

Los enfoques ágiles y la teoría del conocimiento aplicados a los proyectos de desarrollo de software

Carolina Arboleda Blandón

Trabajo de grado para optar al título de Magíster en Administración de Negocios

Asesora

PhD., MSc. Mónica Henao Cálad

Universidad EAFIT

Escuela de Administración

Maestría en Administración de Negocios (MBA)

Medellín

2018

Contenido

1. Introducción	6
2. Marco teórico de gestión de proyectos	10
2.1 ¿Qué es un proyecto?	12
2.2 Metodologías ágiles	15
2.2.1 Estado de la implementación de metodologías ágiles en proyectos de software	23
2.2.2 Riesgos con el uso de metodologías ágiles	27
2.3 Metodologías tradicionales	31
2.3.1 Estado de la implementación de metodologías tradicionales en proyectos de software	41
2.3.2 Riesgos en el uso de aplicación de metodologías tradicionales	46
2.4 Teoría del conocimiento	47
3. Aspectos metodológicos de la investigación	53
4. Análisis de proyectos de desarrollo de software en el mercado, desde los elementos de la teoría de creación de conocimiento	57
4.1 Presentación de casos	59
4.1.1 Caso LAND	59
4.1.1.1 Resumen del proyecto	59
4.1.1.2 Generación de valor para el cliente en el proyecto	61
4.1.1.3 Participación de los usuarios en el proyecto	61
4.1.1.4 Desarrollo de conocimiento explícito y tácito en el proyecto	63
4.1.1.5 Gestión de conocimiento en el proceso de requisitos	64
4.1.1.6 Liderazgo de equipos	65
4.1.2 Caso STORM	66

4.1.2.1	Resumen del proyecto	66
4.1.2.2	Generación de valor para el cliente en el proyecto	68
4.1.2.3	Participación de los usuarios en el proyecto	69
4.1.2.4	Desarrollo de conocimiento explícito y tácito en el proyecto	71
4.1.2.5	Gestión de conocimiento en el proceso de requisitos	72
4.1.2.6	Liderazgo del equipo	73
4.1.3	Caso SILVER	74
4.1.3.1	Resumen del proyecto	74
4.1.3.2	Generación de valor para el cliente en el proyecto	77
4.1.3.3	Participación de los usuarios en el proyecto	78
4.1.3.4	Desarrollo de conocimiento explícito y tácito en el proyecto	80
4.1.3.5	Gestión de conocimiento en el proceso de requisitos	81
4.1.3.6	Liderazgo del equipo	83
4.1.4	Caso ROCK	83
4.1.4.1	Resumen del proyecto	83
4.1.4.2	Generación de valor para el cliente en el proyecto	85
4.1.4.3	Participación de los usuarios en el proyecto	87
4.1.4.4	Desarrollo de conocimiento explícito y tácito	88
4.1.4.5	Gestión de conocimiento en el proceso de requisitos	889
4.1.4.6	Liderazgo del equipo	90
4.1.5	Caso PLATINUM	91
4.1.5.1	Resumen del proyecto	91
4.1.5.2	Generación de valor para el cliente en el proyecto	94
4.1.5.3	Participación de los usuarios en el proyecto	95
4.1.5.4	Desarrollo de conocimiento explícito y tácito en el proyecto	97
4.1.5.5	Gestión de conocimiento en el proceso de requisitos	98
4.1.5.6	Liderazgo del equipo	100
4.2	Análisis comparativo entre los proyectos presentados	100
5.	Planteamiento de una propuesta de mejora en el desarrollo de proyectos de software	108
6.	Conclusiones	113
	Referencias	116

Lista de tablas

	Pág
Tabla 1. Características de los proyectos en estudio	58
Tabla 2. Actividades dentro de iteraciones	60
Tabla 3. Actividades en las iteraciones	75
Tabla 4. Actividades dentro del proyecto	84
Tabla 5. Actividades en el proyecto	92
Tabla 6. Calificación de variables caracterizadas por proyecto	102
Tabla 7. Resultados por proyecto	102
Tabla 8. Calificación de las fases de conocimiento por proyecto	106
Tabla 9. Resultados por proyecto	107

Lista de ilustraciones

	Pág
Ilustración 1. Encuesta CHAOS, reporte 2015	17
Ilustración 2. Fuentes de incertidumbre en proyectos	18
Ilustración 3. Procesos de un proyecto	32
Ilustración 4. Áreas de conocimiento y grupos de procesos en dirección de proyectos	35
Ilustración 5. Interacción entre proyectos, programas y portafolios	36
Ilustración 6. Capacidad de entregar valor al cliente	44
Ilustración 7. Métricas de rendimiento de proyectos	45
Ilustración 8. Dimensiones de creación del conocimiento de Nonaka y Takeuchi (The Knowledge creating company)	49
Ilustración 9. Modos para la creación de conocimiento	50
Ilustración 10. Modelo en cinco fases del proceso de creación de conocimiento organizacional	52
Ilustración 11. Conversión de conocimiento en el proyecto LAND	64
Ilustración 12. Conversión de conocimiento en el proyecto STORM	72
Ilustración 13. Conversión de conocimiento en el proyecto SILVER	81
Ilustración 14. Conversión de conocimiento en el proyecto ROCK	89
Ilustración 15. Conversión de conocimiento en el proyecto PLATINUM	98

Resumen

Las metodologías de gestión de proyectos en general, pretenden organizar las actividades al interior del proyecto con el fin de garantizar el logro de los resultados esperados por el cliente y por la empresa que lo implementa. Sin embargo alcanzar el éxito es cada vez más distante de la realidad del sector y es evidente que año tras año el volumen de proyectos fallidos supera el margen esperado, lo cual desalienta los esfuerzos que las comunidades de tecnología realizan al definir nuevas prácticas y promulgar metodologías emergentes, que si bien son asumidas y ejecutadas por unas pocas empresas, no alcanzan a ser digeridas e interiorizadas por la gran mayoría de empresas del sector.

Standish Group (2015) indicó que solo el 29% de los proyectos que se planean terminan con éxito y los 71% restantes terminan con cambios o no terminan. Estos resultados indican que es imperativo realizar cambios sustanciales que más allá de estar enfocados en las variables típicas de tiempo, costo y alcance, estén centradas en las personas de la organización y en la interacción que estas generan con el cliente. Takeuchi y Nonaka (1986) visualizaron que en los productos de innovación o proyectos tecnológicos, la generación de valor está en las entrañas del conocimiento y en el flujo de energía que produce conocimiento en el trabajo de equipos interfuncionales, que conviven en un conflicto dinámico de ideas y que posibilitan la creación y la transferencia de conocimiento.

El presente trabajo genera un estudio teórico de los principales conceptos que rodean la ejecución de proyectos de desarrollo de software y lo confronta con una investigación explicativa y un análisis detallado de la implementación de una muestra de proyectos del mercado, para finalmente definir propuestas que desde el ámbito de la teoría de conocimiento logren aportar un punto de vista diferente en la búsqueda del éxito de los proyectos de software.

Palabras claves: Teoría del conocimiento, Proyectos de desarrollo de software, Metodologías ágiles, Metodologías tradicionales.

Abstract

The methodologies of project management in general, intend to organize the activities inside the project in order to ensure the achievement of the results expected by the client and by the company that implements it. However, reach success it is increasingly distant from the reality of the sector and it is evidente that year after year the volume of failed projects exceeds the expected margin, which discourages the efforts made by technology communities to define new practices and enact a emerging methodologies, that although they are assumed and executed by a few companies, do not manage to be digested and internalized by the great majority of companies in the sector.

Standish Group (2015) indicated that only 29% of the projects that are planned end successfully and the remaining 71% end up with changes or do not end. These results indicate that it is imperative to make substantial changes that, beyond being focused on the typical variables of time, cost and scope, are centered on the people of the organization and on the interaction they generate with the client. Takeuchi and Nonaka (1986) visualized that in the products of innovation or technological projects, the generation of value is in the bowels of knowledge and in the flow of energy that produces knowledge in the work of interfunctional teams, which coexist in a dynamic conflict of ideas and that enable the creation and transfer of knowledge.

Key words: Knowledge theory, Software development projects, Agile methodologies, Traditional methodologies.

1. Introducción

De acuerdo con el último estudio publicado por Gartner (2016), el desarrollo de proyectos en las organizaciones no está evolucionando hacia mecanismos que permitan obtener los resultados planeados y cumplir las expectativas del cliente; en 2015 solo el 60% de los proyectos se completaron en el tiempo esperado y solo el 71% finalizó con el presupuesto programado.

La variación respecto al tiempo esperado fue del 23% y la variación respecto al presupuesto fue del 16% aproximadamente. A su vez, dichos proyectos se ejecutaron en un 56% bajo metodologías tradicionales, 21% bajo metodologías iterativas y 23% bajo metodologías ágiles. De estos proyectos el 39% tuvo una duración inferior a tres meses y el 20% superó el año de duración.

En este mismo sentido, Standish Group (2015) indica que en 2015 solo el 29% de los proyectos terminaron con éxito y 71% terminaron con cambios o no terminaron. De los proyectos que concluyeron con éxito, 39% se desarrollaron con metodologías ágiles y el 11% con metodologías tradicionales, de los proyectos que no terminaron o tuvieron cambios en su proceso, 61% fueron desarrollados con metodologías ágiles y 89% fueron desarrollados con metodologías tradicionales.

Adicional a esto, la comparación en el tiempo, realizada por Standish Group (2015) no muestra evolución con el transcurrir de los años, en 2011 los proyectos fallidos o con cambios corresponden al 71%, mientras que los terminados con éxito corresponde al 29%; en 2012 los proyectos fallidos o con cambios corresponden al 73%, mientras que los terminados con éxito corresponde al 27%; en 2013 los proyectos fallidos o con cambios corresponden al 69%, mientras que los terminados con éxito corresponde al 31%; en 2014 los proyectos fallidos o con cambios corresponden al 72%, mientras que los terminados con éxito corresponde al 28%.

Los resultados indican que se presentan problemas que impiden alcanzar el éxito consistente en los proyectos y que se deben buscar estrategias que permitan

realizar cambios sustanciales en la definición y gestión de los proyectos, con miras a la consecución de los resultados esperados por la organización.

Si se ahonda en la metodología ágil, las principales barreras que enfrenta una organización para adoptar la filosofía ágil están relacionadas en un 63% por una cultura organizacional en desacuerdo con los principios ágiles y un 47% por la falta de habilidades o experiencia en el uso de metodologías ágiles.

De otro lado, si se ahonda en las metodologías tradicionales, el estudio publicado por Standish Group (2009) y citado por Donaval Neil Thompson (Neil, 2006), presenta una clasificación acerca de las principales causas de los fracasos de los proyectos distribuida en: 13,1% atribuibles a requerimientos incompletos, 12,4% falta de involucramiento de los usuarios y 10,6% falta de recursos.

Conforme a lo expuesto por Matthew Hodgson (2012), la diferencia entre las metodologías ágiles y metodologías tradicionales en el desarrollo de proyectos, trasciende los problemas de cultura organizacional, planificación, análisis y comunicación, y llevan a cuestionar si podría existir de fondo un problema de conocimiento. El autor formula que mientras en los enfoques ágiles el conocimiento tácito, fundamentado en la experiencia, las aptitudes, las percepciones, las ideas que residen en las personas y en los equipos, son fundamentales para la consecución de los objetivos, en las metodologías tradicionales no es permitido.

Mientras que en la metodología ágil no es obligatorio el conocimiento explícito que se codifica y se transporta a otros, a través de medios de comunicación, como documentos; en metodologías tradicionales sí es necesario, dado que se centra en la documentación para transferir el conocimiento y las necesidades del proyecto.

Desde el planteamiento mismo de Takeuchi y Nonaka en 1986, se ha tratado de profundizar en la esencia del conocimiento que genera “ba”, el flujo de energía que produce conocimiento en medio de equipos interfuncionales involucrados en el conflicto dinámico de ideas, que permite la creación y transferencia de conocimiento y que forma nuevos productos (Takeuchi & Nonaka, 1986).

Posterior a esto, las metodologías o técnicas desarrolladas en el mundo occidental podrían ser vistas como efectos secundarios de la generación de dicho conocimiento. Galvis Lista (2016) ha planteado en su estudio exploratorio sobre la implementación de procesos de gestión de conocimiento en organizaciones de tecnología en Colombia, que si bien existe un reconocimiento por la importancia de la gestión del conocimiento, la implementación de dichos procesos tiene un estado apenas aceptable y aún existe una marcada necesidad de fortalecer estos procesos, para generar la ventaja competitiva y la sostenibilidad requerida por la industria.

Esto evidencia la necesidad de continuar explorando en la investigación de generación de conocimiento en las organizaciones, como eje fundamental de innovación y calidad. La mejora continua esperada en el desarrollo de proyectos es una necesidad que ha sido explorada desde los inicios de la teoría de conocimiento, documentada por Takeuchi y Nonaka en Toyota, Honda, Canon, Fuji-Xerox, 3M, HP y otras organizaciones de alto desempeño y, es a partir de sus experiencias y prácticas, que se podrían plantear las acciones y estrategias requeridas en el desarrollo de proyectos, que logren los objetivos propuestos en el tiempo, costo y presupuesto planteados por la organización.

Así, se definió como objetivo general de este trabajo el proponer alternativas que posibiliten el logro de los objetivos de los proyectos de software, a partir del análisis comparativo de los enfoques ágiles y enfoques tradicionales en el marco de la teoría del conocimiento. Como objetivos específicos se tienen: reconocer las bases filosóficas y teóricas de la concepción y evolución de los enfoques ágiles, los enfoques tradicionales, la teoría del conocimiento y la gestión del conocimiento para el desarrollo de proyectos; caracterizar los proyectos de desarrollo de software que han sido exitosos en el mercado; caracterizar los proyectos de desarrollo de software que han sido fallidos en el mercado; realizar análisis comparativo entre los proyectos exitosos y proyectos fallidos en el mercado; proponer mejoras aplicables en el desarrollo de proyectos con base en la filosofía de proyectos ágiles, la teoría de conocimiento y su gestión.

El presente documento contiene una exploración de los conceptos que definen las metodologías ágiles, metodologías tradicionales y la teoría del conocimiento, para obtener postulados que sirvan de insumo en el análisis de mercado de casos de proyectos de software exitosos y fallidos, que fueron implementados bajo las dos metodologías, y encontrar las características diferenciales que finalmente permitan exponer propuestas de mejoras, que logren beneficiar la industria tecnológica.

2. Marco teórico de gestión de proyectos

La gestión de proyectos ha existido hace cientos de años, pasando por eventos significativos como la construcción del arca de Noé, las pirámides egipcias, la muralla china, el coliseo y los acueductos romanos; sin embargo, la práctica como se conoce actualmente, explotó durante la Guerra Fría en la postguerra de la segunda guerra mundial, cuando el desarrollo de proyectos alcanzó dimensiones sin precedentes y se hizo necesario plantear herramientas para planear los proyectos y ejecutarlos de manera eficiente.

Henry Gantt, principal exponente de técnicas de planteamiento y control, quien es famoso por la creación del diagrama de Gantt (1910), creada en su época como herramienta de gestión de proyectos, y Henri Fayol, reconocido por la creación de las cinco funciones de gestión, estudiaron las teorías sobre la organización científica y crearon diferentes herramientas para la gestión de proyectos que facilitaban el control del alcance y la asignación de los recursos requeridos por el proyecto. Fue en la década de los 50's que se dio inicio a la gestión moderna de proyectos con el reto de crear una visión estructurada, que permitiera alcanzar a través de la definición de procesos secuenciales, los objetivos del proyecto.

A medida que el mundo ha evolucionado y como parte de un esfuerzo para mejorar la competitividad en los negocios, desarrollar estrategias que permitan lograr los objetivos organizacionales, materializar sus metas y lograr su visión, las empresas han estado en una búsqueda constante de mejorar la definición de sus proyectos. Esto implica encontrar la manera adecuada de gestionar las personas, desarrollar el conocimiento, gestionar el alcance pactado con el cliente y cumplir con el proyecto en el tiempo prometido y con el costo presupuestado.

Lograr esta gestión apropiada de proyectos implica que la organización conozca sus fortalezas y debilidades, identifique su entorno para aprovechar oportunidades y

mitigar amenazas y, sobre todo, que pueda reconocer en su proceso de aprendizaje, la manera apropiada en que desarrolla y gestiona el conocimiento en sus procesos internos.

En su obra, *El juego de desarrollo de nuevos productos*, Hirotaka Takeuchi e Ikujiro Nonaka (1986), ya hablaban de la evolución de la competitividad en los mercados, en los que los desarrollos de nuevos productos requerían más que alta calidad, bajos costos y diferenciación. Se hacía imperativo entonces ganar velocidad y flexibilidad en las compañías que querían evolucionar en su entorno. Esto requería trascender de las metodologías secuenciales tradicionales de desarrollo de nuevos productos, a una metodología holística que podría ser ejemplificada por el juego de “rugby” en el que un equipo trabaja como una unidad, pasando de tirar y recoger la pelota a mejorar su estrategia, utilizando requerimientos competitivos. Esta es quizás la primera aproximación a metodologías ágiles, donde ya se trataban términos como *scrum*, equipos auto organizados y desarrollo de productos con un enfoque de mejoramiento incremental (Takeuchi & Nonaka, 1986).

En dicho artículo, Takeuchi y Nonaka (1986) evalúan cómo algunas compañías de Japón y EEUU utilizaron el “rugby” como un modelo de equipo de proyectos auto-organizados, con multi-aprendizaje, transferencia organizacional de conocimiento, que formaban un proceso rápido y flexible para el desarrollo de nuevos productos, con el propósito de tomar un nuevo enfoque para la consecución de los resultados. Los autores contribuyeron desde entonces a una investigación enfocada en el mercado, la competencia global, la organización estratégica y el mercadeo (Takeuchi & Nonaka, 1986).

A partir del planteamiento de los autores, se podría cuestionar el desarrollo de productos bajo una metodología tradicional con fases secuenciales, separadas o segmentadas, en contraposición con la competitividad que se logra alcanzar en metodologías ágiles, en las que existe una constante interacción en equipos

multidisciplinarios que trabajan juntos para comenzar y terminar el proyecto, en fases de desarrollo del producto superpuestas y transferencia del aprendizaje organizacional.

Una gestión adecuada de proyectos no solo implica contar con una metodología apropiada para las condiciones de la organización, ni basta con estructurar una serie de pasos, documentos o consideraciones en la administración del proyecto, sino que el reto es trascender en la capacidad de administrar, aplicar el conocimiento de manera efectiva y considerar la gestión del conocimiento como una responsabilidad de gestión estratégica, basada en la experiencia de aprendizaje colaborativo organizacional.

En medio de la era del conocimiento y en un ambiente globalizado en el que las condiciones y reglas para el desarrollo de proyectos que fomenten la innovación han estado cambiando, se ha descubierto que las empresas deben trascender de los principios básicos de alta calidad, bajo costo y diferenciación, para sobresalir en el entorno competitivo actual. En medio de esta evolución entran a predominar conceptos como la velocidad y la flexibilidad, reflejados en el desarrollo de nuevos productos que entran a jugar un papel imperativo en el desarrollo de nuevas ventas y nuevas ganancias para la empresa.

Este apartado abordará las bases epistemológicas de la gestión de proyectos, la concepción de la teoría del conocimiento como eje fundamental en el éxito de la organización y cómo la gestión de proyectos bajo metodologías ágiles o proyectos gestionados bajo metodologías tradicionales, puede contribuir al logro de las expectativas y los objetivos planteados por la organización.

2.1 ¿Qué es un proyecto?

Algunas definiciones por considerar, emitidas por autores e instituciones reconocidas, son:

Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. La naturaleza temporal de los proyectos implica que un proyecto tiene un principio y un final definidos. El final se alcanza cuando se logran los objetivos del proyecto, cuando se termina el proyecto porque sus objetivos no se cumplirán o no pueden ser cumplidos, o cuando ya no existe la necesidad que dio origen al proyecto. (Guide, 2013, p. 4)

“Un proyecto es un esfuerzo complejo, no rutinario, limitado por el tiempo, el presupuesto, los recursos y las especificaciones de desempeño, que se diseñan para cumplir con las necesidades del cliente” (Larson & Gray, 2011, p. 27).

“Un proyecto es un esfuerzo por lograr un objetivo específico mediante una serie especial de actividades interrelacionadas y la utilización eficiente de los recursos” (Tobergte & Curtis, 2013, p. 2).

En este sentido, se podría considerar que un proyecto debe tener como características fundamentales, su carácter de temporalidad, es decir, debe finalizar en una fecha programada, debe contar con unos recursos presupuestados y tiene un desempeño que se mide en términos del cumplimiento de los objetivos planteados en la definición del proyecto.

Los trabajos secuenciales o repetitivos que existen en la organización no serían considerados proyectos, dado que no existe una fecha de finalización que permita medir el logro del resultado. Es importante entonces diferenciar lo que sería un proyecto para la empresa y lo que debe ser considerado como actividades de soporte y mantenimiento que no tiene el carácter de proyecto. El trabajo operativo es aquel que se realiza de forma constante para mantener el negocio y los procesos de la organización, mientras que el trabajo de proyecto tiene un inicio, un alcance determinado y se finaliza cuando se cierra el proyecto.

A pesar del esfuerzo desde las diferentes metodologías de definir proyectos que cumplan con las expectativas de la organización, en la actualidad se tienen altas cifras de fracaso que cuestan para la industria del software miles de dólares y el cierre de pequeñas y medianas empresas, que no pueden subsistir ante pólizas y contratos establecidos con sus clientes.

En el estudio titulado The Chaos Report publicado por Standish Group en 2015 (Neil, 2006), se presenta la siguiente clasificación acerca de las causas de los fracasos de los proyectos:

- 13,1% Requerimientos incompletos
- 12,4% Falta de involucramiento de usuarios
- 10,6% Falta de recursos
- 9,9% Expectativas no realistas
- 9,3% Falta de soporte ejecutivo
- 8,7% Requerimientos cambiantes
- 8,1% Falta de planificación
- 7,5% No se necesita
- 6,2% Falta de gestión de tecnologías de la información
- 4,3% Desconocimiento tecnológico
- 9,9% Otros

De acuerdo a este estudio, tanto la completitud de los requerimientos como la falta de involucramiento de los usuarios, constituyen el principal motivo para el fracaso de los proyectos de desarrollo de software, lo cual se evidencia en los múltiples proyectos que se incumplen en la fecha de entrega y en el desfase en los costos estimados. En la investigación abordada en el presente trabajo, se espera contrastar el resultado de los proyectos en estudio con el proceso de gestión de requerimientos y el involucramiento de los usuarios para determinar prácticas inadecuadas que

originen este comportamiento y, a partir de esto, definir posibles oportunidades de mejora.

2.2 Metodologías ágiles

Son métodos para la gestión de proyectos, enfocados en orientar equipos de trabajo en un desarrollo conjunto de manera efectiva para alcanzar los objetivos del negocio. Buscan resolver los problemas que empezaron a detectarse en la década de los 90's, cuando las expectativas y necesidades de los usuarios finales de los proyectos de desarrollo de software, se incumplían y las empresas se vieron inmersas en una serie de proyectos que no lograban sus objetivos; eran demasiado costosos y a la hora de salir al mercado ya no eran competitivos.

Las metodologías ágiles ofrecen, entonces, una alternativa a los métodos tradicionales que están caracterizados por el énfasis en la documentación, la definición de un alcance fijo, el establecimiento de un cronograma de ejecución del proyecto y un costo específico. Esta metodología está orientada en el logro de objetivos del proyecto, sin dar relevancia a las tareas que no son imprescindibles y reconoce como factor fundamental de éxito la construcción de un buen equipo de trabajo, por encima de los procesos y herramientas que sean requeridos en el proyecto.

Con esta metodología se pretende hacer un uso eficaz del conocimiento de las personas (QRP International, 2014) y permitir adaptar la forma de trabajo del equipo a las condiciones del proyecto. De esta manera, se logra flexibilidad y prontitud en la respuesta ante circunstancias específicas del entorno o ante cambios en los requerimientos iniciales del proyecto, que pueden implicar cambios sencillos en la necesidad inicial del cliente, surgimiento de nuevos productos y evolución de la competencia o de las tendencias tecnológicas.

En la actualidad las empresas necesitan implementar metodologías que las lleven a realizar entrega de productos y servicios de calidad, cumpliendo con el tiempo y costo pactado y, por medio de las metodologías ágiles, pueden encontrar la adaptabilidad requerida a las exigencias del mercado.

Las empresas que llegan a implementar metodologías ágiles logran mejorar la gestión de sus proyectos, lo cual les significa una disminución de costos y un aumento de productividad al concebir lo incierto e impredecible como oportunidades de cambio que son bienvenidas en el proyecto, con el fin de cumplir con las expectativas del cliente. Sin embargo, este escenario no es el común de las empresas a nivel mundial, aún existe una gran brecha en el número de proyectos de software que terminan de manera exitosa y que logran el objetivo inicial planteado de manera eficaz e incrementa su propuesta de valor para el cliente (Alliance, 2016).

Standish Group (2015) analizó 50.000 proyectos de la industria de desarrollo de software alrededor del mundo, y evidenció que existe una diferencia porcentual considerable entre la probabilidad de que un proyecto con metodologías ágiles finalice cumpliendo con las expectativas esperadas, respecto a un proyecto que se desarrolla de manera tradicional o en cascada.

Ilustración 1. Encuesta CHAOS Report 2015

	2011	2012	2013	2014	2015
EXITOSO	29%	27%	31%	28%	29%
CON DESAFIOS	49%	56%	50%	55%	52%
FALLIDO	22%	17%	19%	17%	19%

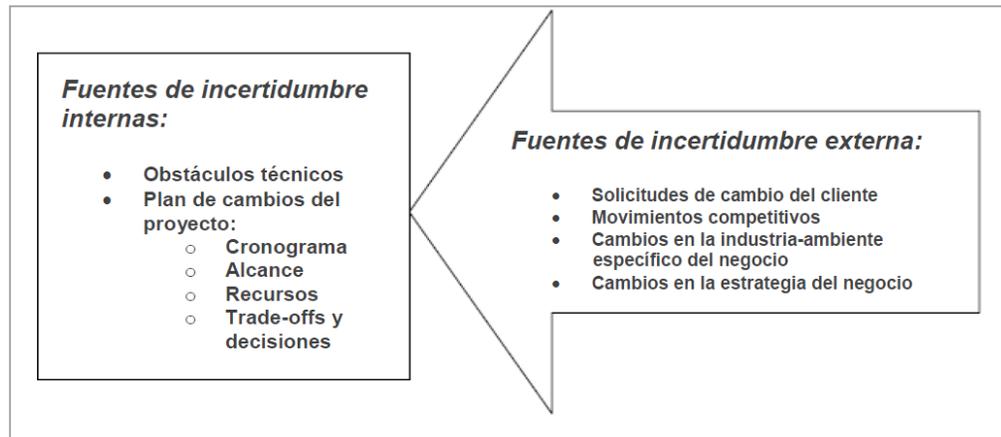
TAMAÑO	METODOLOGIA	EXITOSO	CON DESAFIOS	FALLIDO
Proyectos de todos los tamaños	Agil	39%	52%	9%
	Cascada	11%	60%	29%
Proyectos grandes	Agil	18%	59%	23%
	Cascada	3%	55%	42%
Proyectos medianos	Agil	27%	62%	11%
	Cascada	7%	68%	25%
Proyectos pequeños	Agil	58%	38%	4%
	Cascada	44%	45%	11%

Fuente: Elaboración propia basada en Lynch, Jennifer. Standish Group 2015 Chaos Report - Q&A with Jennifer, 2015.

La comparación en el tiempo realizada por Standish Group (2015) no muestra evolución; los resultados indican que las empresas tienen problemas para alcanzar el éxito en los proyectos y es necesario buscar estrategias que permitan realizar cambios sustanciales en la definición y gestión de los proyectos, con miras a la consecución de los resultados esperados por la organización.

Se hace imperativo entonces considerar que los proyectos no se mueven en un ambiente estático y controlado. Según Chin (2004), las fuentes de incertidumbre en un proyecto pueden ser de dos tipos: internas, que son las relacionadas con el alcance, costo y tiempo del proyecto y, externas, que están relacionadas con la industria de la organización, los tipos de clientes y la estrategia del negocio.

Ilustración 2. Fuentes de incertidumbre en proyectos



Fuente: Elaboración propia tomada de Chin, 2004.

Las metodologías ágiles buscan dar una respuesta a esta necesidad, haciendo uso de los principios definidos en 2001, cuando surgió el manifiesto ágil en los Estados Unidos, específicamente en el estado de Utah. En este año un grupo de diecisiete personas se reunieron con la necesidad de buscar alternativas diferentes a los procesos pesados y robustos para el desarrollo de software. El grupo estaba conformado por practicantes de metodologías como Extreme Programming, Adaptive Software Development, Crystal, Pragmatic Programming, entre otras.

Aunque fue difícil llegar a un consenso por los diferentes puntos de vista, allí se gestó un hito histórico declarado el “Manifiesto ágil para el desarrollo de software”, que tiene como principios fundamentales:

1. “Individuos e interacciones sobre procesos y herramientas” (Beck et al., 2001), lo cual garantiza mayor productividad para el equipo de trabajo.

Las metodologías ágiles estiman el equipo como principal factor de éxito y consideran que contar con personas calificadas, con la capacidad técnica apropiada, con disposición para adaptarse al entorno, con capacidad de trabajo en equipo, con destreza para interactuar con los usuarios, proporciona mayor

probabilidad de éxito que contar con herramientas y procesos rigurosos en el proyecto.

2. “Software funcionando sobre documentación extensiva” (Beck et al., 2001), con lo cual se prioriza el desarrollo de software por encima de la documentación extensa que no agregue valor al usuario.

Si bien la producción de documentos es importante, al igual que identificar el tiempo y costo de mantener la documentación completa y actualizada, la relevancia de dicha documentación está sujeta a la producción de documentos que sean necesarios, los cuales deben contener información fundamental, dando prioridad al contenido y bajo la premisa de que sean cortos y fácilmente digeribles.

3. “Colaboración con el cliente sobre negociación contractual” (Beck et al., 2001). Los responsables de negocio y los desarrolladores trabajan juntos de forma cotidiana durante todo el proyecto. Esto marca una gran diferencia con las metodologías tradicionales en las que típicamente el usuario asume el rol de indicar qué debe hacer el software y luego espera que los resultados cumplan con sus expectativas, en el plazo pactado.

En este escenario frecuentemente las dos partes, cliente y equipo de trabajo, asumen roles distantes con tal prevención que se requiere dedicar un tiempo considerable a redactar, depurar y firmar un contrato que contenga la mayor información posible de alcance, tiempo y costo, para de este modo garantizar que a futuro se van a cumplir las restricciones de ambas partes. Bajo una metodología ágil, el cliente se incluye en el equipo de trabajo de manera directa y con un alto nivel de compromiso, se busca que el cliente participe desde el inicio hasta el fin del proyecto, que corrija o haga recomendaciones en cualquier

momento del proyecto y que exista un beneficio común tanto para la empresa como para el cliente.

4. “Respuesta ante el cambio sobre seguir un plan” (Beck et al., 2001). En ágil, es aceptable que los requisitos cambien, incluso en etapas tardías del desarrollo. Se aprovecha el cambio para proporcionar ventaja competitiva al cliente.

Por la naturaleza cambiante de los productos, la tecnología, la continua evolución de la sociedad y del mercado, un proyecto de software no puede ser estático y debe enfrentarse a cambios durante su ejecución. Las metodologías ágiles tienen en su naturaleza un esquema de trabajo que facilita la respuesta a dichos cambios, lo que permite que el equipo de trabajo se adapte y reciba los cambios con una posición de apertura.

5. Se entrega software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.

El software funcionando es la medida principal de progreso y la prioridad del equipo es satisfacer al cliente mediante la entrega temprana y continua de software de valor. En metodologías ágiles se trabaja realizando entregas parciales, pero funcionales del producto en el menor intervalo de tiempo posible.

6. Los proyectos se desarrollan alrededor de individuos motivados. Al equipo de trabajo hay que darle el entorno y el apoyo que necesita, y confiarle la ejecución del trabajo.
7. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y, entre sus miembros, es la conversación cara a cara.

8. Ágil promueve el desarrollo sostenible. Los promotores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida.
9. Existe atención continua a la excelencia técnica, la simplicidad, el arte de realizar lo esencial y la capacidad de adaptación a las necesidades del cliente.
10. Las mejores arquitecturas de software, requisitos y diseños emergen de equipos auto-organizados.
11. En intervalos regulares de tiempo el equipo debe reflexionar sobre cómo ser más efectivo para ajustar y perfeccionar su comportamiento. Es decir, se hace imprescindible realizar una revisión y retrospectiva constante de lo alcanzado, para definir acciones de mejora que puedan ser introducidas rápidamente en el día a día del equipo.

Las metodologías ágiles e incrementales están en uso desde hace varias décadas, dado que la comunidad de desarrollo de software ha buscado resolver las falencias de las metodologías tradicionales o en cascada, explorando maneras de adaptarse y evolucionar hacia una práctica eficaz, maniobrable, escalable y, sobre todo, centrada en el involucramiento activo de los usuarios de software y en el conocimiento tácito dentro del equipo por encima de la documentación.

Las primeras metodologías con orientación ágil que surgieron como V - Model, Spiral – Model, Rational Unified Process (RUP), estaban altamente orientadas a la documentación y a la planeación estructurada, por encima de la adaptabilidad y de la orientación a las personas, que requería el ambiente tecnológico turbulento y cambiante en el que es imposible anticipar un conjunto de requisitos desde el principio del ciclo de vida del proyecto.

Sin embargo, identificar las falencias de los procesos, encontrar la solución e implementar los cambios con éxito, no ha sido una labor sencilla. De acuerdo con el último estudio publicado por Gartner (2016), el desarrollo de proyectos en las organizaciones no está evolucionando hacia mecanismos que le permitan obtener los resultados planeados y cumplir las expectativas del cliente.

Desde hace varios años se ha identificado que el éxito de un mercado global está en su capacidad de producir pequeños lotes de productos, que puedan satisfacer las crecientes demandas de los mercados globalizados y emergentes. Sin embargo, la industria de software continúa conteniendo enormes costos y excesos de programación que afectan de manera directa el rendimiento de los proyectos. Algunas cifras de referencia enunciadas por Hass Kathleen (2007) en *The Blending of Traditional and Agile Project Management*, son:

- De 80 a 145 mil millones por año se invierte en proyectos fallidos y cancelados (The Standish Group, 2014).
- Entre 25% a 40% del gasto en proyectos se pierde como resultado de rehacer el trabajo (The Standish Group, 2014).
- 50% de los errores son detectados en producción por parte del cliente (Gartner, 2015).
- 40% de los problemas es detectado por los usuarios finales al momento de operar la solución (Gartner, 2015).
- Las aplicaciones mal definidas han dado lugar a una falta de comunicación entre las empresas y la industria de tecnología. Esto contribuye a una tasa de fracaso de proyectos del 66%, lo que cuesta negocios de por lo menos \$30 mil millones cada año (The Standish Group, 2014).
- Entre el 60% a 80% de las falencias de los proyectos son atribuibles directamente a la mala recolección de requisitos, análisis y gestión (Meta Group, 2015).

Se podría inducir entonces que a pesar de ser una metodología definida y explorada desde finales de los 80's, aún es un método poco difundido y, sobre todo, aplicado en el desarrollo de proyectos. Si bien la coyuntura de la globalización y el auge de las economías de escala a nivel mundial demanda procedimientos flexibles y desarrollo de nuevos productos en tiempos cortos, las empresas tienen en sus manos el reto de tomar lo enunciado por la metodología ágil y adaptarlo a su entorno competitivo.

Este enfoque innovador para la gestión de proyectos pretende contribuir a que las personas puedan trabajar juntas de manera efectiva y en pro de desarrollar el conocimiento como eje fundamental del crecimiento organizacional.

La metodología ágil más que un proceso forzado con un conjunto de pasos a seguir, es un marco adaptativo que flexibiliza las normas de las metodologías tradicionales actuando como una guía que combina el uso eficaz de los conocimientos de las personas, junto con las herramientas (QRP International, 2014), y considera la comprensión de los elementos del negocio: el entorno, el desarrollo de la gestión de proyectos y la promoción de las necesidades de las personas, en un ambiente colaborativo (Chin, 2006).

2.2.1 Estado de la implementación de metodologías ágiles en proyectos de software

Partiendo del estudio publicado en la revista *Computer Standards & Interfaces* en el año 2017, sobre requerimientos en entornos de metodologías ágiles, en el que se analizaron 42.808 artículos y 965 estudios, se podrían considerar los siguientes aspectos relevantes en el manejo actual de los proyectos de software bajo metodologías ágiles:

1. La comunicación y la colaboración continuas entre el cliente y el equipo de trabajo son relevantes para involucrar las partes interesadas en el proceso y lograr la efectividad y eficiencia propuesta por la metodología.
2. Para el éxito de la metodología es indispensable la participación del cliente desde el inicio del proyecto, con el fin de obtener de parte del equipo la información de sus necesidades y mantener este relacionamiento a medida que evoluciona el proyecto y surgen ajustes que modifiquen lo inicialmente definido.

La participación de las partes interesadas y los usuarios durante el proyecto es esencial para establecer un entorno de colaboración con ciclos constantes de retroalimentación, con el objetivo de ofrecer productos competitivos con una experiencia de usuario adecuada (Schon, Thomaschewski & Escalona, 2017).

3. Se detectó que uno de los principales problemas en el uso de ágil es la dificultad de compartir conocimiento entre los miembros del equipo, lo cual dificulta el diseño y desarrollo participativo y muestra que existen dificultades inherentes a la gestión del conocimiento en las organizaciones.

Para una empresa de desarrollo de software, el principal activo es el conocimiento y debe por lo tanto tratar con problemas en dicho ámbito. Sin embargo, la tarea no es trivial ni sencilla, no se limita a la codificación, organización o almacenamiento de datos que deban ser consultados por las personas de la organización en sus actividades diarias; por el contrario, consiste en una tarea compleja que parte de la creación de conocimiento, el proceso dinámico y evolutivo para difundirlo en la empresa y la incorporación de lo aprendido en las nuevas tecnologías.

Los procesos de innovación requeridos en los proyectos de desarrollo de software requieren una definición organizacional de creación de conocimiento

que haga referencia a las entidades de creación de conocimiento en diferentes niveles: individual, grupal, organizacional e interorganizacional (Nonaka & Takeuchi, 1995b) y el rol de la organización en este sentido es el de proveer las condiciones apropiadas para facilitar las actividades individuales y grupales de creación y acumulación de conocimiento, para desarrollar la ventaja competitiva que reside en lo que la compañía sabe, cómo emplea lo que sabe y cómo gestiona su capacidad de aprender cosas nuevas.

Es importante entonces entender el conocimiento en los proyectos de desarrollo de software como algo que debe emerger a partir de la participación de los miembros del equipo; el conocimiento en la organización no puede surgir sin la participación de sus individuos y es preciso entender la naturaleza de dicho conocimiento para facilitar las actividades que pueden desarrollarlo y mantenerlo.

4. El análisis de los estudios revisados muestra que la implementación de ágil se enfrenta a algunos problemas con respecto a la documentación de requisitos, este es un desafío especial con respecto a la configuración del proyecto, dado que existe una variedad de artefactos diferentes que las empresas deben usar de acuerdo a la necesidad de su entorno. Los más relevantes son las historias de usuarios, prototipos, casos de uso, escenarios y tarjetas de historias que pueden ser claves en el relacionamiento entre el cliente y el equipo de trabajo.
5. La arquitectura empresarial de una organización que lleva a cabo proyectos de desarrollo de software es fundamental para la implementación de una metodología ágil. Si se considera la arquitectura empresarial como la responsable de la alineación entre el negocio y los procesos de tecnologías de información (Consortio de Sistemas y Software), se podría apreciar la relevancia de tratar la transformación de la empresa no desde un cambio ligero en los procesos, sino en la profundidad de su cultura y sus políticas o modelos.

En la actualidad los líderes empresariales reconocen la arquitectura empresarial como un tema importante para llevar a cabo la alineación de las estructuras y los procesos empresariales con las estrategias comerciales a largo plazo (Kaddoumi & Watfa, 2016). En uno de sus artículos en el sitio web de la asociación de arquitectos empresariales, Allen Brown (presidente y CEO de Open Group, 2016) estimó la capacidad ágil de adaptarse a los requisitos comerciales en constante cambio al afirmar: “el desarrollo de software ágil se ha convertido en una de las formas en que los desarrolladores de tecnología se adaptan a los requisitos del cambio constante” (Kaddoumi & Watfa, 2016, p. 55). También sugirió que “la adaptación de los doce principios del desarrollo ágil a la disciplina de arquitectura empresarial sería un punto interesante para comenzar” (Kaddoumi & Watfa, 2016, p. 55).

Las empresas solo podrán subsistir ante la demanda de un mercado global que espera respuestas veloces, si tanto la transformación cultural de la empresa, como la adopción de prácticas ágiles y la arquitectura empresarial, caminan de la mano hacia un mismo horizonte. Se hace imperativo unir la evolución que se ha logrado en el “hacer ágil”, ámbito logrado desde el cambio de los procesos y el “ser ágil” (Sahota, 2012), ámbito que solo se aborda cuando la organización llega a una transformación de su cultura y de la arquitectura empresarial que ha definido. De este modo, se podría hablar de una metodología ágil que trasciende de la adopción de procesos, a la transformación ágil que trabaja en el trasfondo de la cultura organizacional (Sahota, 2012).

6. En la transformación hacia una cultura basada en metodologías ágiles es indispensable considerar la administración de los cambios, como la manera que tiene la organización de afrontar la evolución de su entorno. Algunos cambios pueden estar originados en la demanda de los clientes, los avances técnicos, las

nuevas legislaciones y, en general, todos los aspectos que puedan generar un cambio en los parámetros establecidos (Kaddoumi & Watfa, 2016).

Se hace indispensable generar adaptabilidad a los cambios del entorno empresarial para lograr sobresalir ante las empresas que no pueden hacerlo y, para esto, es fundamental contar con una cultura que promueva las metodologías ágiles, no solo tratando temas superficiales de procesos y herramientas, sino tratando el fondo de los valores y principios que determinan la esencia misma de la organización.

7. Los principios ágiles se complementan para adaptarse continuamente a un entorno volátil con criterios cambiantes en la consecución de objetivos. Es fundamental adoptar los métodos de entregas de software iterativas que mejoren la lentitud y retraso en el retorno de las inversiones y que mitiguen las grandes dificultades en la construcción de valor de los proyectos (Kaddoumi & Watfa, 2016). De este modo, se alcanza mayor velocidad y eficiencia realizando entregas parciales y funcionales del producto.

2.2.2 Riesgos con el uso de metodologías ágiles

Las metodologías ágiles, a menudo son tratadas como agentes mitigadores de los riesgos que puede presentar un proyecto y, en este mismo sentido, son vistas como la respuesta que resuelve el factor que amenaza el éxito de los proyectos. Sin embargo, se debe reconocer que el uso de metodologías ágiles per se, no debe ser visto como la fórmula mágica de éxito de los proyectos y que la metodología utilizada de manera incorrecta puede acarrear situaciones de riesgo para la empresa.

De acuerdo al estándar ISO 31000, se podría definir riesgo como el efecto de la incertidumbre sobre un objetivo determinado, entendiendo “efecto” como la

desviación de algo que se espera y por “incertidumbre” el estado de deficiencia de información ligada al conocimiento de un evento. Es decir, como riesgo se podría definir los eventos positivos o negativos que pueden afectar un proyecto para mejorar o para degradar sus resultados.

Estos riesgos incluyen tanto amenazas para el proyecto como oportunidades que podrían ser aprovechadas en beneficio de los proyectos, bajo metodologías ágiles. Los siguientes riesgos se identificaron en la industria de telecomunicaciones con base en la investigación realizada sobre proyectos llevados a cabo en diferentes países, distintos clientes y equipos de proyecto, que procedían de diversas culturas (Walczak & Kuchta, 2013).

- Las metodologías ágiles están diseñadas de tal manera que permiten cambios muy baratos en el alcance de un proyecto y esto es especialmente útil en el caso de proyectos con requisitos en continua evolución. Esto implica que los cambios en los requisitos y en el alcance del proyecto se consideran una oportunidad de llegar a un producto de valor que logre de manera temprana la satisfacción del cliente.

Desde esta perspectiva, la gestión de cambios en los requisitos está bajo control del equipo en común acuerdo con el cliente, lo que permite a su vez, generar una relación de confianza y elimina las inhibiciones del equipo de desarrollo para llegar al máximo de su capacidad creativa.

- En el auge de las metodologías ágiles se presenta gran interés de los actores de la comunidad de software por incurrir en el uso de dichas metodologías. Sin embargo, la celeridad exigida por el mercado y la falta de conocimiento profundo de la metodología, induce a que algunas personas que dicen ser ágiles y entienden las prácticas de manera superficial, no razonen en la mentalidad ágil. “Cada vez es más común encontrar profesionales ágiles que entienden las prácticas, pero que no entienden los valores y principios. El argumento aquí es

que ágil está destinado al fracaso, ya que su mensaje y significado se tergiversan” (Sahota, 2012, p. 20).

- La oportunidad de que todo el equipo trabaje colectivamente, apoya de manera significativa el logro de los objetivos del proyecto.
Las metodologías ágiles aumentan la probabilidad de tener el equipo del proyecto trabajando colectivamente hacia el objetivo del proyecto, a través de prácticas como: reuniones con todos los miembros del equipo, distribución efectiva de la información del progreso del proyecto, disminución de las barreras de comunicación y eliminación de las metas individuales.
- La auto organización del equipo podría ser vista como una ventaja del uso de metodologías ágiles, sin embargo, el equipo podría no ser capaz de auto organizarse o podría verse abrumado por la cantidad de información que debe procesar y el número de decisiones a tomar (Walczak & Kuchta, 2013).
Este riesgo puede presentarse con mayor probabilidad en organizaciones donde el equipo de proyecto solo se establece para un proyecto y se disuelve una vez concluya.
- Bajo la metodología ágil, las decisiones grupales son mejores que las decisiones de los individuos.
Sin embargo, existe el riesgo de que las decisiones sean incorrectas cuando el equipo no tiene una visión completa del problema o cuando los miembros del equipo no toman una decisión basada en el mérito sino en otros criterios, como beneficios particulares o determinación de miembros individuales.
- La organización puede proveer herramientas de desarrollo y pruebas que funcionan bien para métodos tradicionales, pero no son compatibles con la flexibilidad requerida en metodologías ágiles, como prácticas de desarrollo impulsado por prueba o integración continua.

- En metodologías ágiles existe un control mejorado sobre el proyecto, dada la visibilidad del estado en cualquier momento del tiempo, de modo que cualquier discrepancia con el plan puede ser inmediatamente detectado. Esto implica que el cliente esté informado de manera temprana de los riesgos o de las limitaciones del proyecto, dejando tiempo suficiente para reaccionar.
- Desde la perspectiva del cliente, las metodologías ágiles permiten incrementos de producto de manera temprana, con lo cual se pueden lograr beneficios tempranos.
La aceptación final del producto al concluir el proyecto dura un tiempo corto, dado que la gran mayoría de mejoras han sido abordadas anteriormente y los desacuerdos sobre los requisitos se han aclarado en las primeras etapas del proyecto.
- El éxito de la adopción de metodologías ágiles requiere del involucramiento activo del cliente; sin embargo, existen proyectos donde el cliente es incapaz de proporcionar retroalimentación valiosa y completar al equipo del proyecto, por ejemplo, respuestas a las preguntas acerca de los requisitos, prioridades o la retroalimentación sobre los incrementos de producto generados.
- Varios autores han detectado el riesgo que existe en la gestión y monitoreo de costos en metodologías ágiles, indican que existe poca evidencia acerca de la medición de costos y que esta se basa comúnmente en juicios de expertos y no en procesos repetibles y mejorables.

En la medida que no exista un proceso definido y medible, no va a ser posible lograr una optimización ni reducción de los costos del proyecto. Sin embargo, la industria viene avanzando en estudios para mejorar los métodos de medición usados para la estimación, de manera tal que sin atentar contra los principios de

medición de avance, se puedan generar modelos matemáticos con medidas predictivas que puedan apoyar el proceso de estimación de tiempo, esfuerzos y costos (Hugo & Cuauhtémoc, 2014).

2.3 Metodologías tradicionales

En el marco de metodologías tradicionales se identifica que el eje fundamental de acción se encuentra en la dirección de proyectos, que es la encargada de la aplicación de conocimientos, habilidades, herramientas y técnicas en las actividades del proyecto, para cumplir con los requisitos del mismo. Se logra mediante la aplicación e integración adecuada de los 47 procesos de la dirección de proyectos, agrupados de manera lógica, y categorizados en cinco grupos de procesos:

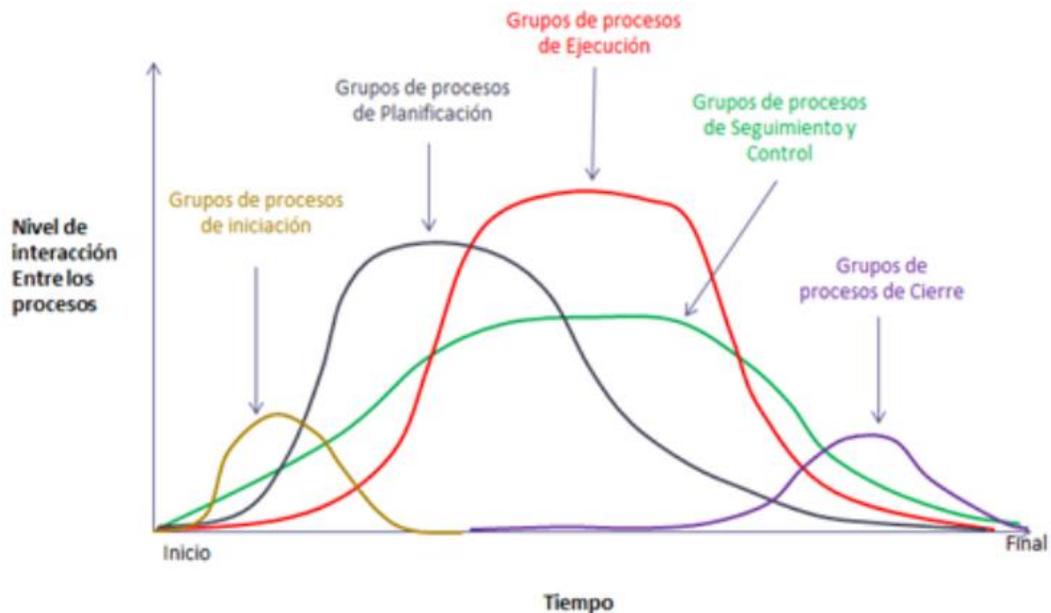
- Inicio: incluye la definición del alcance inicial, la descripción de los objetivos del proyecto, la identificación de los interesados internos y externos que van a participar y la selección del director de proyecto. El propósito principal es alinear la expectativa de los interesados con el propósito del proyecto.
- Planificación: consiste en establecer el alcance total del esfuerzo a realizar, definir y refinar los objetivos y desarrollar la línea de acción requerida para alcanzar dichos objetivos. El propósito principal es desarrollar el plan para la dirección del proyecto y elaborar los documentos del proyecto que se utilizarán para llevarlo a cabo.
- Ejecución: está compuesto por los procesos que se deben realizar para completar el trabajo definido en el plan para la dirección del proyecto, a fin de cumplir con las especificaciones del mismo. Implica coordinar personas y recursos, gestionar las expectativas de los interesados, así como integrar y realizar las actividades del proyecto conforme al plan para la dirección del proyecto.
- Monitoreo y control: está compuesto por aquellos procesos requeridos para rastrear, analizar y dirigir el progreso y el desempeño del proyecto, para identificar áreas en las

que el plan requiera cambios e iniciar los cambios correspondientes. El desempeño del proyecto se mide y se determinan las acciones para reorientar el proyecto hacia el resultado esperado.

- Cierre: implica finalizar todas las actividades de todos los grupos de procesos de la dirección de proyectos, a fin de completar formalmente el proyecto, una fase del mismo u otras obligaciones contractuales. En esta fase se establece formalmente que el proyecto o fase del mismo ha finalizado. (Project Management Institute, citado en Guide, 2013, p. 53)

Los grupos de procesos en el ciclo de vida del proyecto interactúan con base en el ciclo planificar – hacer – verificar – actuar y también pueden entrecruzarse entre las fases del proyecto, como se ilustra en la siguiente gráfica:

Ilustración 3. Procesos de un proyecto



Fuente: Elaboración propia basada en el PMBOK (Guide, 2013).

Dirigir un proyecto, bajo este tipo de metodologías, implica identificar requisitos, gestionar necesidades, encargarse de las dudas y expectativas de los interesados

del proyecto que provienen de la organización, el cliente y el mismo equipo de trabajo, planificar el proyecto de manera detallada, ejecutar el proyecto, establecer canales de comunicación efectivos entre los interesados, gestionar el cumplimiento del alcance y nivelar las restricciones que pueden originarse respecto al alcance, calidad, cronograma, presupuesto, recursos y riesgos (Guide, 2013).

Un programa contiene un grupo de proyectos relacionados que, al ser agrupados, pueden ser coordinados bajo la misma dirección y de este modo logran aprovechar interdependencias entre dichos proyectos para alcanzar una disminución en riesgos, generar economías de escala y lograr una dirección mejorada (Mulcahy, 2013). Al unir proyectos relacionados, la organización puede alcanzar objetivos que respalden la consecución de metas organizacionales. Este tipo de alineación en el marco para la dirección de proyectos tradicionales, permite alcanzar una ventaja competitiva, dado que estimula el aprovechamiento de capacidades transversales a la organización, que serán traducidas en eficiencias y en mayor productividad.

Un portafolio incluye un grupo de programas, proyectos individuales y otros trabajos operativos relacionados, que se priorizan y se implementan para lograr un objetivo estratégico de negocio específico. Al igual que ocurre con la dirección de programas, combinar los proyectos, los programas y las operaciones en uno o más portafolios ayuda a optimizar el uso de recursos, mejora los beneficios de la organización y reduce el riesgo (Mulcahy, 2013). Es decir, la gestión de portafolios permite alcanzar de manera apropiada las metas estratégicas planteadas a nivel corporativo.

La relación entre la planeación de los proyectos y la estrategia corporativa de la organización garantiza que los proyectos definidos sean los apalancadores de los objetivos definidos en el plan estratégico de la empresa y, para garantizar dicha sinergia, se crea una oficina de dirección de proyectos, definida como la estructura de gestión que estandariza los procesos de gobierno relacionados con el proyecto

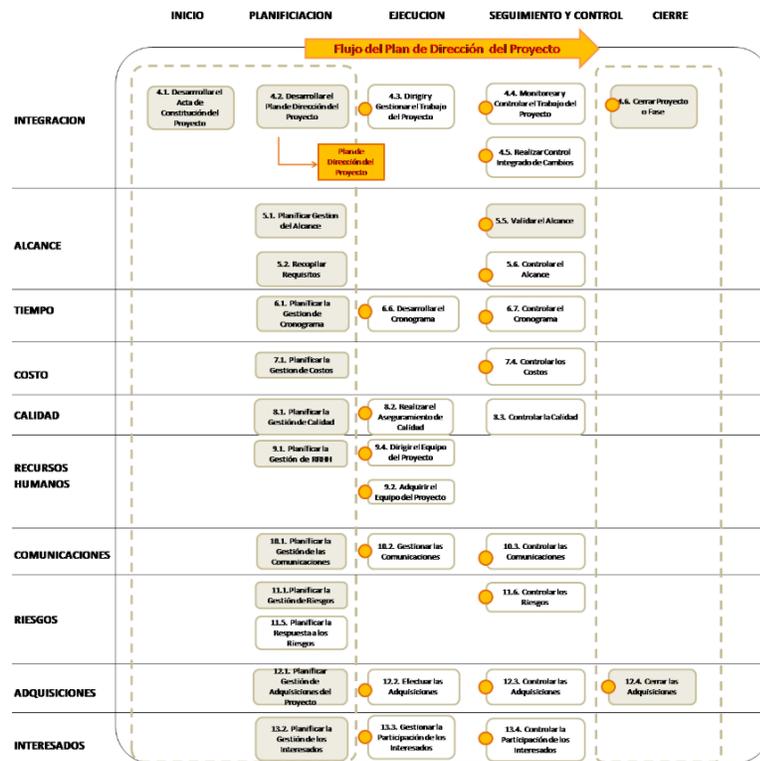
y hace más fácil compartir recursos, metodologías, herramientas y técnicas. Las responsabilidades de una PMO (oficina de dirección de proyectos) pueden abarcar desde el suministro de funciones de soporte para la dirección de proyectos hasta la responsabilidad de la propia dirección de uno o más proyectos.

La PMO integra los datos y la información de los proyectos estratégicos corporativos y evalúa hasta qué punto se cumplen los objetivos estratégicos de alto nivel. La PMO constituye el vínculo natural entre los portafolios, programas y proyectos de la organización y los sistemas de medida corporativos (p.ej., el cuadro de mando integral) (Guide, 2013).

En la gestión de proyectos, programas y portafolios, bajo la metodología tradicional, es importante tener presente que todo debe estar alineado con los objetivos estratégicos de la organización, esto garantiza que la competitividad esté enfocada en alcanzar valor agregado en todos los eslabones de los proyectos y que se trabaje de manera conjunta para evitar una mala planeación o ejecución de un proyecto.

El Project Management Institute (Guide, 2013) considera la relación de portafolios, programas y proyectos, alineados con la estrategia organizacional, por lo cual define diez áreas de conocimiento y cinco grupos de procesos que son transversales a esas áreas y que están integrados y relacionados, como lo muestra la siguiente ilustración:

Ilustración 4. Áreas de conocimiento y grupos de procesos en dirección de proyectos

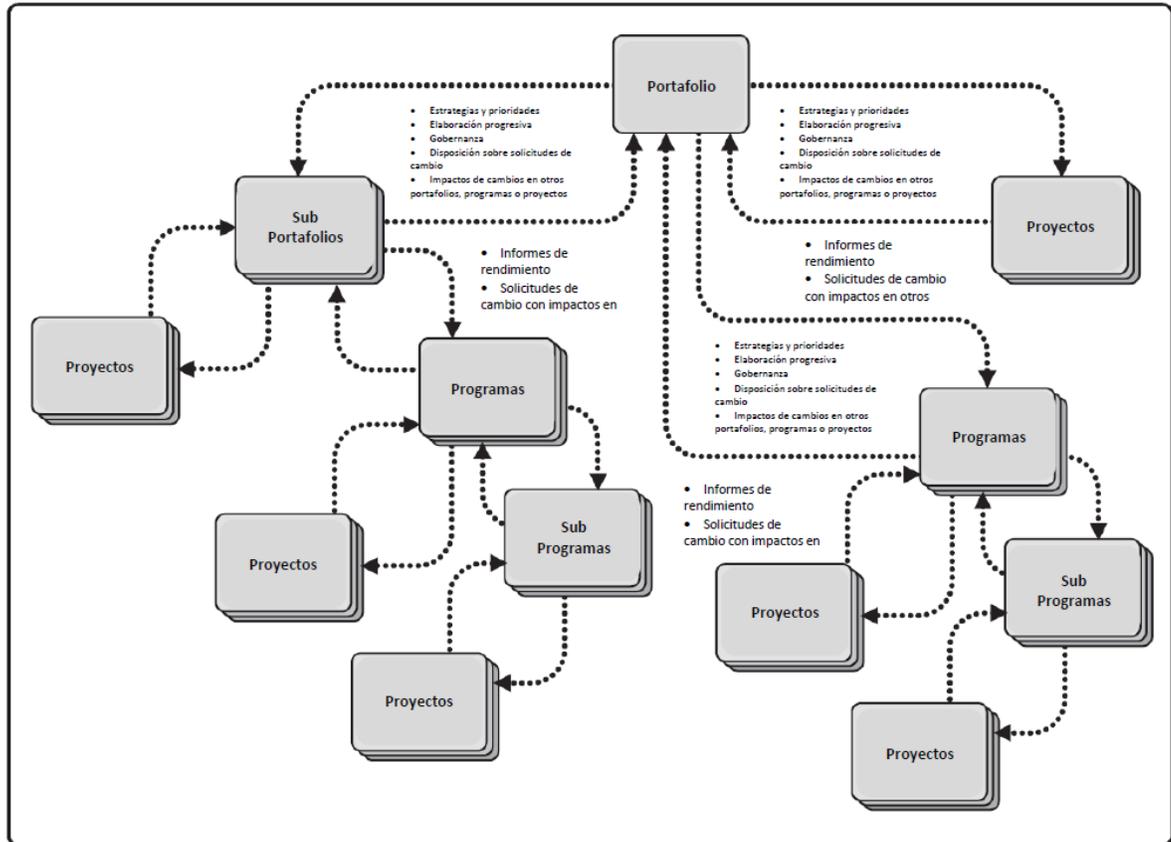


Fuente: Bs Grupo Conocimiento para crecer – basada en el PMBOK (Guide, 2013).

Adicional a esto, el PMI (Guide, 2013) desarrolló un marco de referencia para la ejecución de proyectos, programas y portafolios, denominado Gestión de Proyectos Organizacional (*OPM: Organizational Project Management*), el cual se define como un marco que integra buenas prácticas, permite alcanzar los objetivos definidos y lograr mejores resultados en pro de obtener una ventaja competitiva y sostenible para la organización (Project Management Institute, citado en Guide, 2013).

El siguiente cuadro describe la relación e interacción entre proyectos, programas y portafolios:

Ilustración 5. Interacción entre proyectos, programas y portafolios



Fuente: PMBOK (Guide, 2013).

Como principal ejecutor de la metodología tradicional, se encuentra el rol de director de proyectos, que es el responsable de desarrollar el arte de gestionar recursos y personas, a través de un proceso sistemático que involucra procesos técnicos, roles, responsabilidades y niveles de autoridad en el equipo de proyecto.

Una metodología de gestión de proyectos tradicional busca facilitar la gestión de ideas e iniciativas al interior de la organización, dado que brinda elementos estratégicos que dan orden de principio a fin al proyecto, tanto en la planeación como en la ejecución del mismo. Este esquema es fundamental para el director de proyectos y para los demás interesados, dado que marca la ruta a seguir en cada fase del proyecto, permite medir de manera estricta el avance respecto al alcance

planteado, facilita la toma de acciones para enrutar el cumplimiento del proyecto y ayuda a identificar y minimizar riesgos desde la fase de planeación, hasta la ejecución y cierre del proyecto.

La implementación de la metodología tradicional para la gestión de proyectos permite que la empresa materialice su estrategia corporativa, a través de procesos que facilitan el logro del alcance propuesto, buscando la optimización de recursos. De acuerdo con lo estipulado por este tipo de metodologías, los proyectos deben pasar por un ciclo de vida que se divide en cuatro fases, que se identifican claramente en la administración de proyectos propuesta por Guido y Clements (2013):

- Identificación de la necesidad para determinar el problema u oportunidad que va a ser el eje central del proyecto.
- Desarrollar un plan de solución para encontrar la resolución a la necesidad planteada y presentar la propuesta al cliente, para que este determine si cumple con sus expectativas y aprueba el alcance a implementar.
- Ejecutar la solución propuesta cumpliendo con el objetivo planteado en la planeación inicial.
- Cerrar el proyecto, documentando mejoras a todo nivel de la organización, que serán considerados en futuros proyectos.

En la mayoría de proyectos bajo esta metodología, cada una de las etapas se ejecuta de manera secuencial, para garantizar la evolución del proyecto de manera efectiva con el menor riesgo posible.

Aunque la metodología tradicional proporciona beneficios en la organización, adaptarla a las políticas y al entorno cultural de la misma, requiere realizar un análisis de las variables que rodean la organización y determinar en su cadena de valor el aporte que va a realizar, cuánto costará, que beneficios trae, cuáles personas están preparadas para trabajar en este esquema de trabajo, preparación de la jerarquía organizacional y los niveles de autoridad establecidos para dicha metodología (Charvat J, 2003).

Basados en el PMBOK (Guide, 2013), se podrían concluir las siguientes actividades a realizar en las áreas de procesos que enmarcan la ejecución de un proyecto, bajo metodologías tradicionales:

1. Gestión de la integración: consiste en reunir los procesos del proyecto, con el fin de unir todas las piezas en un todo cohesivo y buscar la alineación con los objetivos del proyecto. Implica que los artefactos de los procesos sean consistentes con lo definido en las demás etapas del proyecto, por ejemplo, que los artefactos de alcance sean coherentes con lo estipulado a nivel de tiempo, costo, calidad, recursos humanos y comunicaciones.
2. Gestión del alcance: es el proceso que implica definir el trabajo que se debe desarrollar y después garantizar que todo ese trabajo y solo ese trabajo sea realizado. En metodologías tradicionales este alcance debe estar definido claramente antes de que inicie el trabajo del proyecto y mientras el proyecto se completa, se deben realizar revisiones para garantizar que el equipo esté implementando todo el trabajo y solo el trabajo incluido en el plan para la dirección del proyecto
3. Gestión del tiempo: la principal responsabilidad en este proceso es la del director de proyectos, quien es el encargado de crear un cronograma realista para cumplir con la fecha de finalización requerida por el proyecto y crear opciones

para hacer que esto suceda. El cronograma del proyecto debe estar definido antes de iniciar la ejecución del proyecto y no debería tener cambios que afecten lo planeado al inicio del proyecto. Una buena gestión de tiempo requiere descomponer los paquetes de trabajo en componentes más pequeños o actividades que puedan ser gestionadas por el director de proyectos.

4. Gestión de los costos: implica planificar, gestionar y controlar los costos del proyecto, creando un presupuesto apropiado para las necesidades del proyecto. El responsable de esta actividad es el director de proyecto y, al igual que en la gestión del tiempo, se espera que este plan no tenga desviaciones durante la fase de ejecución del proyecto.
5. Gestión de la calidad: en este tipo de metodologías, la calidad está definida como el grado en que el proyecto cumple con los requisitos y gestionarla implica definir y seguir políticas y procedimientos para asegurar que el proyecto alcance las necesidades del cliente. Para el director de proyectos, una falta de calidad puede llevar a reprocesos y defectos que cuestan tiempo, dinero y repercuten en la posibilidad de no alcanzar el cronograma y costo planeados para el proyecto.
6. Gestión de los recursos humanos: en este proceso el director de proyectos es responsable de la disponibilidad de los recursos humanos requeridos, debe crear un sistema de reconocimientos y recompensas para garantizar la motivación del equipo de proyecto y es a su vez responsable de mejorar las competencias de los miembros del equipo.
7. Gestión de las comunicaciones: consiste en planificar la forma de almacenar, mantener, distribuir y recuperar información, y lo que sucederá con toda la información del proyecto cuando este esté finalizado. Es indispensable para el director de proyectos mantener una comunicación efectiva en aras de garantizar

que los mensajes que viajan entre interesados, sean entendidos de manera adecuada.

8. Gestión de los riesgos: en metodologías tradicionales la gestión de riesgos es fundamental, dado que existe un marco de planeación previa muy fuerte, que al ser quebrantado, podría traer consecuencias graves para el proyecto. En este sentido, el trabajo del director del proyecto no está en centrarse en lidiar con problemas, sino enfocarse en prevenirlos a través de una gestión eficiente de riesgos, mediante la cual el director trabaja para incrementar la probabilidad e impacto de las oportunidades en el proyecto (eventos positivos) y reducir la probabilidad e impacto de las amenazas en el proyecto (eventos negativos).
9. Gestión de las adquisiciones: en metodologías tradicionales las adquisiciones comprenden el proceso formal para obtener los bienes y servicios requeridos por el proyecto. El director de proyectos debe estar en capacidad de crear, leer y administrar contratos; es el responsable del plan de gestión de las adquisiciones, las solicitudes de cambio, la documentación requerida en las adquisiciones y las lecciones aprendidas.
10. Gestión de los interesados: es responsabilidad del director de proyectos identificar los interesados, determinar sus requisitos, definir sus expectativas, evaluar sus intereses, conocer el nivel de influencia, planificar la manera apropiada de dirigir, seleccionar la estrategia de comunicación adecuada y controlar el compromiso de los interesados.

En metodologías tradicionales, las relaciones son indispensables; los interesados deben participar, y su participación debe ser coordinada por el director del proyecto.

11. Responsabilidad profesional: el código de ética y conducta profesional del director de proyectos implica la responsabilidad profesional y social en categorías como: respeto, equidad y honestidad, cuidando que su comportamiento no tenga efectos perjudiciales para el proyecto, para la organización y para el gremio de directores de proyectos.

Bajo el marco de estas metodologías, todo cambio no planeado en los procesos mencionados, puede ser perjudicial para el proyecto e implica una evaluación de la causa que origina la desviación para determinar si se requiere un sobre-esfuerzo del equipo de trabajo o un cambio en las condiciones pactadas inicialmente con el cliente. Es decir, para las metodologías tradicionales los cambios no son bienvenidos, dado que afectan la planeación realizada desde el inicio del proyecto.

2.3.1 Estado de la implementación de metodologías tradicionales en proyectos de software

En la versión 2018 del estudio *Pulse of the Profession* del Project Management Institute, realizada a 4.455 profesionales de gestión de proyectos, 447 altos ejecutivos y 800 directores de la oficina de gestión de proyectos (PMO), de una gama de industrias diversas como gobierno, tecnología de la información, telecomunicaciones, energía, manufactura, cuidado de la salud y construcción, en empresas de América del Norte, Asia Pacífico, Europa, Medio Oriente, África, América Latina y regiones del Caribe, se encuentran los siguientes datos relevantes del estado actual de proyectos desarrollados bajo metodologías tradicionales:

1. Entre 2010 y 2020, se crearán 15,7 millones de nuevas funciones de gestión de proyectos a nivel mundial, y la industria de gestión de proyectos está programada para crecer en \$6,61 billones.

2. 9,9% de cada dólar se desperdicia debido al bajo rendimiento del proyecto, es decir, \$99 millones para cada \$1 mil millones invertidos. \$1 millón se desperdicia cada 20 segundos o \$2 billones cada año.
3. El dinero se está desperdiciando por el bajo rendimiento de los proyectos, debido a que las empresas no logran alinear el diseño del proyecto frente a la ejecución y no se da importancia a la gestión de proyectos, como eje fundamental de la ejecución de la estrategia organizacional.
4. De 2013 a 2018 se presentó una disminución del 27% en la cantidad de dinero que las organizaciones están desperdiciando, debido al bajo rendimiento de sus proyectos.
5. El 41% de los proyectos de bajo rendimiento y el 17% de los profesionales de proyectos entrevistados, indican que el apoyo inadecuado de los clientes o patrocinadores es la causa fundamental del fracaso en el proyecto. En metodologías tradicionales el impacto e influencia de los clientes puede traer dichas implicaciones, dado que su involucramiento en el proyecto se realiza en determinadas fases, principalmente al inicio y en el cierre del proyecto.

Se hace necesario entonces para los proyectos, bajo metodologías tradicionales, crear una cultura de relacionamiento entre el gerente del proyecto, el cliente y desarrollar una hoja de ruta de ejecución colaborativa del proyecto.

6. El 52% de los proyectos completados en el último año han sido afectados por cambios en el alcance inicialmente pactado o por cambios no controlados por el gerente de proyectos y el cliente. Esto implica un aumento del 43% respecto a los cinco años anteriores.

La complejidad de los proyectos también afecta el alcance y aumenta el porcentaje de riesgo de fracaso. De 2013 a 2018 los proyectos de alta

complejidad aumentaron del 35% al 41%, esto indica que la evolución del entorno requiere proyectos con una complejidad cada vez mayor, lo cual demanda una gestión mayor del alcance y de los interesados que lo definen.

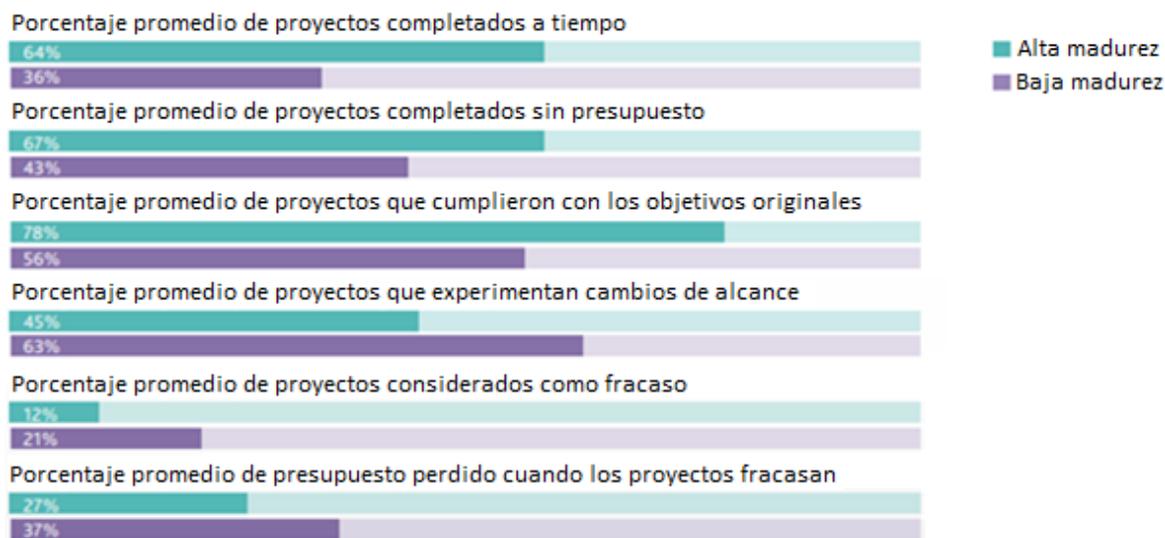
Como se ha indicado en apartados anteriores, bajo metodologías tradicionales, un cambio en el alcance del proyecto puede implicar riesgo en la ejecución del presupuesto planeado, incumplimiento en la entrega del proyecto e insatisfacción del cliente, que podría recibir un producto o servicio diferente a sus expectativas. Este es quizás el mayor reto de las metodologías tradicionales que deben afrontar la ejecución de un proyecto, solventando las vicisitudes que puede ocasionar la definición del alcance del proyecto en la fase inicial de planeación y luego controlar cambios que no atenten contra lo pactado en dicha fase del proyecto.

7. El nuevo entorno de gestión de proyectos requiere combinar las habilidades tradicionales de un gerente de proyectos con habilidades en el ámbito técnico, liderazgo, habilidades estratégicas y gestión empresarial. Los gerentes de proyectos no deben actuar solo como implementadores, deben ser consultores internos que exploren casos de negocio, determinen el valor económico, transformen las tecnologías e iniciativas y determinen a tiempo si los proyectos son viables.
8. Es importante reconocer la necesidad de desarrollar nuevas formas de trabajar. Las organizaciones necesitan replantearse y adoptar cambios continuos para responder ante las necesidades de los clientes y de sus empleados. En la industria de software se hace necesario trabajar tecnologías disruptivas que aceleren la inversión en capacidades, vinculándose a la generación de valor para los clientes.

9. La creación de valor para el cliente en los proyectos entregados es uno de los principales retos de la industria de desarrollo de software, porque es este el indicador real del éxito o fracaso del proyecto y, solo en la medida en que la industria logre fidelizar sus clientes, podrá lograr la competitividad requerida en un mundo de software globalizado.

Partiendo del estudio realizado, solo 1 de cada 10 organizaciones tiene una madurez alta y orienta sus capacidades en la entrega de valor en los proyectos. El 31% de dichas empresas tiene como prioridad el desarrollo de capacidades integradas para la entrega de productos o servicios de valor al cliente.

Ilustración 6. Capacidad de entregar valor al cliente



Fuente: Elaboración propia basada en Wrike Operational Excellence Report (2018).

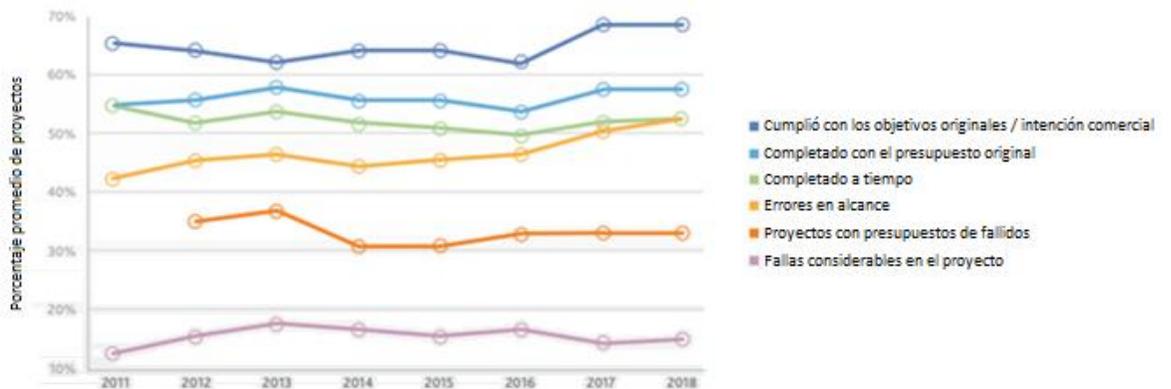
10. De acuerdo con el informe de crecimiento de empleo y brecha de talento del PMI, para 2027, la industria de proyectos necesitará 87,7 millones de personas en roles orientados a gestión de proyectos, lo cual aumentará la necesidad de gerentes de proyectos, expertos en gestión de programas y en gestión de portafolios. Este escenario requiere profesionales competitivos, con capacitación en temas formales, amplia trayectoria profesional y capacidad de transferencia

de conocimiento. Se considera que el papel del gerente de proyectos se debe expandir a:

- Asesores estratégicos que planean, ejecutan y entregan el proyecto
- Innovadores que actúan como propietarios y desarrolladores del producto
- Comunicadores con claridad y manejo de la audiencia
- Adaptables, flexibles y con inteligencia emocional
- Gerentes versátiles con experiencia en diversas metodologías como cascada y *scrum*.

11. La mayoría de variables que miden el éxito del proyecto en metodologías tradicionales como tiempo, costo y alcance, se han mantenido sin cambios a lo largo del tiempo, sin embargo, ha aumentado el porcentaje de proyectos que se ven afectados por cambios en el alcance contemplado al inicio del proyecto. Este cambio podría estar relacionado con los avances en la industria de software, que requiere responsabilidades y habilidades en el equipo de trabajo y considerar nuevas formas emergentes de trabajo.

Ilustración 7. Métricas de rendimiento de proyectos



Fuente: Elaboración propia basada en Wrike Operational Excellence Report (2018).

2.3.2 Riesgos en el uso de aplicación de metodologías tradicionales

El alto nivel de predictibilidad requerida por un proyecto bajo estas metodologías y la planeación que debe realizarse en la fase inicial, implica que exista un alto riesgo en el cumplimiento de los objetivos planteados y en el logro de las expectativas del cliente.

Los principales riesgos de este tipo de proyectos, basado en las estadísticas de gestión de proyectos de Wrike Operational Excellence Report 2018, son:

1. En promedio un proyecto funciona 45% por encima del presupuesto, un 7% por encima del tiempo y ofrece un 56% menos de lo esperado por el cliente. Esto implica que uno de los principales riesgos sea en la fase de inicio y planeación, en tanto que el gerente de proyectos no pueda determinar las variables apropiadas para generar con el cliente acuerdos que garanticen la entrega de un proyecto que satisfaga sus necesidades.
2. Uno de cada seis proyectos de tecnologías de la información tiene un sobrecosto del 200% y casi el 45% de los proyectos reconocen no tener claro su objetivo comercial. En algunos casos el contrato pactado con el cliente indica que estos sobrecostos deben ser asumidos por la empresa responsable del proyecto, lo cual implica un gran riesgo financiero, que amenaza con la sostenibilidad en el tiempo de este tipo de organizaciones.
3. En la encuesta realizada, solo el 34% de los encuestados indica que los proyectos de tecnologías de la información aportan valor a su negocio y el 21% indica que algunas veces genera valor. Este es quizás uno de los mayores retos de los proyectos tradicionales, dado que la falta de involucramiento del cliente durante el proyecto, activa el riesgo de entregar un producto o servicio que no cumpla con lo esperado por el cliente.

4. El 75% de los gerentes de proyecto consideran que sus proyectos desde el inicio están expuestos a riesgos que no pueden controlar y el 78% considera que los requisitos que entrega el cliente no están sincronizados con su negocio. Esto implica que el gerente de proyectos y el equipo deban generar estrategias para obtener la información de requisitos que definen el alcance del proyecto y que definan planes estratégicos para controlar el riesgo de desviación frente a la realidad del negocio del cliente.
5. Un riesgo importante por considerar es la desmotivación del talento en los equipos de trabajo, el 80% de los equipos expone que dedican al menos la mitad del tiempo de su trabajo a reelaborar tareas que ya consideraban terminadas.

Se concluye entonces que los principales contribuyentes al fracaso de proyectos de tecnologías de la información son los objetivos poco claros, la planificación reactiva, posterior a los compromisos realizados con el cliente, los requisitos cambiantes y la falta de integración del equipo con el cliente.

2.4 Teoría del conocimiento

El conocimiento es el eje fundamental sobre el cual las economías están evolucionando y es por esto que es considerado un recurso vital en la transformación de la estrategia que crea innovación y desarrolla ideas para organizaciones y seres humanos (Sugiyama & Meyer, 2008). Viedma en el año 2000 consideró la era actual como la sociedad del conocimiento, definiendo además, que la principal fuente de ventaja competitiva para una organización reside en su conocimiento, en lo que sabe, cómo emplea lo que sabe, cuál es capacidad de aprender cosas y desarrollarlas en sus individuos.

El desarrollo del conocimiento organizacional en las empresas de desarrollo de software es fundamental, no solo porque es una capacidad requerida en la creación de nuevos productos, sino porque permite generar esfuerzos para superar los

competidores en ámbitos como: calidad, servicio al cliente, estrategia organizacional, acciones de gestión del conocimiento y liderazgo de la organización para transformarse.

El arte de crear software reside en las personas, no es una actividad mecánica que pueda trasladarse a otros por medio de documentos o carpetas; es por esto que requiere un reconocimiento fundamental en la evolución de los proyectos de este tipo de empresas, en las que la relación de conocimiento e innovación pueden garantizar el desarrollo de la ventaja competitiva, a través de los procesos apropiados para captar conocimiento, documentarlo, recuperarlo, reutilizarlo, así como para crearlo, transferirlo e intercambiarlo.

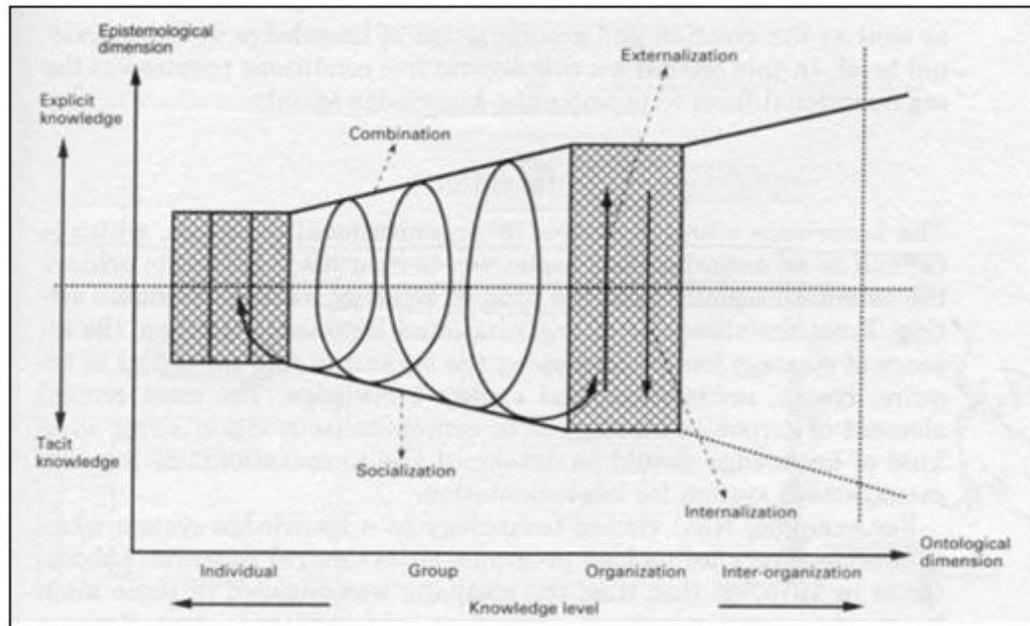
Nonaka & Takeuchi en 1995 definieron el conocimiento como “creencia verdadera justificada, bastante diferente de la información. El conocimiento es acerca de la creencia, acción, significado. El conocimiento es específico y relacional” (Nonaka & Takeuchi, 1995, p. 1), del mismo modo los autores definen la creación de conocimiento organizacional como “la capacidad de una empresa en su conjunto para crear nuevos conocimientos, así como difundirlo en toda la organización y que queden establecidos en productos, servicios y sistemas” (Nonaka & Takeuchi, 1995, p. 2).

A partir de este concepto se profundizará en la influencia que tiene el conocimiento en el logro de objetivos de los proyectos de software y la relevancia que tiene dicho conocimiento en la aplicabilidad de metodologías ágiles o metodologías tradicionales, dado que es en esta capacidad que puede alcanzarse la ventaja competitiva que busca la organización.

Nonaka y Takeuchi (1995) exponen que la innovación requiere la definición de una teoría organizacional que oriente la creación de conocimiento a niveles individual, grupal y organizacional, y para esto crean “la espiral de creación de conocimiento”.

El conocimiento es creado en primera instancia por los individuos dentro de las organizaciones y el mismo se convierte en conocimiento organizacional.

Ilustración 8. Dimensiones de creación del conocimiento de Nonaka y Takeuchi (The Knowledge creating company)



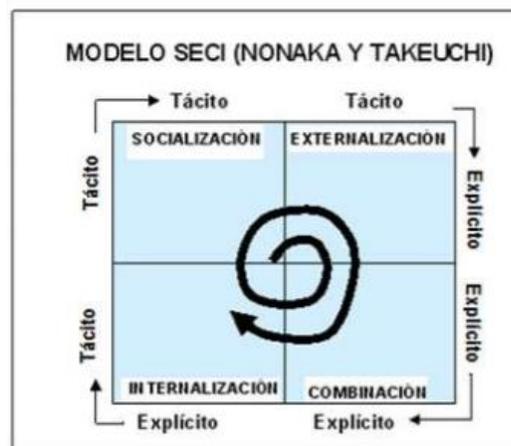
Fuente: Journal 2017, 2(4), 30-37, La Gestión del Conocimiento basado en la Teoría de Nonaka y Takeuchi (1995).

Para determinar la diferencia entre conocimiento tácito y conocimiento explícito es importante entender el conocimiento desde las dos dimensiones enunciadas por Nonaka y Takeuchi (1995), la dimensión ontológica que considera el conocimiento a nivel organizacional de manera opuesta al conocimiento individual y que no puede darse sin la participación de sus miembros, viendo la organización como facilitadora para la creación de dicho conocimiento, y la dimensión epistemológica basada en la relación entre el conocimiento tácito y el conocimiento explícito (Nonaka & Takeuchi, 1995).

“El conocimiento explícito es el conocimiento que puede ser escrito y transferido con relativa facilidad de una persona a otra. Por otro lado, el conocimiento tácito es más difícil de articular, porque a menudo surge por parte de la experiencia” (Briones & Dávila, 2017, p. 30). La espiral se crea cuando se da la interacción entre el conocimiento tácito y explícito, desde la parte ontológica a los niveles más altos de la organización (Nonaka & Takeuchi, 1995).

Con esta propuesta, Nonaka y Takeuchi plantean un relacionamiento en la creación dinámica de conocimiento, proponen que la creación de conocimiento entre tácito y explícito no es unidireccional, sino que puede, dentro de la espiral, crear conversión en cualquier sentido, como lo ilustra la siguiente gráfica:

Ilustración 9. Modos para la creación de conocimiento



Fuente: Tomado de I, Nonaka. “Dynamic Theory of Organizational Knowledge Creation”, (2000).

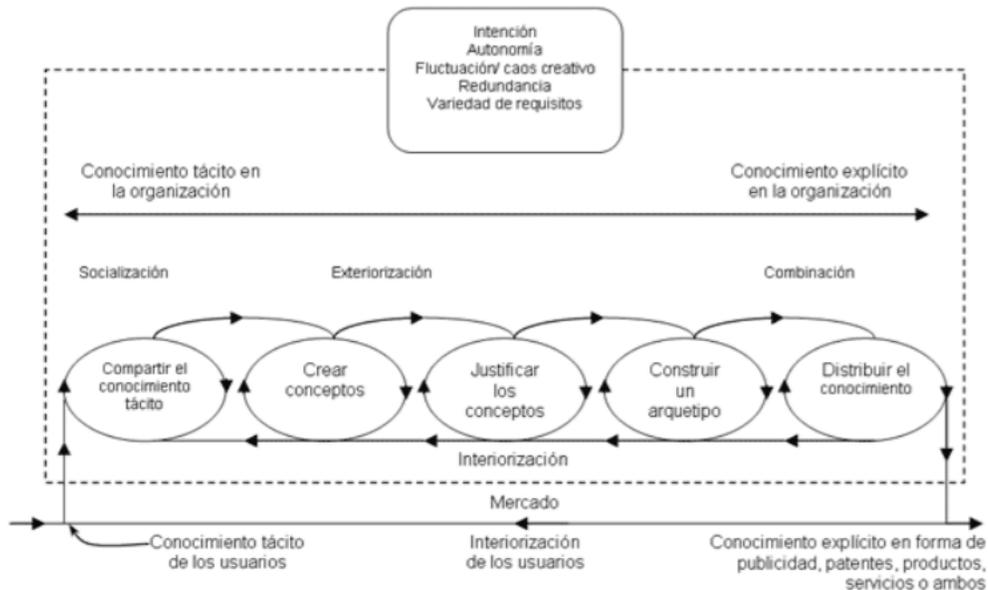
La socialización se da cuando las personas divulgan experiencias y se consigue mediante la observación, imitación y práctica. Este conocimiento es tácito y se comparte como modelos mentales y habilidades técnicas. La externalización permite llevar el conocimiento tácito a conocimiento explícito y es un proceso clave tanto en la creación como en la gestión del conocimiento, dado que requiere el uso

de analogías, hipótesis o metáforas y se desarrolla mediante el diálogo y la reflexión. La combinación es el proceso que convierte el conocimiento en explícito como, por ejemplo, definición de un producto y la adaptación al entorno empresarial. La internalización es la conversión de conocimiento explícito en tácito y está relacionada con “aprender haciendo”, requiere documentación de lo aprendido para interiorizarlo y poder difundirlo (Aguilera, 2017).

En cuanto a las empresas de desarrollo de software, el proceso de creación de conocimiento requiere el desarrollo de las actividades apropiadas en contextos grupales y la creación y acumulación de conocimiento a nivel individual, de modo que pueda ser promovida la espiral de conocimiento y se pueda activar el ámbito de la construcción de conocimiento, el diálogo, el vínculo entre el conocimiento explícito y el aprender haciendo.

Nonaka y Takeuchi (1995) definen el proceso de creación de conocimiento por medio de cinco fases: primero, compartir conocimiento tácito, segundo, creación de conceptos, tercera, justificación de conceptos, cuarta, construcción de arquetipos y quinta, nivelación transversal de conocimientos.

Ilustración 10. Modelo en cinco fases del proceso de creación de conocimiento organizacional



Fuente: Nonaka & Takeuchi (1995).

El método expuesto por Nonaka y Takeuchi pretende plantear un sistema de definición y gestión de conocimiento que sea adaptable a la organización, teniendo en cuenta además, la integración entre teoría de conocimiento y gestión de conocimiento para el desarrollo de la innovación.

En la era del conocimiento y, más aún, en el ámbito de desarrollo de proyectos de software, el conocimiento es el capital más importante para generar una ventaja competitiva en la organización, que sea sostenible en el tiempo. “Una organización puede lograr conocimiento y experiencias con la ayuda de una gestión sistemática y global y puede utilizarlas para proteger su éxito y competitividad en el largo plazo” (Briones & Dávila, 2017, p. 25).

A través del uso del conocimiento que busca ser creado y desarrollado al interior de la organización, la teoría de Nonaka y Takeuchi establece la forma de encontrar, crear, hacer accesible, aplicar capitales intangibles de la organización y fortalecer

la cultura del aprendizaje continuo y el intercambio en la organización (Monavarian & Asgari, 2009).

Se podría inducir entonces, que tanto para gestión de proyectos en metodologías tradicionales como en metodologías ágiles, el conocimiento debe estar en el eje central de la organización, solo de este modo se lograría relacionar la evolución de la organización con la participación entusiasta, activa, con sentido de liderazgo de las personas y garantizar el éxito del proyecto, que al fin y al cabo, depende más de las personas que del método que se desea emplear.

3. Aspectos metodológicos de la investigación

Desde el ámbito de alcance y profundidad, la investigación planteada se clasifica en el tipo de investigación explicativa, pues “pretende establecer las causas de los eventos, sucesos o fenómenos que se estudian y está dirigido a responder por las causas de los eventos y fenómenos físicos o sociales. Busca encontrar las razones o causas que provocan ciertos fenómenos” (Hernández, Fernández & Baptista, 2014, p. 70). Se pretende entonces explicar por qué razones los proyectos de desarrollo de software son fallidos o exitosos y bajo qué condiciones se presenta cada caso. El enfoque se centra en responder por las causas que determinan que un proyecto cumpla o no con los objetivos planteados desde su formulación, explicar en qué condiciones se manifiesta dicho fenómeno y qué relación puede existir con la filosofía de la metodología ágil o la metodología tradicional (Hernández, Fernández & Baptista, 2014).

Desde el ámbito de manejo del objeto de estudio y manera de tratar la información, la investigación se clasifica como cualitativa, dado que:

Se relaciona más con el descubrimiento y el hallazgo que con la comprobación o la verificación, donde el conocimiento es una creación compartida entre el investigador

y el investigado, en la cual los valores median o influyen la construcción del conocimiento; lo que hace necesario meterse en la realidad objeto de análisis para poderla comprender tanto en su lógica interna como en su especificidad. La subjetividad de este tipo de investigación se concibe como un medio y no como un obstáculo para el conocimiento de la realidad humana. (Casilimas, 1996, p. 102)

Para reconocer las bases filosóficas y teóricas de la concepción y evolución de los enfoques ágiles, los enfoques tradicionales, la teoría del conocimiento y la gestión del conocimiento para el desarrollo de proyectos, se realizó un análisis profundo de los conceptos que fundamentan las metodologías ágiles, las metodologías tradicionales y la teoría del conocimiento, así como una revisión de las estadísticas de los proyectos desarrollados en los últimos años, con el fin de determinar si existe alguna variable que caracterice el éxito o fracaso de los proyectos de software. En este mismo sentido, se investigaron los principales riesgos que rodean proyectos en ambas metodologías, con el objetivo de determinar comportamientos típicos al interior de los proyectos o en el entorno que los rodea, que logren sustentar indicios de comportamientos o variables típicas que afecten el resultado final de los proyectos.

A partir de este estudio fue posible categorizar cinco variables que enmarcan el resultado de los proyectos tanto en lo que define el éxito de cara al cliente final, como el éxito intra-organizacional que determina el logro de la ventaja competitiva e impulsa la sostenibilidad de la empresa en una industria tan dinámica, como lo es el ámbito tecnológico:

- La generación de valor para el cliente, determina el grado de éxito que el proyecto alcanzó de cara al cliente, para la organización que lo implementa y para el equipo de trabajo que participa en el proyecto.

- La participación de los usuarios en el proyecto define la conformación del equipo, la asignación de roles y responsabilidades, el ambiente físico en que se desenvuelven, la dinámica de comunicación, las prácticas que facilitan o imposibilitan la generación de conocimiento y los artefactos usados en el proyecto para propiciar ambientes apropiados de comunicación y para formalizar el conocimiento.
- El desarrollo del conocimiento tácito y explícito, determina la manera en que el conocimiento fluye desde el interior de cada individuo hacia los demás miembros y, cómo finalmente, se convierte en el conocimiento explícito que es documentado y transferido a la organización. Teniendo en cuenta que el proceso es cíclico, se hace fundamental detectar las prácticas al interior y al exterior del proyecto, que pueden facilitar o entorpecer dicho proceso.
- La gestión del conocimiento en la fase de requisitos es fundamental, dado que la necesidad del cliente y la determinación del alcance es el primer punto de interacción de conocimiento, que debe manejar el equipo del proyecto de manera estratégica y además la falencia en el entendimiento con el cliente y la determinación de requisitos apropiados es la principal causa de fracaso de los proyectos en los últimos años.
- El liderazgo de equipos muestra la manera en que la organización y los miembros del equipo asumen posiciones de jerarquía o de liderazgo constructivo, que posibilitan la transferencia de conocimiento.

Una vez identificadas las variables de estudio, se realizó una investigación de proyectos del mercado, que incluyó la revisión de proyectos exitosos y proyectos fallidos, que estuvieran implementados bajo metodologías ágiles, tradicionales o híbridas. Se seleccionaron cinco proyectos, cuatro provistos por Diane Elizabeth Strode (Strode, 2012) en su tesis doctoral para la universidad Victoria de Wellington

y un proyecto implementado en una multilatina de origen colombiano. Para cada proyecto se estudiaron y analizaron las visitas realizadas a las empresas, informes anuales públicos de las empresas, sitios web de las empresas, documentación de los proyectos y entrevistas realizadas a gerentes de proyectos, desarrolladores de software, analistas de pruebas, expertos del negocio, entrenadores *scrum*, analistas de negocio y diseñadores técnicos.

A partir de la caracterización realizada, se generó un análisis comparativo que permitió determinar cuáles proyectos se acercaron al éxito o al fracaso y cuáles circunstancias externas, prácticas ejecutadas o metodologías implementadas, facilitaron el resultado del proyecto. A nivel de variables estándar usadas en la medición de proyectos de softwares, se analizó la documentación que existe en cuanto a cumplimiento del proyecto en tiempo, costo, alcance, banco de lecciones aprendidas, indicadores de productividad o desempeño de los equipos de trabajo y satisfacción del cliente.

Para efectuar la comparación entre los proyectos exitosos y proyectos fallidos, se realizó un análisis comparativo, donde se asignaron calificaciones a las variables descritas en la caracterización de los proyectos y se estableció una ponderación a cada atributo, acorde a su relevancia en el ciclo de vida de un proyecto de desarrollo de software. Los resultados esperados en esta comparación fueron sometidos a un análisis documental, sustentado en los conceptos definidos en el marco conceptual, con el fin de buscar, revisar, analizar y clasificar las posibles causas que originan el éxito o fracaso del proyecto.

Finalmente, y sustentado en el marco teórico estudiado y en los proyectos de mercado analizados, se definieron propuestas de mejora que pueden ser aplicadas a la industria de software, para encontrar el camino práctico hacia el anhelado éxito de los proyectos.

4. Análisis de proyectos de desarrollo de software en el mercado, desde los elementos de la teoría de creación de conocimiento

A partir de la investigación de cuatro proyectos de desarrollo de software, realizada por Diane Elizabeth Strode en su tesis doctoral para la universidad Victoria de Wellington (Strode, 2012) y un proyecto implementado en una multilatina de origen colombiano, se realizará la caracterización de los siguientes aspectos, con el fin de determinar el éxito o fracaso de los proyectos:

- Generación de valor para el cliente en el proyecto
- Participación de los usuarios en el proyecto
- Desarrollo conocimiento explícito y tácito en el proyecto
- Gestión de conocimiento en el proceso de requisitos
- Liderazgo de equipos

Este capítulo presenta un análisis de tres proyectos de desarrollo de software ágil, un proyecto tradicional y un proyecto híbrido con prácticas tradicionales y ágiles, proporcionando una visión general de los casos y una caracterización enmarcada en la teoría del conocimiento, que nos permitirá realizar un análisis comparativo entre dichos proyectos y definir mejoras orientadas a la optimización y logro de resultados en los proyectos de desarrollo de software.

Con el objetivo de mantener la confidencialidad en la información, no se utilizarán los nombres de las empresas o los nombres originales de los proyectos, sino que se hará uso de nombres como LAND, STORM, SILVER, ROCK y PLATINUM. A continuación se presenta un resumen general de las características de los proyectos como: tamaño del equipo, metodología implementada y características de las organizaciones en las que fueron ejecutados.

Tabla 1. Características de los proyectos en estudio

	PROYECTOS ANALIZADOS				
	LAND	STORM	SILVER	ROCK	PLATINUM
Tipo de organización	Gobierno	Proveedor de servicios comerciales	Empresa de desarrollo de software comercial	Banco Comercial	Empresa de desarrollo de software
Tamaño de la organización	2000 en Nueva Zelanda	200 en Australasia, Asia y Europa	20 en Nueva Zelanda	5000 en Nueva Zelanda	1500 en Colombia
Propósito del proyecto	Proyecto para mejorar las interacciones de las organizaciones con el público	Proyecto para migrar un sistema heredado crítico a una plataforma tecnológica moderna	Proyecto para proporcionar un nuevo sistema de informes para un cliente externo	Proyecto para reemplazar y mejorar un sistema en línea para ver información de transacciones	Proyecto para desarrollar un nuevo sistema de información que de cumplimiento a los cambios regulatorios establecidos por las normas internacionales financieras
Metodología de desarrollo	Ágil - Scrum	Ágil - Scrum y XP	Ágil - Scrum y XP	Híbrido - Tradicional y Kanban	Tradicional
Tamaño del equipo	6	10	5	7-15	10-15

Fuente: Elaboración propia, 2018.

Para el caso de los cuatro proyectos referenciados en la tesis doctoral mencionada, la información expuesta en este apartado hace parte de visitas realizadas a las empresas, informes anuales públicos de las empresas, sitios web de las empresas, revisión de documentación de los proyectos y entrevistas realizadas, en las que se involucraron gerentes de proyectos, desarrolladores de software, analistas de

pruebas, expertos del negocio, entrenador de *scrum*, analistas de negocio y diseñadores técnicos, entre otros.

Para el caso del proyecto desarrollado en una empresa de software colombiana, la información referencia en este capítulo parte de documentación evaluada del proyecto y entrevistas realizadas al gerente de proyecto, desarrolladores de software, analistas funcionales y analistas de pruebas.

4.1 Presentación de casos

A continuación se presenta el análisis de cada caso.

4.1.1 Caso LAND

4.1.1.1 Resumen del proyecto

El proyecto fue desarrollado en un departamento gubernamental de Nueva Zelanda y su propósito fue modernizar y optimizar el servicio al cliente al cambiar la forma en que el público interactúa con la organización a través de su sitio web. El nuevo sistema debía transformar el proceso existente de una serie de páginas web y formularios en papel, a un sistema dinámico completamente web, garantizando que los formularios y el pago realizado por los usuarios se completaran y enviaran en línea para su almacenamiento en una base de datos relacional y que se realizara el depósito automático en el banco, utilizando el sistema de banca electrónica segura. Se esperaba la creación de un sitio web dinámico y la racionalización del procesamiento de solicitudes de servicio.

El proyecto tenía una alta criticidad en la organización, debido a que impactaba el proceso comercial, el nuevo sistema era parte de una actividad organizativa clave y por esto el gerente de proyecto indicó que era la principal prioridad para la organización. El proyecto tuvo una etapa de planeación previa de 18 meses en los

que finalmente se tomó la decisión de desarrollarlo con el equipo de TI interno y probar utilizando metodologías ágiles.

El equipo siguió el marco de *scrum* que comenzó con una iteración cero para la planificación, y luego mantuvo un sprint semanal con actividades particulares programadas en ciertos días de la semana. Cada sprint incluyó una reunión principal para informar el progreso y llevar a cabo la definición de los requisitos. Se elaboró un sistema de administración de documentos compartido para dar acceso a todos los miembros del equipo a los documentos del proyecto y se utilizaron dos documentos principales del proyecto: uno mostraba el progreso del proyecto en forma de historias de usuario y su estado de finalización, y el otro era un documento de especificaciones de requisitos.

Tabla 2. Actividades dentro de iteraciones

Iteración	Actividades
Iteración 0	<ul style="list-style-type: none"> • Estudiar y decidir sobre la metodología a usar • Definir la plataforma tecnológica y el lenguaje de desarrollo • Definir la duración de iteración de una semana • Crear un plan de proyecto de alto nivel • Capacitar a los miembros del equipo sobre la metodología ágil • Realizar un taller para identificar a los principales usuarios del sistema, desarrollar perfiles de usuario y crear historias épicas (historias de usuarios de alto nivel)
Iteración 1 - 7	<ul style="list-style-type: none"> • Revisar y desglosar historias épicas • Identificar requisitos • Desarrollar software • Realizar pruebas de aceptación del sistema y del usuario • Identificar fallas, corregir y actualizar el servidor de prueba
Iteración 8	<ul style="list-style-type: none"> • Finalización del proyecto: prueba final de software en entorno de prueba formal, incluido el proceso de implementación • Crear documentación • Lanzamiento a producción de acuerdo con los procesos documentados de la organización

Fuente: Elaboración propia, 2018.

4.1.1.2 Generación de valor para el cliente en el proyecto

La organización consideró que el proyecto fue exitoso, dado que se entregó en la última iteración el software funcionando y se avanzó en la implementación de metodologías ágiles para el desarrollo de proyectos internos. El área de TI había experimentado falta de éxito en los proyectos recientes y la sincronización alcanzada en el equipo de trabajo se convertía en un muy buen aliciente para seguir afrontando proyectos de desarrollo de software al interior de la organización.

El proyecto afrontó dos problemas: el primero estaba relacionado con la escasez de personas para el equipo de trabajo. Los miembros del equipo no tenían dedicación completa al proyecto y por tanto tenían solicitudes de su trabajo diario en competencia con el proyecto que no permitían lograr el avance esperado. Esto desencadenó un inconveniente en la recopilación y verificación de requisitos, dado que no todos los miembros conocían los detalles de requisitos y no alcanzaban a tener dominio en el conocimiento y a participar activamente en la solución de problemas. Aunque existía una reunión diaria de coordinación, los miembros no siempre estaban involucrados y esto afectó el progreso del proyecto.

El segundo problema estaba relacionado con la restricción de tiempo en la duración del proyecto, debido a que el desarrollador experto no estaría disponible desde una fecha determinada, lo que implicaba terminar el proyecto en una fecha límite de finalización no negociable.

4.1.1.3 Participación de los usuarios en el proyecto

El equipo del proyecto estaba conformado por seis personas, expertos en dominio comercial de la funcionalidad, expertos técnicos en el sistema existente, *scrum* master o analista de sistemas, un desarrollador, un diseñador web y propietarios del producto final o clientes del nuevo sistema. El equipo no tenía experiencia en la

ejecución de proyectos bajo metodologías ágiles, sin embargo, el gerente del proyecto tenía conocimiento teórico y gran interés en la aplicabilidad de la metodología, a pesar de no tener experiencia en la ejecución.

El equipo estaba ubicado en el mismo lugar, excepto los expertos comerciales que estaban dispersos por el edificio en pisos separados. Esto generaba un inconveniente de proximidad o cercanía física, que causaba problemas en la coordinación de tareas, consumía tiempo y generaba retraso en la gestión de solicitudes de información requerida para el entendimiento de los requisitos del proyecto.

La participación de los integrantes del equipo en el proyecto estaba coordinada por diferentes sesiones de equipo, una iteración cero donde realizaban la planificación del proyecto, definiendo historias de usuario épicas (de alto nivel) y discutiendo aspectos técnicos, una sesión de planificación semanal por iteración, para realizar ajustes a la planeación de la semana, una actividad de seguimiento por iteración, en la que el equipo analizaba las historias de usuario y el estado de finalización actual, una actividad por iteración, en la que el equipo analizaba las historias de usuario, su alcance, prioridad, duración y tareas para implementarla, una sesión diaria e informal de equipo para determinar el estado de las tareas y los impedimentos que debían remover y, finalmente, el lanzamiento de software que implicaba entregar al grupo de usuarios finales, el software funcionando. Estas actividades lograban en el equipo una comprensión común del trabajo a realizar, una sincronización de quien estaba haciendo que para lograr el alcance de la historia de usuario y desarrollaba en el equipo la capacidad de sustitución de miembros con habilidades múltiples, que podría suplir las tareas de los demás y no afectar el cronograma por eventualidades o riesgos materializados en el proyecto.

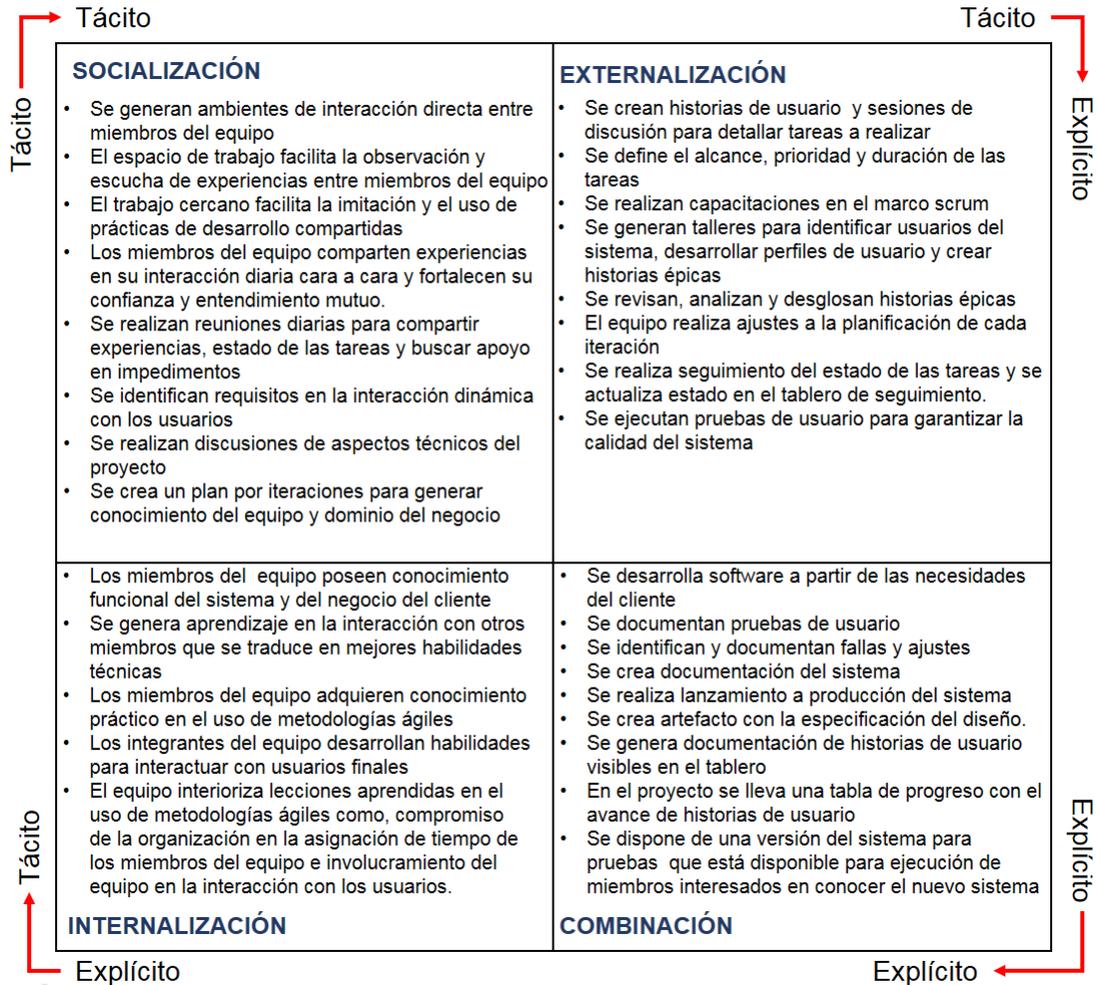
Con el fin de mantener la sincronización del equipo, se utilizaron artefactos para almacenar la información del proyecto, las especificaciones de diseño, las historias

de usuario, una tabla de progreso que mostraba el avance en las historias de usuario respecto al tiempo de ejecución del proyecto, una versión en ejecución del sistema en desarrollo para permitir realizar pruebas de la funcionalidad existente o para conocer el software en ejecución. Estos artefactos permitían la interacción entre los miembros del equipo y los interesados externos, con el fin de mantener información alineada entre los miembros del equipo y con el propósito de proporcionar estatus del proyecto a la organización.

4.1.1.4 Desarrollo de conocimiento explícito y tácito en el proyecto

El uso de metodologías ágiles facilitó el flujo de conocimiento al interior del proyecto, potencializando la generación de conocimiento tácito que partía de las habilidades y experticia de los miembros del equipo, que tenían dominio comercial de la funcionalidad y de los expertos técnicos del sistema anterior, para convertirlo en conocimiento explícito que se almacenaba, distribuía y transmitía a través de la interacción continua del equipo, los repositorios de tareas, documentos de especificación de diseño, software funcionando, cuadros de seguimiento y descripción de historias de usuario. Este ciclo finalmente concluía en la generación de nuevo conocimiento tácito en los miembros del equipo, que se potencializó a través de una interacción dinámica facilitada por sesiones del marco *scrum*. A continuación se realiza una descripción de las actividades realizadas en el proyecto para crear conocimiento y generar flujo dinámico en la organización, partiendo de la espiral de creación de conocimiento definida por Nonaka.

Ilustración 11. Conversión de conocimiento en el proyecto LAND



Fuente: Elaboración propia, 2018.

4.1.1.5 Gestión de conocimiento en el proceso de requisitos

El proceso de requisitos fue flexible, el equipo asistía a una reunión de planificación en el que se realizaba un trabajo conjunto para formalizar el detalle de requisitos y para que todos los miembros del equipo conocieran los detalles de las tareas requeridas para cumplir la necesidad del cliente. El conocimiento tácito surgía a través de la observación, la percepción del sistema anterior y la socialización entre el equipo y el cliente.

Se trataban requisitos a través de historias de usuario que se detallaban en tareas y se perfeccionaban a medida que el proyecto avanzaba en iteraciones. Este esquema de definición y refinamiento de requisitos facilitó la generación de conocimiento de manera rápida y natural, dado que se privilegian el uso de iteraciones cortas y la refactorización de los requisitos ya implementados en el software en pro de beneficiar el resultado final esperado por el cliente.

Sin embargo, la definición de requisitos fue un proceso difícil de abordar, dado que las personas con dominio o conocimiento de requisitos no estaban cerca del equipo de desarrollo, la información no estaba disponible y esto dificultaba el aprendizaje del equipo, porque no permitía que el proceso de transformación de conocimiento tácito a explícito surgiera de manera natural, a partir de la interacción entre miembros del equipo y el cliente. Los usuarios finales no podían ser consultados directamente y el equipo debía trabajar la mayor parte del tiempo con personas de supervisión y gerentes que actuaban como usuarios finales, esta situación generó el riesgo de no recibir la información correcta y adquirir el conocimiento apropiado para suplir la necesidad del negocio.

4.1.1.6 Liderazgo de equipos

Dado que el proyecto se desarrolló bajo metodologías ágiles, se contaba con un liderazgo cooperativo, en el que todos los miembros se sentían empoderados de las actividades por realizar y del éxito del proyecto.

El papel de mayor relevancia lo ejerció el analista de sistemas que asumió el rol de *scrum* master, con un estilo FORMADOR, que estaba enfocado en la generación de espacios de acompañamiento con el equipo, para promover el desarrollo de habilidades personales y apoyarlos en la solución de impedimentos. Su estilo de liderazgo estaba fundamentado en generar apoyo a las necesidades particulares de

cada miembro del equipo, para potencializar su desempeño en el proyecto y motivarlos para hacer de ellos personas más eficientes.

4.1.2 Caso STORM

4.1.2.1 Resumen del proyecto

El proyecto fue desarrollado en Nueva Zelanda, en una organización del sector público encargada de proporcionar servicios de información a los sectores públicos y privados. Debido a la obsolescencia tecnológica del sistema que soportaba el negocio, se definió en un periodo de duración esperado de 18 meses, un proyecto cuyo propósito fue migrar el sistema existente a una tecnología y lenguaje de programación vigente, para garantizar su mantenimiento y la sostenibilidad de las actividades de la organización. El proyecto constaba de 7 sub-proyectos con un grupo diferente de usuarios finales y se esperaba que el sistema actual dejara de funcionar paulatinamente, a medida que los sub-proyectos salieran a producción.

Dado el tamaño del sistema existente y teniendo en cuenta que no iba a ser posible reemplazarlo con el equipo interno que estaba ocupado en sus labores diarias, se contrató para la ejecución del proyecto un equipo independiente a la organización, liderado por un gerente de proyectos con la visión de cumplir con un presupuesto fijo y una fecha de finalización esperada por la organización.

El sistema anterior había operado durante 30 años y había llegado a un tamaño y obsolescencia tecnológica en el que no era viable inyectar mejoras sin renovar el sistema completo, de aquí se desprendía el alto grado de criticidad que tenía para la organización, dado el impacto en la operación del negocio, su complejidad y su tamaño. El reemplazo del sistema existente implicaba mejorar la interfaz con el usuario, mejorar las funciones de procesamiento, optimizar la plataforma y, de este modo, brindar servicios efectivos al público estatal y a la comunidad internacional.

El proyecto se implementó con metodología ágil bajo el marco *scrum*, el gerente del proyecto pasó aproximadamente dos meses analizando el sistema actual y recopilando la poca documentación existente, realizando reuniones y debates al interior de la organización, hasta que se identificaron y priorizaron los componentes a implementar. Posterior a esto, se definieron iteraciones y actividades entre iteraciones; la duración de cada iteración se estableció en dos semanas, pero con el avance del proyecto se ajustó a una semana, dado que el desglose de tareas tomaba mucho tiempo antes de empezar a diseñar y desarrollar.

Por cada sub-proyecto, se definió una iteración cero para determinar los requisitos de alto nivel iniciales y para preparar el registro de historias de usuarios en talleres realizados con el cliente; luego las historias eran tomadas por el equipo y ajustadas con detalles técnicos. En la iteración cero se utilizaban historias épicas y tareas para determinar requisitos funcionales y no funcionales del sistema, las historias épicas eran divididas posteriormente en historias de usuario más pequeñas y, estas, a su vez, eran divididas en tareas, formularios e interfaces. Aunque la mayoría de historias de usuario surgieron durante la iteración cero, también era común que surgieran durante la ejecución de las siguientes iteraciones y esto generaba ajustes a la planeación inicialmente realizada. Diariamente se llevaba a cabo una reunión a la que asistían todos los miembros del equipo para describir el progreso que tenían el día anterior y determinar cualquier problema que debían enfrentar. Semanalmente se actualizaba el gráfico de avance del proyecto que contenía el cálculo del trabajo realizado y el trabajo pendiente de la iteración.

El proceso de desglose de historias tenía como objetivo principal elaborar el detalle de las historias de usuario y crear las tareas relacionadas; en esta sesión se generaban diferentes opiniones de los miembros del equipo sobre la mejor manera de solucionar la tareas y, a medida que avanzaba la discusión, se realizaban análisis sobre la documentación y la necesidad de realizar pruebas en diferentes partes del sistema. Una vez se detallaban las tareas, el equipo asignaba la duración a cada

una, utilizando un sistema de votación que funcionaba por la regla de mayoría y, si alguien no estaba de acuerdo, debía dar las justificaciones del caso para ser consideradas por el equipo. Para la implementación de la tarea se auto asignaba un miembro del equipo que se encargaba de investigar con el usuario final los detalles de la tarea a realizar.

Finalmente, durante cada iteración, se realizaban actividades a demanda como mantenimiento de registros, programación de pares y pruebas de software.

4.1.2.2 Generación de valor para el cliente en el proyecto

El proyecto se consideró exitoso, dado que se realizaban entregas regulares que mostraban el avance efectivo y no se presentaban errores en producción que generaran sobre-costos, ni reproceso en funcionalidades que tuvieran que iniciar desde cero. Se percibió que el equipo disfrutaba su trabajo y los resultados avalaban el buen desempeño que estaban logrando.

El equipo afrontó dos problemas: la interacción con los usuarios finales y la complejidad del sistema de información existente. El rol de usuarios finales fue asumido por ingenieros altamente calificados ubicados en el mismo edificio, en un salón separado y se había estimado que tendrían disponibilidad para consultas sobre requisitos en sus horas de trabajo. Sin embargo, dichos ingenieros trabajaban por turnos y por lo general no eran fáciles de contactar, en ocasiones tenían trabajo crítico y no podían ser abordados, porque su trabajo tenía prioridad sobre las necesidades del proyecto. Como no existía un usuario específico responsable de la función de aclarar requisitos, también era común que se contara con disponibilidad de ingenieros con menor antigüedad y experticia en el sistema existente o que los ingenieros con los que se había empezado un trabajo no estuvieran disponibles para el equipo por los turnos que tenían asignados. Esta situación implicaba que el

equipo se viera en la obligación de cambiar de tarea hasta que los ingenieros apropiados estuvieran libres para ser contactados.

La complejidad del sistema existente se hizo evidente cuando en las fases de pruebas se empezaron a detectar fallos, dada la divergencia que existía entre las funcionalidades y las explicaciones recibidas de las partes interesadas, esto sumado a que cada reproceso ejercía presión sobre el tiempo y el presupuesto del proyecto, originó en el equipo alta presión y un alto grado de desafío ante la meta planteada.

4.1.2.3 Participación de los usuarios en el proyecto

El equipo del proyecto estaba conformado por un gerente de proyecto, ocho desarrolladores de software, un especialista de pruebas y un desarrollador experto en el sistema anterior. Los miembros del equipo se seleccionaron por su capacidad técnica en desarrollo de software y por sus habilidades de comunicación y experticia para trabajar en equipo. El rol de usuarios finales fue asumido por un grupo de 60 a 80 ingenieros que generaban requisitos, aclaraban dudas y aprobaban requisitos. Se contaba con un comité de dirección a quien el gerente de proyectos informaba mensualmente y un equipo que establecía los estándares de comunicación internacional que debían ser cumplidos por el proyecto.

El equipo trabajaba con una dedicación de tiempo completo y estaba ubicado en una habitación grande con los usuarios finales ubicados en el mismo edificio, pero en una habitación diferente. El espacio de trabajo contaba con escritorios alrededor de la habitación sin ninguna clase de división y espacios de reuniones informales con fotografías del proyecto y tableros que facilitaban las actividades de co-creación entre los miembros del equipo. Aunque la expectativa inicial fue que los usuarios finales formaran parte del equipo de proyecto, no se logró y los ingenieros estaban

por fuera del equipo, pero con la responsabilidad de entregar la información requerida por el proyecto.

La participación de los integrantes del equipo en el proyecto estaba coordinada por diferentes sesiones de equipo y auto asignación de tareas, para que cada miembro se sintiera comprometido y se enfocara en el logro de la tarea que él mismo seleccionó, se realizaba una sesión de descomposición de historias en tareas y se actualizaba el tablero de trabajo por orden de prioridad, se realizaba una sesión diaria de pie para que cada miembro informara en qué estaba trabajando, qué hizo ayer y cualquier problema que estuviera afrontando en ese momento. En la sala de trabajo cualquier miembro podía realizar preguntas a los demás miembros y se generaban conversaciones para detectar las soluciones más efectivas y eficientes; cuando un desarrollador tenía un problema con el código o con el entendimiento de alguna funcionalidad, podía libremente pedir ayuda a otros programadores experimentados, lo cual generaba un ambiente de ayuda conjunta ante las dificultades. Las discusiones, negociaciones y toma de decisiones normalmente involucraban a todo el equipo, lo cual enriquecía el conjunto de acciones viables a implementar. En el avance de cada iteración se contaban con sesiones de integración conjunta, que implicaban realizar compilaciones de todos los componentes de software y resolver fallas de manera rápida.

Estas actividades lograban en el equipo proximidad para verse y encontrarse muy cerca ante las necesidades y dificultades, disponibilidad constante para trabajar en equipo, posibilidad de sustitución entre miembros, con lo cual se facilitaba asumir el trabajo del otro en aras de finalizar tareas pendientes, contar con un rol de coordinador que garantiza el cumplimiento de la metodología, lo cual fue un apoyo fundamental en la remoción de los impedimentos que encontraba el equipo.

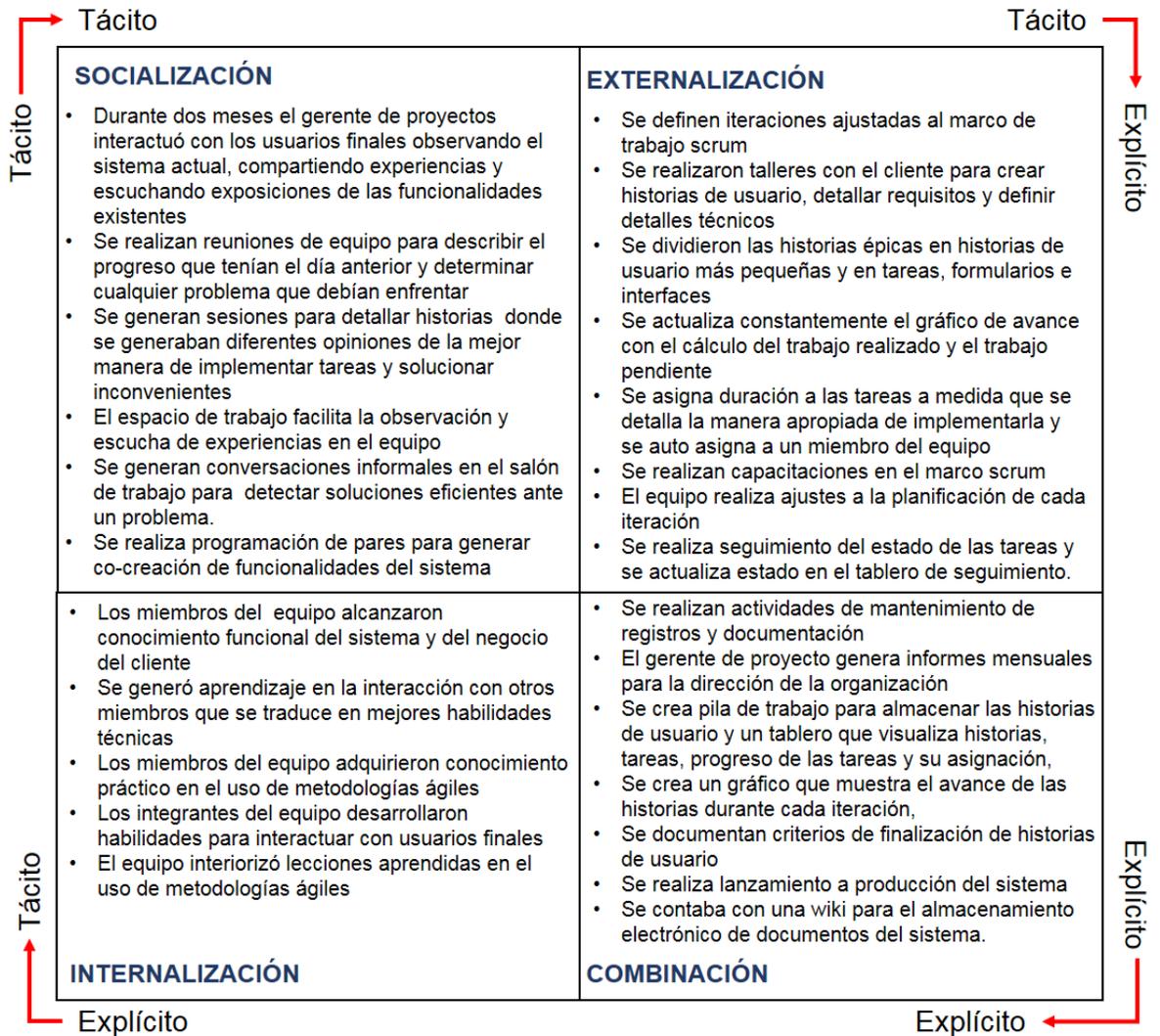
Para mantener la sincronización del equipo se utilizaron artefactos como, una pila de producto en excel para almacenar las historias de usuario, un tablero que hacía

visible al equipo las historias, tareas, progreso de las tareas y su asignación, un gráfico que mostraba el avance de las historias durante cada iteración, una lista de criterios que debe cumplir cada historia para considerarla finalizada, una versión del software funcionando para realizar pruebas y poder observar el nuevo sistema en ejecución y una wiki o sitio de almacenamiento electrónico, para mantener los documentos del proyecto disponibles, tanto para los miembros del proyecto como para la organización.

4.1.2.4 Desarrollo de conocimiento explícito y tácito en el proyecto

El proyecto STORM se desarrolló bajo prácticas ágiles, lo cual potencializó el conocimiento tácito de sus miembros, al rodearse de prácticas basadas en las personas en las que la calidad del resultado se encuentra en mayor medida en el "saber hacer" que construyen las personas y que se exterioriza y formaliza a los demás miembros del equipo, a través del compartir experiencias y tradiciones, la comunicación directa, la creación de pilas de historias de usuario, definición de tareas, diseño de criterios de aceptación y la gestión de un sitio electrónico con documentación. A continuación se realiza una descripción de las actividades realizadas en el proyecto, para crear conocimiento y generar flujo dinámico en la organización, partiendo de la espiral de creación de conocimiento definida por Nonaka.

Ilustración 12. Conversión de conocimiento en el proyecto STORM



Fuente: Elaboración propia, 2018.

4.1.2.5 Gestión de conocimiento en el proceso de requisitos

El equipo del proyecto contaba con empleados permanentes que venían del área de aplicaciones y asumieron el rol de expertos con conocimientos técnicos, tenían dominio en los requisitos del sistema y estaban continuamente disponibles para atender las necesidades de los miembros del equipo. La comunicación con estas personas facilitaba la generación de conocimiento tácito de parte de cada miembro

del equipo, impulsaba la aplicación del conocimiento adquirido en la ejecución del proyecto y propendía la conversión a conocimiento explícito, a través de la definición de historias épicas y tareas que representaban requisitos funcionales y no funcionales que debía cubrir el sistema. La ubicación del equipo en una sala con los especialistas de dominio del sistema generó conversaciones formales e informales, que venían traducidas en aprendizaje del negocio del cliente y que serían aplicadas para la entrega de un producto de valor agregado.

No existía documentación de las funcionalidades del sistema anterior, es decir, no existía un conocimiento formalizado que sirviera de fuente de consulta para el equipo de proyecto y esto generó una alta necesidad de recurrir a expertos de la organización, que entendieran el procesamiento de los datos y que conocieran las funcionalidades del sistema. Aunque se contaba con expertos al interior del equipo, se presentaban dudas o cuestionamientos funcionales que no podían ser atendidas por los expertos y debían ser escaladas a los ingenieros que estaban ubicados en el mismo edificio en una habitación contigua. Bajo este escenario, el proceso de generación de conocimiento tácito y su transformación a conocimiento explícito se hacía complejo, porque era usual que los usuarios que trabajaban en turnos no estuvieran disponibles para consultas sobre los requisitos y que el equipo se viera en la obligación de gestionar la información con personas de diferentes niveles de experticia en la organización o que, en los casos que no se aclaraban, se detuviera el desarrollo de la historia de usuario hasta tener la información apropiada.

4.1.2.6 Liderazgo del equipo

El rol sobresaliente en cuanto a liderazgo del equipo lo ejerció el gerente de proyectos, que fue el promotor de la ejecución del proyecto bajo metodologías ágiles, que impulsó en el equipo el deseo de auto capacitarse en prácticas de *scrum*, para orientar sus esfuerzos a un resultado óptimo para la organización y para el crecimiento personal de cada miembro del equipo.

En el líder del proyecto se evidencian características de liderazgo orientadas con mayor relevancia a un estilo FORMADOR, dado que revisaba continuamente el trabajo adelantando y generaba sugerencias o acciones de mejora para perfeccionar el resultado, contaba con habilidades de escucha y consideraba que el mejoramiento se hallaba en las sugerencias de los demás miembros del equipo, lo cual permitió una gestión efectiva de las necesidades de los interesados del proyecto y, sobre todo, de los miembros del equipo. El líder propiciaba espacios de auto asignación de tareas por parte de los miembros del equipo, de modo que cada persona evolucionara constantemente en el desarrollo de sus habilidades.

4.1.3 Caso SILVER

4.1.3.1 Resumen del proyecto

El proyecto se realizó en una empresa de TI que proporciona servicios de diseño, gestión de redes, desarrollo de software y asistencia técnica para clientes externos, cuyo mercado principal está orientado a organizaciones sin fines de lucro, sindicatos, empresas de servicios profesionales, organismos reguladores y agencias de financiación.

Dado que el sistema anterior estaba implementado en una tecnología antigua y no cumplía las necesidades del negocio, el propósito del proyecto fue desarrollar una aplicación comercial para reemplazarlo, migrarlo a un lenguaje de programación vigente y a una nueva base de datos. Se esperaba construir un nuevo sistema para mejorar la administración de registros obsoletos e inadecuados para un cliente sin fines de lucro, que brindaba servicios al sector salud; esto implicaba que se podía utilizar el sistema existente como guía, pero se debían considerar las necesidades del negocio que no se estaban cubriendo con el sistema actual.

El proyecto era importante para la empresa, puesto que se esperaba mejorar la experiencia de la organización en el uso de nuevas tecnologías y, por esto, se dedicaron todos los desarrolladores de la empresa y se contrató un entrenador para orientar el proyecto bajo metodología ágil.

El proyecto se desarrolló bajo el marco *scrum* con algunas prácticas de programación extrema. Se definió una iteración inicial para planeación de alto nivel y para la definición de historias de usuario épicas (requisitos de alto nivel), seguida de iteraciones con duración de dos semanas que incluían una reunión de planeación orientada al alcance de la iteración, una retrospectiva al final de la iteración y una demostración del producto al cliente. Se realizaron reuniones diarias con duración máxima de 15 minutos para determinar el avance de cada miembro del equipo y para detectar inconvenientes o impedimentos por gestionar. Al final de cada iteración, el equipo entregaba software funcionando en un ambiente de pruebas para que el cliente realizara su revisión y diera una retroalimentación o aceptación.

Tabla 3. Actividades en las iteraciones

Actividad	Descripción
Creación del backlog	Es un repositorio que contenía la acumulación de historias épicas que estaban priorizadas, tenían un tamaño estimado y esperaban ser abordadas en las iteraciones del proyecto. A medida que las historias se iban desarrollando se retiraban del backlog
Definir actividades de cada iteración	Sesión denominada sprint planning que se realizaba por cada iteración. En esta sesión se analizaban los detalles de las historias de usuario, se priorizaban las historias a implementar y se calculaba la velocidad promedio para pronosticar la fecha de finalización del proyecto.
Crear plan de proyecto	El entrenador scrum creó un plan gráfico de las iteraciones a desarrollar, indicando la velocidad esperada del proyecto a partir de la planeación de alto nivel realizada. Esto proporcionó a la organización un registro visual del proyecto con una fecha de finalización estimada.

<p>Definición de historias y tareas</p>	<p>Al comienzo de cada iteración, se seleccionaban las historias de mayor prioridad, se analizaban con detalle, se dividían en tareas individuales y se llevaban al tablero.</p> <p>Se hacía uso de un tablero en la pared que tenía 4 columnas: por hacer, en progreso, en pruebas y finalizado. Las historias y tareas se movían a través de este tablero para que el equipo tuviera la posibilidad de identificar visualmente y en todo momento el progreso de cada tarea.</p> <p>Al comienzo de la iteración, las historias y sus tareas se llevaban a la columna denominada “por hacer”. A medida que cada desarrollador seleccionaba, de acuerdo a su preferencia, la tarea que iba a desarrollar, la pasaba a la columna “en progreso”. Una vez se completaba la tarea, se realizaban las pruebas y se cumplían con los criterios de aceptación, el desarrollador pasaba la tarjeta a la columna “finalizado”.</p>
<p>Desglose de tareas</p>	<p>El equipo descomponía cada historia en tareas, le asignaba criterios de aceptación y, posterior a esto, cada desarrollador se auto asignaba las tareas, a partir de sus conocimientos y preferencias.</p>
<p>Pruebas</p>	<p>El equipo utilizaba tres niveles de pruebas, pruebas unitarias de desarrollador, pruebas de aceptación de historias de usuario y pruebas de aceptación de usuario final.</p>
<p>Programación en pares</p>	<p>Los desarrolladores estaban juntos para ayudarse mutuamente, esto generaba un ambiente colaborativo y fortalecía los lazos entre el equipo de trabajo.</p>
<p>Reunión diaria</p>	<p>El equipo sostenía una reunión diaria de pie con una duración de 5 a 10 minutos para informar el progreso. Cada desarrollador debía responder tres preguntas de manera puntual: qué hice ayer, cuál es mi trabajo para el día de hoy y qué impedimentos tengo.</p>
<p>Demostración del producto</p>	<p>Al finalizar cada iteración, el equipo demostraba a los interesados relevantes, una nueva versión del software que contenía las historias de usuarios desarrolladas. El responsable de la demostración podía ser cualquier miembro del equipo que se seleccionaba de manera aleatoria por un juego de “pajitas”. El demostrador debía preparar un guion y debía conocer con detalle el código desarrollado para comprender las funcionalidades implementadas. En el proceso de preparación se debían realizar</p>

	<p>pruebas y se detectaban errores que debían corregirse inmediatamente.</p> <p>Después de la demostración, las historias completadas se actualizaban en un gráfico de velocidad del proyecto, que reflejaba el ritmo del equipo y la fecha de finalización proyectada.</p>
Retrospectiva	<p>En esta sesión se realizaba una reunión de equipo al final de cada iteración para discutir qué funcionó bien, qué no funcionó bien y qué impedimentos se estaban presentando. Si el equipo determinaba que alguna práctica del marco scrum no estaba generando valor, podían tomar la decisión de abandonarla.</p>
Lanzamiento de software	<p>Cada dos semanas el equipo actualizaba el servidor del cliente con la versión de software que estaba lista para su validación. Esto generaba en el equipo una sensación de finalización y de avance en el software de valor entregado al cliente.</p>

Fuente: Elaboración propia, 2018.

4.1.3.2 Generación de valor para el cliente en el proyecto

A nivel del equipo de trabajo se consideró que el proyecto fue exitoso, dado que se estaba trabajando de manera coordinada y todos se sentían útiles; el equipo se estaba divirtiendo, sentían cohesión y buena comunicación, lo cual estaba generando resultados de valor para la organización.

El principal problema que debió afrontar el equipo fue la interacción con el cliente; el patrocinador del proyecto asignado tenía una participación nula y designó un propietario de producto que no realizaba comentarios significativos del alcance y no mostraba interés en las demostraciones del software entregado. Esto generó desconfianza de la capacidad del propietario de producto y de su idoneidad para asumir un rol relevante en el proyecto. Dado que el alcance del proyecto no estaba claramente definido por el cliente y no se presentaban comentarios apropiados para las historias creadas, era necesaria la participación activa de los usuarios finales que realmente usarían el sistema de información y que conocían el proceso actual del negocio.

A nivel de presupuesto existía una alta incertidumbre desde el inicio del proyecto, lo cual generaba un riesgo mayor si se unía con el alcance incierto del nuevo sistema de información. Esta situación llevó al equipo a manejar una sesión en el tablero de tareas bloqueadas por falta de información o definición de parte del cliente.

A nivel del equipo se tenía una alta dependencia del líder técnico, lo cual generaba que las personas no realizaran el trabajo de una manera adecuada, porque estaban a la espera de que el líder detectara fallos o les realizara recomendaciones de las tareas realizadas. La pérdida de autonomía del equipo de desarrollo era notoria y se veía evidenciada en el reproceso durante las fases de pruebas. En la demostración al cliente de la primera iteración, el equipo se vio enfrentado a grandes desaciertos durante la presentación, lo cual generó cambios en la manera en la que venían trabajando tanto en sus responsabilidades a nivel individual como a nivel grupal.

4.1.3.3 Participación de los usuarios en el proyecto

El equipo del proyecto estaba conformado por tres desarrolladores, un líder técnico y un entrenador *scrum*. Existía además un equipo paralelo de soporte técnico compuesto por 14 personas que estaban ubicadas en una sala contigua al proyecto. El cliente estaba representado por tres personas que tenían el rol de proveer requisitos y verificar la calidad del nuevo sistema, una de estas personas asumió el rol de patrocinador, otro asumió el rol de dueño del producto y otro ocupó el rol de representante de los usuarios finales. Los usuarios finales del sistema formaban un grupo de 30 personas que utilizaban el sistema anterior y que posteriormente serían capacitados para usar el nuevo sistema.

El equipo trabajaba con una dedicación de tiempo completo, excepto por el entrenador *scrum* que trabajaba tres días a la semana. Para facilitar las actividades

del marco *scrum*, se había adecuado una habitación donde cada miembro del equipo tenía su propio escritorio, estaban distribuidos alrededor del salón y no existían divisiones, el equipo podía verse, escucharse todo el tiempo y disponían de un espacio con una gran mesa redonda para las discusiones y para las demostraciones del producto. La sala tenía en sus paredes gráficos de avance, tableros con historias de usuario y avisos relevantes del proyecto.

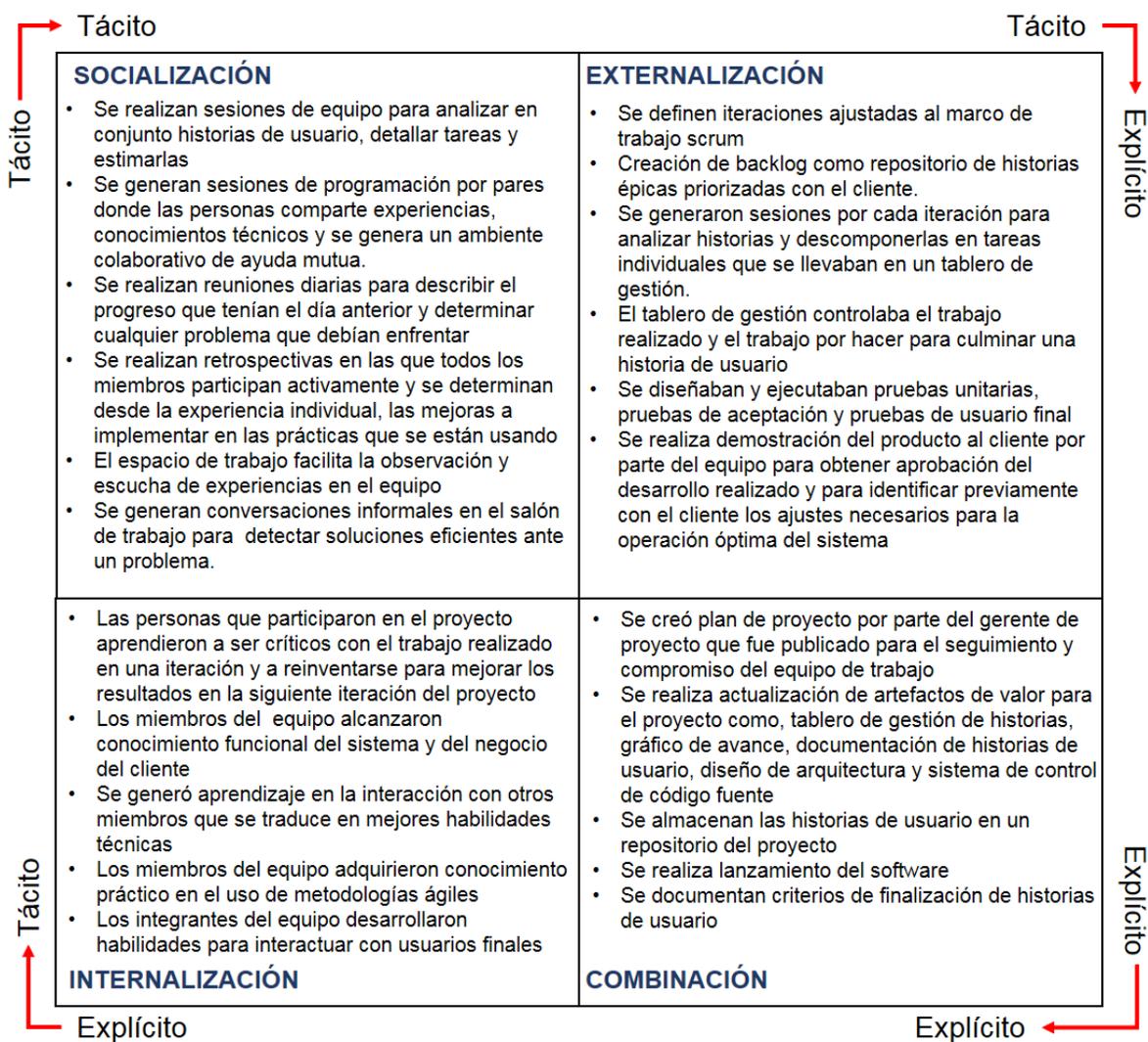
La participación de los integrantes del equipo en el proyecto estaba coordinada por diferentes sesiones, una sesión al inicio del proyecto para preparar la arquitectura básica del sistema, una sesión de planeación por iteración para seleccionar historias de usuario y detallar tareas, una sesión de retrospectiva para discutir qué funcionó bien y qué no funcionó bien durante la iteración, una reunión diaria corta para que cada miembro indicara lo que hizo ayer, lo que planea hacer hoy y cuáles son sus impedimentos, charlas informales entre el equipo que eran facilitadas por el espacio físico designado y en las que todos los integrantes podían escuchar la pregunta, la respuesta y cualquier otra conversación sobre el proyecto, programación por pares para ayudarse mutuamente ante las dificultades técnicas o funcionales, notificaciones a todo el equipo por correo electrónico de temas relevantes que debían quedar por escrito, sesiones de demostración al cliente, que eran lideradas por cualquier miembro y actividades de integración continua para unir compilaciones de diferentes componentes del sistema y garantizar que estaban funcionando como un todo.

Para mantener la sincronización del equipo se utilizaron artefactos como: tablero en la pared con historias y tareas de usuarios, gráfico de avance, avatares para los miembros del equipo y lista de historias terminadas. Todos los artefactos eran visibles para los miembros del equipo desde sus escritorios y se mostraban en una sola pared. Otros artefactos relevantes fueron los estándares de código, software funcionando en un ambiente de pruebas, arquitectura diseñada en tres capas y el sistema de control de código fuente.

4.1.3.4 Desarrollo de conocimiento explícito y tácito en el proyecto

El proyecto SILVER puede ser categorizado como aquel en que se siguieron prácticas ágiles con mayor dedicación, dada la participación del entrenador ágil, que fue contratado por la organización para realizar una implementación basada en prácticas completamente ágiles. Esto condujo el proyecto a ceñirse al manifiesto ágil y seguir el principio de valorar más las respuestas a los cambios, a través del uso de fuertes interacciones con los usuarios y entre los desarrolladores, que el seguimiento estricto a una planificación. Este enfoque en el proyecto dio una mayor relevancia al proceso de comunicación, colaboración y al desarrollo del conocimiento tácito que es compartido con otros a través del contacto personal, promoviendo la transformación a conocimiento explícito en un ambiente de relación directa entre personas, donde la evolución está implícita en el hecho de “aprender haciendo”. A continuación se realiza una descripción de las actividades realizadas en el proyecto para crear conocimiento y generar flujo dinámico en la organización, partiendo de la espiral de creación de conocimiento definida por Nonaka.

Ilustración 13. Conversión de conocimiento en el proyecto SILVER



Fuente: Elaboración propia, 2018.

4.1.3.5 Gestión de conocimiento en el proceso de requisitos

En el proyecto SILVER el levantamiento de requisitos se realizó a través de un método ágil de diseño de casos de prueba, que hacía uso de un formato útil para discutir los requisitos con los interesados del proyecto. Cada caso de prueba preguntaba, “dada la situación actual, cuando esto sucede, entonces se hace esto”. Los talleres para gestionar la información de requisitos se realizaban con los representantes del

cliente que incluían tres personas de la organización, cuyo papel era proporcionar detalles sobre los requisitos y la calidad del sistema. Una de estas personas era el patrocinador designado que tenía el cargo de gerente sénior en la organización, otro tenía el rol de propietario del producto, que era un representante de la administración de la organización y el tercero era un representante de los usuarios finales. Para detallar las historias de usuario, el equipo definió una iteración cero donde se detallaban las tareas a realizar, se discutían aspectos técnicos y se definían los criterios de aceptación. Este análisis conjunto del equipo facilitaba el aprendizaje conjunto y lograba que todos los miembros desarrollaran conocimiento del sistema completo y no solo de la parte que les correspondiera desarrollar.

El grupo de usuarios finales estaba conformado por 30 usuarios con los que el equipo no tenía ninguna interacción por restricciones presupuestarias, esta situación bloqueó el proceso de generación de conocimiento tácito de manera directa con las personas que tenían el verdadero conocimiento del sistema. A medida que el proyecto avanzaba se presentaban mayores dificultades con los requisitos, dado que la respuesta a las solicitudes de información era lenta y no se proporcionaban los detalles sobre la historia a tiempo para que el equipo del proyecto lograra mantener su agenda. El equipo carecía de confianza en la capacidad del representante del cliente para proporcionar comentarios significativos sobre el sistema y sentían que los usuarios finales proporcionarían mejores comentarios, lo cual era frustrante porque no podían ser contactados.

Esta situación dificultó el proceso de socialización en el ámbito tácito del conocimiento, puesto que no existía integración y dinámica de comunicación constante con los usuarios que generara conocimiento a partir de la observación, imitación o práctica de parte de los miembros del equipo. Al existir estas dificultades en el proceso de socialización, no era posible que se generara el flujo de conocimiento hacia el conocimiento explícito en los procesos de exteriorización y combinación.

4.1.3.6 Liderazgo del equipo

El entrenador ágil ejerció el rol de *scrum* master y podría ser considerado como la persona con un rol de liderazgo predominante en el equipo del proyecto. Su estilo demostraba características modo COACH, que se evidenciaba en acciones como: promover en el equipo el uso de herramientas que optimizaran el trabajo y facilitaran el logro de los resultados, promover la auto gestión para que cada miembro del equipo potencializara sus habilidades y se sintiera empoderado de su trabajo, en las sesiones de trabajo realizaba buenas preguntas para apoyar el equipo con su mejor orientación, estaba atento a las necesidades de los miembros del equipo y demostraba en sus decisiones que la persona siempre estaba por encima del resultado de la tarea, buscaba en cada sesión orientar las conversaciones para lograr soluciones efectivas, siempre estuvo comprometido con el reconocimiento a las personas por sus resultados y demostró escucha empática ante los planteamientos o cuestionamientos de los miembros del equipo.

4.1.4 Caso ROCK

4.1.4.1 Resumen del proyecto

El proyecto fue desarrollado en Nueva Zelanda en el área comercial de un banco que tenía centros de operación y sucursales por todo el país, ofrecía servicios a personas naturales, empresas y departamentos gubernamentales locales e internacionales. El proyecto tenía como propósito la creación de una aplicación en línea que permitiría a los clientes ver y administrar sus estados de cuenta de tarjetas de crédito y realizar transacciones en moneda extranjera. Era indispensable que la nueva aplicación operara integrada con las funcionalidades bancarias existentes y con los sistemas externos del banco, por ejemplo empresas de correo, empresas de alerta de servicios móviles y una línea área internacional.

El proyecto no era crítico para la empresa, pero sí tenía una importancia alta, dado que el desarrollo mejoraría la experiencia en línea del cliente, reduciría pantallas en el flujo del proceso de registros, ahorraría costos al disminuir la necesidad de enviar a los clientes estados de cuenta en papel y aumentaría la posición competitiva del banco en el mercado. El proyecto tardó 1,5 años en completarse y se implementó con una metodología híbrida, que tomaba prácticas de metodologías tradicionales en cascada, prácticas ágiles y prácticas de kanban. Tres semanas después del lanzamiento del proyecto, la empresa debió implementar modificaciones al sistema para reparar funcionalidades que no se estaban desempeñando de acuerdo a la necesidad del negocio.

El proyecto inició con una fase de requisitos en la que se empleó un analista de negocios para producir un documento de alcance de alto nivel, a partir de consultas con las partes interesada; esta fase del proyecto incluía levantar requisitos, documentarlos y crear los casos de uso para entregar al equipo de desarrollo. Posterior a esto, se conformó un equipo de proyecto que fue cambiando de tamaño durante el ciclo de vida del proyecto y no se definieron iteraciones, sino que se estructuró un flujo de trabajo continuo que se visualizaba en un tablero de trabajo, denominado kanban.

Tabla 4. Actividades dentro del proyecto

Actividad	Descripción
Sesión de planificación	Esta sesión se llevó a cabo al inicio del proyecto para determinar la complejidad de cada elemento de trabajo.
Elaboración de requisitos	Se creó un documento de 200 páginas aproximadamente para almacenar las necesidades que debía suplir el nuevo sistema.
Diseño del sistema	Se creó un documento de diseño con los detalles de procesamiento requeridos por el nuevo sistema.
Gestión de pruebas	Dado que el ambiente de pruebas era compartido con otros proyectos, se creó una hoja de excel para organizar el uso de datos y garantizar la gestión apropiada de los datos de pruebas.

Reuniones diarias de seguimiento	El equipo sostenía reuniones diarias de pie con duración máxima de 15 minutos, para que cada persona indicara lo que había hecho el día anterior, lo que planeaba hacer hoy y cualquier problema a enfrentar. En esta sesión se discutían problemas del proyecto, se publicaban en el tablero como notas de impedimento y se movían las tarjetas que cambiaban de estado en el kanban.
Gestión de tarjetas del kanban	Las tarjetas kanban eran pequeñas tarjetas de cartón que se usaban para mostrar una funcionalidad por desarrollar y su movimiento a lo largo del tablero indicaba si estaba en progreso en fase de análisis, desarrollo, pruebas, integración o si estaba finalizada. Si una tarjeta llegaba a la columna finalizada, la funcionalidad estaba lista para su lanzamiento al entorno de producción. Cada tarjeta incluía la descripción y los criterios de aceptación para darla por finalizada. Los elementos de trabajo en las tarjetas se priorizaban de acuerdo a la importancia de la funcionalidad y a la capacidad de completarla, incluyendo pruebas de integración.
Retrospectivas	El equipo celebró dos reuniones de retrospectiva durante el proyecto para analizar las prácticas que debían mejorar.
Pruebas de unidad y del sistema	Los desarrolladores realizaban pruebas unitarias automatizadas para garantizar la funcionalidad desarrollada y las registraban en un documento de aprobación del trabajo, que era un registro requerido por el banco. Posterior a la integración de todas las piezas del software, se realizaban pruebas de sistema bajo varios escenarios planeados con anterioridad, que se desprendían de los casos de uso definidos al inicio del proyecto.
Lanzamientos	Se realizó puesta en producción de ocho versiones del sistema, a medida que se tenían funcionalidades listas y mejoradas de la aplicación.

Fuente: Elaboración propia, 2018.

4.1.4.2 Generación de valor para el cliente en el proyecto

El equipo de proyecto consideró que alcanzó el éxito dado que se entregó el alcance esperado por el banco, cumpliendo con el requisito inicial de tarjeta de crédito en línea y adicionando la funcionalidad de moneda extranjera que no estaba incluida

en la planeación inicial. Sin embargo es importante considerar que para alcanzar esta meta, el equipo del proyecto aumentó en tamaño a aproximadamente 15 personas, ocho semanas antes de la finalización del proyecto y que luego de la salida a producción fue necesario realizar ajustes por tres semanas, con el fin de ajustar el sistema a las necesidades del negocio.

El proyecto debió afrontar varios problemas, entre ellos, se resalta la fluctuación del personal asignado al equipo. Con miras a alcanzar la meta planteada y a medida que avanzaba el proyecto, se asignaban personas adicionales, lo cual generó problemas al integrarse en una fase avanzada del proyecto, un equipo tan grande dificultó las reuniones diarias, las cuales se volvieron inmanejables para el equipo, el intercambio de conocimiento se redujo, dado que no todas las personas dominaban las funcionalidades del sistema y no existía un mecanismo de interacción entre todos los miembros del equipo.

Posterior al lanzamiento en producción se detectaron fallos ocasionados por complejidades no identificadas durante la planeación y ejecución del proyecto. Este problema tuvo un impacto en el cumplimiento del plazo del proyecto y se generó porque las estructuras de datos existentes no se ajustaban al nuevo sistema y el equipo debió trabajar fuertemente para detectar las inconsistencias de datos que no permitían la interacción adecuada entre el nuevo sistema, los demás sistemas del banco y los sistemas externos existentes.

El ambiente de pruebas compartido entre varios proyectos afectó la integridad del set de datos definido para las pruebas, esto requirió una negociación del equipo con otras áreas del banco, para controlar el conjunto de datos y vigilar quién utilizaba los datos, quién los modificaba y, de este modo, controlar los cambios en una hoja de excel para garantizar la información a usar en la ejecución de pruebas del nuevo sistema.

4.1.4.3 Participación de los usuarios en el proyecto

El equipo del proyecto lo conformaron: un gerente de proyectos que estaba asignado a múltiples proyectos de manera simultánea, un analista de negocios que desarrolló el documento de alcance y apoyaba la solución de impedimentos a medida que surgían en el proyecto, un especialista técnico que realizó el diseño de los componentes, dos desarrolladores expertos que tenían amplio conocimiento del sistema existente y dos personas de pruebas que debían garantizar la calidad del nuevo sistema. A medida que el proyecto avanzó, se adicionaron al equipo cuatro desarrolladores y un especialista de software experto en comunicación entre aplicaciones, redes, hardware y sistemas operativos. De parte del banco, se contaba con interesados que influían en el desarrollo del nuevo sistema como el jefe de banca para minoristas y el jefe de tarjetas de crédito, que desempeñaron el rol de clientes, el jefe de banca en línea, jefe de división legal y jefe de marketing, que se desempeñaron como un grupo de interesados que debía estar informado del avance del proyecto. De manera externa al banco, se contaba con el banco australiano y tres organizaciones no bancarias que estaban interesadas en el avance efectivo del proyecto, porque estaban relacionadas con el negocio de tarjetas de crédito y serían impactadas por los procesos a implementar con el nuevo sistema.

El equipo constituido al inicio del proyecto trabajaba con una dedicación de tiempo completo, la mayoría había trabajado para el banco por al menos dos años, y algunos durante muchos años más. Durante el desarrollo del proyecto, el equipo se ubicó en una sala grande donde se llevaban a cabo varios proyectos.

La participación de los integrantes del equipo en el proyecto estaba coordinada por sesiones, como reunión al inicio del proyecto para explicar los requisitos y el diseño de sistema, reunión diaria de sincronización del equipo de trabajo, lanzamiento de software al final del proyecto, chat para hacer preguntas entre miembros del equipo,

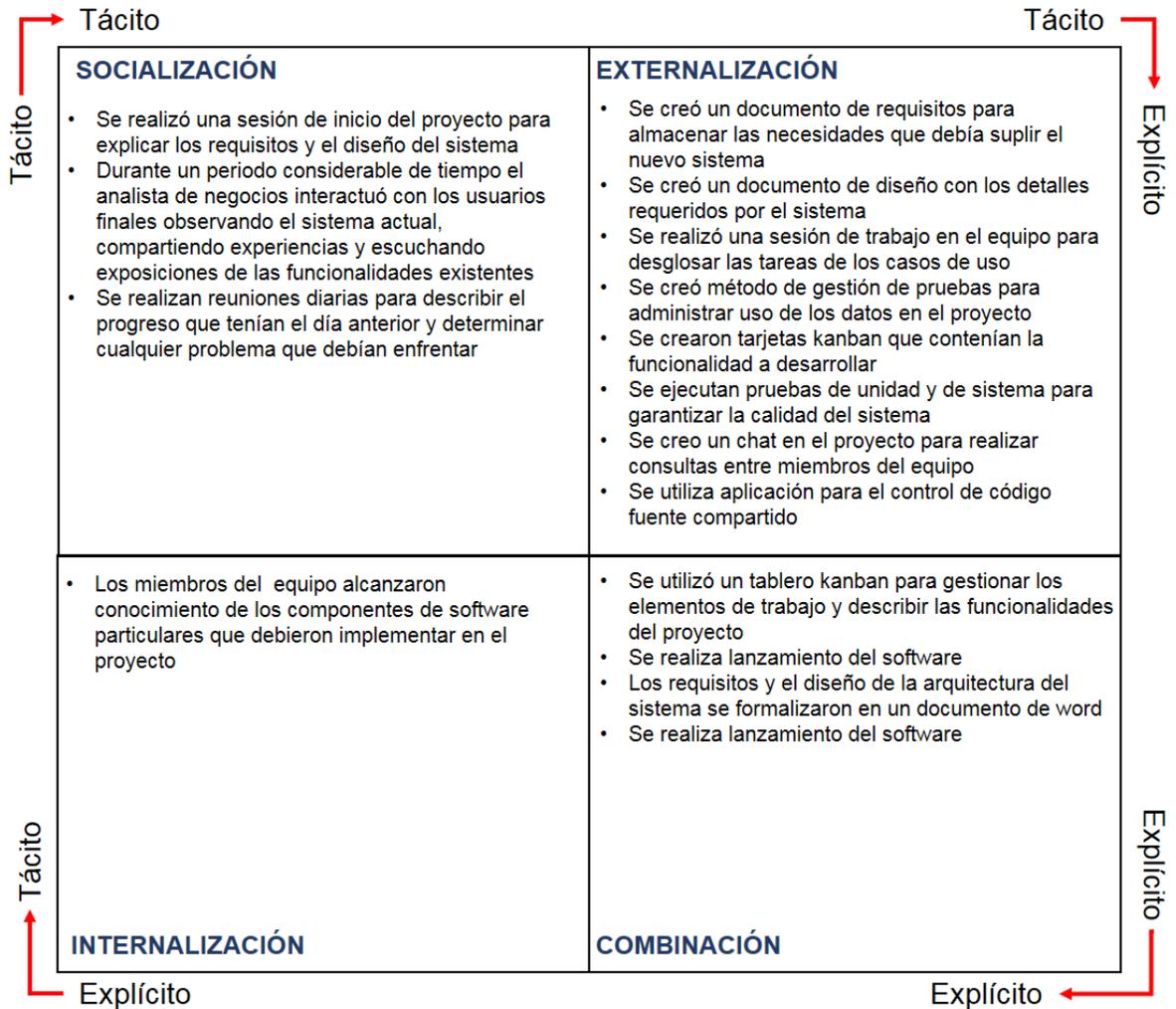
solucionar problemas y realizar consultas, sesión de planificación durante el día para detenerse a planear el trabajo futuro, negociaciones informales del equipo para abordar mejoras o solucionar inconvenientes, sesión de desglose de casos para dividir los casos de uso diseñados en elementos de trabajo, conferencias con usuarios interesados para aclarar requisitos, integración y pruebas continuas para unir las diferentes piezas construidas en un solo sistema.

Para mantener la sincronización del equipo, se utilizaron artefactos como tablero kanban en la pared, para exponer el trabajo actual en forma de elementos de trabajo, tarjetas kanban para describir las funcionalidades a desarrollar, avatar de desarrolladores asignados a las tarjetas, diseño de la arquitectura definida para el nuevo sistema y una aplicación para el control de código fuente, con el objetivo de permitir el uso compartido del código desarrollado.

4.1.4.4 Desarrollo de conocimiento explícito y tácito

El proyecto SILVER fue desarrollado como un híbrido entre prácticas de metodologías ágiles y metodologías tradicionales, por lo cual se evidencia una alta orientación a la planeación en el inicio del proyecto, sin presencia de fases iterativas, con fases en cascada y con una alta orientación a la documentación, para capturar conocimiento en las actividades del ciclo de vida del proyecto. El proyecto basa la gestión del conocimiento en la creación de documentos de requisitos, diseño y arquitectura, con el fin de darle información a los miembros del equipo y, de este modo, asegurar que los requerimientos, el diseño, el desarrollo, el mantenimiento y la gestión del proyecto sean entendidos y ejecutados por los involucrados. A continuación se realiza una descripción de las actividades realizadas en el proyecto para crear conocimiento y generar flujo dinámico en la organización, partiendo de la espiral de creación de conocimiento definida por Nonaka.

Ilustración 14. Conversión de conocimiento en el proyecto ROCK



Fuente: Elaboración propia, 2018.

4.1.4.5 Gestión de conocimiento en el proceso de requisitos

El proceso de requisitos y gestión de alcance en el proyecto ROCK se desarrolló siguiendo prácticas de metodologías tradicionales, un analista de negocio creó un documento en word de 200 páginas, que estaba almacenado en un sistema de gestión documental y que contenía los casos de uso de la información levantada con los usuarios. Esta información fue publicada al equipo en una reunión de inicio del proyecto, en la que el analista explicó los requisitos y el diseño del sistema. A

partir de esta información, se realizó una sesión de desglose de casos en los que el equipo se reunía a trabajar en la división de casos de uso en elementos de trabajo, que registraban en tarjetas kanban y que contenían el detalle de las tareas a realizar. El proyecto presentó problemas con los requisitos definidos, porque el involucramiento de los usuarios se dio de manera activa en la fase inicial del proyecto y no se contó con la disponibilidad apropiada para solucionar dudas o inconsistencias en la fase de ejecución del mismo, lo cual limitó el aprendizaje de parte de los miembros del equipo.

El proceso de conocimiento en esta fase del proyecto tuvo un enfoque mayor hacia el conocimiento explícito que hacia el conocimiento tácito y se externalizó a través de la creación y publicación de documentos de requisitos y documentos de diseño, para darle visibilidad al equipo de desarrollo y asegurar que los requerimientos, el diseño, la implementación y el mantenimiento fueran entendidos y ejecutados por los involucrados. Si bien este modo de trabajo genera una ventaja en la gestión de conocimiento a través de la documentación y reduce las posibilidades de pérdida de conocimiento, al mismo tiempo frustra la generación de conocimiento tácito, que es el más difícil de gestionar y que es el que puede generar una ventaja competitiva sostenible que no sea fácilmente imitada por los competidores

4.1.4.6 Liderazgo del equipo

Aunque en el proyecto se estableció un gerente de proyectos encargado de las labores de planeación, seguimiento, control e informes a los directivos de la organización, el liderazgo predominante fue ejercido por el analista de negocios que mostró un estilo EFECTIVO que se evidenciaba en acciones como: acompañamiento constante en el uso de prácticas ágiles en sincronía con las prácticas tradicionales definidas para el proyecto, creación de una visión y estrategia para alcanzar la meta, gestión de los interesados relevantes para el logro de los resultados del proyecto, gestión de espacios de crecimiento en el equipo para facilitar ambientes de

desarrollo y aprendizaje, incentivación del equipo para generar sentido de pertenencia en el proyecto y obtener compromiso de cada miembro, realización de seguimiento y control para garantizar el logro de los resultados esperados por la organización y búsqueda constante de estrategias para satisfacer los requerimientos del cliente, promoviendo el crecimiento y desarrollo de las personas.

4.1.5 Caso PLATINUM

4.1.5.1 Resumen del proyecto

El proyecto fue desarrollado en Colombia en una organización dedicada a la creación de sistemas de información, que tiene como objetivo central conectar a las personas con el Estado y a las empresas con sus clientes y proveedores, a través del desarrollo de productos con innovaciones tecnológicas que favorezcan el desarrollo sostenible y la competitividad de las empresas que atienden como clientes.

El proyecto PLATINUM surge por la necesidad de mejorar el proceso de solicitud y entrega de medicamentos desde las bodegas de las entidades hospitalarias hacia los diferentes servicios de la institución, como urgencias, pisos de atención y hospitalización de pacientes. Para suplir esta necesidad se hizo necesaria la creación de un aplicativo móvil que permitiera automáticamente sumar en la cuenta del paciente los medicamentos que iban siendo ordenados por el médico para aplicar al paciente y descontar automáticamente de la cuenta los medicamentos que, una vez actualizados, se decidía no administrar al paciente. Estas funcionalidades debían estar completamente sincronizadas con los inventarios de la bodega y pretendían cruzar la información de los medicamentos solicitados contra los despachados y aplicados a los pacientes, para evitar errores y reprocesos entre la solicitud, dispensación (entrega del medicamento) o la devolución a la bodega del medicamento.

El proyecto tenía una alta criticidad para la organización, dado que impactaba los procesos de inventarios, facturación y administración de medicamentos al paciente para una de las mejores instituciones de salud del país, con lo que se esperaba disminuir costos por pérdida de medicamentos, contar en línea con información de la facturación de los medicamentos que se administraban al paciente, disminuir costos de reproceso por comunicación ineficiente entre las áreas y prestar un servicio con mayor calidad para los pacientes. El proyecto tuvo una etapa de planeación previa de 6 meses, en los que la empresa tomó la decisión de desarrollar el software con el equipo de ingenieros de sistemas al interior de la organización y no utilizar mecanismos de subcontratación con otra empresa de desarrollo de software.

El equipo siguió una metodología de desarrollo tradicional, comenzando con una fase inicial de definición y planeación del proyecto, seguida por fases de análisis de requisitos, diseño del sistema, codificación, pruebas, implementación del sistema en el cliente y mantenimiento. En algunos casos, el inicio de una fase esperaba a que se finalizara la anterior y, en otros casos, era posible iniciar siempre y cuando existiera un artefacto o entregable terminado, que sirviera de insumo para iniciar previamente con la siguiente fase. Al final de cada fase, se generaron los controles de calidad necesarios para determinar si el proyecto estaba listo para avanzar a la fase siguiente.

Tabla 5. Actividades en el proyecto

Actividad	Descripción
Planeación del proyecto	Esta actividad se llevó a cabo una vez se conoció la necesidad del aplicativo por parte del cliente y se establecieron algunos parámetros para identificar la viabilidad del proyecto.
Recopilar requisitos, realizar análisis y documentación	Un equipo compuesto por dos analistas funcionales y un arquitecto de software viajaron a la ciudad donde reside el cliente, para conocer en detalle la necesidad funcional e identificar requerimientos no funcionales y

	técnicos. Una vez se tenía claridad del alcance, el equipo regresó a su ciudad origen para crear un documento de requisitos formal que debía ser aprobado por el cliente.
Sesión de inicio y sesión de lanzamiento del proyecto	Una vez hubo claridad entre las partes con la publicación y aprobación del documento de requisitos, se dio inicio formal al proyecto con la reunión de lanzamiento.
Realizar diseños funcionales de la solución, diagramas de requisitos y modelo entidad relación	Conforme se realizaba el diseño funcional del aplicativo móvil y se definía la arquitectura del mismo, se construía un prototipo que iba a ser validado con el cliente y más específicamente con los usuarios finales, para poner a prueba su utilidad y el valor agregado que generaría en el proceso de cargar medicamentos a la cuenta del paciente.
Implementar desarrollos - codificación de funcionalidades	Se realizó el desarrollo del aplicativo móvil con un lenguaje de programación multiplataforma y se realizaron pruebas con los mismos dispositivos móviles que el cliente adquirió para la implementación del proyecto.
Crear casos de prueba para cada requerimiento	Se definieron escenarios de prueba reales para acercarse a la utilización real del sistema en un ambiente de producción.
Actualizar documentación de versión y documentación de instalación	Con la finalización del desarrollo y sus respectivas pruebas, se procedió a la documentación de usuario, técnica y de instalación, a la par que un equipo de pruebas viajaba a la ciudad del cliente para probar en conjunto la aplicación y hacer los ajustes del caso.
Generar documentación de cierre del proyecto	Una vez se realizó la entrega y posterior aceptación del cliente, se procedió a cerrar documentalmente el proyecto.
Sesiones de seguimiento semanales con en el equipo del proyecto	El gerente de proyectos realizaba semanalmente una reunión oficial de seguimiento con todo el equipo del proyecto, en la cual se verificaban avances, dificultades, estado de riesgos y actualización del plan de proyecto.
Sesiones de seguimiento diario de avance modo "daily"	Aunque el proyecto no se desarrolló bajo una metodología ágil, se realizaban reuniones diarias en las cuales se daban a conocer los impedimentos para realizar las actividades por parte del equipo del proyecto, se conocían los avances del día anterior y

	se coordinaban actividades conjuntas para desarrollos y pruebas.
Sesiones para crear lecciones aprendidas con el equipo del proyecto	Cada tres semanas se recopilaban oficialmente las lecciones aprendidas que se daban a conocer en los seguimientos semanales o en los daily's y se registraban en el artefacto de lecciones aprendidas. Posteriormente al finalizar el proyecto (en su etapa de cierre), se publicaron en un banco de lecciones aprendidas.
Realizar informes de avance del proyecto y publicar indicadores	Como parte del proceso de gestión de proyectos, se generaban informes semanales con el avance del proyecto, incluyendo indicadores cuantitativos como: cumplimiento del cronograma (SPI), cumplimiento del presupuesto (CPI) e índice de reproceso.

Fuente: Elaboración propia, 2018.

4.1.5.2 Generación de valor para el cliente en el proyecto

El gerente de proyectos asignado considera que el proyecto no fue exitoso, dado que a pesar de seguir rigurosamente la metodología de desarrollo de proyectos establecida por la organización, el proyecto presentó varios controles de cambio por funcionalidades, que se debían incluir y no estaban consideradas en el documento de requisitos firmado por el cliente; además, se incumplió el plazo de tiempo establecido para la entrega del aplicativo, pues cuando esta se realizó, la necesidad del usuario final había evolucionado, debido a que era necesario registrar información adicional, lo cual imposibilitaba la utilización adecuada de una pantalla de un dispositivo móvil.

El cliente decidió utilizar el aplicativo y posponer para una próxima versión los ajustes que al momento de la entrega eran requeridos, se recibió una retroalimentación positiva dado que, aunque la necesidad había evolucionado y los tiempos se incumplieron, el aplicativo generaba valor sobre el proceso que

actualmente se llevaba, principalmente porque reducía tiempos y evitaba errores en el cargue y devolución de los medicamentos.

De parte de la oficina de dirección de proyectos se recibieron buenos comentarios, dado que era el primer aplicativo móvil que se construía en la organización y aunque hubo inconvenientes con el cliente, terminó aceptando el desarrollo e implantándolo en producción. Además, los indicadores del proyecto estuvieron sobre la media comparado con los demás proyectos implementados en la organización.

Al interior del equipo se contaba con un ambiente cálido de trabajo en equipo, aunque era el primer aplicativo móvil que se construía en la organización, se asumió el reto con responsabilidad; con el avance del proyecto se presentaron inconvenientes técnicos y de rendimiento, que fueron solucionados con el apoyo y compromiso de todos los miembros del equipo.

4.1.5.3 Participación de los usuarios en el proyecto

El equipo del proyecto estaba conformado por un gerente de proyectos, dos analistas funcionales, tres ingenieros de desarrollo, dos analistas de calidad, un arquitecto de software y contaba con la participación parcial del jefe de investigación y desarrollo. El equipo del cliente estaba ubicado en una ciudad diferente al equipo del proyecto y contaba con un analista encargado de definir las necesidades funcionales, además se contaba con el apoyo de dos enfermeras que sirvieron de puente con los usuarios finales y fueron imprescindibles a la hora de definir requerimientos que optimizaran el proceso de carga y devolución de medicamentos y apoyaran las pruebas de aceptación del aplicativo.

El equipo trabajaba con una dedicación de tiempo completo y estaba ubicado en puestos de trabajo contiguos, para mejorar la comunicación y evitar el reproceso en desplazamientos. Los roles y actividades de cada integrante del equipo estaban

plenamente identificados; los analistas funcionales eran responsables de la definición y aprobación de requisitos, posteriormente realizaban el diseño funcional; los ingenieros de desarrollo tomaban como base el diseño funcional para codificar la solución y luego entregar su insumo al área de pruebas, que era la encargada de validar la calidad del aplicativo y posteriormente realizar pruebas conjuntas con el cliente; tanto el arquitecto de software como el jefe de investigación y desarrollo, definieron la arquitectura del aplicativo móvil y daban pautas de buenas prácticas de desarrollo. El gerente del proyecto trabajaba de manera transversal cumpliendo sus funciones de seguimiento a las actividades, identificación de riesgos, actualización de riesgos, generación de indicadores y actuaba como mediador en el equipo.

Durante el proyecto, el equipo utilizó artefactos para garantizar la sincronización de sus actividades, entre los más importantes se encuentra el plan de proyecto, el cual definió el rumbo, dado que integra los objetivos del mismo, su alcance, su cronograma, la estructura de desglose de trabajo, el plan de comunicaciones, la estrategia para la gestión de interesados, el plan de costos, los requerimientos de compras y adquisiciones y la gestión de los riesgos. A nivel de requisitos y análisis fueron fundamentales los documentos de requisitos y el diseño funcional, insumo principal para los ingenieros de desarrollo, como también los documentos de arquitectura que dieron la pauta para la construcción del aplicativo móvil.

En conclusión, la organización vela por el cumplimiento del proceso, generando la documentación necesaria para la gestión del conocimiento y para recolectar información histórica que le permita planear, estimar y ejecutar futuros proyectos con mayor acercamiento a la realidad y a las prácticas adoptadas.

4.1.5.4 Desarrollo de conocimiento explícito y tácito en el proyecto

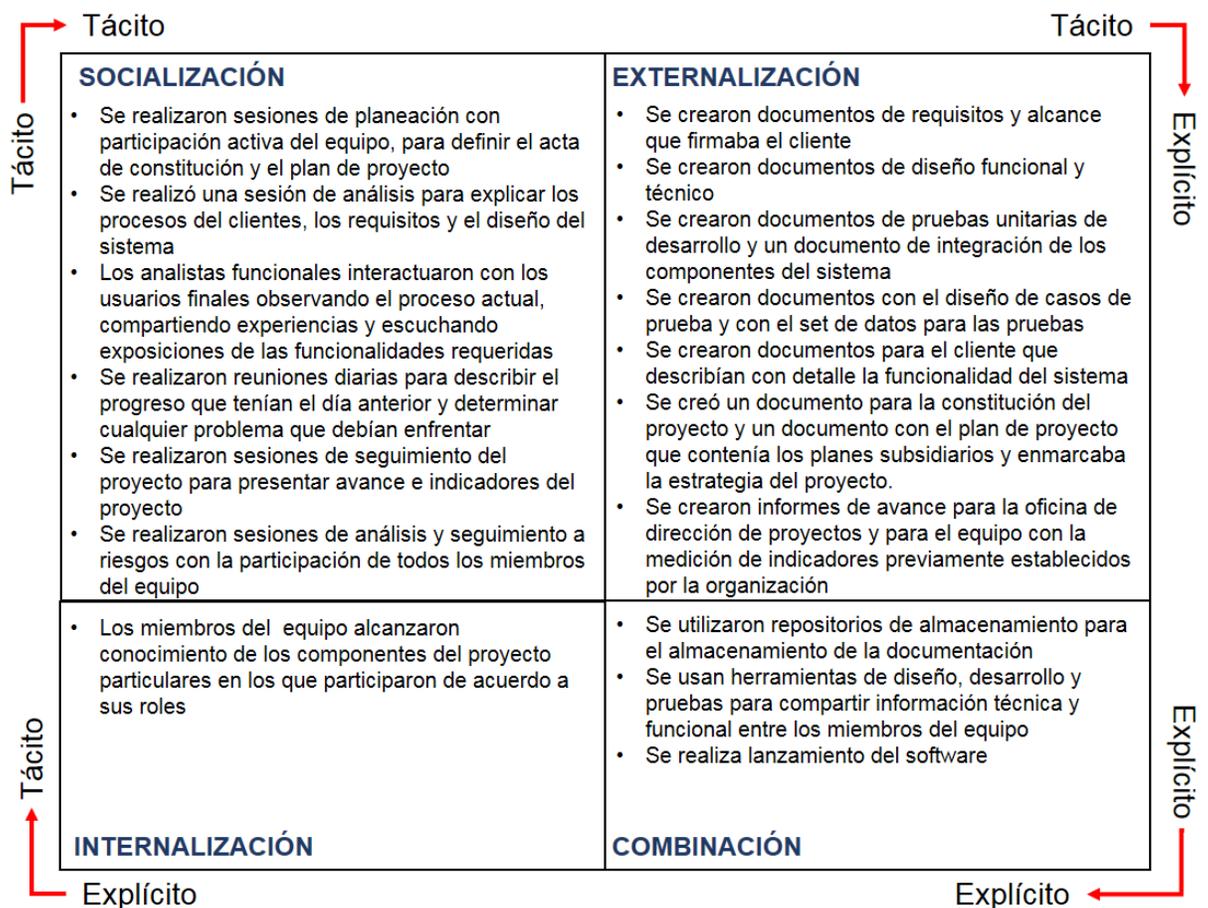
En el proyecto PLATINUM la gestión del conocimiento estuvo basada en la documentación generada en cada fase del proyecto, para capturar conocimiento que se trasladaba a la siguiente fase del ciclo de vida. Bajo la metodología tradicional asumida, el proceso de conocimiento tiene un enfoque explícito, dado que el intercambio de conocimiento entre los diferentes roles está fundamentado en documentos que se producen y entregan a las personas responsables de la próxima etapa del proyecto.

En la fase de inicio y planeación se crearon documentos de constitución del proyecto y un plan de proyecto que fue la base de conocimiento del alcance, tiempo, costo, restricciones y riesgos a gestionar en el proyecto, así como el conjunto de acciones definidas para alcanzar el objetivo planteado; en la fase de requisitos se creó un documento que contenía el alcance firmado por el cliente, en la fase de diseño se crearon casos de uso, modelo entidad relación en la herramienta corporativa, documentos técnicos de diseño y la arquitectura del sistema, en la fase de desarrollo se codificó el sistema, se crearon documentos de pruebas unitarias, documentos de integración de los componentes del sistema y se generó documentación en herramientas de control de código fuente, en la fase de calidad se crearon documentos de casos de prueba en la herramienta corporativa y set de datos para las pruebas, finalizando el ciclo de vida se creó documentación de usuario para el cliente, con el fin de describir de manera detallada la funcionalidad del nuevo aplicativo.

Esta documentación se debió estructurar y actualizar de manera cuidadosa por parte del equipo de proyecto, dado que la organización consideró que era la manera apropiada para asegurar la calidad de sus productos y que el conocimiento se podía capturar, distribuir, medir y gestionar a través de la actualización de la información en repositorios, que dieran la posibilidad de reutilización y consulta por todos los

miembros de la organización. De este modo, se reducen las posibilidades de pérdida de conocimiento como resultado del personal que abandona el proyecto y se mejora a través del enfoque de exteriorización de conocimiento, el proceso de perfeccionamiento continuo en los proyectos futuros de la organización

Ilustración 15. Conversión de conocimiento en el proyecto PLATINUM



Fuente: Elaboración propia, 2018.

4.1.5.5 Gestión de conocimiento en el proceso de requisitos

El proceso de requisitos y gestión de alcance en el proyecto PLATINUM estuvo enfocado en prácticas de metodologías tradicionales, con las que se pretendía lograr una comprensión detallada del problema a solucionar en el proyecto, y para

esto era fundamental identificar las personas que harían uso del sistema y especificarían los requisitos y necesidades de los usuarios, identificar las restricciones del sistema, es decir, las características del entorno empresarial que limitan la funcionalidad o el rendimiento del sistema, realizar entrevistas, grupos focales y reuniones de grupo para determinar las funcionalidades a implementar, motivar la participación de muchas personas para que se definan los requisitos desde diferentes puntos de vista, realizar prototipos para crear una imagen gráfica de las pantallas del nuevo sistema y, de este modo, facilitar el entendimiento de los usuarios, solucionar ambigüedad en los requisitos para obtener un documento coherente y consistente entre características del sistema, crear escenarios de interacción entre el usuario y el sistema para presentar de manera didáctica el cambio en los procesos cuando entre en funcionamiento el nuevo sistema y gestionar la aprobación del documento final por parte del cliente, para luego garantizar que el proyecto desarrolle únicamente el alcance pactado en el acuerdo firmado.

Para los cambios de requisitos que se presentaron durante la ejecución del proyecto, el gerente del proyecto y los analistas funcionales realizaron un análisis detallado del impacto en variables como tiempo, costo, satisfacción del cliente, calidad, riesgos, alcance, y tramitaron el correspondiente control de cambios con la aprobación del cliente y de la oficina de dirección de proyectos.

El proceso de gestión de conocimiento en esta fase del proyecto tuvo un alto enfoque en el conocimiento explícito, dado que se fundamentó en un modelo determinístico que supone que los requerimientos y necesidades del proyecto deben ser completamente definidos y documentados desde el inicio del proyecto y no se concibe cómo construir un sistema exitoso sin tener total claridad de los requerimientos y el compromiso de que estos sean inamovibles y no cambien durante la ejecución del proyecto. Tanto la exteriorización como la combinación en el proceso de conocimiento fueron fundamentales en el proyecto, buscando mitigar

la posible pérdida de memoria corporativa o fuga de conocimiento en las personas, con una documentación detallada que debe ser almacenada en los repositorios corporativos.

Sin embargo, el gerente de proyectos manifestó su preocupación, porque la gran cantidad de documentación que dejó el proyecto no fue actualizada de manera constante, por los cambios propios de la evolución en los procesos del cliente en un ambiente tan cambiante como el desarrollo tecnológico.

4.1.5.6 Liderazgo del equipo

En el proyecto PLATINUM, el rol de liderazgo fue ejercido por el gerente de proyectos, dada la autoridad que la oficina de dirección de proyectos y la organización le otorgaron. Se evidencia un estilo de liderazgo EFECTIVO con características provenientes de prácticas tradicionales, orientadas al cumplimiento de las tareas definidas en el cronograma del proyecto, sin embargo, también se distingue un liderazgo orientado a satisfacer las necesidades del equipo y a promover espacios colaborativos de confianza, donde las personas puedan expresar su experiencia, sus emociones y las mejoras que consideran deben ser implementadas para alcanzar los resultados esperados por el cliente. El líder se ocupó de crear una atmósfera de colaboración y comunicación en la que consultaba constantemente la opinión de los demás miembros del equipo y apreciaba los consejos recibidos, esto generó un alto nivel de participación y un respaldo constante a las decisiones que se tomaban en el proyecto.

4.2 Análisis comparativo entre los proyectos presentados

A continuación se presenta un análisis comparativo entre los cinco proyectos estudiados, donde se asigna una calificación a cada una de las variables analizadas en la caracterización de los proyectos y se asigna una ponderación a cada atributo, acorde a su relevancia en el ciclo de vida de un proyecto desarrollo de software.

Las variables, generación de valor para el cliente en el proyecto y desarrollo de conocimiento explícito y tácito, son las de mayor relevancia, dado que cubren los dos pilares fundamentales de un proyecto: la satisfacción y fidelización de los clientes a través de una estrategia integral que permita cautivar a los clientes nuevos y a los que ya tienen un proyecto con la empresa, ofrecerles un valor extra que garantice una relación a largo plazo; y la gestión de conocimiento organizacional que es aquella que permite el desarrollo de una ventaja competitiva al interior de la empresa de software y que solo podrá ser alcanzada en la medida en que la empresa desarrolle la capacidad de crear conocimiento nuevo, diseminarlo en la organización e incorporarlo en productos, servicios y sistemas (Nonaka, 1988).

Para facilitar la calificación de las variables, se asignan las siguientes notaciones:

- Valor: variable correspondiente a la generación de valor para el cliente en el proyecto. Se asigna un peso del 25%
- Usuarios: variable correspondiente a la participación de los usuarios en el proyecto. Se asigna un peso del 15%
- Conocimiento: variable correspondiente al desarrollo de conocimiento explícito y tácito en el proyecto. Se asigna un peso del 25%
- Requisitos: variable correspondiente a la gestión de conocimiento en el proceso de requisitos. Se asigna un peso del 20%
- Liderazgo: variable correspondiente al liderazgo de equipos. Se asigna un peso del 15%

Tabla 6. Calificación de variables caracterizadas por proyecto

VARIABLES	LAND	STORM	SILVER	PLATINUM	ROCK
VALOR	3,5	3	3,5	3	3
USUARIOS	2,5	2,5	2,5	3	2
CONOCIMIENTO	3,5	3	4	2,5	2,5
REQUISITOS	3	3	4	3	2,5
LIDERAZGO	3	2,5	3,5	3	2,5

Fuente: Elaboración propia, 2018.

A continuación se presentan los resultados de los proyectos, partiendo de la ponderación antes mencionada.

Tabla 7. Resultados por proyecto

VARIABLES	LAND	STORM	SILVER	PLATINUM	ROCK
VALOR (25%)	0,875	0,75	0,875	0,75	0,75
USUARIOS (15%)	0,375	0,375	0,375	0,45	0,3
CONOCIMIENTO (25%)	0,6	0,6	0,8	0,6	0,5
REQUISITOS (20%)	0,45	0,375	0,525	0,45	0,375
LIDERAZGO (15%)	0,875	0,75	0,875	0,75	0,75
TOTAL	2,3	2,1	2,575	2,25	1,925

Fuente: Elaboración propia, 2018.

El proyecto con mayor grado de éxito fue el proyecto SILVER, ya que logró alcanzar un mayor nivel de satisfacción al cliente, superando barreras como la disponibilidad de los usuarios finales, para interactuar con el equipo y entregar información de valor, logró además mitigar la incertidumbre que existía con el presupuesto, dada la falta de claridad en el alcance y sobreponerse a una entrega fallida en la primera demostración realizada al cliente. Este proyecto demostró que el incremento de

software diseñado, programado, testeado, integrado y liberado a través de iteraciones aumenta la efectividad, dado que abre la posibilidad de comunicación y colaboración entre cliente y desarrolladores, y logra a través de la experimentación y la revisión del software funcionando, retroalimentación temprana del cliente para realizar ajustes o refactorización del sistema y detectar problemas a tiempo, logrando la satisfacción del cliente por encima de cumplir con un documento de requisitos firmado que requeriría detallar cada funcionalidad desde el inicio del proyecto y que no contaría con la evolución de necesidades propias del mercado tecnológico, que se ve inmerso en escenarios de rápido y constante cambio en los que los requerimientos se hacen inestables.

En un esquema como el manejado en el proyecto SILVER, los cambios a los requerimientos son bienvenidos, porque prima la satisfacción del cliente por encima de la documentación y el cumplimiento de los procesos. La alta adaptabilidad del equipo del proyecto y el uso de procesos técnicos y de gestión bajo el marco *scrum*, lograron que la evolución de los requerimientos se aprovecharan como cambios de provecho que surgen del conocimiento generado en conjunto con el usuario, durante el proceso iterativo de desarrollo.

Cuando el cliente tiene la posibilidad de revisar versiones previas del sistema y no recurrir a su imaginación o capacidad de descripción para plasmar la necesidad en un documento de requisitos, se alcanza un ambiente de empatía que favorece la socialización, la colaboración, el entendimiento del sistema y una cultura de confianza que genera un entorno de interacción y permite que los miembros del equipo y el cliente compartan sus experiencias, su cultura, sus emociones y esquemas mentales, transfiriendo así el conocimiento tácito de un miembro del equipo a otro.

En cuanto a la gestión del conocimiento y remitiéndonos a lo planteado por Nonaka, el proyecto SILVER logró desarrollar sesiones de trabajo, métodos y técnicas de

comunicación que permitían que los miembros del equipo internalizaran el conocimiento explícito y que, a su vez, logran compartir el conocimiento tácito que cada uno cada uno traía en su “inconsciente” y que se ejecutaba de manera mecánica. Sesiones como la retrospectiva, la reunión diaria de seguimiento, las charlas informales entre el equipo, la programación por pares y las sesiones de demostración al cliente, que eran precedidas por cualquier miembro del equipo, lograron fomentar la socialización por medio de la comunicación cara a cara, se promulgaba la colaboración entre miembros del equipo y el cliente para compartir conocimiento tácito y se transfería constantemente conocimiento, lo cual demuestra el paso de la exteriorización a la internalización del conocimiento. Ligado completamente a lo expuesto en el manifiesto ágil, el proyecto privilegió a las personas por encima de los procesos y resultados, lo cual facilitó el proceso de compartir conocimiento tácito y promulgó una cultura de confianza y motivación, que posibilitaron las acciones ejercidas por los miembros del equipo y el cliente para compartir conocimiento explícito.

Finalmente, el proyecto logró cumplir con uno de los preceptos fundamentales del manifiesto ágil: “construya los proyectos alrededor de individuos motivados. Deles el ambiente y apóyelos en lo que necesiten. Y confíe en ellos” (Beck et al., 2001, p. 3). Esto generó un gran sentido de pertenencia de cada miembro del equipo, cada persona llegó a sentir los resultados del proyecto como propios y trabajaron por la creación de resultados sobresalientes, la colaboración en momentos de problemas, la socialización, la difusión del conocimiento, el apoyo y compromiso por los resultados de su empresa.

El proyecto con el desempeño más bajo fue ROCK y esto se evidenció en el reproceso que surgió luego de la salida a producción, por la interacción inadecuada entre el nuevo sistema, los demás sistemas del banco y los sistemas externos. Esta situación implicó modificaciones por tres semanas, con el fin de ajustar el sistema a las necesidades del negocio. Adicional a esto, el proyecto requirió aumentar el

tamaño del equipo en aproximadamente 15 personas, ocho semanas antes de la finalización del proyecto. Dicho reproceso posterior a la entrega al cliente y el aumento de personas en el equipo a pocas semanas de finalizar, debió originar un sobre costo a la empresa por la falta de control en los diferentes procesos del proyecto y, especialmente, en el aseguramiento de la calidad del producto que debía haber detectado los fallos durante la planeación y ejecución del proyecto.

La mezcla de prácticas ágiles y prácticas tradicionales en sí, no debió constituir un inconveniente para el proyecto, dado que es usual encontrar en la industria de software, que ante la realidad cambiante de las empresas se encuentren posturas que a la hora de definir la metodología de proyectos a usar, deben optar por aquella que se adecue a la necesidad del negocio o por una adaptación que favorezca la organización y el proyecto. El problema se presenta cuando el proyecto muestra síntomas de la mala planificación, ausencia de comunicación asertiva, falta de involucramiento del equipo en la consecución de los resultados, asignación de roles y funciones, sin tener en cuenta las preferencias de las personas y falta de estrategias que generen la gestión apropiada del conocimiento entre miembros del equipo y con el cliente.

Al analizar los problemas evidenciados en el proyecto ROCK, se puede concluir que, aun siendo situaciones que plantean soluciones de maneras diferentes desde una metodología ágil o una tradicional, no es en la rigidez de las prácticas o en el uso del marco de trabajo que esté de moda en el mercado, donde el proyecto encuentra la garantía de éxito. Por el contrario, se requiere afianzar conceptos básicos como: el trabajo en equipo, la auto gestión y empoderamiento de las personas, la comunicación efectiva, la gestión del conocimiento, la disponibilidad de espacios de interacción y la generación de conversaciones informales y de escucha entre el equipo, incluyendo tanto a miembros como a usuarios finales. Es en estos contrastes donde se logra visualizar una diferencia entre el proyecto SILVER y el proyecto ROCK y es lo que determina que, así ambos celebren sesiones similares o

cuenten con equipos con la misma calidad, al final del día tengan un resultado diferente.

A continuación se presenta un análisis comparativo entre los cinco proyectos estudiados, para determinar la interacción del conocimiento en cada uno. El análisis parte del modelo de espiral enunciado por Nonaka (2000), asignando una calificación a cada una de las fases del ciclo: socialización, externalización, internalización, combinación y otorgando una ponderación para cada atributo acorde a su relevancia en el ciclo de vida de cada proyecto estudiado. Se asignan las siguientes ponderaciones a cada fase, partiendo de la influencia que cada etapa podría tener en un proyecto de desarrollo de software y en una organización de tecnología:

- Socialización: se asigna un peso del 40%
- Externalización: se asigna un peso del 20%
- Internalización: se asigna un peso del 25%
- Combinación: se asigna un peso del 15%

Tabla 8. Calificación de las fases de conocimiento por proyecto

VARIABLES	LAND	STORM	SILVER	PLATINUM	ROCK
SOCIALIZACIÓN	3,5	3,5	4	2,5	3
EXTERNALIZACIÓN	3,5	4	4	2,5	3
INTERNALIZACIÓN	4	4	4	3,5	3,5
COMBINACIÓN	3,5	3,5	3,5	4	3,5

Fuente: Elaboración propia, 2018.

A continuación, se presentan los resultados de los proyectos, partiendo de la ponderación antes mencionada.

Tabla 9. Resultados por proyecto

VARIABLES	LAND	STORM	SILVER	PLATINUM	ROCK
SOCIALIZACIÓN (40%)	1,4	1,4	1,6	1	1,2
EXTERNALIZACIÓN (20%)	0,7	0,8	0,8	0,5	0,6
INTERNALIZACIÓN (25%)	1	1	1	0,875	0,875
COMBINACIÓN (15%)	0,525	0,525	0,525	0,6	0,525
TOTAL	3,625	3,725	3,925	2,975	3,2

Fuente: Elaboración propia, 2018.

El proyecto con mayor grado de éxito, en cuanto al flujo de conocimiento, fue SILVER, dado que se implementaron prácticas ágiles con las que se alcanzó un alto nivel de comunicación, colaboración y desarrollo del conocimiento tácito, a través del contacto personal; esto a su vez, motivó la transformación a conocimiento explícito en un ambiente de relación directa entre personas, donde el crecimiento estuvo implícito en la participación activa de cada miembro del equipo.

El proyecto con menor grado de éxito, en cuanto al flujo de conocimiento, fue PLATINUM, dado que se implementaron prácticas tradicionales con una alta orientación a la documentación. Esto hizo el proyecto muy fuerte en la fase de combinación y muy débil en el ámbito tácito del conocimiento, que fue evidenciado en la falta de transferencia de conocimiento natural entre los miembros del equipo. Debido a este comportamiento, el proyecto tuvo un alto grado de desarrollo de conocimiento explícito, que fue insuficiente, al no provenir de buen insumo de conocimiento tácito.

5. Planteamiento de una propuesta de mejora en el desarrollo de proyectos de software

A partir del marco teórico estudiado y del análisis minucioso realizado a la muestra de proyectos del mercado, es posible plantear las siguientes acciones de mejora en beneficio de la industria de desarrollo de proyectos de software:

1. Es necesario que el equipo de proyecto esté definido desde el inicio y que exista una dedicación de tiempo apropiada y unas prácticas que permitan la cohesión entre sus miembros, de manera que las personas tengan la oportunidad de “pensar” y no solo de “ejecutar”. Se requiere dar todo el protagonismo al equipo, que es el dueño del conocimiento y el que, a través del trabajo colaborativo, la motivación y la dedicación, puede garantizar el éxito del proyecto.

No es conveniente que el tamaño del equipo fluctúe en pro de alcanzar los resultados, y que se tengan personas que entren y salgan del proyecto, porque se entorpece el intercambio de conocimiento y se rompe el proceso de interacción comunicacional, al no contar de manera constante con las personas que están adquiriendo dominio en las funcionalidades del sistema.

2. La organización debe renunciar a las jerarquías y al liderazgo, que es ejercido por una persona de manera particular, que se encarga de definir la estrategia, la planeación y las acciones para ejecutar el proyecto. Es necesario volcarse a un enfoque en el que gran parte de las decisiones se realizan desde el punto de vista de un equipo y en el que todos tienen la misma importancia.
3. El gerente de proyectos o *scrum* master debe balancear de manera apropiada las restricciones de tiempo, costo o alcance y gestionar la solución a cualquier impedimento que se presente, para evitar que el equipo de trabajo sea

perturbando y pueda enfocar toda su atención en la generación de un producto de valor para el cliente.

4. Siendo los requisitos y el conocimiento del cliente insumos fundamentales para lograr un sistema que genere valor agregado, es necesario explorar métodos de comunicación eficientes con los usuarios finales. Se deben definir actividades creativas que posibiliten el paso del conocimiento tácito que se encuentra en los usuarios, al conocimiento explícito que debe ser interiorizado por el equipo del proyecto.
5. Es fundamental que el equipo de trabajo y los usuarios finales compartan un mismo espacio físico de trabajo, donde la comunicación y el conocimiento fluyan de manera natural, sin estar supeditados a espacios controlados de reuniones, que se limitan al trato entre un emisor y un receptor que transportan un mensaje a través de un canal determinado.

El arte de desarrollar software exige una interacción diferente entre el equipo y el cliente, de modo que la información siempre esté disponible y se facilite el aprendizaje del equipo, al permitir que el proceso de transformación de conocimiento tácito a explícito surja de manera natural, a partir de la interacción en un mismo espacio entre miembros del equipo y el cliente.

6. Se hace imperativo que la interacción del equipo de trabajo con el cliente, permita el acceso a los usuarios finales, que son los que tienen el verdadero conocimiento de las necesidades del negocio. La comunicación con supervisores, líderes o gerentes de área puede ser considerada para temas específicos, pero no son estas personas las llamadas a entregar la información de las necesidades del cliente.

Si este tipo de prácticas se sigue presentando de manera frecuente, se seguirá encontrando en las causas de fracaso de proyectos de software, el riesgo de no

recibir la información correcta y adquirir el conocimiento apropiado para suplir la necesidad del negocio.

7. Para alcanzar el éxito esperado por toda organización que se embarca en un proyecto de software, es necesario que la persona o el equipo que asume el rol del cliente, cuente con la disponibilidad requerida para interactuar continuamente con el equipo del proyecto y que se desarrolle el mismo nivel de compromiso.

El cliente no debe ser visto ni tratado como un ente externo al proyecto que se consulta solo al inicio o cuando existe una duda específica de algún miembro del equipo; por el contrario, debe ser tratado como un miembro más del equipo y desarrollar una interacción natural en todo momento del proyecto.

8. De acuerdo al tipo de proyecto es importante evaluar la experticia de la persona que asume el rol del cliente. En proyectos de innovación que buscan realizar cambios disruptivos de procesos o funcionalidades de negocio que aún no son operadas por la organización, no sería relevante que el conocimiento del cliente tuviera un gran nivel de experticia en el negocio; pero en los casos de reemplazo de sistemas existentes o sistematización de procesos que ya operan en la institución, sería oportuno y beneficioso para el proyecto, contar con personas con experticia en la organización y no personas con poco conocimiento o experiencia que no cuenten con el detalle del negocio.
9. En entornos con alto grado de volatilidad en que los requisitos son inciertos y cambiantes, tiene mayor eficiencia y eficacia para el cliente y para el equipo del proyecto, crear el software por iteraciones por encima de firmar documentos de requisitos que contengan los detalles inciertos de las funcionalidades esperadas.

Por el contrario, se espera crear en cada iteración software que podría ser probado por un usuario final y, con estas entregas tempranas al cliente, generar los hallazgos de las mejoras a realizar y afinar las funcionalidades del sistema. De este modo se crea un ambiente de mejora constante y no se presiona al cliente y al equipo a definir un documento de requisitos que no va a ser útil en un entorno tecnológico cambiante.

10. El uso de metodologías ágiles favorece en alto grado el éxito del proyecto, dado que hace uso de prácticas fundamentadas en la socialización, que promueven la comunicación directa entre personas, la colaboración, el trabajo en equipo por encima de la definición de roles o jerarquías, la programación en parejas y esto conlleva a la generación de conocimiento tácito entre los desarrolladores y usuarios, que fluye en un proceso dinámico entre la exteriorización e interiorización.

El uso de un enfoque iterativo posibilita la retroalimentación constante del cliente y genera aprendizaje continuo entre los usuarios y desarrolladores. Dicho conocimiento está estrechamente relacionado con la persona que lo desarrolla y es compartido con otras personas, a través de contactos personales donde se comparten experiencias, vivencias, tradiciones y cultura.

11. Seguir de manera fehaciente el principio ágil de valorar más las respuestas a los cambios, a través del uso de fuertes interacciones con los usuarios y entre los desarrolladores, que el seguimiento estricto a una planificación.

De este modo se facilita el proceso de comunicación, colaboración y desarrollo del conocimiento tácito, que es compartido con otros a través del contacto personal y se promueve la transformación de conocimiento tácito a conocimiento explícito en un ambiente de relación directa entre personas, donde la evolución está implícita en el hecho de “aprender haciendo”.

12. Para lograr el desarrollo de conocimiento al interior de la organización que se traduzca en una ventaja competitiva en el mercado, es necesario que en el equipo se promuevan prácticas que faciliten la socialización del conocimiento en el ámbito tácito y que exista integración y dinámica de comunicación constante con los usuarios, que generará conocimiento a partir de la observación, imitación o práctica de parte de los miembros del equipo. De este modo, se hace posible la generación de flujo de conocimiento hacia el conocimiento explícito, en medio de los procesos de exteriorización y combinación.

13. Es un error enfocarse únicamente en la exteriorización de conocimiento y desconocer que el proceso adecuado debe pasar por: recopilación de conocimiento explícito, personalización de conocimiento explícito, conversión a conocimiento tácito, aplicación del conocimiento adquirido en la ejecución de proyectos, aprendizaje del proyecto en forma de conocimiento tácito y conversión del aprendizaje en conocimiento explícito.

Si la organización se apropia de este proceso y facilita el uso de metodologías y prácticas que lo favorezcan, alcanzaría una ventaja competitiva a través de la transformación de información y su apropiación en conocimiento organizacional, que la conduciría en el desarrollo de escenarios apropiados para la creación e innovación en el desarrollo de sus proyectos.

6. Conclusiones

1. La generación de conocimiento es el principal componente que origina la dinámica de innovación organizacional y es el más grande propulsor del desarrollo de la ventaja competitiva que requiere la industria de software, en un contexto socioeconómico y tecnológico en el que se hace imperativo contar con empresas flexibles, innovadoras, capaces de aprender de forma continua, orientadas al aprendizaje y capaces de generar conocimiento para crecer y adaptarse. La gestión de lo que la empresa sabe, cómo emplea lo que sabe, su capacidad de aprender y de desarrollar en sus individuos la sensibilidad e inteligencia para socializar con otros, apalanca el desarrollo creativo y la capacidad de invención de nuevos productos, lo cual, a su vez, permite generar esfuerzos diferenciales para superar los competidores en ámbitos como: calidad, servicio al cliente, estrategia organizacional y liderazgo de la organización para transformarse.

Podríamos considerar entonces, que las acciones para mejorar las posibilidades de éxito y disminuir los índices de fracaso en proyectos de desarrollo de software, deberían estar orientadas en el diseño de estrategias que promuevan la creación de conocimiento a niveles individual, grupal y organizacional y que permitan a través de prácticas apropiadas en la ejecución de los proyectos, captar conocimiento, documentarlo, recuperarlo, reutilizarlo, así como crearlo, transferirlo e intercambiarlo.

2. Partiendo de lo expuesto por Nonaka y Takeuchi, reconocemos el conocimiento tácito como aquel que está vinculado con la experiencia del individuo y que incluye las creencias personales, la perspectiva propia y el propio sistema de valores, y reconocemos el conocimiento explícito como aquel que puede transmitirse utilizando el lenguaje formal y sistemático. El flujo entre ambos tipos de conocimiento se da a través de un proceso denominado espiral del

conocimiento, que implica la interacción entre el conocimiento tácito y explícito y se mueve a través de la recopilación de conocimiento explícito, personalización de conocimiento explícito, conversión a conocimiento tácito, aplicación del conocimiento adquirido a la ejecución de proyectos y aprendizaje del proyecto en la forma de conocimiento tácito, teniendo especial importancia en este modelo la conversión de tácito en explícito. En este sentido, las metodologías ágiles tienen una alta orientación a la comunicación cara a cara entre personas, a la colaboración, al desarrollo de actividades conjuntas, lo cual posibilita un alto grado de socialización y permite compartir conocimiento tácito entre desarrolladores y usuarios. El desarrollo por iteraciones que permite la retroalimentación temprana de parte del cliente, las reuniones periódicas, las sesiones de planeación conjunta, el ambiente colaborativo posibilitado por el espacio físico de trabajo, genera un entorno de aprendizaje continuo entre el equipo y los usuarios, lo cual facilita compartir conocimiento tácito y promueve una cultura de confianza y motivación en la que se posibilita, a través de herramientas, la conversión en conocimiento explícito.

3. En la búsqueda de factores que posibiliten el éxito de los proyectos es indispensable considerar que en la actualidad, tanto los proyectos desarrollados bajo metodologías ágiles como los proyectos desarrollados bajo metodologías tradicionales, han fallado en el involucramiento apropiado de los usuarios y la comunicación efectiva y constante de estos con el equipo de trabajo. Se hace necesario que el proyecto cuente con la participación de los usuarios finales como miembros activos del equipo de trabajo y que se definan actividades creativas que posibiliten el paso del conocimiento tácito, que se encuentra en los usuarios, al conocimiento explícito que debe ser interiorizado por el equipo del proyecto.

Buscando que la información siempre esté disponible y que se facilite el aprendizaje, es fundamental que el equipo de trabajo y los usuarios finales

compartan un mismo espacio físico de trabajo, donde la comunicación y el conocimiento fluyan de manera natural, sin estar supeditados a espacios controlados de reuniones y donde el proceso de transformación de conocimiento tácito a explícito surja de manera natural, a partir del trato personal constante y de la interacción constante entre miembros del equipo y el cliente.

4. Gran parte de los reprocesos y retrasos detectados en los proyectos de desarrollo de software, son causados porque el equipo de proyecto no está definido desde el inicio, no existe una dedicación apropiada para las actividades del proyecto y la persona debe convivir con sus actividades diarias y la asignación del proyecto; se presentan fluctuaciones constantes entre el equipo constituido y es usual que se tengan nuevos miembros o que los que fueron definidos entren y salgan constantemente del proyecto. Esto genera inconvenientes en el intercambio de conocimiento, dado que se rompe el proceso de interacción comunicacional, al no contar de manera constante con las personas que están adquiriendo dominio en las funcionalidades del sistema y que se están nutriendo de conocimiento tácito en la interacción con los usuarios finales y los demás miembros del equipo.

Referencias

Aguilera, L. M. A. (2017). *LA GESTIÓN DEL CONOCIMIENTO ORGANIZACIONAL*. Retomado el 13 de abril de 2018 de: <https://www.researchgate.net/publication/312155359>

Alliance, S. (2016). *The State of Scrum Report 2017 Edition*. Scrum Alliance. 31 pages. Retomado el 13 de abril de 2018 de: https://www.scrumalliance.org/scrum/media/ScrumAllianceMedia/FilesandPDFs/StateofScrum/StateOfScrum_2016_FINAL.pdf

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M. & Kern, J. (2001). *Manifesto for agile software development*. Retomado el 17 de mayo de 2018 de: <http://www.agilealliance.org>.

Briones, C. L. T., & Dávila, R. S. R. (2017). La Gestión del Conocimiento basado en la Teoría de Nonaka y Takeuchi. *INNOVA Research Journal*, 2(4), 30-37.

Casilimas, C. A. S. (1996). *Investigación cualitativa*. Bogotá: ICFES/ASCUN.

Charvat, J. (2003). *Project management methodologies: selecting, implementing, and supporting methodologies and processes for projects*. Wiley, New York (NY): John Wiley & Sons.

Computer Standards & Interfaces (2017). *The International Journal on the Development and Application of Standards for Computers, Software Quality, Data Communications, Interfaces and Measurement*. Editorial Board, Editor-in-Chief: Rory O'Connor

Chin, G. (2004). *Agile project management*. New York: AMACOM.

Chin, G. (2006). *Agile project management: how to succeed in the face of changing project requirements*. AMACOM Div American Mgmt Assn.

Galvis Lista, E. A., Sánchez-Torres, J. M., & González-Zabala, M. P. (2016). Hacia un modelo de referencia de procesos de gestión del conocimiento para organizaciones desarrolladoras de software: validación por expertos. *AD-minister*, (26).

Gartner Research (2016). *The end of the waterfall as we know it*, August, 20. Published ID: G00291841. Retomado 17 de mayo de 2018 de: <https://www.gartner.com/doc/3188962/end-waterfall-know-it>

Guide, A. (2013). Project management body of knowledge (pmbok® guide). In *Project Management Institute*. 3rd ed., Project Management Institute, Newtown Square, PA.

Guido, J., & Clements, J. (2003). *Administración exitosa de Proyectos*. 2ª Edición. Ciudad de México: Thomson Learning.

Hass, K. B. (2007). The blending of traditional and agile project management. *PM world today*, 9(5), 1-8.

Hernández Sampieri R., Fernández Collado, C., & Baptista Lucio, P. (2014). *Metodología de la investigación*. México DF: Editorial Mc Graw Hill.

Hodgson, Matthew (2012). *Is a self-organising agile team a self-managing agile team?* Posted by MATTHEW HODGSON on APRIL 4, 2018. Retomado el 13 de abril de 2018 de: <https://zenexmachina.wordpress.com/2018/04/04/is-a-self-organising-agile-team-a-self-managing-agile-team/>

Hugo, A. M. H., Edgar, O. M., & Cuauhtémoc, L. O. (2014). Estimación y control de costos en métodos ágiles para desarrollo de software: un caso de estudio. *Ingeniería, investigación y tecnología*, 15(3), 403-418.

Jack Gido (2012). *Administración exitosa de proyectos*. Quinta edición. Universidad Eafit, Medellín, Colombia Cengage Learning Editores.

Kaddoumi, T., & Watfa, M. (2016, August). A proposed agile enterprise architecture framework. In *Innovative Computing Technology (INTECH), 2016 Sixth International Conference on* (pp. 52-57). IEEE.

Larson, E. W., & Gray, C. F. (2011). *Project management: The managerial process*. Maidenhead: McGraw-Hill.

Meta Group (2015). *Waterfall vs. Agile – a knowledge problem, not a requirements problem*. Posted by MATTHEW HODGSON on JULY 19, 2015. Retomado el 25 de abril de 2018 de: <https://zenexmachina.wordpress.com/2012/07/19/waterfall-vs-agile-a-knowledge-problem-not-a-requirements-problem/>

Monavarian, A., & Asgari, N. (2009). *Organizations in the age of industry, information and knowledge*. Tehran: Tehran University Press (In Persian)

Mulcahy, R. (2013). *Preparación para el examen PMP* (Octava ed.). New York: RMC

Publications, Inc.

Neil, D. (2006). Proyectos Informáticos: Fracasos y Lecciones Aprendidas. *Revista de Derecho y Tecnologías de la Información*, 4, 1-11.

Nonaka, I. (1988). Toward middle-up-down management: accelerating information creation. *MIT Sloan Management Review*.

Nonaka, I., & Takeuchi, H. I. R. O. T. A. K. A. (1995). *The knowledge-creating company: how Japanese companies create the dynamics of innovation*. Oxford: Oxford University Press.

Nonaka, I. (2000). A Dynamic Theory of Organizational Knowledge Creation, In Knowledge, groupware and the internet” Organization. *Science*, 5(1), pp. 14-37.

QRP International (2014). Retrieved from *What-is-agile*. Retrieved 2017 QRP S.R.L. Retomado el 13 de abril de 2018 de: <http://www.qrpinternational.es/index/agile/what-is-agile>

Sahota, M. (2012). *Una guía de supervivencia a la adopción y transformación ágil : trabajando con cultura organizacional*. Copyright © 2012, P 1- 85. survival guide: Working with organizational culture. s.l.:InfoQ.

Schön, E. M., Thomaschewski, J., & Escalona, M. J. (2017). Agile Requirements Engineering: A systematic literature review. *Computer Standards & Interfaces*, 49, 79-91.

Standish Group (2009). Chaos summary 2009. *Online report*. Accessed June, 20 – 2009. Standish Group, Boston, MA, Estados Unidos de América. Retomado el 13 de abril de 2018 de: <https://www.standishgroup.com/>

Standish Group (2015). *Chaos Report-Q&A with Jennifer Lynch* (2015).

Retomado el 13 de abril de 2018 de: <https://www.standishgroup.com/>

Strode, E, Diane (2012). *A theory of coordination in agile software development projects*. A thesis submitted to the Victoria University of Wellington in fulfillment of the requirements for the degree of Doctor of Philosophy. Florida, Estados Unidos.

Sugiyama, K., & Meyer, B. (2008). Knowledge process analysis: Framework and experience. *Journal of Systems Science and Systems Engineering*, 17(1), 86-108.

Takeuchi, H. & Ikujiro Nonaka (1986). The New Product Development Game. *Harvard Business Review*.

The Standish Group (2014). *Law of the Five Deadly sins*. Copyright © 2014 International, Inc. 2014 - 06. Retomado el 25 de abril de 2018 de: [https://cdn.infoq.com/statics_s1_20161011-0321/resource/articles/standish-chaos-2015/en/resources/5Deadly sins_2014-5.pdf](https://cdn.infoq.com/statics_s1_20161011-0321/resource/articles/standish-chaos-2015/en/resources/5Deadly%20sins_2014-5.pdf)

Tobergte, D. R., & Curtis, S. (2013). No Title No Title. *Journal of Chemical Information and Modeling*, 53(9), 1689-1699.

Viedma, J. M. (2000). La gestión del conocimiento y del capital intelectual. *Nueva Empresa*. 454 (22), 99-105.

Walczak, W., & Kuchta, D. (2013). Risks characteristic to Agile project management methodologies and responses to them. *Operations Research and*

Decisions, 23.

Wrike (2018). *Wrike Operational Excellence Report*. Retomado el 18 de abril de 2018 de: <https://www.wrike.com/library/ebooks/wrike-operational-excellence-report-2018/>