

**UNIVERSIDAD DON BOSCO
DIRECCIÓN DE EDUCACIÓN A DISTANCIA**



**TRABAJO DE GRADUACIÓN PARA OPTAR AL GRADO DE:
MAESTRO EN ARQUITECTURA DE SOFTWARE**

PROYECTO:

IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN SI BOARD EN EL
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS UDB UTILIZANDO LA
METODOLOGÍA AGILE 2

PRESENTADO POR:
JOSÉ MARIO MENJIVAR PINEDA
JOSHUA VILLAVICENCIO

ASESOR:
MG. JOSÉ MAURICIO FLORES AVILÉS

ANTIGUO CUSCATLÁN, LA LIBERTAD, EL SALVADOR, CENTRO AMÉRICA
JUNIO 2022

Agradecimientos

Agradecimientos Mario Menjivar

Agradezco primeramente a Dios por todas sus bendiciones brindadas a mi vida, entre las que se encuentra el haberme permitido cursar la Maestría en Arquitectura de Software en la Universidad Don Bosco, sin su ayuda esto no habría sido posible.

Agradezco a todo el personal de la Dirección de Educación a Distancia y a los docentes de la Maestría en Arquitectura de Software por su inestimable ayuda y por todo lo que han aportado a mi formación y crecimiento profesional.

Agradezco al Mg. Mauricio Flores Avilés por haber sido el asesor de este trabajo de graduación, él realizó una muy buena labor brindando su ayuda siempre que fue necesario.

Agradezco a mi compañero de trabajo de graduación y amigo, Joshua Villavicencio por su colaboración en la realización de este trabajo.

Agradezco a mi madre, Maria Pineda; a mi padre, Mario Menjivar; y a mi hermana, Katya Menjivar, por su constante amor y apoyo el cual me demuestran diariamente.

Agradezco a mi novia, Sara Chávez, por su apoyo incondicional durante el tiempo que cursé esta maestría, de igual manera, agradezco a mis amigos por su apoyo incondicional y por sus palabras de aliento siempre que fueron necesarias.

Agradecimientos Joshua Villavicencio

Agradezco a mi madre, Lilian Villavicencio por su apoyo y amor incondicional durante toda mi vida y por impulsarme a ser el mejor profesional y persona posible día a día.

A mi compañero de trabajo, José Mario Menjívar por su apoyo, ética y profesionalismo no solo durante la realización del trabajo de grado sino durante toda la carrera.

Agradezco al Mg. José Mauricio Flores por su guía, disposición y aportes durante todo el proceso de realización del trabajo de graduación.

Agradezco al personal de la Dirección de Educación a Distancia, personal, asesores y docentes por su inestimable acompañamiento durante todo mi proceso de formación profesional.

Resumen

En el presente trabajo se identifican los procesos vinculados al desarrollo de software llevados a cabo en el Departamento de Sistemas Informáticos de la Universidad Don Bosco, la cual desde 2018 ha adoptado el marco de trabajo Scrum; una vez identificados dichos procesos, se determinan las oportunidades de mejora dentro del departamento. Se pretende solventar dichas oportunidades a través del desarrollo de una guía que brinde los lineamientos necesarios para la creación de un sistema de gestión llamado SI Board, el cual es desarrollado a partir de los principios y valores de la metodología Agile 2. Una vez desarrollado este sistema, se evalúa el impacto que este tiene dentro de la organización y se determina las mejoras que se obtuvieron gracias al uso de la metodología Agile 2.

Índice

Agradecimientos	2
<i>Agradecimientos Mario Menjivar</i>	2
<i>Agradecimientos Joshua Villavicencio</i>	3
Resumen	4
Índice	5
Lista de Abreviaturas y Símbolos	8
Introducción	9
Formulación General del Proyecto	11
<i>Valor Pedagógico y/o Innovación que Presenta el Proyecto</i>	11
<i>Datos Sobre la Organización Objetivo</i>	12
<i>Relevancia Social</i>	16
<i>Objetivos del Proyecto</i>	16
<i>Descripción de Productos o Procesos de Innovación</i>	17
Fundamentación Teórica	19
<i>Agile</i>	19
<i>Scrum</i>	21
<i>Esfuerzos Previos por Mejorar las Metodologías Ágiles</i>	37

<i>Agile 2</i>	43
Metodología	52
<i>Alcance</i>	52
<i>Limitaciones</i>	52
<i>Hoja de Ruta para la Recolección de Datos</i>	53
<i>Proceso de Recolección de Datos</i>	58
<i>Manejo Actual de Scrum y Retos en su Implementación</i>	70
<i>Problemas Encontrados</i>	72
<i>Hoja de Ruta para la Elaboración de la Propuesta de Solución</i>	75
<i>Opinión Experta sobre el Uso de Agile 2</i>	76
Propuesta de Solución	80
<i>Oportunidades de Mejora y Prácticas de Agile 2 Identificadas para Solventar los Problemas de la Organización</i>	80
<i>Cuadro Resumen de los Principios de Agile 2 a Adoptar</i>	93
<i>Requerimientos del Sistema de Gestión SI Board</i>	96
<i>Guía de Implementación del Sistema de Gestión SI Board</i>	100
<i>Cuadro Resumen de las Estrategias Contenidas en la Guía</i>	105
<i>Marco de Trabajo para Implementar el Sistema de Gestión SI Board</i>	108
<i>Herramientas Externas Recomendadas</i>	119

<i>Hoja de Ruta para Implementar el Sistema de Gestión SI Board</i>	<i>121</i>
<i>Factores de Éxito en la Implementación de la Guía de Adopción de Agile 2</i>	<i>125</i>
<i>Verificación de la Implementación de la Guía.....</i>	<i>127</i>
<i>Estrategias Aplicadas para Solventar los Problemas Encontrados en la Organización</i>	<i>145</i>
<i>Mejoras Obtenidas Gracias a la Implementación de la Guía</i>	<i>152</i>
Conclusiones.....	155
Recomendaciones.....	156
Glosario	157
Referencias	158
Anexos.....	161
<i>Anexo A - Encuesta inicial realizada a los trabajadores de la organización objetivo.....</i>	<i>161</i>
<i>Anexo B - Encuesta final realizada a los trabajadores de la organización objetivo.....</i>	<i>175</i>
<i>Anexo C - Entrevista Realizada al jefe del Departamento de Sistemas Informáticos</i>	<i>184</i>
<i>Anexo D - Entrevista Realizada a Cliff Berg, Principal Fundador de Agile 2</i>	<i>188</i>
<i>Anexo E - Fichas de Registro de Observación</i>	<i>193</i>
<i>Anexo F - Ejemplo de Historia Épica</i>	<i>202</i>
<i>Anexo G - Ejemplo de User Story Map.....</i>	<i>204</i>

Lista de Abreviaturas y Símbolos

CTIC: Centro de Tecnología de la Información y las Comunicaciones.

PO: Product Owner.

QA: Quality Assurance (aseguramiento de la calidad).

SI UDB: Departamento de Sistemas Informáticos de la Universidad Don Bosco.

SM: Scrum Master.

ST: Scrum Team.

TI: Tecnologías de la Información.

UDB: Universidad Don Bosco.

XP: Extreme Programming.

Introducción

El departamento de Sistemas Informáticos de la Universidad Don Bosco fue creado en el año 2008 con la finalidad de poder desarrollar software a la medida que ayude a la realización de las diversas tareas que desarrollan cada uno de los trabajadores de esta institución. El departamento inició labores con dos empleados quienes eran profesionales en informática y pusieron a disposición del departamento sus conocimientos técnicos.

El departamento de Sistemas Informáticos fue creciendo con el paso de los años, se han ido añadiendo nuevos empleados los cuales contribuyen en los procesos de realización de proyectos de software. Los dos empleados iniciales del departamento asumieron un rol de liderazgo para con los demás empleados convirtiéndose en los jefes del departamento.

Desde el año 2008 hasta el año 2018, los procesos organizativos y administrativos referente a cada uno de los desarrollos que se realizaban en el departamento, se hacían de manera rudimentaria, debido a que no se aplicaba una metodología o marco de trabajo que brindará las bases o una cultura que determinará la manera en la que se realizaban los desarrollos de software dentro del departamento.

Fue en el año 2018 que esta realidad cambió, se realizó un importante cambio dentro de la jefatura del departamento, esto trajo consigo cambios entre los cuales se encontraba la adopción de una metodología o marco de trabajo ágil que proporcionará herramientas útiles para el desarrollo eficaz y eficiente de proyectos de software, el marco de trabajo elegido para ser adoptado fue Scrum.

Para la organización, Scrum ha sido un marco de trabajo de mucha utilidad durante los últimos cuatro años, pues les ha permitido determinar los lineamientos para realizar sus procesos. Si bien, Scrum es uno de los marcos de trabajo ágiles más adoptados a nivel mundial, su uso no es exclusivo dentro del campo del desarrollo de software, sino que se ha convertido en un marco de trabajo de uso general el cual puede ser utilizado por organizaciones de cualquier rubro, esto ha generado que haya preponderancia en el enfoque hacia habilidades blandas en lugar de habilidades técnicas, es por esta razón que la jefatura del departamento está dispuesta a buscar nuevas opciones que fortalezcan y mejoren sus actuales procesos, esto a través de la adopción de una nueva metodología, ellos quieren explorar la oportunidad de adoptar la metodología Agile 2.

Agile 2 es una revisión a las metodologías ágiles tradicionales, brindando un gran enfoque al desarrollo de software, esto lo hace centrándose más en el área técnica antes que en el área de habilidades blandas. La adopción de esta metodología dentro del Departamento de Sistemas Informáticos de la Universidad Don Bosco fortalecerá sus procesos de desarrollo y garantizará que los productos o servicios desarrollados sean eficientes, eficaces y de calidad.

La adopción de la metodología Agile 2 se realizará a través de la implementación de un sistema de gestión llamado SI Board. Dicho sistema será desarrollado implementando los principios y valores que componen la metodología Agile 2.

Formulación General del Proyecto

Valor Pedagógico y/o Innovación que Presenta el Proyecto

En la actualidad, el desarrollo de software es comúnmente llevado a cabo implementando ciertas prácticas comunes que vienen determinadas por metodologías que garantizan resultados satisfactorios. Es de esta manera, que organizaciones enfocadas al desarrollo de software han adoptado marcos de trabajo como Scrum o XP y herramientas como Kanban; los cuales nacieron en la década de los 90 's. Con un fuerte enfoque en el área del desarrollo de software, pero con el tiempo de acuerdo con Berg (2021):

Vimos el ascenso de dos nuevas profesiones (Scrum Master y Agile Coach), que rápidamente fueron dominadas por gente no técnica surgidos de cualquier tipo de profesión y, debido a que su experiencia es no técnica, se enfocaron en el lado humano de los equipos de software: confianza del equipo, felicidad del equipo, etc. (p. 8).

Debido a este desenfoco que hubo en las metodologías y marcos de trabajo ágiles, sucedió que en octubre de 2019 se lanzó la metodología Agile 2, según Berg (2021) el objetivo de Agile 2 es “realizar una corrección de curso de Agile, pero de una forma abierta, aditiva, inclusiva y no dogmática” (p. xxii) la cual tiene la premisa de expandir el uso de Agile más allá de la industria del Software, sin descuidar los aspectos técnicos de la misma. En la actualidad, hay pocas empresas que han optado por la adopción de esta metodología debido a que es reciente con respecto a las metodologías más populares de la actualidad.

La adopción de esta metodología permitirá al Departamento de Sistemas Informáticos de la Universidad Don Bosco, ser uno de los pioneros a nivel regional en el uso de Agile 2, puesto que dicha metodología otorga mayor flexibilidad a los principios ágiles tradicionales e integra aspectos como la entrega continua al proceso de creación de productos. Esto fortalecerá las habilidades técnicas del personal que compone el departamento, ayudando en la obtención de desarrollos de software que sean eficientes y eficaces.

La Universidad Don Bosco sería una pionera en la adopción de esta metodología, sería el espejo en el cual pueden mirarse otras organizaciones que en el futuro deseen adoptar Agile 2, lo cual puede verse reflejado en la posibilidad de que los miembros del Departamento de Sistemas Informáticos puedan brindar capacitaciones, entrenamientos o consultorías sobre la metodología en cuestión.

Datos Sobre la Organización Objetivo

La organización objetivo de este trabajo es el Departamento de Sistemas Informáticos, dicho Departamento pertenece a la Universidad Don Bosco de El Salvador, el Departamento fue fundado en el año 2008 por los ingenieros Cesar Espinoza, quien actualmente ostenta el cargo de director de CTIC (Centro de Tecnología de la Información y las Comunicaciones); y Jorge Martínez, quien ya no labora más en la organización.

El Departamento de Sistemas Informáticos de la UDB desarrolla proyectos de software hechos a la medida, es decir, según las necesidades específicas de sus clientes.

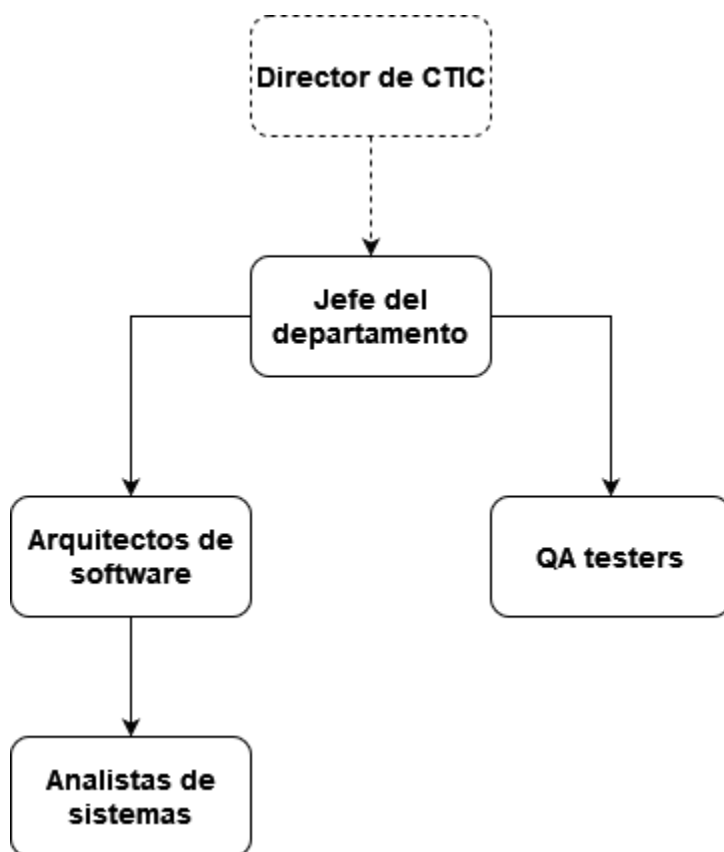
Los requerimientos son solicitados por los 126 Departamentos que componen la Universidad Don Bosco y que cuentan con una necesidad específica que esperan solventar a través de la implementación de un producto de software.

La organización desarrolló múltiples proyectos a la vez, esto debido a que sus clientes son múltiples y las solicitudes por parte de estos son recurrentes, el tiempo de desarrollo de un proyecto varía dependiendo de la complejidad de la necesidad a solventar y de la cantidad de desarrolladores de software que trabajan en dicho proyecto.

Actualmente la organización cuenta con 11 miembros, distribuidos de la siguiente manera: jefe de departamento de Sistemas Informáticos, arquitectos de software, QA tester y analistas de sistemas. Cabe destacar que existe una figura auxiliar a la organización, esta es la del director de CTIC, esto debido a que él ayuda en la realización de ciertas actividades relativas a la organización, pero él no es parte de la organización como tal. El organigrama organizacional se presenta en la figura 1.

Figura 1

Organigrama del Departamento de Sistemas Informáticos UDB



Nota: La línea punteada denota que el director de CTIC realiza labores auxiliares y no es parte de la organización como tal.

A continuación, se detallan las funciones que realizan los miembros de la organización según su cargo:

- Director de CTIC: Se encarga de gestionar el área de TI de la Universidad Don Bosco, cumple un rol auxiliar dentro del Departamento de Sistemas Informáticos ya que les

ayuda con el mantenimiento de servidores, lanzamiento de proyectos y ejecución de cambios en las bases de datos de los proyectos desarrollados.

- Jefe del Departamento de Sistemas Informáticos UDB: Es el encargado de gestionar al equipo de desarrollo de software, él gestiona cada proyecto realizado y se asegura que cada desarrollador realice de manera adecuada sus labores.
- Arquitectos de software: Son los desarrolladores de software con mayor experiencia y conocimiento dentro del equipo, ellos tienen a su cargo a su propio grupo de analistas de sistemas. Sus labores consisten en desarrollar proyectos de software además de gestionar y ayudar en sus labores a los analistas de sistemas que están a su cargo.
- QA testers: Ellos trabajan de la mano con los desarrolladores para dar seguimiento a los proyectos de software durante las distintas etapas de su ciclo de vida, esto para validar que dichos proyectos cumplen con criterios de calidad que garanticen un producto satisfactorio para el cliente.
- Analistas de Sistemas: Se encargan de desarrollar los distintos proyectos de software que son solicitados al Departamento de Sistemas Informáticos UDB. Ellos se encargan de definir la hoja de ruta que seguirán para realizar los proyectos que les son asignados.

A continuación, en la figura 1 se presenta un organigrama que muestra cómo está organizado el orden jerárquico dentro del Departamento de Sistemas Informáticos UDB:

Relevancia Social

La implementación del proyecto propuesto “Implementación del sistema de gestión SI Board en el Departamento de Sistemas Informáticos UDB utilizando la metodología Agile 2” permitirá determinar el grado de mejora que puede alcanzar el Departamento de Sistemas Informáticos, utilizando la metodología Agile 2.

A través de este proyecto se desarrollará una herramienta que permitirá gestionar los diferentes artefactos y documentos necesarios para el desarrollo de proyectos bajo metodologías ágiles, los cuales pueden ser utilizados por los Departamentos fuera del campo de la gobernanza de TI y permitirá al equipo de desarrollo tener una experiencia de primera mano con una nueva metodología de desarrollo, el nombre de esta herramienta a desarrollar será SI Board.

Objetivos del Proyecto

Objetivo General

- Realizar la implementación del sistema de gestión SI Board en el Departamento de Sistemas Informáticos UDB utilizando la metodología Agile 2.

Objetivos Específicos

- Investigar las características principales de la metodología Agile 2 y sus diferencias con las metodologías ágiles tradicionales.
- Establecer la situación actual de la organización y las metodologías de desarrollo utilizadas.

- Crear una guía de implementación para la creación de un sistema de gestión (SI Board) utilizando la metodología Agile 2.
- Diseñar un marco de trabajo y hoja de ruta para llevar a cabo la implementación del sistema SI Board.
- Aplicar la guía y el marco de trabajo diseñado en el Departamento de Sistemas Informáticos.
- Verificar la correcta implementación del sistema de gestión propuesto.
- Identificar las mejoras y el impacto obtenido en los procesos de la organización gracias a la implementación del sistema de gestión SI Board.

Descripción de Productos o Procesos de Innovación

A través del presente proyecto se espera implementar la metodología Agile 2 dentro del Departamento de Sistemas Informáticos, para ello se desarrollará una guía que brinde las pautas a seguir para que los miembros del departamento en cuestión adopten la metodología propuesta.

Dicha guía será diseñada en base a los principios y prácticas que forman parte de la metodología Agile 2, será de fácil entendimiento tanto para personas con conocimiento o experiencia previa sobre metodologías ágiles como para personas sin experiencia en dicha área.

Para verificar la correcta adopción y mejora en la producción de desarrollo de Software, el Departamento de Sistemas Informáticos UDB desarrollará un proyecto piloto utilizando la metodología Agile 2, este proyecto recibirá el nombre de SI Board y sus funcionalidades estarán

enfocadas a la gestión de los procesos internos que se llevan a cabo en el Departamento de Sistemas Informáticos.

A través del proyecto piloto se espera determinar si la adopción de Agile 2 ha sido exitosa y que tanto han mejorado los procesos de producción de software de la organización. A partir de dicho dictamen se espera identificar las mejoras obtenidas gracias a la metodología Agile 2 y las ventajas competitivas que esta metodología presenta comparándola contra metodologías ágiles tradicionales.

Fundamentación Teórica

Agile

Es un conjunto de prácticas, recursos y soluciones utilizadas en un esfuerzo colaborativo auto organizado, con el objetivo de desarrollar productos funcionales enfocados en otorgar valor al cliente, incorporar procesos de mejora continua y proporcionar respuestas rápidas a los cambios en requerimientos, capacidades o problemas que surjan durante el proceso.

Según SCRUMStudy (2013) las metodologías de desarrollo iterativo de software han existido desde la década de los 50's (p. 316) con la teoría de manejo de proyecto evolutivo y desarrollo de software adaptativo, pero fue hasta el año 2001 que un grupo de desarrolladores de software publicaron el manifiesto de desarrollo ágil de software, utilizando el término "Manifiesto Agile" para representar la capacidad de respuesta adaptativa al cambio como pilar del marco del trabajo.

De acuerdo con Agilemanifesto.org (<http://agilemanifesto.org/>), el documento establece doce principios para el desarrollo de Software ágil los cuales se detallan a continuación:

1. La más alta prioridad es la satisfacción del cliente a través de la entrega continua de valor del producto.
2. Integrar cambios en los requerimientos, aun cuando estos se presenten tarde en el ciclo de desarrollo del producto, para entregar una ventaja competitiva al cliente.
3. Entregar productos de software funcionales de manera frecuente.

4. La gente del negocio y los desarrolladores de software deben trabajar juntos diariamente durante el proyecto.
5. Construir proyectos alrededor de individuos motivados, otorgar el apoyo que necesitan y confiar en que cumplirán con su trabajo.
6. La forma más eficiente y efectiva de transferir información entre miembros del equipo es cara a cara.
7. Software funcional es la principal métrica de progreso.
8. Los procesos ágiles promueven el desarrollo sostenible, otorgando la capacidad del equipo, clientes y patrocinadores de trabajar a un ritmo constante de manera indefinida.
9. La atención constante a la excelencia técnica promueve la agilidad.
10. La simplicidad es esencial.
11. Las mejores arquitecturas, requerimientos y diseños emergen de equipos auto organizados.
12. El equipo debe reflexionar cómo volverse más efectivos en intervalos de tiempo regulares para luego ajustar su comportamiento acorde a las conclusiones.

Scrum

Según SCRUMStudy (2013) Scrum se define como “un marco de trabajo adaptativo, iterativo, rápido, flexible y efectivo, diseñado para entregar valor de manera rápida y significativa a través de un proyecto” (p. 2); de tal forma, Scrum no es una metodología sino un conjunto de herramientas, artefactos y técnicas que permiten a un equipo la aplicación de los principios de Scrum para crear un producto u ofrecer un servicio.

Roles de Scrum

1. Product Owner (PO): es el responsable de obtener el máximo valor posible del proyecto, articulando los requerimientos del cliente y manteniendo la justificación del negocio de acuerdo con SCRUMStudy (2013) (p. 41) de tal forma que este rol puede considerarse con el vínculo entre el equipo de trabajo y las necesidades del cliente.
2. Scrum Master (SM): se asegura que el equipo tenga todas las herramientas necesarias para completar el producto ya sean recursos, capacitaciones, competencias prácticas o limpiando impedimentos que pudieran afectar al equipo y de acuerdo a SCRUMStudy (2013) “El trabajo de un Scrum Master es el de facilitador y él o ella se encuentra en el mismo nivel jerárquico que los demás miembros del equipo” (p. 41).
3. Scrum Team o equipo de Scrum (ST): son los responsables de convertir los requerimientos del product owner y las herramientas del scrum master en un producto exitoso que aporte valor al negocio del cliente.

4. Stakeholders o interesados: son el colectivo que interactúa con el Product Owner, el Scrum Master y el equipo. Este colectivo tiene un grado de interés en el éxito del proyecto y sus opiniones deben de ser tomadas en cuenta durante el ciclo de desarrollo del proyecto. Los stakeholders pueden ser clientes, usuarios, patrocinadores, etc.

Principios de Scrum

Scrum cuenta con 6 principios, los cuales son guías que deben seguirse durante todo el proyecto, sin importar el tipo de entregable o el giro de la empresa y deben ser implementados para asegurar la correcta aplicación de Scrum según SCRUMStudy (2013) (p. 21) por lo cual la omisión o inadecuada adhesión a uno de ellos invalida la implementación del marco de trabajo en la organización. A continuación, se detallan los 6 principios:

1. Control de proceso empírico: Las decisiones dentro de un proyecto deben realizarse en base a la observación y experimentación en detrimento de planificación anticipada, este principio recae en las ideas de la transparencia, la inspección y la adaptación.
2. Auto organización: Los miembros del equipo deben ser automotivados y deben buscar responsabilidades dentro de la jerarquía de la organización, el trabajo en conjunto debe de estar alineado con la visión del proyecto.
3. Colaboración: Scrum fomenta la comunicación no solo entre los miembros del equipo sino entre stakeholders, product owners, scrum masters, etc. con el objetivo de que todos los roles puedan aportar ideas que produzcan un producto superior a la suma de sus partes. Para alcanzar dicha colaboración los individuos del equipo deben de estar al tanto

del trabajo realizado por los demás miembros, articular las tareas en unidades más pequeñas para ser reintegradas posteriormente y adaptar tecnologías e ideas que puedan ser útiles durante el desarrollo del producto.

4. Priorización basada en valor: Uno de los objetivos principales de Scrum es el de entregar la mayor cantidad de valor en el menor tiempo posible por lo que las tareas que componen el proyecto deben de ser priorizadas de acuerdo a cuánto valor entregan al cliente.
5. Time-boxing: El término puede definirse como la determinación de la cantidad fija de tiempo que un proceso o actividad que Scrum lleva a cabo. entre los que se encuentran los sprints, daily standup meetings, sprint planning meeting, sprint review meeting, retrospective sprint meeting.
6. Desarrollo iterativo: Permite la corrección de curso a medida que todos los actores involucrados logran un mayor entendimiento de los requerimientos del proyecto y la posibilidad de volver a etapas tempranas de la planificación y el desarrollo para aplicar dicho conocimiento.

Ceremonias Scrum

También conocidos como eventos Scrum, son cinco reuniones en las cuales los miembros del equipo Scrum llevan a cabo un control de los desarrollos en curso y adaptan los artefactos Scrum según sea necesario y de acuerdo con Schwaber y Sutherland. (2020) “Estos eventos están específicamente diseñados para permitir la transparencia requerida. El fracaso al operar

cualquiera de los eventos conlleva a la pérdida de oportunidades para inspeccionar y adaptarse.” [Comentario en la página web *Scrum Guide*]. Las características de las ceremonias Scrum se presentan a continuación.

Sprint. En Scrum, el trabajo a realizar se divide en ciclos o iteraciones, estos ciclos son conocidos como Sprints. La duración de un Sprint es determinada por los miembros del equipo, habitualmente estos duran entre 1 a 4 semanas. Cada Sprint tiene un objetivo que es el que se pretende alcanzar en dicho ciclo.

Sprint Planning. Esta reunión es la que marca el punto de partida de un Sprint, en ella se reúnen todos los miembros del equipo de trabajo junto con el Product Owner para seleccionar las actividades que se habrán de llevar a cabo en el presente Sprint, estas actividades se eligen del listado de ítems del Product Backlog.

Para determinar qué actividades se habrán de realizar, el Product Owner se encarga de indicar el objetivo del presente Sprint, y se seleccionan aquellos ítems del backlog que vayan en consonancia con el dicho objetivo, cabe destacar que en esta selección influye grandemente el grado de prioridad que tienen los ítems del backlog. Además, en esta reunión se define el rol y las tareas que habrá de desarrollar cada miembro del equipo.

Daily Scrum. Son reuniones cortas realizadas con periodicidad diaria en donde cada miembro del equipo brinda un breve resumen de las actividades que realizó el día anterior, las dificultades que tuvo para llevarlas a cabo y qué actividades habrá de realizar ese día. El Scrum Master funge como moderador y se asegura que la reunión sea corta, concisa y que cada

miembro del equipo participe. La duración habitual de esta reunión no debe de ser mayor a 15 minutos.

Sprint Review. Es una reunión la cual se lleva a cabo al finalizar el Sprint, a esta reunión asisten el equipo de desarrollo, el Scrum Master, el Product Owner y los stakeholders del proyecto. En esta reunión el equipo de desarrollo presenta las nuevas funcionalidades desarrolladas, estas funcionalidades son conocidas como incremento del proyecto. En esta reunión se determina si se alcanzó el objetivo del Sprint, y además, se ajusta el Product Backlog para reflejar los items que ya fueron desarrollados.

Sprint Retrospective. También conocida como reunión retrospectiva, al igual que el Sprint Review, esta reunión se realiza al finalizar el Sprint, pero difiere en los participantes y en el objetivo de su realización. A esta reunión asisten el equipo de desarrollo, el Scrum Master y el Product Owner, la finalidad de esta reunión es que el equipo haga un análisis de sus esfuerzos e identifique oportunidades de mejora que pueden ser adoptadas en los siguientes sprints, además de determinar que se ha aprendido en el Sprint que finaliza.

Artefactos Scrum

Los artefactos Scrum representan el trabajo a realizar o el valor entregado en un proyecto, entiéndase “valor” como los entregables que se le hacen llegar al cliente, dicho entregable debe de tener la característica de ser utilizable. Las características de los artefactos Scrum se presentan a continuación.

Product Backlog. Comúnmente conocido como Backlog, es el listado de funcionalidades o requerimientos que se pretende desarrollar en un proyecto, los ítems de ese listado son habitualmente representados a través de historias de usuario, además, los ítems de este listado son categorizados por prioridad, esto con la finalidad de dar una ponderación de importancia a cada ítem. Ítems con un alto grado de prioridad, deben de ser atendidos con mayor importancia o urgencia.

Sprint Backlog. El Sprint backlog está compuesto por el listado de actividades que habrán de llevarse a cabo en un sprint, estas actividades deben de estar alineadas con la consecución del objetivo del sprint en curso. La finalidad de las actividades contenidas en el Sprint backlog es generar un entregable (incremento) en base al objetivo del sprint.

Cualquier miembro del equipo Scrum puede acceder y modificar el contenido de este artefacto, esto debido a que dicho artefacto se maneja en tiempo real pues se actualiza constantemente a medida que se progresa en las actividades que lo componen.

Incremento. Un incremento es un entregable el cual aporta hacia la consecución del objetivo de un proyecto, cada incremento se suma a los incrementos previos ya desarrollados. Cada vez que se genera un incremento, se debe de validar que este se integra correctamente con los incrementos previos.

Se puede crear más de un incremento durante un sprint, habitualmente la entrega de dichos incrementos se realiza al finalizar el sprint durante la realización de la ceremonia Sprint

review, aunque hay que considerar que un incremento puede ser entregado antes de finalizar el sprint en curso, si el cliente o el equipo scrum así lo consideran conveniente.

Herramientas Utilizadas en Scrum

Historias de Usuario. Es una forma de presentar las funcionalidades que un usuario final espera de una parte de un sistema de software a desarrollar, su forma habitual de representación es a través de pequeñas tarjetas físicas o digitales en donde se escribe en lenguaje sencillo una o más funcionalidades que se espera que posea el sistema de software. Se espera que todas las funcionalidades de un proyecto sean representadas a través de historias de usuario.

Historias Épicas. Son historias de usuario demasiado grandes debido a la naturaleza de la funcionalidad que se espera desarrollar, por ello, las historias épicas suelen ser divididas en historias de usuario más pequeñas.

Scrum Board. Esta herramienta es utilizada para llevar un control visual de las actividades que está realizando el equipo de trabajo. Scrum Board es un tablero dividido por estados, en donde en cada estado se colocan las actividades que en ese momento se encuentran en dicho estado. La manera de implementación de esta herramienta y los estados a utilizar dependen de la organización que haga uso de esta herramienta. Los estados que más habitualmente se manejan en esta herramienta son:

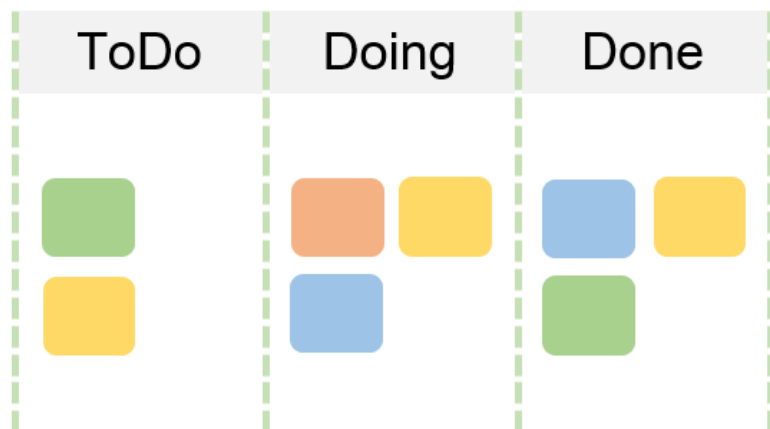
- To do: para determinar las actividades que aún falta por realizar.
- Doing: para determinar aquellas actividades que en ese momento se encuentran en desarrollo.

- Done: para determinar aquellas actividades que ya se finalizaron.

En la figura 2, se muestra un ejemplo de un Scrum Board en donde se utilizan los estados To do, Doing y Done.

Figura 2

Ejemplo de Scrum Board



Nota: El tablero está dividido en tres secciones, cada sección indica un estado, los cuadros de colores representan las actividades que se encuentran dentro de cada estado.

Adaptado de Kanban y dos herramientas para su gestión: Kanbanize y Flow-e, por Sergi Sales, 2017, IThink (<https://www.ithinkupc.com/es/blog/kanban-y-dos-herramientas-para-su-gestion-kanbanize-y-flow-e>). Obra de Dominio Público.

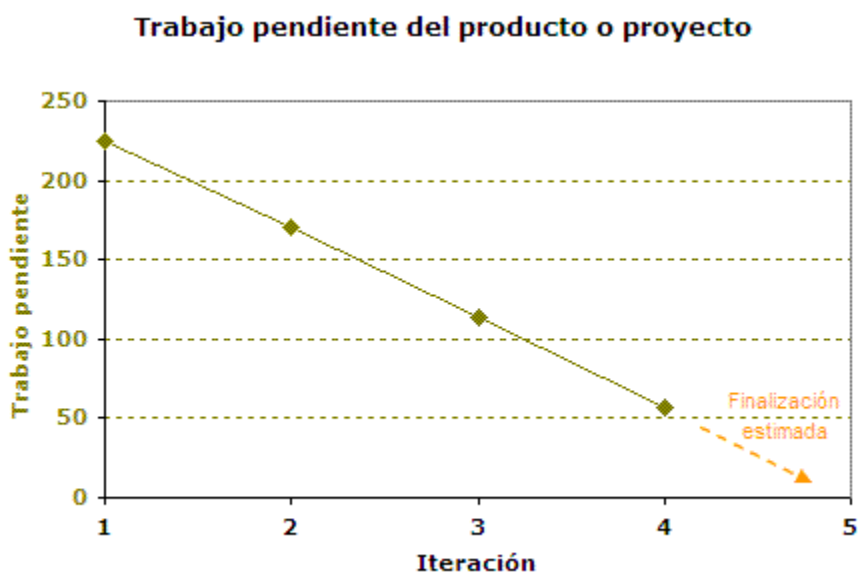
Burndown Chart. Es una herramienta que sirve para cuantificar el avance de un proyecto. Su forma de implementación es a través de un gráfico de dos dimensiones en donde se compara el trabajo pendiente contra el tiempo de desarrollo del proyecto. Cabe destacar que las

unidades de medida para trabajo pendiente y tiempo de desarrollo son determinadas según la información que se desee obtener a partir de dicho gráfico.

El encargado de actualizar este gráfico es el Scrum Master, esta tarea se realiza al finalizar cada Sprint. Gracias a este gráfico se pueden identificar posibles desviaciones referente a la planificación del proyecto. En la figura 3 se muestra un ejemplo de un Burndown chart en donde las unidades de medida son “Cantidad de trabajo pendiente vs Número de iteraciones”.

Figura 3

Ejemplo de Burndown Chart



Nota: El gráfico presenta a través de la línea verde la relación existente entre la cantidad de trabajo pendiente y el número de iteración. Adaptado de Gráficos de trabajo pendiente, por Proyectos Agiles, 2022, ProyectosAgiles (<https://proyectosagiles.org/graficos-trabajo-pendiente-burndown-charts/>). CC BY-SA 4.0

Registro de Deuda Técnica. Una deuda técnica son todas aquellos errores o deficiencias deliberadas que se realizan cuando un proyecto es realizado con prisa o con pocos recursos, estos elementos habitualmente se documentan en un registro de deuda técnica. Si dentro de un proyecto, no se solventa la deuda técnica, esta puede seguir creciendo hasta el punto de no ser sostenible. Debido a que, en Scrum, los proyectos se desarrollan de manera iterativa, es posible realizar una revisión del proyecto y corregir los elementos de la deuda técnica.

Log de impedimentos. Es un listado que es actualizado regularmente por el Scrum Master, en este listado se anotan todos aquellos impedimentos que frenan o ralentizan el desarrollo normal del Sprint. Además, es tarea del Scrum Master buscar posibles soluciones a los impedimentos identificados.

Procesos de Scrum

Son 19 actividades que componen el flujo de desarrollo de un producto bajo el marco de trabajo Scrum y son de carácter obligatorio sin importar el tipo de producto a entregar, estas actividades están divididas en 5 fases para facilitar su implementación. Las actividades que componen los procesos de Scrum se presentan en la tabla 1.

Tabla 1

Actividades que componen los procesos de Scrum

Símbolo	Etapas	Procesos
I	Iniciar	<ol style="list-style-type: none"> 1. Crear la visión del proyecto 2. Identificar al Scrum Master y los Stakeholders

-
- | | | |
|-----|-------------------|--|
| | | 3. Formar el equipo de Scrum |
| | | 4. Desarrollar las historias épicas |
| | | 5. Crear y priorizar el backlog del producto |
| | | 6. Llevar a cabo la planeación de la liberación del producto |
| II | Planear y estimar | 7. Crear historias de usuario |
| | | 8. Estimar historias de usuario |
| | | 9. Establecer historias de usuario |
| | | 10. Identificar tareas |
| | | 11. Estimar tareas |
| | | 12. Crear los backlogs de los sprints |
| III | Implementar | 13. Crear entregables |
| | | 14. Conducir la reunión Daily Scrum |
| | | 15. Refinar el backlog del producto |
| IV | Revisar | 16. Demostrar y validar el sprint |
| | | 17. Retrospectiva del sprint |
| V | Liberar | 18. Liberar entregables |
| | | 19. Reunión retrospectiva del proyecto |

Nota. Reproducido de *A guide to the scrum body of knowledge (Sbok guide)* (3rd ed., p 15) por SCRUMstudy.

Responsabilidades de los Actores en los Procesos de Scrum

Product Owner. Según SCRUMStudy (2013) “El product owner debe mantener una visión dual. Él o ella deben comprender y apoyar los intereses y necesidades de los stakeholders, mientras entiende las necesidades del SCRUM Team” (p.44) Dichas responsabilidades se muestran en la tabla 2:

Tabla 2

Responsabilidades del product owner.

Proceso	Responsabilidad
I-1	<ul style="list-style-type: none"> - Define la visión del proyecto - Ayuda a crear el presupuesto del proyecto
I-2	<ul style="list-style-type: none"> - Ayuda a seleccionar al SM - Identificar Stakeholders
I-3	<ul style="list-style-type: none"> - Ayuda a seleccionar miembros de ST - Ayuda a desarrollar el plan de colaboración - Ayuda a desarrollar el plan para la selección de ST
I-4	<ul style="list-style-type: none"> - Crea las historias épicas
I-5	<ul style="list-style-type: none"> - Prioriza los elementos del product backlog - Define el criterio de “Terminado”
I-6	<ul style="list-style-type: none"> - Crea el calendario de liberación - Determina la longitud de los sprint

II-7	- Ayuda a crear las historias de usuario
	- Define el criterio de aceptación de una historia de usuario
II-8	- Clarifica las historias de usuario
II-9	- Ayuda a completar las historias de usuario
II-10	- Explica las historias de usuario al ST mientras se realiza la lista de tareas
II-11	- Guía y clarifica al ST en la estimación del esfuerzo por tarea
II-12	- Clarifica los requerimientos del ST al crear el backlog del sprint
III-13	- Clarifica las necesidades del negocio al ST
III-15	- Refina el backlog del producto
IV-16	- Acepta o rechaza los entregables
	- Aporta retroalimentación a PO y ST
	- Actualiza el backlog del producto y el plan de liberación
V-19	- Ayuda en el despliegue de los entregables
V-20	- Participa en las reuniones de retrospectivas

Scrum Master. Según SCRUMStudy (2013) “El Scrum Master es un ‘líder sirviente’ del Scrum Team que modera y facilita las interacciones como coach y motivador del equipo” (p.47)

Las responsabilidades del Scrum Master se detallan en la tabla 3:

Tabla 3

Responsabilidades del scrum master

Proceso	Responsabilidad
I-2	- Ayuda a identificar a los Stakeholders del proyecto
I-3	- Facilita la selección de ST - Facilita la creación del plan de colaboración y el plan de selección del ST - Se asegura que el equipo cuente con recursos de respaldo
I-4	- Facilita la creación de historias épicas
I-5	- Ayuda al PO en la creación del backlog del producto y la definición de “Terminado”
I-6	- Coordina la creación del plan de liberación - Determina la duración del sprint
II-7	- Asiste al ST en la creación de las historias de usuario
II-8	- Facilita las reuniones del ST para estimar las historias de usuario
II-9	- Facilita las reuniones del ST para completar las historias de usuario
II-10	- Facilita las reuniones del ST para identificar las tareas para el siguiente sprint
II-11	- Asiste al ST en estimar el esfuerzo requerido para la finalización de las tareas
II-12	- Asiste al ST en el desarrollo del backlog del sprint y en la creación del

Sprint Burndown Chart

- III-13 - Apoya al ST en la creación de los entregables acordados en el sprint
 - Actualiza el log de impedimentos y el Scrum Board
 - III-14 - Actualiza el log de impedimentos y el Scrum Board
 - III-15 - Facilita las reuniones para la revisión del backlog
 - IV-16 - Facilita la presentación de los entregables terminados por el ST para la aprobación del PO
 - IV-17 - Facilita las condiciones que permitan al ST tener éxito en sus sprints
 - V-19 - Representa al ST para proveer las lecciones aprendidas durante el proyecto
-

Scrum Team. Según SCRUMStudy (2013):

El Scrum Team es a veces referido como el equipo de desarrollo porque son los responsables de desarrollar el producto, servicio u otro resultado. Consiste en un grupo de individuos que trabajan en las historias de usuario en el Backlog de cada Sprint para crear los entregables del proyecto (p.50).

Las responsabilidades del Scrum Team se detallan en la tabla 4:

Tabla 4*Responsabilidades del scrum team*

Proceso	Responsabilidad
I-3	- Aporta opiniones en la creación del plan de colaboración
I-4	- Se asegura de comprender el objetivo de las historias épicas creadas
I-5	- Se asegura de entender las historias de usuario y el backlog del producto
I-6	- Acuerda la duración del sprint - Busca la clarificación de nuevos productos o cambios en productos existentes en el backlog del producto
II-7	- Asiste y provee opiniones al PO en la creación de las historias de usuario
II-8	- Estima las historias de usuario aprobadas por el Product Owner
II-9	- Ayuda a determinar las historias de usuario a incluirse en el sprint
II-10	- Desarrolla la lista de tareas basado en las historias de usuario acordadas.
II-11	- Estima las tareas acordadas.
II-12	- Desarrolla el backlog del sprint y el Sprint Burndown Chart.
III-13	- Crea entregables. - Identifica riesgos y los mitiga de ser posible - Actualiza el log de impedimentos y sus dependencias
III-14	- Actualiza el log de impedimentos, el Scrum Board y el Burndown Chart - Discute problemas sucedidos a miembros individuales del equipo y busca

	soluciones a dichos problemas
	- Identifica riesgos
	- Crea solicitudes de cambios
III-15	- Participa en reuniones de priorización del backlog
IV-16	- Presenta entregables al PO para su aprobación
IV-17	- Identifica oportunidades de mejora en el sprint actual para mejorar el sprint siguiente
V-19	- Participa en la reunión de retrospectiva

Esfuerzos Previos por Mejorar las Metodologías Ágiles

Antes de la existencia de Agile 2, hubo otros esfuerzos por mejorar o revolucionar las metodologías ágiles para que estas fueran de mayor utilidad dentro del área del desarrollo de software. Estos esfuerzos se generaron debido a que hubo grupos de desarrolladores que consideraron que las metodologías ágiles tienen un alto enfoque en habilidades blandas en lugar de centrarse en conocimientos técnicos. El más conocido de estos esfuerzos es la metodología DevOps.

DevOps

DevOps nació como una iniciativa de los desarrolladores de software Patrick Debois y Andrew Clay Shafer, ellos denotaron los inconvenientes existentes en el desarrollo de software debido a la división que hay entre el área de desarrollo (devs) y el área de operaciones (ops).

El área desarrollo está compuesta por los desarrolladores de software, es decir, quienes codifican los proyectos de software; mientras que el área de operaciones está compuesta por aquellos trabajadores que se encargan de gestionar el ambiente que rodea a los productos en desarrollo, dicho ambiente incluye la compra y configuración de software, contratación y mantenimiento de servidores, gestión de la seguridad de cada producto desarrollado, creación de planes de recuperación, control de cambios, gestión de la red, entre otras tareas.

Ellos denotaron que los desarrolladores a menudo velaban por desarrollar software de manera rápida adecuándose a las necesidades de los clientes y del mercado, mientras que el área de operaciones velaba por ofrecerle a los clientes productos que fueran seguros y estables. A menudo el diferente enfoque de ambas áreas chocaba, dificultando el desarrollo de productos de software.

Es por ello que, en el año 2009, Debois y Shafer lanzaron la metodología DevOps, la cual tiene como finalidad integrar las áreas de desarrollo de software y de operaciones, para que estas trabajen en conjunto remando hacia una misma dirección.

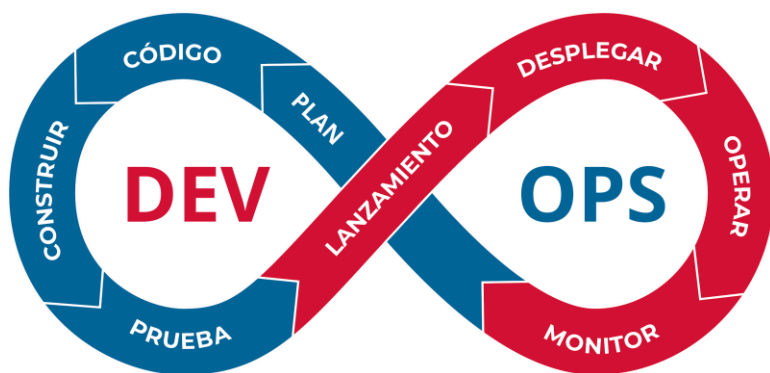
Con esta metodología se buscó dar un fuerte enfoque al desarrollo de software, buscando generar productos de manera rápida y eficaz, para ello se auxiliaron de técnicas ya existentes en el mercado como lo es el automatizar tareas para reducir los tiempos de producción y el manejo de diversos ambientes para pruebas o desarrollos.

Bajo la metodología DevOps, el desarrollo de un producto se trabaja de manera cíclica, es decir, que un proyecto no se considera como finalizado una vez que este ha sido lanzado, sino

que, a partir de ese momento el proyecto ingresa en un proceso de mejora continua en donde este es monitoreado para identificar, planificar, codificar y lanzar nuevas funcionalidades o correcciones que mejoren dicho proyecto. Las fases del ciclo de vida de un proyecto desarrollado bajo la metodología DevOps se distribuyen entre los miembros del área de desarrollo y del área de operaciones. La distribución de fases entre ambos equipos se puede visualizar a través de la figura 3, presentada a continuación:

Figura 3

Resumen del ciclo de vida de un proyecto bajo la metodología DevOps



Nota. La figura presenta las fases del desarrollo de un proyecto utilizando la metodología DevOps, las flechas azules muestran las fases realizadas por el área de desarrollo (dev), mientras que las flechas rojas representan las fases realizadas por el área de operaciones (ops). Adaptado de DevOps con AWS, por SoftwareOne, 2022, SoftwareOne (<https://www.softwareone.com/es-pa/solutions/publisher-advisory-services/aws/devops-with-aws>). Obra de Dominio público.

Para DevOps, es fundamental el lanzamiento periódico de nuevas versiones de un producto, estos lanzamientos deben de ser realizados tan a menudo como sea posible. para ello,

DevOps utiliza 3 procesos los cuales son: integración continua, entrega continua y lanzamiento continuo, sus características se detallan a continuación:

Integración Continua. Cuando un desarrollador ha generado nuevo código para añadirlo a un producto, dicho código es añadido a una rama (una versión del proyecto) de un ambiente de pruebas. A partir de esta rama se crea una compilación y está se pone a prueba a través de tests automatizados. Los tests realizados son: test unitarios para validar de manera individual cada componente del código generado, test de integración para validar el correcto acoplamiento del nuevo código con los componentes ya existentes del proyecto y test de aceptación o de aprobación para validar que el código cumple con el propósito para el cual fue creado.

Del proceso de integración continua, se puede proseguir ya sea al proceso de entrega continua o al proceso de lanzamiento continuo, por lo que, el proceso de integración es el punto de partida del cual se pueden ejecutar cualquiera de los otros dos procesos, queda a consideración del implementador el elegir cuál de los procesos disponibles utilizar.

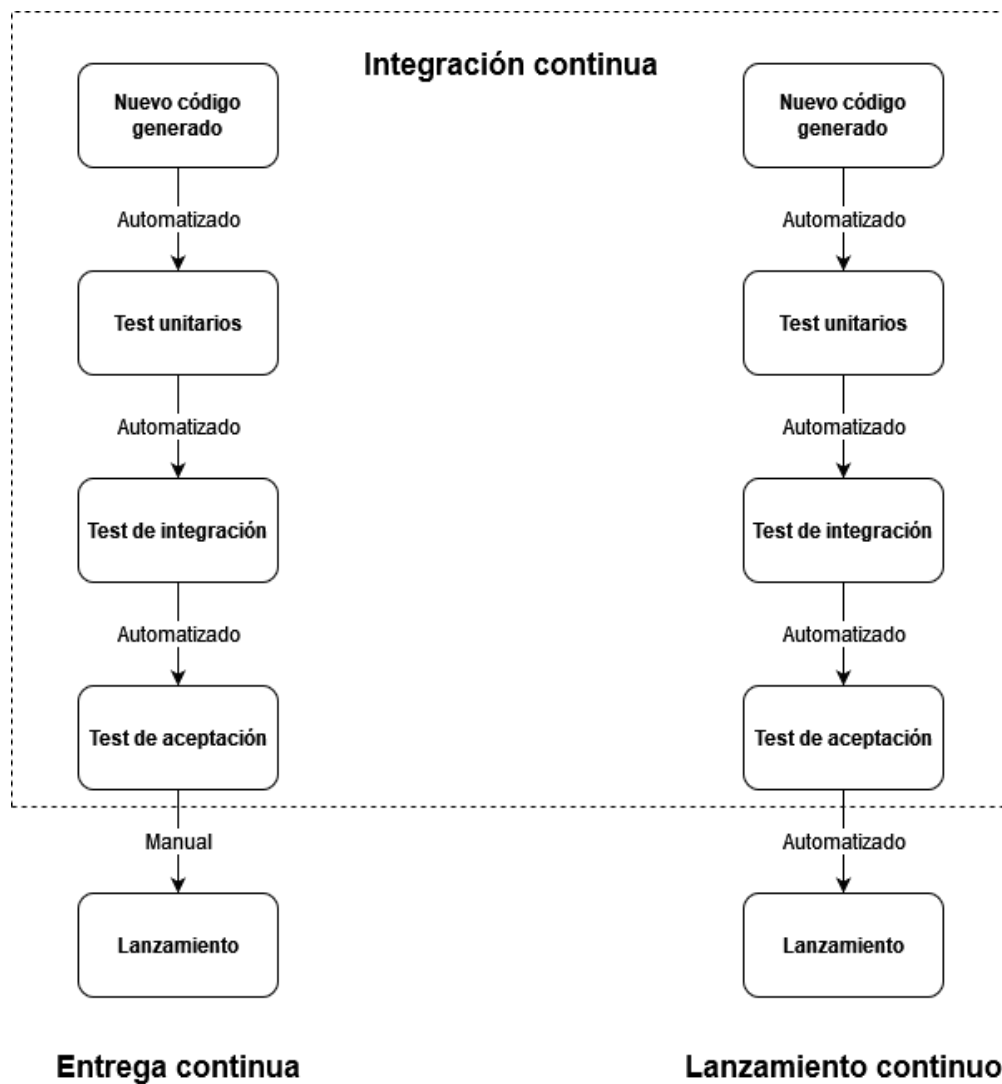
Entrega Continua. Una vez que los tests automatizados del proceso de integración continua han sido aprobados, se pasa a la fase de entrega continua en donde el código aprobado es lanzado al ambiente correspondiente, como pueden ser otros ambientes de desarrollo, ambiente de QA, ambiente de producción, etc. Este proceso es auxiliado por la figura de un administrador de lanzamientos, el cual determina si el código aprobado es válido para ser lanzado.

Lanzamiento continuo. Este proceso es más drástico que el proceso de entrega continua ya que tras la realización de los tests automatizados, el código aprobado es publicado directamente en la versión de producción del proyecto, sin la necesidad de tener la aprobación de un administrador de lanzamientos. Este proceso es más rápido que el proceso de entrega continua ya que no posee tantas validaciones para aprobar el lanzamiento de una nueva versión del proyecto, pero es más riesgosa debido a que no pasa por un proceso de validación humana.

A continuación, en la figura 4, se puede visualizar un diagrama en donde se muestra el orden secuencial de las actividades que componen los 3 procesos mencionados.

Figura 4

Lanzamiento de nuevas versiones de un producto bajo la metodología DevOps



Nota: El texto en las flechas indica si la actividad que le prosigue es realizada de manera automatizada o de manera manual. La línea punteada indica las actividades que pertenecen al proceso de integración continua.

Agile 2

Agile 2 es una metodología que pretende ser una revolución dentro de las metodologías ágiles tradicionales. Su principal creador es Cliff Berg, un consultor especializado en fusionar metodologías ágiles con DevOps. Su principal fundador es Cliff Berg (2021), quien consideraba que las metodologías ágiles se han convertido en algo dogmático, es decir, que quienes adoptan estas metodologías siguen sus prácticas al pie de la letra sin considerar el ambiente o las circunstancias bajo las cuales se está adoptando dichas metodologías (p. 6).

En 2014, Cliff Berg reunió a un grupo de expertos en metodologías ágiles para crear una nueva metodología que fuera rompedora respecto al estado actual de las metodologías ágiles, fueron 15 las personas que contribuyeron en el desarrollo de una nueva metodología a la cual llamaron Agile 2, su desarrollo llevó cerca de 5 años de trabajo, hasta su lanzamiento a finales de octubre de 2019.

Uno de los principales preceptos de Agile 2 es la palabra “Depende”, esto se puede interpretar como que cada proceso dentro de una organización no tiene un camino único a través del cual puede llevarse a cabo, sino que las alternativas de las que se dispone “dependen” del ambiente o las circunstancias que rodean el proceso en cuestión.

Cabe destacar, que Agile 2 tiene un fuerte enfoque al área de desarrollo de software, lo cual la diferencia de otras metodologías ágiles que si bien, fueron concebidas como metodologías para ser usadas por personal con conocimiento técnico, con el paso del tiempo han ido mutando

para poder ser adoptadas por áreas de trabajo de diversa índole, y que no necesariamente son áreas técnicas.

Agile 2 busca ser flexible en cuanto a las soluciones o prácticas que brinda, dicha metodología busca fomentar el pensamiento reflexivo dentro de sus usuarios, esto lo hace brindando una serie de valores y principios para que las organizaciones que decidan adoptar esta metodología tengan una guía y fundamentos que ayuden en la mejora de sus procesos internos.

Algo muy importante a resaltar sobre la metodología Agile 2, es que esta no pretende venir a desechar todas las ideas ya plasmadas en otras metodologías ágiles, sino que Agile 2 pretende complementar o mejorar dichas ideas, de manera que estas sean más efectivas para el área de desarrollo de software.

Valores

Los valores de Agile 2 son las cualidades bajo las que se rige la metodología, estos siempre se declaran en pares, debido a que se busca que los valores que componen cada par sean equilibrados o que se combinen para afrontar situaciones concretas. A continuación, se presentan los valores de Agile 2:

1. **Consideración y prescripción.** Con la consideración se pretende que, al afrontar una situación, primero se debe de analizar y comprender el contexto de esta y una vez realizada esta tarea, ya se pueden implementar medidas para satisfacer dicha situación, mientras que con la prescripción se busca que existan pasos o medidas ya definidas a implementar ante una situación, sin analizar su contexto.

2. **Resultados y salidas.** Se debe de entender la diferencia entre resultados y salidas, los resultados son todo aquello que se genera directa e indirectamente al realizar una acción, por ejemplo, un resultado directo puede ser un producto mientras que un resultado indirecto puede ser el conocimiento que el equipo de trabajo adquirió al realizar una acción. Por su parte, las salidas solo hacen referencia a lo que se genera directamente al realizar una acción, es decir, solo se refiere a los productos como tal. Cabe destacar que los resultados tienen mayor valor que las salidas.
3. **Individuos y equipos.** Se debe de entender que una persona dentro de una organización es importante, a su vez, esta persona debe velar por el beneficio del equipo al que pertenece. Hay que destacar que tanto los intereses de la persona como los intereses del equipo al que pertenecen deben de estar en equilibrio, uno no es más importante que el otro.
4. **Entender el negocio y la tecnología.** En una organización, el personal del área de negocios debe de tener dominio de conocimientos técnicos al igual que el personal del área técnica debe de tener dominio de la lógica del negocio.
5. **Empoderamiento individual y buen liderazgo.** Se debe otorgar a los miembros del equipo la capacidad de decidir cómo llevar a cabo su propio trabajo, esto se realiza brindándoles libertad para innovar, equivocarse y expresar nuevas ideas, al hacer esto se fomenta el liderazgo de dichas personas. Por su parte, los líderes ya establecidos deben de medir cuánta libertad le pueden brindar al personal que está a su cargo.

6. **Adaptabilidad y planeación.** La adaptabilidad indica que se debe de esperar a que los planes de una situación pueden cambiar y hay que estar preparado para afrontar dichos cambios, mientras que la planeación indica que debe de haber una planificación que brinde una hoja de ruta para afrontar una situación.

Principios

Los principios de Agile 2 son una serie de prácticas sugeridas para la implementación de la metodología. Estos están divididos en 10 categorías, cada una haciendo referencia a un área en específico. A continuación, se presentan los principios de Agile 2 divididos por categoría:

1. Planear, transicionar y transformar
 - Toda iniciativa necesita una visión y un objetivo, así como un plan que sea flexible, manejable y orientado a resultados.
 - Todas las transformaciones significativas (cambios o adopciones de metodologías o marcos de trabajo) son un camino de aprendizaje, no solo un proceso de cambio.
 - El cambio ocurre desde arriba (desde los líderes de la organización).
 - El desarrollo de productos va más allá de una implementación, también es un camino de aprendizaje.
2. Productos, portafolios y stakeholders
 - De manera continua se debe de obtener retroalimentación tanto de stakeholders como del mercado.

- La única manera de probar el éxito de un producto es a través de la retroalimentación por parte del cliente y de los stakeholders.
- Se debe de trabajar iterativamente en pequeñas fases o iteraciones.
- El diseño y la implementación de productos deben de integrarse.
- Toda documentación creada debe poder ser compartida y comprensible para quien la lea.
- Quienes ofrezcan productos y servicios deben de ser responsables por el impacto que puedan ocasionar los defectos que estos posean.

3. Datos

- Los datos poseen valor estratégico.
- El modelo de información de la organización es estratégico.
- La información recolectada debe ser utilizada para validar el producto desarrollado.

4. Marcos de trabajo y metodologías

- Se debe buscar un marco de trabajo ágil que encaje con la cultura y las circunstancias de la organización.
- Todas las organizaciones necesitan un marco de trabajo inicial que satisfaga sus necesidades.

5. Dimensión y fluidez técnica

- La agilidad técnica y la agilidad del negocio son inseparables.
- Los líderes del negocio deben comprender cómo sus productos y servicios son creados e implementados.

- Los líderes del área de desarrollo tecnológico deben comprender cómo funciona el desarrollo tecnológico.
- Los líderes y equipos de desarrollo tecnológico necesitan entender la lógica del negocio.

6. Individuos vs equipos

- Todo el equipo soluciona todo el problema, es decir, todo el equipo de trabajo se debe involucrar en buscar una manera de desarrollar los proyectos que tienen a cargo.
- Se debe de facilitar la diversidad de comunicación y la diversidad de estilos de trabajo.
- Las personas importan tanto como importa el equipo.
- Tanto especialistas como personas con conocimientos generales son importantes dentro de un equipo.
- Las diferentes certificaciones ágiles que existen tienen un valor desigual, por lo que requieren un escrutinio de su validez y valor.

7. Equipo vs. Organización

- Se debe favorecer la formación de equipos autónomos los cuales tengan autoridad para actuar.
- Se debe fomentar la colaboración entre equipos a través de objetivos compartidos.
- Se debe fomentar la creación de equipos estables (equipos de larga duración) y se debe convertir la experiencia de esos equipos en una ventaja competitiva.

8. Mejora continua

- Se debe de imponer límites en aspectos o prácticas que causen retrasos o lentitud.

- Las inconsistencias que pueden encontrarse en un proyecto deben de ser solventadas rápidamente, este proceso debe de hacerse frecuentemente.
- Cada cierto tiempo, el equipo debe reflexionar y proponer cambios que mejoren los procesos de la organización.
- Nunca se debe utilizar la capacidad máxima de trabajo de la organización, siempre se debe de dejar una porción de capacidad disponible, esto para realizar nuevas posibles asignaciones.

9. Enfoque

- Solo se debe interrumpir el flujo de trabajo de una persona cuando sea realmente necesario.
- Se debe facilitar que las personas puedan realizar su trabajo de manera enfocada y sin interrupciones.
- Se debe apoyar el realizar cambios profundos, en caso de que estos sean necesarios.

10. Liderazgo

- El paradigma de liderazgo es el factor más importante para el éxito de una organización.
- Se debe de proveer liderazgo que empodere tanto a individuos como a equipos y se debe de establecer la dirección del mismo.
- Una organización está compuesta por colecciones de equipos, por lo que, los problemas que afectan a un equipo afectarán a los demás equipos de su colección.

- Los modelos organizacionales y de liderazgo deben de tener la capacidad de evolucionar para adaptarse a cambios que permitan lograr un mejor rendimiento.
- Los buenos líderes son abiertos a escuchar la opinión o pensamientos de los demás.
- Un equipo a menudo necesita más de un líder, cada uno de un tipo diferente.
- La autogestión y autonomía son aspiraciones y deben otorgarse acorde a la capacidad del equipo de trabajo.
- Se debe de evaluar y validar ideas y sugerencias a través de pequeños experimentos.
- El desarrollo profesional de los individuos es esencial.

Diferencias entre la Metodología Agile 2 y las Metodologías Ágiles Tradicionales

Si bien, Agile 2 es una metodología ágil, esta presenta ciertas cualidades que la diferencian de las metodologías ágiles tradicionales, dichas diferencias se presentan a continuación:

- Agile 2 busca ser una reinención de las metodologías ágiles, más no busca reemplazarlas, esto lo hace complementando o mejorando procesos de las metodologías ágiles tradicionales.
- Agile 2 no brinda una hoja de ruta específica para su adopción, en su lugar brinda una serie de valores y principios que dictan las normas que una organización puede seguir para realizar la adopción.

- Agile 2 no es dogmático, esto significa que no pretende que sus valores y principios sean considerados como algo irrefutable e innegable, sino que estos deben de ser discernidos por el usuario para determinar su forma de implementación.
- El proceso de adopción de Agile 2 es un proceso prolongado, esto debido a que es un proceso de aprendizaje a partir de las características de la organización en donde se realiza la adopción. Cada organización es diferente y por ello, sus necesidades a suplir y elementos de la metodología a adoptar, también son diferentes.

Metodología

Alcance

El presente proyecto pretende ayudar en la implementación de la metodología Agile 2 en el Departamento de Sistemas Informáticos de la Universidad Don Bosco de El Salvador, dicho proceso será llevado a cabo durante el primer semestre del año 2022.

La población beneficiaria de dicho proyecto serán los miembros del Departamento de Sistemas Informáticos, el cual está conformado por 11 personas las cuales ostentan los siguientes cargos:

- Analista de sistemas.
- Arquitecto de Software.
- QA tester (Especialista en el aseguramiento de la calidad).
- Jefe del Departamento de Sistemas Informáticos.

Actualmente, la organización ha adoptado un marco de trabajo ágil tradicional, el cual es Scrum, dicha adopción se realizó en el año 2018. Ahora, 4 años más tarde la organización está abierta a adoptar una nueva metodología que les permita potenciar sus procesos de desarrollo de software hecho a la medida.

Limitaciones

- El área objetivo del proyecto se limitará al Departamento de Sistemas Informáticos de la Universidad Don Bosco, no se involucrará a ningún otro departamento de dicha Universidad.

- Se partirá de la base que los miembros de la organización ya tienen conocimientos previos sobre lo que son las metodologías ágiles, esto debido a que actualmente utilizan el marco de trabajo ágil Scrum.
- El proceso de adopción de la metodología Agile 2 espera llevarse a cabo durante el rango de fechas que comprende desde el 1 de enero de 2022 hasta el 30 de junio del mismo año.
- La recolección de información necesaria para el desarrollo del presente proyecto se realizará tanto de manera presencial en las instalaciones de la organización como a través de los medios digitales que se consideren adecuados.

Hoja de Ruta para la Recolección de Datos

Datos Necesarios por Recolectar

Para llevar a cabo los objetivos planteados en el presente proyecto, se necesita obtener la siguiente información:

- Manejo actual del marco de trabajo ágil Scrum por parte de la organización.
- Retos que tuvo la organización para adoptar Scrum.
- Oportunidades de mejora que poseen los procesos de desarrollo de software que actualmente ejecuta la organización.
- Conocimiento actual sobre metodologías ágiles por parte de los miembros de la organización.

- Características de la metodología Agile 2.
- Dudas y expectativas de los miembros de la organización al momento de adoptar una nueva metodología de trabajo.

Métodos de Recolección de Datos

Para obtener la información planteada, se hará uso de los siguientes instrumentos de recolección de datos:

Reuniones. Se realizará una reunión inicial con el jefe de la organización para conocer y determinar sus expectativas respecto a la implementación de un sistema de gestión utilizando la metodología Agile 2. Posteriormente, se realizarán reuniones de periodicidad mensual para obtener retroalimentación del jefe de la organización sobre el proceso de implementación del sistema de gestión.

Encuestas. Se realizarán las encuestas que se consideren necesarias para obtener la información necesaria que permita conocer la realidad actual de la organización en cuestión, estas encuestas se prevé realizarlas de manera digital a través de herramientas web que faciliten la realización de encuestas. Se espera poder encuestar a todos los miembros de la organización, esto para saber el punto de vista de cada uno de los miembros de nuestra población objetivo.

Entrevistas. Se entrevistará al jefe del Departamento de Sistemas Informáticos UDB para que nos brinde información detallada sobre cómo fue el proceso de adopción de Scrum en el año 2018 y cuáles son sus expectativas referentes a la adopción de una nueva metodología, dicha entrevista se realizará de manera presencial en las instalaciones de la organización objetivo.

Visitas presenciales. Se visitarán las instalaciones de la organización para observar y tomar nota de cómo transcurre un día normal de trabajo para cada uno de sus miembros, esto permitirá conocer de primera mano cuáles procesos o instrumentos se han adoptado del marco de trabajo ágil Scrum. Toda la información obtenida se irá recopilando a través de fichas de registro de observación, el cual es un instrumento que permite crear bitácoras de la información obtenida. Las visitas presenciales se programarán con el jefe de la organización para poder determinar el periodo de tiempo más adecuado para concertar las visitas.

Correos electrónicos. Se ha acordado que el correo electrónico será la forma de comunicación oficial con la organización, a través de este medio se podrá consultar a los miembros de la organización sobre dudas que se posean referentes al proyecto en curso.

Documentación Utilizada Internamente dentro de la Organización. Se solicitará a la organización que nos proporcionen documentación valiosa sobre la manera en que actualmente manejan sus procesos utilizando el marco de trabajo Scrum, esto incluye, definición de procesos internos, definición de su cultura organizacional, roles y responsabilidades de cada uno de los trabajadores, conjunto de buenas prácticas adoptadas y toda documentación que brinde información relevante al tema en cuestión.

Documentación sobre la Metodología Agile 2. Se buscarán medios fiables que brinden información relevante sobre los principios que constituyen a la metodología Agile 2, dichos medios pueden ser libros, revistas, guías desarrolladas por empresas que hayan adoptado

previamente dicha metodología, sitios web cuyas fuentes sean confiables, personas con experiencia en el área en cuestión, entre otros.

Calendarización de la Implementación de los Instrumentos de Recolección de Datos

En la tabla 5 se indica la manera en que se implementarán los instrumentos de recolección de datos:

Tabla 5

Calendarización de la implementación de los instrumentos de recolección de datos

Instrumento	Forma de Implementación	Fecha de implementación	Implementador	Personas evaluadas
Reunión	Presencial	Enero 2022 - junio 2022	Mario Menjivar y Joshua Villavicencio	Jefe de la organización
Encuesta	Digital	Enero 2022	Mario Menjivar	Todos los miembros de la organización
Entrevista	Presencial	Enero 2022	Joshua Villavicencio	Jefe de la organización

Visitas Presenciales	Presencial	Enero 2022	Mario Menjivar y Joshua Villavicencio	Todos los miembros de la organización
Correos Electrónicos	Digital	Enero 2022 – junio 2022	Mario Menjivar y Joshua Villavicencio	Jefe de la organización
Documentación interna	Digital e impresa	Enero 2022	Mario Menjivar y Joshua Villavicencio	Jefe de la organización
Documentación sobre metodología Agile 2	Digital, impresa o presencial	Enero 2022 – junio 2022	Mario Menjivar y Joshua Villavicencio	Investigación propia

Métodos de Validación de los Datos Obtenidos

Se deberá constatar la validez de la información obtenida, debido a que es brindada por miembros de la organización objetivo, no se puede comprobar la veracidad de la misma recolectada a través de fuentes externas, es por ello por lo que el instrumento de visitas presenciales servirá como medida de verificación pues este instrumento permitirá fungir como

observadores de las actividades diarias de la organización, lo que permitirá visualizar y determinar si la tal información es veraz.

Referente a la validación de información sobre la metodología Agile 2, se utilizará como medida de verificación la documentación oficial sobre ella, esta documentación está compuesta por el libro Agile 2: The next iteration of Agile (múltiples autores) y los valores y principios de Agile 2 que se pueden encontrar en el sitio web oficial de los creadores de la de la misma, el cual es agile2.net.

Proceso de Recolección de Datos

Los principales instrumentos utilizados para recabar la información necesaria para identificar la situación actual de la organización fueron la encuesta, entrevista y observación. La información que obtuvimos fue recabada durante el mes de enero de 2022; la población de la encuesta y del proceso de observación fueron todos los miembros del Departamento de Sistemas Informáticos UDB, mientras que para la entrevista se contó con la participación del jefe del departamento. A continuación, se presenta una síntesis de los datos obtenidos a través de los mencionados instrumentos de recolección de datos:

Encuesta

Para aplicar este instrumento de recolección de datos, se brindó a todos los miembros de la organización un formulario digital con preguntas que nos permitieron recolectar información relevante sobre sus interacciones actuales con metodologías ágiles.

A partir de la encuesta realizada, obtuvimos información valiosa respecto al panorama y a la situación actual de la organización. La totalidad de los miembros de la organización participaron en la encuesta lo que permite tener una variedad de opiniones y respuestas. Los insumos obtenidos a través de este instrumento de recolección pueden ser leídas en el Anexo A.

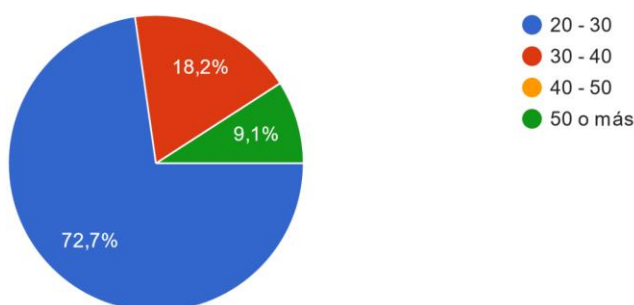
A continuación, se presenta una síntesis con los datos recolectados gracias a la encuesta.

Rango de Edad. Los encuestados son en su mayoría personas de entre 20 a 30 años. El miembro del equipo de mayor edad es el jefe del departamento, quien es mayor de 50 años. En la figura 5, se presentan gráficamente los rangos de edad de los miembros del equipo de trabajo, este gráfico se generó a partir de las respuestas obtenidas cuando se le consultó su edad a los miembros del equipo de trabajo.

Figura 5

Rango de edad de los encuestados

Rango de edad
11 respuestas



Nota: El rango de edad está expresado en años; los porcentajes indican la proporción de personas por cada rango de edad, la totalidad del porcentaje es 100.

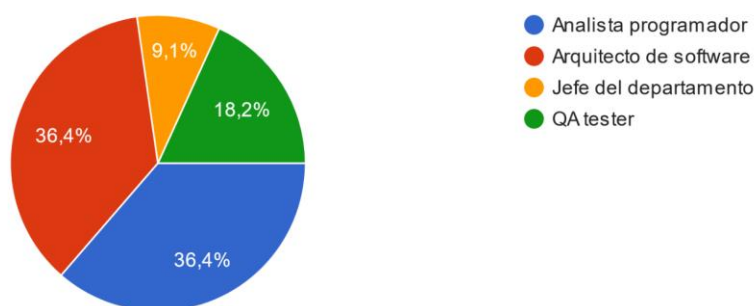
Cargo Desempeñado en la Organización. La mayor parte de los encuestados labora como desarrolladores de software, la creación de software es llevada a cabo tanto por los analistas de sistemas y por los arquitectos de software, siendo un 72.8% de ellos quienes realizan dichas labores; un 18.2% realizan labores de aseguramiento de la calidad fungiendo como QA testers mientras que el 9.1% restante realiza labores de jefatura. En la figura 6, se presenta de manera gráfica las respuestas obtenidas cuando se preguntó a los miembros del equipo de trabajo el cargo que desempeñan dentro de la organización.

Figura 6

Cargo desempeñado por los encuestados

Cargo que desempeña en SI UDB

11 respuestas



Nota: Los porcentajes indican la proporción de personas por cada cargo laboral, la totalidad del porcentaje es 100.

Instrumentos de Scrum que el Equipo de Trabajo Declara Utilizar. El marco de trabajo Scrum posee varios instrumentos para ser implementado, cada organización utiliza los elementos que considera convenientes en base a sus características o limitaciones, en síntesis, los encuestados declararon que dentro de la organización se utilizan los siguientes instrumentos de Scrum:

- Daily Scrum.
- Sprints.
- Historias de usuario.
- Scrum retrospective.
- Backlog de tareas.

Ventajas de Utilizar Scrum. Los encuestados brindaron su opinión respecto a las que consideran las ventajas que obtienen al utilizar el marco de trabajo ágil Scrum:

- Mejorar el trabajo en equipo lo cual ayuda a la productividad.
- Scrum está diseñado específicamente para la producción de software y considera las buenas prácticas obtenidas a partir de la experiencia del desarrollo de software.
- Permite trabajar un proyecto por fases, lo que permite realizar entregas pequeñas.

- El cliente puede estar involucrado en las diferentes etapas del proceso de desarrollo.
- Scrum es flexible ante posibles cambios en el desarrollo de un proyecto.
- Permite al equipo autogestionarse y compartir información sobre el estado del proyecto de manera más rápida
- Permite la centralización de un marco de trabajo alrededor de toda la empresa.
- Permite gestionar los riesgos de un proyecto

Desventajas de Utilizar Scrum. Si bien para los encuestados Scrum es un marco de trabajo que aporta una gran variedad de ventajas, también consideran que este posee las siguientes desventajas:

- La adopción de este marco de trabajo lleva tiempo para ser implementada de manera correcta, por lo que es necesario un equipo con experiencia para poder ser implementado, así mismo, esto conlleva a que en equipos grandes es un gran desafío incorporar este marco de trabajo ya que esto requiere parametrizar y controlar cada una de las funciones que estos desempeñan.
- Es difícil de utilizar en equipos grandes o cuando los proyectos son más complejos de lo que se esperaba. Las tareas deben definirse según el plazo de tiempo dado.
- Al momento de implementar Scrum todo el equipo debe conocer el marco de trabajo.

- La implementación inadecuada da pie a exceso de burocracia y actividades que no contribuyen directamente al proyecto, retrasando de esa manera la entrega de valor.
- Dificultad de implementación.
- Se puede caer en una mala definición de las historias de usuario.
- El equipo de trabajo puede llegar a tomarse más tiempo del recomendado en los daily scrum (reuniones cortas de periodicidad diaria).

Aspectos a Mejorar. A continuación, se presenta un listado con los aspectos que los encuestados consideran que deben de mejorarse dentro del Departamento de Sistemas Informáticos UDB:

- Mejorar el control de los tiempos de entrega.
- Mejorar la comunicación entre los equipos de trabajo, para saber el desarrollo actual de cada equipo de trabajo.
- Mejorar la obtención de requerimientos del cliente.
- Se debe de tener reuniones con el Product Owner antes de realizar el lanzamiento de un proyecto.
- La definición de requerimientos se debe de realizar acorde a los deseos del cliente y no a los del Product Owner.

- Se debe de refinar la manera en que se solicitan cambios en los proyectos.
- Se debe de crear un plan de desarrollo de software al inicio de cada proyecto.
- Cada desarrollo de software debe de ser publicado hasta que se hayan resuelto todos sus bugs (errores).
- Al realizar el lanzamiento de un producto, realizar una demostración de este a todos los involucrados.
- Adaptar la metodología ágil actualmente utilizada en base a las labores diarias.

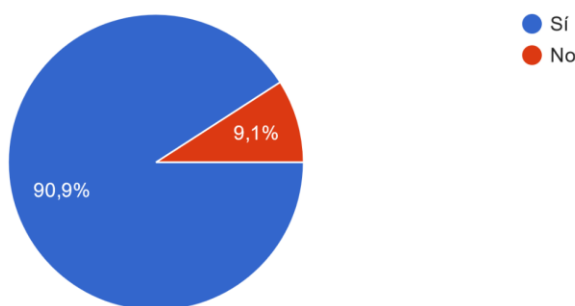
Cabe destacar que un 18.2% de los encuestados no consideran que haya oportunidades de mejora dentro del departamento.

Disposición a Adoptar una Nueva Metodología Ágil. Se consultó a los encuestados si estarían dispuestos a adoptar una nueva metodología ágil dentro de la organización, resultando en que un 90.9% de la población encuestada estaría abierta a esta posibilidad y solamente un 9.1% brindando una respuesta negativa. Esto denota que el equipo de trabajo si bien considera que Scrum brinda una gran variedad de ventajas, no se cierra a la posibilidad de adoptar una nueva metodología que venga a solventar aquellas desventajas de dicha metodología. En la figura 7, se muestra una representación gráfica de las respuestas obtenidas cuando se le consultó a los miembros del equipo de trabajo de la organización, su disposición a adoptar una nueva metodología ágil dentro del Departamento de Sistemas Informáticos UDB.

Figura 7*Disposición de adoptar una nueva metodología ágil*

¿Usted estaría abierto a adoptar una nueva metodología/framework ágil dentro de SI UDB?

11 respuestas



Nota: Los porcentajes indican la proporción de las respuestas brindadas, la totalidad del porcentaje es 100.

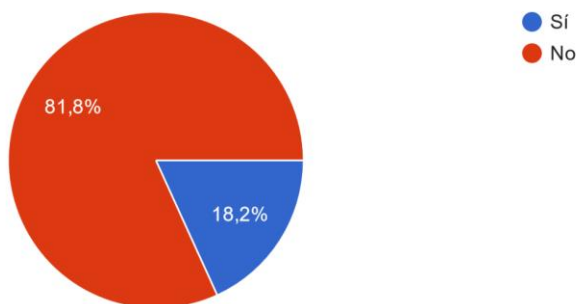
Conocimientos sobre Agile 2. Al adoptar una nueva metodología, es importante tener conocimientos previos sobre esta para saber de antemano los procesos o cambios que dicha adopción conlleva; se le consultó a los encuestados si conocen o han escuchado sobre la metodología Agile 2, un 81.8% respondió que desconocía la metodología mientras que el restante 18.2% tenía conocimiento de ella, la razón de dicho desconocimientos puede deberse a que Agile 2 es una metodología bastante reciente ya que fue lanzada a finales de 2019 y su popularidad aún se está formando. En la figura 8 se puede visualizar una gráfica que presenta las respuestas obtenidas cuando se les consultó a los miembros del equipo de trabajo de la organización si conocen o han escuchado sobre la metodología Agile 2.

Figura 8

Conocimiento previo de los encuestados sobre Agile 2

¿Conoce o ha escuchado sobre la metodología Agile 2?

11 respuestas



Nota: Los porcentajes indican la proporción de las respuestas brindadas, la totalidad del porcentaje es 100.

Dificultades al Adoptar una Nueva Metodología. Se consultó a los encuestados cuáles son las mayores dificultades a la hora de adoptar una nueva metodología ágil, a continuación, se presentan sus respuestas:

- Adaptación y resistencia al cambio.
- Gestionar y acoplar cada una de las actividades en el marco de trabajo de la nueva metodología en cada uno de los procesos de negocio que se realizan en el departamento.
- La refundación de procesos arraigados dentro de la cultura del equipo y el impacto inicial en el tiempo necesario para la generación de valor.

- Los cambios de mentalidad y paradigmas siempre son difíciles.
- Puede que no todos los involucrados tengan conocimientos sobre la nueva metodología a adoptar.
- Ser capaz de acoplarse de manera rápida a cómo se organizan los flujos de trabajos y demás actividades.

Entrevista

El método de la entrevista fue utilizado para obtener la opinión del jefe del Departamento de Sistemas Informáticos UDB, pues fue él, el principal impulsor de la adopción de Scrum en dicho departamento en el año 2018. Ahora, él está involucrado con la adopción de Agile 2, es por ello que su opinión es importante para este proyecto. Las preguntas y respuestas obtenidas a través de este instrumento de recolección pueden ser leídas en el Anexo C.

A continuación, se presenta una síntesis de la información recolectada gracias a la entrevista realizada:

- El director de CTIC y el jefe del Departamento de Sistemas Informáticos fungen como los Product Owner de los proyectos que se desarrollan.
- La estimación del tiempo de desarrollo de un proyecto se determina en base al criterio y experiencia de quien habrá de llevar a cabo dicho proyecto.
- No hay un proceso formal de adición de cambios a un proyecto en desarrollo.

- El jefe del Departamento de Sistemas Informáticos UDB es un entusiasta de las metodologías y marcos de trabajo ágiles, es por ello por lo que él impulsó la adopción de Scrum en el año 2018.
- El jefe del Departamento de Sistemas Informáticos UDB considera que la adopción de una metodología o marco de trabajo ágil debe realizarse de manera gradual, esto debido a que es habitual que las personas involucradas presenten resistencia al cambio.
- Al adoptar una metodología o marco de trabajo ágil, los jefes son quienes habitualmente más sufren el cambio debido a que estos no están acostumbrados a delegar responsabilidades a los demás miembros del equipo de trabajo, este aspecto puede ser interpretado por los jefes como una pérdida de poder.
- El jefe del Departamento de Sistemas UDB se muestra optimista y expectante ante la adopción de la metodología Agile 2, esto debido a que él tiene conocimientos previos sobre la metodología y le atrae por su enfoque hacia el área técnica del desarrollo de software.

Observación

Este método fue utilizado para validar que los datos obtenidos a través de la encuesta y la entrevista sean fidedignos, el Departamento de Sistemas Informáticos UDB nos permitió fungir como observadores durante el periodo del 24 al 27 de Enero de 2022, durante este lapso de tiempo pudimos observar los procesos habituales que realizan los miembros del departamento para cumplir con sus labores diarias, a partir de la observación, se recopilaron los hallazgos a

través de fichas de registro de observación. El formato de las fichas de observación y la información recopilada se puede encontrar en el Anexo D.

A continuación, se presenta una síntesis de los hallazgos realizados gracias al periodo de observación:

- El equipo diariamente realiza reuniones cortas llamadas Daily Scrum, estas reuniones se realizan por pequeños grupos de hasta 5 personas, en estas reuniones cada participante brinda una actualización sobre las actividades en las que está trabajando.
- La comunicación entre los miembros del equipo es fluida, se denota un ambiente ameno dentro de la organización.
- Existen dos cargos dentro de los desarrolladores de software, estos son los analistas de sistemas y los arquitectos de software, estos últimos denotan tener un mayor conocimiento del área técnica y también sobre la lógica del negocio; los analistas de sistemas se apoyan bastante en los arquitectos de software, pues recurrentemente les consultan sobre cualquier duda que tengan referente a los desarrollos de software.
- Los QA tester participan en las diversas ceremonias Scrum que realiza la organización, son ejemplo de estas ceremonias las reuniones Sprint Planning, Sprint Retrospective y Daily Scrum.
- El eje central de la organización es el jefe de departamento, es él quien asigna tareas a los desarrolladores y QA tester, además se encarga de la comunicación con los clientes.

- No se planifica el desarrollo de cada nuevo proyecto, son los desarrolladores los que determinan la hoja de ruta para realizar los proyectos que desarrollan.
- El jefe del departamento es quien asiste a las reuniones de toma de requerimientos, es él quien le hace llegar las solicitudes de los clientes al equipo de desarrollo.
- Los desarrolladores realizan una estimación del tiempo de desarrollo de un proyecto en base a su experiencia o en base al tiempo que considere que le tomara desarrollar un proyecto de software.
- Los cambios que se realizan en las bases de datos de los diversos proyectos que la organización ha desarrollado, se manejan a través de scripts, estos scripts contienen la lógica a cambiar ya sea dentro de una tabla, trigger, vista, función o procedimiento almacenado, dichos scripts son creados por los desarrolladores de software y son ejecutados por el director de CTIC.

Manejo Actual de Scrum y Retos en su Implementación

El Departamento de Sistemas Informáticos inició el proceso de adopción del marco de trabajo Scrum en el año 2018 mediante la implementación progresiva de las diferentes actividades y herramientas que componen dicho marco de trabajo. Esta adopción inició con la división del equipo en 3 grupos de trabajo, los cuales pueden trabajar en un proyecto en común o en múltiples desarrollos en simultáneo; dichos equipos realizan reuniones diarias llamadas Daily Scrum las cuales se desarrollan en un marco de tiempo de 15 minutos como máximo.

Posteriormente, se implementó el uso de historias de usuario para registrar todas las actividades realizadas por los miembros del equipo, para ello se determinó utilizar un tablero Kanban como herramienta para la realización del seguimiento del flujo de trabajo. Durante el periodo de implementación, el equipo utiliza diferentes metodologías para establecer el tiempo de desarrollo de cada historia, enfocándose en encontrar el método que resulte más cómodo para el equipo en vez de obtener un método que brinde la máxima exactitud posible; finalmente el equipo decidió adoptar el método Scrum Poker para realizar estimaciones de tiempo.

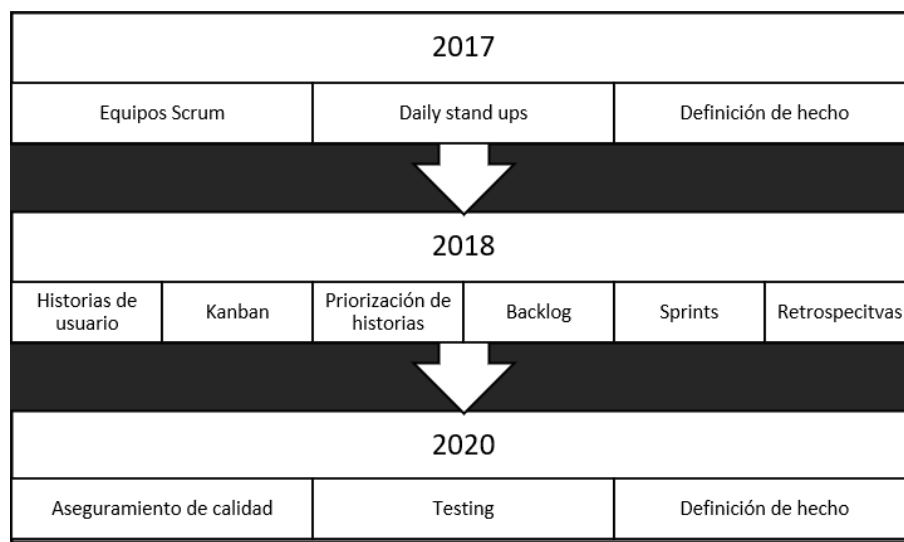
Luego se adoptó el uso de Sprints para llevar a cabo los desarrollos de proyectos en curso, la duración de cada sprint se estableció que será de 15 días, al final de cada sprint se realiza una reunión retrospectiva con una duración de 1 hora como máximo. Al inicio de cada sprint, cada equipo es responsable de crear el backlog de historias a desarrollar durante ese sprint, estas historias deben de estar enfocadas en la generación de funcionalidades que aporten valor al proyecto en desarrollo.

Se estableció el término “definición de hecho” como el estado de un producto de software en el cual este ya está implementado y listo para su uso y que además cuenta con la aprobación del cliente y del scrum master.

Finalmente, se implementaron protocolos de aseguramiento de la calidad y se incorporaron QA testers al departamento, por lo cual, se modificó la definición de hecho para que ésta incluyera la aprobación de los protocolos de aseguramiento de la calidad. En la figura 9 se puede visualizar la cronología de los instrumentos Scrum adoptados por la organización.

Figura 9

Cronología del uso de instrumentos Scrum dentro de la organización objetivo



Entre los retos afrontados por el departamento de sistemas informáticos durante la implementación de Scrum se encuentran:

- El correcto establecimiento del tiempo requerido para el desarrollo de cada historia.
- Establecer la definición de hecho de manera que esta satisfaga las necesidades técnicas y de negocio de los clientes.
- Aportes significativos al mejoramiento de procesos a través de las retrospectivas de cada sprint.
- Mantener el rapport establecido en el equipo con la llegada de la pandemia y la necesidad del trabajo remoto.

Problemas Encontrados

A través de los diversos acercamientos que se tuvo con el equipo de trabajo de la organización, se encontraron una serie de problemas referentes al estado actual de sus procesos.

Dichos problemas se presentan a continuación:

- No existe una metodología para la definición de requerimientos en el desarrollo de nuevos productos.
- Solamente el jefe y líderes de equipo acuden a reuniones de levantamientos de requerimientos.
- Durante las distintas fases de desarrollo, el equipo recibe requerimientos contradictorios sin poder establecer canales de comunicación con el cliente para solventar tal situación.
- El equipo debe destinar recursos de tiempo adicionales para acomodar nuevos requerimientos y cambios de enfoque surgidos durante el desarrollo del producto.
- Los miembros del equipo deben de dejar el desarrollo de sus tareas asignadas para realizar cambios y correcciones de errores.
- La coordinación con el cliente para realizar reuniones de avance de proyecto o capacitaciones de productos terminados es inconsistente.
- El proceso de implementación de productos terminados y cambios a productos existentes es lento, lo que dificulta brindar de manera ágil un resultado de valor a los clientes.
- El equipo de desarrollo de software a menudo tiene desconocimiento sobre la lógica de negocio de los clientes.

- Cuando un cliente solicita la creación de un nuevo producto, no se realiza una planificación que determine los pasos a seguir para llevar a cabo dicho desarrollo.
- No existen métricas que permitan calificar el nivel de éxito de los proyectos o productos desarrollados.
- Los desarrolladores de software no siempre tienen la oportunidad de aportar nuevas ideas o implementar nuevas tecnologías que puedan mejorar u optimizar los actuales procesos de desarrollo de software.
- No se realizan análisis de los datos que se obtienen de los diversos productos que la organización ha desarrollado, esto imposibilita el conocer patrones o comportamientos que podrían ayudar a identificar nuevos desarrollos para sustentar necesidades del mercado o identificar errores que se generan de manera recurrente en el proceso de desarrollo de software.
- Todos los líderes de la organización se enfocan en el área técnica.
- No existen repositorios en donde se centralice toda la documentación que se genera dentro de la organización. Esta documentación incluye manuales, scripts para gestionar cambios en los elementos de las bases de datos, registros de deuda técnica y el catálogo de productos ya desarrollados.
- No existen espacios para que los desarrolladores puedan trabajar sin interrupciones por parte de los demás miembros de la organización.

Hoja de Ruta para la Elaboración de la Propuesta de Solución

Una vez que se hayan empleado los instrumentos de recolección de datos, se procederá de la siguiente manera para adoptar la metodología Agile 2:

- Identificación de la situación actual de la organización.
- Determinar los objetivos y oportunidades de mejora que la organización pretende alcanzar gracias a la adopción de Agile 2.
- Determinar los componentes de Agile 2 que se pretende implementar dentro de la organización.
- Desarrollar una guía que contenga los componentes de Agile 2 que se adoptaran, roles, responsabilidades, métodos de implementación y plazos de tiempo para llevar a cabo la adopción de la metodología en cuestión.
- Proveer a los miembros de la organización la guía de adopción de la metodología para que conozcan de qué manera se llevará a cabo dicho proceso.
- Ayudar a los miembros de la organización a realizar la adopción de Agile 2 fomentando y monitoreando la creación de un nuevo sistema de gestión de procesos internos el cual se llamará SI Board, este sistema será creado utilizando los componentes adoptados de Agile 2.
- Finalizando el proyecto, se validará que la adopción se ha realizado correctamente acudiendo de manera presencial a la organización para observar la nueva manera en que se llevan a cabo los procesos de desarrollo de software.

- Tomando como medida de verificación el desarrollo de SI Board se determinará el grado de mejora alcanzado en la eficiencia, eficacia y calidad referente a los procesos de desarrollo de software.
- En base a la validación realizada, se identificará y determinará posibles desviaciones o dificultades que se hayan presentado en la adopción de la metodología, en base a estos hallazgos se creará un plan correctivo para solventar dichos inconvenientes.

Opinión Experta sobre el Uso de Agile 2

Para obtener mayor conocimiento sobre el correcto uso de la metodología Agile 2, entrevistamos al principal fundador de la metodología, el señor Cliff Berg, quien radica en Virginia, Estados Unidos y tiene más de 25 años de experiencia en el uso de metodologías ágiles. Actualmente, Cliff Berg es cofundador y socio gerente de Agile 2 Academy, la cual es una institución que brinda entrenamientos y consultoría sobre el proceso de adopción de Agile 2.

La entrevista completa se puede leer en el Anexo D. A continuación, presentamos una síntesis de la información más relevante obtenida a través de la entrevista realizada a Cliff Berg.

Viaje de Aprendizaje

La adopción de una metodología ágil dentro de una organización no es un proceso con pasos definidos e inalterables, es por ello por lo que no es efectivo el intentar adoptar una metodología al pie de la letra sin considerar el entorno y las circunstancias de la organización. La adopción de una metodología ágil es un viaje de aprendizaje, es un viaje prolongado en el cual la

metodología se adapta para que esta pueda suplir las necesidades actuales y emergentes de la organización.

Para emprender dicho viaje de aprendizaje, los encargados de la adopción deben de comprender la cultura, la situación y las necesidades de la organización, a partir de eso, deben identificar cuáles son los procesos y herramientas ágiles por adoptar, y en base a ello realizar una hoja de ruta de su implementación.

Las Metodologías Ágiles son Cambiantes

El manifiesto ágil fue lanzado en el año 2001, dicho manifiesto determina los valores que siguen las metodologías ágiles. Dichos valores son importantes debido a que siembran las bases de lo que hoy en día se considera Agile, el problema radica cuando esos valores pasan de ser lineamientos y se convierten en algo dogmático, es decir se convierten en una verdad irrefutable e indiscutible.

Miembros de la comunidad Agile, han observado este comportamiento dogmático, es por ello que han optado por lanzar sus propias revisiones o mejoras de lo que es Agile, ejemplo de ello son las siguientes iniciativas:

- Klaus Leopold (pionero de Kanban) y su libro Repensando ágil: Por qué los equipos ágiles no tienen nada que ver con la agilidad empresarial.
- Mathew Skelton (director de la empresa Conflux) y Manuel Pais (consultor organizacional de TI) y su libro Topologías de equipo.

- Dr. Nicole Forsgren (actual vicepresidente de investigación de GitHub), Jez Humble (considerado el padre de Entrega Continua) y Gene Kim (considerado el padre de DevOps) y su libro Acelerar.

Dichas iniciativas plantean la idea de que actualmente existen dos vertientes de Agile, el primero de ellos es “el viejo Agile”, en el cual se defiende de manera dogmática los valores del manifiesto ágil, la segunda vertiente es “el nuevo ágil” en donde se agrupan aquellos esfuerzos por innovar las metodologías ágiles. Esto denota que las metodologías ágiles son cambiantes y que estas evolucionan o se adaptan en base a las necesidades que identifica la comunidad Agile.

Medición de una Correcta Adopción de Agile 2

La medición de resultados es la manera en que se puede medir la correcta adopción de la metodología Agile 2, esto debido a que los resultados no se limitan solamente al producto directo que se genera de un proceso, sino que también considera los productos que se obtienen de manera indirecta.

Un producto directo es aquel que se espera generar a través de un desarrollo, por ejemplo, al desarrollar un proyecto de software se espera obtener un sistema informático que cuente con funcionalidades que suplan las necesidades de un cliente; mientras que los productos indirectos hacen referencia al conocimiento, aprendizaje, documentación y experiencia obtenidos gracias al desarrollo del proyecto.

Por lo tanto, la adopción de Agile 2 se denotará a través de una mejora en el proceso de desarrollo de productos y a través de un crecimiento personal y profesional de los trabajadores que han adoptado los principios y valores de la metodología.

El Desarrollo de Productos debe de ser Adaptable

El desarrollo de productos de software no es comparable con los procesos de fabricación industrial de productos, esto debido a que el desarrollo de software no es un proceso repetible, sino que cada desarrollo sigue su propia hoja de ruta dependiendo de las funcionalidades o características que se espera que el producto final posea. Por ello, cada desarrollo es diferente y estos deben de ser manejados en base a sus características.

Propuesta de Solución

Oportunidades de Mejora y Prácticas de Agile 2 Identificadas para Solventar los Problemas de la Organización

A partir del listado de problemas encontrados en los procesos de la organización, se han identificado varias oportunidades de mejora a las que se les pretende dar solución gracias a los principios y valores que componen la metodología Agile 2. Cabe destacar que uno de los valores de Agile 2 es la adaptabilidad, la cual determina que se adoptan prácticas específicas según las necesidades a cubrir.

A continuación, se presentan las oportunidades de mejora identificadas y las prácticas de Agile 2 que se sugiere implementar para darles solución:

Levantamiento de Requerimientos

Para conocer las necesidades que el cliente espera satisfacer al momento que solicita la creación de un nuevo software a la medida, es de suma importancia identificar y documentar de manera precisa cada una de las necesidades que se busca suplir. Estas funcionalidades pueden ser reflejadas en requisitos funcionales como no funcionales.

Para identificar estas necesidades es necesario que el cliente y el equipo de trabajo a cargo del proyecto se reúnan a conversar sobre las expectativas que se tiene sobre el proyecto a desarrollar, comúnmente a estos espacios de diálogo se les conoce como reuniones de levantamiento de requerimientos. En estas reuniones a menudo se comete el error que por parte del equipo de desarrollo solamente acude el jefe o encargado del equipo, lo que ocasiona que

este se convierta en un transmisor de las expectativas del cliente, es decir, se convierte en un transmisor de un mensaje, lo cual puede generar que el mensaje entregado al equipo de desarrollo no sea el adecuado o no concuerde con las necesidades específicas que declara el cliente.

Es por ello que Agile 2 sugiere que todos los miembros del equipo de desarrollo o al menos que estén más activamente involucrados con este, puedan ser parte de las reuniones de levantamiento de requerimientos, esto para que cada uno de ellos conozca de primera mano cuales son las funcionalidades a desarrollar.

Cabe destacar que Agile 2, no sugiere una metodología específica para el proceso de toma de requerimientos, pero dentro de sus principios se encuentra el poder adoptar un marco de trabajo que se integre a la cultura y a las circunstancias de la organización, esto para poder suplir las necesidades que se posean. Para la situación concreta de la organización objetivo, se puede optar por adoptar metodologías o técnicas de levantamiento de requerimientos tales como:

- Entrevistas
- Análisis de documentación ya existente
- Cuestionarios
- Observación
- Prototipado

Gestión de Cambios

Los cambios en un proyecto de desarrollo de software son algo inminente, los cambios se generan debido a discrepancias entre el producto presentado al cliente y las expectativas de este o debido a cambios repentinos que el cliente desee hacer a un proyecto aún en desarrollo, y ya sea por una u otra razón, estos cambios deben de ser integrados al proyecto, lo cual modifica su hoja de ruta.

Para gestionar dichos cambios, Agile 2 profesa entre sus valores la adaptabilidad que debe de tener un equipo de trabajo para poder integrar cambios dentro de la planificación existente, para hacer más sencilla esta tarea, Agile 2 posee entre sus principios el trabajar iterativamente en pequeños lotes, cada iteración contendrá una serie de funcionalidades a desarrollar, al finalizar la iteración, se presenta el producto al cliente el cual brinda su retroalimentación a partir de la cual se pueden obtener solicitudes de cambio.

El manejar el desarrollo del proyecto a través de iteraciones, permite seccionar el trabajo, lo cual facilita el integrar un cambio dentro de una iteración. La duración de una iteración y la cantidad de funcionalidades a desarrollar depende de la fuerza de trabajo y el tiempo con el que se cuente.

Dificultad para Obtener Retroalimentación por Parte de los Clientes

Una visión tradicional del desarrollo de software habitualmente propone desarrollar el software completamente y llegado a ese punto, se presentaba el proyecto al cliente y en ese momento se recibía por parte de este una retroalimentación con sus pensamientos acerca del

funcionamiento del producto desarrollado, esta visión ha venido a cambiar con las metodologías ágiles, pues estas han apostado por la figura de un product owner el cual está en constante comunicación con el cliente para de esa manera obtener una retroalimentación de este.

Para Agile 2, la figura del Product Owner ha venido a crear confusión pues es habitual que esta persona se convierta en un mero “receptor de pedidos”, pues es quien presenta el producto y luego se encarga de hacer llegar la retroalimentación al equipo de trabajo con el listado de cambios o nuevas funcionalidades que el cliente ha solicitado.

Para solventar dicha confusión respecto a dicha figura, Agile 2 propone que sean todos los miembros del equipo de trabajo deben de estar en contacto con los clientes y stakeholders, es decir, la tarea de obtener una retroalimentación no se debe de limitar a una sola persona del equipo de trabajo.

Además, se sugiere que la obtención de retroalimentación también incluya a los futuros usuarios de los sistemas informáticos en desarrollo, ya que la satisfacción de estos es indispensable para que el software desarrollado sea exitoso. También se debe rastrear el mercado objetivo de los productos que solicita el cliente para validar si las funcionalidades que se pretende desarrollar cumplen o satisfacen con las necesidades que el mercado demanda.

Las metodologías para obtener retroalimentación son variadas, estas pueden realizarse de las siguientes maneras:

- Reuniones presenciales o virtuales para presentar avances del proyecto a los clientes o stakeholders.

- Realización de pruebas del sistema con posibles usuarios finales.
- Lanzamiento de prototipos funcionales del sistema para el uso y testeo de los stakeholders o usuarios finales.
- Realización de cuestionarios de satisfacción a usuarios finales de sistemas ya lanzados.

Planeación de Proyectos

Una de las mayores carencias identificadas en la organización es que el ciclo de vida de sus proyectos no es planificado antes de que estos sean desarrollados, sino que luego del levantamiento de requerimientos, queda a discreción del desarrollador los pasos que seguirá para crear el proyecto solicitado.

Esto a menudo incide en que el tiempo de desarrollo sea elevado pues no existe una secuencia de pasos ya definida que determine la hoja de ruta del proyecto, además, debido a la falta de planificación no se pueden identificar posibles dependencias entre las funcionalidades a desarrollar y tampoco es posible identificar la ruta crítica del proyecto, la cual determina la mayor cantidad de tiempo que podría llevar el completar el desarrollo del proyecto en cuestión.

Cabe destacar que, al no haber una planificación, no se puede dividir un proyecto en partes o micro actividades, por lo que la estimación de tiempo de realización se calcula para todo el proyecto, es decir a nivel macro y no micro, lo cual conlleva en una estimación inexacta e imprecisa, a diferencia de los resultados que se podrían obtener realizando este mismo cálculo

sumando las estimaciones de cada micro actividad, con lo cual se obtendría una estimación más fidedigna.

Agile 2 sugiere dos principios específicos para la gestión de la planificación, el primero es que cualquier iniciativa (es decir, la creación de un producto, producto o servicio) debe poseer una visión u objetivo y un plan flexible orientado a los resultados; el segundo principio en cuestión es crear documentación para compartir con los involucrados, la cual permita una mayor comprensión de lo que se pretende alcanzar.

La planificación del proyecto debe de ser documentada adecuadamente, de manera que esta perdure durante todo el ciclo de vida del proyecto o cuando se le requiera. Un proyecto debe de tener una visión u objetivo que determine el estado al que se pretende llegar, este estado debe de satisfacer las necesidades del cliente, la planificación debe de ir en sintonía con la visión u objetivo.

La planificación debe de contener todos los aspectos que han de componer el proyecto, se debe de explicar todo al detalle tanto como sea necesario, esto con la finalidad de que haya claridad en lo que se pretenda alcanzar. La planificación debe de ser comprensible para cualquier persona que solicite dicha documentación, por lo que el lenguaje utilizado debe de ser de fácil lectura sin explayarse demasiado en definiciones técnicas.

Dificultad para Adoptar Elementos Innovadores

La industria del desarrollo de software es un elemento en constante evolución debido a que regularmente se desarrollan nuevas tecnologías que velan por la innovación, es por ello que

los miembros de un equipo de desarrollo de software deben de estar al día con los avances en esta área, para que dichos avances puedan ser adoptados en los desarrollos que se realicen.

La falta de innovación en los desarrollos de software puede conllevar a la realización de productos utilizando tecnologías desfasadas, las cuales pueden no ser las óptimas para la realización del producto deseado. Además, el no innovar puede verse reflejado como una desventaja competitiva contra empresas que ofrecen productos a la vanguardia de la innovación.

Para solventar las dificultades a la hora de adoptar elementos innovadores, Agile 2 promueve que periódicamente, el equipo de trabajo reflexione e identifique posibles cambios que pueden ser implementados dentro de la organización, entre dichos cambios se encuentra el identificar posibles elementos tecnológicos innovadores los cuales pueden aportar valor a los desarrollos de software que estén en desarrollo o que habrán de ser desarrollados.

El proceso de innovación dentro de la organización debe de ser constante, una organización nunca puede considerar que ha llegado a un punto en el que sus productos no pueden ser mejorados; cabe destacar que el implementar un elemento innovador dentro de la organización no es tarea exclusiva del jefe del equipo, sino que es tarea de cada uno de los miembros del equipo, por ello, cada miembro del equipo debe de tener voz y voto a la hora de decidir adoptar un elemento innovador.

Falta de Conocimientos del Negocio por Parte de los Desarrolladores de Software

Cuando un desarrollador de software es asignado a un nuevo proyecto, este se enfrenta al reto de comprender las necesidades que el cliente posee, para comprender dichas necesidades

debe de tener un conocimiento mínimo del área al que irá enfocada el proyecto en cuestión, estos conocimientos son llamados conocimientos del negocio o lógica del negocio.

Un desarrollador no puede limitarse a solamente tener dominio de conocimientos técnicos, sino que además debe de conjuntar este conocimiento con conocimientos del área de negocios para tener una visión más amplia del producto que está desarrollando. La falta de conocimientos del negocio puede incidir en que el producto desarrollado no concuerde con las expectativas del cliente debido a que la visión de ambas partes difiere.

Para evitar esta discordancia, Agile 2 sugiere que los miembros del equipo de desarrollo, incluidos los jefes o líderes, cumplan con un mínimo de conocimientos sobre el área de negocios al que pertenecen los proyectos en desarrollo, esto les permitirá el tener una visión más precisa de las necesidades y expectativas del cliente.

Entre las ventajas competitivas que brinda que los equipos de desarrollo cuenten con conocimientos del negocio tenemos:

- Mayor entendimiento de las necesidades de los clientes.
- Mayor facilidad de implementación de las funcionalidades solicitadas por el cliente gracias a que el desarrollador tiene comprensión del resultado esperado.
- Capacidad de sugerir mejoras o correcciones a las solicitudes del cliente.
- Mayor capacidad para validar los productos desarrollados gracias a que el desarrollador comprende la finalidad de cada funcionalidad solicitada por el cliente.

Falta de Métricas o Indicadores que Determinen la Satisfacción del Cliente

Un aspecto fundamental que se debe considerar una vez que un sistema informático ha sido liberado es la satisfacción del cliente, entiéndase satisfacción como un grado elevado de agrado del cliente hacia el producto brindado. Las razones por las que un cliente puede sentirse satisfecho pueden ir desde:

- Expectativas respecto al proyecto cumplidas.
- Producto intuitivo y vistoso.
- El producto aporta valor a la organización.
- Los usuarios finales se sienten cómodos utilizando el producto.
- El producto es rentable respecto a la inversión realizada.

Actualmente, es habitual que el equipo de desarrollo de software no realice un seguimiento que le indique el grado de satisfacción del cliente respecto al producto lanzado, esto incapacita al equipo a conocer si los productos que están desarrollando satisfacen tanto a sus clientes como a los usuarios finales.

Agile 2 sugiere que el equipo de desarrollo, una vez lanzado el producto, se mantenga en contacto con el cliente para identificar indicadores o métricas tangibles las cuales puedan brindar una visión del valor que la organización ha ganado a través de la implementación del sistema informático desarrollado, esta ganancia de valor incidirá directamente con la satisfacción del cliente.

Para que las métricas o indicadores puedan brindar resultados tangibles, se puede hacer uso de los resultados comerciales que se vean involucrados con la implementación de los sistemas informáticos que haya realizado el equipo de desarrollo, entre tales resultados comerciales se pueden considerar los siguientes:

- Crecimiento del valor de la organización en el mercado.
- Aumento de ingresos.
- Aumento de la productividad.
- Productividad de los empleados.
- Ingresos por cliente.
- Mejora de la imagen de la organización.

Falta de Análisis de Datos

Los datos son todos aquellos registros que una organización almacena, dichos datos pueden ser recopilados a través de diversas fuentes, la manera más habitual es la obtención de datos a partir de proyectos de software. Los datos en crudo (datos sin procesar o analizar) no tienen mucho valor pues no se está sacando provecho de ellos, ya sea obteniendo patrones de comportamiento de los clientes, usuarios finales o del mercado, o ya sea obteniendo cualquier otra información relevante.

El Departamento de Sistemas Informáticos UDB no brindan la suficiente atención al análisis de los datos que recopilan a través de sus diferentes desarrollos de software, esto se puede deber a que hay desconocimiento de cómo se realiza esta tarea o porque los datos son

almacenados de manera desordenada, lo que dificulta realizar relaciones entre las distintas entidades que componen los datos con lo que se cuenta, lo que dificulta el análisis de datos.

Agile 2 declara que los datos son de suma importancia en el desarrollo de un proyecto por el valor que estos aportan a una organización, es por ello que los datos deben de ser parte central del desarrollo de software y el manejo de ellos debe de ser el adecuado durante todo el ciclo de vida de un proyecto. El cuidado de los datos no puede limitarse solamente a fases iniciales del proyecto como lo pueden ser las fases de diseño o planificación, sino que dicho cuidado debe de ser transversal a todo el desarrollo del proyecto, lo que indica que continuamente se debe de validar que los datos están siendo manejados de manera correcta y además, se debe de analizar posibles hallazgos a partir de estos datos, dichos hallazgos pueden ser patrones, indicadores o comportamientos.

Para realizar las tareas de análisis de datos, Agile 2 sugiere la implementación de la figura de un especialista de datos dentro del equipo de trabajo, este especialista se debe encargar de aquellas tareas relacionadas con la obtención de valor a partir de los datos recopilados.

La figura de un especialista de datos puede ser implementada ya sea contratando a un especialista en datos o capacitando a un miembro del equipo de trabajo en el área de análisis de datos. Cabe destacar que el especialista de datos no debe de ser el único miembro del equipo con conocimientos sobre este tema, sino que el debe de fungir como un orientador que indique a los demás miembros del equipo la manera más adecuada para integrar un correcto manejo de datos en cada proyecto a desarrollar.

Elección de Líderes de Diferentes Tipos

Los líderes dentro de una organización son fundamentales para orientar a los equipos para alcanzar las metas y objetivos que se posean. Depende de las habilidades de cada líder la manera en que este gestiona a su equipo. Habitualmente, los líderes son seleccionados por una jefatura mayor para que ejerzan las labores de dicho cargo.

El liderazgo puede ser una habilidad intrínseca de una persona o puede ser una habilidad aprendida, además, las formas de liderazgo pueden ser representadas a partir de diferentes enfoques, hay líderes enfocados al área técnica, hay otros enfocados al área humana y existen otros enfocados a los procesos realizados en la organización.

Agile 2 determina que centralizar a todos los líderes en un mismo enfoque de tipo de liderazgo o en una sola persona puede ser problemático debido a que puede que estas personas fallen en sus labores lo que ocasiona que no haya más líderes que puedan guiar de manera correcta al equipo de trabajo.

Para solventar dicha problemática, se puede fomentar o elegir a nuevos líderes de diferentes enfoques para que sean estos quienes brinden guía al equipo. Las labores del cargo de líder dentro de la organización pueden ser repartidas entre todos los líderes que haya en el equipo de trabajo, descentralizando estas labores en una sola persona o en un solo tipo de enfoque de liderazgo.

Falta de Espacios para Trabajar sin Interrupciones

Los miembros de la organización diariamente se enfrentan a diversas tareas laborales las cuales pueden variar en su dificultad debido a las características de dicha tarea, dicha dificultad determina el grado de concentración y esfuerzo que una tarea amerita, es decir, a mayor grado de dificultad, mayor debe de ser la concentración y esfuerzo que el trabajador le dedicará a esa tarea.

En ocasiones, uno de los miembros de la organización puede estar trabajando en una tarea de alta dificultad y puede perder su enfoque debido a que puede ser interrumpido por otro empleado o porque tiene que acudir a una reunión.

Para solventar dicha problemática, Agile 2 sugiere el uso de espacios que permitan que una persona pueda trabajar libre de interrupciones. La implementación de esta práctica puede realizarse de manera variada, ya sea a través de un espacio físico como una habitación en la cual una persona pueda trabajar de manera aislada de los demás miembros del equipo, o a través de un sistema de estados que sea visible para todos los miembros de la organización, en donde se pueda indicar que uno de los empleados no quiere ser interrumpido durante un lapso determinado.

Manejo Ineficiente de la Documentación

Se encontró que dentro de la organización existen diversos esfuerzos para generar documentación relevante con los proyectos que se desarrollan. Estos esfuerzos están esparcidos

por toda la organización, esto debido a que cada miembro de la organización crea documentación propia la cual luego no es compartida con los demás miembros del equipo.

La documentación habitual que crean los miembros de la organización se compone por manuales, scripts, registros de deuda técnica y el listado total de proyectos que se han desarrollado, esta documentación suele ser alojada en la computadora personal de cada trabajador.

Agile 2 fomenta que la documentación generada debe de ser compartida dentro de una organización, dicha documentación debe de poder ser accedida por cualquier miembro de la organización y debe de ser comprensible para cualquier persona que lo lea, con esto, se fomenta que todos los miembros del equipo de trabajo tengan conocimientos sobre los desarrollos realizados.

Cuadro Resumen de los Principios de Agile 2 a Adoptar

En la tabla 6 se presentan de manera resumida las oportunidades de mejora y los principios de Agile 2 a utilizar como insumos para la creación de la guía de implementación del sistema de gestión SI Board.

Tabla 6

Cuadro resumen de las prácticas de Agile 2 a adoptar

Oportunidad de mejora	Principios de Agile 2 a adoptar	Categoría de la práctica
------------------------------	--	---------------------------------

Levantamiento de requerimientos	Se debe buscar un marco de trabajo ágil que encaje con la cultura y las circunstancias de la organización.	Marcos de trabajo y Metodologías
Gestión de cambios	Se debe de trabajar iterativamente en pequeñas fases.	Producto, Portafolio y Stakeholders
Dificultad para obtener retroalimentación por parte de los clientes	De manera continua se debe de obtener retroalimentación tanto de stakeholders como del mercado.	Producto, Portafolio y Stakeholders
Dificultad para adoptar elementos innovadores	Cada cierto tiempo, el equipo debe reflexionar y proponer cambios que mejoren los procesos de la organización.	Mejora continua
Planeación de proyectos	Toda iniciativa necesita una visión y un objetivo, así como un plan que sea	Planificación, Transición y Transformación

	flexible, manejable y orientado a resultados.	
	Toda documentación creada debe poder ser compartida y comprensible para quien la lea.	Producto, Portafolio y Stakeholders
Falta de conocimientos del negocio por parte de los desarrolladores de software	Los líderes y equipos de desarrollo tecnológico necesitan entender la lógica del negocio.	Dimensión y Fluidez Técnica
Falta de métricas o indicadores que determinen la satisfacción del cliente	La única manera de probar el éxito de un producto es a través de la retroalimentación por parte del cliente y de los stakeholders.	Producto, Portafolio y Stakeholders
Falta de análisis de datos	Los datos poseen valor estratégico.	Datos
Elección de líderes de diferentes tipos	Un equipo a menudo necesita más de un líder,	Liderazgo

	cada uno de un tipo diferente.	
Falta de espacios para trabajar sin interrupciones	Se debe facilitar que las personas puedan realizar su trabajo de manera enfocada y sin interrupciones.	Enfoque
Manejo ineficiente de la documentación	Toda documentación creada debe poder ser compartida y comprensible para quien la lea.	Producto, Portafolio y Stakeholders

Requerimientos del Sistema de Gestión SI Board

A partir de la reunión inicial que se tuvo con el equipo de trabajo de la organización y en base a las oportunidades de mejora identificadas, se establecieron los requerimientos primarios y funcionalidades básicas que el producto SI Board debe cumplir para satisfacer la problemática que se desea solventar, dichos requerimientos se presentan en la tabla 7.

Tabla 7

Cuadro resumen de los requerimientos primarios de SI Board

Id	Nombre	Descripción
SIB-001	Registro de productos	Los usuarios deben de ser capaces de registrar los productos generados por el departamento de sistemas

		informáticos, incluyendo la categorización de las tecnologías utilizadas en cada uno de ellos.
SIB-002	Registro de desarrollos	Los usuarios deben de ser capaces de registrar los desarrollos realizados a un producto específico incluyendo la descripción del trabajo a desarrollar, los requerimientos solicitados y las fechas de inicio y finalización.
SIB-002.1	Equipos de trabajo	Los usuarios deben de ser capaces de asignar un equipo de trabajo a un desarrollo establecido, incluyendo roles de cliente, scrum master, QA y desarrolladores.
SIB-002.2	Gestión de desarrollos	Los usuarios deben de ser capaces de retrasar la fecha de finalización de un desarrollo de ser requerido o finalizarlo para que avance a la instancia de QA.
SIB-003	Gestión de historias	Los usuarios deben de ser capaces de registrar historias de usuario incluyendo el tipo de historia, el cuerpo de la historia, el tamaño de la historia, el miembro del equipo responsable y los productos a los cuales está dirigida la historia.
SIB-004	Gestión de sprints	Los usuarios deben de ser capaces de registrar sprints para organizar el trabajo del equipo en los diferentes desarrollos incluyendo el nombre del sprint, fechas de

		inicio y finalización, objetivo y estado inicial del sprint.
SIB-004.1	Asignación de historias	Los usuarios deben de ser capaces de agregar una historia de usuario creada al backlog de historias de un sprint especificado.
SIB-004.2	Retrospectiva de sprint	Los usuarios deben de ser capaces de realizar la retrospectiva de un sprint en su finalización; incluyendo la capacidad de registrar las estrategias a reducir, continuar e incrementar.
SIB-005	Tablero de historias	Los usuarios deben de ser capaces de crear tableros Kanban que representen el progreso de un sprint específico, la capacidad de mover historias a través de las diferentes columnas del tablero y finalizar una historia al haber concluido.

De igual manera, en conjunto con el equipo de trabajo de la organización, se definieron los requerimientos secundarios para el producto SI Board, los cuales representan funcionalidades de apoyo a los procesos ágiles y la gestión de productos por parte del equipo de trabajo. Estos requerimientos se presentan en la tabla 8.

Tabla 8

Cuadro resumen de los requerimientos secundarios de SI Board

Id	Nombre	Descripción
SIB-101	Deuda técnica	Los usuarios deben de ser capaces de registrar la deuda técnica que se presente durante los procesos de aseguramiento de la calidad realizados a los productos desarrollados por el equipo de trabajo.
SIB-002	Registro de desarrollos	Los usuarios deben de ser capaces de registrar los desarrollos realizados a un producto específico incluyendo la descripción del trabajo a desarrollar, los requerimientos solicitados y las fechas de inicio y finalización.
SIB-101.1	Desarrollo de deuda técnica	Los usuarios deben de ser capaces de crear desarrollos con el objetivo de solventar la deuda técnica registrada.
SIB-102	Manuales de usuario	Los usuarios deben de ser capaces de crear gestionar la documentación existente para los productos disponibles.
SIB-103	Diseño de correos	Los usuarios deben de ser capaces de crear gestionar las cuentas de correo electrónico utilizadas en los procesos internos de la organización y gestionar el formato de los mensajes enviados a través de ellas.
SIB-104	Casos de uso	Los usuarios deben de ser capaces de crear casos de uso

		para los diferentes procesos de aseguramiento de la calidad, asignarlos a un desarrollo específico y monitorear el estado de los mismos.
SIB-105	Entrega de scripts	Los usuarios deben de ser capaces de crear un repositorio para la transferencia de archivos de bases de datos con el objetivo de completar la definición de hecho para un desarrollo o ticket.

Guía de Implementación del Sistema de Gestión SI Board

Luego de determinar la situación actual de la organización y los retos a solventar, se han determinado una serie de estrategias generales basadas en la metodología Agile 2, las cuales serán utilizadas para el desarrollo del producto llamado SI Board, esto con el objetivo de que sean adoptadas como parte de la cultura organizacional, estas estrategias están divididos en 5 categorías:

1. Nuevas herramientas ágiles
2. Empoderamiento del rol de arquitecto de software
3. Empoderamiento del equipo de QA
4. Levantamiento de requerimientos
5. Organización del equipo de desarrollo

Cabe destacar que estas estrategias basadas en Agile 2, no vienen a reemplazar en su totalidad a los procesos de Scrum que la organización ya ha adoptado, sino que vienen a complementar dichos procesos con la finalidad de mejorarlos.

Posteriormente, es responsabilidad del equipo de trabajo el desarrollar o adoptar las herramientas que crean convenientes para cumplir con la finalidad de cada estrategia. Las estrategias contenidas en cada categoría se presentan a continuación:

Nuevas Herramientas Ágiles

El equipo diversificará los medios de comunicación entre sus miembros o hacia los Product Owners, abandonando el modelo de Scrum en el cual toda comunicación debe ser en persona, para ello se fomentará el uso de herramientas como videoconferencias y chats. De igual manera se establecerán formas de comunicación asíncrona con el fin de potenciar el aporte de los miembros más introvertidos del equipo, además, se implementarán herramientas que permitan a los actores involucrados en cada proyecto el poder otorgar retroalimentación de los avances que se les presentan.

Las reuniones retrospectivas que se realizan tras la finalización de los sprint, serán simplificadas para enfocarse en reducir, incrementar o continuar prácticas, experiencias o metodologías puestas en práctica durante la duración del sprint.

Se deberá de generar un repositorio común para poder almacenar toda la documentación que se genera en la organización, este repositorio debe de ser accesible por cualquier miembro

del equipo de trabajo, cada uno de ellos debe de tener la capacidad tanto de obtener como de añadir nuevos documentos al repositorio.

Se implementará un sistema de estados, el cual permitirá a cualquier miembro del equipo el poder tener un espacio y tiempo sin interrupciones, esto con la finalidad de poder abordar mejor un problema al cual se enfrenta, dicho estado debe ser visible para todos los miembros del equipo y debe determinarse el tiempo de uso para cada estado que sea implementado.

Empoderamiento del Rol de Arquitecto de Software

El rol del arquitecto de software será transformado para potenciar su capacidad de liderazgo dentro del equipo, los arquitectos serán responsables de recolectar la retroalimentación por parte del equipo de desarrollo y el equipo de QA, promoverán el cambio y la innovación de procesos y tecnologías y acordarán la dirección a tomar por parte del equipo de desarrollo.

El arquitecto de software a nivel individual será responsable de conocer la lógica del negocio de los productos a desarrollar, apoyar a los miembros del equipo de desarrollo en la resolución de problemas y será responsable de la productividad del equipo a su cargo.

Empoderamiento del Equipo de QA

El rol del equipo de aseguramiento de la calidad será enfocado en determinar todos los procesos a utilizar para garantizar la calidad de los diferentes productos desarrollados por la organización. El desarrollo, implementación y socialización de dichos procesos será también

responsabilidad de dicho equipo, a su vez, ellos estarán a cargo de la creación y el mantenimiento de los estándares a aplicar y de los mecanismos de evaluación a implementar.

De cara al Product Owner, el equipo de aseguramiento de la calidad debe de garantizar que los objetivos esperados para los diferentes productos en desarrollo sean debidamente documentados, además ellos deben de gestionar todos los artefactos generados durante cada desarrollo (manuales, evidencias, hojas de ruta, planificaciones, hojas de levantamientos de requerimientos), esto con la finalidad de garantizar la preservación de dichos artefactos.

Levantamiento de Requerimientos

El proceso de levantamiento de requerimientos será rediseñado para mitigar las deficiencias del proceso actual, iniciando por la obligatoriedad de la reunión inicial con el Product Owner, en la cual estará presente al menos un arquitecto de software, un miembro del equipo de QA y el equipo de desarrolladores a cargo del producto. Durante este acercamiento inicial se discutirán los objetivos que el Product Owner espera alcanzar con el producto, las fechas estimadas de finalización y los procesos de negocio involucrados, de manera que sirvan como insumo al equipo de desarrollo para conocer la lógica del negocio, si así fuera necesario.

A partir de la reunión inicial se creará una historia épica, la cual será refinada a lo largo de las reuniones de seguimiento. Una vez la historia épica sea finalizada, se deberá representar de manera visual el orden y prioridad de los requerimientos a desarrollar.

Las reuniones posteriores al inicio del desarrollo de un proyecto estarán centradas en identificar funcionalidades emergentes y posibles desviaciones respecto al objetivo de proyecto, además se hará entrega de funcionalidades finalizadas por parte del equipo de desarrollo, en dichas reuniones, la presencia de al menos un arquitecto de software y un miembro del equipo de QA son requeridas para discutir cambios y retroalimentación por parte del Product Owner, mientras que los miembros del equipo de desarrollo realizarán demostraciones prácticas de los requerimientos finalizados y la entrega final del producto cuando este ya haya sido completado.

Cuando sean requeridos cambios a productos existentes (ya finalizados), el equipo de desarrollo será responsable de adquirir el conocimiento previo sobre el funcionamiento del producto y los procesos de negocios que lo componen, esto se hará mediante el apoyo de la documentación de ese producto y la realización de una reunión previa con los miembros del equipo que desarrollaron ese producto, de manera que durante el proceso de levantamiento de nuevos requerimientos, haya mayor comprensión de las necesidades del cliente y se le pueda asistir de manera eficaz.

Cambios en el Equipo de Desarrollo

Se adoptarán prácticas que conduzcan a una mayor participación en la toma de decisiones por parte del equipo de desarrolladores, estas prácticas se verán reflejadas en nuevas responsabilidades tales como convertir los requerimientos en historias de usuario, asignar el tiempo esperado de cumplimiento de cada historia, presentar al cliente los requerimientos ya finalizados y cuando un producto haya sido completado, deberán socializar el funcionamiento de dicho

producto con el resto del equipo con el objetivo de compartir el conocimiento técnico adquirido y experiencias que serán insumos para futuros desarrollos.

Se adoptará un proceso de capacitación llevado a cabo por los miembros del equipo, esta capacitación será realizada de manera periódica por un miembro del equipo, en ella se presentará la propuesta de un proceso, herramienta o tecnología que se desea implementar en el ciclo de desarrollo del departamento; dicha propuesta luego será discutida por el equipo ya sea para aprobar su adopción o para acordar realizar una investigación más a fondo sobre su viabilidad.

Finalmente, el jefe de departamento será responsable de diseñar métricas que midan el desempeño del equipo y su velocidad en las diferentes tareas que desempeñan, tales como el desarrollo de productos, corrección de errores y requerimientos de cambios a productos existentes, esto con el objetivo de identificar oportunidades de mejora en el equipo, dichas oportunidades serán entregadas a los arquitectos de software para ser incluidos dentro del proceso de capacitación llevado a cabo dentro el equipo de trabajo.

Cuadro Resumen de las Estrategias Contenidas en la Guía

A continuación, en la tabla 9, se enumeran los diferentes ítems que componen cada estrategia sugerida en la guía y las oportunidades de mejora que estos habrán de solventar.

Tabla 9

Cuadro resumen de las estrategias contenidas en la guía

Item	Categoría	Oportunidad de mejora solventada
Implementación	Nuevas herramientas	Levantamiento de requerimientos

de diversos métodos de comunicación	ágiles	Dificultad para obtener retroalimentación por parte de los clientes Planeación de proyectos
Refinamiento de las reuniones retrospectivas	Nuevas herramientas ágiles	Planeación de proyectos
Implementación de un sistema de estados de trabajo	Nuevas herramientas ágiles	Falta de espacios para trabajar sin interrupciones
Creación de un repositorio para almacenar documentación	Nuevas herramientas ágiles	Manejo ineficiente de la documentación
Nuevo rol del arquitecto de software	Empoderamiento del rol de arquitecto de software	Falta de conocimientos del negocio por parte de los desarrolladores de software Dificultad para obtener retroalimentación por parte de los clientes
Nuevo rol del	Empoderamiento del	Elección de líderes de diferentes tipos
		Elección de líderes de diferentes tipos

equipo de QA	equipo de QA	Gestión de cambios
		Falta de métricas o indicadores que determinen la satisfacción del cliente
Acercamiento inicial con el Product Owner	Levantamiento de requerimientos	Levantamiento de requerimientos Planeación de proyectos Falta de conocimientos del negocio por parte de los desarrolladores de software
Creación de historias épicas	Levantamiento de requerimientos	Levantamiento de requerimientos Planeación de proyectos
Representación visual de los requerimientos a desarrollar	Levantamiento de requerimientos	Levantamiento de requerimientos Gestión de cambios
Reuniones de seguimiento	Levantamiento de requerimientos	Levantamiento de requerimientos Gestión de cambios
Presentación al cliente de los requerimientos ya desarrollados	Cambios en el equipo de desarrollo	Dificultad para obtener retroalimentación por parte de los clientes Falta de conocimientos del negocio por parte de los desarrolladores de software Elección de líderes de diferentes tipos

Socializar productos desarrollados	Cambios en el equipo de desarrollo	Falta de conocimientos del negocio por parte de los desarrolladores de software
Capacitaciones del equipo	Cambios en el equipo de desarrollo	Dificultad para adoptar elementos innovadores
Métricas de desempeño	Cambios en el equipo de desarrollo	Falta de análisis de datos Elección de líderes de diferentes tipos

Marco de Trabajo para Implementar el Sistema de Gestión SI Board

A continuación, se presentan las prácticas y herramientas específicas que se habrá de implementar para dar ejecución a la guía de implementación del Sistema de Gestión SI Board utilizando la metodología Agile 2, dicha implementación está seccionada por categorías. Cabe destacar que, para suplir algunas estrategias contenidas en la guía, se optará por la adopción de herramientas externas, es decir, herramientas no desarrolladas por la organización.

Nuevas Herramientas Ágiles

Los cambios presentados en esta categoría se enfocan en la adopción de herramientas ágiles que faciliten las labores dentro de la organización.

1. **Adopción de la Herramienta Externa Google Hangouts.** Se adoptará la herramienta Google Hangouts como medio oficial para realizar videoconferencias con clientes o Product Owners.

2. **Adopción de la Herramienta Externa Google Chat.** Se adoptará la herramienta Google Chat como medio oficial para mantener conversaciones por chat con clientes o Product Owners.
3. **Creación de una Lista de Distribución Gmail.** Se creará una cuenta de correo institucional a la cual pueda escribir cualquier interesado para obtener información o brindar retroalimentación sobre los desarrollos de software en curso, dicha cuenta de correo será configurada como una lista de distribución (también conocida como distro), lo cual permitirá que los correos recibidos en esa cuenta sean distribuidos a la cuenta de correo institucional de cada miembro del equipo de trabajo, el proveedor de este servicio será la herramienta Gmail.
4. **Creación del Módulo Correos Autogenerados.** Se creará una herramienta que auxilie al equipo de trabajo generando correos automáticos, esto permitirá mantener una comunicación asíncrona con sus clientes o usuarios finales. Dicha herramienta se implementará a través del desarrollo de un módulo en el sistema de gestión SI Board, el cual se llamará “Módulo Correos autogenerados”.

El contenido y los escenarios bajo los cuales serán enviados los correos autogenerados, podrán ser definidos desde las opciones de configuración del módulo en cuestión.

5. **Creación del Módulo Registro de Retrospectivas.** Se redefinirá el uso de las reuniones retrospectivas, las cuales se realizan al finalizar un Sprint, éstas ahora servirán para identificar, determinar y documentar la siguiente información:

- Listado de prácticas que fueron realizadas de manera satisfactoria en el sprint que finaliza y que, por ende, deben de continuar realizándose.
- Listado de prácticas habrá que reducir su uso, debido a que no ayuda o no es eficaz en el desarrollo del proyecto.
- Listado de prácticas que serían de utilidad para el desarrollo del proyecto y que deberían de ser añadidas al siguiente Sprint.

Para documentar esta información, se desarrollará un módulo en la herramienta SI Board para llevar un control de las reuniones retrospectivas, el nombre de este módulo será “Registro de retrospectivas”.

6. **Creación del Módulo Manuales.** Dentro del sistema de gestión SI Board se desarrollará un módulo que permita almacenar de manera centralizada los manuales de todos los productos que la organización ha desarrollado.

7. **Creación del Módulo Repositorio de Scripts.** Dentro del sistema de gestión SI Board, se desarrollará un módulo el cual permita almacenar los scripts que se generan cada vez que un desarrollador solicita un cambio dentro de los elementos de las bases de datos, el nombre del módulo será “Repositorio de scripts”.

Cada desarrollador será capaz de añadir scripts al módulo. Los scripts añadidos serán divididos en dos categorías:

- Scripts activos: Son aquellos scripts los cuales aún no han sido ejecutados, estos scripts pueden ser descargados, editados o eliminados.
- Scripts históricos: Son aquellos scripts los cuales ya han sido ejecutados, estos scripts solamente pueden ser descargados.

8. **Creación del Módulo Registro de Deuda Técnica.** Se desarrollará un módulo llamado “Registro de deuda técnica”, en el cual se llevará un registro y control de todas las deudas técnicas que poseen los proyectos ya desarrollados.

9. **Creación del Módulo Productos.** Dentro del sistema de gestión SI Board se desarrollará un módulo que muestre los distintos productos (proyectos de software) que la organización ha desarrollado, este módulo se llamará “Productos”. Cada producto mostrará su estado actual, las opciones de estado disponibles se muestran a continuación:

- Activo: El producto ha sido lanzado y está en funcionamiento.
- Inactivo: El producto fue lanzado, pero dejó de estar en funcionamiento.
- En desarrollo: El producto aún no ha sido lanzado y está siendo desarrollado por los miembros del equipo de trabajo.

10. **Creación del Módulo Estados SI UDB.** Se adoptará un sistema de estados de trabajo que permitirá conocer si un miembro del equipo necesita tiempo para estar concentrado en una tarea en específico. Este sistema de estados debe de ser visible para todos los

miembros del equipo, para ello, dentro del Sistema de gestión SI Board se desarrollará un módulo informático llamado “Estados SI UDB” en donde cada miembro del equipo debe de indicar el estado actual en el que se encuentra, todos los miembros del equipo de trabajo tendrán acceso a este módulo. Los posibles estados a seleccionar en dicho módulo se detallan en la Tabla 10.

Tabla 10

Estados de trabajo disponibles para ser utilizados en el módulo Estados SI UDB

Estado	Descripción	Duración
Disponible	Indica que el empleado se encuentra dentro de su horario de trabajo y está disponible para atender a otros miembros del equipo, en caso de así ser necesario.	Su duración estándar es de 8 horas, ya que esta es la cantidad de horas que trabajan diariamente los miembros de la organización.
Concentrado	Determina que el empleado está enfocado en la realización de una tarea la cual le exige un alto nivel de concentración, por ello, el empleado no está disponible para atender a otros miembros del equipo.	Cada miembro del equipo de trabajo debe de solicitar al jefe del departamento el uso de este estado, es con él con quien se determinará su duración, dicha duración dependerá de la complejidad de tarea que se abordará.

Ausente	Indica que, en ese momento, el empleado no se encuentra en su estación de trabajo, por lo que no está disponible para atender a los demás miembros del equipo.	Su duración depende del tiempo que un empleado se encuentre fuera de su estación de trabajo.
Desconectado	Indica que el empleado está fuera de su horario laboral, y por lo tanto, no está disponible para atender a los demás miembros del equipo de trabajo.	Su duración está sujeta al tiempo libre del trabajador.

Empoderamiento del Rol de Arquitecto de Software

Estos cambios van enfocados en implementar nuevos procesos que les den un mayor poder a los arquitectos de software, esto potenciará sus capacidades de liderazgo a la vez que descentraliza dicho liderazgo dentro de la organización, pues dicha capacidad ya no sería exclusiva del jefe de departamento.

1. **Adopción de Reuniones Internas para Obtener Retroalimentación.** Periódicamente, los arquitectos de software se reunirán con los desarrolladores de software y con el equipo de QA para conocer y recolectar sus opiniones sobre los desarrollos en curso, a partir de esta información recolectada, se identificarán posibles desviaciones respecto al

objetivo del desarrollo y posibles oportunidades de mejora. La periodicidad y duración de estas reuniones quedará a criterio de cada arquitecto de software.

2. **Adopción del Proceso de Búsqueda de Innovación.** Los arquitectos de software serán responsables de sondear el mercado en busca de nuevas herramientas o tecnologías que aporten al crecimiento de la organización. Para realizar esta tarea, los arquitectos de software pueden ser auxiliados por los analistas de sistemas o por el equipo de QA.
3. **Adopción del Proceso de Documentación de la Lógica del Negocio.** Los arquitectos de Software deberán estar presentes en el proceso de toma de requerimientos en proyectos en los que se encuentren involucrados, será su responsabilidad el consultar con el cliente la lógica del negocio, el arquitecto debe de ahondar en este proceso hasta que la lógica del negocio le sea clara y concisa, posteriormente, es tarea del Arquitecto de Software el documentar la lógica del negocio y almacenarla en la funcionalidad de documentación de proyectos del módulo “Detalle de proyectos”. Esto servirá para que el arquitecto tenga el conocimiento necesario para solventar cualquier consulta que le realicen los miembros del equipo de trabajo respecto a la lógica del negocio del proyecto consultado, o en su defecto, para que el arquitecto dirija hacia la documentación generada al miembro del equipo de trabajo que realizó la consulta.

Empoderamiento del Equipo de QA

Con los cambios a realizar al equipo de QA, se pretende que estos tengan un mayor involucramiento en cada proyecto que se desarrolle en la organización, además les permitirá auxiliar a los desarrolladores gestionando todos los artefactos que dichos proyectos generan.

1. **Adopción del Proceso de Definición de Requerimientos de Calidad.** El equipo de QA estará involucrado en la planificación de cada proyecto, esto con la finalidad de determinar y documentar, los requerimientos de calidad que garantizarán la satisfacción del cliente, una vez que el proyecto sea completado.
2. **Creación del Módulo Detalle de Proyectos.** En cada proyecto desarrollado, los miembros del equipo de QA harán llegar a los desarrolladores de software un listado con los requerimientos de calidad identificados, estos requerimientos serán visibles en todo momento gracias a un módulo desarrollado en la herramienta SI Board. Dicho módulo será llamado “Detalle de proyectos”, en él se registrarán todos los requerimientos que contempla el proyecto, tanto funcionales, como no funcionales y de calidad, además, en dicho módulo se llevará un registro del personal que ha trabajado en el proyecto.
3. **Redefinición del Manejo de la Documentación de un Proyecto.** El equipo de QA, a través del módulo “Detalle de proyectos”, tendrá la capacidad de anexar archivos referentes al proyecto en cuestión, esto les permitirá llevar un control de todos los artefactos que se generan durante el desarrollo de un proyecto.

4. **Adopción del proceso de Definición de Mecanismos de Validación de**

Requerimientos. Una vez que se hayan identificado los requerimientos de calidad, el equipo de QA determinará cuales son los mecanismos que utilizarán para evaluar el cumplimiento o implementación de los requerimientos del proyecto.

Levantamiento de Requerimientos

Los cambios propuestos a través de esta categoría buscan brindar procesos y herramientas que favorezcan a la obtención y entendimiento de los requerimientos que los clientes poseen al solicitar el desarrollo de un proyecto de software.

1. **Redefinición de las Reuniones de Toma de Requerimientos.** Se redefinirá el funcionamiento de las reuniones de toma de requerimientos para que en ellas siempre estén involucrados el Product Owner, un arquitecto de software, un QA tester y los desarrolladores de software a cargo de la realización del proyecto. En esta reunión se velará por obtener todos los requerimientos del proyecto y que todos los miembros del equipo de trabajo comprendan la lógica del negocio del proyecto a realizar.
2. **Adopción del Proceso de Creación de Historias Épicas.** A partir de los requerimientos obtenidos, se creará una historia épica con la cual representar las funcionalidades que se espera que posea el proyecto a desarrollar. Se puede visualizar un ejemplo de historia épica en el Anexo F.

3. **Adopción de la Herramienta Externa Miro.** Para una fácil comprensión de todas las funcionalidades que posee un proyecto, se procederá a adoptar una herramienta externa llamada Miro, la cual permite crear y gestionar User Story Maps.

Un User Story Map es una guía visual para ayudar al equipo de desarrollo a entender el alcance del trabajo a realizar, esta herramienta ayudará a priorizar las tareas de acuerdo con las necesidades de los product owners y permitirá detectar riesgos o acciones adicionales a llevar a cabo, finalmente, todo el conocimiento generado por el User Story Map se convertirá en los requerimientos del producto de software a desarrollar.

Los User Story Maps están compuestos de actividades dispuestas en orden secuencial las cuales serán llevadas a cabo por los usuarios del producto a desarrollar, dichas actividades son divididas en acciones y estas en tareas, las cuales se organizan de acuerdo con el valor otorgado por el cliente.

En el anexo G se puede ver un ejemplo de User Story Map utilizando la herramienta Miro.

4. **Redefinición de las Reuniones de Seguimiento.** Durante las reuniones de seguimiento de un proyecto, se debe de procurar que estén presentes todos los miembros que acudieron a la reunión inicial de toma de requerimientos. En estas reuniones, los desarrolladores del proyecto deben de realizar demostraciones del avance del proyecto.
5. **Redefinición del Proceso de Obtención de Información de un Proyecto.** Cuando un cliente solicite un cambio a un proyecto ya lanzado y funcional, el desarrollador a cargo

del cambio deberá de buscar la documentación respectiva en el módulo “Detalle de proyectos” del sistema de gestión SI Board, para posteriormente adquirir el conocimiento necesario sobre el proyecto en cuestión. En caso de que el desarrollador aún posea dudas sobre el funcionamiento y la lógica del proyecto, puede solventar sus dudas consultando a quienes trabajaron previamente en ese proyecto, ese registro de personal lo puede obtener a través del módulo mencionado previamente “Detalle de proyectos”.

Cambios en el Equipo de Desarrollo

Los cambios propuestos para ser implementados en esta categoría aportarán procesos y herramientas que mejorarán la manera de trabajar de los desarrolladores de software.

1. **Adopción del Proceso de Diseño y Planificación de Proyectos.** Todos los miembros del equipo de trabajo que participen en las reuniones de toma de requerimientos de un proyecto, deberán de reunirse nuevamente con la finalidad de diseñar y planificar el proyecto a desarrollar. Toda documentación generada deberá ser almacenada en el módulo “Detalle de proyectos”.
2. **Creación del Módulo Sprints.** Dentro del sistema de gestión SI Board, se desarrollará un módulo llamado “Sprints” en el cual se gestionan todos los requerimientos que habrán de desarrollarse en los sprints llevados a cabo en la organización. Los requerimientos serán representados a través de historias de usuario.

3. **Adopción del Proceso de Presentación de Proyectos.** Cada vez que se finalice un proyecto, sus desarrolladores realizarán una presentación a los demás miembros de la organización para mostrarle las funcionalidades que componen dicho proyecto.
4. **Adopción de Capacitaciones Internas.** Periódicamente, los miembros del equipo de trabajo participarán en capacitaciones en donde se presentarán herramientas o procesos que puedan ser adoptados dentro de la organización, estas capacitaciones serán realizadas dentro del Departamento de Sistemas Informáticos UDB y serán impartidas ya sea por el jefe del departamento o por alguno de los arquitectos de software. La periodicidad de estas capacitaciones será de 3 semanas, el nombre que tendrán será “viernes del saber”, esto debido a que serán realizadas los días viernes.

Al finalizar la capacitación, los miembros del equipo de trabajo decidirán si se adopta la herramienta o proceso presentado.

5. **Adopción de un Sistema de Métricas.** Se adoptará el uso de una metodología que brinde métricas que permitan obtener información relevante a partir de los datos obtenidos en los diversos proyectos que la organización ha desarrollado, la metodología a utilizar será Key Performance Indicator (KPI).

Herramientas Externas Recomendadas

Adicionalmente al desarrollo de los módulos que contendrá el sistema de gestión SI Board, se recomienda la adopción de herramientas externas que permitan suplir todas las

oportunidades de mejora que se identificaron en la organización. Las herramientas externas recomendadas se muestran en la tabla 11.

Tabla 11

Cuadro resumen de las herramientas sugeridas para la implementación de la guía

Herramienta	Estrategia	Descripción
Google Hangouts	Implementación de métodos de comunicación síncronos y asíncronos.	Herramienta gratis la cual permitirá la comunicación a través de videollamadas.
Google Chat	Implementación de métodos de comunicación síncronos y asíncronos.	Herramienta gratis la cual permitirá la comunicación a través de chats.
Gmail	Creación de una lista de distribución de correo de tipo institucional.	Herramienta gratis que permitirá que todos los miembros de la organización puedan recibir retroalimentación de los clientes o usuarios finales.
Miro	Documentación de las historias épicas y los user story maps.	Sitio web y aplicación que permite organizar historias de usuario de manera colaborativa.

Key Performance Indicator (KPI)	Evaluación de rendimiento en base a los datos que almacena cada proyecto ya desarrollado.	Metodología de medición de rendimiento enfocada en la identificación de objetivos clave a través de un tiempo determinado.
---------------------------------	---	--

Hoja de Ruta para Implementar el Sistema de Gestión SI Board

Metodología

Se procederá a implementar gradualmente todos los procesos y herramientas sugeridos en el marco de trabajo, esto para implementar el sistema de gestión SI Board. El orden de implementación de los elementos del marco de trabajo se decidirá a partir del tiempo necesario para que la organización adopte dicho elemento. Se le dará prioridad a elementos que requieren una gran cantidad de tiempo para ser adoptados.

El proceso de implementación del marco de trabajo será realizado a partir del mes de febrero de 2022. En dicha implementación intervienen Mario Menjivar y Joshua Villavicencio, quienes supervisan dicho proceso, además intervienen todos los trabajadores de la organización pues ellos son la población objetivo que se ve beneficiada con este trabajo.

Calendarización

El jefe del Departamento de Sistemas Informáticos UDB será el responsable de realizar la asignación de los responsables que habrán de desarrollar las herramientas o que impulsarán la

adopción de las prácticas contenidas en el marco de trabajo. Esta asignación se realizará de manera mensual, esto con la finalidad de que los miembros del equipo de trabajo tengan tiempo suficiente para realizar la tarea que se les asigne. La calendarización de las tareas a realizar se muestra en la tabla 12.

Tabla 12

Calendarización de las tareas a realizar para implementar el marco de trabajo

Práctica o herramienta por implementar	Categoría	Tipo de implementación	Fecha de implementación
Creación del módulo Productos	Nuevas herramientas ágiles	Creación de un módulo en SI Board	Febrero 2022
Creación del Módulo Estados SI UDB	Nuevas herramientas ágiles	Creación de un módulo en SI Board	Febrero 2022
Adopción de reuniones internas para obtener retroalimentación	Empoderamiento del rol de arquitecto de software	Adopción o modificación de un proceso	Febrero 2022
Adopción del proceso de búsqueda de innovación	Empoderamiento del rol de arquitecto de software	Adopción o modificación de un proceso	Febrero 2022
Adopción del proceso de definición de mecanismos de validación de requerimientos	Empoderamiento del equipo de QA	Adopción o modificación de un proceso	Febrero 2022
Redefinición de las reuniones de toma de requerimientos	Levantamiento de requerimientos	Adopción o modificación de un proceso	Febrero 2022

Adopción de la herramienta externa Google Hangouts	Nuevas herramientas ágiles	Adopción de herramienta externa	Febrero 2022
Adopción de la herramienta externa Google Chat	Nuevas herramientas ágiles	Adopción de herramienta externa	Febrero 2022
Creación de una Lista de Distribución Gmail	Nuevas herramientas ágiles	Adopción de herramienta externa	Febrero 2022
Creación del Módulo Detalle de proyectos	Empoderamiento del equipo de QA	Creación de un módulo en SI Board	Marzo 2022
Creación del Módulo sprints	Cambios en el equipo de desarrollo	Creación de un módulo en SI Board	Marzo 2022
Adopción de la herramienta externa Miro	Levantamiento de requerimientos	Adopción de herramienta externa	Marzo 2022
Redefinición de las reuniones de seguimiento	Levantamiento de requerimientos	Adopción o modificación de un proceso	Marzo 2022
Adopción del proceso de presentación de proyectos	Cambios en el equipo de desarrollo	Adopción o modificación de un proceso	Marzo 2022
Creación del módulo Repositorio de scripts	Nuevas herramientas ágiles	Creación de un módulo en SI Board	Abril 2022
Creación del módulo Registro de retrospectivas	Nuevas herramientas ágiles	Creación de un módulo en SI Board	Abril 2022

Adopción del proceso de diseño y planificación de proyectos	Cambios en el equipo de desarrollo	Adopción o modificación de un proceso	Abril 2022
Adopción del proceso de documentación de la lógica del negocio	Empoderamiento del rol de arquitecto de software	Adopción o modificación de un proceso	Abril 2022
Adopción del proceso de definición de requerimientos de calidad	Empoderamiento del equipo de QA	Adopción o modificación de un proceso	Abril 2022
Redefinición del manejo de la documentación de un proyecto	Empoderamiento del equipo de QA	Adopción o modificación de un proceso	Abril 2022
Adopción del proceso de creación de historias épicas	Levantamiento de requerimientos	Adopción o modificación de un proceso	Abril 2022
Redefinición del proceso de obtención de información de un proyecto	Levantamiento de requerimientos	Adopción o modificación de un proceso	Abril 2022
Creación del módulo Correos autogenerados	Nuevas herramientas ágiles	Creación de un módulo en SI Board	Mayo 2022
Creación del módulo Manuales	Nuevas herramientas ágiles	Creación de un módulo en SI Board	Mayo 2022
Creación del módulo Registro de deuda técnica	Nuevas herramientas ágiles	Creación de un módulo en SI Board	Mayo 2022

Adopción de capacitaciones internas	Cambios en el equipo de desarrollo	Adopción o modificación de un proceso	Mayo 2022
Adopción de un sistema de métricas	Cambios en el equipo de desarrollo	Adopción o modificación de un proceso	Mayo 2022

Factores de Éxito en la Implementación de la Guía de Adopción de Agile 2

De acuerdo con la metodología Agile 2, la manera de determinar el éxito de una implementación es mediante la definición de objetivos medibles, para la guía desarrollada se ha creado una serie objetivos y el grado de éxito alcanzado que este representa para cada estrategia propuesta:

Nuevas herramientas ágiles:

- **Éxito parcial:** Existen herramientas de comunicación síncrona y asíncrona.
- **Éxito completo:** Existe una herramienta para la implementación de los estados del equipo de trabajo.

Empoderamiento del rol de arquitecto de software:

- **Éxito parcial:** Se socializan los cambios en el rol del arquitecto de software con el equipo de trabajo, dichos cambios se ponen en práctica en subsecuentes desarrollos.
- **Éxito completo:** El departamento de recursos humanos de la organización cuenta con un perfil para cada puesto ocupado, la descripción específica del rol debe reflejar las nuevas tareas y procesos contenidos en la estrategia.

Empoderamiento del rol de QA:

- **Éxito parcial:** Se socializan los cambios en el rol del equipo de QA con el equipo de trabajo, dichos cambios se ponen en práctica en subsecuentes desarrollos.
- **Éxito completo:** El departamento de recursos humanos de la organización cuenta con un perfil para cada puesto ocupado, la descripción específica del rol debe reflejar las nuevas tareas y procesos contenidos en la estrategia.
- **Éxito completo:** Existe un protocolo que gestione los artefactos generados en cada desarrollo.

Levantamiento de requerimientos:

- **Éxito parcial:** Existe documentación de las reuniones de acercamiento inicial y seguimiento.
- **Éxito completo:** Existe un formato definido para las historias épicas y los user story map.
- **Éxito completo:** El conocimiento derivado de las historias épicas y los user story map es utilizado para construir los requerimientos del desarrollo.

Cambios en el equipo de desarrollo:

- **Éxito parcial:** Existen diplomas de asistencia a las capacitaciones del equipo.
- **Éxito parcial:** Existen métricas para evaluar el rendimiento del equipo.
- **Éxito completo:** Existen instrumentos para medir la satisfacción de los product owner.

Verificación de la Implementación de la Guía

El proceso de verificación de la implementación consta de dos partes: el grado de cumplimiento de objetivos establecidos para la aplicación de la guía y que el producto de software final cumpla con los requerimientos establecidos.

Cumplimiento de objetivos

Durante el proceso de implementación de la guía se cumplieron diez de los objetivos establecidos, logrando el éxito completo de dos de las cinco estrategias propuestas (nuevas herramientas ágiles y levantamiento de requerimientos) y éxito parcial en las tres estrategias restantes, obteniendo un porcentaje de éxito igual o mayor al 50% de los objetivos cumplidos. En la tabla 13 se presenta un resumen de los objetivos cumplidos durante la aplicación de la guía.

Tabla 13

Cuadro resumen de los objetivos cumplidos durante la aplicación de la guía

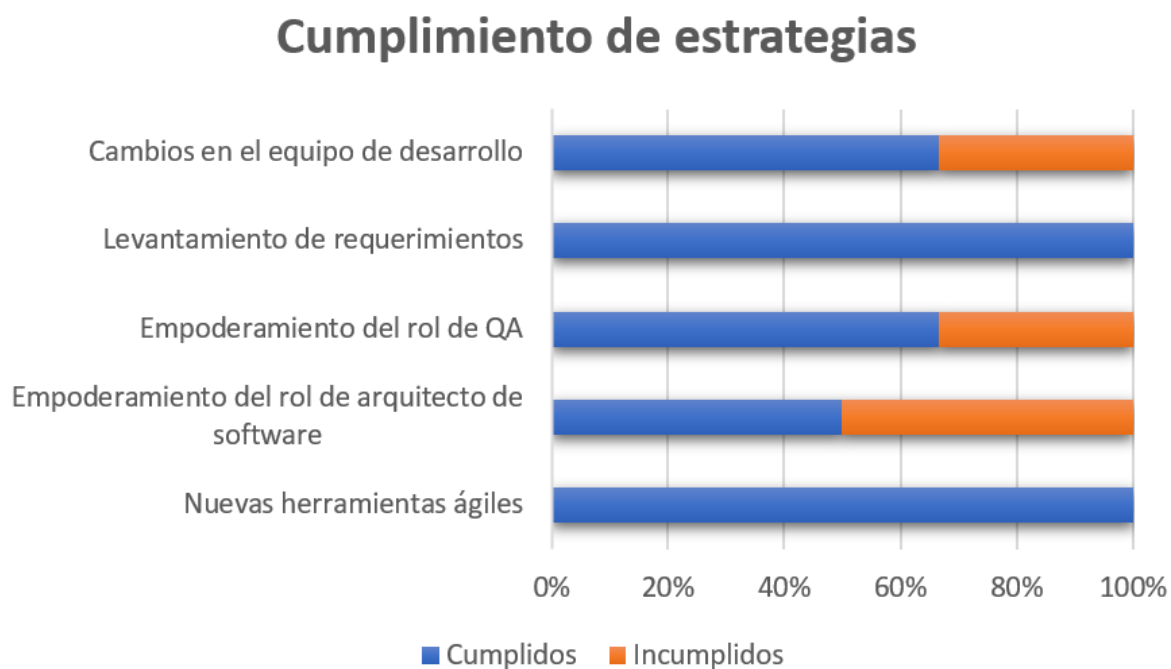
Objetivo	Cumplimiento
Existen herramientas de comunicación síncrona y asíncrona	Si
Existe una herramienta para la implementación de los estados del equipo de trabajo	Si
Se socializan los cambios en el rol del arquitecto de software con el equipo de trabajo, dichos cambios se ponen en práctica en subsecuentes desarrollos	Si

El departamento de recursos humanos de la organización cuenta con un perfil para cada puesto ocupado, la descripción específica del rol de arquitecto de software debe reflejar las nuevas tareas y procesos contenidos en la estrategia	No
Se socializan los cambios en el rol del equipo de QA con el equipo de trabajo, dichos cambios se ponen en práctica en subsecuentes desarrollos	Si
El departamento de recursos humanos de la organización cuenta con un perfil para cada puesto ocupado, la descripción específica del rol de QA debe reflejar las nuevas tareas y procesos contenidos en la estrategia	No
Existe un protocolo que gestione los artefactos generados en cada desarrollo	Si
Existe documentación de las reuniones de acercamiento inicial y seguimiento	Si
Existe un formato definido para las historias épicas y los user story map	Si
El conocimiento derivado de las historias épicas y los user story map es utilizado para construir los requerimientos del desarrollo	Si
Existen diplomas de asistencia a las capacitaciones del equipo	Si
Existen métricas para evaluar el rendimiento del equipo	Si
Existen instrumentos para medir la satisfacción de los product owner	No

En la figura 10, se presenta el porcentaje de cumplimiento de cada una de las categorías que contienen las estrategias propuestas en la guía de implementación del sistema de gestión SI Board.

Figura 10

Cumplimiento de las estrategias propuestas en la guía.



Nota: Las barras azules representan el porcentaje de cumplimiento de cada estrategia, el porcentaje máximo alcanzable es 100%.

Evidencia del producto Final

Durante el proceso de desarrollo del producto de software SI Board se cumplieron todos los requerimientos establecidos por los product owners, el sistema de gestión SI Board ya ha sido implementado y en uso por parte del equipo de desarrollo.

En las figuras 11 a la 24, se presentan capturas de pantalla que demuestran el desarrollo y uso de los módulos del sistema de gestión SI Board y de la herramienta externa sugerida, Miro. Por razones de confidencialidad, no se agregan capturas de pantalla de las herramientas externas adoptadas Google Hangouts, Google Chat y Gmail. Además, para no desvelar datos de sus clientes, la organización prefirió no brindarnos capturas de pantalla del sistema de métricas que han adoptado.

Figura 11

Módulo Productos

Productos

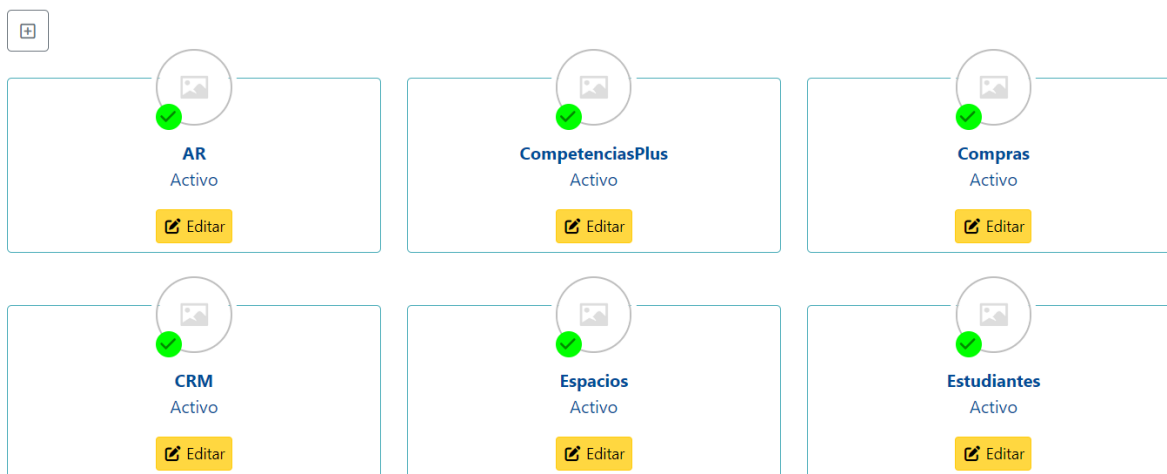


Figura 12

Módulo Detalle de proyectos opción listado de proyectos

Desarrollo

+ Agregar Todos		<input type="text" value="Nombre Desarrollo"/>
Cambios sobre el Formulario de Ingreso	Activo	
[CMP] - Liquidaciones	Activo	
[ACAD] - Nuevas modalidades de asignaturas	QA	
[ACAD] - Cuarta matrícula	Requerido	

Figura 13

Módulo Detalle de proyectos opción listado de requerimientos

#126	[ACAD] - Nuevas modalidades de asignaturas			
QA	Se debe modificar la modalidad en que se configuraran los grupos de clases			
Joshua.villavicencio Lun/6/jun/2022 vie/10/jun/2022 Espacios				
Requerimientos + Requerimiento				
Requerimiento	Dueño	Fecha	Estado	Opciones
Semipresencial: Esta modalidad debe de exigir al menos un horario presencial, y pueden contener también horarios virtuales, debe verificar choque de horarios	cesar.espinosa	06/06/2022	Activo	
Virtual: No es requerido incluir horarios presenciales, sus horarios pueden ser totalmente virtuales. Además, debe verificar choque de horarios	cesar.espinosa	06/06/2022	Activo	
Educación a Distancia: No es requerido colocar horarios, puede colocar horarios. Además, no verifica choques de horarios	cesar.espinosa	06/06/2022	Activo	
Las demás modalidades deben ser quitadas	cesar.espinosa	06/06/2022	Activo	

Figura 14

Módulo Detalle de proyectos opción equipo de trabajo del proyecto

Equipo





Product Owner	Scrum Master	Developer	QA
 cesar.espinosa Activo	 mauricio.flores Activo	 joshua.villavicencio Activo	 josseline.gomez Activo

Figura 15

Módulo Detalle de proyectos opción documentación del proyecto

	SI_T003_Casos de Prueba Nueva Modalidad.xlsx 115.14 KB / jue/16/jun/2022 4:31 p.m.	⋮
	SI_T003_Reporte QA Testing_ Nueva Modalidad.xlsx 41.85 KB / jue/16/jun/2022 4:32 p.m.	⋮
	SI_T003_Reporte QA Testing_ Nueva Modalidad.docx 702.46 KB / jue/16/jun/2022 4:33 p.m.	⋮
	SI_T003_Evidencia de Bug resueltos_ Nueva Modalidad.docx 1.1 MB / jue/16/jun/2022 4:33 p.m.	⋮

Figura 16

Módulo Sprints opción listado de sprints

Sprints

Sprint

Año: Todos Estado: Todos









Sprint	Duración	Estado	Opciones
Grupo KCJJ 13/06/2022 - 24/06/2022	13/06/2022 - 24/06/2022 11 día(s)	Iniciado	   
Grupo CLP 13/06/2022 - 24/06/2022	13/06/2022 - 24/06/2022 11 día(s)	Iniciado	   

Figura 17

Módulo Sprints opción historias de usuario

Tablero / Sprint

Ir a Grupo CLP 13/06/2022 - 24/06/2022 13/6/2022 24/6/2022 Historia Ver solo mis historias Iniciado

To Do	Doing	Done
<p>AR - Opciones de Plan Cuota - Detalle XXS Doc. Referencia jorge.rivera AR</p> <p>Homologación de Horas Sociales - Servicio Social M kevin.ponce PortalWeb</p>	<p>AR - Opciones de Plan Cuota - Detalle XXS Cuota jorge.rivera AR Tareas 1 / 4</p> <p>cambio de link S gustavo.melendez Recursos Humanos Espacios Costos</p>	<p>Arreglos en Nuevo Ingreso Pregrado y Virtual S kevin.ponce publico</p> <p>AR - Opciones de Plan Cuota L jorge.rivera AR Tareas 6 / 6</p>

Figura 18

Módulo Registro de retrospectivas

Retrospectiva:
Se mejora el ingreso de los objetivos al inicio de cada sprint

Continuar	Reducir	Incrementar
<p>Continuar la comunicación para reducir los errores cuando se realiza trabajo compartido 🔗 🗑️</p> <p>Utilizando índices para mejorar el rendimiento de los sistemas desarrollados 🔗 🗑️</p>	<p>Olvidarse de poner los objetivos al inicio del sprint 🔗 🗑️</p>	<p>Uso de deuda técnica para documentar los errores eventuales encontrados 🔗 🗑️</p> <p>Utilizar la categoría "QA Item" para el tiempo utilizando en la resolución de bugs 🔗 🗑️</p> <p>Creación de épicas para desarrollos 🔗 🗑️</p>

Figura 19*Módulo Estados SI UDB***Estado del equipo**

Empleado	Cargo	Estado	Inicio	Duración
joshua.villavicencio	Arquitecto de software	Disponible	8:00 a.m.	5 horas 0 minutos
javier.martinez	Arquitecto de software	Ausente	7:00 a.m.	6 horas 0 minutos
wendy.guzman	QA tester	Concentrado	10:30 a.m.	2 horas 30 minutos
oscar.chevez	Analista programador	Disponible	9:00 a.m.	4 horas 0 minutos

Figura 20*Módulo Manuales***Manuales**
















<input type="checkbox"/> Agregar				
Módulo	Versión	Modificado por	Fecha modificación	Opciones
Notas	1	mauricio.flores	28/09/2021	  
Planificaciones	1	mauricio.flores	28/09/2021	  
Pensum	1	mauricio.flores	28/09/2021	  
Pagos	1	mauricio.flores	28/09/2021	  
Retiros	1	mauricio.flores	11/10/2021	  

Figura 21*Módulo Correos autogenerados***Diseño**

← Guardar

Correo: Credenciales

Código

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:so="urn:schemas-microsoft-com:office:office" style="font-family:arial, 'helvetica neue', helvetica, sans-serif">
<head>
<meta charset="UTF-8">
<meta content="width=device-width, initial-scale=1" name="viewport">
<meta name="x-apple-disable-message-reformatting">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta content="telephone=no" name="format-detection">
<title>Nueva plantilla</title>
<!--[if (msn 16)]>
<style type="text/css">
a {text-decoration: none;}
</style>
</if-->
```

Figura 22*Módulo Repositorio de scripts***Repositorio de Scripts**

Nueva Entrega
 Activos
 Históricos

Registros: 124

Nombre	Fecha	Programador	Referencia	Opciones
[REDACTED]	16/06/2022 03:15:46 p.m.	javier.martinez	Ticket-16988	
[REDACTED]	16/06/2022 10:07:32 a.m.	javier.martinez	Ticket-16990	
[REDACTED]	15/06/2022 11:31:01 a.m.	erick.noyola	Ticket-16978	

Figura 23

Módulo Registro de deuda técnica

Deuda Técnica





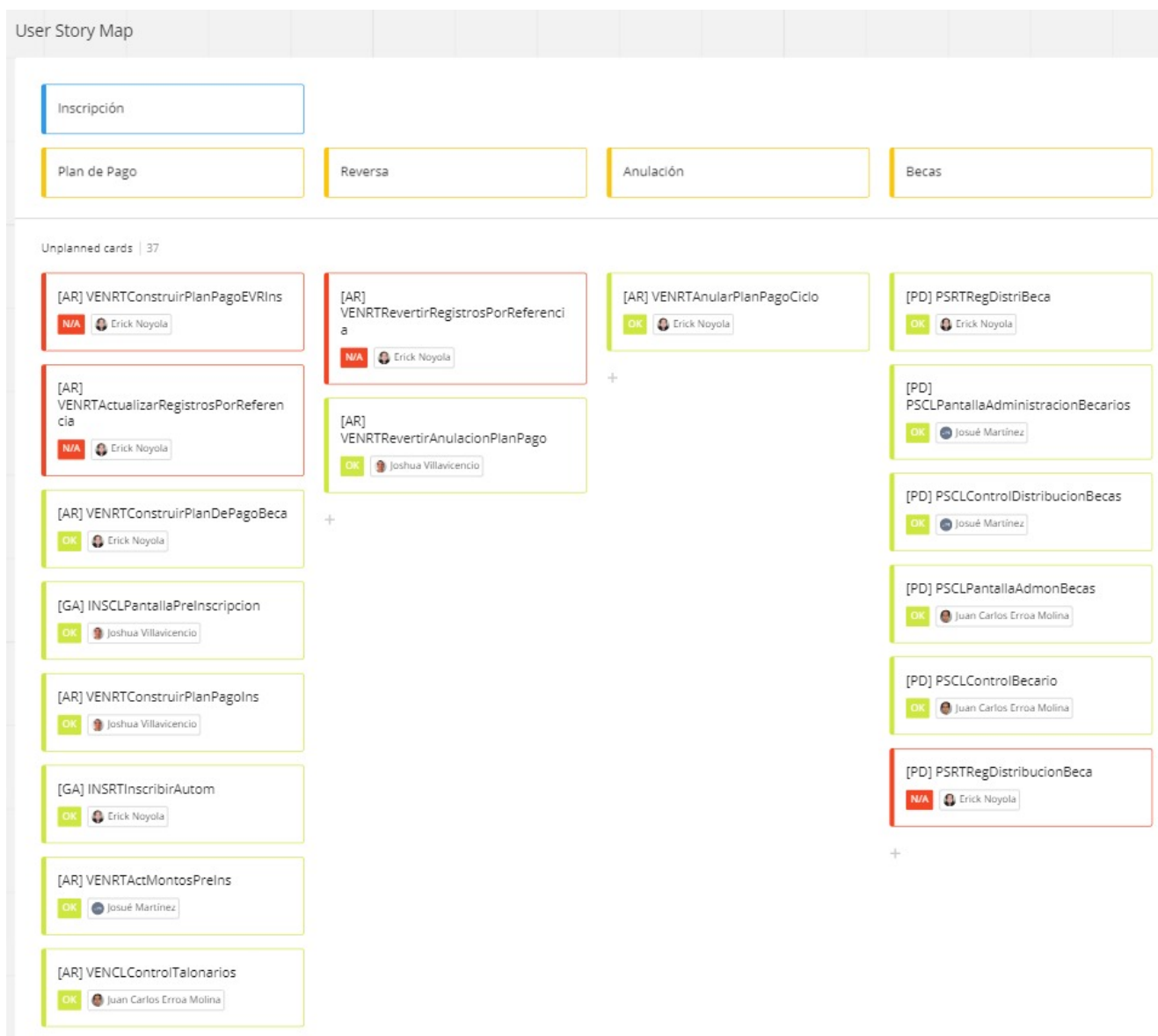
Producto	Nombre	Descripción	Estado	Creado Por	Opciones
Estudiantes	Agregar ticket de tercero	Se debe poder agregar un ticket de tercero.	Reportado	wendy.guzman	   
Estudiantes	Cuarta matrícula	No se debe de poder realizar una solicitud de cuarta matrícula cuando una materia en cuarta matrícula ha sido reprobada.	Reportado	joshua.villavicencio	   
Portal	Mensaje de Error	Mostrar mensaje de error o advertencia cuando no permita reenviar de correo de socioeconómico [Fecha de caducidad desfasada]	QA	erick.noyola	   
Estudiantes	Iconos de menu	Hay dos iconos iguales, favor de corregir	Reportado	mauricio.flores	   

Figura 24

Herramienta externa Miro



Encuesta de Satisfacción sobre la Implementación de la Guía

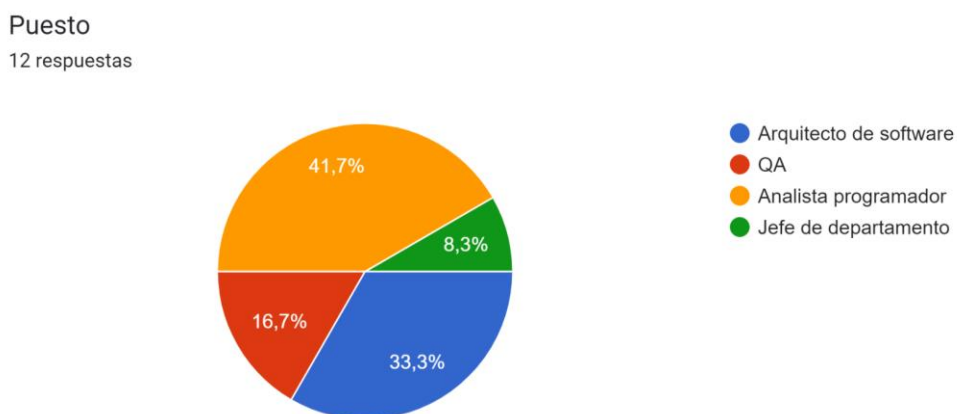
Para aplicar este instrumento de recolección de datos, se brindó a todos los miembros de la organización un formulario digital con preguntas que nos permitieron recolectar las impresiones de los miembros del equipo involucrados tanto en el desarrollo del producto como de la aplicación de la guía de implementación. Durante el proceso de implementación de la guía, un analista de sistemas fue contratado para formar parte del equipo de trabajo, elevando la población del instrumento a doce participantes. Las preguntas y respuestas obtenidas a través de este instrumento de recolección pueden ser leídas en el Anexo B.

A continuación, se presenta una síntesis de los resultados obtenidos gracias a la encuesta realizada:

Puesto. La mayor parte de los encuestados se dedica al desarrollo de los productos de software ofrecidos por el departamento en los roles de arquitecto de software y analistas de sistemas, los cuales conforman el 75% de los encuestados, 16.7% se desempeña en el rol QA realizando los procesos de aseguramiento de la calidad y el 8.3% restante realiza labores de jefatura en el rol de jefe de departamento. En la figura 25, se presenta de manera gráfica las respuestas obtenidas cuando se preguntó a los miembros del equipo de trabajo el puesto que ocupan dentro de la organización.

Figura 25

Puestos ocupados dentro de la organización.



¿Participó en el desarrollo de SI Board? El 83.3% de los encuestados participaron activamente en el desarrollo de SI Board, mientras que el 16.7% restante se incorporó al equipo de trabajo posterior al desarrollo del producto. En la figura 26 se presentan gráficamente los resultados obtenidos a partir de esta pregunta.

Figura 26

Participación en el desarrollo de SI Board.

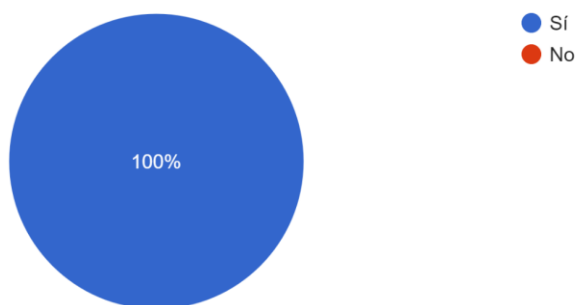


¿Participó usted en la implementación de la guía de Agile 2 en el departamento? La totalidad de los encuestados formaron parte del proceso de implementación de la guía de Agile 2 en el departamento, tal y como lo muestra la figura 27.

Figura 27

Participación en la implementación de la guía.

¿Participó usted en la implementación de la guía de Agile 2 en el departamento?
12 respuestas



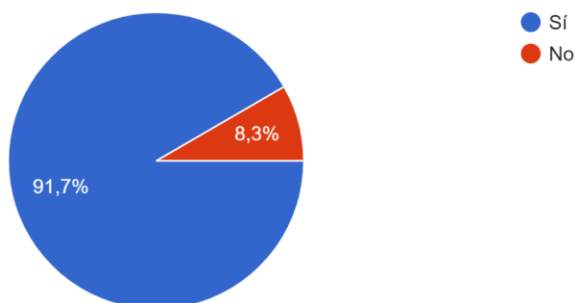
¿Considera que la guía tiene objetivos claros? El 91.7% de los encuestados consideró que la guía posee objetivos claros solo los cambios que se desean implementar en el departamento, tal y como lo muestra la figura 28.

Figura 28

Claridad en los objetivos de la guía.

¿Considera que la guía posee objetivos claros?

12 respuestas



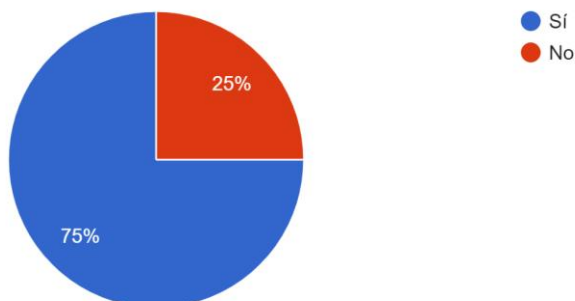
¿Considera que la guía es fácil de implementar? El 75% de los encuestados consideró que la implementación de la guía y la adopción de sus estrategias fue fácil mientras el 25% consideró lo contrario, tal y como lo muestra la figura 29.

Figura 29

Claridad en los objetivos de la guía.

¿Considera que la guía es fácil de implementar?

12 respuestas

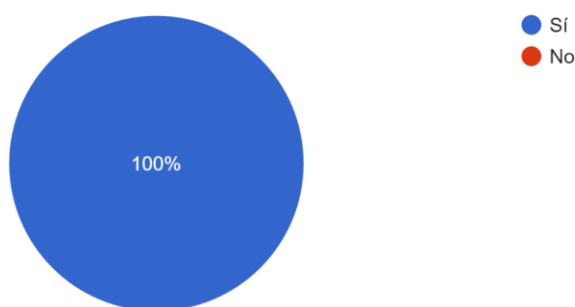


¿Considera que la implementación de la guía ha mejorado el desempeño de sus labores? La totalidad de los encuestados consideró que la guía ha mejorado su desempeño individual en sus roles correspondientes, tal y como se puede observar en la figura 30.

Figura 30

Desempeño posterior a la implementación de la guía.

¿Considera que la implementación de la guía ha mejorado el desempeño de sus labores?
12 respuestas



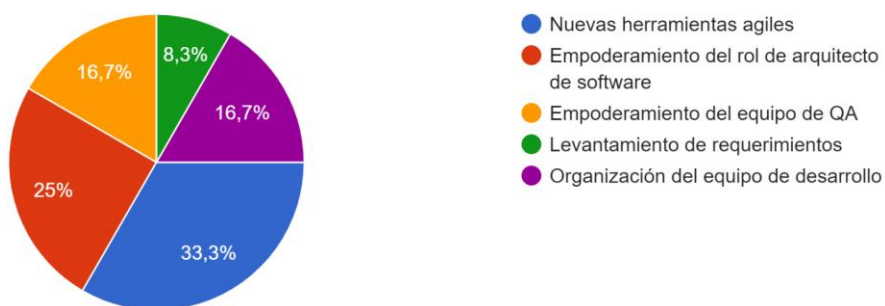
¿Cuál de las 5 categorías de estrategias considera es más útil en el desempeño de sus labores? El 33.3% considera que las nuevas herramientas ágiles es la categoría que mejor potencia el desempeño de sus labores, el 25% de los encuestados señala el empoderamiento del rol de arquitecto de software al igual que la organización del equipo de desarrollo, un 16.7% considera que el empoderamiento del equipo de QA potencia su desempeño, el 16.7% considera la organización del equipo de desarrollo como un potenciador de su desempeño individual y finalmente el 8.3% de los encuestados valora el levantamiento de requerimientos como el conjunto de estrategias más valioso como se muestra en la figura 31.

Figura 31

Utilidad de las estrategias por categoría.

¿Cuál de las 5 categorías de estrategias considera es más útil en el desempeño de sus labores?

12 respuestas



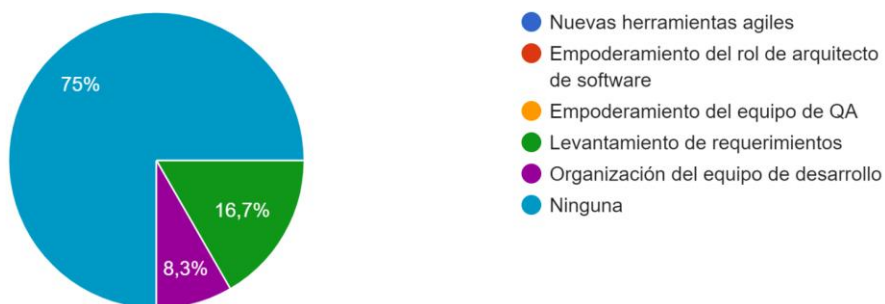
¿Cuál categoría considera que puede mejorarse? El 75% de los encuestados considera que ninguna categoría de estrategias debe mejorarse, un 16.7% señala el levantamiento de requerimiento y el 8.3% restante la organización del equipo de desarrollo, estos resultados se presentan de manera gráfica en la figura 32.

Figura 32

Categorías con oportunidades de mejora.

¿Cuál categoría considera que puede mejorarse?

12 respuestas



¿Tiene alguna estrategia que desearía estuviera incluida en la guía? Los encuestados sugirieron las siguientes estrategias:

- Una forma más clara de establecer los cambios en los requerimientos que se presentan.
- Formas más fáciles de incorporar a nuevos miembros del equipo a la implementación de la guía.
- Estrategias para promover el desarrollo móvil.

¿Tiene alguna herramienta que desearía estuviera incluida en la guía? los encuestados sugirieron las siguientes herramientas:

- Herramientas para administrar los casos de uso empleados durante el proceso de QA.

Estrategias Aplicadas para Solventar los Problemas Encontrados en la Organización

A continuación, en la tabla 14 se presenta un cuadro comparativo en donde se realiza un análisis de los problemas encontrados en la organización y las estrategias aplicadas para solventar dichos problemas. Estas estrategias han sido aplicadas a partir de los valores y principios de la metodología Agile 2.

Tabla 14

Cuadro comparativo de problemas encontrados contra estrategias aplicadas

Problemas encontrados	Estrategias aplicadas
No existe una metodología para la definición de requerimientos en el desarrollo de nuevos productos.	Se adoptó la herramienta externa Miro para crear User Story Maps los cuales permiten visualizar de manera gráfica los requerimientos de un proyecto a desarrollar, además se adoptó el proceso de creación de historias de usuario épicas para determinar el alcance esperado de un proyecto.
Solamente el jefe y líderes de equipo acuden a reuniones de levantamientos de requerimientos.	Se adoptó un proceso a través del cual se determinó que siempre asistirán a las reuniones de levantamiento de requerimientos el jefe del departamento, un

	arquitecto de software, un QA tester y los analistas de software que habrán de trabajar en el proyecto.
Durante las distintas fases de desarrollo, el equipo recibe requerimientos contradictorios sin poder establecer canales de comunicación con el cliente para solventar tal situación.	Se adoptaron herramientas externas como Google Hangouts, Google Chat y una lista de distribución de Gmail, esto con la finalidad de mejorar la comunicación con los clientes lo que facilite el proceso de obtención y definición de requerimientos. Además, dentro del sistema de gestión SI Board se desarrolló un módulo llamado “Correos autogenerados” el cual permite enviar correos predefinidos a clientes o usuarios finales bajo escenarios en donde esto sea requerido.
El equipo debe destinar recursos de tiempo adicionales para acomodar nuevos requerimientos y cambios de enfoque surgidos durante el desarrollo del producto.	Dentro del sistema de gestión SI Board se desarrolló un módulo llamado “Detalle de proyectos” el cual permite llevar un control de todos los requerimientos que contiene

Los miembros del equipo deben de dejar el desarrollo de sus tareas asignadas para realizar cambios y correcciones de errores.	un proyecto y además permite realizar ajustes a la planificación de cada proyecto. Se redefinió el proceso de obtención de información de un proyecto, ya que ahora toda la información referente a un proyecto ya finalizado se encontrará documentada en el módulo “Detalle de proyectos”, esto facilitará la comprensión de los componentes y la lógica del proyecto por parte de cualquier desarrollador de software al cual se le asigne la realización de un cambio a un determinado proyecto.
La coordinación con el cliente para realizar reuniones de avance de proyecto o capacitaciones de productos terminados es inconsistente.	Gracias a las herramientas externas adoptadas Google Chat, Google Hangouts y la lista de distribución Gmail, se facilita la comunicación con el cliente y se diversifica la forma en que se pueden llevar a cabo las reuniones de seguimiento de un proyecto, ya que estas pueden ser llevadas a cabo a través de una videollamada.

<p>El proceso de implementación de productos terminados y cambios a productos existentes es lento, lo que dificulta brindar de manera ágil un resultado de valor a los clientes.</p>	<p>Se mejoró el desarrollo y gestión de sprints, el cual es un proceso heredado de la previa adopción de Scrum que la organización había llevado a cabo. Para ello, dentro del sistemas de gestión SI Board se desarrolló un módulo llamado “Sprints”, el cual permite llevar un control de los Sprints realizados y las historias de usuario que los componen.</p>
<p>El equipo de desarrollo de software a menudo tiene desconocimiento sobre la lógica de negocio de los clientes.</p>	<p>Se redefinió el proceso de obtención de la lógica del negocio de un proyecto en desarrollo, siendo ahora los arquitectos de software los encargados de obtener este conocimiento a través de las reuniones de toma de requerimientos, posteriormente, el arquitecto de software se encargará de documentar y difundir el conocimiento adquirido, además deberá de asegurarse de que todos los miembros de la organización</p>

	involucrados en el proyecto comprendan la lógica del negocio.
Cuando un cliente solicita la creación de un nuevo producto, no se realiza una planificación que determine los pasos a seguir para llevar a cabo dicho desarrollo.	Los miembros de la organización presentes en las reuniones de toma de requerimientos se encargarán de diseñar y planificar el proyecto a realizar, esto será documentado y almacenado dentro del módulo “Detalle de proyectos”.
No existen métricas que permitan calificar el nivel de éxito de los proyectos o productos desarrollados.	Se adoptó un proceso para determinar los requerimientos de calidad de un proyecto, los cuales garanticen la satisfacción del cliente, dichos requerimientos serán determinados por el equipo de QA al iniciar el desarrollo del proyecto, estos requerimientos serán documentos y estarán visibles para todos los miembros de la organización a través del módulo “Detalle de proyectos”. A su vez, será el equipo de QA quien se encargue de determinar los mecanismos a

	utilizar para validar el cumplimiento de los requerimientos de calidad determinados.
Los desarrolladores de software no siempre tienen la oportunidad de aportar nuevas ideas o implementar nuevas tecnologías que puedan mejorar u optimizar los actuales procesos de desarrollo de software.	Se ha adoptado un nuevo proceso a través del cual los arquitectos de software se encargan de sondear el mercado con la finalidad de encontrar herramientas o procesos que aporten a la innovación dentro del departamento. Además, también se ha adoptado un proceso de capacitación periódica llevada a cabo por los mismos miembros de la organización, dichas capacitaciones reciben el nombre del “viernes del saber” y en ella el capacitador se encarga de presentar una herramienta o proceso que el proponga para ser adoptado dentro de la organización.
No se realizan análisis de los datos que se obtienen de los diversos productos que la organización ha desarrollado, esto	Se adoptó un sistema de métricas que permite evaluar el rendimiento del equipo, y posibles patrones o comportamientos

<p>imposibilita el conocer patrones o comportamientos que podrían ayudar a identificar nuevos desarrollos para sustentar necesidades del mercado o identificar errores que se generan de manera recurrente en el proceso de desarrollo de software.</p> <p>Todos los líderes de la organización se enfocan en el área técnica.</p>	<p>dentro de los datos que almacenan de sus diversos proyectos ya desarrollados. Estas métricas son gestionadas de manera interna por el jefe del departamento.</p> <p>El sistema de métricas adoptado es KPI, el cual brinda indicadores según las necesidades de la organización en cuestión.</p> <p>Se han adoptado diversos procesos cuya finalidad es empoderar a los arquitectos de software y QA testes, esto con la finalidad de que estos puedan ejercer labores de liderazgo y para que el liderazgo organizacional no este centralizado en el jefe del departamento.</p>
<p>No existen repositorios en donde se centralice toda la documentación que se genera dentro de la organización. Esta documentación incluye manuales, scripts para gestionar cambios en los elementos de las bases de datos, registros de deuda</p>	<p>Dentro del sistema de gestión SI Board se desarrollaron los módulos “Detalle de proyectos”, “Manuales”, “Repositorio de scripts”, “Registro de retrospectivas” y “Registro de deuda técnica”, los cuales sirven para almacenar toda la</p>

técnica y el catálogo de productos ya desarrollados.

documentación que se genera dentro de la organización. Esto permite tener centralizada toda la documentación generada dentro de un mismo sistema informático.

No existen espacios para que los desarrolladores puedan trabajar sin interrupciones por parte de los demás miembros de la organización.

Dentro del sistema de gestión SI Board se desarrolló un módulo llamado “Estados SI UDB” en el cual se determina el estado de trabajo de cada miembro de la organización. Esta herramienta permite que un empleado tenga pueda indicar a sus compañeros que ha solicitado tiempo para poder trabajar sin ser interrumpido.

Mejoras Obtenidas Gracias a la Implementación de la Guía

Después de finalizar el proceso de implementación de la guía en el mes de mayo, se han identificado las siguientes mejoras dentro de los procesos de desarrollo del equipo de trabajo:

- Existe documentación que valida los requerimientos generados por los product owner y detalla de qué manera dichos requerimientos han cambiado a través del ciclo de vida del desarrollo.

- El equipo de QA es capaz de generar casos de prueba a partir de la documentación generada durante el levantamiento de requerimientos.
- El equipo de QA genera documentación que certifica los desarrollos entregados.
- Existe transparencia en el trabajo realizado por el equipo de trabajo.
- El equipo de trabajo tiene la oportunidad de ejercer sus habilidades de liderazgo en el desempeño de sus tareas.
- El equipo de trabajo tiene mecanismos para presentar herramientas o metodología que desearían implementar dentro del flujo de trabajo del departamento.
- Se han adoptado herramientas que permiten visualizar las actividades que componen un desarrollo en curso.
- Se han adoptado herramientas que permiten centralizar toda la documentación generada dentro de la organización.
- Se ha facilitado el proceso de comunicación con los clientes, lo que ayuda a poder obtener retroalimentación de estos de manera más efectiva.
- Se adoptó una herramienta que posibilita el conocer el estado de trabajo de cada miembro del equipo, esto a su vez posibilita que un trabajador pueda laborar sin interrupciones en caso de ser necesario.
- Se ha adoptado una herramienta que permite llevar un control de las deudas técnicas de los diversos proyectos que desarrolla la organización.

- Se optimizó el proceso de obtención de la lógica del negocio de un proyecto a desarrollar, de manera que esta sea comprensible para todos los involucrados en dicho proyecto.
- La organización ahora vela por la búsqueda de innovación dentro de sus procesos, lo cual se traduce en la adopción de un proceso de mejora continua.

Conclusiones

- Se identificaron las diferencias entre la metodología Agile 2 y las metodologías ágiles tradicionales.
- Se identificaron las oportunidades de mejora de la organización y se les dio solución a través de los principios y valores de la metodología Agile 2.
- Agile 2 permitió empoderar a todos los miembros del equipo de trabajo para sugerir e implementar cambios significativos dentro del flujo de trabajo del departamento.
- Agile 2 otorga al equipo de trabajo la libertad de implementar la metodología o herramienta ágil que crean más conveniente, sin depender de un marco de trabajo rígido.
- Establecer objetivos de éxito que sean medibles y corroborables es la mejor manera de determinar el éxito de un cambio de metodología de trabajo.
- La adopción de la metodología Agile 2 no ha reemplazado la previa adopción del marco de trabajo Scrum, sino que Agile 2 ha venido a potenciar y complementar las capacidades de los procesos de Scrum.
- El desarrollo del sistema de gestión SI Board se llevó a cabo de manera exitosa, en este proceso se involucraron todos los miembros de la organización.
- Gracias al desarrollo del sistema de gestión SI Board, se adoptaron herramientas que mejoran los procesos de la organización.

Recomendaciones

- Deben establecerse mecanismos para evaluar el conocimiento de las nuevas incorporaciones al equipo de trabajo, para que así, la introducción a la metodología de trabajo sea exitosa.
- Los formatos de los documentos generados durante los procesos de desarrollo deben de someterse a un proceso de refinamiento y mejora continua.
- En actualizaciones futuras, el equipo debe incorporar los casos de uso utilizados por el equipo de QA al sistema de gestión SI Board.
- La organización debe de estar abierta ante la posible adopción de nuevas prácticas ágiles que favorezcan y potencien los procesos actuales con los que cuentan.
- Todos los miembros de la organización deben de tener voz y voto en el proceso de búsqueda y proposición de nuevos procesos que puedan ser adoptados dentro de la organización.
- Los principios y valores de la metodología Agile 2, no deben de ser tomados como algo dogmático, pues estos solamente presentan los lineamientos que una organización puede tomar para implementar dicha metodología, pero dichos lineamientos no son inalterables o irrefutables.
- Dentro de la organización, se debe de buscar el crecimiento tanto del equipo de trabajo como de los individuos, esto debido a que el crecimiento de un individuo incidirá en el crecimiento del equipo al que pertenece.

Glosario

Definición de hecho: criterio de común acuerdo que debe ser cumplido para que un equipo de trabajo considere un aspecto de un proyecto como completo o concluido.

Deuda técnica: también conocida como deuda de código o de diseño, son todos los errores o deficiencias resultantes del desarrollo apresurado de productos de software, la cual incurre en costos de estandarización y corrección.

Rapport: una relación caracterizada por el entendimiento de todas las partes involucradas que permite que la comunicación sea fácil y comprensible.

Scrum Poker: también llamado póker de estimación o póker de planeación es una técnica de estimación utilizada por equipos de trabajo para estimar el tamaño de una historia de usuario o el esfuerzo requerido para completarla.

Script: es toda aquella porción de código que puede ser ejecutada sin necesidad de ser compilada.

Tablero Kanban: herramienta de administración de proyectos ágiles que permite visualizar el trabajo completado, las tareas por hacer y el flujo de eficiencia del equipo de trabajo.

Test unitario: fragmento de software creado por el equipo de desarrollo el cual ejecuta una funcionalidad específica de un producto y valida los resultados obtenidos.

Referencias

Agile 2 Official Documentation. (2022). The Values and Principles of Agile 2. Retrieved 23 May 2022, from <https://agile2.net/agile-2/the-values-and-principles-of-agile-2/>

Bass, L., Weber, I., Zhu, L., Xu, F., & Ren, F. (2015). DevOps: A Software Architect's Perspective.

Berg, C., Cagle, K., Cooney, L., Fewell, P., Lander, A., Nagappan, R., & Robinson, M. (2021). Agile 2: the next iteration of Agile.

Berg, C. (2022). *Old Agile vs New Agile - DZone Agile*. dzone.com. Retrieved 27 June 2022, from <https://dzone.com/articles/old-versus-new-agile>.

Found, D. (2022). *What is CI/CD/CD? The Differences & Benefits & DevOps Application*. Indellient. Retrieved 8 June 2022, from <https://www.indellient.com/blog/whats-the-difference-between-continuous-integration-continuous-delivery-and-continuous-deployment/>.

Francisco González, B. (2022). *Metodologías vs. Frameworks*. LinkedIn.com. Retrieved 18 June 2022, from <https://www.linkedin.com/pulse/metodologías-vs-frameworks-betsua-nohemí-francisco-gonzález/>

- Garzas, J. (2010). El primer método ágil, DSDM - Javier Garzas. Retrieved 16 February 2022, from <https://www.javiergarzas.com/2010/01/primer-metodo-agil-dsdm.html>
- Logan, M. (2022). *Defining the Dev and the Ops in Devops*. DevOps.com. Retrieved 8 June 2022, from <https://devops.com/defining-the-dev-and-the-ops-in-devops/>.
- NetApp Documentation. (2022). *¿Qué es DevOps y para qué sirve? | NetApp*. Netapp.com. Retrieved 5 June 2022, from <https://www.netapp.com/es/devops-solutions/what-is-devops/>.
- Redacción KeepCoding. (2022). *Historia y origen de DevOps | KeepCoding Tech School*. KeepCoding Tech School. Retrieved 8 June 2022, from <https://keepcoding.io/blog/historia-y-origen-de-devops/>.
- SCRUMstudy, A brand of VMEdU, Inc. (2013). *A guide to the scrum body of knowledge (Sbok guide)* (3rd ed.).
- Agilemanifesto.org. (2022). *Manifesto for Agile Software Development*. Retrieved 16 July 2022, from <http://agilemanifesto.org/>.
- Schwaber, K., & Sutherland, J. (2020). *Scrum Guide | Scrum Guides*. Scrumguides.org. Retrieved 28 June 2022, from <https://scrumguides.org/scrum-guide.html>.
- Silwa, C. (2020). New Agile 2 development aims to plug gaps, complement DevOps. Retrieved 16 February 2022, from

<https://www.techtarget.com/searcharchitecture/news/252492739/New-Agile-2-development-aims-to-plug-gaps-complement-DevOps>

SoftwareOne. (2022). *DevOps con AWS*. SoftwareONE. Retrieved 7 June 2022, from <https://www.softwareone.com/es-pa/solutions/publisher-advisory-services/aws/devops-with-aws>.

Anexos

Anexo A - Encuesta inicial realizada a los trabajadores de la organización objetivo

Tema: Encuesta inicial realizada a los trabajadores del Departamento de Sistemas Informáticos UDB.

Objetivo: Recopilar e identificar los conocimientos de los miembros del Departamento de Sistemas Informáticos UDB sobre metodologías ágiles, enfocándonos principalmente en sus conocimientos del marco de trabajo Scrum y la metodología Agile 2.

Encuestador: Mario Menjivar.

Encuestados: Trabajadores del Departamento de Sistemas Informáticos UDB.

Cantidad de personas encuestadas: 11.

Fecha de realización: 17 de enero de 2022.

Formulario de la Encuesta

Figura A1

Encuesta metodologías ágiles SI UDB 2022

1. Correo
2. Nombre
3. Edad
4. Cargo que desempeña en SI UDB
5. ¿Cuánto tiempo lleva laborando en SI UDB?
6. ¿Sabe que son las metodologías/frameworks ágiles?
7. En base a sus conocimientos y experiencia, ¿Para qué sirven las metodologías/frameworks ágiles?
8. ¿Qué acciones o actividades específicas del marco de trabajo ágil Scrum se han adoptado en SI UDB?
9. ¿Cuáles considera que son las ventajas de utilizar Scrum?
10. ¿Cuáles considera que son las desventajas de utilizar Scrum?
11. ¿Considera que el uso de Scrum fomenta que el desarrollo de software sea eficaz y eficiente?
12. ¿Qué aspectos considera que deben o pueden ser mejorados dentro de los procesos de desarrollo de software de SI UDB?
13. ¿Usted estaría abierto a adoptar una nueva metodología/framework ágil dentro de SI UDB?
14. ¿Cuál considera que podría ser la mayor dificultad a la hora de adoptar una nueva metodología/framework ágil?

15. ¿Conoce o ha escuchado sobre la metodología Agile 2?
16. ¿Ha cursado capacitaciones o cursos sobre metodologías/frameworks ágiles?
17. ¿Posee alguna certificación sobre metodologías/frameworks ágiles? En caso de que su respuesta sea afirmativa, por favor mencione que certificación posee.

Respuestas Obtenidas

1. Correo

Las respuestas a esta pregunta son anónimas.

2. Nombre

Las respuestas a esta pregunta son anónimas.

3. Edad

Encuestado 1: 24 años.

Encuestado 2: 25 años.

Encuestado 3: 29 años.

Encuestado 4: 24 años.

Encuestado 5: 25 años.

Encuestado 6: 32 años.

Encuestado 7: 26 años.

Encuestado 8: 22 años.

Encuestado 9: 26 años.

Encuestado 10: 33 años.

Encuestado 11: 52 años.

4. Cargo que desempeña en SI UDB

Encuestado 1: Analista de sistemas.

Encuestado 2: Analista de sistemas.

Encuestado 3: Analista de sistemas.

Encuestado 4: Arquitecto de software.

Encuestado 5: Arquitecto de software.

Encuestado 6: QA tester.

Encuestado 7: Arquitecto de software.

Encuestado 8: Analista de sistemas.

Encuestado 9: QA tester.

Encuestado 10: Arquitecto de software.

Encuestado 11: Jefe de departamento.

5. ¿Cuánto tiempo lleva laborando en SI UDB?

Encuestado 1: 6 Meses.

Encuestado 2: 7 meses.

Encuestado 3: año y medio.

Encuestado 4: 1 año.

Encuestado 5: 2 años y 8 meses.

Encuestado 6: 2 años.

Encuestado 7: 11 Meses.

Encuestado 8: 5 meses.

Encuestado 9: 2 meses aprox.

Encuestado 10: 5 años.

Encuestado 11: 11 años.

6. ¿Sabe que son las metodologías/frameworks ágiles?

Encuestado 1: Sí.

Encuestado 2: Sí.

Encuestado 3: Sí.

Encuestado 4: Sí.

Encuestado 5: Sí.

Encuestado 6: Sí.

Encuestado 7: Sí.

Encuestado 8: Sí.

Encuestado 9: Sí.

Encuestado 10: Sí.

Encuestado 11: Sí.

7. En base a sus conocimientos y experiencia, ¿Para qué sirven las metodologías/frameworks ágiles?

Encuestado 1: Sirve para adaptar y responder ante los cambios que puedan surgir en diferentes proyectos.

Encuestado 2: Gestionar proyectos con más rapidez.

Encuestado 3: Para optimizar tiempo y tener un mejor orden de los procesos a realizar sobre los proyectos.

Encuestado 4: Nos ayudan a poder adaptar el trabajo que se realiza a las necesidades del proyecto de una manera rápida y flexible a los cambios.

Encuestado 5: Para gestionar proyectos con rapidez, flexibilidad y adaptabilidad a los cambios.

Encuestado 6: Para optimizar tiempo en el desarrollo de un sistema, la organización en el equipo para realizar las diferentes entregas y para disminuir errores durante la producción.

Encuestado 7: Aumenta la eficiencia entre las entregas en un proceso de negocio.

Encuestado 8: Son aquellas que permiten adaptar la forma de trabajo a los requerimientos del proyecto.

Encuestado 9: Sirven para gestionar proyectos/sistemas que se encuentran en producción y a medida se realizan las tareas cortas de cada integrante del equipo, se van tomando ideas, mejoras constantes con el desarrollo del sistema/proyecto.

Encuestado 10: Permite a los equipos de trabajo adaptarse rápidamente a cambios dentro del proyecto e iterar sobre requerimientos cambiantes.

Encuestado 11: Para dar respuestas a entornos cambiantes y complejos.

8. ¿Qué acciones o actividades específicas del marco de trabajo ágil Scrum se han adoptado en SI UDB?

Encuestado 1: Daylis, Retrospectiva, Sprints.

Encuestado 2: Dailys.

Encuestado 3: Dailys, Sprint, Retrospectiva.

Encuestado 4: La realización de dailys para mantener al tanto al equipo de que se está trabajando y hacer sprint para cada desarrollo.

Encuestado 5: Sprint, Scrum Daily Meeting y Sprint Retrospective.

Encuestado 6: Planeación del Sprint, Sprint, Daily, Demostración y Retrospectiva.

Encuestado 7: Historias de Usuario, Planeación (Sprint), Reuniones planificando los objetivos del día, Retrospectiva.

Encuestado 8: Uso de Sprints, uso del Daily y Retrospectiva

Encuestado 9: Sprints y creación de historias, los Daily, retrospectiva del Sprint, revisión de cumplimiento de objetivos.

Encuestado 10: Backlog de tareas priorizadas, Daily Scrum meetings, sprints y retrospectivas de sprints.

Encuestado 11: Daily, sprint, retrospective, tableros y más.

9. ¿Cuáles considera que son las ventajas de utilizar Scrum?

Encuestado 1: Mejorar el trabajo en equipo, Ayuda la productividad.

Encuestado 2: Buena gestión para los riesgos de un proyecto.

Encuestado 3: Mejor comunicación con los compañeros de trabajo para entender procesos y resolución de conflictos.

Encuestado 4: Fomenta la participación, el trabajo en equipo, desarrollos adaptables a las necesidades.

Encuestado 5: Permite trabajar por fases el proyecto, realizando entregas pequeñas y es flexible para aplicar a cuál área o contexto.

Encuestado 6: Que agiliza el proceso para el desarrollo de software, el cliente puede estar involucrado en las diferentes etapas del proceso, gestión del riesgo, se brindan resultados concretos en corto tiempo.

Encuestado 7: Centralización de un marco de trabajo en la empresa. Comunicación activa entre los compañeros de trabajo. Diversificación de ideas en un mapa conceptual entre los compañeros de trabajo. Establece un roadmap a corto y mediano plazo según el objetivo de la empresa.

Encuestado 8: Mejora el trabajo en equipo y la cooperación. Se organizan mejor los flujos de trabajo de acuerdo al rol que se ejerce en el proyecto.

Encuestado 9: Hay resultados rápidos en períodos de tiempo cortos, promueve trabajo en equipo, flexible ante posibles cambios en el desarrollo del proyecto.

Encuestado 10: Permite al equipo autogestionarse y compartir información sobre el estado del proyecto de manera más rápida.

Encuestado 11: Mayor entrega de valor.

10. ¿Cuáles considera que son las desventajas de utilizar Scrum?

Encuestado 1: Funciona más que nada con equipos reducidos.

Encuestado 2: Sin la repartición del trabajo es muy difícil

Encuestado 3: Tomarse más tiempo del recomendado en los daily.

Encuestado 4: En equipos muy grandes se necesita dividir grupos con objetivos específicos concretos para funcionar.

Encuestado 5: Es eficiente en grupos pequeños y no se puede trabajar si no se tienen plazos o tareas definidas.

Encuestado 6: Al momento de implementar Scrum todo el equipo debe conocer el marco de trabajo. Si por algún motivo quedan tareas sin finalizar, el resto de tareas pendientes comenzarán a posponerse indefinidamente.

Encuestado 7: La adopción de este marco de trabajo lleva tiempo para ser implementada de manera correcta, por lo que es necesario un equipo con experiencia para poder ser implementado, así mismo esto conlleva a que en equipos grandes es un gran desafío incorporar este marco de trabajo ya que esto requiere parametrizar y controlar cada una de las funciones que estos desempeñan.

Encuestado 8: Funciona de mejor manera solo con equipos reducidos.

Encuestado 9: Quizás sea difícil de utilizar en equipos grandes o cuando el proyecto sea más complejo de lo que se esperaba. Las tareas deben definirse según el plazo de tiempo dado.

Encuestado 10: La implementación inadecuada da pie a exceso de burocracia y actividades que no contribuyen directamente al proyecto, retrasando de esa manera la entrega de valor.

Encuestado 11: Dificultad de implementación.

11. ¿Considera que el uso de Scrum fomenta que el desarrollo de software sea eficaz y eficiente?

Encuestado 1: Sí.

Encuestado 2: Sí.

Encuestado 3: Sí.

Encuestado 4: Sí.

Encuestado 5: Sí.

Encuestado 6: Sí.

Encuestado 7: Sí.

Encuestado 8: Sí.

Encuestado 9: Sí.

Encuestado 10: Sí.

Encuestado 11: Sí.

12. ¿Qué aspectos considera que deben o pueden ser mejorados dentro de los procesos de desarrollo de software de SI UDB?

Encuestado 1: Adaptación de la metodología al día a día.

Encuestado 2: La comunicación entre los equipos de trabajo, para saber que desarrolla el otro equipo.

Encuestado 3: Por el momento no tengo presente aspectos a considerar.

Encuestado 4: Comunicación sobre fase de desarrollo y requerimientos del cliente.

Encuestado 5: Control de los requerimientos para los proyectos y creación de un plan de desarrollo de software al inicio de cada proyecto.

Encuestado 6: Crear un documento con los requerimientos del sistema brindados por el Product Owner, que el software se publique hasta que se hayan resuelto todos los bug, que se realice la demostración del producto a todos los involucrados siempre.

Encuestado 7: Definición de requerimientos acorde a los deseos del cliente y no al del product owner. Reuniones con el product owner antes de un release del proyecto.

Encuestado 8: El Refinamiento de la lista de requerimientos y cambios en los proyectos.

Encuestado 9: Es funcional.

Encuestado 10: La definición de requerimientos durante todo el proceso de desarrollo, comunicación con los clientes y definiciones claras del valor otorgado por los proyectos emprendidos.

Encuestado 11: El involucramiento de los clientes, convirtiéndolos en verdaderos Product Owner.

13. ¿Usted estaría abierto a adoptar una nueva metodología/framework ágil dentro de SI UDB?

Encuestado 1: Sí.

Encuestado 2: Sí.

Encuestado 3: Sí.

Encuestado 4: Sí.

Encuestado 5: Sí.

Encuestado 6: No.

Encuestado 7: Sí.

Encuestado 8: Sí.

Encuestado 9: Sí.

Encuestado 10: Sí.

Encuestado 11: Sí.

14. ¿Cuál considera que podría ser la mayor dificultad a la hora de adoptar una nueva metodología/framework ágil?

Encuestado 1: Constancia.

Encuestado 2: Adaptación, la etapa inicial de un cambio.

Encuestado 3: Que no todos los involucrados tengan conocimientos sobre la nueva metodología.

Encuestado 4: La generación de una cultura en la organización.

Encuestado 5: Resistencia al cambio.

Encuestado 6: La adaptación al cambio por parte del equipo.

Encuestado 7: Gestionar y acoplar cada una de las actividades en el marco de trabajo de la nueva metodología en cada uno de los procesos de negocio que se realizan en el departamento.

Encuestado 8: Ser capaz de acoplarse de manera rápida a cómo se organizan los flujos de trabajos y demás actividades.

Encuestado 9: La gestión y actitud de los equipos ante el cambio de metodología. También que cada proyecto es diferente, según las necesidades y recursos que se manejen, lo largo y/o complicado del proyecto.

Encuestado 10: La refundación de procesos arraigados dentro de la cultura del equipo y el impacto inicial en el tiempo de necesario para la generación de valor.

Encuestado 11: Los cambios de mentalidad y paradigmas siempre son difíciles.

15. ¿Conoce o ha escuchado sobre la metodología Agile 2?

Encuestado 1: No.

Encuestado 2: No.

Encuestado 3: No.

Encuestado 4: No.

Encuestado 5: No.

Encuestado 6: No.

Encuestado 7: No.

Encuestado 8: No.

Encuestado 9: No.

Encuestado 10: Sí.

Encuestado 11: Sí.

16. ¿Ha cursado capacitaciones o cursos sobre metodologías/frameworks ágiles?

Encuestado 1: Sí.

Encuestado 2: Sí.

Encuestado 3: No.

Encuestado 4: Sí.

Encuestado 5: Sí.

Encuestado 6: Sí.

Encuestado 7: No.

Encuestado 8: No.

Encuestado 9: No.

Encuestado 10: Sí.

Encuestado 11: Sí.

17. ¿Posee alguna certificación sobre metodologías/frameworks ágiles? En caso de que su respuesta sea afirmativa, por favor mencione que certificación posee.

Encuestado 1: Scrum Fundamentals.

Encuestado 2: No poseo.

Encuestado 3: No.

Encuestado 4: Scrum certification foundations.

Encuestado 5: No.

Encuestado 6: No.

Encuestado 7: No.

Encuestado 8: No.

Encuestado 9: No.

Encuestado 10: Ninguna.

Encuestado 11: Scrum Master de Scrumstudy.

Anexo B - Encuesta final realizada a los trabajadores de la organización objetivo

Tema: Encuesta final realizada a los trabajadores del Departamento de Sistemas Informáticos UDB.

Objetivo: Recopilar la opinión de los miembros del Departamento de Sistemas Informáticos respecto a ejecución de la guía de implementación del sistema de gestión SI Board.

Encuestador: Joshua Villavicencio.

Encuestados: Trabajadores del Departamento de Sistemas Informáticos UDB.

Cantidad de personas encuestadas: 12.

Fecha de realización: 17 de enero de 2022.

Formulario de la Encuesta

A continuación, en la figura B1 se presenta el formulario con las preguntas realizadas a los miembros de la organización para conocer sus impresiones tras la implementación de la guía para crear el sistema de gestión SI Board.

Figura B1

Encuesta de satisfacción de aplicación de la guía

1. Nombre
2. Correo electrónico
3. Puesto
4. ¿Participó usted en el desarrollo de SI Board?
5. ¿Participó usted en la implementación de la guía de Agile 2 en el departamento?
6. ¿Considera que la guía posee objetivos claros?
7. ¿Considera que la guía es fácil de implementar?
8. ¿Considera que la implementación de la guía ha mejorado el desempeño de sus labores?
9. ¿Cuál de las 5 categorías de estrategias considera es más útil en el desempeño de sus labores?
10. ¿Cuál categoría considera que puede mejorarse?
11. ¿Tiene alguna estrategia que desearía estuviera incluida en la guía?
12. ¿Tiene alguna herramienta que desearía estuviera incluida en la guía?

Respuestas Obtenidas

1. Nombre

Las respuestas a esta pregunta son anónimas.

2. Correo electrónico

Las respuestas a esta pregunta son anónimas.

3. Puesto

Encuestado 1: Arquitecto de software.

Encuestado 2: Analista de sistemas.

Encuestado 3: Arquitecto de software.

Encuestado 4: Arquitecto de software.

Encuestado 5: Arquitecto de software.

Encuestado 6: QA.

Encuestado 7: Analista de sistemas.

Encuestado 8: Analista de sistemas.

Encuestado 9: Analista de sistemas.

Encuestado 10: Analista de sistemas.

Encuestado 11: Jefe de departamento.

Encuestado 12: QA.

4. ¿Participó usted en el desarrollo de SI Board?

Encuestado 1: Si.

Encuestado 2: Si.

Encuestado 3: Si.

Encuestado 4: Si.

Encuestado 5: Si.

Encuestado 6: No.

Encuestado 7: Si.

Encuestado 8: Si.

Encuestado 9: Si.

Encuestado 10: No.

Encuestado 11: Si.

Encuestado 12: Si.

5. ¿Participó usted en la implementación de la guía de Agile 2 en el departamento?

Encuestado 1: Si.

Encuestado 2: Si.

Encuestado 3: Si.

Encuestado 4: Si.

Encuestado 5: Si.

Encuestado 6: Si.

Encuestado 7: Si.

Encuestado 8: Si.

Encuestado 9: Si.

Encuestado 10: Si.

Encuestado 11: Si.

Encuestado 12: Si.

6. ¿Considera que la guía posee objetivos claros?

Encuestado 1: Si.

Encuestado 2: Si.

Encuestado 3: Si.

Encuestado 4: No.

Encuestado 5: Si.

Encuestado 6: Si.

Encuestado 7: Si.

Encuestado 8: Si.

Encuestado 9: Si.

Encuestado 10: Si.

Encuestado 11: Si.

Encuestado 12: Si.

7. ¿Considera que la guía es fácil de implementar

Encuestado 1: Si.

Encuestado 2: Si.

Encuestado 3: Si.

Encuestado 4: Si.

Encuestado 5: Si.

Encuestado 6: No.

Encuestado 7: Si.

Encuestado 8: Si.

Encuestado 9: No.

Encuestado 10: No.

Encuestado 11: Si.

Encuestado 12: Si.

8. ¿Considera que la implementación de la guía ha mejorado el desempeño de sus labores?

Encuestado 1: Si.

Encuestado 2: Si.

Encuestado 3: Si.

Encuestado 4: Si.

Encuestado 5: Si.

Encuestado 6: Si.

Encuestado 7: Si.

Encuestado 8: Si.

Encuestado 9: Si.

Encuestado 10: Si.

Encuestado 11: Si.

Encuestado 12: Si

9. ¿Cuál de las 5 categorías de estrategias considera es más útil en el desempeño de sus labores?

Encuestado 1: Empoderamiento del rol de arquitecto de software.

Encuestado 2: Organización del equipo de desarrollo.

Encuestado 3: Empoderamiento del rol de arquitecto de software.

Encuestado 4: Empoderamiento del rol de arquitecto de software.

Encuestado 5: Levantamiento de requerimientos.

Encuestado 6: Empoderamiento del equipo de QA.

Encuestado 7: Nuevas herramientas ágiles.

Encuestado 8: Nuevas herramientas ágiles.

Encuestado 9: Organización del equipo de desarrollo.

Encuestado 10: Nuevas herramientas ágiles.

Encuestado 11: Nuevas herramientas ágiles.

Encuestado 12: Empoderamiento del equipo de QA.

10. ¿Cuál categoría considera que puede mejorarse?

Encuestado 1: Ninguna.

Encuestado 2: Ninguna.

Encuestado 3: Ninguna.

Encuestado 4: Ninguna.

Encuestado 5: Ninguna.

Encuestado 6: Ninguna.

Encuestado 7: Levantamiento de requerimientos.

Encuestado 8: Ninguna.

Encuestado 9: Levantamiento de requerimientos.

Encuestado 10: Organización del equipo de desarrollo.

Encuestado 11: Ninguna.

Encuestado 12: Ninguna.

11. ¿Tiene alguna estrategia que desearía estuviera incluida en la guía?

Encuestado 1: Ninguna.

Encuestado 2: Ninguna.

Encuestado 3: Ninguna.

Encuestado 4: Estrategias para promover el desarrollo móvil.

Encuestado 5: Ninguna.

Encuestado 6: Ninguna.

Encuestado 7: Ninguna.

Encuestado 8: Ninguna.

Encuestado 9: Una forma más clara de establecer los cambios en los requerimientos que se presentan.

Encuestado 10: Formas más fáciles de incorporar a nuevos miembros del equipo a la

implementación de la guía.

Encuestado 11: Ninguna.

Encuestado 12: Ninguna.

12. ¿Tiene alguna herramienta que desearía estuviera incluida en la guía?

Encuestado 1: Ninguna.

Encuestado 2: Ninguna.

Encuestado 3: Ninguna.

Encuestado 4: Ninguna.

Encuestado 5: Ninguna.

Encuestado 6: Ninguna.

Encuestado 7: Ninguna.

Encuestado 8: Ninguna.

Encuestado 9: Ninguna.

Encuestado 10: Ninguna.

Encuestado 11: Ninguna.

Encuestado 12: Herramientas para administrar los casos de uso empleados durante el proceso de QA.

Anexo C - Entrevista Realizada al jefe del Departamento de Sistemas Informáticos

Tema: Entrevista al jefe del Departamento de Sistemas Informáticos para conocer cómo fue el proceso de adopción de Scrum y sus expectativas sobre Agile 2.

Objetivo: Entrevistar al jefe del Departamento de Sistemas Informáticos para saber cómo se implementó el marco de trabajo Scrum en el año 2018 dentro de dicho departamento y qué expectativas tiene sobre la posible adopción de la metodología Agile 2.

Entrevistador: Joshua Villavicencio.

Entrevistado: Jefe del Departamento de Sistemas Informáticos UDB.

Tipo de entrevista: Entrevista estructurada.

Fecha de realización: 24 de enero de 2022.

Entrevista Realizada

1. ¿De qué manera actúa la figura del Product Owner en cada proyecto?

Actualmente lo que se está haciendo es que el director de CTIC y el jefe de sistemas (jefe del Departamento de Sistemas Informáticos UDB) están actuando prácticamente como Product Owner, eso es una deuda que tenemos de poder incorporar al verdadero Product Owner al equipo, es decir, a la dinámica de trabajo, porque para mí el verdadero Product Owner es el único que nos puede garantizar los mejores requerimientos.

2. ¿De qué manera se estima el tiempo total de desarrollo de un proyecto?

Ahorita tampoco hay un proceso, lo hacemos en base a aproximados, en cada proyecto decimos “esta tarea se aproxima a esto”, pero hemos estado hablando de usar pronósticos para ir viendo si en verdad cumplimos, pero no hay tiempos en verdad bien exactos.

3. ¿Qué procedimiento se sigue para adoptar cambios durante el desarrollo de un proyecto de Software que está actualmente en curso?

Procedimiento si no hay, lo que hacemos es que los cambios solicitados se meten en el log del desarrollo, se ponen como un requerimiento más, pero en verdad es bien informal ese proceso, no tenemos un proceso formal de control de cambios.

4. ¿Qué lo motivó en su momento a adoptar Scrum dentro de SI UDB?

En primer lugar yo tenía ya años de estar usando, empapándome y proponiendo metodologías ágiles, quizás desde el 2014 o 2015, así entonces cuando llegué (cuando inició labores como jefe en 2018) hice un diagnóstico de que había muchos problemas, por ejemplo, no había un Git o algo para manejar las versiones, no había un manejador de versiones y era una locura, llevar las versiones era súper complicado y también había un papel central en donde el jefe o el director de CTIC hacían lo más difícil y a la demás gente solo le repartían lo más básico, entonces eso lo que

crea es debilidad en el equipo porque no le da al equipo la experiencia ni el poder para poder manejar las demás cosas, entonces ahí vi que era una buena opción cambiar y en verdad se mejoró, ya que en los últimos cinco años hemos hecho más proyectos que en los 10 años anteriores, y además de eso, los proyectos anteriores se renovaron, quizás nada más un proyecto no se ha renovado pero es porque ya no se va a usar, pero de ahí todos se renovaron, eso quiere decir que haber adoptado un marco de trabajo ágil funciona porque se produjo más y se renovaron totalmente los proyectos anteriores, empezamos a usar MVC (modelo vista controlador) y estamos más o menos actuales.

5. ¿Cuáles considera usted que fueron las mayores dificultades al adoptar Scrum?

Primero, siempre cuando uno hace un cambio siempre hay una reacción al cambio, nadie quiere cambiar porque todo mundo está acostumbrado a algo y el cambio siempre genera miedo, segundo, la agilidad lastimosamente al jefe le quita mucho poder porque la idea es que todo sea colaborativo y en las metodologías ágiles las jerarquías son horizontales o eso tratan, sobre todo el que más sufre cuando se implementa una metodología ágil son las jefaturas porque no están acostumbradas a delegar ni están acostumbradas a perder poder, entonces deben de entender que su rol pasa a ser un líder y a dejar de ser jefe, en las metodologías ágiles no existen jefes, sólo líderes, además de eso, debe de ser un líder servicial, entonces pasar de ser jefe a líder servicial y eso no todo mundo lo puede digerir, entonces eso es lo más difícil, de ahí, a la gente si la

llevamos poco a poco acepta el cambio y lo celebra. Si se quiere implementar de un solo, no funciona.

6. ¿La adopción de Scrum debe de ser gradual?

Sí, porque uno puede llegar a un grupo donde todo mundo ya sabe qué es ágil, que es Scrum, que es esto, que es lo otro, entonces ahí es bien fácil (la adopción se vuelve fácil), pero muchas veces uno llega a un grupo donde no saben nada, entonces hay que capacitar primero, por ejemplo, una cosa de las que adopté es que todo mundo nos tratáramos igual, en el sentido de no decirnos puestos y tratar de invertir en el nosotros y no en el yo, para ágil todo es nosotros pero eso también es difícil porque el ego es de lo más complicado de manejar en una metodología ágil, entonces el proceso debe ser gradual, poco a poco, y aun así hay gente que es muy cuadrada, entonces uno debe de ser disruptivo, ser innovador, ser creativo, esas son nuevas skills (habilidades) que si alguien no los tiene, pues no sirve para ágil, es decir, hay que fomentar esas nuevas skills.

7. ¿Qué expectativas tiene usted respecto a la posible adopción de una nueva metodología en SI UDB, en este caso específicamente hablando de Agile 2?

Según lo que conozco de Agile 2 o lo que más o menos he entendido, a mí me gusta la propuesta en el punto de vista de que va más directa a lo que es desarrollo de software, un poco lo que es DevOps con Scrum. Actualmente, casi todo mundo dice que hay una división entre DevOps y

Scrum y quizás eso debe estar unido, entonces ese sería el primer punto que me gusta de Agile 2, que une estas dos metodologías, el otro punto que me gusta es que va más enfocado al software y hay algún par de cosas más que me gustan, como por ejemplo que habla más sobre prueba y error, y creo que es una metodología un poco ya más madura, entonces todo cambia en la vida y hay que adaptarse también a los cambios y adaptarlos lo más pronto que se pueda.

Anexo D - Entrevista Realizada a Cliff Berg, Principal Fundador de Agile 2

Tema: Entrevista a Cliff Berg, principal fundador de la metodología Agile 2.

Objetivo: Entrevistar a Cliff Berg para conocer cuáles fueron las razones que lo impulsaron a fomentar la creación de la metodología Agile 2 y para, además, obtener consejos de su parte sobre cómo llevar a cabo una correcta y exitosa adopción de la metodología.

Entrevistador: Mario Menjivar.

Entrevistado: Cliff Berg.

Tipo de entrevista: Entrevista estructurada.

Fecha de realización: 9 de junio de 2022.

Entrevista Realizada Versión Original en Idioma Inglés

- 1. What was the main reason that motivated you to promote the creation of Agile 2?**

Organizations need agility - now more than ever - but there is abundant evidence that the approaches being advocated by much of the “Agile community” do not work for most organizations.

The core ideas of Agile are sound, but they are too extreme and too simplistic. We felt that a more robust, more nuanced, and more science-based approach was needed. We did not want to “throw out” the core ideas of Agile, but rather to adjust them to make them more effective.

2. What do you think could be the biggest obstacle for an organization that intends to adopt Agile 2?

Executives must understand that agility is not something that they can “install” or “roll out” like a plan. Agility arises from behavior - not from business processes. Thus, adopting processes such as SAFe (Scaled Agile Framework) will not result in agility. Achieving agility is a protracted learning journey. It is necessary to address behavior, knowledge (of Lean and Flow patterns), and organizational culture. Process is irrelevant.

3. In which environment do you think Agile 2 can be best adopted, in large or small companies?

Company size does not matter.

4. Do you think there is a roadmap to follow to adopt Agile 2 or should each organization design its own roadmap based on its characteristics and needs?

There is a roadmap. It is called Constructive Agility. Rather than being a workflow process like SAFe or Scrum, it is a path through which organizations define their own processes, learning new behaviors along the way.

5. How do you think it can be measured whether an organization has successfully adopted Agile2?

By measuring outcomes.

6. What vision or expectations do you have for Agile 2 in the future?

We believe that Agile 2 represents more recent writing about agility. This article explains why:

<https://dzone.com/articles/old-versus-new-agile>

7. What would be the main advice you would give to a company that wants to adopt Agile2?

To realize that agility is extremely important, and that it cannot be attained by following one of the Agile “frameworks” - but that one must understand how agility actual will manifest in one’s organization. Also, to recognize that creative work such as product development is not like manufacturing: it does not consist of repeatable processes. That is why for such work, agility will not result from changing one’s process.

Entrevista Realizada Versión Traducida al Idioma Español

1. ¿Cuál fue la razón principal que lo motivó a impulsar la creación de Agile 2?

Las organizaciones necesitan agilidad, ahora más que nunca, pero existe abundante evidencia de que los enfoques defendidos por gran parte de la "comunidad ágil" no funcionan para la mayoría de las organizaciones.

Las ideas centrales de Agile son sólidas, pero son demasiado extremas y simplistas. Sentimos que se necesitaba un enfoque más sólido, más matizado y más basado en la ciencia. No queríamos "desechar" las ideas centrales de Agile, sino ajustarlas para hacerlas más efectivas.

2. ¿Cuál cree que podría ser el mayor obstáculo para una organización que pretende adoptar Agile 2?

Los ejecutivos deben comprender que la agilidad no es algo que puedan "instalar" o "implementar" como un plan. La agilidad surge del comportamiento, no de los procesos comerciales. Por lo tanto, adoptar procesos como SAFe (Scaled Agile Framework) no resultará en agilidad. Lograr la agilidad es un viaje de aprendizaje prolongado. Es necesario abordar el comportamiento, el conocimiento (de los patrones Lean y Flow) y la cultura organizacional. El proceso es irrelevante.

3. ¿En qué entorno cree que se puede adoptar mejor Agile 2, en empresas grandes o pequeñas?

El tamaño de la empresa no importa.

4. ¿Cree que hay una hoja de ruta a seguir para adoptar Agile 2 o cada organización debe diseñar su propia hoja de ruta en función de sus características y necesidades?

Hay una hoja de ruta. Se llama Habilidad Constructiva. En lugar de ser un proceso de flujo de trabajo como SAFe o Scrum, es un camino a través del cual las organizaciones definen sus propios procesos, aprendiendo nuevos comportamientos en el camino.

5. ¿Cómo cree que se puede medir si una organización ha adoptado con éxito Agile 2?

Midiendo los resultados.

6. ¿Qué visión o expectativas tiene para Agile 2 en el futuro?

Creemos que Agile 2 representa la escritura más reciente sobre la agilidad. Este artículo explica por qué: <https://dzone.com/articles/old-versus-new-agile>.

7. ¿Cuál sería el principal consejo que le daría a una empresa que quiera adoptar Agile 2?

Darse cuenta de que la agilidad es extremadamente importante y que no se puede lograr siguiendo uno de los "marcos de trabajo" ágiles, pero que uno debe comprender cómo se manifestará la agilidad real en la propia organización. Además, hay que reconocer que el trabajo

creativo, como el desarrollo de productos, no es como la fabricación: no consiste en procesos repetibles. Es por eso que, para tal trabajo, la agilidad no resultará de cambiar el proceso de uno.

Anexo E - Fichas de Registro de Observación

Tema: Fichas de observación utilizadas para recopilar los hallazgos encontrados a través del proceso de observación llevado a cabo en el Departamento de Sistemas Informáticos UDB.

Objetivo: Observar y documentar las labores diarias realizadas por los miembros del Departamento de Sistemas Informáticos UDB.

Observadores: Mario Menjivar y Joshua Villavicencio.

Población a Observar: Todos los miembros del Departamento de Sistemas Informáticos UDB.

Fecha de realización: Del 24 al 27 de enero de 2022.

En la figura E1 se puede observar el formato de las fichas de registro de observación que se utilizaron para recabar información relevante sobre la manera en que laboran los miembros de la organización.

Figura E1

Formato de las fichas de registro de observación

Ficha de registro de observación	
Implementador:	Fecha:

Lugar:	Hora:
Actividad:	
Objetivo:	
Descripción de lo observado	
Análisis:	

Del 24 al 27 de enero de 2022, se realizaron visitas presenciales al Departamento de Sistemas Informáticos UDB, en dicho periodo de tiempo se observaron y documentaron las labores realizadas por los miembros de la organización. Cada día se observó a los miembros de un rol distinto, los roles observados fueron:

- Jefe del Departamento de Sistemas Informáticos UDB.
- Arquitectos de software.
- Analistas de sistemas.
- QA tester.

Las fichas de observación utilizadas se presentan a continuación en las Figuras E2, E3, E4 y E5.

Figura E2

Ficha de registro de observación tomada el 24 de enero de 2022

Ficha de registro de observación	
Implementador: Mario Menjivar	Fecha: 24 de enero de 2022
Lugar: Departamento de Sistemas Informáticos UDB	Hora: de 8 am a 5 pm
Actividad: Evaluación de las labores del jefe del Departamento de Sistemas Informáticos UDB.	
Objetivo: Identificar todas las actividades que realiza el jefe del Departamento de Sistemas Informáticos UDB en un día normal de labores.	
<p>Descripción de lo observado: En el periodo de tiempo en que se implementó este instrumento, se observó el siguiente comportamiento en las actividades que realiza el jefe del departamento:</p> <ul style="list-style-type: none"> - El jefe del departamento es la figura más importante dentro de la organización ya que él gestiona al equipo de desarrolladores y al equipo de QA, además, él está en constante comunicación con el jefe del área de TI de la UDB. - Diariamente, él asiste a varias reuniones con los clientes. - Tiene un alto dominio de conocimientos técnicos y también conoce sobre la lógica del negocio. - Denota ser una persona capacitada para el puesto. - El fomenta un buen ambiente amigable dentro de la organización. - Diariamente realiza muchas tareas debido a que una gran parte de las responsabilidades de liderazgo dentro de la organización, recaen sobre él. - Él es quien se encarga de asignar las tareas que llevan a cabo los equipos de desarrollo y de QA. 	

<ul style="list-style-type: none"> - Habitualmente, los demás miembros del equipo de trabajo le consultan sobre dudas que ellos poseen y que son referentes a los distintos proyectos que se tienen en desarrollo. - Él puede fungir como Product Owner en proyectos en donde no hay un Product Owner proveniente del lado del cliente.
<p>Análisis: En base a lo observado se puede determinar lo siguiente del jefe del Departamento de Sistemas Informáticos UDB:</p> <ul style="list-style-type: none"> - El jefe del departamento es el eje bajo el que gira la organización, su figura es de suma importancia, lo cual puede generar problemas en caso de que él se llegue a ausentar. - Las labores de liderazgo en su mayoría están centralizadas bajo su figura. - Es una persona muy competente que realiza de muy buena manera sus labores.

Figura E3

Ficha de registro de observación tomada el 25 de enero de 2022

Ficha de registro de observación	
Implementador: Mario Menjivar	Fecha: 25 de enero de 2022
Lugar: Departamento de Sistemas Informáticos UDB	Hora: 8 am a 5 pm
Actividad: Observación de las labores de los arquitectos de software.	
Objetivo: Identificar todas las actividades que realizan los arquitectos de software en un día normal de labores.	

Descripción de lo observado: En el periodo de tiempo en que se implementó este instrumento, se observó el siguiente comportamiento en las actividades que realizan los arquitectos de software:

- Se encargan de desarrollar los diferentes proyectos de software que se llevan a cabo en la organización.
- Denotan un buen dominio de conocimientos técnicos y tienen cierto dominio de la lógica del negocio de los proyectos que desarrollan.
- Ocasionalmente se reúnen con el jefe del departamento para discutir cambios que se realizarán en los proyectos en desarrollo, luego ellos se encargan de transmitir estos cambios a los demás miembros del equipo.
- Tienen a su cargo a pequeños equipos de trabajo, los cuales están formados por el arquitecto como tal y uno o más analistas de sistemas. Cabe añadir que algunos equipos también cuentan con un QA tester dentro de sus miembros.
- Los demás miembros del equipo al que tienen a su cargo, frecuentemente les consultan sobre dudas referentes a los proyectos que están desarrollando.
- Los analistas de sistemas con mayor longevidad dentro de la organización y que demuestran un buen rendimiento, son los principales candidatos para promover al puesto de arquitecto de software.
- La cantidad y elección de arquitectos de software con los que cuenta la organización, son determinados por el jefe del departamento en conjunto con el jefe del área de TI de la Universidad Don Bosco.
- Diariamente, brinda a su equipo de trabajo una actualización de sus actividades en la reunión Daily Scrum.
- Cuando necesitan realizar cambios a los elementos de una base de datos, crean un script con la lógica del cambio a implementar y dicho script se lo hacen llegar al director de CTIC para su posterior ejecución.

Análisis: En base a lo observado se puede determinar lo siguiente de los arquitectos de software:

- Los arquitectos de software realizan labores de desarrollo de software a la vez que tienen la responsabilidad de ayudar a los analistas de sistemas que tienen a su cargo.
- Los arquitectos de software están abiertos a poder ayudar a los miembros de su equipo.
- En ocasiones, los arquitectos de software fungen como un canal de transmisión de información entre el jefe del departamento y los analistas de sistemas.

Figura E4

Ficha de registro de observación tomada el 26 de enero de 2022

Ficha de registro de observación	
Implementador: Joshua Villavicencio	Fecha: 26 de enero de 2022
Lugar: Departamento de Sistemas Informáticos UDB	Hora: 8 am a 5 pm
Actividad: Observación de las labores de los analistas de sistemas.	
Objetivo: Identificar todas las actividades que realizan los analistas de sistemas en un día normal de labores.	
<p>Análisis: En base a lo observado se puede determinar lo siguiente de los analistas de sistemas:</p> <ul style="list-style-type: none"> - Ellos se encargan de desarrollar los proyectos de software que le son solicitados a la organización. 	

- Si bien, poseen un buen dominio de conocimientos técnicos, aun presentan falencias o desconocimiento sobre algunos elementos de la lógica del negocio de los proyectos en los que trabajan, esto se puede deber a falta de experiencia o que, al inicio del proyecto no se les brindó una correcta orientación y explicación sobre dicha lógica del negocio.
- Los analistas de sistemas realizan la hoja de ruta y la estimación de tiempo de desarrollo de los proyectos que llevan a cabo, esto puede ocasionar que estos instrumentos no devuelvan resultados eficientes o exactos debido a la falta de experiencia de los analistas de sistemas realizando estas actividades.

Descripción de lo observado: En el periodo de tiempo en que se implementó este instrumento, se observó el siguiente comportamiento en las actividades que realizan los analistas de sistemas:

- Sus labores diarias se enfocan en desarrollar los proyectos de software que los clientes solicitan.
- Cada analista de sistemas pertenece a un equipo de trabajo el cual está liderado por un arquitecto de software.
- Los analistas de sistemas, son los miembros del equipo de trabajo que cuentan menor experiencia y longevidad dentro de la organización.
- Cuando ellos han finalizado de desarrollar un proyecto, ellos le hacen llegar dicho proyecto al equipo de QA para que ellos evalúen si este cumple con los requerimientos y expectativas del cliente.
- Los analistas de sistemas denotan dominio de conocimientos técnicos, sin embargo, en ocasiones denotan desconocimiento respecto a algunos elementos de la lógica del negocio de los proyectos que desarrollan.

- Para solventar dudas que poseen sobre los proyectos en los que están trabajando, le consultan sus dudas al arquitecto de software del equipo al que pertenecen o al jefe del departamento.
- Diariamente, brinda a su equipo de trabajo una actualización de sus actividades en la reunión Daily Scrum.
- Cuando a un analista de sistemas se le asigna el desarrollo de en un nuevo proyecto, el cual será desarrollado desde cero, es responsabilidad de dicho analista de sistemas el determinar la hoja de ruta a seguir para llevar a cabo ese proyecto, esa hoja de ruta se crea a partir de los requerimientos que el cliente haya brindado.
- Ellos son los encargados de realizar la estimación de tiempo de desarrollo de un proyecto, esta estimación la realizan en base a su experiencia.
- Al igual que los arquitectos de software, gestionan los cambios a los elementos de las bases de datos a través de scripts.

Figura E5

Ficha de registro de observación tomada el 27 de enero de 2022

Ficha de registro de observación	
Implementador: Mario Menjivar	Fecha: 27 de enero de 2022
Lugar: Departamento de Sistemas Informáticos UDB	Hora: 8 am a 5 pm
Actividad: Observación de las labores de los QA tester.	

Objetivo: Identificar todas las actividades que realizan los QA tester en un día normal de labores.

Descripción de lo observado: En el periodo de tiempo en que se implementó este instrumento, se observó el siguiente comportamiento en las actividades que realizan los QA tester:

- Los QA tester pertenecen a un equipo de trabajo pequeño en donde trabajan en conjunto con un arquitecto de software y uno o más analistas de sistemas.
- Una de las labores de los QA tester es identificar los requerimientos funcionales que debe de poseer un proyecto una vez que este esté finalizado, en base a ello determinan los criterios a evaluar para determinar si ese proyecto ha sido desarrollado de manera correcta.
- Los QA tester almacenan en Google Drive toda la documentación que se genera cuando se desarrolla un proyecto.
- Las pruebas que realizan para validar que un proyecto cumple con su cometido, son realizadas de manera manual, no se realizan pruebas de manera automatizada.
- Ellos participan en las diversas ceremonias Scrum que se llevan a cabo dentro de la organización, son ejemplo de dichas ceremonias las reuniones Sprint Planning, Sprint Retrospective o las reuniones de periodicidad diaria Daily Scrum.
- Una vez que un desarrollador de software ha finalizado un proyecto, dicho proyecto se transfiere a un QA tester el cual valida su correcto funcionamiento.
- Una vez que el QA tester ha validado un proyecto, este proyecto es presentado al jefe del departamento para que apruebe dicho desarrollo, si esta aprobación ocurre, el jefe del departamento se reúne con el jefe del área de TI de la Universidad Don Bosco para discutir los detalles del lanzamiento del proyecto.

Análisis: En base a lo observado se puede determinar lo siguiente de los QA tester:

- Sus labores diarias están muy involucradas con las de los desarrolladores de software, esto debido a que los QA tester velan por la correcta implementación de las funcionalidades que deben poseer los proyectos realizados por los desarrolladores de software.
- Ellos participan en los diversos instrumentos Scrum que se llevan a cabo en la organización.
- Los QA tester son el primer filtro que determina si un desarrollo de software ha sido realizado de manera correcta, siendo los siguientes filtros, el jefe del departamento y el jefe del área de TI de la Universidad Don Bosco.

Anexo F - Ejemplo de Historia Épica

Tema: Ejemplo de Historia épica.

Objetivo: Presentar un ejemplo de una historia épica, esta es presentada desde el punto de vista de un Product Owner pues se asume que es esta persona quien solicita los elementos contenidos en la historia épica.

Desarrollo de SI Board

Los integrantes del equipo del departamento de sistemas informáticos necesitan una herramienta que les permita documentar los procesos internos llevados a cabo cuando se requiere un producto nuevo o una modificación por parte de los clientes.

Como Product Owner, solicito un registro de todos los productos desarrollados y mantenidos por el departamento, documentar el desarrollo de productos nuevos y cambios en

productos existentes, dicha documentación debe incluir los requerimientos del cliente, el equipo asignado al desarrollo, así como todo documento generado durante el proceso de aseguramiento de la calidad. Los desarrollos deben incluir la fecha prevista de finalización, el estado del desarrollo (activo, QA, certificado o demorado).

Los miembros del departamento deben de ser capaces de crear historias de usuario que registren el trabajo desempeñado por cada uno de ellos en los diferentes desarrollos existentes.

Los miembros del departamento deben de ser capaces de administrar los sprints de desarrollo, establecer fechas de inicio y finalización, agrupar historias de usuario de acuerdo con los miembros que participan en un sprint seleccionado y tener la capacidad de generar una retrospectiva del sprint por parte de sus participantes. Cada sprint debe poseer un tablero en donde los integrantes puedan mover sus historias de usuario entre las categorías To Do, Doing y Done.

Los usuarios deben de tener una serie de herramientas para la gestión de las diferentes actividades realizadas por el departamento: ser capaces de almacenar los scripts utilizados en los diferentes desarrollos, así como realizar entregas relacionadas a soluciones de tickets utilizando la misma interfaz, organización de los manuales creados para los productos entregados, una interfaz donde pueda configurarse el diseño de los correos generados automáticamente por los diferentes productos disponibles y una interfaz en la cual se lleve un registro de la deuda técnica encontrada por el equipo de QA.

Anexo G - Ejemplo de User Story Map

Tema: Ejemplo de User Story Map.

Objetivo: Presentar un ejemplo de uso de un User Story Map.

A continuación, se presenta un ejemplo de uso de un User Story Map utilizando la herramienta externa Miro. En este ejemplo se representan todas las actividades realizadas durante el desarrollo del sistema de gestión SI Board. El tamaño total del User Story Map es extenso, por ello, para una mejor lectura se ha dividido este recurso en 5 partes, mostrando cada parte un área específica de dicho recurso, a continuación, se detalla el contenido de cada parte:

- En la parte 1 (figura G1), se presentan las actividades relacionadas con el manejo del catálogo de productos con el que cuenta la organización.
- En la parte 2 (figura G2), se presentan las actividades relacionadas con la creación de historias de usuario.
- En la parte 3 (figura G3), se presentan las actividades relacionadas con el desarrollo de proyectos de software.
- En la parte 4 (figura G4), se presentan las actividades relacionadas con el manejo de sprints.
- En la parte 5 (figura G5), se presentan las actividades relacionadas con la gestión de la documentación generada en la organización.

Figura G1

User Story Map del producto SI Board Parte 1

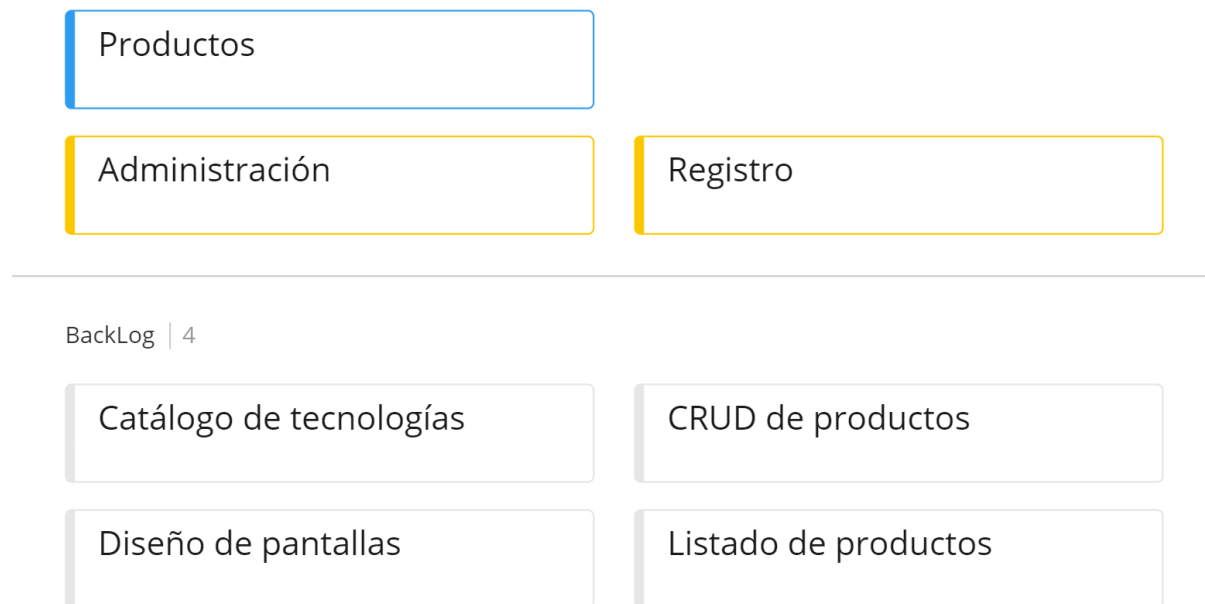


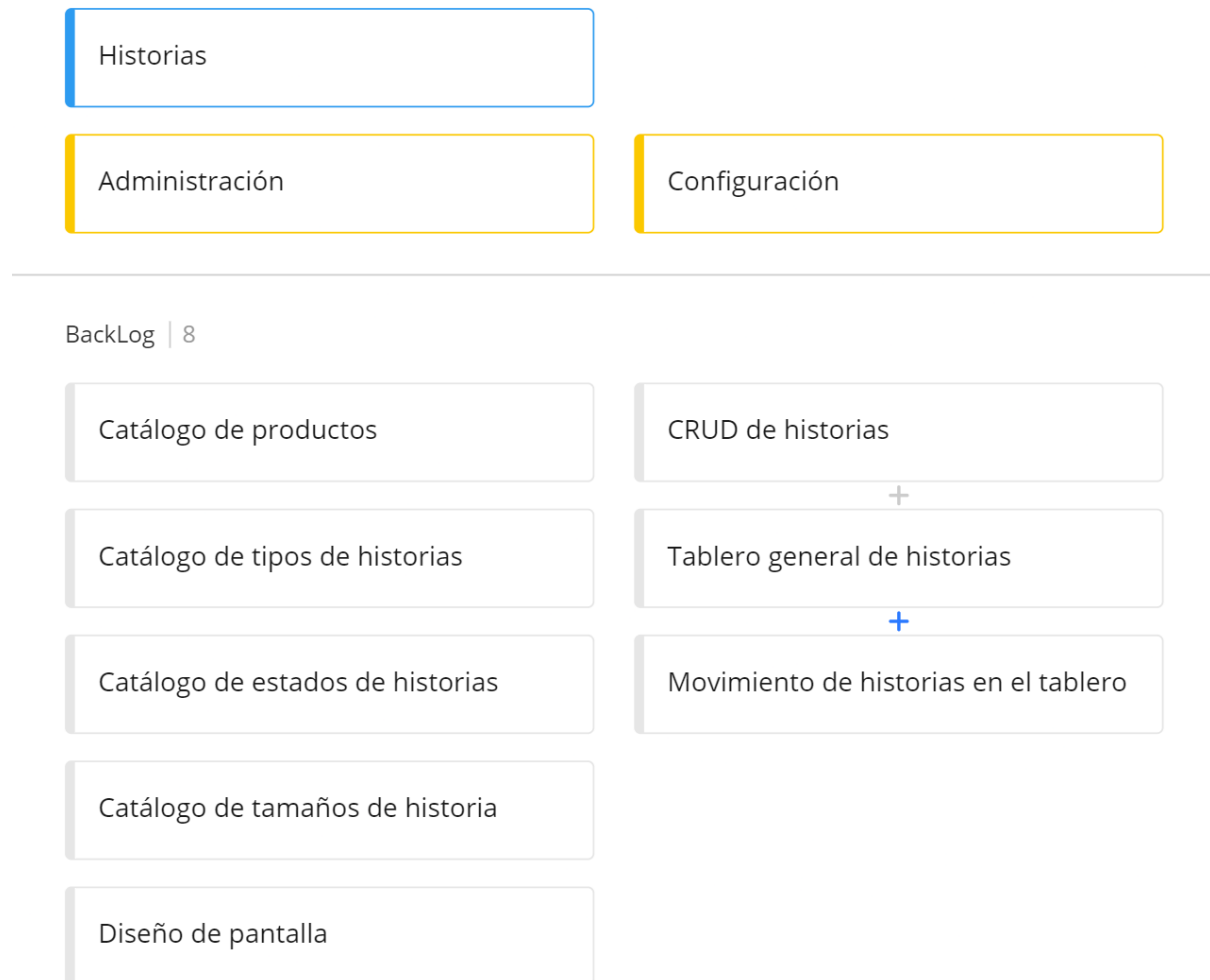
Figura G2*User Story Map del producto SI Board Parte 2*

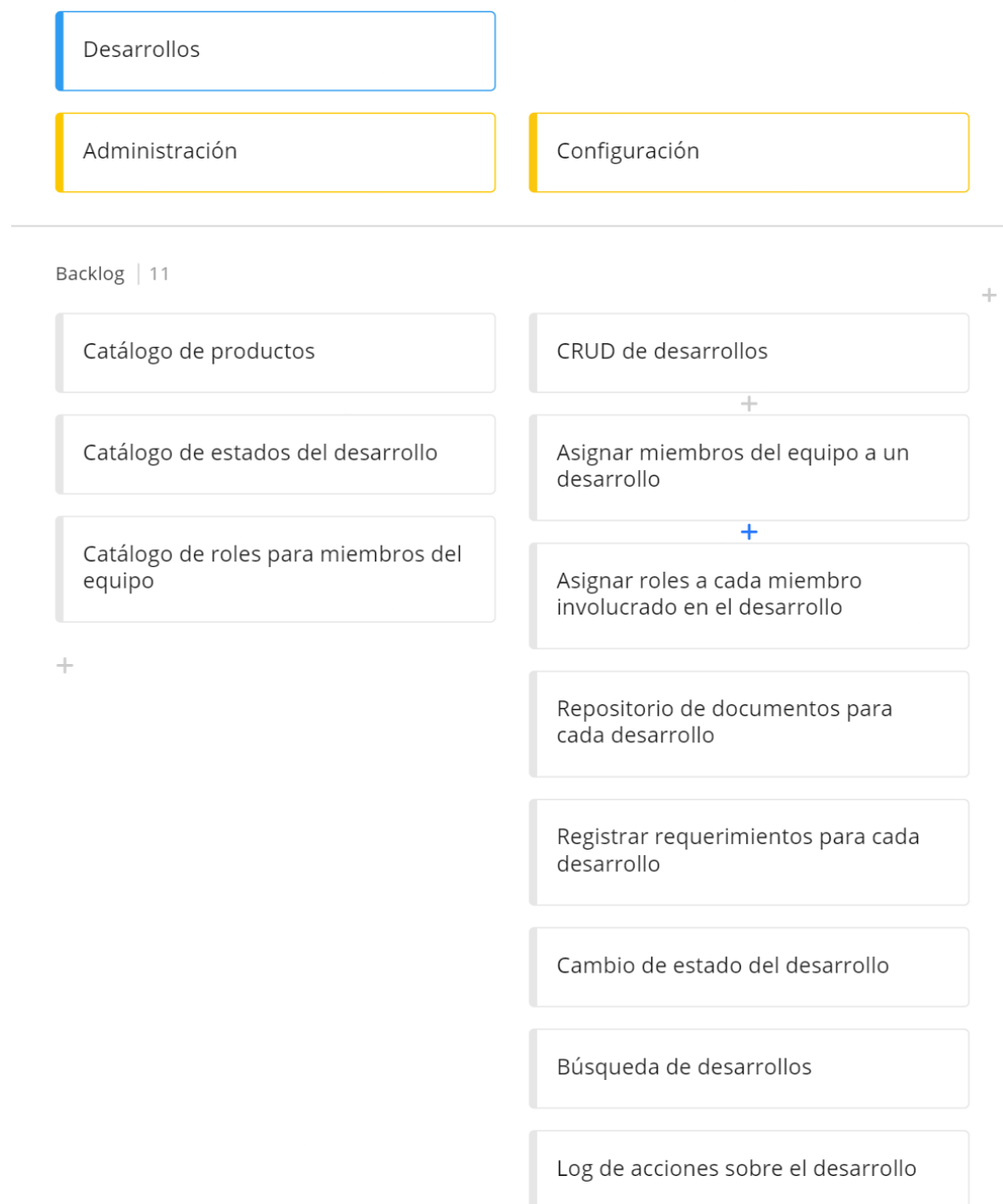
Figura G3*User Story Map del producto SI Board Parte 3*

Figura G4

User Story Map del producto SI Board Parte 4

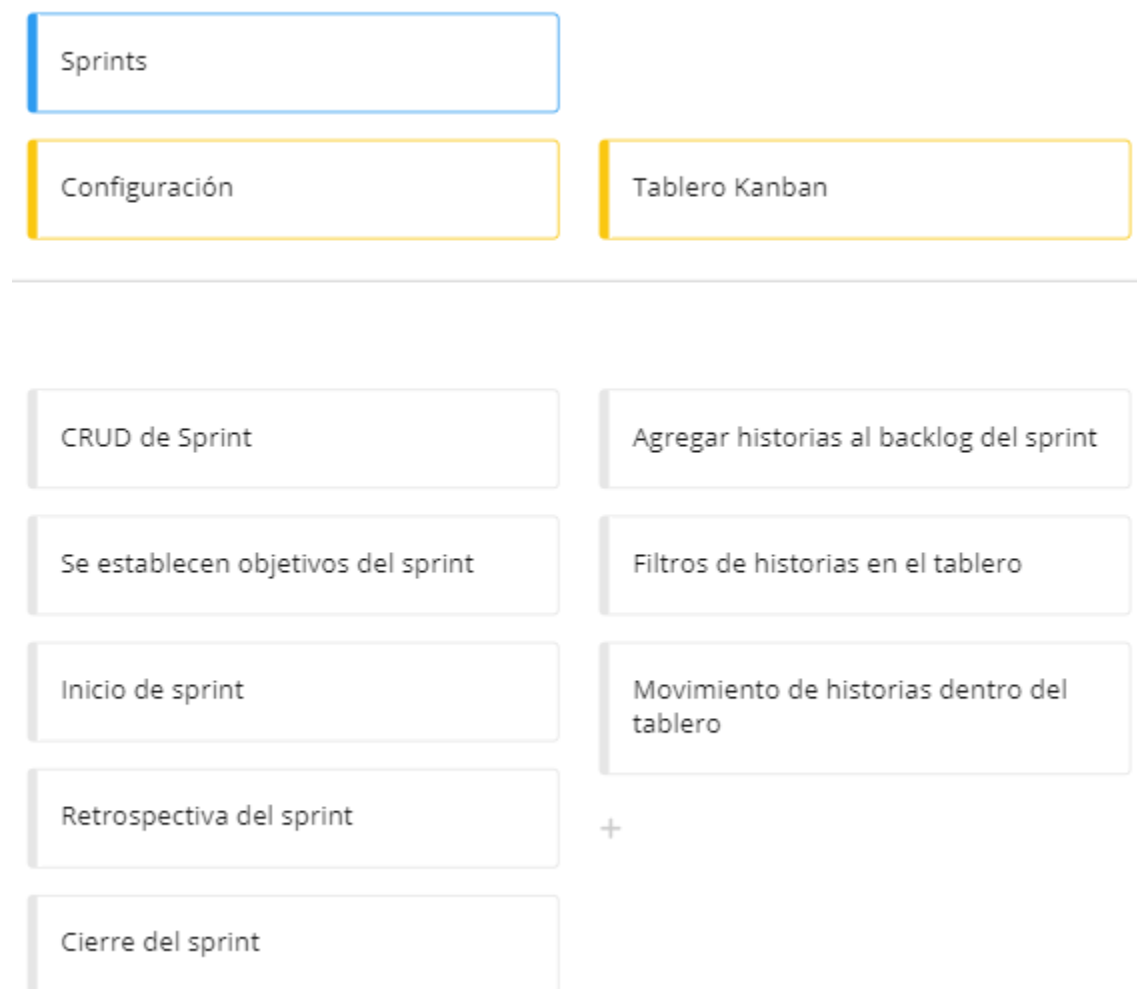
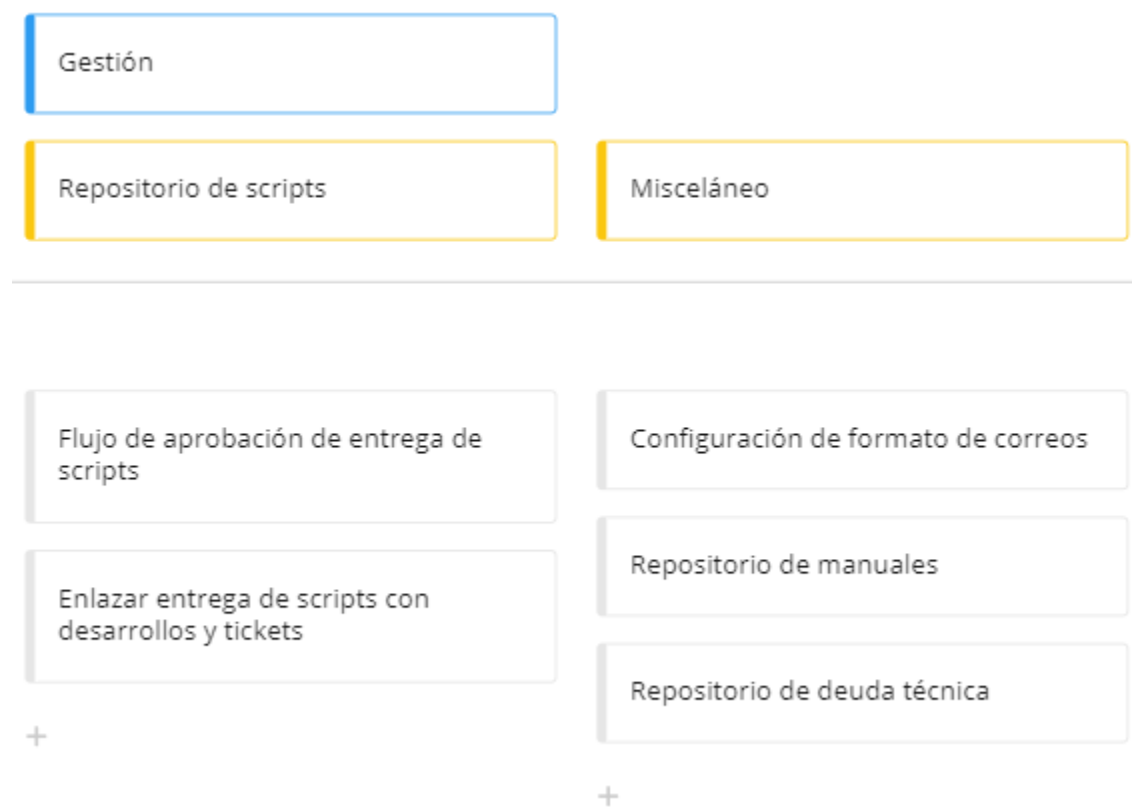


Figura G5

User Story Map del producto SI Board Parte 5



A Continuación, en la figura G6 se presenta la dimensión total del User Story Map.

Figura G6

User Story Map del producto SI Board Dimensión general

