



Universidad Abierta Interamericana - UAI

Facultad de Ciencias Económicas

Licenciatura en Ingeniería Comercial

“La gestión empresarial y su integración con las metodologías ágiles de desarrollo de software”

Romaniuk Correa, Andrés Abelardo

Diciembre 2018

AGRADECIMIENTOS

A mi novia Samanta por la motivación para lograr este objetivo, a mis padres por su apoyo en todo momento, a mis profesores, a mis compañeros de estudio y trabajo.

¡Muchas gracias!

ÍNDICE

Capítulo 1: Introducción y Objetivos.....	4
1.1 Introducción.....	4
1.2 Objetivos.....	6
Capítulo 2: Metodologías ágiles de desarrollo.....	7
2.1 Agilidad.....	7
2.2 Manifiesto Ágil.....	8
2.3 SCRUM - Fundamentos.....	10
2.4 Metodologías tradicionales y ágiles.....	17
2.4.1 Ventajas de las metodologías tradicionales.....	19
2.4.2 Desventajas de las metodologías tradicionales.....	19
2.4.3 Ventajas de las metodologías ágiles.....	20
2.4.4 Desventajas de las metodologías ágiles.....	20
2.4.5 Elección de las metodologías a utilizar.....	21
Capítulo 3: Integración de las metodologías ágiles en las organizaciones.....	22
3.1 De la gestión empresarial tradicional hacia la ágil	22
3.2 Nuevos roles en las organizaciones	23
3.3 Integración por indicadores de gestión.....	24
3.4 Implementación de metodologías ágiles en una organización tradicional.....	26
Conclusiones.....	29
Bibliografía.....	31

Capítulo 1 – Introducción y Objetivos

1.1 Introducción

Las organizaciones actualmente están conviviendo en un contexto de permanente cambio y se enfrentan a rápidamente a nuevos desafíos. Durante las dos últimas décadas se han presentado marcados cambios en las tendencias de los mercados, en las competencias, en las tecnologías, en la cultura y en la sociedad.

Para lograr maximizar sus beneficios, las empresas en la actualidad deben responder ágil y eficientemente a las exigencias que el contexto presenta, minimizando el time to market, buscando ventajas competitivas y el desarrollo económico a largo plazo.

La implementación de Tecnologías de Información y Comunicación (TIC) en una empresa, puede proporcionar una ventaja competitiva o no, dependiendo de si se realiza el análisis de los factores afectados por la implementación de un nuevo producto o metodología tecnológica, para saber cuál debe ser la estrategia que debe adoptar la organización frente al cambio, pero cierto y fundamental es que la gestión empresarial vaya de la mano con la evolución de la sociedad, los avances tecnológicos y la globalización.

Esta situación impacta muy especialmente al desarrollo de software que actualmente es el pilar que sustenta y aporta valor al resto de sectores industriales y ha sido uno de los factores con mayor peso en la transformación de la economía mundial (Software Engineering-Pfleeger,2010).

En el actual entorno turbulento, la ingeniería de software ha reflexionado sobre el grado de adaptabilidad de las metodologías utilizadas en el desarrollo de sistemas software hasta el momento, conocidas -frente a otras nuevas corrientes- como metodologías tradicionales. ¿Se adaptan bien las metodologías convencionales a las actuales características de dinamismo y variabilidad del contexto? Kent Beck, uno de los padres del modelo ágil, ha escrito:

“Cada cosa en software cambia. Los requisitos cambian, el diseño cambia, el negocio cambia, la tecnología cambia, el equipo cambia y los miembros del equipo cambian. El problema no es el cambio, porque el cambio, inevitablemente, ocurrirá. El problema, realmente, es nuestra inhabilidad para hacer frente a los cambios”.

Si extrapolamos la afirmación anterior al campo de la gestión empresarial, podemos observar que lo dicho también puede tomarse como válido. Y ahora nos preguntamos: ¿Se adaptan bien los métodos de gestión empresarial a las actuales características de dinamismo y variabilidad del contexto digital?

Durante el desarrollo de este trabajo se intentará responder a este interrogante analizando las características de las variables que inciden en ella.

El presente trabajo se realiza a fin de exponer un análisis resultante de la investigación documental acerca de la gestión empresarial y la integración con las actuales metodologías de desarrollo de software, finalizando con una conclusión de lo investigado.

Los motivos por los cuales se ha llevado adelante la investigación están relacionados al interés y a la constante actualización de conocimiento que ello requiere y motiva.

1.2 Objetivos

Objetivo general:

- Analizar y describir las tareas, roles y habilidades necesarias en una organización tradicional para poder llevar adelante proyectos de desarrollo de software aplicando metodologías ágiles mediante *SCRUM*.

Objetivos específicos:

- Describir las características de las metodologías ágiles de desarrollo *SCRUM*.
- Describir la comparación entre metodologías ágiles y metodología tradicional de desarrollo.
- Describir técnicas de gestión empresarial entre las distintas áreas de una organización tradicional para la exitosa implementación de metodologías ágiles de desarrollo.

Capítulo 2 – Metodologías Ágiles de desarrollo

Las metodologías ágiles son uno de los temas más recientes en la ingeniería de software y actualmente cuenta con gran protagonismo en la industria, despertando interés y un poco de debate.

Durante los años 90, se comenzó a concebir la definición de desarrollo ágil de software como reacción contra las metodologías aplicadas hasta el momento, por considerarse demasiado rígidas en su carácter normativo y requerir de cuantiosa documentación de planificaciones detalladas previas al desarrollo.

En el año 2001, diecisiete miembros destacados de la comunidad de software se reunieron en Utah (EE. UU) y adoptaron el nombre de “Metodologías ágiles” para denominar a esta nueva corriente de desarrollo. Poco después, algunos de estos miembros formaron la conocida como “Alianza ágil”, una organización sin fin de lucro que tiene como objetivo promover el desarrollo ágil de aplicaciones. Desde ese momento hasta la actualidad las metodologías ágiles han ido adquiriendo gran auge dentro de la industria software y las organizaciones comienzan a apostar por este nuevo enfoque para desarrollar sus sistemas.

Las metodologías ágiles han revolucionado las áreas de tecnología de la información, aumentando considerablemente las tasas de éxito en el desarrollo de software, mejorando la calidad y la velocidad del time to market como también la motivación y la productividad de los equipos de TI (Standish Group Chaos Studies 2013-2017).

2.1 Agilidad

Es necesario que, al estar utilizando el concepto de agilidad para definir estas metodologías, definamos agilidad desde una perspectiva de software. En el consenso de varias definiciones, Qumer y Henderson, especialistas en tecnología, ofrecieron la siguiente definición de agilidad:

“La agilidad es un comportamiento persistente o habilidad, de entidad sensible, que presenta flexibilidad para adaptarse a cambios, esperados o inesperados, rápidamente; persigue la duración más corta en tiempo; usa instrumentos económicos, simples y de calidad en un ambiente dinámico; y utiliza los conocimientos y experiencia previos para aprender tanto del entorno interno como del externo.”

Por lo tanto, el desarrollo ágil no impone específicamente procesos o métodos a seguir, aunque se han desarrollado algunas prácticas asociadas a este concepto.

El desarrollo ágil es conceptualmente una filosofía de desarrollo de software y su comienzo fundamental fue establecido en las ideas del Manifiesto Ágil resultante de la reunión en Utah, el cual es un documento que resume tal filosofía en 4 valores y 12 principios.

2.2 Manifiesto Ágil

Tal como describe el Manifiesto Ágil, se valora:

Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente es el principal factor de éxito de un proceso software. Este primer valor enuncia que es preferible utilizar un proceso indocumentado con buenas interacciones personales que un proceso documentado con interacciones hostiles. Se considera que no se debe pretender construir primero el entorno y esperar que el equipo se adapte automáticamente sino al revés, construir primero el equipo y que éste configure su propio entorno. El talento, la habilidad, la capacidad de comunicación y de tratar con personas son características fundamentales para los miembros de un equipo ágil.

Desarrollar software que funcione por encima de una completa documentación. El propósito de este valor es acentuar la supremacía del producto por encima de la documentación. El objetivo de todo desarrollador es obtener un producto que funcione y cumpla las necesidades del cliente y la documentación es un artefacto que utiliza para cumplir su objetivo. Por tanto, no se trata de no documentar sino de documentar aquello que sea necesario para tomar de forma inmediata una decisión importante. Los documentos deben ser cortos y centrarse en lo fundamental. Dado que el código es el valor principal que se obtiene del desarrollo, se trabaja en seguir ciertos estándares de programación para mantener el código legible y documentado.

La colaboración con el cliente por encima de la negociación contractual. Se plantea una interacción continua entre el cliente y el equipo de desarrollo de forma tal

que el cliente forme un tándem con el equipo de desarrollo. Se pretende no diferenciar entre las figuras cliente y equipo de desarrollo, sino que se apuesta por un solo equipo persiguiendo un objetivo común.

Responder a los cambios más que seguir estrictamente un plan. Planificar el trabajo a realizar es muy útil y las metodologías ágiles consideran actividades específicas de planificación a corto plazo. Sin embargo, adaptarse a los cambios es vital en la industria software actual y, por lo tanto, también consideran mecanismos para tratar los cambios de prioridades. La regla es “planificar es útil, seguir un plan es útil hasta que el plan se distancia de la real situación actual.

Para cumplir estos valores se siguen doce principios que establecen algunas diferencias entre un desarrollo ágil y uno tradicional:

1. La 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 que funciona 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 autoorganizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para, a continuación, ajustar y perfeccionar su comportamiento en consecuencia.

Estos principios marcan el ciclo de vida de un desarrollo ágil, así como las bases para las prácticas y procesos a utilizar.

2.3 SCRUM - Fundamentos

Aunque el manifiesto ágil es la piedra angular sobre la que cimientan todas las metodologías ágiles (Extreme Programming, Crystal Methodologies, Dynamic Software Development Method, Feature-Driven Development, Lean Software Development), en el presente trabajo se explicará la metodología *SCRUM* por ser aquella con mayor difusión en la actualidad.

SCRUM es un nombre que describe una forma de desarrollar productos de software iniciada en Japón. No es un concepto nuevo, ya en 1987 Nonaka y Takeuchi fijaron este término, una estrategia usada en rugby en la que todos los integrantes del equipo actúan en conjunto para avanzar la pelota y ganar el partido, para denominar un nuevo tipo de proceso de desarrollo de productos de software. El nombre fue escogido por las similitudes que consideraban que existía entre el juego de rugby y el tipo de proceso que

estaban proponiendo con características adaptables, rápido, auto-organizable y con pocos descansos.

SCRUM es un proceso para la gestión y control de desarrollo que trata de eliminar la complejidad en las áreas responsables del mismo para centrarse en la construcción de software que satisfaga las necesidades del negocio. Es simple y escalable, ya que no establece prácticas de ingeniería del software, sino que se aplica o combina, fácilmente, con otras prácticas ingenieriles, metodologías de desarrollo o estándares ya existentes en la organización.

SCRUM se concentra, principalmente, al nivel de las personas y equipo de desarrollo de software que construye el producto. Su objetivo es que los miembros del equipo trabajen juntos y de forma eficiente obteniendo productos complejos y sofisticados.

También se puede explicar *SCRUM* como un tipo de ingeniería social que pretende conseguir la satisfacción de todos los que participan en el desarrollo, fomentando la cooperación a través de la autoorganización. De esta forma se favorece la franqueza entre el equipo y la visibilidad del producto. Pretende que no haya problemas ocultos u obstáculos que puedan poner en peligro el proyecto. Los equipos se guían por su conocimiento y experiencia más que por planes de proyecto formalmente definidos. La planificación detallada se realiza sobre cortos espacios de tiempo lo que permite una constante retroalimentación que proporciona inspecciones simples y un ciclo de vida adaptable. Así, el desarrollo de software de productos se produce de forma incremental y con un control empírico del proceso que permite la mejora continua.

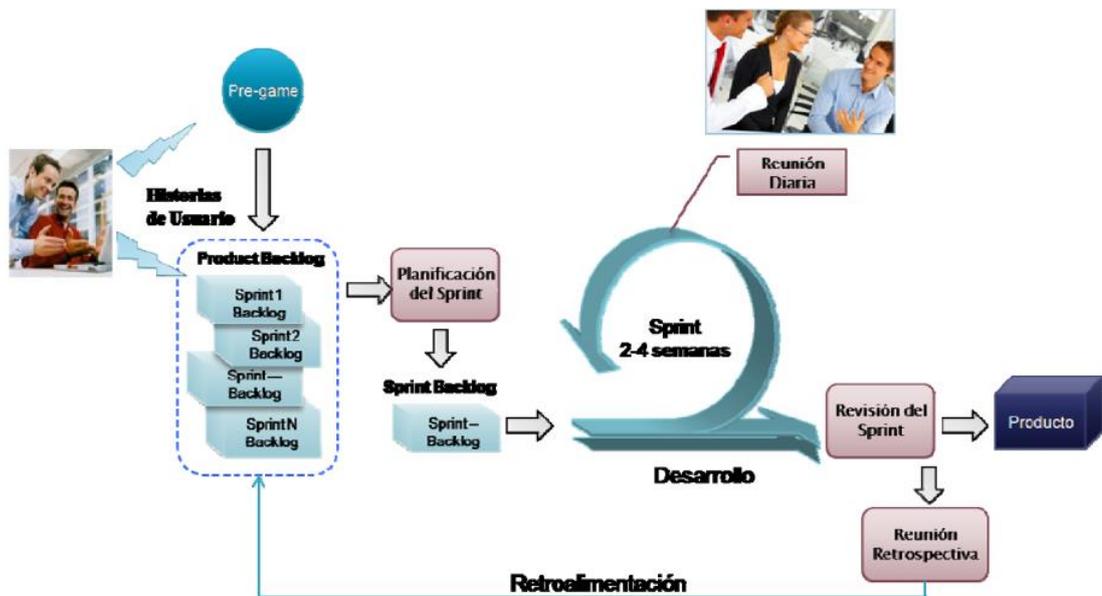


Fig. 1 - Modelo de desarrollo aplicando Scrum-www.agilealliance.com

Independientemente del tipo de metodología que se utilice, cualquier desarrollo software parte siempre de un mismo problema: conocer las necesidades de los clientes.

SCRUM, al igual que el resto de las metodologías ágiles, pretende no centrar las tareas de desarrollo en un conjunto de requisitos formalmente definidos, sino que aboga por la incorporación del cliente como un miembro más del equipo de desarrollo. De este modo, no se considera el proceso de definición de requisitos como un fin dentro del desarrollo del proyecto, sino que los requisitos aparecen implícitamente dentro del contenido de las denominadas historias de usuario.

Las *historias de usuario* son el elemento base que utiliza *SCRUM* para representar las características que el usuario espera que tenga el software que se va a desarrollar. Por lo tanto, pueden incorporar tanto a cuestiones relacionadas con las funciones del sistema como con cualquier otro aspecto de este (restricciones, rendimiento, etc.).

Las *historias de usuario* se presentan desde la perspectiva del usuario. No se describen utilizando una terminología técnica, sino que se escriben utilizando un lenguaje cercano al dominio de la aplicación que se está desarrollando de forma que sea comprensible por los clientes y por los desarrolladores. Las historias de usuario se construyen bajo un mismo esqueleto que centra el foco de las características del producto:

- Primero se determina quién propone la historia de usuario,

- luego se describe la característica que se cubre con la historia de usuario y
- finalmente se especifica la razón por la que dicha característica es necesaria.

El proceso comienza con la fase de Pre-game, en la que se realiza de forma conjunta con el cliente una definición sencilla y clara de las características que debe tener el sistema que vaya a ser desarrollado, definiendo las *historias de usuario* que van a guiar el proceso de desarrollo. Posteriormente, cada *historia de usuario* se refina de manera que se identifican las tareas que son necesarias para llevar a cabo el desarrollo de la historia de usuario. En un principio no es necesario detallar de forma completa todas las *historias de usuario* sino sólo las que tienen un mayor nivel de prioridad por parte del usuario. Esto permite que el proceso de desarrollo pueda adaptarse a posteriores modificaciones de las necesidades de usuario, de forma que el proceso de desarrollo se vuelve más flexible.

El resultado de esta fase de Pre-game es lo que se denomina en *SCRUM* “Product Backlog”, que contiene una lista de todas las *historias de usuario* priorizadas, así como de las tareas que se deben llevar a cabo para la realización del proyecto. Este enfoque basado en quién, qué y por qué facilita la tarea de priorización de las historias de usuario, lo que permite realizar la planificación inicial del proyecto. Como puede deducirse de lo anteriormente expuesto, la obtención del Product Backlog se realiza con la ayuda del cliente/usuario del sistema que se va a desarrollar, por lo que puede considerarse que se realiza en directo y, por lo tanto, las posibles dudas en la definición de las *historias de usuario* puede resolverse en ese mismo instante.

Una vez identificadas y priorizadas las *historias de usuario* del Product Backlog, se realiza la separación de *historias de usuario* en etapas de corta duración (no más de 30 días) denominadas *Sprints*. Para cada *sprint* se realiza una reunión de planificación de lo que se va a llevar a cabo en ese *sprint* y se establece la fecha de finalización de este. El objetivo es mover aquellas historias de usuario con mayor prioridad para el cliente al denominado “Sprint Backlog”, que contiene el conjunto de tareas a desarrollar en ese *sprint*, incluyendo diseño, desarrollo, pruebas, integración, etc. Las historias de usuario se congelan en el Sprint Backlog de forma que durante dicho periodo no puedan producirse cambios sobre los aspectos que se encuentran en fase de desarrollo.

De forma iterativa, todos los días que dure el *sprint*, se realiza una reunión operativa, informal y ágil con el equipo de desarrollo, de un máximo de quince minutos y recomendablemente parados, en donde a cada integrante del equipo se le hacen tres preguntas:

- ¿Qué tareas hiciste desde la última reunión? Es decir, tareas realizadas en un día.
- ¿Qué tareas vas a hacer hoy?
- ¿Qué ayuda necesitas para poder realizar este trabajo?

Es decir, identificación de obstáculos o riesgos que impiden o pueden impedir el normal avance.

Una vez finalizado el *sprint* se debería obtener parte del producto, un entregable o algo que se pueda mostrar y que enseñe los avances acometidos en el *sprint*. En este punto se procede a una fase de revisión del proyecto para mostrar dichos avances e incorporar, si fuera necesario, nuevas historias de usuario al Product Backlog. Para mantener la calidad del producto se establece que una historia de usuario está 100% completa si supera las pruebas unitarias, pasa las pruebas de aceptación, supera las pruebas adicionales para mantener la calidad del producto, el código está construido e integrado satisfactoriamente, está basado en un diseño factorizado sin duplicaciones, el código es claro, estructurado y autodocumentado y, por supuesto, es aceptado por el cliente.

Además, tras la realización de cada *sprint* se lleva a cabo una reunión retrospectiva que permite aprender de los conocimientos y experiencias adquiridas hasta el momento y mejorar de forma continua el proceso de desarrollo. Se revisará con el equipo los objetivos marcados inicialmente en el Sprint Backlog concluido, se aplicarán los cambios y ajustes si son necesarios, y se marcarán los aspectos positivos (para repetirlos) y los aspectos negativos (para evitar que se repitan) del *sprint*, que servirán de retroalimentación para el siguiente *sprint*.

Los elementos que servirán de entrada y guiarán la planificación de un nuevo *sprint* serán el Product Backlog, las capacidades o habilidades del equipo, las condiciones que en ese momento se den del negocio, el avance tecnológico que se haya podido producir y el incremento que se pretenda hacer al producto que se está desarrollando. De esta

forma, en contraposición con las metodologías convencionales, el proceso de desarrollo adquiere flexibilidad, agilidad y capacidad para adaptarse a su entorno.

Para orquestar este proceso *SCRUM* distingue distintos actores con diferentes papeles dentro del proceso. De forma general, podemos distinguir al propietario del producto o Product Owner, maestro de *SCRUM* o *SCRUM* Máster, equipo de desarrollo o Scrum Team y cliente o usuario.

El *Product Owner* es la única persona encargada de la dirección y control del Product Backlog, es decir, de las *historias de usuario* que debe cumplir el sistema. Se trata de una persona física (solamente una persona para eliminar las posibles confusiones o interferencias), no una organización o comité. Puede ser el propio cliente in situ en el lugar de desarrollo u otra persona que tenga el conocimiento suficiente sobre el producto o pueda estar en continuo contacto con el cliente para marcar las prioridades del proyecto. Es la persona oficialmente responsable del proyecto que, de forma visible, vocal y objetiva debe tomar todas las decisiones de negocio para el producto.

Para que el Product Owner tenga éxito, el resto del equipo de la organización tiene que respetar sus decisiones. En cuanto a su implicación debe estar en constante interacción con el equipo de desarrollo. Debe asistir, al menos, a las reuniones de planificación y de revisión de cada sprint y estar en continuo contacto con el equipo para proporcionar detalles sobre las historias de usuario y constante retroalimentación que rija el desarrollo del sprint.

El *SCRUM Máster* es la persona responsable del éxito al aplicar la metodología *SCRUM* en el desarrollo del proyecto o producto, asegurando que los valores, prácticas y reglas son seguidos por el resto del equipo. Es la persona que asegura el seguimiento de la metodología guiando las reuniones, ayudando al equipo ante cualquier problema que pueda aparecer y controlando que el trabajo siga el ritmo adecuado. Debe tomar decisiones inmediatas y eliminar los impedimentos que vayan surgiendo en el momento, aunque en ocasiones no cuente con toda la información necesaria. Su responsabilidad es entre otras, la de hacer de paraguas ante las presiones externas y motivar al resto del equipo. Por tanto, la labor de *SCRUM Máster* requiere una fuerte personalidad ya que debe facilitar el trabajo del equipo sin imponer autoridad.

El *SCRUM Team* lo conforman las personas responsables de implementar la funcionalidad o funcionalidades elegidas por el Product Owner. Es un conjunto de personas motivadas con habilidades y capacidades complementarias que estén comprometidos por un propósito común, cumplir el objetivo del sprint. El equipo tiene plena autoridad para tomar todas las decisiones que consideren adecuadas en el desarrollo del proyecto, autoorganizándose y auto disciplinándose. Así, por ejemplo, en las reuniones de planificación el *Product Owner* y el *SCRUM Team* deben llegar a un acuerdo realista sobre las historias de usuario que se van a completar en el siguiente sprint y, si en algún momento el *SCRUM Team* considera que algunas de las prioridades del *Product Owner* no es razonable, dispone de libertad absoluta para remarcar este caso y obligar al *Product Owner* a variar sus prioridades.

Teóricamente, se estima que debería estar formado por un número de entre 7 u 8 miembros como máximo y 2 como mínimo.

Finalmente, el *cliente* son los beneficiarios finales del producto quienes, viendo los progresos, pueden aportar ideas, sugerencias o necesidades. Su participación es importantísima e imprescindible en esta metodología.

La efectividad de la metodología para la gestión de proyectos se basa en un conjunto de valores fundamentales que deben seguir todos los integrantes del equipo, principios sobre los que reposan el resto de las prácticas:

- Los miembros del equipo deben estar dispuestos a comprometerse con el objetivo de cada sprint y del proyecto en general. *SCRUM* proporciona al equipo toda la autoridad que necesiten para obtener a cambio su compromiso.
- El equipo se tiene que comprometer a hacer su trabajo. Cada miembro debe concentrar todos sus esfuerzos y habilidades en cumplir con el trabajo que se han comprometido a realizar sin desviarse en otros aspectos, realizando todas sus labores con esmero.
- Todos los aspectos del proyecto son visibles para todo el equipo por lo que un valor fundamental es también la franqueza. Además, los miembros del equipo están avalados por sus conocimientos y experiencias, el respeto es un valor fundamental con cada uno de ellos.

- Finalmente, cada miembro del equipo debe tener el coraje suficiente para comprometerse, actuar, ser transparente en el desarrollo y respetar y hacer que lo respeten.

2.4 Metodologías tradicionales y ágiles

En toda organización madura, debe requerirse seguir una metodología de software bien definida para el correcto desarrollo de su producto. Los modelos de ciclo de vida de desarrollo de ayudan a las empresas a producir software de buena calidad dentro de los límites permitidos de tiempo, alcance y de costos.

Desarrollar un buen software depende de un gran número de actividades y etapas, donde el impacto de elegir la metodología para un equipo en un determinado proyecto es trascendental para el éxito del producto.

Las metodologías no ágiles son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo; llamadas también metodologías tradicionales, donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema.

Las metodologías tradicionales de desarrollo resultan de las bases del pensamiento tayloriano que legitima el management y la división de distintas tareas, la especialización de los trabajadores y el control del tiempo necesario para realizar las tareas en un proceso de producción industrial con el objetivo de maximizar la productividad. Además, adoptan el aporte de unos de los discípulos de Taylor, H. Gantt quien, en 1915, publicó para Occidente el método de gráfico de Gantt para obtener, mediante técnicas de programación como PERT y ruta de camino crítico, una representación visual de la planificación y programación de tareas secuenciales.

Las metodologías tradicionales son denominadas, a veces, de forma despectiva, como metodologías pesadas. Centran su atención en llevar una documentación exhaustiva de todo el proyecto, la planificación y control de este, en especificaciones precisas de requisitos y modelado y en cumplir con un plan de trabajo, definido todo esto, en la fase inicial del desarrollo del proyecto.

Estas metodologías tradicionales imponen una disciplina rigurosa de trabajo sobre el proceso de desarrollo del software mediante un modelo predictivo, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y, una vez que esté todo detallado, comienza el ciclo de desarrollo del producto de software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. A continuación, se presenta una figura del proceso de desarrollo de software mediante metodologías tradicionales:

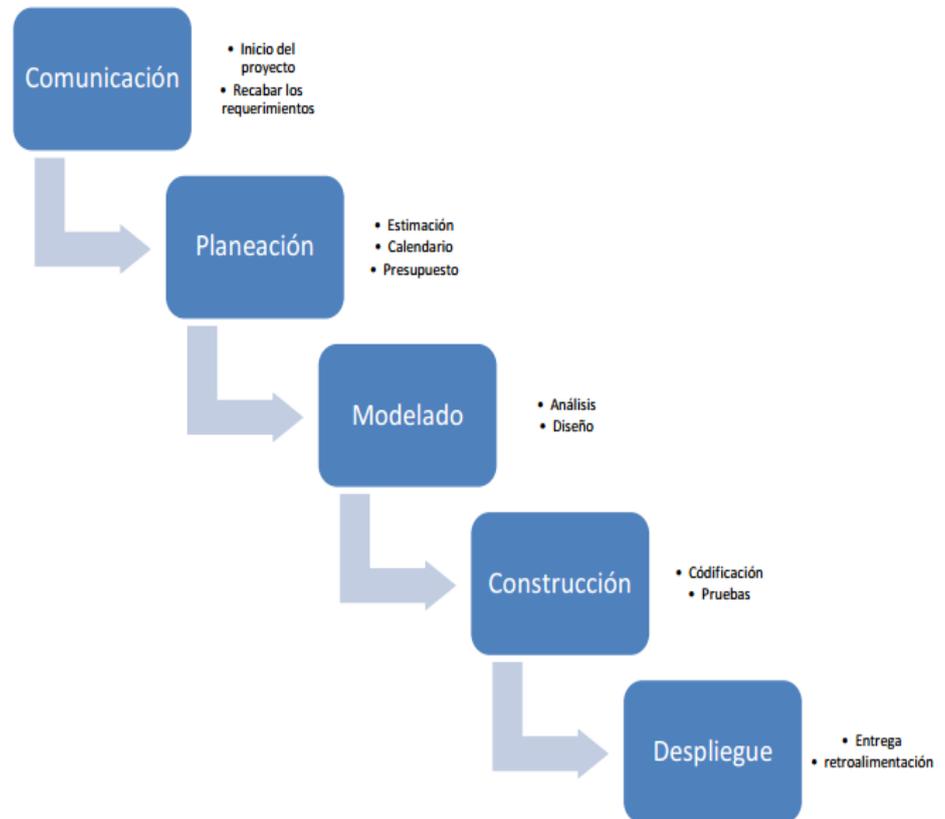


Figura 1. Modelo en cascada - Pressman

Como se puede ver en la figura, las tareas del proceso se realizan de forma de cascada, secuencial y sistemática ya que el inicio de una depende de la finalización de la anterior. Cada tarea es realizada por áreas con personal especializado en las actividades que la integran y no se formalizan actividades de colaboración entre ellas.

2.4.1 Ventajas de las metodologías tradicionales

- La principal ventaja es que proporciona una estructura para organizar y controlar un proyecto de desarrollo de software.
- Los detalles y errores de diseño son detectados en fases anteriores a la programación.
- La elaboración de documentos ayuda al proceso de mantenimiento del desarrollo.
- Si el procedimiento se sigue correctamente, se puede obtener una estimación de costos y tiempos de precisión.
- Hace énfasis en la documentación, lo cual genera que cualquier nuevo trabajador que se incorpore al equipo de proyecto pueda ponerse al día rápidamente.
- Un plan de proyecto puede ser utilizado para proyectos similares a futuro, es plan queda como un activo de la empresa.

2.4.2 Desventajas de las metodologías tradicionales:

- Se requiere que todos los requisitos se especifiquen detalladamente en el comienzo del proyecto.
- Al ser un modelo lineal, no permite cambios para ser aplicados.
- No tiene formalmente una retroalimentación de actividades, no hay una manera de corregir errores iniciales en actividades posteriores.
- Se pierde tiempo en la finalización de las actividades predecesoras.
- El cliente no tiene un producto de software funcionando hasta el final del proyecto.

Se puede interpretar que las metodologías tradicionales no se adaptan naturalmente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno donde los requisitos no puedan predecirse o bien puedan variar, ya generarían altos costos de implementaciones de cambios.

Las metodologías ágiles aparecen como una opción atractiva en contraposición con las metodologías tradicionales basadas en principios de estabilidad y control del contexto. Las metodologías ágiles no se centran en habilidades de predicción, ni pretenden tener un sistema perfectamente definido como paso previo a su construcción, sino que perciben cada respuesta al cambio como una oportunidad para mejorar el sistema e incrementar la satisfacción del cliente, considerando la gestión de cambios como un aspecto inherente al propio proceso de desarrollo software y, permitiendo de este modo, una mejor adaptación en entornos turbulentos.

Por otro lado, un porcentaje muy alto del desarrollo software actual se centra en la evolución de productos ya construidos, con el objetivo de maximizar la inversión y reducir costos, ya sea para variar parte de su funcionalidad, añadir otra nueva o, por ejemplo, adaptar el sistema a un nuevo dominio de aplicación.

2.4.3 Ventajas de las metodologías ágiles

- Toma en consideración los requisitos y la naturaleza de los mismos es cambiante por lo que es escalable para proyectos de alta incertidumbre.
- Promueve el trabajo en equipo, diferentes áreas de la gerencia de desarrollo y negocio trabajan en forma simultánea.
- El cliente tiene un producto de software funcionando en las primeras fases de desarrollo.
- No se debe documentar todo el trabajo, solo las mínimas normativas.
- Es un proceso de planificación muy flexible.
- Promueve la comunicación ya que las áreas de desarrollo y el cliente trabajan en conjunto.

2.4.4 Desventajas de las metodologías ágiles

- Difícil seguimiento de dependencias complejas entre áreas.
- Difícil estimación de tiempo de entrega. Se inicia con alta incertidumbre.
- Se requiere de trabajadores con experiencia para tomar decisiones.
- Necesidad de la implicación del cliente en el día a día de desarrollo.

- No hay documentación para la rápida integración de nuevos trabajadores.
- Los contratos comerciales deben ser flexibles respecto a las variables tradicionales.

2.4.5 Elección de las metodologías a utilizar

Jeff Sutherland, co-creador de la metodología *SCRUM* observa que la decisión de implementar la metodología es tomada por las gerencias de TI, quienes deberán buscar implementar un proceso de desarrollo que garantice, en la mejor velocidad posible, la disponibilidad de productos de software y que a su vez sea compatible con las demás normas, procedimientos y mejores prácticas actuales de la empresa.

Las organizaciones tradicionales en donde se implementen metodologías ágiles deberán compartir en mayoría el concepto de ser una empresa que aprende y debe permitirse equivocarse y, a partir de allí, los altos cargos de las áreas de TI deberán detectar y controlar los drivers que puedan hacer un mejor proceso de gestión híbrido entre ambos estilos de trabajo.

No existe verdad absoluta en este campo, las metodologías ágiles tienen una filosofía y origen diferente a las metodologías tradicionales, aunque pueden coexistir. La elección de la metodología de desarrollo de software a utilizarse debe ser una decisión táctica de TI, para lo cual deberá evaluar la situación del entorno partiendo de las siguientes variables internas y externas resultantes de la investigación:

- Solidez de los requerimientos y especificaciones.
- Tiempo limitado de entrega del producto de software.
- Necesidad de prototipado funcional.
- Necesidad de documentación.
- Posibilidad de participación colectiva cliente/desarrollo.
- Necesidad de software seguro.
- Seniority del personal disponible.

Capítulo 3: Integración de las metodologías ágiles en las organizaciones

3.1 De la gestión empresarial tradicional hacia la ágil

Con el fin de lograr continuamente buenos resultados y ser exitosas, las empresas deben adaptarse hábilmente a un entorno competitivo, buscar zonas atractivas para competir, y construir una ventaja competitiva. Esta debe ser la intención de una estrategia de negocios.

Esta adaptabilidad requiere agilidad para adaptarse rápidamente al medio ambiente, para detectar oportunidades en el entorno o crear nuevas oportunidades por sí mismas, permitiendo controlar su curso estratégico de manera inteligente

Para que una empresa megra pueda implementar metodologías ágiles de desarrollo de SW, debe tener una cultura colaborativa orientada al cliente y una organización estructural adaptable a los nuevos tipos de interacciones laborales internas compatibles con los valores pilares del pensamiento ágil.

Las empresas tradicionales vigentes desarrollaron su organización estructural en lo que Henry Mintzberg (1994) describe como una burocracia mecánica, una gestión empresarial mediante modelos basados por las teorías weberianas de orden por jerarquía según el paradigma clásico y en la especialización de las tareas prescriptas entre varios conceptos resultantes de los años de la era industrial de mediados del siglo XX.

El paradigma clásico tuvo la virtud explicar, sistematizar y perfeccionar a las primeras prácticas gerenciales, sacándolas de su concepción meramente intuitiva, con la pretensión (aunque no lograda) de erigir al management en una disciplina de carácter científico (Zalazar,2012).

Sus aportes se resumen en un conjunto de herramientas destinadas a hacer más eficiente la acción administrativa, a saber:

- Herramientas vinculadas a la organización básica de tareas: en esta categoría incluimos a aquellos principios de orden práctico como la división del trabajo - especialización, los esquemas de métodos y tiempos, los dispositivos de supervisión funcional.

- Herramientas orientadas a la eficacia en la gestión administrativa: relacionadas con los principios clásicos y neoclásicos de administración delimitadores de las funciones y deberes de un administrador.
- En el ámbito del taller, surge como nueva temática la “administración de producción”, herramienta imprescindible en el aprovechamiento de las modernas tecnologías de producción.
- La concepción de organización como instrumento formal, sistema ordenador de acciones y recursos.

Si bien se han ido actualizando y mejorando las prácticas administrativas, los gerentes actuales de muchas empresas han hecho sus estudios y su trayectoria laboral dentro de los modelos clásicos de administración. Sus empresas pueden ser exitosas en términos de ROI o de optimización de recursos, pero raramente son pioneras en la creatividad y la innovación, lo que resulta en tener gran posibilidad de pagar un alto costo ante los cambios.

Los avances tecnológicos de la actualidad y la democratización de la información generaron que muchas de las empresas de esta época decidan rever su visión y misión, transformar su un sistema de valores a un modo más abierto y tener una burocracia óptima orientada al servicio interno y externo, desarrollando la habilidad de ir aprendiendo del entorno para poder anticiparse a los cambios y, mediante la creatividad, llegar a innovar en el mercado creando y tomando valor.

Las empresas tradicionales que cuentan con un área de TI seguramente hayan evaluado en esta época la factibilidad de implementar metodologías de desarrollo ágil. Para impulsarlo es necesario tener una visión de futuro desde una perspectiva sistémica de la empresa, una estrategia sólida y recursos precisos para poder llevar a cabo la tarea transformacional.

3.2 Nuevos roles en las organizaciones

Para la implementación de *SCRUM*, algunas áreas de la empresa necesitarán incorporar inicialmente a su estructura dos nuevos roles y nuevas habilidades en la empresa.

Para conformar un equipo de desarrollo ágil, se debe contar con un grupo de personas que, en su conjunto y por sus niveles de conocimiento, puedan desarrollar valor hacia el cliente a través de TI. El *SCRUM Team* debe tener claro el por qué y para qué están desarrollando la aplicación.

El equipo estará conformado por personas especialistas en diseño, programación, calidad, UX y un *Product Owner* quien deberá tener conocimiento suficiente del producto que desarrollarán. Es común que los trabajadores asignados en ese rol vengan desde áreas de negocio y deberán estar familiarizados y capacitados en metodologías ágiles. El *Product Owner* se puede explicar básicamente como el cliente que forma parte del proceso de desarrollo.

El segundo rol es el de *SCRUM Master*, el cual será una especie entrenador de la metodología *SCRUM* y quien fomentará y motivará la correcta dinámica ágil bajo el modelo. Naturalmente un *SCRUM Master* tiene un alcance multiproyecto ya que su implicación no altera la capacidad de un equipo, sino que busca la colaboración entre toda la comunidad ágil.

Si el crecimiento de equipos ágiles de desarrollo en la empresa va en aumento, se debe considerar agrupar a los mismos según atributos comunes conformando una comunidad y crear el rol de *responsable de comunidad* quien colaborará en la resolución de problemas comunes entre los equipos de desarrollo.

Las organizaciones que estratégicamente decidan comenzar a innovar también pueden incorporar trabajadores con el rol de *Intrapreneur* para fomentar la creatividad y buscar grandes ideas.

3.3 Integración por indicadores de gestión

El éxito o fracaso de la integración de metodologías ágiles de desarrollo al modelo de gestión de una empresa tradicional estará íntimamente relacionado con el valor que perciba la empresa que el producto de software a desarrollar le brinda al cliente final.

Tomando los conceptos de Porter (1986) y la cadena de valor, y de la investigación realizada para este trabajo, puedo observar que actualmente las áreas de desarrollo de

muchas empresas dejaron de ser parte de las áreas de tipo de soporte para ser actividades primarias.

Por este motivo, como resultado del relevamiento de investigaciones relacionadas al tema, es un hecho positivo para la implementación de la metodología que la gerencia de desarrollo incorpore como indicadores clave a los siguientes factores:

- Satisfacción del cliente: Para integrarse motivando la colaboración y la transparencia. Se puede utilizar para relacionarse con áreas de marketing, comerciales, de investigación de mercado y producto. Se lo puede obtener a través de métodos de ponderación a los requerimientos priorizados por el Product Owner y a la percepción promedio del usuario final.
- Eficiencia Operativa: Para integrarse motivando el cuestionamiento del status quo. Se puede utilizar para obtener un contacto con las áreas de operación, de control de gestión, de riesgos y de recursos humanos entre otras.
- Ventaja del mercado: Para integrarse motivando la creatividad e innovación. Se relaciona integralmente con las áreas de marketing, de finanzas y con áreas del negocio principal de la empresa.

Razonablemente, los factores detallados arriba están relacionados a los valores y principios de las metodologías ágiles. Se sugiere fomentar la filosofía ágil a través de métodos cuantificables dentro de la empresa para lograr su posicionamiento en todas las áreas funcionales posible de la empresa. La gerencia de desarrollo en este caso cumplirá un rol de aceleradora de la adopción de los nuevos métodos.

El análisis y publicación de indicadores de percepción del cliente final de los productos o servicios de una empresa ayuda a homologar la metodología ágil y los motivos y propósitos de su implementación. Con la publicación y comunicación de indicadores se colabora con la idea de tener una visión compartida de empresa en a través de toda la estructura de la organización.

3.4 Implementación de metodologías ágiles en una organización tradicional

Para buscar implementar metodologías ágiles de desarrollo exitosamente en una empresa tradicional con una madurez organizacional media-alta, es recomendable tener en cuenta las siguientes tácticas:

- **Obtener el compromiso de la gerencia:** La alta y media gerencia deben capacitarse y recibir información sobre los beneficios de las prácticas ágiles. Para una organización, un cambio abrupto hacia prácticas ágiles podría ser difícil, por lo que es considerable proponer a los gerentes altos la aplicación de metodologías ágiles como tácticas de recuperación para proyectos con resultados actualmente fallidos. La aceptación para tales proyectos debería ser fácil ya que el deseo de rescatar el proyecto y probar una nueva metodología es naturalmente alto.
- **Relevar y analizar cuanti y cualitativamente el historial de indicadores de plazos de entrega de proyectos:** Si una organización tiene un historial negativo, es probable que la organización apoye el cambio de metodología sin problemas, en cambio, si el historial es positivo, la aceptación al cambio se deberá buscar desde la mejora en los tiempos de entrega final de desarrollo y su reducción de consumo de recursos.
- **Articular capacitación y soporte:** Las gerencias de la empresa y principalmente la gerencia de desarrollo y sus integrantes, deben estar familiarizados con los principios, las prácticas y las metodologías ágiles antes de implementarlas. Se debe proporcionar capacitación y soporte para eliminar cualquier tipo de concepto erróneo y mitigar la resistencia al cambio. Tiene que estar claro el concepto que el desarrollo ágil es un método que implica disciplina para utilizarla, que una implementación parcial de la metodología no producirá los resultados esperados, que el software se entrega funcionando y sin errores, que la planificación depende de los desarrolladores y que los gerentes no deben quejarse si algunas tareas llevan más tiempo que el previsto.

- Integrarse con procesos externos: Es importante que las metodologías ágiles se adapten para integrarse con las prácticas no ágiles existentes en la empresa. Esto significa estar listo para producir entregables de menor valor y crear planes o informes de estado en formatos tradicionales para obtener la aprobación necesaria. Una vez que los beneficios de los procesos ágiles son claramente visibles, se pueden iniciar negociaciones sobre el equilibrio de enfoques ágiles y no ágiles. Al introducir un proceso ágil, la gerencia debe comprender el impacto en grupos fuera del grupo de desarrollo. El rendimiento del equipo de desarrollo podría verse afectado negativamente por otros grupos que utilizan procesos no ágiles. Los gerentes deben encontrar maneras de resolver las diferencias de opinión entre estos equipos. La gerencia de recursos humanos debe estar informada sobre cualquier equipo de desarrollo que se transforme a las metodologías ágiles y podría compartir inquietudes sobre los resultados imprecisos de la adopción metodológica.
- Usar un proyecto piloto: Las empresas deben usar proyectos piloto para demostrarse que las metodologías ágiles les funcionan, usar proyectos de prueba para determinar cómo integrar procesos ágiles en procesos existentes y luego pasar a proyectos de producción. Se deben usar los filtros de idoneidad del proyecto para determinar el proyecto apropiado para el piloto. El proyecto piloto debe contar con sponsors empresariales que demuestren que el proyecto tuvo un buen patrocinio y que ofrece un alto valor comercial y para demostrar que las prácticas ágiles pueden producir software de alta calidad en poco tiempo y en una corta duración para obtener resultados.
- Inspeccionar y Adaptar: El éxito del proyecto piloto podría mostrarse para atraer más sponsors en la adopción de prácticas ágiles. La gerencia de desarrollo debe crear conciencia y permanecer abierta a la retroalimentación. Deben utilizar las experiencias del equipo involucrado en el proyecto piloto para planificar futuros proyectos. El equipo del primer proyecto piloto debería comenzar a actuar como mentores para los equipos que trabajarán en proyectos ágiles posteriormente.

- Mantener la agilidad: Luego de haber adoptado los procesos ágiles, es necesario mantener el ambiente de trabajo ágil. Se deben proporcionar estímulos constantes para evitar que las personas vuelvan a los enfoques antiguos. Se debe promover un ambiente de discusión a través de grupos de interés especial, conferencias, talleres ágiles, etc. Además, las gerencias deben asumir un rol enfocado más en gestionar y facilitar en lugar de controlar y reclamar. Tomando los aportes de Bass B (1998), las características del estilo de liderazgo necesario se pueden definir como *transformacional*. Las gerencias deberán influir con la idealización, con motivación inspiradora, con la estimulación intelectual y tener una consideración especial con sus subordinados. Si los equipos de desarrollo no ven a los gerentes trabajando mediante un modelo de conducta coherente con la agilidad, no creerán en el valor real de lo que están haciendo.

Conclusiones

Tras la realización de este trabajo de investigación acerca de la gestión empresarial y su integración con las metodologías ágiles de desarrollo, se pudo observar que las empresas tradicionales que adopten estas metodologías de desarrollo también estarán adoptando transformaciones que van desde la forma de medir los resultados hasta la forma de pensar la empresa.

Los conceptos que trae el agilismo resumidos en: trabajo colaborativo mediante equipos autoorganizados, transparencia, continuo aprendizaje y construcción veloz de valor llevan a los equipos ágiles a dudar de las estructuras empresariales tradicionales y su manera de llevar el mando y control, de la forma que se toman las decisiones y a preguntarse para qué hacen lo que hacen. Estará en la habilidad del responsable de las áreas de TI la manera de lograr que ambas formas de trabajar sean concluyentes en una mayor entrega de valor a través de los drivers que desarrolle para tal fin.

Se observa también que el agilismo, si bien tiene desarrollado métodos y técnicas para implementarse, será difícil de hacerlo en aquellas empresas donde no se homologue el concepto de agilidad dentro de la cultura empresarial, para lograr esto es fundamental la deconstrucción de los patrones burocráticos rígidos junto con una comunicación interna eficiente para posicionar la metodología por parte de los sponsors y los integrantes de los equipos de desarrollo ágiles.

En los tiempos actuales, las áreas de TI han evolucionado de papel dentro de las empresas pasando de ser un área de soporte a la producción a ser parte estratégica de la organización. La implementación de metodologías ágiles genera que las áreas técnicas tengan claro el objetivo de su trabajo y mediante una visión sistémica, lograr mayor empatía con el usuario final o cliente.

Del resultado de la investigación se concluye que los gerentes y mandos medios tradicionales que interactúen o tengan a su cargo grupos de trabajo que apliquen metodologías ágiles, si bien pueden adaptarse a los cambios, tendrán que desarrollar en mayor nivel sus competencias interpersonales e intrapersonales necesarias para coexistir en la coyuntura actual contribuyendo al cambio de filosofía a través de sus habilidades.

La pregunta introductoria de este trabajo, acerca de si se adaptan bien los métodos de gestión empresarial a las actuales características de dinamismo y variabilidad del contexto digital, tras haber realizado la investigación puede responderse afirmativamente sí y solo si los valores y principios de la filosofía que promueve la agilidad ya hayan traspasado las puertas de las gerencias técnicas, sea evidenciado en el diálogo cotidiano entre los trabajadores y hayan sido incorporados a la estrategia, táctica y operación de la organización.

Bibliografía:

- Fredrickson, J. W. (1986). *The strategic decision process and organizational structure. Academy of Management Journal*, 280-297.
- Grosso, F. (2016). ¿Se están diseñando las organizaciones correctas? *Revista Argentina de Investigación en Negocios*, 2(1-2).
- Marjański, A., (2004). *Inteligencia strategiczna-Strategic and Strategy*, Varsovia, Polonia, Rebis.
- Mintzberg, H. (1994). *Diseño de Organizaciones Eficientes*. Buenos Aires, Argentina, El Ateneo.
- Palacio, J. (2015). *Gestión de proyectos Scrum Manager*. Buenos Aires, Argentina, Scrum Manager Ed.
- PMI. (2017). *PMBOK. (6ta. Ed)*. Pensilvania.
- Pressman, R. (2010). *Ingeniería del software: un enfoque práctico. (7ma. Ed)*. México DF, México. McGraw-Hill.
- Qumer, A., Henderson-Sellers, B. (2008). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and Software Technology*, 50(4), 280-295.
- Sajdak, M. (2015). The Concept of an Agile Enterprise Research Model. *International Journal of Social, Behavioral, Educational, Economic and Management Engineering*, 6, 1764-1769.
- Senge, Peter. (2004). *La quinta disciplina: Como impulsar el aprendizaje en la organización inteligente*. Buenos Aires, Argentina, Granica.
- Sutherland, J. (2014). *Scrum: the art of doing twice the work in half the time*. New York, U.S.A. E-book.
- Takeuchi, H., Ikujiro, N. (1986). The New New Product Development Game. *Harvard Business Review*, 64(1).
- Verma, S., Kannan, V., y Jhajharia, S. (2014). Agile vs waterfall: A Comparative Analysis. *International Journal of Science, Engineering and Technology Research (IJSETR)*, 3.

www.agilealliance.org

www.agilemanifesto.org

Zalazar, J. (2012). *Introducción a la administración: Paradigmas en las organizaciones*. Málaga, España. Eumed.net.