

# INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

---

RECONOCIMIENTO DE VALIDEZ OFICIAL, ACUERDO SEP. NO. 15018  
PUBLICADO EN EL DIARIO OFICIAL DE LA FEDERACIÓN  
EL 29 DE NOVIEMBRE DE 1976.

DEPARTAMENTO DE  
**MAESTRÍA EN INFORMÁTICA APLICADA**



## ITESO

### **GESTIÓN DE PROYECTOS DE DESARROLLO DE SOFTWARE EN SCRUM CON FUNDAMENTOS EN EL PMBOK.**

PROYECTO PROFESIONALIZANTE DE DESARROLLO PARA OBTENER EL  
**GRADO DE MAESTRO EN INFORMÁTICA APLICADA**

PRESENTA:

**Moisés Cano Valdés**

**Asesora de Trabajo de Obtención de Grado**  
**Mtra. Sara Josefa Ortiz Cantú**

# Dedicatoria

A mis padres que me dieron las herramientas y los valores para culminar cualquier proyecto que me proponga.

A mi esposa e hijo que he conocido en este camino y son mi motor para seguir avanzando y ser un mejor ser humano y un mejor profesionista.

Mi esposa que tanta paciencia y apoyo me brindó durante todo este tiempo para poder culminar uno de nuestros sueños y metas.

Mi hijo que es todo para mí, mi legado y la actual causa de buscar mejorar en todos y cada uno de los aspectos de mi vida.

A Dios que siempre me ha cobijado con su manto y en los momentos más difíciles ha escuchado mis suplicas y oraciones.

A mis hermanos que me han apoyado y animado en todos los proyectos de mi vida.

A mis profesores y compañeros que me apoyaron y me guiaron para lograr obtener este título tan significativo, importante y arduo en mi carrera profesional.

Al ITESO, esta grandiosa institución que me brindó la oportunidad de adoptar una segunda alma mater y de la cual siempre estaré orgulloso y representaré con honor, integridad, dignidad y honestidad.

A todos ustedes gracias infinitas...

## Contenido

1. Resumen .....	5
2. Introducción .....	6
3. Planteamiento del problema.....	7
4. Hipótesis .....	7
4.1 Tecnología actual: .....	7
4.2 Tecnología buscada: .....	8
4.3 Desarrollo de Hipótesis:.....	8
5. Objetivos .....	8
5.1 Objetivos Generales .....	8
5.2 Objetivos Particulares.....	9
6. Justificación .....	9
7. Marco teórico. ....	10
7.1 Antecedentes de los modelos de gestión de proyectos .....	10
7.2 Administración de proyectos.....	13
7.3 Dirección de proyectos .....	16
7.4 Definición de las áreas de conocimiento del PMBOK. ....	18
7.5 Metodologías de desarrollo de proyectos de Software.....	20
7.5.1 Metodología de Cascada .....	20
7.5.2 Metodología del modelo en espiral.....	22
7.5.3 Metodologías Ágiles .....	23
7.6 SCRUM .....	27
7.6.1 Manejo de requerimientos en SCRUM .....	29
7.6.2 Correlación de áreas del conocimiento con SCRUM.....	31
7. 7 Análisis final .....	40
8. Estrategia metodológica.....	42
8. 1 Hoja de concepto.....	43
8.1.1 Definición mercadológica .....	43
8.1.2 Mercado(s) meta/ características .....	43
8.1.3 Necesidades básicas a satisfacer .....	44
8.1.4 Variables de competitividad .....	44
8.1.5 Características funcionales .....	44
8.1.6 Hábitos de consumo actuales .....	44
8.1.7 Vida funcional prevista o ciclo de vida.....	45
8.1.8 Restricciones .....	45

8.1.9 Productos en competencia.....	46
8.1.10 Aspectos tecnológicos.....	46
8.1.11 Ángulos de diferenciación .....	46
8.1.12 Productos complementarios .....	47
8.1.13 Propuesta de valor .....	47
8.1.14 Costos.....	47
8.2 Enunciado del alcance.....	47
8.2.1 Justificación del proyecto: .....	47
8.2.2 Alcance del proyecto:.....	48
8.2.3 Metodología del proyecto .....	48
8.2.4 Productos a entregar.....	49
8.2.5 Delimitación .....	49
8.3 Desarrollo de Propuesta .....	49
8.3.1 Iniciación:.....	50
8.3.2 Planeación.....	56
8.3.3 Ejecución.....	61
8.3.4 Monitoreo y control.....	67
8.3.5 Cierre .....	72
8.3.6 Marco referencial de Hitos clave: .....	77
8.3.7 Herramientas de gestión de proyectos y sus características.....	78
8.4 Conclusiones:.....	88
11. Bibliografía .....	91
Glosario .....	92
Anexos.....	92
Anexo 1 Cuestionario sobre el uso de SCRUM.....	92
Anexo 2 Planilla “Plan de Proyecto” .....	94
Anexo 3. Fases del Proyecto.....	97
Anexo 4 Esquemas de metodología PMP y SCRUM.....	100

## 1. Resumen

En los inicios de los servicios tecnológicos y surgimiento de software como herramienta tecnológica básica y complemento del hardware, los expertos en diversas materias han buscado una forma de gestionar y administrar los proyectos y servicios relacionados al software y a todo el campo de las tecnologías de información en general.

Hoy en día las metodologías de gestión y administración de desarrollo de software son cada vez más tendientes hacia los procesos ágiles, rápido, baratos y adaptativos. Muchas organizaciones han migrado de metodología de administración de proyectos, administración de recursos como hardware y administración de sistemas como software hacia dicha tendencia y por consiguiente han desarrollado nuevos métodos de gestión algunos muy radicales mientras que otros más tradicionalistas.

Probablemente se piense que los métodos ágiles, con el afán de ser rápidos y cambiantes, sacrifican calidad en los entregables; sin embargo, en la actualidad muchos de estos viejos tabúes han sido resueltos de forma exitosa.

A finales de los años 90 con el surgimiento de las metodologías ágiles y sus entramados, muchas compañías en la industria de TI (Tecnologías de la Información) surgieron con bases sólidas y con una idea clara en sus procesos de desarrollo de software, este surgimiento generó una gran polémica y cierta confusión para los gerentes de proyectos, ya que por un lado tenían el método tradicional de gestionar y controlar sus proyectos por medio de fases, siendo sucesivo y asegurando, de cierta manera, la calidad de los entregables; por otro lado un nuevo método adaptativo, dinámico y cumpliendo los criterios de aceptabilidad básicos.

La industria hoy por hoy enfrenta diversas complicaciones para la implementación de sus métodos de trabajo, ya que desafían dos escenarios: El primer escenario es sobre los proyectos de corta envergadura, que, si bien, requieren la misma atención y calidad que los demás, se gestionan por medio de metodologías ágiles por la falta de conocimiento y alcance de las mismas. En el segundo escenario, están los proyectos medianos o de gran tamaño que normalmente se manejan bajo un esquema de administración en su marco teórico del PMBOK y desarrollo del producto en cascada.

Sin embargo, por la demanda de tiempos y esfuerzos, la industria ha modificado ambas metodologías, llegando a crear sistemas híbridos que muchas veces cumplen con los tiempos establecidos del proyecto, pero no con la calidad del producto o viceversa. Esto no quiere decir que las propuestas alternativas con contenido híbrido sean malas, al contrario; es el preámbulo de la creación de una mejor propuesta metodológica adaptada a las necesidades de la organización. No obstante, este tipo de propuestas muchas veces se hacen también por la falta de conocimiento integral y las bondades de métodos ágiles los cuales se crearon con la idea de adaptarse a cualquier producto de software.

## 2. Introducción

El contenido de este trabajo de investigación pretende mostrar las características principales de las metodologías de gestión de proyectos, catalogándolas y describiéndolas en dos principales tendencias: los métodos ágiles y los métodos tradicionales; a través de experiencias laborales y opiniones de autores expertos en el tema, se busca hacer un análisis minucioso de ambas metodologías con el supuesto que a pesar que se tienen algunas deficiencias con respecto a las metodologías tradicionales como por ejemplo de cascada, interacción con prototipos, modelo V, modelo en espiral, por mencionar algunos; también cuentan con ciertas cualidades esenciales en la gestión de proyectos. Sin embargo, con base a las tendencias en la industria de TI, los métodos ágiles como *SCRUM*, *DSDM (Dynamic Systems Development Method)* y *Extream Programming*, entre otros, tienden a dominar y adquirir nuevas utilidades en diversas áreas, por esta razón la presente propuesta, busca aportar una idea de los posibles escenarios.

En los últimos años las metodologías ágiles han irrumpido en la industria con fuerza en un intento de despojar la gestión de desarrollo de software con bases metodológicas convencionales, por esta razón muchas organizaciones son ahora las que se suman a esta nueva propuesta, pero ¿qué hacen novedosas estas metodologías? Algunos estudios realizados recientemente indican que la productividad y calidad del software aumenta utilizando los principios y valores que rigen dichos métodos. No obstante, la mayoría de estos estudios se limitan a narrar observaciones cualitativas. Algunos utilizan datos empíricos con el fin de apoyar sus conclusiones; los resultados arrojados se muestran tan dispersos como un 337% en la productividad y una disminución del 44% en el tiempo. (Dalcher & Thorbergsson, 2005). Por este motivo las organizaciones solicitan un reporte inicial y estudios sobre las metodologías ágiles para así poder probar y constatar sus beneficios con la alta dirección.

Es lógico pensar que uno de los factores que han influido en los resultados descritos con anterioridad, es el rápido cambio en diversos sectores industriales, la globalización como punto fundamental de intercambio de bienes y servicios entre naciones, potencias tecnológicas como India, China y Singapur abaratan la mano de obra global y crean un ambiente dinámico de cambios rápidos y efectivos, muchas organizaciones buscan gestionar y controlar sus productos dentro del mercado global, que por ende se vuelve cada vez más competitivo y voraz. Como consecuencia de esta competitividad híper rítmica, la industria tecnológica opta cada vez más y más, como se menciona anteriormente, por adoptar metodologías de gestión, control y desarrollo ágiles que ayuden a competir y adaptarse de forma integral.

A lo largo del desarrollo de este trabajo se buscará sustentar y explicar desde un punto de vista teórico-práctico la evolución de un producto de software en ambos escenarios; un enfoque tradicional de desarrollo de cascada y ciclo de vida del software, así como un desarrollo ágil y sus interacciones adaptativas y enfocadas a la satisfacción del cliente, se pretende tratar ciertos puntos básicos en la configuración de ambas metodologías con el fin de aterrizar los conceptos.

Algunos de los factores claves a evaluar son características fundamentales del producto y su evolución en el ciclo de vida, criterios de aceptabilidad y calidad, dinamismo en el desarrollo del producto, además la estimación del esfuerzo de adopción de ambas tendencias metodológicas, dicho esfuerzo se estima cuantitativa y cualitativamente desde un punto de vista de capital humano y capital de trabajo.

### 3. Planteamiento del problema

En la administración de proyectos de software, muchas veces los responsables de la gestión plantean un escenario bipartito para completar en tiempo y forma los requisitos solicitado por el cliente. Con base en la experiencia de algunos especialistas en la materia, se tiene la idea que algunos proyectos, por la complejidad del producto a entregar, no cumplen los requisitos mínimos para ser gestionados con metodologías ágiles y por el contrario productos de menor complejidad, no deben ser administrados en un contexto progresivo de cascada. Esta idea no es del todo cierta pues muchas veces la oficina de administración de proyectos desconoce el alcance de las metodologías ágiles por ser relativamente nuevas.

Las metodologías ágiles fueron creadas aproximadamente hace 20 años y a lo largo de ese tiempo han demostrado gran utilidad y productividad cuando se usan en proyectos medianos y pequeños, que tienen como característica en común los cambios constantes de requerimientos. Sin embargo, en proyectos de dimensiones mayores los administradores de los mismos, no confían aún en este tipo de metodologías, ya sea por falta de difusión o reconocimiento; lo que conlleva a seguir usando metodologías tradicionales o también conocidas como metodologías pesadas, causando que los proyectos presenten un sobre gasto de tiempo, dinero y esfuerzo, por consiguiente una baja calidad en los entregables.

Principalmente *Agile* es el nombre que se le da a todo el conjunto de métodos que se encargan de desarrollar productos y servicios principalmente de software con bases de gestión “ligeras”. SCRUM se puede considerar como un componente o un marco de referencia de tendencia ágil, es un conjunto de buenas prácticas para la gestión y administración de proyectos. En otras palabras, SCRUM es una de las diversas formas de poner en práctica los principios del manifiesto ágil, pilar de dicha metodología.

La cambiante industria de tecnologías de información evoluciona rápidamente creando un entorno turbulento, el cual genera cierto grado de incertidumbre en la gestión de proyectos tecnológicos; por esta razón, la adaptabilidad de las metodologías utilizadas en la construcción de sistemas de Software hasta el momento, está enfocada a las metodologías convencionales. No obstante, surge la pregunta ¿se adaptan bien las metodologías convencionales a las actuales características de dinamismo y variabilidad del mercado? ¿Necesitamos nuevas alternativas que se ajusten de un modo más fehaciente a dichas necesidades? Según Kent Beck, uno de los padres del modelo ágil, “cada cosa en software cambia. Los requisitos cambian, el diseño cambia, el negocio cambia, la tecnología cambia, el equipo cambia y los miembros del equipo cambian. El problema no es el cambio, porque el cambio, inevitablemente, va a ocurrir. El problema, realmente, es nuestra inhabilidad para hacer frente a los cambios”. (Kent, Xtream Programming, p.56) por tal motivo, parte del mercado de software está teniendo variaciones, por lo que las organizaciones se obligan cada vez más a abordar estos cambios con nuevas alternativas que les permitan adaptarse mejor a la actual situación.

### 4. Hipótesis

#### 4.1 Tecnología actual:

Desarrollo de proyectos de software en cascada basado principalmente en un enfoque SDLC (*Software Develop Live Cycle*), algunos proyectos bancarios en México se administran de forma tradicional bajo las normas y principios del PMBOK, la figura metodológica muestra tendencias de gestión tradicional como: Documentación específica por fases, enunciados del alcance del

proyecto, documento de especificaciones exactas, documento de análisis, acta constitutiva del proyecto, creación del EDT (Estructura de desglose del trabajo), concatenado de los interesados del proyecto, entre otras.

#### 4.2 Tecnología buscada:

La tecnología que se busca es la migración a SCRUM como base metodológica y marco referencial de gestión de proyectos; dicha migración se pretende llevar a cabo por medio de fases que cumplan ciertas características particulares como son: contar solamente con dos documentos de control del proyecto (1. Los criterios de aceptabilidad y 2. Las historias de usuarios). También se busca establecer cortos periodos de tiempo en el desarrollo del producto, con el fin de presentarle al usuario final eventualmente una pequeña pero valiosa parte del producto final; otra característica que se busca es tener reuniones periódicas para el refinamiento de dicho producto; reuniones con el denominado equipo SCRUM con la finalidad de remarcar las fortalezas y debilidades de cada miembro.

La parte de la comunicación clara y oportuna es fundamental como parte de otro elemento a considerar, puesto que, por medio de llamadas diarias de estatus, el gestor del proyecto denominado Maestro SCRUM da seguimiento a la evolución del trabajo y facilita el mismo a los miembros de su equipo.

Con todo esto, el objetivo primordial desde la perspectiva empresarial, es buscar generar un impacto económico positivo hacia el dueño o cliente final del producto a desarrollar, a través de optimizar tiempos, maximizar recursos materiales y humanos, estos últimos auto-gestionados con objetivos específicos y características distintas, pero con una meta en común.

#### 4.3 Desarrollo de Hipótesis:

La migración de proyectos de diversos tamaños y características, inicialmente formados con principios lineales-secuenciales bajo un esquema metodológico “pesado”, que para este fin se denomina “de cascada”; con rumbo hacia un marco de referencia “ágil” e incremental denominado SCRUM. Bajo el supuesto que, con dicha migración el usuario final del producto de software a desarrollar estará ahorrando tiempo y esfuerzo.

Otro factor importante que se pretende desarrollar en este trabajo es sobre la postura de las bondades en las metodologías ágiles con respecto a las tradicionales, con lo anterior se remarcarán ciertas características de ambas tendencias y una propuesta de solución sobre la migración de un proyecto de tamaño medio con gran impacto que actualmente se desarrolla con el método de cascada y se pretende migrar a SCRUM.

## 5. Objetivos

### 5.1 Objetivos Generales

Aplicar la metodología SCRUM en la administración y gestión de proyectos de desarrollo de software.

Este aplicativo se hará bajo un esquema comparativo en términos metodológicos, demostrando los beneficios de adoptar SCRUM como marco de referencia, así como sustentar el supuesto

que cualquier proyecto, no importando sus características, puede ser gestionado bajo este modelo ahorrando tiempo, esfuerzo y recursos.

## 5.2 Objetivos Particulares

Número	Descripción del objetivo	Beneficios a obtener
1	Ejemplificar la migración metodológica de un proyecto real con lineamientos tradicionales hacia SCRUM.	Exponer los beneficios de la adopción de dicha metodología con un enfoque tangible y sustentable.
2	Remarcar las ventajas y desventajas de la utilización de SCRUM como metodología de gestión de proyectos.	Conocer las principales razones de adoptar SCRUM entre el resto de las metodologías ágiles.
3	Crear un marco referencial para aquellos administradores de proyectos que busquen una referencia puntual de la utilización de SCRUM en la gestión.  Detallar los puntos principales del ciclo de vida en el desarrollo de un producto de software.	Brindar un panorama específico de la utilización de SCRUM como herramienta de gestión de proyectos.

## 6. Justificación

Una de las principales limitantes sobre la utilización del marco del PMP, es lo que describe Swenson, (2002, p.87), "El PMBOK es una metodología rígida y poco dinámica, no apta para entornos complejos y susceptibles a cambios, como son la mayoría hoy en día".

Sin embargo, desde un punto de vista teórico el PMBOK, es un marco general de administración de proyectos y es aplicable a cualquier tipo de proyecto por lo cual no prescribe ninguna metodología concreta, en otras palabras, la guía del PMBOK no es una metodología; es un marco de referencia, un compendio de buenas prácticas, una guía sobre aplicación de técnicas y herramientas de administración de proyectos, un convenio sobre nomenclatura. El PMBOK elige cuáles de los procesos propuestos aplican al proyecto a desarrollar, y pueden ser combinados con cualquier metodología de gestión.

SCRUM es una metodología específica, definida paso por paso, para desarrollo de productos, sobre todo en el mundo de Software. Define desarrollo de producto, y es tan amplio como el administrador de proyectos decida. Por lo anterior, a continuación se describen un par de características de cada uno.

- Guía del PMBOK ofrece una visión global y generalista de varias facetas de la gestión de proyectos, que sin duda resulta útil como referencia, sin ser limitadora, para cualquier entorno, practique la metodología que practique.
- SCRUM propone enfoques dinámicos e invita a un cambio de la cultura de trabajo de las empresas, incluso si no se sigue esta metodología ágil a pie de la letra, usando sólo algunos de sus elementos. (Abrahamsson, 2002).

De acuerdo con el PMBOK un proyecto se denomina como un esfuerzo temporal para obtener un resultado único el cual tiene cierto número de entradas y de salidas que sirven para lograr dicho resultado. (PMBOK, 6ta edición, 2017).

Dentro de las entradas, el control de cambios es fundamental y se debe gestionar bajo el supuesto que los cambios son excepcionales por el alto costo en el impacto del proyecto; ya que al modificar el alcance, por ende se modificará el tiempo y el costo del mismo. Debido principalmente a esta característica específica del PMBOK, los gerentes de proyecto enfrentan grandes conflictos al momento de modificar el alcance del mismo, lo cual de cierta manera genera inconformidad por parte del dueño del producto (cliente o usuario final); la propuesta de las metodologías ágiles es tener un canal de comunicación abierto de forma constante con el representante del cliente, con el objetivo de realizar cambios en tiempo y forma para así adaptar el producto a entregar de acuerdo a las necesidades específicas del cliente, por lo anterior el tiempo que se invierta en el desarrollo del proyecto así como los recursos, será estrictamente lo necesario para obtener una satisfacción real de dicho cliente.

SCRUM es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en SCRUM son el *SCRUM Master*, que mantiene los procesos y trabaja de forma similar al director de proyecto, el *ProductOwner*, que representa a los *stakeholders* (interesados externos o internos), y el equipo que incluye a los desarrolladores; dichos roles no son limitativos ya que se pueden rolar de acuerdo a las necesidades del proyecto. SCRUM está catalogada como una metodología de desarrollo ágil con ciclos secuencias con solapamiento. (Abrahamsson, 2002)

SCRUM permite la creación de equipos auto organizado impulsando la co-localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

Un principio clave de SCRUM es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren o necesitan, y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, SCRUM adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

Existen varias implementaciones de sistemas para gestionar el proceso de SCRUM, que van desde notas amarillas "post-it" y pizarras hasta paquetes de Software. Una de las mayores ventajas de SCRUM es que es muy fácil de aprender, y requiere muy poco esfuerzo para comen-zarse a utilizar.

## 7. Marco teórico.

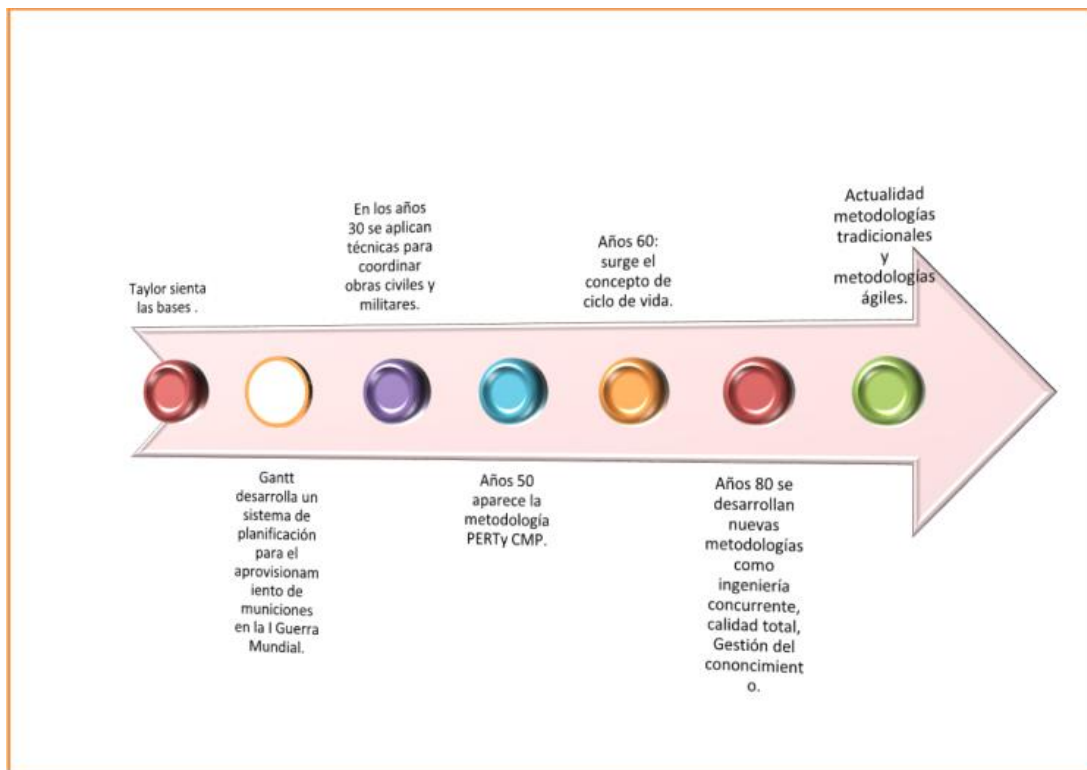
### 7.1 Antecedentes de los modelos de gestión de proyectos

En conjunto, la utilización de todos estos elementos (entre otros), se denomina gestión de proyectos; esta técnica surgió en los años 50 dentro del ámbito militar; debido a que los proyectos que se desarrollaban eran de gran magnitud y requerían de recursos materiales y humanos especialistas en diversas áreas, por lo tanto, la coordinación de estos grupos era tan solo un paso natural.

Fue a partir de esos momentos cuando comienzan a surgir diversas herramientas, técnicas y métodos que facilitarían la gestión de un proyecto, algunos ejemplos son: El ciclo de vida, descomposición en tareas, realización de gráficos, entre otros. (Palacios, 1998, p67)

En la figura siguiente (Fig. 1); se muestra una breve descripción de la evolución en la administración de proyectos, cabe señalar que todos estos acontecimientos, se desarrollaron debido al dinamismo y tendencias globalizadas, así como a la necesidad del ser humano de organizar y sistematizar los procesos no importando las dimensiones de estos.

**Figura 1. Evolución Histórica de la Gestión de Proyectos.**



Fuente: Trigas, 2016, p55

La administración de proyectos tiene sus orígenes a finales del siglo XIX. En este artículo se explica cómo las metodologías científicas, sociales y empresariales desarrolladas a lo largo de más de un siglo influyeron en la teoría moderna de la administración de proyectos.

La administración de proyectos, en su forma moderna, comenzó a afianzarse hace solo unas décadas. A partir de principios de los años sesenta del siglo pasado, las empresas y otras organizaciones comenzaron a observar las ventajas de organizar el trabajo en forma de proyectos. Esta perspectiva de organización centrada en proyectos evolucionó aún más cuando las organizaciones empezaron a entender la necesidad fundamental de que sus empleados se comuniquen y colaboren entre sí al tiempo que integran su trabajo en diferentes departamentos, profesiones y, en algunos casos, industrias completas.

En la actualidad, los preceptos básicos de administración de proyectos están representados por el triángulo del proyecto, un símbolo que popularizó Harold Kerzner, Ver Figura 2.

**Figura 2: La triple restricción en la gestión de proyectos.**



Fuente: Kerzner, 1979, p 86

Podemos remontarnos aún más atrás, a la segunda mitad del siglo XIX, cuando el mundo empresarial comenzaba a ser cada vez más complejo, para observar cómo la administración de proyectos evolucionó a partir de principios básicos de administración. Los proyectos gubernamentales a gran escala fueron el impulso para tomar decisiones importantes que se convirtieron en la base de la metodología de la administración de proyectos. Por ejemplo, en los Estados Unidos el primer proyecto verdaderamente grande del gobierno fue el ferrocarril transcontinental, cuya construcción comenzó en los años sesenta del siglo XIX. De repente, los líderes empresariales debieron enfrentarse a la enorme tarea de organizar el trabajo manual de miles de trabajadores, además del procesamiento y montaje de las materias primas en cantidades sin precedentes.

Entre finales del siglo XIX y principios del XX, Frederick Taylor (1856–1915) comenzó a realizar estudios detallados del trabajo. Aplicó el razonamiento científico y demostró que el trabajo puede analizarse y mejorarse si se centra en las partes fundamentales. Puso en práctica sus ideas en las tareas realizadas en las fundiciones de acero, como recoger arena con la pala y levantar, y trasladar piezas. Anteriormente, la única manera de mejorar la productividad era exigir a los trabajadores más esfuerzo y más horas de trabajo. Taylor presentó el concepto de trabajar con más eficiencia en lugar de más esfuerzo y tiempo. La inscripción en la tumba de Taylor en Filadelfia avala su lugar en la historia de la administración: "El padre de la administración científica". (Thamhain, 1986, p122)

Taylor, Henry Gantt (1861 – 1919) estudió detalladamente el orden de las operaciones de trabajo. Sus estudios de administración enviaron aportaron bastante valor durante la primera guerra mundial. Su diagrama de Gantt con barras de tareas y marcadores de hitos, la secuencia y la duración de todas las tareas en un proceso de esquemas. Dicho diagrama resultó ser una herramienta analítica eficaz para los administradores, la cual ha permanecido prácticamente sin cambios por casi cien años. No fue sino hasta la década de 1990 que Microsoft Office Project agregó por primera vez las líneas de vínculo a estas barras de tareas, para remarcar dependencias entre dichas tareas.

En la actualidad, el legado de Henry Gantt se recuerda con una medalla entregada en su nombre por la Sociedad Estadounidense de Ingenieros Mecánicos (American Society of Mechanical Engineers).

Taylor, Gantt y otros contribuyeron a que la administración de proyectos sea una función empresarial destacada que requiere de estudio y disciplina. En las décadas anteriores a la Segunda Guerra Mundial, los métodos de marketing, la psicología industrial y las relaciones humanas comenzaron a tener un lugar como partes fundamentales de la administración de proyectos. (Cleland, 1999).

Para mediados del siglo XX, durante la segunda guerra mundial, el gobierno de Estados Unidos manejó proyectos militares y una importante reducción de tiempo a sus respectivas obras lo cual generó, nuevas estructuras de organización. Diagramas de red complejos, denominados diagramas PERT y el método de ruta crítica; dichas herramientas ofrecieron un mayor control sobre la ingeniería de proyectos muy complejos (como sistemas militares de armas con una gran variedad de tareas y varias interacciones en muchos puntos cronográficos).

En poco tiempo, estas técnicas se extendieron a todos los tipos de industrias a medida que los líderes empresariales buscaban estrategias y herramientas nuevas de administración para controlar su crecimiento en un mundo competitivo en constante evolución. A comienzos de la década de los sesenta, las empresas comenzaron a aplicar teorías generales del sistema a las interacciones empresariales. En su libro *The Theory and Management of Systems*, Richard Johnson, Fremont Kast y James Rosenzweig describieron cómo una empresa moderna se parece al organismo de un ser humano, con el sistema óseo, muscular, circulatorio, nervioso y demás.

Hoy en día, Este punto de vista de considerar a una empresa como el organismo de un ser humano implica que, para que un negocio pueda sobrevivir y prosperar, todas sus piezas funcionales deben trabajar en conjunto hacia objetivos o proyectos específicos. En las décadas posteriores a los años sesenta, este enfoque hacia la administración de proyectos comenzó a afianzarse en sus formas modernas. Aunque varios modelos empresariales distintos evolucionaron durante este período, todos comparten una estructura subyacente común: un administrador del proyecto administra el proyecto, reúne un equipo y garantiza la integración y comunicación horizontal del flujo de trabajo en los diferentes departamentos. (Saladis, 2009)

Durante las últimas décadas, la administración de proyectos no ha dejado de evolucionar. Existen dos tendencias importantes emergentes:

- Planeación descendente y revisión: Esta tendencia se caracteriza por la toma de decisiones en toda la empresa sobre la cartera de proyectos que una organización debería tener y por permitir que las tecnologías de minería de datos hagan más transparente la información de la cartera.
- Planeación ascendente: Esta tendencia enfatiza el uso de diseños de proyectos más sencillos, ciclos más breves, colaboración eficiente entre los miembros del equipo, participación más sólida de los miembros del equipo y toma de decisiones. Este enfoque se conoce ampliamente como administración de proyectos ágil e incluye varias metodologías relacionadas, como *SCRUM*, *Crystal*, *Extreme Programming*, *Unified Process* y muchas otras.

## 7.2 Administración de proyectos

Desde una perspectiva general surge la pregunta ¿En esencia que es un proyecto y como se define?

Según el PMI en su guía PMBOK, un proyecto se podría definir como “un esfuerzo temporal que se lleva a cabo para crear o alcanzar un producto, servicio o resultado único” (PMBOK, 2017, p3)

Se dice entonces que un proyecto tiene un inicio y un fin, limitado por tres factores principalmente; el tiempo, el presupuesto y el alcance. Dichas restricciones trabajan en correlación, ya que si alguna se ve afectada positiva o negativamente, las otra dos también sufrirán impacto. Por ejemplo, si el alcance del proyecto se ve afectado por algún requisito adicional o la mala interpretación

de algún requerimiento de negocio, automáticamente el proyecto está propenso a sufrir retrasos; en su defecto, si el administrador de proyectos decide finalizar en los tiempos establecidos, tendría que incorporar recursos adicionales o establecer horarios extendidos de trabajo, lo cual por ende genera un incremento de costos.

Con este principio, en un proyecto, la consecución de los objetivos al final del mismo es la máxima deseada, pero la mayor parte de las veces, ya sea por una mala planificación, alguna eventualidad o bien por una mala gestión de los recursos, es imposible finalizar el proyecto con éxito. Sin embargo, aun así, se puede considerar finalizado.

Existen diversos factores que pueden hacer que el proyecto no llegue a alcanzar los objetivos finales establecidos por los interesados en el proyecto (*stakeholders*). El seguimiento y gestión apropiados de dan por diversos factores, algunos considerados esenciales se muestran en la figura 3; Sin embargo, también algunos otros factores considerados como marco de referencia de un proyecto que se considera ideal cumple con lo siguiente:

- Entorno: No sufre modificaciones de forma rápida, sino que se alarga en el tiempo.
- Cliente: Tiene mucha claridad en que se necesita, sabe cómo transmitirlo y el resto de los equipos entenderá perfectamente sus necesidades.
- Equipo: Disponemos del equipo de profesionales necesario para poder atender esta necesidad y además sabemos cómo resolverla.
- Fases: Las fases se harían de una forma lineal, organizada y no surgiría ningún problema durante su realización.

**Figura 3. Características estándares de un proyecto**



Fuente: Trigas, 2016, p25

Tal como define el PMBOK a un esfuerzo de trabajo permanente, el cual es por lo general un proceso repetitivo que sigue los procedimientos existentes de una organización. En cambio, debido a la naturaleza única de los proyectos, pueden existir incertidumbres o diferencias en los productos, servicios o resultados que el proyecto genera. Las actividades del proyecto pueden ser nuevas para los miembros del equipo del proyecto, lo cual puede requerir una planificación con mayor dedicación que si se tratara de un trabajo de rutina. Además, los proyectos se llevan

a cabo en todos los niveles de una organización. Un proyecto puede involucrar a una única persona o a varias personas, a una única unidad de la organización, o a múltiples unidades de múltiples organizaciones. (PMBOK, 2017, p3)

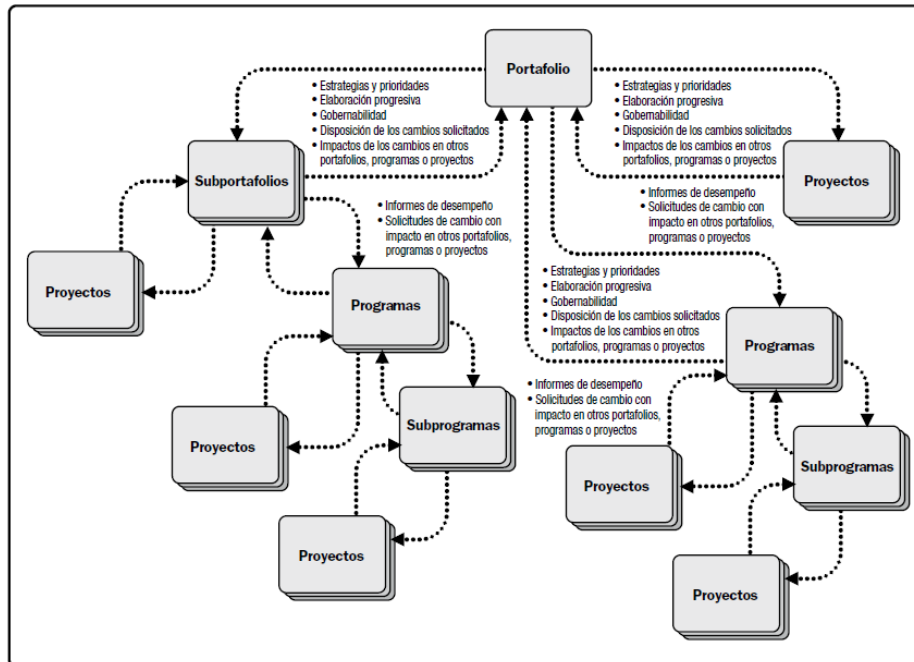
Por ende, un proyecto puede lograr los siguientes puntos:

- Un producto, que puede ser un componente de otro elemento, una mejora de un elemento o un elemento final en sí mismo.
- Un servicio o la capacidad de realizar un servicio (p.ej., una función de negocio que brinda apoyo a la producción o distribución).
- Una mejora de las líneas de productos o servicios existentes (p.ej., Un proyecto Seis Sigma cuyo objetivo es reducir defectos).
- Un resultado, tal como una conclusión o un documento (p.ej., un proyecto de investigación que desarrolla conocimientos que se pueden emplear para determinar si existe una tendencia o si un nuevo proceso beneficiará a la sociedad).

Un proyecto puede ser cualquier proceso de transformación en cualquier ámbito, desde un desarrollo Software de un producto en específico, la infraestructura de telecomunicaciones, cambio de procesos en alguna empresa, el desarrollo de una investigación minuciosa que se registre adecuadamente, por mencionar algunos.

Es importante destacar también la relación que existe entre las dimensiones de gestión de proyectos, puesto que para este fin existen tres niveles: “La relación entre portafolios, programas y proyectos es tal que un portafolio se refiere a un conjunto de proyectos, programas, subconjuntos de portafolios y operaciones que se gestionan como un grupo para alcanzar determinados objetivos estratégicos”, ver figura 4. (PMBOK, 2017, p4)

**Figura 4. Interacción entre Dirección de proyectos, programas y portafolios**



Fuente: PMBOK, 2017, p7

De acuerdo con la gráfica anterior, la relación que existe entre los programas como nivel macro y los proyectos como nivel micro, pasando por los portafolios en un punto medio; es de suma importancia debido a los lineamientos a los que se apegan de forma tripartita, en otras palabras, los programas y proyectos deben estar alineados al enfoque y objetivos del portafolio. Trabajar en función a este, para no desviar recursos y esfuerzos a un punto fuera de la visión empresarial.

### 7.3 Dirección de proyectos

Según el PMBOK (2017, p9) la dirección de proyectos se define como la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos del mismo. También menciona que esta dirección se logra mediante la aplicación e integración adecuada de los 49 subprocesos de la gestión de proyectos, agrupados en forma lógica y que a su vez se catalogan en cinco grupos de procesos como son: 1. Inicio, 2. Planificación, 3. Ejecución, 4. Monitoreo y Control y 5. Cierre.

Generalmente el director del proyecto debe considerar aspectos clave como son la identificación de requisitos; concatenación de necesidades, inquietudes y expectativas de los interesados en que el proyecto se realice; comunicaciones efectivas entre dichos interesados; gestionar a los interesados para lograr culminar los entregables establecidos; equilibrar algunas restricciones como el alcance, el tiempo, los costos, la calidad, los riesgos, por mencionar algunos.

Como se menciona anteriormente, la relación entre estos factores es tal que, si alguno de ellos cambia, es probable que al menos otro de ellos se vea afectado. Por ejemplo, si el cronograma es acortado, a menudo el presupuesto necesita ser incrementado a fin de añadir recursos adicionales para completar la misma cantidad de trabajo en menos tiempo. Si no fuera posible aumentar el presupuesto, se podría reducir el alcance o los objetivos de calidad para entregar el resultado final del proyecto en menos tiempo y por el mismo presupuesto. Los interesados en el proyecto pueden tener opiniones diferentes sobre cuáles son los factores más importantes, creando un desafío aún mayor. La modificación de los requisitos o de los objetivos del proyecto también puede generar riesgos adicionales. El equipo del proyecto necesita ser capaz de evaluar la situación, equilibrar las demandas y mantener una comunicación proactiva con los interesados a fin de entregar un proyecto exitoso.

Cabe mencionar que basado en entrevistas y comentarios de expertos, se tiene la idea que un proyecto gestionado bajo los principios del PMBOK no tiene la capacidad de manejar los cambios de forma dinámica y agregarlos rápidamente al alcance; dado el potencial de cambios, el desarrollo del plan de gestión es muy iterativo ya que su elaboración es progresiva a lo largo del ciclo de vida del proyecto. Esto significa que progresivamente los cambios se van incorporando en el proyecto por medio de un control estricto de documentación, dicho control consume un tiempo considerable para el gestor del proyecto.

En diversas organizaciones, la dirección de proyectos está a cargo de una oficina denominada PMO (Project Management Office), dicha oficina es una 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 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 (como por ejemplo cuadro de mando integral). Puede que los proyectos que la PMO apoya o dirige no guarden más relación entre sí que la de ser gestionados conjuntamente. La forma, la función y la estructura específicas de una PMO dependen de las necesidades de la organización a la que ésta da soporte.

“Una PMO puede tener la autoridad para actuar como un interesado integral y tomar decisiones clave a lo largo de la vida de cada proyecto, hacer recomendaciones, poner fin a proyectos o tomar otras medidas, según sea necesario, a fin de mantenerlos alineados con los objetivos de negocio” (PMBOK, 2017, p13). Asimismo, la PMO puede participar en la selección, gestión e utilización de recursos de proyectos compartidos o dedicados.

Otro factor que destacar es la importancia de la gestión de las operaciones en las organizaciones, dicha gestión se puede realizar por medio de cambios o ajustes a la visión y estrategia organizacional, por medio de proyectos de implementación o de cualquier otro tipo. A pesar de su naturaleza temporal, los proyectos pueden contribuir al logro de los objetivos de la organización cuando están alineados con su estrategia. Las organizaciones modifican a veces sus operaciones, productos o sistemas mediante la generación de iniciativas estratégicas de negocio que se desarrollan e implementan a través de proyectos. Los proyectos requieren actividades de dirección de proyectos y conjuntos de habilidades, mientras que las operaciones requieren gestión de procesos de negocio, actividades de gestión de las operaciones y conjuntos de habilidades.

Por tal motivo los cambios en las operaciones de negocio pueden dar lugar a un proyecto específico, en particular cuando se producen cambios sustanciales en las mismas como consecuencia de la entrega de un nuevo producto o servicio. Las operaciones permanentes están fuera del alcance de un proyecto; sin embargo, existen puntos de intersección en que se cruzan ambas áreas.

Cuando en el alcance del proyecto, el gestor desconoce las limitantes del producto o servicio a desarrollar, usualmente se llega a caer de forma eventual, a la ejecución de operaciones, en vez de acotar el alcance del entregable se ejecuta un ciclo de vida constante de las mismas actividades previamente implementadas. Por lo anterior, el administrador de proyectos debe conocer los requisitos del cliente a detalle y saber si se debe o no implementar un cambio documentado para ajustar dicho alcance a los criterios de aceptabilidad del usuario final.

Desafortunadamente el desarrollo de un producto o servicio relacionado a TI o Software, no es tan sencillo. Esto es que debido al entorno al que se desarrolla como nuevas tecnologías, incremento o disminución de funcionalidades, competencia, entre otras; sufre modificaciones de forma constante, de tal manera que la idea del proyecto esencial o que se contempló en un inicio, termina mutándose en otra cosa muy diferente.

Cuando se define el planteamiento del proyecto o producto a desarrollar, se deben considerar los siguientes factores:

- Cliente: Será la persona que está requiriendo una solución para un problema específico.
- Usuarios: La persona que utiliza este utilizando la solución o producto desarrollado.
- Tiempo: Donde las fechas de inicio y final son establecidas.
- Costos: La cantidad de dinero que se utilizará en función al tiempo y esfuerzo utilizados.

- Administrador de Proyectos: Es el o los individuos que se necesitan para fungir como gestor y administrador de recursos, así como la planificación y control del mismo.

Otras características descriptivas del PMBOK que pueden ser tomadas con enfoque negativo debido a su naturaleza estática son:

- El jefe de proyecto puede no tener conocimientos técnicos y dedicarse exclusivamente al control siguiendo los procedimientos establecidos y limitándose a la generación de informes, actas, diagramas de Gantt, WBS (Estructura de desglose de trabajo, por sus siglas en inglés), así como herramientas que facilitan la gestión pero que no forman parte del objetivo del proyecto. Un jefe de proyecto con estas características no podrá participar activamente en la toma de decisiones técnicas.
- En proyectos largos, ceñirse a un plan estático puede provocar que el producto final ya no cubra la totalidad de las necesidades del cliente dado que estas han cambiado desde el inicio. Por tal motivo, durante el propio desarrollo del producto, es posible que se deban ampliar las características diseñadas inicialmente con tal de que no sea obsoleto antes de su salida al mercado.
- Incertidumbre: Debido al entorno rápido e inestable que se vive, donde cumplir el plan inicial no garantiza el éxito. La idea de “producto terminado” puede perder su sentido en determinados sectores, por ejemplo, en el desarrollo de software, dado que el producto está evolucionando constantemente. La capacidad de adaptación a partir de la retroalimentación e incorporación de nuevas ideas es fundamental en la industria tecnológica.

#### 7.4 Definición de las áreas de conocimiento del PMBOK.

Dentro de la estructura de la gestión de proyectos definida por el PMI, se han identificado 5 procesos y 10 áreas del conocimiento que integran la administración de proyectos, estos se describen a continuación:

Procesos en la administración de proyectos:

1. El grupo de procesos de iniciación. Aquellos procesos realizados para definir un nuevo proyecto o una nueva fase de un proyecto ya existente, mediante la obtención de la autorización para comenzar dicho proyecto o fase.
2. El grupo de procesos de planificación. Aquellos procesos requeridos para establecer el alcance del proyecto, refinar los objetivos y definir el curso de acción necesario para alcanzar los objetivos para cuyo logro se emprendió el proyecto.
3. El grupo de procesos de ejecución. Aquellos procesos realizados para completar el trabajo definido en el plan para la dirección del proyecto a fin de cumplir con las especificaciones de este.
4. El grupo de procesos de seguimiento y control. Aquellos procesos requeridos para dar seguimiento, analizar y regular el progreso y el desempeño del proyecto, para identificar áreas en las que el plan requiera cambios y para iniciar los cambios correspondientes.
5. El grupo de procesos de cierre. Aquellos procesos realizados para finalizar todas las actividades a través de todos los grupos de procesos, a fin de cerrar formalmente el proyecto o una fase del mismo.

Con respecto a las 10 áreas de conocimiento:

1. **Gestión de la integración del proyecto.** Incluye los procesos y actividades necesarios para identificar, definir, combinar, unificar y coordinar los diversos procesos de la dirección de proyectos dentro de los grupos de procesos de dirección de proyectos. La gestión de la integración del proyecto implica tomar decisiones en cuanto a la asignación de recursos, balancear objetivos y manejar las interdependencias entre las áreas de conocimiento; así como también la gestión del conocimiento del proyecto.
2. **Gestión del alcance del proyecto.** Incluye los procesos necesarios para garantizar que el proyecto incluya todo el trabajo requerido para completarlo con éxito. El objetivo principal de esta área, es definir y controlar qué se incluye y qué no, en el proyecto.
3. **Gestión de la programación del proyecto.** Incluye los procesos requeridos para administrar la finalización del proyecto a tiempo, los cuales son: Definir las actividades, secuenciar las actividades, estimar los recursos de las actividades, estimar la duración de las actividades, desarrollar el cronograma y controlar el cronograma.
4. **Gestión de los costos del proyecto.** Incluye los procesos involucrados en estimar, presupuestar y controlar los costos de modo que se complete el proyecto dentro del presupuesto aprobado.
5. **Gestión de la calidad del proyecto.** Incluye los procesos y actividades de la organización ejecutante que determinan responsabilidades, objetivos y políticas de calidad a fin de que el proyecto satisfaga las necesidades por las cuales fue emprendido.
6. **Gestión de los recursos del proyecto.** Incluye los procesos que organizan, estiman, gestionan y conducen el equipo del proyecto. El equipo del proyecto está conformado por aquellas personas a las que se les ha asignado roles y responsabilidades para completar el proyecto. También los recursos se deberán controlar. (PMBOK, 2017, p15)
7. **Gestión de las comunicaciones del proyecto.** Incluye los procesos requeridos para garantizar que la generación, la recopilación, la distribución, el almacenamiento, la recuperación y la disposición final de la información del proyecto sean adecuados y oportunos.
8. **Gestión de los riesgos del proyecto.** Incluye los procesos relacionados con llevar a cabo la planificación de la gestión, la identificación, el análisis, la planificación de respuesta a los riesgos, la implementación de la respuesta a los riesgos previamente analizados, así como su monitoreo y control en un proyecto.
9. **Gestión de las adquisiciones del proyecto.** Incluye los procesos de compra o adquisición de los productos, servicios o resultados que es necesario obtener fuera del equipo del proyecto. Incluye los procesos de gestión del contrato y de control de cambios requeridos para desarrollar y administrar contratos u órdenes de compra.
10. **Gestión de los interesados del proyecto.** Incluye los procesos necesarios para identificar a las personas, grupos u organizaciones que pueden afectar o ser afectados por el proyecto, para analizar las expectativas de los interesados y su impacto en el proyecto.

## 7.5 Metodologías de desarrollo de proyectos de Software

En el desarrollo del software, se entiende que la tecnología es aquella disciplina que indica que métodos, técnicas y herramientas se deben usar en cada fase del ciclo de vida de desarrollo del software. Existen varias metodologías para la gestión de proyectos, desde un punto de vista técnico se puede decir que el proceso antes mencionado es un marco de trabajo de las tareas que se requieren para construir software de alta calidad. El proceso proporciona una iteración entre los usuarios y los diseñadores, entre los usuarios y las herramientas de desarrollo, y entre los diseñadores y las herramientas de desarrollo (tecnología). Es un proceso interactivo donde la herramienta de desarrollo se usa como medio de comunicación, con cada iteración del dialogo se obtiene mayor conocimiento.

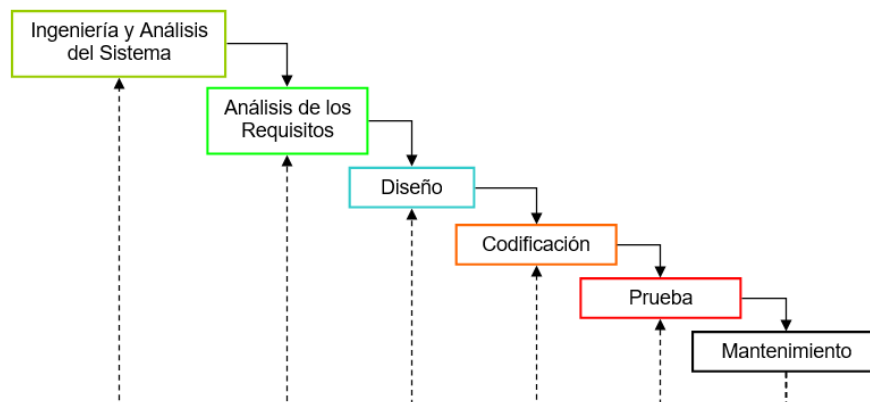
### 7.5.1 Metodología de Cascada

También conocida como modelo de cascada, fue creada por Bennington en 1956, y posteriormente modificado por Royce en 1970. Pressman lo presenta como ciclo de vida clásico y se denomina en cascada porque su característica principal es que no se comienza con un paso hasta que no se ha terminado el anterior. La metodología de cascada establece que el software debe ser construido, rigurosamente, a través de una transformación sucesiva de documentos, siguiendo una estrategia lineal de desarrollo. Primero saber qué se quiere y después, cuando se conozca todo lo que se quiere, empezar a construirse comienza la construcción.

El método es también conocido como modelo lineal secuencial sugiere un enfoque sistemático, secuencial para el desarrollo del software que comienza en un nivel de sistemas y avanza con el análisis, diseño, fase de construcción, fase de pruebas y la post implementación.

El primer paso es conseguir un documento con la especificación completa, exacta, no ambigua de los requisitos del sistema software que debe ser desarrollado. Este documento inicial es transformado en un documento de análisis, supuestamente alejado de la máquina. Después, a partir del análisis, se obtiene otro documento, el diseño. Por último, del diseño se obtiene el documento final: el código. Para asegurar que no se introducen equivocaciones al transformar un documento (modelo) en otro, se hacen pruebas, al terminar cada uno. Las pruebas son planificadas desde el principio y se documentan como se vayan realizando. Antes de la entrega final del software, se valida que satisface los requisitos definidos en el documento inicial, tal como refiere la siguiente figura (fig. 5), las fases del método en cascada son:

**Fig. 5 Fases Metodología de Cascada**



Fuente: Zibert, 2016, p1

Las fases de la figura 5 implican:

1. Ingeniería y Análisis del Sistema: Debido a que el software es siempre parte de un sistema mayor, el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.

2. Análisis de los requisitos del software: el proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.

3. Diseño: el diseño del software se enfoca en cuatro atributos distintos del programa; la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.

4. Codificación: el diseño debe traducirse en una forma legible para la máquina. Si el diseño se realiza de una manera detallada, la codificación puede realizarse mecánicamente.

5. Prueba: una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.

6. Mantenimiento: el software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debidos a que se haya encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos) o a que el cliente requiera ampliaciones funcionales o del rendimiento.

En la metodología vemos una ventaja evidente y radica en su sencillez, ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software. Pero el método se aplica en un contexto, así que debemos atender también a él y saber que los proyectos reales raramente siguen el flujo secuencial que propone el modelo. Siempre hay iteraciones y se crean problemas en la aplicación del paradigma.

Normalmente, al principio, es difícil para el cliente establecer todos los requisitos explícitamente. El ciclo de vida clásico lo requiere y tiene dificultades en acomodar posibles incertidumbres que pueden existir al comienzo de muchos productos.

El cliente debe tener paciencia. Hasta llegar a las etapas finales del proyecto no estará disponible una versión operativa del programa. Un error importante que no pueda ser detectado hasta que el programa esté funcionando, puede ser desastroso. (Ble, 2010).

Con estas características metodológicas descritas con anterioridad, se debe considerar un riesgo alto en las implementaciones de sistemas nuevos, ya que las especificaciones y el diseño deben estar muy bien definidos, de lo contrario al ser un modelo descendente en su ejecución, el usuario final deberá esperar hasta la fase de pruebas para detectar y/o modificar algún elemento que considere no cumple con los criterios de calidad descritos en los requerimientos.

Por el contrario, la mayor ventaja de este tipo de metodología es que si se conoce y comprende a detalle el requerimiento, y se utilizan tecnologías conocidas; la implementación será más sencilla por su predictibilidad.

### 7.5.2 Metodología del modelo en espiral

El modelo espiral propuesto originalmente por Boehm en 1988, es un modelo de proceso de software evolutivo ha sido desarrollado para cubrir las mejores características tanto del ciclo de vida clásico, como de la creación de prototipos, añadiendo al mismo tiempo un nuevo elemento: el análisis de riesgo. Proporciona el potencial para el desarrollo rápido de versiones incrementales del software. En este modelo el software se desarrolla en una serie de versiones incrementales. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. En las últimas iteraciones, se producen versiones cada vez más completas del sistema diseñado.

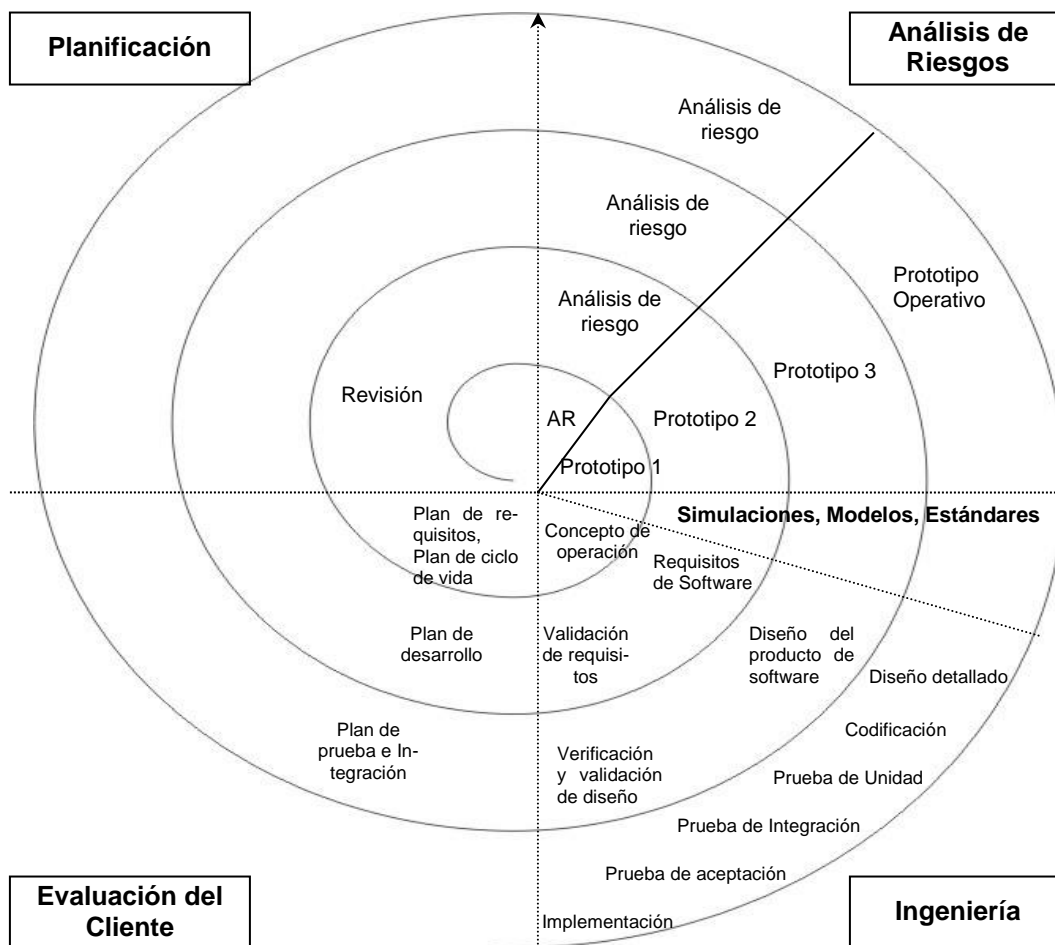
El modelo representado mediante la espiral de la figura 6 define cuatro actividades principales, también llamadas regiones de tareas o sectores:

1. Planificación. Durante la primera vuelta alrededor de la espiral se definen los objetivos, las alternativas y las restricciones, se analizan e identifican los riesgos. Si el análisis de riesgo indica que hay una incertidumbre en los requisitos, se puede usar la creación de prototipos en el cuadrante de ingeniería para dar asistencia tanto al encargado de desarrollo como al cliente.
2. Análisis de riesgo. El proyecto se revisa y se toma la decisión de si se debe continuar con un ciclo posterior de la espiral. Si se decide continuar, se desarrollan los planes para la siguiente fase del proyecto.
3. Ingeniería. En este sector se desarrolla y se valida. Después de la evaluación de riesgos, se elige un modelo para el desarrollo del sistema.
4. Evaluación del cliente. El cliente evalúa el trabajo de ingeniería (cuadrante de evaluación de cliente) y sugiere modificaciones. Sobre la base de los comentarios del cliente se produce la siguiente fase de planificación y de análisis de riesgo. En cada bucle alrededor de la espiral, la culminación del análisis de riesgo resulta en una decisión de "seguir o no seguir". (Sommerville, 2005).

De acuerdo con Sommerville Un ciclo en espiral inicia con la elaboración de objetivos, como el rendimiento y la funcionalidad. Después se enumeran formas alternativas de alcanzar estos objetivos y las restricciones impuestas en cada una de ellas. Cada alternativa se evalúa contra cada objetivo y se identifican las fuentes de riesgo del proyecto. Lo siguiente es resolver los riesgos mediante actividades de recopilación de información como la de detallar más el análisis, la construcción de prototipos y la simulación. Ya que se evaluaron los riesgos, se lleva a cabo cierto desarrollo, seguido de una actividad de planificación para la siguiente fase, ver figura 6.

El paradigma del modelo en espiral para la Ingeniería de Software es actualmente el enfoque más realista para el desarrollo de software y de sistemas a gran escala. Utiliza un enfoque evolutivo, permitiendo al desarrollador y al cliente entender y reaccionar a los riesgos en cada nivel. Utiliza la creación de prototipos como un mecanismo de reducción de riesgo, pero, lo que es más importante permite a quien lo desarrolla aplicar el enfoque de creación de prototipos en cualquier etapa de la evolución de prototipos.

Fig. 6 Modelo de espiral de Boehm



Fuente: Sommerville, 2005, p247

### 7.5.3 Metodologías Ágiles

En años anteriores la tendencia en el desarrollo de proyectos de software de los productos y servicios tecnológicos era muy lenta, y se producía la mayor parte del tiempo en un entorno estable con muy pocas variaciones o cambios.

En cambio, en la actualidad el entorno donde se mueven los productos de software, así como los servicios de TI, es demasiado inestable y con cambios continuos, por esta razón las pesadas metodologías rígidas de cascada han estado, poco a poco, perdiendo popularidad y utilidad en la industria debido a la falta de adaptación y velocidad sin sacrificar la calidad.

Por esta razón, principalmente se diferencian las metodologías ágiles de las pesadas convencionales; estas últimas requieren necesariamente conocer desde el inicio que desea el cliente. Tampoco se pueden cambiar los requisitos iniciales puesto que a medida que se sigue construyendo el proyecto las modificaciones y la corrección de los errores es más costosa. Además, todos los cambios que se produzcan los sufrirá económicamente el cliente.

Posterior a la identificación de las organizaciones sobre este tipo de restricciones, y con el entorno cada vez más cambiante, con tiempos de desarrollo más cortos, surge el concepto de “metodología ágil” como una alternativa bastante atractiva para las empresas de software.

El desarrollo ágil está centrado en la interacción, comunicación y reducir elementos intermedios. El desarrollo con interacciones se realiza normalmente en porciones de tiempo pequeñas, denominadas “*timeboxes*” y se ocupará de desarrollarlas un equipo multidisciplinario, auto organizado y auto controlado, ya que ellos mismos decidirán como desarrollar las tareas de la interacción en cuestión.

Otro factor a destacar es que el método de desarrollo ágil fomenta la comunicación entre los miembros del equipo lo que previene problemas que en otra metodología quedan escondidos. Dicha comunicación no solo se establece de forma cerrada entre los miembros del equipo, sino que también se realiza con la figura que representa el cliente.

La persona que representa al cliente o gerente de alto nivel, es de suma importancia y apoyo para el equipo de desarrolladores, puesto que será la persona a la que se le podrán hacer las preguntas necesarias y que, junto con el resto de personas involucradas en el negocio, comprobarán si se cumplen o no con los objetivos establecidos. Por lo tanto, si se trabaja con una gran comunicación entre el equipo involucrado, se hará más fácil poder tomar decisiones de forma más rápida, así como aplicarlas.

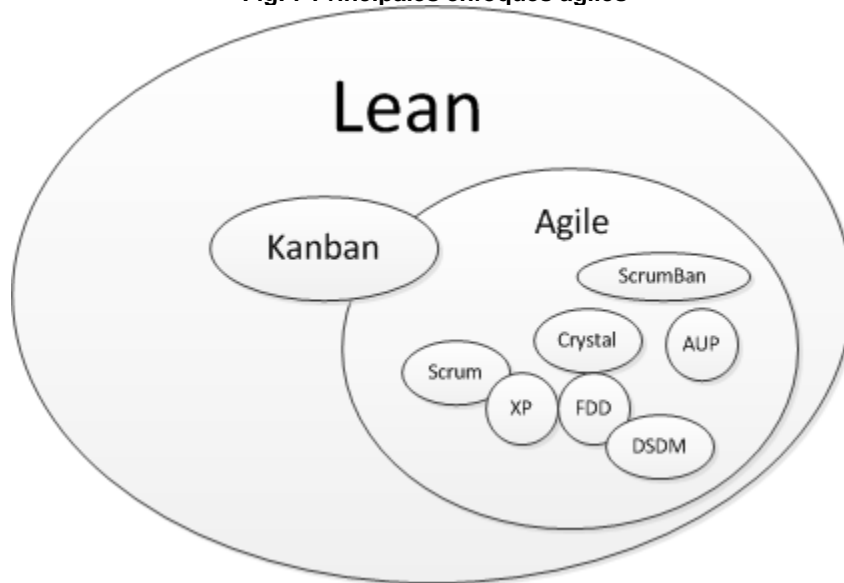
La característica realmente nueva que aportan estas metodologías, es reconocer a las personas como el principal valor para que un proyecto consiga terminarse de forma correcta. “El factor más importante en el desarrollo de software no son las técnicas y las herramientas que emplean los programadores, sino la calidad de los programadores”. (Robert L. Class, 2001).

A continuación, se enlistan algunas de las principales características y ventajas de las metodologías ágiles: (Fowler, 1999, p111)

1. “Gran capacidad de respuesta ante los cambios, los cuales no se definen como un problema sino como algo necesario para que el producto sea mejor y satisfaga al cliente. Los cambios formarán parte del proceso de desarrollo.
2. Las entregas no se hacen al final, sino que se hacen pequeñas entregas. Estas entregas permiten valorar el producto al cliente, además de poder ir trabajando o interactuando con algunas funcionalidades.
3. Los ciclos cortos de entrega ayudarán a disminuir los riesgos, sobre todo al inicio del proyecto.
4. Se trabaja en equipo entre el cliente y los desarrolladores, mediante una comunicación casi diaria para evitar errores y documentación innecesaria.
5. Eliminar el trabajo que no es necesario y que realmente no aporta un valor al negocio.
6. Buscar la mejor técnica y mejor diseño para conseguir productos o servicios de calidad.
7. Mejorar los procesos y al equipo que realiza el desarrollo.
8. Está basado en la iteración, cada iteración debe liberar una nueva funcionalidad completa.
9. Se centra en el tiempo, al final de cada iteración la funcionalidad desarrollada debe estar lista para ponerse en producción.
10. Está diseñada para soportar altos grados de cambios por lo que sus procesos flexibles permiten establecer cambios de forma fiable a lo largo del desarrollo.

11. La Arquitectura y la documentación son incrementales durante el desarrollo del proyecto.
12. El desarrollo se basa en la realización de funcionalidades en iteraciones cortas”.

Fig. 7 Principales enfoques ágiles



Fuente: PMI & Agile Alliance, 2017, p11

En la figura 7 se muestra que el término *Agile* hace referencia a varios tipos de enfoques y métodos, los cuales a su vez tienen diferentes y variados marcos referenciales. También en dicha figura se muestra *Agile* y *Kanban* como un subconjunto de *Lean*; históricamente el pensamiento *Lean* surge con definiciones conceptuales de “enfoque en el valor”, “pequeñas piezas como entregables”, “eliminación de desperdicios” entre otros, antes del boom del enfoque ágil.

La principal idea del pensamiento ágil no debe confundirse con la entrega prematura de hitos o la forma de trabajar dinámica, sino debe enfocarse en la implementación de cambios de forma que encaje con el progreso de los valores principales y principios tanto de la empresa, como del proyecto en sí. Algunas prácticas como el uso de cuadros de progreso para la creación de *features*, la refinación de los mismos; otras técnicas como desmenuzar los requerimientos en pequeñas tareas fáciles de estimar, ayudan a la implementación de la metodología ágil no solo por el nombre, sino por el flujo del valor continuo que se entrega a los clientes y un mejoramiento continuo de los entregables de negocio. (Agile Practice Guide, 2017, p12)

#### 7.5.3.1 Manifiesto Ágil

En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término “ágil” aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue diseñar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto.

Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. Tras esta reunión se creó *The Agile Alliance*, una organización, sin

ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida es fue el Manifiesto Ágil, un documento que resume la filosofía “ágil”. (Abrahamsson, 2002, p186).

Según el Manifiesto se valora:

- *El individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.* La gente es el principal factor de éxito de un proyecto software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- *Desarrollar software que funciona más que conseguir una buena documentación.* La regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental.
- *La colaboración con el cliente más que la negociación de un contrato.* Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- *Responder a los cambios más que seguir estrictamente un plan.* La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.

Los valores anteriores inspiran los doce principios del manifiesto. Son características que diferencian un proceso ágil de uno tradicional. Los dos primeros principios son generales y resumen gran parte del espíritu ágil. El resto tienen que ver con el proceso a seguir y con el equipo de desarrollo, en cuanto metas a seguir y organización del mismo. Los principios son:

1. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
2. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
3. Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
4. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
5. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
6. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
7. El software que funciona es la medida principal de progreso.
8. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
9. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
10. La simplicidad es esencial.
11. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
12. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento. (Bob Martin, 2001)

## 7.6 SCRUM

SCRUM es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en SCRUM son el SCRUM Master, que mantiene los procesos y trabaja de forma similar al director de proyecto, el *Product Owner*, que representa a los *stakeholders* (interesados externos o internos), y el Equipo que incluye a los desarrolladores. SCRUM está catalogada como una metodología de desarrollo AGILE con ciclos secuencias con solapamiento.

SCRUM permite la creación de equipos auto organizado impulsando la co-localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

Un principio clave de SCRUM es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado *requirements churn*), y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, SCRUM adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

Existen varias implementaciones de sistemas para gestionar el proceso de SCRUM, que van desde notas amarillas “post-it” y pizarras hasta paquetes de software. Una de las mayores ventajas de SCRUM es que es muy fácil de aprender, y requiere muy poco esfuerzo para comen-zarse a utilizar. (Genaro J Rodriguez, 2012).

El proceso del desarrollo de SCRUM es básicamente muy simple, como se muestra en la figura 7.3 factores esenciales para el desarrollo de este proceso son los roles, los componentes y las reuniones.

Fig. 7 SCRUM Ficha Sinóptica



Fuente: Velázquez, 2012, p34

Componentes:

Según Velázquez, SCRUM se divide en tres principales componentes, la pila del producto o *Product Backlog*, la Pila del Sprint o *Sprint Backlog* y los incrementos o *features*.

El *Product Backlog (PB)* es la lista de requerimientos a nivel general, los cuales se modifican, priorizan y ajustan de acuerdo a las necesidades del dueño del producto, en este componente, el detalle no es necesario y la evolución del mismo va a depender del avance de los “*Sprints*” o iteraciones.

El *Sprint backlog (SB)* es el detalle de los requerimientos obtenidos en el PB pero que están planeados y comprometidos por el equipo dentro del Sprint siguiente, este tipo de “pila” tiene un nivel de detalle tal, que permite su ejecución en los parámetros establecidos por la metodología.

Los *features* son componentes o partes del SB, también se identifican como tareas por desarrollar que cumplen con el principio de utilidad e incrementalidad; estas pueden ser pruebas de usuario, codificación limpia y documentada, pruebas de aceptabilidad, entre otras.

Reuniones:

SCRUM se puede interpretar de diversas maneras, pues no solo es una metodología flexible enfocada al manejo de proyectos ágiles, sino también es un marco referencial que cualquier compañía puede seguir para obtener beneficios, por esta razón no existe un número determinado de reuniones forzosas a lo largo del desarrollo de un proyecto.

Sin embargo, algunos autores como como Velázquez identifican 3 reuniones mínimas a lo largo del proyecto. La llamada diaria de sprint, la reunión de planeación del sprint y la reunión de revisión de sprint.

La llamada o reunión diaria de sprint sirve para identificar por medio de 3 preguntas básicas, que trabajo han hecho los miembros del equipo, que trabajo harán durante el día y si tuvieran alguna restricción o impedimento para seguir avanzando. Es una forma fácil de dar seguimiento a las actividades del equipo.

La reunión de planeación de Sprint es para determinar las prioridades y aclarar dudas por parte del dueño del producto, mientras que el equipo estima el esfuerzo de acuerdo a dicha prioridad y se determina el SB, el maestro SCRUM, hace un pronóstico del tiempo y velocidad del proyecto para también así determinar junto con el dueño del producto, los *features* que quiere realizar en cada sprint para así entregar un valor de dicho producto.

La reunión de revisión se lleva a cabo cada que el sprint finaliza y tiene una duración aproximada de 4 horas (depende mucho de la organización), esta se realiza con la finalidad de revisar el *Sprint Backlog* y los *features* desarrollados en dicho sprint, también sirve para afinar detalles del proceso y del equipo y es la forma en que el equipo SCRUM le muestra al dueño del producto el avance y progreso que se ha tenido, es también el preámbulo de la planeación del siguiente sprint.

Como dato curioso, la palabra SCRUM procede de la terminología del juego de Rugby, donde designa al acto de preparar el avance del equipo en unidad pasando la pelota a uno y otro jugador (aunque hay otras acepciones en circulación). Igual que el juego, SCRUM es adaptativo, ágil, auto-organizado y con pocos tiempos muertos. (Hirotaka Takeuchi, 1986)

#### 7.6.1 Manejo de requerimientos en SCRUM

Las metodologías que usan metodologías ágiles como SCRUM, requieren que todo el equipo de desarrollo esté involucrado en todas las etapas del proyecto en constante contacto con el usuario final de la aplicación. Los requerimientos Agile son representados como “Historias de Usuario” (*Userstories*), la cual es una pequeña descripción del requerimiento del usuario descrito en un lenguaje lo más aproximado a su jerga diaria. Se hace un énfasis muy fuerte a que este tipo de comunicación esté orientada al usuario final y que sea fácil de entender. Estas historias definen alcances fáciles de identificar para poder efectuar planes iterativos en definición, revisión y confirmación de alcances. No se les exige a los usuarios finales que tengan que conocer desde un principio todos sus requerimientos. Sin embargo, se pide que no se cambien dichos requerimientos o “pila del *Sprint*” durante los ciclos iterativos (*Sprints*) que se definieron previamente para el desarrollo del producto, ver figura 8.

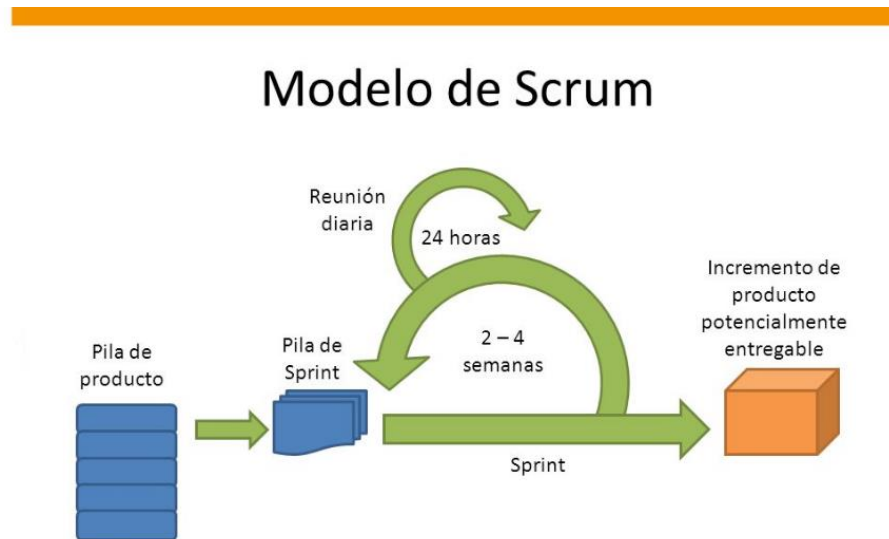
En SCRUM los requisitos se expresan como elementos del *Product Backlog*. Es una lista viva de requisitos funcionales y no funcionales priorizados por su valor para el cliente. Al decir que se trata de una lista viva, dejamos claro que los requisitos que en ella aparecen y el orden de los mismos, es cambiante a lo largo de la vida del proyecto. En SCRUM, los requisitos se van abordando en *Sprints* en el orden en que aparecen en el *Product Backlog*.

Utilizar el *Product Backlog* como principal artefacto para la gestión de requisitos nos permite atajar los problemas relacionados con la gestión de requisitos que antes hemos descrito.

Aunque construir y mantener el *Product Backlog* es una tarea ardua, es mucho más simple que hacer una captura de requisitos tradicional. El motivo es simple, solo hay que expresar a grandes rasgos en que consiste el requisito, no es necesario un nivel de detalle muy elevado. Solo es necesario el nivel de detalle suficiente que nos permita estimar los requisitos y priorizarlos. A

menudo se utilizan historias de usuario para expresar estos requisitos. Los requisitos que aparecen en el *Product Backlog* deben ser independientes, negociables, evaluables, estimables y no demasiado grandes. Deben ser independientes pues el orden en el que serán implementados puede cambiar. Deben ser negociables en el sentido de que son un punto de partida para comenzar en desarrollo no un contrato cerrado. Deben ser evaluables desde el punto de vista del retorno de la inversión que proporcionan a los clientes. Deben ser estimables pues es imposible priorizar algo de lo que se desconoce la magnitud. Y, por último, deben ser de un tamaño que permita estimarlos sin tener demasiadas incertidumbres sobre cuál es el alcance concreto del requisito, aunque cierto nivel de incertidumbre siempre va a existir. (Corral, 2007)

Fig. 8 Modelo de interacción Agile-SCRUM



Fuente: Velázquez, 2012, p50

Evidentemente antes de su implementación será necesario refinar en mayor detalle los requisitos. En SCRUM esto es algo que se pospone hasta que se planea el Sprint en el que vamos a abordar esos requisitos en concreto. Posponer el refinado detallado de los requisitos hasta un momento cercano a su implementación permite que cuando se realice esta actividad se tenga en cuenta de nuevo las necesidades del cliente que pueden haber cambiado y así poder contar con la información que se ha ganado durante la implementación de otros requisitos, de esta manera, al incrementarla se tendrá muchas más posibilidades de actuar correctamente a la hora de describir y estimar en detalle los requisitos. El fundamento para lo anteriormente expuesto es claro, es posible definir y estimar las actividades del proyecto en cuestión, cercanas en el tiempo con un buen margen de confianza, pero es muy complejo hacerlo con actividades cuyo momento de comienzo se encuentra lejano en el tiempo. Las actividades cercanas en el tiempo tienen una probabilidad mucho menor de sufrir cambios en su alcance que obliguen a cambiar su estimación y, en consecuencia, su planificación.

“Los cambios en los requisitos ocurren, por mucho que se trate de evitarlos. En SCRUM y en las metodologías ágiles en general, se asume que estos cambios van a ocurrir por lo que se trata de gestionar dichos cambios como oportunidades de que los clientes obtengan lo que necesitan. Evidentemente para que esto funcione la gestión del *Product Backlog* debe reaccionar de manera continua para reflejar las prioridades del cliente. El *Product Owner* (dueño del producto) es el

encargado de gestionar el *Product Backlog*, introduciendo nuevos requisitos según se descubren y priorizándolos en función de las necesidades del cliente y el retorno de la inversión que el cliente recibirá de su implementación”. (Corral, 2007, p98)

Otra aportación relevante en el análisis de SCRUM, fue propuesta por Kniberg y Skarin, 2010; la cual a continuación enlista algunos elementos que componen un desarrollo de Software bajo esta premisa:

### **Reuniones**

Planificación de sprint: Jornada de trabajo previa al inicio de cada sprint en la que se determina cuál va a ser el trabajo y los objetivos que se deben cumplir en esa iteración.

Reunión diaria: Breve revisión del equipo del trabajo realizado hasta la fecha y la previsión para el día siguiente.

Revisión de sprint: Análisis y revisión del incremento generado.

### **Roles**

SCRUM clasifica a todas las personas que intervienen o tienen interés en el desarrollo del proyecto en: propietario del producto, equipo, gestor de SCRUM (también *SCRUM Manager* o *SCRUM Master*) y “otros interesados”.

Propietario del producto: El responsable de obtener el mayor valor de producto para los clientes, usuarios y resto de implicados.

Equipo de desarrollo: Grupo o grupos de trabajo que desarrollan el producto.

*SCRUM Manager*: gestor de los equipos que es responsable del funcionamiento de la metodología SCRUM y de la productividad del equipo de desarrollo.

### **Valores**

SCRUM es una “carrocería” para dar forma a los principios ágiles. Es una ayuda para organizar a las personas y el flujo de trabajo; como lo pueden ser otras propuestas de formas de trabajo ágiles:

Delegación de atribuciones, la cual consta de empoderar al equipo para que pueda auto-organizarse y tomar las decisiones sobre el desarrollo.

Respeto entre las personas. Los miembros del equipo deben confiar entre ellos y respetar sus conocimientos y capacidades.

Responsabilidad y auto-disciplina (no disciplina impuesta).

Trabajo centrado en el desarrollo de lo comprometido.

Información, transparencia y visibilidad del desarrollo del proyecto.

#### 7.6.2 Correlación de áreas del conocimiento con SCRUM

En capítulos anteriores se habló sobre los 5 procesos y 10 áreas del conocimiento según el PMBOK, a continuación, se busca explicar la correlación de dichas áreas con el marco referencial de SCRUM.

INICIO: En el grupo procesos de inicio, normalmente lo que se busca desde la perspectiva del administrador del proyecto es, realizar procesos relacionados a la definición de un nuevo proyecto, o la continuidad de una fase de proyecto ya existente; esto normalmente se logra mediante la obtención de la autorización para comenzar ya sea la fase o el proyecto en sí.

En un ambiente ágil, en los procesos de inicio, en ocasiones denominados SCRUM 0, se hace una pre-planeación general del proyecto estimando tiempos y costos, se crea una visión general del proyecto a desarrollar por medio de un *'Backlog' (requisito priorizado)* y posteriormente se comienza la planeación del *'Sprint'*; la definición del alcance es equivalente al *Product Backlog*, así como la revisión del alcance del producto y la descomposición en partes. (PMBOK, 2017, p87)

**PLANEACIÓN:** Son aquellos procesos requeridos para establecer el alcance del proyecto, refinar los objetivos y definir el curso de acción necesario para alcanzar los objetivos para cuyo logro se emprendió el proyecto. En esta fase, se realiza la estructura general del desglose de trabajo y se hace una planeación de tiempos y costos, así como un análisis minucioso de los interesados en el proyecto y la planeación de las comunicaciones a lo largo del proyecto.

La propuesta de SCRUM es crear una planeación de la interacción o *Sprint* por medio de una revisión del Backlog, quiere decir que se revisa de manera particular cada historia de usuario contenida en un *feature* el cual está dentro del *Sprint Backlog* que a su vez se encuentra contenida en un *Product Backlog*; esta actividad de desagregar en pequeñas piezas de requerimiento es muy similar al sugerido por el PMBOK, la principal diferencia radica en el formato de redacción que se maneja como una historia, por ejemplo, "yo como usuario, requiero una aplicación de reservas de avión, para realizar mis compras desde mi teléfono móvil".

También es importante resaltar que dichas historias son dinámicas y adaptables, lo cual, al momento de estimar el tiempo del proyecto por medio de un 'mapa de camino' o una gráfica de *'Burn down'*, dichas estimaciones son variables *sprint* por *sprint* y se van a basar prácticamente en la velocidad con que se ejecutan los 'puntos de usuario' que contienen cada historia de usuario, por parte del equipo SCRUM.

Otro factor esencial es que dicha planeación se hace periódicamente de acuerdo al inicio de los *sprints*, esto no quiere decir que en los procesos de planeación del PMBOK no se realice, pues la planeación se va ajustando a los tiempos y costos, sin embargo, en SCRUM es mucho más fácil cambiar los requerimientos del usuario para adaptar el producto final, exactamente a las necesidades de dicho usuario. (PMBOK, 2017, p38)

**EJECUCIÓN:** Aquellos procesos realizados para completar el trabajo definido en el plan para la dirección del proyecto a fin de cumplir con las especificaciones de este.

En esta sección el equipo involucrado en el proyecto ejecuta el trabajo previamente planeado, por ejemplo, en el desarrollo de un proyecto de software, se realiza la construcción del código, se ajustan los ambientes de pruebas, se realizan pruebas de aseguramiento de la calidad, entre otras actividades.

Cabe señalar que como se muestra en la figura 9, estos procesos son iterativos, y se repiten de acuerdo a las necesidades del proyecto, tal cual pasa en un ambiente SCRUM, donde las historias de usuario se ejecutarán en cada *Sprint*, de acuerdo a la planeación del *Product Backlog*, con la diferencia que esta ejecución se lleva a cabo en periodos más cortos de tiempo, lo cual se traduce en modificaciones más constantes del producto para terminarlo con la mayor satisfacción hacia el usuario final.

**MONITOREO Y CONTROL:** Aquellos procesos requeridos para dar seguimiento, analizar y regular el progreso y el desempeño del proyecto, para identificar áreas en las que el plan requiera

cambios y para iniciar los cambios correspondientes. Este proceso es muy común en el área de software, ya que los requerimientos iniciales que se planearon, muchas veces no cumplen con la calidad o funcionalidad requerida por el usuario, este proceso es la principal razón por la que las metodologías ágiles comenzaron a adquirir mayor popularidad en la industria; pues sugiere un control constante de cambios, los cuales acercan el producto a la mayor calidad requerida, un ejemplo son las pruebas unitarias de usuario, donde el analista de calidad, monitorea el comportamiento de la aplicación en algún ambiente semejante a producción para así poder replicar y corregir cualquier error en el código de la aplicación.

Desde el punto de vista de SCRUM, este proceso es esencial, tal como se menciona en el párrafo anterior, es la principal correlación entre los procesos de monitoreo y control del PMBOK; existen algunas formas de llevar a cabo dicho control, la primera es una junta diaria con el equipo SCRUM donde se medirá el trabajo realizado y se identificarán los obstáculos para avanzar; también existe la forma de controlar el avance realizado en un *Sprint* o iteración completa, y es por medio de una reunión periódica (cada *Sprint*) con el representante del negocio o dueño del producto, donde se va a mostrar la evolución del producto a entregar con el respectivo valor planeado en los *Product Backlogs*. En dicha reunión se muestra la funcionalidad ya probada y funcionando, también se lleva a cabo una reunión denominada de 'Refinamiento' donde se van a ajustar las prioridades del usuario final, también se vuelve a estimar el esfuerzo a realizar por historias de usuario y finalmente se lleva a cabo la planeación del nuevo *Sprint*. (PMBOK, 2017)

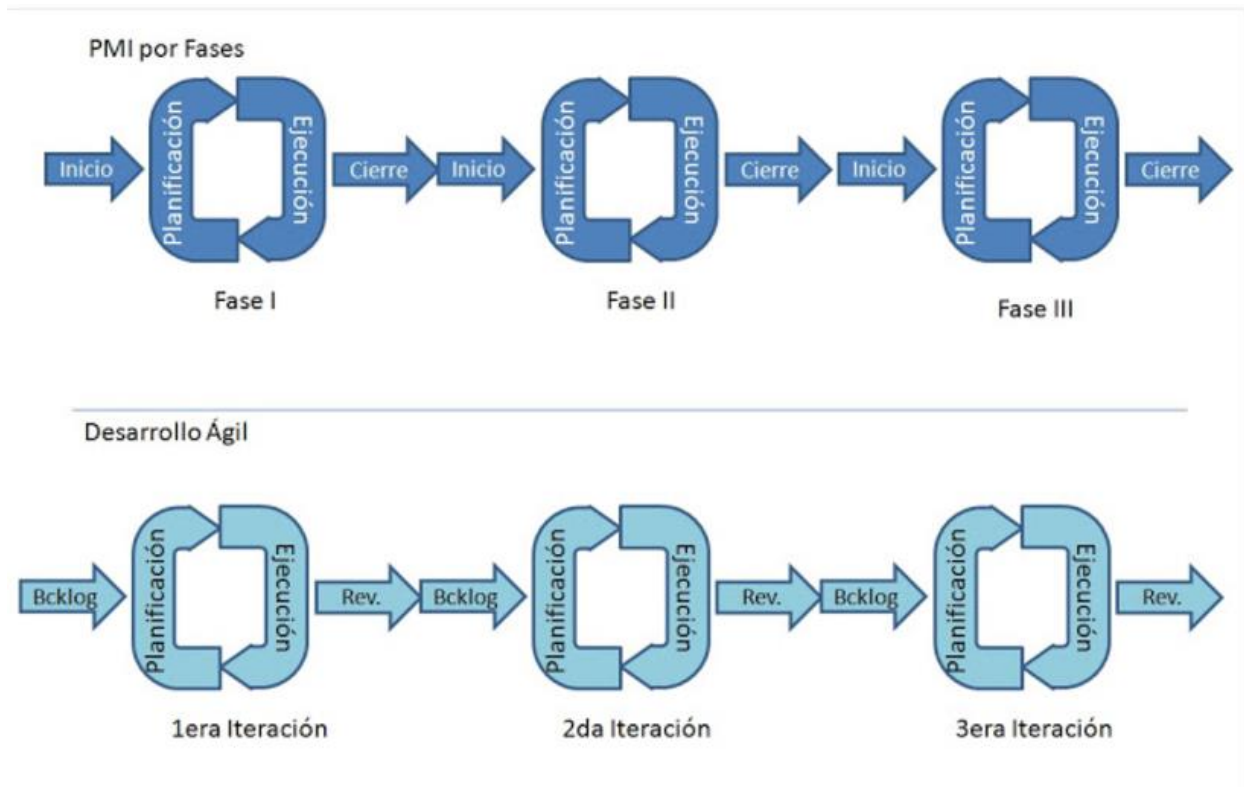
**CIERRE:** Aquellos procesos realizados para finalizar todas las actividades a través de todos los grupos de procesos, a fin de cerrar formalmente el proyecto o una fase del mismo. Normalmente se lleva a cabo cuando se da la firma de que el proyecto fue terminado, cabe destacar que, aunque el proyecto no esté completado, también se puede cerrar dicho proyecto por diversos factores.

También se lleva un control de las lecciones aprendidas para saber qué factores fueron cruciales para obtener el resultado que se obtuvo a lo largo del proyecto, y por último se hace un cierre de las adquisiciones que se contrataron durante el proyecto como pago de los proveedores de servicios, *testers*, desarrolladores, etc. (PMBOK, 2017)

En relación a los proyectos ágiles, se lleva un registro de las actividades realizadas durante los *Sprints* que podrían traducirse en lecciones aprendidas, este registro se hace en una reunión de retrospectiva con el equipo SCRUM para definir los pros y contras del proceso realizado en el *Sprint* anterior.

El cierre en un proyecto de SCRUM se da por medio de la aceptación de los criterios de aceptabilidad por parte del dueño del producto, puede ser por la firma de aceptación y aprobación o por medio de un correo electrónico que conste dicho acto.

Fig. 9 Elaboración progresiva de PMI comparado con Desarrollo Ágil



Fuente: [Oficinaproyectosoftware.com](http://Oficinaproyectosoftware.com), 2012

Con respecto a las 10 áreas de conocimiento:

**Gestión de la integración del proyecto.** En esta etapa, implica tomar decisiones referidas a la asignación de recursos, balancear objetivos y manejar las interdependencias entre las áreas de conocimiento. Principalmente se desarrolla el acta de constitución del proyecto y el enunciado del alcance del proyecto de manera preliminar. Prácticamente lo que se busca es integrar la información y los insumos que se estarán utilizando a lo largo de éste.

Se puede determinar que, desde el punto de vista ágil, también se lleva a cabo un proceso de integración pues el maestro SCRUM, que para este fin es el administrador del proyecto, se encarga de reunir la visión general por parte del dueño del producto, reunir el equipo y ajustar todo lo necesario para comenzar el desarrollo del proyecto. (Mulcahy, 2013)

**Gestión del alcance del proyecto.** Incluye aquellos procesos requeridos para garantizar que el proyecto cuente con todo el trabajo necesario para completarlo exitosamente. Su objetivo principal es definir y controlar lo que se incluye y no en el proyecto. En esta etapa, se entregarán la planificación del alcance, la definición del alcance y la creación del enunciado del desglose del trabajo (EDT). La correlación que existe con los procesos de planeación es remarcable en este punto, al planear la gestión del alcance, recolectar requerimientos, la validación de estos se da en los procesos de ejecución y por último el control del alcance.

Sobre este punto, la correspondencia con SCRUM es cuando a partir de la visión del proyecto, se crean los *Product backlogs* y se hace el desglose del trabajo en pequeños componentes hasta llegar a un punto de historias de usuario, también aquí la planeación es esencial ya que de

acuerdo lo que se estableció en el mapa del camino o en el pizarrón de actividades, se comenzará a elaborar y ejecutar el Sprint. También en este punto, se controlan y gestionan los puntos de historias para saber cuántos puntos quedan pendientes y en cuantos Sprints. (Mulcahy, 2013)

**Gestión de la programación del proyecto.** Como se menciona en puntos anteriores, esta etapa se encarga de administrar la finalización del proyecto a tiempo, por medio de: definir las actividades, secuenciar las actividades, estimar los recursos de las actividades, estimar la duración de las actividades, desarrollar el cronograma y controlar el cronograma.

Sobre este punto solo agregar que en un proyecto desarrollado en SCRUM, el tiempo es crucial en la definición de la duración de los Sprints y el número aproximado de los mismos, es aproximado debido a la flexibilidad con que se desarrolla el proyecto, el dinamismo del refinamiento del producto a lo largo de los *Sprints* y la autogestión de los integrantes del equipo SCRUM que trabajan en el desarrollo y calidad del producto a entregar. (Class, 2001)

**Gestión de los costos del proyecto.** Son los procesos involucrados en estimar, presupuestar y controlar los costos del proyecto de modo que se complete el mismo dentro del presupuesto aprobado. En esta área se debe planear el costo de cada tarea específica, de cada actividad a desarrollar y de cada proveedor a utilizar, así como las instalaciones, hardware, servidores, etc. Que de utilizarán a lo largo de la ejecución del proyecto.

De manera muy similar, un proyecto ágil debe realizar una estimación aproximada al inicio. Aquí se debe considerar el esfuerzo en puntos de historia que se estarán utilizando por *Product Backlog*, así como el costo de cada Sprint; al realizarse las iteraciones de manera más corta en tiempo, el ciclo financiero es más rápido, lo cual ayuda a determinar el retorno de inversión de la implementación a desarrollar, así como la posibilidad de ajustarlo. Esta es una de las principales ventajas del desarrollo de un proyecto ágil. (Mulcahy, 2013)

**Gestión de la calidad del proyecto.** Área del conocimiento donde se encuentran los procesos y actividades que determinan responsabilidades, objetivos y políticas de calidad para que el proyecto sea ejecutado satisfactoriamente. Existen diversas herramientas para entregar el producto con una calidad satisfactoria como son: matrices de causa efecto donde se determina la raíz de los problemas, errores o defectos. El diagrama de Pareto que permite asignar un orden de prioridades. El diagrama permite mostrar gráficamente el principio de Pareto (pocos vitales, muchos triviales), es decir, que hay muchos problemas sin importancia frente a unos pocos muy importantes. (Diagrama de Pareto, 2017, Wikipedia.com)

A pesar de que muchos autores catalogan los proyectos ágiles con deficiencia en la calidad (Kerzner, 2013, p187), esto no es del todo cierto, ya que principalmente la calidad de un proyecto en agile se mide iteración por iteración y esta calidad se puede ajustar de acuerdo con las necesidades del cliente o usuario final. En las juntas de refinamiento se ajusta el producto de acuerdo con los criterios de aceptabilidad del dueño del producto. De hecho, lo que describe Kerzner, 2003, p.56 en su libro principios de SCRUM, es que solamente deberá haber dos documentos en un proyecto SCRUM, las historias de usuarios y los criterios de aceptabilidad por parte del dueño del producto.

**Gestión de los recursos del proyecto.** Se consideran los procesos relacionados con la organización, gestión y conducción del equipo del proyecto. Este equipo es conformado por las personas a quienes se les asigna roles y responsabilidades para completar el proyecto.

Parte de los procesos que se deben manejar en este punto son la planificación de los recursos humanos, la adquisición y contrato del equipo del proyecto, también el desarrollo e integración del mismo equipo, y en el proceso de ejecución, la gestión del equipo.

A diferencia de un proyecto PMI, en SCRUM se desarrolla una autogestión de los recursos humanos, los mismos miembros se asignan trabajo y ellos mismos se encargan de completarlo. Si llegara a existir alguna dependencia de algún otro miembro o grupo, el maestro SCRUM se encargará de resolverlo.

La etapa de selección del equipo se lleva a cabo por medio de la participación del dueño del producto y del maestro SCRUM como principales interesados en el proyecto; y esta se basa principalmente en las experiencias de cada miembro o candidato del equipo. (Abrahamsson, 2002)

**Gestión de las comunicaciones del proyecto.** Implementa los procesos necesarios mediante los cuales se busca que la generación, recopilación, distribución, almacenamiento, recuperación y disposición final de la información del proyecto sean adecuados y oportunos.

En el proceso de iniciación, se identifican a los participantes, posteriormente se planifican las comunicaciones, se distribuye la información como parte de los procesos de ejecución, también en esta etapa se gestionan las expectativas de los participantes y en el monitoreo y control se informa el rendimiento del proyecto.

Para los proyectos basados en SCRUM las comunicaciones son uno de los factores clave del éxito de un buen proceso, esto se realiza mediante 5 reuniones periódicas realizadas por cada iteración o Sprint. La primera es la llamada diaria donde la comunicación del gestor o maestro del proyecto es directa con los miembros del equipo SCRUM, y sirve para medir el rendimiento del equipo, así como las suposiciones y restricciones que pudieran estar enfrentando. La segunda reunión es la junta de revisión donde el dueño del producto y el maestro SCRUM se reunirán para discutir el avance del producto, su funcionalidad, el valor para el negocio, entre otros puntos; en esta reunión también está el equipo SCRUM y sirve para que todos los interesados en el proyecto estén en sintonía. La tercera reunión es la junta de refinamiento donde el equipo SCRUM y el dueño del producto, van a discutir sobre las mejoras o cambios que se harán en el producto, así como un reajuste en las prioridades del usuario final. La cuarta reunión es una retrospectiva entre los miembros del equipo y el maestro SCRUM, donde se van a discutir las lecciones aprendidas, los puntos buenos y malos del desarrollo del Sprint. Por último, tenemos la reunión de planeación donde se repiten actividades de estimación y priorización de las historias de usuario y del *Product Backlog* en general.

Por lo antes mencionado, la gestión en las comunicaciones dentro de un proyecto SCRUM, es vital para el desarrollo de esta metodología. (Mulcahy, 2013)

**Gestión de los riesgos del proyecto.** Aquí se desarrollan los procesos relacionados con la planificación de la gestión, la identificación, el análisis cuantitativo y cualitativo, la planificación de respuesta a los riesgos, así como su monitoreo, control y minimización en un proyecto. Cabe mencionar que parte del aseguramiento de la calidad dentro de un proyecto desarrollado en un marco del PMI, se enfoca en esta área del conocimiento de manera muy específica, incluso existen capítulos completos para la gestión de riesgos, tal vez por esta razón los proyectos desarrollados bajo este esquema se consideran más seguros.

Sin embargo, en un ambiente ágil específicamente el desarrollo de proyectos de software en SCRUM, la gestión y control de riesgos se da de manera cotidiana; ya que, por medio de una correcta gestión de las comunicaciones en el momento de la ejecución, los riesgos se mitigan diariamente y se incluyen dentro de la planeación del reléase dentro del *Product Backlog*.

Otro factor a resaltar es que dado que la base de SCRUM es la gestión de proyectos y el trabajo en equipo, CMMI y PMBOK se pueden utilizar para complementar SCRUM cuando sea necesario en las áreas que SCRUM no aborda, que pudieran estar fuera de su alcance, por ejemplo, la gestión de riesgos (que en SCRUM son más reducidos por hacer iteraciones cortas con retroalimentación tanto del cliente como del equipo y por limitar la complejidad que cabe en una iteración), la institucionalización en la empresa de los procesos ágiles, la gestión de proveedores, entre otros. (Abrahamsson, 2002, p288)

**Gestión de las adquisiciones del proyecto.** Abarca los procesos de compra o adquisición de los insumos, bienes y servicios que se requiere para hacer realidad el proyecto. En la planeación se realizan actividades relacionadas a la planificación de las compras y adquisiciones, también la planificación de las contrataciones de proveedores externos o servicios externos, propuestas de vendedores, decisión de hacer o comprar, activos de los procesos de la organización, entre otros. Dentro del proceso de monitoreo y control se realiza un plan para la dirección del proyecto, se gestionan los documentos realizados para las adquisiciones, acuerdos y solicitudes de cambios aprobadas, informes del desempeño del trabajo, así como los datos del desempeño del trabajo realizado por dichos proveedores.

La forma en contratar los servicios y proveedores en SCRUM es muy similar que lo sugerido por el PMBOK, se realiza como base en la primera planeación del primer Sprint, se debe cotizar de acuerdo a la estimación aproximada de trabajo, se realiza un plan para la dirección de proyectos, y todo lo necesario para tener al equipo SCRUM listo para arrancar el Sprint.

Otro punto importante que resaltar son los factores ambientales de la empresa, ya que el maestro SCRUM y el dueño del producto, se pondrán de acuerdo para la realización de las estimaciones y el dueño del producto propondrá un presupuesto al cliente final, el cual aprobará para así arrancar el primer Sprint y los subsecuentes según se desarrolle el proyecto. (Mulcahy, 2013)

**Gestión de los interesados del proyecto.** Desarrolla los procesos que hacen posible la identificación de las personas, grupos u organizaciones que pueden afectar o ser afectados por el proyecto. Se busca conocer y evaluar las expectativas de los interesados y su impacto en el proyecto. Como entradas principales en esta área, se tiene el acta de constitución del proyecto donde se involucran a las personas necesarias para financiar, ejecutar y beneficiarse del desarrollo del mismo. También una vez identificados a todos aquellos que se ven impactados positiva o negativamente por el proyecto, denominados *stakeholders*, se procede a realizar un plan para la comunicación y gestión de estos.

Una vez ejecutándose el proyecto, se realizan registros de incidentes, solicitudes de cambio, actualizaciones del plan, actualizaciones a los documentos del proyecto, entre otros procesos. Por medio del plan de la dirección del proyecto y el registro de los incidentes, se realizan constantemente informes del estatus del proyecto a los involucrados, se realizan las solicitudes de cambio, y el proceso se repite cada vez que sea necesario hasta llegar al cierre del proyecto.

Cuando hablamos de SCRUM, la gestión de los interesados se realiza de manera muy similar a lo anterior descrito, salvo con algunas excepciones; por ejemplo, resalta el dinamismo con el que

se comunican los cambios periódicamente a los usuarios finales o consumidores finales por medio del representante del negocio denominado dueño del producto. También el tener canales de comunicación bastante abiertos por medio de las reuniones diarias de SCRUM, y las de revisión y refinamiento cada cierre de Sprint. Esto además de generar una importante reducción de riesgos y costos (por adaptarse el producto en tiempo y forma a las necesidades del usuario), genera mayor percepción del valor del producto a desarrollar, puesto que el dueño del producto estará satisfecho con lo que se esté desarrollando a lo largo de las iteraciones, se podría comparar a un servicio 'On Demand' donde el usuario paga por lo que usa. (Class, 2001)

### Comparación entre Administración de Proyectos y SCRUM

Como se explica en los puntos anteriores del marco teórico, existe una similitud evidente entre diversas metodologías de administración de proyectos de Software; sin embargo, algunas etapas, elementos, roles, procesos, etc. se adaptan y mejoran conforme se desarrolla y evoluciona la metodología SCRUM.

1. Etapa de iniciación: En esta etapa se menciona la descripción del proyecto a desarrollar, su factibilidad, necesidades, objetivos y estrategias a gran escala.

Prácticamente es el equivalente a un sprint 0 en la metodología SCRUM donde se define la pila del producto, estimaciones pre-trabajadas, mapa de ruta, estimación cero del reléase por medio de puntos de historia, entre otras actividades.

2. Etapa de planeación: Se realizan diversas actividades, la principal es la reunión de planeación del *sprint*. Donde se va a revisar de qué forma se desarrollará la pila del producto dentro de dicho sprint; de qué forma va a trabajar el equipo y todos los interesados del proyecto; así como las estimaciones aproximadas en esfuerzo y tiempo, la velocidad probable de progreso y las fechas de entrega para la revisión.

La planeación en SCRUM se hace en repetidas ocasiones, cada vez que se va a iniciar un *sprint*, y su finalidad es saber que trabajo se pretende realizar a lo largo de éste, los entregables que se estarán haciendo, así como el plan de comunicación y reuniones en este periodo.

Cabe señalar que la primer planeación denominada planeación de *sprint* cero, es de suma importancia ya que en esta se planeará el proyecto completo incluyendo las juntas de planeación por sprint.

3. Etapa de ejecución: Básicamente en esta etapa, se va a desarrollar todo el análisis y construcción de las historias de usuario, conectividades y tramas entre los diferentes medios, las historias de usuario se ajustan y mejoran en esta etapa, y se deben involucrar todos los interesados en el proyecto los cuales ayudarán al equipo SCRUM a mejorar y enfocar las historias de usuario.

También como parte de la ejecución, se da seguimiento al progreso de los *Sprints* y la resolución de problemas y restricciones que surjan.

Una vez iniciado el *Sprint* 1, se llevarán a cabo actividades de monitoreo y control; la etapa de ejecución dura el tiempo necesario para completar los *Sprints* planeados y completar el producto a entregar en 100% con la calidad requerida.

4. Etapa de monitoreo y control: Esta etapa inicia desde la fase de ejecución ya que es cíclica de acuerdo al inicio y termino de cada *sprint*, principalmente se llevan a cabo reuniones de monitoreo que se denominan juntas de revisión y refinamiento, donde el equipo SCRUM presenta el progreso por *sprint*, se discuten temas de relevancia sobre la administración del equipo y de la pila

del producto y prácticamente todos los interesados deberán atender estas reuniones; desde el dueño del producto, el maestro *SCRUM*, el equipo, calidad, etc.

El resultado de esta etapa se repite en cada junta de planeación nueva para cada *sprint* y desemboca en la etapa de cierre cuando ya se han completado todas las iteraciones del proyecto.

5. Etapa de cierre: En esta etapa se va a presentar el resultado final del producto ya con todas las modificaciones remarcadas previamente, en algunas ocasiones también se considera cierre del proyecto cuando la implementación ya se llevó a cabo y se realizaron pruebas del funcionamiento para asegurar la calidad de los entregables, la etapa de garantía normalmente es la duración de un *sprint* 2-3 semanas.

En esta etapa se crean lecciones aprendidas para futuros proyectos y se cierran los contratos pendientes con proveedores de recursos humanos.

En teoría la forma en que se cierran los proyectos tanto en una metodología de cascada como en una metodología ágil como *SCRUM*, es muy similar sin embargo en esta última es más dinámico ya que cada *sprint* se lleva a cabo una reunión de refinamiento y revisión del progreso del producto, se cierra cada *sprint* con estas reuniones, pero en la última etapa se colectan las reuniones aprendidas y se liquidan los recursos contratados.

La implementación de Software en *SCRUM* se realizará en todas sus fases, como se grafica en la figura 10. Primeramente, se recibe información por parte de clientes, equipos, gerentes y/o ejecutivos interesados en el proyecto, esta información es proporcionada a un “dueño del producto” (*Product Owner*) el cual se encargará de analizar dicha información, clarificar en caso de dudas con el negocio, una vez definido el alcance, se encargará de priorizar el trabajo a realizar a lo largo de los *Sprints* y también como parte de su función, revisar el avance del producto a desarrollar. Otra de las funciones del dueño del producto es crear la Pila del Producto o *Product Backlog*, el cual está compuesto de historias épicas, características e historias de usuario, todas ellas contienen el alcance completo del proyecto.

Con la Pila del Producto, el dueño del producto y el equipo *SCRUM* se lleva a cabo una reunión de planificación del *Sprint* siguiente donde, se desmenuzará las historias épicas en características o *features* e historias de usuario o *user stories* las cuales son estimadas por el equipo *SCRUM* y priorizadas por el dueño del producto, el junto con el maestro *SCRUM*, definirán la carga de trabajo, de acuerdo con las estimaciones, dentro del *Sprint*.

Regularmente el *Sprint* dura de una a cuatro semanas, cabe señalar que durante ese periodo no se pueden realizar cambios de las tareas asignadas; el maestro *SCRUM* será el responsable de darle seguimiento y cumplimiento a las historias de usuario, también se encarga de resolver cualquier clase de impedimentos y restricciones que enfrente el equipo durante el periodo de *Sprint*. Para ayudarse a darle un seguimiento cercano y personalizado al equipo, realiza diariamente una reunión denominada “Junta diaria y actualización de artefactos” donde el equipo deberá contestar tres preguntas básicas: 1. ¿Qué actividad hiciste ayer?, 2. ¿Qué actividad vas a realizar hoy? Y 3. ¿Tienes algún impedimento para continuar tu trabajo?

Posteriormente una vez finalizado el *Sprint*, se lleva a cabo una reunión de revisión del producto y avance de éste; donde participarán tanto el dueño del producto, el maestro y el equipo *SCRUM*. En cada revisión se validará que la pieza entregada tenga un valor y un criterio de aceptabilidad por parte del dueño del producto, por lo que el dará su visto bueno del progreso

El objetivo de tener interacciones cortas en tiempo y bien definidas en alcance es principalmente mostrar un incremento asequible del producto final el cual se parcializa en varios entregables, con dicho incremento se puede palpar desde el primer *Sprint* valor tangible para el negocio y así

encaminar los que futuros apegándose a los requerimientos iniciales o adaptándose a los cambios solicitados en las revisiones.

Por último, como parte del proceso de SCRUM, se lleva a cabo una reunión de retrospectiva con el objetivo de valorar y corregir el desempeño del equipo, del maestro SCRUM e incluso del dueño del producto. Es aquí donde todos los interesados en el proyecto aclaran las diferencias, recalcan los hallazgos y definen la estructura del personal para el siguiente *Sprint*.

Fig. 10 Proceso a seguir en SCRUM para el desarrollo de Software



Fuente: Elaboración propia

## 7. 7 Análisis final

La gestión de proyectos no es tarea fácil ya que en realidad no existe una metodología universal para hacer frente con éxito a cualquier proyecto y específicamente de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc. Históricamente, las metodologías tradicionales han intentado abordar la mayor cantidad de situaciones de contexto del proyecto, exigiendo un esfuerzo considerable para ser adaptadas, sobre todo en proyectos pequeños y con requisitos muy cambiantes. Las metodologías ágiles ofrecen una solución casi a medida para una gran cantidad de proyectos que tienen estas características. Una de las cualidades más destacables en una metodología ágil es su sencillez, tanto en su aprendizaje como en su aplicación, reduciéndose así los costos de implantación en un equipo de desarrollo. Esto ha llevado hacia un interés creciente en las metodologías ágiles y sobre todo en proyectos de desarrollo de Software donde los requerimientos son adaptables a las necesidades del usuario final.

Muchas compañías y gestores especialistas en administración de proyectos (ver anexos), opinan que en su experiencia nunca han visto gestionarse un producto con una metodología 100% pura, incluso en PMBOK menciona que, si bien es un libro especializado en buenas prácticas de administración de proyectos de cualquier característica, no es una fórmula para elaborar proyectos exitosos y con grandes retornos de inversión asegurados; se describe a sí mismo como un marco de referencia de buenas prácticas, e incluso sugiere complementarse con alguna otra metodología de desarrollo, en el caso particular de Software, con métodos ágiles o lineales.

Por esa razón al realizar un ajuste de correlaciones entre SCRUM y los 5 grupos de procesos, 10 áreas de conocimiento y 49 procesos descritos en el PMBOK, se logran identificar elementos importantes que sirven de guía para una mejor gestión de proyectos de desarrollo de Software.

A pesar de que varios autores hablan de las bondades y beneficios del uso de las metodologías ágiles, estas también tienen ciertas restricciones que se mencionan a continuación:

- Para que un grupo de desarrollo adopte una metodología ágil debe poseer experiencia trabajando con metodologías tradicionales, ya que la experiencia es la que predomina en los momentos cruciales del proyecto, además debe tener la capacidad de ser equipos auto-gestionados, altamente motivados y con gran innovación.
- Las metodologías ágiles permiten disminuir costos y brindar flexibilidad a los proyectos de software sin embargo la incertidumbre está presente.
- Las metodologías ágiles se deberían aplicar en proyectos donde exista mucha incertidumbre donde el entorno es volátil, donde los requisitos no se conocen con exactitud, mientras que las metodologías tradicionales obligan al cliente a tomar las decisiones al inicio del proyecto.

En la siguiente tabla (tabla 1), se describen 8 elementos que con base al juicio de expertos en materia de gestión de proyectos ágiles (ver anexos), se pueden considerar claves si se compara el método SCRUM contra el método de cascada:

Dichos expertos en el tema consideran que un enfoque SCRUM se caracteriza por tener poca documentación, un equipo reducido de integrantes, realizar cambios constantes enfocados a las necesidades del usuario final, cuenta con interacciones en periodos cortos, el costo y el alcance se enfocan en las necesidades del cliente o del dueño del producto, entre otras características.

**Tabla. 1 elementos del desarrollo de proyectos de software**

CARACTERISTICA	SCRUM	CASCADA
Enfoque	Ágil	Lineal secuencial
Documentación	Mínima (3 máximo)	8 por lo menos
Equipo	Máximo 6	Mínimo 10
Producto	Enfocado al cliente final	Alineado al plan inicial
Cambios	Dinámicos y constantes	Lentos y difíciles
Tiempos	Iteraciones cortas (4 semanas)	Releases largos (meses)
Costo	Enfocado a las necesidades del cliente	Enfocado al contrato inicial y cambios costo extra
Alcance	Adaptable	Fijo

Fuente: Elaboración propia

La tabla siguiente 2 muestra las ventajas y desventajas de la utilización de SCRUM, así como un enfoque general de gestión de proyectos ágiles. Hope toma referencia de algunos casos de éxito de la implementación de SCRUM como metodología, incluso modificando términos para ejecutarla. Algunos casos de éxito como Spotify o Amazon; marcan una tendencia al alza en el ámbito de Tecnologías de Información. Para la adopción cada vez mayor de SCRUM como metodología ágil. (Hope, 2003,)

**Tabla. 2 ventajas y desventajas de SCRUM**

VENTAJAS	DESVENTAJAS
----------	-------------

Normalmente, si se busca la entrega en tiempo y forma de las piezas del producto; dichas piezas consideradas funcionales, se van desarrollando a lo largo de los <i>Sprints</i> , y siempre se busca generar un valor para que el cliente pueda percibirlo.	Si algún miembro del equipo SCRUM no tiene suficiente conocimiento del uso de la metodología, solo buscará su beneficio propio y no se enfocará en la autogestión, esperando recibir instrucciones precisas de acciones.
Al tener iteraciones cortas, la planeación es constante y el producto se ajusta a las necesidades del cliente; lo que se traduce a menores costos de re-trabajo y estimaciones de esfuerzo.	Se puede llegar a crear un ambiente de anarquía si el maestro SCRUM no tiene suficiente control del equipo, incluso la colaboración entre miembros se puede ver afectada.
En las juntas de refinamiento del producto, el dueño del mismo puede priorizar las historias de usuario las veces que sea necesario, incluso puede cancelar requerimientos que considere ya no cumplen con las necesidades del cliente.	Demasiadas reuniones para poco avance, a veces es muy cansado y estresante reunirse demasiadas veces por el mismo tema, algunos van perdiendo el interés en el proyecto.
Se involucra desde un principio y se da un rol a todos los interesados en el proyecto.	Las escalaciones a nivel recursos humanos son muy escasas, ya que uno de los principios de agile es la colaboración de las personas de forma voluntaria, esto se traduce a retrasos si algún miembro del equipo no trabaja al ritmo del resto.
El proceso de comunicaciones es casi siempre transparente y fluido, a pesar de que tener varias reuniones en corto tiempo, puede considerarse desventaja, el hecho de tener siempre un canal de comunicación abierto entre los interesados es una gran ventaja.	La zona horaria se convierte en una desventaja, ya que muchos de los proyectos de desarrollo de software tienen su manufactura en otros países, y SCRUM estipula que es mucho más conveniente tener al equipo, al dueño de producto y al maestro SCRUM en una misma oficina.
Los riesgos se mitigan de manera más fácil ya que cada Sprint se hace una evaluación de suposiciones y restricciones y se ajusta el <i>Product Backlog</i> para hacer el proyecto más eficiente.	Muchas veces la gestión de las adquisiciones se realiza en una caja negra, lo que significa que en muchos casos se desconocen los contratos y acuerdos entre las partes involucradas.

Fuente: Hope, 2003

## 8. Estrategia metodológica.

Se pretende realizar la comparación práctica de la gestión y administración de proyectos de desarrollo de Software bajo un esquema metodológico de cascada contra un esquema ágil en SCRUM bajo los principios propuestos por el PMBOK en su sexta edición; como base y guía se utiliza la siguiente hoja de conceptos y enunciado del alcance del proyecto.

La finalidad de este trabajo es buscar crear un punto de referencia que sirva de sinergia para ajustar las etapas propuestas por el PMBOK de un desarrollo de ciclo de vida del software deno-

minado *Software Development Life Cycle* (SDLC) en cascada hacia un flujo de gestión ágil denominado *SCRUM*, el cual es expuesto y explicado a detalle en forma progresiva de acuerdo a la adopción del mismo de principio a fin.

Para ello se expone de manera teórico-práctica, la migración de metodología de gestión de proyectos de una aplicación Web basada en cascada hacia un enfoque ágil de *SCRUM*, con la finalidad de mostrar paso a paso y con base a la experiencia llevada a cabo, de qué forma se substituye el proceso, los entregables, las reuniones y otros elementos importantes en dicha gestión.

La estrategia metodológica que se seguirá para la elaboración del TOG implica las etapas de la metodología, descritas anteriormente:

1. Iniciación
2. Planeación
3. Ejecución
4. Monitoreo y control
5. Cierre

Como punto de partida se desarrolló la hoja de concepto y el enunciado del alcance del proyecto, tomando en cuenta un proyecto desarrollado bajo el marco metodológico de cascada donde, en resumen, se registran los requerimientos de negocio, se hace el análisis de las necesidades del mercado, posteriormente se realiza la construcción del código y una vez realizada esta última, se comienzan a hacer pruebas para posteriormente dar soporte post implementación.

## 8. 1 Hoja de concepto

La hoja de conceptos que se presenta a continuación es la elaborada para el TOG En el caso de la propuesta de *SCRUM*, se desarrolla en la etapa previa a la planeación de la metodología con la que se desarrollará el proyecto.

En los próximos puntos se pretende exponer la forma en que se trabaja la gestión y administración de proyectos en un ambiente ágil, con los fundamentos metodológicos de *SCRUM*, y resaltando algunas ventajas y desventajas en la adopción de dicha metodología.

### 8.1.1 Definición mercadológica

Diseño de un esquema de gestión de proyectos por medio del cual se incorporen en las fases de administración de proyectos propuestas por el PMBOK las fases propuestas por la metodología *Scrum* en el desarrollo de un software.

La aportación principal de este documento es resaltar las ventajas y desventajas en la migración y adopción de una metodología ágil para la gestión de proyectos de desarrollo de software, por medio de la exposición teórico-práctica de migraciones realizadas.

### 8.1.2 Mercado(s) meta/ características

Gestores y administradores de proyectos de Software, así como todos aquellos profesionales interesados en temas de gestión de proyectos de Software los cuales busquen un marco de referencia entre la metodología de cascada y la metodología ágil de *Scrum* bajo los principios estructurados del PMBOK.

#### 8.1.3 Necesidades básicas a satisfacer

- Reducción de incertidumbre con respecto a la metodología *SCRUM* como herramienta de gestión de proyectos diversos.
- Mapeo estructural de los procesos y áreas de conocimiento del PMBOK embebido en las fases propuestas por *SCRUM*.
- Propuesta de proyectos gestionados *on-demand* ahorrando tiempo, dinero y esfuerzo.

#### 8.1.4 Variables de competitividad

- Tiempo de desarrollo de proyectos.
- Nómina de personal enfocada a un solo proyecto.
- Precisión en el mapeo de planeación de actividades.
- Calidad del servicio de gestión de proyectos.
- Precio del desarrollo del proyecto.
- Facilidad comparativa y de referencia entre metodologías ágiles y de cascada.

#### 8.1.5 Características funcionales

1. Diagnóstico de gestión de proyectos.
  - 1.1 Mapeo de fase actual de proyecto a analizar.
  - 1.2 Identificación del proceso y área del conocimiento de las tareas a realizar.
  - 1.3 Utilización práctica del uso de metodologías ágiles y como aplicar *SCRUM* a los proyectos.
2. Adaptación del proceso de gestión de proyectos.
  - 2.1 Permite identificar en la tabla de gestión, las opciones de la metodología *SCRUM* para la gestión del proyecto analizado.
  - 2.2 Identificar ventajas y desventajas de *SCRUM* en comparación con la metodología de cascada en ciertas tareas de gestión.
  - 2.3 Crear un vínculo comparativo entre la gestión de proyectos en cascada contra el uso y utilización de *SCRUM* para definir, desarrollar y monitorear proyectos de software.
3. Adaptabilidad de cambios por interacciones de acuerdo a las fases de desarrollo de proyectos de software.
  - 3.1 La metodología *SCRUM* acepta cambios cada revisión en cada sprint. Dichos cambios se generan con base a las reuniones de revisión y refinamiento periódicamente en un máximo de 3 semanas. A diferencia de un proyecto desarrollado en cascada que requiere documentos de control de cambios denominados CR (*Change Request*).
  - 3.2 Mostrar de forma sistemática como se emparejan las fases de refinamiento en un ambiente de desarrollo *SCRUM* versus un ambiente PMP para el manejo de cambios.

#### 8.1.6 Hábitos de consumo actuales

Los gestores y administradores de proyectos se enfocan en el desarrollo integral de proyectos de inicio a fin, ellos buscan tener herramientas funcionales para dicha gestión. Sin embargo, en ocasiones debido a la falta de conocimiento del manejo, control e incluso amplitud de tendencias metodológicas, específicamente hablando sobre administración de proyectos; eligen metodologías que normalmente utilizan en el día a día, dejando a un lado nuevas opciones que pudieran resultar funcionales e incluso más apropiadas con base a las tendencias de la industria.

En otras palabras, algunos de los *Project Managers* tienen la idea que la mejor forma de gestionar proyectos grandes y complejos es basándose en metodologías progresivas con características específicas como documentación abundante, ciclos cerrados y bien definidos desde un inicio, etc. Mientras que proyectos pequeños con bajo riesgo se planean desarrollar con métodos ágiles, debido a su flexibilidad y rapidez. En el peor de los escenarios, gestionan proyectos con una

mezcla ineficiente de ambas tendencias metodológicas limitando las bondades y características positivas de cada una.

Por ejemplo, proyectos de mantenimiento o mejoras continuas de alguna aplicación, usualmente se desarrollan en un marco metodológico ágil como *SCRUM* ya que normalmente se compone de implementaciones cortas y no requiere complejidad en las configuraciones de sistemas impactados; mientras que un proyecto de dimensiones mayores en termino de esfuerzo con elementos más complejos, dependencias de diversos sistemas, bases de datos, etc. Es usual que se gestione con los principios del PMBOK y bajo un esquema progresivo de cascada por el control de documentos, cumplimiento de requerimientos, control de cambios y aprobaciones de cada fase hasta la aprobación final pos-implementación.

Cabe señalar que esto no quiere decir que la mezcla de ambas tendencias sea mala o ineficiente, sino la falta de conocimiento del alcance de estas; pues se ha comprobado en casos de éxito descritos con anterioridad, la eficiencia de la administración de proyectos con principios de *SCRUM*, aunque estos sean de gran tamaño, y por el contrario fracasos en proyectos desarrollados con un marco teórico del PMBOK de tamaño pequeño.

Otro punto importante para resaltar es que la propuesta metodológica del PMI en su última edición, contempla ya las tendencias metodológicas ágiles, sin limitar la forma en que se lleve a cabo la gestión, el PMBOK sexta edición, desglosa por fases y áreas de conocimiento la importancia de tener en cuenta un enfoque ágil y todos sus respectivos elementos de administración.

#### 8.1.7 Vida funcional prevista o ciclo de vida

La duración de la propuesta es esencialmente de 2 años, sin embargo, debido a las tendencias en la industria de Software, las metodologías ágiles han adquirido mayor popularidad y se prevén incrementos importantes en la adopción de la misma, en diferentes industrias comerciales.

En el caso específico de este documento, se muestra dentro del desarrollo de la propuesta y sus anexos, la variabilidad del tiempo que toma la migración en la gestión de proyectos; así como algunos ejemplos reales del desarrollo de dicha migración con diferentes productos. Cabe resaltar que el ciclo de vida de un proyecto gestionado en Scrum, varía del número de *Sprints* planeados y desarrollados.

#### 8.1.8 Restricciones

**Restricciones tecnológicas:** Pocos productos o herramientas de gestión donde se lleve a cabo un análisis minucioso y comparativo de las tareas asignadas del proyecto en un marco referencial del PMI con respecto a la propuesta de gestión de la metodología *SCRUM*.

**Restricciones económicas:** En la primera fase del proyecto donde se busca explicar la transición de metodologías, no existe una restricción económica remarcable; sin embargo, en fases posteriores de implementación y publicidad, existe restricciones de presupuesto y horas hombre.

**Restricciones Legales:** En temas de auditoría, el gestionar proyectos basados en *SCRUM*, muchas veces se vuelve complicado ya que la metodología sugiere la menor utilización de documentación para así agilizar el proceso, pero se le aplican auditorias enfocadas a métodos convencionales donde se genera un documento por fase.

Restricciones comerciales: Posicionamiento de metodologías y herramientas de gestión basadas en el PMI, incluso en temas educativos, la adquisición de una certificación en PMP es mucho más cara que una certificación como *SCRUM Master* de la *SCRUM Alliance*.

#### 8.1.9 Productos en competencia

Productos similares: Muchos autores proponen modelos de gestión de proyectos con un enfoque ágil, otros con una mezcla de cascada con ágil.

Productos alternativos: Esquemas de gestión de proyectos ligeros en desarrollo ágil. Algunos modelos complementarios como Kanban ~~KANBAN~~ que describen la organización de actividades por medio de una pizarra con elementos adicionales como, por ejemplo, el número de tareas que se pueden ejecutar por iteración.

Otra metodología es *Xtream programing*, donde el desarrollo se los scripts de hacen en pares; un desarrollador y un *tester* que interactúan, reciben y generan retroalimentación inmediata para ir completando piezas del producto final con valor intrínseco.

#### 8.1.10 Aspectos tecnológicos

##### 8.1.10.1 Innovación tecnológica

- Tecnología actual: Gestión de proyectos con fundamentos estructurales de cascada en fases de procesos y áreas de conocimientos propuestos por el PMBOK.
- Tecnología buscada: Vinculo de migración metodológica hacia *SCRUM* como marco referencial por medio de un esquema de mapeo de actividades.
- Hipótesis: La mayoría de los proyectos de diversas características pueden ser gestionados por medio de principios de *SCRUM* y metodologías ágiles de administración.

##### 8.1.10.2 Transferencia tecnológica

Se pretende esquematizar, ejemplificar y exponer la migración teórico-práctica del desarrollo de proyectos de Software desde una perspectiva metodológica de cascada hacia un marco teórico ágil denominado *SCRUM* bajo los parámetros expuestos por el PMBOK.

Se aplicará el ciclo completo de *SCRUM*, destacando las limitantes, lecciones aprendidas, elementos desarrollados y otros factores importantes en dicha migración.

Por medio de un esquema de mapeo se identificará en que área y proceso se encuentra el proyecto a migrar, haciendo la migración de forma ágil y bien estructurada para cubrir cualquier punto o característica específica de la fase en la que se realice.

Esta migración tecnológica se hace con herramientas específicas como RTC (*Rational Team Concert*) o *JIRA*, donde se crean historias épicas equivalentes a requerimientos generales por parte del usuario final, posteriormente se crean "*features*" (granulaciones del requerimiento general con características más específicas) y finalmente se crean historias de usuario, equivalentes a tareas específicas en una metodología de cascada, que son requerimientos al nivel granular más bajo.; más adelante se explican dichas herramientas.

##### 8.1.11 Ángulos de diferenciación

- Disponibilidad de información completa y útil para la toma de decisiones con respecto a la metodología de gestión de proyectos a utilizar.
- Registro de cambios y adaptaciones del proyecto desarrollado en un marco referencial *SCRUM*.
- Mapeo de tareas (historias de usuario) por fases dependiendo la etapa del proyecto.

- Comparativa de las metodologías SCRUM versus cascada para identificar los pros y contras de ambas en la administración de un proyecto de tecnologías de información.

#### 8.1.12 Productos complementarios

- Esquema de correlación entre las 10 áreas del conocimiento y los 49 procesos del PMBOK con las fases de SCRUM y desarrollo de *Sprints* a lo largo del proyecto.
- Lecciones aprendidas a lo largo de la migración de proyectos de desarrollo de Software desarrollados en cascada hacia SCRUM.
- Descripción detallada de las fases de un proyecto ágil y el equivalente en un proyecto en cascada.
- Información relacionada a la diferencia entre la gestión de proyectos tradicionales y métodos alternativos.

#### 8.1.13 Propuesta de valor

- Enfoque más claro y conciso sobre la utilización de la metodología de administración de proyectos SCRUM; Este enfoque se pretende enriquecer por medio de ejemplos de casos de éxito de adaptación de SCRUM como herramienta de gestión de proyectos.
- Esquema de mapeo para identificar la fase del proyecto en la que se está trabajando desde 2 perspectivas, enfoque ágil por medio de SCRUM y enfoque tradicional o de cascada por medio del PMBOK.

Dicho esquema funcionará como una herramienta de referencia para identificar en que área de conocimiento, fase y proceso se encuentra el proyecto y que oportunidades de mejora tiene.

#### 8.1.14 Costos

- En el desarrollo del esquema de mapeo se invertirán horas hombre con la finalidad de completar el análisis y las pruebas de la migración.

## 8.2 Enunciado del alcance

### 8.2.1 Justificación del proyecto:

El objetivo primordial de este proyecto es hacer una comparativa de las fases de desarrollo de la gestión de proyectos de software, se busca principalmente exponer los elementos del uso de la metodología de cascada aunado al marco teórico del PMBOK en comparación a las etapas de un proyecto desarrollado bajo un esquema ágil, específicamente en la metodología *SCRUM*.

De acuerdo a las tendencias de la industria, expuestas en el marco teórico de este trabajo; nace la necesidad del mercado y de la industria de TI para la administración y gestión de proyectos de desarrollo de software de una manera más simple, barata y ágil donde las restricciones administrativas se disminuyan a medida de lo posible para pasar a una gestión de proyectos enfocada a satisfacer las necesidades de los clientes y a reorganizar los requisitos del producto, para poder entregar valor al cliente desde las primeras semanas de desarrollo.

También como parte del análisis arriba descrito, se pretende desarrollar un diagrama donde contenga las fases de *SCRUM* y los 49 procesos propuestos por el PMBOK en su sexta edición. Con este diagrama se ejemplificará una migración de un proyecto de desarrollo de Software estructurado en metodología de cascada hacia *SCRUM*.

### 8.2.2 Alcance del proyecto:

El contenido del trabajo especificará, de acuerdo con las etapas de administración de proyectos del PMBOK, la vinculación de las etapas propuestas por la metodología *SCRUM*. En otras palabras, va a definir y comparar cada una de las fases de ambas metodologías, asumiendo que el PMBOK no delimita a ninguna metodología, pero refiere a un marco teórico tradicional como el de cascada.

También como parte de los entregables del proyecto, se va a elaborar un caso de la vida real para ejemplificar el desarrollo en ambos escenarios descritos anteriormente; así como un diagrama de flujo que contenga las fases de ambos marcos referenciales.

Esta propuesta deberá tener las siguientes características:

1. Contar con un contenido claro y bien definido de lo que se busca exponer en este trabajo.
2. Una comparativa objetiva de las metodologías de administración de proyectos de Software.
3. Ejemplificar con un caso real de una migración de metodología de cascada a *SCRUM*.
4. Un diagrama donde se podrán identificar la fase, etapa y elementos de la gestión de proyectos con ambas propuestas (PMBOK cascada y *SCRUM*).

### 8.2.3 Metodología del proyecto

La metodología por utilizar en la migración es agile *SCRUM*, la cual se caracteriza por:

- Cumplimiento de expectativas
- Flexibilidad de cambios
- Maximización del retorno de inversión
- Reducción del tiempo
- Mayor productividad
- Predicciones de tiempos
- Reducción de riesgos

El desarrollo del proyecto se pretende implementar con las siguientes etapas técnicas:

- Investigación bibliográfica y del estado del arte.
- Búsqueda en la web de trabajos similares.
- Compilar, examinar, detallar y comprobar requerimientos.
- Gestionar requerimientos.
- Diseñar diagrama de flujo de la migración de gestión de proyectos.
- Diseñar diagrama inclusivo de los procesos y subprocesos del PMBOK con las fases de *SCRUM* y Cascada.
- Aplicar la metodología *SCRUM* en un proyecto de desarrollo de software para una aplicación bancaria en línea de un menú y submenú de pagos y transferencias.
- Desarrollo de lecciones aprendidas en el ejemplo del módulo de inversiones tomado de la migración de Cascada a *SCRUM*.

Los entregables se irán desarrollando en un lapso no mayor a 3 semanas, este periodo se denominará Sprint, y al final de cada Sprint se va a entregar una pequeña parte del producto final, pero con valor para el usuario y se hará una revisión del mismo, también se llevara a cabo una retrospectiva del trabajo desarrollado así como un refinamiento en caso necesario.

Las actividades por desarrollar son:

Actividad	Descripción
1. Análisis de los 49 subprocesos contenidos en el PMBOK.	Se pretende desarrollar un análisis minucioso sobre la utilización y enfoque de los 49 procesos de administración de proyectos contenidos en el PMBOK sexta edición.
2. Análisis de los elementos de SCRUM	Desarrollo minucioso de las fases y elementos de SCRUM descritos en el marco teórico de este trabajo.
3. Desarrollo de un diagrama SCRUM-PMP	Elaboración de un diagrama donde se incorporen algunos de los 49 procesos del PMBOK en un esquema de SCRUM.
4. Explicar un caso real sobre la migración metodológica en un proyecto de software de "cascada" a "SCRUM".	Ejemplificar la migración de la gestión de un proyecto de software con un caso real con el objetivo de mostrar la propuesta objeto de este trabajo.

#### 8.2.4 Productos a entregar

- Diagrama de flujo de la migración de metodologías para la gestión de proyectos de software.
- Diagrama de inclusión de los procesos y subprocesos del PMBOK con las fases de SCRUM y el método de Cascada.
- Libreta de lecciones aprendidas durante la migración, limitaciones, condiciones mínimas necesarias (capacitación).

#### 8.2.5 Delimitación

Ningún elemento adicional al propuesto, el cual tiene como fin ejemplificar la utilización de la metodología SCRUM como una alternativa viable de gestión de proyectos. Tampoco se considera algún entregable adicional al esquema de mapeo arriba descrito.

Cualquier otro producto o proyecto que no esté descrito en el alcance de este trabajo, tampoco será considerado.

### 8.3 Desarrollo de Propuesta

La estructura de los proyectos de desarrollo de software en la industria bancaria es, de cierta manera complejos, por diversos factores.

Uno de los principales es la complejidad de documentación que se requiere por temas de auditoría, otro es el riesgo y los parámetros de seguridad que se manejan. Por tal motivo, dichos proyectos se manejan bajo una metodología secuencial de cascada, atendiendo los principios del PMBOK, sin embargo y debido a las tendencias globales, se optó por migrar la gestión hacia SCRUM.

En la práctica, en mi opinión encuentro ciertas similitudes en ambas metodologías de gestión de proyectos en el sentido de las fases, pero con diferencia en la forma de progreso y el tiempo en la entrega de valor.

A continuación, se describen las fases estructuradas por el PMBOK y la forma de adopción, ventajas y desventajas en la gestión bajo la metodología de cascada y SCRUM.

### 8.3.1 Iniciación:

Específicamente hablando de la fase de iniciación de un proyecto de desarrollo de Software, desarrollado bajo la metodología de cascada, el proyecto era sobre un menú y submenú de una aplicación bancaria en línea. Sobre dicha metodología, se consideraba una fase de requerimientos donde se realizaba un análisis general de las necesidades del cliente para determinar las características del software a desarrollar, en este caso los dueños de la aplicación en línea eran los responsables de la entrega del enunciado del alcance del proyecto, el acta de constitución del proyecto y el requerimiento de negocio (ejemplos en anexos 2, 3 y 4). Cabe mencionar que en esta fase se debía poner especial atención en la definición del producto a entregar, ya que bajo este esquema no se podían añadir nuevos requerimientos a mitad del proceso de desarrollo.

Ejemplo del Enunciado del alcance que se utilizaba en el proyecto: Se puede ver en la Figura 11.

**Fig. 11 Enunciado del alcance**

**3.1 Descripción/Objetivos**  
 XXXX

**Beneficios para el cliente:**

**Quantitativos**

Requerimientos	Actual
Cuentas actuales	nnn
Promedio de volume de cuentas	nnn
Volume de cuentas por día y por año	nnn
Horas de operación	Desde nn a nn
Días de operación	xxx
Tiempo de respuesta/Seg	nnn
Lugares o cuentas distribución geográfica	México
Tipo de COB	nnn

**Cualitativos**

- P

**Beneficios internos**

- O

**Impactos si no se aprueba el proyecto**

- I

**3.2 Justificación**  
 XXXX

**3.3 Suposiciones**  
 XXX

**3.4 Restricciones**  
 XXX

**3.5 Dependencias**  
 XXX

**3.6 Criterios de Aceptabilidad del Producto**  
 XXX

**3.7 Portafolios Impactados**  
 XXX

**3.8 Detalle del Alcance**  
 XXX

Fuera del Alcance  
 XXX

Fuente: Elaboración propia

También en esta fase, se desarrollaba una estimación general de costos del producto a desarrollar, los riesgos probables, plazos, suposiciones y restricciones que enfrentará el proyecto; basada en la descripción de los requisitos del software los cuales cumplen los parámetros y necesidades del cliente en cuestión.

Se hacía un análisis minucioso de los interesados y posibles involucrados en el proyecto, por medio de una matriz con datos de contacto, escalaciones, roles y responsabilidades.

En la Figura 12 se puede ver un ejemplo de Plan de Comunicación.

Fig. 12 Plan de Comunicaciones

Fases	Descripción	Método de entrega	Frecuencia	Responsables	Audiencia	Principales entregables
Requisitos Diseño Implementación	Análisis de las necesidades del cliente Etapa donde se describe estructura interna del Software	Web Ex, Conferencias telefónicas, Cara a cara	Una vez por semana Quincenal	Administrador del proyecto	Equipo técnico, Negocio, bases de datos, etc.	Documento de requerimientos de negocio Documento de requerimientos funcionales Código desarrollado
Verificación Mantenimiento	Se verifican los componentes del sistema y que cumplan con los requisitos.	Web Ex, Conferencias telefónicas, Cara a cara.	De acuerdo a las necesidades	El negocio.	Equipo técnico, clientes.	Casos de pruebas, matriz de incidentes.

Fuente: Elaboración propia

Cabe mencionar que en esta fase, se llevaban a cabo diversas reuniones con el cliente, para obtener estimaciones preliminares denominadas E0, en dichas reuniones se revisaba el alcance a un nivel general, se resolvían dudas genéricas por parte de los equipos técnicos (desarrollo y QA), se validaban los posibles impactos a sistemas de *Backend* y la forma en que se harían las conectividades, los protocolos de comunicación y todas aquellas configuraciones técnicas y administrativas, necesarias para la implementación de la aplicación.

Un punto importante que se validaba en los documentos de alcance era la descripción y objetivos del proyecto; por ejemplo, se tenía un apartado para descripción de los beneficios de negocio en caso de implementación, métodos para medir los beneficios, regiones donde impactan los beneficios, criterios de éxito y aceptación de factores críticos, entre otros.

Es común, por lo menos en la industria de donde se desarrolla este TOG, que la mayoría de los interesados en el producto firmen o estén de acuerdo con lo plasmado en los diversos documentos que se desarrollan en la fase de inicio, para esto tanto el documento del alcance, el acta constitutiva del proyecto, el enunciado del alcance, entre otros; deben tener una lista de aprobadores y revisores que otorgarán el visto bueno para que se pueda pasar a la siguiente fase.

Un ejemplo de lista de aprobadores y revisores se puede ver en la figura 13:

Fig. 13 Información General del Proyecto

Lista de Revisores

IDENTIFICADOR	ROL	NOMBRE
XyxxX123	Gerente de Producto	Nombre completo
XyxxX123	Representante de Negocio	Nombre completo
XyxxX123	Gerente de Proyecto	Nombre completo

Lista de Aprobadores

IDENTIFICADOR	ROL	NOMBRE
XyxxX123	Gerente Regional de Proyecto	Nombre completo
XyxxX123	Gerente de Producto de Negocio	Nombre completo
XyxxX123	Patrocinado	Nombre completo

Fuente: Elaboración propia

Como se explica en los puntos anteriores, la industria de tecnologías de información comenzó a sufrir diversos cambios, los cuales impulsaron a los altos directivos a tomar decisiones con respecto a la forma de gestionar los proyectos, las herramientas utilizadas y el enfoque que se tomaría para estar en sincronía con la industria. Por esta razón, se comenzó a migrar la gestión y desarrollo de proyectos de software hacia un enfoque ágil.

Cuando comenzó la migración a la metodología de Scrum, fue de forma paulatina, ya que primero la fase de inicio se realizan reuniones iniciales denominadas *Scrum cero* donde prácticamente se revisan los requisitos de negocio de forma general, se hacen estimaciones preliminares del número de *Sprints* probables, la duración de cada uno y los posibles equipos involucrados que afectarán el producto final. También se estiman recursos y presupuesto disponible en relación horas/hombre.

Es importante señalar que depende la criticidad en sentido de priorización e impacto para el negocio, en la misma fase inicial de Scrum se comienza a desarrollar en paralelo la Pila del producto ya que, de esta forma en la fase de desarrollo, se comienzan a trabajar los primeros entregables que brindarán ganancias al cliente.

En otras ocasiones cuando el proyecto comienza con muy corto tiempo o se retoma una funcionalidad a desarrollar previamente analizada, la fase de iniciación y desarrollo se deben desarrollar en paralelo con la finalidad de brindar agilidad e iteraciones cortas que comienzan a generar valor para el cliente. Incluso en algunas industrias la fase de inicio en Scrum la manejan como un *sprint* inicial parte de la Pila del producto, que también se desarrolla en dicho *Sprint*. En figura 14 se muestra la pila del producto en una herramienta de gestión.

Fig. 14 Pila del Producto en una Herramienta de Gestión

Summary	Story Points	Effective Effort	Priority	Rank
Dynamic format in text label after adding a Payee	--	--	High	1
Generals	--	--	High	2
1.Png's	0 pts	--	Unassigned	2.1.1
2.Browsers	0 pts	--	Unassigned	2.1.2
Functionality	--	--	Low	2.2

Fuente: Elaboración propia, 2018

A pesar de que existen diversas similitudes entre las metodologías progresivas y las iterativas e incrementales, esencialmente en el caso particular aquí descrito, se tuvieron muchas dificultades para la adopción de Scrum en las áreas de desarrollo y pruebas. Mucha gente involucrada en estas tareas aún conserva la idea de que las metodologías ágiles son utópicas ya que, si bien trabajan con diferentes herramientas, al final hacen las mismas funciones, entonces en ocasiones donde hay que hacer cambios en el alcance, modificar el producto final, extender los tiempos de entrega, etc. Ponen muchas trabas y solicitan documentos que se solían pedir cuando se administraban proyectos bajo el esquema de cascada, específicamente hablando de "Requerimientos de Cambios", los cuales no tienen razón de ser en un esquema de Scrum pues los cambios son, por ende, parte de la definición continua del alcance.

Como se explica en la tabla siguiente, una de las ventajas de la fase de iniciación en cascada es el conocimiento completo de los requerimientos de negocio, sin embargo, parte de las desventajas es la poca disponibilidad de cambios en el alcance y la casi nula flexibilidad en las fases.

En cambio, la fase de iniciación de un proyecto en *Scrum*, tiene la flexibilidad en las fases y disponibilidad en los cambios y alcance, sin embargo, el conocimiento del producto final no es tan necesario ya que se irá moldeando a lo largo del desarrollo del proyecto.

Metodología	Ventajas	Desventajas
Cascada	<ul style="list-style-type: none"> <li>- Conocimiento completo de requerimientos que proporciona negocio.</li> <li>- Tiempo invertido en diseño en primeras fases evita costos muy elevados en fases futuras de desarrollo.</li> <li>- Una definición bien estructurada se vuelve más fácil comprender el alcance.</li> <li>- Esta metodología es idónea para proyectos estables en el alcance donde hay requisitos claros y no van a cambiar a lo largo del proceso de desarrollo, en el ejemplo de esta migración, resultó ineficiente</li> </ul>	<ul style="list-style-type: none"> <li>- Pocos o nulos cambios en el alcance, ya establecido en el documento de alcance.</li> <li>- Poca flexibilidad en fases</li> <li>- Completar una fase para pasar a la otra.</li> <li>- Desconocimiento de los clientes de los requisitos hasta tener primera versión del Software funcionando, solo entonces se hacen cambios y se regresa el proceso.</li> <li>- No se va mostrando al cliente el producto a medida que se va desarrollando, si no que se ve el resultado una vez ha terminado todo el proceso. Lo cual provoca inseguridad por parte del cliente bancario que quiere ir viendo los avances en el producto.</li> <li>- Muchas veces los diseñadores no tenían en cuenta todas las dificultades que se encontrarían cuando estaban diseñando el software, lo que conllevaba a rediseñar el proyecto para solventar el problema en fases avanzadas.</li> </ul>
SCRUM	<ul style="list-style-type: none"> <li>- Flexibilidad de alcance en la migración.</li> <li>- Iteración con el cliente y retroalimentación continua del menú y submenú.</li> <li>- Dinamismo en la definición de alcance y requerimientos.</li> <li>- Se tiene un mayor control sobre el producto final, ya que se registra y</li> </ul>	<ul style="list-style-type: none"> <li>- Poca visibilidad del alcance final</li> <li>- En las fases iniciales es fácil perderse del objetivo principal.</li> <li>- Poca documentación lo cual en ocupaciones para algunos miembros del equipo, se interpreta como desinterés y falta de compromiso.</li> </ul>

	<p>se le da seguimiento continuo al alcance.</p> <ul style="list-style-type: none"> <li>- El trabajo en equipo es un plus que ayuda a solventar y desarrollar el producto de manera más fácil, ya que el bien común del equipo es una de las razones principales de esta metodología.</li> <li>- El incremento del producto final es una gran ventaja ya que se puede medir de forma palpable el software que se está desarrollando.</li> <li>- En las primeras fases se palpan piezas funcionales del producto final.</li> <li>- El alcance puede ser redefinido cuantas veces sea necesario a lo largo del proyecto con el objetivo de mantener al cliente satisfecho y cumplir los parámetros establecidos por las reglas de negocio y requerimientos.</li> </ul>	
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

Dificultades confrontadas durante la migración
<ul style="list-style-type: none"> <li>- Resistencia al cambio de varios interesados en el Proyecto.</li> <li>- Debido a que son grupos estructurales muy grandes, se vuelve muy difícil coordinar los cambios y la retroalimentación de los equipos.</li> <li>- Muchos de los equipos involucrados en el proceso de desarrollo de software, no están familiarizados con los principios de agile ni Scrum, por lo que la colaboración es casi nula, la autogestión es un tema utópico y por el contrario existen muchas trabas en el proceso.</li> <li>- Después de haber trabajado durante muchos años bajo los principios de la metodología de cascada, para varios miembros del equipo que tienen mayor antigüedad en el equipo, resulta muy complicado tener poca documentación durante la fase inicial del proyecto, ya que no se cuenta con un acta constitutiva del proyecto, ni un enunciado del alcance firmados, aprobados y revisados por los principales interesados en el proyecto.</li> <li>- En Scrum se trabaja solamente con un documento virtual que se desarrolla en una herramienta denominada RTC, donde se contienen las historias de usuario y sus respectivos criterios de aceptabilidad, en muchas ocasiones resulta difícil darle seguimiento a los requerimientos con el detalle que contenían los BRDs (<i>Business Requirement Documents</i>) que son los documentos de requerimientos de negocio que se utilizaban para definir a detalle los deseos del cliente.</li> </ul>

En conclusión se puede decir que ambas metodologías trabajan bajo el esquema del PMBOK, pero ambas tienen características específicas en la fase de iniciación, de acuerdo a mi experiencia cuando he trabajado en Scrum, resulta mucho más sencilla la implementación de cambios en el alcance y desde esta fase se pueden mostrar avances tangibles para el cliente, lo cual se interpreta en ganancias tempranas y un elevado grado de satisfacción, pues al ser una aplicación en línea los clientes o usuarios finales retroalimentan de forma positiva, y así crean referencias positivas que, a su vez, crean lealtad y convencimiento de nuevos usuarios; sin embargo la poca o nula documentación se traduce en falta de recursos para seguimiento y cumplimiento de fechas programadas en los entregables.

### 8.3.2 Planeación.

Continuando con el proyecto descrito en la fase de inicio, una vez teniendo las aprobaciones de los principales interesados del proyecto, así como los vistos buenos de los aprobadores y revisores descritos en el documento del alcance, el acta constitutiva del proyecto, el enunciado del alcance y la matriz de interesados; se puede pasar a la fase de planeación del proyecto.

El principal entregable que servía de base para las siguientes fases del proyecto, era el desarrollo del plan para la dirección del proyecto, este plan se consideraba una línea base temporal que permite tener la referencia del alcance en tiempo, calidad y costos; también en las reuniones de dirección, servía como punto de referencia para comparar el progreso real del proyecto.

Como parte fundamental de este plan, varios elementos tenían que ser incluidos en el mismo, por ejemplo el plan de gestión de los involucrados o *stakeholders* del proyecto, donde se hacía una lista de los posibles equipos, gerentes de dichos equipos, puntos de contacto, roles y responsabilidades para que, en caso de ser requeridos, se contactara con ellos y de esta manera a lo largo del proyecto saber quién podía responder y en qué momento; también con la finalidad de mantener una línea directa de comunicación para compartir el estatus del avance del proyecto.

Otro elemento es el plan de gestión de las adquisiciones donde se describía la lista de contratos, proveedores y empresas que brindan servicios externos de diferentes tipos (QA, desarrollo, integración, etc.) y que estos a su vez, contribuyen con el proyecto y también deben estar contenidos en el plan inicial de presupuesto.

El plan de gestión de riesgos se establecía desde la fase de inicio, sin embargo, en esta fase, se hacía un plan de mitigación y respuesta de riesgos, posibles involucrados, priorizaciones y características de estos, posibles soluciones para evitar, persuadir, mitigar riesgos negativos o en su defecto impulsar riesgos positivos como la reutilización de una solución técnica antes utilizada. En este plan, se debían considerar tanto un análisis cualitativo y cuantitativo de riesgos una vez identificados, con la finalidad de determinar las características antes mencionadas y así poder tener en control uno de los principales elementos de la fase de planeación del proyecto.

Dentro del plan de dirección, también colocábamos un plan de comunicaciones en línea vertical y horizontal, en otras palabras, se hacía un plan de reuniones, diarias, semanales, quincenales o mensuales con diversos interesados en el proyecto, así como también ciertas presentaciones hacia la alta dirección, con la finalidad de mostrarles avances y/o posibles riesgos.

La planeación de los recursos humanos también formaba parte de los principales elementos, ya que se hacía un plan de la utilización de los recursos, se creaban equipos equitativos donde normalmente había asociados de más experiencia con otros de poca experiencia, las horas planeadas a cada equipo debían coincidir con las estimaciones hechas en el documento del alcance.

La planeación de la gestión de la calidad, los costos, el cronograma y el alcance se manejaba en un solo plan, denominado plan de gestión en los entregables; también era parte del plan de

dirección, e incluía además de los elementos descritos anteriormente, las características de calidad del entregable final, características funcionales y no funcionales, los costos por hora de cada recurso, el calendario con los principales hitos a entregar y las revisiones del alcance para medir y comparar el progreso con el trabajo pendiente.

Cabe mencionar que, con respecto a los costos, una vez planeada la gestión de los mismos se debe hacer una estimación aproximada para que sirva de referencia en el momento de determinar el presupuesto del proyecto, incluyendo los entregables del plan antes mencionado.

En el caso del proyecto de desarrollo de software expuesto en este trabajo, un área específica de representantes del negocio se encargaba de todo lo relacionado al plan de entregables, ya que ellos cubrían o negociaban el presupuesto destinado a ciertos proyectos y el cual se debía adaptar al número total de proyectos en estado de desarrollo.

Sobre la gestión de los requisitos, también se llevaba a cabo un plan donde se describían todos los requisitos funcionales y no funcionales del negocio, posteriormente se procedía a crear el plan con base en las estimaciones preliminares o estimaciones cero, se comprobaban los impactos en los sistemas de "backend" y toda aquella información técnica necesaria para el desarrollo de los mismos requisitos. Dentro de esta sección, se incluían las aprobaciones de los representantes de negocio o del cliente. Y también se veía impactada por un plan de mejora de procesos, así como de gestión de cambios donde se consideraban los posibles cambios que impactarían los requisitos o requerimientos realizados.

Otro factor por resaltar es la creación del WBS o Estructura del desglose del trabajo, donde se desmenuzaban los requisitos del negocio en pequeñas tareas, las cuales a su vez tenían actividades asignadas que se podían estimar en tiempo y esfuerzo, así como secuenciar de acuerdo a la importancia o criticidad.

Parte de la definición del alcance total del producto a desarrollar, también se incluyen dentro del plan de gestión del proyecto, las ventajas y limitaciones del producto a desarrollar, así como el enunciado del alcance del trabajo (adjunto ejemplo en la sección de anexos).

Una vez que se tenían completos los documentos, tanto el plan de dirección de proyecto, como el plan de gestión de los entregables, debidamente formados, aprobados y revisados por los principales interesados, y después de una junta de revisión de estos documentos con todos los equipos; se procedía a la fase de ejecución.

En la figura 15 se muestran los principales elementos que se usan en el Documento del Plan de Dirección de Proyecto, también se agrega en el anexo 5 el contenido completo.

**Fig. 15 Plan de Dirección de Proyecto**

CONTENIDO .....	2
PLAN DE PROYECTO .....	3
FUNDAMENTO .....	3
PRINCIPALES TÉCNICAS Y HERRAMIENTAS A UTILIZAR PARA LA DIRECCIÓN DEL PROYECTO ...	4
PRINCIPALES TÉCNICAS Y HERRAMIENTAS A UTILIZAR PARA LA GENERACIÓN DEL PRODUCTO	4
PRINCIPALES DOCUMENTOS A ELABORAR .....	5
Respecto al proyecto .....	5
Respecto al producto .....	5
FASES DEL PROYECTO - .....	6
PLAN DE GESTIÓN DE ALCANCE .....	6
PLAN DE GESTIÓN DEL CRONOGRAMA .....	6
PLAN DE GESTIÓN DE COSTOS .....	6
PLAN DE GESTIÓN DE LA CALIDAD .....	7
PLAN DE MEJORAS DEL PROCESO .....	7
PLAN DE GESTIÓN DEL PERSONAL .....	8
PLAN DE GESTIÓN DE LAS COMUNICACIONES .....	8
PLAN DE GESTIÓN DE RIESGOS .....	8
PLAN DE GESTIÓN DE LAS ADQUISICIONES .....	8
.....	9

Fuente: Elaboración propia

Cuando comienza la migración hacia Scrum, surgen conflictos de primera mano que se explican en la tabla de dificultades confrontadas durante la migración, pues muchos interesados en el proyecto veían una debilidad en la metodología Scrum, especialmente en la fase de planeación, si bien bajo el esquema de cascada existen muchos elementos necesarios para considerar completa esta fase, en Scrum se vuelve muchas veces imperceptible cuando comienza y termina la planeación y la ejecución del proyecto.

Sin embargo, de acuerdo con la experiencia obtenida en el proyecto, se identifica que comienza la fase de planeación una vez que se crea la lista de producto o pila del producto en el *Sprint* inicial.

Normalmente se tiene una junta de alrededor de ocho horas (para la planeación de *Sprints* de cuatro semanas), la cual se repite cada *Sprint* planeado, y se lleva a cabo con la finalidad de que el equipo Scrum completo (desarrolladores, analistas de negocio, QA, bases de datos, etc.) pueda planificar las actividades, tareas y roles que se llevarán a cabo a lo largo del *sprint*. Es aquí donde se va a desmenuzar la pila del producto en pila del *Sprint*, básicamente es un proceso similar al WBS o estructura del desglose de trabajo, donde los mismos equipos Scrum determinarán por medio de estimaciones, en este caso en puntos de historia (*Story Points*), cuantas historias de usuario tienen planeado completar en el periodo de duración del *Sprint*.

Respecto a los entregables bajo el esquema de Scrum, en esta fase se identifican dos principalmente, una planilla en Excel denominada *Product Backlog Template*, donde se define el identificador y nombre del proyecto a desarrollar, la prioridad, el tipo de ítem de trabajo, la descripción del mismo, los criterios de aceptabilidad, el estado actual de la tarea como pendiente, o comprometido o terminado, etc; el nombre de la persona asignada al seguimiento (usualmente el *Project Manager* o el *Scrum Master*) así como la estimación del ítem; también dentro del mismo documento existe una pestaña denominada *Sprint Backlog Template*, la cual contiene información similar a la antes mencionada pero en relación a los entregables planeados por *Sprint*. En otras palabras, contiene detalle de la parte proporcional a desarrollar en el periodo definido, complementando la pila del producto más otros datos relacionados a la estimación en horas, el identificador del *Sprint*, y el nombre del *Sprint* con la nomenclatura establecida. En la figura 16 se ejemplifica el plan del proyecto, tiempos y duración.

Fig. 16 Tiempo y Duración de Spints en Plan de Proyecto

Sprint	Funcionalidades a Gran Escala	Se libero a producción?	Duración	Datos de Entrega
5	NLP Model – Demo Sprint 5	No	2 semanas	May 08, 2018
6	NLP Model – Demo Sprint 6	No	2 semanas	May 22, 2018
7	NLP Model – Demo Sprint 7	No	2 semanas	Jun 05, 2018
8	NLP Model – Demo Sprint 8	No	2 semanas	Jun 19, 2018

Fuente: Elaboración propia, 2018

También dentro del documento antes mencionado, se incorpora una gráfica denominada *Sprint Burndown Chart*, que contiene el progreso del desarrollo de las historias de usuario por día y por tamaño de esfuerzo. Por último, contiene una gráfica denominada *Project Burnup Chart*, la cual contiene información del progreso por Sprint en relación con el esfuerzo completado en horas y el esfuerzo pendiente por completar en el proyecto.

El otro entregable es una planilla de aprobaciones con los datos administrativos del proyecto, los cuales se complementan con el documento de *Product Backlog Template*; esta planilla deberá ser revisada y aprobada por los principales dueños del producto, negocio o el cliente que paga por el proyecto.

Cabe mencionar que, para los proyectos administrados bajo el esquema de Scrum, la fase de planeación es repetitiva y progresiva, pues se va replanteando de acuerdo al trabajo completado y a los cambios requeridos por el negocio o cliente que solicita los requerimientos. Lo cual significa que se llevará una junta de planeación por cada *Sprint* durante el número de *Sprints* planeados para la entrega final del producto.

En la tabla siguiente se describen las ventajas y desventajas de la implementación en ambas metodologías objeto de este trabajo, en la fase de planeación.

Metodología	Ventajas	Desventajas
Cascada	<ul style="list-style-type: none"> <li>- Debido a que la planeación es extensa y detallada, es poco probable que los riesgos no se hayan contemplado y puedan causar mayor problema en fases futuras.</li> <li>- Se vuelve más sencillo entender la fase en que va corriendo el proyecto ya que es muy estructurado.</li> <li>- En un escenario donde se conoce bien el alcance y es poco probable que se modifique, los tiempos y costos coinciden con el plan al cien por ciento.</li> </ul>	<ul style="list-style-type: none"> <li>- A menudo, el cliente o equipo de negocio con los que se está creando el producto, no saben lo que realmente necesitan por adelantado y desconocen los alcances posibles con la tecnología disponible. No es un buen método para llevar un buen servicio hacia el cliente.</li> <li>- El proceso es bastante cerrado en sentido de cambios y modificaciones en el alcance.</li> <li>- En la fase de planeación no se tiene en cuenta las necesidades cambiantes de un cliente. Si el</li> </ul>

	<ul style="list-style-type: none"> <li>- Antes de desarrollar el código, es importante que los requisitos y diseño estén muy claros, lo cual asegura un desperdicio mínimo de tiempo y esfuerzo, y reduce el riesgo de desfase en el cronograma (<i>Schedule slippage</i>) o que las expectativas del cliente no se cumplan.</li> <li>- Es mucho más fácil y barato detectar y corregir posibles fallas en la etapa de diseño que en la etapa de prueba.</li> </ul>	<p>cliente se da cuenta que necesita más de lo que inicialmente pensó, y exige un cambio, el proyecto se entregara tarde e impactara en el presupuesto.</p> <ul style="list-style-type: none"> <li>- No es el mejor modelo para proyectos complejos y orientados a objetos.</li> <li>- El cliente que está pagando por el proyecto, usualmente va a gastar más dinero en caso de que el alcance sea modificado, incluso también en tiempo se verá afectado el proyecto al salirse del plan inicial.</li> </ul>
SCRUM	<ul style="list-style-type: none"> <li>- El proyecto está abierto a un mayor esfuerzo en desarrollo con relación al plan, hasta la fase de cierre. El entregable se puede cambiar en cualquier tiempo durante las fases de planeación y <i>Sprint</i>.</li> <li>- La parte del control de la planeación, incluyendo la administración de riesgos, se determina en cada iteración de los <i>Sprints</i> para evitar el caos y maximizar la flexibilidad.</li> <li>- Desde la planeación de la pila del producto por parte de su dueño, se identifican las mejoras necesarias que serán implementadas.</li> <li>- Se considera el marco de tiempo necesario para obtener una ventaja competitiva.</li> <li>- La calidad siempre se irá revisando a lo largo de los <i>Sprints</i>, la cual a su vez otorgará valor al cliente desde el inicio.</li> <li>- Existe más a detalle de lo que vendrá próximo y esto será menos perdida en planificación.</li> </ul>	<ul style="list-style-type: none"> <li>- La planeación se hace de acuerdo al número de <i>Sprints</i> planeados, por lo que muchas veces es un proceso poco estable.</li> <li>- La documentación es prácticamente muy escasa, pues esta metodología se enfoca más en entrega de producto funcional que documentación.</li> <li>- Puede ser difícil para un Maestro Scrum planificar, estructurar y organizar un proyecto que carece de una definición clara.</li> </ul>

Dificultades confrontadas durante la migración

El enfoque agile puede resultar complicado de entender para personas que llevan muchos años trabajando bajo una estructura lineal secuencial de cascada, en la migración descrita fue muy complicado convencerlos de la utilización de Scrum

La planeación se puede mal interpretar a un proceso falto de compromiso y sin documentos de evidencia de progreso, los equipos técnicos tuvieron que ser bien gestionados para mitigar este tema.

El software que se desarrolló impactaba diferentes canales y medios como son: *backend*, *middleware*, *frontend*, bases de datos, servidores, entre otros. Los interesados tenían resistencia al cambio y fue difícil resolver las dependencias en este tipo de situaciones.

La gran variación del alcance impacta la pila del producto y la pila del *sprint*, por lo que en ocasiones se vuelve complicado darles seguimiento a los requerimientos de negocio a lo largo del ciclo de vida del proyecto. Para el caso de la migración, se tuvo un correcto seguimiento.

En resumen, la forma de gestionar la planeación en un esquema lineal secuencial, se vuelve mucho más extensa y en muchas de las ocasiones efectiva, sin embargo como se menciona en el cuadro de las desventajas, resulta un fuerte impacto económico para el negocio o cliente al hacer modificaciones al plan de dirección de proyectos en fases avanzadas de desarrollo o pruebas; en cambio bajo un esquema ágil, es mucho más flexible la forma de agregar o quitar elementos al alcance de acuerdo a las necesidades del negocio, sin embargo como parte de las principales desventajas, es el hecho de que al no existir documentación relacionada a los compromisos en fechas, el seguimiento es mucho más informal y en ocasiones hasta se vuelve complicado poder cumplir las fechas compromiso y los elementos planeados en la pila.

### 8.3.3 Ejecución.

En esta fase comienza el desarrollo del producto, se comienzan a ejecutar las tareas y trabajos planeados en el cronograma. Se inicia la codificación con base en la configuración descrita en los diseños funcionales, el desarrollo de requerimientos funcionales, la configuración de bases de datos, las planillas de integración de protocolos, entre otras actividades.

Un proceso que se consideró muy importante en la fase de ejecución del proyecto expuesto en este trabajo, fue dirigir y gestionar el trabajo del proyecto; lo que significa que una vez teniendo los entregables de la fase de planeación, se comienzan a ejecutar dichos planes, primeramente se inició la codificación basándose en los diseños y requerimientos funcionales con las características de autenticación y funcionalidad, siempre tomando en cuenta el involucramiento de los interesados en el proyecto, así como también una correcta gestión del conocimiento del producto que se debía entregar. Es por eso por lo que se vuelve relevante la participación de los involucrados, pues en esta fase surge el mayor número de conflictos y se comienzan a materializar los riesgos.

Por lo anterior, es muy importante tener un registro y desglose de actividades por responsable, fechas compromiso y estatus. En la figura 17 se muestra un ejemplo de las actividades que se deben ejecutar con fechas específicas.

Fig. 17 Seguimiento de Actividades

NOTES:

Critic Delay
Deleay
In Progres / Future Work
Important
Closed

No.	Descripción de Actividad	Responsable	Fecha Inicio	Fecha Fin	Comentarios	Estatus
<b>Activities February 29 to March 04, 2016</b>						
<b>Next Generation (Release 6.5)</b>						
1						

Fuente: Elaboración propia, 2018

También durante la ejecución del proyecto, se irá monitoreando el progreso tanto de desarrollo como de la ejecución de pruebas. Para esto último se van a ejecutar pruebas en diferentes niveles, desde las denominadas MIT y SIT que se hacen desde la fase de construcción, hasta las pruebas UAT que sirven de validación para el usuario, simulando un escenario con las características de un ambiente productivo, pero con datos que no afectan o alteran las cuentas reales.

En el caso específico expuesto en este documento, también la adquisición de recursos que se utilizarán tanto en el desarrollo del código, como la gestión de pruebas basadas en los requerimientos funcionales y no funcionales, se vuelve trascendente en la fase de ejecución. Usualmente desde la planeación, se sabe que proveedor será el encargado de codificar y probar; esto bajo los esquemas específicos del negocio y con la supervisión de algún interesado de la misma institución bancaria.

Una vez formado el equipo, establecidas las reglas del juego, se comienza el proceso antes mencionado de codificar la aplicación del menú y submenú de transferencias y pagos, para dicho objetivo se crea una "receta de cocina" que no es otra cosa sino una guía del tipo de código que se está utilizando, las reglas de negocio, las excepciones, las configuraciones de conectividad, protocolos, mensajes y otros factores importantes para que los desarrolladores puedan codificar de manera eficaz y de estas manera comenzar las pruebas de funcionalidad para cerciorarse de que la aplicación funcione de acuerdo a lo planeado.

Otro punto importante que resaltar es la diferencia entre el diseño de alto nivel o arquitectónico y el diseño detallado.

El primero de ellos tiene como objetivo definir la estructura de la solución (el menú y submenú impactados) identificando los módulos a gran escala (por ejemplo, las funciones de pagos y transferencias y los diferentes cruces de estos) y sus relaciones. Con ello se definía la arquitectura de la solución que se implementó. El segundo define los algoritmos empleados y la organización del código para comenzar la implementación, es aquí donde se desarrolla el *Cook Book* o recetario mencionado anteriormente. Se muestra un ejemplo de la plantilla del párrafo anterior, con el documento de diseño técnico y control de versiones en la Figura 18 la cual es crucial en la ejecución.

Fig. 18 Documento de diseño técnico

Project Name:	<Project Name>
Project ID:	<Project ID>

**Document Control**

a. Document History

Version	Date	Status: <Draft/ Final>	Author	Comment / Changes from Prior Version

b. Document Reviewers/Approvers

Name	Position (e.g. Client Manager, Project Manager, Client Manager delegate etc.)	Reviewer (only)	Reviewer and Approver
	Project Manager		X
	Application Manager		X
	Architect (review only required for projects that have been deemed to be architecturally significant)	X	
	Technology Information Security Officer (TISO) (only required for a Technical Design that affects an application in the High Risk Subset of Internet Applications)		X

Fuente:(Elaboración propia)

Durante la fase de ejecución, se lleva a cabo el plan de comunicación donde el gestor o administrador del proyecto, deberá tener llamadas diarias, semanales o quincenales según lo requerido con la finalidad de transmitir y sincronizar a los equipos en el avance y seguimiento del proyecto.

Como se muestra en la imagen de Figura 19, se deben tener datos específicos de contacto de los principales interesados del proyecto.

Fig. 19 Ficha de roles y responsabilidades

Puesto	Compañía	Nombre	e-mail	Ext/Celular
Negocio	BMX	[Redacted]	[Redacted]	[Redacted]
BRs	AGS-Nasoft	[Redacted]	[Redacted]@cti.com	[Redacted]
PMO	BMX	[Redacted]	[Redacted]@cti.com	[Redacted]
ST (SME Local)	AGS-Nasoft	[Redacted]	[Redacted].com.mx	[Redacted]
ST (SME Local)	AGS-Nasoft	[Redacted]	[Redacted]	[Redacted]
ST (SME Local)	AGS-Nasoft	[Redacted]	[Redacted]	[Redacted]

Fuente: Elaboración propia

Por ejemplo, en la implementación objeto de estudio de este trabajo, al inicio de la fase de ejecución, se tenían llamadas diarias con los equipos técnicos, tanto desarrolladores como *Testers* (equipo QA); las llamadas y reuniones con negocio se llevaban a cabo semanalmente y las reuniones con la alta gerencia eran de manera quincenal.

Posteriormente cuando el proyecto comenzó a evolucionar y la fase de pruebas tomó lugar, las reuniones con el equipo técnico llegaban a ser hasta 3 veces al día, con la finalidad de resolver errores y defectos, dudas de funcionalidad y configuración con bases de datos y sistemas de *backend*, también en estas reuniones se revisaba el avance de todos los miembros del equipo, así como dependencias con otros equipos o interesados. Con respecto a las reuniones con negocio, se llegaban a tener hasta 3 por semana o en su defecto, cadenas de correos continuas,

con la finalidad de resolver dudas y/o obtener aprobaciones. Las reuniones con la alta gerencia se desarrollaban semanalmente y se trataban temas de presupuesto y alineación tanto de alcance, calidad, riesgos y tiempos. La estructura de una minuta desarrollada durante la fase de ejecución se puede ver en la Figura 20

**Fig. 20 Estructura de Minuta de Reunión**

REFERENCIA: Reunión telefónica		FECHA: 10.08.2017 – 4:30 P.M.		
<b>AGENDA</b>				
<b>Reunión</b>		<b>Duración :</b>	<b>Elaboró:</b>	
1. El objetivo de la reunión es detallar los pasos a seguir para la migración de diversos requerimientos		30 min.	Yazmin Hernández	
<b>ASISTENTES</b>				
Mendoza, Lopez Rocio (RM)				
Guerrero Curiel, Janet (ES)				
Cano, Moises (MC)				
Valencia, Gerardo (GV)				
<b>Puntos Tratados</b>				
<ul style="list-style-type: none"> <li>• Scope</li> <li>• Capacidad de los equipos</li> </ul>				
<b>Compromisos</b>				
Responsable	Descripción	Fecha	Fecha Compromiso	Fecha Compromiso
Moises C.	Buscar estimaciones con los equipos para los requerimientos, dará un update en la siguiente sesión.	03 Agosto 2017	10 Agosto 2017	18 Agosto 2017
Moises C.	Resolver conflicto de migración de metología.	10 Agosto 2017		11 Agosto 2017

**Fuente: Elaboración propia**

Esta fase, termina en la implementación, donde se programaban los requisitos especificados haciendo uso de las estructuras de datos diseñadas durante la fase de construcción. Como se menciona anteriormente, una vez construido el código e implementado en los ambientes de prueba, se hace un copiado del código en un ambiente simulado de producción, donde se utiliza la aplicación a nivel usuario para detectar los últimos errores o defectos antes de salir a producción.

Cabe mencionar que varios de los procesos de la fase de monitoreo y control, se llevan a cabo durante la ejecución del proyecto, sobre este tema se profundizará en el siguiente capítulo.

Con respecto a la migración a Scrum, esta fase de cierta manera fue la menos complicada ya que se comenzó con un cambio de herramienta de gestión, donde en vez de escribirse documentos de requerimientos funcionales, se comenzaron a crear historias de usuario.

En la gestión Scrum, la fase de ejecución es la más extensa, ya que comienza inmediatamente después de terminar la junta de planeación del *Sprint* y se extiende hasta la junta de revisión del mismo. Por lo que el incremento que se está creando del producto final, tiene como característica ser utilizable y es potencialmente un entregable; hablando del menú y submenú, en el primer sprint se entrega la funcionalidad del menú con 3 opciones, de las cuales aún no se ve el submenú que está planeado para el tercer *Sprint*. Ver figura 21.

**Fig. 21 Forma de ejecutar las historias de usuario y la división de la Pila del Producto**

ID del Proyecto:	ID	NOTA: Si estas usando ALM, JIRA, RTC o TFS, no es necesario completar esta plantilla. Solamente copia y pega la Pila del Producto y especifica el Sprint y los links de ALM para mayor referencia.						
Nombre del Proyecto:	Nombre							
NOTA: Esta plantilla es para documentar la lista de todos los documentos deseables que definen el proyecto completo.								
ID	Prioridad	Tipo de Item	Titulo	Descripción	Criterio de Aceptabilidad	Estado	Asignado a	Estimacion del PBI
94001	1	Product Backlog Item	Sample PBI Title 1	Sample PBI Description 1	Sample PBI Acceptance Critería 1.1 Sample PBI Acceptance Critería 1.2 Sample PBI Acceptance Critería 1.3 Sample PBI Acceptance Critería 1.4	Committed	PBI Assignee Name	M

Fuente: Elaboración propia

Otra característica importante en Scrum durante esta fase es la llamada que se tiene diariamente denominada Scrum diario; donde el equipo de desarrollo se está sincronizando para acordar que trabajo van a estar entregando dentro de las próximas 24 horas. Un factor que se considera en esta reunión es que se hace todos los días laborales a la misma hora y de preferencia en el mismo lugar, ya sea en persona o de forma remota, pero en videoconferencia.

En esta reunión, también se revisa por parte del administrador de proyecto o maestro Scrum, así como el dueño del producto, un seguimiento del trabajo planeado del día anterior, y el trabajo que se pretende completar hacia la meta del *Sprint* el cual, a su vez, está complementando el trabajo planeado durante la Pila del producto.

Como experiencia personal, se puede decir que estas reuniones sirven para mejorar las comunicaciones entre el equipo y los interesados en el proyecto, así como eliminar obstáculos para el desarrollo del producto, también se promueve la rápida toma de decisiones y ayuda al equipo a seguirse auto gestionando pues la información es clara y transparente para todos, así como los objetivos generales y específicos del proyecto.

A pesar de que la metodología Scrum es muy flexible en cuanto a los cambios del alcance, es importante asegurar que, durante el *Sprint*, no se lleve a cabo ningún cambio que pudiera afectar el objetivo del *Sprint*. Tampoco disminuir los objetivos de calidad y criterios de aceptación, sin embargo, en la práctica algunos *Sprints* llegan a ser muy largos, en este caso la definición de lo que se construye cambia constantemente y esto produce un aumento en la complejidad y los riesgos que muchas veces ha llegado a provocar la revocación del proyecto por falta de visibilidad.

Sin embargo, en la migración expuesta en este documento, siempre se trató de mantener un periodo corto en los *Sprints*, con la finalidad de controlar la visibilidad, la calidad y la utilización de los entregables, sin sacrificar la optimización por parte del cliente en cada junta de revisión.

Ventajas y desventajas en la fase de ejecución:

Metodología	Ventajas	Desventajas
Cascada	<ul style="list-style-type: none"> <li>- Estructuración clara y bien definida de las tareas.</li> <li>- Ejecución delimitada por tiempos y actividades.</li> <li>- Documentación exhaustiva de desarrollo y pruebas.</li> <li>- Control de riesgos.</li> </ul>	<ul style="list-style-type: none"> <li>- Inflexibilidad en cambios pos-planificación.</li> <li>- En caso de algún error en el plan de dirección del proyecto, representa retraso y aumento del costo del proyecto.</li> </ul>

		<ul style="list-style-type: none"> <li>- Cambios demasiado documentados y aprobados por diferentes interesados en el proyecto.</li> <li>- Si existiera alguna omisión en el código o en las matrices de prueba, de deberá levantar un cambio y esperar aprobaciones.</li> <li>- En fases avanzadas como ejecución, algún cambio en el alcance resulta muy caro en tiempo y dinero.</li> <li>- Procesos muy rígidos y con elevado grado de dependencia de otros procesos.</li> </ul>
<p>SCRUM</p>	<ul style="list-style-type: none"> <li>- Flexibilidad en el alcance, tiempo y presupuesto.</li> <li>- Si el alcance cambia de acuerdo al plan inicial, no existe mayor riesgo.</li> <li>- La calidad no se sacrifica por las entregas tempranas de Software.</li> <li>- Desde los primeros <i>Sprints</i> se entregan pequeñas porciones del producto, ya funcionando y que aportan un valor al negocio o dueño del producto.</li> <li>- Dentro de los criterios de aceptabilidad de cada historia de usuario, se describe de forma clara y concisa, la calidad que deberá tener la funcionalidad.</li> <li>- Los cambios se pueden realizar de manera libre a lo largo del proyecto.</li> <li>- En las juntas de revisión, el cliente puede ir moldeando el producto final de acuerdo a sus necesidades.</li> <li>- Se lleva un control diario del avance del proyecto, y de las actividades del equipo técnico.</li> <li>- En la ejecución del <i>Sprint</i>, se puede tener mejor conocimiento</li> </ul>	<ul style="list-style-type: none"> <li>- Flexibilidad en proyectos estables donde el alcance es fijo y no cambia.</li> <li>- Poca documentación de control.</li> <li>- Mala interpretación de la metodología.</li> <li>- Falta de disciplina del equipo técnico en la autogestión.</li> </ul>

	de los riesgos y así evitarlos o mitigarlos.	
--	----------------------------------------------	--

Dificultades confrontadas durante la migración
<p>La falta de conocimiento de la metodología SCRUM, hizo que las personas que impactaban el proyecto tuvieran un grado elevado de renuencia al tomar nuevos procesos que resultarían en la implementación ágil.</p> <p>La forma de trabajo y los procesos de cascada resultaban lógicos para muchos interesados en el proyecto, de tal suerte que la secuencia era natural y fácil de seguir, al implementar Scrum, muchas personas tuvieron que reorganizar todos sus procesos para adaptarse a los principios de dicha metodología.</p> <p>Los primeros <i>Sprints</i>, se llegaban a alargar más tiempo del definido, ya que muchas personas no estaban acostumbradas a trabajar bajo un esquema dinámico, y les costaba mucho trabajo adaptarse a los tiempos y a ser auto-gestionados, les resultaba complicado trabajar sin un capataz que los estuviera persiguiendo para realizar las entregas y medir los tiempos.</p>

En resumen, la forma de trabajo de Scrum en la ejecución de los *Sprints*, representa la mayor parte del esfuerzo de todo el proyecto, sin embargo al ser una metodología cíclica, esta fase se repite en diversas ocasiones, es la parte central y medular del proyecto, pues es donde se desarrolla el Sprint y se hace una revisión del mismo con el cliente, para que este pueda dar su visto bueno y aprobar o rechazar el avance periódico del producto final, también es importante resaltar que en esta fase el cliente, ya puede disfrutar de los beneficios de la mínima parte funcional del producto, resultado del *Sprint*, lo que se traduce a ganancias tempranas de la empresa.

#### 8.3.4 Monitoreo y control.

La fase de monitoreo y control, como su nombre lo dice, es donde se estará vigilando la ejecución del proyecto; dentro de esta fase el equipo de administración de proyectos se encargada de monitorear todo el trabajo planeado y darle seguimiento desde la perspectiva colectiva de los interesados del proyecto, principalmente el área de negocio y los equipos técnicos.

Con la finalidad de cumplir lo descrito en el párrafo anterior, así como también monitorear el involucramiento de los interesados en el proyecto de desarrollo del menú y submenú de transferencias y pagos, se manejaban tarjetas de roles y responsabilidades para un mejor seguimiento; tal como se muestra en la figura 22, se mencionaba el nombre del proyecto, el número de registro con la finalidad de tener un control del mismo, el área que debía pagar por el proyecto, los nombres de los principales interesados del mismo y la estimación en tiempo del esfuerzo de ejecución así como la fecha compromiso del mismo.

Fig. 22 Planilla de monitoreo del proyecto

### Bancanet Personal

Num. de Proyecto: <input type="text"/>		Nombre Oficial del Proyecto: IMPLEMENTAR <input type="text"/> FASE 1	
Depto(s). a Facturar: <input type="text"/>		Departamento Usuario: <input type="text"/>	
Area de negocio: <input type="text"/>		Nombre y puesto del solicitante: <input type="text"/>	
Nombre del Líder: <input type="text"/>		Categoría del Proyecto: <input type="text"/>	
Fecha Acordada Término: 29/07/2013		Tipo de Proyecto: <input type="text"/>	
Esfuerzo total (Estimación nivel 1): <input type="text"/>		Tipo de Esfuerzo: <input type="text"/>	
Presupuesto Inv. Gasto Consultoría (Estimación nivel 1): <input type="text"/>		Núm. de Proy. Alfa: <input type="text"/>	

Fuente: Elaboración propia

Otro punto a resaltar en esta fase es el control integrado de cambios, donde se vigilan todas aquellas modificaciones que sufre el alcance del proyecto; en tal documento se menciona el número de cambio dentro de la base de datos de registro, una breve descripción del cambio, la fecha de inicio y fin con relación a la implementación del mismo, el impacto que sufrirá el calendario de la oficina de administración de proyectos, en caso de existir algún comentario, también se debe registrar por ejemplo, si es expedito, o de alto riesgo, etc. El estatus de este, como pueden ser aprobado, pendiente, denegado; por último, el nombre de responsable que solicita dicho cambio en el cual caerá la responsabilidad del impacto. Ver Figura 23.

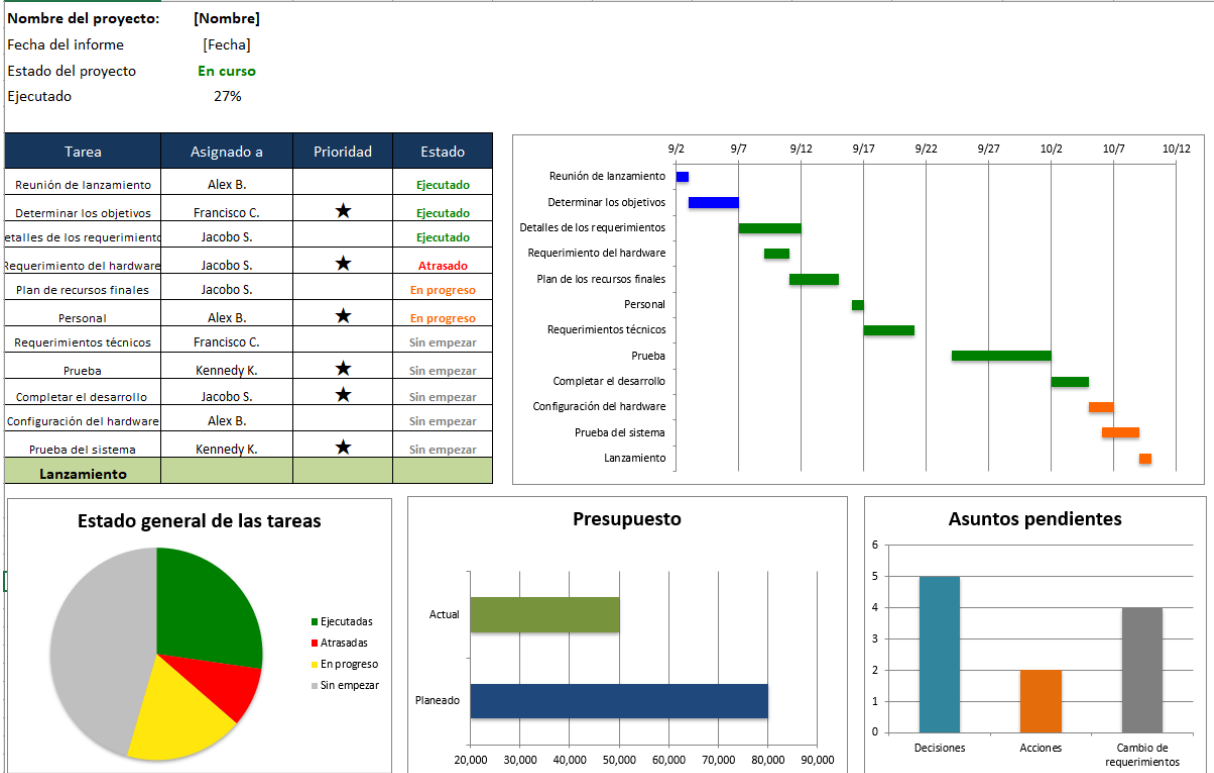
**Fig. 23 Administración de Control de Cambios**

No. Cambio	Descripción	Fecha Inicio	Fecha Fin	Impacto Calendario (DIAS)	Comentarios	Estatus	Solicita y Asume Impacto

Fuente: Elaboración propia, 2018

La calidad de los entregables, las fechas de entrega, el monitoreo de las comunicaciones, riesgos y recursos, así como el control del alcance se van a gestionar en el documento de control del proyecto, el cual se ve en la Figura 24.

**Fig. 24 Documento de Control Integral del Proyecto**



Fuente:(Elaboración propia)

La importancia de este documento en la gestión de proyectos bajo una metodología de cascada es crucial para saber el estatus actual del proyecto, mantener alineados los tiempos y costos, tomar medidas necesarias para resolver algún problema, mitigar riesgos, etc. Siempre y cuando los datos con lo que se nutra el documento sean reales, los equipos involucrados en el desarrollo del proyecto estarán atentos a dicha información; la forma de gestionar este tipo de comunicación de estado y progreso del proyecto era por medio de un repositorio de información denominado *Sharepoint* el cual debía estar actualizado con la última información correspondiente a cada equipo, y el responsable del mantenimiento de tal repositorio era el gerente de proyectos.

La gestión de riesgos también se monitorea en esta fase, se utilizaba un documento denominado administración de riesgos, el cual contiene toda la información relacionada a los riesgos materializados durante la fase de planeación y ejecución, también contiene información sobre la verificación e instalación del Software como parte del proceso de ejecución. En el ejemplo de la Figura 25 se muestran los elementos principales que contiene el documento de riesgos, como son la descripción, fecha de reporte, fecha crítica, el responsable, el impacto, la probabilidad de ocurrencia, la severidad, el estatus y la posible solución del riesgo.

Fig. 25 Control y Administración de Riesgos

No. Risk	Descripción	Fecha Reporte	Fecha Critica	Responsable	Impacto Alto(3) Medio(2) Bajo(1)	Probabilidad de ocurrencia Baja: 0.25 Media: 0.50 Alta: 0.75 Issue: 1.0	Severidad Impacto x Probabilidad Alta > 1.5 Media > 0.5 y < 1.5 Baja < .5	Estatus	Mitigación de Riesgo

Fuente: Elaboración propia

También dentro el desarrollo del proyecto en Scrum se considera muy importante la fase de revisión, monitoreo y control del producto; sin embargo, a diferencia de la metodología de cascada, la revisión y refinamiento del producto se realiza cada *Sprint* y va correlacionado a la pila del producto tanto como a la pila del *Sprint*.

Con este preámbulo, la migración metodológica de realizó de forma ágil, ya que todos los equipos estaban familiarizados con actividades de monitoreo y control del proyecto; con pequeñas variaciones, por ejemplo, en la forma de verificar el producto, en este caso el código desarrollado y la funcionalidad en formato Demo.

En la herramienta de gestión de Scrum se incorporan datos de requerimientos de negocio, historias de usuario, épicas, funcionalidades, estimaciones, casos de prueba, casos de uso, matrices de trazabilidad de pruebas, gráficos de comportamiento y seguimiento, entre otros elementos fundamentales en esta fase. En la figura 26, se muestra una matriz de monitoreo de pruebas.

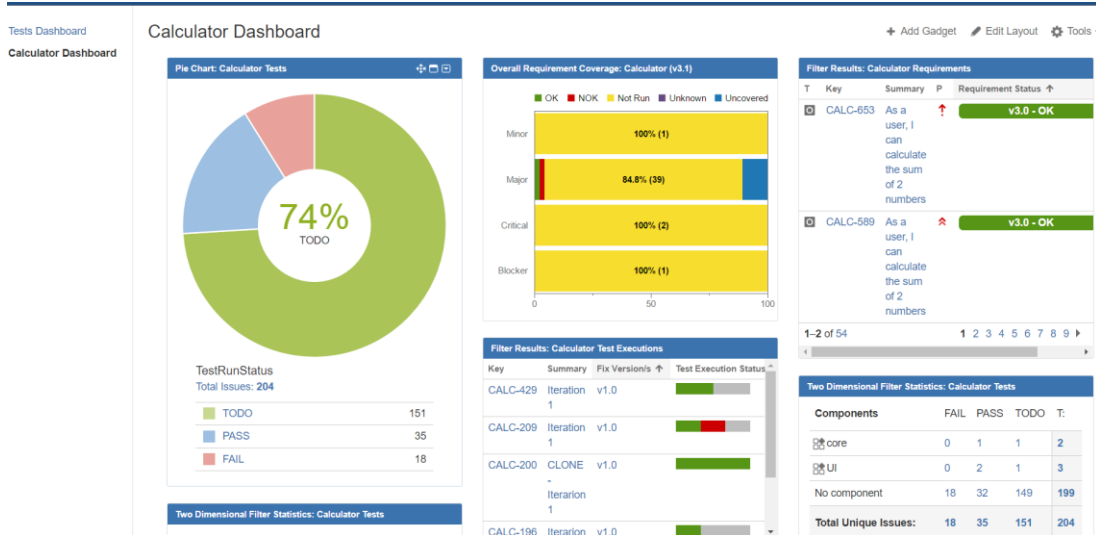
Fig. 26 Matriz de Monitoreo de pruebas

Ciclo	Total # Casos	Ejecutados		Pasados		Fallidos		No ejecutados		Comentarios
		#	%	#	%	#	%	#	%	
1	1866	1866	100%	1849	99%	17	1%	0	0	

Fuente: Elaboración propia

En la Figura 27 se aprecia una pizarra de la herramienta JIRA, con la que actualmente se gestionan los *Sprints* y el producto como tal.

Fig. 27 Pizarra de gestión del Producto



Fuente: Elaboración propia

Ventajas y desventajas en la fase de monitoreo y control:

Metodología	Ventajas	Desventajas
Cascada	<ul style="list-style-type: none"> <li>- Diversos documentos de rastreo y predicción de comportamiento del proyecto.</li> <li>- Las fases de verificación, instalación y mantenimiento se desarrollan en paralelo con la ejecución del proyecto.</li> <li>- Al ser un proyecto predecible, se vuelve menos complicado el rastreo.</li> </ul>	<ul style="list-style-type: none"> <li>- El control de cambios es muy elaborado.</li> <li>- Si se detecta alguna modificación de alcance, se afectan tanto el cronograma como el presupuesto.</li> <li>- Se requieren diversas aprobaciones para continuar a la siguiente fase.</li> <li>- Inflexible en el control de calidad, recursos, plan de comunicación, plan de riesgos, cronograma y costos.</li> </ul>
SCRUM	<ul style="list-style-type: none"> <li>- Dentro de una herramienta de gestión de proyectos, se incorporan todos los elementos de las fases de planeación, ejecución, monitoreo y control.</li> <li>- Los cambios son muy ágiles.</li> <li>- Se mantiene una estrecha comunicación con el cliente y todos los equipos.</li> </ul>	<ul style="list-style-type: none"> <li>- Muchas veces el equipo no es objetivo en su autogestión.</li> <li>- El monitoreo y control del proyecto, así como del producto es muy flexible.</li> <li>- El rastreo se vuelve complicado al ser proyectos impredecibles.</li> </ul>

	<ul style="list-style-type: none"> <li>- Los equipos se autoevalúan y hacen una retrospectiva de su rendimiento.</li> <li>- Las gráficas de <i>burnup</i> y <i>burn-down</i> son excelentes indicadores de desempeño.</li> <li>- Se incrementa la satisfacción del cliente.</li> <li>- El control y monitoreo de las actividades es mucho más dinámico y productivo.</li> <li>- El alcance tiene gran flexibilidad lo cual genera un producto ADOC.</li> </ul>	<ul style="list-style-type: none"> <li>- Se combinan fases de forma involuntaria.</li> </ul>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------

Dificultades confrontadas durante la migración
<p>En esta fase, la migración fue más sencilla que en las tres pasadas, ya que como se ha mencionado previamente, los equipos de cierta manera estaban familiarizados con el monitoreo y control del proyecto; el único inconveniente fue adoptar la herramienta de gestión JIRA, pues no todas las personas tienen facilidad de interacción con la misma. En esta industria hay colaboradores con más de veinte años de experiencia, lo cual vuelve complicado para ellos la adopción de nuevos procesos y tecnologías.</p>

A modo de recopilación, destacan factores como la duración de las fases de verificación, instalación y mantenimiento de la metodología de cascada contra la fase de revisión, refinamiento y retrospectiva de Scrum que se lleva a cabo de forma dinámica y en no más de 2 días se pueden obtener las aprobaciones del proyecto tanto como del producto.

Otro factor importante es el hecho de que la gestión de un proyecto bajo un esquema ágil, desde la planeación hasta el cierre, se puede hacer en una herramienta, mientras que bajo un esquema de cascada se necesitan diversas herramientas y elementos para obtener aprobaciones por fases.

### 8.3.5 Cierre

La fase de cierre en cascada consiste principalmente en la obtención de las aprobaciones de los principales interesados del proyecto. Para obtener estas aprobaciones de los involucrados que se enlistan desde la fase de planeación en el documento de dirección de proyectos, es necesario cubrir ciertos puntos que se mencionan a continuación:

- a) Verificar que el producto tenga el resultado, la funcionalidad y calidad que se acordó en el documento del alcance.
- b) Se debe validar que el producto se transfiera de manera integral al negocio que este a su vez lo validará con sus clientes finales.
- c) El rendimiento del alcance es otro factor que se considera fundamental en la industria bancaria, ya que monitorea los requerimientos específicamente con relación al plan, por

tal motivo el administrador de proyectos debe completar una lista de preguntas y respuestas con la finalidad de determinar el grado de viabilidad del proyecto, así como beneficios cuantitativos y cualitativos obtenidos con el mismo, y que a su vez este resultado sirva de entrada para futuros proyectos. Ejemplo: ¿Los beneficios empresariales originales que se identificaron inicialmente durante la fase de recopilación de requerimientos, donde se solicitaba la creación de un menú y submenú de transferencias en el aplicativo, cumple con la implementación? ¿Se agregaron nuevos requerimientos durante la fase de pruebas? ¿Cuántos cambios se hicieron durante el proyecto?

- d) También se debe incluir el rendimiento de itinerario donde se detalla de manera evidente como se desempeñó el proyecto según el cronograma original; como se muestra en la figura 28, el contenido de la tabla de mejoras debe detallar que se hizo bien y que se puede mejorar.

Fig. 28 Tabla de mejoras continuas y retroalimentación del proyecto.

Áreas de Mejora	Lo que Funciono
<ul style="list-style-type: none"> <li>• Mejor coordinación de problemas ambientales.</li> </ul>	<ul style="list-style-type: none"> <li>• Recopilación de información y seguimiento de problemas, utilizando la herramienta adecuada ( Microsoft Visio) para visualizar hacia dónde iba el proceso</li> </ul>
<ul style="list-style-type: none"> <li>• Comprensión de los datos de entrada que forman parte de las pruebas y deberá considerarse en los entornos para evitar problemas similares.</li> </ul>	<ul style="list-style-type: none"> <li>• Trabajar en equipo a través de problemas ambientales, especialmente a través de zonas horarias, trabajó duro para impulsar la producción.</li> </ul>
<ul style="list-style-type: none"> <li>• Criterios de éxito mejor definidos.</li> </ul>	<ul style="list-style-type: none"> <li>• Los equipos responden al momento de aplicar los cambios.</li> </ul>
<ul style="list-style-type: none"> <li>• Liberación de proceso de gestión y concéntrese en las mejores prácticas para la gestión de procesos.</li> </ul>	<ul style="list-style-type: none"> <li>• <u>Capacidad de entregar incrementalmente.</u></li> </ul>

Fuente: Elaboración propia

- e) Se deben tener también aprobaciones formales del producto tanto del equipo de negocio, la alta gerencia y todas aquellas personas que tuvieron la facultad de tomar decisiones durante la duración del proyecto.
- f) La gestión de riesgos que se desarrolló a lo largo del proyecto es fundamental para mantener un equilibrio en las etapas de ejecución, monitoreo y control. En la figura 29 se muestra la forma de cómo se identificaron los riesgos del proyecto con planes de mitigación, resolución, probabilidad de ocurrencia, la propiedad asignada y plan ejercido de contingencia. Por lo que, en esta etapa se hace un análisis costo beneficio de los riesgos materializados a lo largo del proyecto, así aquellos que se lograron evitar y las lecciones aprendidas de mitigación de los mismos.

Fig. 29 Registro de Riesgos e Incidentes durante el proyecto.

Tipos de Riesgos y descripciones	Prob. De Ocurrencia	Gravedad de Impacto	Plan de Contingencia/Mitigación
Falta de especificaciones finalizadas	Alto	Alto	Calidad realizará pruebas exploratorias con la asistencia de expertos en materia.
Problemas de hardware	Medio	Alto	Sistema operativo, memoria, espacio libre en disco, # de sistemas
Problemas de software	Alto	Alto	Actualizar el sistema constantemente
Desgaste / Habilidades	Alto	Alto	Transferencia de conocimiento a través de documentación y capacitación colaborativa con los miembros del equipo.

Fuente: Elaboración propia

Para tener un correcto seguimiento del proyecto, junto con todas sus actividades, también es necesario incluir un apartado de la gobernanza en la gestión del proyecto, pues es por medio de este documento donde se desencadenarán las lecciones aprendidas y se utilizará para proyectos futuros como referencia y valor de entrada en la fase de iniciación.

Como puede observar en la figura 30, la matriz de revisión contiene datos de gestión como categorización de las actividades, logros y objetivos no cumplidos.

Fig. 30 Matriz de revisión de las actividades del proyecto.

Categoría de Actividad	Lo que funcionó	Logros/Innovaciones	Lo que no funcionó
Gestión de proyectos	Comunicación y mejora en la colaboración.	Apoyó al equipo en lanzamiento.	Coordinación de problemas ambientales
Gestión del tiempo	Los comentarios sobre el uso del tiempo del equipo se incorporaron después de cada sprint	Gestionó reuniones globales y varias zonas horarias	Seguimiento inconsistente de la velocidad
Manejo de costos	Los presupuestos fueron aprobados y no se incurrió en excesos	No aplica.	Reclutar recursos efectivos fue desafiante
Gestión de Comunicaciones / Informes	Scrums diarios del equipo individual.	Retroalimentación continua	Espacio para mejorar las comunicaciones con los clientes, específicamente en torno a las técnicas de datos
Gestión del cambio	Los procesos estándar de gestión de cambios se adhieren.	Se adoptó implementaciones en todos los entornos.  Las solicitudes regulatorias del cliente se rastrean y en la aplicación del modelo para facilitar la respuesta regulatoria.	Los cambios arquitectónicos no se calcularon de manera efectiva antes de la decisión de implementar

Fuente: Elaboración propia

Sobre el documento de lecciones aprendidas, se deberá enumerar cualquier plan de acción asociado a dichas lecciones, las cuales son debidamente documentadas con la finalidad de reutilizar el conocimiento, la experiencia y en algunos casos como marco de referencia para resolver ciertos escenarios que se presentaron previamente y fueron difíciles de solucionar.

Se pone un ejemplo como referencia al compilado de lecciones aprendidas en la figura 31.

Fig. 31 Tabla de control de Lecciones Aprendidas.

Elemento de acción ID	Lección asociada aprendida	Descripción	Acciones Aplicadas	Dueño
1	Mejor comunicación con el cliente y se mantiene con constante retroalimentación del hacia el cliente.	La falta de atención del cliente	La Epica tiene que ser reestructurado más amigable para el cliente.	Dev Manager
2	Abordar dependencias entre equipos	Las primeras dependencias entre equipos no fueron identificadas	Continuar el esfuerzo colaborativo	All
3	Pruebas de regresión automatizadas	La falta de pruebas de regresión automática causó regresiones relativamente tarde en el ciclo de publicación	Especialista en automatización contratado y trabajando en desarrollar la automatización de pruebas.	Dev Manager+ Leads
4	Mejora los datos de entrada	La falta de datos de entrada consistentes para el lanzamiento	Conjunto de datos unificados en todos los equipos	Dev Leads
5	Mejor definición de criterios de éxito	Criterios de éxito no definidos correctamente al comienzo del proyecto, posteriormente mejorados, pero aún hay margen de mejora	Dedicar tiempo adicional para analizar y definir los criterios de éxito en todos los equipos	RP + Dev Leads
6	Mejor definición de requisitos	Espacio para mejorar en la recopilación de requisitos y el mapeo de Historias	Todas las Historias de usuario deben vincularse a una Epica donde los requisitos están correlacionados y acordados por el cliente	RP + Client
7	Lanzamiento más frecuente durante todo el año	Confianza en un solo lanzamiento, aumentó la complejidad	Mejorar lanzamientos en equipos	All

Fuente: Elaboración propia

Cabe mencionar que la migración de metodología fue transparente, pues de todas las fases pasadas, el cierre es donde aún no se hace una migración cien por ciento a Scrum.

Esto debido a que la alta gerencia, el área de negocio y los patrocinadores de los proyectos, tienen la creencia que se deben completar todos los elementos de cierre para que no exista margen de error.

Sin embargo, referente a la forma de cierre actual que se maneja en Scrum, prácticamente la única diferencia es que ahora se entregan Historias de Usuario aprobadas en vez de requerimientos funcionales.

En el siguiente ejemplo (figura 32) se describe la estructura de un correo de aprobación tanto en un marco metodológico de cascada como en Scrum.

**Fig. 32 Correo de Aprobaciones para cierre de proyecto.**

Subject: correo de aprobaciones

Nombre del Proyecto: <Inserte el Nombre de Proyecto>  
 Project Id: <Inserte el Número de Identificación del Proyecto>  
 Nombre del Sprint/ Iteración: <Inserte el Nombre Sprint/Iteración>  
 Número de Identificación del Sprint/Iteración: <Inserte el Número De Identificación del Sprint/ Iteración>

Fecha de Implementación: <Inserte la fecha de Implementación>

Aprobación requerida por: <Inserte la fecha de la aprobación si es requerida>  
*La aprobación debe estar registrada antes de pasar a la producción*

**Problemas notables <Indique cualquier problema conocido a continuación>**

Este correo electrónico es para anunciar que el proyecto anterior ha completado con éxito el UAT y está listo para ser lanzado a producción, con los Problemas conocidos como se indicó anteriormente. Su aprobación significa que otorga permiso para lanzar este proyecto a producción.

Saludos,

<Nombre del administrador del proyecto>

**\*Notes al remitente para UAT y aprobación de implementación:**

- Asegúrese de que la solicitud de aprobaciones se envíe al Gerente de Cliente Senior, Gerente de Cliente y / o Administrador de Gerencia, así como a otras partes interesadas, según corresponda.
- La aprobación debe estar fechada antes de pasar a producción

Fuente: Elaboración propia

Ventajas y desventajas en la fase de cierre:

Metodología	Ventajas	Desventajas
Cascada	<ul style="list-style-type: none"> <li>- Cuando el proyecto se desarrolla de acuerdo con el plan, del cierre se lleva a cabo en tiempo y forma.</li> <li>- Diversos documentos de evidencia y lecciones aprendidas a lo largo del proyecto.</li> <li>- La revisión de desempeño se hace esquemáticamente debido a los ciclos específicos.</li> <li>- Cada fase hace un cierre al obtener aprobaciones.</li> </ul>	<ul style="list-style-type: none"> <li>- Para proyectos con alcance indefinido, el cierre se prolonga por tiempo indeterminado.</li> <li>- Si el producto entregado no cumple con las especificaciones del cliente, se puede rechazar o crear mala reputación para quien lo desarrolla.</li> <li>- Carece de ciertas características como gestión de adquirentes y recursos.</li> <li>- Al no tener cierto acercamiento con el cliente, la retroalimentación se da hasta el final, lo cual si no es positiva recae en pérdida de credibilidad.</li> </ul>

SCRUM	<ul style="list-style-type: none"> <li>- El cierre se va generando de forma evolutiva y siempre con la certeza de cubrir las expectativas del cliente final.</li> <li>- Lecciones aprendidas dinámicas debido al contenido práctico de proyectos similares exitosos.</li> <li>- Se van conociendo de mejor manera las características específicas del cliente que solicita el producto, así como sus cualidades y desventajas de los procesos de gestión de proyectos.</li> <li>- A pesar de contar con poca documentación, el control y cierre se lleva a cabo con un índice de efectividad más elevado.</li> <li>- El cliente o usuario final no se deben esperar hasta el último momento para dar retroalimentación del proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>- La fase de cierre frecuentemente se desvanece a lo largo de los <i>Sprints</i> debido a la interactividad y rapidez de las fases.</li> <li>- Cierre de los equipos Scrum, ya que la sinergia que posiblemente se tenía se rompe y volverlo a crear impacta tiempo y dinero.</li> </ul>
-------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Dificultades confrontadas durante la migración
<p>En esta fase, la migración se complicó debido a que los interesados en el proyecto estaban acostumbrados a tener varios documentos firmados y con aprobaciones, sin embargo, bajo la metodología de Scrum solamente un correo aprobatorio sobre la misma herramienta de gestión fue necesario para liberar el aplicativo al ambiente de producción. Otro factor que obstaculizó la transición fue el dinamismo con el que se crean y desvanecen los equipos Scrum, ya que usualmente en el método de cascada, los equipos técnicos se mantienen a lo largo de los <i>Releases</i>, sin embargo, en Scrum dichos equipos se forman y desintegran de acuerdo con el <i>sprint</i> y al producto a entregar y sus respectivas especializaciones.</p>

A manera de resumen, se puede mencionar que la gestión del cierre de los proyectos en ambas metodologías se mantiene muy similar hasta la fecha, salvo algunos pequeños cambios en la nomenclatura de los documentos de diseño y el contenido de los correos.

Considero que, bajo un esquema ágil de Scrum, el cierre se lleva cabo con cada iteración o *Sprint*, y siempre se enfoca a satisfacer las necesidades del cliente ya que, desde las fases iniciales el producto se va adaptando y refinando para cumplir con los criterios de aceptabilidad, y las definiciones de terminado y listo para usarse (*Definition of Ready – Definition of Done*)

### 8.3.6 Marco referencial de Hitos clave:

A continuación se enlistan los principales Hitos por fases en el ciclo de vida del proyecto:

<b>Fase</b>	<b>Metodología de Cascada</b>	<b>Metodología SCRUM</b>
Inicio	<p>Enunciado del alcance del proyecto.</p> <p>Acta constitutiva del proyecto.</p> <p>Requerimientos de negocio.</p> <p>Matriz de interesados del proyecto</p> <p>Aprobaciones y revisiones.</p>	<p>Sprint 0 – Estimaciones preliminares</p> <p>Creación y modificación de la Pila de Producto.</p> <p>Modificación de pila del producto, así como la pila del Sprint.</p> <p>Identificación de dependencias y de instancias que afectan el producto.</p>
Planeación	<p>Plan de Comunicación</p> <p>Plan de Gestión de Interesados</p> <p>Plan de Gestión de adquisiciones</p> <p>Plan de mitigación de Riesgos</p> <p>Plan de dirección del proyecto y aprobaciones.</p>	<p>Forma de presentar el progreso del proyecto (Tipo de gráficas, datos a presentar, periodicidad.)</p>
Ejecución	<p>Ejecución de tareas de la Estructura del desglose del trabajo</p> <p>Diseño de alto nivel y arquitectónico del software.</p> <p>Creación del código fuente de la aplicación y sus aprobaciones.</p> <p>Ejecución de pruebas unitarias y de usuario.</p>	<p>Reuniones diarias de Sprint.</p> <p>Reuniones de refinamiento del producto.</p> <p>Aprobaciones del dueño del producto.</p>
Monitoreo y control	<p>Tarjetas de roles y responsabilidades para escalaciones.</p> <p>Control integrado de cambios</p> <p>Documento de control del proyecto.</p> <p>Documento de administración de riesgos</p>	<p>Reuniones de revisión del producto.</p> <p>Reuniones de retrospectiva del equipo.</p> <p>Aprobaciones del maestro Scrum y dueño del producto.</p>
Cierre	<p>Obtención de aprobaciones del negocio.</p> <p>Documento de rendimiento de itinerario.</p> <p>Registro de Riesgos e incidentes.</p> <p>Lecciones aprendidas.</p>	<p>Aprobación y estatus "HECHO" de las historias de usuario.</p> <p>Reunión de retrospectiva general del proyecto.</p>

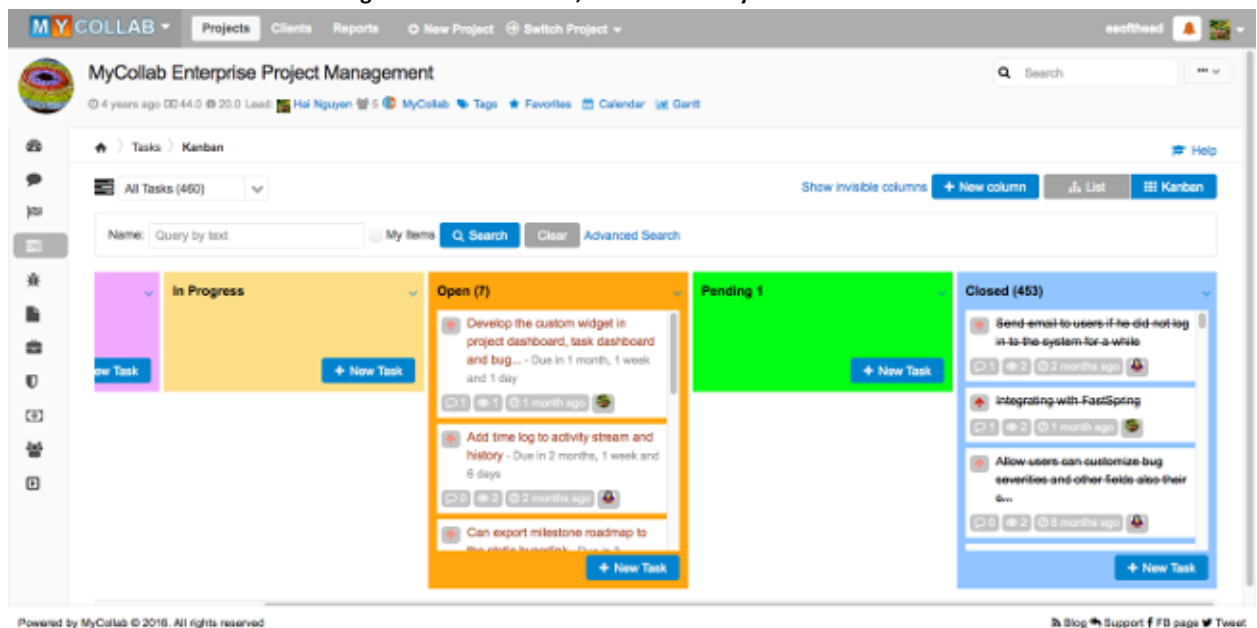
### 8.3.7 Herramientas de gestión de proyectos y sus características.

Existen diversas herramientas de gestión de proyectos que trabajan como un *Toolkit* de fuente abierta.

Cabe señalar que la mayoría de las herramientas para gestionar y administrar proyectos tanto en cascada como en SCRUM, tienen características específicas y agregan valor a la gestión en sí.

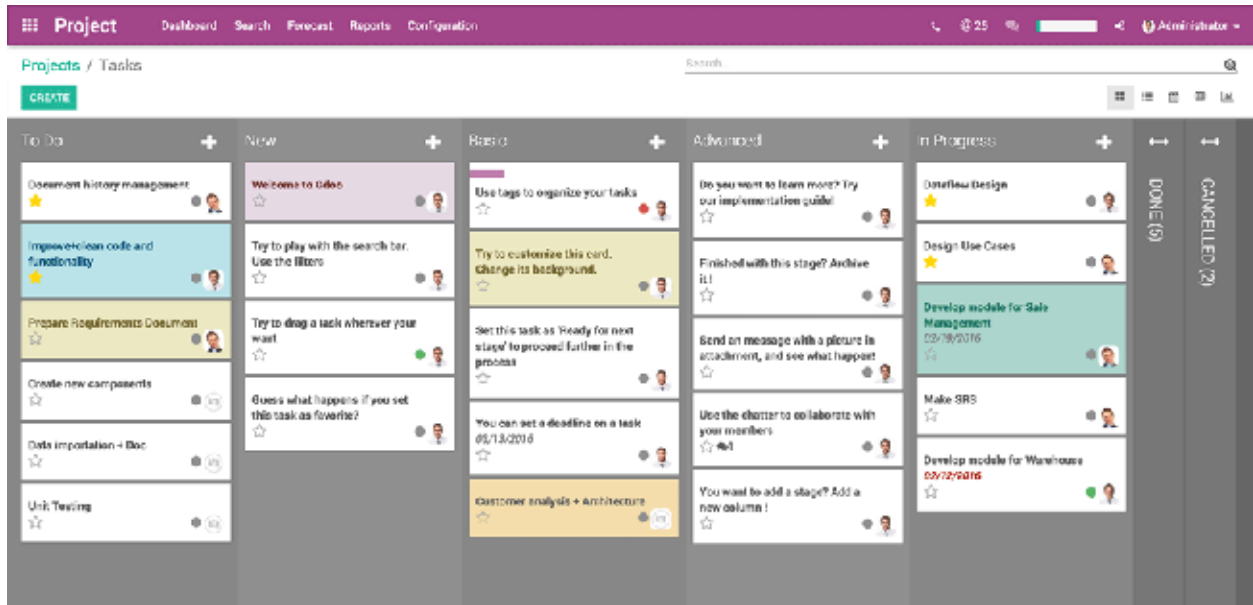
Sin embargo, las herramientas con costo en las licencias suelen ser más utilizadas en las grandes empresas, por ejemplo, en la institución financiera donde colaboro, el uso de la herramienta *Blueprint* y *Quality Center* para desarrollo en cascada, *RTC* y *Jira* para desarrollo SCRUM en esquemas ágiles.

Fig. 33 Panel de control, herramienta My Collab.



**MyCollab** este compuesto de tres módulos de colaboración para pequeñas y medianas empresas: gestión de proyectos, gestión de relaciones con los clientes (CRM) y creación y edición de documentos. Hay dos opciones de licencia: una edición comercial "definitiva", que es más rápida y se puede ejecutar en las instalaciones o en la nube, y la "edición comunitaria" de código abierto, que no tiene una opción de nube y es más lenta, debido a que no utiliza caché de consultas, pero proporciona funciones esenciales de administración de proyectos, incluidas tareas, administración de problemas, flujo de actividad, vista de hoja de ruta y una placa Kanban para equipos ágiles y bajo un esquema de Scrum; esta versión se puede utilizar para gestionar desde la fase de inicio y planeación del Sprint hasta la ejecución, refinamiento, y revisiones del mismo.

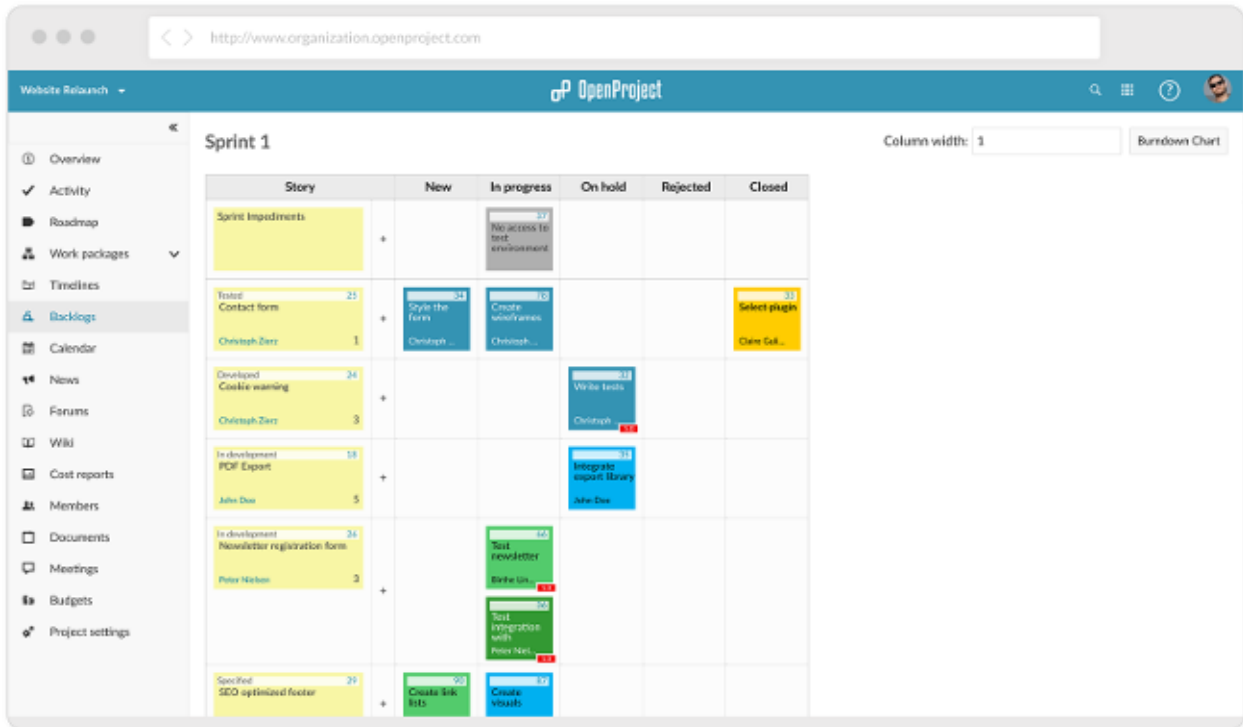
Fig. 34 Panel de control, herramienta Odoo.



**Odoo** se trata de un conjunto completo de aplicaciones empresariales integradas que incluye contabilidad, recursos humanos, sitios web y comercio electrónico, inventario, fabricación, gestión de ventas (CRM) y otras herramientas.

La edición de comunidad de código abierto y gratuito tiene características limitadas en comparación con la suite empresarial de paga. Su aplicación de gestión de proyectos incluye una vista de seguimiento de tareas al estilo Kanban para equipos ágiles, que se actualizó en su última versión, Odoo 11.0, para incluir una barra de progreso y animación para rastrear el estado del proyecto. La herramienta de gestión de proyectos también incluye diagramas de Gantt, tareas, problemas, gráficos y más. Odoo tiene una comunidad próspera y proporciona guías de usuario y otros recursos de entrenamientos.

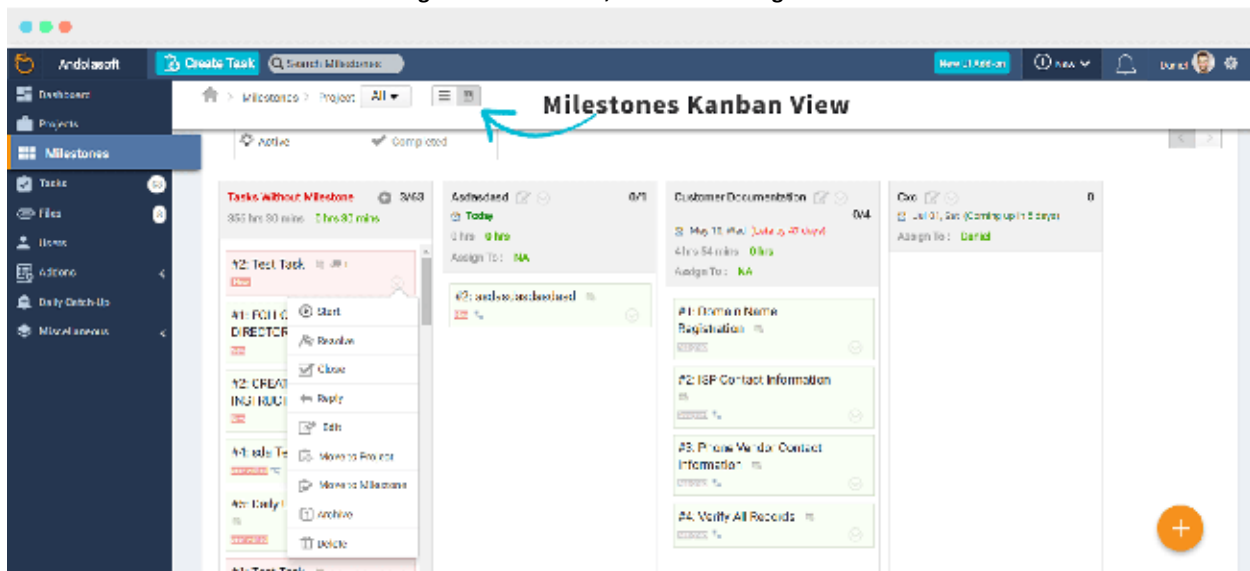
Fig. 35 Panel de control, herramienta Open Project.



**OpenProject** es una poderosa herramienta de administración de proyectos de código abierto que destaca por su facilidad de uso y por las funciones de gestión de proyectos y colaboración de equipos, que abarca desde la fase de inicio, requerimientos, planeación, ejecución, monitoreo y control hasta cierre.

Sus módulos son compatibles con la planificación de proyectos, la programación, la hoja de ruta y la planificación de versiones, el seguimiento del tiempo, los informes de costos, la elaboración de presupuestos, el seguimiento de errores y la administración ágil y de Scrum. Sus funciones ágiles, que incluyen la creación de historias, la priorización de *Sprints* y las tareas de seguimiento, están integradas con otros módulos de *OpenProject*.

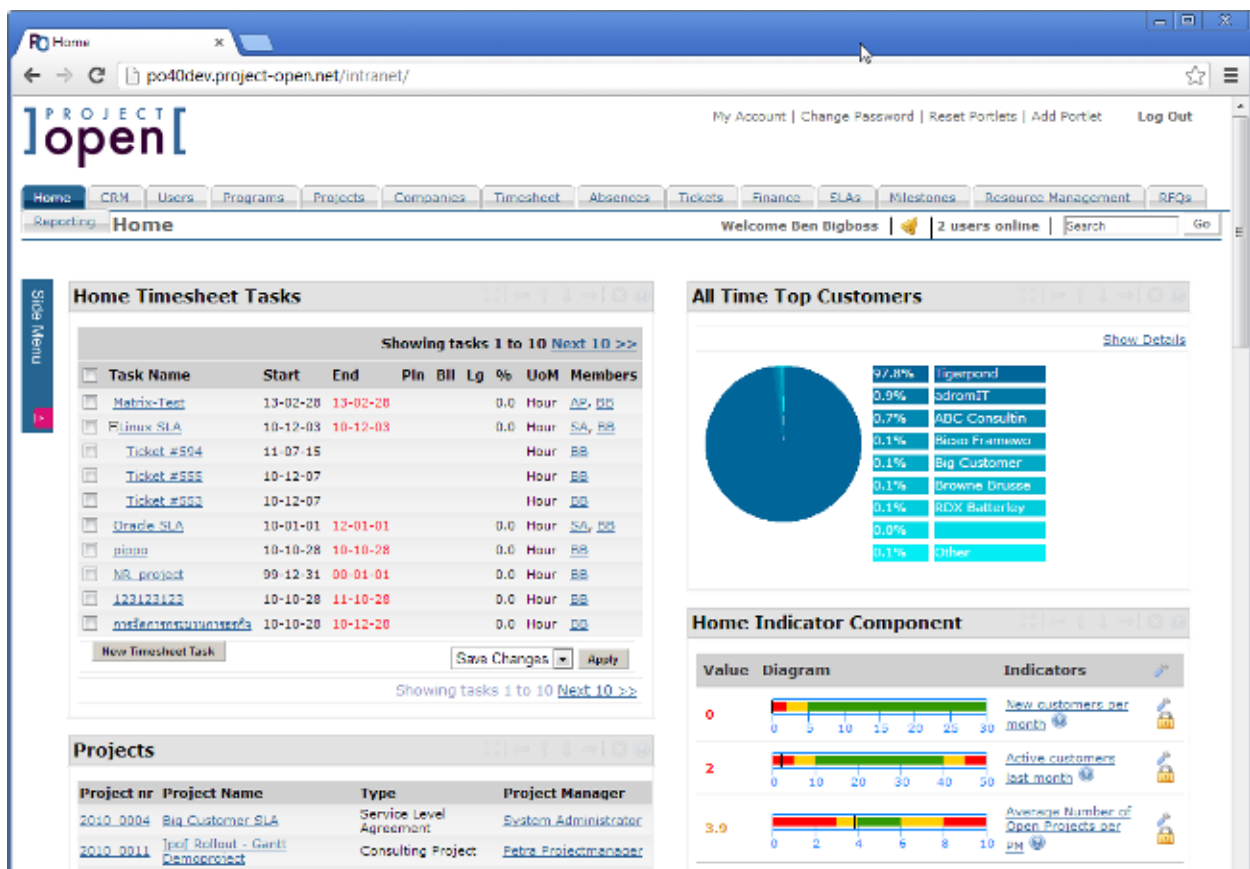
Fig. 36 Panel de control, herramienta Orange Scrum.



**OrangeScrum** adopta metodologías ágiles, específicamente con un tablero de tareas Scrum y una vista de flujo de trabajo de estilo Kanban. Está orientado a organizaciones más pequeñas: autónomos, agencias y pequeñas y medianas empresas.

La versión de código abierto ofrece muchas de las funciones de las ediciones de pagas de *OrangeScrum*, que incluyen una aplicación móvil, la utilización de recursos y el seguimiento del progreso. Otras características, como diagramas de Gantt, registros de tiempo, facturación y administración de clientes, están disponibles como complementos pagos, y las ediciones de pagas incluyen una opción en la nube, y la edición comunitaria no la incluye.

Fig. 37 Panel de control, herramienta Project-Open

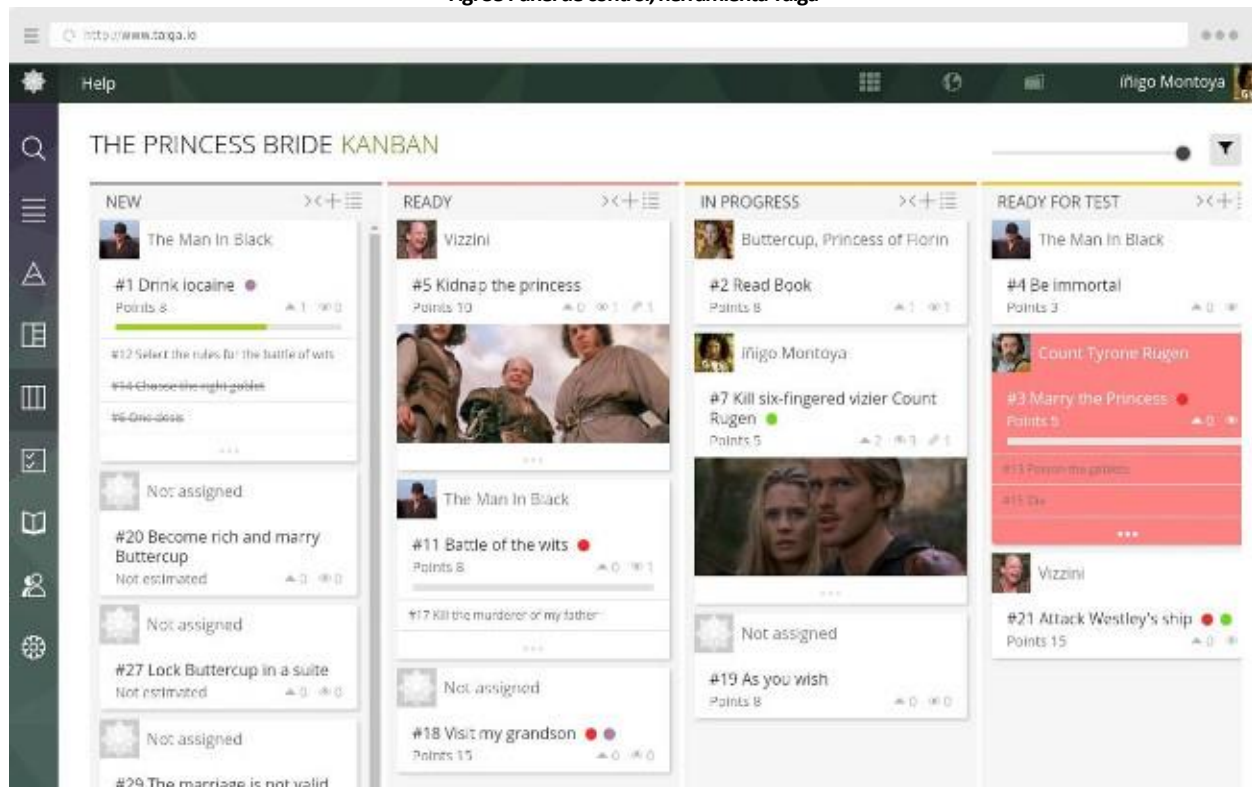


**project-open** es una herramienta de administración de proyectos empresariales de doble licencia, lo que significa que su núcleo es de código abierto, y algunas características adicionales están disponibles en los módulos con licencia comercial. De acuerdo con la comparación del proyecto de las ediciones comunitarias y empresariales, la fuente de código abierto ofrece muchas características para las pequeñas y medianas empresas.

Admite proyectos ágiles con soporte de Scrum y Kanban, así como proyectos clásicos de Gantt / cascada y proyectos híbridos o mixtos; que pueden ir desde las primeras fases, ya que contiene un desarrollador de tareas que se nutre con los requerimientos de negocio en dicha fase, así

como un panel de control para administrar las pruebas en diversos niveles que van desde las pruebas de integración *SIT* hasta *UAT*, e incluso en producción.

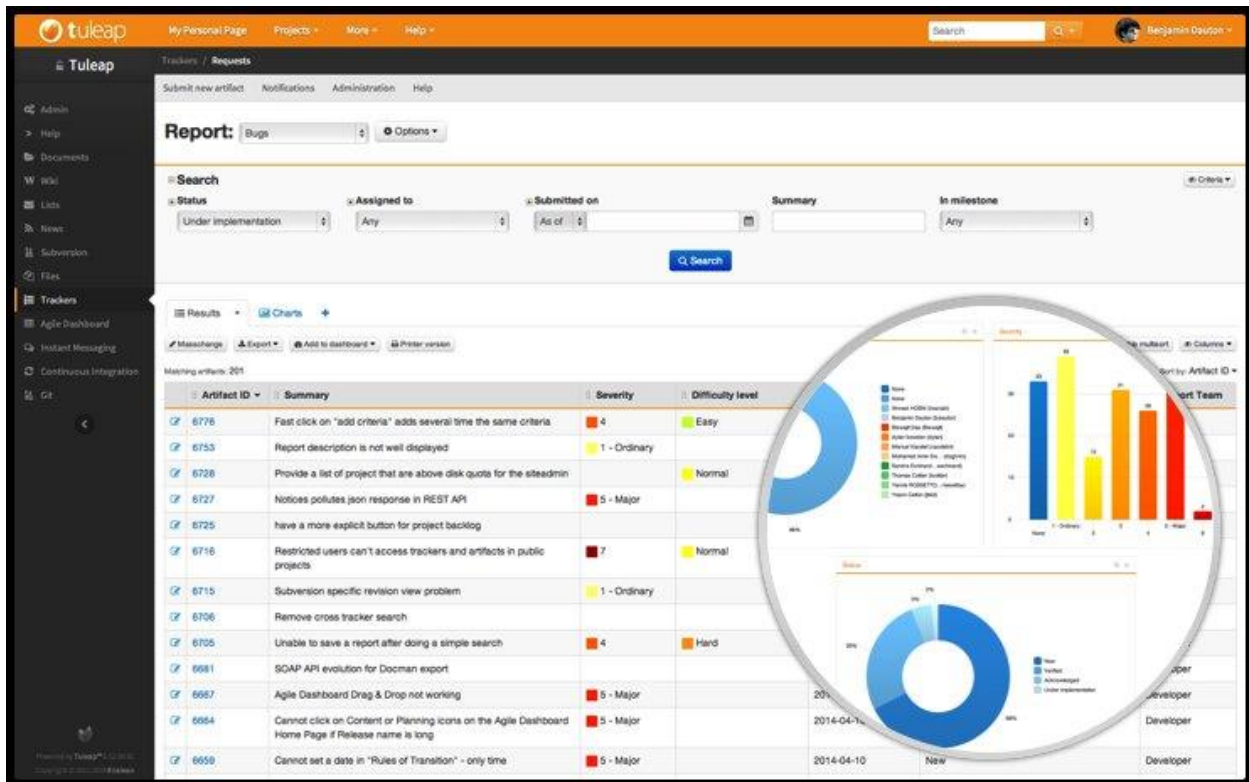
Fig. 38 Panel de control, herramienta Taiga



**Taiga** es una plataforma de administración de proyectos de código abierto que se enfoca en Scrum y en el desarrollo ágil, con características que incluyen una placa Kanban, tareas, *Sprints*, problemas, atrasos y a nivel portafolio (Epopeyas). Otras características incluyen administración de tickets, soporte de múltiples proyectos, páginas wiki e integraciones de terceros.

Es gratis para proyectos públicos, sin restricciones en el número de proyectos o la cantidad de usuarios. Para proyectos privados, hay una amplia gama de planes de pago disponibles bajo un modelo de "*freemium*", pero, en particular, las características del software son las mismas, sin importar qué tipo de plan tenga.

Fig. 39 Panel de control, herramienta Tuleap



Tuleap es una plataforma de administración del ciclo de vida de la aplicación (ALM) que tiene como objetivo administrar proyectos para todo tipo de equipo: pequeño, mediano, grande, en cascada, ágil o híbrido, pero su soporte para equipos ágiles es prominente. Notablemente, ofrece soporte para Scrum, Kanban, *Sprints*, tareas, informes, integración continua, atrasos y más.

Está licenciado bajo GPLv2. Se encuentra disponible más información, incluidas descargas de Docker y CentOS, en su página de Introducción. También puede obtener el código fuente de su última versión, 9.14, en Tuleap's Git.

A continuación, se mencionan las herramientas de código abierto que son utilizadas por empresas pequeñas o experimentales, así como las herramientas de uso común en la industria que ayudan a la gestión de proyectos de desarrollo de Software descritas anteriormente, y también se enlistan en la siguiente tabla las principales características de dichas herramientas, así como las herramientas de licencia comúnmente utilizadas.

Herramienta	Descripción	Ventajas	Desventajas
My Collab	Herramienta con versión gratuita o comercial que ayuda a la gestión de proyectos, gestión de relaciones con clientes y crea	Dos tipos de licencia, la comercial se puede ejecutar desde la nube y es más rápida que la comunitaria. Administra tareas, problemas y	Versión de código abierto, no tiene opción de nube y es más lenta debido a que no utiliza caché de consultas.

	ción y edición de documentos de gestión de Software.	riesgos, flujos de actividad, hoja de ruta y Kanban.	Más enfocado a proyectos híbridos con características de cascada y ágiles.
Odo	Conjunto de aplicaciones empresariales integradas que incluyen contabilidad, recursos humanos, sitios web y comercio electrónico, inventario, fabricación, gestión de ventas (CRM), entre otros.	La herramienta de gestión de proyectos incluye una vista de seguimiento tipo Kanban, para equipos ágiles en la cual incluye una barra de progreso y animación para rastrear el estado del proyecto. Diagramas de Gantt, tareas, gráficos, etc.	Edición gratuita o de código abierto tiene características limitadas en comparación con la suite de paga.
Open Project	Es un software desarrollado en código abierto para equipos de escritorios que permite a los equipos de trabajo que no cuentan con grandes recursos para gestionar sus proyectos con características limitadas.	Predomina por su facilidad de uso y por las funciones de gestión de proyectos y colaboración de equipos.	Medianas y pequeñas empresas con bajo presupuesto y requerimientos de gestión no altos.
OrangeScrum	Adiestrado para la metodología Ágil con un tablero de tareas de Scrum y una vista de flujo de trabajo estilo Kanban.	Permite a los gerentes obtener visibilidad completa en todos los aspectos de sus proyectos, presupuesto, recursos y personal.	Está enfocado a organizaciones más pequeñas.
]Project Open[	Herramienta de doble licencia, y su núcleo está desarrollado en código abierto. Soporte de proyectos orientado en Scrum y Kanban.	Ciertas características adicionales están disponibles en los módulos con licencia comercial.	Ofrece muchas características para las pequeñas y medianas empresas.
Taiga	Es una herramienta enfocada en administración de proyectos ágiles tales como SCRUM o Safe. Con características que incluyen una placa Kanban, tareas, <i>Sprints</i> , problemas, atrasos y a nivel portafolio (Epopeyas). Otras características incluyen administración de tickets, soporte de múltiples proyectos, páginas wiki e integraciones de terceros.	Es gratis para proyectos públicos, sin restricciones en el número de proyectos o la cantidad de usuarios. Para proyectos privados, hay una amplia gama de planes de pago disponibles bajo un modelo de " <i>freemium</i> ", pero, en particular, las características del software son las mismas, sin importar qué tipo de plan tenga	Prácticamente no tiene muchas desventajas desde la perspectiva de ser una herramienta de fuente abierta, sin embargo, llega a tener limitantes en cuestión técnica en comparación con una herramienta como Jira o RTC debido a que los <i>sprints</i> los maneja limitados en el sentido de enriquecimiento.

Tuleap	Plataforma de administración del ciclo de vida de la aplicación (ALM) que tiene como objetivo administrar proyectos para todo tipo de equipo: pequeño, mediano, grande, en cascada, ágil o híbrido.	Su soporte para equipos ágiles es prominente. Notablemente, ofrece soporte para Scrum, Kanban, <i>Sprints</i> , tareas, informes, integración continua, atrasos y más.  Sin embargo, la gestión en cascada es bastante completa incluso en su versión gratuita.	Probablemente una de sus principales desventajas es la forma en que gestiona los proyectos de gran tamaño, ya que tiene una gran dependencia de los requisitos iniciales, si están mal planteados seguramente fracasará el proyecto.
IBM RTC	Es una herramienta para colaborar entre equipos, gestionar código, llevar a cabo <i>standups</i> , planificar <i>sprints</i> , y hacer un seguimiento del trabajo. Disponible en sitio local y en la nube	Gestiona planes, tareas, estados de los proyectos y el vínculo fundamental entre el trabajo entregado y el flujo de trabajo. RTC tiene una flexibilidad única para adaptarse a cualquier proceso, que permite a las empresas adoptar más velozmente los ciclos de lanzamiento y administrar las dependencias entre proyectos de desarrollo, tanto pequeños como complejos.	Realiza tareas adecuadas con elementos de trabajo establecidos de forma predictiva, vinculando planes con la ejecución de desarrollo por lo que, si se hace alguna modificación en el plan, la ejecución se afectará de forma reactiva.  Otra desventaja significativa es el alto precio de la licencia, incluso cuando se contrata un solo producto.
Jira	Aplicación web que permite la organización de proyectos, el seguimiento de errores y la gestión de incidencias. Mecanismo de procesos útil y moldeable de acuerdo con propósitos corporativos específicos. Este Software es compatible con cualquier metodología ágil de gestión de proyectos de desarrollo de software.	Su interfaz es sencilla y el manejo de los indicadores a través de sus diferentes vistas en un <i>dashboard</i> proporciona recursos suficientes, tanto a gerentes como a personal base, para mantenerse al día de lo que está sucediendo en el proyecto.  Administra actividades <i>end to end</i> de product y Proyecto.  Compatible con la mayoría de bases de datos.	Tiene funcionalidades de red social, lo que significa que puedes mantener "posteados" a los interesados en el elemento que se creó, muchas veces si no les pones un @ antes de su nombre, no podrá ser posible mantenerlos informados de los cambios o modificaciones, también es muy difícil borrar elementos creados.
BluePrint Software Sys	Aplicación web de gestión de requerimientos, planes, arquitectura y diseño de Software, enfocado a procesos de análisis de requerimientos de nego-	En su composición técnica es una herramienta muy útil para crear y almacenar artefactos que van desde casos de uso, diagramas diversos, anexos, reglas de negocio,	Licencia de uso de Software con un costo elevado para empresas de pequeño y mediano tamaño; por el tipo de gestión en cascada con un mínimo mensual de

	<p>cio, requerimientos funcionales y no funcionales, así gestión de proyectos en fases de planeación, ejecución, monitoreo y control.</p>	<p>entre otros dentro de requerimientos funcionales y de negocio, con la finalidad de ser compartidos con áreas de aseguramiento de la calidad y pruebas unitarias, así como con el negocio o usuario intermedio para su implementación.</p>	<p>carga de datos, es recomendable en organizaciones que generen alrededor de cinco proyectos trimestrales.</p>
HP ALM	<p><i>HP Quality Center</i> es una herramienta de gestión de pruebas de la división de software de Hewlett-Packard.</p> <p>El objetivo de HP Quality Center es el control de la calidad software, incluyendo la gestión de requisitos, gestión de pruebas y los procesos de negocio para entornos y aplicaciones IT.</p>	<p>Cuenta con diversos módulos como son:</p> <ul style="list-style-type: none"> <li>• <i>Releases</i>: gestiona las diferentes entregas y ciclos del proyecto.</li> <li>• <i>Requirements</i>: gestión de requisitos.</li> </ul> <p><i>Business components</i></p> <ul style="list-style-type: none"> <li>• <i>Test plan</i>: módulo de especificación y diseño de pruebas.</li> <li>• <i>Test resources</i>: recursos y librerías utilizables en las pruebas.</li> <li>• <i>Test lab</i>: gestión de los ciclos de prueba y ejecución de estas.</li> <li>• <i>Defects</i>: gestión de defectos</li> <li>• <i>Dashboards</i>: paneles con métricas</li> </ul>	<p><i>HP Quality Center</i> es un componente del paquete <i>HP Application Lifecycle Management</i>. Lo que ocasiona un precio elevado por sobre la competencia, ya que si el cliente decidiera adquirir el paquete completo tendría que invertir una cantidad considerable, la cual deberá estar seguro de aprovechar al máximo con respecto al alcance de los módulos de la herramienta y las necesidades de gestión.</p>
Planview	<p>Ofrece las soluciones más completas de la industria diseñadas para la planificación estratégica, la gestión de carteras y recursos, la innovación de productos, la gestión de capacidades y tecnología (arquitectura empresarial), la entrega Lean y Agile, y el trabajo colaborativo y la gestión de proyectos.</p>	<p>Líder mundial en trabajo y gestión de recursos.</p> <p>Sus soluciones abarcan todas las clases de trabajo, recursos y organización para abordar las diferentes necesidades de equipos, departamentos y empresas diversos y distribuidos.</p> <p>Planview atienden a 5,000 clientes en todo el mundo a través de una cultura de liderazgo tecnológico innovador, profunda experiencia en el mercado y comunidades altamente comprometidas.</p>	<p>En sus versiones básicas no cuenta con interfaces en Web.</p> <p>La información generada en cada una de sus etapas debe adecuarse para transferirse a otros sistemas, generando actividades.</p> <p>No se contemplan procesos para revisar el impacto de los resultados en la organización.</p>

En la siguiente tabla se comparan todas las herramientas mencionadas con antelación y una comparativa entre cada una. También se especifica la eficiencia en relación con la utilización, elementos, características y efectividad con respecto a su uso basado en la experiencia de algunos administradores de proyectos que fueron entrevistados (ver anexo 1).

Las herramientas de eficiencia media cumplen en un rango del 50% al 79% con los requisitos mínimos necesarios para administrar un proyecto *end to end*.

Aquellas que tienen eficiencia alta, cumplen en un rango del 80% al 100% con los requisitos necesarios para gestionar y administrar proyectos ya sea en cascada o Agile.

Herramienta	Uso	Eficiencia
<b>My Collab</b>	Cascada	Media
<b>Odo</b>	Agile	Media
<b>Open Project</b>	Agile	Alta
<b>OrangeScrum</b>	Agile	Media
<b>]Project Open[</b>	Cascada	Media
<b>Taiga</b>	Agile	Alta
<b>Tuleap</b>	Cascada y Agile	Alta
<b>IBM RTC</b>	Agile	Media
<b>Jira</b>	Agile	Alta
<b>BluePrint Software Sys</b>	Cascada	Media
<b>HP ALM</b>	Cascada	Media
<b>Planview</b>	Cascada y Agile	Alta

#### 8.4 Conclusiones:

Se planteó como objetivo primordial de este trabajo, la implementación y aplicación de la metodología *Scrum* en la administración de proyectos de desarrollo de Software; para lo cual se cumple con dicho objetivo de manera sistémica, ya que se comparan tanto la metodología de gestión en cascada como la metodología ágil de Scrum, resaltando ventajas, desventajas y recomendaciones en el uso de ambas, pero sobre todo la forma de gestión integral bajo un esquema ágil tanto en el proyecto expuesto de mejora del módulo de pagos y transferencias de un aplicativo web, como de cualquier otro proyecto relacionado al desarrollo de Software. Sobre este punto, es trascendental resaltar las limitantes del alcance de este trabajo, ya que si bien se buscaba destacar las bondades del uso de Scrum en la gestión de cualquier tipo de proyecto; se acotó dicho objetivo con el fin de presentar elementos sustentables sobre la experiencia del uso de esta metodología.

También con respecto al objetivo particular de referenciar un marco para todos aquellos interesados en la gestión de proyectos de software, se cumplió con dicha premisa pues en el desarrollo

de la propuesta (Sec. 8.3), se mencionan detalladamente las fases de administración de proyectos según el PMBOK, así como sus áreas de conocimiento y procesos, pero adaptados a un marco de referencia tanto para “Cascada” como para “Scrum”; así como diversas herramientas utilizadas en la gestión de ambas metodologías.

De manera concreta, y como resultado del desarrollo de este documento, se recomienda para la gestión de proyectos de Software, realizar un análisis costo beneficio sobre las ventajas y desventajas, escenarios, recursos, limitantes y oportunidades para cada caso particular de gestión.

Sin embargo, con base a mi experiencia profesional, recomiendo la utilización de SCRUM como metodología en la gestión de proyectos para desarrollo de Software por tres puntos principalmente: En primer lugar (como se expuso a lo largo del TOG) esta metodología trabaja bajo los principios de *Agile*, y la forma de gestionar tanto la planeación, ejecución e implementación se vuelve más eficiente; siempre teniendo en mente las necesidades y satisfacción del cliente como nuestro principal objetivo.

En segundo lugar, al momento de liberar el Software funcional a producción, este proceso puede ser hasta tres veces más rápidos que la metodología de cascada, debido a las continuas iteraciones que desarrollan el producto final paulatinamente, teniendo en cuenta un beneficio mayor al obtener un ROI (Retorno de Inversión) más rápido y también con menores costos.

En tercer lugar, recomiendo el uso de SCRUM, debido a que en la actualidad se está llevando a cabo una enorme transformación tecnológica, la cual apunta indiscutiblemente a los enfoques ágiles de gestión y administración de proyectos de desarrollo de Software.

Con lo antes mencionado, es altamente probable que el administrador de proyectos decida tomar como base los fundamentos de gestión ágil de *Scrum*, pues las ventajas y beneficios se vuelven palpables en muy corto tiempo debido a que esta metodología tiene un enfoque cien por ciento canalizado a la satisfacción del cliente en interacciones cortas y efectivas donde el valor del producto se puede palpar desde las primeras fases.

Otra recomendación importante, es adoptar las mejores prácticas de *Scrum* en dado caso que la alta gerencia decida mantener un enfoque de cascada o incluso administrar sus proyectos con metodologías híbridas, esto con la finalidad de obtener beneficios que brida *Scrum* como primera impresión, y con este supuesto mantener una tendencia de transformación hacia lo que actualmente la industria está adoptando y que por ende genera mayor valor para la compañía.

Finalmente es importante mencionar que, si bien el objetivo de este trabajo es exponer todos los puntos positivos y negativos de la metodología Scrum, no se busca en ningún momento persuadir al lector a optar tal o cual metodología, simplemente se busca exponer desde una perspectiva teórico-práctica las bondades de la adopción de Scrum bajo la premisa de un proyecto migrado que se desarrolló positivamente en dicha metodología.

Durante el desarrollo de este documento surgieron diversas dudas con respecto a la presentación comparativa de dos metodologías poderosas y populares en la gestión de proyectos de Software, sin embargo poniendo como base la metodología del PMP que en ningún caso es limitativa con respecto a la forma de gestionar proyectos (ya sea la tradicional metodología progresiva de cascada o la radical metodología ágil de SCRUM), se logró presentar a lo largo de este trabajo, las ventajas, desventajas, características y tendencias no solo de dichas metodologías, sino también otras tendencias que en esencia se asemejan pero tienen características distintas.

Otro punto positivo que resaltar fue la forma de desarrollar el conocimiento a lo largo de este trabajo; ya que se dividió en fases basadas en la propuesta del PMBOK. Se comenzó explicando los hitos de la fase de iniciación tanto para *Waterfall* como *SCRUM*, también se presentaron

diversas herramientas de gestión de proyectos que ofrecen un híbrido de ambas metodologías y su respectiva utilización en la fase de inicio.

En la fase de planeación y ejecución, también se mencionan los puntos favorables y desfavorables de ambas metodologías, así una breve recomendación de herramientas de código abierto y otras con costo en la licencia. En este punto es importante resaltar que entre la gestión de proyectos de Software por medio de cascada y de *Scrum* difiere en gran medida, sobre todo al momento de integrar y aprobar los principales hitos; por un lado en el método progresivo de cascada, se deberá crear un plan de dirección de proyectos muy bien estructurado y completo, y una vez comenzada la ejecución del mismo, se vuelve más costoso en tiempo y presupuesto cualquier tipo de cambio; se deben crear requerimientos específicos para ellos y obtener aprobación por parte de los principales interesados del proyecto.

Por otro lado, cuando la planeación y ejecución se llevan a cabo bajo un esquema ágil de *Scrum*, ambas fases se combinan, inclusive con el monitoreo y control, y con respecto a los cambios, estos se realizan de forma natural en el proceso de gestión.

Al crear una pila del producto que puede ser desarrollada bajo la premisa de entregar valor al cliente desde las primeras fases, hace más factible adaptar y moldear el producto final a entregar de acuerdo con los comentarios y retroalimentación del cliente.

Sobre la fase de monitoreo y control en cascada, se lleva a cabo desde el momento que se comienza a ejecutar la codificación del Software hasta la fase de pruebas unitarias por parte del equipo de *testing* de negocio. Sobre esta fase, es importante resaltar la obtención de aprobaciones tanto del código funcional como del cumplimiento de los requerimientos no funcionales o de soporte.

Con respecto a la fase de cierre, cabe mencionar que no existe una gran diferencia entre las dos metodologías, producto de la comparación de este trabajo; incluso ambas cumplen de cierta manera lo que se menciona para esta fase en el PMBOK. También es importante resaltar que basado en la experiencia personal de la implementación de *Scrum* en la institución donde laboro, tanto la alta gerencia como el equipo de negocio, han optado por mantener un enfoque de cascada cuando se cierran los proyectos, ya que elementos como bitácora de lecciones aprendidas, documentos de aprobación y aceptación final, análisis cuantitativo y cualitativo de los recursos utilizados en el proyecto, entre otros.

## 11. Bibliografía

1. Abrahamsson, P., Salo, O., Ronkainen, J. y Warsta, J. (2002). Agile software development methods Review and analysis. Nueva Inglaterra: VTT Publications.
2. Calos Ble. (2013). Libros Web. Creative Commons Atribución SinDerivadas 3.0. [En línea] <[http://librosweb.es/libro/tdd/capitulo\\_1/modelo\\_en\\_cascada.html](http://librosweb.es/libro/tdd/capitulo_1/modelo_en_cascada.html)> [2017, abril 05].
3. Carolina Zibert, (2006). Ciclos de vida de Ingeniería de Software [En línea] <[carolina.terna.net/ingsw2/Datos/Cascada-ModeloV.doc](http://carolina.terna.net/ingsw2/Datos/Cascada-ModeloV.doc)> [2017, marzo 12].
4. Cleland, D. y Ireland, L. (1999). Project management: strategic design and implementation. E.U.A: Mc. Graw Hill 5ª edición.
5. Harold Kerzner. (2013). Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 11th Edition. E.U.A: Wiley
6. Thamhain, H. Van Nostrand, R. (1986). Project Management Operating Guidelines. E.U.A.: Van Nostrand Reinhold Company
7. Saladis, F. y Kerzner, H. (2009). What Executives Need to Know about Project Management. E.U. A: Wiley
8. Fowler, M., Beck, K., Brant, J. (1999). Refactoring: Improving the Design of Existing Code. E.U.A: Addison-Wesley.
9. Pressman Roger S. (2003). Ingeniería del software. México: Mc Graw Hill, 5ª edición.
10. PMI. (2017). Guía de los fundamentos para la dirección de proyectos PMBOK. E.U.A: Global Standard 6ª edición
11. Sommerville, I. (2005). Ingeniería de software. Madrid: Pearson Addison Wesley 7ª edición.
12. Solares, P. y Romero, E. (2014). Gobierno de tecnologías de información. Bolivia: Ecorfan.
13. Kerzner, H. (2017). Project management: a systems approach to planning, scheduling, and controlling. E.U.A: Wiley 12ª edición.
14. Martin Series, Robert. (2005). Agile Estimating and Planning. NJ. E.U.A. Pearson.
15. Hope, J y Fraser, R. (2003). Beyond Budgeting. Boston, E.U.A. Harvard Business School Press.
16. Kniberg, H. Skarin, M. (2010). Kanban y SCRUM – Obteniendo lo mejor de ambos. E.U.A. C4 Media Inc.
17. Mulcahy, R. (2013). Preparación para el Examen PMP. E.U.A. RMC Publications.
18. PMI. Agile Alliance (2017). Agile Practice Guide. E.U.A.: Global Standard 6ª edición.

## Glosario

PMP – Project Management Professional  
PMBOK – Project Management Body of Knowledge  
SCRUM – Marco metodológico de administración de proyectos ágiles.  
Sprint – Iteración de tiempo que se utiliza para ejecutar tareas en SCRUM.

## Anexos

### Anexo 1 Cuestionario sobre el uso de SCRUM.

*Cuestionario realizado a 2 Maestros SCRUM. Alejandra Vieyra (AV) y Francisco Alvarado (FA).*

#### 1. ¿Cuál es la duración de tus iteraciones?

SCRUM recomienda que las iteraciones duren hasta un mes, aunque existen diferentes opiniones de cuál es la duración óptima. Probablemente no haya una duración única recomendable para diferentes equipos y proyectos, por lo que la tendencia más aceptada corresponde a que cada equipo escoja la duración de sus iteraciones como parte del aprendizaje y evolución usando este proceso.

Una duración demasiado corta puede introducir una sobrecarga innecesaria de gestión, pruebas o reuniones, mientras que una duración excesiva puede hacer “perder el ritmo” tanto dentro del equipo como entre el equipo y el cliente.

Mi experiencia es que la mayoría de los grupos que han madurado SCRUM suelen ir a una duración de 2 semanas, combinándolas con iteraciones de 3 semanas en caso que la iteración produzca un reléase. He visto otras duraciones y sí tengo constancia que casi todas las que iban más allá de 4 semanas provocaban problemas de desviaciones y bajadas en la confianza del equipo.

AV: 1 a 2 semanas

FA: 3 a 4 semanas

#### 2. ¿Cuál es el tamaño de un equipo?

El tamaño de los equipos normalmente queda fuera del alcance de decisión de los equipos y es un factor ligado usualmente al tamaño de la empresa. Las empresas más pequeñas suelen realizar proyectos de menor tamaño y su dimensión no les permite formar equipos grandes.

En el caso de los participantes, es unánime el tamaño pequeño de los equipos, aunque este dato no puede correlacionarse con otros como el tamaño de la empresa o de los proyectos al carecer de estos últimos datos.

AV: 3

FA: 4

### 3. ¿Cuántos proyectos has desarrollado con SCRUM?

Como se menciona en otras preguntas, la madurez en el uso de SCRUM suele influir en algunos parámetros de uso, como el tamaño de las iteraciones o mejora en la confianza que éste genera en la organización.

En este caso, ambos participantes en la encuesta han usado bastante SCRUM, aunque el planteamiento de las preguntas no deja claro si alguno de estos no ha usado este proceso en proyectos reales.

AV: Más de 5

FA: Más de 8

### 4. ¿Cuáles herramientas de gestión de proyectos conoces y cuál es su eficacia en tu experiencia?

- My Collab
- Odo
- Open Project
- OrangeScrum
- ]Project Open[
- Taiga
- Tuleap

## Anexo 2 Planilla “Plan de Proyecto”

<b>Asunto:</b>	<b>Plan de Proyecto</b>
<b>Fecha:</b>	<b>dd/mm/yyyy</b>

### Historial de Revisiones

<b>Fecha</b>	<b>Versión</b>	<b>Descripción</b>	<b>Autor</b>
dd/mm/yyyy	01	Versión inicial	[Nombre Autor]

## PLAN DE PROYECTO

### Fundamento

El plan de proyecto es “un documento formalmente aprobado que define cómo se ejecuta, supervisa y controla un proyecto”

El mismo incluye:

- Procesos de dirección seleccionados y nivel de implementación de cada uno
- Principales documentos a elaborar
- Descripción de técnicas y herramientas que se van a utilizar
- Como se ejecutará el trabajo a ser realizado
- Cómo se ejecutará el control integrado de cambios
- Cómo se realizará la gestión de la configuración y cómo se asegurará la integridad de la línea base de tiempo, costo y calidad (objetivos de calidad del proyecto)
- Técnicas de comunicación
- Ciclo de vida
- Revisiones claves

Planes subsidiarios que contiene:

- Plan de gestión de la integración
- Plan de gestión del alcance
- Plan de gestión del cronograma
- Plan de gestión de costes
- Plan de gestión de la calidad

- Plan de gestión del proceso
- Plan de gestión del personal
- Plan de gestión de las comunicaciones
- Plan de gestión de riesgos
- Plan de gestión de las adquisiciones

### Principales técnicas y herramientas a utilizar para la dirección del proyecto

#### Técnicas:

- Juicio de expertos
- Técnica del documento único
- Manejo de conflictos
- Estimación por analogía
- Gestión de la configuración
- Control de cambios
- Valor ganado
- Planillas de riesgos

#### Herramientas:

- Open Office o similar que trabaje con formato ODF
- OpenProj - <http://openproj.org/>
- Correo electrónico

### Principales técnicas y herramientas a utilizar para la generación del producto

Depende de cada producto. Si es un producto de software, a vía de ejemplo.

#### Metodologías de desarrollo de software, incluyendo:

- Casos de uso
- Trazabilidad desde el análisis de impacto hasta la puesta en producción
- Gestión de la configuración
- Casos de prueba

#### Herramientas:

- xxx
- Gestión de la configuración

### Principales documentos a elaborar

- Acta de Constitución
- Enunciado de alcance
- Matriz de interesados

- Plan de proyecto, conteniendo:
  - Control integrado de cambios
  - Aceptación formal de entregables
  - Transferencia tecnológica
  - Lista de entregables
- EDT (Estructura de Desglose del Trabajo)
- Roles y responsabilidades del proyecto
- Matriz de comunicaciones
- Cronograma
- Presupuesto
- Registro de riesgos
- Registro de actividades
- Registro de incidencias
- Memo de reuniones de trabajo
- Informe de avance
- Lecciones aprendidas
- Depende del producto que se genera en el proyecto

## Anexo 3. Fases del Proyecto

### Revisiones claves

Las revisiones son procesos en los cuales se analizan los entregables clave y se valida su buen funcionamiento, como base para la construcción de nuevos entregables.

### Plan de gestión de alcance

Describe los procesos de gestión de alcance, la forma de ejecutarlos y las herramientas que se utilizan.

En particular, se debe especificar siempre que se va a realizar la EDT.

### Plan de gestión del cronograma

El Plan de gestión del cronograma establece los “criterio y las actividades para desarrollar y controlar el cronograma del proyecto”

Contiene la descripción de:

- Formato del cronograma, herramienta a utilizar, criterios para su desarrollo
- Criterios de seguimiento a utilizar
- Procesos de identificación de actividades, establecimiento de secuencia, recursos, duración, desarrollo y control del cronograma

### Plan de gestión de costos

Fija el formato y establece los criterios para estructurar y controlar los costos.

Contiene una descripción de:

- Formato de reporte de costos
- Unidades de medida
- Umbrales de control
- Reglas de valor ganado – 0 – 50 - 100
- Procesos de estimación, presupuestación y control

### Plan de gestión de la calidad

Describe como se implementará la política de calidad. Contiene las normas de calidad relevantes para el proyecto y la forma de satisfacerlas.

Contiene:

- Métricas de calidad (densidad de defectos, disponibilidad del sistema, cobertura de las pruebas)

### Plan de mejoras del proceso

Detalla los pasos de análisis de procesos que facilitarán la identificación de actividades que no agregan valor, y permitirá que del proyecto piloto quede definido un proceso que sea repetible y que se pueda mejorar en cada iteración.

Contiene información de:

- Límites del proceso – cual es el alcance del proceso a realizar
- Configuración del proceso – una descripción de los pasos
- Métricas del proceso – las métricas a elaborar como forma de medir los resultados
- Objetivos de mejora acordados – objetivos que se aplican a cada una de las iteraciones

#### Plan de gestión del personal

Determina los roles y las responsabilidades, cómo y cuándo ingresan las personas al proyecto y cuando se liberan, normas de funcionamiento, políticas de seguridad y confidencialidad, capacitación. Hay planilla aparte para describirlo.

#### Plan de gestión de las comunicaciones

Describe las necesidades de información y comunicación de los interesados: qué información, cuándo, cómo, quién la suministra y en que formato. Hay planilla aparte para describirlo.

También detalla el proceso de escalamiento de los asuntos que no se pueden resolver dentro del equipo de trabajo.

#### Plan de gestión de riesgos

Detalla cómo se estructurará la gestión de riesgos del proyecto.

Contiene la descripción de:

- Metodología, herramientas y fuentes de información a utilizar, formatos de informe
- Roles y responsabilidades
- Preparación del presupuesto
- Características del seguimiento: periodicidad, participantes, disparadores
- Categoría de riesgos
- Formato de la matriz de probabilidad e impacto
- Niveles de tolerancia admitidos

Hay planilla aparte para la identificación de riesgos y para el repositorio de riesgos de la institución.

#### Plan de gestión de las adquisiciones

El plan de gestión de las adquisiciones identifica las necesidades de adquirir bienes y servicios y detalla cómo se efectuarán las compras.

Describe:

- Tipos de contratos
- Criterios de evaluación
- Pliegos y contratos
- Gestión de proveedores

- Restricciones y supuestos
- Garantías de cumplimiento de contrato
- Cláusulas de propiedad
- Hitos principales

El plan de gestión del contrato describe como se administrarán los contratos del proyecto, incluyendo documentación necesaria, requisitos de rendimiento, criterios para el contrato del mantenimiento.

# Anexo 4 Esquemas de metodología PMP y SCRUM.

