas personas pueden manejarlos, otras requieren apoyo, normalmente porque necesitan un punto de vista diferente. No necesariamente, el Scrum Master debe ser una persona que sirva en la mediación de esos conflictos y no un juez (...) debe velar porque los conflictos sean resueltos por los miembros del equipo y no se llegue a una solución impuesta que degrade la unión del equipo y genere que haya una pérdida de compromiso hacia los objetivos que se han propuesto.

No siempre (...) se deben resolver en equipo, el Scrum Master debe propiciar espacios para resolverlos, enseñar a las personas del equipo a dar y recibir feedback (...) generar confianza dentro del equipo para que cualquiera levante la mano.

Existe consenso en identificar al *Scrum Master* como un mediador que puede apoyar a los equipos a solucionar los conflictos que se presentan, más poco se le identifica como el responsable de solucionarlos, lo cual está de acuerdo con lo que dicta la metodología de trabajo SCRUM y demuestra también conocimiento en administración de conflictos entre los entrevistados.

Papel del conflicto en el mejoramiento continuo. Ante la pregunta: ¿Cree usted que el mejoramiento continuo de los equipos de trabajo está ligado directamente al surgimiento y la resolución de conflictos?, los entrevistados respondieron:

Sí, (...) un equipo motivado es un equipo eficaz y veloz es un equipo que rinde (...) la no disolución de conflictos pueden crear problemas de comunicación que es uno de los pilares del éxito de un equipo auto gestionado y ágil. Conflicto y la forma de resolverlos es una clara métrica de maduración de un equipo. Pienso que los equipos mejoran en la medida que cada uno de sus individuos esté en mejoramiento de capacidades interpersonales y técnicas (...) las habilidades blandas juegan un rol definitivo para que las personas se sientan parte de un colectivo.

Los conflictos generan mejoramiento continuo porque el equipo aprende, afianza o renuncia, creo que eso mejora los equipos. Sí, basado en lo que he dicho antes. Sí. Conflictos no son solo diferencias entre personas (...) es no estar entrenados en la tecnología (...) no tener los recursos (...) falta de conocimiento en el negocio, etc (...) que el grupo logre identificar y resolver estos conflictos de manera efectiva hace que sea cada vez mejor. No. Sí, un equipo que aprende a solucionar todo tipo de conflictos es un equipo que está enrutado a solucionar cualquier situación que se presente en su camino.

Depende netamente del equipo (...) siguen sucediendo cosas después de un año (...) no estamos en ese nivel que uno considere el equipo de alto desempeño y siguen ocurriendo conflictos. En parte porque el conflicto si se resuelve, normalmente genera aprendizaje, pero en mi concepto también está relacionado con el deseo de hacer cada vez mejor cada cosa. No necesariamente, considero que los conflictos son una parte importante de los equipos de trabajo y que deben ser correctamente enfocados. Sin embargo, pienso que la generación de compromisos y la unión de los integrantes del equipo son más importantes a la hora de hacer crecer a los equipos. No, el mejoramiento continuo se da cuando juiciosamente se monitorea y evalúa el trabajo realizado por el equipo, se toman acciones que ayuden a mejorar y repetir dicho ciclo.

En términos generales, hay consenso en la percepción de que el conflicto está relacionado con el mejoramiento continuo y la maduración de los equipos de trabajo, puesto que las personas consideran que cuando el equipo es capaz de sortear los obstáculos que los conflictos le pueden presentar, aprende de la experiencia y se convierte en un mejor equipo. Se resalta el hecho de que los *Scrum Master* consideran que para que el fenómeno anterior se dé es importante la buena gestión de los conflictos a tiempo, para evitar que estos dañen la

relación entre los integrantes del equipo o afecten los resultados de proyectos a puntos en los que ya no se pueden corregir.

Vale la pena aclarar que aquellas personas que dicen que el mejoramiento de los equipos no está ligado directamente al conflicto, pertenecen al grupo de las personas que no muestran capacidades de identificación o diferenciación de conflictos.

CONCLUSIONES

Las personas entienden el hecho de que los conflictos son una parte inherente al trabajo en equipo y comprenden que su identificación y tratamiento a tiempo, no solo evitan sus posibles consecuencias negativas, sino que también ayudan a la maduración de los equipos de trabajo.

Existe una marcada diferencia en la forma como los equipos maduros (entendiendo “equipo maduro” como uno que ya ha adoptado la metodología de trabajo y tiene una dinámica de grupo con suficiente confianza para discutir las situaciones que se requieran) y los no maduros gestionan la situación conflictiva. Los primeros entienden que estas situaciones son normales, y que es importante trabajar en la solución como equipo de manera constructiva, mientras que los segundos parecen tratar de ocultar las situaciones, lo que lleva a que eventualmente la situación se convierta en un problema.

Los equipos de trabajo que usan la metodología SCRUM comprenden lo que significa que sean autogestionados y ven en el *Scrum Master* una persona que los puede guiar y ayudarlos, pero saben también que la responsabilidad de la resolución de todos los inconvenientes que puedan presentarse durante la ejecución de un proyecto son de la responsabilidad del equipo como un todo, lo que demuestra un buen conocimiento de la metodología y puede evitar la presentación de conflictos por este ítem.

Este trabajo presenta evidencia de la importancia de la correcta identificación, diferenciación y administración de los conflictos dentro del equipo de trabajo que sigue la metodología de trabajo SCRUM. Las entrevistas realizadas apoyan la conjetura referida a que los conflictos dejan en los equipos un aprendizaje suficiente para llegar a un punto de maduración en el que la gestión de los conflictos ayuda a los equipos a mantener el mejoramiento continuo.

De acuerdo con la información recolectada en las entrevistas, los tipos de conflicto más frecuentes en los equipos de trabajo con proyectos de desarrollo de *software* bajo la metodología SCRUM son los conflictos de proceso, debido a que a las personas les cuesta un poco adaptarse a la metodología de trabajo de cada nuevo equipo. Los tipos de conflictos que se presentan en menor

frecuencia son los conflictos de relación, sin embargo, estos últimos son los más difíciles de administrar por su naturaleza.

La relevancia de este trabajo radica en lo importante que es SCRUM en el tiempo de desarrollo del mismo, puesto que de acuerdo con los estudios presentados, es la más ampliamente usada entre las metodologías de desarrollo ágil.

Las situaciones conflictivas son inherentes al ser humano y siempre estarán presentes en menor o mayor medida en los equipos de trabajo; la correcta gestión de los conflictos juega entonces un papel fundamental dentro de la administración de este tipo de equipos. Por lo anterior, se considera fundamental que las personas que piensan utilizar esta metodología comprendan que los conflictos son una parte fundamental de la implementación y que, para evitar rompimientos en las relaciones en los equipos y garantizar el éxito de los proyectos, los equipos autogestionados que trabajan con SCRUM deben aprender a manejar correctamente los conflictos.

REFERENCIAS

Agile Alliance (2001a). *History: The Agile Manifesto*. Obtenido de <http://agilemanifesto.org/history.html>

Agile Alliance (2001b). *Principios del Manifiesto Ágil*. Obtenido de <http://agilemanifesto.org/iso/es/principles.html>

Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M. y otros (2001). *Manifiesto por el Desarrollo Ágil de Software*. Obtenido de <http://agilemanifesto.org/iso/es/manifesto.html>

Bell, T. E. & Thayer, T. A. (13 de 10 de 1976). Software requirements: Are they really a problem? *ICSE '76 Proceedings of the 2nd international conference on Software engineering*, 61-68.

Boehm, B. W. (1988). A Spiral Model of Software Development and Enhancement. *Computer*, 21 (5), 61-72.

Beynon-Davies, P., Carne, C., Mackay, H., & Tudhope, D. (1999). Rapid application development (RAD): an empirical review. *European Journal of Information Systems*, 8 (3), 211-223.

Crock, P. (2008). Best Practices for Software Projects - JAD Sessions.

Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2012). *Scrum Primer: Una introducción básica a la teoría y práctica de Scrum*. Recuperado el 25 de 10 de 2016, de Scrum Primer: <http://scrumprimer.org/scrumprimer20.pdf>

Fowler, M. & Highsmith, J. (2011). The Agile Manifesto. *Software Development Magazine*, 9, 28-35.

Gacitúa Bustos, R. A. (2003). Métodos de desarrollo de software: El desafío pendiente de la estandarización. *Theoria*, 12, 23-42.

González, D. y López, F. (2012). Poder y conflicto en la organización: Una aproximación desde el tratamiento funcionalista de los libros de texto. En: S. Arcand, R. Muñoz, J. Facal, J. Dupuis u F. E. EAFIT (Ed.), *Sociología de la empresa. Del marco histórico a las dinámicas internas*. Bogota: Siglo del Hombre.

Grupo ISSI (2003). Metodologías Ágiles en el Desarrollo de Software. *VIII Jornadas de Ingeniería del Software y Bases de Datos/*

Jian, Z. (2005). Why IT Projects Fail. *Computerworld*, 39 (6), 31-31

Kinnander, M. (2011). *Conflict Management How to manage functional conflicts within project teams*. Unpublished dissertation, Nørthumbria University, Sweden

Martínez-Corts, I., Boz, M., Medina, F. J., Benítez, M. y Munduate, L. (2011). Afrontamiento de conflicto interpersonal en el trabajo en pequeñas empresas: el papel moderador del apoyo del supervisor y del compañero. *Revista de Psicología del Trabajo y de las Organizaciones*, vol.27 no.2 Madrid ago. 2011. Pp.117-129.

Standish Group. (2015). *2015 CHAOS Report*. Recuperado de <https://www.standishgroup.com/store/services/chaos-report-2015-blue-pm2go-membership.html>.

Royce, W. W. (1970). Managing the development of large software systems. En: *Proceedings of IEEE WESCON*, 26 (8), 328-338.

Rational Unified Process. (2001). Best practices for software development teams. *A Rational Software Corporation White Paper*. Recuperado de: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

VersionOne. (2016). *10th Annual State of Agile Report*. Obtenido de <https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf>

Paetsch, F., Eberlein, A. & Maurer, F. (2003). Requirements Engineering and Agile Software Development. *Proceedings of Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. Linz, Austria.

Riehle, D. (2001). A comparison of the value systems of adaptive software development and extreme programming: how methodologies may learn from each other. En: G. Succi & M. Marchesi, *Extreme programming examined*. Boston, MA, USA: Addison-Wesley Longman Publishing Co.

Scrum.Org. (2014). *The Scrum Guide*. Recuperado de <https://www.scrumalliance.org/why-scrum/scrum-guide>

Schwaber, K., & Sutherland, J. (2016). *The Definitive Guide to Scrum: The Rules of the Game*. Berna, Suiza: Scrum.org.

Nazareno, R., Leone, H. y Gonnet, S. (2013). Trazabilidad de Procesos Ágiles: un Modelo para la Trazabilidad de Procesos Scrum. *XVIII Congreso Argentino de Ciencias de la Computación*. Universidad Nacional de La Rioja, La Rioja, Argentina 2013

Scrum.org. (2016). *Scrum Framework*. Recuperado de https://www.scrum.org/Portals/0/Documents/Collateral/ScrumFramework_17x11.pdf

Michael, J., Maurer, F. & Tessem, B. (2005). Human and social factors of software engineering: workshop summary. *ACM SIGSOFT Software Engineering Notes*, 30 (4), 1-6.

Crawford, B., Soto, R., Leon de la Barra, C., Crawford, K., & Olguin, E. (2014). Agile software teams can use conflict to create a better product. *Communications in Computer and Information Science*, 434, 24-29.

Karn, J. (2008). An ethnographic study of conflict in software engineering teams. *Journal of Information, Information Technology & Organizations*, 3, 106-133.

Jehn, K. & Mannix, E. (2001). The dynamic nature of conflict: a longitudinal study of intragroup conflict and group performance. *Academy Of Management Journal*, 44 (2), 238-251.

West, M. & Anderson, N. (1996). Innovation in top management teams. *Journal Of Applied Psychology*, 81(6), 680-694.

Jehn, K., Northcraft, G. & Neale, M. (1999). Why Differences Make a Difference: A Field Study of Diversity, Conflict, and Performance in Workgroups. *Administrative Science Quarterly*, 44(4), 741-763.

Turner, M. & Pratkanis, A. (1994). Social identity maintenance prescriptions for preventing groupthink reducing identity protection and enhancing intellectual conflict. *International Journal Of Conflict Management*, 5(3), 254-270.