CONFLICTO Y DESEMPEÑO EN EQUIPOS ÁGILES EN UNA DE LAS EMPRESAS DE TECNOLOGÍA MÁS IMPORTANTE DE COLOMBIA

JUAN FERNANDO VALENCIA TORO

UNIVERSIDAD EAFIT

ESCUELA DE ADMINISTRACIÓN

MAESTRÍA EN GERENCIA DE PROYECTO

MEDELLÍN

2017

CONFLICTO Y DESEMPEÑO EN EQUIPOS ÁGILES EN UNA DE LAS EMPRESAS DE TECNOLOGÍA MÁS IMPORTANTE DE COLOMBIA

JUAN FERNANDO VALENCIA TORO

TRABAJO DE GRADO para optar por el título de MÁSTER EN GERENCIA DE PROYECTOS

ASESOR TEMÁTICO: FRANCISCO LÓPEZ GALLEGO

UNIVERSIDAD EAFIT

ESCUELA DE ADMINISTRACIÓN

MAESTRÍA EN GERENCIA DE PROYECTO

MEDELLÍN

2017

Resumen

En este artículo se explora la posibilidad de la existencia de una relación entre los estilos de solución de conflictos y el desempeño en equipos de desarrollo de *software* que trabajan usando la metodología Scrum. El estudio está compuesto por una detallada investigación bibliográfica sobre el desarrollo de *software*, desde el punto de vista de la ingeniería y la gerencia de proyectos. De la misma forma, se hace un análisis sobre el conflicto, las posibles razones por las que se genera y los estilos que se pueden adoptar para afrontarlo, así como su relación de estos con el desempeño y el rendimiento en los equipos de trabajo. Finalmente, se revelan los resultados de las respuestas de 30 personas que tomaron el test de Thomas-Kilman, para establecer su estilo de solución de conflictos predominante. Esto, en conjunto con la evaluación de un panel de expertos que tienen información, desde el punto de vista gerencial, sobre el desempeño de los cinco equipos de trabajo a los que pertenecen las personas entrevistadas. Este estudio fue realizado en una de las empresas de tecnología más importantes de Colombia, que es pionera en la implementación de metodologías ágiles en el país; además, cuenta con un poco más de 500 empleados distribuidos en cinco ciudades de Colombia y tiene sedes en México y Estados Unidos.

***Palabras clave*:** Scrum, metodologías ágiles, conflicto, desarrollo de *software*

Abstract

This article deeply explores the existence of a possible relationship between conflict handling skills and performance in software development teams that work using Scrum methodology. This study is made of a detail bibliographical research regarding software development, looked from an engineering and project management point of view. In the same way, there is an analysis of conflict, with its possible root causes and the different modes to face it, as well the relationship between them and the team performance. Finally, the results of the answers of 30 people that took the Thomas-Kilman test are revealed, and it is used to determine their prevailing conflict handling mode. All of that is shown along a management assessment from an expert panel group, regarding each one of the five teams that the interviewed people work on. This study took place in one of the most important technology companies in Colombia, which is a leader company in implementing agile methodologies in software development projects in this country; moreover, it has more than 500 employees distributed in five Colombian cities and has offices in Mexico and the United States.

***Keywords*:** Scrum, agile methodologies, conflict, software development

1. Introducción

El presente trabajo demuestra la posible existencia de una relación entre los diferentes estilos de solución de conflictos, con el desempeño y el rendimiento de los equipos de trabajo que utilizan metodologías ágiles, más específicamente Scrum. El análisis se realizó en una de las empresas de tecnología más importantes de Colombia, la cual tiene más de 500 empleados distribuidos en diferentes ciudades del país y en sus dos sedes en el exterior, una en México y otra en Estados Unidos.

En el inicio del trabajo, se puede encontrar una investigación literaria sobre el desarrollo de *software*, visto desde un punto de gerencial y de ingeniera, en el que se recorre un poco la historia de esta ciencia joven, que abarca desde su nacimiento hasta su llegada a las metodologías ágiles, para adentrarse en la metodología Scrum como marco de trabajo para muchas empresas a nivel mundial. A continuación se presentan algunas definiciones de conflicto, un análisis sobre sus posibles causas y la relación que este tiene con los diferentes estilos de solución de conflictos detallados por Thomas-Kilman.

Después del marco teórico se encuentra la estructura de una serie de entrevistas y evaluaciones, que se llevaron a cabo en el presente trabajo, con el fin de establecer la relación entre el perfil del estilo de solución de conflictos y el desempeño en cinco equipos de desarrollo de *software* en una de las empresas más importante de tecnología colombiana. Para esto se seleccionaron 30 personas que en la actualidad integran cinco equipos de trabajo dentro de la compañía. Estas personas respondieron el test de Thomas-Kilman, con el fin de identificar cuáles son los estilos predominantes en sus respectivos equipos de desarrollo de *software*.

Luego se llevó a cabo un panel, que contó con la participación de un grupo de expertos, los cuales conocían el desempeño y el rendimiento de cada equipo desde un punto de vista gerencial, para realizar una calificación de los grupos seleccionados. Este test sirvió para identificar cuál es el rendimiento y el desempeño de cada uno de los cinco equipos involucrados en este estudio. Finalmente, se estable un tipo de relación entre los estilos de solución de conflictos predominantes de cada equipo con respecto a su desempeño y su rendimiento.

2. *Software* como ingeniería

El término *software* fue utilizado por primera vez por John W. Tukey en 1957 (Leonhardt, 2000); sin embargo, el *software* como ingeniería no sería conocido hasta 1969, un año después de una conferencia que realizó la NATO con el fin de discutir sobre los programas, complejos y llenos de errores, que existían hasta ese momento (Wirth, 2008). A partir de ese entonces, se empezaron a realizar muchos esfuerzos y a definir diferentes metodologías para profundizar más en el tema y mejorar la calidad de los programas y de los sistemas desarrollados en todo el mundo (Randell, 1996).

2.1 Metodología en cascada (*Waterfall*)

Como resultado de esos esfuerzos, nació la metodología en cascada, o *Waterfall Model*, propuesta en 1970 por Royce (1970) para mejorar los resultados del desarrollo de *software*. Esta metodología consiste en una serie de etapas o fases que producen unas salidas, comúnmente conocidas como entregables (Cnfolio, s. f.). Adicionalmente, como el mismo nombre de la metodología lo indica, cada etapa necesita como entrada la información obtenida en la fase anterior para poder completar sus objetivos específicos (Liu & Horowitz, 1989). El modelo en cascada, cuyo esquema se aprecia en la ilustración 1, está compuesto por las fases de requisitos del sistema, análisis del sistema, diseño de la solución, implementación, pruebas y operación (o mantenimiento) (Lott, 1997).

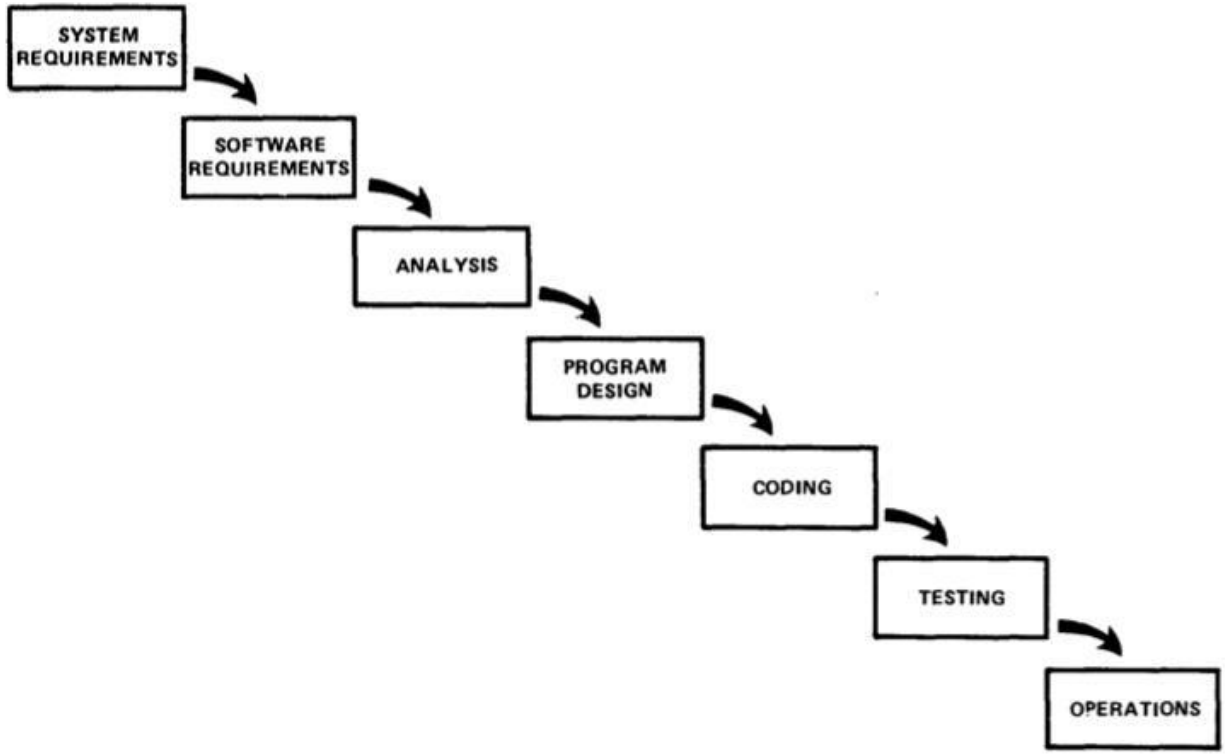

Ilustración 1. Metodología en cascada (Cnfolio, s .f.; Royce, 1970).

Aunque suene redundante, en aquel momento cada vez que un proyecto iniciaba, arrancaba con la primera fase de la metodología: los requisitos del sistema. En esta etapa del proyecto, las empresas invertían grandes cantidades de tiempo y de recursos para dar a entender el negocio y la idea que se iba a desarrollar, y luego los analistas de negocio describían detalladamente el comportamiento que iba a tener el *software* que se iba a construir. Esos detalles explicaban de forma minuciosa la forma en la que el usuario y el sistema iban a interactuar. Estas definiciones se hacían a través de requisitos funcionales y no funcionales (Bassil, 2012).

Estos requisitos funcionales y no funcionales, en las etapas de análisis y diseño de la solución eran recibidos por un equipo de arquitectos de *software* y de diseñadores. Esta fase consistía en revisar las extensas especificaciones del sistema, con el fin de entenderlas, de proponer una solución técnica para el problema en cuestión y de definir unas aproximaciones visuales para el sistema por construir. En estas dos etapas se trataba de encontrar la mejor solución posible, con base en las tecnologías existentes en la época y en las restricciones que pudiera tener la empresa que necesitaba el proyecto (Cusumano & Smith, 1995).

Luego de definirse la solución y de tener una base de diseños visuales, el equipo de desarrollo transformaba los requisitos y las especificaciones del sistema en un *software* concreto (con su respectiva base de datos y pantallas), a través de la programación. En esta fase es donde se escribe el código fuente del sistema, utilizando alguno de los lenguajes de programación existentes. En otras palabras, este es el proceso de convertir largos documentos, llenos de especificaciones y planos del sistema, en *software* funcional (Bassil, 2012).

Sin embargo, a pesar de que se contaba ya con diversos *software* funcionales que podían ser utilizados por usuarios reales, el sistema todavía no se le podía entregar al cliente. Esto era porque hacía falta una de las fases más importantes de toda la metodología: la fase de pruebas. Por la misma naturaleza del *software*, todos los sistemas son propensos a presentar errores o defectos, por lo que es esencial hacer pruebas funcionales del *software* (Kruchten, 2004). En esta etapa se identifican errores en las especificaciones, en los diseños y en la funcionalidad como tal (Singh, 2010).

Al completar las fases anteriormente mencionadas, el proyecto está a punto de ser una realidad, y el paso que falta es el despliegue y la operación del sistema. En esta fase final, el sistema empieza a ser utilizado por los usuarios finales, se detectan errores adicionales y se identifican nuevas necesidades que pueden llevar a nuevas mejoras del sistema (Ali & Govardhan, 2010).

2.2 Proceso Unificado de Desarrollo (RUP)

A pesar de que la metodología en cascada era ampliamente utilizada en los proyectos de desarrollo de *software*, esta aún tenía una serie de desventajas que no le permitían a las empresas de desarrollo de *software* alcanzar un alto porcentaje de éxitos en sus proyectos (Leffingwell, 2007). Las razones comúnmente encontradas como causa de los fracasos en los proyectos fueron la dificultad para devolverse a una etapa anterior, la falta de *software* funcional en etapas tempranas, los altos niveles de riesgo e incertidumbre y la poca flexibilidad al cambio (ISTQB, 2015).

A comienzos de la década de 1990 apareció una nueva metodología conocida como Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés: *Rational Unified Process*), con el fin de cubrir las deficiencias de la metodología en cascada y, sobre todo, de la ingeniería de *software*

(Osorio, Chaudron & Heijstek, 2011). Este nuevo modelo de desarrollo agrupó las mejores prácticas del momento, de forma que fueran útiles para todo tipo de proyectos y empresas. Las prácticas que cubría eran las de desarrollo iterativo, administración de requisitos, administración de cambios y pruebas continuas, entre otros (Kruchten, 2004).

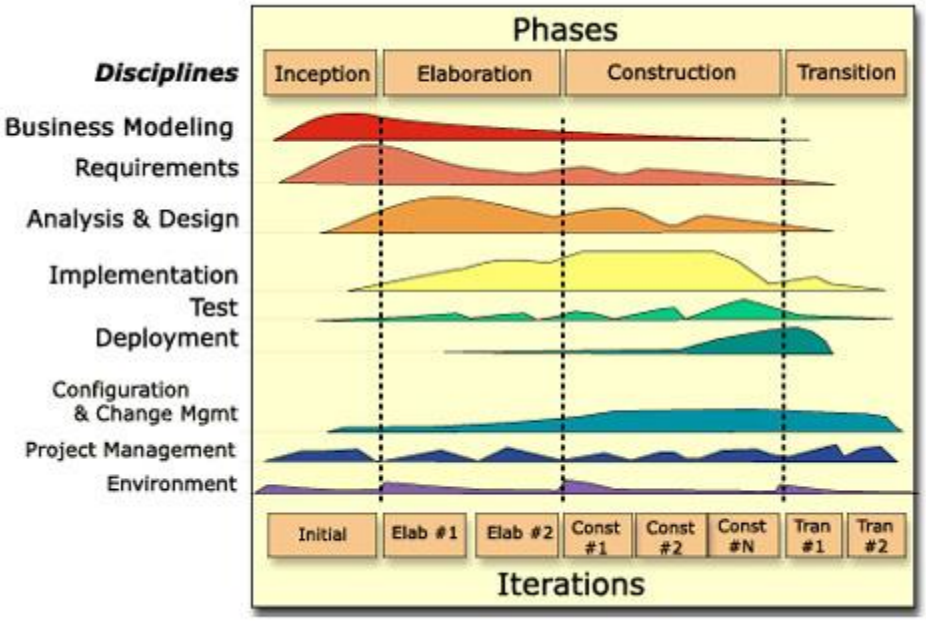

Ilustración 2. Disciplinas, fases e iteraciones de RUP (Kroll & Kruchten, 2003).

A pesar de que a simple vista el proceso unificado de desarrollo de *software* resulta muy parecido a la metodología en cascada, este tiene una gran diferencia, y es la introducción de una aproximación iterativa (ilustración 2), la cual ayuda a construir el sistema en varias etapas, que promueven la retroalimentación por parte del cliente (Kharytonov, 2015). Adicionalmente, esta metodología contó con el apoyo de una de las compañías de tecnología más grandes e importantes del mundo, como lo es IBM, lo que ayudó a su pronta expansión en todos aquellos países pioneros en el desarrollo de *software* (Borges, Monteiro & Machado, 2011).

Sin embargo, las mejoras introducidas en este nuevo modelo no resultaron suficientes para frenar la crisis del *software*, reconocida públicamente a partir del *Reporte CHAOS (1994)* (The Standish Group International, 1994), en el cual se presentaron porcentajes que mostraban que tan solo el 16 % de los proyectos de desarrollo de *software* habían sido exitosos, en contraste con un 53 % que terminaron con fallas y un 31% restante fueron cancelados. Las razones mayormente identificadas como las causantes de esta crisis fueron la falta de requisitos claros, la no participación activa de los clientes y(o) de los usuarios finales y la falta de apoyo gerencial (Wirth, 2008).

3. Metodologías ágiles

A raíz de estos problemas claramente identificados por el reporte CHAOS, la ingeniería de *software* se fijó en las metodologías ágiles con el fin de fortalecer las debilidades anteriormente mencionadas y de devolverle la confianza a una ciencia determinante en la innovación del mundo moderno.

Las metodologías ágiles para la gestión de proyectos han ganado bastante popularidad en las últimas dos década, debido a la alta efectividad que han demostrado en términos de satisfacción de las necesidades de los clientes (Mann & Maurer, 2005). Esto ha sido posible gracias a la adaptabilidad y flexibilidad con la que se cuenta para aceptar los cambios y utilizarlos como una ventaja competitiva dentro de los proyectos de desarrollo de *software* (Ungan, Cizmeli, & Demirörs, 2014). A partir de estos beneficios, grandes empresas de desarrollo del *software* se fueron inclinando hacía las metodologías ágiles, con el fin de mejorar la calidad de los sistemas que construían y de lograr una satisfacción de sus clientes.

A dichas empresas este paso les implicó realizar grandes cambios en sus formas de trabajar que, parafraseando a (Pressman, 2009), incluyen:

- Ser efectivos en la respuesta a los cambios.
- Mejorar la comunicación con los *stakeholders* de los proyectos.
- Involucrar al cliente con el equipo.
- Empoderar al equipo del trabajo a realizar.
- Entregar *software* funcional de manera incremental.
- Eliminar tareas que no aporten valor real.

Estos cambios se fueron dando de forma progresiva, y cada empresa lo hacía de una forma diferente; sin embargo, en el 2001, gracias a la creación de "The Agile Alliance", donde 17 expertos en diferentes metodologías ágiles se reunieron para discutir sobre las mejores prácticas para el desarrollo de *software*, se firmó *The Agile Manifesto* (Cohen, Lindvall & Costa, 2003).

Dicho manifiesto, en su versión en español: Manifiesto por el Desarrollo Ágil de Software (Agile Alliance, 2001), dice:

> 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. (s. p.).

De la misma forma, se crearon 12 “Principios del Manifiesto Ágil”, con los que todos los equipos de desarrollo de *software* tienen que funcionar, para así lograr una diferencia notable con respecto a los equipos que trabajan desde metodologías tradicionales.

Los 12 principios ágiles son:

> 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 del negocio y los desarrolladores trabajan 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 del 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. (Agile Alliance, 2001).

El mundo ágil cuenta con diferentes corrientes y metodologías, entre las cuales se destacan: Programación Extrema (XP, por sus siglas en inglés, *Extreme Programming*), Scrum, Método de Desarrollo de Sistema Dinámicos (DSDM, por sus siglas en inglés, *Dynamic System Development Method*), Desarrollo de Software Adaptativo, Metodología Cristal, Desarrollo Basado en Funcionalidades (FDD, por sus siglas en inglés, *Feature Driven Development*) y Kanban, entre otros (Wirth, 2008). Para este trabajo, nos vamos a centrar en la metodología Scrum, hasta el día de hoy una de las más utilizadas en la presente década (Cohen et ál., 2003).

4. Scrum

Con base en los principios de iteraciones y retroalimentación por parte del cliente, propuestos a partir del proceso unificado de desarrollo y del modelo de agilidad de las empresas de manufactura japonesas, Ken Schwaber y Jeff Sutherland propusieron en 1995 una nueva

metodología llamada Scrum (Richter, 2015), la cual es idónea para proyectos donde los requisitos y necesidades tienden a variar en el tiempo y deben adaptarse rápidamente a los cambios (Vlaanderen, Brinkkemper, Jansen & Jaspers, 2009). Ambas situaciones, muy comunes en el día a día de la ingeniería de *software*. De hecho, en 1995, Ken Schwaber y Jeff Sutherland publicaron el artículo "SCRUM Software Development Process", donde exponen que el alto rendimiento en el desarrollo de nuevos y complejos productos es alcanzado por equipos, que son pequeñas y autoorganizadas unidades de personas, motivados por objetivos y no por la asignación de tareas (Sutherland & Schwaber, 2015). Los mejores equipos son aquellos que reciben orientación acerca de la dirección en la que deben ir, pero con la libertad de escoger y desarrollar la estrategia que consideren óptima para lograr su objetivo. Los equipos que trabajan con la metodología Scrum requieren entonces de cierta autonomía en la toma de decisiones, para alcanzar la excelencia (Schwaber, 1997).

Desde entonces Scrum ha sido el buque insignia de las metodologías ágiles y ha sido ampliamente adaptado a un sinnúmero de empresas de desarrollo de *software*, e incluso a grandes empresa de diferentes sectores a nivel mundial (Leffingwell, 2007). Scrum es un marco de referencia que define un conjunto de actividades, roles y valores, que permiten que cada equipo adapte el modelo a sus necesidades y lo utilice durante los proyectos de desarrollo de *software* (Bass, 2014). El marco de trabajo de esta metodología se aprecia en la ilustración 3.

Ilustración 3. Marco de trabajo de Scrum (Scrum Alliance, s. f.).

Los roles definidos por la metodología son:

- *Scrum Master*, que es el líder espiritual del equipo, la persona encarga de garantizar que los integrantes del equipo de desarrollo tengan todas las habilidades necesarias para

iniciar un proyecto y llevarlo a feliz término. Igualmente, son los responsable de planear las capacitaciones necesarias para el equipo y de mantener una motivación alta en cada uno de los integrantes (Cervone, 2011).

- *Dueño de producto*, que es la persona encargada de conocer las necesidades del cliente y de explicarlas de forma clara al equipo. Esta persona debe representar al cliente y a todos los *stakeholders* (interesados) del proyecto y ser el único medio de comunicación entre estos y el equipo de desarrollo (Vlaanderen et ál., 2009).
- *El equipo de desarrollo*, que son todas las personas que deben participar activamente para completar de manera exitosa la implementación del sistema. Acá sobresale una figura que no está definida por Scrum, pero que juega un papel muy importante en el desempeño del equipo y en el resultado final, y es el arquitecto de *software* encargado de diseñar la solución técnica y de garantizar que los demás desarrolladores cumplan con los requisitos técnicos del proyecto (Cervone, 2011).

4.1 Desafíos de la implementación de Scrum

A pesar de tener una definición clara de lo que es Scrum y de cómo implementarlo en el desarrollo de *software*, esta no es una tarea trivial para una empresa de desarrollo de *software*, debido a que muchas de estas empresas ya habían adoptado otro tipo de metodologías previas, por ejemplo el modelo cascada o el proceso unificado de desarrollo, y las habían venido ejecutando durante mucho tiempo (Romano & Delgado, 2015). Esto ha ocasionado el nacimiento de muchos retos que deberán ser afrontados de manera rápida y abierta, para lograr un cambio de metodología de forma exitosa (Noordeloos, Manteli & Vliet, 2012).

Cabe destacar que han sido muchas las empresas de desarrollo de *software*, inclusive empresas colombianas, que se han visto en la necesidad de evolucionar hacía esta nueva metodología, y que han generado cambios en sus modelos de negocio y en la forma en la que operan; sin embargo, estas transiciones no siempre han sido transparente para los empleados, los altos directivos, los proveedores o incluso para los clientes, por lo que las empresas se ven obligadas a invertir recursos económicos, tiempo, e inclusive sacrificar ingresos, con el fin de innovar en los servicios de desarrollo de *software* (Noordeloos et ál., 2012).

A pesar de no existir una lista oficial de empresas que hayan implementado Scrum como metodología para la gerencia proyectos, sí existen varias listas de los principales retos e impactos intrínsecos a esta migración de modelo (Beecham, Noll & Richardson, 2014). Entre ellos podemos destacar la necesidad de entrenamiento adecuado para el personal existente, el involucramiento de los altos ejecutivos, el tamaño de la organización, la comunicación entre los equipos, el involucramiento de los clientes y el ambiente de trabajo (Livermore, 2008; Cho, 2008).

Aunque existen diferentes documentos de lecciones aprendidas de empresas que han pasado por el proceso de cambios de metodología hacia Scrum, todavía es muy común ver nuevas

compañías que emprenden el mismo viaje para mejorar sus resultados organizacionales, pero que no tienen en cuenta los retos y los impactos que han sido evidenciados en otros lugares del mundo (Noordeloos et ál., 2012).

5. El conflicto

La palabra 'conflicto' puede tener diferentes significados, dependiendo del contexto en el que se esté hablando. Para este trabajo, se va a tomar la definición del *Diccionario de la lengua española* (conflicto, 2014): "Problema, cuestión, materia de discusión". Es necesario tener en cuenta que el contexto al que se hace referencia es un contexto organizacional, donde un grupo de personas está reunido con una meta en común y cada individuo tiene que trabajar en conjunto con las demás personas para alcanzar dicho objetivo.

Según la definición de organización presentada en el párrafo anterior, no se deberían generar conflictos en este tipo de grupos porque todas las personas tienen metas compartidas y quieren alcanzar los mismos objetivos; sin embargo, cuando hay interacciones entre las personas, cada una de ellas tiene un interés en particular, y al percibir que la otra persona se opone o que afecta de forma negativa sus intereses, se producen entonces los conflictos.

Al darse por hecho que dentro de las organizaciones los grupos tienen conflictos, es entonces necesario afrontarlos y manejarlos de modo eficiente, para que estos no impacten de manera negativa el desempeño de la organización. De esta forma, se pueden encontrar cinco estilos para afrontar conflictos (ilustración 4): competencia, colaboración, evasión, cesión y compromiso (Northwest Frontier ATTC, 2009).

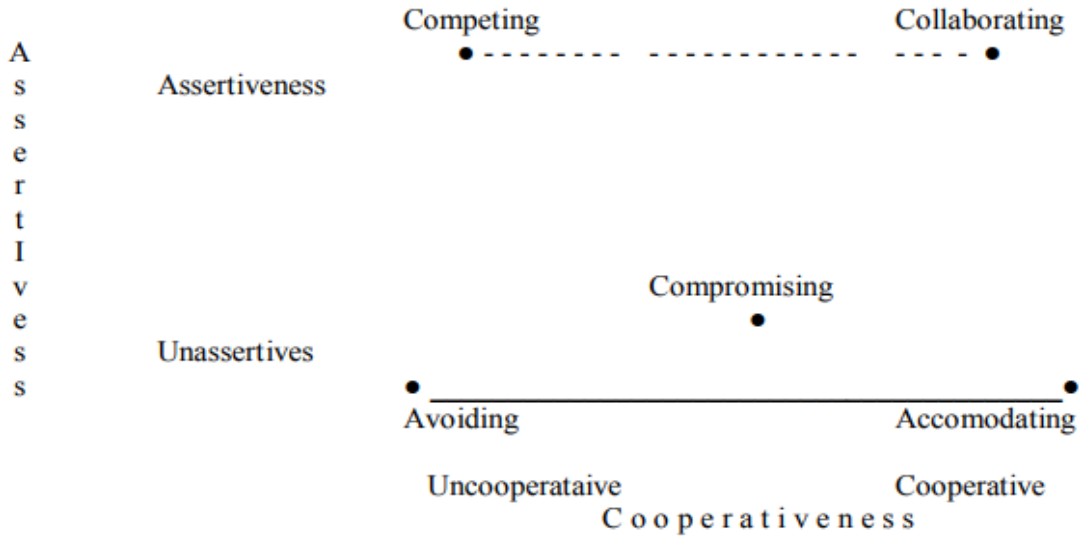

Ilustración 4. Modos de manejo de conflictos (Thomas & Kilman, 1974).

A continuación se describen los cinco estilos para afrontar conflictos:

Competencia: combinación de egoísta y no colaborativo. Es cuando una persona persigue su propio beneficio a cualquier costo. Se busca imponer la propia posición y hacer mucho más énfasis en los intereses propios, y poco en los intereses de la otra persona. Es una forma orientada al poder, en la cual se utiliza cualquier tipo de poder que sirva para ganar.

Evasión: combinación de no egoísta y no colaborativo. Es evitar el conflicto a toda costa. No se persiguen ni siquiera los propios intereses, lo que ocasiona que no se haga nada ni para satisfacer sus propios intereses ni los intereses de la otra persona. Es una forma de no afrontar el conflicto, es una forma diplomática de apartarse del conflicto.

Colaboración: combinación de egoísta y colaborativo. Es el opuesto de la evasión. Se afronta el conflicto de frente, tratando de entenderla a fondo, con el fin de lograr identificar los propios intereses y los intereses reales de la otra persona. Se intentan alcanzar los intereses propios y los de la otra persona. La intención de cada persona es aclarar las diferencias con la otra parte, en lugar de ceder en sus puntos de vista. Este es el estilo al que se debe aspirar.

Cesión: combinación de no egoísta y colaborativo. Es el opuesto de la competencia. Se afronta el conflicto dispuesto a satisfacer a la otra persona, renunciando incluso a sus propios intereses. Se hace muy poco o nada para lograr satisfacer sus propios intereses. Se evita molestar a la otra persona, pensando que de esa forma se puede mantener la relación a largo plazo.

Compromiso: intermedio entre egoísta y colaborativo. Es la forma de encontrar un punto medio que sea aceptado por ambas partes, con el fin tener una convivencia pacífica entre ambas. El objetivo es ganar un poco y ceder un poco, para que las dos personas se sientan bien y se solucione rápido el conflicto. No se evita el conflicto, pero tampoco se explora profundamente, como en la colaboración. Tampoco se satisface totalmente a la otra parte, como en la cesión, pero no se persigue el beneficio propio como único fin, como en la competencia.

5.1 Los conflictos en el desarrollo de *software*

Por la definición de conflicto anteriormente mencionada, es natural que este se presente en las empresas de desarrollo de *software*. En estas empresas, en general, se cuenta con diferentes grupos o equipos de trabajo que llevan a cabo proyectos y diferentes encargos hechos por sus clientes. Estos equipos de trabajo tienen una serie de relaciones, entre ellos mismos y con sus clientes, que pueden llevar a que surjan algunos conflictos entre las distintas partes.

Este es el caso de una de las empresas de desarrollo de *software* más grande del país. Esta empresa tiene su sede principal en la ciudad de Medellín y, adicionalmente, cuenta con sedes en otras de las principales de ciudades de Colombia, Estados Unidos y México. Dicha compañía

cuenta con cerca de 600 empleados y más de 70 proyectos activos al día de hoy (octubre de 2017). En esta empresa se utiliza principalmente la metodología Scrum.

Teniendo en cuenta la cantidad de personas que trabajan en los diferentes proyectos de la empresa, y cada uno de ellos tiene un conjunto de experiencias y conocimientos diferentes, es común encontrar diferentes puntos de vistas en cada proyecto que pueden llevar a conflictos. Estos conflictos suelen afectar el rendimiento de cada grupo de trabajo y que posiblemente pueden llegar a comprometer el éxito de su proyecto.

6. Diseño metodológico

Este proyecto de grado se enfoca en establecer la relación que hay entre el perfil del estilo de solución de conflictos y el desempeño en cinco equipos de desarrollo de *software* en una de las empresas de desarrollo de *software* más grande de Colombia. Para lograrlo, se llevaron a cabo 29 encuestas en cinco equipos diferentes, con el fin de identificar cuál es el estilo de solución de conflictos predominante de cada uno de ellos. Adicionalmente, se identificaron cuatro personas de la empresa que conocen el funcionamiento de cada equipo, desde un punto de vista gerencial, y se les realizó una encuesta sobre la percepción que tienen del desempeño de cada grupo. A este selecto grupo de personas se le seguirá haciendo referencia como "jurado".

Para identificar el estilo de solución de conflictos de cada uno de los integrantes de los grupos encuestados se utilizó el test de Thomas-Kilman. Este test no tiene ninguna pregunta como tal; en realidad, son 30 pares de afirmaciones que describen el posible comportamiento de la persona en situaciones donde sus deseos son diferentes a los de cualquier otra persona. A cada encuestado se le solicitó que seleccionara el comportamiento que más probablemente asumiría en ese tipo de situaciones (Thomas & Kilman, 1974).

Así mismo, al jurado se le hicieron cinco preguntas por cada grupo, y una sexta pregunta en general, con el fin de identificar el desempeño de cada equipo. Para las primeras cinco preguntas, se utilizó una calificación de 1 hasta 5, donde 1 es la calificación más baja y 5 es la más alta. Al finalizar las encuestas, se procedió a sumar la cantidad total de puntos obtenidos por equipo y se identificó cuál era el equipo que tenía el mejor desempeño. Las preguntas que se les hicieron al grupo de jurados fueron:

1. ¿Desde su punto de vista, qué tan satisfecho está el cliente con el proyecto? (Siendo 5 "muy satisfecho" y 1 "nada satisfecho").

2. ¿Qué tanto cree usted que el equipo cumple con los plazos de entrega? (Siendo 5 "siempre cumple" y 1 "nunca cumple").

3. ¿Desde su percepción, qué tan rápido cree que el equipo toma decisiones para solucionar un conflicto? (Siendo 5 “muy rápido” y 1 “muy demorado”).

4. ¿Desde su punto de vista, qué tan buenas son las relaciones internas entre los miembros del equipo? (Siendo 5 “excelente”, y 1 “muy malas”).

5. ¿Cuál considera usted que es la capacidad de solución de conflictos del equipo? (Siendo 5 “autosuficiente” y 1 “dependiente”).

6. ¿Desde su punto de vista, cuál es el mejor proyecto de los 5 anteriormente mencionados, en términos de rendimiento?

7. Análisis de resultados

Como se mencionó anteriormente, se contó con la participación de cinco grupos de desarrollo de *software*, a los cuales se les realizó el test de Thomas-Kilman, y a cada jurado se le preguntó por el desempeño de esos equipos. A continuación, se presentan los resultados de cada uno de los equipos.

7.1 Equipo 1

Este es un grupo que se encarga de construir diferentes sistemas para una importante empresa del sector de venta directa en multinivel y que cuenta con presencia en cerca de 50 países. Este grupo viene trabajando junto por más de dos años y medio, y ha entregado seis sistemas funcionales. Los resultados de los integrantes del equipo obtenidos en el test de Thomas-Kilman y en la evaluación de los jurados, presentados en las tablas 1 a la 7, son los siguientes:

Tabla 1. Equipo 1 – Persona 1

Competencia	Colaboración	Compromiso	Evasión	Cesión
3	6	9	6	6

Tabla 2. Equipo 1 – Persona 2

Competencia	Colaboración	Compromiso	Evasión	Cesión
3	10	6	7	4

Tabla 3. Equipo 1 – Persona 3

Competencia	Colaboración	Compromiso	Evasión	Cesión
5	8	9	5	3

Tabla 4. Equipo 1 – Persona 4

Competencia	Colaboración	Compromiso	Evasión	Cesión
3	9	8	4	6

Tabla 5. Equipo 1 – Persona 5

Competencia	Colaboración	Compromiso	Evasión	Cesión
2	9	11	3	5

Tabla 6. Equipo 1 – Persona 6

Competencia	Colaboración	Compromiso	Evasión	Cesión
2	7	7	6	8

Tabla 7. Equipo 1 – Suma de totales

Competencia	**Colaboración**	**Compromiso**	**Evasión**	**Cesión**
18	49	50	31	32

En las anteriores tablas se puede evidenciar que en este equipo, conformado por seis personas, hay tres que tienen un estilo predominante de compromiso, dos personas que tienen un estilo colaborativo y una con estilo de ceder. De igual forma, al computar los resultados de todos ellos, el estilo predominante en el equipo es el de compromiso, seguido del estilo colaborativo.

Adicionalmente, la respuesta de los jurados ante la pregunta acerca de la satisfacción del cliente con respecto al proyecto (pregunta 1), el equipo obtuvo un puntaje de 19 puntos sobre 20 posibles. Acerca de la pregunta sobre el cumplimiento de los plazos de entrega (pregunta 2), el equipo obtuvo 20 puntos de 20. Para la pregunta sobre toma de decisiones (pregunta 3), el grupo fue calificado con 14 puntos sobre 20. Con respecto a las relaciones internas (pregunta 4), la calificación fue de 17 puntos de 20 posibles. Finalmente, en la pregunta sobre la capacidad de solución de conflictos (pregunta 5), el grupo obtuvo 16 sobre 20.

De acuerdo con lo anterior, se puede concluir que el grupo 1 obtuvo una calificación de 86/100 y también que su estilo de solución de conflictos es de compromiso.

7.2 Equipo 2

Este equipo tiene a cargo la construcción de un sistema de rastreo y almacenamiento de datos de navegación en internet de personas a nivel mundial, con el fin de usar dicha información en la presentación de publicidad digital dirigida. Este grupo lleva seis meses trabajando juntos, por lo que se puede considerar que todavía se están adaptando a trabajar en conjunto. El test de Thomas-Kilman y la evaluación de los jurados arrojaron los resultados que se muestran a continuación en las tablas 8 a la 13.

Tabla 8. Equipo 2 – Persona 1

Competencia	Colaboración	Compromiso	Evasión	Cesión
5	6	4	8	7

Tabla 9. Equipo 2 – Persona 2

Competencia	Colaboración	Compromiso	Evasión	Cesión
4	9	10	6	1

Tabla 10. Equipo 2 – Persona 3

Competencia	Colaboración	Compromiso	Evasión	Cesión
7	5	7	9	2

Tabla 11. Equipo 2 – Persona 4

Competencia	Colaboración	Compromiso	Evasión	Cesión
10	4	8	6	2

Tabla 12. Equipo 2 – Persona 5

Competencia	Colaboración	Compromiso	Evasión	Cesión
7	5	4	8	6

Tabla 13. Equipo 2 – Suma de totales

Competencia	**Colaboración**	**Compromiso**	**Evasión**	**Cesión**
33	29	33	37	18

Para este grupo, conformado por cinco personas, se observa que el estilo de solución de conflictos predominante es el de la evasión. Además se observa algo muy llamativo, y es la alta competencia que también se puede apreciar, tanto como estilo de una persona como en la suma total de los resultados. Esto cobra gran importancia, porque describe a un equipo que, en general, trata de evadir los problemas, y, cuando realmente quiere afrontarlos, cada integrante quiere imponer sus propios intereses sobre los demás.

En la encuesta realizada luego al grupo de jurados, el equipo obtuvo un puntaje de 12 puntos en cuanto a la satisfacción del cliente con respecto al proyecto (pregunta 1). La calificación recibida en el cumplimiento de plazos de entrega (pregunta 2) fue de 12 puntos de 20. En la pregunta acerca de la toma de decisiones (pregunta 3), el resultado fue de 9 puntos sobre 20. En cuanto a las relaciones internas del grupo de trabajo (pregunta 4), el equipo fue calificado con 13 puntos de 20 posibles, y en la capacidad de solución de conflictos (pregunta 5) el resultado fue de 8 puntos sobre 20 posibles. Por lo tanto, se puede concluir que el grupo obtuvo un resultado de

54/100. Adicionalmente, que tiene un estilo de solución de conflictos evasivo, seguido por un estilo de competencia.

7.3 Equipo 3

Este grupo está construyendo un *software* para la gestión de afiliados de una de las EPS más grandes de Colombia, teniendo como retos el volumen de datos, los tiempos máximos de espera y la falta de información por parte del cliente. Este equipo ha estado trabajando en conjunto por más de año y medio. En el test de Thomas-Kilman y en las calificaciones de los jurados, el equipo obtuvo los resultados que se presentan en las tablas 14 a la 20.

Tabla 14. Equipo 3 – Persona 1

Competencia	Colaboración	Compromiso	Evasión	Cesión
6	7	3	6	8

Tabla 15. Equipo 3 – Persona 2

Competencia	Colaboración	Compromiso	Evasión	Cesión
8	5	5	4	8

Tabla 16. Equipo 3 – Persona 3

Competencia	Colaboración	Compromiso	Evasión	Cesión
2	4	7	8	9

Tabla 17. Equipo 3 – Persona 4

Competencia	Colaboración	Compromiso	Evasión	Cesión
9	4	7	7	3

Tabla 18. Equipo 3 – Persona 5

Competencia	Colaboración	Compromiso	Evasión	Cesión
7	5	8	8	2

Tabla 19. Equipo 3 – Persona 6

Competencia	Colaboración	Compromiso	Evasión	Cesión
7	4	7	8	4

Tabla 20. Equipo 3 – Suma de totales

Competencia	**Colaboración**	**Compromiso**	**Evasión**	**Cesión**
39	29	37	41	34

En este grupo, conformado por seis personas, se encontró que el estilo de solución de conflictos más común entre sus miembros es el de la cesión; sin embargo, al totalizar los resultados de todos los integrantes se evidencia que el estilo predominante es el de la evasión. Ambos estilos tienen como factor común la falta de asertividad, lo que afecta directamente las habilidades de comunicación del grupo. Adicionalmente, esta combinación es peligrosa porque el grupo normalmente tratará de evitar afrontar los conflictos, y cuando realmente decidan hacerlo, tenderán a complacer a la otra parte involucrada, lo que llevará al equipo a no satisfacer sus intereses propios como equipo.

Adicionalmente, la valoración acerca de la satisfacción del cliente (pregunta 1), que el grupo obtuvo por parte de los jurados, fue de 13 puntos sobre 20 posibles. Con respecto al cumplimiento de plazos de entrega (pregunta2), el grupo obtuvo un total de 8 puntos sobre 20. En la pregunta sobre la toma de decisiones (pregunta 3), la puntuación obtenida fue de 9 puntos de 20 posibles. Para la pregunta sobre las relaciones entre los miembros del equipo (pregunta 4), el resultado fue de 12 puntos sobre 20 posibles. Y, ante la pregunta sobre solución de conflictos, el jurado le otorgó 7 puntos sobre 20.

En conclusión, el grupo fue calificado con un total de 49 puntos sobre 100. Igualmente, se puede decir que tiene un estilo predominante de solución de conflictos de tipo evasivo.

7.4 Equipo 4

Este equipo trabaja con una empresa dedica a la publicidad digital y se enfoca en construir una herramienta que permite realizar compra y venta de espacios publicitarios en anunciantes de talla mundial a través de subastas en tiempo real. Este grupo viene trabajando en conjunto por más de 2 años. Los resultados obtenidos por sus integrantes en el test de Thomas-Kilman y en la calificación de los jurados, se presentan a continuación en las tablas 21 a la 27.

Tabla 21. Equipo 4 – Persona 1

Competencia	Colaboración	Compromiso	Evasión	Cesión
5	10	8	3	4

Tabla 22. Equipo 4 – Persona 2

Competencia	Colaboración	Compromiso	Evasión	Cesión
6	6	9	5	4

Tabla 23. Equipo 4 – Persona 3

Competencia	Colaboración	Compromiso	Evasión	Cesión
6	4	10	4	6

Tabla 24. Equipo 4 – Persona 4

Competencia	Colaboración	Compromiso	Evasión	Cesión
1	6	10	6	7

Tabla 25. Equipo 4 – Persona 5

Competencia	Colaboración	Compromiso	Evasión	Cesión
8	7	10	4	1

Tabla 26. Equipo 4 – Persona 6

Competencia	Colaboración	Compromiso	Evasión	Cesión
9	5	9	3	4

Tabla 27. Equipo 4 – Suma de totales

Competencia	**Colaboración**	**Compromiso**	**Evasión**	**Cesión**
35	38	56	25	26

Se evidencia que en este equipo hay cinco personas que tienen un estilo de solución de conflictos comprometida y que hay una persona con estilo colaborativo. De igual forma, al sumar los resultados de cada uno de los integrantes se obtuvo que el estilo predominante es el de comportamiento comprometido.

Adicionalmente, la respuesta por parte de los jurados a la pregunta acerca de la satisfacción del cliente con respecto al proyecto (pregunta 1), el equipo obtuvo un puntaje de 20 puntos sobre 20 posibles. Sobre la pregunta de cumplimiento de plazos de entrega (pregunta 2), el equipo obtuvo 20 puntos de 20. Para la pregunta sobre la toma de decisiones (pregunta 3), el grupo fue calificado con 19 puntos sobre 20. Con respecto a las relaciones internas (pregunta 4), la calificación fue de 17 puntos de 20 posibles. Finalmente, en la pregunta sobre la capacidad de solución de conflictos (pregunta 5), el grupo obtuvo 19 puntos sobre 20.

De acuerdo con lo anterior, se puede concluir que el grupo 4 obtuvo una calificación de 95/100. También se evidencia que el estilo de solución de conflictos predominante es el de compromiso.

7.5 Equipo 5

El grupo 5 tiene a su cargo la construcción de sistemas que son utilizados en cárceles, a nivel mundial, para gestionar las celdas de los prisioneros, sus comidas, sus dotaciones y sus medicamentos, entre otros. El equipo ha trabajado en este proyecto por cerca de tres años, y sus resultados en el test de Thomas-Kilman y en la calificación del grupo de jurados son los que se presentan a continuación, en las tablas 28 a la 34.

Tabla 28. Equipo 5 – Persona 1

Competencia	Colaboración	Compromiso	Evasión	Cesión
4	6	11	7	2

Tabla 29. Equipo 5 – Persona 2

Competencia	Colaboración	Compromiso	Evasión	Cesión
11	5	6	7	1

Tabla 30. Equipo 5 – Persona 3

Competencia	Colaboración	Compromiso	Evasión	Cesión
7	6	8	7	2

Tabla 31. Equipo 5 – Persona 4

Competencia	Colaboración	Compromiso	Evasión	Cesión
6	5	9	6	4

Tabla 32. Equipo 5 – Persona 5

Competencia	Colaboración	Compromiso	Evasión	Cesión
2	9	10	6	3

Tabla 33. Equipo 5 – Persona 6

Competencia	Colaboración	Compromiso	Evasión	Cesión
4	5	10	7	4

Tabla 34. Equipo 5 – Suma de totales

Competencia	**Colaboración**	**Compromiso**	**Evasión**	**Cesión**
34	36	54	40	16

En este grupo de seis personas, se evidencia que el estilo de solución de conflictos predominante es el del compromiso. Este estilo es el que más se repite en 5 de los 6 integrantes; de igual forma, al hacer la suma de los resultados como equipo, el estilo que más puntos obtiene es el del compromiso.

Al revisar la evaluación del grupo de jurados, este equipo obtuvo una calificación de 18 puntos en la pregunta sobre la satisfacción del cliente (pregunta 1). Para la pregunta sobre el cumplimiento de los plazos de entregas (pregunta 2), el equipo fue calificado con 19 puntos sobre 20 posibles. Con respecto a la pregunta sobre la toma de decisiones (pregunta 3), el equipo obtuvo un puntaje de 17 puntos de 20. Acerca de la convivencia interna de los miembros del

equipo (pregunta 4), el puntaje obtenido fue de 18 puntos sobre 20. Por último, en la calificación de la solución de conflictos (pregunta 5), el equipo obtuvo 18 puntos de 20 posibles.

Teniendo en cuenta lo anteriormente mencionado, el grupo 5 obtuvo una calificación general de 90 puntos sobre 100 puntos posibles. También se puede concluir que el estilo de solución de conflictos comprometido es el predominante en el grupo.

8. Conclusiones

A continuación, en la tabla 35 se muestra el resumen de las calificaciones que el grupo de jurados le dio a cada uno de los equipos que hicieron parte de este estudio.

Tabla 35. Evaluación de jurados

	Equipo 1	Equipo 2	Equipo 3	Equipo 4	Equipo 5
Pegunta 1	19	12	13	20	18
Pregunta 2	20	12	8	20	19
Pregunta 3	14	9	9	19	17
Pregunta 4	17	13	12	17	18
Pregunta 5	16	8	7	19	18
Total	**86**	**54**	**49**	**95**	**90**
Posición	**3**	**4**	**5**	**1**	**2**

En la anterior tabla se puede observar que, en la posición en la que quedó cada equipo en términos de desempeño, el equipo 4 fue el mejor ubicado, y el equipo 3 fue el que ocupó el último lugar en la lista.

Adicionalmente, se pueden resaltar algunas de las siguientes evidencias:

- El equipo 4 quedó en la primera posición del *ranking* de desempeño, obteniendo 95 puntos sobre 100 posibles. Este puede considerarse un equipo de alto desempeño, con un rendimiento excelente, y cuyo estilo de solución de conflictos predominante es el de compromiso. En las encuestas realizadas al grupo de jurados también se evidencia que este equipo fue catalogado por tres personas del jurado como el mejor de los 5 equipos.

- El equipo 5 quedó en la segunda posición del *ranking*, con una calificación de 90 puntos sobre 100 posibles. Esto se puede considerar como un grupo con un desempeño bastante bueno, con un estilo de solución de conflictos de compromiso.

- El equipo 1 quedó ubicado como tercero en el *ranking* de desempeño, con una evaluación de 86 puntos. Esa calificación le permite ser considerado un equipo de buen desempeño y buen rendimiento. Adicionalmente, tiene un estilo dominante comprometido.

- El equipo 2 quedó en la cuarta posición en la evaluación de desempeño realizada por el jurado, con una calificación de 54 puntos de 100 posibles. Su estilo de solución de conflictos predominante es el evasivo

- El equipo 3 fue el que obtuvo la calificación más baja de los 5 grupos estudiados, obteniendo 49 puntos sobre 100. Esto se puede interpretar como un rendimiento regular y con un margen alto de mejora. Así mismo, su estilo predominante es el de la evasión.

De acuerdo con lo anterior, se evidencia una relación directa entre el estilo predominante de solución de conflictos con el desempeño y el rendimiento de cada equipo. Los tres equipos que obtuvieron un mayor puntaje (entre 86 y 95 puntos) son equipos que solucionan sus problemas de forma comprometida, logrando encontrar soluciones rápidas que satisfacen a las partes implicadas, de modo que todos ganan un poco en cuanto a sus intereses, pero también ceden un poco, algo que les ayuda a tener una convivencia pacífica entre los integrantes de dichos equipos.

Por otro lado, se encuentran los equipos 2 y 3, con calificaciones de 54 y 49 puntos respectivamente, y cuyo estilo de solución de conflictos es evasivo. Esto indica que los integrantes de estos dos equipos son personas que tienden a huir de sus problemas sin antes tratar de satisfacer sus propios intereses, permitiendo, en mucha ocasiones, que los conflictos escalen y sean cada vez más complejos de solucionar.

9. Referencias bibliográficas

Agile Alliance (2001). History: The Agile Manifesto. Recuperado de *Manifesto for Agile Software Development*. Disponible en http://agilemanifesto.org/iso/es/manifesto.html

Ali, N. & Govardhan, A. (2010). A Comparison Between Five Models Of Software Engineering . *IJCSI International Journal of Computer Science Issues, 7*(5), 94-101. Disponible en https://www.ijcsi.org/papers/7-5-94-101.pdf

Bass, J. M. (2014). Scrum Master Activities: Process Tailoring in Large Enterprise Projects. *IEEE Computer Society*, 6-15. DOI: http://dx.doi.org/10.1109/ICGSE.2014.24

Bassil, Y. (2012). A Simulation Model for the Waterfall. *International Journal of Engineering & Technology, 2*(5), 742-749. Disponible en http://iet-journals.org/archive/2012/may_vol_2_no_5/255895133318216.pdf

Beecham, S., Noll, J., & Richardson, I. (2014). Using Agile Practices to Solve Global Software Development problems – A Case Studio. *2014 IEEE International Conference on Global Software Engineeering Workshops*, 5-10. Disponible en DOI: 10.1109/ICGSEW.2014.7

Borges, P., Monteiro, P., & Machado, R. J. (2011). Tailoring RUP to Small Software Development. *2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications*. *Software Engineering and Advanced Applications*, 306-309. DOI: 10.1109/SEAA.2011.55

Cervone, H. F. (2011). Understanding agile project management methods using Scrum. *OCLC* Systems and Services. *International digital library perspectives, 27*(1), 18-22. Bradford: Emerald Group Publishing. DOI: https://doi.org/10.1108/10650751111106528

Cho, J. (2008). Issues and Challenges of Agile Software Development with Scrum. *Issues in Information Systems, IX*(2), 188-195. Disponible en http://iacis.org/iis/2008/S2008_950.pdf

Cnfolio (s. f.). Waterfall model of software development life cycle. *System Analysis.* Recuperado de http://cnfolio.com/systemsanalysis11

Cohen, D., Lindvall, M., & Costa, P. (2003). *DACS State-of-the-Art/Practice Report. Agile Software Development.* Washington D.C.: Fraunhofer Center Maryland. Disponible en http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.201.2704&rep=rep1&type=pdf

Conflicto (2014). *Diccionario de la lengua española* (23ª ed.). Real Academia Española. Disponible en http://dle.rae.es/?id=AGHyxGk

Cusumano, M., & Smith, S. (1995). Beyond the Waterfall: Software Development at Microsoft. *International Center for Research on the Management of Technology*. Sloan School of Management, Massachusetts Institute of Technology, Vancouver. Disponible en https://pdfs.semanticscholar.org/01a1/d9047b43f001acab11bbe367d31560438bf9.pdf

International Software Testing Qualifications Board – ISTQB (2015, Septiembre 04). What is Waterfall model- advantages, disadvantages and when to use it? *ISTQB EXAM CERTIFICATION*. Recuperado de http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/

Kharytonov, S. (7 de septiembre, 2015). Waterfall, RUP and Agile: Which is Right for You? *Ebizq*. Recuperado de http://www.ebizq.net/topics/dev_tools/features/11821.html?page=1

Kroll, P. & Kruchten, P. (2003). *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP.* Boston: Addison-Wesley.

Kruchten, P. (2004). *The Rational Unified Process an Introduction.* Boston: Pearson Education.

Leffingwell, D. (2007). *Scaling Software Agility: Best Practices for Large Enterprises.* Boston: Addison-Wesley Professional.

Leonhardt, D. (2000). John Tukey, 85, Statistician; Coined the Word 'Software'. *The New York Times*, A19. Disponible en http://www.nytimes.com/2000/07/28/us/john-tukey-85-statistician-coined-the-word-software.html

Liu, L.-C., & Horowitz, E. (1989). A Formal Model for Software Project Management. *IEEE Transactions on Software Engineering*, *15*(10), 1280-1293. DOI: 10.1109/TSE.1989.559781

Livermore, J. (2008). Factors that Significantly Impact the Implementation of an Agile Software Development Methodology. *Journal of Software, 3(4)*. Disponible en http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.522.2441&rep=rep1&type=pdf

Lott, C. M. (1997). Breathing new life into the waterfall model. *IEEE Software, 14*(5), 103-105. DOI: 10.1109/52.605938

Mann, C., & Maurer, F. (2005). A Case Study on the Impact of Scrum on Overtime and Customer Satisfaction. *Agile Conference, 2005. Proceedings*. Denver: IEEE. DOI: 10.1109/ADC.2005.1

Noordeloos, R., Manteli, C., & Vliet, H. v. (2012). From RUP to Scrum in Global Software Development: A Case Study. *IEEEE Computer Society*, 31-40. DOI 10.1109/ICGSE.2012.11

Northwest Frontier ATTC (2009). Manejo de Conflictos. *Adiction Messenger. Series 34, 12*(5). Recuperado de https://www.attchub.org/userfiles/file/NorthwestFrontier/Vol_%2012,%20Issue%205-%20NEW%20SPANISH.pdf

Osorio, J., Chaudron, M., & Heijstek, W. (2011). Moving From Waterfall to Iterative Development - An Empirical Evaluation of Advantages, Disadvantages and Risks of RUP. *IEEE Xplore Digital Library*, 453-460. DOI: 10.1109/SEAA.2011.69

Pressman, R. (2009). *Software Engineering: A Practitioner's Approach.* Nueva York: McGraw-Hill.

Randell, B. (1996). *History of Software Engineering.* Newcastle: Schloss Dagstuhl.

Richter, W. (2015). nCanto: An agile software development case study. *2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW),* 1-2. DOI: 10.1109/ICSTW.2015.7107422

Romano, B. , & Delgado, A. (2015). Project Management Using the Scrum Agile Method: A Case Study within a Small Enterprise. *2015 12th International Conference on Information Technology - New Generations*, 774-776.

Royce, D. (1970). Managing the Development of Large Software Systems. *IEEE Wescon*, 1-9.

Scrum Alliance (s. f.). *Learn About Scrum*. Recuperado de https://www.scrumalliance.org/why-scrum

Singh, Y. (2011). Software Testing. *Cambridge University Press,* 37-109. DOI: https://doi.org/10.1017/CBO9781139196185

Sutherland, J. (16 de septiembre, 2015). The Scrum Guide. Recuperado de http://www.scrumguides.org/scrum-guide.html#acknowledgements-history

Schwaber, K. (1997). SCRUM Software Development Process. En Sutherland, J., Patel, D., Casanave, C., Hollowell, G., & Miller J. (eds.). *Business Object Design and Implementation OOPSLA '95 Workshop Proceedings 16 October 1995* (117-134). Londres: Springer. Disponible en http://www.springer.com/la/book/9783540760962

The Standish Group (1994). *The CHAOS Report (1994).* Disponible en https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf

Thomas, K. & Kilmann, R. (1974). *Thomas-Kilman Conflict Mode Instrument (TKI).* Tuxedo: Xicom.

Ungan, E., Cizmeli, N., & Demirörs, O. (2014). Comparison of Functional Size Based Estimation and Story Points, Based on Effort Estimation Effectiveness in SCRUM Projects. *IEEE Computer Society*, 77-80. DOI: 10.1109/SEAA.2014.83

Vlaanderen, K., Brinkkemper, S., Jansen, S., & Jaspers, E. (2009). The Agile Requirements Refinery: Applying SCRUM Principles to Software Product Management. *International Workshop on Software Product Management*, 1-10. DOI: 10.1109/IWSPM.2009.7

Wirth, N. (2008). A Brief History of Software Engineering. *IEEE Annals of the History of Computing, 30*(3), 32-39. DOI: 10.1109/MAHC.2008.33