Comparación de las metodologías cascada y ágil para el aumento de la productividad en el desarrollo de software

Comparison of waterfall and agile methodologies for increasing productivity in software development

Fabián González González[1]
fabian.gonzalez01@usc.edu.co

Sandra Liliana Calero Castañeda[1]
sandra.calero01@usc.edu.co

Diego Fernando Loaiza Buitrago, M.Sc[2]
diego.loaiza02@usc.edu.co

Universidad Santiago de Cali, Facultad de Ingeniería, Programa de Ingeniería Industrial (1)
Universidad Santiago de Cali, Facultad de Ingeniería, Programa de Ingeniería de Sistemas (2)

Resumen
El presente artículo expone una revisión teórica, sustentada en un enfoque descriptivo, de dos modelos empleados en la ingeniería de software, los cuales sirven como herramienta para mejorar los procesos organizacionales. Dichos modelos son el modelo de cascada o denominado tradicional y el modelo ágil. Metodológicamente, se realizó una revisión de artículos académicos junto a algunos libros y publicaciones, tanto en inglés como en español, que abordan temáticas relacionadas con el manejo de dichos modelos para el desarrollo de software. Los resultados obtenidos se presentan en cinco capítulos, los cuales abordan tres objetivos específicos: conocer la historia de los modelos y sus principales características; conocer las técnicas que se utilizan en cada uno de los modelos mencionados; identificar el éxito que ha tenido al momento de implementar software en cada uno de los modelos; igualmente se hace una comparación con el fin de determinar la eficiencia de los modelos, sus límites y ventajas. Los resultados demuestran que la forma de evaluar determina la calidad del software.
Palabras Clave: software, ingeniería de software, modelo cascada, modelo ágil, diseño, productividad.

Abstract

This article presents a theoretical review, based on a descriptive approach, on two models used in software engineering, which serve as tools to improve organizational processes. Such models are: cascade or traditional and the agile model. Methodologically, a review of academic articles was carried out along with some books and publications, both in English and in Spanish, which address issues related to the management of such models for software development. The results obtained are presented in five chapters, which address three specific objectives: to know the history of the models and their main characteristics; know the techniques used in each of the mentioned models; identify the success you have had when implementing software in each of the models; A comparison is also made in order to determine the efficiency of the models, their limits and advantages. The results show that the way to evaluate determines the quality of the software.
Keywords: software, software engineering, waterfall model, agile model, design, productivity.

1. INTRODUCCIÓN

Actualmente, las empresas demandan sistemas de información lo suficientemente robustos que soporten los procesos misionales de las mismas, permitiéndole satisfacer las necesidades de sus clientes. Para ello es necesario que los proveedores de dicha solución tecnológica logren canalizar todos sus requerimientos y desarrollen un software ajustado a sus necesidades, cumpliendo oportunamente los tiempos de entrega del producto. Sin embargo, la productividad en el desarrollo de software puede llegar a ser difícil de evaluar y atacar, ya que el desarrollo de software es una labor creativa que depende en su mayoría de las personas, de sus habilidades, conocimientos y metodologías.

La mayoría de costos de las empresas que se dedican a esta actividad, están en la mano de obra tan específica que requieren, es por esto que la mejora de la productividad de sus ingenieros es de gran valor, pues sus ingresos se estiman en parte por hora/hombre. Por lo anterior resulta importante encontrar una metodología que, sin sacrificar el alcance funcional y tecnológico de los productos de software, su calidad y el compromiso de entrega al cliente, permita optimizar los costos de producción (Pardini, 2018).

Actualmente, existen diferentes metodologías que permiten llevar a cabo esta ardua labor de desarrollo de software.

Las más importantes son: entre las más reconocidas figuran la metodología en cascada y la metodología ágil. La primera sigue un proceso lineal y secuencial, donde una vez se completa una de las etapas, el equipo de desarrollo continúa con la etapa siguiente. El equipo no puede devolverse a una etapa anterior a menos que reinicie el proceso. La metodología ágil, en lugar de segmentar los proyectos por etapas los aborda en su conjunto, logrando manejarlos de manera más fluida, centrándose en la colaboración y funcionalidad cruzada (Randstad, 2017).

Históricamente, el modelo en cascada ha estado asociado al éxito en procesos de desarrollo de sistemas altamente complejos, pues destaca por su utilidad a la hora de establecer compromisos contractuales de tiempo de entrega de un producto de software y el alto grado de control del alcance funcional que tendrá el producto de software construido. Por otra parte, la metodología ágil, propone valorar más al individuo y la forma como interactúa con otros, por encima de los procesos y las herramientas; así mismo, destaca el software en correcto funcionamiento, por encima del uso de documentación excesiva. También considera Beck, et al (2011) que tiene gran relevancia la respuesta al cambio que mantener un plan estrictamente diseñado.

En ese sentido, este artículo contribuye aportando una investigación teórica que permita a las empresas desarrolladoras de software tomar decisiones, informadas y consientes, sobre la metodología más adecuada que se ajuste a sus objetivos particulares. Para ello se realizará una revisión bibliográfica de las metodologías y su comparación, enfocada en la productividad en el desarrollo de software, de manera que se brinde un panorama que permita identificar sus beneficios, implicaciones y demás aspectos importantes, tanto individuales como en conjunto.

Estructuralmente, los capítulos en este artículo se desarrollan de la siguiente manera:

- En el primer capítulo se explica en qué consiste la metodología en cascada, su historia y características.
- En el segundo capítulo se define la metodología ágil, sus orígenes y principales características.
- En el tercer capítulo se dan a conocer las técnicas utilizadas para la administración e implementación, tanto de la metodología en cascada como de la ágil y el impacto en la estructura de la organización, en el recurso humano y en los roles que se desempeñan en el interior de cada una.

En el cuarto capítulo se exponen algunos casos de éxito en organizaciones a nivel mundial en los que se haya aplicado la metodología en cascada y otros en donde se aplicó la metodología ágil. Adicionalmente se abordarán los inconvenientes que se presentaron durante la implementación de cada caso.

El quinto capítulo presenta la comparación entre las dos metodologías, dando a conocer sus ventajas y desventajas desde los diferentes frentes como son: implementación, requisitos de conocimiento teórico/ práctico, administración, desarrollo, impacto en la calidad del producto, fechas de entrega y productividad. Finalmente se analizará cómo sería una posible combinación entre ellas.

Por último, se exponen algunas las conclusiones a las cuales se haya podido llegar, vale la pena recordar que no se pretende acotar toda la información existente y que este trabajo presenta límites metodológicos, como lo es por ejemplo la poca información sobre la evaluación de los casos expuestos.

2. CAPÍTULO 1. METODOLOGÍA EN CASCADA HISTORIA Y CARACTERÍSTICAS

En este apartado, se expone de qué se trata la metodología en cascada para el diseño de Software, cuál ha sido su historia, y sus principales características, esto con el fin de tener un panorama claro sobre la importancia de esta al interior de las organizaciones, así mismo, se identifican algunos resultados que se han obtenido a raíz de la implementación de estos sistemas. Cabe resaltar que dicha herramienta está destinada a solucionar problemáticas relacionadas con el área de planeación y de logística de las empresas.

2.1 Orígenes de la metodología de cascada

La ingeniería de software se centra en aplicar enfoques de manera sistemática para la solución de problemas en distintas áreas, lo que implica un grado de disciplina y de comprobación cuantificable, aplicado al desarrollo de software, a su operación, su mantenimiento y sus mejoras. En relación con el diseño en cascada, se debe decir que sus orígenes se remontan al año de 1970 cuando el ingeniero Winston W. Royce propone una metodología la cual pretendía dar seguimiento al desarrollo de un software, sin embargo, la idea era detectar los posibles errores que se presentaban en el diseño, dicho de una manera más científica es "La disciplina tecnológica y de gestión que concierne a la producción y el mantenimiento sistemático de productos software desarrollados y modificados dentro de unos plazos estipulados y costes estimados" (Fairley citado por Pickin y García, 2015, p. 5)

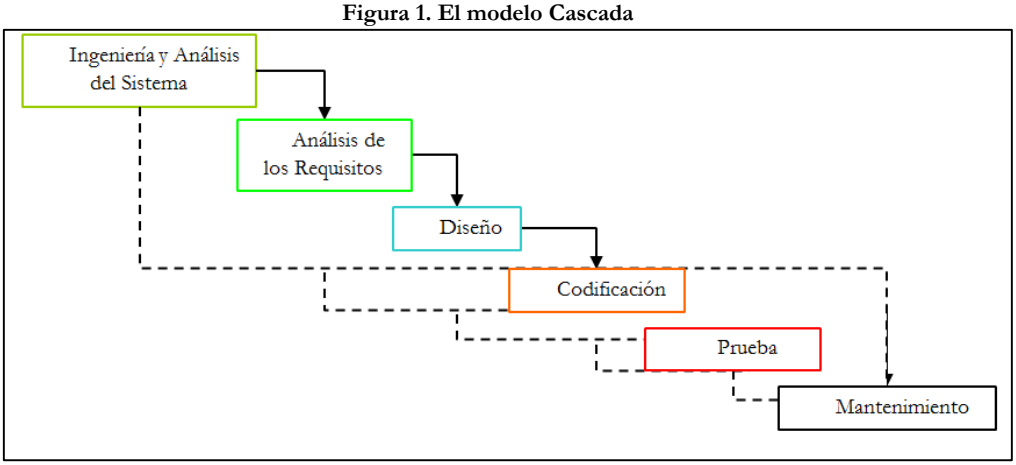
Actualmente, existen un gran número de enfoques y métodos que se mezclan para llegar a los objetivos trazados, ya sea por las organizaciones en general o para un área en especial, la tendencia a evolucionado de una manera acelerada en los últimos 20 años, tanto en el desarrollo de hardware como de software, integrando un gran número de herramientas físicas, virtuales y humanas que configuran un entramado en el cual la información se convierte en el elemento vital para su correcto funcionamiento (Pardini, (2018); López y Ruíz, 2017; Randstad, 2017); así mismo, cabe resaltar que dicha metodología es denominada clásica y tiene una tendencia a la planificación, Arboleda y Jiménez (2005) mencionan que esta planeación se da con único fin de alcanzar resultados predecibles dado que "La experiencia ha mostrado que, como consecuencia de las características del software, los resultados de los procesos no son siempre predecibles y, sobre todo, es difícil predecir desde el comienzo del proyecto cada resultado" (p.2).

Lo anterior permite ver una evolución corta en el tiempo, pero amplia en alcances pues sólo fue hasta 1980 que se logró hacer una revisión de las ideas expuestas; igualmente, como sucede en la ingeniería de sistemas en general, han existido avances relevantes, al tiempo que las nuevas tecnologías siguen avanzando rápidamente, lo que hace que los conceptos también se impulsen y se trasformen para explicarse y poder ser aplicados a las necesidades organizacionales.

2.2 Características del modelo cascada.

La principal característica del modelo de cascada es que sigue una secuencia lineal, esto permite identificar unas etapas específicas a las cuales se les debe dar cumplimiento en orden, a medida tal que se van implementando las adecuaciones pertinentes y las pruebas para mitigar los errores que se puedan presentar, hay que recordar que es un procesos sistémico, analítico, disciplinado y técnico para el desarrollo y mantenimiento de softwares, esto implica un trabajo amplio y enfatizado en la solución de las problemáticas particulares que se hayan detectado al momento de empezar a desarrollar el software (Roa, 2018; Sommerville, 2014; Moreno, 2010; Fernández, 2009). Dicho proceso se representa a continuación de manera gráfica, con el fin de exponer de qué se trata y así dar cuenta de sus características.

Figura 1. El modelo Cascada



Fuente: Rojas y Boucchechter (2005)

Cada uno de esos procesos tiene un objetivo específico, siguiendo lo descrito por Rojas y Boucchechter (2005) estos son:

- Ingeniería y análisis del sistema: todo comienza al establecer qué requisitos tiene el sistema. Para terminar con la asignación de funciones al software.
- Análisis de los requisitos del software: compete al analista revisar la información que recibe el software, para potenciar las funciones, rendimientos e interfaces.
- Diseño: es donde se estructuran los datos, se da arquitectura al software, así como los detalles procedimentales y caracterizar la interface.
- Codificación: con los diseños ya establecidos, se pasa a la traducción de los datos para la interpretación de la máquina, eso es lo que implica la codificación.
- Prueba: con los códigos obtenidos se procede a hacer pruebas del programa, específicamente en el manejo interno del software y sus actividades externas. Se busca de esta manera establecer que los datos ingresados generan los resultados esperados.
- Mantenimiento: esta etapa implica posibles cambios en caso de errores, o bien adaptaciones del software para enfrentar el entorno, también se pueden dar ajustes según los requerimientos del cliente.

Ese sería el ciclo de vida del software a partir de un diseño implementado desde el modelo de cascada, como se puede ver es de manera lineal, tiene una estructura rígida, la cual aporta como ventaja el seguimiento sistemático de la información, en donde la cuantificación del proceso se encuentra permanentemente relacionada con el desarrollo del programa. Los autores mencionados hasta este punto concuerdan en que es una metodología muy versátil, razón por la cual ha ganado adeptos entre los diseñadores de software.

A continuación, se hace énfasis en el proceso de modelado ágil (MA). Sin embargo, no se expone una profunda observación sobre su evolución histórica, por el contrario, se describen sus características, teniendo en cuenta que los orígenes parten directamente del modelo de cascada; esto debido a que, a raíz de la evolución tecnológica, el surgimiento de nuevas necesidades organizacionales y la globalización de la información, los modelos para diseñar nuevos softwares tienden a modificarse constantemente, siendo este el origen del MA.

3. CAPÍTULO 2. CARACTERÍSTICAS DEL MODELO ÁGIL.

Este modelo puede definirse como un cúmulo de procedimientos, estrategias y análisis que buscan desarrollar softwares de manera dinámica; en otras palabras, Ambler (2002) lo define como "una colección de prácticas, guiadas por principios y valores que pueden ser aplicados por profesionales de software en el día a día. AM no es un proceso prescriptivo, ni define procedimientos detallados de cómo crear un tipo de modelo dado" (p. 1). Lo dicho representa grandes ventajas y muchas posibilidades al momento de emprender el diseño de un programa, el cual optimiza de manera general los campos productivos, especialmente se ha usado en los procesos de mercadeo y logística (Roa, 2018; Sommerville, 2014; Moreno, 2010; Fernández, 2009). Retomando a Ambler (2002), este explica tres características importantes, cuando dice que permite:

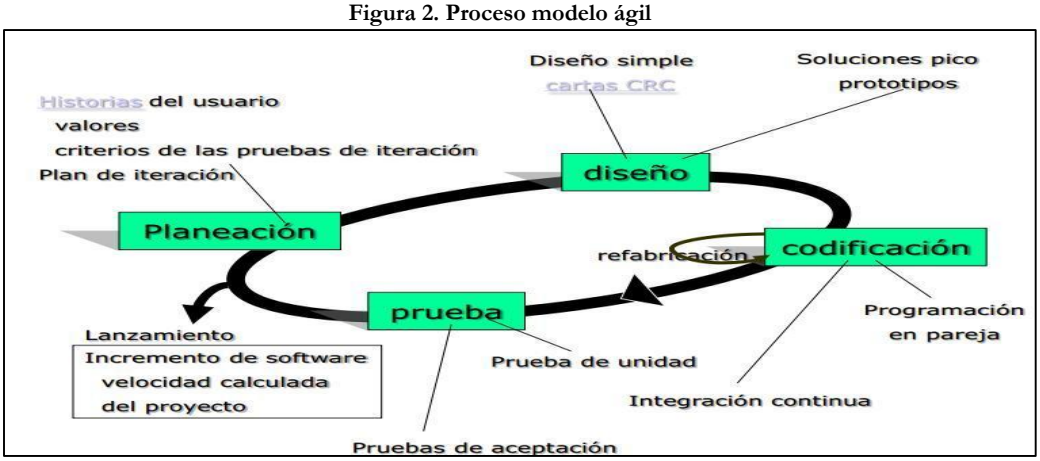
- Definir y mostrar cómo poner en práctica una colección de valores, principios y prácticas que conlleven a un modelado ligero efectivo.
- Explorar la aplicación de técnicas de modelado en proyectos de software a través de un enfoque ágil, tal como XP, DSDM o SCRUM.
- Explorar el cómo mejorar el modelado bajo procesos prescriptivos, tales como el Proceso Rational Unificado (RUP). (p.2)

Son muchos los beneficios que se pueden obtener al momento de planificar un desarrollo se software a través de una MA, dado que presenta "un énfasis en individuos e interacciones sobre procesos, colaboración con clientes sobre contratos y negociaciones formales, y capacidad de respuesta sobre la planificación rígida" (Serrador y Pinto, 2015, p. 1).

Es decir, está enfocado al cliente, a sus necesidades y a las necesidades contextuales de la interacción social, esto abarca un número de variables significativas, lo que mejora la capacidad de análisis y así se disminuye el riesgo de fracaso o de error en el desarrollo del programa.

En la siguiente figura se podrá observar de manera detallada en qué consiste el proceso de modelación ágil, el cual no comprende una secuencia definitiva de pasos a desarrollar, por el contrario, parece ser un vaivén, en el cual se pueden identificar las opciones más adecuadas para implementar en el diseño del programa:

Figura 2. Proceso modelo ágil



Fuente: Serrador y Pinto (2015)

Como se puede observar, en esta metodología hay un enfoque por saber cómo serán las interacciones entre el cliente o usuario y los procesos éticos en relación con los valores sociales y técnicos de los procesos de diseño, es una herramienta que permite, como ya se había mencionado, una especia de vaivén en donde se puede llegar a reestructurar el proyecto dependiendo de las características propias del objetivo que se quiere cumplir, la información y la integración continua es la parte vitar de este modelo de desarrollo de programas (Serrador y Pinto, 2015).

4. CAPÍTULO 3. TÉCNICAS PARA LA ADMINISTRACIÓN E IMPLEMENTACIÓN DE LAS METODOLOGÍAS Y SU IMPACTO EN LA ORGANIZACIÓN

En este apartado se busca examinar las técnicas utilizadas para la administración e implementación, tanto de la metodología en cascada como de la ágil y el impacto en la estructura de la organización, en el recurso humano y en los roles que se desempeñan en el interior de cada una.

4.1 Metodología Ágil, técnicas para la administración e implementación y su impacto.

En principio, se debe decir que la metodología ágil, como su nombre lo indica, se basa sobre técnicas de implementación que incluyen:

> Valores y principios ágiles, como se describe en el Manifiesto Agile, prescribir las filosofías y pautas generales que subyacen en el TEA." Estas indican los principios legales sobre el desarrollo de sistemas integrados de softwares, sus políticas de privacidad y sus métodos de implementación, así, en algunos casos se puede decir que "Los métodos de AII se componen de varias prácticas de AII. Por ejemplo, XP se compone de prácticas de TEA tales como refactorización, propiedad colectiva, e integración continua. (Beck citado por Serrador y Pinto, 2015, p. 12)

Cabe resaltar que en muchas ocasiones las prácticas parecen contradictorias, sin embargo, si existe una contradicción es precisamente porque de eso se trata, de confrontar diferentes técnicas para la administración e implementación.

Serrador y Pinto (2015) mencionan que a partir de las técnicas ágiles se pueden desarrollar softwares adaptativos ya que:

> El software adaptativo de alta calidad es desarrollado por Pequeños equipos que utilizan los principios de la continua. Mejora de diseño y pruebas basadas en rápida retroalimentación y cambio Liderazgo y colaboración Tácito Informal El modelo evolutivo de entrega. Orgánico (flexible y participativo alentador. acción social cooperativa), dirigida a pequeños y organizaciones medianas Control continuo de requerimientos, diseño. y soluciones además de la realización de pruebas continuas. (p.13)

Con lo anterior, se puede establecer que el impacto es alto, pues en él proceso de diseño, hasta el proceso de implementación, se hacen constantes evaluaciones, las cuales permiten medir el diseño del software y su capacidad, requerimientos y limitaciones, tiene un impacto directo en la formación de interacciones entre los procesos y actores, ya sea el cliente o los usuarios de dicho sistema (Manjul, Joey y Weidong, 2018; Highsmith, y Cockburn, 2017; Stoica, y Mircea, 2013; Beng et al, 2012); igualmente, el modelo desarrolla técnicas dialógicas en donde la información circula de manera abierta, esto con el fin de identificar las necesidades, prioridades y ventajas que se tienen que satisfacer de la manera más adecuada.

4.2 Metodología de Cascada, técnicas para la administración e implementación y su impacto.

La metodología de cascada, por su parte presenta un número más limitado de técnicas, las cuales no se pueden integrar en su totalidad; sin embargo, es preciso mencionar que su estructura rígida si permite una comunicación más directa, lo que posibilita prestar atención a un sólo punto por momento, hay que tener en cuenta que el modelo cascado es el primer modelo que existió para el diseño desarrollo e implantación de software al ser una técnica lineal. A riesgo de reiterar lo descrito anteriormente, se enfatiza en la técnica lineal tal como la menciona Hirsch, (2018) cuando dice que en el proceso de establecer qué se requiere para un proyecto es necesario determinar los tiempos de implementación de fases, a fin de alcanzar una predicción confiable de posibles problemas que deban corregirse, para este autor:

> Una vez él se establecen los requisitos, el siguiente paso se mueve hacia la fase de diseño y planificación arquitectónica en donde la infraestructura técnica se produce en forma de diagramas o modelos. Estos traen a la superficie el potencial y los problemas que el proyecto puede enfrentar a medida que avanza y proporciona una hoja de ruta viable para que los desarrolladores puedan implementar. (p.163)

Así pues, las técnicas utilizadas a partir del modelo de cascada son específicas, en cada uno de los procesos se puede implementar una técnica diferente. No obstante, es complicado utilizar diferentes técnicas en los procesos, puesto que no aplican de ninguna manera. Igualmente, se puede decir que dichas técnicas tienen una rigurosidad alta al momento de evaluar cada proceso por separado. La metodología permite evidenciar las problemáticas que se puedan presentar (Hirsch, 2018; Hoeren, y Pinelli, 2018; García, Rodríguez, y Villanueva, 2017; Cecil, 2015; Paasivaara y Lassenius, 2014). El impacto del desarrollo de esta modalidad es intenso, aunque no muy alto, es decir, se puede mejorar las condiciones, siempre y cuando esté dentro de la estructura de desarrollo por etapas y cumpliendo los objetivos que se planteen en cada una de ellas.

5. CAPÍTULO 4. ALGUNOS CASOS DE ÉXITO EN ORGANIZACIONES A NIVEL MUNDIAL

En este apartado se exponen algunos casos de éxito en organizaciones a nivel mundial en los que se haya aplicado la metodología en cascada y otros en donde se aplicó la metodología ágil. Adicionalmente se abordan los inconvenientes que se presentaron durante la implementación de cada caso. Cabe resaltar que existen muy pocos documentos que expongan casos específicos de implementación y evaluación; no obstante, la información rastreada da cuenta de diferentes evidencias, lo cual permite hacer una comparación y mirar sus alcances y sus límites.

5.1 Casos de éxito en el Modelo de Cascada.

Dado que existen limitantes para la implementación de este modelo, sobretodo en la poca y mala comunicación entre los equipos de trabajo y la indisciplina de los desarrolladores, que en muchos casos acarrean grandes costos a la

organización (Hirsch, 2018; Hoeren y Pinelli, 2018), se ha podido establecer que son amplios los casos de fracaso o de reestructuración del software, lo que conlleva a un estado de éxito pobre, "en muchos casos las empresas llegan a destinar hasta un 60% del presupuesto en cuestiones de mantenimiento y correcciones" (Navarro et al, 2013, p12). Lo dicho ya da un punto de partida sobre la evaluación del éxito en los modelos tradicionales. Además, el autor mencionado sostiene que en más de 200 casos registrados en empresas pequeñas, mediana y grandes fue posible encontrar que en la producción de software "se utilizan metodologías ágiles que proveen de una revisión continua a cada parte del proyecto. Sin embargo, esto no es suficiente para minimizar en gran parte la cantidad de errores relacionados al diseño de las interfaces" (p. 23).

Así mismo, un estudio realizado por el Standish Group, enfocado en analizar el desempeño de proyectos de software en empresas de todo el mundo, demuestra que el modelo cascada llegó a presentar en 2013 "un desempeño bajo (….) ya que únicamente el 14% de los proyectos llevados con esta metodología tuvieron resultados satisfactorios sin dificultad en el proceso" (Navarro et al, 2013, p. 30) lo que pone de relieve las dificultades que se presentan al momento de desarrollar un software con un modelo tan rígido como lo es el de cascadas.

5.2 casos de éxito en el modelo ágil.

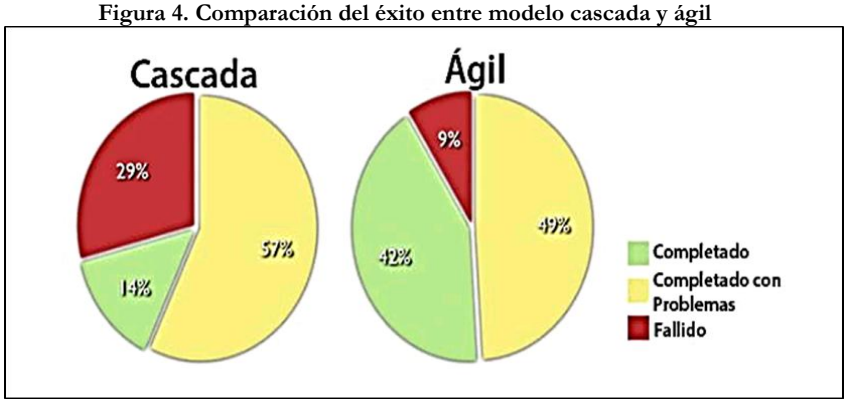
Por su parte, los modelos ágiles tienen "el 42% de los proyectos mostraron resultados satisfactorios" Navarro et al, 2013, p. 29) lo que en primera instancia indica que hay una efectividad por debajo del 50%, la cual no es alta, pero en comparación resulta mejor al momento de diseñar, desarrollar e implementar software con mejores resultados. Igualmente, en los casos expuestos por López y Ruíz (2017); Randstad (2017); Pardini, (2018); Rand, y Eckfeldt, B (2017); Stare, (2014), se puede observar que al haber una mejor interacción entre los componentes y actores en el proceso de diseño e implementación del software se puede hacer una mejor gestión del requerimiento; así mismo, las evaluaciones al ser constantes, van generando resultados de manera consecutiva. En dicho sentido, como sostiene Navarro et al., (2013), a través de la agilidad se consigue fomentar una actitud y estructura de grupo que potencia la comunicación, ya sea entre los diversos equipos, o bien entre las áreas de tecnología, ejecutivos y desarrolladores de software; para este autor "La agilidad adopta al cliente como parte del equipo ágil de desarrollo. Todos los stakeholders trabajan juntos y promueven la colaboración entre ellos. La agilidad pone más énfasis en la entrega rápida de software operacional y menos énfasis en productos de trabajo intermedios" (p.13)

Así pues, existe un mayor número de casos afortunados y de éxito en los softwares desarrollados a partir de un modelo ágil, esto demuestra una vez más la evolución de los conceptos, las prácticas y técnicas que se han venido desarrollando a lo largo de la historia en la ingeniería de sistemas. Es importante recordar que la bibliografía es escasa, de ahí que lo mejor fue estudiar algunos casos y algunas compilaciones, a través de las cuales se pudo dar cuenta del grado de éxito de cada una de las modalidades estudiadas aquí para el diseño, desarrollo e implementación de software.

6. CAPÍTULO 5. COMPARACIÓN DE LAS METODOLOGÍAS DE CASCADA Y ÁGIL

En este capítulo se hace una comparación entre las dos metodologías expuestas, con el fin de analizar las ventajas y desventajas que cada una de ellas ofrecen, para tal fin, en primera medida se presenta una imagen la cual enseña el grado de éxito de las modalidades, y así se empieza el proceso comparativo, de lo más general a lo más particular, profundizando en las técnicas y métodos de cada una de ellas.

Figura 4. Comparación del éxito entre modelo cascada y ágil



Fuente: Prieto (2015)

La anterior figura ya da indicios de las problemáticas que aún existen, tanto en el modelo cascado como en el ágil; sin embargo, los modelos ágiles han demostrado una mayor efectividad al momento de desarrollar software, de todas maneras, el índice de efectividad en el desarrollo y la implementación sigue siendo alto, lo que sugiere que aún falta bastante camino por recorrer en dicho aspecto. La siguiente tabla enseña las principales diferencias de los modelos, esto aportará a la comparación y permitirá identifica las características que les permite a cada modelo mostrar sus ventajas y desventajas

Figura 5. Comparación del modelo de cascada y el modelo ágil

	Desarrollo tradicional	Desarrollo ágil
Suposición fundamental	Los sistemas son totalmente especificables, predecibles y se construyen a través de una planificación meticulosa y extensa.	El software adaptativo de alta calidad es desarrollado por Pequeños equipos que utilizan los principios de la continua.
Estilo de gestión Conocimiento administrativo Comunicación Modelo de desarrollo Forma / estructura organizativa deseada	Comando y control Explícito Formal Modelo de ciclo de vida Mecanismo (burocrático con alta formalización), dirigido a grandes organizaciones	Mejora de diseño y pruebas basadas en rápida retroalimentación y cambio Liderazgo y colaboración Tácito Informal El modelo evolutivo de entrega. Orgánico (flexible y participativo alentador. acción social cooperativa), dirigida a pequeños y organizaciones medianas
Control de calidad	Planificación pesada y control estricto. Pruebas tardías y pesadas	Control continuo de requerimientos, diseño y soluciones. Pruebas continuas

Fuente: Hirsch (2018)

Teniendo en cuenta lo anterior, se puede decir que el modelo ágil se diferencia en gran medida del diseño de cascada, empezando porque el primero tiene un campo de acción más amplio, tienen en cuenta más variables y acumula más información contextual, esto ha ayudado a bajar el riesgo de fracaso al momento de desarrollar un software; sin embargo, tal como lo plantean Roa (2018); Moreno, 2010; el Instituto Nacional de Tecnologías de la comunicación (2009); Fernández (2009), no importa qué modelo se elija para el desarropo de aplicaciones de software, esta actividad implica procesos complejos que son a menudo predispuestos a los errores. Así, Sommerville (2014) menciona que:

> Es por eso que Más allá de la agilidad o el tradicionalismo, un punto importante es el rol que se le da a las pruebas y a la validación. Cualquier Sistema de software de alta calidad, diseñado por profesionales, desde su desarrollo e implementación debe ser probado y validado antes de entrar producción. El cliente debe saber que las especificaciones del proyecto. También, El cliente debe estar seguro del proyecto. Y si su funcionalidad es correcta (p, 56)

De ahí la importancia de la evaluación y he ahí la gran diferencia entre los modelos, mientras que el de cascada hace una evaluación procesal y rígida, la cual en ocasiones no permite la interacción entre los procesos y las técnicas, por su disposición rigurosa, no permite un flujo de información adecuado que permita conocer de manera íntegra las necesidades del cliente; por el contrario los modelos ágiles se desarrollan de manera más libre, lo que permite observar más variables, y su evaluación es constante, sin contar con un sistema rígido que lo estructure. Es por eso que el modelo ágil tiene a ser mejor en organizaciones medianas y pequeñas, mientras que el modelo de cascada es más efectivo en la gran industria (Sommerville, 2014; Fernández, 2009).

7. CONCLUSIONES

Después de haber realizado la correspondiente revisión teórica y documental en relación a los dos modelos de diseño de software mencionados a lo largo de este trabajo, se puede llegar a las siguientes conclusiones, las cuales no pretenden abarcar en su totalidad los estudios expuestos ni mucho menos los existentes, teniendo en cuenta las limitantes expresadas, se presentan de manera ordenada:

- La ingeniería de sistemas y el diseño de software han evolucionado paulatinamente, de manera constante y conjuntamente, uno no es sin el otro, tanto el hardware cambia, inmediatamente el software hace lo correspondiente y viceversa, esto con el objetivo de satisfacer las necesidades tecnológicas de las empresas y de la sociedad en general.

- Al interior de las organizaciones se ha hecho necesario la implementación de software que mejoren los procesos, ya sean logísticos, contables, administrativos etc., sin embargo, para poder diseñar un software se debe tener en cuenta las necesidades del área, su interacción con las otras dependencias y recolectar la mayor información posible.

- El desarrollo de software comprende diferentes momentos, como lo es el diseño, el desarrollo, la implementación y la evaluación; sin embargo, también existen otros modelos de desarrollo los que pueden contar o no con dichas etapas, o las contemplan con algunas variaciones.

- El modelo de cascada fue el primero que se desarrolló, sin embargo, cuenta con múltiples limitaciones, esto debido a que es un sistema que se puede llamar rudimentario, que utiliza la fuerza para obtener resultado, un ejemplo es la rigidez de sus operaciones, la poca integración en los procesos, la poca coordinación de las herramientas y la precaria evaluación de estos, lo que ha hecho que se generen fallas y perdidas por el proceso de rediseño.

- El modelo ágil, por su parte, es un compendio de técnicas, el cual ya ha perfeccionado sus operaciones, pero sigue presentando dificultades, aunque no tanto como el modelo de cascada. Este modelo es un modelo más flexible que ayuda a obtener mayor información al momento de diseñar y desarrollar un programa, esto hace que la evaluación sea constante, y que se presenten idas y vueltas en cada uno de los procesos, esto ha permitido tener un mayor conocimiento de las necesidades del cliente. Además, son software que tienen una capacidad de adaptabilidad lo que permite disminuir el riesgo de fallo.

- Es claro que hace falta evidencia empírica para demostrar la efectividad de las dos metodologías, existen limitaciones en las producciones literarias sobre el tema, lo que dificulta el análisis; no obstante, lo rastreado hasta el momento da cuenta de la importancia de la evaluación, sea cual sea el modelo que se escoja, también se ha demostrado que es más efectivo el modelo de cascada en organizaciones grandes, esto debido a que las evaluaciones de los procesos se hacen por independiente, además, al ser una estructura rígida no da cabida a

procesos aislados en materia de información. Mientras que la metodología ágil ha funcionado mejor en organizaciones pequeñas y medianas, ya que al ser reducidas se pueden integrar de manera más sencilla, teniendo la posibilidad de gestionar mejor los proyectos, al tener menos componentes es más fácil manejarlos.

- Es difícil determinar qué elementos son fundamentales para y en qué momento se hace mejor utilizar el sistema de cascada o el sistema ágil para el desarrollo del software. Aún existen casos de fallo y de reingeniería en muchos de los referentes estudiados, las variables parecen indicar que hacen falta estudios que permitan conocer a fondo las problemáticas existentes, lo que deja el campo abierto para que se analicen las variables que afectan los reprocesos de diseño en una gran cantidad de casos.

8. REFERENCIAS

Ambler, S (2002) Modelado Ágil (AM) Una introducción. Agile Moldeling. Estado Unidos de Norte América.

Beck, K; et al. (2011). Manifiesto por el Desarrollo Ágil de Software. Retrieved September 5, 2018, Recuperado de: http://agilemanifesto.org/iso/es/manifesto.html

Beng, T, et al (2012) Software Development Life Cycle AGILE vs Traditional Approaches. International Conference on Information and Network Technology (ICINT 2012)

Cecil, R. (2015). Agile Software Development, Patterns, and Practices. Pearson Education, Inc. Upper Saddle River, New Jersey 07458

Fernández A, (2009) Procesos de ingeniería de software Universidad El Bosque de Bogotá, Colombia.

Gabino, G. (2017) Qué son las metodologías ágiles y su aplicación en el mercado actual. Universidad de Valwncia, España.

García, M Rodríguez, V y Villanueva, J (2017) Analysis of the agile methodologies applied in software engineering within the framework of the PMBOK knowledge areas. 21th International Congress on Project Management and Engineering. Cádiz, 12th - 14th July 2017

Highsmith, J, Cockburn, A (2017) Agile Software Development: The Business of Innovation.

Hirsch, M (2018) Moving from a Plan Driven Culture to Agile Development. ZQhIke Engineering AG Wiesenstrasse 10a CH-8952 Schlieren

Hoeren, T y Pinelli, S. (2018) Agile programming Introduction and current legal challenges. Computer Law & Security review 34 (2018) 1131–1138

Instituto Nacional de Tecnologías de la comunicación (2009) CURSO DE INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE, Laboratorio Nacional de Calidad del Software. Madrid, España.

Jiménez, H (2005) Modelos de ciclo de vida en desarrollo de software, Revista Facis Edición N° 93 Julio - septiembre de 2005. Recuperado de: http://52.0.140.184/typo43/index.php?id=551

López, P y Ruiz, F (2017) INGENIERÍA DEL SOFTWARE Lenguaje Unificado de Modelado –UML Universidad Cantabria –Facultad de Ciencias.

Manjul G, Joey F., Weidong X (2019) Relationships between IT department culture and agile software development

practices: An empirical investigation. International Journal of Information Management 44 (2019) 13–24

Navarro et al, (2013) Revisión de metodologías ágiles para el desarrollo de software. Universidad Icesi. Cali, Colombia. Recuperado de: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=2ahUKEwjznLH5p57iAhUkhOAKHTFqCcYQFjAAegQIAhAC&url=https%3A%2F%2Fdialnet.unirioja.es%2Fdescarga%2Farticulo%2F4752083.pdf&usg=AOvVaw2Vx1COO2hbVc37Lq4rblXm

Paasivaara M, y Lassenius, C (2014) Deepening Our Understanding of Communities of Practice in Large-Scale Agile Development Department of Computer Science and Engineering, School of Science Aalto University POB 15400, 00076 Aalto, Finland

Pardini, B. (2018). Entrevista VP Product & Technology. Cali.

Pedro, P, (2015) Filosofía Lean aplicada a la Ingeniería del Software. Universidad de Sevilla. España.

Pickin, T y García, M (2015) Introducción a la Ingeniería del Software. Departamento de Ingeniería Telemática. Universidad Carlos III de Madrid. España.

Pressman, R. (2011). Desarrollo Ágil Software Engineering: A Practitioner's Approach. Coordinación de Ciencias Computacionales. INAOE

Pressman, R. (2011) Ingeniería del software un enfoque práctico. Derechos reservados © 2010, 2005, 2002 respecto a la tercera edición en español por Mcgraw-hill.

Prieto, C. (2015). Adaptación de las Metodologías Tradicionales Cascada y Espiral para la Inclusión de Evaluación Inicial de Usabilidad en el Desarrollo de Productos de Software en México. Huajuapan De León, Oax, México.

Rand, C y Eckfeldt, B (2017) Aligning Strategic Planning with Agile Development: Extending Agile Thinking to Business Improvement Cyrus Innovation 200 Varick Street, Suite 902 New York, New York 10014 USA

Randstad. (2017). Qué es la metodología ágil, y por qué es tan popular en TI | Randstad Chile. Retrieved September 5, 2018, Recuperado de: https://www.randstad.cl/tendencias360/archivo/que-es-la-metodologia-agil-y-por-que-es-tan- popular-en-ti_1463/

Roa, L. (2018). Ingeniería de Software Pontificia Universidad Javeriana, Cali.

Rojas, R y Boucchechter, I. (2005). Ingeniería del Software. Ciclo de vida, Mallorca, España.

Serrador, P y Pinto, J. (2015). Does Agile work? — A quantitative analysis of agile project success. ScienceDirect International Journal of Project Management 33 (2015) 1040 – 105.

Sommerville, I. (2014). Software engineering, 9th edition, by Ian Sommerville published by Pearson Education, Inc., publishing as Addison-Wesley, Copyright © 2011. All rights reserved.

Stare, A. (2014). Agile Project Management in Product Development Projects. University of Ljubljana, Faculty of Economics, Kardeljeva plošad 17, Ljubljana 1000, Slovenia.

Stoica, M Mircea, B. (2013). Software Development: Agile vs. Traditional Software Development: Agile vs. Traditional