

FACULTAD
DE CIENCIAS
ECONÓMICAS



Escuela de
Graduados
FCE · UNC



Maestría en Dirección de Negocios

Proyecto de Trabajo Final

“Mejora de Procesos en una empresa de Software”

Autor

Escudero, Ignacio Martín

(Cohorte 2022)

Tutor

Gauna, Marcos

Marzo 2024



Mejora de procesos en una empresa de Software by Ignacio Martín Escudero is licensed under [CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Contenido

1. Introducción.....	3
2. Presentación del Caso.....	4
3. Objetivos.....	11
4. Antecedentes Teóricos/ Marco Referencial.....	12
4.1. Metodologías ágiles:.....	12
4.1.1 Scrum:.....	14
4.1.2 Kanban:.....	19
4.1.3 Lean Software Development.....	23
4.2. Mejora de Procesos.....	25
4.3. Relación entre las metodologías expuestas y sus beneficios en procesos.....	26
5. Metodología.....	29
5.1. Documentar el proceso actual.....	30
5.2. Relevamiento de la estructura.....	32
5.3. Definición de nuevos roles.....	34
5.3.1. Team: Producto (Product Management).....	35
5.3.2. Team: Product Business (Equipo: Negocio de Producto).....	36
5.3.3. Rol: Scrum Product Owner.....	37
5.3.4. Rol: Scrum Master.....	38
5.4. Tableros Kanban.....	39
5.5. Definición de un nuevo proceso.....	43
5.5.1. Artifact: Product Roadmap (Hoja de Ruta de Producto).....	43
5.5.2. Artifact: Release Plan (Plan Estratégico de Lanzamientos).....	44
5.5.3. Artifact: Business Case (Caso de Negocio).....	44
5.5.4. Release.....	45
5.5.5. Refinamiento de Requisitos.....	47
5.5.6. Sprint Review.....	48
5.5.7. Comienzo/Cierre de Sprint.....	49
5.6. Indicadores de Medición.....	52
5.6.1. Requerimientos clasificados por estado y tipo.....	53
5.6.2. Cantidad de requerimientos por proyectos.....	53
5.6.3. Horas dedicadas por cliente.....	54
5.6.4. Tiempo de respuesta real vs. tiempo de respuesta previsto.....	54
5.6.5. Frecuencia de los Sprints.....	55
5.6.6. Planificación de requerimientos vs. Requerimientos ejecutados.....	56
6. Resultados.....	57
7. Conclusiones.....	63
BIBLIOGRAFÍA.....	65

Índice de Ilustraciones

Ilustración 1 - Entrada de requerimientos.....	6
Ilustración 2 - Salida de requerimientos.....	6
Ilustración 3 - Porcentaje de Satisfacción de clientes.....	8
Ilustración 4 - Metodologías ágiles más populares.....	9
Ilustración 5 - Porcentaje de Problemas en implementación de Scrum....	10
Ilustración 6 - Diferencia entre Agile y Waterfall.....	14
Ilustración 7 - Metodología Scrum.....	19
Ilustración 8 - Proceso de desarrollo - Situación inicial.....	31
Ilustración 9 - Organigrama de la empresa.....	33
Ilustración 10 - Rol de Product Management.....	36
Ilustración 11 - Equipo de Product Business.....	37
Ilustración 12 - Rol de Scrum Master.....	39
Ilustración 13 - Tablero Kanban desarrollo.....	40
Ilustración 14 - Tablero Kanban Scrum Master y Product Owner.....	41
Ilustración 15 - Proceso de un Business Case.....	45
Ilustración 16 - Detalle de un ticket.....	47
Ilustración 17 - Proceso completo.....	50
Ilustración 18 - Equipos que forman parte del nuevo proceso.....	51
Ilustración 19 - Documentación del nuevo proceso.....	52
Ilustración 20 - Nueva encuesta satisfacción de clientes.....	59
Ilustración 21 - Indicador de requerimientos por Tipo de Requerimientos	61
Ilustración 22 - Indicador Requerimientos por Estado.....	61

1. Introducción

En una empresa del rubro de IT, se identificó la ausencia de documentación y estandarización del proceso para el desarrollo de software, acompañado por una clara indefinición del mismo. Esto provoca una serie de dificultades y obstáculos que afectan directamente la eficiencia, la calidad del trabajo, la falta de planificación y quejas de los clientes. Además, no solo afecta la productividad interna, sino que también tiene consecuencias negativas en la satisfacción del cliente y la reputación de la compañía. Sin procesos definidos y documentación clara, los proyectos corren el riesgo de exceder los plazos, presentar errores recurrentes y no cumplir con las expectativas del cliente.

En estas empresas, el proceso de desarrollo de software ha experimentado cambios significativos desde principios de 2001 con la publicación del "Manifiesto Ágil". Han surgido nuevas metodologías para la gestión de procesos, tales como Scrum, Kanban y Extreme Programming (XP). Estas metodologías comenzaron a ganar popularidad en la industria debido a su capacidad para abordar los desafíos de los proyectos complejos y cambiantes. Las empresas se dieron cuenta de que al adoptar enfoques ágiles, eran más flexibles, les permitía adaptarse rápidamente a los cambios y entregar valor de manera más efectiva a sus clientes. Hoy en día, las metodologías ágiles son ampliamente utilizadas en empresas de desarrollo de software en todo el mundo. Han demostrado su eficacia al mejorar la colaboración, la calidad del producto y la satisfacción del cliente. La agilidad se ha convertido en un enfoque clave para la innovación y el éxito en el entorno empresarial actual, impulsando la evolución continua en la forma en que se desarrolla el software.

La identificación y resolución del inconveniente que enfrenta la empresa es esencial, ya que la carencia de procesos definidos, la ausencia de documentación y una estandarización deficiente generan ineficiencias que afectan las operaciones diarias. Estas deficiencias no solo comprometen la calidad del trabajo, sino que también pueden resultar en quejas por parte de los clientes, quienes esperan recibir productos o servicios de calidad, entregados puntualmente y que satisfagan sus expectativas.

A medida que la empresa experimenta un crecimiento y enfrenta un aumento en su carga de trabajo, los procesos ineficientes y no estandarizados se vuelven aún más problemáticos, limitando su capacidad de crecimiento y escalabilidad. Este escenario se traduce en pérdidas económicas significativas, que incluyen oportunidades perdidas debido a la falta de soluciones efectivas, la retención de nuevos clientes influida por referencias negativas de clientes insatisfechos, y una disminución en la productividad de los colaboradores.

A primera vista, esta situación podría llevar a la conclusión de que la empresa enfrenta un destino inevitable de fracaso en su industria. Sin embargo, mediante una adecuada implementación de cambios en los procesos, es posible eliminar redundancias, minimizar pasos innecesarios y optimizar

la ejecución de tareas. Este enfoque conduce a una mayor eficiencia en el uso de los recursos disponibles y una mejora sustancial en la productividad general.

Al estandarizar los procedimientos, se proporciona una guía clara para llevar a cabo las tareas de manera precisa y consistente. Esto contribuye a reducir los errores, mejorar la calidad del trabajo y asegurar el cumplimiento de los estándares establecidos. Además, al abordar los problemas en el proceso de desarrollo, se mejora la calidad del producto o servicio final, se garantiza una entrega más rápida y se cumplen las expectativas del cliente, lo que se traduce en una mayor satisfacción y, en última instancia, en el éxito y crecimiento sostenido de la empresa.

En conclusión, la mejora en el proceso del desarrollo de software es fundamental para que la misma se mantenga en un rubro donde predomina la búsqueda de la eficiencia, donde la competencia es cada vez más creciente, la satisfacción del cliente es clave y la reducción de costos es importante. Por lo tanto, se plantea la siguiente interrogante: **¿Cómo se podría implementar y medir adecuadamente un cambio del proceso del desarrollo de software para brindar una mayor productividad para la organización?**. En el presente trabajo, se buscará responder este interrogante.

2. Presentación del Caso

En primer lugar, es fundamental comenzar por comprender la empresa que será objeto de nuestro análisis para la mejora de procesos de desarrollo. Esta organización se dedica principalmente al desarrollo de software orientado al sector de la salud. Sus productos incluyen un sistema hospitalario, un software destinado a obras sociales y prepagas, así como un sistema de planificación de recursos empresariales (ERP) que puede ser comercializado de forma independiente o incorporado a los dos productos mencionados anteriormente. Los principales clientes de la empresa son prepagas y hospitales ubicados principalmente en las provincias de Córdoba y Buenos Aires.

El objetivo central de esta empresa es consolidarse como líder en el sector de la salud en un plazo máximo de cinco años. Esto implica que la organización tiene aspiraciones de crecimiento y liderazgo en un mercado altamente competitivo. En su nómina, la empresa cuenta con un total de 23 colaboradores distribuidos en diversas ubicaciones, siendo 12 de ellos en Buenos Aires, 9 en Córdoba, 1 en Entre Ríos y 1 en Rosario.

Un aspecto crítico para el éxito de esta empresa es la comunicación y coordinación entre sus empleados, dado que están dispersos en distintas provincias. La eficacia de la comunicación y la coordinación es esencial para alcanzar los objetivos estratégicos de la organización. La gestión eficaz de las operaciones, desde la implementación inicial hasta el soporte continuo, es esencial para brindar un servicio de calidad y garantizar la satisfacción del cliente.

En cuanto a las actividades que realiza la empresa, se dividen en dos fases principales: la implementación y el soporte. La etapa inicial comprende la implementación de los sistemas, lo que

incluye capacitación de usuarios, establecimiento de un cronograma de implementación y todas las actividades necesarias para poner en funcionamiento los productos. Posteriormente, la empresa ofrece un servicio de soporte post-implementación. En este caso, los clientes pagan una tarifa mensual para asegurarse asistencia en caso de problemas en los sistemas implementados y, además, se benefician de las nuevas funcionalidades que se incorporan a los productos estándar.

Sin embargo, no se puede pasar por alto el hecho de que esta industria también es altamente competitiva y se caracteriza por un entorno de constante cambio e innovación. La necesidad de adaptarse a nuevas regulaciones, avances tecnológicos y cambios en las preferencias de los clientes es constante. Los sistemas de información en el ámbito de la salud deben ser ágiles y flexibles para responder a estas dinámicas cambiantes.

La industria de la salud se caracteriza por ser dinámica y sujeta a cambios constantes y avances tecnológicos. Un ejemplo relevante es la adaptación que esta empresa tuvo que realizar a sus sistemas al comienzo de la pandemia, para permitir consultas médicas virtuales y la emisión de recetas digitales. Estos cambios, así como cualquier cambio normativo o avance tecnológico en el sector de la salud, demandan respuestas ágiles y eficientes por parte de la empresa. Esto se debe a la necesidad de cumplir con las expectativas de los clientes, que, como mencionamos anteriormente, son hospitales y prepagas reconocidas en la industria.

En un mundo cada vez más digitalizado, la importancia de la tecnología en el sector de la salud es innegable. Los sistemas de información y gestión desempeñan un papel crucial en la eficiencia de los hospitales, la prestación de atención médica de calidad y la administración de los servicios de las obras sociales y prepagas.

No responder con rapidez a estos desafíos puede resultar en una pérdida de oportunidades comerciales y la posibilidad de que la competencia capte a los clientes insatisfechos. En el peor de los casos, la empresa podría perder completamente a sus clientes si no es capaz de ofrecer respuestas rápidas y mejoras en sus productos y servicios.

Dada la relevancia de la industria de la salud y la importancia de la empresa en cuestión, es crucial abordar la mejora de sus procesos de desarrollo con profundidad y cuidado. En este trabajo, explicaremos el proceso que requiere atención y mejora, proponiendo soluciones y estrategias que ayudarán a la empresa a alcanzar sus objetivos a largo plazo y mantenerse competitiva en un entorno de rápida evolución.

Considerando los objetivos perseguidos por la compañía y siendo parte de ella, dentro de la gerencia de operaciones, es que detecto la necesidad de mejorar el proceso de desarrollo. La incorporación de nuevos clientes, a fines de 2022, provocó que los proyectos no avancen al ritmo al que nos habíamos comprometido; es así que se solicitó al área de planificación y la gerencia operativa que realice una medición de los requerimientos de clientes que ingresan vs. requerimientos resueltos en un período de tiempo. De este trabajo, se obtuvieron los siguientes resultados para el período de

agosto del año 2022, donde se comienza a medir la información que surge del sistema de requerimientos internos de la organización denominado “Redmine”:

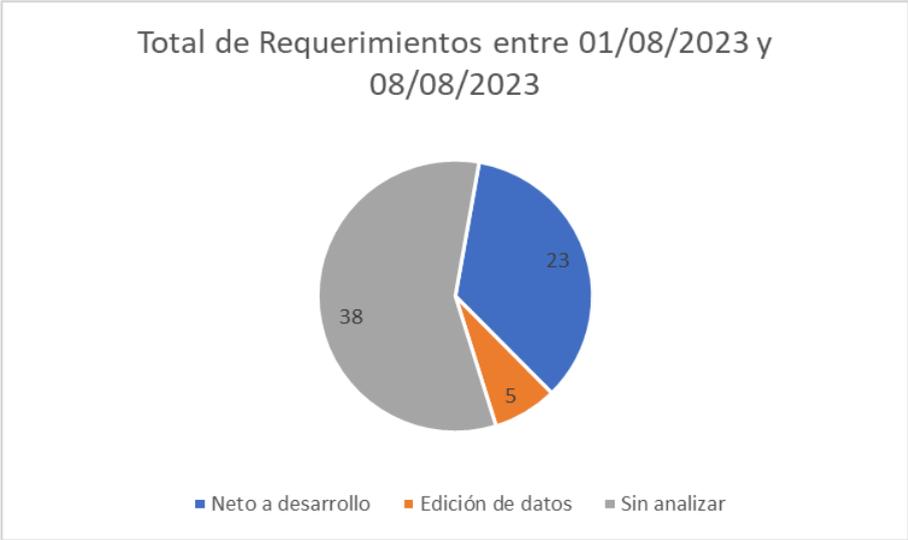


Ilustración 1 - Entrada de requerimientos



Ilustración 2 - Salida de requerimientos

Estos resultados corresponden a la primer semana que se comenzó a medir, donde se observaron 23 requerimientos que ingresaron para resolver, mientras que los requerimientos solucionados para esa misma semana fueron 14. Se observa que existe un 40% de requerimientos sin

realizarse y que los mismos se iban acumulando en el siguiente período, que tenía un comportamiento bastante similar con el mencionado.

A partir de esta medición, se analizaron las posibles causas de ese 40% de requerimientos sin realizarse semanalmente. En primer lugar, se mencionó la posibilidad de poder incrementar la nómina para que pudiera resolverse el inconveniente. Luego, a medida que se profundiza el análisis se observó que los principales inconvenientes surgían del proceso de desarrollo, ya que el mismo no se encuentra documentado. Esto generaba que cada una de las personas que intervienen en el mismo, realicen las tareas a su consideración sin un orden específico, llevando a no poder estandarizar el proceso.

Por otra parte, en reuniones personales con cada una de las personas intervinientes en el proceso, se pudo identificar que existían conflictos entre las áreas debido a falta de especificaciones de las tareas a realizar. La falta de un intermediario entre las áreas del proceso (áreas de análisis funcional y área de desarrollo), generaba discusiones que, en muchos casos, solían llevar demasiado tiempo. Ya no siendo el único problema la falta de un proceso estandarizado, sino que también se mezclaban los roles de los miembros y las tareas que cada uno realizaba.

Por otro lado, en la encuesta de satisfacción al cliente que se realiza de manera anual, se observó que los indicadores de satisfacción no eran los esperados y que si bien las respuestas sobre la calidad y confiabilidad del producto eran las esperadas, no estaban los mismos resultados al momento de que el cliente respondiera sobre la satisfacción sobre el soporte y la velocidad de respuesta antes requerimientos.

A continuación, se presenta un resumen de los resultados finales obtenidos en la última encuesta de satisfacción dirigida a los 14 clientes de nuestra empresa. Cada cliente evaluó diez aspectos clave en una escala del 1 al 10. Los primeros cinco aspectos, diferenciados en el gráfico en azul, se relacionan con el soporte al cliente, mientras que los siguientes tres, resaltados en naranja, abordan la satisfacción con nuestros productos. Los dos últimos aspectos se centran en la valoración de la capacitación recibida sobre nuestro sistema. Observamos que los niveles de satisfacción oscilan alrededor del 70% en los puntos más altos, mientras que aspectos como la velocidad de respuesta, el soporte remoto y la satisfacción con las respuestas recibidas muestran niveles de satisfacción cercanos al 40%, evidenciando que son claras áreas de mejora identificadas por nuestros clientes:



Ilustración 3 - Porcentaje de Satisfacción de clientes

Esto llevó a replantear el proceso interno para poder verificar si un cambio en el mismo puede ser lo suficientemente importante como para poder mejorar estos indicadores que no están siendo buenos, como es el caso de satisfacción de clientes, el bienestar interno de los colaboradores y la buena relación entre las áreas y la capacidad de poder dar respuestas en tiempo según lo solicitado por el cliente.

Debido a ello, se revisó cuáles son las prácticas que se toman en otras empresas del rubro de desarrollo de software, pudiendo observar que gran parte de las compañías trabajan con metodologías ágiles, principalmente se encuentra muy difundido Scrum, como así también el concepto de Kanban y Lean cuyas características ya fueron comentadas en el apartado anterior. Por lo tanto, la implementación de una mejora en el proceso apalancada en estos conceptos no suena descabellada. A largo plazo, impactarán ineludiblemente en un aumento de la productividad, la satisfacción del cliente e incentivará la proactividad del equipo.

Una de las principales metodologías para trabajar dentro de la empresa es el concepto de Scrum y, tal como lo muestra la siguiente estadística, es de gran popularidad dentro de las empresas del rubro:

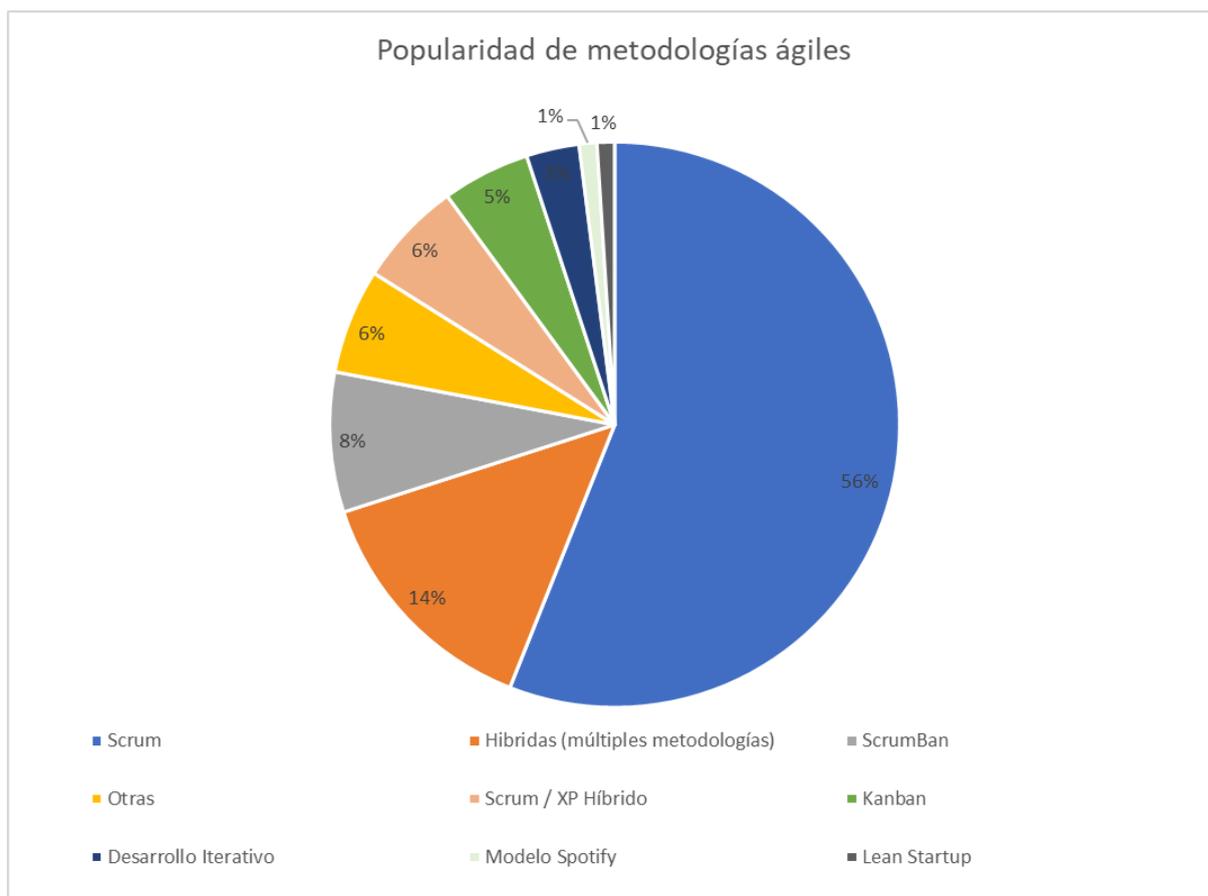


Ilustración 4 - Metodologías ágiles más populares

Además de los hallazgos destacados, es fundamental examinar con mayor profundidad las investigaciones relacionadas con la implementación de metodologías ágiles como Scrum en empresas similares. Un relevamiento exhaustivo se ha llevado a cabo para identificar las posibles causas subyacentes que pueden conducir al fracaso de una implementación efectiva de Scrum. Entre los resultados de esta investigación, se destaca que la principal causa identificada se relaciona con la cultura organizacional, la cual puede representar un obstáculo significativo para la adopción exitosa de esta metodología. Esta conclusión subraya la importancia crítica de comprender y abordar los aspectos culturales dentro de la organización al considerar cualquier cambio en los procesos que afecten al personal.

El análisis detallado de estos resultados revela la necesidad de un enfoque estratégico y multifacético para gestionar la implementación de Scrum en el contexto de estas empresas. Además de abordar los aspectos culturales, es crucial considerar otros factores clave, como la capacitación adecuada del personal, la comunicación efectiva de los objetivos del cambio y la alineación de los procesos y estructuras organizacionales con los principios ágiles. Estos hallazgos respaldan la

importancia de adoptar un enfoque holístico y centrado en las personas para garantizar el éxito a largo plazo de la implementación de Scrum y otras metodologías ágiles en el entorno empresarial.

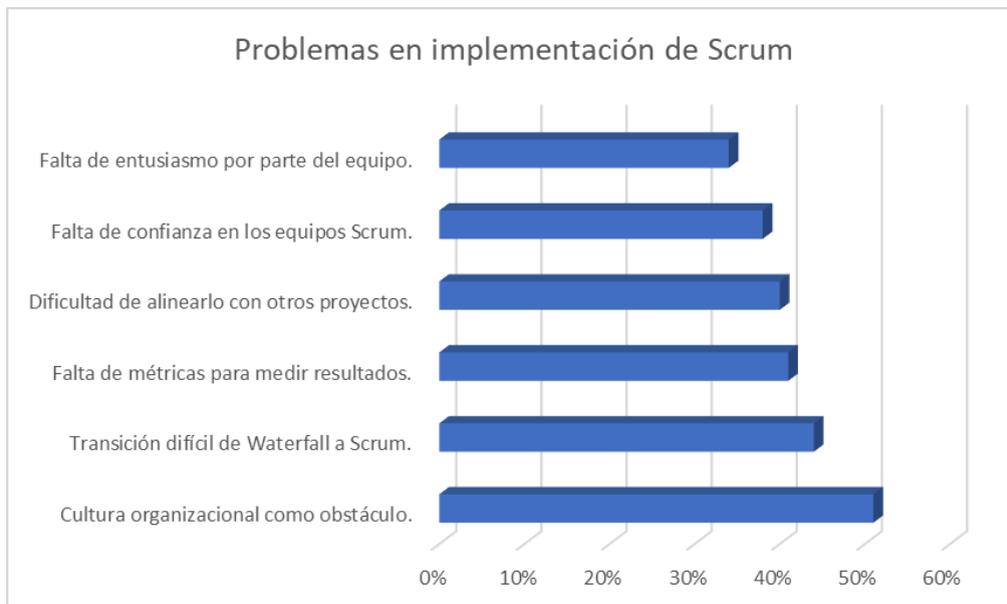


Ilustración 5 - Porcentaje de Problemas en implementación de Scrum

La información anteriormente mencionada es de vital importancia para determinar aquellos aspectos a considerar a la hora de plantear una modificación del proceso. Al mismo tiempo, se tendrán en consideración al momento de realizar la implementación para reducir el impacto negativo que pueda generar en el ambiente laboral y en cada uno de los colaboradores. Para ello se estableció la premisa de basar el cambio del proceso en tres pilares fundamentales:

1) Planificación del Sprint: es la reunión al inicio de un sprint en la que los equipos desglosan las descripciones de alto nivel en tareas detalladas. Esto les ayuda a hacer una estimación detallada de la cantidad de trabajo que necesitan hacer durante el sprint. El 86% de los equipos Scrum utilizan la planificación de sprints para planificar ciclos que suelen durar 2 semanas (Estudio sobre el estado de Scrum 2017-2018).

2) Daily: describe una reunión diaria, de un máximo de 15 minutos, en la que los miembros del equipo discuten el progreso de su trabajo y los obstáculos. La idea básica es aumentar la comunicación dentro del equipo. El 87% de los equipos Scrum utilizan el Scrum Diario (Estudio sobre el estado de Scrum 2017-2018).

3) Retrospectivas: son quizás el evento ágil más importante que puede utilizarse para introducir la mentalidad ágil en los equipos. Al final del sprint de 2 semanas, el equipo Scrum reflexiona sobre lo que ha ido bien y lo que ha ido mal. Las retrospectivas son, por tanto, el evento central de una idea básica de los métodos de trabajo ágiles: la mejora continua.

Según los estudios realizados en diferentes empresas, el factor importante para evitar el fracaso es concentrarse en los tres puntos antes mencionados, ya que de esta manera se pueden anticipar los problemas que puedan llegar a generarse ante la implementación del cambio.

En último lugar, para poder llevar adelante la transformación del proceso, se definió comenzar por la documentación del proceso actual, ya que no se encontraba nada claro. En base a ello, poder detectar los principales puntos en los que se deberá mejorar.

Un factor importante es el estudio del organigrama para verificar que cambios en la estructura son necesarios o si la misma se encuentra preparada para el cambio propuesto.

Una vez que completados los pasos iniciales, se continuará por la redefinición de roles importantes para poder llevar a cabo la implementación del nuevo proceso en base al marco teórico antes mencionado.

Posteriormente, se avanzará con la definición de los tableros Kanban para cada uno de los participantes del proceso con la intención de poder optimizar el trabajo de cada uno de ellos y evitar la sobrecarga.

Una vez definido el nuevo proceso a implementar, mediremos el mismo para evaluar su eficiencia y las mejoras buscadas en los puntos comunicacionales.

3. Objetivos

El objetivo general del presente trabajo consiste en diseñar e implementar un proceso para el desarrollo proyectos que permitan incrementar la productividad y la satisfacción del cliente.

Mientras que como objetivos específicos podemos mencionar los siguientes:

- Establecer la criticidad del proceso dentro de la empresa y sus principales desafíos e inconvenientes.
- Definir los puntos importantes del proceso que deberían ser mejorados con la intención de aumentar la productividad.
- Diseñar una propuesta de mejora en base a la información relevada y que la misma sea acorde a la situación actual de la empresa.
- Medir el impacto de la propuesta realizada en comparación con la situación anterior, para evaluar su efectividad.

4. Antecedentes Teóricos/ Marco Referencial

Contar con un proceso definido es fundamental para las empresas del rubro de desarrollo de software, ya que les permite mejorar la eficiencia y efectividad en el desarrollo de producto; en una industria cada vez más competitiva y exigente, debido a la necesidad de ofrecer soluciones innovadoras y de calidad. Además, *el tamaño del mercado de análisis de composición de software está creciendo a una tasa compuesta anual del 21,7% para 2027(1)*. En este contexto, es fundamental que las empresas de desarrollo de software adopten procesos de desarrollo eficientes y efectivos para adaptarse rápidamente a los cambios y ofrecer productos de alta calidad en un corto período de tiempo. La innovación y la competitividad son factores clave en la industria del software, y las empresas deben estar dispuestas a asumir riesgos y tomar decisiones para mantenerse en el mercado.

En relación a la historia de las empresas del rubro de software en cuanto a las metodologías adoptadas para mejorar los procesos, se puede decir que ha habido una evolución significativa en los últimos años. En el pasado, las empresas de software solían utilizar metodologías tradicionales como el modelo en cascada, que se caracterizaba por ser rígido y poco flexible. Sin embargo, con el tiempo, las empresas comenzaron a adoptar metodologías ágiles como Scrum (Sutherland et al., 1993) y Kanban (Ohno.,1985), que se enfocan en la colaboración, la comunicación y la flexibilidad. Estas metodologías permiten a los equipos de desarrollo adaptarse rápidamente a los cambios en los requisitos del cliente y en el mercado, lo que les permite ofrecer soluciones más innovadoras y de alta calidad. En resumen, las empresas del rubro de software han evolucionado en cuanto a las metodologías adoptadas para mejorar los procesos, pasando de metodologías tradicionales a metodologías ágiles que se enfocan en la colaboración, la comunicación y la flexibilidad.

4.1. Metodologías ágiles:

Al hablar de empresas de desarrollo de software debemos hablar de metodologías ágiles, estas metodologías ágiles no fueron creadas por una sola persona, sino que surgieron como resultado de la colaboración y el trabajo conjunto de diferentes expertos en el campo del desarrollo de software. Sin embargo, uno de los eventos más importantes en la historia de las metodologías ágiles fue la publicación del "Manifiesto Ágil" en el año 2001. Este manifiesto fue redactado por diecisiete profesionales de la industria del software, incluidos Kent Beck, Martin Fowler, Jeff Sutherland y Ken Schwaber, entre otros.

La finalidad de las metodologías ágiles es mejorar la forma en que los equipos de desarrollo trabajan, permitiéndoles adaptarse a las necesidades cambiantes, entregar valor de manera continua y satisfacer a los clientes mediante una mayor colaboración y enfoque en la calidad del producto. Estas

metodologías han revolucionado la forma en que se desarrolla el software y han impulsado una cultura de mejora continua en la industria del desarrollo de software.

Es recomendable utilizar metodologías ágiles en una empresa de desarrollo de software por varias razones:

- **Adaptabilidad:** Las metodologías ágiles permiten a los equipos adaptarse rápidamente a los cambios en los requisitos del cliente o en el entorno del proyecto. Al trabajar en iteraciones cortas y entregas incrementales, las metodologías ágiles brindan la flexibilidad necesaria para responder a las necesidades cambiantes del mercado.
- **Entrega de Valor Continua:** Las metodologías ágiles se centran en entregar incrementos de software funcional en cada iteración o Sprint. Esto permite que el cliente obtenga valor en etapas tempranas del proyecto y brinda la oportunidad de obtener feedback valioso para mejorar el producto en desarrollo.
- **Colaboración y Comunicación:** Las metodologías ágiles fomentan la colaboración y la comunicación activa entre todos los miembros del equipo y con los stakeholders. Esto ayuda a asegurar que todos estén alineados con los objetivos del proyecto y trabajen en conjunto para lograrlos.
- **Mejora Continua:** Las metodologías ágiles incorporan prácticas de retrospectiva que permiten a los equipos reflexionar sobre su desempeño y buscar formas de mejorar continuamente sus procesos y resultados.
- **Reducción de Riesgos:** Al entregar productos de manera incremental y frecuente, las metodologías ágiles permiten identificar y abordar riesgos potenciales más temprano en el proceso, lo que ayuda a minimizar problemas y sorpresas en etapas posteriores del desarrollo.
- **Satisfacción del Cliente:** Al involucrar al cliente de manera activa y continua en el proceso de desarrollo, las metodologías ágiles garantizan que el producto final se alinee mejor con las expectativas y necesidades del cliente.

Metodologías ágiles como Scrum, Kanban, XP (eXtreme Programming), entre otras, comparten principios y valores similares que se centran en la flexibilidad, la entrega de valor y la colaboración efectiva.

A continuación, se presenta un cuadro comparativo entre la metodología ágil mencionada anteriormente, y la metodología en cascada o “waterfall” que es la metodología utilizada anteriormente en las empresas del rubro, en la misma podemos ver claramente diferencias entre los principales aspectos de la gestión.

	Waterfall	Agile
Enfoque	Procesos ✓	Personas
Documentación	Exhaustiva	Mínima requerida ✓
Estilo de procesos	Lineal ✓	Iterativo (Sprints)
Planificación por adelantado	Alta	Baja ✓
Priorización de los requisitos	Conforme al plan de trabajo ✓	Conforme a valor del negocio y del cliente
Aseguramiento de la calidad	Centrada en el proceso ✓	Centrada en el cliente
Organización	Gestionada ✓	Auto-organizada
Estilo de gestión	Centralizado ✓	Descentralizado
Cambio	Sistema formal de gestión del cambio ✓	Basado en la priorización del Backlog
Liderazgo	Mando y control	Colaborativo, liderazgo servicial ✓
Medición del rendimiento	Al fin del proyecto ✓	Valor entregado al negocio
Retorno de inversión (ROI)	Varía en función del ciclo de vida del proyecto	Al comienzo y a lo largo del proyecto ✓
Participación del cliente	Varía en función del ciclo de vida del proyecto.	Alta durante todo el proyecto ✓

Ilustración 6 - Diferencia entre Agile y Waterfall

Sobre cada uno de los puntos se indicó el estado en el que se encuentra actualmente la empresa, donde se puede observar que si bien tiene la mayor parte de sus puntos con un enfoque de “Waterfall”, también tiene ciertos aspectos que se encuentran mas cercanos a la metodología “Agile” y esos aspectos pueden ser los puntos de partida para poder comenzar a trabajar sobre el proceso de la empresa.

4.1.1 Scrum:

La teoría de Scrum es una metodología ágil de gestión y desarrollo de proyectos que se ha convertido en una opción popular para empresas de desarrollo de software. Esta metodología se basa en la colaboración, la adaptabilidad y la entrega incremental, lo que permite a los equipos de desarrollo entregar productos de alta calidad de manera más eficiente. En esta sección, se explorará en detalle la teoría de Scrum, sus principios y prácticas clave, y cómo puede aplicarse en una empresa de desarrollo de software para mejorar la productividad y la satisfacción del cliente.

La teoría de Scrum fue propuesta por primera vez en 1986 por Hirotaka Takeuchi e Ikujiro Nonaka en un artículo titulado "El nuevo nuevo producto: desarrollo de productos y equipos de proyectos" publicado en la revista Harvard Business Review. En este artículo, los autores describieron un enfoque de desarrollo de productos que se centraba en equipos pequeños, multifuncionales y autoorganizados, en contraposición al enfoque tradicional de desarrollo en cascada.

El término "Scrum" proviene del rugby, donde se refiere a una formación de juego en la que los jugadores se unen para avanzar hacia el objetivo común de marcar un gol. Jeff Sutherland y Ken Schwaber fueron los dos primeros defensores de Scrum como metodología de gestión de proyectos en el ámbito de desarrollo de software. En 1995, Schwaber presentó formalmente Scrum en una conferencia, y en 2001, ambos autores, junto con otros diecisiete profesionales, redactaron el "Manifiesto Ágil" que dio lugar al movimiento ágil. La teoría de Scrum se basa en varios principios clave que guían su implementación en una empresa de desarrollo de software:

- **Transparencia:** Toda la información relevante sobre el proyecto debe ser visible y accesible para todos los miembros del equipo y las partes interesadas.
- **Inspección:** Se realiza una revisión regular del progreso y los resultados del trabajo del equipo para identificar posibles problemas y oportunidades de mejora.
- **Adaptación:** Con base en las inspecciones, el equipo adapta su enfoque y planificación para abordar eficazmente los cambios y desafíos que surgen durante el proyecto.
- **Colaboración:** Se fomenta la colaboración activa y continua entre los miembros del equipo y con los clientes para asegurar la alineación y el éxito del proyecto.

En la metodología de Scrum existen roles predefinidos, los mismos fueron definidos inicialmente por Ken Schwaber y Jeff Sutherland en su documento "SCRUM Development Process" en 1995. En Scrum, la metodología de desarrollo de software ágil, existen tres roles clave que trabajan de manera conjunta para llevar a cabo el desarrollo del producto. Cada uno de estos roles tiene responsabilidades específicas y contribuye al éxito del proyecto. Los roles en Scrum son los siguientes:

- **Product Owner (Propietario del Producto):**

El Product Owner es el representante del cliente y los stakeholders (partes interesadas) dentro del equipo de desarrollo. Es responsable de maximizar el valor del producto resultante y gestionar adecuadamente el backlog (lista de tareas) del proyecto. Sus principales responsabilidades incluyen:

Definir la Visión: El Product Owner debe tener una visión clara del producto que se va a desarrollar y comunicarla al equipo. Esta visión guiará las decisiones y prioridades durante el proyecto.

Priorizar el Backlog: El backlog contiene todas las funcionalidades, mejoras y tareas pendientes. El Product Owner debe priorizar los elementos del backlog según su valor y necesidad para el cliente.

Desarrollar Historias de Usuario: Las Historias de Usuario son descripciones breves de las funcionalidades desde la perspectiva del usuario. El Product Owner trabaja con los stakeholders y el equipo para crear historias claras y comprensibles.

Aceptar el Trabajo Terminado: Al finalizar cada sprint (iteración), el Product Owner revisa el trabajo completado por el equipo y asegura que cumpla con los criterios de aceptación y las expectativas del cliente.

Colaborar con el Equipo: El Product Owner trabaja estrechamente con el equipo de desarrollo, brindando claridad sobre los requisitos y aclarando dudas para que el equipo pueda realizar un trabajo eficiente y acorde a las necesidades.

- Scrum Master (Facilitador del Proceso):

El Scrum Master actúa como facilitador y defensor del equipo Scrum, asegurando que se sigan los principios y prácticas de Scrum. Su objetivo es eliminar obstáculos y mejorar continuamente la forma en que el equipo trabaja. Sus responsabilidades clave incluyen:

Facilitar las reuniones: El Scrum Master organiza y lidera las reuniones, como las reuniones diarias, la planificación del sprint, la revisión del sprint y la retrospectiva del sprint.

Eliminar Obstáculos: El Scrum Master trabaja para identificar y eliminar cualquier obstáculo o impedimento que pueda afectar la productividad o el progreso del equipo.

Proteger al Equipo: El Scrum Master protege al equipo de distracciones externas y asegura que se cumplan los acuerdos establecidos en el proceso de Scrum.

Fomentar la Mejora Continua: El Scrum Master impulsa la mejora continua, ayudando al equipo a reflexionar sobre su desempeño y encontrar formas de ser más efectivos en cada sprint.

Facilitar la comunicación: El Scrum Master promueve la comunicación abierta y transparente entre los miembros del equipo y con otras partes interesadas.

- Equipo de Desarrollo:

El Equipo de Desarrollo es un grupo de profesionales multifuncionales y autoorganizados que realizan el trabajo real de desarrollo del producto. Pueden incluir desarrolladores, diseñadores, testers, entre otros. Sus principales características y responsabilidades son:

Desarrollar el Producto: El equipo trabaja en conjunto para diseñar, desarrollar, probar y entregar incrementos de producto durante cada sprint.

Autonomía y Autoorganización: El equipo tiene la libertad de decidir cómo llevar a cabo el trabajo y cómo cumplir con los objetivos del sprint.

Colaboración: La colaboración activa y la comunicación clara dentro del equipo son fundamentales para el éxito de Scrum.

Responsabilidad Compartida: Todos los miembros del equipo son responsables del éxito del sprint y del producto en su conjunto.

Mejora Continua: El equipo busca formas de mejorar su eficiencia, calidad y procesos en cada retrospectiva del sprint.

Cada uno de estos roles en Scrum tiene un papel específico y contribuye al éxito general del proyecto. La colaboración y la comunicación efectiva entre el Product Owner, el Scrum Master y el Equipo de Desarrollo son fundamentales para el funcionamiento adecuado de Scrum y para alcanzar los objetivos del proyecto de desarrollo de software.

En Scrum, la metodología ágil de desarrollo de software, se utilizan prácticas y elementos específicos para gestionar y organizar el trabajo del equipo. Estas prácticas y artefactos están diseñados para promover la transparencia, la inspección y la adaptación, lo que permite que el equipo entregue productos de alta calidad de manera incremental y colaborativa. A continuación, se describen en detalle algunas de las prácticas más importantes en Scrum:

- Reuniones Diarias (Daily Stand-up):

Es una reunión corta y diaria en la que todos los miembros del equipo se ponen al día sobre el progreso del trabajo. Se realiza de pie y suele tener una duración de 15 minutos o menos. En esta reunión, cada miembro del equipo responde a tres preguntas:

¿Qué hice ayer?

¿Qué haré hoy?

¿Hay algún obstáculo o impedimento que me esté bloqueando?

El objetivo de la reunión diaria es fomentar la colaboración, identificar rápidamente los problemas y mantener a todos en la misma página sobre el estado del proyecto.

- Planificación del Sprint (Sprint Planning):

La Planificación del Sprint es una reunión al inicio de cada Sprint (iteración), en la que el equipo selecciona las tareas que se abordarán durante el Sprint y define el plan para alcanzar los objetivos acordados. La reunión se divide en dos partes:

Parte 1: El Product Owner presenta las funcionalidades y tareas más importantes del backlog al equipo. El equipo hace preguntas y obtiene una comprensión clara de los requisitos.

Parte 2: El equipo de desarrollo colabora para estimar el esfuerzo necesario para completar cada tarea. Basándose en estas estimaciones y la capacidad del equipo, se determina qué tareas se incluirán en el Sprint.

- Revisión de Sprint (Sprint Review):

Al finalizar cada Sprint, el equipo lleva a cabo la Revisión de Sprint para demostrar y recibir feedback sobre el trabajo completado durante el Sprint. El Product Owner, los stakeholders y otros interesados pueden asistir a esta reunión. Durante la Revisión de Sprint, el equipo presenta las funcionalidades desarrolladas y recopila comentarios para futuras mejoras.

- Retrospectiva de Sprint (Sprint Retrospective):

Después de la Revisión de Sprint, el equipo realiza la Retrospectiva de Sprint, donde reflexionan sobre el Sprint que acaba de finalizar. El objetivo es identificar lo que funcionó bien, los desafíos enfrentados y las oportunidades de mejora. Con base en estas reflexiones, el equipo define acciones concretas para mejorar el proceso en el próximo Sprint.

En lo que respecta a elementos de la metodología se pueden mencionar las siguientes:

- Product Backlog:

El Product Backlog es una lista priorizada de todas las funcionalidades, mejoras y tareas que se desean desarrollar en el producto. Es mantenido y actualizado por el Product Owner y debe estar siempre disponible y accesible para todo el equipo. El Product Backlog es dinámico y se ajusta a medida que se adquiere un mayor conocimiento sobre el producto y las necesidades del cliente.

- Sprint Backlog:

El Sprint Backlog es una lista de tareas seleccionadas del Product Backlog que el equipo de desarrollo se compromete a completar durante el Sprint actual. Se crea en la reunión de Planificación del Sprint y contiene los elementos que el equipo ha estimado que pueden ser completados dentro del período del Sprint.

- Incremento de Producto:

El Incremento de Producto es el resultado del trabajo realizado durante un Sprint. Es un producto funcional y potencialmente entregable que agrega valor al producto en desarrollo. Cada Incremento debe ser coherente y compatible con todos los Incrementos anteriores. Con cada Sprint, el producto se enriquece y evoluciona hacia una versión más completa y valiosa.

Estas prácticas y elementos en Scrum trabajan en conjunto para organizar y gestionar el trabajo de manera transparente y adaptativa. Al utilizarlos adecuadamente, los equipos de desarrollo de software pueden lograr entregas regulares de valor a sus clientes, mantener una mejora continua y ser más eficientes en sus proyectos.

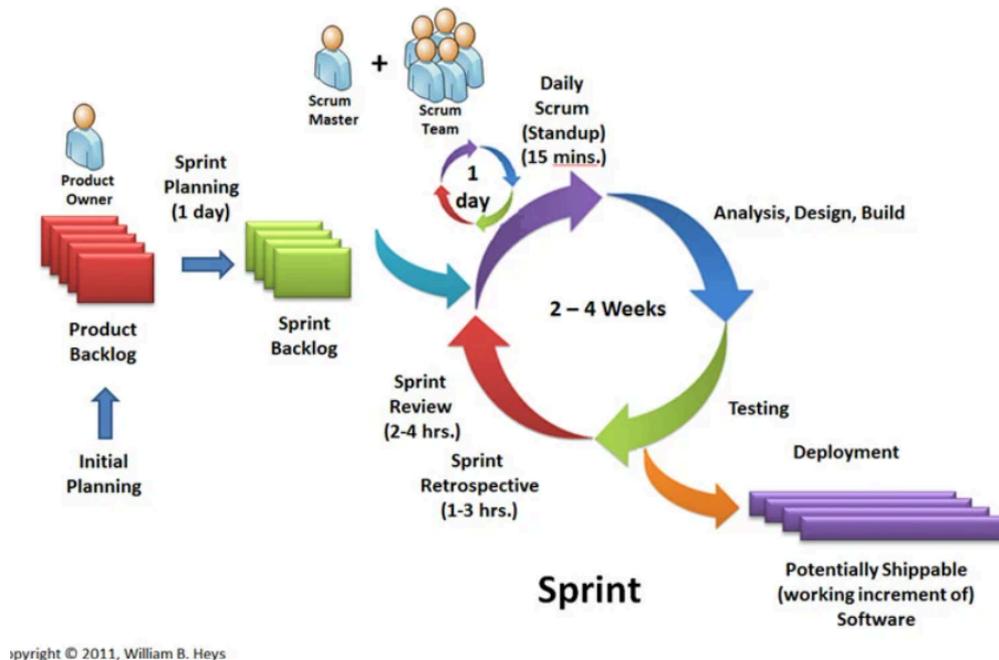


Ilustración 7 - Metodología Scrum

Como conclusión podemos mencionar que la teoría de Scrum en una empresa de desarrollo de software proporciona una metodología ágil que enfatiza la colaboración, la adaptabilidad y la entrega incremental. Sus principios, roles, prácticas y artefactos trabajan en conjunto para mejorar la eficiencia y la satisfacción del cliente en el desarrollo de productos. A través de su enfoque iterativo e incremental, Scrum permite a las empresas adaptarse rápidamente a las necesidades cambiantes del mercado y entregar productos de alta calidad de manera más efectiva. Al adoptar Scrum, las empresas pueden aumentar su capacidad para responder a las demandas del mercado y mantenerse competitivas en la industria del desarrollo de software.

4.1.2 Kanban:

La teoría de Kanban es una parte fundamental de las metodologías ágiles, especialmente en el ámbito del desarrollo de software. Originario de la filosofía Lean Manufacturing, Kanban ha sido adoptado y adaptado para mejorar la eficiencia y la calidad en el desarrollo de productos y servicios en diversas industrias. En esta sección, se explorará en detalle la teoría de Kanban, su evolución histórica, sus componentes principales y los actores involucrados en su implementación. Tiene sus raíces en Toyota, Japón, en la década de 1950. El término "Kanban" significa "tarjeta visual" en japonés, y se refiere a un sistema de señalización que Toyota utiliza para gestionar la producción en su sistema Just-in-Time (JIT). Taiichi Ohno, ingeniero y ejecutivo de Toyota, es considerado el padre del sistema

Kanban. Ohno desarrolló este sistema para minimizar el inventario y maximizar la eficiencia de producción al proporcionar una señal visual para reponer los materiales solo cuando se necesitaban.

A lo largo de los años, la filosofía Lean Manufacturing y el sistema Kanban se extendieron más allá de Toyota y se aplicaron en otras empresas y sectores. En la década de 2000, el método Kanban fue adaptado para el desarrollo de software por David J. Anderson y otros pioneros de la comunidad ágil. Desde entonces, Kanban se ha convertido en una metodología ágil muy utilizada para gestionar y mejorar los flujos de trabajo en proyectos de desarrollo de software y en otros contextos.

La metodología de kanban tiene ciertos componentes que son característicos en esta metodología:

- Tablero Kanban:

El tablero Kanban es una herramienta visual utilizada en la metodología Kanban para representar y gestionar el flujo de trabajo de un equipo o proceso. Es una forma de tablero de tareas que proporciona una vista clara y transparente del estado actual de las actividades en curso y facilita la visualización de los elementos de trabajo a medida que avanzan a través del proceso.

El tablero Kanban se compone de columnas que representan las diferentes etapas o estados del flujo de trabajo. Cada columna tiene un título que describe la etapa en la que se encuentra el trabajo. Las tarjetas Kanban, que representan los elementos de trabajo, se mueven horizontalmente a través del tablero desde una columna a otra a medida que avanzan en el proceso.

Las columnas más comunes en un tablero kanban son: Por hacer (trabajo que aún no ha comenzado), En Progreso (Aquellas actividades que están en curso y aún no finalizado), En Revisión (Trabajo completado, pero que se encuentra en etapa de revisión), Terminado (Trabajo listo y completo para ser entregado o liberado a cliente).

Cada tarjeta Kanban suele contener información relevante, como una descripción del trabajo, el responsable, fechas límite y cualquier otra información necesaria para comprender el estado y el contexto del elemento de trabajo.

El tablero Kanban permite a los miembros del equipo tener una visión clara y en tiempo real del flujo de trabajo, lo que facilita la identificación de cuellos de botella, la priorización adecuada y la optimización del proceso. Además, al ser una herramienta visual, el tablero Kanban promueve la transparencia y mejora la comunicación y colaboración entre los miembros del equipo y otras partes interesadas.

En resumen, el tablero Kanban es una poderosa herramienta visual que permite a los equipos de desarrollo de software y otras áreas de trabajo gestionar de manera efectiva el flujo de trabajo y mejorar la eficiencia en sus procesos.

- Tarjetas kanban:

En el contexto de Kanban, las tarjetas son representaciones visuales de elementos de trabajo individual, como tareas, historias de usuario o cualquier otra unidad de trabajo que forma parte del flujo de trabajo del equipo. Estas tarjetas se utilizan en el tablero Kanban para visualizar y gestionar el estado y el progreso de cada elemento de trabajo a medida que avanza a través del proceso.

Las tarjetas Kanban suelen ser pequeñas y pueden estar hechas de papel, cartulina o incluso de forma digital en herramientas de gestión de proyectos o software específico de Kanban. Cada tarjeta representa una sola unidad de trabajo y contiene información relevante sobre el elemento de trabajo que representa.

Descripción del Trabajo: Una breve descripción de la tarea o elemento de trabajo que se realizará.

Responsable: El miembro del equipo o persona asignada para llevar a cabo la tarea.

Fecha Límite: La fecha límite o plazo para completar la tarea.

Etiqueta o Categoría: Pueden utilizarse etiquetas o categorías para clasificar las tarjetas por tipo de trabajo, prioridad o cualquier otra característica relevante.

Estado: El estado actual del elemento de trabajo, es decir, en qué etapa del flujo de trabajo se encuentra la tarea.

Información Adicional: Cualquier información adicional que pueda ser relevante para el equipo o que facilite la comprensión y ejecución de la tarea.

Las tarjetas Kanban se mueven horizontalmente a través del tablero Kanban, desde la columna "Por Hacer" hasta la columna "Terminado", a medida que avanza el trabajo. El movimiento de las tarjetas entre las columnas del tablero representa el progreso de las tareas y permite a todos los miembros del equipo tener una vista en tiempo real del estado del flujo de trabajo.

Las tarjetas Kanban son una herramienta esencial en la metodología Kanban, ya que proporcionan una manera visual y sencilla de gestionar y optimizar el flujo de trabajo, al tiempo que promueven la transparencia, la comunicación y la colaboración entre los miembros del equipo.

- Límites de Trabajo en Progreso (WIP):

Los Límites de Trabajo en Progreso (WIP) en Kanban son una restricción establecida en cada columna del tablero Kanban que limita la cantidad de elementos de trabajo que pueden estar en curso o en proceso a la vez. Estos límites están diseñados para evitar la sobrecarga del equipo y mantener un flujo de trabajo equilibrado y eficiente.

Cada columna en el tablero Kanban representa una etapa o estado del flujo de trabajo, y los límites de WIP se aplican individualmente a cada columna. Por ejemplo, si se establece un límite WIP de 3 en la columna "En Progreso", significa que el equipo solo puede tener hasta 3 tarjetas Kanban en esta columna a la vez.

El propósito de los límites WIP en Kanban es:

Evitar la Sobrecarga: Al limitar la cantidad de elementos de trabajo en proceso, se evita que el equipo se vea abrumado por una carga excesiva de trabajo y se asegura de que puedan concentrarse en completar las tareas en curso antes de agregar más trabajo.

Identificar Cuellos de Botella: Si una columna alcanza su límite WIP, indica que hay demasiado trabajo acumulado en esa etapa del proceso. Esto puede señalar un cuello de botella y permite que el equipo identifique y aborde los problemas que están afectando el flujo de trabajo.

Fomentar el Enfoque y la Calidad: Al limitar la cantidad de trabajo en curso, el equipo puede enfocarse en completar las tareas antes de tomar nuevas. Esto promueve una mayor dedicación a cada elemento de trabajo y puede conducir a una mayor calidad en las entregas.

Mejorar la Eficiencia: Al mantener un flujo de trabajo equilibrado y evitar la acumulación de trabajo en una etapa, el equipo puede mejorar la eficiencia general del proceso.

Es importante señalar que los límites WIP no son estáticos y pueden ajustarse con el tiempo según las necesidades del equipo y la demanda de trabajo. Si el equipo se siente cómodo manejando más trabajo a la vez o si se resuelven cuellos de botella, los límites WIP pueden aumentarse. Por otro lado, si el equipo experimenta problemas con la carga de trabajo o el flujo, puede ser útil reducir los límites WIP para mejorar la gestión del proceso.

En resumen, los Límites de Trabajo en Progreso (WIP) son una parte fundamental de la metodología Kanban, y su uso adecuado puede ayudar a los equipos a mantener un flujo de trabajo equilibrado, mejorar la eficiencia y garantizar entregas de mayor calidad en el desarrollo de software u otros procesos.

En Kanban se encuentran ciertos actores principales que son aquellos que permiten llevar adelante la metodología, los mismos son los siguientes:

- *Equipo de Desarrollo:* Son los profesionales encargados de realizar el trabajo real de desarrollo del software. Son responsables de actualizar el tablero Kanban, mover las tarjetas y asegurarse de que el trabajo fluya de manera eficiente.
- *Scrum Master o Líder de Proyecto:* El Scrum Master (en el contexto de Scrum) o el Líder de Proyecto (en otros contextos) puede actuar como facilitador y defensor del equipo, asegurando que se sigan los principios y prácticas de Kanban.
- *Product Owner:* El Product Owner (en el contexto de Scrum) u otra figura similar es responsable de gestionar la lista de tareas y priorizar el trabajo del equipo para maximizar el valor entregado al cliente.
- *Stakeholders:* Los interesados en el proyecto, incluidos los clientes, patrocinadores y otros actores relevantes, pueden estar involucrados en el proceso de Kanban para proporcionar feedback y aportar sus perspectivas.

Como conclusión podríamos decir que la teoría de Kanban, como parte de las metodologías ágiles, es una herramienta valiosa para gestionar y mejorar los flujos de trabajo en una empresa de desarrollo de software. Su origen en la filosofía Lean Manufacturing y su evolución hacia el desarrollo de software demuestran su versatilidad y efectividad en diversos contextos. Al utilizar un tablero Kanban, tarjetas Kanban y límites de trabajo en progreso, junto con un enfoque en el flujo continuo, Kanban permite a los equipos mejorar la eficiencia, la calidad y la adaptabilidad en su trabajo, lo que resulta en entregas más rápidas y de mayor valor para el cliente.

El uso de Kanban en el desarrollo de software ofrece varias ventajas, entre ellas:

Mejora la distribución de las tareas: Kanban permite una mejor distribución de las tareas entre los miembros del equipo, lo que ayuda a evitar la sobrecarga de trabajo y a mantener un flujo de trabajo equilibrado.

Agilidad: Kanban es una metodología ágil que permite a los equipos adaptarse rápidamente a los cambios y ajustar su trabajo en consecuencia. Esto permite a los equipos responder de manera más eficiente a las necesidades del proyecto.

Organización de la carga laboral: Kanban ayuda a los equipos a organizar su carga laboral y a priorizar las tareas según su importancia y urgencia. Esto permite a los equipos enfocarse en las tareas más importantes y asegurarse de que se les dé la atención adecuada.

Evita el exceso de trabajo: Kanban utiliza límites WIP para limitar la cantidad de tareas que se pueden trabajar simultáneamente. Esto evita la sobrecarga de trabajo y ayuda a mantener un flujo de trabajo equilibrado.

Aumenta la calidad del trabajo: Kanban permite a los equipos enfocarse en completar las tareas actuales antes de comenzar nuevas, lo que mejora la calidad del trabajo. Al limitar el trabajo en curso, los equipos pueden enfocarse en completar las tareas actuales antes de comenzar nuevas, lo que mejora la calidad del trabajo.

En resumen, el uso de Kanban en el desarrollo de software ofrece varias ventajas, como la mejora de la distribución de tareas, la agilidad, la organización de la carga laboral, la reducción del exceso de trabajo y el aumento de la calidad del trabajo. Estas ventajas permiten a los equipos de desarrollo de software mejorar la eficiencia, la productividad y la transparencia en su trabajo.

4.1.3 Lean Software Development

Lean, dentro del contexto de una empresa de desarrollo de software, es una filosofía y metodología que se inspira en el sistema de producción de Toyota, conocido como "Lean Manufacturing". Su objetivo principal es eliminar desperdicios, mejorar la eficiencia y maximizar el valor entregado al cliente. El enfoque Lean se ha adaptado al desarrollo de software como "Lean Software Development" para abordar los desafíos específicos de esta industria.

El término "Lean" se originó en la década de 1950 cuando Taiichi Ohno y Shigeo Shingo desarrollaron el Sistema de Producción Toyota, también conocido como Toyota Production System (TPS). Este sistema se convirtió en la base de Lean Manufacturing, una metodología ampliamente adoptada en la industria manufacturera para mejorar la eficiencia y reducir el desperdicio. El TPS se enfoca en eliminar cualquier actividad que no agrega valor al cliente, incluyendo sobreproducción, tiempos de espera, movimientos innecesarios y defectos en la producción. Estos principios de eficiencia y mejora continua fueron posteriormente aplicados al desarrollo de software, dando lugar a Lean Software Development.

En el ámbito del desarrollo de software, Lean se aplica para eliminar actividades que no contribuyen al valor del cliente y mejorar la eficiencia en la entrega de software funcional. En lugar de seguir un enfoque tradicional en cascada, donde cada fase del proyecto se completa antes de pasar a la siguiente, Lean Software Development adopta un enfoque iterativo e incremental.

Uno de los conceptos clave de Lean Software Development es la eliminación de desperdicios. Los desperdicios en el desarrollo de software pueden incluir el retrabajo debido a errores, la sobreproducción de características innecesarias, los tiempos de espera debido a dependencias o la falta de comunicación efectiva entre el equipo y los stakeholders. Al identificar y eliminar estos desperdicios, el equipo puede centrarse en entregar funcionalidades relevantes y valiosas de manera más eficiente.

La entrega continua de valor es otro principio fundamental en Lean Software Development. En lugar de esperar a tener un producto completamente terminado para entregarlo al cliente, el desarrollo se divide en incrementos pequeños y entregables. Cada iteración produce un resultado funcional, lo que permite obtener retroalimentación temprana y frecuente del cliente. Esta retroalimentación temprana facilita la adaptación a cambios en los requisitos y prioridades, lo que a su vez permite entregar un producto final más alineado con las necesidades reales del cliente.

Respeto por las personas es otro aspecto clave de Lean en el desarrollo de software. Se valora el talento y la experiencia de los miembros del equipo, y se alienta la colaboración y el trabajo en equipo. Los miembros del equipo son empoderados para tomar decisiones y mejorar el proceso de desarrollo. El enfoque en el desarrollo de las personas contribuye a un ambiente de trabajo más motivador y productivo.

La reducción del tiempo de ciclo es un objetivo importante en Lean Software Development. Al disminuir el tiempo que toma completar una tarea o funcionalidad, el equipo puede responder rápidamente a las necesidades del cliente y mejorar la velocidad y la eficiencia en la entrega. Esto también permite una mayor adaptabilidad y agilidad frente a cambios y nuevos requerimientos.

La mejora continua es un principio esencial en Lean. Se alienta al equipo a reflexionar sobre su trabajo, identificar oportunidades de mejora y aplicar soluciones para optimizar el proceso de desarrollo. La mejora continua ayuda a la empresa a adaptarse a los cambios y a mantenerse competitiva en un mercado en constante evolución.

La limitación del trabajo en progreso (WIP) es otra práctica que promueve Lean Software Development. Limitar la cantidad de trabajo en progreso evita la acumulación de tareas inconclusas y ayuda a mantener un flujo de trabajo constante y equilibrado. Esto también permite al equipo enfocarse en finalizar tareas antes de iniciar nuevas, evitando el desperdicio de tiempo y recursos.

La visualización del flujo de trabajo es una práctica común en Lean. Los equipos utilizan tableros visuales para representar el flujo de trabajo y el estado de las tareas. Esto permite una mejor visualización de las actividades en curso, lo que facilita la identificación de cuellos de botella y la toma de decisiones informadas para optimizar el proceso.

En resumen, Lean dentro de una empresa de desarrollo de software implica adoptar una filosofía de eliminación de desperdicios, entrega continua de valor y mejora continua. Al centrarse en maximizar el valor entregado al cliente, reducir el tiempo de ciclo y fomentar la colaboración y la mejora continua, las empresas de desarrollo de software pueden aumentar su eficiencia y calidad, logrando así un mayor éxito en sus proyectos y una mayor satisfacción del cliente. La adaptación de los principios de Lean Manufacturing al desarrollo de software ha demostrado ser una estrategia efectiva para enfrentar los desafíos de esta industria y mejorar la eficiencia y la calidad en el proceso de desarrollo.

4.2. Mejora de Procesos

La teoría de mejora de procesos es una parte fundamental en la implementación de Scrum o cualquier otra metodología ágil. Esta teoría se basa en el modelo empírico de mejora continua de Scrum, conocido como el Ciclo de Deming o el ciclo PDCA (Planificar, Hacer, Verificar, Actuar). La idea detrás de este ciclo es establecer una base para la mejora continua a través de la experimentación, la inspección y la adaptación.

A continuación, se describe cómo aplicar la teoría de mejora de procesos para describir el proceso actual de una empresa de desarrollo de software que tiene como objetivo implementar Scrum:

- *Planificar (Plan)*: En esta etapa, la empresa debe establecer un plan detallado para la implementación de Scrum. Esto implica definir los objetivos específicos que se desean lograr con la adopción de Scrum, identificar los roles y responsabilidades del equipo Scrum, así como establecer el alcance y la duración de los sprints iniciales. También es importante crear un plan de capacitación para el equipo y las partes interesadas, para asegurarse de que todos comprendan los principios y prácticas de Scrum.
- *Hacer (Do)*: Una vez que se ha establecido el plan, se pasa a la implementación de Scrum en la empresa de desarrollo de software. En esta etapa, el equipo Scrum comienza a trabajar en la ejecución del proceso de desarrollo según lo definido en el marco Scrum. Esto incluye la

realización de las ceremonias de Scrum, como las reuniones diarias de Scrum, la planificación del sprint, la revisión del sprint y la retrospectiva del sprint. Además, se deben implementar las prácticas de Scrum, como mantener un tablero Kanban para visualizar el trabajo y realizar entregas incrementales al final de cada sprint.

- *Verificar (Check)*: En esta etapa, el equipo y las partes interesadas deben realizar una revisión exhaustiva del proceso de desarrollo bajo Scrum. Se deben analizar los resultados obtenidos, la productividad del equipo, la satisfacción del cliente y otros indicadores clave de rendimiento. Esta inspección ayuda a identificar qué aspectos están funcionando bien y cuáles requieren mejoras.
- *Actuar (Act)*: Con base en los hallazgos de la etapa de verificación, se deben tomar acciones correctivas y medidas para mejorar el proceso de desarrollo bajo Scrum. Esto puede incluir ajustar los roles y responsabilidades, revisar la duración de los sprints, optimizar las prácticas de Scrum y brindar más capacitación al equipo o a las partes interesadas. Además, la empresa debe fomentar una cultura de mejora continua en la que el equipo Scrum esté siempre buscando formas de optimizar su trabajo y aprender de sus experiencias.

Una vez que se han tomado las acciones correctivas, se repite el ciclo PDCA para continuar mejorando el proceso de desarrollo de software bajo Scrum. La teoría de mejora de procesos es esencial para garantizar que Scrum se adapte adecuadamente a la empresa y genere resultados positivos y sostenibles en el tiempo.

4.3. Relación entre las metodologías expuestas y sus beneficios en procesos

Las teorías de Scrum y Kanban son enfoques ágiles altamente exitosos y ampliamente adoptados para la mejora de procesos en empresas de desarrollo de software. Estas metodologías se han convertido en pilares fundamentales en la industria tecnológica debido a sus enfoques flexibles, centrados en el cliente y orientados hacia la mejora continua. A lo largo de los años, han demostrado su eficacia al permitir que las organizaciones se adapten de manera eficiente a los desafíos cambiantes del desarrollo de software y mejoren la calidad de sus productos.

Scrum, una de las metodologías ágiles más populares, se basa en un enfoque iterativo e incremental para la gestión de proyectos. La metodología se enfoca en la entrega de valor incremental y en responder de manera ágil a los cambios en los requisitos del cliente. Un aspecto clave de Scrum es el uso de sprints, que son períodos de tiempo fijos y cortos en los que el equipo de desarrollo se

compromete a entregar un conjunto de funcionalidades específicas. Al final de cada sprint, se presenta al cliente un incremento del producto en funcionamiento, lo que le permite visualizar y experimentar rápidamente los resultados tangibles del trabajo del equipo.

La flexibilidad de Scrum se deriva de su naturaleza adaptativa. A medida que surgen nuevos requisitos o cambian las prioridades, el equipo puede ajustar el plan y las funcionalidades a entregar en el próximo sprint. Esto garantiza una mayor alineación con las necesidades cambiantes del cliente y del mercado, permitiendo que el producto final sea más relevante y útil.

Además de la entrega incremental y la adaptabilidad, la mejora continua es un pilar fundamental de Scrum. Después de cada sprint, el equipo realiza una retrospectiva para reflexionar sobre su desempeño, identificar áreas de mejora y establecer acciones para optimizar su trabajo. Esta cultura de mejora constante permite al equipo aprender de sus éxitos y fracasos, ajustar su enfoque y mantener un alto nivel de eficiencia y productividad.

Por otro lado, Kanban es otra metodología ágil que ha ganado popularidad en el desarrollo de software. Kanban se centra en la visualización del flujo de trabajo y la mejora continua. A través del uso de tableros visuales, el equipo puede tener una representación clara y detallada del estado actual de las tareas y los procesos en curso. Cada tarea se representa con una tarjeta en el tablero, y las columnas indican las diferentes etapas del proceso, desde "Por hacer" hasta "Hecho". Este enfoque visual facilita la identificación de cuellos de botella, excesos de trabajo y áreas donde se pueden mejorar los tiempos de entrega.

La flexibilidad inherente a Kanban se traduce en la capacidad de ajustar la cantidad de trabajo en progreso (Work In Progress, WIP) según la capacidad del equipo y la demanda de las tareas. Al limitar la cantidad de tareas en cada etapa del proceso, Kanban ayuda a evitar la sobrecarga y el agotamiento del equipo, lo que a su vez conduce a una mayor eficiencia y calidad.

La colaboración y comunicación efectiva son fundamentales en Kanban, al igual que en Scrum. Al visualizar el flujo de trabajo, el equipo tiene una comprensión común de las prioridades y el estado de las tareas, lo que mejora la coordinación y permite una toma de decisiones más informada. La comunicación frecuente con los stakeholders y el cliente asegura que todos estén alineados con los objetivos del proyecto y que se satisfagan las necesidades del cliente.

Un aspecto destacado de Kanban es su capacidad para controlar los riesgos de manera efectiva. La identificación temprana de cuellos de botella y problemas en el flujo de trabajo permite una respuesta rápida para resolverlos antes de que se conviertan en obstáculos significativos. Al

mantener un flujo de trabajo constante y equilibrado, Kanban evita la acumulación de trabajo y asegura una entrega estable y predecible.

Finalmente, tanto Scrum como Kanban enfatizan la importancia de la calidad del software. Scrum incorpora prácticas como la revisión continua del código y las pruebas automatizadas para garantizar la calidad en cada entrega. Kanban, por su parte, busca eliminar los desperdicios y los errores en el proceso, lo que también contribuye a la entrega de un producto más confiable y satisfactorio para el cliente.

En conclusión, las teorías de Scrum y Kanban son ampliamente consideradas como exitosas en la mejora de procesos en empresas de desarrollo de software debido a su enfoque ágil, flexible y centrado en el cliente. Estas metodologías ofrecen beneficios significativos, como la entrega incremental de valor, la adaptabilidad a cambios, la mejora continua, la visualización del flujo de trabajo, la colaboración y comunicación efectiva, el control de riesgos y el enfoque en la calidad del producto. Al implementar Scrum y Kanban de manera efectiva, las empresas pueden mejorar su eficiencia, calidad y satisfacción del cliente, lo que los convierte en valiosos enfoques para la mejora de procesos en el campo del desarrollo de software.

5. Metodología

Como se comentó anteriormente, es necesario comenzar por la definición de los procesos principales de la compañía, su estandarización y la creación de documentación pertinente, conduciendo una ineludible mejora en la eficiencia en las operaciones. Pero debemos conocer bien la empresa, antes de poder implementar cualquier cambio será necesario relevar la situación inicial, siendo crítico para una adecuada ejecución.

A continuación, se exponen algunas razones por las cuales es esencial revisar el proceso antes de implementarlo:

1. *Entender el estado actual*: antes de embarcarse en la implementación de Scrum o cualquier metodología ágil, es fundamental tener una comprensión clara y completa del proceso actual de la empresa. Esto implica mapear todos los pasos, actividades y flujos de trabajo que actualmente se utilizan para desarrollar proyectos de software. Al entender el estado actual, es posible identificar las fortalezas y debilidades del mismo.

2. *Identificar ineficiencias y desafíos*: al revisar el proceso actual, es probable que se identifiquen ineficiencias, cuellos de botella y desafíos en la gestión de proyectos. Estos problemas pueden incluir retrasos en la entrega, falta de visibilidad, dificultades en la comunicación y problemas en la calidad del producto. Identificar estos puntos problemáticos es esencial para abordarlas de manera efectiva durante la implementación de Scrum.

3. *Alineación con los principios de scrum*: scrum se basa en una serie de principios ágiles que promueven la colaboración, la adaptabilidad y la entrega de valor continuo. Antes de implementar Scrum, es importante evaluar si el proceso actual de la empresa se alinea con estos principios. Si existen discrepancias significativas, será necesario realizar ajustes para garantizar que Scrum pueda aplicarse de manera efectiva.

4. *Determinar roles y responsabilidades*: scrum introduce roles específicos, como el Product Owner, el Scrum Master y el Equipo de Desarrollo. Revisar el proceso actual permite identificar cuáles son los roles y las responsabilidades existentes en la organización y cómo se pueden adaptar con los roles de Scrum. Esto es crucial para evitar confusiones y superposiciones de responsabilidades durante la transición.

5. *Establecer métricas y objetivos claros*: una vez que se comprende el proceso actual, es posible establecer métricas y objetivos claros para medir el éxito de la implementación de Scrum. Esto implica definir indicadores clave de rendimiento (KPI) que reflejen la eficiencia, la calidad y la productividad del proceso. Sin una línea de base sólida proporcionada por la revisión del proceso actual, sería difícil evaluar el impacto de Scrum.

6. *Planificación estratégica*: la revisión del proceso actual también desempeña un papel fundamental en la planificación estratégica. Permite a la empresa definir su visión y sus objetivos para

la implementación de Scrum. Además, ayuda a identificar los recursos necesarios, los riesgos potenciales y los pasos específicos que deben seguirse para llevar a cabo la transición de manera exitosa.

7. *Alineación con las necesidades del cliente*: scrum pone un fuerte énfasis en la satisfacción del cliente y la entrega de valor continuo. Revisar el proceso actual permite evaluar cómo se están satisfaciendo las necesidades del cliente en la situación actual y cómo Scrum puede mejorar este aspecto. Esto incluye la identificación de posibles oportunidades para involucrar más al cliente en el proceso de desarrollo.

8. *Comunicación y compromiso*: la revisión del proceso brinda una oportunidad importante para involucrar a todos los miembros del equipo y partes interesadas en el proceso de cambio. Fomenta la comunicación abierta y la comprensión compartida de los objetivos y beneficios de la implementación de Scrum. Esto es esencial para garantizar el compromiso de todos los involucrados en el proceso.

A raíz de lo expuesto previamente, se procede a efectuar el análisis del estado actual, de la siguiente manera:

5.1. Documentar el proceso actual

La optimización del flujo del proceso de desarrollo es fundamental para asegurar la eficiencia y efectividad en la entrega de productos y servicios. En este contexto, se ha llevado a cabo un exhaustivo análisis del flujo actual del proceso, con el objetivo de identificar áreas de mejora y proponer soluciones concretas.

Para comenzar, se realizó un detallado relevamiento con cada uno de los participantes del proceso, lo cual permitió recopilar información valiosa sobre las etapas, responsabilidades y puntos críticos del mismo. Esta información se documentó en un diagrama de flujo utilizando la herramienta Bizagi, proporcionando así una representación visual clara y concisa del proceso actual.

Uno de los hallazgos más significativos de este análisis es la naturaleza mayormente lineal del proceso, que involucra varias áreas, pero carece de etapas de revisión entre estas para garantizar un flujo efectivo de los requerimientos. Esta falta de revisión ha generado problemas de comunicación y desacuerdos, especialmente debido a la entrega de requerimientos mal definidos o no alineados con las expectativas.

Además, se observa que la gestión del backlog no sigue las mejores prácticas, ya que no se realiza una planificación periódica ni estructurada del mismo. Esto ha resultado en una asignación poco eficiente de los requerimientos, especialmente aquellos provenientes de la alta dirección.

Otro aspecto importante para destacar es la falta de un período definido para cada versión del software, lo que dificulta la planificación y previsibilidad del proceso de desarrollo. En lugar de seguir una metodología como Scrum (que utiliza sprints con una duración específica), las versiones se liberan según la necesidad, lo que plantea desafíos adicionales en la gestión del proyecto.

Finalmente, se identifica la ausencia de ciertas reuniones que podrían mejorar la eficiencia del equipo, como las retrospectivas. Estas reuniones son fundamentales para evaluar el desempeño del equipo, mejorar la comunicación entre las áreas y identificar oportunidades de mejora.

Diagrama de Flujo (Situación inicial):

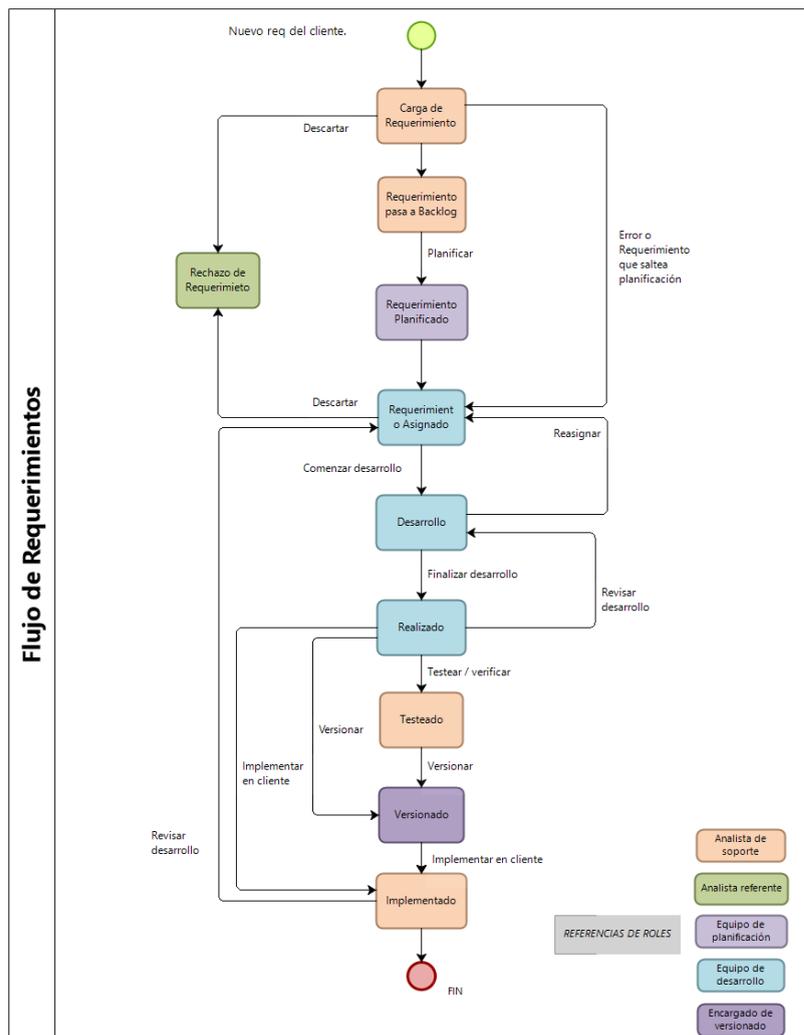


Ilustración 8 - Proceso de desarrollo - Situación inicial

5.2. Relevamiento de la estructura

La revisión de la estructura de una empresa es una parte fundamental en la planificación y ejecución exitosa de un cambio de proceso, principalmente en la implementación de Scrum en la gestión de proyectos de desarrollo de software. Antes de embarcarse en este tipo de transformación, es necesario entender y evaluar la estructura organizativa existente, ya que esta tiene un impacto significativo en cómo se adoptará y se adaptará Scrum en la organización. En este contexto, es esencial abordar esta consideración debido a los siguientes aspectos claves:

- La implementación de Scrum no se limita únicamente al equipo de desarrollo de software. Afecta a toda la organización, desde la alta dirección hasta los equipos operativos. Por lo tanto, es esencial revisar la estructura de la empresa para determinar cómo se alinearán y colaborarán las diferentes áreas funcionales y niveles jerárquicos para respaldar la adopción de Scrum de manera efectiva.
- Scrum introduce roles específicos, como el Product Owner, el Scrum Master y el Equipo de Desarrollo. Cada uno de estos roles tiene responsabilidades bien definidas. Revisar la estructura de la empresa implica determinar quiénes asumirán estos roles y cómo se relacionarán con otras partes de la organización. Esto puede requerir cambios en las descripciones de trabajo y en la asignación de responsabilidades.
- Scrum fomenta la comunicación y colaboración continua entre los miembros del equipo y con las partes interesadas. La estructura organizativa actual puede tener barreras a la comunicación que deben identificarse y eliminarse. Es posible que se necesiten ajustes en la forma en que los equipos se comunican y colaboran en comparación con las prácticas anteriores.
- La jerarquía y la toma de decisiones en la empresa pueden influir en la agilidad de la organización. Scrum promueve la toma de decisiones descentralizada y la autoridad de los equipos autoorganizados. Esto puede entrar en conflicto con una estructura organizativa jerárquica y centralizada. Revisar la estructura implica determinar cómo se delegan las decisiones y cómo se ajustará la jerarquía para facilitar la agilidad.
- La implementación de Scrum puede requerir habilidades y competencias específicas que quizás no estén presentes en la organización en este momento. Revisar la estructura organizativa también significa evaluar las necesidades de capacitación y desarrollo de habilidades, así como las estrategias de reclutamiento y retención de talento necesario para el éxito de Scrum.
- La cultura de la empresa es un factor determinante en la implementación de Scrum. Si la cultura actual es resistente al cambio, la revisión de la estructura puede ayudar a identificar áreas donde se necesita un cambio cultural para alinear la organización con los valores y

principios de Scrum. Esto puede incluir la promoción de la transparencia, la inspección y la adaptación en todos los niveles de la empresa.

- La implementación de Scrum puede variar en escalabilidad, desde pequeños equipos hasta grandes organizaciones. Revisar la estructura implica evaluar cómo se escalan los procesos de Scrum en la empresa y cómo se gestionan múltiples equipos y proyectos concurrentes.
- La revisión de la estructura organizativa puede ayudar a identificar posibles obstáculos antes de que surjan problemas. Al identificar y abordar los obstáculos potenciales de antemano, se pueden tomar medidas proactivas para mitigarlos y garantizar que la implementación de Scrum se desarrolle sin contratiempos.

Por las consideraciones previas expuestas, se solicita el organigrama actual de la empresa, el cual se presenta a continuación:

Thinksoft Argentina

ORGANIGRAMA GENERAL

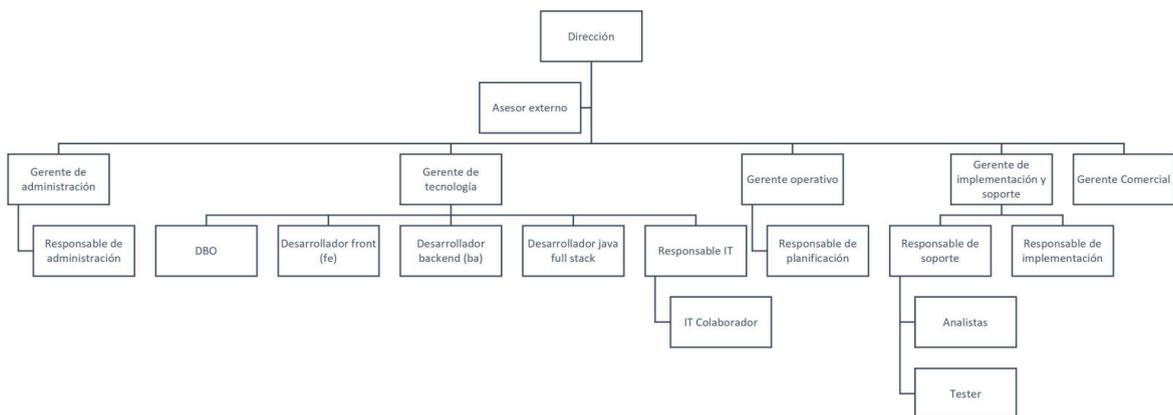


Ilustración 9 - Organigrama de la empresa

Tras analizar el organigrama, se identifica que sigue una estructura jerárquica o piramidal, lo que implica una clara jerarquía y líneas de autoridad definidas. Este diagrama facilita la toma de decisiones, la asignación de responsabilidades y la comunicación dentro de la organización. Sin embargo, en un entorno empresarial que enfrenta constantes cambios inherentes a su sector, esta rigidez puede limitar la capacidad de respuesta y adaptación oportuna de la empresa, generando una lentitud que dificulta la agilidad necesaria para enfrentar los requerimientos dinámicos que exige los clientes y la competencia.

5.3. Definición de nuevos roles

Luego del relevamiento de la estructura se comenzó a realizar ajustes en los roles de los involucrados en el proceso, asignándoles responsabilidades específicas acorde a un marco basado en la metodología Scrum. Estos roles fueron definidos considerando la estructura organizativa previamente establecida.

En primer lugar, se designó el rol de Scrum Master, cuya función principal radica en facilitar la comunicación entre las distintas áreas, abordando así el desafío de mejorar la interacción y eficiencia en el flujo de trabajo. Esta persona será responsable de convocar reuniones para resolver dudas o problemas relacionados con los requerimientos diarios, así como coordinar encuentros breves diarios para monitorear el progreso individual de los colaboradores y detectar posibles obstáculos.

Los encargados de desempeñar el rol de Scrum Master serán el gerente operativo y el responsable de planificación. Aunque la principal responsabilidad recae en el responsable de planificación, el gerente operativo actuará como sustituto cuando sea necesario, colaborando de manera conjunta para generar soluciones a los desafíos que surjan.

Asimismo, se definió el rol de Product Owner, compartido entre el gerente de Implementación y Soporte junto con el gerente de tecnología. Estos líderes serán responsables de identificar las funcionalidades técnicamente posibles y necesarias para satisfacer los requerimientos del cliente, en coordinación con el gerente de implementación y soporte, quien representa los intereses del cliente y prioriza las tareas del equipo de desarrollo.

El equipo de desarrollo, compuesto por analistas funcionales y desarrolladores, conserva sus responsabilidades anteriores, aunque se les asignan funciones adicionales para facilitar la implementación del nuevo proceso. Estas funciones se detallarán más adelante en la tesis, pero es importante destacar que dentro de este equipo, los analistas funcionales supervisan los requerimientos del cliente, mientras que los desarrolladores se encargan de la programación, y ambos colaboran para implementar las soluciones en los clientes.

Cada rol tiene responsabilidades claramente definidas, lo que proporciona claridad y eficiencia en la toma de decisiones y la ejecución del proyecto. Esto contrasta con la falta de definición de roles en la empresa previamente analizada, lo que puede generar confusión y superposición de responsabilidades, obstaculizando así la eficacia operativa.

El Gerente de Implementación y Soporte debe trabajar en estrecha colaboración con el cliente para identificar y documentar los requerimientos funcionales del producto. Estos requerimientos deben ser priorizados en un backlog de producto, asegurando que las características más valiosas se trabajen en primer lugar. Debe colaborar con el equipo de desarrollo para planificar los sprints, asegurándose de que el trabajo seleccionado para cada sprint esté alineado con las prioridades del negocio y los objetivos del proyecto.

El Gerente de Tecnología debe evaluar la viabilidad técnica de las funcionalidades propuestas. Debe considerar aspectos como la arquitectura, la tecnología existente y los recursos disponibles para garantizar que las funcionalidades sean factibles desde el punto de vista técnico, como así también deberá asistir a las revisiones de sprint para evaluar y validar el trabajo realizado por el equipo de desarrollo desde una perspectiva técnica y asegurarse de que cumple con los estándares de calidad y las expectativas técnicas.

La metodología Scrum, al enfatizar el papel del cliente a través del Product Owner, garantiza que el producto final esté alineado con las expectativas del cliente, lo que aumenta sus posibilidades de éxito en el mercado.

5.3.1. Team: Producto (Product Management)

Se define específicamente el equipo de Product Management es un equipo interdisciplinario y transversal a la empresa, que está representado por colaboradores idóneos en el negocio. Este equipo es el que planea y gestiona los productos de la empresa, es el que define y gestiona el Product Portfolio, Product Roadmap y Release Plan, estos tres instrumentos están contenidos en uno solo llamado Product Board. A los participantes de este equipo se los denomina Product Manager (PM), estos son los sponsors internos de los productos y sus lanzamientos.

Hay que aclarar que Product Manager es un rol y no un puesto dentro la empresa, e incluso en el caso de la empresa está formado por más de una persona. Este equipo tiene un Product Manager Lead, que en el caso de la empresa será el gerente de operaciones. El Product Manager Lead es un gerente de producto de alto nivel, el mismo se encarga de manejar la visión del producto y guiar al equipo.

Por otro lado, cada uno de los Product Managers que forman parte del equipo toman el rol de Delivery Manager, para hacer responsable explícitamente a una persona de la gestión y seguimiento de cada delivery de producto en producción. Cada uno de ellos trabaja en conjunto con consultores, sponsors externos, especialistas, expertos, colaboradores tecnológicos, etc, estos consultores pueden ser staff fijo o temporal, interno o externo.

El Product Manager es quien debe interactuar fluidamente con los Clientes (directores, gerentes, stakeholders importantes) y el equipo de Business Analysts. El Product Manager es quien cumple con la misión de sponsor, es el sponsor al que le responderán los equipos de Business Analyst y equipos de tecnología.

A continuación se puede visualizar de manera ilustrativa las tareas que realiza las personas que cumplen con el rol mencionado:



Ilustración 10 - Rol de Product Management

5.3.2. Team: Product Business (Equipo: Negocio de Producto)

Este equipo está integrado por los Business Analysts (BA), quienes se encargan de analizar y diseñar en profundidad las características de los productos. El BA es quien media entre los Product Managers y el Equipo de Desarrollo, es quien traduce las necesidades del negocio a un idioma más tecnológico y viceversa.

El BA, por encargo de los Products Managers, completa y detalla la información de una versión (Product Roadmap), es el que recopila los requerimientos, hace un análisis de los mismos y diseña la solución. Los diseños deben estar basados en User Centric Design (Diseño Centrado en el Usuario), bajo la validación de de UX/UI. Por eso el equipo de analistas debe tener asistencia y soporte de perfiles de UX/UI, QA y colaboradores tecnológicos. Es indispensable que los BA posean habilidades en UX/UI y QA.

El BA es quien tiene una interacción fluidamente con los Usuarios y el Equipo de Producto. El BA responde y asiste directamente a los Product Managers. Por otro lado, es el encargado de monitorear el producto: Al lanzar el producto debe realizarse un seguimiento para saber si tiene el impacto deseado, además de atender a los indicadores cualitativos de resultados, también se monitorea el desempeño y se vuelve a recopilar información de los usuarios.

El BA debe saber en qué contexto usan los productos los usuarios, él mismo debe usar los productos, en etapas de testing como en la vida real. Para entender cómo satisfacer mejor al usuario, tiene que vivir la misma experiencia de los usuarios reales. A continuación se observa la interacción entre que posee el rol:

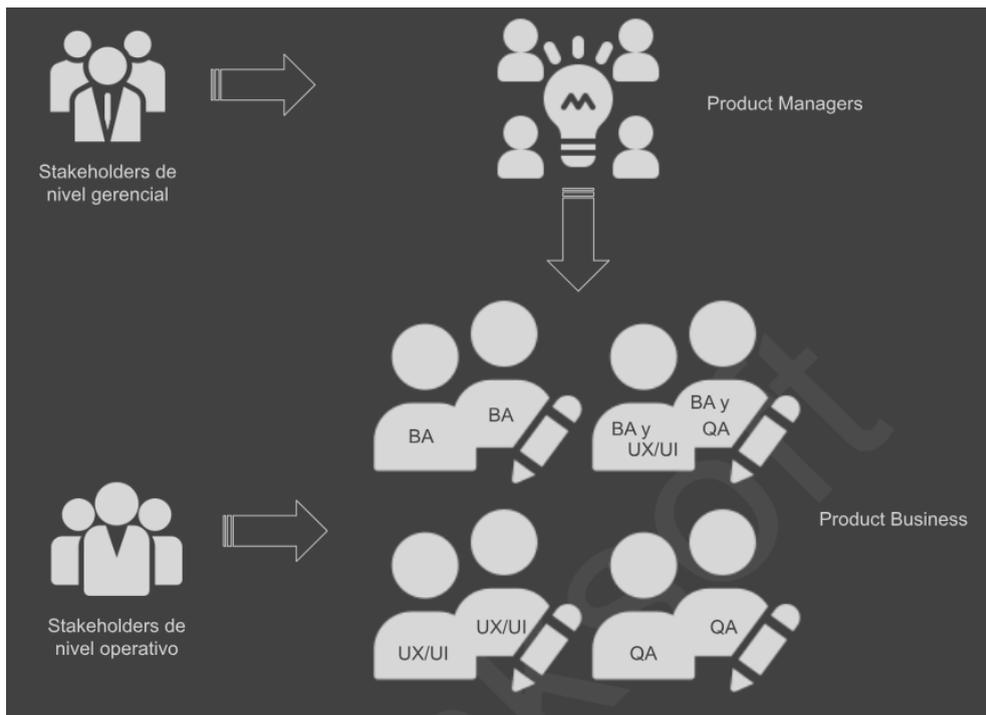


Ilustración 11 - Equipo de Product Business

Hay que aclarar que el proceso de análisis y diseño de cualquier iniciativa, problema o necesidad puede ser soportado por un un staff interdisciplinario, los encargados principales son los BA, pero estos suelen apoyarse en distintas personas de la empresa, como por ejemplo los desarrolladores. Generalmente suelen existir varias reuniones de prediseño y diseño en conjunto con los desarrolladores.

5.3.3. Rol: Scrum Product Owner

El Product Owner (PO) es el responsable principal de un Business Case, es al que le responderán los participantes del Equipos de Desarrollo, y es el responsable de hacer que un Business Case se construya en tiempo y forma.

El Product Owner (PO) es el dueño y principal responsable de la solicitud, seguimiento y validación de los entregables que satisfacen un Business Case y debe hacer una presentación formal del Business Case ante los colaboradores tecnológicos. Esta reunión se denomina “Refinamiento de Requisitos” que se detalla en los próximos apartados. Al menos se debe realizar una vez y repetirse todas las veces que sea necesario.

El PO es el asistente principal de los colaboradores tecnológicos (desarrolladores, IT, etc) y trabaja estrechamente con ellos. Él es el referente principal y tiene la responsabilidad de asegurarse

que los colaboradores tecnológicos entendieron los objetivos y requerimientos, es decir que entendieron el Business Case.

Si es necesario el PO puede hacer hasta un seguimiento diario de los requerimientos en construcción y tiene la capacidad de gestionar desviaciones de lo que se especificó en los requisitos, para esto se puede apoyar en los líderes de tecnología. Esta reunión de seguimiento se denomina Sprint Checkpoint y su objetivo principal es detectar de forma temprana cualquier desvío. También es quién responde ante el equipo de Product Manager.

Hay que aclarar que Product Owner es un rol y no un puesto dentro la empresa, este rol puede ser cumplido por cualquier Business Analytics que defina el equipo de Product Managers. Para el cumplimiento de este rol se definen diferentes personas según el producto del que se trate el requerimiento o el módulo de proyecto, ya que generalmente dentro de la empresa hay definido un referente para cada uno de los módulos.

5.3.4. Rol: Scrum Master

El Scrum Master es el “Encargado del Plan Táctico”, debe cumplir con la gestión del mismo y ayuda a los Product Managers a planificar y priorizar las Historias de Usuario e Incidentes.

El Scrum Master es el principal responsable de asegurarse el cumplimiento de los objetivos, de que los miembros del equipo cuenten con lo necesario para que puedan ejecutar sus tareas y de eliminar cualquier obstáculo que ponga en riesgo lo planificado.

Hay que aclarar que es un rol y no un puesto dentro la empresa, este rol puede ser cumplido por cualquier colaborador, tal como se comentó anteriormente actualmente este rol es ejercido por el responsable del área de planificación, aunque en su ausencia también estará a cargo del gerente de operaciones.

El Scrum Master es el dueño de Product Backlog y de los Sprint Backlog, también gestiona y modera las siguientes ceremonias que serán detalladas más adelante:

- Sprint Planning (Planificación del Sprint)
- Sprint Refinement (Refinamiento de requisitos)
- Sprint Daily (Scrum Diario)
- Sprint Review (Entrega de valor - Demo)
- Sprint start/end (Cierre/Comienzo de Sprint)
- Sprint Retrospective (Retrospectiva del sprint terminado)
- Sprint Checkpoint (en conjunto con el PO)

Hay que aclarar que nadie puede modificar el Product Backlog y los Sprint Backlog sin el consentimiento del Scrum Master. A continuación se presenta una ilustración que describe las principales tareas esperadas del mismo:

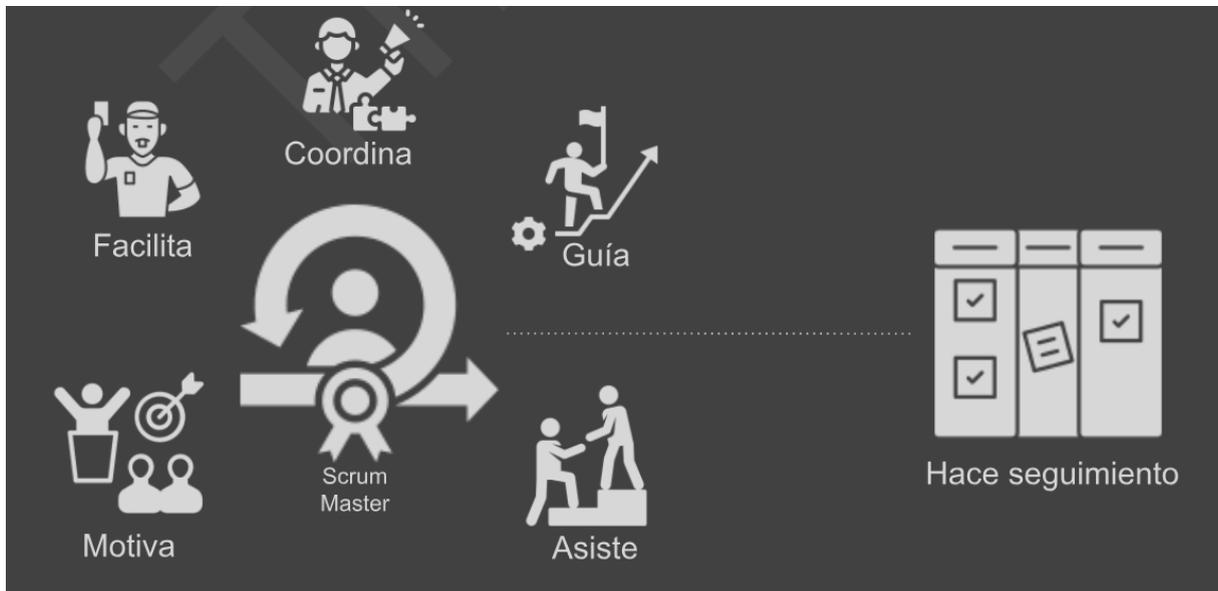


Ilustración 12 - Rol de Scrum Master

5.4. Tableros Kanban

Para lograr una optimización sobre la gestión del trabajo de cada uno de los participantes del equipo de desarrollo se utilizó Kanban, y en base a dicho marco conceptual se realizó un cambio en la gestión del trabajo.

En primer lugar, sobre la herramienta que utiliza la organización para gestionar los requerimientos se incorpora un nuevo plug-in denominado “Redmine Agile Plugin” que permite una visualización de tableros kanban para cada uno de los participantes del equipo de desarrollo, esta nueva herramienta permite poder diseñar los estados que debería tener cada una de las tareas asignadas a cada persona. Para ello se definieron los siguientes estados:

- Asignado
- Desarrollo
- Pausado
- Realizado

Sobre el estado Asignado se definió un WIP de cinco requerimientos, esto permite que no se le acumulen demasiados requerimientos para una determinada persona e ir controlando la carga de trabajo de cada uno de ellos, una vez que la persona toma el requerimiento debe pasar la tarjeta al estado “Desarrollo” para que el resto del equipo tome conocimiento que se está trabajando sobre el mismo. En el caso que se encuentre esperando alguna definición necesaria para poder continuar, se define el estado “Pausado”, que también puede utilizarse en el caso que se encuentre momentáneamente pausado porque se le ha solicitado que continúe con otro requerimiento. Una vez que haya finalizado con el trabajo, el estado “Realizado” indica que se finalizó con las tareas asociadas al mismo.

A continuación, se observa un ejemplo de cómo se visualiza el tablero de cualquier persona que se encuentra dentro del equipo de desarrollo:



Ilustración 13 - Tablero Kanban desarrollo

El Scrum Master y el Product Owner, encargados de supervisar el proceso general de las tareas, tienen la capacidad de llevar a cabo una revisión exhaustiva de todas las tarjetas de cada una de las personas ubicadas en los distintos estados mencionados. Esto se complementa con la inclusión de una columna adicional llamada "Planificado", la cual contiene aquellos requerimientos que han sido planificados dentro del sprint, pero aún no han sido asignados a ningún miembro del equipo de desarrollo. Estos requerimientos esperan a que se liberen recursos para poder asignarlos correspondientemente:



Ilustración 14 - Tablero Kanban Scrum Master y Product Owner

Además, con el fin de mejorar la eficiencia en el área y prevenir posibles inconvenientes o malentendidos, se llevó a cabo una especificación detallada de los datos mínimos que deben incluirse en las tarjetas al momento de su creación. A continuación, se detallan las definiciones que se comunicaron a todos los miembros del equipo, divididas según el tipo de requerimiento que se esté registrando:

- **Requerimiento de error:**

Al recibir un requerimiento de error, el primer paso consiste en intentar reproducirlo en un entorno de prueba, dado que la mayoría de los errores son reproducibles. Posteriormente, al cargar el requerimiento desde el sistema de seguimiento (Tracking) utilizado por el cliente al sistema interno de gestión Redmine, se deben incluir los siguientes elementos:

- Ambiente: el lugar donde se originó el caso o donde ocurrió el error, por ejemplo, en la base de producción del cliente.
- Especificación de la ruta de navegación: detallar el camino seguido para llegar a la pantalla donde se produce el error, incluyendo menús, submenús y opciones.
- Datos utilizados para la prueba: proporcionar información sobre los datos utilizados durante la reproducción del error, como detalles de pacientes, datos afiliatorios, o cualquier conjunto de datos necesario para recrear el problema.
- Versión en la que se debe realizar la corrección: especificar la versión del sistema en la que se debe implementar la solución, especialmente si se trata de un requerimiento bloqueante que afecta la operación normal del sistema.

- Capturas de pantalla y registro de errores (Log): adjuntar las capturas de pantalla que ilustran el error experimentado por el usuario, así como el registro de errores, que debe buscarse en el archivo correspondiente al proyecto. Este registro debe adjuntarse junto con las capturas de pantalla para brindar una visión completa del problema.
- ***Requerimiento de Funcionalidad Nueva o Cambio Funcional:***

Cuando se carga una solicitud de nueva funcionalidad o cambio solicitado, es necesario realizar una evaluación inicial para determinar si requiere algún cambio en el modelo existente. En caso afirmativo, se debe especificar con el mayor detalle posible la funcionalidad deseada y luego pasar el requerimiento para su planificación.

Una vez completada la etapa de planificación, y antes de que el requerimiento sea asignado al desarrollador correspondiente, es fundamental verificar que el script elaborado sea coherente con la funcionalidad solicitada. No se debe avanzar con el requerimiento sin realizar esta revisión.

Es importante dividir el requerimiento en varios elementos según sea necesario. Por ejemplo, si implica modificaciones en diferentes dominios, se debe crear un requerimiento por cada dominio, siguiendo un formato similar al utilizado para los requerimientos de error. Además, se debe evitar agrupar múltiples funcionalidades en un solo requerimiento. Si una solicitud abarca varias opciones o pantallas, estas deben ser separadas en requerimientos individuales.

Es esencial proporcionar una descripción detallada de cómo se espera que se realice la funcionalidad solicitada. Esto incluye especificar el menú, la pantalla, las opciones, el módulo y la funcionalidad asociada.

Sin una especificación detallada del requerimiento, el desarrollo del mismo no se iniciará, incluso si el script ya ha sido preparado y confirmado. No se debe crear un requerimiento que únicamente adjunte el script, sin especificar claramente los detalles funcionales del desarrollo.

Para concluir, podemos mencionar que a partir de la implementación de Kanban se pudo obtener una vista clara y en tiempo real del estado del trabajo en curso, que permite a todos los miembros del equipo comprender qué se está haciendo, quién está trabajando en qué, y cuánto trabajo queda por hacer. Por otro lado, permite identificar cuellos de botella y retrasos en el proceso de trabajo, que facilita la toma de decisiones informadas para eliminar obstáculos y mejorar la eficiencia.

Al limitar la cantidad de trabajo en progreso (WIP), Kanban ayuda a evitar la sobrecarga del equipo y el agotamiento y esto fue adoptado por la empresa al definir el WIP para cada persona que se encuentra incluida en el equipo de desarrollo, esto conduce a un flujo de trabajo más equilibrado y una mejor gestión de la capacidad.

En último lugar, la intención de realizar la implementación de Kanban se centra en eliminar el trabajo innecesario y el desperdicio, lo que conduce a una mayor eficiencia y ahorro de recursos. La

retroalimentación constante y la revisión regular del proceso de trabajo fomentan la mejora continua en la empresa.

5.5. Definición de un nuevo proceso

Luego de haber cumplido los pasos mencionados anteriormente se comenzó a formalizar el nuevo proceso, para realizar se tomó de base el marco conceptual mencionado en el punto 3 del presente trabajo.

A la hora de realizar la definición del proceso fue importante poder definir algunos conceptos claves relacionados a los productos que posee la empresa, para poder documentarlos y expresar claramente a cada uno de los colaboradores cuales son las características de los mismos.

En este apartado se enumeran los equipos que son definidos en el proceso, la especificación de roles que se mantienen de manera definitiva, la denominación de las partes del proceso y ceremonias que se deben realizar para poder cumplir con el proceso de manera adecuada. Cada uno de los puntos que se definen en este apartado fue documentado y compartido con cada uno de los colaboradores de la empresa para que cada uno de ellos esté en conocimiento de las características del nuevo procesos y de los puntos claves. Luego fue puesto a disposición en una carpeta compartida que tienen acceso todo el personal para que lo puedan consultar y revisar en el momento que lo deseen.

5.5.1. Artifact: Product Roadmap (Hoja de Ruta de Producto)

Se define el concepto de Product Roadmap que es el instrumento que se utilizará para detallar principalmente objetivos potenciales y futuros, de esta forma se visibiliza cómo es probable que evolucione el producto a lo largo de lanzamientos importantes. En el siguiente cuadro se puede ver claramente los puntos necesarios e importantes del Product Roadmap, tomando como ejemplo el definido en la empresa:

Producto Thinksoft Hospital			
Release	9.286	9.287	9.288
Goals	objetivos	objetivos	objetivos
Big Features	características	características	características
Result	S / I / N	S / I / N	S / I / N

La primera fila indica el nombre de la versión (release), para la nomenclatura que se utiliza dentro de la institución, la misma se encuentra dividida por tres niveles divididos por puntos. En el caso del ejemplo de Thinksoft Hospital un ejemplo es: “9.286.0”.

La segunda fila es la parte más importante, como sugiere su nombre, establece el objetivo de un lanzamiento. Los objetivos podrían ser un nuevo módulo, nueva tecnología, etc. Se establece que no más de tres objetivos por versión e identificados con un número, ej. 9.288.1.

La tercera fila enumera los Big Features (características) del producto que son necesarias para alcanzar el objetivo. En este punto se destaca lo que realmente importa; limitando a cinco características (big features) por versión, es importante que las características sean importantes e identificadas con un número, ej. 9.288.1-2.

La cuarta y última fila define que tan bueno es el resultado, primero definimos si el resultado es (2) Satisfactorio / (1) Insatisfactorio / (0) No Desarrollado. Si el resultado es Insatisfactorio, se debe analizar, investigar y definir propuestas de mejora para que no vuelva a suceder.

5.5.2. Artifact: Release Plan (Plan Estratégico de Lanzamientos)

Es el instrumento que contiene los lanzamientos de todos los productos, ubicados en trimestres por año. No hay límites para la cantidad de años planificados. En este instrumento se visualiza qué etapas deben realizarse para alcanzar el lanzamiento. Es muy importante identificar las etapas finales, identificando así en qué momento se da por terminado el lanzamiento.

Las etapas deben ser muy generales, en el que se identifican las fases de un proyecto. También es importante identificar los diferentes hitos, estos ayudan a la planificación de los puntos de control del plan.

5.5.3. Artifact: Business Case (Caso de Negocio)

Los detalles de los goals (objetivos) y big features (características principales) del Product Roadmap mencionados anteriormente, se dejan asentados en un documento denominado Business Case, en el que la complejidad, estructura, profundidad, diseño y organización de estos son a criterio de los Business Analysts.

Un Business Case especifica claramente funciones, características, el diseño del producto que permite usarlo, la experiencia del usuario, procesos que involucra y métricas de negocio. Todo esto en la medida de que sea necesario y según el caso.

En base a la idoneidad del caso se espera que los Business Analysts documenten los Business Case de un producto con:

- BPMN con sus actores, artefactos (estructuras de datos), KPIs.
- Análisis y diseño basado en UX/UI como bocetos, wireframes, prototipos, definición de usuarios, analytics de las pantallas, validaciones de inputs.
- Casos de prueba para aplicar los test de QA.

Las tareas y relaciones mencionadas se resumen en el siguiente gráfico:



Ilustración 15 - Proceso de un Business Case

5.5.4. Release

El Release está compuesto por trabajo planificado y por trabajo que viene por un camino de necesidades excepcionales, trabajo que si o si se agrega a la planificación por que el negocio así lo requiere.

Las nuevas funcionalidades o requerimiento funcional contempla el camino planeado, es algo que se diseñó y definió según un cronograma preestablecido, mientras que el trabajo no planificado se identifica como Incidente o Error estos son una dolencia en general de los clientes y usuarios, pueden ser bug, crash, performance, error de ortografía, etc. Dentro de lo no planificado también pueden ser trabajos en general, estos se denominan Solicitud General, estas no son para construcción de software, son pedidos genéricos, como por ejemplo instalar un aplicativo. También pueden existir pedidos de Nuevas Funcionalidades que no fueron planificadas, a estas excepciones se las denomina Adendas de Release.

Toda la documentación necesaria, análisis, adjuntos, links, historial de comentarios deben quedar en los incidentes o nuevas funcionalidades, que deberán cumplir con las condiciones que se definieron en el punto 5.4.

Existen diferentes tipificaciones que están asociadas a los incidentes o a las nuevas funcionalidades, estas ayudan mucho a identificar el scope del trabajo a realizar, si bien la tarea tiene objetivo, no todas las tareas son de la misma naturaleza. Se enumeran las diferentes tipificaciones:

- Feature
- Fix (sólo para Incidents, y se aplica en una versión futura)
- Hotfix (sólo para Incidents, y se aplica sobre la versión en PROD)
- Technical work (Trabajo técnico en general, It, DBA, etc)
- Knowledge acquisition (Investigación, Cursos, etc)
- Technical analysis (Análisis de código, apis, viabilidad, etc)
- Product delivery (environments)
- Testing
- Validation

Cabe aclarar que las Funcionalidades, Incidentes y Solicitud General pueden durar más de un Sprint, estas pueden moverse de un Sprint a otro si es que no se terminaron.

Un requerimiento al llegar al área de desarrollo entra en el kanban de cada uno de ellos pudiendo tener los siguientes estados: Asignado, Desarrollo, Pausado y Realizado. Una tarea puede ser bloqueada en casos que no se tenga bien claro que hacer, se tengan dudas sobre los objetivos o cómo resolverlos técnicamente, otro motivo simplemente puede ser que se frena la tarea por otra.

Business Case vs Nueva Funcionalidad:

Un Business Case es un documento centralizado de un diseño, explica los detalles de los requisitos y funcionalidades de un producto o parte de uno.

Cualquier tarjeta (ticket) de tipo Nueva Funcionalidad o Incidente son trabajos a realizar, son objetivos a cumplir, no son un documento, ni pretenden serlo, ni deben serlo, por ende apuntan o referencian al Business Case, ya que estos son la fuente centralizada de los requisitos y especificaciones del producto o parte de él.

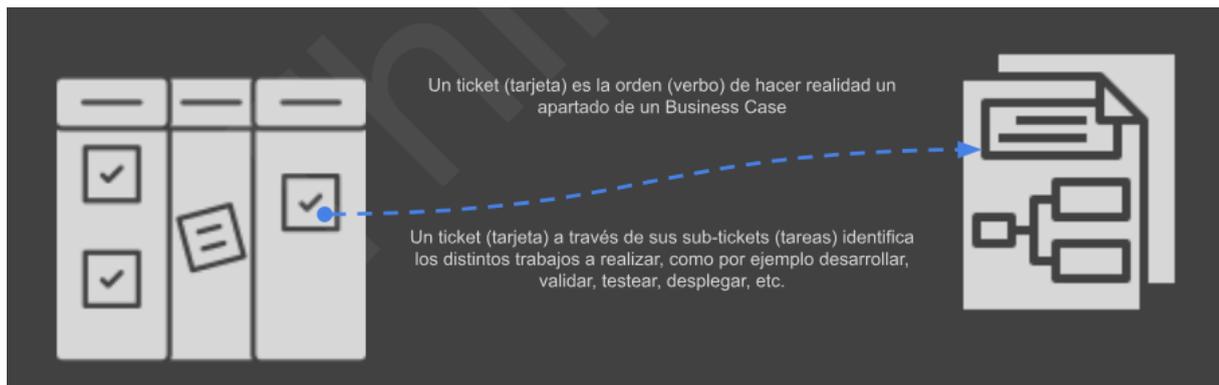


Ilustración 16 - Detalle de un ticket

Las Nuevas Funcionalidades no deberían contener documentación embebida, ya que estas no son un repositorio de documentación, para tal fin existen los Business Case, un requerimiento sirve para identificar qué apartado de un Business Case se debe crear, por lo que en la Story sólo se puede colocar referencias a la documentación, es decir el Business Case. Puede que en algunos casos una Nueva Funcionalidad apunte a todo el Business Case, es decir hay un mapeo 1 a 1 entre el requerimiento y el documento (Business Case).

Es importante aclarar que para el caso de Incidentes y Solicitud General, el ticket o tarjeta sí debe funcionar como documento, no tienen un Business Case asociado, la misma tarjeta debe tener toda la información necesaria, tanto como descripción, análisis, problema, imágenes, etc.

Una aclaración importante es decir que pueden existir otros tipo de tickets, los que sean necesarios, sin embargo lo importante es tener en cuenta que lo que llega a BACKLOG es lo que se considera que se va a hacer, que ya paso todos los procesos necesarios para determinar si se va a hacer o no.

El ticket que llegue a BACKLOG significa que se decidió que se va a realizar y se decidió su prioridad, esta está dada por su lugar en la pila, mientras más arriba más prioridad.

5.5.5. Refinamiento de Requisitos

El PO al menos debe hacer una presentación formal de los requerimientos ante los colaboradores tecnológicos, esta reunión tiene por objetivo transmitir y explicar a los colaboradores tecnológicos los requisitos que se les van a pedir en un futuro.

Es responsabilidad de los colaboradores tecnológicos dar feedback al PO, como por ejemplo viabilidad técnica, performance posible, dificultades, mejoras, soluciones superadoras, etc.

Este tipo de reunión es responsabilidad del PO, debe estar planificada, pactada y calendarizada por él, siempre en coordinación con el Scrum Master.

Participantes necesarios:

- Líderes de los equipos tecnológicos.
- Product Owner (PO).
- Colaboradores tecnológicos involucrados.
- Stakeholders

Estas reuniones deben realizarse continuamente, tanto en etapas tempranas del análisis y diseño de los Business Case, como al momento de que se planifican.

Si es importante tener refinado y lo suficientemente claro cualquier requisito que se encuentre en un Sprint por comenzar, un Sprint no puede comenzar con requerimientos no refinados y comprendidos por parte de los colaboradores tecnológicos.

5.5.6. Sprint Review

Toda solución que entregan los colaboradores tecnológicos deben hacerlo formalmente, mediante una reunión pactada (con el PO y Scrum Master) y calendarizada. Tiene por objetivo la comunicación y presentación al PO del resultado logrado, mediante una demostración, entrega de documentación y cualquier otro entregable asociado al trabajo.

Es función como parte del PO la validación del resultado, con dos posibles decisiones, ACEPTACIÓN o RECHAZO, estos pueden ser parciales, debiendo quedar perfectamente aclarado y detallado qué se RECHAZA y qué se ACEPTA.

El resultado de la ceremonia es un informe de los motivos de rechazo (en la tarea) y la calendarización de posibles reuniones extras, como para Capacitaciones, Pruebas, Uso del Producto entregado, planificación de despliegue en el cliente, etc.

Si es un incidente, la confirmación de en qué versión del Producto entra. Esta decisión debe quedar registrada en el Incidente en coordinación con los Product Managers. La versión se detalla al momento de crear el Incidente. Se espera que el PO tome todos los recaudos necesarios para validar la calidad del entregable, entre los cuales se recomienda por ejemplo “usar el Producto él mismo”. Si el equipo de Product Business cuenta con uno o más colaboradores de QA, la calidad se debe evaluar con estos colaboradores, en base a los casos de pruebas ya creados.

El equipo de tecnología está a disposición del PO para asistirlo en lo que necesite para validar el entregable.

Participantes necesarios:

- Líderes tecnológicos (Gerente de Desarrollo, Techlead, etc.).
- Product Owner (PO)
- Colaboradores tecnológicos involucrados
- Stakeholders

Esta ceremonia puede ser pactada al momento de terminar uno o más requerimientos, lo normal es que se realice al momento de terminar un Sprint, sin embargo queda a criterio del PO.

5.5.7. Comienzo/Cierre de Sprint

Esta ceremonia consiste en repasar el Sprint Backlog, identificando requerimientos terminados de los que no están terminados. Los PO son los responsables de dar este feedback. Los requerimientos que no se terminaron deben pasarse a otro sprint o dejarlos en espera para futuros sprints. Esto lo determinan los Product Managers.

Los requerimientos terminados (desarrollo) deben cerrarse y crear nuevos requerimientos con objetivos de despliegue en ambientes del cliente como TEST, PROD, etc. Estas consisten en nuevas tareas asociadas a las Funcionalidades e Incidentes.

Esta ceremonia es moderada, planificada y calendarizada por el rol de Scrum Master. De esta ceremonia se espera que se complete el Sprint Backlog del próximo Sprint y se dé inicio al mismo.

Es ideal que los Product Managers dispongan de un Sprint Backlog borrador como input de la ceremonia, para tomar requerimientos del borrador y sumarlos a los requerimientos no terminados del sprint que se cierra.

Es importante que el Scrum Master identifique todos los requerimientos agregados, estos son aquellos que no fueron planificados inicialmente para el sprint. El Scrum Master debe llevar una estadística de requerimientos agregados.

El Scrum Master identifica la cantidad de requerimientos completados del sprint y la cantidad sin completar, además debe llevar estadística por cada sprint, para ajustar las futuras planificaciones de Sprint Backlog.

Si es importante tener refinado y lo suficientemente claro cualquier requisito que se encuentre en el Sprint por comenzar, un Sprint no puede comenzar con requerimientos no refinados y comprendidos por parte de los colaboradores tecnológicos.

El Scrum Master debe prever un espacio de capacidad para los Incidentes, es decir si el equipo de desarrollo tiene una capacidad de 50 requerimientos por sprint, sólo se debe planificar 30, como para dar un ejemplo, dejando así un espacio de 20 para incidentes o agregados.

Participantes necesarios:

- Líderes tecnológicos (Gerente de Desarrollo, Techlead, etc.).
- Product Owners (PO).
- Product Managers.
- Otros colaboradores que el Scrum Manager invite.

Esta ceremonia se debe hacer el día en que termina el sprint y como requisito fundamental es tener pre-validado los requisitos (terminados) por parte de los PO, los mismos son responsables de realizar las Sprint Review previamente a la ceremonia de Cierre de Sprint.

A continuación, se puede observar una ilustración del proceso planteado, en el que se puede observar que se realiza un principal cambio del constante feedback que se realiza desde la finalización del sprint y la priorización del nuevo sprint. Es muy importante que se pueda realizar una retrospectiva sobre el trabajo realizado para evaluar las tareas realizadas, y verificando cuales son los puntos en los que se puede mejorar para el próximo sprint.

Por otro lado, se destaca la priorización de cada uno de los requerimientos que ingresan, junto con la planificación, que anteriormente no estaba claramente definida.

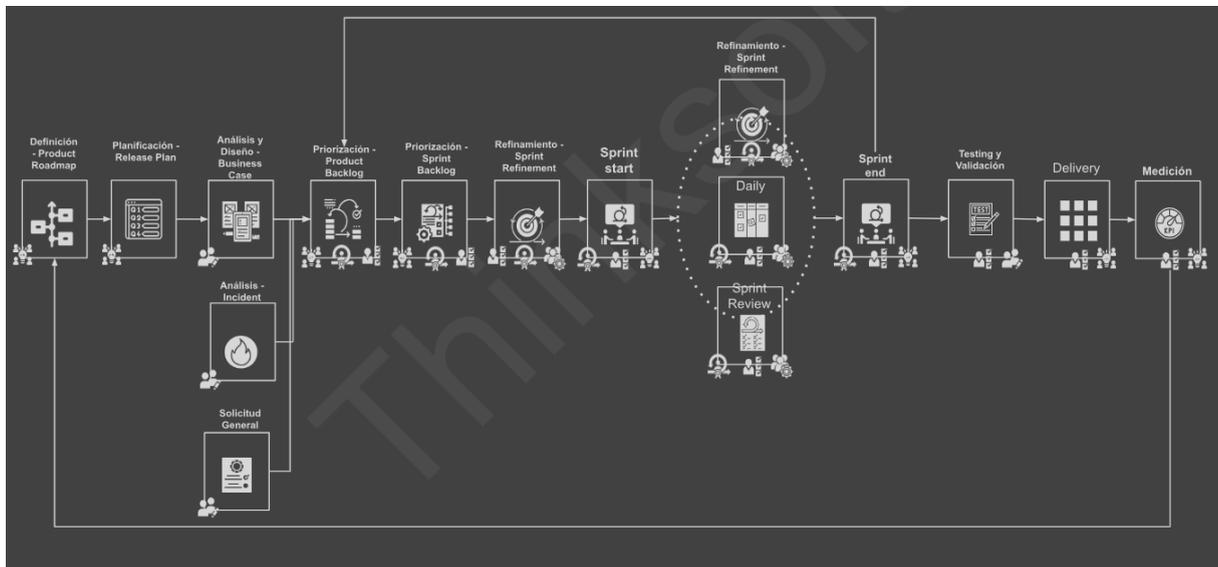


Ilustración 17 - Proceso completo

Por último, cabe destacar la relevancia de los nuevos roles que han sido definidos como parte de la reestructuración, los cuales desempeñan un papel fundamental en la consecución de los objetivos organizacionales. Estos roles han sido meticulosamente concebidos para facilitar una planificación efectiva de los requerimientos operativos, así como para mediar entre las áreas de Business Analysts y

desarrolladores, abordando así un punto crítico que previamente generaba conflictos y fricciones entre dichas áreas. Es crucial que cada rol identifique claramente las responsabilidades asignadas y se comprometa con su ejecución para asegurar el flujo adecuado de los procesos y la consecución de los objetivos establecidos.

La definición y asignación de estos nuevos roles no solo busca optimizar la coordinación y colaboración entre las diferentes áreas funcionales, sino que también apunta a mejorar la eficiencia operativa y reducir los tiempos de respuesta ante los desafíos y demandas del entorno empresarial. Además, estos roles están diseñados para fomentar una mayor integración entre los equipos de trabajo, promoviendo así un ambiente laboral colaborativo y orientado hacia el logro de resultados. En este sentido, se espera que la claridad en las funciones asignadas y la adecuada delimitación de responsabilidades contribuyan a fortalecer la sinergia organizacional y a potenciar el desempeño global de la empresa en el mercado competitivo actual.

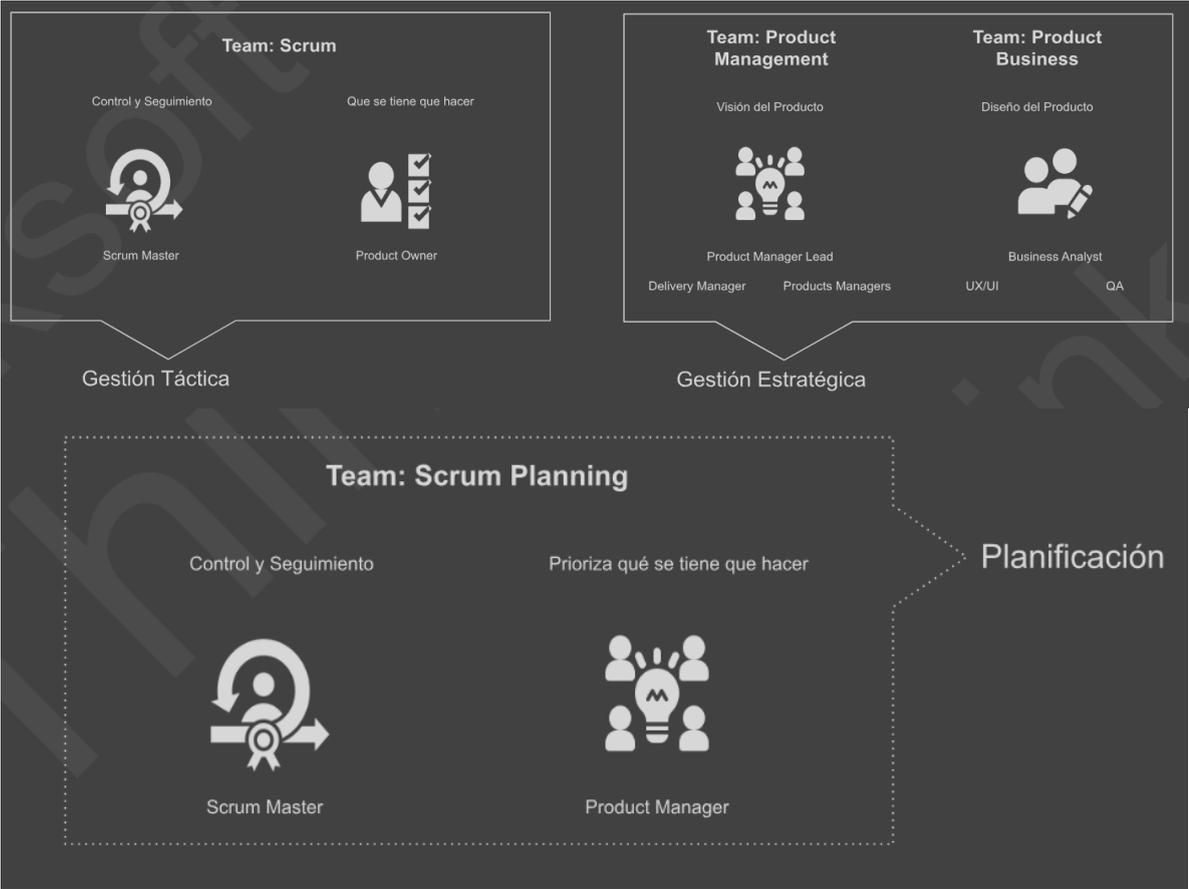


Ilustración 18 - Equipos que forman parte del nuevo proceso

Después de establecer los nuevos roles involucrados en el proceso, definir las nuevas ceremonias y abordar los diversos aspectos que se han integrado, procedemos a documentar el nuevo proceso delineado y a compartirlo en toda la institución para garantizar su comprensión generalizada. Esta información se encuentra alojada en una carpeta compartida accesible para todos los miembros de la organización en cualquier momento. Es relevante señalar que el tiempo de sprint está establecido en un período de dos semanas. Además, se destaca la inclusión de una tarea de medición de resultados y la importancia de la retrospectiva para identificar lecciones aprendidas y áreas de mejora específicas dentro del sprint.

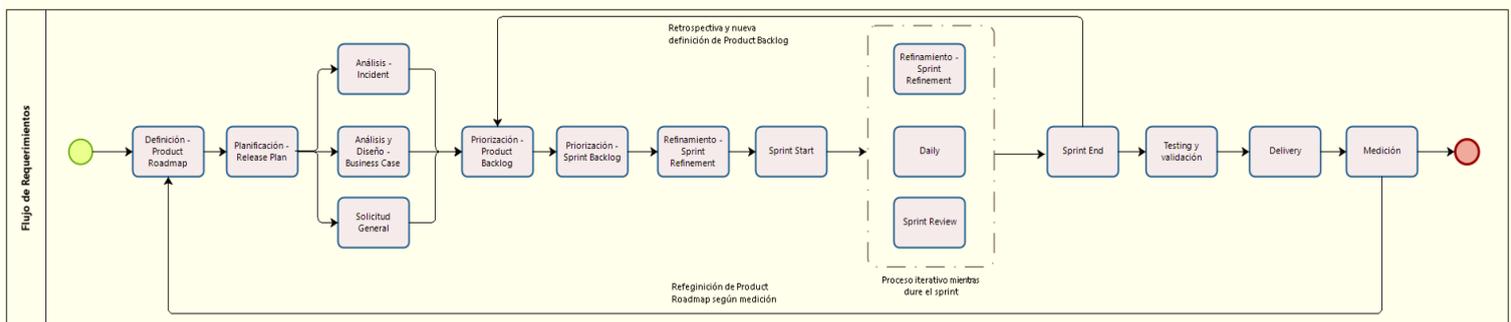


Ilustración 19 - Documentación del nuevo proceso

5.6. Indicadores de Medición

Con el objetivo de establecer indicadores que posibiliten la evaluación de la eficacia y eficiencia del proyecto, se procede a definir los siguientes indicadores:

- Requerimientos clasificados por estado y tipo.
- Cantidad de requerimientos por proyectos.
- Horas dedicadas por cliente.
- Comparación entre el tiempo de respuesta real y el tiempo de respuesta previsto.
- Frecuencia de los sprints.
- Planificación de requerimientos vs. requerimientos efectivamente ejecutados.

Es importante destacar que, si bien se reconoce la necesidad de ampliar la variedad de indicadores a lo largo del tiempo, se ha optado por comenzar con los mencionados previamente. A continuación, se describen detalladamente cada uno de estos indicadores..

5.6.1. Requerimientos clasificados por estado y tipo

Este indicador tiene por objetivo recopilar la cantidad de requerimientos que han sido cargados por los clientes en un período de tiempo. El tiempo de medición es mensual, y se evalúa sobre la cantidad de requerimientos que han sido cargados, un porcentaje de aquellos que han sido solicitados como errores diferenciados de aquellos que han sido solicitados como nuevas funcionalidades. En este aspecto, permite contar con el porcentaje de requerimientos de errores en relación a las nuevas funcionalidades o cambios funcionales. Un aumento en la cantidad de errores significa que las entregas del producto no son satisfactorias y que no cumplen con los mínimos de calidad necesarios para poder ser entregados al cliente.

Por otro lado, otro de los aspectos que se evalúa en este indicador es la cantidad de requerimientos que han sido finalizados en el período vs. la cantidad que aún se encuentran pendientes. Los plazos del sprint son de dos semanas y debido a que el plazo de medición es mensual se realiza la evaluación conjunta de dos sprints. En este sentido, el objetivo es PENDIENTES < 35%. Es decir que de la totalidad de los requerimientos que han sido cargados en el mes, al menos el 65% debe ser resuelto en dicho período. Se deja un margen de 35% sobre todo para aquellas funcionalidades que requieren un tiempo de desarrollo que sea mayor a dos sprints.

5.6.2. Cantidad de requerimientos por proyectos

El indicador de cantidad de requerimientos por proyectos permite medir la cantidad de requerimientos que ingresan clasificados por los proyectos que tiene la empresa (Hospital / Prepaga / ERP). Según la complejidad, la cantidad de clientes y la importancia de los proyectos antes mencionados se definió un determinado porcentaje máximo de requerimientos que pueden recibirse de clientes, que también es el mismo porcentaje que debe planificarse de requerimientos de cada uno de los proyectos sobre el total. El objetivo del indicador es el siguiente:

HOSPITAL < 80%

PREPAGA < 15 %

ERP < 5%

El período de medición es mensual, y se toman en cuenta la cantidad de requerimientos totales cargados por los clientes por un lado sobre eso se realiza un análisis para poder verificar que se cumpla con el objetivo. Por otro lado, se realiza un segundo análisis que se basa en los requerimientos planificados, en donde se toma la cantidad de requerimientos planificados en el período y se lo compara con el objetivo para verificar el cumplimiento del mismo.

En los casos en los que se produzca un desvío se analizan las causas que los producen, y en base a ello se toma alguna acción correctiva, o se dejan documentadas las causas que producen la desviación, y se determina el tiempo en el que se considera que debería estabilizarse. Esto último sucede puntualmente, con nuevas implementaciones que generalmente se produce un aumento de los requerimientos durante el tiempo que dura la implementación de un nuevo cliente.

5.6.3. Horas dedicadas por cliente

Mediante este indicador se mide la cantidad de horas que son dedicadas a cada uno de los clientes, en relación a un criterio específico que determina la cantidad de tiempo que se debe dedicar a cada uno de ellos.

Para ello se utilizó el importe de la facturación mensual emitida a cada cliente, y se estableció la relación de este importe con el total facturado por la empresa en el período. Esto da como resultado un porcentaje que indica cuál es el porcentaje máximo de horas que debe ser dedicada a cada cliente por mes.

El porcentaje de horas dedicadas, se toma como total de horas posibles de ser utilizadas, la cantidad de empleados de la empresa por la cantidad de horas laborables en el mes. Este total de horas será comprado con el total de horas dedicadas a un cliente específico y en base a ello se determina cuánto representa sobre el total. Como hora dedicada a cada cliente se computan las horas asociadas a requerimientos de Incidentes, Funcionalidades o cualquier hora dedicada en soporte presencial, reuniones o cualquier otra hora del personal demandada por el cliente.

En caso de que no se cumpla el objetivo con algún cliente, se procede a analizar la causa del desvío. Esto puede ser justificado de manera excepcional si, por ejemplo, durante un mes se dedicó un porcentaje mayor de horas debido a la implementación de un proyecto crítico o para resolver un inconveniente importante. También podría tratarse de un desvío que se prolonga durante dos meses o más, en cuyo caso se coordina con el área comercial para regularizar la situación. Esto podría implicar estabilizar las horas demandadas o bien aumentar la facturación para dicho cliente.

5.6.4. Tiempo de respuesta real vs. tiempo de respuesta previsto

En el contrato de soporte o implementación que se firma con el cliente, se establecen los tiempos de respuesta ante las diferentes peticiones. En este sentido, se identifican como incidentes que afecten la operatoria del cliente de manera directa (caída de un servidor, error en un proceso central) deberá tener respuesta en un plazo de 24 hs como máximo. Ante otro tipo de incidentes deberá tener

respuesta en un plazo de 24 a 72 hs. Mientras que para los casos de Funcionalidades se establece un tiempo de 5 días hábiles. Es necesario aclarar que este tiempo no implica el compromiso de plazo de solución, sino que implica que alguien de la empresa tomará el tema y lo gestionará.

Por lo tanto, se considera necesario realizar una medición sobre los tiempos de respuesta, debido a que ello afecta directamente la satisfacción del cliente sobre el soporte brindado, que es uno de los puntos que se necesita mejorar y es un factor importante que llevó a tomar la decisión de realizar este cambio de proceso.

La medición de este indicador es de manera mensual, y se toman en consideración todas las peticiones que han sido cargadas por el cliente, y de acuerdo al tipo de petición se compara con el tiempo que el Business Analyst dió una respuesta o generó un determinado curso de acción interno para poder solucionar el inconveniente. Los objetivos están definidos de acuerdo a lo indicado según el contrato y que se explicó en el primer párrafo. En casos de desvío se verifican las causas y se solicita al BA responsable del cliente las explicaciones para conocer los motivos del desvío. Se consideró necesario que una de las posibles acciones que se tomen ante esta situación es realizar una capacitación al personal sobre manejo y trato con los clientes.

5.6.5. Frecuencia de los Sprints

Con el fin de garantizar la entrega continua de valor a los clientes, es imperativo ejercer un control sobre la frecuencia de los sprints. Esto asegura que se cumplan los plazos establecidos por la dirección, lo que a su vez posibilita la entrega constante de soluciones a problemas y nuevas funcionalidades, generando así un incremento en la satisfacción del cliente. Además, es crucial respetar el período de los sprints debido a que el cierre de cada uno, junto con la celebración de la revisión y la retrospectiva, proporciona al equipo de desarrollo un valioso feedback interno. Esta retroalimentación es esencial para identificar áreas de mejora de cara al próximo sprint.

La dirección considera necesario que cada sprint tenga una duración de dos semanas, que ante casos excepcionales podría llegar a tener una duración de máximo tres semanas. Cada sprint comenzará el primer día de la semana, y terminará preferentemente un día viernes para dar cierre al sprint un último día de la semana.

5.6.6. Planificación de requerimientos vs. Requerimientos ejecutados

Este indicador se controla al cierre de cada uno de los sprints, forma parte del cierre del mismo y es necesario para poder tener claro la efectividad de la planificación realizada al comienzo del mismo.

Al definir el indicador, se consideró como objetivo principal verificar el cumplimiento del plan establecido y utilizarlo como herramienta de control para monitorear la ejecución del plan y evaluar mejoras en este aspecto. El objetivo de este indicador es lograr un cumplimiento de al menos el 60%. Aunque este porcentaje inicialmente puede parecer poco exigente, se tomó en cuenta que al inicio de la implementación del proceso, podría haber una estimación menos precisa hasta que el equipo adquiriera experiencia y mejore su capacidad para estimar los tiempos. Con el tiempo, se espera que este porcentaje de cumplimiento aumente.

6. Resultados

Durante el primer mes, se comienza a observar los desafíos iniciales que presenta la implementación de un cambio de proceso, donde se puede evidenciar en primera instancia una resistencia al cambio que se manifiesta en las diferentes áreas que participan del mismo, y que se produce principalmente en las personas que poseen mayor antigüedad en la empresa.

Los principales factores de resistencia al cambio que se encontraron durante la implementación del nuevo proceso son dos factores claves:

El primero de ellos es la sensación de pérdida de control. Sobre este aspecto los principales referentes de las áreas de desarrollo y de análisis funcional sintieron que estaban perdiendo autonomía y control sobre el área al que pertenecían, principalmente por estar acostumbrados a una estructura jerárquica con la que operaba anteriormente la empresa.

Otro de los factores destacados fue el miedo a lo desconocido que implica la adaptación de un nuevo proceso, lo que genera ansiedad y temor en los participantes del mismo. Principalmente, debido a que muchas de las personas que se encuentran trabajando tienen pocas o nulas experiencias con metodologías ágiles.

Estos aspectos provocaron, no sólo el descontento inicial de ciertos miembros del equipo, sino que también planteó el riesgo de disminuir la confianza de aquellos que estaban comprometidos con el cambio, poniendo en duda el valor de sus esfuerzos y potencialmente desmotivando a todo el personal.

Además, en las etapas iniciales, surgieron conflictos entre los participantes del nuevo proceso, lo cual es un riesgo en la efectividad de la implementación del cambio. Este escenario provoca que pueda resultar en un fracaso en la ejecución del cambio, dejando sin concretar los resultados y objetivos previamente establecidos a pesar del tiempo y esfuerzo invertidos.

Por lo tanto, en ese primer mes se tomaron ciertas decisiones para poder abordar esta resistencia al cambio, entre ellas podemos enumerar las más importantes aplicadas:

- Se realizó un especial énfasis en la comunicación para que la misma pueda ser clara y continua: se comunicaron reiteradamente los beneficios y objetivos del cambio. Indicando detalladamente como las metodologías ágiles mejoran el proceso de desarrollo y el producto final. Además se realiza un seguimiento por parte del scrum master a cada uno de los responsables del proceso para que la comunicación sobre lo que debería hacerse sea lo más clara posible.
- Durante la etapa de preparación del cambio, se ofrecieron cursos de capacitación interna y externa para ayudar a cada uno de los colaboradores a comprender y adoptar metodologías ágiles. Para que la capacitación pueda ser efectiva se realizó una evaluación al final de la misma para evaluar la eficacia, y además también se ofrece que las horas de capacitación

puedan ser compensadas con permisos particulares, esto último se adopta para motivar la asistencia a capacitaciones por parte del personal.

- Otro aspecto importante para poder bajar el impacto que produce la resistencia al cambio es el compromiso que se demostró desde los líderes que manifiestan constantemente el respaldo hacia el cambio, y además fueron las primeras personas que estuvieron involucradas desde el comienzo realizando las actividades que se les solicitaba de manera activa. Esto es fundamental, ya que con el ejemplo se influye en la actitud y la percepción que tiene el resto de las personas en relación al cambio.

El éxito rotundo de las estrategias implementadas para contrarrestar la resistencia al cambio ha generado un impacto tangible en un corto período de cuarenta y cinco días desde su aplicación inicial. El personal, que inicialmente mostraba cierta reticencia hacia la adopción de las nuevas prácticas, ha logrado una adaptación notable a la nueva dinámica operativa. Este hito marca un importante punto de inflexión en el proceso de transformación organizacional, ya que los equipos comienzan a experimentar los primeros frutos de sus esfuerzos y compromiso con el cambio.

Ante estos resultados alentadores, la alta dirección ha optado por aprovechar esta oportunidad para comunicar y celebrar los logros alcanzados hasta el momento. La intención es destacar públicamente los aspectos positivos derivados de la nueva metodología implementada, con el fin de demostrar de manera tangible los beneficios y las oportunidades que esta trae consigo. Esta acción estratégica busca no solo reconocer y motivar al personal por su contribución al éxito inicial del proceso de cambio, sino también convencer a aquellos que aún albergaban dudas sobre la efectividad y el valor de la transformación en curso. Al destacar los primeros resultados positivos, se pretende generar un impulso adicional y consolidar el respaldo necesario para asegurar la continuidad y el éxito del proyecto de cambio a largo plazo.

Tras cuatro meses desde la implementación del cambio de proceso, se procedió a llevar a cabo una nueva encuesta de satisfacción de clientes, la cual se ejecuta de manera regular cada año. Es relevante destacar que en el período anterior, los resultados de esta encuesta no alcanzaron los estándares deseados, como se evidencia en la ilustración 5. Sin embargo, en esta ocasión, los resultados obtenidos han sido excepcionalmente positivos, marcando una diferencia significativa en la percepción general del cliente respecto al servicio y producto ofrecido por la empresa.

La comparación entre los resultados actuales y los del año anterior ofrece una visión clara del progreso logrado tras la implementación del cambio de proceso. Los datos recopilados revelan un aumento notable en la satisfacción del cliente, así como una mejora significativa en diversos aspectos clave, como la calidad del servicio, la eficiencia operativa y la atención al cliente. Estos resultados no solo reflejan el impacto positivo del cambio de proceso en la experiencia del cliente, sino que también validan la efectividad de las estrategias y acciones implementadas para abordar las deficiencias

identificadas en el pasado. En este sentido, estos hallazgos respaldan la relevancia y el éxito de la iniciativa de cambio de proceso como parte integral de la estrategia empresarial para mejorar la satisfacción del cliente y fortalecer la posición competitiva en el mercado.



Ilustración 20 - Nueva encuesta satisfacción de clientes

Durante la ejecución de la encuesta de satisfacción de clientes, se llevó a cabo una revisión exhaustiva del cuestionario existente, la cual resultó en la incorporación de nuevas preguntas destinadas a capturar de manera más precisa la percepción del cliente sobre la empresa y sus servicios.

Además de estas adiciones, se introdujo una innovación significativa al permitir que los clientes compartieran sus comentarios personales acerca de la visión general de la empresa. Esta iniciativa dio lugar a una respuesta positiva por parte de los clientes, quienes expresaron su sorpresa y satisfacción ante las mejoras observadas en el servicio. Algunos de los comentarios recibidos destacan el notable progreso en la calidad de la atención al cliente y la percepción de una mayor organización dentro de la empresa. Por ejemplo, se registraron comentarios como: "Me agrada la mejora en la atención, se percibe una mayor organización", "La atención es de excelencia" y "Siempre están dispuestos a ayudar, se nota una mejora tanto en la atención como en el sistema".

Adicionalmente, se incluyó en la encuesta la opción para que los clientes evalúen la probabilidad de recomendar la empresa a otras personas, utilizando una escala del 1 al 10, donde 10 indica una alta probabilidad de recomendación y 1 una baja probabilidad. Es destacable que, en ningún

caso, se obtuvo una calificación inferior a 7, lo que sugiere una satisfacción generalizada entre los clientes encuestados. Los resultados revelaron que las puntuaciones más frecuentes fueron 8 y 10, lo que indica un alto nivel de satisfacción y una alta propensión de los clientes a recomendar los servicios de la empresa a sus contactos y redes de influencia. Este hallazgo respalda la eficacia de las iniciativas implementadas para mejorar la experiencia del cliente y fortalecer la reputación de la empresa en el mercado.

Como se mencionó anteriormente, los resultados obtenidos en base a la encuesta de satisfacción han mejorado notablemente y eso es un factor muy positivo y deseable por la empresa, ya que claramente se puede observar que la mejora de procesos interna ha tenido un impacto en las relaciones con los clientes que genera mejores condiciones de trabajo en conjunto y la posibilidad de seguir creciendo.

Por otro lado, también se logra observar luego de seis meses de la implementación del cambio que se ha incrementado el interés en el mercado del rubro de la salud por el producto. Según las encuestas realizadas, gran parte de ellas ha surgido por la recomendación de clientes actuales. Esto también deja como enseñanza que no solo el impacto es en los clientes actuales sino también en los clientes potenciales que pueden obtener principalmente por la recomendación de los primeros cuando sienten que se le está dando un servicio acorde a sus necesidades y que se encuentran conformes con el resultado que obtienen.

También comienza a tener importancia los indicadores de medición mencionados en el apartado 5.6 que establecen el cumplimiento del proceso correctamente, y que sirven para poder establecer la eficacia del mismo. Se realiza un control mensual de los indicadores para verificar su eficacia, y sobre los mismos se observan los resultados.

Se procede inicialmente a realizar un análisis de los resultados obtenidos, que se cotejan en primer término con los objetivos predefinidos para verificar su alineación. En caso de desviaciones respecto a dichos objetivos, se procede a llevar a cabo un análisis exhaustivo de las causas subyacentes que hayan generado dicho incumplimiento, con el fin de identificar áreas de mejora y tomar medidas correctivas pertinentes. En situaciones donde se evidencien tales desviaciones, aunque se haya registrado un número limitado de casos, particularmente durante los dos primeros meses de adaptación al cambio, se han implementado acciones correctivas centradas principalmente en la capacitación del personal y en intervenciones destinadas a fomentar la motivación y gestionar el proceso de cambio de manera efectiva. Estas medidas incluyen, entre otras, programas de formación específicos y análisis detallados de los puntos críticos de falla en el proceso.

La revisión de los indicadores se realiza mediante una reunión de manera mensual, donde participan el scrum master, los PM y los jefes de equipo, y es realizada el primer viernes de cada mes. Luego de finalizada dicha reunión, mediante el envío de un mail se pone a disposición la información a todas las personas de la organización para puedan tener la información de los resultados obtenidos en

el mes anterior, ya que se considera que es una información valiosa para que pueda ser compartida para todos, principalmente para que todas las personas estén enteradas que las revisiones y que los controles sobre los mismos son aplicados. Por otro lado, también permite a todos los integrantes poder tener contacto de primera mano con la información sobre el éxito en los casos que los resultados se encuentren bien, y también de qué manera pueden mejorar en los casos que algún indicador no se llegue a cumplir. Esto último implicaba un desafío importante para la empresa, ya que inicialmente se generó un debate sobre las ventajas de compartir la información con todos los colaboradores, pero los resultados de hacerlo y el nivel de involucramiento que se generó en el personal fue tan positivo que hasta el día de la fecha se sigue realizando de esta manera.

A continuación se presenta un ejemplo del indicador “Requerimientos clasificados por estado y tipo” con los datos reales correspondientes al período diciembre 2023, los mismos son enviados y distribuidos a todas las personas de la empresa. Así como puede observarse para este caso particular, también la información del resto de los indicadores es generada mediante gráfico para ser de mayor rapidez su lectura, pero además se encuentra toda la información correspondiente al detalle de los datos, con la información disponible a disposición para ser analizado por quien lo desee:



Ilustración 21 - Indicador de requerimientos por Tipo de Requerimientos



Ilustración 22 - Indicador Requerimientos por Estado

En conclusión, la empresa experimentó inicialmente desafíos y resistencia al cambio durante la implementación de nuevos procesos, especialmente debido a la pérdida de control percibida y al miedo a lo desconocido por parte de algunos empleados. Sin embargo, mediante decisiones estratégicas como una comunicación clara y continua, capacitación interna y externa, compromiso de los líderes y el seguimiento de indicadores clave, la organización logró superar con éxito la resistencia al cambio.

Después de los primeros dos meses y se acentúa aún más a partir del cuarto mes, los resultados positivos comenzaron a ser evidentes, con una encuesta de satisfacción de clientes reflejando mejoras significativas. La introducción de preguntas adicionales y la posibilidad de comentarios permitieron una comprensión más profunda de las necesidades de los clientes, contribuyendo al cambio positivo en la percepción del servicio y el producto entregado.

Además, se observó un aumento en el interés del mercado y nuevas oportunidades de negocio, destacando la importancia de las recomendaciones de clientes satisfechos. La implementación de un sistema de medición y control de indicadores mensuales demostró ser efectiva para evaluar el cumplimiento del proceso y tomar acciones correctivas cuando fuera necesario. La transparencia en la comunicación de los resultados, a través de reuniones mensuales y distribución de información a todo el personal, generó un alto nivel de involucramiento y contribuyó al éxito continuo de los nuevos procesos.

En última instancia, la empresa no solo logró mejorar sus procesos internos, sino que también experimentó un impacto positivo en la satisfacción del cliente, el reconocimiento en el mercado y la cultura organizacional, demostrando que la gestión efectiva del cambio puede conducir a resultados exitosos y sostenibles.

7. Conclusiones

La implementación de cambios en el proceso de desarrollo de esta empresa de software dedicada al sector de la salud, es un desafío significativo que requiere una cuidadosa planificación y gestión. Este trabajo de tesis se centró en analizar, proponer y ejecutar mejoras en el proceso de desarrollo existente para permitir que la empresa alcance sus objetivos estratégicos y se consolide como líder en su industria en un plazo de cinco años.

Al llevar a cabo el primer objetivo específico, nos encontramos que la empresa enfrentaba problemas críticos relacionados con la falta de documentación y estandarización en su proceso de desarrollo de software, lo que resultaba en ineficiencias operativas y en la insatisfacción tanto de los clientes como de los empleados. La necesidad de cambio se hizo evidente cuando la incorporación de nuevos clientes generó un desequilibrio entre los requerimientos entrantes y los resueltos, afectando la productividad y la satisfacción del cliente.

Posteriormente, se realizó un relevamiento del estado inicial de la empresa para poder llevar a cabo el segundo objetivo, en el mismo se identificaron conflictos entre las áreas debido a la falta de especificaciones y roles claros, así como la ausencia de un proceso definido. Este escenario condujo a la búsqueda de mejores prácticas en la industria del desarrollo de software, destacando la relevancia de las metodologías ágiles, particularmente Scrum.

Como siguiente punto, se realiza una propuesta para mejorar el proceso, mediante la implementación de Scrum que se basó en una planificación detallada, la introducción de reuniones diarias y retrospectivas, y la redefinición de roles.

Uno de los puntos importantes a destacar en los primeros meses de implementación es la resistencia al cambio que fue un desafío inicial, destacándose la sensación de pérdida de control y el miedo a lo desconocido entre los empleados. Sin embargo, la empresa adoptó medidas estratégicas, como una comunicación clara y continua, cursos de capacitación y el compromiso visible de los líderes, para abordar estos problemas y fomentar una transición suave.

La introducción de un sistema de medición y control de indicadores mensuales permitió evaluar la eficacia del nuevo proceso y tomar acciones correctivas cuando fue necesario. La transparencia en la comunicación de los resultados, a través de reuniones mensuales y distribución de información a todo el personal, generó un alto nivel de involucramiento y contribuyó al éxito continuo de los nuevos procesos, permitiendo así alcanzar el cuarto objetivo específico.

A medida que la empresa avanzó en la implementación de Scrum, los resultados positivos se reflejan no solo en la eficiencia operativa interna, sino también en la satisfacción del cliente y en la capacidad para atraer nuevos negocios. La encuesta de satisfacción del cliente demostró mejoras significativas en áreas críticas, como el soporte y la velocidad de respuesta, lo que contribuyó a

fortalecer las relaciones con los clientes existentes y a atraer nuevos clientes gracias a recomendaciones positivas.

Una conclusión fundamental de este estudio es la necesidad crítica de contar con un proceso claro y bien definido de desarrollo. Un proceso bien estructurado proporciona la guía necesaria para las operaciones diarias, asegurando la consistencia y la eficiencia en todas las etapas del desarrollo de software. La claridad en los roles, responsabilidades y flujos de trabajo contribuye significativamente a la reducción de malentendidos y conflictos internos.

La inclusión de indicadores de medición adecuados es un componente esencial de cualquier proceso de mejora. Estos indicadores proporcionan una visión objetiva del rendimiento y permiten realizar ajustes proactivos antes de que se generen desviaciones significativas.

Uno de los resultados más significativos de este estudio es la conexión directa entre un proceso bien gestionado y la satisfacción del cliente. Cuando los procesos internos son eficientes y bien coordinados, los clientes experimentan entregas más rápidas, productos de mayor calidad y una comunicación más clara.

La mejora de procesos no solo se trata de eficiencia operativa y satisfacción del cliente; también tiene un impacto directo en la rentabilidad de la empresa. Al optimizar los flujos de trabajo, reducir los tiempos de desarrollo y mejorar la calidad del producto, la empresa logra una mayor eficiencia en costos y una utilización más efectiva de los recursos.

En última instancia, esta investigación y aplicación práctica de mejoras en los procesos de desarrollo de software no solo se ve como una iniciativa operativa, sino como una estrategia integral que abarca la cultura organizacional, el compromiso del personal, la satisfacción del cliente y la rentabilidad empresarial, sino que también sentó las bases para la continuación de un crecimiento sostenible y competitivo en el dinámico y cambiante entorno de la industria de la salud y el desarrollo de software.

BIBLIOGRAFÍA

- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., & Thomas, D. (2001). Manifiesto for agile software development.
- Jeff Sutherland, John Scumniotales y Jeff McKenna (1993) “SCRUM: El Arte de Hacer el Doble del Trabajo en Mitad del Tiempo”
- Taiichi Ohno (1985) “Kanban y Just in Time en Toyota”
- <https://blog.buhoos.com/diferencias-entre-agile-y-waterfall/>
- <https://echometerapp.com/es/estadisticas-del-scrum/>
- Sutherland, J., & Sutherland, J. (2014). Scrum: The Art of Doing Twice the Work in Half the Time. Random House.
- Mercado-Ramos, Victor Hugo, Julián Zapata y Yony Fernando Ceballos. "Herramientas y buenas prácticas para el aseguramiento de calidad de software con metodologías ágiles". REVISTA DE INVESTIGACIÓN, DESARROLLO E INNOVACIÓN 6, n.º 1 (9 de diciembre de 2015)
- Rivadeneira Molina, Silvia Gabriela. "Metodologías ágiles enfocadas al modelado de requerimientos". Informes Científicos Técnicos - UNPA 5, n.º 1 (11 de junio de 2014)
- Cadavid, Andrés Navarro. "Revisión de metodologías ágiles para el desarrollo de software". Prospectiva 11, n.º 2 (20 de septiembre de 2013)-
- Paez Cardenas, Patricia, Christian David Arias Peralta y Luis Felipe Wanumen Silva. "Metodología para elaboración de requerimientos en aplicaciones con servicios web". Revista vínculos 15, n.º 2 (6 de noviembre de 2018)