**Grupo Consigue Ventas E.I.R.L.**

**ARQUITECTURA DE DESARROLLO ÁGIL DEL SOFTWARE SISTEMA WEB DE ASISTENCIA DE LA EMPRESA CONSIGUE VENTAS**

**-**

**Jefe directo:** Jhoel Fernandéz Alvarado

**Área: Sistemas**

Departamento Estratégico

**LIMA – PERÚ**

**Junio, 2023**

| **CONTROL DE VERSIONES** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Versión** | **Hecha por** | **Revisada por** | **Aprobada por** | **Fecha** | **Motivo** |
| 1.0 | Equipo | Jhoel Fernandéz | Jhoel Fernandéz | 25/05/2023 | Sistema Web |
| 2.0 | Equipo | Jhoel Fernandéz | Jhoel Fernandéz | 26/05/2023 | Sistema Web |

# RESUMEN

Aplicar el marco ágil a un sistema web de asistencia de trabajadores implica utilizar un enfoque flexible y colaborativo en el desarrollo e implementación del proyecto. En contraposición a un enfoque tradicional y lineal, el enfoque ágil permite una adaptación rápida a los cambios y la entrega de valor de forma incremental. Por lo tanto, nuestro objetivo será incorporar este marco de trabajo al proyecto y desarrollar cada aspecto establecido en él, previo a una definición detallada de la metodología que se aplicará dentro de la empresa Consigue Ventas.

# INTRODUCCIÓN

La arquitectura de desarrollo ágil del software para el sistema web de asistencia de la empresa “Consigue Ventas” se plantea como una solución para abordar los desafíos y las necesidades de calidad en el desarrollo de aplicaciones y sistemas. En un mercado cada vez más competitivo, es fundamental ofrecer soluciones de software que cumplan con las expectativas de los clientes y brindan un nivel de servicio óptimo.

Por lo anterior, se puede entender que surgen diversos cuestionamientos acerca del manejo o la funcionalidad de este tipo de aplicativos, donde se podrían requerir de nuevos mecanismos para encontrar la optimización esperada dentro de los procesos. Es así, para dar inicio a este artículo se planteó la siguiente pregunta, teniendo como referencia la empresa Consigue Ventas: ¿Es posible implementar una arquitectura (Framework) utilizando metodologías ágiles y principios lean, en marco a la producción de software en la empresa Consigue Ventas, para el aumento de la calidad y nivel de servicio ofrecido a sus clientes(Colaboradores)?.

El objetivo del presente informe se centra en diseñar una arquitectura marco para la ejecución del proceso productivo de desarrollo de software basada en metodologías ágiles y principios lean, aplicables en Consigue Ventas. Seguidamente, se busca diseñar una arquitectura o un framework basado en metodologías ágiles (SCRUM) y conceptos Lean (Lean software Development), que puedan ser aplicados como herramientas para los diferentes procesos productivos del desarrollo de software.

De este modo es como se pretende construir el presente documento basado en la operación de la empresa Agencia Consigue Ventas E.I.R.L, sustentado en las falencias actuales que presenta su operación, desencadenando así, la afectación de su nivel de servicio de cara al cumplimiento oportuno y con la calidad exigida por parte de sus colabores.

Existen ciertas metodologías que implementadas total o parcialmente permiten generar un marco aplicable en empresas, las cuales generan mejoras en los procesos productivos de la creación de software y sus diferentes subprocesos, garantizando así, la calidad y la generación de valor en los aplicativos que se entregan a sus colaboradores. Sustancialmente, se pretende la inclusión de conceptos y metodologías basadas en la agilidad (Scrum, Kanban, Lean software development), que a su vez sustentan el concepto metodológico del proyecto y que se describen a continuación.

**Scrum**: Es una metodología ágil con prácticas específicas basada en los principios del desarrollo gradual e interactivo. Sus primeros autores fueron ken shwaber, Mike beedle y Jeff Sutherland Se enfoca en la colaboración de todo un equipo o varios equipos en la consecución de un objetivo. Scrum es un marco de trabajo adaptativo enfocado en la entrega de valor constante. Se puede implementar en proyectos grandes, medianos y pequeños además de diferentes actividades de la vida cotidiana.

**Kanban**: Palabra de origen Japonés que significa tarjeta con signos o señal visual. Kanban es un método para la gestión del trabajo a realizar en los procesos productivos. Es implementada

por primera vez en Toyota Production System (TPS) a finales de la década de los 40s, con el objetivo de conseguir un flujo lineal y sin interrupciones en las operaciones.

**Lean software development**: Metodología basada en el concepto Lean desarrollado en Toyota en los años 50 posterior a la segunda guerra mundial. Lean software development fue adaptada y desarrollada por Mary y Tom Poppendieck aplicada en el desarrollo de software. Esta metodología se fundamenta en el concepto de “eliminar los desperdicios” (Cualquier tipo de artefacto o proceso que no genere avance o mejora) basado en la mejora continua en los diferentes procesos y subprocesos que intervienen en la generación de software.

Este tipo de prácticas derivadas de la metodología JIT, implementadas de una manera adecuada en conjunto, pueden generar un flujo de trabajo ágil, flexible, y lineal con la menor cantidad de desperdicio posible. El marco se conforma de la implementación de los principios de las metodologías en los procesos macro de producción y sus subprocesos.

El presente informe se desarrolla basados en el análisis del proceso productivo de la empresa ConsigueVentas es una agencia de marketing digital que se dedica a la realización de embudos de ventas online y creación de páginas web para ayudar a empresas a alcanzar su máximo potencial en ventas y fidelización de clientes. En concordancia con las herramientas, datos y acceso a información necesaria para el desarrollo del artículo, las cuales serán suministradas por la empresa ConsigueVentas y por la documentación oficial acerca de los temas, metodologías y prácticas que se analizaran en el desarrollo del documento.

El presente artículo se organiza en diferentes capítulos, iniciando con la descripción de las diferentes metodologías involucradas en el capítulo Materiales y Métodos. Continuando en el capítulo 2, Resultados y Discusión, en donde se desplegará el diagnóstico realizado a la compañía, el desarrollo de la arquitectura y su forma de aplicación en la empresa ConsigueVentas. El artículo finaliza con las conclusiones

# 1. MATERIALES Y MÉTODOS

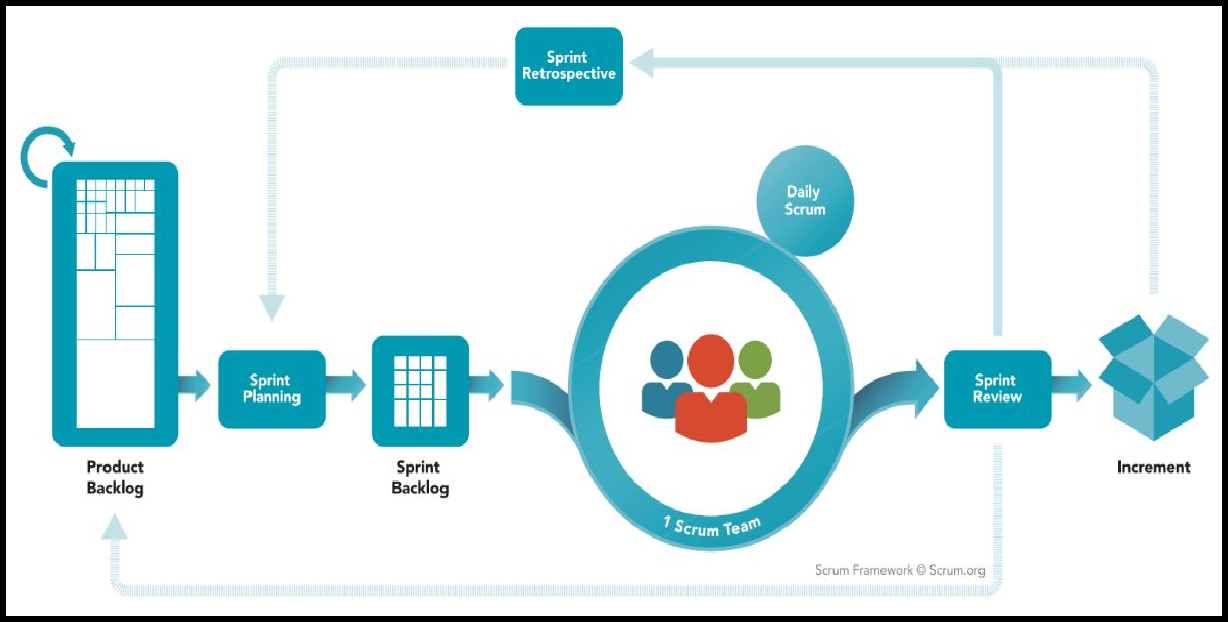
En la elaboración de este proyecto, se utilizaron recursos y métodos para desarrollar la arquitectura propuesta, tomando en cuenta los conceptos de las herramientas previamente mencionadas. Además, se buscó obtener información de fuentes confiables que brindarán un panorama preciso de diversos aspectos dentro de la empresa Consigue Ventas, con el fin de realizar un diagnóstico adecuado y comprender en detalle los elementos que requerían mejoras. Durante la construcción de este nuevo modelo, se emplearon diferentes herramientas y metodologías, tales como:

**Metodología Scrum**: Dentro de la arquitectura propuesta, se ha incorporado el marco de trabajo SCRUM como un enfoque para gestionar los requerimientos y el desarrollo de productos derivados de los distintos proyectos en curso. La metodología ágil de SCRUM se ha aplicado para coordinar el flujo de trabajo, el cual puede provenir de diversas fuentes y no es constante.

Mediante la implementación de SCRUM, se busca generar un mayor valor para los clientes a través de sus distintas etapas y roles. El objetivo principal es adaptar los principios y las buenas prácticas de esta metodología, lo que permitirá al equipo de trabajo desarrollar software de manera ágil y responder a los cambios, incluso en las etapas más avanzadas de los proyectos.

SCRUM se basa en una generación constante e incremental de valor para el cliente. Para lograrlo, propone la formación de equipos autoorganizados y adaptativos, fomentando una cultura de agilidad a través de sus artefactos, roles y eventos. La utilización de sprints asegura la entrega iterativa de valor, acelerando así el proceso productivo.

* **Roles:** Los participantes clave en la metodología tienen roles y responsabilidades definidos, desempeñando funciones específicas para generar valor en cada sprint o iteración.
  + **Scrum Master:** El Scrum Master cumple el rol de líder en Scrum, estableciendo los parámetros y asegurando el cumplimiento completo de la metodología. El Scrum Master está involucrado en el día a día del proyecto y es responsable de impulsar el progreso.
  + **Product Owner:** El Product Owner desempeña el papel de arquitecto, siendo responsable de traducir las necesidades de los clientes. Su objetivo principal es maximizar el valor del producto. En algunas ocasiones, el Product Owner toma decisiones en representación del cliente.
  + **Equipo de Desarrollo:** El equipo de desarrollo es el responsable de ejecutar los requerimientos y producir los artefactos. Son quienes aportan directamente al valor incremental del proyecto.
* **Ritos:** Son las ceremonias o rituales que se deben llevar a cabo en la metodología. A medida que el equipo va ejecutando estos rituales, los interioriza y se vuelve experto en ellos.
  + **Daily:** Una reunión diaria en la que todo el equipo realiza una revisión de avance. Cada miembro del equipo responde a las siguientes preguntas relacionadas con las Historias de Usuario definidas para el Sprint en el Planning: 1. ¿En qué se avanzó ayer? 2. ¿En qué se trabajará hoy? 3. ¿Hay algún bloqueo o inconveniente? El tiempo asignado para esta reunión es de 15 minutos, y el Scrum Master garantiza que solo el equipo de desarrollo participe en el Daily. El equipo es responsable de liderar esta reunión y el Scrum Master es el responsable.
  + **Planning:** Una ceremonia en la que se define el plan de ejecución o el Sprint Backlog para la siguiente iteración. Esta reunión es liderada por el Product Owner, quien conoce las necesidades y prioridades del cliente. El equipo se reúne y se familiariza con el plan a seguir y los responsables de las Historias de Usuario que se ejecutarán.
  + **Retrospective:** Es el momento de realizar una mejora continua. Se lleva a cabo una evaluación en la que se identifica lo que salió bien y lo que salió mal durante el sprint. En este ritual, se deben responder y registrar las siguientes preguntas: 1. ¿Qué cosas fueron exitosas en términos de rutinas, actitudes o herramientas durante el sprint? 2. ¿Qué cosas salieron mal y perjudicaron al equipo o su rendimiento? 3. ¿Qué podemos implementar para mejorar?
  + **Review:** Una ceremonia en la que se revisa lo que se ha logrado durante el Sprint, se valida el incremento obtenido y se realiza una revisión general de la ejecución de cada Historia de Usuario. En esta reunión, se responde a la pregunta: ¿Qué elementos se completaron y cuáles no durante el Sprint?
  + **Release:** Un lanzamiento del producto, ya sea que esté completo o en una etapa avanzada de desarrollo, siempre asegurando un incremento de valor.
  + **Sprint:** Un objetivo de corta duración con un período de tiempo definido. Es el núcleo de Scrum, son las iteraciones en períodos de tiempo que generan valor para el producto. Normalmente, la duración de un Sprint varía de 1 a 4 semanas

Figura 1. Framework para la ejecución de la metodología Scrum

**Kanban**: La aplicación de Kanban se llevó a cabo considerando sus cuatro principios y seis prácticas. En la arquitectura propuesta, se implementó Kanban con el objetivo de lograr un flujo de trabajo lineal, sin retrasos y con la claridad necesaria para que el equipo pueda comprenderlo adecuadamente.

El método Kanban se formuló como un enfoque para el proceso evolutivo e incremental y el cambio de sistemas en las organizaciones de trabajo. El método se centra en llevar a cabo las tareas pendientes y se pueden dividir los principios más importantes en cuatro principios básicos y seis prácticas.

Los principios fundamentales de la metodología son:

1. **Empezar con lo que se hace actualmente:** Kanban se puede implementar en cualquier tipo de organización sin necesidad de realizar cambios drásticos o configuraciones complejas.
2. **Cambios incrementales y evolutivos:** Kanban se basa en realizar cambios de manera gradual y continua, permitiendo avances cortos pero constantes en el proceso.
3. **Respeto a los procesos y responsabilidades:** Aunque Kanban no rechaza el cambio, respeta los procesos existentes y no los sobrescribe. Se adapta a los flujos de trabajo actuales sin interrumpirlos.
4. **Liderazgo en todos los niveles:** Kanban promueve una cultura de mejora continua en equipo, fomentando el liderazgo y la responsabilidad en todos los niveles de la organización.

Para la implementación de la metodología, se consideraron cinco prácticas comunes durante el proceso de ejecución:

1. **Visualizar el flujo de trabajo:** Es esencial tener un tablero visual con al menos tres columnas (por hacer, en progreso y hecho) que muestre de manera sencilla el progreso de las diversas actividades.
2. **Cero interrupciones:** Se establece un límite para la capacidad máxima de elementos en ejecución, con el fin de evitar desperdicios causados por el multitasking. Esto garantiza un flujo de trabajo continuo.
3. **Gestionar el flujo:** Es importante asegurar un flujo constante de elementos a través del tablero Kanban. Un flujo constante de tarjetas permite un progreso interactivo y una mayor generación de valor.
4. **Fomentar la visibilidad:** Todos los participantes deben tener una comprensión clara del proceso y el progreso. Por lo tanto, es necesario compartir y comunicar los procesos y ajustes relevantes.
5. **Retroalimentación continua:** Se promueven reuniones de retroalimentación, breves y con objetivos claros, al igual que en la filosofía Lean.

**Lean Software Development:** Esta metodología aplicada al desarrollo de software se basa en 7 principios que buscan mejorar los procesos de producción de software. Se convierte en una herramienta enfocada en los equipos de desarrollo, sirviendo como guía para generar valor incremental a los clientes y minimizar desperdicios. Al aplicar esta metodología en la arquitectura, se proponen mejoras en los procesos y mejores prácticas basadas en sus principios:

* 1. **Eliminar desperdicios:** Reducir al mínimo los desperdicios generados en el proceso, como código innecesario, comprensión incorrecta de los requisitos del cliente o documentación no necesaria.
  2. **Amplificar el aprendizaje:** Fomentar el aprendizaje continuo dentro de los equipos de desarrollo, adaptado a la creación de software, mediante la actualización constante en nuevas tecnologías y lenguajes de programación.
  3. **Tomar decisiones lo más tarde posible:** A diferencia del enfoque en cascada, se recomienda estar abiertos a los cambios incluso en etapas avanzadas del proyecto.
  4. **Entregar lo antes posible:** Generar valor de forma constante al cliente mediante entregas priorizadas y en intervalos cortos de tiempo.
  5. **Potenciar el equipo:** Fomentar equipos autoorganizados donde los miembros estén capacitados para la toma de decisiones y se eliminen barreras, valorando su rol dentro del equipo y aprovechando sus conocimientos y experiencia.
  6. **Crear integridad:** Implementar tecnologías como pruebas automatizadas de código en la creación de software.
  7. **Visualizar el conjunto completo:** Buscar la integración del software con otras herramientas para mejorar la experiencia del usuario y la calidad del software en general.

# 2. RESULTADOS Y DISCUSIÓN

* 1. Diagnóstico:

Se realiza un diagnóstico en el cual se analiza las categorías o variables a priori definidas en procesos macro y subprocesos que de estos se desprenden en proceso productivo de la compañía Consigue Ventas. El proceso de construcción de software se define en tres variables a priori, planeación, estimación y desarrollo de los requerimientos. En la actualidad se generan requerimientos de proyectos nuevos o proyectos en curso, los cuales se desarrollan para clientes activos o nuevos clientes. A continuación, se definen las categorías analizadas y

los diferentes subprocesos por los cuales pasan los requerimientos para convertirse en software funcional. En el proceso productivo actual de la compañía se tienen en cuenta diferentes roles, que conforman el equipo de producción de software, este equipo está conformado por diferentes roles con especialidades específicas y con objetivos claros, estos roles se describen a continuación:

* + - Director de desarrollo: Encargado del área de desarrollo, si bien es el director del área también es el encargado de desarrollar, validar, y aprobar los requerimientos del plan de trabajo. Es el miembro con mayor experiencia en el equipo y con conocimientos en diversas áreas del desarrollo.
    - Desarrollador Backend: Parte del equipo de desarrollo, encargado de la implementación del back de las plataformas web (Arquitectura, funcionalidad, etc). Este rol suele ser ajustable a las necesidades y no es excluyente en el desarrollo Frontend.
    - Desarrollador Frontend: Parte del equipo de desarrollo, encargado de la implementación front de las plataformas web (Vista, diseño, botones de acción. etc). Este rol suele ser ajustable a las necesidades y no es excluyente en el desarrollo Backend.
    - Diseñador web: Encargado de definir y diseñar los conceptos web de acuerdo a las necesidades y parámetros dados por el cliente. Miembro encargado del diseño e implementación Web y móviles a nivel visual.
    - Analista de Calidad: Encargado de realizar los procesos de calidad de software y las funcionalidades implementadas, su función adicional de realizar los test de calidad es registrar y actualizar los cambios realizados en código, documentar las novedades y actualizar los manuales del cliente de manera periódica.
    - Coordinador de proyectos: Es la persona encargada de conectar al equipo de trabajo con el cliente. Es quien mantiene informado al equipo de los requerimientos solicitados en los planes de trabajo y al cliente acerca del avance de los mismos. Otra de sus funciones se centran en priorizar el plan de trabajo de acuerdo a la necesidad del cliente. De este rol depende que se entienda bien la necesidad y se entregue al cliente lo solicitado en los tiempos estipulados.

A continuación se describe el proceso productivo de la compañía.

* + - Planeación: Proceso en el cual se realiza el entendimiento del requerimiento presentado de manera escrita por parte del cliente y la planeación de ejecución del requerimiento por parte del equipo de desarrollo. La salida o entregable de este proceso es una Historia de usuario aprobada por el cliente. La variable planeación cuenta con tres subprocesos:
      * Analizar la necesidad del cliente: Se realiza por medio de una reunión virtual o presencial con el cliente el entendimiento de los requerimientos solicitados, en esta reunión se entiende la expectativa y el objetivo del requerimiento.
      * Estudiar el requerimiento: Reunión con el director del equipo de desarrollo en la cual se analiza la viabilidad del requerimiento, se analizan los diferentes riesgos y restricciones en la ejecución y se define un requerimiento ajustado.
      * Aprobar requerimiento: Se realiza un encuentro con los coladores, donde se expone el requerimiento ajustado con las características de implementación definida, la cual él colaborador acepta o rechaza, en caso de no ser de conformidad para los coladores se revisa nuevamente el requerimiento con el director de desarrollo. Si el requerimiento es aprobado por líder se solicita una aceptación escrita con la cual continúa el proceso productivo.
    - Estimación: Proceso en el cual se realiza la estimación de recursos, esfuerzos, horas de desarrollo y fecha de entrega para el requerimiento aprobado por el cliente. Este proceso inicia una vez se obtiene la aprobación del cliente al despliegue propuesto por el equipo de desarrollo. Esta estimación se realiza basado en los conceptos y conocimientos del desarrollador encargado, quien otorga una estimación en horas con la aprobación de la cual es traducida en una fecha de entrega por
      * Asignar recursos: Se Asigna el requerimiento aprobado por el cliente al o a los desarrolladores, diseñadores y administradores de bases de datos, necesarios para la ejecución y el desarrollo del requerimiento.
      * Realizar estimaciones: Proceso en el cual de estimar los esfuerzos en horas por parte de los implicados en el desarrollo del requerimiento para el cumplimiento total o parcial del requerimiento. Una vez se cuenta con una estimación validada por el director de desarrollo, el coordinador del proyecto procede a realizar el cálculo de la fecha estimada de entrega.
      * Autorizar la estimación: Se envía para aprobación del cliente el requerimiento a ejecutar con todos los datos de estimación, este puede aceptar o no el requerimiento.
    - Desarrollo: Proceso en el cual se realiza el desarrollo del requerimiento aprobado por el cliente total o parcialmente, en este proceso se realizan los procesos asociados a la calidad del software para una posterior entrega al cliente. En este proceso se ven vinculados todos los miembros del equipo de desarrollo y es supervisado por el director de desarrollo quien es el encargado de cargar y bajar los cambios aplicados en el código. Este proceso se pone en marcha una vez se obtiene el visto bueno del cliente en la ficha de estimaciones.
      * Desarrollar el requerimiento: Desarrollo e implementación del código necesario para cumplir el requerimiento expreso en la orden de desarrollo en este proceso se incluyen diseños, manejos de base de datos y creación de código (Frontend, Backend).
      * Subir cambios ambiente test: Proceso en el cual se realiza la subida de los cambios implementados para una posterior bajada en ambiente test.
      * Q.A ambiente test: Pruebas de calidad y funcionamiento en la cual se garantiza la funcionalidad en el aplicativo de manera correcta, en caso dado que la implementación no cumpla con las expectativas del cliente o a nivel técnico se devolverá al equipo de desarrollo para su revisión.
      * Bajar cambios: Proceso en el cual se implementan los cambios aprobados por el analista de calidad y el coordinador de proyectos posterior a la validación en el ambiente test para que queden implementados en ambiente producción.

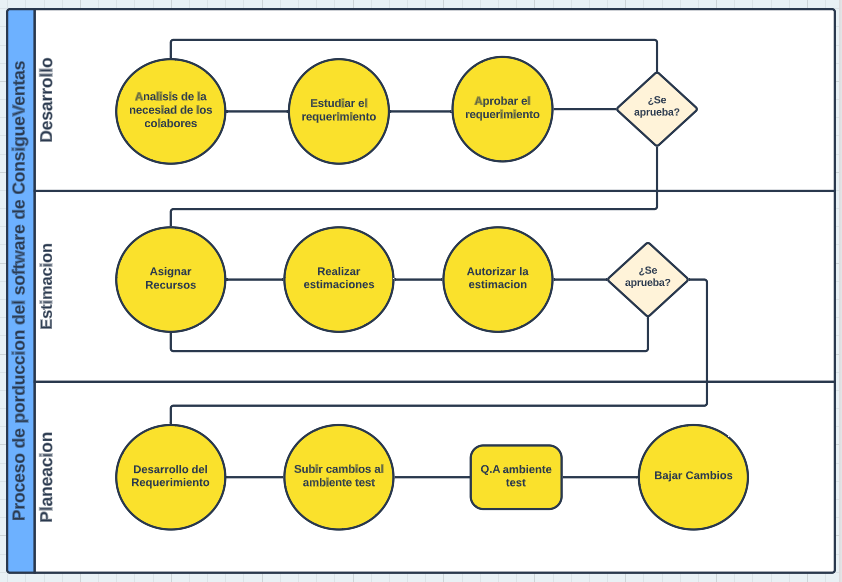


Figura 2. Diagrama general proceso productivo actual de ConsigueVentas Fuente: Elaboración propia

* 1. Desarrollo de la arquitectura:

La presente arquitectura fue basada en los conceptos y principios de las metodologías ágiles, Scrum, Kanban y Lean Software Development, aplicados en nuevas prácticas, ajustes a los procesos, adopción de roles e implementación de estrategias para el aumento de la calidad y el desarrollo de software con la menor cantidad de desperdicio en relación con el valor generado al cliente.

Entendiendo que los requerimientos que ingresan al área de producción de software tienen diferente naturaleza (Tickets, planes de trabajo, ajustes, requerimientos puntuales, mantenimientos. etc) esto exige una línea productiva con la flexibilidad suficiente para que con los recursos limitados con los que usualmente cuentan las Startups y pequeñas empresas, se mantenga un nivel de servicio favorable. Es habitual que empresas de estas características mantengan un enfoque tradicional. Los procesos de producción de software tradicional o tipo cascada son utilizados habitualmente en las empresas de desarrollo de software para la ejecución de los proyectos y el desarrollo de requerimientos a solicitud de los clientes. Sin embargo, “El modelo de cascada está diseñado para proyectos de tamaño reducido y complejidad controlada. No resulta nada recomendable su aplicación cuando existe una gran influencia de la incertidumbre” . Caso contrario al que presentan las pequeñas empresas, en donde se presenta una alta variabilidad e incertidumbre, afectando la capacidad de respuesta de su equipo limitado.

Por lo anterior, la arquitectura propone un sistema productivo donde además de cumplir con los acuerdos de nivel de servicio, se genere valor por medio de las entregas funcionales en cortos periodos de tiempo. Para esto se vinculan las tres metodologías propuestas, utilizando la metodología SCRUM como línea principal en el proceso productivo apoyada en Kanban por medio de su tablero aplicado y sus principios. En busca de una mejora continua de impacto y un proceso transparente, se incluyen los principios Lean aplicados en mejores prácticas a nivel de planificación y ejecución con el menor desperdicio posible.

* + 1. **Aplicación de la metodología SCRUM**: La metodología Scrum brinda un marco de trabajo en el cual se genera una entrega de valor al cliente de manera constante en cortos periodos de tiempo. En el marco de trabajo, se propone la inclusión de Scrum en el proceso productivo, incluyendo conceptos propios de la metodología como sus roles, eventos, artefactos, priorización y medición.

Para la implantación de Scrum no se tienen requisitos de alto nivel que imposibiliten o restrinjan su aplicación. Sin embargo, si se debe contar con una disposición favorable al cambio por parte de todos los miembros que de alguna manera se vinculan con el ciclo productivo, esto incluye al Scrum master, Product Owner, Equipo de trabajo e incluso el mismo cliente del proyecto o de la línea de producción. Es primordial para la aplicación, una disposición previa y favorable del equipo, además de implantar en ellos la mentalidad de autoorganización y descentralizar la toma de decisiones por niveles. Esto debido a que es necesario eliminar las diferentes barreras que podrían desvirtuar la metodología ágil.

Inicialmente, para la aplicación de Scrum es necesario definir la figura de Scrum Master (SM) un rol fundamental en la implementación, debido a que el es encargado de soportar la teoría de Scrum asegurando la aplicación de la metodología, definiendo los parámetros, teniendo en cuenta los recursos disponibles, su capacidad y flexibilidad. Se recomienda que el rol de SM lo ocupe un conocedor de la metodología, tanto a nivel teórico como práctico. Esto debido a la necesidad de claridad en la implementación, anticipándose al impacto que pueda percibir el equipo de trabajo, acostumbrados a otras metodologías o procesos.

Una vez definido el rol del SM es necesario se identifiquen los demás roles, definiendo el número de células o equipos de trabajo que entrarán en función. El número de grupos o células se definen dependiendo el tamaño del equipo y el número de proyectos a los que estos respondan; sin embargo, se debe tener en cuenta que en Startups y pequeñas empresas usualmente se cuenta con un único equipo de trabajo encargado de responder a la totalidad de la demanda a nivel de requerimientos. Esta célula debe estar compuesta por los miembros del área de producción de software, garantizando los recursos necesarios para un desarrollo integral (Diseñadores, desarrolladores, analistas de calidad, coordinador de proyectos, director de desarrollo entre otros.) Una vez identificado nuestro equipo de trabajo, es necesario definir el rol de Product Owner (PO) quien será el encargado de conectar al equipo de trabajo con las necesidades del cliente. Este rol es de vital importancia ya que del entendimiento de las necesidades del cliente depende la generación constante de valor. Usualmente el PO es un rol ocupado por el Gerente de proyecto, coordinador de proyecto o incluso por el mismo director de desarrollo, esto debido a que es necesario contar con el nivel necesario para guiar al equipo de desarrollo al cumplimiento de los objetivos, cumpliendo con las expectativas de los clientes. Por último en la definición de roles es primordial identificar quien es el cliente de cada proyecto en caso de proyectos externos o el cliente interno cuando los requerimientos vienen de un proyecto o necesidad interna.

Una vez se tienen definidos los roles principales para la implementación del marco Scrum, se debe definir junto con el SM cómo se ejecutarán los eventos o ritos siempre teniendo en cuenta las rutinas y obligaciones habituales. Inicialmente se recomienda definir el timebox (Tiempo definido para el evento) de los Sprints, este se sugiere no ser inferior a una semana ni superior a cuatro, esto basado en las necesidades de los clientes y el alcance de los proyectos. Una vez definido, es necesario planear la ejecución de los demás eventos que se desarrollan dentro de cada uno de los Sprints. Para la planeación de los diferentes eventos que dan lugar a Scrum, es necesario definir su propio timebox, estas definiciones se realizan con base a la capacidad del equipo, de manera consensuada, teniendo en cuenta que para cada uno de los ritos es necesario contar con la disposición del equipo y los demás participantes. A continuación se expone el flujo en el que se debería desarrollar cada uno de los diferentes eventos.

Sprint Planning (Planeación del Sprint): Se sugiere que su timebox se defina de acuerdo a la duración del sprint en relación de dos horas de planeación por cada semana del sprint. Es importante que el PO llegue a este evento con una propuesta de Sprint Backlog (Lista de requerimientos que se tendrán en cuenta en el Sprint ) con base a una reunión previa con el

cliente. En la figura 3 se puede evidenciar los componentes y actores del Planning de acuerdo a las responsabilidades de cada rol.



Figura 3. Diagrama ejecución Sprint Planning Fuente: Elaboración propia

Sprint Review (Revisión del Sprint): Rito en el cual se da visibilidad a lo ejecutado durante el sprint, su impacto o incremento que será presentado al cliente. En la figura 4 se puede identificar los puntos importantes del Sprint Review y los responsables en su ejecución.



Figura 4. Diagrama ejecución Sprint Review

Fuente: Elaboración propia

Sprint Retrospective (Retrospectiva del Sprint): Evento en el cual el equipo realiza un análisis en busca de mejoras en la ejecución de los sprints. Se analizan los puntos fuertes y débiles en la ejecución. En el siguiente esquema (figura 5) se evidencian los componentes y actores del evento Retrospective de acuerdo a las responsabilidades de cada rol.

Figura 5. Diagrama ejecución Sprint Retrospective Fuente: Autor

Daily Meeting (Reunión diaria): Reunión de control diaria en la cual se analiza el avance conseguido por el equipo, el plan de trabajo para el presente día y los inconvenientes que se presentaron en la ejecución. En la figura 6 se puede identificar los puntos importantes del Daily y sus lineamientos para la ejecución.

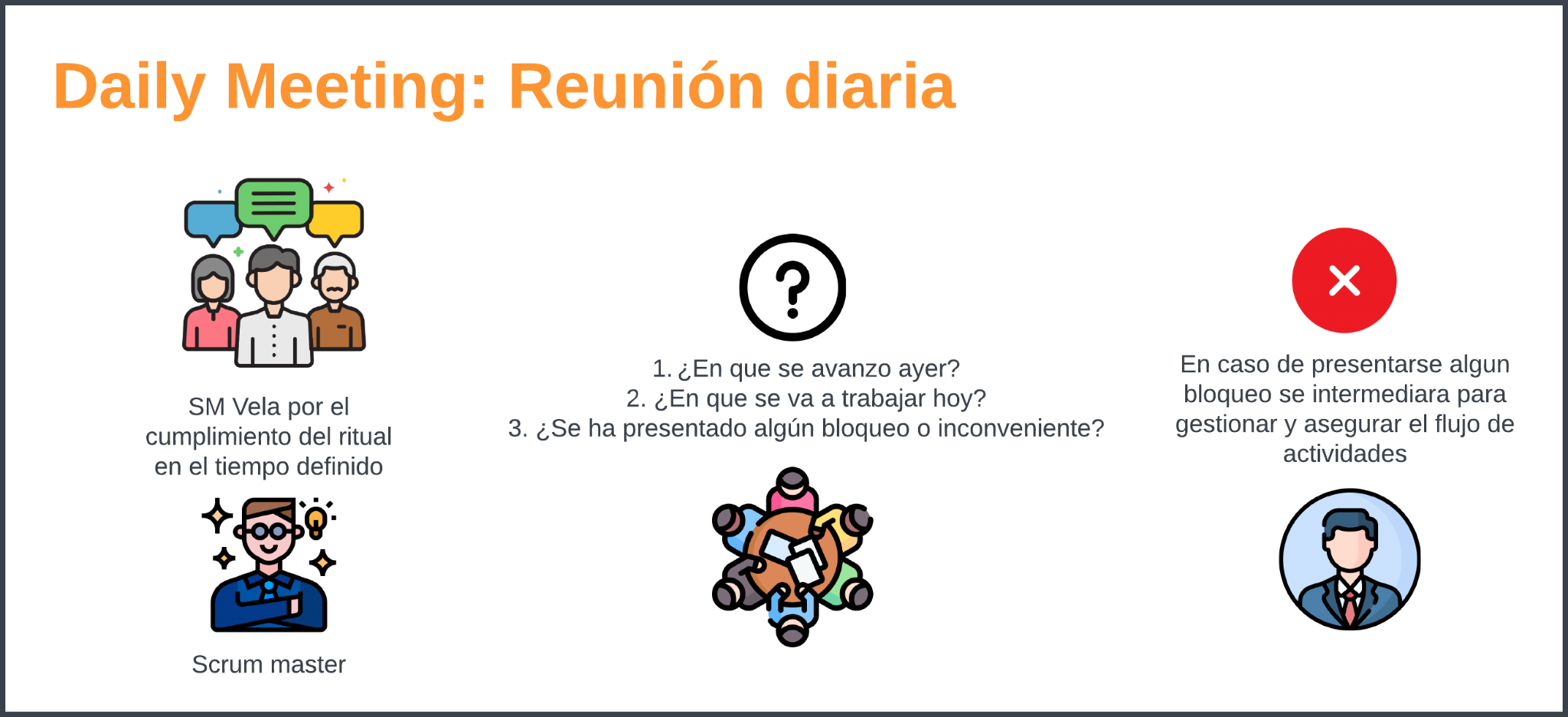


Figura 6. Diagrama ejecución Daily Meeting Fuente: Autor

Los eventos dentro de la metodología Scrum se encuentran definidos en su forma; sin embargo, la aplicación dentro de cada proceso productivo es flexible en su implementación dependiendo la capacidad de la empresa. El SM es quien debe garantizar la ejecución de cada uno de los eventos, garantizando que se cumpla el objetivo en el timebox definido.

En la aplicación de la metodología es fundamental que todos los elementos cuenten con la transparencia necesaria para el entendimiento de los requerimientos o artefactos desarrollados en el sprint por parte de cada miembro que interactúa en el proceso productivo. Para el flujo de trabajo en una empresa con las características definidas es necesario contar con un Product backlog conformado con los requerimientos priorizados por el PO de acuerdo a las necesidades de los diferentes clientes y proyectos que se estén ejecutando, para esto, es indispensable una continua comunicación con el cliente realizada por el PO. Para la conformación del Sprint backlog es necesario contar con la claridad suficiente de las necesidades priorizadas, traducidas en Épicas, Historia de usuario (HU) o tareas, dependiendo la complejidad y magnitud de los requerimientos o planes de trabajo. A continuación se describen los diferentes artefactos en los cuales se traducen los requerimientos inmediatos

Epica: Objetivo global que se quiere alcanzar en un periodo de tiempo definido, el cual genera las HU.

Hu: Historias que aportan valor de forma incremental que contienen algunos criterios de estructura y contenido.

Tareas: Es una acción clave que permite el cumplimiento de la HU. Debe contar con un responsable. y puede convertirse en una nueva HU

Estos artefactos deben contener la información suficiente para el entendimiento de la necesidad o el requerimiento a desarrollar, además de esto, los artefactos deben contar con una estructura estándar de conocimiento abierto al equipo. Estos artefactos cuentan con diferentes componentes que permiten su caracterización y facilitan el entendimiento, al igual que permiten el seguimiento a nivel de cumpliento.

En la figura 7 se describen cada uno de los elementos que identifican y componen los artefactos. Se deben definir de acuerdo a la necesidad de alcance.



Figura 7. Estructura de los artefactos

Una vez definidos los roles, eventos, y los componentes de los diferentes artefactos, se debe definir cómo se controlará el flujo de trabajo y el nivel de avance del equipo, Para esto se propone la inclusión de la metodología Kanban dentro del proceso.

* + 1. Aplicación de la metodología Kanban:

En un proceso productivo es indispensable contar con un flujo de información transparente y continuamente actualizado, en el cual se tengan claras las especificaciones de los requerimientos y sus respectivos responsables. Aquí es donde cobra importancia la metodología Kanban, la cual por medio de su tablero y la aplicación de sus principios permite un flujo regulado de trabajo.

El tablero Kanban es un artefacto de implementación sencilla, en la actualidad el mercado web ofrece variadas opciones de implementación digital en la nube; sin embargo, este se puede implementar en un tablero o una pared. El concepto a aplicar es su sistema de columnas, encargado de dar claridad al flujo de trabajo y permite una identificación rápida del trabajo en curso. Como se sabe el tablero Kanban propone manejar el flujo de trabajo en 3 columnas básicas, pendiente, en ejecución y terminada. Sin embargo, para la presente arquitectura se propone implantar tantas columnas como el equipo, PO y SM consideren necesarias en el proceso. Dentro de estas columnas adicionales se recomienda manejar una para el Product Backlog, en la cual se listan todos los artefactos sin proceso de priorización. Otra columna en la cual se maneja el Sprint backlog con los artefactos priorizados y columnas para procesos de validación. Estas columnas contendrán los artefactos definidos y aprobados y permitirán el flujo de las tarjetas de manera práctica.

Una vez implementadas las columnas pertinentes, el tablero Kanban se convertirá en la hoja de ruta del equipo y permitirá a todos los interesados contar con un conocimiento actualizado acerca del plan de trabajo. Dentro del tablero Kanban se deben realizar algunas definiciones a nivel de procesos multitarea (Multitask). Se debe definir una capacidad máxima para la ejecución de artefactos, de acuerdo a los parámetros dados por Kanban cada miembro como máximo debe tener 3 tarjetas en la columna de ejecución y debe mantener constantemente actualizada su actividad.

Una de las herramientas propuestas para la implementación del tablero Kanban es la plataforma Trello, quien ofrece un entorno útil y una versión gratuita; sin embargo, si se busca una mayor integración entre las dos metodologías, se recomienda la herramienta JIRA, la cual cuenta con opciones propias para la ejecución de Scrum por medio de Kanban.

Una vez implementado el sistema de trabajo y la herramienta por la cual se manejara su flujo, se deben implantar lineamientos, los cuales garanticen que el sistema productivo sea eficiente para esto y con base a los pilares de la metodología Lean Software Development.

* + 1. Aplicación Lean Software Development:

La aplicación de esta metodología dentro de la arquitectura propuesta se fundamenta en sus principios y la aplicación de estos, por tal motivo de acuerdo a sus pilares se propone:

* + - 1. Eliminar desperdicios: Se propone centrarse en los desperdicios generados en el código, agregando procesos de validación del mismo, identificando y depurando código innecesario. Por otra parte, con base a mediciones de indicadores se proponen ajustar los procesos que generan desperdicio, identificando cuellos de botella en cada interacción y realizando seguimientos de mejora continua.
      2. Amplificar el aprendizaje: Dentro de los equipos de desarrollo, implementar la cultura del aprendizaje continuo, brindando por parte de la compañía herramientas de actualización para su equipo productivo, lo que permitirá mantenerse a la vanguardia de las nuevas tecnologías.
      3. Tomar decisiones lo más tarde posible: En compañías con una capacidad limitada es fundamental estar abierto a los cambios aún en una fase avanzada del sprint o su planeación, es necesario tomar decisiones acerca de la ejecución contando con un entorno estable y con toda la información posible.
      4. Entregar lo antes posible: Es de vital importancia generar valor constante al cliente por medio de entregas continuas, por lo tanto, se propone mantener un sistema flexible de implementación de cambios (Subidas y bajadas del código madre). Priorizando los elementos completos y validados en la implementación en ambiente de producción.
      5. Potenciar el equipo: Promulgar los equipos auto organizados es primordial, en la actualidad una de las mayores fuentes de cuellos de botella son los procesos

centralizados dependiendo el rol dentro del equipo. Se propone promulgar la autoorganización de tal manera que las decisiones de nivel medio y bajo puedan ser tomadas por los miembros del equipo, esto reducirá los desperdicios de tiempo generados por tiempos en toma de decisiones.

* + - 1. Crear la integridad: Se recomienda de ser posible, teniendo la capacidad por parte de la empresa, invertir en sistemas automatizados de validación de código. En caso de no ser posible, se recomienda incluir un proceso de validación en el cual intervengan los desarrolladores y el director de desarrollo.
      2. Visualizar todo el conjunto: En el desarrollo de software es necesario no solo implementar los artefactos para el cumplimiento de la necesidad puntual del cliente, se debe también tener en cuenta que tanto el aplicativo se integra con los demás sistemas, esto nos lleva a siempre pensar en que los aplicativos se ajusten a los sistemas actuales.
  1. Aplicación en la empresa Consigue Ventas:

La compañía en la actualidad cuenta con un proceso definido, el cual contiene principios ágiles y tradicionales, para la aplicación de la arquitectura se aconseja en cabeza del coordinador de proyecto y con aprobación del gerente de la compañía y del director de desarrollo definir los parámetros propuestos para la aplicación de la metodología Scrum (Roles, Eventos, Artefactos), una vez definidos y aprobados estos parámetros, se debe proceder con la capacitación pertinente y adecuación del equipo. La segunda fase consiste en definir los TImebox de cada uno de los eventos y su planeación de ejecución, para esto es necesario contar con el concepto y experiencia del director de desarrollo, pues es quien conoce a fondo las limitaciones del equipo.

De acuerdo a los ajustes aprobados, es necesario reestructurar el proceso actual, incluyendo la implementación de Scrum, el manejo de del tablero Kanban y los procesos de mejora definidos. El proceso se recomienda sea diagramado y puesto a conocimiento de todos los interesados.

Por último, es necesario para dar inicio al proceso, realizar por parte del Coordinador de proyectos o en este caso Product Owner un barrido de los requerimientos actuales, clasificándolos por cliente y prioridad. Una vez definidos, realizar una reunión con los diferentes clientes para así generar el primer Product Backlog a ejecución del equipo productivo de la compañía dados por épicas, historias de usuario o tareas de acuerdo a la necesidad.

En la actualidad la compañía Consigue Ventas se encuentra en un proceso de crecimiento tanto en su tamaño como en sus proyectos, lo cual implica procesos cada vez más ajustados y ágiles. Sin embargo, la compañía aún no tiene una definición clara en sus procesos, lo que genera diferentes puntos de mejora. Se sugiere un manejo ágil en sus procesos por medio de las metodologías que se proponen en esta arquitectura.

# 3. CONCLUSIONES

* + - Como se evidenció, es posible aplicar un marco de trabajo general en un proceso estándar de producción de software, teniendo en cuenta los recursos con los que cuente la compañía. La aplicación del marco es posible si se cuenta con un equipo flexible y ajustado a las necesidades del negocio.
    - Se concluye que es posible generar un marco de trabajo basado en el concepto ágil y la filosofía Lean por medio de la aplicación de una o más metodologías que hacen parte de este pensamiento (Scrum, Kanban y Lean Software Development). Con una aplicación ajustada y conjunta de estas metodologías se puede desarrollar un proceso productivo que genere valor constante al cliente.
    - Se recomienda para la aplicación del marco propuesto y para la implementación de cualquier marco ágil, que este se realice de manera ajustada a cada empresa y su necesidad, teniendo en cuenta diferentes variables como lo son, su equipo, recursos, clientes, capacidad, etc. Aplicando los principios de las metodologías de manera total, parcial o ajustada de acuerdo a la estructura de la compañía y sus procesos ya constituidos.
    - Se recomienda la aplicación del presente marco en empresas con un equipo de trabajo limitado o reducido, que respondan a proyectos de diferentes naturalezas pero con una demanda constante a nivel de solicitudes y requerimiento proveniente de los clientes. Para su aplicación se recomienda un conocimiento previo de las metodologías presentadas.
    - Con la aplicación de la arquitectura propuesta es necesario visualizar una descentralización de los equipos productivos, impulsando equipos autoorganizados capaces de solucionar, de forma autónoma, los retos que se presentan en el día a día.
    - La aplicación de la arquitectura no es restringida a la cadena productiva del desarrollo de software, esta también es aplicable en la ejecución de proyectos, implementación de nuevas metodologías de trabajo, Creación de nuevos productos y servicios, además de actividades que vincule la consecución de objetivos de manera ágil y con resultados generadores de valor.