

Estilos de aprendizaje en equipos Scrum
Caso de estudio en una de las aseguradoras más grandes de Colombia

Luis Felipe Londoño Londoño

Trabajo de grado presentado como requisito parcial para optar al título de
Magíster en Administración

Asesor:
Francisco López Gallego

Universidad EAFIT
Escuela de Administración
Medellín
2017

Contenido

Introducción	4
1. Metodologías ágiles	5
1.1 Scrum	10
1.2 Artefactos	11
1.3 Roles.....	12
1.4 Sprint	14
1.5 Aprendizaje	16
1.6 Teoría de aprendizaje experiencial.....	18
1.7 El inventario de estilos de aprendizaje	19
2. Metodología	21
2.1 Resultados	22
2.2. Hallazgos finales	26
2.3 Futuras investigaciones	27
Referencias.....	27

Lista de tablas

Tabla 1. Diferencias entre metodologías ágiles y no ágiles.....	8
Tabla 2. Relación entre los estilos de aprendizaje y cinco niveles de comportamiento	21

Lista de figuras

Figura 1. Marco de trabajo Scrum	15
Figura 2. Ciclo de aprendizaje de cuatro etapas de Kolb.....	18

Lista de gráficos

Gráfico 1. Equipo 1.....	22
Gráfico 2. Equipo 2.....	23
Gráfico 3. Equipo 3.....	23
Gráfico 4. Equipo 4.....	24
Gráfico 5. Equipo 5.....	24
Gráfico 6. Equipo 6.....	25
Gráfico 7. Todos los equipos	25

Resumen

El programa de investigación en aprendizaje por experiencia, iniciado por D. A. Kolb (1984), ha encontrado evidencia empírica que hace presumir que los equipos de trabajo conformados por integrantes que tienen diferentes prioridades en sus estilos de aprendizaje, encuentren complementariedades que los hagan más eficaces en su trabajo articulado y mancomunado. Con esa premisa básica de trabajo se evaluaron, utilizando la versión 1 de la herramienta de Kolb, la totalidad de los miembros de seis equipos Scrum de una de las más grandes aseguradoras del país. Dicha evaluación se realizó entre los meses de marzo y abril de 2017, arrojando como resultado que solamente un tercio de los equipos presentan complementariedades en los estilos de aprendizaje de sus miembros, lo cual genera interrogantes sobre los niveles de eficacia de dichos equipos según el modelo teórico.

Palabras clave: Scrum, Aprendizaje, Equipos, Inventario de estilos de aprendizaje.

Abstract

The learning-by-experience research program initiated by D. A. Kolb (1984) has found empirical evidence to suggest that work teams composed by members who have different priorities in their learning styles find complementarities that make them more effective in their articulated and joint work. With this basic working premise, all the members from 6 Scrum teams in one of the largest insurers in the country were evaluated using version 1 of the Kolb tool. This evaluation was carried out between March and April 2017, with the result that only one third of the teams present complementarities in the learning styles of their members, which raises questions about the efficiency levels of these teams according to the theoretical model.

Key words: Scrum, Learning, teams, Learning styles inventory.

Introducción

El marco de trabajo Scrum define a los equipos como un colectivo de personas a las que se les presenta un objetivo común que deben cumplir. Dicho marco propone que el equipo de trabajo sea auto-gestionado, auto-organizado y auto-suficiente, haciendo que las personas que lo conforman deban tener una visión clara del objetivo, encontrar la mejor manera de cumplirlo y contar con todas las habilidades necesarias requeridas a lo largo del proyecto (Deemer, Benefield, Larman, & Vodde, 2012). Adicionalmente, Scrum define que los equipos de trabajo deben estar comprometidos con la mejora constante y continua, intentando que en cada iteración de trabajo (*Sprint*) se tenga un equipo mejor. Por lo tanto, debido a que el equipo es la columna vertebral de la metodología y los equipos están conformados por personas, la selección de personal se convierte en una tarea esencial, ya que la falta de conocimiento, la carencia de las habilidades requeridas o la incompatibilidad de personalidades pueden llevar al equipo a no alcanzar el objetivo planteado (Licorish, Philpott, & MacDonell, 2009). Sin embargo, este importante tema no ha sido ampliamente estudiado, generando que muchos equipos no cumplan con las expectativas, a pesar de estar conformados por profesionales comprometidos (Bootla, Rojanapornpun, & Mongkolnam, 2015).

Una característica importante de las personas que conforman los equipos Scrum es la forma cómo aprenden y desarrollan nuevas habilidades, ya que la metodología exige una mejora continua, que sólo puede ser lograda cuando los miembros del equipo pueden alcanzar dichas habilidades. Esto genera que comprender el modo cómo las personas aprenden sea un factor relevante al momento de seleccionar a los miembros de un equipo.

Una teoría ampliamente aceptada sobre el aprendizaje fue postulada por D. A. Kolb (1984), mostrando que los individuos tienen naturalmente una forma única de aprender, que puede ser clasificada dentro de cuatro grandes categorías: divergente, convergente, acomodador y asimilador; cada una con diferentes características que pueden ser provechosas en distintas situaciones.

Diferentes estudios han encontrado que los equipos son más efectivos cuando son conformados por personas que cuentan con estilos de aprendizaje complementarios, es decir, dentro del equipo se pueden encontrar los cuatro estilos de aprendizaje postulados por Kolb (D. A. Kolb & Kolb, 2013). Por lo tanto, este factor debería ser relevante al momento de seleccionar los miembros de los equipos Scrum de una organización, buscando siempre el equilibrio entre las habilidades

técnicas requeridas, las características de las personas y su estilo de aprendizaje, logrando obtener un equipo de trabajo que sea capaz de mejorar en el tiempo y alcanzar los objetivos planteados. Este trabajo presenta evidencia de que la selección y conformación de los equipos Scrum en la empresa objeto de estudio, no tiene en cuenta los hallazgos de las investigaciones recopiladas por D. A. Kolb & Kolb (2013), en las que se ha demostrado la importancia de convertir a los estilos de aprendizaje en un factor relevante al momento de elegir a las personas que van a conformar el equipo.

Este estudio realizó un censo, utilizando el cuestionario de la versión 1 de los estilos de aprendizaje de Kolb (cuestionario que se puede usar libremente), que detalla la forma como están conformados los equipos Scrum del departamento de desarrollo de software de una importante compañía aseguradora colombiana. Se encuentra que la mayor parte de los equipos están constituidos por sólo dos o tres estilos de aprendizaje. Adicionalmente, se evidencia una predominancia por el estilo divergente dentro de las personas censadas.

1. Metodologías ágiles

Desde que se crearon los computadores, se ha estado debatiendo sobre las metodologías de trabajo que se deberían utilizar para el desarrollo de software. En el inicio se contaba con las denominadas metodologías tradicionales que se centraban principalmente en el control minucioso del proceso de desarrollo, estableciendo clara y rígidamente las actividades involucradas, los artefactos que se deben producir y cada una de las herramientas y notaciones que deben usarse durante el desarrollo (Canós, Letelier, & Penadés, 2003), logrando así entonces tener un completo control sobre los procesos de desarrollo y su evolución, sin embargo, se pierde toda la flexibilidad que requiere el cambiante entorno corporativo moderno y se da mayor importancia a los procesos usados para la creación del software que al software en sí.

Las nuevas tendencias en metodologías de desarrollo de software se están centrando en dimensiones diferentes al control riguroso y la extensa documentación, favoreciendo, por ejemplo, el factor humano o el producto de software terminado. A las metodologías que han surgido, producto de esta nueva tendencia, se les denomina metodologías ágiles y se enfocan en destacar el valor de los individuos que componen los equipos, la colaboración con el cliente y el desarrollo de software basado en iteraciones cortas e incrementales. Estas metodologías han mostrado que son

efectivas en proyectos con requisitos altamente cambiantes, logrando reducir notoriamente los tiempos de desarrollo y manteniendo una alta calidad en el producto de software entregado (Canós et al., 2003).

El término “ágil”, aplicado al desarrollo de software, nace en una reunión realizada en Utah, EE.UU, donde un grupo de 17 expertos en software se centraron en lograr un consenso de valores y principios, que deberían ayudar a los equipos a crear software rápidamente, respondiendo eficientemente a los cambios que se puedan presentar a lo largo de los proyectos (Cockburn, 2006). De lo acordado en esta reunión se pueden destacar tres cosas: la creación de *The Agile Alliance*, una organización sin ánimo de lucro dedicada a promover todos los conceptos del desarrollo ágil entre las organizaciones; la creación del Manifiesto ágil, un documento que resume los principios de la filosofía ágil (Canós et al., 2003) y la formulación de los doce principios que se desprenden del Manifiesto ágil (Fowler & Highsmith, 2001).

El Manifiesto ágil es la piedra angular de las metodologías que se han creado bajo la perspectiva agilista (Fowler & Highsmith, 2001). Este manifiesto, que cuenta con cuatro premisas valora en mayor medida a:

- **Individuos e interacciones sobre procesos y herramientas.** Las personas son las que llevan a cabo el proyecto de construcción de software, por lo tanto, son el principal factor de éxito. Es más importante entonces construir un buen equipo que cuente con las capacidades y la motivación necesarias para llevar un proyecto a buen término, que construir el entorno de trabajo. Muchos proyectos ejecutados, usando metodologías tradicionales han cometido el error de construir primero el entorno, esperando que los miembros del equipo se adapten a él automáticamente. En la mayoría de los casos, este enfoque no ha funcionado, y se ha demostrado que es mejor constituir el equipo y que éste tenga la libertad de crear su entorno de desarrollo con base en sus propias necesidades (Canós et al., 2003).

- **Software funcionando sobre documentación exhaustiva.** La documentación que se debe crear debe ser la mínima requerida para la toma de decisiones, esperando que estos documentos sean cortos, concisos y centrados en la parte fundamental de la decisión que apoyan (Canós et al., 2003).

- **Colaboración con el cliente sobre la negociación de un contrato.** Se enfatiza en la necesidad de una interacción constante entre el cliente y el equipo de

desarrollo, promoviendo que esta relación sea abierta y cordial. Esta colaboración mutua marca el ritmo del proyecto y asegura que lo entregado se ajuste completamente a las necesidades del cliente. Los proyectos tradicionales cuentan con contratos poco flexibles que impiden realizar cambios sobre la marcha, sin caer en la burocracia de una renegociación; en las metodologías ágiles se asume un entorno altamente cambiante, por ende, se entiende que los contratos deben ser flexibles, permitiendo que las prioridades y las actividades del proyecto cambien constante (Canós et al., 2003).

- **Responde al cambio sobre seguir un plan estricto.** La planificación de los proyectos ágiles debe ser flexible y abierta, logrando que los cambios que se presenten a lo largo del proyecto (cambios en requisitos, tecnología, equipo, entorno, etc.) puedan ser atendidos rápidamente sin afectar el rendimiento del equipo de trabajo (Canós et al., 2003).

Los principios que se desprenden de dicho manifiesto muestran las diferencias que existen entre un proyecto que adopta una metodología ágil y uno que se enmarca en las metodologías tradicionales. Cabe destacar que los dos primeros principios son generales y hacen un resumen de la esencia del espíritu ágil. Los otros están enfocados en el proceso que se debe seguir para lograr las metas y la organización del equipo. Los principios son (Agile Manifiesto, 2001):

1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana, durante todo el proyecto.
5. 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.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros, es la conversación cara a cara.
7. El software funcionando es la medida principal de progreso.

8. Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para, a continuación, ajustar y perfeccionar su comportamiento en consecuencia.

En la siguiente tabla extraída del artículo *Métodologías ágiles en el desarrollo de software*, elaborado por José H. Canós, Patricio Letelier y Ma Carmen Penadés en el 2003, se destacan en resumen, las principales diferencias entre las metodologías tradicionales y las metodologías influenciadas por la filosofía ágil.

Tabla 1. Diferencias entre metodologías ágiles y no ágiles

Metodologías ágiles	Metodologías tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlados, con pocos principios	Proceso mucho más controlados, con numerosas políticas/normas
No existe contrato tradicional o al menos es	Existe un contrato prefijado

bastante flexible	
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Fuente: (Canós et al., 2003).

Esta filosofía de trabajo ha sido adoptada por una serie de metodologías, que basan sus procesos en el manifiesto ágil y los principios que se desprenden de él. Algunas metodologías que se han destacado a lo largo de los años, son:

- **Adaptive Software Development:** define un ciclo de desarrollo basado en tres actividades: especular, colaborar y aprender (Highsmith, 2013). Este ciclo dinámico incentiva el aprendizaje continuo de los miembros del equipo y la adaptación a los cambios emergentes que se presentan en el proyecto. Las características de un proyecto que cuenta con un ciclo de vida basado en ASD son: enfocado en el objetivo, basado en características, iterativo, impulsado por el riesgo, tiempos fijos, tolerante al cambio (Boehm & Turner, 2004).
- **Dynamic Systems Development Method (DSDM):** es una metodología que define un marco de trabajo para RAD (Rapid Application Development), adicionando levantamiento de requisitos, estudio de viabilidad y estudio de negocio a los pasos que se deben cumplir antes del inicio del proyecto (Paetsch, Eberlein, & Maurer, 2003).
- **Extreme Programming (XP):** sus valores principales son la simplicidad, la comunicación, la retroalimentación y el coraje. Es por esto que propone tener a todo el equipo de trabajo reunido en torno a simples pero efectivas prácticas de desarrollo,

otorgando la suficiente retroalimentación que permita al equipo ver hacia dónde deben dirigirse en el constante cambio que se presenta en los proyectos (Beck & Andres, 2005).

- **Joint Application Development:** se basa en sesiones de trabajo con un enfoque en el análisis estructurado. Durante estas sesiones los clientes y desarrolladores discuten acerca de las características del producto deseado, logrando de esta manera involucrar a los usuarios en el proceso de diseño y construcción del software, agilizando los tiempos de desarrollo y logrando satisfacer a mayor cabalidad las necesidades del negocio (Macaulay, 1996).

- **Metodologías Cristal:** son una familia de diferentes metodologías que se centran en los miembros del equipo de trabajo. Adicionalmente, se enfocan en que la cantidad de artefactos entregados sea mínima. Se basan en un sistema de etiquetas de colores para clasificar el tamaño de los equipos (Cockburn, 2006).

- **Rapid Application Development:** se enfoca en proporcionar velocidad a la construcción del software, mediante el uso de prototipado, enfoque constante, reutilización de componentes y una desformalización de todas las comunicaciones del equipo. Con esto se logra entregar rápidamente valor al negocio. Debido a que acoge los prototipos, como una parte esencial de la metodología, el negocio puede estar constantemente probando las nuevas características, velando así, durante todo el proyecto, por la calidad del producto final (Martin, 1991).

El presente trabajo se centra en equipos que utilizan la metodología Scrum, la más usada de las corrientes ágiles (VersionOne, 2016). Por este motivo, dicha metodología será descrita a mayor profundidad.

1.1 Scrum

Jeff Sutherland y Ken Schwaber crearon el marco de trabajo Scrum en 1995 para presentarlo en la conferencia Oopsla en Austin, Texas (EEUU). Posteriormente publicaron el paper “Scrum Software Development Process” que establece las primeras pautas de la metodología, sus valores, sus principios, sus fases y ceremonias («Scrum History | Scrum Guides», 2017).

Ken y Jeff heredaron el nombre *Scrum* del paper ‘The New New Product Development Game’ escrito por Takeuchi y Nonaka en 1986, en el que el término fue tomado del rugby y describe un tipo de jugada en donde cada equipo se une para apartar al otro y obtener el balón, se usa para demostrar la importancia y el poder de los equipos. La investigación de Nonaka y Takeuchi muestra que un rendimiento sobresaliente en el desarrollo de nuevos y complejos productos es alcanzado, cuando los equipos, pequeñas y auto-organizadas unidades de personas, son alimentados con objetivos y no tareas. Los mejores equipos son aquellos que apuntan a un objetivo común. Los equipos requieren autonomía para alcanzar la excelencia («Scrum History | Scrum Guides», 2017).

Scrum es un marco de trabajo en el que las personas abordan problemas complejos, mientras creativamente entregan productos del más alto valor posible. Scrum no es un proceso o técnica para construir productos, sino que se enfoca en la mejora constante de los equipos de trabajo, apoyándose en otras técnicas o procesos para desarrollar software. Es por esta razón que la metodología está conformada por una serie de roles, ceremonias y artefactos pensados para promover la interacción abierta y la mejora constante de las habilidades de los miembros del equipo (Schwaber & Sutherland, 2013).

1.2 Artefactos

Son los elementos definidos por Scrum para facilitar el proceso de construcción del producto. Se caracterizan por ser simples e intuitivos para que puedan ser comprendidos fácilmente por todos los miembros del equipo (Schwaber, 2004). Los principales artefactos definidos por Nazareno, Leone & Gonnet (2013), son:

Producto: es el objetivo final del proyecto, entregado de manera incremental e iterativa para permitir la retroalimentación constante de los usuarios finales.

Product Backlog: una lista ordenada de todo lo que se necesita para la creación del producto, siendo el único sitio donde se guardan los requerimientos.

Elemento del Backlog: también conocidas como Historias de Usuario, son las necesidades específicas que deben ser satisfechas por el producto de software. Cada Historia de Usuario debe ser desarrollada y probada en su totalidad por el equipo. Cuentan con una descripción clara del

requerimiento, unos criterios mínimos de aceptación y una puntuación que es asignada por el equipo según su dificultad.

Backlog del Sprint: son las funcionalidades, tomadas del Product Backlog, que el equipo desarrollará durante un Sprint.

Incremento del Producto: es el resultado de las tareas ejecutadas durante un Sprint. Se presenta en una de las ceremonias de Scrum llamada Revisión del Sprint, que será descrita posteriormente.

1.3 Roles

Scrum define tres roles principales para las personas que componen los equipos: Product Owner, Development Team y Scrum Master. A continuación se presenta la definición y función de cada uno de estos roles dentro de los equipos Scrum:

Product Owner (PO)

También conocido como dueño de producto, es la persona encargada de maximizar el retorno de la inversión por medio de la identificación de las características clave del producto, convirtiéndolas en una lista priorizada de elementos dentro del Product Backlog, conocida como Product Backlog. La priorización que realiza el PO es esencial para garantizar el éxito del proyecto, lo que exige que el PO comprenda completamente las necesidades de los diferentes *Stakeholders*, permitiéndole definir cuáles necesidades deben satisfacerse primero, porque están alineadas con objetivos estratégicos, atacan riesgos o permiten la mejora del proyecto. Adicionalmente, el PO es fundamental durante la revisión de los Sprints, proporcionando retroalimentación constante que permite la mejora continua del equipo (Deemer et al., 2012).

El PO debe ser una persona, no un comité. El PO puede representar los deseos de un comité en el Product Backlog, pero cualquier persona que quiera cambiar la prioridad del Product Backlog debe dirigirse al PO. Por lo tanto, para que el PO sea exitoso es necesario que toda la organización respete sus decisiones (Schwaber & Sutherland, 2013).

Development Team

Entre cinco y nueve profesionales constituyen el equipo de desarrollo que construye el producto que el Product Owner indica. Ellos determinan cuántas Historias de Usuario, de las entregadas por

el PO, se van a realizar por Sprint (Schwaber & Sutherland, 2013). También pueden ofrecer retroalimentación al PO para mejorar el producto que están creando.

Scrum ha identificado que los equipos más productivos y efectivos son aquellos que se enfocan en un solo producto por Sprint, con miembros que son estables durante todo el proyecto. Este producto es entregado de manera iterativa e incremental, fomentando que exista una retroalimentación constante (Deemer et al., 2012).

Los equipos son auto-organizados e interfuncionales, de manera que son autónomos para seleccionar el método de trabajo que mejor les ayude a lograr su objetivo, en vez de ser dirigidos por personas ajenas, y se garantiza que el equipo tenga todas las competencias necesarias para lograr dicho objetivo sin depender de agentes externos (Schwaber & Sutherland, 2013).

Dentro de los equipos de desarrollo sólo existen *Team Members* (miembros de equipo), haciendo que los títulos fijos, como arquitecto, líder del proyecto o diseñador no sean utilizados. Debido a esto, cada persona puede aportar en tareas que no sean de su habilidad principal, promoviendo el aprendizaje y la colaboración. Por lo tanto, los miembros del equipo trabajan juntos de la manera más apropiada para lograr el objetivo que se han propuesto (Deemer et al., 2012).

Scrum Master (SM)

Es la persona encargada de ayudar al equipo a aprender y aplicar Scrum para alcanzar los objetivos planteados. El SM sirve al equipo removiendo impedimentos, protegiéndolo de interferencia externa y promoviendo las prácticas de desarrollo modernas. Para ser un buen Scrum Master no es necesario contar con una disciplina o carrera específica (Deemer et al., 2012).

De acuerdo con Schwaber & Sutherland (2013), el SM juega un papel fundamental, acompañando al PO y al equipo de desarrollo. Apoya al PO principalmente:

- Encontrando técnicas para un manejo efectivo del Product Backlog.
- Enfatizando en el equipo de desarrollo la necesidad de contar con un Product Backlog claro y conciso.
- Ayudándole a entender cómo se realiza la planeación en un ambiente empírico.

- Aportando herramientas que permitan priorizar el Product Backlog correctamente, maximizando el valor entregado en cada Sprint.

- Facilitando las ceremonias de Scrum.

Apoya al equipo de desarrollo principalmente:

- Enseñando cómo ser auto-organizados y buscando que sea interfuncional.
- Ayudando a crear productos de alto valor.
- Removiendo los impedimentos que frenan el progreso del equipo.
- Entrenando a los miembros del equipo para trabajar adecuadamente en ambientes organizaciones que no han adoptado completamente Scrum.

1.4 Sprint

Es el corazón de la metodología, al ser los ciclos de trabajo que se utilizan para la creación del producto. No deben tener una duración mayor a un mes cada uno y se debe realizar uno después del otro sin pausas (Deemer, Benefield, Larman, & Vodde, 2010). Tienen una duración fija, haciendo que terminen sin importar si las historias asignadas en el Sprint Backlog están incompletas (Mahnich & Drnovscek, 2005).

Dentro de cada Sprint se realizan una serie de ceremonias que constituyen el proceso de desarrollo basado en Scrum. Las ceremonias son las siguientes:

Planeación del Sprint: es la ceremonia con la que se inicia el Sprint, en ella participa todo el equipo. Durante esta ceremonia se definen cuántas y cuáles Historias de Usuario van a ser ejecutadas durante el Sprint, dependiendo de la capacidad de desarrollo del equipo (Cervone, 2011).

Daily: es una reunión corta, no más de 15 minutos, que se realiza diariamente para evaluar dentro de cada Sprint el avance sobre las historias definidas en la planeación (Schwaber & Beedle, 2002). Normalmente cada uno de los miembros del equipo se hace tres preguntas que ayudan a entender el estado actual del Sprint, las preguntas son:

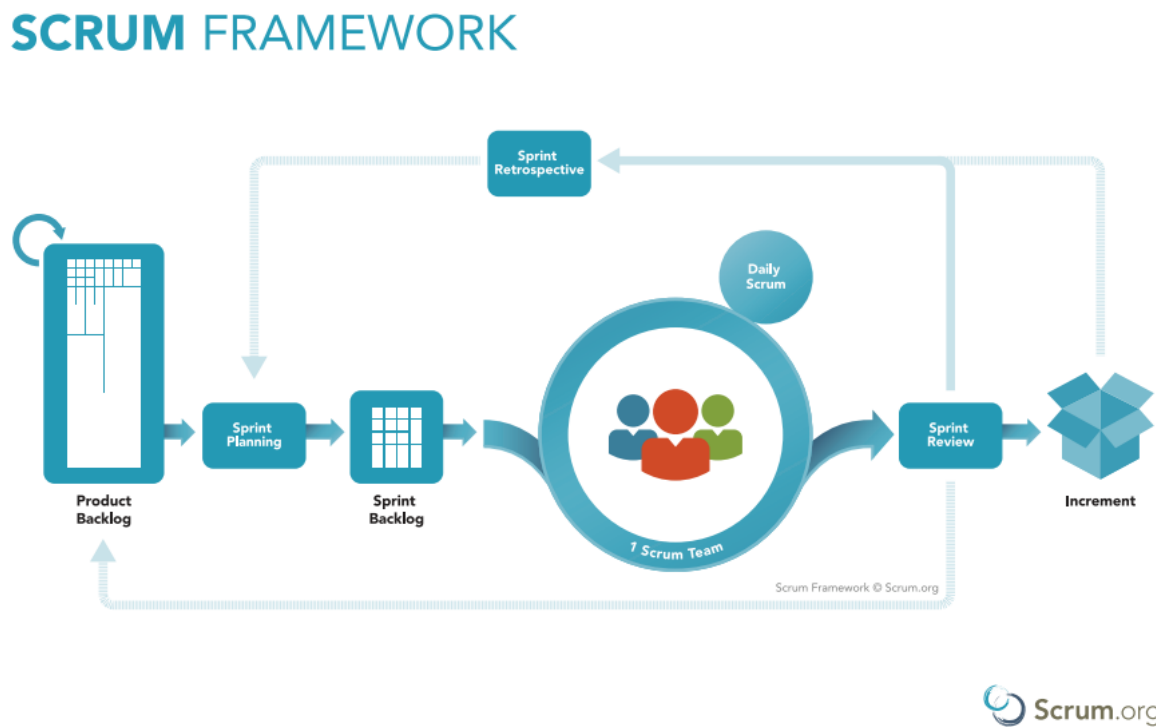
- ¿Qué hice ayer?
- ¿Qué voy a hacer hoy?
- ¿Qué impedimentos tengo para cumplir mi meta del Sprint?

Revisión del Sprint: se realiza al finalizar el Sprint. En esta reunión el equipo de desarrollo presenta el resultado del trabajo realizado durante el Sprint, para que el Product Owner valide que las funcionalidades pueden pasar a ser utilizadas por los usuarios finales. Adicionalmente, se realiza una revisión del Backlog del producto para identificar qué funcionales deben ser completamente definidas para poderse trabajar el siguiente Sprint (Schwaber & Sutherland, 2013).

Retrospectiva del Sprint: es la ceremonia que ayuda a realizar mejoras al interior del equipo. Se realiza después de la revisión del Sprint y está enfocada en identificar qué cosas se deben cambiar, qué cosas se deben mejorar y qué cosas nuevas se van a implementar en el equipo. Con esta información recolectada, se crea un plan de mejoras enfocadas en permitir entregar mayor valor en los posteriores Sprints (Deemer et al., 2012).

En la figura 1 se visualiza una representación gráfica de las interacciones que existen durante un Sprint, entre las ceremonias, los diferentes roles y los artefactos definidos en Scrum.

Figura 1. Marco de trabajo Scrum



Fuente: (Scrum.org, 2016).

1.5 Aprendizaje

No existe consenso general respecto a la definición del aprendizaje como concepto, lo que existe en realidad son algunas definiciones que parecen ser más aceptadas por la comunidad científica que otras. Para la elaboración de este trabajo se utilizará la definición de Shuell (1986), en la que el aprendizaje es un cambio relativamente permanente en la conducta de un individuo, en tanto sus asociaciones y sus representaciones mentales aparecen como resultado de la experiencia. La definición de Shuell se puede complementar con la definición de Langley & Simon (1981), según la cual, el aprendizaje es cualquier proceso que modifica un sistema, de manera más o menos irreversible, haciendo que el mismo grupo de individuos mejore la posterior realización de la misma tarea.

De acuerdo con Shuell, se puede decir que el aprendizaje cuenta con tres criterios (citado en Schunk, 2012).

- El aprendizaje implica un cambio.
- El aprendizaje perdura a lo largo del tiempo.
- El aprendizaje ocurre por medio de la experiencia.

El primer criterio muestra que las personas aprenden cuándo pueden realizar acciones nuevas o de una manera diferente, implicando por lo tanto, que el aprendizaje es inferencial y no es posible observarlo de manera directa, sino a través de sus productos o resultados. El segundo criterio demuestra que el aprendizaje puede no ser permanente, debido a que las personas pueden olvidar con el paso del tiempo, abriendo el debate respecto al tiempo que deben durar los cambios en los individuos para ser clasificados como aprendizaje. El tercer criterio define el aprendizaje como algo que se adquiere, por ejemplo, practicando u observando las acciones de los demás, haciendo que se excluyan del aprendizaje los cambios de conducta determinados por la herencia (Schunk, 2012).

Las personas aprenden por diferentes razones, cómo obtener recompensas externas, alcanzar una sensación de triunfo o hacer más fácil su vida, consiguiendo no sólo habilidades y conocimiento sino también valores, actitudes y reacciones emocionales. Las recompensas son importantes durante el aprendizaje, porque hacen que la persona preste más atención a la información que tiene que aprender. Sin embargo, cuando una situación concreta llama la atención de un individuo hacia lo que debe aprender, se puede producir aprendizaje sin requerir un estímulo externo (Ormrod, 2005).

Existen dos grupos principales de teorías en este amplio concepto de aprendizaje, el conductismo y el cognitivismo. El primer grupo se refiere al cambio externo en la conducta que se puede palpar y observar, por ejemplo, atarse los zapatos, resolver correctamente un problema aritmético o hacerse el **enfermo** para no ir al colegio. El segundo, se centra en el cambio en las representaciones o asociaciones mentales, siendo este un cambio interno que no podemos ver, haciendo que las teorías cognitivistas no se centren en la conducta, sino en los procesos de pensamiento implicados en el aprendizaje humano. Algunos ejemplos de este último grupo de teorías pueden ser el hecho de encontrar la relación entre la multiplicación y la división o utilizar trucos mnemotécnicos para recordar vocabulario de otra lengua (Ormrod, 2005). Ambas teorías tienen perspectivas diferentes para influenciar el aprendizaje, haciendo que las teorías conductistas se enfoquen en cambiar el entorno para estimularlo y las teorías cognitivas se enfoquen en cambiar a la persona que está aprendiendo (Shuell, 1986). Sin embargo, en cualquiera de las dos perspectivas, se puede apreciar que ha ocurrido un aprendizaje cuando se observa un cambio en la conducta de una persona, apareciendo una nueva respuesta o incrementando la frecuencia de una que ya existía (Ormrod, 2005).

Para aclarar más el concepto de las teorías conductuales, es factible decir que son las que plantean que aprender consiste en formar asociaciones entre estímulos y respuestas que, según la perspectiva de Skinner (1953), citada en Shuell (1986), una respuesta a un estímulo puede que se repita en el futuro dependiendo de las consecuencias de las respuestas previas, haciendo que el reforzamiento aumente la probabilidad de que se repita una respuesta, mientras el castigo reduce esa probabilidad. Un ejemplo de esto se da cuando a un niño se le regala un billete cada vez que se amarre los zapatos correctamente y se le castiga quitando algo que le guste cada vez que no lo haga, logrando que el estímulo sea suficiente para conseguir que aprenda a atarse los zapatos sin ayuda.

Por su parte, las teorías cognitivistas resaltan el proceso de adquisición de conocimiento y habilidades, la formación de estructuras mentales y el procesamiento de la información y las creencias, haciendo que desde su perspectiva, el aprendizaje sea considerado como un fenómeno mental interno que se infiere a partir de lo que la gente dice y hace. Uno de sus temas centrales se basa en mostrar cómo realiza la mente el procesamiento de la información, buscando explicar cómo construye, adquiere, organiza, codifica, repite y almacena una persona los datos que aprende. Este tipo de teorías hacen especial énfasis en la función de los pensamientos, las creencias, las

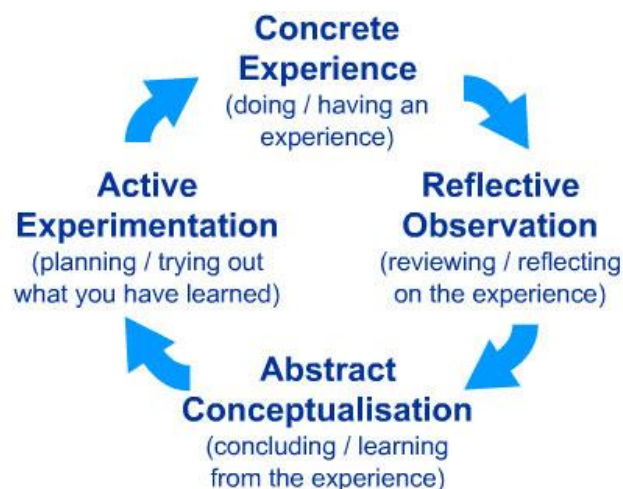
actitudes y los valores de la persona que está aprendiendo. Esto muestra que los individuos que dudan de su capacidad para aprender retardan el aprendizaje por falta de dedicación o entusiasmo (Shuell, 1986).

1.6 Teoría de aprendizaje experiencial

La teoría de aprendizaje experiencial (ELT por sus siglas en inglés), define al aprendizaje como “el proceso por el cual el conocimiento es creado a través de la transformación de la experiencia. El conocimiento es el resultado de combinar el acaparamiento y la transformación de la experiencia” (D. A. Kolb, 1984, p. 37), destacando que el aprendizaje es el mayor determinante del desarrollo humano y que la manera como los individuos aprenden moldea el curso de su desarrollo personal (A. Y. Kolb, 2005). Se utiliza el término “experiencial” para diferenciar esta teoría de las teorías cognitivas, que enfatizan el proceso cognitivo sobre los estímulos, y de las teorías de comportamiento, que indican que la experiencia subjetiva no tiene influencia en el proceso de aprendizaje (D. A. Kolb, Boyatzis, Mainemelis, & Others, 2001).

En el modelo ELT se define que las personas cuentan con dos modos para acaparar experiencia: la Experiencia Concreta (CE por sus siglas en inglés) y la Conceptualización Abstracta (AC por sus siglas en inglés) y dos para transformarla, la Observación Reflexiva (RO por sus siglas en inglés) y la Experimentación Activa (AE por sus siglas en inglés) (D. A. Kolb et al., 2001). Estos modos se relacionan en el ciclo de aprendizaje de cuatro etapas de Kolb, descrito en la figura 2.

Figura 2. Ciclo de aprendizaje de cuatro etapas de Kolb



Fuente: (McLeod, 2010).

Un aprendizaje efectivo se logra cuando la persona puede pasar por cada uno de los pasos del ciclo de aprendizaje, viviendo una experiencia, reflexionando sobre lo vivido, analizando y concluyendo de las reflexiones hechas y, por último, actuando, basado en las conclusiones logradas para tener nuevas experiencias (A. Y. Kolb, 2005). Por lo tanto, ELT sugiere que el aprendizaje requiere habilidades que son completamente opuestas, y por ende, la persona debe seleccionar cuáles usar para cada situación de aprendizaje a la que se enfrente. Es por esto que las personas desarrollan una preferencia por un modo de aprendizaje, basándose en su composición genética, sus experiencias de vida o las demandas de su ambiente. Esta preferencia se muestra como un patrón característico, al que se le denomina “estilo de aprendizaje” (D. A. Kolb et al., 2001).

La investigación actual ha mostrado que ETL es útil para entender el aprendizaje de los equipos y su eficiencia (Kayes, Kayes, & Kolb, 2005). Según D. A. Kolb & Kolb (2013), varios estudios demuestran que un equipo es más efectivo si aprende de la experiencia y hace énfasis en los cuatro estilos de aprendizaje. En esta información se basa el presente trabajo, que evalúa cómo están conformados los equipos de una empresa aseguradora de Medellín.

1.7 El inventario de estilos de aprendizaje

En 1971 David Kolb desarrolló el Inventario de Estilos de Aprendizaje (LSI por sus siglas en inglés). Este inventario fue creado para cumplir con dos propósitos:

1. “Para servir como una herramienta educativa para incrementar el entendimiento de los individuos del proceso de aprendizaje basado en la experiencia” (D. A. Kolb & Kolb, 2013, p. 39). Al enseñarle a la persona cómo es el proceso de aprendizaje por experiencia, esta persona adquiere una mayor capacidad para realizar un control metacognitivo de su proceso de aprendizaje, haciendo que pueda seleccionar conscientemente el estilo de aprendizaje que más se acomode a sus necesidades en una situación determinada. Por esta razón, los resultados de la prueba de estilos de aprendizaje no deben ser tomados como definitivos, sino como un punto de inicio para entender cuál es la forma más cómoda de aprendizaje que tiene el individuo (D. A. Kolb & Kolb, 2013).

2. “Para proveer una herramienta de investigación que apoye la teoría de aprendizaje experiencial y las características de los estilos de aprendizaje individuales” (D. A. Kolb & Kolb, 2013, p. 39).

Por estos motivos, el LSI es construido como un ejercicio de autoevaluación y una herramienta para la validación de ELT, que mide el grado en el cual los individuos exhiben diferentes estilos de aprendizaje (A. Y. Kolb, 2005).

Este inventario cuenta con varias versiones y revisiones que ha realizado su autor a través de los años, encontrándose actualmente en la versión 4.0. El presente trabajo se realiza utilizando la versión 1.0 que consta de un corto cuestionario (9 preguntas), en las que se le pide al encuestado clasificar de mayor a menor cuatro frases que correspondan a los cuatro modos de aprendizaje descritos en la teoría de aprendizaje experiencial.

Este proceso de clasificación ha encontrado que las personas pueden ser clasificadas en cuatro estilos de aprendizaje, que son asociados a los diferentes modos de ELT. Estos estilos de aprendizaje son:

Divergente: persona que tiene la experiencia concreta y la observación reflexiva (CE y RO), como sus habilidades de aprendizaje dominantes, haciendo que el individuo tenga facilidad de observar una situación concreta desde diferentes puntos de vista (A. Y. Kolb, 2005). Prefieren trabajar en grupos para poder obtener información de varias fuentes y poder escuchar con mente abierta. No son buenos ejecutores debido a su preferencia por la observación y la reflexión (Seel, 2011).

Asimilador: persona que tiene la conceptualización abstracta y la observación reflexiva (AC y OR), como sus habilidades de aprendizaje dominantes, haciendo que el individuo pueda entender un amplio rango de información y la pueda clasificar de una manera lógica y concisa (A. Y. Kolb, 2005). Les interesa en menor medida la interacción con las personas, prefiriendo las ideas abstractas y los conceptos, generando que prefieran trabajar solos y que no tomen decisiones apresuradas (Seel, 2011).

Convergente: persona que tiene la conceptualización abstracta y la experimentación activa (AC y AE), como sus habilidades de aprendizaje dominantes, haciendo que el individuo encuentre fácilmente usos prácticos para las ideas y las teorías (A. Y. Kolb, 2005). Prefieren resolver problemas técnicos y no son buenos enfrentándose a problemas sociales (Seel, 2011).

Acomodador: persona que tiene la experimentación activa y la experiencia concreta (AE y CE), como sus habilidades de aprendizaje dominantes, haciendo que el individuo aprenda de experiencias en las que tiene que interactuar activamente (A. Y. Kolb, 2005). Para resolver problemas, prefieren consultar a otras personas para obtener información. Debido a que están centrados en la acción y no en la reflexión, tienden a realizar acciones antes de pensar (Seel, 2011).

A continuación se presenta una tabla con la relación que existe entre los estilos de aprendizaje y cinco diferentes niveles de comportamiento.

Tabla 2. Relación entre los estilos de aprendizaje y cinco niveles de comportamiento

Nivel de conducta	Divergente	Asimilador	Convergente	Acomodador
Tipos de personalidad	Introvertido Emotivo	Introvertido Intuitivo	Extrovertido Pensador	Extrovertido Sentimental
Especialización educativa	Artes, lengua, historia, psicología	Matemáticas, física pura	Ingeniería, medicina	Educación, comunicación, enfermería
Carrera profesional	Servicios sociales, artes	Ciencia, investigación, información	Ingeniería, medicina, tecnología	Ventas, servicios sociales, educación
Trabajos actuales	Trabajos con personas	Trabajos de información	Trabajos técnicos	Trabajos ejecutivos
Competencias adaptativas	Habilidades de valoración	Habilidades de análisis	Habilidades de decisión	Habilidades de acción

Fuente: (D. A. Kolb & Kolb, 2013).

2. Metodología

El presente trabajo se dividió en dos fases: investigación documental y realización de un censo de estilos de aprendizaje, en una de las aseguradoras más grandes de Colombia. Durante la investigación documental se logró definir dos conceptos claves para el estudio, el aprendizaje y la metodología Scrum.

El concepto de aprendizaje fue definido desde la perspectiva de diferentes autores, encontrando que al día de hoy no se ha creado una tesis que sea aceptada en consenso por los investigadores

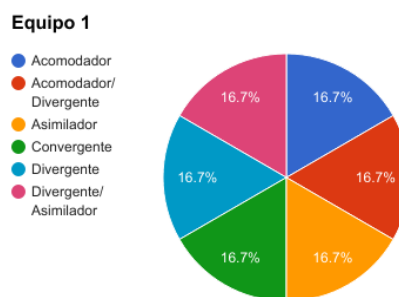
del área, sin embargo, para el presente trabajo se utilizó la definición de Shuell (1986), que describe al aprendizaje como un cambio en el individuo por sus experiencias vividas.

Posteriormente, se realizaron indagaciones en las principales fuentes literarias, para lograr definir qué son las metodologías ágiles de desarrollo de software, cuáles son sus premisas y principios y cuáles han sido los principales marcos de trabajo que se basan en este concepto. Debido a que este trabajo se centra en equipos que implementan la metodología ágil, más ampliamente utilizada en la industria del desarrollo de software (VersionOne, 2016), Scrum, se la definió con especial detalle para alcanzar un amplio entendimiento de sus características.

Al finalizar la recolección de todo el marco teórico, se realizó un censo, utilizando la versión 1 del inventario de estilos de aprendizaje definidos por D. A. Kolb (1984). Este censo fue realizado entre los meses de marzo y abril del año 2017, tomando como población a los *Team Members* de los equipos Scrum de la de las principales aseguradoras de Colombia, cuya sede principal se encuentra en la ciudad de Medellín, departamento de Antioquia. A los participantes se les pidió que realizaran un cuestionario que se encontraba en una página web, creada para apoyar el presente trabajo. Dentro del cuestionario se especificó las características de las preguntas y se les indicó que todas las respuestas serían utilizadas sólo con fines académicos.

2.1 Resultados

Gráfico 1. Equipo 1

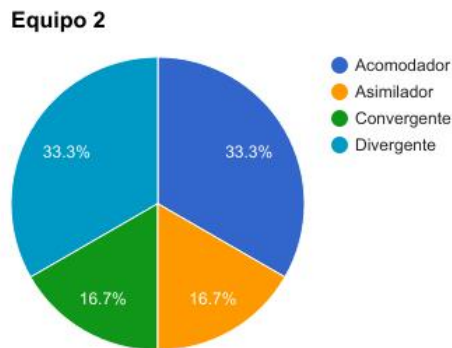


Fuente: Elaboración propia, 2017.

El primer equipo evaluado está conformado por seis personas, dos de ellas sin una clara diferenciación entre dos de los estilos de aprendizaje y, por lo tanto, pueden hacer uso de ambos

estilos, dependiendo de la situación a la que se enfrenten. Adicionalmente, este equipo presenta una distribución completamente heterogénea, logrando cumplir con las recomendaciones de D. A. Kolb & Kolb (2013).

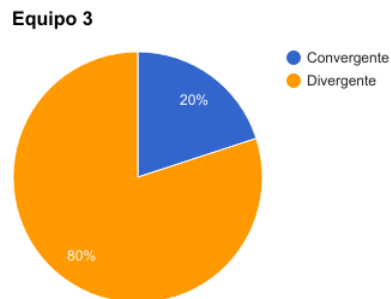
Gráfico 2. Equipo 2



Fuente: Elaboración propia, 2017.

El segundo equipo está conformado por seis personas, teniendo dos acomodadores, dos divergente, un convergente y un asimilador. Este equipo también se encuentra cumpliendo con las recomendaciones de D. A. Kolb & Kolb (2013). En este equipo se empieza a encontrar una preferencia por el estilo divergente.

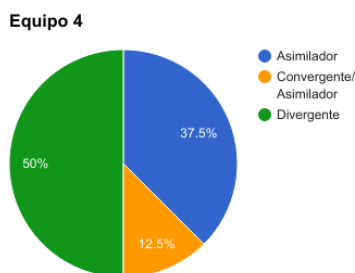
Gráfico 3. Equipo 3



Fuente: Elaboración propia, 2017.

El tercer equipo evaluado, junto con el equipo seis, son los más pequeño de los seis censados, sólo contando con cinco miembros. Tiene una marcada preferencia por el estilo divergente, con cuatro de sus miembros clasificándose en esta categoría. El miembro restante hace parte de los convergentes, haciendo de éste un equipo mal formado bajo la luz de las recomendaciones y hallazgos de D. A. Kolb & Kolb (2013).

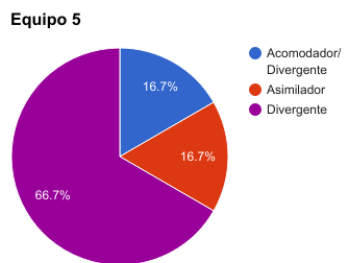
Gráfico 4. Equipo 4



Fuente: Elaboración propia, 2017.

El cuarto equipo cuenta con ocho personas, haciéndolo el más grande de los equipos Scrum de la organización. A pesar de ser numeroso, no cumple con las recomendaciones de D. A. Kolb & Kolb (2013) y la mitad de sus miembros están clasificados como divergentes. Otra particularidad de este equipo es que cuenta con un miembro que se puede desenvolver como convergente o asimilador, según la situación a la que se enfrente, ya que muestra igual preferencia por ambos estilos de aprendizaje. Los otros tres miembros restantes se encuentran clasificados como asimiladores.

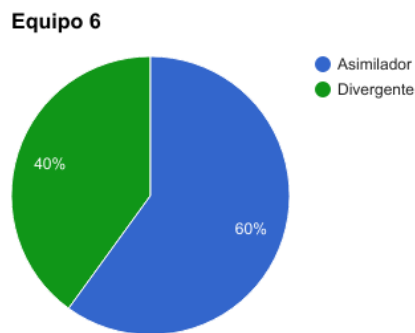
Gráfico 5. Equipo 5



Fuente: Elaboración propia, 2017.

El quinto equipo está conformado por seis personas, con cuatro de sus miembros clasificados como divergentes. Adicionalmente, cuenta con un miembro que puede desempeñar el rol de acomodador y divergente, por sus preferencias por ambos estilos. El miembro restante es un asimilador. Debido a la ausencia del estilo convergente dentro del equipo, se puede decir que no se está cumpliendo con las recomendaciones de D. A. Kolb & Kolb (2013).

Gráfico 6. Equipo 6



Fuente: Elaboración propia, 2017.

El sexto y último equipo está conformado por cinco personas, de igual tamaño que el equipo tres. Es el único equipo de los que no cumple con las recomendaciones de D. A. Kolb & Kolb (2013), que está conformado en su mayoría por asimiladores y no divergentes.

Gráfico 7. Todos los equipos



Fuente: Elaboración propia, 2017.

En este gráfico se puede encontrar la distribución de los estilos de aprendizaje de las treinta personas censadas en la organización. Se nota una fuerte tendencia por el estilo divergente, con el 47.2% de los encuestados ubicándose en esta categoría. El estilo de aprendizaje que le sigue a los divergentes es el asimilador, con el 25% de las personas. Esto muestra que en sólo dos de los cuatro estilos de aprendizaje se puede clasificar al 72.2% de los miembros de los equipos Scrum de la organización, evidenciando que las recomendaciones de D. A. Kolb & Kolb (2013) no son un factor relevante en los procesos de selección y conformación de equipos en esta empresa aseguradora.

2.2. Hallazgos finales

Los seis equipos censados cuentan con distribuciones diferentes de sus estilos de aprendizaje, sin embargo, sólo dos de ellos presentan los cuatro estilos simultáneamente. Las principales características de cada uno de los equipos censados son: el primer equipo cuenta con todos los estilos de aprendizaje dentro de sus miembros; el segundo es de similares distribuciones a las del primero, conformado por todos los estilos; el tercero sólo tiene cinco miembros y cuenta únicamente con los estilos divergente y convergente; el cuarto es el equipo más numeroso con ocho miembros, la mayoría de ellos clasificados dentro del estilo divergente; el quinto sigue la línea de tendencia de los equipos, contando con la mayoría de sus miembros en la categoría divergente; el sexto está conformado por cinco miembros, distribuidos en sólo dos de los cuatro estilos de aprendizaje, siendo peculiar en este equipo que la mayor parte de sus miembros son asimiladores y no divergentes.

El amplio bagaje investigativo del profesor David Kolb, iniciado en la década de los 80, y continuado por él y sus seguidores hasta el presente, ha permitido acopiar información relevante que hace presumir sólidamente que los equipos de trabajo encuentran mayor desempeño cuando están conformados por personas cuyos estilos de aprendizaje son diferentes y complementarios. En los equipos Scrum censados, en una de las más grandes empresas aseguradoras colombianas, pudo verificarse, a partir del uso de los instrumentos diseñados por el grupo de investigación del profesor Kolb, que sólo la tercera parte de dichos equipos están conformados por personas de estilos de aprendizaje complementarios y que la mayoría de las personas censadas se encuentran clasificadas dentro de la categoría divergente, haciendo que la conformación de los equipos esté alejada del modelo planteado por la teoría de estilos de aprendizaje.

2.3 Futuras investigaciones

Ulteriores investigaciones podrían enfocarse en examinar las relaciones posibles entre la complementariedad de los estilos de aprendizaje y el rendimiento, eficacia y eficiencia de los equipos Scrum.

Referencias

- Agile Manifesto (2001). *Principios del Manifiesto Ágil*. Recuperado el 30 de marzo de 2017, de <http://agilemanifesto.org/iso/es/principles.html>
- Beck, K., & Andres, C. (2005). *Extreme Programming Explained: Embrace Change*. Boston, MA, USA: Addison-Wesley.
- Boehm, B. W., & Turner, R. (2004). *Balancing Agility and Discipline: A Guide for the Perplexed*. Boston, MA, USA: Addison-Wesley Professional.
- Bootla, P., Rojanapornpun, O., & Mongkolnam, P. (2015). Necessary skills and attitudes for development team members in Scrum: Thai experts' and practitioners's perspectives. En *2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE)* (pp. 184-189).
- Canós, J., Letelier, P., & Penadés, M. C. (2003). Metodologías Ágiles en el Desarrollo de Software. En P. Letelier & E. A. Sánchez (Eds.), *VIII Jornadas de Ingeniería del Software y Bases de Datos, JISBD 2003*. Valencia, España: Universidad Politécnica de Valencia.
- Cervone, H. F. (2011). Understanding agile project management methods using Scrum. *OCLC Systems & Services: International digital library perspectives*, 27(1), 18-22.
- Cockburn, A. (2006). *Agile Software Development: The Cooperative Game*. Boston, MA, USA: Pearson Education.
- Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2010). The Scrum Primer. *The Scrum*

- Primer*. Recuperado el 20 de febrero de 2017 de
<http://www.brianidavidson.com/agile/docs/scrumprimer121.pdf>
- Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2012). A Lightweight Guide to the Theory and Practice of Scrum. *Scrum Primer*. Recuperado el 20 de febrero de 2017 de
<http://scrumprimer.org/scrumprimer20.pdf>
- Fowler, M., & Highsmith, J. (Ago 2001). The Agile Manifesto.pdf. *Software Development Magazine*. Recuperado el 15 de febrero de 2017 de
http://dmsboiv.uqac.ca/8INF851/web/part1/introduction/The_Agile_Manifesto.pdf
- Highsmith, J. (2013). *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. Boston, MA, USA: Addison-Wesley.
- Kayes, A. B., Kayes, D. C., & Kolb, D. A. (2005). Experiential learning in teams. *Simulation & gaming*, 36(3), 330-354.
- Kolb, A. Y. (2005). The Kolb learning style inventory-version 3.1 2005 technical specifications. Boston, MA: *Hay Resource Direct*, 200, 72.
- Kolb, D. A. (1984). *Experiential Learning: Experience as the Source of Learning and Development*. Englewood Cliffs, NJ, USA: Prentice-Hall.
- Kolb, D. A., Boyatzis, R. E., Mainemelis, C., & Others. (2001). Experiential learning theory: Previous research and new directions. *Perspectives on thinking, learning, and cognitive styles*, 1(2001), 227-247.
- Kolb, D. A., & Kolb, A. Y. (2013). *The Kolb Learning Style Inventory 4.0: Guide to Theory, Psychometrics, Research & Applications*. Kaunakakai, HI, USA: Experience Based Learning Systems.
- Langley, P., & Simon, H. A. (1981). The central role of learning in cognition. *Cognitive skills*

and their acquisition, 361-380.

Licorish, S., Philpott, A., & MacDonell, S. G. (2009). Supporting Agile Team Composition: A Prototype Tool for Identifying Personality (In)Compatibilities. En *Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering* (pp. 66-73). Washington, DC, USA: IEEE Computer Society.

Macaulay, L. (1996). *Requirements Engineering*. London, UK: Springer.

Mahnic, V., & Drnovscek, S. (2005). Agile Software Project Management with Scrum. En *EUNIS 2005 Conference-Session papers and tutorial abstracts*. Ljubljana, Slovenia.

Martin, J. (1991). *Rapid Application Development*. New York, NY, USA: Macmillan Publishing Company.

McLeod, B. S. (2010). *Kolb's Learning Styles and Experiential Learning Cycle | Simply Psychology*. Recuperado 6 de abril de 2017, de <https://www.simplypsychology.org/learning-kolb.html>

Nazareno, R., Leone, H. P., & Gonnet, S. M. (2013). Trazabilidad de procesos ágiles: un modelo para la trazabilidad de procesos scrum. En *XVIII Congreso Argentino de Ciencias de la Computación*. Recuperado el 12 de febrero de 2017 de http://sedici.unlp.edu.ar/bitstream/handle/10915/32423/Documento_completo.pdf?sequence=1

Ormrod, J. E. (2005). *Aprendizaje humano*. Madrid, España: Pearson Educación.

Paetsch, F., Eberlein, A., & Maurer, F. (2003). Requirements engineering and agile software development. En *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003*. (pp. 308-313).

Schunk, D. H. (2012). *Learning Theories: An Educational Perspective*. Boston, MA, USA:

Pearson.

Schwaber, K. (2004). *Agile Project Management with Scrum*. Redmond, WA, USA: Microsoft Press.

Schwaber, K., & Beedle, M. (2002). *Agile Software Development with Scrum*. Saddle River, NJ, USA: Prentice Hall.

Schwaber, K., & Sutherland, J. (2013). *The Scrum Guide*. Recuperado el 20 de febrero de 2017 de <http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf>

Scrum History | Scrum Guides (2017). Recuperado 31 de marzo de 2017, de <http://www.scrumguides.org/history.html>

Seel, N. M. (2011). *Encyclopedia of the Sciences of Learning*. (N. M. Seel, Ed.). New York, NY, USA: Springer Science & Business Media.

Shuell, T. J. (1986). Cognitive Conceptions of Learning. *Review of educational research*, 56(4), 411-436.

VersionOne (2016). *10th Annual State of Agile Report*. Recuperado el 30 de marzo de 2017, de <https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf>