

**ARQUITECTURA PARA EL DESARROLLO ÁGIL EN LA PRODUCCIÓN DE SOFTWARE
BASADO EN LA EMPRESA TUPROYECTO S.A.S. PROYECTADO A STARTUPS Y
PEQUEÑAS EMPRESAS**

**AUTOR
HERNAN DARIO GOMEZ CUERVO**

INGENIERO INDUSTRIAL

u9501000@unimilitar.edu.co

Artículo Trabajo Final del programa de Especialización en Gerencia Logística Integral



La U
acreditada
para todos

**ESPECIALIZACIÓN EN GERENCIA LOGÍSTICA INTEGRAL
UNIVERSIDAD MILITAR NUEVA GRANADA
FACULTAD DE INGENIERÍA
JUNIO, 2020**

**ARQUITECTURA PARA EL DESARROLLO
ÁGIL EN LA PRODUCCIÓN DE SOFTWARE
PARA LA EMPRESA TUPROYECTO S.A.S.
PROYECTADO A STARTUPS Y PEQUEÑAS
EMPRESAS.**

**ARCHITECTURE FOR AGILE DEVELOPMENT IN THE
PRODUCTION OF SOFTWARE FOR THE COMPANY
TUPROYECTO S.A.S. PROJECTED TO STARTUPS AND
SMALL BUSINESSES**

Hernán Darío Gómez Cuervo
Ingeniero Industrial
u9501000@unimilitar.edu.co

RESUMEN

El presente artículo presenta una arquitectura marco aplicable a los procesos productivos del desarrollo de software, basado en metodologías ágiles analizando el proceso productivo de la compañía TuProyecto S.A.S. El documento presenta, un diagnóstico actual de las variables a priori y los subprocesos productivos de la compañía, así mismo se desarrolla la arquitectura con los métodos y pasos sugeridos tomando como base las metodologías ágiles SCRUM, Kanban y Lean Software Development. Por último se realiza un paso a paso del proceso de implementación del marco sugerido a la compañía. A través de la ejecución del documento presentado, fue posible generar un arquitectura marco aplicable a Startups y pequeñas empresas del sector, para una implementación ligera que constituyan buenas prácticas para la generación de valor constante de cara a los clientes.

Palabras Clave: Metodologías Ágiles, Scrum, Kanban, Lean Software Development, Producción de software.

ABSTRACT

This article presents framework architecture applicable to the productive processes of software development based on agile methodologies based on the productive process of the company TuProyecto S.A.S. The document presents a current diagnosis of the a priori variables and the productive threads of the company, and the architecture is developed with the suggested methods and steps based on the agile methodologies SCRUM, Kanban and Lean Software Development, and finally it is carried out a step-by-step process of implementing the suggested framework for the company. Through the execution of the presented document, it was possible to generate framework architecture applicable to Startups and small companies in the sector, for a light implementation that constitutes good practices for the generation of constant value for customers.

Keywords: Agile Methodologies, Scrum, Kanban, Lean Software Development, Software Production.

INTRODUCCIÓN

En sectores no convencionales como lo es el del desarrollo de software, cada vez son más las Startups y pequeñas empresas enfocadas en brindar diferentes soluciones a nivel de desarrollo de aplicativos y programas funcionales, enfocados en automatizar, optimizar y dar mayor transparencia a diferentes procesos que se realizan en empresas públicas y privadas de variados sectores productivos. En el mercado, cada vez la oferta y demanda de servicios relacionados al desarrollo de software es mayor y variada. Sin embargo, la calidad de estos aplicativos, páginas y sistemas, a menudo no cumplen la necesidad real del cliente, bien sea por un entendimiento básico del problema por parte del equipo, poca claridad por parte del cliente o procesos deficientes de producción del software o calidad del mismo.

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 Tuproyecto: ¿Es posible implementar una arquitectura (Framework) utilizando metodologías ágiles y principios lean, en marco a la producción de software en la empresa TuProyecto, aplicable en pequeñas empresas y startups para el aumento de la calidad y nivel de servicio ofrecido a sus clientes ?.

El objetivo del presente artículo 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 startups y pequeñas empresas. 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 en startups o pequeñas empresas. Como bien lo explica el autor Pino, en las últimas dos décadas, la comunidad de Ingeniería de Software ha expresado especial interés en la mejora del proceso de software (SPI) en un esfuerzo por aumentar la calidad del producto de software, así como la productividad del desarrollo de software[1].

De este modo es como se pretende construir el presente artículo basado en la operación de la empresa TuProyecto S.A.S, 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 del cliente.

En la actualidad, y tal como lo sustenta Fayad [2] la mayoría de las empresas de desarrollo de software son pequeñas (Menos de 50 personas). Estas constantemente se enfrentan a diferentes retos de alto nivel que exigen prácticas eficientes en el desarrollo y la ingeniería de software, siempre siendo flexibles a su tamaño y capacidad técnica. A partir de la década de los 90 compañías del sector y diferentes investigadores, determinan la necesidad y el interés en mejorar las prácticas y procesos de la producción de software, creando el término internacional SPI (Software Process Improvement). Sin embargo, dentro de la industria y la academia, se ha tenido la percepción que estos sistemas de mejora sólo son aplicables en empresas de gran tamaño debido a la estructura organizacional de las PYMES, pequeñas empresas y Startups, puesto que estas compañías no tendrían la capacidad de implementar programas de mejora a nivel de los entes reguladores como el Software Engineering Institute (SEI) y el International Organization for Standardization (ISO). Por tal motivo, se sostiene que estos parámetros no han sido redactados para ser aplicados en empresas pequeñas y tipos Startups.[3][4]

Existen ciertas metodologías que implementadas total o parcialmente permiten generar un marco aplicable en pequeñas empresas y Startups, 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 los clientes. 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 artículo 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 Schwaber, 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. [5]

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. [6]

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. [7].

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 artículo se desarrolla basados en el análisis del proceso productivo de la empresa TuProyecto S.A.S, una Startup con 6 años en el mercado dedicada a ofrecer soluciones y servicios tecnológicos con base en el desarrollo de software, con experiencia en la ejecución de proyectos con entidades del sector público y privado. En concordancia con las herramientas, datos y acceso a información necesaria para el desarrollo del artículo, las cuales serán suministradas por la empresa TuProyecto SAS y por la documentación oficial acerca de los temas, metodologías y prácticas que se analizaran en el desarrollo del documento, se determina la viabilidad del artículo en un tiempo establecido de 2 meses en el cual se espera el documento con la arquitectura definida aprobada por la compañía TuProyecto S.A.S.

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 TuProyecto S.A.S. El artículo finaliza con las conclusiones

1. MATERIALES Y MÉTODOS

Para el desarrollo del presente artículo fue necesario utilizar recursos y métodos que permitieran diseñar la arquitectura propuesta, esto con base a los conceptos de cada una de las herramientas planteadas anteriormente. De manera complementaria, fue necesario contar con información de primera mano que permitiera conocer el estado real y actual de diferentes factores dentro de la empresa TuProyecto S.A.S, para asimismo lograr construir un diagnóstico acertado y conocer en detalle los elementos a mejorar. Para la construcción de este nuevo modelo, se trabajaron herramientas y metodologías, como:

Metodología Scrum: Dentro de la arquitectura se incluye el marco de trabajo o framework SCRUM para la ejecución de los requerimientos o desarrollo de productos derivados de los diferentes proyectos en ejecución. Se aplicó el marco de esta metodología ágil para coordinar el flujo de trabajo, que si bien no es constante, si puede generarse de diferentes fuentes.

Para esto, Scrum permite generar mayor valor a los clientes por medio de sus diferentes fases y roles. Su objetivo principal es adaptar los principios y buenas prácticas que permita al equipo de trabajo una generación de software a gran velocidad, con una capacidad de responder a los cambios que se puedan presentar incluso en fases tardías de los proyectos [8].

Scrum se basa en una generación constante e incremental de valor al cliente, para esto, la metodología propone con base equipos auto organizados y adaptativos, crear una cultura de agilidad por medio de sus artefactos, roles y eventos, buscando acelerar el proceso productivo por medio de Sprints que aseguran una generación de valor constante por medio de entregas iterativas.

- **Roles:** Actores de la metodología, cuentan con un rol y una responsabilidad definida, encargados de generar valor en cada sprint o iteración.
 - **Scrum Master:** Es el ingeniero de Scrum, dicta los parámetros y asegura el cumplimiento a cabalidad de la metodología. El SM hace parte del día a día de Scrum y es quien debe garantizar el avance.
 - **Product Owner:** Es el arquitecto encargado de traducir las necesidades de los clientes. Es el responsable de maximizar el valor del producto. En ocasiones toma decisiones a nombre del cliente.
 - **Development team:** Son los encargados de ejecutar los requerimientos o artefactos. Son quienes directamente dan valor incremental.
 -
- **Ritos:** Son las ceremonias o ritos que se deben ejecutar en la metodología. A medida que se van ejecutando los rituales el equipo los interioriza y se vuelven expertos.
 - **Daily:** Ritual diario de todo el equipo, en el cual se realiza una revisión de avance. Cada miembro del equipo deberá contestar las siguientes preguntas de cara las HU definidas para el Sprint en el planning. 1. ¿En que se avanzó el día de ayer? 2. ¿En qué se va a trabajar hoy? 3. ¿Se ha presentado algún bloqueo o inconveniente?. El tiempo de ejecución es de 15 min y el scrum master asegura la regla que solo el equipo de desarrollo participa en el Daily. el equipo es quien debe estar a cargo de esta reunión y el responsable.
 - **Planning:** Ceremonia en la cual se define el plan de ejecución o Sprint Backlog para la siguiente iteración. Este ritual va dirigido por el Product Owner quién es el que conoce la necesidad y la prioridad del cliente. El equipo debe estar

reunido y conocer el plan a seguir y los responsables de las historias de usuario que se ejecutará.

- **Retrospective:** Es el momento donde se presenta la oportunidad de realizar una mejora continua. Se realiza una evaluación en la cual se identifica que salió bien y que salió mal durante el sprint. En este ritual se debe contestar y registrar las siguientes preguntas. 1. ¿Cosas que han ido bien: rutinas, actitudes o herramientas positivas en el sprint? 2. ¿Cosas que han ido mal: aquellas que han perjudicado al equipo o su rendimiento? 3. ¿Qué implementamos para mejorar?
- **Review:** Ritual en el cual se realiza una revisión de lo ejecutado en el Sprint, se valida el incremento que se obtuvo, se realiza una revisión general de la ejecución de cada historia de usuario. En este ritual se debe contestar la siguiente pregunta: ¿Qué elementos se terminaron y cuáles no para el Sprint?
- **Release:** El release es un lanzamiento del producto, bien sea terminado o en un gran avance de desarrollo siempre garantizando el incremento de valor.
- **Sprint:** Objetivo de corta duración con un timebox definido. Es el corazón de Scrum, son las iteraciones en periodos de tiempo que generan valor al producto, usualmente el tiempo del sprint de 1 a 4 semanas

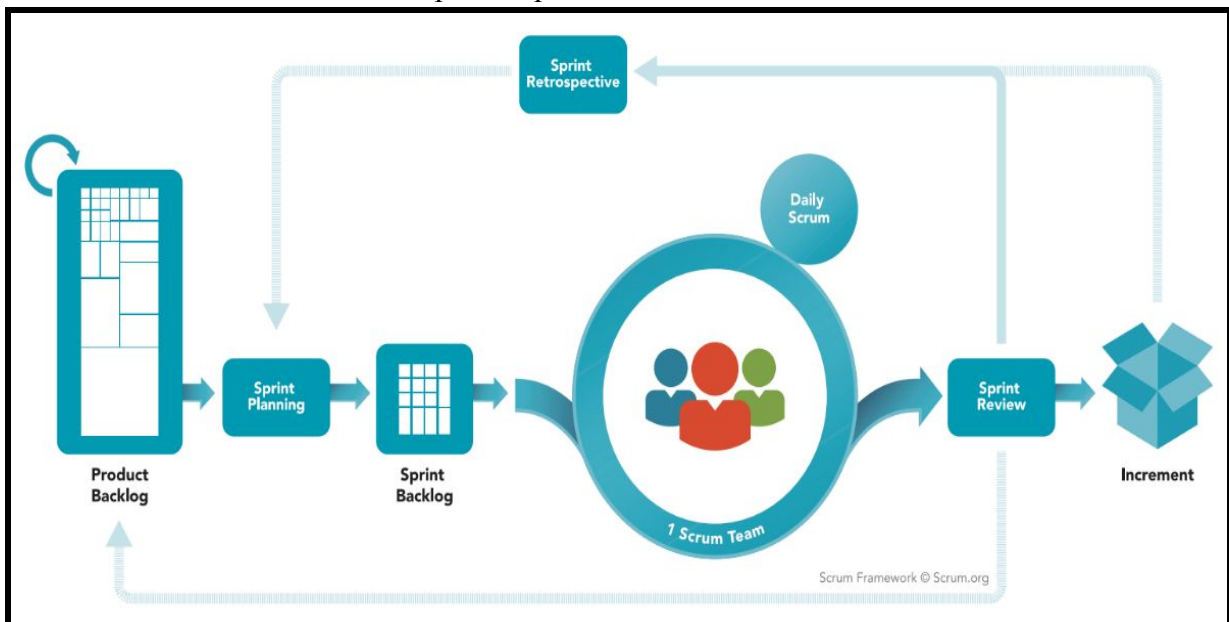


Figura 1. Framework para la ejecución de la metodología Scrum

Fuente: Curso: Scrum Master Certificación Mundial. Plataforma: Udemy. Video: Clase 4. El manifiesto Agile). 2020 [5]

Kanban: Su aplicación se realizó teniendo en cuenta sus cuatro principios y sus seis prácticas. Dentro de la arquitectura propuesta se implementó para lograr un trabajo lineal sin retrasos y con la claridad necesaria para el buen entendimiento del equipo.

“David J. Anderson (reconocido como el líder de pensamiento de la adopción del Lean/Kanban para el trabajo de conocimiento) formuló el método Kanban como una aproximación al proceso evolutivo e incremental y al cambio de sistemas para las organizaciones de trabajo. El método está enfocado en llevar a cabo las tareas pendientes y los principios más importantes pueden ser divididos en cuatro principios básicos y seis prácticas.” [6]

Los principios básicos de la metodología son:

1. Empezar con lo que hace ahora: Es implementable en cualquier tipo de organización no requiere configuración, ni cambios drásticos.
2. Cambios incrementales y evolutivos: Kanban es una práctica con una resistencia mínima en su implementación, busca por medio de su sistema generar cambios incrementales, avances cortos pero continuos.
3. Respetar procesos y responsabilidades: Este método no es repulsivo al cambio, sin embargo no sobrescribe los procesos que están en curso.
4. Liderazgo en todos los niveles: kanban fomenta una cultura de mejora continua en equipo, un hábito de cumplimiento y coordinación de todos los participantes.

Para la ejecución de la metodología se tuvieron en cuenta cinco prácticas habituales durante los procesos de ejecución:

1. Visualizar el flujo de trabajo: Es necesario contar con un tablero visual con mínimo tres columnas (por hacer, en progreso y hecho) en las cuales se visualice de una manera simple el progreso de las diferentes actividades.
2. Cero interrupciones: Se debe tener un límite a la hora de establecer la capacidad máxima de elementos en ejecución, esto para evitar desperdicios a causa del multitask. De esta manera se asegura un flujo continuo de trabajo.
3. Gestionar el flujo: Se debe asegurar un flujo continuo de elementos a través del tablero Kanban. El constante flujo de tarjetas asegura un progreso interactivo y una mayor generación de valor.
4. Fomentar la visibilidad: Es necesario que todos los participantes cuenten con la claridad suficiente del proceso y el progreso, por lo tanto es necesario se divulguen los procesos y ajustes del mismo.
5. Retroalimentación continua: Se promueve al igual que en toda la filosofía lean las reuniones de retroalimentación, cortas y con objetivos claros.

Lean software development [9]: Esta metodología aplicada al desarrollo de software, se fundamenta en 7 principios aplicables, buscando una mejora en los procesos de producción de software. Convirtiéndose así en una herramienta enfocada en equipos de desarrollo, para ser utilizado como guía al momento de generar valor a los clientes de manera incremental y con la

menor cantidad de desperdicios posibles. Para la aplicación de la metodología en la arquitectura, se propusieron mejoras en procesos y mejores prácticas basadas en sus principios:

1. Eliminar desperdicios: Reducir al mínimo los desperdicios generados por el proceso, el código, el entendimiento de los requerimientos de los clientes la documentación no necesaria entre otros.
2. Amplificar el aprendizaje: Dentro de los equipos de desarrollo es necesario el aprendizaje continuo, adaptado a la creación de software, la continúa actualización en nuevas tecnologías y nuevos lenguajes de programación.
3. Tomar decisiones lo más tarde posible: A diferencia del método cascada en la ejecución lean se recomienda estar abiertos a cambios sin importar que el proyecto esté en una etapa avanzada.
4. Entregar lo antes posible: Generar valor constante al cliente por medio de entregas priorizadas y en intervalos cortos de tiempo.
5. Potenciar el equipo: Promulgar los equipos auto organizados con el objetivo que los miembros del equipo estén capacitados para la toma de decisiones, y la eliminación de barreras, otorgando importancia a su rol dentro del equipo y aprovechando sus conocimientos y experiencia.
6. Crear la integridad: implementando tecnología como pruebas automatizadas de código en la creación de software.
7. Visualizar todo el conjunto: Buscar una integración del software con las demás herramientas que se tengan en búsqueda de una mejora integran en el software y en la experiencia del usuario.

El desarrollo del artículo se fundamenta en 3 fases. En la primera fase se realizó un diagnóstico al proceso productivo actual de la compañía TuProyecto S.A.S, realizando un análisis de las diferentes variables a priori y los subprocesos que las compone. En la segunda etapa se describe el marco de la arquitectura propuesta explicando los aportes de cada metodología y la estructura de la misma. para finalizar se describirán los pasos para una futura implementación en la compañía.

2. RESULTADOS Y DISCUSIÓN

2.1 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 TuProyecto S.A.S. En 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 realizado 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 el cliente, donde se expone el requerimiento ajustado con las características de implementación definida, la cual el cliente acepta o rechaza, en caso de no ser de conformidad para el cliente se revisa nuevamente el requerimiento con el director de desarrollo. Si el requerimiento es aprobado por el cliente 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 del 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 se 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 enviar 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 realiza 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.

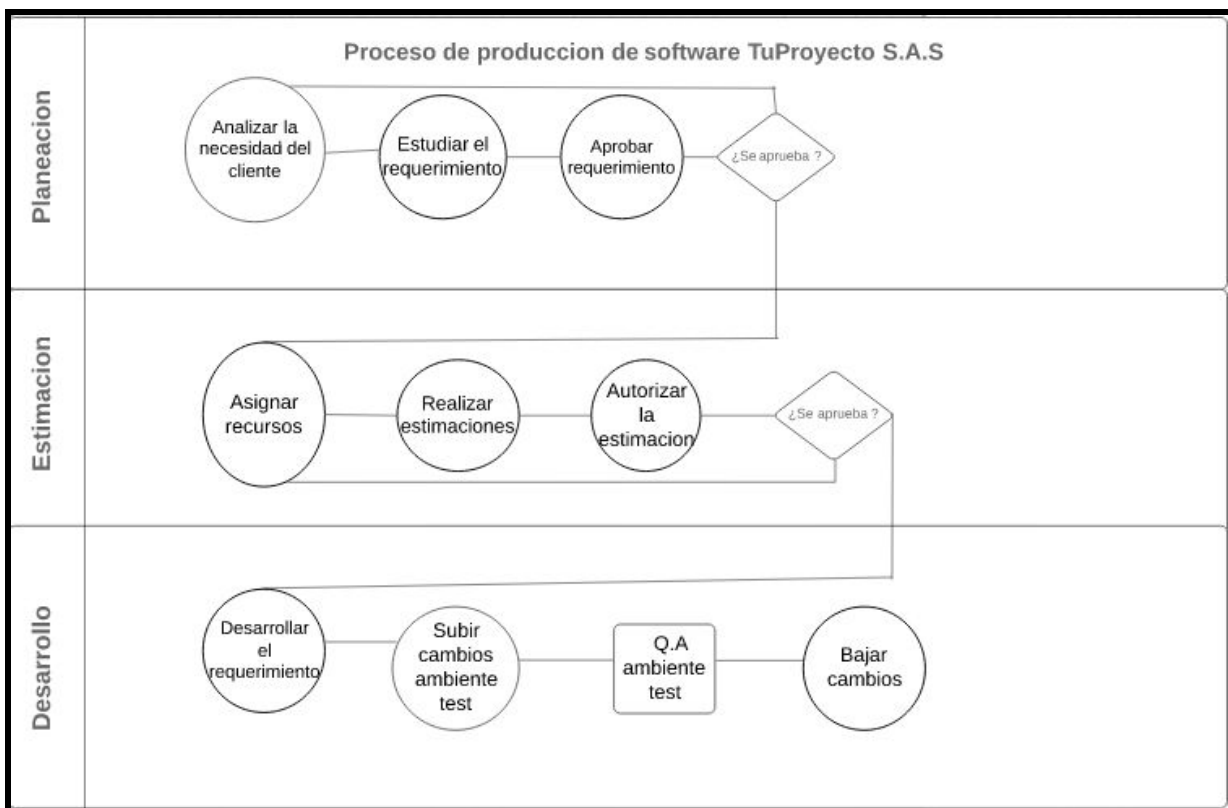


Figura 2. Diagrama general proceso productivo actual de TuProyecto S.A.S
Fuente: Elaboración propia

2.2 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 como lo menciona la OBS Business School “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” [10]. 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.

2.2.1 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 evidencia los componentes y actores del evento Retrospective de acuerdo a las responsabilidades de cada rol.

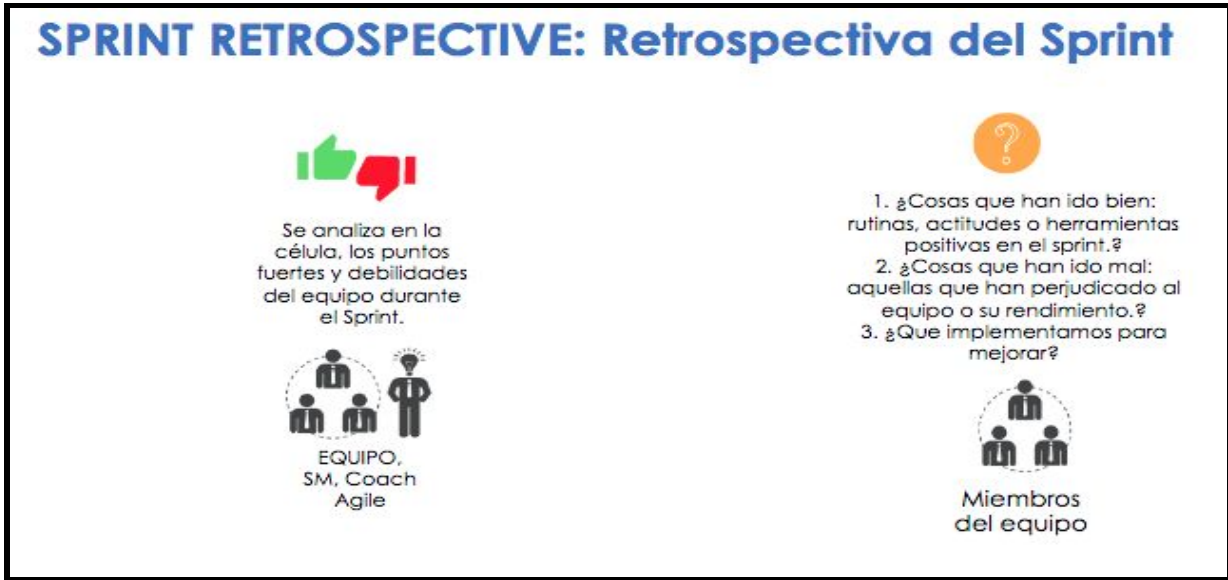


Figura 5. Diagrama ejecución Sprint Retrospective

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.

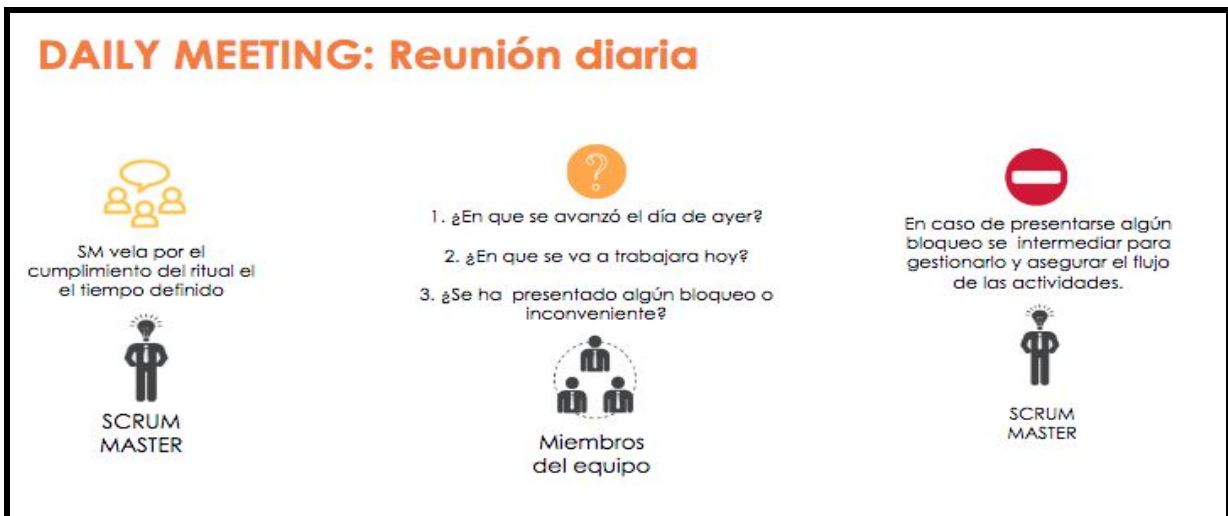


Figura 6. Diagrama ejecución Daily Meeting

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 a 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 de una Start up 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 cumplimiento.

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.

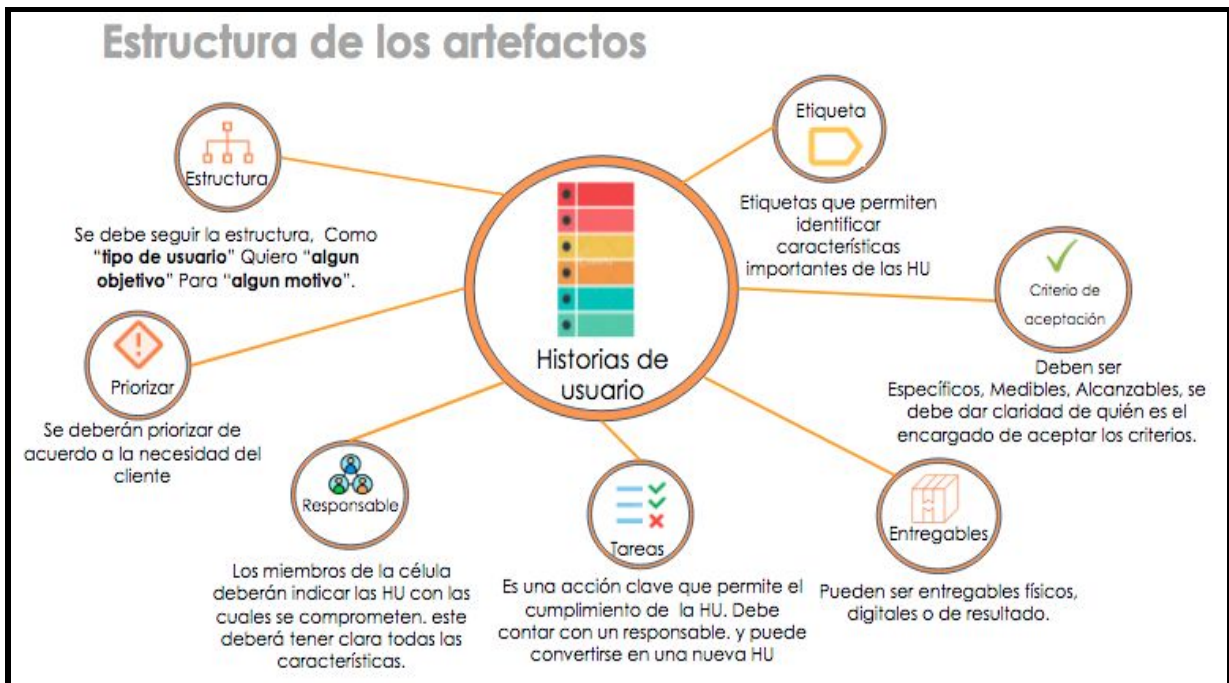


Figura 7. Elementos de un artefacto

Fuente: Autor

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.

2.2.2 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 maneje 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 maneja 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.

2.2.3 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 aun en una fase avanzada de los 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.

6. 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.
7. 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.

2.3 Aplicación en la empresa TuProyecto S.A.S:

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 TuProyecto S.A.S 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 evidencio, es posible aplicar un marco de trabajo general en un proceso estándar de producción de software, teniendo en cuenta los recursos con los que cuenta 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 naturaleza 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.

REFERENCIAS

- [1] F. J. Pino, F. García, M. Piattini, “Software process improvement in small and medium software enterprises: a systematic review”, *Software Quality Control*, vol. 16, pp. 237-261, Junio 2008.
- [2] M. E. Fayad, R. P. Ward y M. Laitinen, “Software Engineering in the Small”. *Communications of the ACM*, 2000. 43(3): p. 115-118.
- [3] L. Hareton, y Y. Terence, A process framework for small projects. *Software Process: Improvement and Practice*, 2001. 6(2): p. 67-83.
- [4] H. Saiedian, y N. Carr Characterizing a software process maturity model for small organizations. *ACM SIGICE Bulletin*, 1997. 23(1): p. 2-11

- [5] Curso: Scrum Master Certificación Mundial. Plataforma: Udemy. Video: Clase 4. El manifiesto Agile).
- [6] Kanbanize (2020). Kanban: explicación para principiantes [En línea] <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban>
- [7] M. Poppendieck, y T. Poppendieck. Implementing Lean Software Development: From Concept to Cash. The Addison Wesley Signature Series
- [8] J. Canós, H. Letelier. M. Penadés,. Metodologías ágiles en el Desarrollo de Software. Universidad Politécnica de Valencia. Metodologías Ágiles en el Desarrollo de Software Alicante España, 12 de Noviembre de 2003. pp. 18.
- [9] A. Rayo (2016/09/27), Lean Software Development (LSD): Los siete principios [En línea] Available: <https://www.bit.es/knowledge-center/lean-software-development-bsd-los-siete-principios>
- [10] OBS Business School (2020) Gestión de proyectos siguiendo el modelo de cascada [En línea] Available: <https://obsbusiness.school/es/blog-project-management/actualidad-project-management/gestion-de-proyectos-siguiendo-el-modelo-de-cascada>

Bogotá D.C. Julio 06 del 2020

Señores
Universidad Militar Nueva Granada
Repositorio Institucional UMNG

Por medio de la presente, nos permitimos autorizar la publicación en acceso abierto al Repositorio Institucional UMNG de la información inherente a la compañía TuProyecto S.A.S plasmada en el artículo titulado “ARQUITECTURA PARA EL DESARROLLO ÁGIL EN LA PRODUCCIÓN DE SOFTWARE BASADO EN LA EMPRESA TUPROYECTO S.A.S. PROYECTADO A STARTUPS Y PEQUEÑAS EMPRESAS” elaborado por el estudiante de la especialización en Gerencia Logística Integral, Hernan Dario Gomez Cuervo identificado con documento de identidad No. 1.032.455.513 y codigo estudiantil No. 9501000.

QUIEN AUTORIZA



Diego Armando Gómez Cuervo
Gerente General
TuProyecto S.A.S