

# Integración de enfoques ágiles para el mejoramiento continuo de procesos de software

Un reporte de experiencia en el Sector Público

## *Integration of Agile Approaches for Continuous Software Process Improvement*

*An Experience Report in the Public Sector*

Nombres de Autores de Primera Institución

Línea 1 (Institución) Dep., Universidad, Organización

Línea 2 (Institución)

Línea 3: Ciudad, País

Línea 4: Correo electrónico

Nombres de Autores de Segunda Institución

Línea 1 (Institución) Dep., Universidad, Organización

Línea 2 (Institución)

Línea 3: Ciudad, País

Línea 4: Correo electrónico

**Resumen** — La Ingeniería de Software Continua (CSE) implica la adopción de herramientas de automatización para facilitar la mejora continua en el desarrollo de software. Esta adopción requiere la implementación de procesos de software que permitan el análisis, la evaluación y la mejora continua, adaptables a las necesidades de negocio, desarrollo y operación (BizDevOps). Las organizaciones buscan integrar la Ingeniería de Software Continua con el objetivo de incrementar la calidad de los procesos y productos de software. Es crucial combinar estrategias para la automatización y medición continua de las fases del ciclo de vida del desarrollo, especialmente para aquellas organizaciones que adoptan metodologías ágiles e híbridas, buscando flexibilidad, colaboración y entregas incrementales. Este artículo presenta un reporte de experiencia sobre la integración de distintos enfoques ágiles, incluyendo la adopción del marco de trabajo Scrum y la metodología de planificación Objetivos y Resultados Clave (OKR), para la mejora continua de procesos centrada en prácticas de gestión de la calidad e indicadores clave de rendimiento (KPI). Nuestro objetivo es proporcionar evidencia y discutir las lecciones aprendidas en la adopción de enfoques ágiles y herramientas de medición, que han conducido a iniciativas de mejora continua de procesos de software en los últimos tres años de una Dirección Tecnológica de una institución educativa pública, regulada por normativas nacionales basadas en marcos tradicionales como COBIT e ITIL.

**Palabras Clave** - Ingeniería de Software Continua; Mejora de Procesos de Software; Desarrollo Ágil de Software; Indicadores Clave de Rendimiento; Tablero de Indicadores.

**Abstract** — Continuous Software Engineering (CSE) involves adopting automation tools to facilitate continuous improvement in software development. This adoption requires implementing software processes that enable continuous analysis, evaluation, and improvement, adaptable to business, development, and operational needs (BizDevOps). Organizations aim to integrate Continuous Software Engineering to enhance the quality of software processes and products. Combining strategies for

continuous automation and measurement of the development lifecycle phases is crucial, especially for organizations adopting agile and hybrid methodologies that seek flexibility, collaboration, and incremental deliveries. This paper presents an experience report on integrating various agile approaches, including the adoption of the Scrum framework and the Objectives and Key Results (OKR) planning methodology, for continuous process improvement focused on quality management practices and key performance indicators (KPIs). Our objective is to provide evidence and discuss the lessons learned in adopting agile approaches and measurement tools, which have led to continuous software process improvement initiatives over the past three years in a technology unit of a public educational institution, regulated by national standards based on frameworks such as COBIT and ITIL.

**Keywords** - Continuous Software Engineering; Software Process Improvement; Agile Software Development; Key Performance Indicators, Dashboard Indicators.

### I. INTRODUCCIÓN

La Ingeniería de Software Continua (CSE, por sus siglas en inglés) ofrece una visión integral del ciclo de vida del desarrollo de software, integrando y manteniendo un flujo de trabajo continuo entre la estrategia de negocio, el desarrollo de software y las operaciones, conocido como BizDevOps [1]. Este enfoque permite realizar entregas rápidas y frecuentes de productos de software, desarrollados de manera iterativa e incremental y en ciclos cortos, por ejemplo, mediante marcos de trabajo como Scrum o Kanban, lo cual es crucial para adaptarse a las cambiantes necesidades del negocio [2][3].

El modelo "Stairway to Heaven" describe cómo, a partir de una base de desarrollo tradicional, una organización puede evolucionar hacia una estructura ágil, adoptando procesos de integración y despliegue continuo apoyados por herramientas de

automatización, hasta alcanzar un estado de innovación continua siguiendo las prácticas de BizDevOps [2].

Desde el año 2021, la Dirección Tecnológica de la Universidad Estatal a Distancia, institución pública de educación a distancia, enfrentó el reto de integrar enfoques de desarrollo ágiles para lograr la transformación desde entornos tradicionales y rígidos a procesos de desarrollo adaptables y colaborativos, incluyendo la medición continua de la calidad. Esta transición correspondió al plan estratégico de la institución que responde a iniciativas a nivel país como la Estrategia de Transformación Digital hacia la Costa Rica del Bicentenario 4.0, en el que se reconoce el auge del uso de las tecnologías de la información y las comunicaciones (TIC). El Gobierno plantea la hoja de ruta para realizar las transformaciones digitales en las instituciones del sector público y en la sociedad a fin de potenciar el desarrollo socio-económico del país y asegurar una mejor calidad de vida para todos los habitantes de manera inclusiva [4]. En particular, la Directriz No.019-MP-MICITT emitida en agosto del 2018, establece que se deben tomar las medidas administrativas, técnicas y financieras necesarias para la consecución de los objetivos del Gobierno Digital del Bicentenario [5]. Lo anterior incluye la modernización de los procesos de desarrollo de software y la atención continua a la calidad de los servicios que ofrecen las instituciones del sector público.

En el año 2019, la institución trazó en su Plan de trabajo [6] el proyecto de transformación digital de la universidad. Durante el 2021, la Estrategia de Transformación Digital de la UNED fue declarada de interés institucional por parte del Consejo Universitario, la cual tiene como objetivo “Gestionar la transformación cultural, académica, administrativa y tecnológica de la UNED mediante un modelo flexible, utilizando la prospectiva y con el estudiante como eje central para el alcance de la misión de la universidad” [7]. Este proyecto promueve el tema de metodologías ágiles como Scrum, Kanban y Lean, centrada en la oferta de productos y servicios que generen valor para la institución [8]. En la hoja de ruta se incluyen actividades para planes piloto en la institución con el fin de establecer nuevos departamentos ágiles y brindar capacitaciones a todo el personal de la institución en estas metodologías, razón por la que se proyecta para el año 2025, un avance importante en la adopción de procesos ágiles en la institución [9]. Adicionalmente, el Plan Estratégico de TI 2020-2024 establece el desarrollo de un modelo de gestión de TI que incorpore conceptos de calidad de software para la mejora de los servicios que se le brindan en la universidad: la definición de una metodología para desarrollo ágil que incorpore herramientas que permitan dar acompañamiento en el proceso de análisis, definición de requerimientos de software, gestión de calidad y pruebas de software, desde la perspectiva de calidad [10]. Finalmente, uno de los objetivos de calidad de la institución, establece la necesidad de impulsar la mejora continua de los procesos de los sistemas de información que incluye las plataformas educativas, a través de la investigación, desarrollo e innovación [11], objetivo que refuerza las iniciativas de mejora de procesos institucionales.

Durante los últimos tres años, la Dirección Tecnológica integró distintos enfoques ágiles, incluyendo la adopción del marco de trabajo Scrum y la metodología de planificación

Objetivos y Resultados Clave (OKR, por sus siglas en inglés), para la mejora continua de procesos centrada en prácticas de gestión de la calidad e indicadores clave de rendimiento (KPI) que permitan la medición de los resultados. Para esto, integró herramientas de gestión y medición en sus procesos y productos, con el objetivo de implementar una mejora de procesos que incluya la medición continua automatizada de la gestión de calidad durante el desarrollo ágil.

Nuestro reporte presenta las acciones realizadas para implementar el proceso de mejora continua, incluyendo la integración del marco de trabajo Scrum, el uso de metodologías de planificación OKR y los Indicadores Clave de Rendimiento (KPI), en cumplimiento de normas técnicas [12] y marcos de gobierno y gestión de la institución [13], que se basan en modelos como ITIL y COBIT. Tomando como referencias iniciales el modelo "Stairway to Heaven" y BizDevOps, se lograron resultados significativos en la transformación de los procedimientos internos de trabajo, el uso de plataformas de gestión de proyectos, la gestión documental, la capacitación continua de los equipos de trabajo, y el diseño e implementación de un modelo de medición continua ágil apoyado por tableros de indicadores clave de rendimiento (KPI). Lo anterior ha guiado la transformación hacia el ágilísimo, y procesos de desarrollo de software que combinan las ventajas del desarrollo ágil y tradicional, esto es, un enfoque de desarrollo híbrido.

En el 2018, la institución había implementado una mejora de procesos de desarrollo centrada en la gestión de la calidad combinando el modelo IDEAL y los enfoques GQM, las normas técnicas (NTCGR), el marco de trabajo COBIT, prácticas de aseguramiento de la calidad (SQA), el modelado Business Process Model and Notation (BPMN) y el modelo de Aceptación Tecnológica (TAM) para los procesos de desarrollo tradicionales que realizaba [14]. Actualmente, en su proceso de transformación tecnológica, que busca promover y cumplir con los estándares de calidad junto con la adopción de enfoques ágiles y la medición continua para la gestión de sus proyectos, se ha trabajado en este nuevo modelo de mejora continua de procesos centrada en prácticas ágiles, automatización y gestión de la calidad, lo que representa una nueva mejora a los procesos establecidos anteriormente. La Dirección Tecnológica gestiona el desarrollo y mantenimiento de sistemas de información y sitios web en la institución para una población estudiantil de más de 40,000 personas. Al 2024, la Dirección Tecnológica gestiona 92 aplicaciones de software en producción y un total de 118 aplicaciones en su portafolio de proyectos, tanto administrativos como académicos, además del sitio web institucional y la plataforma educativa que atiende a las personas estudiantes y aproximadamente 3,500 personas docentes y administrativas. Por lo que la mejora continua de procesos tecnológicos es una prioridad para la institución.

La sección II del artículo presenta los conceptos teóricos, la sección III presenta los trabajos relacionados, la sección IV presenta la metodología y la integración de enfoques ágiles para el mejoramiento continuo de procesos de software, la sección V analiza los resultados de la integración de enfoques ágiles y la sección VI presenta las conclusiones y trabajo futuro.

## II. MARCO TEÓRICO

En esta sección se describen los conceptos teóricos principales.

### A. Normas técnicas para el sector público costarricense

El conjunto de las normas técnicas para la gestión y el control de las Tecnologías de Información (TI), es un marco de gestión de TI de acatamiento obligatorio para las instituciones adscritas al sector público. Contiene 14 procesos, de los cuales, tres de los procesos de mayor relevancia para este estudio son: (1) Calidad de los procesos tecnológicos, (2) Desarrollo, implementación y mantenimiento de sistemas de información y (3) Aseguramiento [12]. En Costa Rica, las normas técnicas han sido mapeadas hacia modelos como COBIT e ITIL [15] [16].

Como complemento al marco de gestión de TI anterior, la institución cuenta con un Marco de gobierno y gestión de TI orientado a la institución, que está constituido por seis objetivos de gobierno que se dividen en objetivos de gestión, que, a su vez, se dividen en prácticas con sus respectivas actividades y productos esperados. Dentro de los cuales, para esta investigación se consideró el objetivo de mejora continua, y el objetivo de gestión de calidad de los servicios de TI [13]. Ambos marcos se complementan y son de acatamiento obligatorio para la Dirección Tecnológica.

Las normas técnicas para la gestión y el control de las Tecnologías de Información (TI) [12], son utilizadas por las organizaciones como una guía para la implementación de buenas prácticas basadas en COBIT 2019 [17][18]. El Marco de gobierno y gestión de TI [13] también hace referencia a: (1) COBIT 2019, que es un marco de referencia para el gobierno y la gestión de las tecnologías de la información (TI) de una organización. Este marco está diseñado para que las organizaciones puedan alinear sus procesos de TI con los objetivos estratégicos del negocio, mejorar la gobernanza de TI, administrar los riesgos de TI de manera efectiva y garantizar el cumplimiento de las normas y regulaciones [17]. (2) ITIL (Information Technology Infrastructure Library), que es un marco de trabajo basado en buenas prácticas para la gestión de servicios de TI (ITSM). Proporciona una guía para planificar, diseñar, entregar, operar y mejorar los servicios de TI. Este marco es utilizado por organizaciones para mejorar la calidad de sus servicios de TI, reducir costos y aumentar la satisfacción del cliente [19].

### B. Ingeniería de Software Continua (CSE)

CSE es un enfoque integral que abarca el ciclo de vida del desarrollo de software, procurando la integración y la entrega continuas de software. Se centra en la automatización y la mejora continua de los procesos que permiten a las organizaciones responder a los cambios del mercado y a las necesidades de los clientes. Incluye prácticas y principios que buscan optimizar y agilizar todos los aspectos del desarrollo de software, fomentando una cultura de colaboración y mejora continua [1].

BizDevOps se refiere a un enfoque holístico para la gestión de la innovación y el crecimiento organizacional que integra las prácticas de negocio, desarrollo de software y operaciones en el ciclo de vida de desarrollo de software [1]. La fig. 1 presenta

actividades continuas (BizDevOps) con una base de mejora continua, experimentación e innovación.

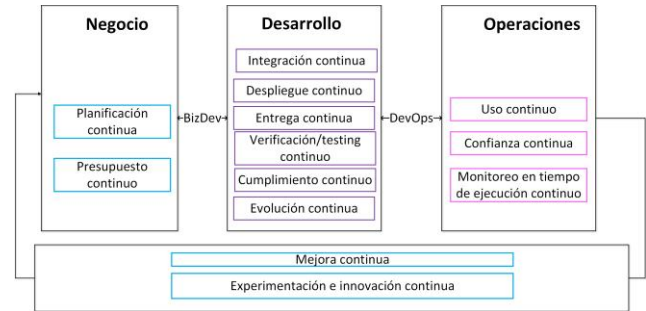


Figure 1. Continuo: una visión holística de las actividades desde el negocio, el desarrollo, las operaciones y la innovación [1]

BizDev incluye las prácticas de planificación continua y presupuesto continuo. Por otro lado, las principales actividades de desarrollo de software que son análisis, diseño, codificación y verificación / prueba, en las que DevOps promueve la integración continua, el despliegue continuo, la entrega continua y la verificación / prueba continua. Además, se incluye el enfoque al cumplimiento continuo y las actividades de seguridad continuas [1].

La mejora continua busca la mejora de los procesos y productos de software [1]. Finalmente, la innovación continua busca establecer un proceso sostenible que responda a las condiciones cambiantes del negocio y que se base en métricas apropiadas a lo largo de todo el ciclo de vida del software. La innovación y la experimentación se deben considerar en las propuestas y validaciones de nuevas formas que incluyan automatización de procesos, evaluaciones y análisis de datos que apoyen los procesos en la actualidad cambiantes [1].

### C. Mejora de procesos SPI

Una mejora de procesos de software (Software Process Improvement, SPI, por sus siglas en inglés) es la secuencia de tareas y técnicas que se realizan para la planificación y aplicación de las actividades que permiten definir de manera efectiva los procesos de una organización que desarrolla software [20].

Es un proceso esencial para las organizaciones que desean desarrollar software de alta calidad de manera eficiente y eficaz. Al implementar un enfoque sistemático y continuo para la mejora de procesos, las organizaciones pueden lograr una serie de beneficios, como una mayor eficiencia, mejor calidad, menores costos, un tiempo de entrega del producto más rápido y una mayor satisfacción del cliente [20].

### D. Gestión de la calidad del software

El aseguramiento de la calidad (SQA, por sus siglas en inglés) es el conjunto de actividades sistemáticas y planificadas para brindar la confianza de que, el proceso de desarrollo de software o el proceso de mantenimiento de un producto software cumple con los requisitos técnicos funcionales establecidos, los requisitos administrativos para mantener el cronograma y operar dentro del presupuesto establecido [20].

En el marco de trabajo Cobit 2019 el objetivo de gestión APO11 — Gestionar la calidad incluye prácticas como: establecer un sistema de gestión de la calidad, enfocar la gestión de la calidad al cliente, gestionar los estándares, prácticas y procedimientos de calidad e integrar la gestión de la calidad en los procesos y soluciones clave y llevar a cabo la monitorización, control y revisiones de calidad [17].

Este objetivo de gestión APO 11, se relaciona con las normas técnicas para la gestión y el control de las Tecnologías de Información (TI), con respecto a la Calidad de los Procesos Tecnológicos [18], donde se establece que las instituciones deben implementar prácticas que permitan controlar los procesos organizacionales, posibilitando la mejora continua de productos y servicios, buscando asegurar la satisfacción de las necesidades institucionales, manteniendo estándares de documentación de los lineamientos requeridos, esquemas para la medición del desempeño y control sobre la vigencia de las prácticas aplicables a los procesos. Igualmente, debe generar servicios de TI de conformidad con los requerimientos de los usuarios con base en un enfoque de eficiencia y mejoramiento continuo de los procesos que habilitan la gestión de las tecnologías de información [12]. Por lo tanto, el proceso de desarrollo de software debe enfocarse en generar productos de conformidad con los requerimientos de las personas usuarias.

#### E. Modelo Stairway to Heaven

La tendencia de escalar del desarrollo de software basado en enfoques tradicionales como Cascada hacia enfoques ágiles es cada vez mayor. El modelo *Stairway to Heaven* consta de cinco etapas como se presenta en la fig. 2. Partiendo del desarrollo tradicional, el segundo escalón es la integración de prácticas de trabajo ágiles en la organización. Se incorporan prácticas de trabajo con equipos pequeños y empoderados, una pila de producto de trabajo pendiente, una serie de reuniones y cortos periodos para realizar entregas de productos de manera incremental mediante iteraciones [2].

Una vez que se han adoptado prácticas de trabajo ágiles, normalmente los equipos comenzarán a expresar su frustración por su incapacidad para probar el código que construyen en el contexto más amplio del sistema. Esto conduce al tercer escalón que consiste en la integración continua y al cuarto escalón de despliegue continuo con prácticas para el desarrollo impulsado por pruebas por equipos ágiles, un entorno de prueba y compilación automatizado, y la definición de requisitos funcionales, heredados y de calidad del sistema o familia de sistemas. Para llegar finalmente al quinto escalón, que corresponde a una organización que centra sus procesos automatizados y ágiles con aspectos relacionados con la investigación, innovación y mejora continua [2].



Figure 2. *Modelo Stairway to Heaven* [2]

#### F. Desarrollo ágil e híbrido de software

El término Ágil un enfoque de gestión de proyectos basado en la entrega de requisitos de forma iterativa e incremental a lo largo del ciclo de vida. Desarrollo ágil es un término general específico para metodologías iterativas de desarrollo de software como Scrum, Lean, eXtreme Programming [21].

El marco de trabajo Scrum se define como un marco ligero que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptables para problemas complejos. Scrum se basa en tres pilares empíricos de transparencia, inspección y adaptación. La inspección se centra en tres puntos durante cada Sprint que son: (1) el progreso durante la reunión diaria, (2) el producto durante la revisión del Sprint y (3) el proceso durante la retrospectiva del Sprint. Además, Scrum, se enfoca en cinco valores que son el compromiso, enfoque, apertura, respeto y coraje. La unidad fundamental del marco de trabajo es el equipo Scrum, que está conformado por diversos roles incluyendo las personas desarrolladoras, el dueño o propietario del producto y el Scrum master [3]. Seguidamente se resumen sus principales características.

En Scrum, el Sprint es el principal evento de longitud fija de máximo un mes de duración, en los que se llevan a cabo otros eventos como: el refinamiento de la pila de producto, la planificación del Sprint, la reunión diaria, la revisión del Sprint y la retrospectiva del Sprint. Los artefactos que se trabajan en Scrum son la pila de producto, la pila del Sprint y el incremento del producto.

El desarrollo híbrido de software es una combinación de enfoques ágiles o tradicionales que una unidad organizacional adopta y personaliza según las necesidades de su propio contexto (por ejemplo, dominio de aplicación, cultura, procesos, proyecto, estructura organizacional, técnicas y tecnologías) [22] [23].

Existen contextos de organizaciones que requieren una mejor adopción al cambio, por lo que, optan por enfoques ágiles como Scrum. Sin embargo, pueden contar con regulaciones que provienen de enfoques más tradicionales como Cascada, que permite etapas secuenciales más controladas. Esto desencadena en la opción de combinar prácticas de ambos enfoques que respondan a las necesidades de la organización, tomando el rumbo del desarrollo híbrido de software [23].

#### G. Objetivos y Resultados Clave (OKR)

La metodología Objetivos y Resultados Clave (Object Key Result, OKR, por sus siglas en inglés) es un marco de gestión de objetivos y resultados clave que ayuda a las organizaciones a definir metas claras y medir su progreso. John Doerr, popularizó el concepto con su libro *Measure What Matters* [24]. OKR presenta dos elementos principales:

- **Objetivos:** Establecen qué se quiere lograr. Deben ser cualitativos y claros. Se definen como una meta simple de una oración que comienza con un verbo. A menudo toma la forma de: “verbo + qué quieres hacer + para /qué” [24].
- **Resultados Clave:** Establecen cómo se medirá el logro del objetivo. Deben ser cuantitativos, específicos y medibles. Cada objetivo debe tener entre dos a cuatro resultados clave

que indiquen el progreso hacia el logro del objetivo. Los resultados clave son valores cuantitativos que miden el logro del objetivo. Puede tomar la forma: “verbo + lo que vas a medir + de x aspecto a y aspecto” [24].

Se define un periodo para la revisión de cada objetivo y los resultados clave definidos. Si alguno presenta riesgos, se debe hacer un plan para cambiar el rumbo. Se debe realizar una revisión de cada resultado clave según su cumplimiento y definir el por qué sucedió. Con base en el éxito o el fracaso de los resultados clave, se toma la decisión de si se ha cumplido o no con el objetivo. Si no es así, se mantiene en el siguiente periodo, con los mismos resultados clave o nuevos dependiendo de si se completan los resultados clave en revisión. Si se determina que se ha cumplido el objetivo, se crea un nuevo objetivo para el próximo periodo [24].

#### H. Indicadores Clave de Rendimiento (KPI)

El marco de gobierno y de gestión de TI define un Indicador Clave de Rendimiento (key Performance Indicator, KPI, por sus siglas en inglés), como una métrica importante que se usa para evaluar el éxito en la consecución de un objetivo [13].

KPI es una herramienta que permite la medición y mejora continua de los procesos de TI, mediante la evaluación del rendimiento, la identificación de las áreas de mejora y asegurar que los servicios y procesos de TI estén alineados con los objetivos estratégicos del negocio [17][19][13].

Los KPI son mediciones de negocio que se utilizan para visualizar el estado y las tendencias de la organización [25]. El tablero KPI es una herramienta que permite vista consolidada y resumida de los principales indicadores clave de rendimiento de una organización, proyecto o proceso. Estos tableros permiten a las personas usuarias monitorear y analizar el estado y el progreso hacia los objetivos estratégicos y operativos [25].

### III. TRABAJOS RELACIONADOS

El incremento de la adopción de prácticas ágiles en las organizaciones que trabajaban con las metodologías tradicionales de desarrollo de software como el modelo en cascada, iniciaron un camino hacia el cambio a partir del año 2000 cuando se adoptan las prácticas de la Web 2.0 y el software como servicio (SaaS, por sus siglas en inglés). La integración de marcos de trabajo ágiles han permitido realizar entregas de software al negocio en periodos más cortos, aunque estas actividades traen desafíos en cuanto a la calidad de procesos, productos y servicios, así como la participación activa del negocio en el proceso de desarrollo de software [2].

Sin embargo, la implementación de marcos de trabajo ágiles ha generado nuevas necesidades relacionadas con la continuidad de las actividades en el ciclo de vida del desarrollo de software. Esto requiere una fluidez y comunicación constante entre las áreas de negocio, desarrollo y operaciones (BizDevOps). Muchas soluciones se apoyan en herramientas de automatización y generación de grandes cantidades de datos [2].

La literatura menciona los desafíos que enfrentan las organizaciones en la adopción de la Ingeniería de Software Continua. En este contexto, se refiere al modelo *Stairway to Heaven*, que inicia con el desarrollo tradicional, seguido de la integración de prácticas de trabajo ágiles incluyendo

investigación y desarrollo (I + D). Tras la adopción de prácticas ágiles, las organizaciones avanzan hacia procesos más automatizados [2]. La participación del negocio es crucial para la puesta en producción de software o las nuevas versiones, y en esta etapa se requiere control de calidad riguroso. Las organizaciones se dan cuenta de que la implementación frecuente de software conlleva actividades continuas, como pruebas de software, y que el negocio puede determinar qué características se utilizan y cuáles son necesarias [2].

Diversos estudios han investigado la implementación las actividades de CSE, enfocándose en cómo los profesionales capturan y utilizan la retroalimentación de las personas usuarias para gestionar los requerimientos [26], desarrollando marcos de implementación genéricos para DevOps [27], y sistemas de monitoreo continuo de excepciones que sirven como base de conocimiento para sugerir prácticas de calidad que eviten errores en el código. Estos enfoques permiten monitorear continuamente el software en operación, recopilar problemas automáticamente y recomendar soluciones a los desarrolladores [28]. La automatización de procesos y la detección de los errores durante la integración continua pueden afectar directamente el aseguramiento de la calidad [29].

En cuanto al uso de KPI para la medición del ciclo de desarrollo de software, se han analizado las características deseables de lo que se considera un “buen KPI” y su representación mediante tableros de visualización [25]. Además, se han desarrollado modelos para evaluar el desempeño del proceso Scrum en organizaciones a través de indicadores clave de rendimiento KPI, generando recomendaciones para gestionar tareas y mejorar los procesos adoptados. Estos estudios proponen clasificaciones de KPI relacionados directamente con las actividades del marco trabajo Scrum [30].

Finalmente, la aplicación de la metodología OKR y Scrum en proyectos de desarrollo ágil de software ha sido documentada, proporcionando recomendaciones para la implementación de OKR medibles que ofrezcan visibilidad de la colaboración y seguimiento del progreso de las actividades con prioridades claras. Esto ha mejorado el rendimiento en el tiempo de entrega de software en equipos ágiles que trabajan con Scrum [31].

### IV. INTEGRACIÓN DE ENFOQUES ÁGILES PARA LA MEJORA DE PROCESOS DE SOFTWARE

En esta sección se describen las etapas de la metodología y sus principales actividades.

#### A. Metodología

La fig. 3 presenta la metodología para la integración de enfoques ágiles para el mejoramiento continuo de procesos de software. Basada en el modelo *Stairway to Heaven* [2] la metodología inicia con un análisis del modelo tradicional para avanzar a un proceso ágil de desarrollo de software. Posteriormente, se implementa de automatización y la innovación. Además, se combinan otros enfoques como la metodología OKR, la definición de los indicadores claves de rendimiento (KPI), y las prácticas de CSE, manteniendo un mapeo hacia las normas técnicas de cumplimiento obligatorio (Cobit, ITIL, otras). A continuación, se detallan las etapas de la metodología aplicada.



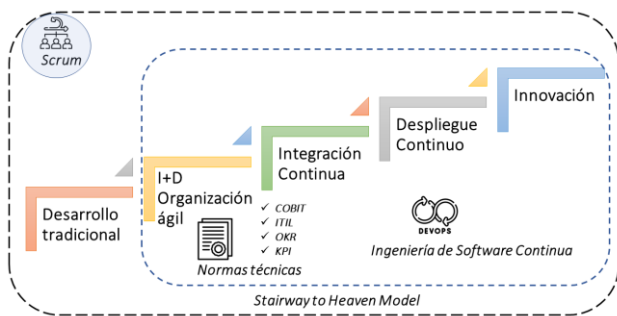


Figure 3. Metodología de integración de enfoques ágiles para el mejoramiento continuo de procesos de software

### 1) Etapa de preparación

Esta etapa abarca las acciones necesarias para iniciar con el proyecto de integración de enfoques ágiles para el mejoramiento continuo del proceso de desarrollo de software. Las principales acciones son las siguientes:

- Conformación del equipo de trabajo: el equipo está conformado por cuatro personas participantes: una persona asesora experta en agilidad empresarial, una persona coordinadora de unidad, dos personas coordinadoras de proyectos informáticos y una persona gestora de calidad.
- Elaboración de plan de trabajo: Se sigue el marco de trabajo Scrum, con sesiones de trabajo de cuatro horas semanales. Se creó una pila de trabajo en el año 2021 que se actualiza y prioriza continuamente al año 2024.
- Capacitación del equipo: se incluyen temas de agilidad empresarial y marcos de trabajo ágiles.
- Presupuesto para el proyecto: Se incluyen rubros en el presupuesto de la Dirección Tecnológica para las actividades de ejecución del proyecto.

### 2) Etapa 1: Análisis del desarrollo tradicional de software

La Dirección Tecnológica presentaba un enfoque tradicional de desarrollo de software caracterizado por ciclos de desarrollo amplios, fases secuenciales, con una planificación rigurosa donde los requerimientos se establecen desde el inicio del proyecto. La interacción entre la gestión del producto software, desarrollo, pruebas y negocio se realiza en Cascada con ciclos de retroalimentación largos y dificultades para integrar los comentarios de los clientes durante el desarrollo del producto. La comunicación con el cliente tiene lugar al final del ciclo de vida del proyecto [2]. En este contexto, se realizan las siguientes actividades:

- Análisis del contexto de desarrollo tradicional: Se selecciona la mejora del proceso de desarrollo y mantenimiento de software como parte de las acciones de la Dirección Tecnológica, en respuesta a las propuestas de transformación digital y planes estratégicos.
- Diagnóstico del estado actual del proceso de desarrollo de software: Se integra la documentación vigente del proceso tradicional y se analiza con el equipo de trabajo. El diagnóstico se realiza paso a paso, priorizando los elementos de trabajo de la pila de producto. Se establecen

las fuentes de documentación y herramientas utilizadas, identificando información requerida por cumplimiento normativo y que genere valor para la Dirección Tecnológica, llevando un control con el nombre del documento, los comentarios de mejora / actualización/eliminación según corresponda, fecha que se analizó. Además, incluye el mapeo de la documentación tradicional y la sustitución a documentación ágil, con su respectiva herramienta propuesta para almacenar los datos requeridos. En el caso de las herramientas tecnológicas, se analizaron las herramientas y licenciamiento con las que contaba la Dirección Tecnológica, en especial, la de gestión y desarrollo de proyectos de software. En cuanto a herramientas, el objetivo era identificar las actualizaciones / migraciones / adquisiciones y respaldos requeridos

- Resultados del diagnóstico se espalda la información de los proyectos almacenados en la herramienta Team Foundation Serve, se adquiere y configura la herramienta Azure DevOps Server, y se verifica el licenciamiento de otras herramientas como Project Online, Viva Engage, SharePoint. Con respecto a la documentación de los procedimientos: se mapeó la documentación existente tradicional a documentación para desarrollo ágil entre los que destacan: pasar de una plantilla para especificación de requerimientos a documentar requerimientos en historias de usuario en Azure DevOps, en el caso de los documentos para gestionar las pruebas pasó de plantillas al uso del Test Plans en Azure DevOps. La documentación técnica de cada proyecto se almacena en un repositorio seguro. El diagnóstico permitió identificar las herramientas requeridas, así como, la documentación requerida para la mejora del proceso de desarrollo de software.

### 3) Etapa 2: Organización ágil – Establecer el desarrollo ágil de software

Esta etapa se centra en la integración de enfoques ágiles mediante la formación de equipos de desarrollo multifuncionales, la definición de Sprints cortos de desarrollo con entregas de software funcionales y una planificación continua en la que el cliente participa para la retroalimentación constante [2]. En el contexto de la Dirección Tecnológica se realizan las siguientes actividades:

- Estudio y selección de enfoques ágiles: El equipo de trabajo de agilismo estudió enfoques ágiles como SAFe, Scrum y Kanban, decidiendo implementar Scrum para los equipos de desarrollo de la Dirección Tecnológica. Una vez instalada la herramienta de Azure DevOps Server, se configuró la plantilla ágil para trabajar con los elementos de trabajo de la épica, historias de usuario y tareas. El elemento funcionalidad o característica se incluirá en un nivel de madurez posterior.
- Incorporación en el presupuesto: las actividades de la integración de enfoques ágiles se han incluido el presupuesto anual desde el año 2021.
- Capacitación continua: Se proporcionó capacitación continua en temas como agilismo empresarial, marco de

trabajo ágil Scrum, estructura de las historias de usuario y el modelo INVEST [32], elaboración y de casos de prueba, pruebas de aceptación, automatización de pruebas, mejores prácticas de desarrollo de software dentro del plan de capacitaciones y su respectiva ejecución continua.

- Formación de equipos pilotos Scrum: Se seleccionaron y formaron dos equipos Scrum pilotos con roles de Scrum Master, persona dueña del producto, y equipo de desarrollo para iniciar la implementación del marco de trabajo Scrum y el uso de las nuevas herramientas.
- Adopción de prácticas ágiles: Se definió el Sprint de un mes, planificación del Sprint, revisión del Sprint, Retrospectiva del Sprint, Refinamiento. Se elaboró una nueva versión de los procedimientos con enfoques ágiles adoptados y documentación adicional requerida como instructivos para el uso de nuevas herramientas y que respondan a las normas técnicas.
- Formalización de procedimientos y documentación técnica: especialistas de la institución formalizaron los procedimientos y documentación técnica como el uso de control de versiones de código fuente, instructivos del uso de plataformas y pruebas de software.
- Plan de comunicación: Se elaboró y ejecutó un plan de comunicación para la mejora de proceso, iniciando con una matriz de temas relevantes sobre agilidad, el marco de trabajo Scrum y los equipos pilotos. Se implementó una herramienta de comunicación abierta a la comunidad.
- Acompañamiento y concientización: se llevaron a cabo sesiones de divulgación y formalización tanto a lo interno de la Dirección Tecnológica como a la organización en general. Estas sesiones explicaron los procesos, procedimientos y avances de la incorporación de equipos ágiles.
- Definición de los flujos de trabajo: se definieron los flujos de trabajo para las épicas, historias de usuario e incidentes, este trabajo consistió en definir el paso a paso del trabajo en el proceso de desarrollo de software y contextualizado a la organización. La herramienta Azure DevOps Server se configuró para documentar el tiempo de cada etapa del flujo de trabajo y automatizar los indicadores clave de rendimiento.
- Definición de OKR y KPI: se establecieron OKR y KPI para el proceso de desarrollo utilizando buenas prácticas ITIL [19], Cobit [17] y normas técnicas[12] [13]. Se realizaron sesiones de trabajo para identificar objetivos y resultados clave alineados a los objetivos institucionales, priorizando los indicadores propuestos según su importancia y utilidad percibida del equipo de trabajo. Los KPI se documentaron y analizaron en estas sesiones de trabajo considerando (1) cumplimiento, (2) valor para la Dirección Tecnológica y la institución (3) alcance (3) viabilidad con respecto a los recursos disponibles como software, infraestructura tecnológica, capacidad del talento humano, entre otros aspectos. Se

documentaron los campos de objetivo estratégico, objetivo de la Dirección Tecnológica, los resultados claves considerando Cobit e ITIL, y KPI con sus respectivos datos de descripción del KPI para el flujo de desarrollo, métrica, el rol responsable de verificar que se recolecten los datos, la herramienta o fuente de datos como: Azure DevOps (historias de usuario, épicas, pruebas), formularios de SharePoint (reporte de investigaciones activas, encuestas de satisfacción).

- Encuestas de satisfacción: Se diseñaron encuestas de satisfacción para evaluar la atención de los equipos ágiles y la encuesta para la satisfacción de incidentes atendidos. Como parte del ciclo iterativo e incremental, cuando se detectan mejoras o ajustes se aplican, en este caso, para la documentación de incidentes se creó y configuró la plantilla del elemento de trabajo de Incidente en Azure DevOps.
- Capacitación continua técnica y de investigación continua: se incluyó capacitación para las personas desarrolladoras mediante el plan de capacitaciones para el fortalecimiento y actualización en áreas como: programación, análisis de datos, experiencia del usuario y de investigación científica como parte de los proyectos internos de la Dirección Tecnológica establecidos en el plan estratégico que fomentan las prácticas de gestión y de investigación.
- Implementación de CSE: adopción del marco de trabajo Scrum y su respectivo procedimiento formal, en el que, se incluye planificación en Sprint o iteraciones cortas y entrega incremental de valor, incorpora la participación del negocio como parte de los equipos y se establece las prácticas de retroalimentación continua. Este paso a permitido valorar el camino hacia la automatización de tareas repetitivas y el uso de herramientas que faciliten el proceso de desarrollo de software.

#### 4) Etapa 3: Establecer la integración Continua

En la tercera etapa, la Dirección Tecnológica se centra en la integración frecuente del trabajo, lo que incluye compilaciones diarias y confirmaciones rápidas de cambios, por medio de la automatización de las compilaciones y las pruebas. Para ello, ha iniciado actividades de capacitación para personas desarrolladoras enfocadas en la automatización de pruebas utilizando Jenkins y Selenium.

Como parte del procedimiento de desarrollo ágil de software se ha implementado el uso de control de versiones de código fuente a través la herramienta Repos de Azure DevOps Server utilizando Git. Actualmente, cada proyecto registrado en Azure DevOps Server cuenta con su repositorio. Se han realizado sesiones de trabajo para transmitir e incentivar el uso de esta herramienta, aspecto importante para resaltar es la incorporación de las personas desarrolladoras en la elaboración de guías e instructivos, así como las pruebas de estas herramientas.

La implementación de la automatización continuará conforme se consolide la etapa de la organización ágil para

todos los equipos de la unidad, siguiendo las actividades de CSE como la ejecución automática de pruebas de pruebas unitarias, de integración y el monitoreo de compilaciones y pruebas [1].

5) Etapa 4: Despliegue Continuo

En lugar de grandes compilaciones y lanzamientos planificados, esta etapa propone la publicación continua de cambios en el código lo que permite obtener retroalimentación continua del cliente, aprender del uso del cliente, eliminar el trabajo que no aporta valor al cliente. Investigación y Desarrollo I+D, gestión de producto y clientes participan en un ciclo de desarrollo ágil y rápido, con un tiempo de respuesta corto [2].

Una vez incorporados los conocimientos de automatización, se habilitará el uso de la herramienta de Pipelines del Azure DevOps Server. Adicionalmente, se incorporan personas colaboradoras del área de infraestructura tecnológica como parte de la capacitación que se requiera para integración de enfoques ágiles y CSE.

6) Etapa 5 Innovación

En la quinta etapa la Dirección Tecnológica responde a la retroalimentación instantánea del cliente. Se considera que el despliegue real de la funcionalidad del software es una forma de validarla. Se presenta a los clientes una implementación parcial de la funcionalidad y se utiliza su retroalimentación para determinar su valor [2].

La Dirección Tecnológica ha incentivado la participación de las personas desarrolladoras a fortalecer mediante las capacitaciones y uso de herramientas para la investigación, diseño de experimentos, diseño y aplicación de encuestas y entrevistas a las personas usuarias, recopilación de información en las retrospectivas de los Sprint, diseño de prototipos, actividades de desarrollo de software se han centrado en la entrega de software funcionando.

V. RESUMEN Y DISCUSIÓN DE LOS RESULTADOS

Los principales resultados de la integración de enfoques ágiles para el mejoramiento continuo de mejoras de procesos de software se presentan a continuación.

A. Indicadores clave de rendimiento

La medición continua y la automatización son componentes esenciales para la mejora del proceso de desarrollo de software y la generación de los informes. Como parte de la pila de producto del equipo de trabajo, se incluyeron actividades relacionadas con el proceso de definición de los indicadores clave de rendimiento. La fig. 4 muestra los pasos realizados para definir los indicadores clave de rendimiento iniciales del proceso. Para esto, se aplicó el método OKR para alinear los objetivos institucionales con los de la Dirección Tecnológica, utilizando como insumos principales la documentación formal de la institución y las normas técnicas.

Mediante una combinación de enfoques, como las normas técnicas, COBIT 2019 e ITIL junto con los objetivos establecidos, se procede a definir los resultados clave que generan valor para la Dirección Tecnológica según el equipo de trabajo de Agilismo. Este trabajo se llevó a cabo de manera iterativa e incremental siguiendo el marco de trabajo Scrum.

Durante las sesiones de trabajo se analizaron las normas técnicas y se fueron consultando las prácticas contenidas en los enfoques de consulta COBIT e ITIL. La Tabla I presenta la lista de indicadores, que incluye el resultado clave, el indicador, la métrica y la práctica fuente. Para cada uno de los KPI, se definieron los campos del rol responsable de recolectar el dato, la herramienta (ya sea una plataforma o artefactos), la periodicidad de recolección de los datos y la clasificación según el área estratégica, de gestión u operativa.

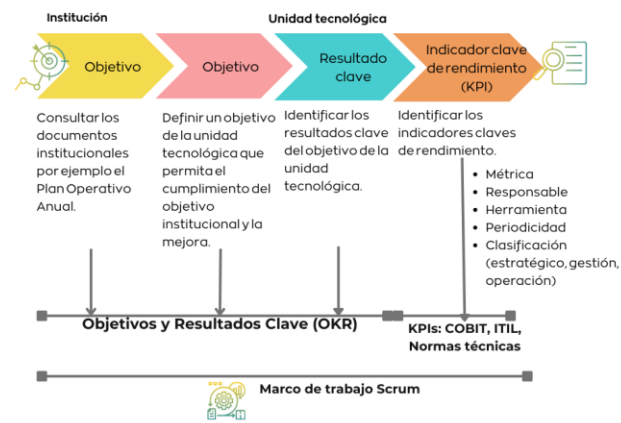


Figure 4. Proceso para la definición de KPI

Como resultado principal se generaron los principales indicadores clave de rendimiento para la Dirección Tecnológica para un total de 15, que serán tratados para la generación de información que colaboren para la toma de decisiones y la mejora continua del proceso.

TABLE I. INDICADORES CLAVE DE RENDIMIENTO

Resultado clave	KPI		
	KPI	Métrica	Práctica
Alineamiento de TI y la estrategia de la institución	Nivel de satisfacción de las partes interesadas con los requerimientos	Promedio general de encuesta de satisfacción: Escala Likert	COBIT
	Porcentaje de requisitos reelaborados debido a la falta de alineación con las necesidades y expectativas de la institución	(Cantidad de requisitos reelaborados / Cantidad total de requisitos)*100	COBIT
Entrega de servicios de TI de acuerdo a los requisitos de la institución	Número de errores descubiertos	Cantidad de errores por proyecto	COBIT



Resultado clave	KPI		
	KPI	Métrica	Práctica
Agilidad de las TI	Tiempo de retraso para desplegar un nuevo sistema (producto mínimo viable) en ambiente productivo	Promedio del tiempo de espera de un nuevo sistema para pasar a producción	COBIT
	El tiempo de duración en las etapas de diseño, desarrollo, pruebas Desarrollo, pruebas Aceptación de Usuario y seguridad.	Sumatoria de los tiempos de duración de cada una de las etapas	COBIT
	Porcentaje de historias aceptadas por PO	(Cantidad de requisitos aceptados / Cantidad total de requisitos por proyecto)*100	COBIT
	Esfuerzo - Puntos de historia cargados por sprint	Cantidad de puntos de historia por proyecto	COBIT
Seguridad de la información, infraestructuras de procesamiento y aplicaciones	Número de vulnerabilidades descubiertas durante las pruebas	Cantidad de vulnerabilidades detectadas durante las pruebas por proyecto	COBIT
	Bug clasificado seguridad: Tiempo dedicado a remediar vulnerabilidades	Cantidad de tiempo horas dedicado a remediar vulnerabilidades	COBIT
Conocimiento experiencia e iniciativas para la innovación	Frecuencia de la investigación y exploración del entorno realizadas para identificar ideas y tendencias innovadoras	Cantidad de investigaciones realizadas cada dos años	COBIT
Incidentes	Incidentes por nivel de prioridad	Número total incidentes por nivel de prioridad	ITIL
	Incidentes escalados al proveedor	Número total de incidentes escalados al proveedor	ITIL
	Porcentaje de satisfacción del usuario líder de servicio con la solución y recuperación del incidente	Promedio general de las encuestas de satisfacción. Escala Likert	ITIL
	Tiempo promedio de resolución del incidente (tiempo incidente estado de abierto – cerrado)	Promedio de tiempo de resolución de incidentes	ITIL
	Incidentes por categorías	Número de incidentes por las categorías definidas	ITIL

## B. Tablero de indicadores

Como parte de la integración de enfoques ágiles, el equipo de trabajo identificó la necesidad de mantener los indicadores clave de rendimiento de manera automatizada posible, ya que el mantenimiento manual implica una inversión adicional de tiempo para los equipos de trabajo y el personal de coordinación. A partir de la información definida, se inició un proceso para integrar los resultados de diversas fuentes de datos, incluyendo algunos indicadores que se recolectaban por primera vez.

Considerando la herramienta para extraer los datos de cada indicador, se procedió a la elaboración de un tablero automatizado para el cálculo de todos los KPI. Este tablero extrae datos de formularios en línea, como encuestas de satisfacción, registros de investigaciones y registros de restauraciones. Además, consulta información de la plataforma de gestión de proyectos, como el tiempo de duración en las etapas de diseño, desarrollo, pruebas de desarrollo, pruebas de Aceptación de Usuario y revisión de seguridad.

La herramienta utilizada para esta integración es PowerBI que consulta datos del Azure DevOps Server y formularios de encuesta / listas de datos de sitios en SharePoint.

Las figuras fig. 5 presenta los indicadores clave de rendimiento relacionados con los proyectos de software que gestiona la Dirección Tecnológica. Permite el uso de filtros por proyecto o bien contabilizar resultados de todos los proyectos, permite filtrar por fechas o bien por periodos de cuatrimestres y años. Puede consultar los datos de elementos de trabajo por cantidad y porcentaje ya sea para ver su estado o por tipo de elemento. Incluye datos generales de cantidad de defectos, el total de elementos de trabajo, las vulnerabilidades registradas, la encuesta de satisfacción con respecto a los requerimientos elaborados en promedio. La fig. 6 muestra específicamente, los datos de la encuesta de satisfacción de los requerimientos de software donde el promedio de satisfacción máximo es de 5 según la escala utilizada en el diseño de la encuesta. La fig. 7 presenta los datos de los incidentes clasificados por las categorías establecidas en el nuevo procedimiento de atención de incidentes y por nivel de prioridad.

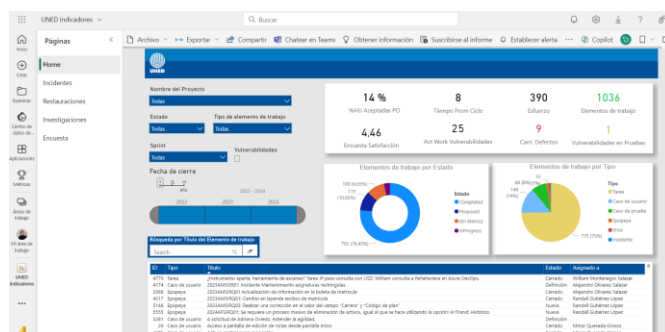


Figure 5. Tablero de indicadores - información de los proyectos

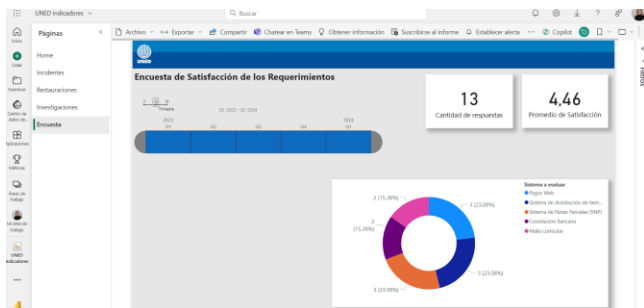


Figure 6. Tablero de indicadores – información de encuestas

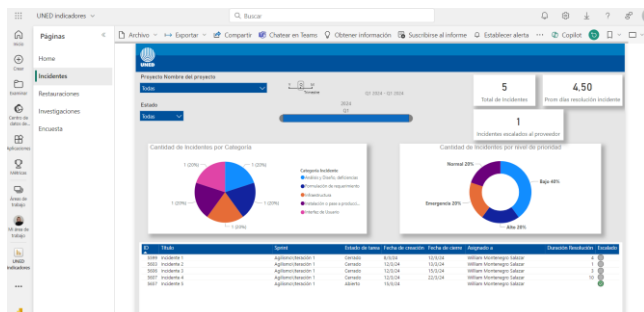


Figure 7. Tablero de indicadores – información de incidentes

### C. Mejora en la comunicación

El plan de comunicación surge como parte de la transparencia que requiere un proceso de integración de enfoques ágiles. Es una tarea que forma parte de la pila del producto del proyecto base de Agilismo, en el que se desarrollan las ideas de las temáticas. Para esto, se requirió de una persona adicional en desarrollo del material escrito y en video. Los comunicados inicialmente durante los años 2021-2022 se enviaron por correo electrónico institucional.

Sin embargo, por recomendaciones de las personas usuarias, surgió la necesidad de mejorar, lo que generó la investigación acerca de diversas herramientas hasta construir una comunidad ágil interna en la institución, que al 2024 aún está activa con material acorde a los temas requeridos enfocados a teoría de prácticas ágiles en general y en el contexto institucional, noticias, testimonios, talleres impartidos, elogios, encuestas de conceptos básicos de la teoría, entre otros, como se presenta en la fig.8. La comunidad ágil fue de uso privada para los equipos de trabajo de los proyectos pilotos, posteriormente, se abrió a la comunidad de la institución cuenta con 91 miembros activos en la comunidad. Sin embargo, las publicaciones llegan a alcanzar entre 50 a más de 300 visualizaciones por semana.

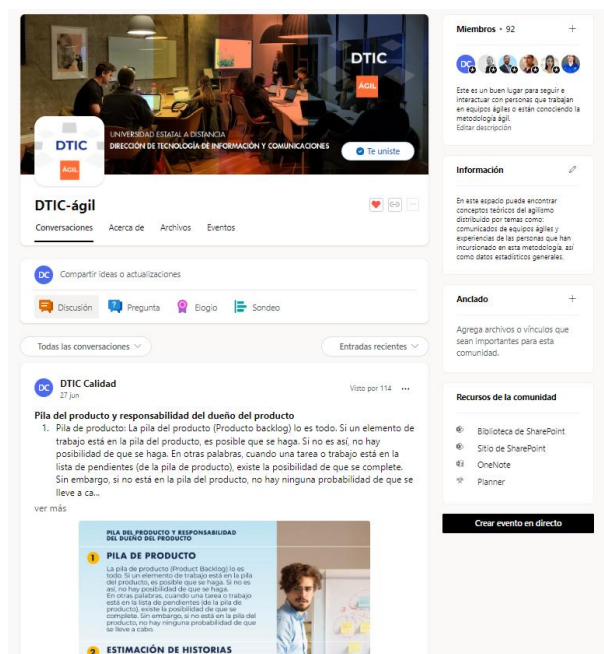


Figure 8. Comunidad ágil

### D. Gestión documental

La Dirección Tecnológica, como parte de institución del sector público, está sujeta a procesos de auditoría y cumplimiento normativo. Esto motivó la inclusión de tareas dedicadas a una revisión exhaustiva de la documentación utilizada en el enfoque tradicional dentro de las actividades del equipo de trabajo de Agilismo. Estas tareas incluyeron:

- Identificar la información estrictamente requerida para documentar.
- Incluir información relevante como parte de las actividades diarias de trabajo en el registro de artefactos contenidos en la nueva herramienta de trabajo como: la pila de producto de los proyectos, registro y control de pruebas, repositorio de código fuente.
- Configurar herramientas con el fin de agilizar las tareas de documentación. Esto permitió elaborar instructivos de uso de las herramientas como: (1) Azure Repos para el control de versiones y Git (2) Azure Test Plans para la documentación de planes de pruebas y ejecuciones de casos de pruebas (3) Instructivo para el uso del Azure DevOps en el registro, seguimiento y consulta de épicas, historias de usuario, configuración de los Sprints, capacidad del equipo (4) flujos de trabajo configurados en la herramienta para la épica y la historia de usuario, (5) configuración y uso de sitios de proyecto en Project Online (6) configuración de sitios de SharePoint con repositorios de documentación de los nuevos procedimientos formalizados e información relevante con la integración de la herramienta de comunidad ágil de comunicación (7) Guía de seguridad en el proceso de desarrollo de software.
- Identificar las fuentes de información requeridas para informes de seguimientos, evidencias de auditorías e

información que genera valor para la toma de decisiones de la Dirección Tecnológica.

- Documentar y formalizar en la institución los procedimientos de trabajo resultados de la mejora de procesos.

Como resultado, se creó una biblioteca de documentos internos que contiene procedimientos, instructivos, formularios y documentos tanto formalizados a nivel institucional como a nivel interno por la Dirección Tecnológica. En busca de la integración de la información, la biblioteca incluye secciones de información relevante interna como capacitaciones realizadas y en proceso, además de acceso vinculado a la comunidad ágil interna.

#### E. Flujos de trabajo

Un resultado importante fue la definición de los flujos de trabajo para poder automatizar algunas de las métricas relacionadas con la gestión de los proyectos. Esto requirió, la definición en el equipo de trabajo y configurarlos en los paneles de trabajo de la plataforma Azure DevOps Server.

Los principales flujos de trabajo configurados se presentan en la fig. 9 para la Épica se definieron 5 etapas: inicio, activa, aprobación de la persona dueña del producto, lanzamiento y terminado. Para la historia de usuario se definieron 9 etapas: definición, diseño, desarrollo, pruebas desarrollo, pruebas aceptación de usuario, revisión de seguridad, aprobación dueña de producto, release y terminado.



Figure 9. Flujos de trabajo

La metodología propuesta de una integración de enfoques ágiles para el mejoramiento continuo de mejoras de software basado en el modelo Stairway to Heaven y la Ingeniería de Software Continua, y el uso de otros métodos y herramientas como COBIT, ITIL, normativas técnicas ha permitido el avance continuo en las actividades del ciclo de vida del desarrollo ágil de software.

Se rescatan los valores de Scrum para el éxito basados en el compromiso, enfoque, apertura, respeto y coraje necesario para enfrentar el cambio. La comunicación continua, conocer el contexto de implementación y contar con recursos es indispensable para la mejora de procesos. La Dirección Tecnológica incluye en los presupuestos anuales fondos para: capacitación, pago de licenciamiento de algunas herramientas en uso e inversión de talento humano en los últimos tres años y

continuarán su labor con los siguientes escalones para alcanzar la innovación.

#### VI. CONCLUSIONES Y TRABAJO FUTURO

La integración de enfoques ágiles ha facilitado una evolución continua y la adaptabilidad en los procesos de desarrollo de software, permitiendo a la Dirección Tecnológica adaptarse rápidamente a las cambiantes necesidades del negocio y a las nuevas tecnologías.

En cuanto al impacto de la calidad se concluye que la integración de enfoques ágiles se proyecta en una mejora significativa en la calidad del software desarrollado, gracias a la implementación de prácticas de automatización y a la monitorización continua mediante KPI.

La colaboración entre distintos departamentos, incluyendo infraestructura tecnológica y desarrollo de software, es clave para el éxito de la integración de enfoques ágiles, fomentando un entorno de trabajo más cohesionado y eficiente. Adicionalmente, la colaboración del negocio como parte de los equipos ágiles permite obtener la retroalimentación durante el ciclo de vida de desarrollo ágil de software en contraste con el enfoque tradicional.

La sostenibilidad de los cambios implementados se ha visto fortalecida por la capacitación continua y el enfoque en la mejora incremental, asegurando que los procesos ágiles sean mantenidos y perfeccionados a lo largo del tiempo.

La integración de enfoques ágiles para el mejoramiento continuo de procesos de software, apoyada por la capacitación continua y la integración de herramientas de automatización, permite una mejora sustancial en los procesos de desarrollo de software en la Dirección Tecnológica, posicionándola como un modelo de referencia en la transformación digital y la gestión de la calidad con la experiencia obtenida en los últimos tres años.

Como trabajo futuro se espera que la consolidación de estos enfoques ágiles y la automatización de procesos no solo mejoren la eficiencia y calidad del desarrollo de software, sino que también contribuyan a la innovación continua dentro de la institución, alineándose con los objetivos estratégicos y las iniciativas de transformación digital.

#### AGRADECIMIENTOS

Se agradecen los invaluables aportes de las personas colaboradoras del equipo de trabajo de Agilismo de la DTIC-UNED: Paola Arias Ching, Randall Gutiérrez López y William Montenegro Salazar, así como el apoyo brindado por Francisco Durán Montoya y el grupo de profesionales que han colaborado en las actividades del proyecto. Se agradece al proyecto de Transformación Digital de la UNED, el cual ha apoyado esta iniciativa de mejora para la adopción de prácticas ágiles en la institución. Esta investigación forma parte del proyecto de investigación No. PROY0013-2024 de la UNED, y del Proyecto Nodos-Investiga de la DTIC-UNED. El trabajo fue parcialmente apoyado por el proyecto No. 834-C1-011 de la UCR.

## REFERENCIAS

- [1] B. Fitzgerald and K. J. Stol, "Continuous software engineering: A roadmap and agenda," *J. Syst. Softw.*, vol. 123, pp. 176–189, 2017, doi: 10.1016/j.jss.2015.06.063.
- [2] J. Bosch, *Continuous software engineering*, vol. 9783319112. 2014. doi: 10.1007/978-3-319-11283-1.
- [3] K. Schwaber, J. Sutherland, and L. G. Definitiva, "La Guía Scrum," 2020.
- [4] MICITT, *Estrategia de Transformación Digital*. San José, Costa Rica, 2018.
- [5] Sistema Costarricense de Información Jurídica, "Directriz N° 019-MP-MICITT Desarrollo del Gobierno Digital del Bicentenario," 2018. [https://www.pgrweb.go.cr/scij/Busqueda/Normativa/Normas/nrm\\_texto\\_completo.aspx?param1=NRTC&nValor1=1&nValor2=87415&nValor3=113907&strTipM=TC](https://www.pgrweb.go.cr/scij/Busqueda/Normativa/Normas/nrm_texto_completo.aspx?param1=NRTC&nValor1=1&nValor2=87415&nValor3=113907&strTipM=TC)
- [6] R. Arias Camacho, *Plan de Trabajo - Volver a Crecer 2019-2024*. San José, Costa Rica, 2019.
- [7] UNED, "CR-2021-1882-acuerdo tomado por el Consejo de Rectoría, sesión 2197-2021, Artículo I, inciso 9), celebrada el 15 de noviembre del 2021," 2021. <https://www.uned.ac.cr/sites/default/files/content/tutoria/Declaratoria de Interes Institucional y Conformacion Equipo Lider.pdf>
- [8] UNED, "Estrategia Transformación digital en la UNED," 2021. <https://www.uned.ac.cr/sites/default/files/inline-files/Propuesta de Estrategia de Transformación digital en la UNED.pdf>
- [9] UNED, "Transformación digital de la UNED Hoja de ruta 2023-2025," 2023. [https://www.uned.ac.cr/sites/default/files/inline-files/Hoja de Ruta\\_TD 2023-2025-3.pdf](https://www.uned.ac.cr/sites/default/files/inline-files/Hoja de Ruta_TD 2023-2025-3.pdf)
- [10] UNED-DTIC, *Plan estratégico de la Dirección de Tecnologías de Información y Comunicaciones 2020-2024*. San José, Costa Rica, 2020.
- [11] UNED, "Objetivos institucionales de calidad," 2023. <https://www.uned.ac.cr/docencia/images/cidreb/Normativa/Objetivoscalidad.pdf>
- [12] O. T. Chacón, J. M. Flores, and J. M. Flores, "Normas técnicas para la gestión y el control de las Tecnologías de Información," 2021.
- [13] A. Cecilia et al., "Marco de gobierno y gestión de la UNED v1.0".
- [14] E. Hernández-agüero et al., "Una Combinación de Enfoques para una Mejora de Procesos de Software," pp. 47–57, 2018.
- [15] E. Hernández-Agüero, "Diseño de un Plan de mejoras dirigido a la Gestión de la Calidad para la Dirección de Tecnologías de Información y Comunicaciones de la Universidad Estatal a Distancia," Universidad de Costa Rica, 2018.
- [16] F. Cocozza, E. Brenes, G. L. Herrera, M. Jenkins, and A. Martínez, "Application of GQM+strategies in a small software development unit," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8892, pp. 108–118, 2014, doi: 10.1007/978-3-319-13835-0\_8.
- [17] ISACA, *COBIT 2019 Framework: Governance and Management Objectives*. 2019.
- [18] MICITT, "Normas MICITT Matriz Guía Implementación Prácticas de Gobierno y Gestión 2021-Excel," 2024. <https://www.micitt.go.cr/node/630>
- [19] Axelos, *ITIL Foundation*, 4 edition. 2019.
- [20] D. Galin, *Software Quality Assurance From theory to implementation*. England: Pearson Education, 2004.
- [21] Association for Project Management, "Agile project management glossary," <https://www.apm.org.uk/resources/find-a-resource/agile-project-management/glossary/>
- [22] M. Kuhrmann et al., "Hybrid Software Development Approaches in Practice: A European Perspective," *IEEE Softw.*, vol. 36, no. 4, pp. 20–31, 2019, doi: 10.1109/MS.2018.110161245.
- [23] C. R. Kuhrmann, Marco and Diebold, Philipp and Munch, Jurgen and Tell, Paolo and Garousi, Vahid and Felderer, Michael and Trektere, Kitija and McCaffery, Fergal and Linssen, Oliver and Hanser, Eckhart and Prause, "Hybrid Software and System Development in Practice: Waterfall, Scrum, and Beyond," 2017, doi: 10.1145/3084100.3084104.
- [24] J. L. Butler, T. Zimmermann, and C. Bird, "Objectives and Key Results in Software Teams: Challenges, Opportunities and Impact on Development," *ACM Int. Conf. Proceeding Ser.*, no. Section 6, pp. 358–368, 2024, doi: 10.1145/3639477.3639747.
- [25] M. Staron, W. Meding, K. Niesel, and A. Abran, "A key performance indicator quality model and its industrial evaluation," *Proc. - 26th Int. Work. Softw. Meas. IWSM 2016 11th Int. Conf. Softw. Process Prod. Meas. Mensura 2016*, pp. 170–179, 2017, doi: 10.1109/IWSM-Mensura.2016.033.
- [26] J. O. Johanssen, A. Kleebaum, B. Paech, and B. Bruegge, "Practitioners' eye on continuous software engineering: An interview study," *ACM Int. Conf. Proceeding Ser.*, pp. 41–50, 2018, doi: 10.1145/3202710.3203150.
- [27] A. D. Nagarajan and S. J. Overbeek, *A DevOps Implementation Framework for Large Agile-Based Financial Organizations*, vol. 11229 LNCS. Springer International Publishing, 2018. doi: 10.1007/978-3-030-02610-3\_10.
- [28] A. Janes, V. Lenarduzzi, and A. C. Stan, "A continuous software quality monitoring approach for small & medium enterprises," *ICPE 2017 - Companion 2017 ACM/SPEC Int. Conf. Perform. Eng.*, pp. 97–100, 2017, doi: 10.1145/3053600.3053618.
- [29] M. Zahedi, R. N. Rajapakse, and M. A. Babar, "Mining Questions Asked about Continuous Software Engineering: A Case Study of Stack Overflow," *ACM Int. Conf. Proceeding Ser.*, pp. 41–50, 2020, doi: 10.1145/3383219.3383224.
- [30] A. Wannes and S. A. Ghannouchi, "KPI-Based Approach for Business Process Improvement," *Procedia Comput. Sci.*, vol. 164, pp. 265–270, 2019, doi: 10.1016/j.procs.2019.12.182.
- [31] P. Sowkasem, Chutiwan and Kirawanich, "A Deliverable Delay Management of Software Development in Railway Project using an OKR-Based Scrum Process," 2021, pp. 10–16. doi: 10.1145/3451471.3451473.
- [32] CertiProf, *Scrum Master Professional Certificate*. 2022.