Evaluación de la conveniencia de adoptar DevOps como estrategia para la entrega continua de productos de software en la Alianza Bioversity – CIAT sede las Américas, Colombia.

Jose Libardo Molina Galvis

Nota de Aceptación

Certificamos que el presente Trabajo de Grado Satisface, en alcances y calidad, todos los requisitos que demanda un Trabajo de Grado de Maestría.

FELIPE MEJÍA V.

Felipe Mejía Villegas
Director

Luisa Fernanda Rincón
Codirectora

Mario Mora
Jurado

David Noreña
Jurado

Aprobado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana Cali, para optar el título de Magister en Ingeniera de Software. ✓

HERNAN CAMILO ROCHA NIÑO Ph. D.
Decano Facultad de Ingeniería y Ciencias

JUAN CARLOS MARTÍNEZ ARIAS
Director Posgrados de Ingeniería y Ciencias

Santiago de Cali, 10 de octubre del 2022

Acta de Correcciones al Documento de Trabajo de Grado

Santiago de Cali, octubre 6 de 2022

Autor: Jose Libardo Molina Galvis

Título del Trabajo de Grado: "Evaluación de la conveniencia de adoptar DevOps como estrategia para la entrega continua de productos de software en la Alianza Bioversity CIAT sede las Américas, Colombia"

Director: Felipe Mejía Villegas

Como indica el artículo 2.13 de las Directrices para Trabajo de Grado de Maestría, he verificado que el estudiante indicado arriba ha implementado todas las correcciones que los Jurados del Proyecto de Trabajo de Grado definieron que se efectuaran, como consta en el Acta de Evaluación correspondiente.

FELIPE MEJÍA V.

Firma del Director del Trabajo de Grado

Datos actuales del estudiante

Nombre completo: Jose Libardo Molina Galvis

Dirección: Calle 4 # 2b-32, Guacarí-Valle

Teléfonos de contacto: 3013325282

Correo electrónico: jlmolina@javerianacali.edu.co

Profesión: Ingeniero de sistemas

Universidad: Universidad del Valle

Empresa: Alianza de Bioversity International y el Centro Internacional de Agricultura Tropical (CIAT)

Cargo: Analista de sistemas

Pontificia Universidad
JAVERIANA
Cali

Evaluación de la conveniencia de adoptar DevOps como estrategia para la entrega continua de productos de software en la Alianza Bioversity – CIAT sede las Américas, Colombia.

REALIZADO POR:

Jose Libardo Molina Galvis

Trabajo de grado para optar al título de Magister en Ingeniera de Software

DIRIGIDO POR:

Director
Felipe Mejia, M.Sc.

Codirectora
Luisa Fernanda Rincón, Ph.D

Facultad de Ingeniería y Ciencias
Maestría en Ingeniería De Software

Santiago de Cali, 2022

Índice general

1. Aspectos Generales **1**
1.1. Planteamiento del problema . . . 1
1.1.1. Problema . . . 3
1.2. Objetivos del proyecto . . . 3
1.2.1. Objetivo general . . . 3
1.2.2. Objetivos específicos . . . 3
1.2.3. Resultados esperados . . . 4
1.3. Alcance . . . 4

2. Revisión de literatura **5**
2.1. Marco teórico . . . 5
2.1.1. Ciclo de vida del desarrollo de software . . . 5
2.1.2. Modelos de proceso de software . . . 6
2.1.2.1. El modelo en cascada (*waterfall*) . . . 6
2.1.2.2. Agile . . . 6
2.1.3. DevOps . . . 7
2.1.3.1. Conceptos y definiciones: . . . 7
2.1.3.2. Historia . . . 9
2.1.3.3. Prácticas DevOps . . . 10
2.1.3.4. ¿Qué no es DevOps? . . . 12
2.1.3.5. Métricas de rendimiento . . . 13
2.1.3.6. Pilares y Objetivos DevOps . . . 13
2.1.3.7. Estrategias de adopción DevOps . . . 14
2.2. Estado del arte . . . 16
2.2.1. Trabajos relacionados . . . 16
2.2.2. Modelos de Madurez DevOps . . . 17
2.2.2.1. Accelerate . . . 18
2.2.2.2. Modelo CALMS . . . 20
2.2.2.3. OWASP DevSecOps Maturity Model . . . 22
2.2.3. Modelos de medición de DevOps . . . 23
2.2.3.1. Devops Health Radar Assessment . . . 23
2.2.3.2. Dora Assessment . . . 24
2.3. Antecedentes . . . 25
2.3.1. Alianza Bioveristy - CIAT . . . 25
2.3.2. CIAT y DevOps . . . 26

2.4. Resumen del capítulo 27

3. Propuesta del artefacto de evaluación 29
3.1. Modelo de capacidades 29
3.1.1. Dimensión 31
3.1.2. Capacidad 32
3.1.3. Pregunta 33
3.1.4. Respuesta 34
3.1.5. Plan de acción 34
3.2. Formato de respuestas y sistema de puntaje 34
3.2.1. Captura de información de entrada 34
3.2.2. Transformación de resultados en unidades de análisis 35
3.2.3. Herramienta de implementación del instrumento 37
3.2.4. Presentación de resultados 38
3.2.5. Proceso para usar el instrumento de evaluación 39
3.2.6. Limitaciones 40
3.3. Resumen del capítulo 40

4. Caso de estudio 43
4.1. Ejecución del caso de estudio 43
4.1.1. Configuración del caso de estudio y objetivos del análisis . . . 43
4.1.2. Ejecución del caso de estudio 43
4.2. ¿Cómo iniciar la adopción DevOps? 46
4.2.1. Identificar el Estado objetivo 46
4.2.2. Entender el estado actual 47
4.2.2.1. Visualización de datos 47
4.2.2.2. Resultados 48
4.2.2.3. Análisis de los resultados 49
4.2.3. *Análisis final* 54
4.2.4. Determinar el mejor camino a seguir 55
4.2.4.1. Aspectos a tener en cuenta 58
4.2.5. ¿CIAT si necesita DevOps? 60
4.3. Calidad de la evaluación 60
4.3.1. Riesgo de la validez 60
4.4. Resumen del capítulo 61

5. Conclusiones y trabajos futuros 63
5.1. Conclusiones 63
5.2. Lecciones aprendidas 65
5.3. Trabajos futuros 66

A. Encuesta de medición de la percepción de los productos y los servicios prestados 67

B. Modelo de capacidades para la evaluación DevOps **73**
B.1. Preguntas del modelo de capacidades 73

C. Plan de acción **117**

Bibliografía **136**

Índice de figuras

2.1. Modelo en cascada 6
2.2. Características generales de los marcos de trabajo Agile 7
2.3. Enfoques y marcos de trabajo ágiles 8
2.4. Cultura DevOps 9
2.5. DevOps en el ciclo de vida del desarrollo de software 10
2.6. Integración continua 11
2.7. Entrega continua 11
2.8. Diferencias entre integración continua, entrega y despliegue 12
2.9. Escala de rendimiento por métricas 14
2.10. Evolución de la industria en escala de rendimiento 15
2.11. Áreas clave para la adopción DevOps 15
2.12. Modelo de competencias – propuesto por Rico de Feijte 18
2.13. Modelo general de la investigación propuesto por Forsgren et al. (2018) 20
2.14. Modelo CALM para DevOps 22
2.15. Modelo de madurez Devsecops de OWASP 23
2.16. Modelo de evaluación de la salud de DevOps - DevOps Health Radar 24
2.17. Ejemplo resultado evaluación DORA 24
2.18. Estructura Organizacional del área de integración tecnológica Alianza Bioversity -CIAT 26
2.19. Herramientas DevOps instaladas en CIAT 27
2.20. Pipeline de despliegue implementado en CIAT como piloto DevOps . 27

3.1. Estructura del *modelo de capacidades* propuesto 30
3.2. Ejemplo estructura de preguntas del *modelo de capacidades* 31
3.3. Modelo de referencia 33
3.4. Transformación de resultados en unidades de análisis 35
3.5. Ejemplo de reporte dashboard 39
3.6. Proceso de medición de capacidades 39

4.1. Número de respuestas de la encuesta de capacidades en CIAT 45
4.2. *Modelo de capacidades*, distribución de encuestados 45
4.3. Nota final del caso práctico 46
4.4. Pasos claves para iniciar una transformación DevOps 46
4.5. Tiempos de despliegue 2021 de aplicaciones del área de integración de sistemas – CIAT 47

4.6. Frecuencia de entrega a producción 2021 de aplicaciones del área de integración de sistemas – CIAT 47
4.7. Resumen de los resultados del caso de estudio 48
4.8. Resultados de evaluación por categoría y dimensión 49

A.1. Resultado encuesta de percepción de los productos y los servicios prestados por la unidad de Integración de Sistemas del departamento de Tecnología de Información 69
A.2. Encuesta de percepción respuestas a la pregunta 1 69
A.3. Encuesta de percepción respuestas a la pregunta 2 70
A.4. Encuesta de percepción respuestas a la pregunta 3 70
A.5. Encuesta de percepción respuestas a la pregunta 4 71
A.6. Encuesta de percepción respuestas a la pregunta 5 71
A.7. Encuesta de percepción respuestas a la pregunta 6 71
A.8. Encuesta de percepción respuestas a la pregunta 6, comentarios . . . 72

B.1. Cultura y liderazgo transformacional 115

Índice de tablas

3.1. Nivel de escala de notas 35
3.2. Escala de categorías por calificación 37

4.1. Resultado escala de clasificación - Oportunidades de mejora 50
4.2. Resultado escala de clasificación Aceptable 52
4.3. Resultado escala de clasificación Fortalezas 53
4.4. Capacidades representativas por etapa 59

C.1. Plan de acción resultante de la evaluación CIAT 135

PALABRAS CLAVE

DevOps, Transformación Digital, Desarrollo de Software, Modelo de Madurez, Artefacto de Evaluación DevOps, Entrega Continua, Automatización

Abstract

The adoption of a widespread and popularized set of best practices for software development such as DevOps in an organization is a task that is not achieved overnight, it involves evaluating and adapting how the lifecycle of software development stages are being implemented in a company, in addition to a change in the organizational culture and process management according to the particularities of each company. With this type of transformation, the goal is to deliver software products to end users in the shortest possible time, ensuring quality and reliability, through the identification of bottlenecks in processes, procedures and work methodologies.

This document evaluates the convenience of adopting a set of best practices for software development such as DevOps to a company whose business focus is not to produce software, but an agricultural research center that uses software for its operational and functional support through internal applications and that, like any other company, requires agile and timely processes to improve its operation.

One of the aspects to highlight is the proposal of an artifact for DevOps assessment in an organization that through indirect questions designed for the segment of people who participate in the software development life cycle and are unfamiliar with DevOps terminology, can assess the state of corporate software development practices from a multidimensional point of view. The proposed model evaluates dimensions such as culture, process management, LEAN management, architecture and continuous delivery, with these elements it was possible to determine the maturity status of a group of practices that support better performance in the delivery of quality software products. The proposed model was designed with the benefit of relating an action plan from each response option of the evaluated aspects.

The Bioversity International Alliance and the International Center for Tropical Agriculture (CIAT) was evaluated as a particular study case of the proposed evaluation artifact and from these results an analysis was generated to prioritize which elements should be addressed according to their level of maturity and begin the progressive adoption of DevOps practices focusing on improving the speed in the frequency of deployments and the reduction of deployment time.

Resumen

La adopción de un conjunto de buenas prácticas muy difundidas y popularizadas para el desarrollo de software como lo es DevOps en una organización es una tarea que no se logra de la noche a la mañana, implica evaluar y adecuar cómo se están implementando las etapas del ciclo de vida del desarrollo de software en una compañía, además de un cambio en la cultura organizacional y la gestión de los procesos acorde con las particularidades de cada compañía. Con este tipo de transformaciones se busca entregar productos de software a los usuarios finales en el menor tiempo posible, asegurando calidad y confiabilidad, a través de la identificación de cuellos de botellas en procesos, procedimientos y metodologías de trabajo.

Este documento busca evaluar la conveniencia de adoptar un conjunto de buenas prácticas muy difundidas y popularizadas para el desarrollo de software como lo es DevOps a una empresa cuyo foco de negocio no es producir software, sino un centro de investigación en agricultura que utiliza el software para su soporte operativo y funcional por medio de aplicativos internos y que como toda empresa requiere procesos agiles y oportunos para mejorar su operación.

Uno de los aspectos a destacar es la propuesta de un artefacto para la evaluación DevOps en una organización que a través de preguntas indirectas pensadas para el segmento de personas que participan en el ciclo de vida de desarrollo de software y no estén familiarizadas con la terminología DevOps, puedan evaluar el estado de las prácticas de desarrollo de software corporativas desde un punto de vista multidimensional. El modelo propuesto evalúa dimensiones como la cultura, la gestión de procesos, la gestión LEAN, la arquitectura y la entrega continua, con estos elementos se pudo determinar el estado de madurez de un grupo de prácticas que apalancan mejores rendimientos en la entrega de productos de software de calidad. El modelo propuesto se diseñó con el beneficio de relacionar un plan de acción a partir de cada opción de respuesta de los aspectos evaluados.

La Alianza Bioversity International y el Centro Internacional de Agricultura Tropical (CIAT) fue evaluada como caso de estudio particular del artefacto de evaluación propuesto y a partir de estos resultados se generó un análisis para priorizar que elementos deben ser tratados acorde a su nivel de madurez y así iniciar la adopción progresiva de las practicas DevOps enfocándose en mejorar la velocidad en la frecuencia de despliegues y la reducción de tiempo en el despliegue.

Capítulo 1

Aspectos Generales

1.1. Planteamiento del problema

La Alianza Bioversity y el Centro internacional de agricultura tropical - CIAT es un centro de investigación en agricultura sin ánimo de lucro con presencia mundial que se creó para enfrentar las crisis de cambio climático, pérdida de la biodiversidad, degradación ambiental y malnutrición, es parte del CGIAR, el mayor consorcio mundial en investigación e innovación agrícola enfocado a contribuir a la seguridad alimentaria y nutricional, reducir el hambre y la pobreza. La compañía tiene presencia a nivel mundial en cuatro sedes regionales, Américas, Europa, África y Asía, para el caso colombiano, la sede regional para las Américas está ubicada en la ciudad de Palmira, Colombia.

La organización centra sus esfuerzos humanos y económicos en la investigación científica y producción de conocimiento en agricultura. A nivel estructural y organizacional la división entre procesos de investigación y no investigación es marcada por lo que muchas actividades presentan diferentes grados relevancia de ejecución y priorización. La informática, tiene un papel representativo transversal enfocado a soporte de procesos científicos y administrativos acorde con las divisiones estructurales de la organización. Del lado de investigación, el alcance informático se centra en el desarrollo de soluciones a tareas particulares de cada área, en donde salvo contadas excepciones solo se entrega la aplicación con poca documentación de su ciclo de vida sin un responsable técnico definido. En la parte administrativa, se tiene un área tecnológica definida con diferentes unidades de soporte, desarrollo de software, operaciones e infraestructura, se cuenta con un portafolio de soluciones informáticas compuesto por un ERP para procesos administrativos y financieros institucionales y un grupo de aplicaciones desarrolladas a la medida denominadas aplicaciones periféricas, estas últimas fueron desarrolladas para uso interno principalmente dentro de la sede las Américas y algunas para todas las oficinas regionales con la que cuenta la organización.

Debido al tipo de organización, los proyectos de software son internos con duraciones cortas y medianas no mayores a 6 meses, normalmente cubiertos con personal propio, donde el factor económico no juega el rol decisivo en el inicio de los proyectos

como en una empresa comercial, por el contrario, la responsabilidad de inicio o no de un proyecto recae en la alta gerencia del departamento de acuerdo a la disponibilidad del recurso humano.

En los últimos 10 años, el área de desarrollo en administración ha perdido rigurosidad en el uso de directrices y metodologías para gestionar y agilizar el ciclo de vida de un producto de software, trasladando muchas de las decisiones técnicas y buenas prácticas a juicio de las personas involucradas en el desarrollo de los proyectos, predominando muchos procesos manuales durante todo el ciclo de vida del desarrollo. Por ejemplo, se tiene una deuda técnica acumulada durante los años, en donde las tareas relacionadas a la compilación y construcción de los proyectos se siguen haciendo de forma manual sin ninguna herramienta o tarea de automatización. En pruebas y calidad las tareas se dejan a la experticia de los desarrolladores, en ocasiones sin criterios de evaluación bien definidos y tareas manuales. Por último, el despliegue de soluciones en los ambientes de pruebas y producción también se hace de forma manual, incrementando los tiempos de entrega de las soluciones a los usuarios finales.

Desde inicios del 2020 el área de desarrollo y soporte administrativo se reestructuraron en una nueva área llamada integración de sistemas. Esta área tiene el propósito de mejorar, agilizar y potenciar las tareas del ciclo de vida del desarrollo de software. De esta forma se llegó al primer acercamiento teórico al término DevOps con sus buenas prácticas y recomendaciones, a finales del 2020 en conjunto con las áreas de redes se dio el primer paso tangible a DevOps. A través de un modelo de consultoría se contrató a una compañía experta en prácticas y herramientas DevOps para la instalación y configuración de herramientas particulares, como, por ejemplo, un orquestador de tareas, un gestor de dependencias y un analizador de código estático.

La iniciativa anterior no se acompañó de un plan de adopción claro ni de la reestructuración de procedimientos internos y culturales por lo que no tuvo gran acogida en la organización, por lo que al año 2022 no se les ha dado un uso extendido y constante a las herramientas instaladas. Con el ánimo de retomar la iniciativa para mejorar los tiempos de entrega, potencializar el trabajo colaborativo entre el desarrollo y operaciones, este trabajo busca evaluar la conveniencia de adoptar prácticas DevOps en las áreas de integración de sistemas y redes (operaciones) del CIAT, oficina regional Colombia a partir de un diagnóstico de las capacidades técnicas y operativas de las áreas.

Adoptar la filosofía DevOps implica una transformación a nivel de procesos, cultura y tecnología de las organizaciones. Una transformación de este tipo implica de una parte conocer cuál es el estado de la organización para asumir el cambio y por otra parte conocer cuáles son los elementos que se deben considerar para llevar a cabo la transformación digital en DevOps. Sin embargo, el CIAT no es experto en estas temáticas ni tiene las capacidades técnicas y culturales para ejecutar un cambio de este tipo, por lo tanto, este trabajo permitirá evaluar el estado de los procesos de desarrollo de software las áreas de desarrollo y operaciones frente a las recomendaciones DevOps y brindará los criterios y orden de priorización de los elementos requeridos para la adopción de la filosofía acorde al estado de madurez de los procesos de la compañía. El propósito es que las áreas de Integración de Sistemas y Redes y Seguridad del CIAT tengan los elementos necesarios para iniciar un proceso de adopción

progresivo de un piloto DevOps definido y estructurado acorde a los resultados de la evaluación de la organización en el marco de evaluación DevOps propuesto y las recomendaciones brindadas en este documento.

1.1.1. Problema

A partir de identificar las prácticas de desarrollo software que implementan las áreas de Integración de Sistemas y Redes en la sede para las Américas del CIAT durante el ciclo de vida de las aplicaciones y evaluándolas frente a un modelo que permite identificar el estado actual de un grupo de prácticas corporativas frente a las recomendaciones y buenas prácticas que ofrece DevOps para poder iniciar un proceso de transformación digital particularmente en las áreas de Integración de Sistemas y Redes de la compañía, la principal pregunta para contestar con el presente trabajo es:

¿Cómo evaluar la conveniencia de adoptar DevOps como estrategia para la entrega continúa de productos de software de la Alianza Bioversity CIAT - Sede Regional para las Américas - Palmira, Colombia?

En particular se espera responder las siguientes preguntas:

- ¿Cómo evaluar si DevOps es una alternativa que ayudaría al CIAT a iniciar un proceso de transformación de sus procesos de entrega continua de productos de software?
- ¿Cómo diagnosticar los procesos de entrega de software versus las buenas prácticas propuestas por DevOps acorde al tipo de empresa que es CIAT?
- ¿Cómo planear la adopción progresiva de DevOps en los procesos de entrega continua de productos de software acorde a las necesidades puntuales de CIAT?

1.2. Objetivos del proyecto

1.2.1. Objetivo general

Evaluar la conveniencia de adoptar DevOps como estrategia para la entrega continúa de productos de software de la Alianza Bioversity - Centro Internacional de Agricultura Tropical – CIAT – Sede Regional para las Américas - Palmira, Colombia

1.2.2. Objetivos específicos

1. Definir los factores a considerar para evaluar la conveniencia de adoptar DevOps como estrategia de entrega continua de productos de software en la organización.
2. Proponer un artefacto para evaluar la conveniencia de adoptar DevOps en el CIAT a partir de los factores identificados.

3. Diagnosticar el estado de los procesos de entrega de software en el CIAT con el instrumento propuesto para determinar si DevOps es una solución adecuada a las necesidades del CIAT.

4. Proponer un plan de acción a partir del diagnóstico obtenido.

1.2.3. Resultados esperados

- Artefacto para diagnosticar el nivel de madurez de las prácticas actuales en la entrega de productos software en el CIAT versus las buenas prácticas que ofrece DevOps
- Un artefacto que permita evaluar el estado de las prácticas DevOps en una organización con la capacidad de relacionar las opciones de respuesta de cada pregunta a un plan de acción.
- Resultado de la evaluación de las necesidades de CIAT versus DevOps
- Plan de acción a partir de los resultados del análisis de las necesidades particulares en la entrega de productos de software en el CIAT
- Una guía de recomendaciones para mejorar el proceso de entrega de productos de software en el CIAT en caso de que la conclusión del diagnóstico sea NO usar DevOps

1.3. Alcance

Este trabajo define criterios de evaluación para construir un artefacto para diagnosticar el estado de madurez de los procesos de entrega de productos de software por parte del área de integración de sistemas de CIAT versus las recomendaciones y buenas prácticas que ofrece DevOps para la entrega continua de productos de software.

A partir del resultado del diagnóstico se determinó que las recomendaciones y buenas prácticas DevOps se pueden aplicar a cualquier tipo de organización por lo que son convenientes y pertinentes para atender las necesidades de CIAT dado sus particularidades y tipo de compañía.

Por último, se propone una estrategia de adopción DevOps enfocada en mejorar el rendimiento en los tiempos de entrega de software en la organización. Este plan se genera acorde a los hallazgos encontrados al evaluar un conjunto de prácticas DevOps del artefacto de evaluación en la organización y que se expresan con una calificación del nivel de madurez y con un plan de acción por cada elemento evaluado. A partir de estos resultados la CIAT tiene los elementos necesarios para iniciar y estructurar una adopción progresiva de una transformación digital DevOps.

Capítulo 2

Revisión de literatura

En este capítulo se presenta el marco teórico y los trabajos previos que apalancan la realización del presente proyecto. Se abordarán temas de transformación digital DevOps y como este se involucra en el ciclo de vida de los proyectos de software, sus prácticas, sus pilares y los diferentes modelos de madurez y medición en las organizaciones.

2.1. Marco teórico

2.1.1. Ciclo de vida del desarrollo de software

El Desarrollo de software se compone de un grupo de procesos, actividades y tareas que se relacionan entre sí para elaborar un producto de software, a este grupo de métodos se les conoce como ciclo de vida del desarrollo de software en inglés por sus siglas Software Development Life Cycle (SDLC). Dentro del mercado existe una gran variedad de modelos, entre los más destacados de la industria están Cascada, Espiral y Ágil.

Sommerville (2011) define cuatro actividades fundamentales que son comunes a todos los procesos de desarrollo de software.

1. **Especificación del software**, donde clientes e ingenieros definen el software que se producirá y las restricciones en su operación.

2. **Desarrollo del software**, donde se diseña y programa el software.

3. **Validación del software**, donde se verifica el software para asegurar que sea lo que el cliente requiere.

4. **Evolución del software**, donde se modifica el software para reflejar los requerimientos cambiantes del cliente y del mercado.

A nivel de estándares y normatividades, existe la norma ISO ISO/IEC 12207 que define las etapas básicas del proceso de desarrollo de software.

- **Análisis:** Se identifican los requisitos que debe cumplir el software y se construye un modelo de dichos requisitos.
- **Diseño:** A partir del modelo de análisis se identifican los procesos y las estructuras de datos en las que se descompone el sistema y además se construye un modelo del sistema a desarrollar.
- **Codificación:** Se construye el sistema en sí mismo
- **Pruebas:** Se comprueba que el sistema construido es correcto y cumple con el modelo de requisitos
- **Mantenimiento:** Esta fase tiene lugar tras la entrega del producto acabado y en ella se trata de asegurar que el sistema siga funcionando y adaptándose a los nuevos requisitos.

2.1.2. Modelos de proceso de software

2.1.2.1. El modelo en cascada (*waterfall*)

Davis et al. (2016) define el modelo de cascada como un mecanismo de entrega de proyectos en el que las tareas se completan en un orden secuencial para lograr un objetivo específico, cada fase solo inicia si la fase anterior está completa.

La Figura 2.1 presenta las diferentes etapas que componen el modelo en cascada, donde se implementan las actividades fundamentales del proceso como son: Análisis y definición de requerimientos, el diseño, la implementación, la integración, las pruebas, la operación y el mantenimiento (Sommerville, 2011).

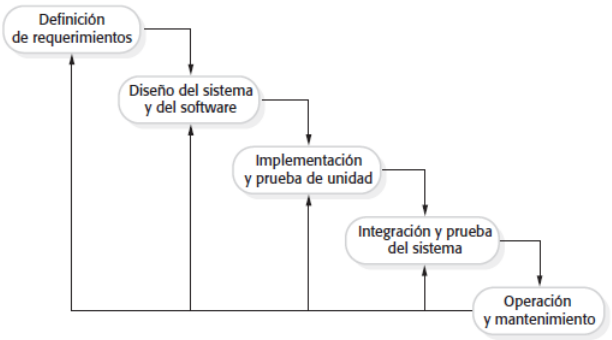

Figura 2.1: Modelo en cascada
Imagen tomada de (Sommerville, 2011)

2.1.2.2. Agile

Agile es el nombre dado a un grupo de marcos de trabajo de desarrollo de software que están diseñadas para ser más ligeras y flexibles que los métodos considerados tradicionales como la cascada (Davis et al., 2016).

El término "Ágil" es un término relativamente nuevo, acuñado oficialmente en 2001 cuando se publicó un documento llamado Manifiesto Ágil que define el siguiente conjunto de valores: (Beck et al., 2001)

- Individuos e interacciones sobre procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual
- Respuesta ante el cambio sobre seguir un plan

La Figura 2.2 representa algunos de los elementos que componen los marcos de trabajo ágiles. Las tareas se dividen en bloques de tiempo, con el fin de entregar características específicas para un lanzamiento. Se adopta un enfoque iterativo y después de cada iteración se entrega una versión del software funcional y cada versión es incremental en términos de características, por último, la versión final contiene todas las características requeridas por el cliente (Project Management Institute, 2017).

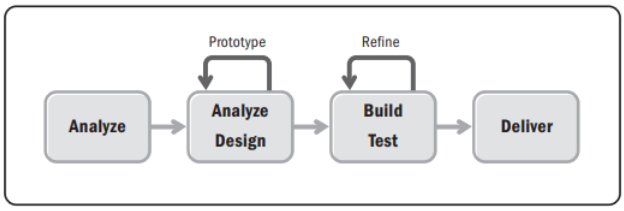

Figura 2.2: Características generales de los marcos de trabajo Agile
Imagen tomada de (Project Management Institute, 2017)

Los marcos de trabajo ágiles más populares en la industria son Extreme Programming, Scrum, Rational Unified Process, Crystal Clear, Adaptive Software Development, entre otras.

En la Figura 2.3 se muestra la relación de composición entre los marcos de trabajo *Agile* y los enfoques *Agile*, donde la agilidad y el método *Kanban* se presentan como subconjuntos de *Lean* porque comparten conceptos particulares, como: "centrarse en el valor", "lotes de pequeño tamaño" y "eliminación de residuos" (Project Management Institute, 2017).

2.1.3. DevOps

2.1.3.1. Conceptos y definiciones:

Por definición DevOps es la unión de las palabras "Desarrollo" (development) and "Operaciones" (operation), es una práctica que fomenta el trabajo colaborativo y la comunicación entre las personas involucradas en desarrollo de software.

El término DevOps, es un concepto tecnológico relativamente nuevo que la industria del software ha adoptado. Muchos autores no tienen un consenso definitivo de

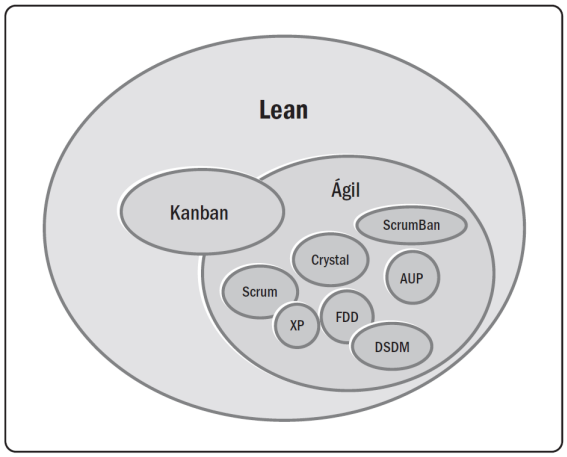

Figura 2.3: Enfoques y marcos de trabajo ágiles
Imagen tomada del libro (Project Management Institute, 2017)

una definición universal, a continuación, se presenta la versión de diferentes autores referentes de la industria:

Davis et al. (2016): "DevOps es una forma de pensar y de trabajar. Es un marco para compartir historias y desarrollar la empatía, permitiendo a las personas y a los equipos aplicar sus conocimientos de manera eficaz y duradera. Es parte del tejido cultural que da forma a cómo trabajamos y por qué. Mucha gente piensa en DevOps como herramientas específicas como Chef o Docker, pero herramientas por sí solas no son DevOps. Lo que hace que las herramientas sean "DevOps" es la forma de su uso, no características fundamentales de las propias herramientas."

Gonzalez (2017): "DevOps es una filosofía más que un conjunto de herramientas o un procedimiento: tener a tus ingenieros expuestos a todo el ciclo de vida de tu producto requiere mucha disciplina, pero te da un enorme control sobre lo que se está construyendo. Si los ingenieros entienden el problema, lo resolverán; es lo que saben hacer."

Smith (2020): "DevOps es un conjunto de prácticas de desarrollo de software que combina una mentalidad de desarrollo de software con otras funciones de la organización.

DevOps pone énfasis en las responsabilidades compartidas entre todos los equipos a lo largo del ciclo de vida del desarrollo de software. Los límites del trabajo se suavizan, ya que los miembros del equipo de operaciones asumen tareas que tradicionalmente estaban más centradas en los desarrolladores, y los miembros del equipo de desarrollo hacen lo mismo. El término DevOps se asocia más comúnmente con el desarrollo (Dev) y las operaciones de TI (Ops), pero el enfoque puede extenderse también a otros grupos, incluyendo, entre otros, la seguridad (DevSecOps), el control de calidad, las operaciones de bases de datos y redes."

Gartner: "DevOps representa un cambio en la cultura de TI, centrándose en la rápida prestación de servicios de TI mediante la adopción de prácticas ágiles y ajustadas en el contexto de un enfoque orientado al sistema. DevOps hace hincapié en las personas (y la cultura), y busca mejorar la colaboración entre los equipos de operaciones y

de desarrollo. Las implementaciones de DevOps utilizan la tecnología, especialmente las herramientas de automatización que pueden aprovechar una infraestructura cada vez más programable y dinámica desde la perspectiva del ciclo de vida."

En resumen, DevOps se puede definir como una filosofía que está en constante evolución que agrupa buenas prácticas de industria para mejorar la entrega de productos de software a través de un cambio cultural que abarca personas, procesos y herramientas tecnológicas.

La Figura 2.4 presenta los pilares de la cultura DevOps en una organización; colaboración, procesos y herramientas. La conjunción de ellas permite alcanzar una transformación digital en DevOps en pro de mejorar el rendimiento en la entrega de productos de software en las organizaciones.

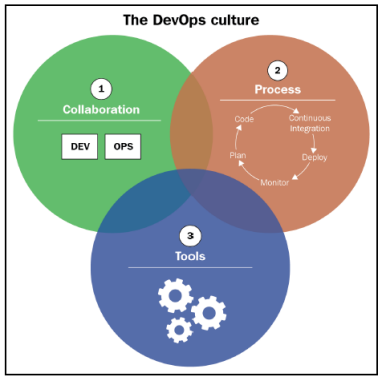

Figura 2.4: Cultura DevOps

Imagen tomada de (Verona et al., 2016)

2.1.3.2. Historia

El origen de la palabra DevOps y los primeros días del movimiento se centran en una persona que lideró la popularización de la idea inicial del concepto DevOps: *Patrick Debois*, un desarrollador de software y consultor con experiencia en muchos campos dentro de las TI que se sentía frustrado por la división entre los desarrolladores y el personal de operaciones.

Inspirado en la charla "*10 Deploys per Day: Dev and Ops Cooperation at Flickr*" dictada por *John Allspaw* y *Paul Hammond* en la conferencia Velocity de 2009, en la que describieron que la única manera de construir software viable era integrar operaciones y desarrollo a través de objetivos compartidos. Ese mismo año Patrick decidió entonces organizar un evento en Gante, Bélgica, llamado DevOpsDays y posteriormente el nombre se abrevió a DevOps.

En años siguientes el evento DevOpsDays se popularizó en el resto del mundo convirtiéndose en eventos muy concurridos popularizando el término DevOps para un movimiento cultural (Davis et al., 2016).

2.1.3.3. Prácticas DevOps

DevOps está compuesto por un conjunto de prácticas que implementan las etapas del ciclo de vida del desarrollo de software ágil para alcanzar mejoras en la entrega de productos de software en las organizaciones

La Figura 2.5 muestra las diferentes fases del ciclo de vida del desarrollo de software y cómo estas están relacionadas con las prácticas DevOps. Adicionalmente, la gráfica muestra la complementariedad que existe entre el desarrollo ágil y DevOps, donde el desarrollo ágil se encarga de las etapas iniciales como planeación, desarrollo y pruebas y DevOps se ocupa de las fases de integración y entrega del producto hacia los usuarios finales.

A continuación, se explican las tres fases principales en DevOps integración continua, despliegue continuo y entrega continua:

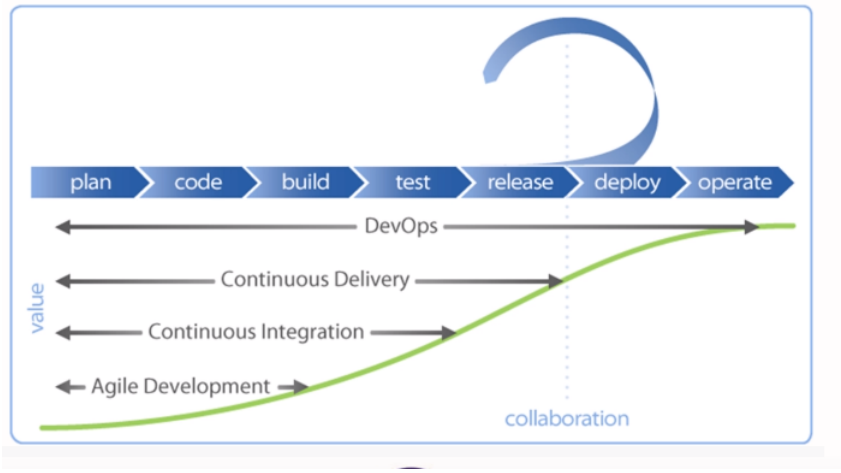

Figura 2.5: DevOps en el ciclo de vida del desarrollo de software
Imagen tomada del curso de DevOps - Introducción y Fundamentos no técnicos (https://www.udemy.com/course/devops-en-espanol/)

- **Integración Contínua (*CI - Continuous Integration*):**

Es la fase de desarrollo de software que permite integrar rápidamente el código recién desarrollado con el resto del código de la aplicación (al menos una vez al día) y con otros componentes, tras lo cual se ejecutan versiones y pruebas automáticas (Sharma, 2017).

En la Figura 2.6 se muestra un ejemplo de integración continua entre varios equipos y componentes.

- **Entrega continua (*CD - Continuous delivery*):**

Esta fase permite obtener productos listos para ser desplegados, está estrechamente relacionada con la *CI*, va más allá de simplemente asegurarse de que los cambios superen las diferentes fases de pruebas sin causar regresiones, la entrega continua

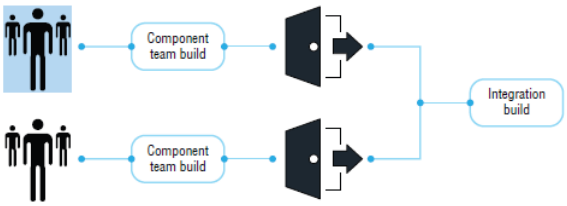

Figura 2.6: Integración continua

Imagen tomada de (Sharma, 2017)

significa que estos cambios ya están listos para ser desplegados de forma manual o parcialmente automatizada al entorno de producción (Coupland, 2021).

La Figura 2.7 presenta como la entrega continua es la fase final del ciclo de vida DevOps para entregar un producto o un cambio al cliente final de forma rápida y confiable.

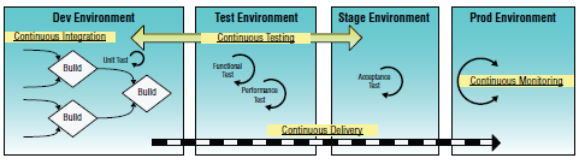

Figura 2.7: Entrega continua

Imagen tomada de (Sharma, 2017)

- **Despliegue continuo (*Continuous Deployment*):**

Esta fase se encarga del proceso de despliegue de los cambios en producción en donde los cambios en el código se compilan, prueban y preparan automáticamente para minimizar riesgo y los errores así como entregar los cambios lo más rápido posible al cliente (Davis et al., 2016).

El despliegue continuo es un paso más allá de la entrega continua porque el despliegue se hace sin intervención humana: una prueba fallida, en esta etapa, impedirá nuevas liberaciones a producción (Sharma, 2017).

La Figura 2.8 presenta de manera gráfica la diferencia entre entrega continua y despliegue continuo para DevOps.

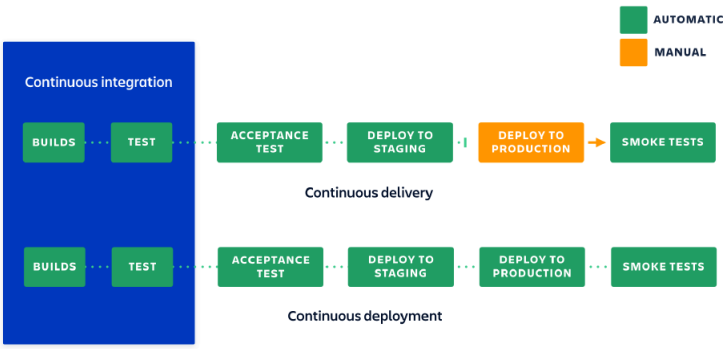

Figura 2.8: Diferencias entre integración continua, entrega y despliegue
Imagen tomada de (Coupland, 2021)

2.1.3.4. ¿Qué no es DevOps?

DevOps es solo herramientas y automatización
DevOps es más que un conjunto de herramientas implementadas e interconectadas que automatizan las fases del proceso de desarrollo, primero se debe enfocar la cultura visto cómo las personas, luego el proceso y después las herramientas. "Puedes instalar la última versión de Jenkins o apuntarte a CircleCI, pero si no tienes un conjunto de pruebas sólido, no sirve de nada" (Smith, 2020).
Devops sólo involucra a los desarrolladores y administradores de sistemas
En sus orígenes el DevOps nació como la interacción entre desarrolladores y operaciones, las ideas y conceptos DevOps van más allá e incluye a quien sea necesario para mejorar el proceso de entrega de software por lo que no hay una lista definitiva de qué equipos o personas participan, todo depende de las necesidades de cada organización (Davis et al., 2016).
Hay una "forma correcta" (o "incorrecta") de hacer Devops
No existe un modelo genérico o manual paso a paso que contenga las instrucciones necesarias para realizar una adopción de prácticas DevOps. "El hecho de que una empresa presente su estrategia de éxito en DevOps no significa que estos mismos procesos sean la forma correcta de hacer DevOps en todos los entornos" DevOps, debe adaptarse acorde a la situación inicial y tipo de empresa donde quiera aplicarse (Davis et al., 2016).
Devops sólo es relevante para los startups
El término DevOps es un término que no es exclusivo de grandes empresas tecnológicas, como Netflix, Facebook, LinkedIn o Google, o de organizaciones a la vanguardia en innovación, los principios de la cultura DevOps son abiertos y cualquier compañía los puede adoptar, algunas sin saberlo ya aplican alguna de las premisas, como por ejemplo Agile o Lean, entre otros (Davis et al., 2016).
Devops es un título laboral
EL "ingeniero DevOps" es un título de trabajo que cada vez es más frecuente en el mercado laboral, aún no se tiene un consenso claro en las organizaciones sobre el rol. Sus funciones giran en torno a un administrador de sistemas que sabe programar y a un desarrollador que conoce sobre administración de sistemas e infraestructura. En

las etapas tempranas de adopción DevOps se puede dar el caso en que los desarrolladores desplieguen el código y mantengan la infraestructura, pero a medida que la organización madura y crece, tiene sentido que las personas se especialicen más en sus funciones. "DevOps es, en esencia, un movimiento cultural, y sus ideas y principios deben utilizarse en toda la organización para que sean eficaces" (Davis et al., 2016).

2.1.3.5. Métricas de rendimiento

Forsgren et al. (2018) presenta 4 características para medir el rendimiento en la entrega de software que se han extendido y popularizado en la industria, Estas métricas son comunes en los reportes anuales State of DevOps de las compañías Puppet y el proyecto DORA.
Deployment frequency: Es la frecuencia con que la organización despliega el código en producción o lo libera para los usuarios finales
Lead time: Es el tiempo que se tarda en pasar de un commit de código a un código que se ejecuta con éxito en producción (cuánto tiempo dedica el equipo a implementar, testear y entregar código a los usuarios.)
Change Fail Rate: Es el porcentaje de cambios de código que requieren correcciones en vivo u otros ajustes que afecten el rendimiento y degraden el servicio después del despliegue en producción (se mide como cambios fallidos vs cambios exitosos)
Mean Time to Restore (MTTR): Es el tiempo medio que se necesita para restablecer un servicio cuando ocurre un fallo en producción.

La Figura 2.9 tomada del reporte Accelerate State of DevOps 2021(DORA's, 2021), presenta la tendencia de industria frente a las métricas DevOps para las cuatro medidas de rendimiento, donde el nivel elite se caracteriza por tener tiempos de respuesta de menos de una hora, ejemplo frecuencias de despliegue por día bajo demanda, tiempo de espera de despliegue de cambios de menos de una hora, tiempo de restablecimiento ante fallos de menos de una hora y por último una tasa de errores en producción entre el 0 y 15 %.

La Figura 2.10 muestra la evolución de la industria en los últimos tres años sobre las cuatro categorías de madurez. En esta imagen se evidencia la mejora en cifras porcentuales de las categorías Elite y High, lo que presenta una tendencia de mejora constante de las compañías por potenciar sus capacidades y métricas de entrega de software. Se resalta el incremento del 19 % de la categoría Elite en solo 3 años (2018 y 2021).

2.1.3.6. Pilares y Objetivos DevOps

Coupland (2021) indica que la adopción DevOps en una organización difiere de muchos elementos y condiciones que lo hacen particular. Desde el punto de vista de impulsores del negocio los objetivos DevOps más comunes que ayudan a determinar las métricas y capacidades en que la organización se debe centrar son:

- **Deployment frequency:** Es la frecuencia con que la organización despliega el código en producción o lo libera para los usuarios finales

Software delivery performance metric	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per week and once per month	Between once per month and once every 6 months	Fewer than once per six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one hour	Between one day and one week	Between one month and six months	More than six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day	Between one day and one week	More than six months
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0%-15%	16%-30%	16%-30%	16%-30%

Figura 2.9: Escala de rendimiento por métricas
Imagen tomada de: (DORA's, 2021)

- **Faster time to market:** Reducir la cantidad de tiempo que se necesita desde el inicio de la idea hasta el lanzamiento del producto
- **Lower failure rates:** Lograr menores tasas de fallos gracias a que los equipos (multifuncionales) colaboran entre sí y se comunican mejor
- **Shorter lead times:** Reducir el tiempo que transcurre entre el inicio del trabajo en una historia de usuario y la publicación de la misma en producción.

  Puede tratarse de cualquier cosa, desde la planificación, en la que se obtienen los requisitos de forma más eficaz, hasta la construcción de la infraestructura más rápida que antes.
- **Improved recovery time:** El tiempo de indisponibilidad representa una pérdida de ingresos y un daño a la reputación de su organización, por lo que es muy importante reducir ese nivel de indisponibilidad.

2.1.3.7. Estrategias de adopción DevOps

Basado en Sharma (2017), la Figura 2.11 presenta las estrategias tácticas de mejora más comunes en las organizaciones al momento de iniciar un camino de adopción DevOps:

1. **Mejora de los procesos:** ¿cómo hacer que los procesos sean ágiles y eficientes mediante la eliminación de los residuos?
2. **Herramientas para la automatización:** ¿cómo automatizar los procesos de mejora con herramientas para hacerlos repetibles y escalables, y para reducir los errores?

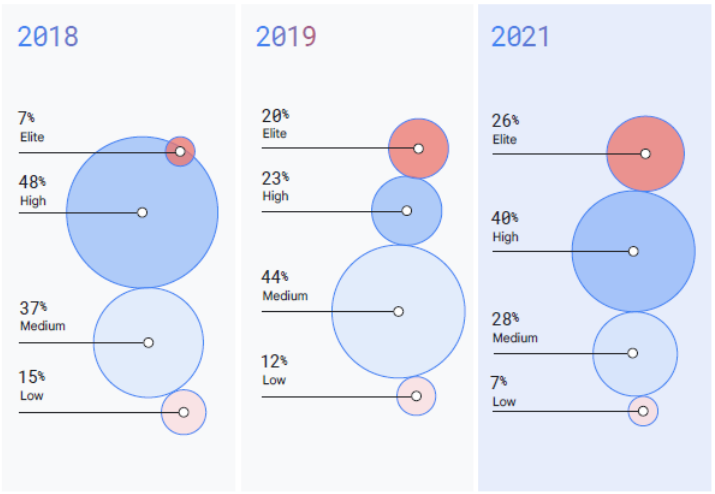

Figura 2.10: Evolución de la industria en escala de rendimiento
Imagen tomada de: DORA's (2021)

3. **Plataforma y entornos:** ¿cómo hacer que las plataformas y los entornos para la canalización de la entrega de aplicaciones, desde los requisitos hasta la producción, sean resistentes, elásticos, escalables y capaces de gestionar las configuraciones?

4. **Cultura:** ¿cómo fomentar una cultura de confianza, comunicación y colaboración?

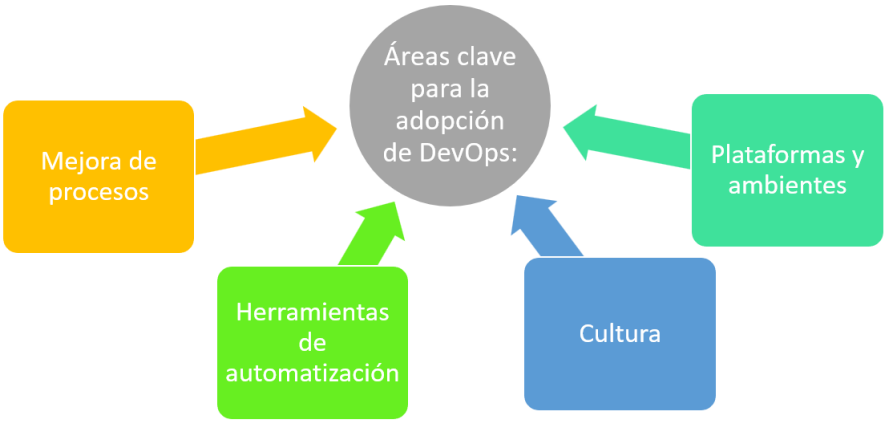

Figura 2.11: Áreas clave para la adopción DevOps
Fuente elaboración propia basado en (Sharma, 2017)

2.2. Estado del arte

2.2.1. Trabajos relacionados

En la búsqueda de literatura sobre estos temas existen miles de artículos que enfocan DevOps y muchos más en calidad de software, lo que se busca con este análisis es identificar los documentos más relevantes que unan estos dos conceptos en pro de asegurar que se entreguen soluciones funcionales en periodos de tiempo moderados y que cumplan con buenas prácticas de calidad, documentación y mantenibilidad.

Ordóñez and Buchelli (2020) presenta un conjunto de lineamientos utilizados en la implementación del modelo CALMS para DevOps en miPymes de desarrollo de software en el contexto surcolombiano. Describen los aspectos en términos de cambios organizacionales a tener en cuenta para alcanzar una mejora en el rendimiento en la entrega de software, por ejemplo, aspectos técnicos relacionados a las herramientas utilizadas para la creación de un pipeline de integración y despliegue continuo. En el documento hacen referencia a los pasos de adopción DevOps que proponen los principios CALMS lo cual tiene una relación directa con el presente documento en términos del uso de un modelo de evaluación de madurez DevOps en una organización para determinar los pasos para iniciar su adopción.

Los autores realizaron la medición de prácticas empresariales a través una encuesta que se basa en preguntas directas sobre conceptos DevOps lo cual difiere del modelo propuesto en este trabajo que se enfoca en preguntas indirectas pensado en el público que se está iniciando en el mundo DevOps, además, el modelo de encuesta planteado no presenta un plan de acción concreto para cada escenario de respuesta de la encuesta.

Por último, los autores presentan los resultados de aplicar dicho modelo en un grupo de empresas mipymes de desarrollo de software, obteniendo grandes resultados en términos de despliegues, pasando de un despliegue semanal a un despliegue diario. Como conclusión adicional los autores indican que "la adopción de DevOps debe ser gradual, iterativa e incremental"

Mishra and Otaiwi (2020) presenta un mapeo sistemático del impacto de DevOps en la calidad del software, analizando las implicaciones de las características de DevOps en la calidad del software, proporcionando una mejor comprensión de la estrecha relación que se genera entre DevOps y la calidad, concluyendo que DevOps contribuye positivamente en el aseguramiento de la calidad de software.

Perera et al. (2017) realiza un estudio sobre el impacto de la práctica DevOps en la calidad del software basándose en explorar las prácticas actuales de DevOps en la industria y los factores clave que llevaron a esta conclusión. Los autores indican que se basaron en un sondeo a partir de la experiencia de expertos para determinar cómo mejorar la calidad de software a partir de la práctica de DevOps. Esta medición fue expresada por los autores a partir de análisis de regresión múltiple generando una ecuación para la medición cualitativa de la calidad de software a partir de los siguientes pilares: Cultura, Automatización, Medición, Compartir.

Cruz and Albuquerque (2018) presentan una experiencia de despliegue de DevOps para aplicaciones Legacy o heredadas. En este documento se presentaron las herra-

mientas y las modificaciones necesarias para que un proyecto Legacy se integre en un flujo DevOps basándose en los siguientes subprocesos

- Integración continua.
- Automatización de la implementación
- Automatización de pruebas
- Automatización del monitoreo
- Automatización de la infraestructura

Asociación Colombiana de Ingenieros de Sistemas (2018) hace referencia a un estudio a 1.200 ejecutivos de TI sobre el uso de *Agile* y DevOps en la transformación digital en el que presenta las características de los "Maestros en Agilidad" (18 % de los encuestados), de las cuales forman parte las organizaciones más cercanas a la adopción total y que realizan las principales, o casi todas, las acciones para hacer de Agile y DevOps una parte esencial de sus operaciones diarias. Estos "Maestros en Agilidad" también están más propensos a utilizar prácticas del enfoque Agile en otras funciones de la empresa, por lo que el aumento del 60 % en los ingresos y ganancias y la probabilidad de expandir sus negocios es 2,4 veces mayor con una tasa de aumento por encima del 20 % , no es sólo una coincidencia.
de Feijter et al. (2018) presenta un modelo de madurez DevOps que busca definir un camino de crecimiento para las organizaciones productoras de software a partir de una herramienta de evaluación de DevOps que el autor plantea para realizar autoevaluaciones del estado de madurez DevOps.

La Figura 2.12 muestra la estructuración del modelo, compuesto por 3 perspectivas definidas por el autor que agrupan un conjunto de áreas de interés que representan las prácticas DevOps evaluadas en una escala de medición entre 1 y 10, donde el rango 1 a 5 representa estados de no adopción DevOps, los niveles entre 5 y 10 presentan un grado de madurez y adopción DevOps dejando los niveles 5 y 6 como etapas de transición.

Este modelo de madurez tiene similitudes con el modelo propuesto ya que comparten la forma de plantear prácticas DevOps y un grupo de categorías de clasificación. Los modelos difieren en la escala de calificación ya que el autor propone una escala de respuestas numéricas entre 1 y 10 vs a diferencia de la escala de calificación propuesto en este documento que va entre 1 y 5, además de que este modelo utiliza un sistema de priorización y peso a las activades evaluadas.

2.2.2. Modelos de Madurez DevOps

En la literatura no existe un método único o definitivo para implementar DevOps, esto se da porque cada organización es diferente y particular, en donde la adopción DevOps se da a partir de analizar y medir la organización en términos de capacidades particulares y modelos de referencia:

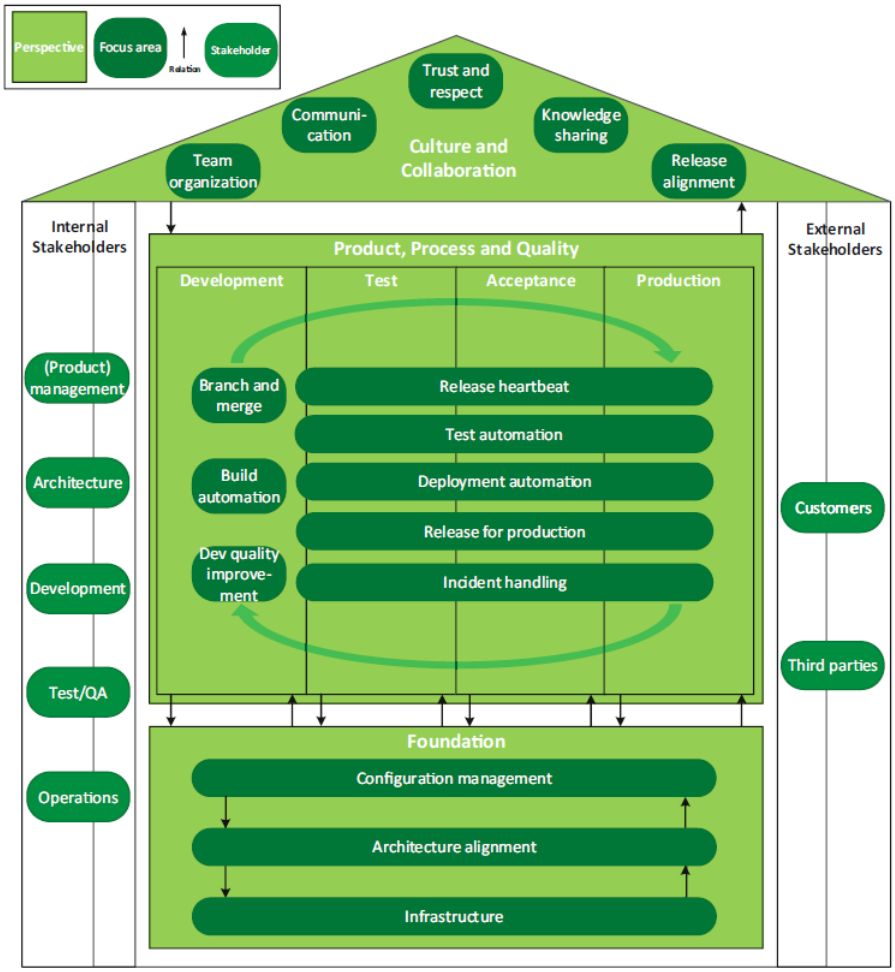

Figura 2.12: Modelo de competencias – propuesto por Rico de Feijte
Imagen tomada de (de Feijter et al., 2018)

2.2.2.1. Accelerate

El libro Accelerate (Forsgren et al., 2018) presenta el resultado de analizar 23000 respuestas de encuestas de organizaciones y profesionales que trabajaban con DevOps recopilados en 4 años de investigación entre 2014 y 2017, hallando 24 capacidades, agrupadas en 5 categorías dimensionales que a nivel estadístico impulsan las mejoras en el rendimiento de la entrega de software y el desempeño organizacional de la siguiente forma: (ver Figura 2.13 que describe la relación entre los diferentes componentes).

Dimensiones o Categorías del modelo

- Entrega continua
- Arquitectura
- Producto y Procesos

- Gestión Lean y Monitorización
- Cultura

Capacidades y categorías
Entrega continua

1. Utilizar control de versiones para todos los artefactos de producción
2. Automatizar el proceso de despliegue
3. Implementar integración continua
4. Utilizar métodos de desarrollo "Trunk-Based"
5. Implementar pruebas automáticas
6. Implementar gestión de los datos de prueba
7. Incorporar los aspectos de seguridad en las fases de diseño y pruebas (shift left de la seguridad)
8. Implementar entrega continua de software

Arquitectura

1. Utilizar arquitecturas de bajo acoplamiento
2. Permitir a los equipos definir sus propias arquitecturas

Producto y Procesos

1. Recolectar e implementar las recomendaciones (feedback) de los clientes
2. Hacer visible el flujo de trabajo durante todo el ciclo
3. Trabajar con lotes o cantidades pequeñas
4. Fomentar y habilitar al equipo para realizar experimentos

Gestión Lean y Monitorización

1. Tener un proceso liviano para la aprobación de cambios
2. Monitorizar las aplicaciones y la infraestructura para tomar decisiones de negocio
3. Monitorizar la salud de los servicios y aplicaciones proactivamente
4. Administrar y establecer límites para el trabajo en progreso (Work-In-Progress)
5. Visualizar el trabajo para monitorizar la calidad y comunicarlo al equipo

Cultura

1. Implementar una cultura tipo generativa
2. Animar y apoyar al equipo a aprender
3. Apoyar y facilitar la colaboración entre los equipos
4. Proporcionar recursos y herramientas para que el trabajo sea significativo para los individuos
5. Apoyar el liderazgo transformacional

Los conceptos listados se profundizarán en mayor detalle en el siguiente capítulo donde se describe el modelo de evaluación propuesto que se fundamentó en estos conceptos.

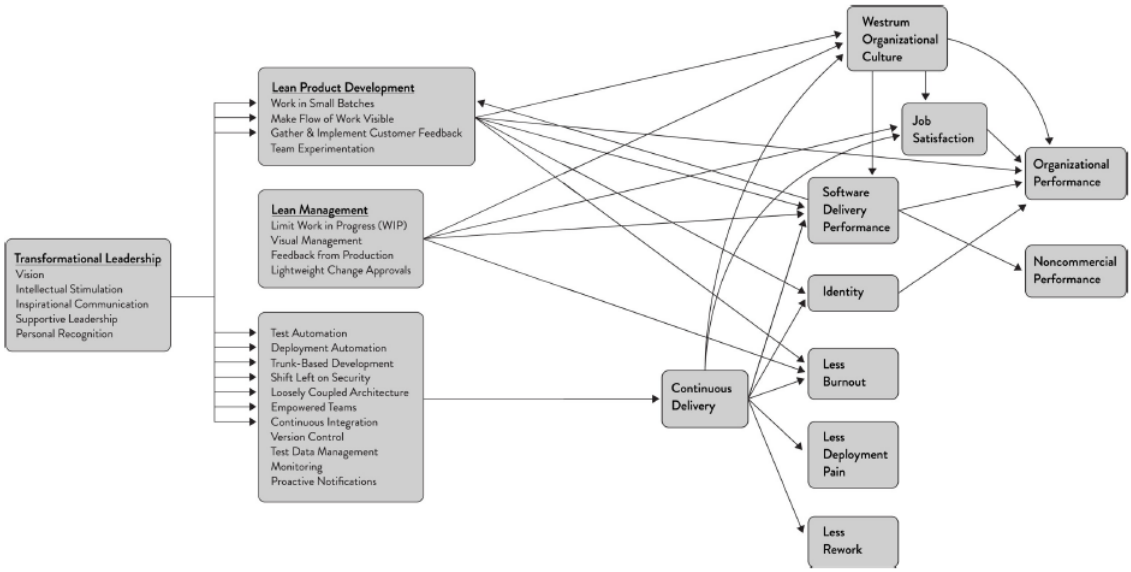

Figura 2.13: Modelo general de la investigación propuesto por Forsgren et al. (2018)
Imagen tomada del libro: Accelerate: The Science of Lean Software and DevOps Building and Scaling High Performing Technology Organizations (Forsgren et al., 2018)

2.2.2.2. Modelo CALMS

CALMS, es un acrónimo que significa Cultura, Automatización, Lean, Medición y Sharing (Compartir), una serie de puntos clave que resultan particularmente útiles para analizar la estructura de una organización (Devopsgroup, Atlassian, b).

Es un modelo fundacional para DevOps y un punto de partida útil para entender DevOps como proceso cultural, la Figura 2.14 representa como la cultura es base para alcanzar una transformación DevOps.

A continuación, se describen cada una de los componentes del modelo CALMS.

Cultura: Este punto es crucial en una organización. El conocimiento de cómo funciona una organización y conseguir involucrar a sus actores para generar valor es especialmente importante para lograr que todos los actores se involucren y no solo los departamentos técnicos.

La cultura en una organización es el conjunto de saberes, creencias y pautas de conducta, incluyendo los medios materiales que usan sus miembros para comunicarse entre sí, y resolver sus necesidades.

Automatización: Liberar a los equipos de tareas repetitivas o redundantes, logra que los diferentes equipos puedan concentrarse en tareas más productivas.

En este sentido, DevOps propone la aplicación de herramientas que automatizan gran parte de los procesos existentes a lo largo del ciclo de vida como:

- Automatización de las revisiones de código.
- Automatización de la construcción y del despliegue.
- Automatización de los controles de calidad.
- Automatización del aprovisionamiento de la infraestructura.

Lean: El desarrollo de software Lean tiene 7 principios, y el primero es "eliminar los desperdicios", es decir, todo aquello redundante y que supone una pérdida de tiempo ya que no aporta valor. Estos son:

- Código o funcionalidad innecesaria en las aplicaciones.
- Sobrepasar la capacidad del equipo, o empezar más de lo que puede terminarse.
- Requisitos poco claros o con cambios constantes.
- Burocracia.
- Comunicación lenta o inefectiva
- Trabajo parcialmente terminado
- Defectos o problemas de calidad
- Cambio de tareas

Medición (Metrics) Lo que puede medirse puede gestionarse. Dicho de otra manera, lo que se mide se puede gestionar de forma eficaz, y DevOps también pone el foco en esta parte.

El uso de métricas da visibilidad sobre el estado real de un proyecto de forma que los responsables de los equipos, y sus miembros coordinen acciones coherentes con los objetivos corporativos, alineando la visión y la estrategia de la organización.

La medición es el combustible que alimenta el motor DevOps que se centra en la mejora continua, basada en el "empirismo", y por tanto en la experiencia y los datos obtenidos.

Compartir (Share) La creación de valor surge de la colaboración, del aprovechamiento de experiencias y conocimientos multidisciplinares del equipo. Gracias a compartir conocimiento los integrantes del equipo, y de otras áreas, pueden tener acceso al mismo, de modo que formarán parte del modelo de referencia para un proceso iterativo de aprendizaje y mejora continua, optimizando iterativamente los procesos de la organización.

Compartir también implica invertir en políticas de desarrollo de talento e innovación, que faciliten o permitan cambios dentro de la cultura de la organización.

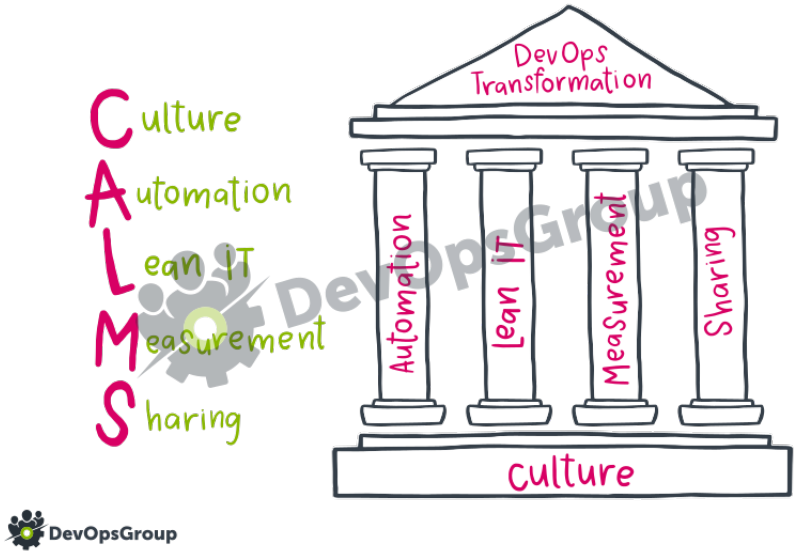

Figura 2.14: Modelo CALM para DevOps
Imagen tomada de (Devopsgroup)

2.2.2.3. OWASP DevSecOps Maturity Model

El Modelo de Madurez DevSecOps (DSOMM), se enfoca en analizar aspectos de seguridad que muchos de los modelos usados en el mercado descuidan, creando un modelo enfocado en mejorar la seguridad.

OWASP permite complementar las estrategias DevOps en términos de fortalecer y brindar un espacio de mayor relevancia a las medidas de seguridad que se aplican cuando se utilizan estrategias DevOps ya que la evidencia práctica indica que estos aspectos se suelen descuidar o al menos no se tienen suficientemente en cuenta durante las fases de ciclo de vida DevOps (OWASP Foundation).

Los niveles de implementación en DSOMM son:

- Nivel 1: Comprensión básica de las prácticas de seguridad
- Nivel 2: Adopción de prácticas básicas de seguridad
- Nivel 3: Adopción elevada de prácticas de seguridad

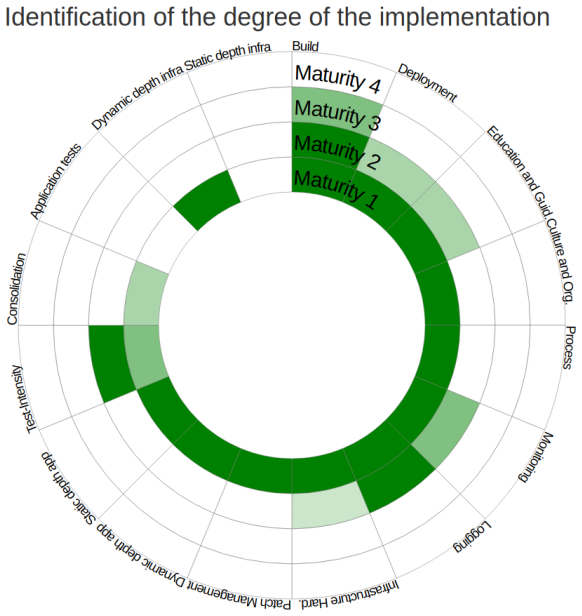

Figura 2.15: Modelo de madurez Devsecops de OWASP
Imagen tomada de (OWASP Foundation)

- Nivel 4: Implantación avanzada de prácticas de seguridad a escala

Dimensiones de DevOps

- Construcción y despliegue
- Cultura y organización
- Recogida de información
- Endurecimiento
- Prueba y verificación

2.2.3. Modelos de medición de DevOps

2.2.3.1. Devops Health Radar Assessment

Es un modelo de evaluación de las capacidades DevOps desarrollado por la compañía *AgilityHealth* para para los programas y/o equipos de entrega de tecnología con el fin de medir y visualizar su estado de madurez DevOps e identificar las áreas clave de crecimiento y desarrollar un plan de acción de crecimiento concreto(AgilityHealth).

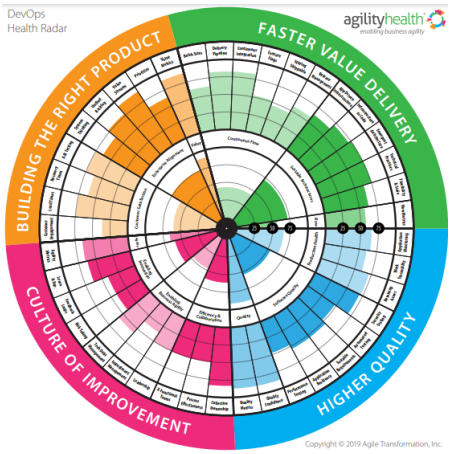

Figura 2.16: Modelo de evaluación de la salud de DevOps - DevOps Health Radar
Imagen tomada de https://agilityhealthradar.com/wp-content/uploads/2018/09/DevOps-Health-Radar_Web.pdf

2.2.3.2. Dora Assessment

El proyecto DevOps Research and Assessment (DORA) presenta un modelo de evaluación corto que se enfoca en cubrir cinco preguntas con opciones de respuesta cerrados en torno a cómo la organización aborda las cuatro métricas de rendimiento (Google):

- Lead time
- Deploy frequency
- Time to restore
- Change fail percentage

Este chequeo rápido (https://www.devops-research.com/quickcheck.html) permite tener una visión general del rendimiento de la organización, pero no permite entrar a evaluar y medir cómo opera y funcionan los aspectos técnicos, culturales y de procesos que permite que una organización alcance los tiempos definidos.

La Figura 2.17 presenta un ejemplo del resultado que entrega la plataforma luego de realizar una evaluación

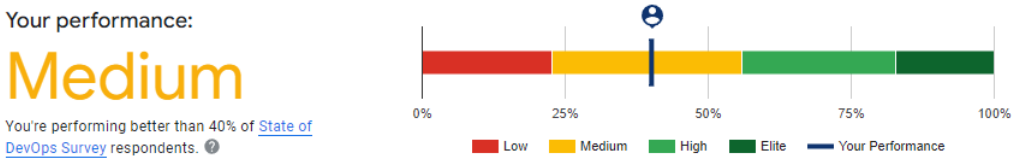

Figura 2.17: Ejemplo resultado evaluación DORA
Imagen tomada de (DORA's, u)

Algo a destacar del proyecto DORA, es que los autores del libro guía y del modelo de referencia usado en este documento, participaron del proyecto por lo que las capacidades propuestas en esta investigación están basadas en el mismo estudio y comparten muchos aspectos. Adicionalmente en su portal oficial describen cómo implementar, mejorar y medir cada una de las capacidades de la investigación.

2.3. Antecedentes

2.3.1. Alianza Bioveristy - CIAT

La Alianza entre Bioversity International y el Centro Internacional de Agricultura Tropical, en adelante CIAT, es unión de fuerzas entre dos centros de investigación que en conjunto suman más de 100 años de experiencia aplicando ciencia para abordar retos mundiales. Esta alianza se creó para enfrentar las crisis de cambio climático, pérdida de la biodiversidad, degradación ambiental y malnutrición, y para maximizar las oportunidades combinando nuestras fortalezas en puntos claves en el sistema alimentario. La alianza es parte del CGIAR, el mayor consorcio mundial en investigación e innovación agrícola para un futuro sin hambre, dedicado a reducir la pobreza, contribuir a la seguridad alimentaria y nutricional y mejorar los recursos naturales.

La alianza oficializó su operación conjunta en el 2020 con presencia a nivel mundial representada en 4 sedes distribuidas en 4 continentes:

- Oficina Principal- Roma, Italia
- Sede Regional para las Américas - Palmira, Colombia
- Sede Regional para África - Nairobi, Kenia
- Sede Regional para Asia - Hanoi, Vietnam

El manejo de los procesos y recursos informáticos de la organización está bajo el manejo del área de integración tecnológica cuya principal responsabilidad es apoyar la operación de la organización a través de la prestación y soporte de servicios tecnológicos en cada una de las sedes regionales. El área tiene alcance y cobertura global concentrando la mayor parte de su recurso humano en la oficina principal (antiguo Bioversity) y en la sede de las Américas (antiguo CIAT).

La Figura 2.18 presenta la distribución del área de Integración Tecnológica y sus cinco subáreas especializadas:

El departamento de integración tecnológica cubre todo el soporte operativo de la organización a nivel global, a nivel de desarrollo de software las áreas involucradas son:

- Integración de sistemas, responsable por el funcionamiento y operación de las soluciones y servicios relacionados con la operación de la compañía. El área está compuesta por 10 personas de los cuales 6 están directamente relacionados con

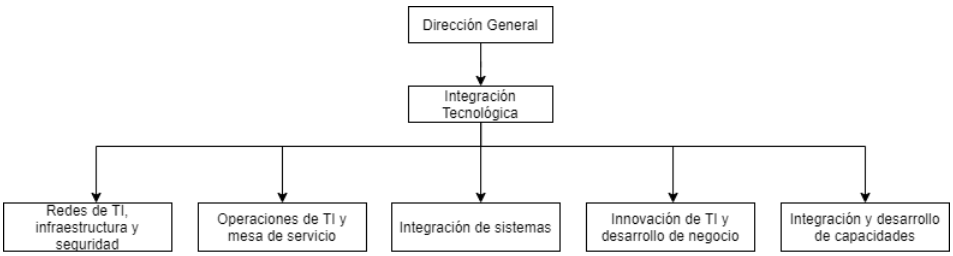

Figura 2.18: Estructura Organizacional del área de integración tecnológica Alianza Bioversity -CIAT

desarrollo de software así: 3 desarrolladores, 1 administrador de bases de datos, 1 jefe de desarrollo y el gerente del área, el resto del área tienen dedicación exclusiva a temas del ERP.

- Redes de TI, infraestructura y seguridad, encargados de la gestión de la infraestructura física de red y operación de los servidores y servicios informáticos de la compañía. En esta área solo hay 3 personas directamente involucradas con los procesos del ciclo de vida de desarrollo y entrega de productos de software. Su participación se centra en tareas particulares como administración de servidores, seguridad de red y servicios y la gerencia del área.

- Innovación de TI y desarrollo de negocio, es otra área donde se hace desarrollo de software, focalizados en soluciones de gestión para programas de investigación a nivel del consorcio CGIAR, el área está compuesta de 10 personas de los cuales 6 están dedicados a temas asociados a desarrollo de software.

A nivel de infraestructura la compañía cuenta con un centro de datos propio administrado por el área de redes de TI, allí están alojados mediante servidores virtualizados los diferentes ambientes de servidores y servicios utilizados en las diferentes etapas del desarrollo de software.

El área de integración de sistemas está compuesta por un total de 11 personas, un director global, un coordinador regional y 9 analistas de sistemas, dentro del equipo solo 3 analistas y el coordinador tienen relación directa con desarrollo de aplicaciones, el resto del equipo cumple tareas funcionales con el ERP. El equipo de analistas de desarrollo se encarga de todas las actividades relacionadas con el mantenimiento, soporte, implementación de cambios y nuevos desarrollos internos que sean solicitados por las áreas de operación administrativa.

2.3.2. CIAT y DevOps

El CIAT, es una empresa de investigación en agricultura, que centra sus objetivos de negocio en la investigación y no se especializada en temas relacionados al desarrollo de software, a pesar de que en la actual modernidad demanda productos de software cada día más especializados e integrados para apoyar tanto los procesos científicos como la operación administrativa y operativa. La compañía ha enfocado sus esfuerzos

en mantener la operación y desarrollar soluciones a medida para sus procesos particulares. El término DevOps no ha sido ajeno a la organización, a finales del 2020 los equipos integración de sistemas y redes recurrieron a consultores externos para la implementación de un piloto de adopción técnica de DevOps que no se popularizó de la forma esperada dentro de la misma área.

Uno de los principales factores para el estancamiento del piloto de adopción DevOps, fue que el proceso solo se centró en la parte técnica, instalando herramientas e implementando mecanismos de integración entre ellas y las herramientas existentes, dejando de lado el análisis del estado los procesos de desarrollo y entrega de software, la capacidad de los equipos, las políticas y metodologías de aseguramiento de calidad.

La Figura 2.19 presenta el grupo de aplicaciones que contaba la organización antes y las herramientas que se instalaron e integraron después de adoptar el piloto DevOps, en este punto se resalta que aprovechando que la compañía cuenta con un centro de datos propios, todos los servicios mencionados en la gráfica fueron instalados localmente en servidores virtuales propios para uso interno.

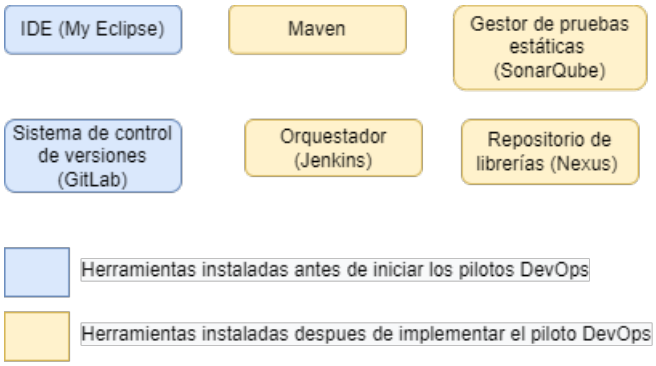

Figura 2.19: Herramientas DevOps instaladas en CIAT

El piloto de integración DevOps implementado, contempló pasos básicos de integración continua como instalar un orquestador, construir la aplicación a través de la herramienta Maven tomando librerías de repositorio local (Nexus), ejecutar un análisis de código estático y desplegar el producto final generado en un servidor de aplicaciones de pruebas (Ver Figura 2.20).

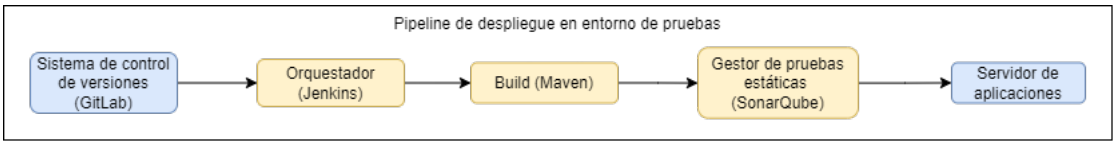

Figura 2.20: Pipeline de despliegue implementado en CIAT como piloto DevOps

2.4. Resumen del capítulo

El principal objetivo de este capítulo fue presentar las diferentes metodologías de desarrollo de software que dieron origen y que abrieron camino para que el concep-

to DevOps naciera como un movimiento cultural para la entrega de productos de software.

DevOps no tiene una única definición generalizada, todas buscan expresar una perspectiva que refleje el movimiento cultural que busca entregar productos de software a los clientes en el menor tiempo posible, a través de la implementación de buenas prácticas en procesos de desarrollo, entrega y calidad de software dando prevalencia a las personas, procesos sobre las herramientas.

En el mercado existen un número creciente de herramientas que apoyan cada una de las fases del proceso de entrega DevOps brindando un abanico gigante de opciones para automatizar los procesos de desarrollo de software.

Por último, se describió la composición del departamento informático y en especial al área de desarrollo software administrativo de la compañía CIAT, en donde se describió cómo se abordó el primer acercamiento de la compañía a los conceptos, buenas prácticas y herramientas DevOps que se hizo por medio de un piloto técnico que se enfocó a la instalación e integración de servicios que complementan la infraestructura ya existente, dejando de lado el análisis y validación de procesos de la cadena de valor para la entrega de soluciones de software. Lo que condujo al estancamiento de la adopción incremental de DevOps en el área.

En el siguiente capítulo se describe como se definió, estructuró e implementó una propuesta de un artefacto de evaluación de capacidades DevOps para una organización, basada en un modelo dimensional que agrupa capacidades culturales, técnicas y de procesos para mejorar el rendimiento en la entrega de productos de software a través de automatización de procesos.

Capítulo 3

Propuesta del artefacto de evaluación

Adoptar una transformación digital en DevOps, es un proceso que no se puede llevar a cabo siguiendo una hoja de ruta genérica, por el contrario, cada organización debe pensar y definir su propio proceso acorde a cómo funciona, opera y gestiona el ciclo de vida de las aplicaciones, iniciando con la identificación de debilidades, cuellos de botella y demás elementos que sean factor de retraso en el proceso de desarrollo de software. Con estos elementos base identificados, la organización tiene criterios de decisión y análisis para definir la estrategia de adopción que más le convenga a la organización para lograr el objetivo de mejorar el rendimiento en la entrega de productos de software.

Para facilitar la evaluación del estado de los procesos de la organización, se diseñó un artefacto de medición de capacidades enfocado en el segmento de profesionales TI con pocos conocimientos previos de las prácticas DevOps. El marco de evaluación se representa a través de un cuestionario diseñado en forma de preguntas indirectas sobre una práctica puntual desde el punto de vista de las características y funcionalidades del desarrollo de software y su ciclo de vida. Con este enfoque se buscó minimizar sesgos, confusiones y malas interpretaciones de las características que en su mayoría no son fáciles de medir, por lo que son abordados de forma abstracta, logrando una aproximación a partir de unidades definidas o recomendaciones de industria.

El marco de evaluación diseñado se basa en un modelo de referencia en donde se identifican los elementos claves para mejorar el rendimiento en la entrega de software a través de prácticas DevOps, a partir de estas capacidades se diseñó un cuestionario de preguntas cerradas que utiliza una escala de respuestas predefinida al que se asoció un plan de acción con el propósito de escalar el nivel de adopción de la capacidad.

3.1. Modelo de capacidades

Se diseñó un artefacto de medición de competencias DevOps, denominado *modelo de capacidades*, que se fundamenta en el resultado de la investigación del libro Accelerate (Forsgren et al., 2018), donde identificaron 24 capacidades y 5 dimensiones que abarcan ámbitos técnicos, gestión de procesos y cultura que a nivel estadístico impulsan mejoras en el rendimiento en la entrega de software en una compañía.

El *modelo de capacidades* propuesto complementa las capacidades y dimensiones propuestos en el modelo de referencia descrito en (Forsgren et al., 2018) con 3 elementos: un grupo de preguntas, un conjunto de opciones de respuestas a cada pregunta que se representa a través de una escala de calificación y un plan de acción para resolver cada opción de respuesta (Cada uno de estos elementos serán descritos en secciones posteriores de este documento). La Figura 3.1 muestra la relación de cardinalidad entre las preguntas del *modelo de capacidades* y el marco de referencia, resaltando como el modelo de referencia agrupa las capacidades en categorías dimensionales y el modelo propuesto presenta la relación entre cada capacidad y las preguntas propuestas. Esta estructura permite futuros escalamientos del modelo de evaluación ya que una capacidad puede estar relacionada en una o muchas preguntas. Adicionalmente, cada pregunta del *modelo de capacidades* propuesto se complementa con una justificación del enfoque o propósito de lo que se quiere evaluar.

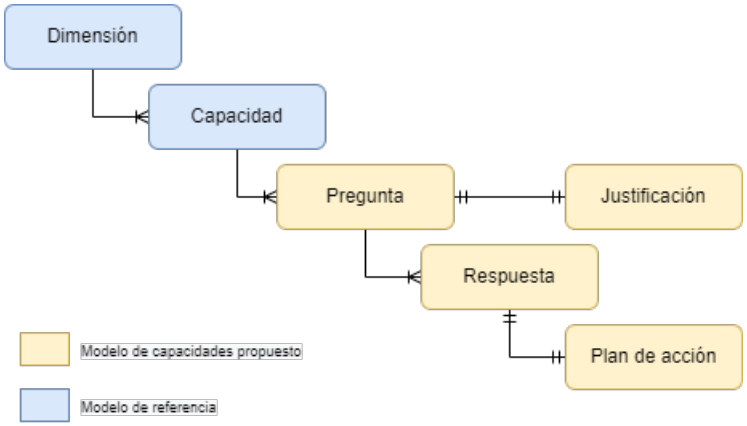

Figura 3.1: Estructura del *modelo de capacidades* propuesto
Fuente: Elaboración propia

En cifras totales, el *modelo de capacidades* propuesto está compuesto por los siguientes elementos:

- 5 dimensiones (modelo de referencia)
- 24 capacidades (modelo de referencia)
- 25 preguntas (cada pregunta representa una única capacidad)
- 125 opciones de respuesta (5 opciones por cada pregunta)
- 125 planes de acción (un plan de acción por cada opción de respuesta)

La Figura 3.2 presenta un ejemplo de la estructura del *modelo de capacidades* propuesto, cada pregunta está acompañada de una justificación del porqué se aborda la capacidad, las 5 opciones de respuesta en la escala definida y un plan de acción diseñado para resolver cada opción de respuesta. (El modelo completo se detalla en profundidad en el anexo Modelo de capacidades para la evaluación DevOps)

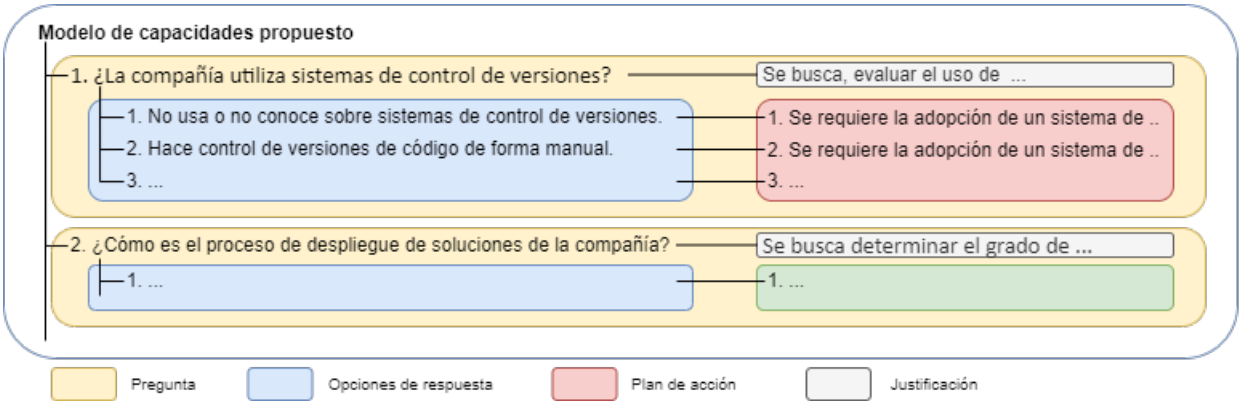

Figura 3.2: Ejemplo estructura de preguntas del *modelo de capacidades*
Fuente: Elaboración propia

Los elementos del modelo de capacidades como las preguntas del cuestionario, las opciones de respuestas y el plan de acción fueron diseñadas en base a un análisis e investigación y experiencias de diferentes fuentes académicas, y empresariales, ejemplo artículos, libros, y programas de investigación de empresas referentes de la industria del software a nivel mundial, entre otras, con el propósito de evitar caer en sesgos, flexibilidad, positivismo y puntos de vista personales al momento de su redacción y orientación.

Con el fin de brindar una alternativa de respuesta cuando el encuestado no tiene el suficiente conocimiento o experticia sobre el tema preguntado, al momento de la implementación se incluyeron en el formulario de encuesta dos opciones de respuesta: No aplica y No sabe /No responde por cada pregunta.

En las siguientes secciones se describirán cada uno de los elementos que componen el *modelo de capacidades* propuesto: dimensiones, capacidades, preguntas, opciones de respuesta y planes de acción.

3.1.1. Dimensión

El modelo de referencia propone un conjunto de categorías desde los puntos de vista técnico, procesos, gestión de software y cultura que agrupan un conjunto de características clave que impulsan las mejoras en el rendimiento en la entrega de software en las organizaciones (Forsgren et al., 2018).

Las dimensiones del modelo de referencia son (Forsgren et al., 2018):

Entrega continua: Agrupa las capacidades técnicas para medir los componentes de integración de procesos que permiten alcanzar la entrega frecuente y confiable de software. Estas capacidades permiten pasar cambios a producción o uso de los usuarios finales de manera segura, rápida y sostenible.

Arquitectura: Agrupa las capacidades de arquitectura de software, hardware y servicios que permiten reducir el tiempo de despliegue y mejorar la estabilidad del proceso de entrega de soluciones de software.

Producto y Procesos: Agrupa las capacidades de desarrollo de productos Lean, mediante el enfoque y análisis hacia el cliente, gestión de prototipos tempranos y la división del trabajo

Gestión Lean y Monitorización: Agrupa las capacidades de gestión de proyectos del movimiento Lean, su aplicación a la entrega de software y el modo en que impulsa el rendimiento en la entrega de software.

Cultura: Agrupa las capacidades del manejo de personas desde el punto de vista de satisfacción, identidad y compromiso de los empleados. Además de la influencia de la cultura al implementar prácticas de DevOps.

La Figura 3.3 muestra el modelo completo planteado por los autores donde se agrupan las dimensiones en dos categorías que representan su enfoque particular: aspectos técnicos y componentes de gestión que apalancan la entrega de software en las organizaciones.

3.1.2. Capacidad

Las capacidades son los elementos claves para mejorar la entrega de software en una organización, identificados en el modelo de referencia (Forsgren et al., 2018) como el resultado del análisis estadístico de una investigación donde determinaron los elementos claves que impulsan mejoras el rendimiento en la entrega de software en las organizaciones. Estas capacidades cubren ámbitos desde el punto de vista de gestión de procesos, técnicos, culturales, entre otras con la finalidad de escalar su rendimiento en la entrega de software en aspectos particulares. La Figura 3.3 representa el modelo general planteado por los autores, donde se refleja la relación entre capacidades y las dimensiones que los agrupa.

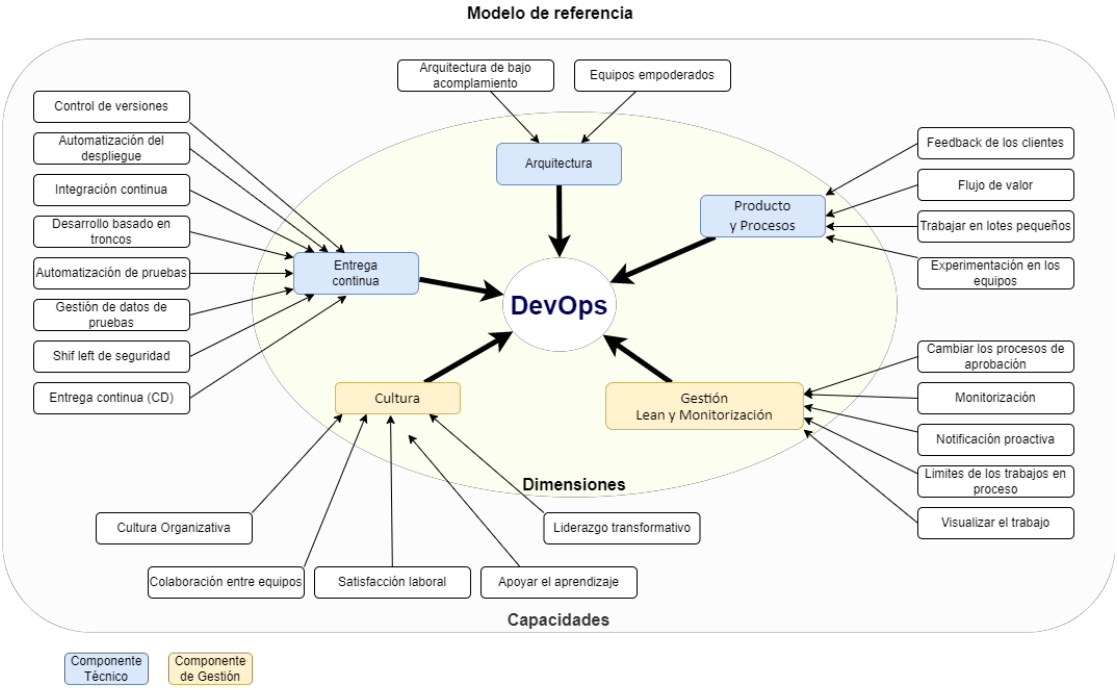

Figura 3.3: Modelo de referencia
Fuente: (Accelerate Forsgren 2018 (Forsgren et al., 2018)])

3.1.3. Pregunta

Cada pregunta del cuestionario propuesto está definida en un modelo de pregunta cerrada con una única opción de respuesta, cubriendo una única capacidad del modelo de referencia. Cada pregunta fue estructurada usando una estrategia de pregunta indirecta con el fin de reducir la complejidad conceptual y ambigüedades al momento de indagar al encuestado. Esta estrategia se diseñó pensando en simplificar el concepto abordado y evitar malas interpretaciones al momento de responder el cuestionario de evaluación, facilitando la medición de las capacidades especialmente para el público encuestado que es nuevo en el tema o no domina los diferentes términos y definiciones propios de DevOps.

El diseño de cada pregunta del marco propuesto fue producto de una revisión bibliográfica de diferentes modelos de madurez DevOps como **DevOpsHealthRadarAssessment** (AgilityHealth), **Dora reseach program** (Google), **Owaps** (OWASP Foundation), entre otros (para mayor detalle consultar la sección (ver 2.2.3), los cuales presentan modelos de evaluación completos, con la particularidad que en su gran mayoría son presentados al usuario final con conceptos técnicos, difíciles de entender para el encuestado no familiarizado con los conceptos propios de DevOps. El propósito al diseñar cada pregunta fue reducir estos aspectos de uso de lenguaje técnico avanzado de la terminología DevOps, con el fin de reducir la complejidad de lo que se está preguntando, enfocando el cuestionario para el segmento con poco o ningún conocimiento previo de las prácticas DevOps.

3.1.4. Respuesta

Las opciones de respuesta para cada pregunta del modelo fueron diseñadas en una escala de 1 a 5 usando una estrategia de valores extremos, donde 1 indica la ausencia o no uso de la práctica DevOps y 5 el uso avanzado y extendido de la capacidad abordada, después se definió la opción 3 como un valor intermedio equivalente a los extremos, por último, para los puntos 2 y 4 se usó la estrategia de analizar cada opción como estado de transición de las dos opciones de respuesta vecinas.

3.1.5. Plan de acción

El plan de acción presenta un conjunto de acciones concretas propuestas para abordar y mitigar cada una de las opciones de respuesta planteadas en el modelo de capacidades diseñado, su construcción se basó en la consolidación de diferentes fuentes de información especializadas en la capacidad abordada con el fin de proponer recomendaciones necesarias para potenciar y llevar la práctica evaluada al siguiente nivel de madurez apoyándose en la estrategia de resaltar los elementos claves a considerar acorde al nivel de desarrollo de la capacidad representado por la escala de madurez de las opciones de respuesta de cada pregunta.

La construcción del plan de acción fue la tarea más desafiante del marco de evaluación debido a que cada uno de ellos debió ser analizado y medido acorde a la opción de respuesta que lo representa. Uno de los principales obstáculos durante la etapa de construcción fue el tratamiento del punto 5, que debe resaltar que llegar a este nivel no es la cúspide del proceso y no haya nada más por mejorar, por el contrario, DevOps es un proceso de mejora y aprendizaje continuo que la organización debe adoptar como cultura del día a día, aquí se recomiendan los puntos claves a considerar para revaluar y optimizar las prácticas existentes además de estar abierto a la adopción de nuevos conceptos.

3.2. Formato de respuestas y sistema de puntaje

3.2.1. Captura de información de entrada

El *modelo de capacidades* propuesto está compuesto por un cuestionario de preguntas cerradas descrito en el punto Modelo de capacidades donde se busca medir el nivel de adopción o uso de un grupo de capacidades que representan prácticas DevOps para fortalecer el rendimiento en la entrega de software.

Para poder analizar las respuestas del cuestionario de preguntas se planeó una estrategia para definir las opciones de respuesta en una escala de 1 a 5. La Tabla 3.1 representa un ejemplo de la unidad de análisis que representa cada opción de respuesta clasificada en 5 niveles de progreso y adopción de la capacidad, donde la nota 1 es la ausencia o no manejo de la capacidad y la nota 5 equivale a un nivel de uso avanzado de la capacidad.

Nota	*Nivel*	*Descripción*
1	Sin experiencia	Las actividades técnicas y de gestión se realizan en procesos completamente manuales
2	Novato	Se cuenta con conocimientos fundamentales en procesos técnicos y de gestión de procesos parcialmente automatizados, pero no integrados entre sí
3	Intermedio	Se cuenta con procesos estandarizados, que apoyan la toma de decisiones compartidas, se tienen procesos centrales automatizados y parcialmente integrados a lo largo del ciclo de vida del desarrollo de software
4	Destacado	Se usan procesos estandarizados que apoyan la toma de decisiones compartidas, gran parte de los procesos de automatización y pruebas ya están integrados
5	Avanzado	Se usan procesos estandarizados avanzados en toma de decisiones compartidas, procesos completamente integrados y automatizados a lo largo del ciclo de vida del desarrollo de software

Tabla 3.1: Nivel de escala de notas

3.2.2. Transformación de resultados en unidades de análisis

Analizar los resultados de la encuesta de medición de capacidades, combinó de un grupo de fases que se encargaron de transformar los datos en crudo de la encuesta en unidades de análisis. La Figura 3.4 presenta las diferentes fases implementadas para obtener unidades de análisis definidas para representar los resultados de la encuesta, los cuales se detallan a continuación:

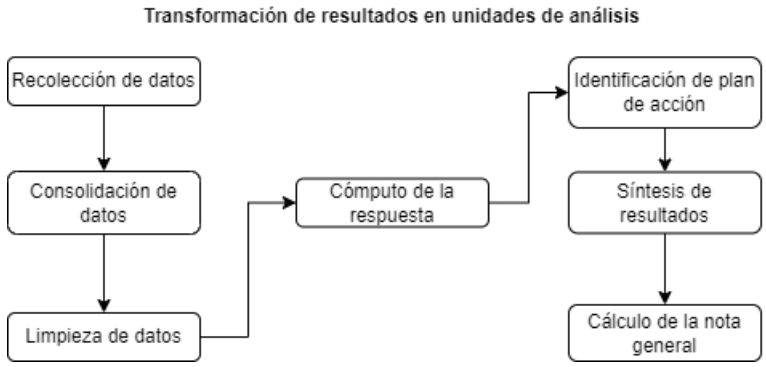

Figura 3.4: Transformación de resultados en unidades de análisis
Fuente: Elaboración propia

Recolección de datos: Esta fase inicia con la definición del medio de captura de datos del cuestionario, por ejemplo, Forms de Office 365 o Google Forms. El

principal requisito para la herramienta de captura de información es que permita crear preguntas cerradas, elegir una única respuesta por pregunta y que al final del proceso poder exportar las respuestas capturadas en una hoja electrónica como un documento Excel.

Consolidación de datos: Una vez finalizada la etapa de recolección de datos, la herramienta de captura de información debe permitir exportar los resultados a un archivo Excel con el fin de procesar y consolidar los datos obtenidos con el modelo de referencia desarrollado. Este punto consta de dos fuentes de datos, el primero son los datos obtenidos de la encuesta y el segundo es un archivo fijo de referencia que contiene el modelo de referencia completo, las escalas de evaluación y los rangos de notas.

Limpieza de datos: Este proceso se encarga de limpiar y depurar los datos obtenidos de la encuesta con el propósito de excluir del análisis las opciones de respuesta que no estén contempladas dentro del *modelo de capacidades*.

Cómputo de la respuesta: Teniendo en cuenta que cada opción de respuesta está expresada en una escala de 1 a 5, se utilizó la estrategia de tomar la mínima opción de respuesta para determinar la opción de respuesta que va a presentar cada una de las preguntas del modelo.

Se usó la estrategia de valor mínimo con el propósito de identificar alertas tempranas en las prácticas organizacionales que se están evaluando desde el punto de vista de la relevancia de su uso y adopción para el encuestado.

Por otro lado, esta estrategia permite obtener un panorama general de los eslabones más débiles de la organización, lo que permite identificar los elementos y las herramientas necesarias para la toma de decisiones para iniciar la adopción DevOps fundamentada en la mejora de las practicas organizaciones partiendo de la premisa de fortalecer las bases.

Esta estrategia genera una especial atención a la hora de definir el grupo representativo idóneo para presentar el cuestionario de evaluación y así conseguir que el modelo pueda capturar la realidad más aproximada del estado de las practicas organizacionales.

Identificación de plan de acción: Una vez obtenida la nota que representa cada pregunta del *modelo de capacidades* y dado que el modelo diseñado relaciona un plan de acción por cada opción de respuesta de una pregunta, se puede asociar un plan de acción por cada pregunta a partir de la nota calculada.

Síntesis de resultados: Consiste en realizar un análisis descriptivo de la nota calculada por cada pregunta, usando una escala de categorías de calificación definida en la Tabla 3.2 en donde se definen 3 categorías de evaluación que permiten crear niveles de agrupación a partir de la nota de cada pregunta.

Título	*Rango de notas*	*Descripción*
Oportunidades de mejora	Notas menores a 3.0 (<3.0)	Representa los elementos con nota baja los cuales son caracterizados como oportunidades de mejora, son los primeros que deben ser abordados y tratados por la organización al momento de iniciar el camino de adopción de prácticas DevOps.
Aceptable	Notas entre 3.0 y 4.0 (>=3.0 y <4.0)	Son los elementos que están en un estado intermedio por encima de las oportunidades de mejora, pero no lo suficiente para definirse como fortaleza. Los elementos clasificados dentro de esta categoría deben ser analizados con el fin de fortalecerlos y alcanzar el siguiente nivel
Fortalezas	Notas mayores o iguales a 4.0 (>=4.0)	Esta categoría define los elementos que están fuertes en la organización, los cuales no pueden ser descuidados, por el contrario, también deben ser manejados como procesos de mejora continua

Tabla 3.2: Escala de categorías por calificación

Utilizando la estructura que compone el *modelo de capacidades* presentado en la Figura 3.1 donde existe una relación directa entre preguntas del cuestionario y las capacidades del modelo de referencia, se extendió el análisis descriptivo de las categorías a las capacidades y las dimensiones, donde la nota del nivel se obtuvo promediando la nota de cada pregunta que lo integra.

Cálculo de la nota general: Complementario al análisis descriptivo, se promedió la nota de cada pregunta del cuestionario, obteniendo un valor numérico que presenta la nota alcanzada por la organización en el análisis de capacidades DevOps en la escala de 1 a 5.

3.2.3. Herramienta de implementación del instrumento

Para el caso académico se utilizó la cuenta de Office 365 de la Universidad para la implementación del formulario de encuesta y captura de datos y Power BI como herramienta para el análisis y presentación de los resultados a través de gráficos analíticos

El cuestionario del *modelo de capacidades* diseñado se implementó en un Formulario de Encuestas de Office 365, que permite al encuestado responder cada pregunta eligiendo una única opción de respuesta y al final del proceso la herramienta permite descargar los resultados de la encuesta en un archivo Excel. El formulario puede ser visualizado y copiado a través del siguiente link: https://tinyurl.com/4pue2sz7

El análisis de los datos se realizó utilizando la herramienta Power BI en su versión de escritorio en donde se calcularon las unidades de información definidas en el punto anterior a partir de la fase de limpieza de datos en adelante. Una vez se calcularon las unidades de información definidas se generaron gráficos analíticos con los resultados obtenidos en diferentes perspectivas en un dashboard.

El informe final con los gráficos obtenidos después de analizar los resultados de la evaluación de capacidades DevOps en la organización fueron cargados en el espacio de trabajo gratuito de Power BI que ofrece la cuenta académica de la universidad.

3.2.4. Presentación de resultados

Para la presentación de los datos, se creó un reporte gráfico estilo dashboard en donde se construyeron diferentes gráficos y tablas que presentan resúmenes y consolidación de datos a partir de las unidades de análisis obtenidas de los resultados de la encuesta y del *modelo de capacidades*. El informe de resultados fue diseñado basado en una estrategia de buenas prácticas para el diseño de gráficos que permite presentar la información que vaya de lo general a lo específico, donde los elementos generales como la nota final, las calificaciones por dimensión se ubican en la parte superior del reporte y los elementos más específicos del modelo como detalle de nota por capacidad, agrupación de categorías se ubican de la mitad de la página hacia a abajo.

El principio de diseño implementado buscó mostrar diferentes perspectivas acordes a la unidad de análisis, por ejemplo, se utilizaron gráficos de barra para representar la nota promedio de cada dimensión resaltando la nota mínima de la escala de calificación como línea de corte, también se crearon tablas con las capacidades agrupadas en cada rango de calificación.

La escala de calificación presentada en laTabla 3.2 se complementó con la presentación de los resultados a través de una escala de colores inspirado en un semáforo para representar sus 3 categorías, rojo para las oportunidades de mejora, amarillo para los aceptables y verde para las fortalezas.

La elección de Power BI como herramienta de visualización de los resultados permitió el uso de gráficos dinámicos, botones y eventos que permite la creación de sub reportes con el propósito de generar niveles de detalle para visualizar bloques de información particulares complementarios.

La Figura 3.5 muestra un ejemplo de algunos gráficos y los colores usados en el dashboard para presentar los resultados de la evaluación y la escala de calificación del caso práctico que será descrito en el siguiente capítulo.

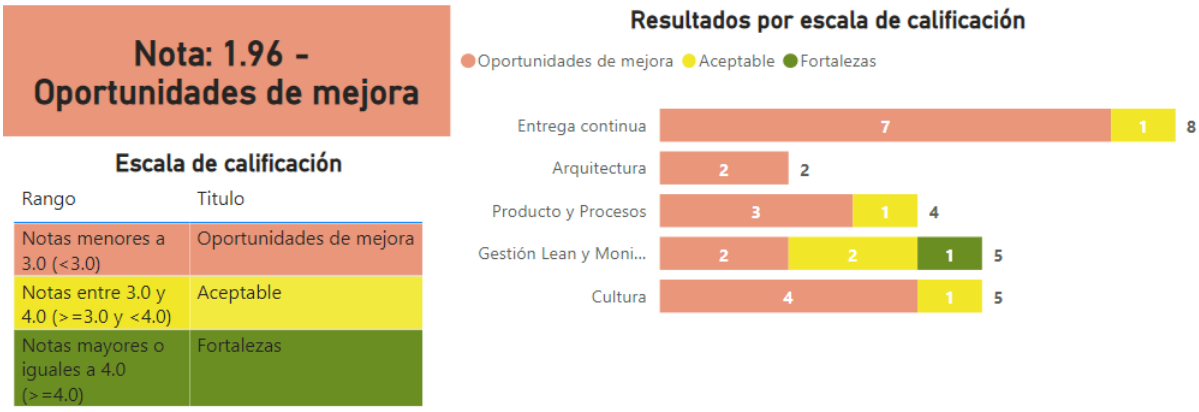

Figura 3.5: Ejemplo de reporte dashboard

Fuente: Elaboración propia

3.2.5. Proceso para usar el instrumento de evaluación

El proceso de medición de capacidades de una organización fue diseñado en tres etapas: Preparación, ejecución y resultados, la Figura 3.6 muestra una representación gráfica de las etapas y los elementos que componen cada fase.

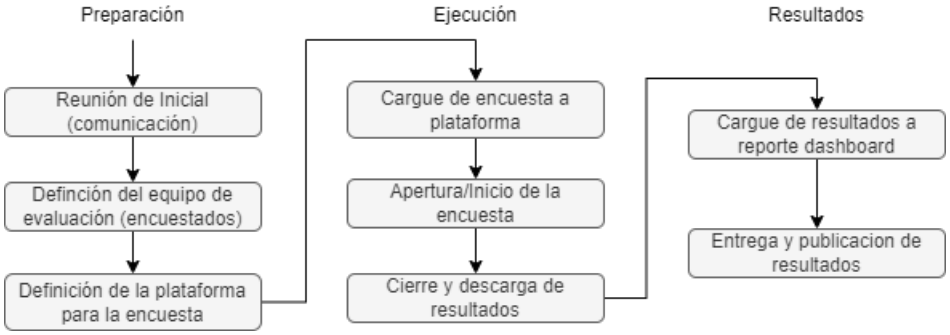

Figura 3.6: Proceso de medición de capacidades

Fuente: Elaboración propia

Preparación: Esta fase definió quién, cómo y cuándo se va a realizar la encuesta de medición de capacidades en la organización, inicia con una reunión inicial con los representantes de los directivos de la organización donde se definen los participantes en la medición, la plataforma a utilizar y el tiempo para que el grupo definido diligencie la medición de capacidades. Se recomienda que el grupo de encuestados este compuesto por representantes de los equipos de desarrollo, infraestructura incluyendo líderes técnicos y demás personas que participen directamente el proceso de desarrollo de software.

Ejecución: En esta fase se cargaron las preguntas del *modelo de capacidades* propuesto a la plataforma de encuesta elegida, una vez se complete este paso, se habilita el cuestionario para que el grupo de encuestados lo diligencie bajo el

tiempo definido, al momento de finalizar el sondeo en la organización se obtienen los resultados en crudo de la plataforma para que sean depurados y transformados en unidades de análisis.

Resultados: Esta fase se encargó de transformar los datos obtenidos en la medición de capacidades en unidades de análisis, para sintetizar los resultados en gráficos expresado en un reporte estilo dashboard y obtener el plan de acción de cada pregunta. Esta fase toma como entrada el resultado de la encuesta exportado de la herramienta utilizada, donde se ejecutó un proceso de depuración y limpieza de datos que permitió realizar los cálculos analíticos descritos en el punto Transformación de resultados en unidades de análisis, obteniendo un reporte dinámico con el diagnóstico de los factores de evaluación claves para determinar el estado de las capacidades DevOps de la organización.

El caso de estudio donde se aplicó el *modelo de capacidades* propuesto a través de la implementación de las diferentes fases de implementación y análisis será explicado en detalle en el siguiente capítulo.

3.2.6. Limitaciones

El modelo de evaluación de capacidades diseñado y propuesto fue una creación propia orientada para evitar ambigüedades y sesgos conceptuales a la hora de medir el uso de prácticas DevOps en una organización. Cada pregunta del cuestionario se basó en una capacidad del modelo de referencia, por lo que en algunos casos el modelo planteado puede quedarse corto en aspectos complementarios que no están cubiertos en el modelo de referencia.

Al igual que las preguntas del cuestionario, las opciones de respuestas y el plan de acción fueron creación propia, inspiradas en el modelo de referencia, por lo que pueden existir conceptos DevOps no contemplados y cubiertos en el modelo de evaluación propuesto.

A partir del alcance y limitación de este documento, el *modelo de capacidades* propuesto solo fue probado en un único caso particular, en este caso la compañía Alianza Bioversity CIAT como se detalla en el siguiente capítulo Caso de estudio.

Por último, este documento se enfocó en definir un marco de evaluación DevOps general y se particularizó en una única compañía, por lo que la captura de datos y su posterior análisis no están integrados en un único proceso o aplicación que automatice todas las propuestas.

3.3. Resumen del capítulo

La adopción DevOps en una compañía no se puede llevar a cabo siguiendo una hoja de ruta genérica, por el contrario, cada organización debe definir su propio proceso acorde a cómo funciona, opera y gestiona el ciclo de vida de las aplicaciones, identificando debilidades, cuellos de botella y prácticas corporativas que de alguna

manera afecten la entrega de productos de software. Para facilitar este tipo de evaluaciones, se diseñó un artefacto de medición de capacidades enfocado en el segmento de profesionales TI con nuevos o con pocos conocimientos de las prácticas DevOps. El artefacto de medición denominado *modelo de capacidades*, que se fundamenta en el resultado de la investigación del libro Accelerate (Forsgren et al., 2018), donde identificaron 24 capacidades que abarcan ámbitos técnicos, gestión de procesos y cultura que a nivel estadístico impulsan mejoras en el rendimiento en la entrega de software en una compañía.

El *modelo de capacidades* está compuesto por un cuestionario de 25 preguntas cerradas que representa cada una de las 24 capacidades del modelo de referencia, una justificación del porqué de la pregunta, 5 opciones de respuesta por cada pregunta expresadas en una escala de 1 a 5 y un plan de acción asociado a cada opción de respuesta, la Figura 3.2 muestra un ejemplo de la composición de una pregunta del *modelo de capacidades*.

En cifras totales, el *modelo de capacidades* propuesto está compuesto por los siguientes elementos:

- 5 dimensiones (modelo de referencia)
- 24 capacidades (modelo de referencia)
- 25 preguntas (cada pregunta representa una única capacidad)
- 125 opciones de respuesta (5 opciones por cada pregunta)
- 125 planes de acción (un plan de acción por cada opción de respuesta)

Analizar los resultados de la encuesta de medición de capacidades, combinó de un grupo de fases que se encargaron de transformar los datos en crudo de la encuesta en unidades de análisis, iniciando con la definición del medio de captura de datos del cuestionario, la consolidación y limpieza de los datos para poder determinar la opción de respuesta más elegida por los encuestados en cada pregunta y que representa la nota de la pregunta. A partir de calcular la nota por cada pregunta, se asoció el plan de acción propuesto por el modelo, por último, con estos datos calculados se realizó un análisis descriptivo apoyado en las categorías descritas en la Tabla 3.2 con los que se obtuvo unidades de análisis representables en gráficos analíticos.

El informe de resultados se basó en una estrategia de buenas prácticas para el diseño de gráficos con el objetivo de presentar la información con una estrategia que va de lo general a lo específico, donde los elementos generales se ubicaron en la parte superior y los elementos más particulares del modelo se ubicaron en las zonas de mitad de página hacia a abajo.

El siguiente capítulo describirá cómo se aplicaron las diferentes fases del modelo de capacidades propuesto en un caso de estudio particular para la compañía CIAT.

Capítulo 4

Caso de estudio

Acorde al alcance definido este documento, el caso de estudio se llevó a cabo únicamente en la compañía Alianza Bioversity – CIAT, donde se realizó el diagnóstico del estado actual de los procesos de entrega de software usando el *modelo de capacidades* propuesto y detallado en el capítulo anterior. Con esta evaluación se buscó determinar la conveniencia de adoptar DevOps como estrategia para la entrega continua de productos de software en la organización.

Durante el proceso de medición y captura de datos se contó con el apoyo y disposición de los equipos de desarrollo e infraestructura de la organización especialmente con los coordinadores de cada área, por lo que el acceso a la información de la compañía fue ágil y oportuno.

4.1. Ejecución del caso de estudio

4.1.1. Configuración del caso de estudio y objetivos del análisis

La medición de capacidades DevOps en la compañía Alianza Bioversity – CIAT a través del modelo propuesto, buscó ir más allá de solo medir que tan evolucionados y automatizados están sus procesos de entrega de software. El principal objetivo fue brindar una herramienta ligera que minimice la complejidad conceptual al momento de medir aspectos que impulsan la entrega de software en la organización, a su vez identificar y clasificar estos aspectos en puntos débiles y fuertes para que sean abordados a través de un plan de recomendaciones por cada capacidad evaluada en forma de plan de acción.

4.1.2. Ejecución del caso de estudio

El proceso de medición de las capacidades DevOps en la compañía Alianza Bioversity – CIAT, implementó las etapas de uso del *modelo de capacidades* descrito en la Figura 3.6 y la transformación de datos en unidades de análisis se realizó acorde a lo detallado en la Figura 3.4.

Preparación: Este paso se ejecutó en acuerdo y coordinación directa con los directores de las áreas de integración de sistemas y redes e infraestructura, participantes directos en la ejecución del ciclo de vida del desarrollo de software para las áreas administrativas y operativas de la organización.

En la reunión inicial con los directores del área de integración de sistemas se presentó el objetivo y alcance de la medición que se llevó a cabo y se recibió el aval para la medición de capacidades DevOps en el área. Al momento de definir el grupo de encuestados se involucró al área de infraestructura, ya que la compañía no cuenta con área de calidad de software. El criterio de selección de los participantes en el sondeo se basó en que las personas elegidas debían estar directamente involucradas en el proceso de desarrollo y entrega de software. Al final se definió un total de 8 encuestados, 4 representantes por área, que incluyó a los dos coordinadores técnicos de las áreas.

Para este caso práctico, aprovechando la cuenta académica de la universidad y propósito académico de este trabajo se eligió la herramienta Forms de Office 365 para implementar la encuesta de medición de capacidades, adicionalmente y con el fin de brindar un espacio libre de presión para que los encuestados pudieran responder con total franqueza el cuestionario de medición de capacidades DevOps, se configuró la herramienta de encuesta para que los encuestados pudieran registrar sus respuestas de forma anónima. La comunicación hacia el equipo de evaluación se hizo a través de un correo electrónico indicándoles el propósito y alcance de la medición y la característica de anonimato a la hora de responder el cuestionario junto con el enlace de acceso a la herramienta.

Ejecución: En este paso se configuró el cuestionario del *modelo de capacidades* y se cargaron las preguntas con sus opciones de respuestas al formulario de encuestas de Office 365. Con el fin de brindar una alternativa de respuesta cuando el encuestado no tenía el suficiente conocimiento o experticia sobre el tema preguntado, se agregaron dos opciones de respuesta (No aplica y No sabe /No responde) a cada pregunta del *modelo de capacidades* en el formulario de encuesta.

Para identificar el perfil del encuestado, se agregó una pregunta adicional a la encuesta preguntando el área de la organización (Desarrollo (System Integration) o Infraestructura Redes y seguridad) a la que pertenece el encuestado.

Una vez finalizada la encuesta, desde Microsoft Forms se exportó el resultado consolidado de las respuestas del cuestionario de medición de capacidades a un archivo Excel para su posterior análisis. La Figura 4.1 presenta el consolidado que la herramienta generó indicando el número de respuestas por parte de los participantes definidos en la etapa de preparación y el tiempo promedio para responder el cuestionario.

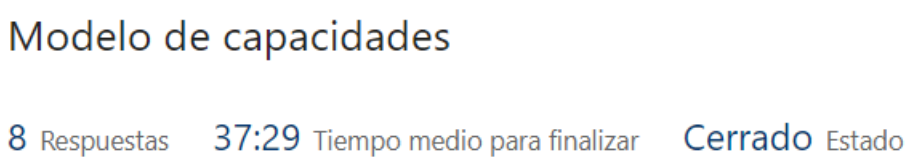

Figura 4.1: Número de respuestas de la encuesta de capacidades en CIAT
Fuente: Elaboración propia

La Figura 4.2 confirma la distribución equitativa de los presentantes de cada área en la medición, acorde a lo definido. para la ejecución, un total de 8 personas distribuidas en 4 del área de redes e infraestructura y 4 del área de desarrollo.

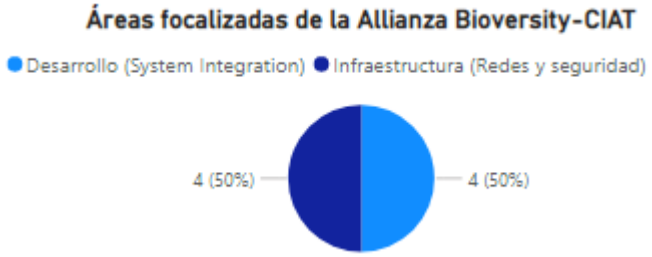

Figura 4.2: *Modelo de capacidades*, distribución de encuestados
Fuente: Elaboración propia

Resultados: En este paso se creó un segundo archivo Excel que contiene el *modelo de capacidades* completo, es decir, el modelo de referencia, el cuestionario de preguntas, la relación entre el cuestionario y el modelo de referencia, las opciones de respuestas a cada pregunta, los planes de acción, la escala de medición y el rango de notas. Junto con el archivo de los resultados de la encuesta, fueron cargados en un reporte analítico desarrollado en la herramienta Power BI, que se encargó de realizar todo el proceso de transformación de resultados en unidades de análisis.

La limpieza de los resultados de la encuesta consistió en quitar las opciones de respuestas *No aplica y No sabe /No responde* de cada pregunta con el fin de omitir del análisis las opciones de respuesta no alineadas al *modelo de capacidades* y centrarse solo en los elementos definidos en el modelo propuesto.

Al finalizar la fase de limpieza de datos, se ejecutaron las diferentes fases de la transformación de los resultados en unidades de análisis, iniciando por el cálculo de la moda por cada pregunta, obteniendo la nota que representa la opción de respuesta más elegida por los encuestados en cada pregunta del cuestionario.

Acorde con la definición del *modelo de capacidades* cada opción de respuesta tiene asociado un único plan de acción, a partir de la nota de cada respuesta se obtuvo el plan de acción para cada una de las preguntas del modelo de evaluación. El anexo Plan de acción presenta el detalle de los planes de acción obtenidos para el caso de estudio.

La Figura 4.3 muestra la nota general de la medición del caso práctico con un valor de 1.96 en una escala de 1 a 5. Esa cifra se obtuvo promediando la nota de

cada pregunta del cuestionario de medición y representa un alcance del 39.2 % de los conceptos básicos DevOps para la organización.

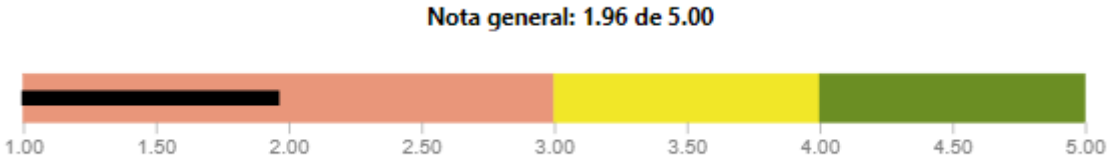

Figura 4.3: Nota final del caso práctico

Fuente: Elaboración propia

4.2. ¿Cómo iniciar la adopción DevOps?

Basado en lo propuesto por Sharma (2017), la Figura 4.4 presenta un conjunto de pasos recomendados para iniciar la adopción DevOps en una organización. Para el caso particular de CIAT y acorde al alcance del presente trabajo se define el siguiente plan de adopción:

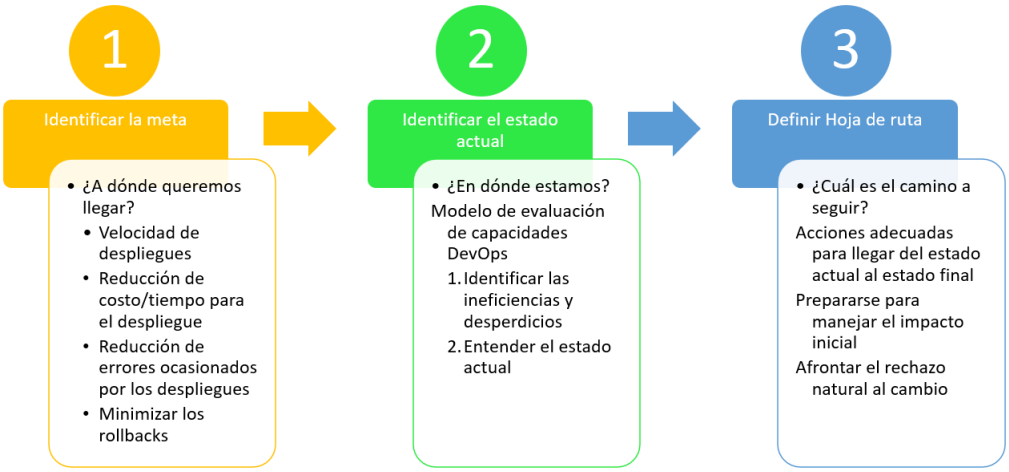

Figura 4.4: Pasos claves para iniciar una transformación DevOps

Gráfico generado a partir de lo indicado en (Sharma, 2017)

4.2.1. Identificar el Estado objetivo

Acorde al enfoque del presente documento y apoyado en una revisión interna del CIAT, donde se midió la frecuencia y tiempos de despliegue de área de integración de sistemas en el 2021, se determinó encaminar las recomendaciones de adopción para los indicadores de negocio:

- Velocidad en la frecuencia de despliegues
- Reducción de tiempo en el despliegue

La Figura 4.5 presenta el tiempo (medido en días calendario) de despliegue en producción para el año 2021 en el CIAT, en este caso la unidad de medida fue horas con un promedio 2021 de 15.45 horas. El segundo aspecto que determinó el enfoque es la frecuencia de entrega de productos, la Figura 4.6 presenta un promedio de entrega de productos de 1.41 meses (43 días)

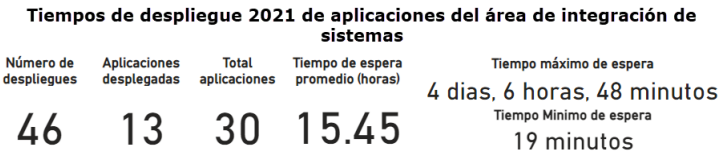

Figura 4.5: Tiempos de despliegue 2021 de aplicaciones del área de integración de sistemas – CIAT
Fuente: Elaboración propia

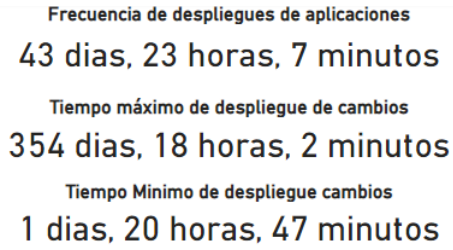

Figura 4.6: Frecuencia de entrega a producción 2021 de aplicaciones del área de integración de sistemas – CIAT
Fuente: Elaboración propia

4.2.2. Entender el estado actual

4.2.2.1. Visualización de datos

Para la presentación de los resultados de la evaluación del caso de uso (en este caso CIAT), se creó un reporte consolidado que a través de gráficos analíticos y una estrategia de visualización que va de lo general a lo específico permitieron identificar el estado actual de la organización acorde a las diferentes dimensiones y capacidades evaluadas en el *modelo de capacidades* propuesto. Por otro lado, apoyados en la escala de evaluación definida el reporte permitió identificar qué elementos se deben priorizar al momento de iniciar la adopción de prácticas DevOps para que la compañía CIAT pueda mejorar los tiempos de entrega

Una vez se depuró y consolidó el resultado de la encuesta de evaluación del caso de uso (en este caso CIAT), se creó un reporte consolidado que por medio de una estrategia de visualización de gráficos analíticos que va de lo general a lo específico, logró identificar el estado actual de la organización acorde a las diferentes dimensiones

y capacidades evaluadas en el *modelo de capacidades* propuesto. Apoyados en la escala de evaluación (definida en el capítulo Propuesta del artefacto de evaluación), el reporte permitió clasificar los resultados en un orden que permite priorizar qué elementos se deben tratar primero al momento de iniciar la adopción de prácticas DevOps en la organización CIAT.

La Figura 4.7 presenta una vista preliminar del reporte con los resultados de la evaluación del caso de estudio CIAT, este reporte está disponible para su consulta web a través del siguiente enlace https://tinyurl.com/4vp9s22c.

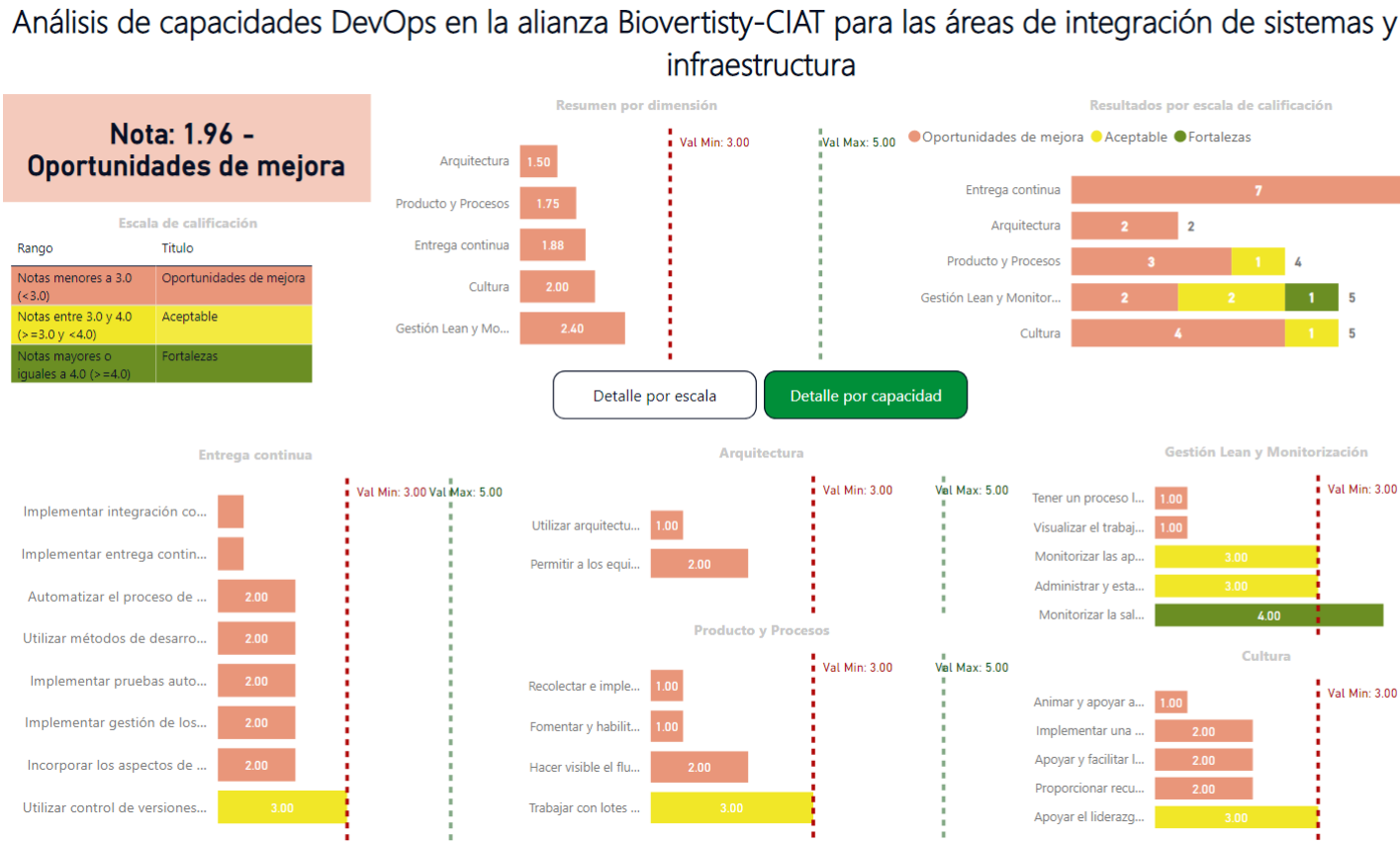

Figura 4.7: Resumen de los resultados del caso de estudio

4.2.2.2. Resultados

Para identificar el estado de madurez de los procesos de desarrollo de software en una organización se utilizó el *modelo de capacidades* propuesto y detallado en el capítulo anterior. Para el caso particular CIAT, en este capítulo se implementó el modelo evaluación propuesto y se obtuvo un grupo de resultados que fueron analizados para determinar la conveniencia de adoptar DevOps como estrategia para la entrega continua de productos de software en la organización.

El objetivo de la evaluación con el *modelo de capacidades* es determinar qué tan robustas están las prácticas de desarrollo de software en la organización frente las recomendaciones DevOps. Basado en lo propuesto por Sharma (2017), el modelo se

centró en evaluar las dimensiones y capacidades con el fin de poder dar respuestas a las siguientes preguntas:

1. ¿Qué tan capaz es la organización de ofrecer rápidamente aplicaciones nuevas e innovadoras y de aprovechar las arquitecturas modernas?

2. ¿Qué tan capaz es la organización de modernizar las aplicaciones existentes y permitir una entrega más rápida y la innovación?

3. ¿Qué tan capaz es la organización para adaptar prácticas en aspectos como la cultura, las herramientas y los procesos?

4.2.2.3. Análisis de los resultados

Basados en la escala de categorías de calificación de la Tabla 4.1, las capacidades y dimensiones del modelo propuesto se agruparon con el objetivo obtener 3 grupos focales acorde a la nota promedio de cada pregunta en la encuesta, logrando clasificar los diferentes elementos del modelo en grupos de mayor a menor priorización, obteniendo un orden recomendado para el tratamiento y adopción de planes de mejora para cada una de las capacidades del modelo.

La Figura 4.8 presenta un resumen consolidado de las cifras porcentuales de los resultados obtenidos desde el punto de vista de la categoría y de la dimensión evaluado

Dimensión vs Categoría

Categoría	Oportunidades de mejora			Aceptable			Fortalezas			Total		
Dimensión	Total	% Dim	% Cat	Total	% Dim	% Cat	Total	% Dim	% Cat	**Total**	**% Dim**	**% Cat**
Entrega continua	7	38.89%	87.50%	1	20.00%	12.50%				**8**	**33.33%**	**100.00%**
Gestión Lean y Monitorización	2	11.11%	40.00%	2	40.00%	40.00%	1	100.00%	20.00%	**5**	**20.83%**	**100.00%**
Cultura	4	22.22%	80.00%	1	20.00%	20.00%				**5**	**20.83%**	**100.00%**
Producto y Procesos	3	16.67%	75.00%	1	20.00%	25.00%				**4**	**16.67%**	**100.00%**
Arquitectura	2	11.11%	100.00%							**2**	**8.33%**	**100.00%**
Total	**18**	**100.00%**	**75.00%**	**5**	**100.00%**	**20.83%**	**1**	**100.00%**	**4.17%**	**24**	**100.00%**	**100.00%**

Figura 4.8: Resultados de evaluación por categoría y dimensión

Oportunidades de mejora:

La Tabla 4.1 detalla las capacidades evaluadas con su dimensión asociada que fueron clasificadas en esta categoría. Un aspecto importante para resaltar con estos resultados obtenidos es que $\frac{3}{4}$ partes de todas las capacidades evaluadas en el modelo están aquí clasificadas, lo que genera una alerta del nivel actual de las prácticas técnicas y de gestión de procesos de la organización.

A este grupo se le realizó un análisis de mayor a menor respecto al porcentaje de capacidades clasificadas como Oportunidad de Mejora, el propósito es analizar cada una de las dimensiones y capacidades presentes en esta escala de categorización acorde al porcentaje de incidencia en esta categoría y así tener elementos de base proponer estrategias para su mejora y priorización.

Dimensión	Capacidad
Entrega continua	Automatizar el proceso de despliegue
Entrega continua	Implementar integración continua
Entrega continua	Utilizar métodos de desarrollo "Trunk-Based"
Entrega continua	Implementar pruebas automáticas
Entrega continua	Implementar gestión de los datos de prueba
Entrega continua	Incorporar los aspectos de seguridad en las fases de diseño y pruebas (shift left de la seguridad)
Entrega continua	Implementar entrega continua de software
Arquitectura	Utilizar arquitecturas de bajo acoplamiento
Arquitectura	Permitir a los equipos definir sus propias arquitecturas
Producto y Procesos	Recolectar e implementar las recomendaciones (feedback) de los clientes
Producto y Procesos	Hacer visible el flujo de trabajo durante todo el ciclo
Producto y Procesos	Fomentar y habilitar al equipo para realizar experimentos
Gestión Lean y Monitorización	Tener un proceso liviano para la aprobación de cambios
Gestión Lean y Monitorización	Visualizar el trabajo para monitorizar la calidad y comunicarlo al equipo
Cultura	Implementar una cultura tipo generativa
Cultura	Animar y apoyar al equipo a aprender
Cultura	Apoyar y facilitar la colaboración entre los equipos
Cultura	Proporcionar recursos y herramientas para que el trabajo sea significativo para los individuos

Tabla 4.1: Resultado escala de clasificación - Oportunidades de mejora

- La entrega continua es la dimensión que presenta mayor debilidad, 38.89 % (7 de las 18) capacidades clasificadas como oportunidad de mejora pertenece a la dimensión entrega continua. Desde el punto de vista de toda la dimensión el 87.5 % (7 de las 8) de sus capacidades están clasificadas en esta categoría; su calificación promedio la sitúa como debilidad critica, ubicándola como una de las dimensiones de alta priorización al momento de iniciar la adopción de prácticas DevOps en la organización. En este punto se destaca la evidente debilidad en aspectos de acción inmediata como la Integración de procesos, Gestión de procesos y Pruebas, para que sean la base para abordar las demás prácticas de entrega continúa clasificadas en esta categoría.

- La cultura es la segunda dimensión con mayor número de capacidades clasificadas en esta categoría con el 22 % de las capacidades clasificadas en esta categoría (4 capacidades). Desde el punto de vista general de la dimensión, el 40 % de sus capacidades se clasificaron como oportunidades de mejora con un promedio ge-

neral de 2.0. Se destacan aspectos a priorizar como: potenciar la colaboración entre los miembros del trabajo en equipo y generar empoderamiento progresivo frente a la organización a través de espacios de aprendizaje en ámbitos técnicos y de trabajo en equipo.

- Producto y procesos con el 16.67 % compuesto por 3 capacidades clasificadas en esta categoría, a nivel dimensional general refleja que el 75 % (3 de 4) de sus capacidades están clasificadas dentro de esta categoría y su promedio general se clasifica en la categoría oportunidad de mejora, en este nivel la priorización está enmarcada en iniciar la adopción de los mecanismos necesarios para recolectar eficientemente las retroalimentaciones de los usuarios finales (clientes) en todo el ciclo de vida del desarrollo del producto de software y de elementos que permitan visibilizar el flujo de actividades para todos los miembros del equipo de trabajo.

- En un doble empate con el 11.11 % de las capacidades clasificadas como oportunidad de mejora se posicionan las dimensiones:

- Arquitectura con 2 capacidades, desde el punto de vista general de toda la dimensión representa el 100 % de sus capacidades clasificadas como oportunidades de mejora. Este caso particular genera un elemento de consideración de alta prioridad porque toda la dimensión tiene la calificación más baja, con un promedio general de 1.5 en la escala de evaluación. Se debe iniciar con la adopción incremental desde las bases para iniciar a justar la infraestructura actual desde los conceptos y alcances para arquitecturas físicas como lógicas de bajo acoplamiento en donde los equipos tengan la posibilidad de definir y aportar ideas en que arquitectura es adecuada para proyecto.

- Gestión LEAN con 2 capacidades clasificada en esta categoría, a nivel de dimensión la representación es del 40 % de todas sus capacidades clasificadas como oportunidad de mejora y una nota promedio de 2.4 (la más alta de toda la medición). En este nivel, la prioridad es redefinir y adoptar mecanismos que permitan reducir los cuellos de botella en la gestión y aprobación de procesos, depurando lo que no genere valor y produzcan retrasos en la cadena de valor en la entrega de software; por otro lado, implementar las herramientas necesarias para que esta información esté siempre disponible y sea de fácil acceso a todos los miembros del equipo de trabajo.

En resumen, los aspectos más importantes clasificados como oportunidad de mejora y que se deben priorizar son:

- Procesos de integración y automatización
- Gestión de cambios y metodología de desarrollo
- Silos y falta de comunicación entre las áreas
- Arquitectura de bajo acoplamiento

Aceptables

La Tabla 4.2 presenta el grupo de 5 capacidades clasificadas como Aceptables (20.83 % de todas las capacidades evaluadas), representan los elementos clasificados como intermedio, donde se cuenta con procesos encaminados a apoyar la toma de decisiones compartidas y mecanismos para controlar la carga de trabajo hacia los miembros del equipo de trabajo, a nivel técnico se destaca el uso intermedio de sistemas de control de versiones.

Dimensión	***Capacidad***
Entrega continua	Utilizar control de versiones para todos los artefactos de producción
Producto y Procesos	Trabajar con lotes o cantidades pequeñas
Gestión Lean y Monitorización	Monitorizar las aplicaciones y la infraestructura para tomar decisiones de negocio
Gestión Lean y Monitorización	Administrar y establecer límites para el trabajo en progreso (Work-In-Progress)
Cultura	Apoyar el liderazgo transformacional

Tabla 4.2: Resultado escala de clasificación Aceptable

- Gestión LEAN y Monitorización, con el 40 % tuvo 2 de sus capacidades clasificadas en esta categoría; desde el punto de vista general de toda la dimensión, representan el 40 % de todas las capacidades de esta dimensión. En este nivel de escala, se parte de prácticas base en operación que se deben abordar en un enfoque de mejora continua para seguir escalando en los niveles de madurez, en este punto se resalta el monitoreo tanto de la infraestructura como de operación de las aplicaciones con el fin de llevarlo a niveles de monitoreo predictivos, por otro lado, potenciar los mecanismos para controlar la carga de trabajo y evitar la sobrecarga excesiva tanto físico como mental del equipo de trabajo.

Con un tripe empate con el 20 % de las capacidades categorizadas como aceptables están las dimensiones:

- Producto y proceso, con 1 capacidad clasificada como aceptable, desde el punto de vista general de la dimensión representa 1 de las 4 capacidades (25 %) de toda la dimensión clasificadas en esta categoría. En este nivel se busca pulir y potenciar la gestión de trabajo de forma ágil, de tal forma que se puedan distribuir las actividades en lotes pequeños para que puedan entregarse en periodos de tiempo cortos acorde con el nivel de experiencia y experticia de los miembros del equipo de trabajo. El primer paso de evolución es que las entregas de productos de valor se de en periodos de tiempos menores a 3 meses y progresivamente con la mejora constante de procesos y adopción de prácticas agiles, esta unidad de tiempo se debe ir reduciendo a entregas aún más cortas como semanas e inclusive días.

- Cultura, con 1 capacidad aquí clasificada y desde el punto de vista de general de la dimensión representa el 20 % del total de sus capacidades clasificadas en esta categoría. En este nivel se busca fortalecer el trabajo en equipo, actividad que debe ir de la mano con lideres que potencien la comunicación abierta y constante para que desarrollo, operaciones mejoren la toma de decisiones compartidas e incrementen el trabajo colaborativo buscando evolucionar el trabajo a equipos multifuncionales que abarquen todo el proceso de entrega de software resaltando aspectos como la calidad y la seguridad de software.

- Entrega continua, 1 de sus capacidades se clasificó en esta categoría, desde el punto de vista de la dimensión esta capacidad representa el 12.5 % del total de las capacidades de la dimensión. Se destaca el uso de sistemas de control de versiones para los artefactos en producción se debe potenciar el uso de herramientas de control versiones en ambientes web para facilitar la integración con los demás procesos de desarrollo.

En resumen, los aspectos más importantes clasificados como aceptables que se deben priorizar son:

- Cultura de trabajo orientada al rendimiento (Generativa)
- Metodologías de trabajo ágiles.
- Gestión de proyectos y herramientas para visibilizar el trabajo.
- Liderazgo y motivación del equipo de trabajo.

Fortalezas

La Tabla 4.3 presenta capacidades clasificadas como Fortalezas, estas presentan los niveles de calificación destacado y avanzado. A este nivel se usan procesos estandarizados y avanzados que apoyan la toma de decisiones compartidas, los procesos de automatización y pruebas ya están integrados a lo largo del ciclo de vida del desarrollo de software.

Dimensión	***Capacidad***
Gestión Lean y Monitorización	Monitorizar la salud de los servicios y aplicaciones proactivamente

Tabla 4.3: Resultado escala de clasificación Fortalezas

- Gestión Lean y monitorización con la única capacidad que alcanzó este nivel de clasificación, a nivel de toda la dimensión representa el 20 % (1 de 5) de todas sus capacidades aquí clasificadas. Esta dimensión se debe enfocar en potenciar y alcanzar niveles de predicción proactiva de la salud de los componentes de la aplicación tanto a nivel físico y lógico, el objetivo es anticipar fallos por alta demanda o picos de trabajo y procesamiento.

En resumen, el único aspecto que se debe potenciar y no descuidar es poder tener los mecanismos necesarios para por anticipar situaciones de alta demanda de las aplicaciones y la infraestructura en función a cambios drásticos en recursos del servidor.

- Monitoreo preventivo y proactivo

4.2.3. *Análisis final*

Por último, se da respuesta a las preguntas planteadas al inicio de la medición del estado actual.

1. ¿Qué tan capaz es la organización de ofrecer rápidamente aplicaciones nuevas e innovadoras y de aprovechar las arquitecturas modernas?

   A nivel general CIAT obtuvo un resultado general de Oportunidad de Mejora con una nota promedio de 1.96, lo que indica que la empresa debe iniciar el camino de transformación DevOps a partir de definir unas bases técnicas y teóricas sólidas. La mayor parte de sus prácticas técnicas y gestión de procesos están entre niveles sin experiencia y novato, lo que conlleva a que la compañía deba categorizar qué capacidades se deben abordar con mayor prioridad.

   Los aspectos claves para abordar al momento de iniciar la adopción DevOps en la organización son:

   - Gestión:
     - Adoptar metodologías de trabajo ágiles.
     - Formar equipos multidisciplinarios para los proyectos.
     - Definir políticas de calidad.
   - Técnico:
     - Integración de herramientas.
     - Automatización de procesos.

   La organización debe potenciar sus procesos en términos de depurar pasos innecesarios, esperas, retrabajos y actividades manuales, apoyar el bienestar del equipo, entre otros, para alcanzar niveles altos y avanzados de madurez en la entrega de productos de software. Las áreas se deben ajustar y en algunos casos redefinir su forma de trabajo (configurarse para el éxito) para lograr encaminarse con la transformación y buenas prácticas que ofrece DevOps.

2. ¿Qué tan capaz es la organización de modernizar las aplicaciones existentes y permitir una entrega más rápida y la innovación?

   Este es uno de los aspectos que la organización debe poner mucha atención ya que la gran mayoría de sus productos están desarrollados en tecnologías y herramientas de más de 10 años de antigüedad. La recomendación es iniciar la adopción hacia nuevos desarrollos y proyectos recientes de máximo 3 años de

vida con el fin de ganar experiencia en los procesos técnicos particulares y luego tratar las aplicaciones legadas.

Un factor por destacar es que la compañía tiene claramente definido su stack tecnológico y el 88 % de todo su portafolio de aplicaciones está categorizado en dos paquetes tecnológicos, por lo que la modernización no será un proceso en extremo complejo, pero sí requerirá de horas de trabajo para cada proyecto.

3. ¿Qué tan capaz es la organización para adaptar prácticas en aspectos como la cultura, las herramientas y los procesos?

   CIAT y particularmente las áreas de desarrollo e infraestructura desde su dirección media y operacional tienen el propósito mejorar sus procesos, por lo que la adopción de prácticas DevOps no se estima que sea una tarea que genera mucha resistencia, debido a que hace un par de años las áreas ya tuvieron un intento de adopción DevOps que no se completó porque solo se le dio un enfoque técnico y no se contemplaron aspectos de cultura y procesos para la gestión del ciclo de vida del desarrollo de software.

4.2.4. Determinar el mejor camino a seguir

En este punto surgen las preguntas: ¿Por dónde iniciar?, ¿Cuáles son los pasos por seguir?

La adopción de una transformación digital como DevOps no ocurre de la noche a la mañana, este debe ser un proceso estructurado y definido que necesita ser fluido, a su vez comprendido por todos los involucrados.

Para el caso de estudio CIAT se presenta la propuesta de orden de adopción de las recomendaciones DevOps para la organización, acorde con la escala de clasificación de Tabla 3.2 qué propone tres fases, cada una se analiza con una estrategia top-down desde la dimensión hacia la capacidad, puntualizando recomendaciones y estrategias para abordar el nivel que complementan el plan de acción descrito para cada capacidad(para mayor detalle consultar el Anexo Plan de acción que contiene todos los planes de acción acorde a los resultados de la medición en el CIAT).

Las tres fases de adopción modelo son:

1. Reevaluar, adoptar e implementar los planes de acción de las capacidades clasificadas en oportunidades de mejora.

2. Ajustar y potenciar las capacidades clasificadas como aceptable.

3. Pulir, robustecer y direccionar las capacidades categorizadas como fortalezas hacia una filosofía de mejora continua.

Los cuales se detallan a continuación:

1. ***Abordar las oportunidades de mejora:*** La primera fase se fundamenta en priorizar la adopción e implementación de las recomendaciones generales por dimensión y del plan de acción sobre cada capacidad clasificada como oportunidades de mejora, la Tabla 4.1 detalla las dimensiones y capacidades clasificadas

en esta categoría, a las que se le realizó un análisis Top-down de dimensión a capacidad.

- **Entrega continua**, es la dimensión que presenta una debilidad muy alta, se debe abordar enfocándose en los siguientes factores clave:
  - Definir y adoptar una política de calidad de software.
  - Implementar procesos técnicos y herramientas para la automatización de pruebas.
  - Adoptar e implementar mecanismos técnicos que permitan la integración de herramientas que intervienen en el ciclo de desarrollo de software.
  - Adoptar procesos de automatización de despliegue permiten alcanzar una mejora en el nivel. • Estrategias que optimicen el uso de ramas o líneas de tiempo.
  - Entrega y despliegue de lotes de cambios pequeños.
  - Por último, reevaluar y potenciar todos los procesos denominados automáticos que intervengan durante el proceso de llevar un cambio desde el desarrollador al cliente final asegurando aspectos de calidad, seguridad y confiabilidad. • Fortalecimiento técnico y preventivo de los aspectos de seguridad desde el inicio del ciclo de vida del desarrollo para que sea una actividad altamente relevante antes, durante y en especial después de la entrega de un producto de software hacia los clientes finales.
- **Cultura**, potenciar el trabajo colaborativo entre los integrantes del equipo de trabajo (equipos multidisciplinarios que cubren representantes de desarrolladores, infraestructura, calidad, seguridad, entre otros) el objetivo es llegar a:
  - La toma de decisiones compartidas.
  - Integrar al trabajo colaborativo en equipos multifuncionales como calidad y seguridad de software.
  - Propiciar espacios de aprendizaje en por de fortalecer la motivación del equipo.

  Este trabajo debe acompañarse con el fortalecimiento de habilidades blandas como liderazgo de parte de los líderes de áreas para empoderar a todos los colaboradores a potenciar sus habilidades técnicas y de trabajo en equipo.
- **Arquitectura**, Es la dimensión con menor calificación promedio por lo tanto se debe abordar con un enfoque y visión desde niveles fundamentales para iniciar la adopción incremental de
  - Arquitecturas físicas como lógicas de bajo acoplamiento permitan gestionar ambientes de software a dinámicamente.

- Empoderar a los equipos en términos de dar libertad de que ellos mismos puedan proponer y definir con fundamentos la arquitectura (física, lógica) que mejor solucione los problemas particulares propios que afronte cada proyecto de software.

- **Gestión LEAN**, redefinir y adoptar mecanismos que permitan reducir los cuellos de botella en la gestión y aprobación de procesos, depurando procesos que no generen valor y sólo produzcan retrasos en la cadena de valor de la entrega de software.

- **Producto y procesos**, Se recomienda abordar esta dimensión basado en los siguientes aspectos:
  - Adoptar los mecanismos necesarios para recolectar eficientemente las retroalimentaciones de los usuarios finales (clientes) en todo el ciclo de vida del desarrollo del producto de software.
  - Pulir y potenciar los procesos clave como visibilizar el flujo de valor del trabajo realizado durante el ciclo de vida del desarrollo de software mejorando la transparencia y la comunicación entre todos los miembros del equipo de trabajo, otro punto importante en esta categoría es propiciar y apoyar la investigación y la cultura de aprendizaje para que los equipos realicen experimentos de nuevos componentes tecnológicos con el fin de ampliar el portafolio de herramientas y procesos para resolver futuros retos tecnológicos.

2. ***Ajustar y potenciar las capacidades aceptables:*** En esta fase el propósito es potenciar las capacidades que están clasificadas en esta categoría intermedia, la Tabla 4.2 presenta el grupo de capacidades clasificadas como Aceptables, y que representan la segunda fase del plan de adopción de recomendaciones y plan de acción.

   - **Gestión Lean y monitorización**, potenciar todos los aspectos de monitoreo tanto de infraestructura como de operación con el fin de prever y anticiparse a situaciones adversas que puedan generar indisponibilidad con los usuarios finales y no que el usuario final la identifique a través de interrupciones del servicio u operación del producto de software.

   - **Cultura**, enfocarse en términos de la gestión de los equipos y la toma de decisiones de la organización u área de implementación, aquí se busca que los equipos tengan iniciativas e inicien su transformación a equipos autogestionados. Los primeros pasos de este hito se logran a través del liderazgo de los directivos de la organización, donde se promueve la creación de equipos multidisciplinarios con propósitos de aprendizaje y trabajo colaborativo.

- **Producto y proceso**, abordar con un enfoque de gestión de trabajo ágil donde las tareas se distribuyen en lotes pequeños y puedan entregarse en periodos de tiempo cortos. En el objetivo principal para la primera etapa es entregar productos de valor en semanas y no en meses, en niveles superiores de madurez ir reduciendo la unidad de medida a días y unidades de tiempo menores.

- **La entrega continua**, fortalecimiento técnico y preventivo en el manejo de sistemas de control de versiones que faciliten la integración con otros procesos del ciclo de vida del desarrollo de software, fundamental para que sea el punto de inicio de la ejecución de los diferentes pipelines que implemente la organización.

3. ***Pulir y robustecer las fortalezas:*** Esta fase se debe abordar con la filosofía de que todo proceso u actividad no se puede quedar estática y debe ser tratada como un proceso de mejora constante. La Tabla 4.3 presenta las capacidades clasificadas como Fortalezas que se deben potenciar, se recomiendan ser tratadas en una tercera etapa después de reforzar las capacidades clasificadas como aceptables.

   - **Gestión Lean y monitorización**, enfocarse en alcanzar niveles de predicción proactiva de la salud de los componentes de la aplicación tanto a nivel físico y lógico, el objetivo es anticipar fallos por alta demanda o picos de trabajo y procesamiento. Por último y no menos importante son los mecanismos que permiten controlar la carga de trabajo paralelo para los equipos de trabajo, donde la definición de límites es clave para evitar el desgaste del equipo y no llegar a niveles de sobrecarga de trabajo que desborden la capacidad del equipo generando, retrasos prolongados y el incumplimiento de entregas. El punto clave es asegurar que el trabajo esté priorizado y equilibrado.

4.2.4.1. Aspectos a tener en cuenta

Acorde al análisis de los resultados obtenidos, la Tabla 4.4 presenta un resumen de los elementos más representativos que se deben abordar en CIAT para alcanzar los objetivos de adopción DevOps (Velocidad de despliegues, Reducción de tiempo para el despliegue).

- **Velocidad en la frecuencia de despliegues:** la primera etapa del proceso es mejorar la frecuencia de despliegues de meses a semanas y progresivamente llevar la unidad de medida a días

- **Reducción de tiempo en el despliegue**: la meta de la primera etapa de adopción es reducir el tiempo de horas a minutos, progresivamente reducir la

intervención externa, con el objetivo de llevar el proceso a un nivel de automatización completa.

Etapa	Mejora de los procesos	Herramientas para la automatización	Plataforma y entornos	Cultura
1	■ Iniciar la adopción de un marco de trabajo ágil ■ Adoptar Value Stream Mapping ■ Definir producto mínimo viable ■ Simplificar la gestión de cambios	■ Integrar herramientas. ■ Automatizar la compilación y la construcción (build). ■ Iniciar la adopción de pruebas	■ Primeros pasos de gestión de la configuración	■ Potenciar el trabajo colaborativo en el equipo ■ Ciclos de entrega completos ■ Bienestar del equipo
2	■ Adoptar y fortalecer el uso de un marco de trabajo ágil ■ Visibilizar el trabajo	■ Definir y adoptar una política de calidad de software ■ Integración continua	■ Infraestructura como código	■ Motivación del equipo de trabajo ■ Equipos multi-disciplinarios
3	■ Gestión de lotes y tamaño de trabajo	■ Despliegue continuo	■ Arquitectura flexible y por demanda	■ Cultura de trabajo orientada al rendimiento. ■ Liderazgo y potenciar las capacidades técnicas y culturales de los miembros del equipo

Tabla 4.4: Capacidades representativas por etapa

4.2.5. ¿CIAT si necesita DevOps?

Al evaluar los resultados de la medición con las métricas de la frecuencia de despliegue y el tiempo de despliegue de la organización, se evidencia el abanico de oportunidades para mejorar los procesos de entrega de productos de software. La adopción de las recomendaciones DevOps en el manejo de procesos a nivel técnico, gestión de procesos y cultura, le traería la oportunidad de mejorar muchísimo la eficiencia y oportunidad para brindar soluciones a los usuarios finales.

La adopción progresiva de prácticas DevOps en las fases propuestas ayudarán a la organización a mejorar aspectos como su eficiencia operativa, gestión de cambios, colaboración, entrega de productos de forma rápida, oportuna y confiable a los usuarios finales. Abordar prácticas de agilidad, apoyado de un trabajo cultural donde se propicie el trabajo colaborativo son los factores claves para alcanzar el éxito en la adopción DevOps, esto no se trata de solo implementar herramientas y automatizarlas.

Se entiende que CIAT no es una empresa dedicada al desarrollo de software al estilo de casa de software lo que no la exime del uso de buenas prácticas para el manejo de procesos acordes a las limitaciones de personal que pueda tener el área y la poca comunicación de los logros a toda la compañía.

A pesar de las limitaciones el área de desarrollo cuenta con altos niveles de aceptación de parte de sus principales usuarios funcionales, la encuesta NPS realizada a este público específico y que se detalla en el anexo Encuesta de medición de la percepción de los productos y los servicios prestados arrojó puntajes positivos por encima de 30 puntos NPS que son considerados de alto valor y muy buena aceptación de los servicios prestados.

4.3. Calidad de la evaluación

4.3.1. Riesgo de la validez

Al evaluar el modelo en un único caso de estudio, este puede presentar sesgos o elementos que requieran ajustarse para alcanzar un nivel de robustez que solo se obtiene probando el modelo en diferentes escenarios.

Al ser esta la primera versión de un modelo de evaluación de capacidades aplicado a una compañía real, pero conservando el enfoque académico, el modelo propuesto en este documento está abierto a futuros ajustes y mejoras para ámbitos académicos de la universidad o sus autores.

Con este trabajo se buscó medir la conveniencia de adoptar prácticas DevOps en una organización, para esto se propuso un modelo de evaluación DevOps genérico para compañías y personas poco familiarizadas con los conceptos y terminologías de estas prácticas.

4.4. Resumen del capítulo

Como caso de estudio de este documento, se realizó el diagnóstico de los procesos de entrega de software en la compañía Alianza Bioversity – CIAT por medio de la implementación del *modelo de capacidades* propuesto. La ejecución del modelo se realizó con un grupo representativo de las áreas de desarrollo e infraestructura de la organización logrando una visión general de los procesos desde dos puntos de vista diferente. Con los resultados del *modelo de capacidades*, se identificaron y clasificaron los aspectos débiles y fuertes acorde a las dimensiones y capacidades evaluadores para que sean abordados a través de un plan de recomendaciones.

Una vez obtenido el resultado de evaluación y evaluando las oportunidades de mejora en cultura, procesos y herramientas, se llegó a conclusión que una transformación DevOps se puede aplicar a cualquier compañía, independiente de su línea de negocio por lo que CIAT como organización se verá beneficiado en mejoras en tiempos y frecuencias de entrega de productos.

La adopción DevOps se planteó basado en lo propuesto por Sharma (2017), donde se definen 3 etapas de adopción:

1. **Identificar el estado objetivo:** A partir de una medición interna con la información disponible en CIAT para el año 2021, se determinó encaminar las recomendaciones de adopción para los indicadores de negocio:
   - Velocidad en la frecuencia de despliegues
   - Reducción de tiempo en el despliegue
2. **Entender el estado actual:** Para identificar el estado de madurez de los procesos de desarrollo de software en una organización se utilizó el *modelo de capacidades* propuesto para el caso particular CIAT. Utilizando la escala de clasificación se presentan los resultados más importantes de cada categoría:
   - **Oportunidades de mejora**
     - Procesos de integración y automatización.
     - Gestión de cambios y metodología de desarrollo.
     - Silos y falta de comunicación entre las áreas.
     - Arquitectura de bajo acoplamiento.
   - **Aceptables**
     - Cultura de trabajo orientada al rendimiento (Generativa).
     - Gestión de proyectos y herramientas para visibilizar el trabajo.
     - Liderazgo y motivación del equipo de trabajo Metodologías de trabajo.
   - **Fortalezas**
     - Monitoreo preventivo y proactivo.
3. **Determinar el mejor camino a seguir:** Acorde a los resultados se definió abordar la adopción DevOps en un modelo de tres fases:

1 Reevaluar, adoptar e implementar los planes de acción de las capacidades clasificadas en oportunidades de mejora

2 Ajustar y potenciar las capacidades clasificadas como aceptables

3 Pulir, robustecer y direccionar las capacidades categorizadas como fortalezas hacia una filosofía de mejora continua

Actividades para mejorar las métricas de los Indicadores de negocio:

- **Velocidad en la frecuencia de despliegues:** la primera etapa del proceso es mejorar la frecuencia de despliegues de meses a semanas y progresivamente llevar la unidad de medida a días
- **Reducción de tiempo en el despliegue**: la meta de la primera etapa de adopción es reducir el tiempo de horas a minutos, progresivamente reducir la intervención externa, con el objetivo de llevar el proceso a un nivel de automatización completa.

La adopción progresiva de prácticas DevOps en las fases propuestas ayudarán a la organización a mejorar aspectos como su eficiencia operativa, gestión de cambios, colaboración, entrega de productos de forma rápida, oportuna y confiable a los usuarios finales.

Capítulo 5

Conclusiones y trabajos futuros

5.1. Conclusiones

El propósito planteado en este trabajo era identificar cómo la filosofía y buenas prácticas en transformación digital que ofrece DevOps puede ser adoptado en cualquier tipo de compañía independiente de su finalidad y la razón de ser, en el caso particular en una organización como CIAT. Para ello se propuso un modelo de evaluación multidimensional de prácticas DevOps que se enfoca en los aspectos técnicos y de gestión de procesos más representativos en la entrega de productos de software (Forsgren et al., 2018).

La propuesta de definir un marco de evaluación propio surgió después de usar y evaluar un grupo de modelos de madurez DevOps del mercado, donde predominó el uso de términos técnicos y complejos, que para compañías o usuarios nuevos en las prácticas DevOps son confusas o desconocidas, generando ambigüedades y malas interpretaciones en los conceptos. El modelo propuesto se diseñó a través de preguntas indirectas usando terminología fundamental propia de prácticas de desarrollo de software que buscan medir el uso y manejo de procesos DevOps en una organización. El modelo presentado permitió evaluar las capacidades DevOps de CIAT y de cualquier organización por medio de un cuestionario que abarcó dimensiones culturales, técnicas, de gestión de procesos que impulsan mejoras en el rendimiento de la entrega de software.

Cada capacidad descrita en el modelo de referencia se expresó en el marco propuesto por medio de una pregunta con opciones de respuesta cerradas y acotada a una escala de madurez, por último, a cada opción de respuesta se le elaboró un plan de acción para potenciar la capacidad y llevarla al siguiente nivel de madurez. La principal finalidad de esta asociación es que el propio marco de evaluación propuesto entregue las recomendaciones expresadas como plan de acción a partir de la opción de respuesta seleccionada de cada pregunta.

Las dimensiones en las que se enfocó el marco de evaluación son: Entrega conti-

nua, Arquitectura, Producto y proceso, Gestión Lean monitorización y Cultura. Cada una de estas dimensiones está compuesta por veinticuatro capacidades que Forsgren et al. (2018) propone como elementos claves para mejorar el rendimiento de la entrega de software. Para cada capacidad se definió una escala de cinco niveles de madurez (Sin experiencia, Novato, Intermedio, Destacado, Avanzado) que establece la escala de evolución de las capacidades. Con el fin de definir un orden de prioridad para adopción de los planes de mejora, cada nivel de la escala de madurez se clasificó en tres categorías (Oportunidades de mejora, Aceptable y Fortalezas).

Los tres primeros niveles de madurez se categorizan como oportunidades de mejora, estos son los primeros elementos que deben tratarse para definir los fundamentos y la filosofía de transformación cuando se inicie la adopción, el cuarto nivel de madurez se categoriza como Aceptable donde ya existe una solidez que se debe seguir potenciando, por último el quinto nivel se categoriza como Fortalezas, aquí el nivel de solidez y madurez es elevado, se debe tratar a través de una filosofía de aprendizaje continuo.

Para el caso de estudio CIAT, al promediar todas las capacidades evaluadas se obtuvo una nota general de 1.96, posicionándolo dentro de la categoría Oportunidad de Mejora, lo que quiere decir que a nivel general la compañía tiene un nivel de desarrollo principiante, con poco conocimiento y experiencia de las prácticas organizaciones y recomendaciones que ofrece DevOps, la cuales se deben potenciar.

Al momento de iniciar la adopción DevOps en la organización, se deben priorizar las dimensiones Entrega Continua, Arquitectura y Cultura, ya que más del 80 % de sus capacidades están clasificadas como Oportunidad de Mejora.

CIAT como organización tiene el 75 % de las capacidades evaluadas con necesidad de atención prioritaria, lo que indica que el proceso de adopción se debe enforcar en fortalecer las bases teóricas y técnicas de todos los eslabones del equipo de trabajo que intervienen en el ciclo de vida de entrega de productos de software.

Las organizaciones deben ser asertivas al momento de definir el personal que va a participar en la medición. Este grupo idóneo de personas debe conocer y participar activamente en los procesos de desarrollo y entrega de productos de software para poder reflejar con mayor exactitud la realidad de las prácticas empresariales de la organización.

Después del análisis de los resultados, se concluye que DevOps es una práctica en transformación digital que le conviene a CIAT. A partir de las recomendaciones y buenas prácticas DevOps la organización tiene muchos aspectos a mejorar y potenciar para lograr una mejora representativa en los tiempos de entrega de productos de software a sus usuarios finales. A través de un proceso de adopción incremental, el objetivo de la primera etapa es reducir los tiempos de desarrollo y entrega productos de meses a semanas y llevar los tiempos de despliegue de horas a minutos. Progre-

sivamente y en etapas posteriores, cuando se alcancen nuevos niveles de madurez, el objetivo de la filosofía de mejora continua es reducir los tiempos de entrega a escala de días y los despliegues a unos pocos minutos, apoyados en procesos automáticos confiables y seguros.

5.2. Lecciones aprendidas

El desarrollo de esta investigación me permitió conocer cómo DevOps es una filosofía que agrupa buenas prácticas de industria para mejorar la entrega de productos de software a través de un cambio cultural que abarca personas, procesos y herramientas tecnológicas.

El modelo de capacidades propuesto fue diseñado a partir de la consolidación de muchas fuentes de información especializadas en temas DevOps y agilidad, se resaltan dos fuentes que han sido esenciales para el desarrollo del presente trabajo: el resultado de la investigación del libro Accelerate Forsgren et al. (2018) y del programa de investigación DORA's de Google. Estas fuentes de información fueron la base de fundamentación para la definición de las preguntas del modelo de capacidades, la escala de respuestas y especialmente la construcción del plan de acción para cada opción de respuesta o grupo de ellas.

Procesar y analizar los resultados del modelo de evaluación propuesto fue un gran reto que requirió establecer estrategias de consolidación, análisis y presentación de datos. Especialmente patrones y buenas prácticas para exteriorizar la información como es el caso de estrategia Top-Down que se utilizó para presentar la información de general a lo específico en un barrido de izquierda a derecha.

La adopción de DevOps en una organización es un proceso que debe ser abordado de una forma consciente y acorde a sus necesidades puntuales, no existe un modelo genérico o receta de adopción única, por lo que cada compañía debe ser evaluada mediante un mecanismo que permita estimar el estado de sus procesos de entrega de software, así poder definir y estructurar el plan de adopción.

La adopción DevOps en una organización tiene muchas formas de iniciar, lo más recomendable es arrancar con pilotos incrementales y focalizados ya que los cambios culturales y de procesos no son inmediatos y requieren tiempo para planearse y adoptarse en consenso con las áreas implicadas. Se resalta que los cambios a nivel de cultura y gestión de procesos son cruciales a la hora de iniciar la automatización de los procesos técnicos ya que "no se puede automatizar el desorden".

Un aspecto a resaltar durante el desarrollo del presente documento fue la grata experiencia de tener dos evaluadores complementarios, cada uno desde su experticia me brindó el acompañamiento constante y paciente con las herramientas necesarias para aterrizar conceptos, abordar dudas en temas que en un inicio eran completamente

nuevos para mí, de igual forma este camino recorrido me permitió afinar mi análisis crítico ante situaciones, especialmente al momento de la redacción y elaboración de este documento.

5.3. Trabajos futuros

El presente documento nació con el propósito de evaluar la conveniencia de DevOps en una organización como CIAT. Dentro de este proceso se definió y estructuró un marco de evaluación propio basado en unos resultados de investigación confiables, este modelo de evaluación de capacidades se implementó por medio de un cuestionario web que un grupo definido de presentantes de la organización (Desarrollo, infraestructura, seguridad y calidad de software), por último, la limpieza, consolidación y análisis de los resultados a través de gráficos se desarrolló en una herramienta de inteligencia de negocios que permitió relacionar los resultados de la encuesta con el modelo propuesto.

Como trabajos futuros se plantea integrar el proceso de captura de información del modelo de capacidades representado en una encuesta con el proceso de limpieza de datos y generación de los gráficos en una aplicación web que sirva de mecanismo de fácil acceso para realizar la evaluación DevOps en cualquier compañía. De esta forma, se pueden implementar mecanismos que unifiquen y centralicen el modelo propuesto con el análisis de datos, abriendo la posibilidad de evolucionar fácilmente el modelo de evaluación propuesto, agregando o modificando los elementos que lo componen (por ejemplo, preguntas, opciones de respuesta, planes de acción y escalas de calificación y categorización).

Al ser la primera versión de un mecanismo de evaluación de capacidades DevOps y dado el alcance definido en este documento, el modelo propuesto sólo fue aplicado en un único caso de uso. Por esta razón, se recomienda que el modelo sea usado en diferentes compañías con el fin de poder evaluar los resultados obtenidos y así recolectar retroalimentaciones que permitan fortalecer los componentes del artefacto de evaluación.

Se propone, crear un espacio de trabajo compartido para que la comunidad universitaria pueda utilizar el modelo propuesto en un marco de trabajo colaborativo donde puedan proponer mejoras a los componentes y estos siempre estén disponibles para su uso.

Apéndice A

Encuesta de medición de la percepción de los productos y los servicios prestados

El objetivo de esta etapa fue poder medir la percepción de satisfacción que tienen los usuarios finales de la organización sobre el portafolio de aplicaciones y calidad de servicio que ofrece el área de desarrollo administrativo (Unidad de integración de sistemas). Para esta tarea se utilizó la encuesta como método directo de recolección de información, en donde se aplicó la métrica NPS para medir el grado de satisfacción de los usuarios finales.

La encuesta implementada fue pensada para ser corta y concisa, solo consta de 6 preguntas que busca identificar qué tan útil son los servicios prestados para el día a día del usuario final y el nivel de satisfacción general de los servicios recibidos desde una perspectiva general y en los últimos 6 meses. Por último, dado que la encuesta será contestada de forma anónima, se agregaron dos preguntas descriptivas con el fin de tener un perfil que indique el área a la que pertenece el empleado y el tiempo que tiene vinculado a la organización.

Las preguntas de satisfacción (3,4 y 5) están diseñadas acorde a la métrica NPS por lo que sus opciones de respuesta están en una escala de 0 a 10.

Preguntas encuesta de percepción de la unidad de Integración de sistemas

1. ¿A qué área organizacional perteneces?
   (menú desplegable con las áreas)
2. ¿Cuántos años tienes en la organización?
   - Menos de 1 año
   - Entre 1 y 3 años
   - Entre 3 y 5 años
   - Entre 5 y 10 años

- Más de 10 años

3. En una escala de 0 a 10, donde 10 es "Completamente" y 0 es "En nada".

   ¿Qué tanto la unidad de Integración de sistemas le facilita el trabajo diario?

   (Escala de 0 a 10)

4. 4. En una escala de 0 a 10, donde 10 es "Completamente satisfecho" y 0 es "Completamente insatisfecho".

   ¿Qué tan satisfecho está con el servicio prestado por la unidad de Integración de sistemas?

   (Escala de 0 a 10)

5. 5. En una escala de 0 a 10, donde 10 es "Completamente satisfecho" y 0 es "Completamente insatisfecho".

   ¿Qué tan satisfecho está con el servicio prestado por la unidad de Integración de sistemas en los últimos 6 meses?

6. ¿Tiene usted algún comentario adicional o alguna sugerencia que nos ayude a mejorar algún aspecto del servicio?

   (respuesta abierta)

Resultados:

Para esta encuesta se tuvo en cuenta al menos un representante o líder funcionales del portafolio de aplicaciones periféricas ofrecidos por la unidad de integración de sistemas a las demás áreas de soporte de la organización, definiendo una población objetivo de 20 usuarios funcionales, de los que se obtuvo respuesta de 15 de ellos, equivalente un 75 % de participación de la población objetivo.

La implementación de la encuesta se realizó en un formulario de encuesta de office 365 que soporta preguntas tipo NPS, obteniendo los siguientes resultados.

Los cuales pueden ser consultados en el siguiente link https://tinyurl.com/muer2jrd (ver figure A.1)

1. La pregunta número 1, asociada a identificar el área funcional a la que pertenece el encuestado, se obtuvo respuesta de 9 diferentes áreas funcionales, donde las Americas Regional HR y Facilities and Operations representan cada uno equivalente al 20 % de los encuestados. (Ver figure A.2)

2. La pregunta 2, cuya finalidad era conocer la antigüedad de los encuestados en la organización, se obtuvo que el 60 % de los encuestados tienen más de 10 años vinculados a la compañía

3. La pregunta 3, se obtuvo una medida de opción NPS de 34 %, el cual es un indicador de que positividad en la satisfacción al cliente, indicando que el área si facilita el trabajo diario de un grupo de empleados. (ver Figura 6-4 Encuesta de percepción respuestas a la pregunta 3)

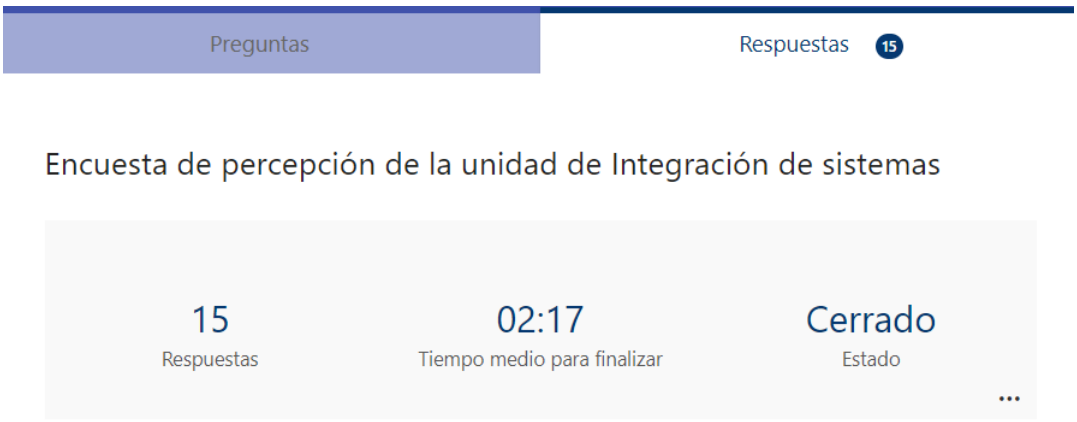

Figura A.1: Resultado encuesta de percepción de los productos y los servicios prestados por la unidad de Integración de Sistemas del departamento de Tecnología de Información

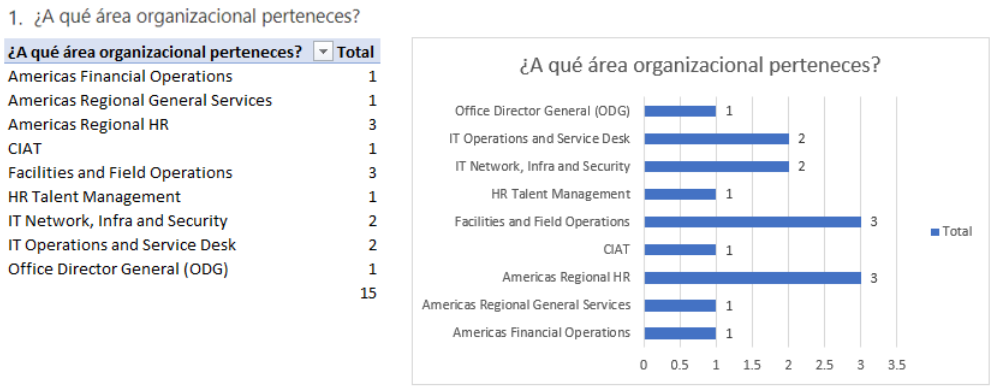

Figura A.2: Encuesta de percepción respuestas a la pregunta 1

4. Para la pregunta 4, se obtuvo una medida de opción NPS de 53 %, el cual es un indicador alto de que positividad en la satisfacción al cliente frente a los servicios recibidos, indicando históricamente el área ofrece servicios que satisfacen las necesidades de los usuarios finales.

5. La pregunta 5, se obtuvo una medida de opción NPS de 46 %, el cual es un indicador alto de que positividad en la satisfacción al cliente frente a los servicios recibidos en los últimos 6 meses, contrastando con la NPS de la pregunta 4, la satisfacción bajó en 7 %, manteniendo la precepción en valores considerados altos. (ver Figura 6-6 Encuesta de percepción respuestas a la pregunta 5).

6. La sexta y última pregunta de esta encuesta fue un espacio abierto para recibir retroalimentación de parte del encuestado. Se recibieron comentarios positivos sobre la percepción de área, resaltando que en los últimos 6 meses se está presentando retraso en la respuesta oportuna a los servicios solicitados y está bajando el NPS del área. Tal y como se muestra en las imágenes Figura 6-7 Encuesta

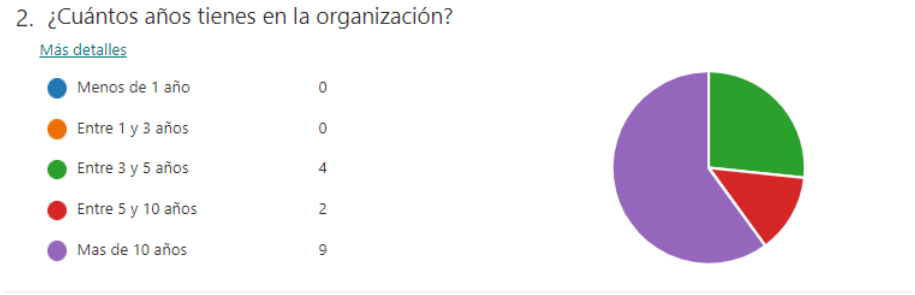

Figura A.3: Encuesta de percepción respuestas a la pregunta 2

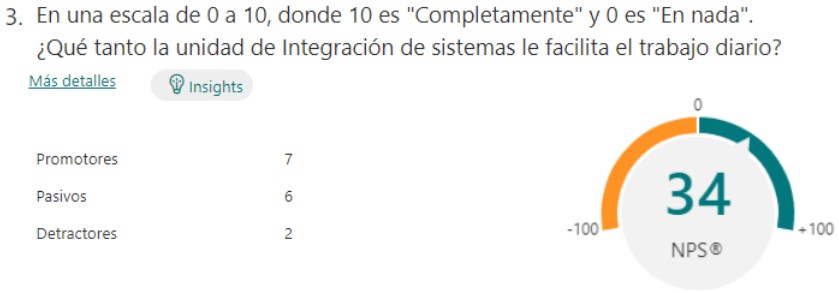

Figura A.4: Encuesta de percepción respuestas a la pregunta 3

de percepción respuestas a la pregunta 6 y Figura 6-8 Encuesta de percepción respuestas a la pregunta 6, comentarios

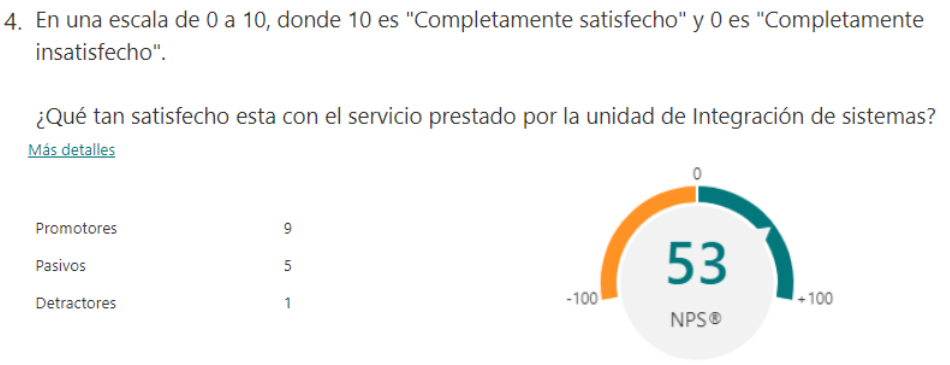

Figura A.5: Encuesta de percepción respuestas a la pregunta 4

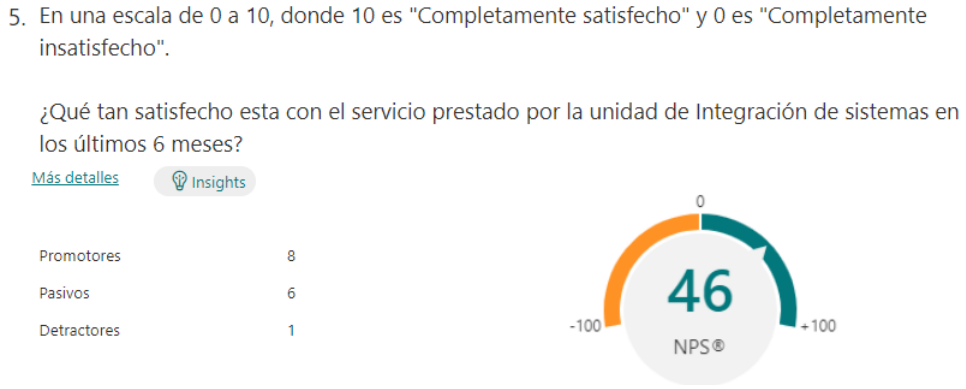

Figura A.6: Encuesta de percepción respuestas a la pregunta 5

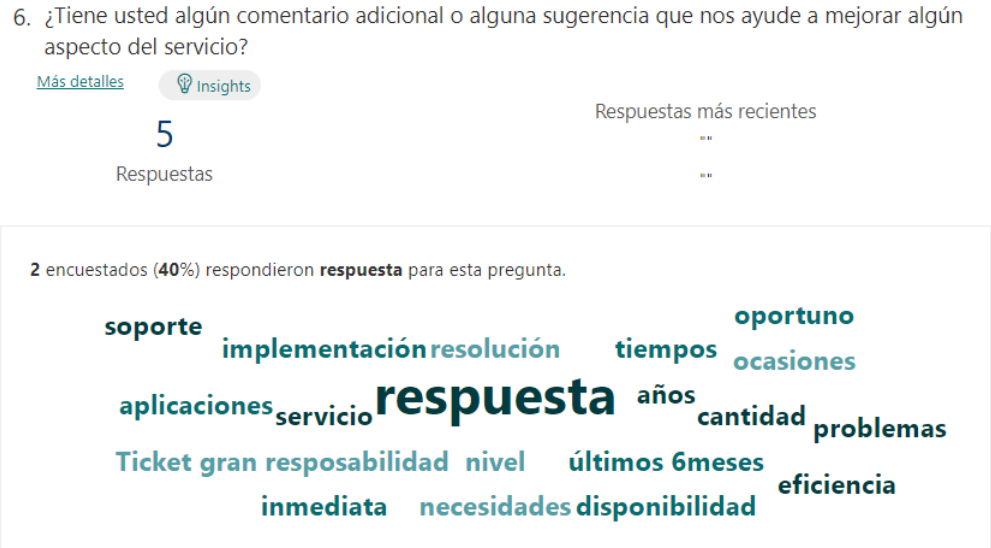

Figura A.7: Encuesta de percepción respuestas a la pregunta 6

6. ¿Tiene usted algún comentario adicional o alguna sugerencia que nos ayude a mejorar algún aspecto del servicio?

5 Respuestas

1	anonymous	En general he estado muy satisfecha con el servicio, sin embargo en los últimos 6meses siento que los tiempos de respuesta se están demorando y en ocasiones no tengo respuesta.
2	anonymous	Tienen muy poco personal para la cantidad de aplicaciones que manejan y la gran resposabilidad que ustedes tienen,
3	anonymous	Mantengan ese nivel de eficiencia y disponibilidad que siempre mantienen a nuestras necesidades.
4	anonymous	seguir prestando el soporte oportuno como lo han venido prestando durante todos estos años.
5	anonymous	No se si tenga que ver, pero con esa implementación del Ticket, la resolución de los problemas se ha vuelto más lenta que antes. Es frustrante cuando la necesidad requiere una respuesta inmediata.

Figura A.8: Encuesta de percepción respuestas a la pregunta 6, comentarios

Apéndice B

Modelo de capacidades para la evaluación DevOps

B.1. Preguntas del modelo de capacidades

Cada una de las preguntas del siguiente cuestionario está orientada a cubrir una de las dimensiones y capacidades que apalancan mejoras en el rendimiento en la entrega de software descritos en el libro (Forsgren et al., 2018) con el que se aborda la adopción de prácticas DevOps a partir de medir el contexto actual de la organización para mejorar y progresar de forma continua sus procesos de entrega de software.

Las preguntas de este artefacto de medición de capacidades están diseñadas para el segmento profesionales TI con conocimientos previos o no de las practicas DevOps, por lo que no se pregunta directamente sobre una práctica puntual, sino que se aborda desde el punto de vista de las características y funcionalidades del desarrollo de software y su ciclo de vida, así de evitar sesgos o malas interpretaciones.

Por otro lado, en este cuestionario de evaluación se abordan características que no son medibles por lo que son abordados de forma abstracta y así lograr una aproximación de la a partir de unidades definidas o recomendaciones de industria.

Cada pregunta, está en definida para ser calificada en una escala de 1 a 5 con lo que se permite obtener una medida del estado de cada capacidad.

Por cada pregunta, se tiene una justificación del por qué se aborda esta pregunta, las opciones de respuesta en la escala definida, el plan de acción por cada respuesta o grupo de ella y una categorización donde de relaciona la pregunta al modelo de capacidades de referencia.

1. **¿La compañía utiliza sistemas de control de versiones?**

   Justificación

   Se busca, evaluar el uso de un sistema de control de versiones como Git o subversion en la organización. Se tiene en cuenta que se incluya el código fuente, configuraciones de la aplicación, scripts de automatización para construcción y configuración de entornos, que son los elementos clave a la hora de iniciar procesos de automatización.

La escala busca reflejar que tan difundido esta el uso de un sistema de control de versiones en la compañía.
Se debe tener en cuenta el uso de servicios como Github, GitLab o Bitbucket para gestionar y centralizar el uso de sistemas de control de versiones.

Guardar la configuración del sistema y las aplicaciones en el control de versiones esta correlacionado con el rendimiento de la entrega de software que solo guardar el código fuente de la aplicación (Forsgren et al., 2018).

Opciones de respuestas

1 No usa o no conoce sobre sistemas de control de versiones.

2 Hace control de versiones de código de forma manual.

3 Usa parcialmente herramientas de control de versiones localmente

4 Utiliza una herramienta centralizada de control de versiones sin mayor profundización en sus herramientas.

5 Cuenta con un sistema de control de versiones donde almacena la aplicación, configuraciones y scripts.

Plan de acción

1 y 2. Se requiere la adopción de un sistema de control de versiones como Git o Subversion con el fin de gestionar la creación, actualización, eliminación y acceso de los archivos del código fuente, configuraciones y otros documentos que hagan parte del proyecto, en especial los elementos en los que se basan los procesos de compilación, incluyendo las herramientas y los entornos de los que dependen.

3. Se debe avanzar para tener sistemas de control de versiones distribuido y administrado con sistemas de alojamiento web como por ejemplo Github, GitLab, Bitbucket que faciliten la integración con otros procesos del ciclo de vida de las aplicaciones con el fin de poder iniciar la ejecución de tareas de compilación, construcción de las aplicaciones.

4. Aprender y adoptar conceptos de ramas y estrategias de mezcla de cambios con el fin de aprovechar el potencial de un sistema de control de versiones para que los equipos puedan confirmar, comparar, combinar y restablecer revisiones de la aplicación con mayor facilidad y así poder definir los flujos claros para el uso y ejecución de scripts de automatización y configuración del sistema.

5. Se tiene adoptado las prácticas de referencia de DevOps para el manejo de un sistema de control de versiones, siendo el punto de partida de flujos de tareas integradas y automatizadas que permite llevar cambios en el código fuente hasta tener un producto listo para ser usado por los usuarios finales. El equipo de trabajo debe adoptar la cultura de mejora continua, evaluando periódicamente los procesos implementados con el fin de buscar oportunidades de mejora o nuevas formas de realizar las tareas, teniendo en cuenta que los sistemas de control de versiones deben ir más allá del código fuente de las aplicaciones, es necesario

gestionar factores como, la configuración del sistema e infraestructura, la configuración de la aplicación, los comandos de automatización de la construcción y configuración.

Fuente:

- DORA's research program, DevOps tech: Version control (DORA's, a).
- Accelerate, (Forsgren et al., 2018).

Tipo: Herramientas

Subtipo: Automatización

Categoría: Entrega continua

Capacidad: Utilizar control de versiones para todos los artefactos de producción

Ultima modificación: 20220311

2. **¿Cómo es el proceso de despliegue de soluciones de la compañía?**

   Justificación

   Se busca determinar el grado de automatización del despliegue de soluciones en la organización, va desde un proceso completamente manual sin ninguna revisión, hasta procesos completamente automáticos.
   Se busca determinar si se hacen o no tareas de control de calidad antes de cada despliegue

   El despliegue o entrega de software son los pasos y capacidades que se deben ejecutar para pasar los cambios de una aplicación desde un sistema de control de versiones hasta el entorno de producción o a los usuarios finales para su uso de forma rápida y segura.

   Opciones de respuestas

   1 El equipo de desarrollo genera manualmente las nuevas versiones de las aplicaciones y ellos mismos la despliegan independientemente del ambiente (pruebas o producción).

   2 El equipo de desarrollo genera nuevas versiones de las aplicaciones y las envían a operaciones para que ellos las desplieguen en su sitio final (normalmente ambientes productivos).

   3 El equipo de desarrollo genera la nueva versión del producto usando scritps y herramientas especializadas sin integración de pruebas y posteriormente la envía a operaciones para su despliegue.

   4 Los equipos de desarrollo y operaciones tienen tareas de generación de producto final y despliegue, pero no están integradas o dependen de aprobación para su despliegue.

5 La construcción y despliegue se hacen completamente automatizados, se ejecutan tareas de verificación que no requieren ninguna intervención, solo se necesita que un agente de inicio al proceso.

Plan de acción

1 Iniciar la adopción de herramientas que automaticen la compilación y generación del producto final, ejemplo adopción de Maven, Gradle para ambientes java, Nuget para C#, adicionalmente integrar procesos de aseguramiento de calidad, como son pruebas manuales (unitarias y de aceptación) por último definir roles y responsabilidades en común acuerdo entre desarrollo y operaciones de quienes son los responsables a la hora de hacer los despliegue.

2 Acordar y adoptar un proceso de responsabilidades compartidas y colaborativo entre las áreas de desarrollo e infraestructura desde la etapa inicial del proyecto en la que se definan roles y responsabilidades, y que herramientas se van a utilizar para las diferentes etapas del ciclo de vida de la aplicación, como gestionar la compilación, construcción, ejecución de pruebas y despliegue en producción de la aplicación.

3 Acordar y adoptar un proceso colaborativo de responsabilidades compartidas entre desarrollo, infraestructura, calidad y seguridad en donde se definan las herramientas a utilizar en cada paso enfocado a lograr unir cada fase en una secuencia de pasos (pipeline), definiendo que scripts se deben ejecutar para compilar y generar el producto final, cuando se deben ejecutar las pruebas (integración y aceptación) del sistema (al menos manuales) y cual es criterio de un producto listo para pasar a producción.

4 Adoptar una herramienta de orquestación de procesos con el que se pueda integrar las tareas entre desarrolladores, pruebas y operaciones para que el despliegue de una aplicación se sea en un flujo de tareas ágil, repetible y confiable donde se minimice cualquier intervención humana ya sea iniciando procesos o aprobando tareas. El despliegue se debe estructurar como una cadena de procesos (Pipeline) que va desde tomar el código fuente y demás elementos de configuración desde un sistema de control de versiones, compilarlo, ejecutar pruebas (ejemplo de unitarias, de integración, de aceptación, etc.), ejecutar tareas complementarias de calidad y ambientes y por último desplegar el producto resultante en el ambiente de producción con la menor intervención humana.

5 El despliegue continuo es la capacidad que permite entregar soluciones de software a los usuarios finales de forma rápida y confiable. En este nivel, la compañía debe adoptar una cultura de mejora constante del proceso de automatización para desplegar aplicaciones en entornos de prueba y producción sin ningún tipo de intervención, donde la simplificación y flexibilidad de la arquitectura juega un papel relevante para garantizar el proceso. Se debe tener en cuenta aspectos como, almacenar la configuración del sistema y scripts de configuración en un sistema de control de versiones, el

objetivo es evitar la intervención manual. El despliegue continuo requiere del trabajo conjunto de equipos multidisciplinarios, integrados por desarrolladores, calidad, seguridad e infraestructura, entre otros para obtener tareas agiles, repetibles y confiables.

Fuentes:

- The DevOps Handbook First Edition (Kim et al., 2016).
- The DevOps Handbook Second Edition, 2021 (Kim et al., 2021).
- Accelerate, Forsgren 2018 (Forsgren et al., 2018).
- DORA's research program, DevOps tech: Continuous delivery: (DORA's, p).
- RedHat: (Redhat, 2018).

Tipo: Herramientas

Subtipo: Automatización

Categoría: Entrega continua

Capacidad: Automatizar el proceso de despliegue

Ultima modificación: 20220311

3. **¿De qué manera la compañía realiza la compilación, construcción y empaquetamiento del producto final?**

   Justificación

   La intención es determinar qué tan automático es el proceso de construcción y empaquetamiento de las soluciones finales.

   Se evalúa procesos manuales donde incluso las librerías se cargan manualmente, hasta el uso de herramientas de gestión de compilación e integrado para automatizar el proceso de construcción y despliegue de los proyectos de software.

   Opciones de respuestas

   1 El desarrollador o líder de desarrollo, ejecuta un proceso manual para obtener el producto final a desplegar

   2 El producto final se genera mediante scripts que un desarrollador ejecuta localmente.

   3 Se utiliza una herramienta de automatización de compilación (ejemplo Maven, Gradle) para la construcción completa del proyecto mediante la ejecución de scripts automáticos.

   4 Se utiliza una herramienta de automatización y orquestación integrada en una secuencia de pasos que permite la compilación, ejecución de pruebas y construcción del proyecto en un empaquetado final listo para desplegar.

5 El proceso de compilación hace parte de un proceso automático de integración basado en secuencias de pasos que permiten compilar, construir y desplegar en un ambiente los proyectos de software

Plan de acción

1. y 2. Se debe iniciar la migración de las aplicaciones a una herramienta especializada para la gestión de proyectos de desarrollo de software por ejemplo Maven, Gradle, Nuget, etc, iniciando con la preparación del proyecto, que consiste en crear un inventario de dependencias y definir la fuente o repositorio donde se obtendrán, segundo, adaptar el proyecto acorde a las especificaciones particulares de la herramienta elegida para que el proyecto se pueda gestionar a partir de comandos repetibles y así cubrir las etapas de compilación, construcción, pruebas y empaquetamiento del proyecto y usarse en procesos posteriores.

3. A partir de tener los proyectos gestionados bajo una herramienta especializada (ejemplo, Maven, Gradle, Nuget, etc.), se debe iniciar con la incorporación de procesos complementarios durante las etapas de construcción del proyecto, por ejemplo, ejecución de pruebas unitarias automáticas durante las etapas de compilación, construcción, ejecución de análisis de código estático, etc. En este punto se recomienda la implementación de una herramienta de orquestación de integración continúa, ejemplo Jenkins, CircleCI, Travis CI, Bamboo, AzureDevOps, etc.

4. Se debe asegurar que todas las etapas del proceso de desarrollo de una aplicación estén integrados y automatizados en secuencias de pasos (producto de un consenso interdisciplinario entre desarrollo, infraestructura, calidad de software, etc) que inicia desde la automatización de la compilación, ejecución de pruebas y aseguramiento de calidad y por ultimo despliegue en entornos de producción, por ejemplo, el proceso debe iniciar desde tomar el código fuente desde un sistema de control de versiones, compilarlo, ejecutar pruebas (unitarias, funcionales, de integración, etc) y desplegar el producto final en un ambiente definido.

5. La integración continua es un proceso que la organización debe adoptar con la filosofía de mejoramiento constante, las tareas de integración y automatización del ciclo de vida de las aplicaciones se deben revisar periódicamente, con el fin de robustecer o agregarle nuevas capacidades al proceso de entrega de software. En este nivel los elementos claves que se deben asegurar son: tener un proceso de compilación automatizado, tener un conjunto de pruebas automatizadas, tener un sistema de orquestación y automatización basados en secuencias de tareas (pipelines) que pueden integrar un sinfín de procesos automáticos que van más allá de la compilación y el despliegue del proyecto. La integración continua requiere del trabajo conjunto de equipos multidisciplinarios, integrados por desarrolladores, calidad, seguridad e infraestructura, entre otros para obtener tareas agiles, repetibles y confiables.

Fuente:

- Accelerate, Forsgren 2018 (Forsgren et al., 2018).
- DORA's research program, DevOps tech: Continuous integration (DORA's, m).

Tipo: Herramientas

Subtipo: Automatización

Categoría: Entrega continua

Capacidad: Implementar integración continua

Ultima modificación: 20220311

4. **¿La compañía usa alguna estrategia de gestión de la configuración?**

   Justificación

   Se busca conocer como la compañía gestiona los cambios en el código fuente de las aplicaciones que se desarrollan.

   Se tienen en cuenta aspectos como no hacer control de cambios, hasta políticas de control de ramas y línea base compartida

   La gestión de cambios dentro de un proyecto es un factor clave para garantizar rendimiento, respaldo y productividad dentro de un proyecto de software, no gestionar los cambios en el código fuente y demás archivos propios del proyecto dificulta el trabajo colaborativo entre todos los miembros del equipo ya que no permite un flujo de trabajo ágil y demás de que imposibilita la automatización de procesos y calidad de software.

   El DevOps Research and Assessment (DORA) muestra que los equipos alcanzan altos niveles de entrega y rendimiento siguen las siguientes practicas:

   - Tener tres ramas activas, o menos, en el repositorio de código de la aplicación
   - Combinar las ramas con el tronco al menos una vez al día
   - No tener congelación de código ni fases de integración

   Opciones de respuestas

   1 No se controlan los cambios.

   2 Los cambios se controlan manualmente.

   3 Se utiliza un sistema de control de versiones como git, subversión, cvs, etc.

   4 Se utiliza la estrategia de rama de producto estable (main/master) y producto en desarrollo (dev/develop).

   5 Se es riguroso, se utilizan métodos y estrategias de integración de versiones de código basados en ramas, separando la línea principal del producto estable (main/master) de las líneas de desarrollo (dev/devlop) incluso ramas exclusivas para funcionalidades especificas (features), se adoptan estrategias de fusión de cambios como pull request al repositorio principal.

Plan de acción

1 Se debe iniciar la adopción de un sistema de control de versiones, un sistema de gestión documental o al menos definir una metodología de trabajo en donde desarrolladores puedan registrar los cambios en el código fuente, archivos de configuración y demás documentos que hagan parte del proyecto de manera que los cambios se puedan consolidar en revisiones en donde debe conservar quien y cuando se hizo algún cambio sobre los archivos.

2 Adoptar un sistema de control de versiones como Git, Subversion y Mercurial, etc. que permita controlar fácilmente los cambios en código fuente y demás archivos propios del proyecto durante todo su ciclo de vida, sin desgastar tiempo y esfuerzo del equipo de trabajo en tareas manuales de consolidación de cambios. La gestión de la configuración es un factor determinante en el rendimiento de los equipos de trabajo y un elemento clave en la entrega continua de productos, por lo que este es el primer paso para mejorarlo.

3 Adoptar estrategias de ramas o división de funcionalidades con el fin de separar el código que esta desplegado en producción de las características que están en etapa de desarrollo, ejemplo uso de rama master/main para el código que está en producción y rama desarrollo/develop para las características que están en construcción.

4 Adoptar un patrón de gestión de ramas para sistemas de control de versiones basado en troncos, en donde el código principal está en una única rama tronco (main o master) y las nuevas funcionalidades se desarrollan en lotes pequeños denominadas ramas de desarrollo de corta duración que suelen durar unas pocas horas, en donde los desarrolladores combinan los cambios constantemente con el tronco/rama principal, reduciendo la complejidad a la hora de integrar y estabilizar ramas.

5 Las estrategias y patrones para un manejo eficiente de un sistema de control de versiones es una tarea de constante aprendizaje y mejora, por lo la tanto la compañía debe enfocar sus esfuerzos en fortalecer practicas ágiles, dividir el trabajo en lotes pequeños, tener ramas de muy corta duración (horas), integrar frecuentemente de lotes de trabajo, adoptar aprobación ligera de cambios, implementar una fuerte cobertura de pruebas automatizadas. En este nivel no debe existir fases de integración y estabilización largas y complejas con la rama principal. El programa de investigación DORA DORA's (l) recomienda las siguientes practicas: Tener máximo 3 ramas activas en el repositorio de código, combinar al menos una vez al día las ramas alternas con la principal, no tener código congelado ni fases de integración.

Fuentes:

- Accelerate, Forsgren 2018, (Forsgren et al., 2018).

- DORA's research program, DevOps tech: Trunk-based development (DORA's, l).
- DORA's research program, DevOps tech: Version control (DORA's, a).
- Trunk Based Development,(Hammant's).

Tipo: Procesos

Subtipo: Metodologías

Categoría: Entrega continua

Capacidad: Utilizar métodos de desarrollo "Trunk-Based"

Ultima modificación: 20220312

5. **¿La compañía implementa procesos de pruebas en los desarrollos?**

   Justificación

   Se busca conocer si la compañía asegura la calidad de sus productos mediante pruebas, se indaga si estas estas hacen parte del proceso de despliegue ya sea como tarea manual o como tarea automática.

   El proceso de pruebas es un mecanismo esencial para el aseguramiento de calidad de los productos de software de una compañía, con el fin de asegurar que los usuarios finales obtengan productos confiables, seguros que generen valor para el negocio o necesidad particular.

   El diseño y ejecución de pruebas dentro del proceso de desarrollo es una labor constante que se debe realizar cada que se hace un cambio o se implementa una nueva funcionalidad sobre el proyecto, hacer estas tareas manualmente sin herramientas que apoyen el procese se vuelve desgastante y tedioso.

   Opciones de respuestas

   1 No se hacen pruebas

   2 Los desarrolladores hacen pruebas manuales sin un marco o herramienta definido.

   3 Los desarrolladores usan frameworks de pruebas manual unitarias automatizadas, sin una definición clara de que probar.

   4 Se utiliza un framework de pruebas estándar basados en un conjunto de definiciones de calidad acordado por el equipo de desarrollo, calidad y operaciones. En este punto se deben ejecutar pruebas unitarias, en el momento de cualquier generación de un producto final, adicional se deben manejar pruebas de integración, componentes de GUI en procesos desacoplados.

   5 Se tiene implementado un proceso de pruebas automático a lo largo del proceso de desarrollo de software, donde se garanticen la ejecución de pruebas funcionales, de integración y de GUI totalmente automáticas.

   Plan de acción

1 Es fundamental que la compañía inicie la adopción de procesos de pruebas durante todo el proceso de desarrollo, por lo tanto, se debe iniciar con la adopción de pruebas manuales de tipo unitarias, aceptación, usabilidad del sistema que se puede ir escalando a lo largo del tiempo, por ejemplo, iniciar con pruebas unitarias sobre el código fuente de las características core o esenciales del proyecto y progresivamente incorporar pruebas funcionales sobre el funcionamiento de la aplicación o GUI.

2 Se recomienda adoptar herramientas o frameworks diseñados para la implementación de pruebas (por ejemplo, JUnit, QUnit, Jasmine, Nunit, entre otros) acorde con el lenguaje de programación en que se desarrolla cada proyecto, con el fin de poder automatizar la ejecución de las pruebas del proyecto, inicialmente se puede enfocar en las pruebas unitarias e ir escalando a los demás tipos de pruebas, con el fin de que estas puedan ser ejecutadas automáticamente en las diferentes etapas del ciclo de vida de la entrega del proyecto.

3 Se debe implementar una hoja de ruta con los criterios y definiciones de prueba de cada aplicación, de forma estructurada en el que debe intervenir desarrolladores, probadores (si existe en la compañía) y demás miembros del equipo de trabajo con el fin de saber que se debe probar, cómo hacerlo y que tipo de pruebas hacer (unitarias, de integración, de GUI, ect). Existen muchas técnicas y metodologías como, TDD, BDD que son útiles para este propósito, por ejemplo, TDD Test-Driven Development (desarrollo dirigido por tests) es una práctica generalmente utilizada para pruebas unitarias en la que primero se diseña la prueba antes de iniciar su codificación.

4 Se debe iniciar con la integración y ejecución automáticas del grupo de pruebas que cuente el proyecto cada vez que se compile y genere un producto final para despliegue, por ejemplo se puede iniciar desde el bajo nivel integrando las pruebas unitarias de la aplicación mediante un proceso automático (ejecución de scripts del framework donde este implementadas) que las ejecute en el momento que se compile y se genere un producto final y progresivamente integrar los demás grupos de pruebas implementados (aceptación, de integración, de GUI) al proceso.

5 La mejora continua es uno de los pilares de la cultura DevOps, el aseguramiento de calidad de los productos de software es una parte clave en este proceso y se debe tener en cuenta desde las etapas iniciales del proyecto. En este nivel, el objetivo es reevaluar y mejorar los procesos existentes, por ejemplo, asegurar que las pruebas automáticas sean fiables y con altos niveles de cobertura en los diferentes tipos de pruebas existentes para lograr minimizar los paquetes de pruebas poco confiables y técnicamente costosos; fortalecer practicas TDD en nuevos desarrollos y trabajar en reducir la deuda técnica, obtener feedback constante de las suites de pruebas y trabajar en conjunto probadores para cubrir pruebas exploratorias, de usabilidad y de aceptación.

Fuente

- Accelerate, Forsgren 2018 (Forsgren et al., 2018).
- DORA's research program, DevOps tech: Continuous testing, (DORA's, o).
- The DevOps Handbook Second Edition 2021 (Kim et al., 2021).

Tipo: Herramientas

Subtipo: Pruebas

Categoría: Entrega continua

Capacidad: Implementar pruebas automáticas

Ultima modificación: 20220226

6. **¿Cómo definen que datos utilizar a la hora de probar las aplicaciones?**

   Justificación

   La gestión de los datos de pruebas es un aspecto que ha cobrado mucha relevancia para asegurar la ejecución de pruebas automáticas, por lo que es necesario conocer como lo maneja la empresa.

   Los datos de prueba deben ser analizados y definidos cuidosamente para permitir validar recorridos comunes o de alto valor, probar casos extremos, reproducir defectos y simular errores.

   Los equipos exitosos tienen los datos de pruebas adecuados para ejecutar sus suites de pruebas totalmente automatizadas y podían adquirir datos adicionales bajo demanda.

   Opciones de respuestas

   1 No se maneja ningún tipo de pruebas.

   2 A criterio del desarrollador en procesos manuales.

   3 Se define en conjunto con el equipo de calidad donde las pruebas se ejecutan manualmente.

   4 Parcialmente definidos en algunos procesos automáticos.

   5 Se tienen los datos adecuados para ejecutar la suite de pruebas totalmente automatizadas y con la posibilidad de adquirirlos bajo demanda.

   Plan de acción

   1 Se debe iniciar con la adopción progresiva de aseguramiento de calidad de software, para este primer nivel se puede iniciar con la implementación y documentación de pruebas manuales, por ejemplo, a nivel de código fuente identificar los procesos y funciones principales del proyecto e implementar pruebas manuales que permitan asegurar que estos cumplen con su

propósito. Por otro lado, a nivel funcional y entorno grafico se debe validar que el producto resultante cumple con los requisitos funcionales y no funcionales del proyecto. El objetivo de este nivel es tener al menos un conjunto de pruebas manuales que respalden los aspectos básicos de calidad del producto de software.

2 Se debe adoptar un proceso colaborativo entre desarrolladores, infraestructura y calidad para definir el criterio, alcance y los datos que se utilizarán durante las pruebas. Se recomienda priorizar las pruebas unitarias ya que son la base de la pirámide de pruebas y progresivamente ir escalando en los demás tipos de pruebas (aceptación, integración, etc.). Hay que asegurar que existan datos de pruebas adecuados, consistentes y realistas para que la ejecución de los casos de pruebas permita asegurar el correcto funcionamiento de la aplicación. El proceso de pruebas se debe adoptar desde las fases iniciales de los proyectos, a través de prácticas de desarrollo que promuevan la implementación de pruebas, como, por ejemplo, (Test-Driven Development).

3 Se debe iniciar con la automatización de las pruebas iniciando con las unitarias, partiendo del hecho que los datos de pruebas sean concisos, bien definidos y disponibles al momento de la ejecución. Permitiendo validar los procesos más comunes o de alto valor de la aplicación (ejemplo un lograr una cobertura del 80 % de los procesos Core) y así probar casos extremos, reproducir defectos y simular errores de la aplicación.

4 La automatización de las pruebas se debe integrar con procesos de construcción y aseguramiento de calidad, en donde los datos de pruebas deben ser adecuados, disponibles según demanda y no ser un factor limitante en la ejecución. Los datos de pruebas deben ser puntuales, aislados y específicos para no generar alta dependencias y lograr ejecuciones en paralelo.

5 El aseguramiento de la calidad y disponibilidad de los datos para la ejecución de pruebas es un factor clave para entregar productos confiables con tasas de fallos, por lo tanto, se debe enfocar los esfuerzos en la mejora constante en la calidad, fiabilidad y disponibilidad de los datos necesarios para la ejecución de los diferentes tipos de pruebas implementadas, así potenciar su cobertura y minimizar las tasas de fallos y errores en los ambientes de producción.

Fuentes:

- Accelerate, Forsgren 2018 (Forsgren et al., 2018).
- DORA's research program, DevOps tech: Test data management (DORA's, c).

Tipo: Procesos

Subtipo: Pruebas

Categoría: Entrega continua

Capacidad: Implementar gestión de los datos de prueba

Ultima modificación: 20220226

7. **¿Cuál es el nivel de relevancia que tiene la seguridad en el ciclo de vida del desarrollo de software?**

Justificación

Medir la seguridad es un aspecto clave para aseguramiento de calidad y confiabilidad en los productos de software.

Integrar la seguridad en las fases de diseño y pruebas del proceso de desarrollo de software e inclusive antes de iniciar el ciclo de vida del desarrollo, es clave para impulsar el rendimiento del área de desarrollo.

El equipo de seguridad de la información debe proporcionar feedback a cada paso del ciclo de vida de la entrega de software, desde el diseño hasta la ayuda con automatización de pruebas.

Al incorporar estas prácticas de seguridad se contribuye a mejorar el rendimiento de la entrega de software.

Opciones de respuestas

1 No se tiene en cuenta

2 Se tiene en cuenta cuando el producto ya está terminado

3 Se tiene en cuenta solo en la etapa al final del desarrollo

4 Se tiene en cuenta al equipo de seguridad, y se adoptan todas las recomendaciones a pesar de que afecten los tiempos de desarrollo y entrega.

5 Se involucra al equipo de seguridad desde la fase inicial del proyecto de forma que no ralenticen el proceso de desarrollo.

Plan de acción

1. Es fundamental que se inicie la adopción de aspectos de seguridad en todo el ciclo de vida de desarrollo particularmente desde el inicio del proyecto, incorporando pruebas y controles de seguridad, por lo tanto, incluir al menos un representante de seguridad de la información desde el inicio de los proyectos.

2 y 3. Se debe incluir una capa de seguridad al equipo de trabajo desde la fase de inicio del ciclo de vida de la aplicación para que apoyen procesos diseño arquitectónico y pruebas, con el fin de tener definiciones claras de librerías y herramientas aprobadas desde el punto de vista de seguridad, también, la adopción de pruebas de seguridad lo más automatizado posible en todas las fases del ciclo de vida del desarrollo.

4. El proceso de pruebas y validaciones de seguridad se debe definir desde las etapas de iniciales del proyecto en común acuerdo entre todos los miembros del equipo (desarrollo, operaciones y seguridad) de forma que no afecte los tiempos de entrega bajo el principio de responsabilidad compartida.

5. Se integra la seguridad en cada paso del ciclo de vida del desarrollo, teniendo en cuenta que es un proceso de mejora constante y que a través de elementos como feedback del equipo y de los usuarios finales permite las revisiones se seguridad de las aplicaciones, aprobación de uso de las versiones de paquetes, librerías y automatización de las pruebas, logrando mejorar constantemente la calidad de los productos que se despliegan a producción.

Fuentes:

- Accelerate (Forsgren et al., 2018).
- DORA's research program, DevOps tech: Shifting left on security (DORA's, r).

Tipo: Procesos

Subtipo: Seguridad

Categoría: Entrega continua

Capacidad: Incorporar los aspectos de seguridad en las fases de diseño y pruebas (shift left de la seguridad)

Ultima modificación: 20220226

8. **¿Ejecutan tareas automáticas al registrar algún cambio?**

Justificación

El objetivo es identificar qué pasos sigue la compañía para desplegar una solución o cualquier tipo de cambio en el ambiente de producción

Se deben tener en cuenta factores como:

- Construcción
- Calidad
- Automatización
- Integración
- Pruebas
- Seguridad

Los principios y las prácticas de la entrega continua aplican a cualquier contexto de software, por ejemplo:

- Actualizar servicios en un sistema distribuido complejo.
- Actualizar el software de la unidad central.
- Realizar cambios en la configuración de la infraestructura.
- Realizar cambios en el esquema de la base de datos.
- Actualizar el firmware de forma automática.

- Lanzar versiones nuevas de una aplicación para dispositivos móviles.

La entrega y la implementación continua son practicas independientes. La implementación continua se da cuando los equipos pasan todos los cambios de una aplicación en producción lo antes posible, esta funciona bien para ambientes web, pero no se puede aplicar a software como firmware o aplicaciones de dispositivos móviles. En cambio, la entrega continua es la capacidad de actualizar cambios de todo tipo a pedido de manera rápida, segura y sustentable y se aplica a todo tipo de software

La entrega continua permite la capacidad de lanzar software y realizar cambios en la producción de forma segura y en cualquier momento, incluso durante el horario laboral o de uso habitual, sin afectar a los usuarios finales.

Si bien la entrega continua a menudo se combina con la integración continua y se abrevia como *CI/CD*, la investigación de DORA, muestra que la integración continua es solo un elemento de la implementación de la entrega continua.

Opciones de respuestas

1 No
2 Se tienen tareas de validación de calidad que se ejecutan manualmente
3 Se ejecutan tareas de calidad de software y pruebas
4 Se compila, construye y se prueba después de registrar cada cambio
5 Se tienen tareas automáticas de construcción, pruebas, calidad de software, etc.

Plan de acción

1 Se debe asegurar que el código fuente del proyecto tenga la capacidad de compilarse y generar un producto final o desplegable por medio de la ejecución de comandos propios de una herramienta de gestión de librerías.
2 Se debe iniciar con la adopción de tareas automáticas partiendo desde un sistema de control de versiones centralizado, en el que puedan ejecutar tareas automáticas como ejecución de pruebas unitarias y análisis de calidad de código estático.
3 Como primera medida, se debe asegurar que todas las tareas automáticas ya existentes estén cargadas en un sistema de control de versiones incluyendo el código fuente. Iniciar con la adopción de una herramienta de orquestación e integración y a través de una secuencia de procesos estas tareas se ejecuten automáticamente al momento de subir un cambio en el sistema de control de versiones.
4 Se debe contar con una herramienta orquestadora de tareas que integre el código fuente y scripts de pruebas y configuración (si existen) que permita que las tareas de compilación, construcción y ejecución de pruebas (unitarias, funcionales, etc.) se integren en una secuencia de procesos automático, resultado en la generación de un producto final listo para el despliegue.

5 La integración continua es un proceso de constate evolución que involucra tener todos los procesos de la construcción de un producto de software completamente automáticos, iniciando desde la construcción del producto final, pasando por ejecución de pruebas (unitarias, funcionales, de integración, etc.), seguridad, calidad de software, etc.

Fuentes:

- Accelerate, Forsgren 2018 (Forsgren et al., 2018).
- DORA's research program, DevOps tech: Continuous delivery (DORA's, p).

Tipo: Procesos

Subtipo: Entrega continua

Categoría: Entrega continua

Capacidad: Implementar entrega continua de software

Ultima modificación: 20220107

9. **¿Qué características tiene la infraestructura de la organización?**

Justificación

Se busca conocer el grado de acoplamiento de la infraestructura de la organización, que va desde ambientes configurados de forma manual, altamente acoplados hasta ambientes desacoplados y dinámicos

Opciones de respuestas

1 Ambientes configurados e instalados manualmente

2 Servidores fijos con configuración dinámica

3 Ambientes parametrizados fuertemente acoplados al hardware

4 Ambientes desacoplados con aprovisionamiento manual

5 Ambientes débilmente acoplados y dinámicos

Plan de acción

1 La compañía debe iniciar un plan de organización y estandarización del stack tecnológico existente con el fin de conocer que elementos pueden ser compartidos y reutilizables en diferentes condiciones, con el fin de que en un futuro se pueda eliminar la necesidad de hacer configuraciones y actualizaciones manuales de las piezas individuales de hardware. Por otro lado, identificar y depurar los conocidos entornos particulares o exclusivos que no pueden reproducirse automáticamente.

2 La compañía debe iniciar un plan de transformación de la arquitectura que soporta la operación de las aplicaciones e incorporar un nuevo concepto conocido como infraestructura como código (IaC) que de primera mano permitiría que la compañía pueda replicar fácilmente la configuración de un servidor a través de scripts. El mercado ofrece un grupo de herramientas para este tipo de funciones, ejemplo: Puppet, Chef, Ansible, AWS cloud formation, Terraform, etc.

3 La adopción del concepto de infraestructura como código IaC, debe ir más allá de solo replicar la configuración de servidor a partir de la ejecución de un script, esta desde estar desligada del hardware físico y se debe enfocar por opciones como virtualización, contenedores o cloud computing. Logrando que los desarrolladores no tengan que preparar ni gestionar manualmente ningún elemento de la infraestructura cada vez que desarrollan o implementan una aplicación.

4 La compañía debe iniciar con la adopción de infraestructura con código (IaC) en la organización e integrado al ciclo de vida de las aplicaciones desde etapas de integración, desarrollo, pruebas hasta el despliegue y distribución. Partiendo de entornos uniformes acordados entre infraestructura y desarrollo se pueden generar los todos los entornos necesarios (incluida producción) de forma automática.

5 La de infraestructura con código (IaC) integrado con las practicas DevOps deber ser adoptado como un proceso de autoevaluación y mejora constante con el fin de garantizar la entrega rápida de software. La compañía debe evaluar constantemente aspectos como, la capacidad de provisionamiento, elasticidad de la plataforma, disponibilidad del servicio. En este nivel la recomendación es trabajar o migrarse a modelos de infraestructura en la nube y adoptar sus prácticas y patrones nativos con el fin de lograr agilidad, estabilidad, disponibilidad y transparencia de costos. Con este nivel de capacidad la organización mejorará el rendimiento de entrega de productos y alto rendimiento organizacional.

Fuente:

- Accelerate, Forsgren 2018 (Forsgren et al., 2018).
- Red Hat, IaC, (Red Hat).
- DORA's research program, DevOps tech: Cloud infrastructure (DORA's, b).

Tipo: Herramientas

Subtipo: ambientes

Categoría: Arquitectura

Capacidad: Utilizar arquitecturas de bajo acoplamiento

Ultima modificación: 20220312

10. **¿Cómo la organización define las herramientas que utilizan los equipos en los proyectos?**

    Justificación

    Busca medir el grado de libertad para elegir las herramientas a utilizar dentro de un proyecto.

    DevOps tech: Empowering teams to choose tools [35]: Permitir que los equipos elijan las herramientas no significa que se dé rienda suelta a cada equipo para que seleccione la herramienta que quiera. Introducir tecnologías sin ningún tipo de restricción puede aumentar la deuda técnica y la fragilidad. Sin embargo, cuando se combina la elección de herramientas con otras capacidades (por ejemplo, una visión completa del sistema, una retroalimentación rápida y la comprensión de que son responsables del código que escriben), se ayuda a los tecnólogos a tomar decisiones acertadas sobre las herramientas que van a utilizar y a las que deben dar soporte.

    Opciones de respuestas

    1 No hay un grupo de herramientas base para los proyectos, nadie es responsable de su mantenimiento.

    2 El stack tecnológico está definido como política institucional y todos los desarrollos se deben regir bajo ellos.

    3 El stack tecnológico del proyecto está definido por los lideres técnicos y no hay opción de ajustarlo.

    4 El stack tecnológico del proyecto está definido por los lideres técnicos, cualquier intento de cambio debe pasar por aprobación de un comité.

    5 Los equipos pueden decidir que herramientas utilizar acorde a las necesidades del proyecto y en consenso general

    Plan de acción

    1 La compañía debe iniciar el inventario completo las herramientas de desarrollo utilizadas en todos los proyectos de software, a partir de esta información se debe identificar herramientas comunes que sirvan como elementos de apoyo y referencia para iniciar la definición la estandarización de las herramientas utilizadas por el equipo de trabajo.

    2 Una de las practicas DevOps, es minimizar el uso de anti patrones como martillo de oro, bala de plata en donde todos los problemas se pretenden solucionar con una única herramienta o metodología, por lo que un stack tecnológico de una organización debe permitir ajustes de acuerdo con las necesidades de los proyectos.

    3 El stack tecnológico de la organización no se debe considerar como un elemento estático al que le pasan los años sin ninguna alteración. La vertiginosa dinámica de productos y herramientas informáticos en nuestra

actualidad requiere que la compañía permita la adopción de nuevas y el desuso de otras. En compañías conservadores este tipo de análisis se puede hacer en periodos anuales o bianuales, con el objetivo de brindar a los equipos las herramientas necesarias para satisfacer las necesidades de los clientes y mejorar los resultados comerciales.

4 El stack tecnológico de la organización deber ser evaluado por lo menos una vez al año con el fin brindar a los equipos las herramientas necesarias para satisfacer las necesidades de los clientes y resultados comerciales.

5 La capacidad de que los equipos puedan decidir que herramientas utilizar acorde a las necesidades de un nuevo proyecto es un trabajo de mejora continua de parte de todo el equipo de trabajo. El stack tecnológico en operación se evaluar con frecuencia (al menos al inicio de cada proyecto) o al cambio de condiciones para productos en operación sin la necesidad de aprobación de personas ajenas al equipo de trabajo. Esta capacidad de autonomía es un factor importante para mejorar el rendimiento, la satisfacción a los clientes y añadir valor a la organización, además de propiciar el espíritu investigativo y proactivo del equipo para proponer nuevas herramientas de trabajo.

Fuente:

- Accelerate, Forsgren (Forsgren et al., 2018).
- DORA's research program, DevOps tech: Empowering teams to choose tools (DORA's, u).

Tipo: Herramientas

Subtipo: Infraestructura

Categoría: Arquitectura

Capacidad: Permitir a los equipos definir sus propias arquitecturas

Ultima modificación: 20220312

11. **¿La compañía recoge el feedback de los clientes respecto a sus productos de software?**

Justificación

Se buscar conocer si la compañía recoge el feedback de los clientes respecto a los productos que despliega, en que etapas y con qué frecuencia lo hace.

El feedback incluye métricas de satisfacción de los clientes y opiniones frente a la calidad del producto y sus funcionalidades.

El programa de investigación de DevOps de DORA (DORA's, u) resalta que los equipos obtienen un mayor rendimiento cuando trabajan en organizaciones que:

- Recopilan métricas de satisfacción del cliente con regularidad.

- Buscan y atienden los comentarios de los clientes sobre la calidad de los productos y las características.
- Usan estos comentarios para ayudar a diseñar productos y características.

Opciones de respuestas

1 Nunca, compañía solo se centra en entregar el desarrollo en los tiempos estipulados

2 Ocasional, pero solo para los productos terminados

3 A veces, solo cuando en producto está en las etapas finales del desarrollo

4 Si, durante la etapa de desarrollo, aunque sin incidir en el proceso de desarrollo

5 Si, de manera activa, desde el inicio del ciclo de vida del producto y su posterior puesta en producción, estos resultados tienen incidencia en el diseño del producto y nuevas funcionalidades.

Plan de acción

1. Se debe iniciar con la adopción progresiva de la práctica de recolectar el feedback de los clientes durante todo del ciclo de vida del producto, en especial en las etapas tempranas del proyecto, estas opiniones se deben tener en cuenta para el diseño del producto.

2 y 3 El feedback de los clientes frente a los productos desarrollados deber ser incorporados dentro de todo el ciclo de vida del producto, a través de la recopilación activa y periódica de la opinión de los usuarios finales y la definición de métricas que miden la satisfacción del cliente.

4. El feedback como métricas de satisfacción y opinión del producto deben ser tomadas periódicamente e incorporadas en la toma de decisiones en cada fase del ciclo de vida del producto.

5. El feedback constante de los clientes en métricas de satisfacción, opinión del producto y sus funcionalidades debe ser un proceso constante e influyente en el diseño del producto desde el inicio del ciclo de vida del producto (desde el diseño hasta etapas posterior al despliegue en producción).

Fuente:

- Accelerate, Forsgren (Forsgren et al., 2018).
- DevOps process: Customer feedback (DORA's, q).

Tipo: Procesos

Subtipo: Metodologías

Categoría: Producto y Procesos

Capacidad: Recolectar e implementar las recomendaciones (feedback) de los clientes

Ultima modificación: 20220227

12. **A nivel de proyectos, ¿cómo fluye la comunicación entre los miembros del equipo de trabajo y los directivos?**

    Justificación

    Se buscar conocer como es el nivel de comunicación y entendimiento del flujo de trabajo de los proyectos entre todos los miembros del equipo

    Opciones de respuestas

    1 Los lideres y stakeholders del proyecto son los únicos que conocen y definen los objetivos, alcances y detalles de los proyectos por iniciar, sin comunicarlos con los equipos técnicos (desarrolladores, calidad e infraestructura).

    2 Los detalles del proyecto no se comunican desde el inicio y se comparten en etapas avanzadas del proyecto, generando reprocesos.

    3 El alcance y objetivos de los proyectos se comunican parcialmente sin mayor nivel de detalle, solo los lideres técnicos los conocen en profundidad.

    4 Todos los miembros del equipo conocen el alcance y objetivos de los proyectos a alto nivel.

    5 Todos los miembros conocen los elementos claves del proyecto y están visibles para todos.

    Plan de acción

    1 La comunicación dentro de los equipos de desarrollo es un factor esencial a la hora de trabajar de forma colaborativa y así lograr mejoras en la entrega de software y el rendimiento de la organización, por lo tanto, la compañía debe iniciar un cambio en la forma en cómo se maneja la información relacionada con los proyectos (personal involucrado, actividades a desarrollar, flujo de información). Se busca que el equipo pueda conocer los detalles de las actividades a realizar en cada etapa del proyecto.

    2 La compañía debe asegurar que el equipo de trabajo conozca y participe desde el inicio de proyecto en la definición del flujo de trabajo para el desarrollo del producto, que va desde la solicitud del cliente hasta la entrega del producto, definiendo bloques de trabajo, el personal involucrado, las activades a desarrollar, los tiempos estimados de las tareas. Por último, se lleva a herramientas de visualización para que todo el equipo pueda consultar el estado de cada etapa durante todo el ciclo de vida del desarrollo de la aplicación.

    3 El flujo de información dentro del equipo es un factor para el entendimiento de los proyectos, en este se debe definir el flujo de trabajo desde el negocio hasta los clientes a través de bloques de procesos o lo que defina la organización en su metodología de trabajo, lo realmente importante es que esta información sea de fácil acceso para el equipo y no sea conocimiento exclusivo de la directiva y los lideres técnicos.

4 La creación de un flujo de valor es una herramienta que permite representar los pasos necesarios para entregar un producto de software a los clientes. Aquí se busca representar el flujo de la información a través de los diferentes procesos desde el punto de vista técnico para cumplir con las necesidades del cliente. Teniendo esta estructura creada y accesible por todos los miembros del equipo se logran mejoras en el rendimiento de la entrega de software.

5 La definición y evolución del mapa de una cadena de valor es un trabajo colaborativo entre todos los miembros del equipo de trabajo que permite descentralizar la toma de decisiones y encontrar dónde están los desperdicios o 'waste' para entregar valor de forma continua al cliente. Una cadena de valor sin visualización y evolución no es útil para que una empresa pueda mejorar sus procesos y generar productos de valor. Por ejemplo, se puede apoyar en herramientas visuales como muro de tarjetas, guiones o tableros Kanban para hacer visible la cadena de valor.

Fuente:

- Accelerate, Forsgren 2018 (Forsgren et al., 2018).
- DORA's research program, DevOps process: Visibility of work in the value stream (DORA's, t).
- Value Stream Mapping, K Martin, 2013 (Martin and Osterling, 2014).

Tipo: Cultura

Subtipo: Comunicación

Categoría: Producto y Procesos

Capacidad: Hacer visible el flujo de trabajo durante todo el ciclo

Ultima modificación: 20220304

13. **¿Cómo los equipos dividen el trabajo a la hora de iniciar el desarrollo de productos y funcionalidades para tener un mínimo producto viable?**

Justificación

La recomendación y buenas prácticas de las metodologías agiles es dividir el proyecto en entregas pequeñas que aporten valor funcional al usuario final.

La investigación DORA indica que trabajar en lotes pequeños trae lo siguientes beneficios en el proceso de desarrollo de software:

- Reduce el tiempo que implica recibir comentarios sobre los cambios, lo que facilita la clasificación y la solución de problemas.
- Aumenta la eficiencia y la motivación.
- Evita que la organización caiga en la falacia del costo irrecuperable.

Entregar productos o cambios de software en periodos superiores a un 1 año se asocia a metodologías tradicionales de desarrollo, por ejemplo, RUP y cascada, etc. donde se basan en modelos de entrega de un único producto final al final del ciclo de vida del desarrollo de la aplicación con un producto totalmente terminado. En estos modelos la interacción con el cliente es escasa y en muchos casos interviene en los extremos del ciclo de vida (fase inicial con la gestión de requisitos y al final en la entrega del producto).

Entregar productos en periodos de 6 meses a 1 año se asocia a metodologías tradicionales, donde aún se corre un riesgo de perder oportunidades de mercado o que las necesidades del cliente evolucionen y se generen reprocesos generando pérdida de tiempo, fuerzo para el equipo de desarrollo

Trabajar en lotes pequeños permite entregarle productos de valor al cliente en periodos cortos, aquí las metodologías agiles juegan un papel importante por que permiten una constante evolución y adaptabilidad a las necesidades del cliente entregando productos mínimos viables o prototipos usables y tener rápidas retroalimentaciones pasando de completar tareas en semanas a días e incluso a horas.

Opciones de respuestas

1 El trabajo se agrupa en proyectos grandes con lanzamientos poco frecuentes de más de un año

2 En lotes medianos de entre 6 meses a 1 año

3 En lotes medianos de 3 a 6 meses

4 En lotes pequeños entre 1 y 3 meses

5 En lotes pequeños que pueda completarse en tres semanas o menos

Plan de acción

1 Entregar productos o versiones usables de un software en periodos mayores a 1 año es una práctica poco recomendable porque se corre el riesgo de perder oportunidad de mercado, o que las necesidades del cliente cambien y entregar productos obsoletos que no cumplan con lo que el cliente requiere, por eso el primer paso para reducir esta situación es iniciar con un cambio en la forma como se entrega el producto final a los clientes, adoptando la estrategia de dividir el trabajo en partes más pequeñas que puedan completarse en periodos menores, en este caso entregar versiones usables en tiempos menores a 1 año. El objetivo es que los desarrolladores adquieran la experiencia de dividir el trabajo de varios años a entregas de menos de 1 año.

2 La compañía debe iniciar la adopción de metodologías de desarrollo agiles (Scrum, Kanban, Lean etc, que permitan facilitar la división de trabajo en lotes medianos que sean independientes, valiosos y verificable y así entregar a los usuarios finales y demás interesadas productos de valor usables en rangos de tiempo entre 3 y 6 meses.

3 Iniciar la adopción de un enfoque en donde prime la división del trabajo en partes pequeñas de forma independiente, valioso, verificable y negociable, logrando que la entrega de productos mínimos viables pueda adaptase a las necesidades del cliente. Descomponer el trabajo en funcionalidades que permitan un desarrollo rápido en periodos menores a 3 meses.

4 El siguiente paso para alcanzar un proceso de entrega ágil de alto rendimiento es conocer los procesos de la organización, identificar y reducir el desperdicio o procesos que no generen valor, y reducir la complejidad del proceso de desarrollo. A partir de estos factores dividir todas las tareas del desarrollo en lotes de trabajo pequeños logrando que estas sean independientes, estimables, verificables y valiosos y que se puedan desplegar en producción en periodos de menos de 3 semanas.

5 La entrega oportuna de productos de valor es un proceso de mejora constante, en donde los equipos de trabajo se deben enfocar para minimizar el esfuerzo necesario para desplegar una aplicación y lograr tiempos de entrega cortos. Actualmente, las grandes empresas de desarrollo de la industria manejan tiempos de despliegues diarios, incluso varios en un mismo día o bajo demanda. Características como, la adopción de metodologías agiles, buenas prácticas de desarrollo, dividir el trabajo en lotes pequeños, automatización de procesos y pruebas, toma rápida de decisiones, etc, son elementos claves para alcanzar la mejora continua de la frecuencia de despliegues en producción, llevando esta tarea a medirse en horas en lugar de días y semanas.

Fuentes:

- Accelerate, Forsgren (Forsgren et al., 2018).
- DORA's research program, DevOps process: Working in small batches (DORA's, h).
- Accelerate State of DevOps 2021 Report (DORA's, 2021).

Tipo: Procesos

Subtipo: Metodologías

Categoría: Producto y Procesos

Capacidad: Trabajar con lotes o cantidades pequeñas

Ultima modificación: 20220312

14. **¿Cuál es nivel de libertad que tiene el equipo de desarrollo para elegir sus propias herramientas en función de lo que sea mejor para el usuario?**

Justificación

Busca medir la flexibilidad de la compañía para que los desarrolladores puedan experimentar con nuevas herramientas y metodologías, apoyando el crecimiento del portafolio de opciones y no limitarse a un grupo definido de aplicaciones

Opciones de respuestas

1 Ninguna, la compañía tiene una lista de herramientas definidas (stack tecnológico) y no se puede usar nada diferente

2 Poca, cualquier cambio o nueva herramienta a utilizar en el proyecto debe venir únicamente de un líder o comité técnico.

3 Moderado, la compañía tiene un stack tecnológico definido y para usar una nueva herramienta debe pasar por varias aprobaciones, un comité, líder técnico, infraestructura, etc.

4 Controlado, los equipos pueden proponer el uso de alguna herramienta previa aval del líder técnico del proyecto.

5 Concertado, los equipos pueden decidir que herramientas utilizar acorde a las necesidades particulares del proyecto debidamente comunicadas.

Plan de acción

1 Se debe iniciar con un acuerdo entre desarrolladores, lideres de desarrollo y comité técnico (si existe) y gerencia para evaluar al menos 2 veces al año el stack tecnológico vigente y decidir adoptar nuevas herramientas para mejorar el rendimiento y funcionalidades de nuevos desarrollos.

2 El stack tecnológico de una organización debe evaluarse al menos 2 veces en el año con el fin de tener un portafolio de opciones para cubrir necesidades particulares de proyectos en curso o por futuras de la organización o mejorar procesos vigentes.

3 La mejora del stack tecnológico para un proyecto de software deber ser manejado como un proceso participativo en donde un líder técnico o comité especializado están siempre dispuestos a escuchar propuestas de mejora de cualquier miembro del equipo de trabajo y decir su adopción en proyectos nuevo, en curso y en ocasiones ya finalizados.

4 Durante el proceso de desarrollo, los equipos pueden elegir sus propias herramientas en función de lo que sea mejor para los usuarios finales, respaldándose en experiencias de mercado, tendencias mundiales fundamentadas en análisis de rendimiento y feedbacks de los usuarios, sin requerir aprobación de personas ajenas al equipo.

5 Permitir al equipo de trabajo la experimentación en nuevas herramientas es un cambio cultural que la compañía debe acoger como una filosofía de mejora constante para potenciar la innovación y adopción de técnicas modernas de desarrollo. El programa de investigación DORA identifica 3 componentes claves en este aspecto: capacidad de trabajar en nuevas ideas de forma independiente, poder crear y ajustar especificaciones durante el desarrollo, sin requerir aprobación de alguien por fuera del equipo. Todo esto permite lograr mayor satisfacción de los clientes, impactando significativamente la capacidad de entrega continua del software.

Fuente:

- Accelerate, Forsgren 2018 (Forsgren et al., 2018).

Tipo: Procesos

Subtipo: Desarrollo

Categoría: Producto y Procesos

Capacidad: Fomentar y habilitar al equipo para realizar experimentos

Ultima modificación: 20220312

15. **¿Cómo es proceso de gestión de cambios de la compañía?**

Justificación

Se busca conocer como la compañía gestiona el control de cambios en los proyectos, usando una métrica de escala de capacidades iniciando en un estado en donde la compaña no tiene un orden o control de los cambios recibidos, hasta llegar a una buena práctica adoptada en DevOps de tener procesos de aprobación ligeros basados en metodologías de evaluación que permitan agilidad en el proceso de desarrollo

Opciones de respuestas

1 No se tiene proceso de evaluación o aprobación de cambios y se realizan en orden de llegada

2 Todos los cambios deben ser evaluados y aprobados por un comité u organismo externo

3 Todos los cambios deben ser evaluados y aprobados por un comité interno.

4 Solo los cambios de alto riesgo que impliquen bases de datos y arquitectura requieren aprobación.

5 Se tiene un proceso de aprobación ligero basado en evaluación apoyado en metodologías de evaluación definidas, ejemplo: evaluación por pares, evaluación 360 entre otras.

Plan de acción

1 Se debe iniciar la adopción de procedimientos y herramientas que permitan gestionar y visualizar los cambios de un proyecto, lo principal es darles orden y visibilidad a dichas solicitudes para que estas puedan ser analizadas y aprobadas por los mecanismos y metodologías que use la compañía (revisión por pares, revisión inter equipo, evaluación 360).

2 Se debe iniciar aliviando la complejidad a la hora de evaluar y aprobar cambios del proyecto, el objetivo es iniciar la definición de un proceso de aprobación ligero a la hora de tomar decisiones. En la investigación (Accelerate Forsgren et al. (2018)) descubrieron que las aprobaciones externas estaban correlacionadas de forma negativa con el tiempo de provisión, la

frecuencia de los despliegues y el tiempo de restauración sin correlación con la tasa de fallos en los cambios.

3 La implementación de un proceso ligero de gestión de cambios permite empoderar a los equipos en la toma de decisiones, trasladar la aprobación de cambios a un comité interno permite ganar agilidad en frecuencia de despliegues respecto a tener la aprobación en un comité externos. Adoptar estrategias como hacer revisión por pares para hacer cambios individuales. Otro paso importante es definir qué tipo de cambios requieren o no la evaluación comité y cuales se dejan a decisión del equipo.

4 Se debe empoderar al equipo de trabajo para que adopte procesos de aprobación ligeros y gestionados dentro del mismo equipo, enfocándose en cambios de alto impacto, por ejemplo, que afecten la base de datos y la arquitectura. Otro factor importante a tener en cuenta es validar el proceso de cambios a nivel macro del proyecto con el fin de identificar cuellos de botella, definir formas de detectar problemas de forma automatizada y anticiparse a posibles fallos.

5 La optimización de la aprobación de los cambios es una actividad de constante evolución y no debe ser tratada como un proceso estático, esta se debe basar en dos objetivos importantes, que son entregar oportunamente los requisitos y minimizar los cambios a raíz de fallos. Para este propósito, se recomienda actividades como, adoptar aprobaciones ligeras basados en revisión por pares, implementar pruebas continuas y de integración para detectar cambios incorrectos, facilitar canales de notificación y coordinación dentro del equipo, etc.

Fuente

- Accelerate, Forsgren (Forsgren et al., 2018).
- DORA's research program, Streamlining change approval (DORA's, n).

Tipo: Procesos

Subtipo: Lean

Categoría: Gestión Lean y Monitorización

Capacidad: Tener un proceso liviano para la aprobación de cambios

Ultima modificación: 20220203

16. **¿La organización monitorea la infraestructura y las aplicaciones?**

Justificación

Se busca conocer, si la empresa implementa o no herramientas de monitoreo para la infraestructura y las aplicaciones que desarrollan, con el fin de obtener información relevante para decisiones empresariales sobre tasas de uso y rendimiento

Opciones de respuestas

1 No

2 Si, solo se tienen herramientas para verificar el estado de salud de la infraestructura

3 Si, se utilizan herramientas para monitorear la infraestructura y las aplicaciones, pero solo las maneja el equipo de operaciones

4 Si, se utilizan herramientas para monitorear la infraestructura y las aplicaciones y todo el equipo puede consultarlas

5 Si, se utilizan herramientas para monitorear diariamente la infraestructura y las aplicaciones para tomar decisiones empresariales

Plan de acción

1 Se debe adoptar e implementar herramientas que permitan verificar el estado básico de la infraestructura y de las aplicaciones en operación enfocados en estadísticas de uso y rendimiento, por ejemplo, uso de CPU, memoria RAM y espacio en disco duro.

2 El monitoreo debe ir más allá de solo tener las estadísticas de uso de los recursos físicos del entorno o el servidor donde se despliegan las aplicaciones, se debe adoptar herramientas de monitoreo del estado de salud las aplicaciones, no basta que el servidor esté funcionando si el usuario final no puede utilizar la aplicación.

3 Los datos de monitoreo de la infraestructura y las aplicaciones deben ser de fácil acceso para todos los miembros del equipo y no solo de uso exclusivo de un grupo particular, el principal objetivo es obtener datos relevantes a través de una supervisión inteligente y focalizada en las áreas claves que tengan el potencial de generar valor para la compañía y de fácil acceso.

4 La información de monitoreo del rendimiento de la infraestructura y las aplicaciones son elementos influyentes para tomar decisiones empresariales, se debe asegurar que los datos recolectados de estos monitoreos sean relevantes y den valor para apoyar la toma de decisiones empresariales, en este punto se debe ir más allá de solo tener en cuenta factores como tasa de fallo, rendimiento, etc y enfocarse en una visión global.

5 El monitoreo de la infraestructura y las aplicaciones son elementos influyentes para apoyar la toma de decisiones empresariales, la organización debe manejar este proceso como una práctica de mejora constante, a partir de revisión de las tareas y el alcance de las mismas para poder brindar información de calidad sobre las áreas claves de la cadena de valor, por ejemplo, estadísticas de rendimiento y tasas de fallo de las aplicaciones. Estas prácticas, permiten identificar con rapidez oportunidades de aprendizaje, incrementan el rendimiento de la entrega de software y también tienen efectos positivos sobre el rendimiento y la cultura de los equipos.

Fuente:

- Accelerate, Forsgren (Forsgren et al., 2018).
- DORA's research program, Monitoring systems to inform business decisions (DORA's, j).

Tipo: Procesos

Subtipo: Lean

Categoría: Gestión Lean y Monitorización

Capacidad: Monitorizar las aplicaciones y la infraestructura para tomar decisiones de negocio

Ultima modificación: 20220312

17. **¿Cómo se maneja las notificaciones que vienen del monitoreo de los sistemas?**

Justificación

Se busca conocer como gestionan y reciben las notificaciones de los sistemas de monitoreo, donde se busca tener alertas tempranas frente a rendimiento y tasas de uso de las aplicaciones y la infraestructura

Opciones de respuestas

1 No se monitorea

2 No se reciben notificaciones

3 Se reciben notificaciones en caso de fallos generados por sistemas de monitorización y registros

4 Se reciben advertencias de umbral de capacidad (Ej uso de CPU por encima del 90 %)

5 Se reciben advertencias de monitoreo en función de tasas de cambio (ej, el uso de la CPU incrementó en un 25 % en los últimos 10 minutos)

Plan de acción

1 Se debe iniciar con el monitoreo básico de la infraestructura, que implica por ver consultar el estado de los recursos de máquina del servidor (disco duro, memoria y CPU) donde se aloja el producto de software.

2 El monitoreo debe ir más allá de solo poder consultar el estado de recursos de maquina donde se aloja el servidor (disco duro, memoria y CPU) sino que se debe definir un canal de envío de notificaciones en caso de alguna falla de infraestructura, estas notificaciones pueden ser vía correo electrónica, mensaje de texto, o mensaje a canal de soporte (slack, Ms Teams, etc).

3 El monitoreo de infraestructura no solo debe limitarse a ser procesos reactivos ante alguna anomalía en la infraestructura, este debe tener la capacidad de anticiparse a hechos que puedan generar alguna situación anómala,

por ejemplo, notificar situaciones de picos de alto consumo de recursos de CPU y memoria con el fin de prevenir caídas de la infraestructura y de las soluciones de software.

4 El monitoreo preventivo y proactivo se debe enfocar en anticipar situaciones de alta demanda del producto de software, en función a tasas de cambio de los recursos del servidor, ejemplo incremento de más de 40 % del uso de la CPU en los últimos 10 minutos)

5 El monitoreo debe ir más allá de la medir el estado de salud de la infraestructura o elementos supervisados, la compañía se debe adoptar una filosofía de mejora constante en los procesos para implementar notificaciones proactivas basados en aspectos como: definir reglas claras para generar alertas, establecer umbrales de activación en categorías especialmente cuando se genera o no un impacto en los usuarios, hacer análisis retrospectivos de los incidentes e implementar medidas de mitigación. El objetivo es ser reactivo y anticiparse a fallas en las aplicaciones con el fin de garantizar la disponibilidad del producto de software hacia los clientes finales.

Fuente:

- Accelerate, Forsgren (Forsgren et al., 2018).
- DORA's research program, Proactive failure notification (DORA's, f).

Tipo: Procesos

Subtipo: Lean

Categoría: Gestión Lean y Monitorización

Capacidad: Monitorizar la salud de los servicios y aplicaciones proactivamente

Ultima modificación: 20220312

18. **¿Cómo controlan la carga de trabajo en los proyectos?**

Justificación

Se busca conocer como la compañía gestiona las cargas de trabajo del equipo de desarrollo.

A partir de la investigación DORA, limitar el trabajo en proceso forman parte de las capacidades para mejorar la entrega de software y el rendimiento organizacional, por lo tanto, lo primero que se debe hacer:

- Prioriza el trabajo.
- Limita la cantidad de personas que trabajan en una tarea.
- Concentrarse en completar pocas tareas de alta prioridad.

Una de las estrategias que recomienda DORA es hacer las siguientes preguntas:

- ¿Conoces el plazo de entrega y la variabilidad promedio de todo el flujo de valor (desde la idea hasta el cliente)?

- ¿Estás buscando maneras de aumentar el flujo y reducir así el plazo de entrega del trabajo?
- ¿Los límites de WIP ayudan a manifestar los obstáculos que te impiden aumentar el flujo?
- ¿Estás tratando de resolver esos obstáculos?

Opciones de respuestas

1 No hay control sobre la cantidad de trabajo que se le asigna a un equipo

2 Los equipos no controlan la cantidad de trabajo y normalmente se sobrecargan.

3 Se identifican las sobrecargas, aunque no se puede hacer nada para limitarlas

4 Los equipos por sí mismos a juicios propios tratan de controlar la cantidad de trabajo para no sobrecargarse

5 Se cuenta con procesos que ayudan a limitar la cantidad de trabajo en proceso apoyado en feedbacks y datos de monitoreo.

Plan de acción

1 Implementar un sistema o metodología que permita registrar, consultar y asignar tareas del proyecto de software a los miembros del equipo de trabajo basados en la priorización del trabajo, limitar la cantidad de personas asignadas a una tarea y enfocado completar las tareas de alta prioridad.

2 Revisar y analizar la forma en que se asignan a cada miembro del quipo las cargas por cada periodo de entrega del proyecto. Identificar donde están los cuellos de botella, preguntándose cuando y por qué se alcanzan los límites de trabajo en curso, para esto se recomienda usar un storyboard o guion gráfico con el fin de tener escrito todo el trabajo que el equipo realiza en las diferentes etapas del proceso de desarrollo.

3 No caer en la tentación de ajustar arbitrariamente la cantidad de trabajo en curso de un equipo de trabajo, este deber ser un proceso de revisión constante y acordado entre los miembros del equipo para que a partir de un análisis fundamentado en el rendimiento y en el cumplimiento de tareas se definan las diferentes estrategias para regular la cantidad de trabajo en progreso, por ejemplo, limitar la cantidad de asignación de trabajo al paso del proceso más lento y respetando esos límites definidos, analizando y buscando mejorar los factores que contribuyen a los retrasos.

4 Los límites del trabajo en curso mejoran el rendimiento y reducen la cantidad de trabajo "prácticamente listo", ya que obliga al equipo a centrarse en un conjunto de tareas más pequeño fomentan la cultura de finalizar tareas específicas y administradas como consenso del equipo de desarrollo, por lo tanto, se debe asegurar que, el trabajo este priorizado, equilibrado (limitar

la cantidad de personas trabajando en una misma tarea) y focalizado en completar las tareas de alta prioridad, para lograrlo se debe hacer visible todas las actividades del proyecto, ejemplo adoptando un guion gráfico o storyboard, definiendo límites de trabajo en curso acorde a la capacidad del equipo.

5 Controlar la cantidad de trabajo en proceso en progreso (WIP) es una práctica de gestión LEAN que debe ser adoptado como un proceso dinámico que está en constante evolución, este se debe apoyar a través del uso de guiones gráficos como storyboards o tableros Kanban para representar el flujo de valor los procesos de entrega y así tener una forma práctica de controlar las cargas de trabajo sobre un equipo o un individuo, encontrar cuellos de botella y evitar sobrecargar a miembros del equipo de trabajo. Estas prácticas son elementos clave para conseguir mejores tasas de entrega, reducirlos riesgos del proyecto y generar efectos positivos en la cultura del equipo.

Fuente:

- ¿Qué son los límites de trabajo en curso? (Atlassian, a).
- Kanban Tool, Limitando el Trabajo en Progreso Kanbantool.
- Accelerate, Forsgren Forsgren et al. (2018).
- DORA's research program, Work in process limits (DORA's, e).

Tipo: Procesos

Subtipo: Lean

Categoría: Gestión Lean y Monitorización

Capacidad: Administrar y establecer límites para el trabajo en progreso (Work-In-Progress)

Ultima modificación: 20220303

19. **¿La empresa utiliza herramientas para visualizar y controlar el avance de los proyectos?**

Opciones de respuestas:

1 No, los proyectos se coordinan informalmente a juicio de expertos, no se genera documentación del estado del proyecto

2 No, los proyectos se coordinan formalmente, se genera documentación del avance del proyecto, pero esta no es publica o de fácil acceso.

3 Si, solo se utilizan repositorios de documentos con información no siempre actualizada del estado del proyecto, no todos pueden acceder a ella.

4 Si, se utilizan herramientas para visualizar el avance del proyecto, pero solo es visible por unos pocos miembros del equipo

5 Si, se utilizan herramientas para que todos los miembros del equipo puedan ver el estado actual del proyecto

Justificación

Se busca conocer los mecanismos que utiliza la compañía para visualizar el avance del proyecto y la visibilidad que tiene esta información entre equipo de trabajo y los clientes

Plan de acción

1 La compañía debe adoptar la cultura de generar documentación mínima necesaria para poder obtener fácilmente el estado actual de un proyecto.

2 La compañía debe propiciar el fácil acceso de todo el equipo una documentación centralizada del proyecto para conocer el estado y avance del proyecto manualmente.

3 Se debe dar el salto de rendimiento y agilidad en la gestión de los proyectos a través de la implementación de una herramienta que permita conocer los detalles generales del proyecto y su avance.

4 Las herramientas de gestión de proyecto deben permitir herramientas visuales para organizar el trabajo, como tableros Kanban, historias de usuario de fácil acceso para los miembros del equipo de trabajo.

5 Tener herramientas de visualización para que todos los miembros del equipo puedan consultar y gestionar el estado de la cadena de valor de proyecto, debe ser adoptado por la compañía como una tarea de mejora continua entre todos los miembros del equipo, en donde resalten elementos como: reflejar información relevante e importante para los objetivos del trabajo, permitir ejecutar o tomar acciones, fácil de entender, mantener actualizada y precisa. El objetivo es mantener al equipo informado con información útil, exacta y actualizada para identificar y quitar obstáculos en el rendimiento a través de elementos visuales como tableros Kanban, storyboards, tarjetas físicas, entre otras.

Fuente:

- Accelerate, Forsgren (Forsgren et al., 2018).
- DORA's research program, Visual management capabilities (DORA's, k).

Tipo: Herramientas

Subtipo: Monitoreo

Categoría: Gestión Lean y Monitorización

Capacidad: Visualizar el trabajo para monitorizar la calidad y comunicarlo al equipo

Ultima modificación: 20220313

20. **¿Como describiría su organización?**

Justificación

Se busca medir el tipo de organización basado en la siguiente escala del modelo Teal

1 Organizaciones Rojo, Lideradas por la Fuerza

2 Organizaciones Ámbar, Lideradas por el Mandante

3 Organizaciones Naranja, Lideradas a través de a la eficiencia

4 Organizaciones Vede, Lideradas a través de la responsabilidad

5 Organizaciones Teal

Mención especial al modelo Westrum

1 Patológicas (Orientadas al poder), equivalente al nivel 1 Rojo

2 Burocráticas (Orientado a reglas), equivalente al nivel 2 Ámbar

3 Generativas (orientado a rendimiento), equivalente a los niveles 3,4 y 5 Naranja, Verde y Teal

Opciones de respuestas

1 Regidas por el ejercicio del poder y el miedo, altamente reactivas.

2 Regida por roles formales en estructura piramidal escalable, top-down command and control, procesos estables y replicables

3 Objetivo vencer a la competencia, a través del crecimiento y la rentabilidad, gestión por objetivos.

4 Foco en la cultura y el empoderamiento para potenciar la motivación de los empleados.

5 Organización vista como un organismo vivo orientada a desarrollar todo su potencial, equipos autogestionados.

Plan de acción

1 Este tipo de organizaciones predomina en entornos normalmente caóticos, donde prima la necesidad de supervivencia y suelen operar de un modo más reactivo, en este caso se debe iniciar con un proceso de redefinición progresiva de su estructura para evolucionar a un modelo piramidal y adoptar un modelo de mando jerárquico en pro de que la organización pueda alcanzar objetivos a largo plazo.

2 Este es un modelo muy propio para estructuras altamente jerarquizadas donde predomina la burocracia y la rigidez organizacional y la antigüedad es un factor muy importante para el escalamiento. ejemplo fuerzas armadas, entidades del sector público o la iglesia. Para evolucionar este modelo la compañía debe aceptar el paradigma de constante evolución y adaptarse

a los cambios del entorno, donde prima el crecimiento para obtener beneficios, basados en el principio de la meritocracia, cumplimiento de objetivos individuales e innovación.

3 Este tipo de organizaciones se centran en el crecimiento y sus principales decisiones se toman a partir de un rendimiento económico, adaptándose a los cambios del entorno. Sus principales características son la meritocracia, los objetivos individuales y la innovación. El siguiente nivel para este tipo de organizaciones es ajustar su foco y dar prioridad a la cultura y el empoderamiento de sus miembros para que estén motivados en su trabajo.

4 Este tipo de organización utilizan los valores a la hora de tomar decisiones, operan bajo una estructura piramidal, dan prioridad a la cultura y al empoderamiento de sus miembros. El siguiente nivel, se basa enfocar el paradigma de la organización a un modelo evolutivo y autogestionado, donde la toma decisiones no se toma en la cima de la organización.

5 Las organizaciones que se clasifican como de alto rendimiento, son organizaciones que tienen como filosofía la mejora constante de sus procesos. En este nivel se deben garantizar que el equipo este comprometido y adopte prácticas laborales de alta colaboración, manejo de riesgos compartidos, orientación a las conciliaciones y motivación a la innovación. Este modelo se centra en que la organización pueda funcionar con mayores niveles de eficacia y eficiencia, sin jerarquía, eliminando estructuras piramidales donde los valores y el factor humano son los protagonistas.

Fuentes:

- Reinventar las organizaciones (Laloux and Maturana, 2016).
- Thepowermba, Organizaciones Teal (ThePowerMBA).
- Euroforum, ¿Sabes lo que son las Organizaciones TEAL?, (Euroforum).
- Accelerate, Forsgren (Forsgren et al., 2018).
- DORA's research program, DevOps culture: Westrum organizational culture (DORA's, g).

Tipo: Cultura

Subtipo: Organización

Categoría: Cultura

Capacidad: Implementar una cultura tipo generativa

Ultima modificación: 20220303

21. **¿En los últimos 6 meses que espacios has tenido para aprender nuevas tecnologías, herramientas y prácticas?**

Justificación

Se busca conocer si la empresa propicia espacios para el aprendizaje y experimentar con nuevas herramientas y tecnologías.

Estos espacios se dan a criterio y disponibilidad de tiempo de la compañía, normalmente se da durante el proyecto para lograr mejora continua, en ocasiones se da al final del proyecto o en la brecha entre el cierre e inicio de un nuevo proyecto.

El proyecto de investigación y evaluación de DevOps DORA muestra que una cultura organizacional que valora el aprendizaje contribuye al rendimiento de la entrega de software con las siguientes características:

- Mayor frecuencia de implementación.
- Plazos de entrega reducidos para los cambios, tiempo para restablecer el servicio y cambio en la tasa de fallas.
- Cultura de equipo sólida.

Opciones de respuestas

1 No he tenido espacio, voluntad, recursos o necesidad

2 Solo lo puedo hacer en tiempo personal con recursos propios

3 Si, en una mezcla entre espacio individual y laboral donde cada parte cubre sus gastos

4 Solo para cubrir las necesidades particulares que requiera la empresa con recursos de la compañía.

5 Si, tengo la opción de proponer temas y tecnologías para aprender lo que se requiera adicional de lo que demande los proyectos de la empresa con recursos de la compañía.

Plan de acción

1. Propiciar los espacios de motivación para que los empleados adquieran nuevos conocimientos con el fin de dar el primer paso de una cultura de aprendizaje

2. Crear un presupuesto interno acorde a la capacidad de la empresa para apoyar la capacitación o asistencia a eventos de nuevas tecnologías y casos de éxito de los miembros del equipo acorde a cuanto la empresa valora el clima de aprendizaje.

3 y 4, Una cultura organizativa debe tratarse como una inversión, para ello debe iniciar propiciando espacios de aprendizaje y compartir información, crear presupuestos de capacitación y apoyar el intercambio de conocimiento en eventos tecnológicos.

5. La cultura del aprendizaje está dada en el momento que la compañía la vea y la trate como una inversión a las capacidades técnicas de su personal y está directamente relacionada con tener equipos empoderados y comprometidos

con la mejora continua, reflejándose en mayor frecuencia de implementación, menores tiempo de entrega

Fuentes:

- Accelerate, Forsgren (Forsgren et al., 2018).
- DORA's research program, DevOps culture: Learning culture (DORA's, i).

Tipo: Cultura

Subtipo: Aprendizaje continuo

Categoría: Cultura

Capacidad: Animar y apoyar al equipo a aprender

Ultima modificación: 20211223

22. **¿Como califica la división de responsabilidades entre los miembros del equipo de trabajo?**

Justificación

Se busca conocer lo bien que interactúan los equipos de desarrollo, operaciones y seguridad, que tradicionalmente trabajan en silos.

Se busca determinar el grado de colaboración y como se comparten las responsabilidades del proyecto

La investigación de DORA demuestra que una cultura generativa y de alta confianza predice la entrega de software y el rendimiento organizacional en la tecnología.

Opciones de respuestas

1 No hay asignación de responsabilidades definidos, cada uno hace lo que considera y los miembros del equipo eluden las responsabilidades

2 Las responsabilidades están limitadas y focalizadas a cada miembro del equipo de trabajo

3 Las responsabilidades se comparten parcialmente

4 Los riesgos son compartidos entre todos los miembros del equipo y existe alguien que asigna y determina responsabilidades

5 Las responsabilidades y riesgos se comparten y son producto de un acuerdo entre todos los miembros del equipo

Plan de acción

1. Identificar y referenciar las responsabilidades de cada miembro del equipo con el fin de generar una visión global del flujo de información las todas las actividades del proceso de entrega de software que se implementan, así crear

canales de comunicación alineados para un mismo propósito e iniciar la toma de decisiones compartida.

2 y 3 Propiciar equipos multidisciplinarios con representantes de cada área funcional (desarrolladores, analistas, personal de calidad, infraestructura, seguridad, etc.) donde existan canales de comunicación abiertos y colaborativos para que la toma de decisiones sea compartida desde el inicio del ciclo de vida de la aplicación para que todos los miembros compartan la responsabilidad de compilar, implementar y mantener un producto de software. La automatización también reduce el riesgo y, con la herramienta adecuada, se puede emplear la colaboración.

4. Se debe definir equipos multidisciplinarios con representantes de cada área funcional que intervenga en el proceso de desarrollo de las aplicaciones con el fin que todos compartan responsabilidades, tomen decisiones compartidas, asuman riesgos compartidos, con el fin de enfocar esfuerzos en mejorar incidencias y problemas del día a día como cuellos de botella con mayor eficacia. La empresa no debe abordar el trabajo técnico solo como un desafío de tecnología y procesos.

5. Adoptar un cambio cultural es una tarea de aprendizaje y mejora continua en donde la compañía debe enfocarse en revisar y fortalecer aspectos de la cultura organizacional como: la alta colaboración, la toma de riesgos y decisiones compartidos, el fortalecimiento de la conciliación, el fomento de la confianza, la gestión proactiva de errores y el potenciamiento de la innovación. La cultura influye en la forma en que fluye la información en una organización, modelos como la cultura generativa de Westrum propician mejores ambientes laborales, alta confianza, calidad y rendimiento de los equipos por ende mejores tiempos de entrega y productividad.

Fuente

- DORA's research program, DevOps culture: Westrum organizational culture (DORA's, g).
- Accelerate, Forsgren (Forsgren et al., 2018).

Tipo: Cultura

Subtipo: Colaboración

Categoría: Cultura

Capacidad: Apoyar y facilitar la colaboración entre los equipos

Ultima modificación: 20220314

23. **¿Como calificaría el trabajo en equipo entre desarrolladores y operación?**

Justificación

Se busca conocer el grado de cooperación que existe entre desarrolladores e infraestructura en la compañía.

Esta escala permite conocer si no hay comparación alguna o si se tienen equipos que cooperan y se ayudan constantemente

El primer paso para la transformación digital de la compañía es derribar lo que en DevOps denominan muro de confusión donde desarrollo y operaciones van por caminos separados en una especie de burbuja donde cada uno se limita a hacer sus funciones sin tener en cuenta a los otros.

La colaboración entre equipos es el eje fundamental para mejorar los tiempos de entrega y calidad de los productos de software. Reforzar y adoptar el pensamiento sistémico entre las dos áreas es fundamental para mejorar la comunicación e iniciar la toma de decisiones compartidas durante todo el ciclo de vida de una aplicación

Opciones de respuestas

1 No hay cooperación entre áreas

2 Cooperación baja menos del 20 % de las tareas conjuntas

3 Cooperación moderada, entre un 20 y 50 % de tareas conjuntas

4 Cooperación alta, entre un 50 y 70 % de tareas conjuntas

5 Cooperación continua y constante, más del 70 % de las tareas son conjuntas

Plan de acción

1 Para esto se debe iniciar con un cambio cultura progresivo adoptando un pensamiento sistémico entre las dos áreas bajo la premisa todo está conectado con lo demás, un primer paso es encontrar sentido al porque trabajar en conjunto identificando eventos, relaciones y actividades comunes entre las dos áreas y así tener un marco común para realizar tareas conjuntas particulares. Una alternativa es emplear actividades de gamificación entre los equipos para reforzar el clima laboral entre las áreas.

2 Una buena práctica para iniciar es que cada equipo involucre a miembros del otro equipo en sus reuniones clave, por ejemplo, que el equipo de desarrollo invite a representantes del equipo de infraestructura a reuniones de planeación, evaluación y demostración de proyectos en curso, por otro lado, que infraestructura involucre a representantes de desarrollo en reuniones que tengan algún impacto sobre las aplicaciones. Esto es un proceso progresivo que no se alcanza de un día para otro.

3 En este nivel la cooperación entre desarrollo y operaciones los equipos, colaboran entre sí y toman decisiones compartidas durante las diferentes etapas del ciclo de vida de una aplicación a través de un pensamiento sistémico y entregar valor los clientes. Otro factor para tener en cuenta es que en esta etapa los equipos deben compartir responsabilidades propias de cada especialidad para lograr un flujo de trabajo continuo.

4 Adoptar un pensamiento sistémico para la entrega de valor y un flujo de trabajo continúo enfocado en la colaboración, toma de decisiones y riesgos compartidos que permitan la distribución efectiva del conocimiento para poder identificar incidencias y cuellos de botella en la entrega de valor al cliente, a través de procesos de retroalimentación rápido y continua de los clientes.

5 El trabajo colaborativo en equipos interdisciplinarios es una capacidad cultural que debe fundamentarse en la mejora constante, en donde la compañía debe potenciar prácticas como, responsabilidad compartida durante el ciclo de vida de la aplicación (compilar, implementar y mantener un producto), toma de decisiones conjunta, empoderamiento individual para la toma de decisiones, fomentar una cultura de experimentación y aprendizaje continuo. En este nivel, no puede existir el denominado muro de confusión DevOps entre desarrollo, infraestructura y demás miembros del equipo, alcanzando así un grado donde se comparte un objetivo común y se tiene un plan para alcanzarlo juntos.

Fuente:

- Udemy, DevOps - Introducción y Fundamentos no técnicos (Udemy).
- Accelerate, Forsgren 2018 (Forsgren et al., 2018).
- The DevOps Handbook First Edition (Kim et al., 2016).
- Atlassian, Marco CALMS (Atlassian, b).

Tipo: Cultura

Subtipo: Colaboración

Categoría: Cultura

Capacidad: Apoyar y facilitar la colaboración entre los equipos

Ultima modificación: 202203014

24. **¿La compañía provee de todas las herramientas necesarias para hacer bien el trabajo?**

Opciones de respuestas

1 No se cuenta con las herramientas claras para los proyectos

2 La empresa tiene un stack tecnológico a los que los miembros del equipo deben adaptar los desarrollos.

3 Solo se provee las herramientas mínimas para cumplir cada proyecto

4 La empresa provee solo las herramientas que varios comités técnicos aprueben.

5 La empresa invierte y provee las herramientas necesarias para que el equipo de desarrollo logre sus objetivos de negocio.

Justificación

Se quiere conocer, el grado de compromiso de la compañía para brindar las herramientas necesarias para el desarrollo e implementación de los proyectos

Plan de acción

1. La compañía debe definir un stack tecnológico base en el que los desarrolladores puedan usar como marco de referencia para el desarrollo de los proyectos, teniendo en cuenta que cada proyecto posee condiciones particulares que no se pueden resolver con las mismas herramientas.

2. y 3. La compañía debe permitir la flexibilidad de ajustar el stack tecnológico corporativo para solventar las particularidades de un proyecto y así brindar al equipo de trabajo las herramientas y recursos adecuados para hacer el trabajo, no todo se puede resolver de la mejor manera con una misma herramienta universal.

4. La satisfacción laboral del equipo es directamente proporcional a empoderar al equipo en la toma de decisiones sobre las herramientas a utilizar en cada proyecto, apoyados en procesos de aprobación ligeros sin burocracia, garantizando poder asegurar las herramientas y recursos necesarios para hacer el trabajo en el menor tiempo posible.

5. La satisfacción laboral impacta la productividad, por lo que es importante que la compañía adopte una filosofía de mejora y aprendizaje continuo, brindando herramientas y recursos adecuados para realizar el trabajo, por ejemplo, recursos técnicos, como el acceso a servidores o entornos necesarios para desarrollar y probar, o recursos necesarios para aprender y desarrollar nuevas habilidades. Otro elemento a destacar, es generar los espacios de empoderamiento con los miembros del equipo para que sientan que su opinión de valora y poder aprovechar sus habilidades y capacidades.

Fuente:

- Accelerate, Forsgren (Forsgren et al., 2018).
- DORA's research program, DevOps culture: Job satisfaction (DORA's, s).

Tipo: Cultura

Subtipo: acceso a recursos

Categoría: Cultura

Capacidad: Proporcionar recursos y herramientas para que el trabajo sea significativo para los individuos

Ultima modificación: 20220303

25. **¿Como describe a los lideres de su organización?**

Opciones de respuestas

1 Ejerce todo el poder y toma las decisiones, mientras que el resto del equipo debe acatar las órdenes sin opción a proponer su visión.

2 Se empeña en realizar un seguimiento absoluto del trabajo de sus subordinados.

3 Comparte la información que tiene y está abierto a debatir las opciones con todo el quipo

4 Deja a su equipo libertad en la toma de decisiones en los puntos o áreas de influencia del equipo

5 Alientan, inspiran y motivan a los empleados a innovar y crear cambios que ayudarán a crecer y dar forma al éxito futuro de la empresa

Justificación

Se busca medir el tipo de liderazgo en la organización basado en la siguiente escala de tipo de lideres, donde cada nivel corresponde a una de las opciones de respuesta

1 Líderes autocráticos

2 Lideres controladores (micromanagement)

3 Líderes democráticos

4 Líderes liberales

5 Líder transformador

La investigación DORA define cinco características de un líder transformador son las siguientes:

- **Visión:** Comprende con claridad hacia dónde se dirige la organización y su equipo y dónde quiere que esté su equipo en cinco años.
- **Comunicación inspiradora:** Se expresa de una manera positiva en cuanto al equipo, hace comentarios que provocan que los empleados se sientan orgullosos de formar parte de su organización y alienta a las personas a ver las condiciones cambiantes como situaciones llenas de oportunidades.
- **Estímulo intelectual:** Desafía a los miembros del equipo a pensar en problemas antiguos de maneras nuevas y a reconsiderar algunas suposiciones básicas sobre sus trabajos. Tiene ideas que obligan a los miembros del equipo a reconsiderar algunos aspectos que nunca se habían cuestionado.
- **Liderazgo de asistencia:** Evalúa los sentimientos del otro antes de actuar, se comporta de un modo que tiene en cuenta las necesidades personales de los demás y comprende que se debe estar atento a los intereses de los miembros del equipo.
- **Reconocimiento personal:** Elogia a los miembros del equipo cuando realizan un trabajo mejor que el promedio, reconoce mejoras en la calidad del trabajo de los miembros del equipo y felicita de forma personal a los

miembros del equipo de forma personal cuando realizan trabajos sobresalientes.

El equipo de Dora creó un modelo para probar la relación entre liderazgo, practicas técnicas y de gestión de productos que se muestra en la B.1

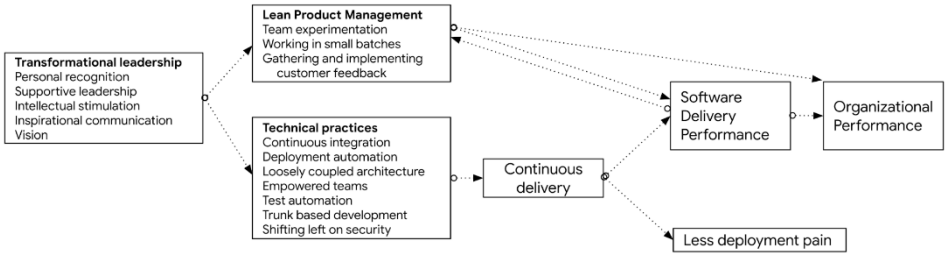

Figura B.1: Cultura y liderazgo transformacional
Imagen tomada de: (Forsgren et al., 2018)

Plan de acción

1 Se debe iniciar un proceso para transformar a los lideres de la organización con el fin que desarrollen y potencien capacidades de liderazgo transformativo que permita inspirar y motivar a quienes están a su alrededor se identifiquen con la organización, se comprometan con el apoyo a los objetivos organizativos. Esto genera ambientes de mayor confianza y apoyo dentro del equipo, a su vez se refleja en mejora de rendimiento y productividad.

2 Se debe iniciar un proceso para transformar a los lideres de la organización hacia un enfoque de líder tecnológico en donde debe propiciar espacios de mayor confianza, apoyo, trabajo colaborativo y abierto con todos los miembros del equipo, esto se refleja en mejora en el rendimiento y productividad. (Accelerate) "un buen líder influye en la capacidad de un equipo para entregar código, crear la arquitectura de sistemas, aplicar principios LEAN en la manera de que el equipo gestiona su trabajo y desarrolla productos".

3 Se debe iniciar un proceso para transformar a los lideres para que potencien las habilidades de comunicación y el apoyo dentro del equipo de trabajo, que son características asociados a un líder transformativo y estas están muy correlacionadas con el rendimiento en la entrega de software.

4 La organización debe apoyar para que sus lideres se involucren en los procesos de transformación tecnológica y dar el siguiente paso para evolucionar de simples lideres a lideres transformacionales con la capacidad de inspirar y motivar a quienes están a su alrededor. Permitiendo generar ambientes colaborativos dentro del equipo, por el lado técnico un buen líder influye en capacidades para entregar código, crear buenas arquitecturas, aplicar principios LEAN a la gestión del trabajo.

5 El papel de líderes y directores de una compañía en los procesos de transformación tecnológica va más allá de coordinar a grupos de personas, estos deben ser los principales actores de la cultura de mejoramiento constante, a continuación, se resaltan las características principales de un líder transformador que la compañía deben adoptar o potenciar:

- Visión: saber dónde quiere que esté el equipo y la organización en el mediano y largo plazo.
- Comunicación: inspirar y mantener motivado al equipo.
- Estimulación intelectual: ayudar al equipo a reconsiderar y buscar alternativas de solución problemas.
- Comprensivo: tener en cuenta las necesidades humanas del equipo.
- Reconocimiento personal: elogiar el trabajo sobresaliente del equipo.

La influencia positiva (o negativa) del liderazgo impacta indirectamente el rendimiento en la entrega de software y rendimiento organizativo, un buen liderazgo facilita que los equipos adopten prácticas técnicas y eficientes en la administración de productos. El liderazgo no solo está representado por ejecutivos y administradores, cualquier miembro del equipo que practique comportamientos de liderazgo puede ser líder.

Fuente:

- Accelerate, Forsgren (Forsgren et al., 2018).
- DORA's research program, DevOps culture: Transformational leadership (DORA's, d).

Tipo: Cultura

Subtipo: Liderazgo

Categoría: Cultura

Capacidad: Apoyar el liderazgo transformacional

Ultima modificación: 20220315

Apéndice C

Plan de acción

Una vez se obtuvo el resultado de la encuesta del modelo de capacidades de las áreas de desarrollo administrativo e infraestructura del CIAT, con el fin de tener la opción de respuesta que represente cada pregunta se calculó la moda para cada una de las preguntas del modelo propuesto. Acorde con la definición del modelo de capacidades cada opción de respuesta tiene asociado un único plan de acción.

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Entrega continua	Utilizar control de versiones para todos los artefactos de producción	¿La compañía utiliza sistemas de control de versiones?	Aceptable	Se debe avanzar para tener sistemas de control de versiones distribuido y administrado con sistemas de alojamiento web como por ejemplo Github, GitLab, Bitbucket que faciliten la integración con otros procesos del ciclo de vida de las aplicaciones con el fin de poder iniciar la ejecución de tareas de compilación, construcción de las aplicaciones.

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Entrega continua	Automatizar el proceso de despliegue	¿Cómo es el proceso de despliegue de soluciones de la compañía?	Oportunidades de mejora	Acordar y adoptar un proceso de responsabilidades compartidas y colaborativo entre las áreas de desarrollo e infraestructura desde la etapa inicial del proyecto en la que se definan roles y responsabilidades, y que herramientas se van a utilizar para las diferentes etapas del ciclo de vida de la aplicación, como gestionar la compilación, construcción, ejecución de pruebas y despliegue en producción de la aplicación.

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Entrega continua	Implementar integración continua	¿De qué manera la compañia realiza la compilación, construcción y empaquetamiento del producto final?	Oportunidades de mejora	Se debe iniciar la migración de las aplicaciones a una herramienta especializada para la gestión de proyectos de desarrollo de software por ejemplo Maven, Gradle, Nuget, etc, iniciando con la preparación del proyecto, que consiste en crear un inventario de dependencias y definir la fuente o repositorio donde se obtendrán, segundo, adaptar el proyecto acorde a las especificaciones particulares de la herramienta elegida para que el proyecto se pueda gestionar a partir de comandos repetibles y así cubrir las etapas de compilación, construcción, pruebas y empaquetamiento del proyecto y usarse en procesos posteriores.

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Entrega continua	Utilizar métodos de desarrollo "Trunk-Based"	¿La compañia usa alguna estrategia de gestion de la configuracion?	Oportunidades de mejora	Adoptar un sistema de control de versiones como Git, Subversion y Mercurial, etc. que permita controlar fácilmente los cambios en código fuente y demás archivos propios del proyecto durante todo su ciclo de vida, sin desgastar tiempo y esfuerzo del equipo de trabajo en tareas manuales de consolidación de cambios. La gestión de la configuración es un factor determinante en el rendimiento de los equipos de trabajo y un elemento clave en la entrega continua de productos, por lo que este es el primer paso para mejorarlo.

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Entrega continua	Implementar pruebas automáticas	¿La compañía implementa procesos de pruebas de los desarrollos?	Oportunidades de mejora	Se recomienda adoptar herramientas o frameworks diseñados para la implementación de pruebas (por ejemplo, JUnit, QUnit, Jasmine, Nunit, entre otros) acorde con el lenguaje de programación en que se desarrolla cada proyecto, con el fin de poder automatizar la ejecución de las pruebas del proyecto, inicialmente se puede enfocar en las pruebas unitarias e ir escalando a los demás tipos de pruebas, con el fin de que estas puedan ser ejecutadas automáticamente en las diferentes etapas del ciclo de vida de la entrega del proyecto.

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Entrega continua	Implementar gestión de los datos de prueba	¿Cómo definen que datos utilizar a la hora de probar las aplicaciones?	Oportunidades de mejora	Se debe adoptar un proceso colaborativo entre desarrolladores, infraestructura y calidad para definir el criterio, alcance y los datos que se utilizarán durante las pruebas. Se recomienda priorizar las pruebas unitarias ya que son la base de la pirámide de pruebas y progresivamente ir escalando en los demás tipos de pruebas (aceptación, integración, etc.). Hay que asegurar que existan datos de pruebas adecuados, consistentes y realistas para que la ejecución de los casos de pruebas permita asegurar el correcto funcionamiento de la aplicación. El proceso de pruebas se debe adoptar desde las fases iniciales de los proyectos, a través de prácticas de desarrollo que promuevan la implementación de pruebas, como, por ejemplo, (Test-Driven Development).

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Entrega continua	Incorporar los aspectos de seguridad en las fases de diseño y pruebas (shift left de la seguridad)	¿Cuál es el nivel de relevancia que tiene la seguridad en el ciclo de vida del desarrollo de software?	Oportunidades de mejora	Se debe incluir una capa de seguridad al equipo de trabajo desde la fase de inicio del ciclo de vida de la aplicación para que apoyen procesos diseño arquitectónico y pruebas, con el fin de tener definiciones claras de librerías y herramientas aprobadas desde el punto de vista de seguridad, también, la adopción de pruebas de seguridad lo más automatizado posible en todas las fases del ciclo de vida del desarrollo.
Entrega continua	Implementar entrega continua de software	¿Ejecutan tareas automáticas al registrar algún cambio?	Oportunidades de mejora	Se debe asegurar que el código fuente del proyecto tenga la capacidad de compilarse y generar un producto final o desplegable por medio de la ejecución de comandos propios de una herramienta de gestión de librerías.

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Arquitectura	Utilizar arquitecturas de bajo acoplamiento	¿Qué características tiene la infraestructura de la organización?	Oportunidades de mejora	La compañía debe iniciar un plan de organización y estandarización del stack tecnológico existente con el fin de conocer que elementos pueden ser compartidos y reutilizables en diferentes condiciones, con el fin de que en un futuro se pueda eliminar la necesidad de hacer configuraciones y actualizaciones manuales de las piezas individuales de hardware. Por otro lado, identificar y depurar los conocidos entornos particulares o exclusivos que no pueden reproducirse automáticamente.

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Arquitectura	Permitir a los equipos definir sus propias arquitecturas	¿Cómo la organización define las herramientas que utilizan los equipos en los proyectos?	Oportunidades de mejora	Una de las practicas DevOps, es minimizar el uso de anti patrones como martillo de oro, bala de plata en donde todos los problemas se pretenden solucionar con una única herramienta o metodología, por lo que un stack tecnológico de una organización debe permitir ajustes de acuerdo con las necesidades de los proyectos.
Producto y Procesos	Recolectar e implementar las recomendaciones (feedback) de los clientes	¿La compañía recoge el feedback de los clientes respecto a sus productos de software?	Oportunidades de mejora	Se debe iniciar con la adopción progresiva de la práctica de recolectar el feedback de los clientes durante todo del ciclo de vida del producto, en especial en las etapas tempranas del proyecto, estas opiniones se deben tener en cuenta para el diseño del producto.

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Producto y Procesos	Hacer visible el flujo de trabajo durante todo el ciclo	A nivel de proyectos, ¿cómo fluye la comunicación entre los miembros del equipo de trabajo y los directivos?	Oportunidades de mejora	La compañía debe asegurar que el equipo de trabajo conozca y participe desde el inicio de proyecto en la definición del flujo de trabajo para el desarrollo del producto, que va desde la solicitud del cliente hasta la entrega del producto, definiendo bloques de trabajo, el personal involucrado, las activades a desarrollar, los tiempos estimados de las tareas. Por último, se lleva a herramientas de visualización para que todo el equipo pueda consultar el estado de cada etapa durante todo el ciclo de vida del desarrollo de la aplicación.

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Producto y Procesos	Trabajar con lotes o cantidades pequeñas	¿Cómo los equipos dividen el trabajo a la hora de iniciar el desarrollo de productos y funcionalidades para tener un mínimo producto viable?	Aceptable	Iniciar la adopción de un enfoque en donde prime la división del trabajo en partes pequeñas de forma independiente, valioso, verificable y negociable, logrando que la entrega de productos mínimos viables pueda adaptase a las necesidades del cliente. Descomponer el trabajo en funcionalidades que permitan un desarrollo rápido en periodos menores a 3 meses.
Producto y Procesos	Fomentar y habilitar al equipo para realizar experimentos	¿Cuál es nivel de libertad que tiene el equipo de desarrollo para elegir sus propias herramientas en función de lo que sea mejor para el usuario?	Oportunidades de mejora	Se debe iniciar con un acuerdo entre desarrolladores, lideres de desarrollo y comité técnico (si existe) y gerencia para evaluar al menos 2 veces al año el stack tecnológico vigente y decidir adoptar nuevas herramientas para mejorar el rendimiento y funcionalidades de nuevos desarrollos.

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Gestión Lean y Monitorización	Tener un proceso liviano para la aprobación de cambios	¿Cómo es proceso de gestión de cambios de la compañía?	Oportunidades de mejora	Se debe iniciar la adopción de procedimientos y herramientas que permitan gestionar y visualizar los cambios de un proyecto, lo principal es darles orden y visibilidad a dichas solicitudes para que estas puedan ser analizadas y aprobadas por los mecanismos y metodologías que use la compañía (revisión por pares, revisión inter equipo, evaluación 360).
Gestión Lean y Monitorización	Monitorizar las aplicaciones y la infraestructura para tomar decisiones de negocio	¿La organización monitorea la infraestructura y las aplicaciones?	Aceptable	Los datos de monitoreo de la infraestructura y las aplicaciones deben ser de fácil acceso para todos los miembros del equipo y no solo de uso exclusivo de un grupo particular, el principal objetivo es obtener datos relevantes a través de una supervisión inteligente y focalizada en las áreas claves que tengan el potencial de generar valor para la compañía y de fácil acceso.

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Gestión Lean y Monitorización	Monitorizar la salud de los servicios y aplicaciones proactivamente	¿Cómo se maneja las notificaciones que vienen del monitoreo de los sistemas?	Fortalezas	El monitoreo preventivo y proactivo se debe enfocar en anticipar situaciones de alta demanda del producto de software, en función a tasas de cambio de los recursos del servidor, ejemplo incremento de más de 40

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Gestión Lean y Monitorización	Administrar y establecer límites para el trabajo en progreso (Work-In-Progress)	¿Cómo controlan la carga de trabajo en los proyectos?	Aceptable	No caer en la tentación de ajustar arbitrariamente la cantidad de trabajo en curso de un equipo de trabajo, este deber ser un proceso de revisión constante y acordado entre los miembros del equipo para que a partir de un análisis fundamentado en el rendimiento y en el cumplimiento de tareas se definan las diferentes estrategias para regular la cantidad de trabajo en progreso, por ejemplo, limitar la cantidad de asignación de trabajo al paso del proceso más lento y respetando esos límites definidos, analizando y buscando mejorar los factores que contribuyen a los retrasos.
Gestión Lean y Monitorización	Visualizar el trabajo para monitorizar la calidad y comunicarlo al equipo	¿La empresa utiliza herramientas para visualizar y controlar el avance de los proyectos?	Oportunidades de mejora	La compañía debe adoptar la cultura de generar documentación mínima necesaria para poder obtener fácilmente el estado actual de un proyecto.

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Cultura	Implementar una cultura tipo generativa	¿Como describiría su organización?	Oportunidades de mejora	Este es un modelo muy propio para estructuras altamente jerarquizadas donde predomina la burocracia y la rigidez organizacional y la antigüedad es un factor muy importante para el escalamiento. ejemplo fuerzas armadas, entidades del sector público o la iglesia. Para evolucionar este modelo la compañía debe aceptar el paradigma de constante evolución y adaptarse a los cambios del entorno, donde prima el crecimiento para obtener beneficios, basados en el principio de la meritocracia, cumplimiento de objetivos individuales e innovación.

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Cultura	Animar y apoyar al equipo a aprender	¿En los ultimos 6 meses que espacios has tenido para aprender nuevas tecnologías, herramientas y prácticas?	Oportunidades de mejora	Propiciar los espacios de motivación para que los empleados adquieran nuevos conocimientos con el fin de dar el primer paso de una cultura de aprendizaje

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Cultura	Apoyar y facilitar la colaboración entre los equipos	¿Como califica la división de responsabilidades entre los miembros del equipo de trabajo?	Oportunidades de mejora	Propiciar equipos multidisciplinarios con representantes de cada área funcional (desarrolladores, analistas, personal de calidad, infraestructura, seguridad, etc.) donde existan canales de comunicación abiertos y colaborativos para que la toma de decisiones sea compartida desde el inicio del ciclo de vida de la aplicación para que todos los miembros compartan la responsabilidad de compilar, implementar y mantener un producto de software. La automatización también reduce el riesgo y, con la herramienta adecuada, se puede emplear la colaboración.

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Cultura	Apoyar y facilitar la colaboración entre los equipos	¿Como calificaría el trabajo en equipo entre desarrolladores y operación?	Oportunidades de mejora	Una buena práctica para iniciar es que cada equipo involucre a miembros del otro equipo en sus reuniones clave, por ejemplo, que el equipo de desarrollo invite a representantes del equipo de infraestructura a reuniones de planeación, evaluación y demostración de proyectos en curso, por otro lado, que infraestructura involucre a representantes de desarrollo en reuniones que tengan algún impacto sobre las aplicaciones. Esto es un proceso progresivo que no se alcanza de un día para otro.

Dimensión	Capacidad	Pregunta	Capacidad	Descripción del plan de acción
Cultura	Proporcionar recursos y herramientas para que el trabajo sea significativo para los individuos	¿La compañía provee de todas las herramientas necesarias para hacer bien el trabajo?	Oportunidades de mejora	La compañía debe permitir la flexibilidad de ajustar el stack tecnológico corporativo para solventar las particularidades de un proyecto y así brindar al equipo de trabajo las herramientas y recursos adecuados para hacer el trabajo, no todo se puede resolver de la mejor manera con una misma herramienta universal.
Cultura	Apoyar el liderazgo transformacional	¿Como describe a los lideres de su organización?	Aceptable	Se debe iniciar un proceso para transformar a los lideres para que potencien las habilidades de comunicación y el apoyo dentro del equipo de trabajo, que son características asociados a un líder transformativo y estas están muy correlacionadas con el rendimiento en la entrega de software.

Tabla C.1: Plan de acción resultante de la evaluación CIAT

Bibliografía

AgilityHealth. DevOps Health Radar Assessment. URL https://agilityhealthradar.com.

A. Asociación Colombiana de Ingenieros de Sistemas. El uso de agile y devops genera un aumento de hasta un 60ganancias de las productoras de software, 2018. URL https://acis.org.co/portal/content/NoticiaDelSector/el-uso-de-agile-y-devops-genera-un-aumento-de-hasta-un-60-en-las-ganancias-de-

D. R. Atlassian. ¿qué son los límites de trabajo en curso?, a. URL https://www.atlassian.com/es/agile/kanban/wip-limits.

I. B. Atlassian. Marco CALMS, b. URL https://www.atlassian.com/es/devops/frameworks/calms-framework.

K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas. Manifiesto por el desarrollo Ágil de software, 2001. URL https://agilemanifesto.org/iso/es/manifesto.html.

M. Coupland. *DevOps Adoption Strategies: Principles, Processes, Tools, and Trends*. Packt Publishing Ltd, 2021. ISBN 9781801076326.

V. L. Cruz and A. B. Albuquerque. A devops introduction process for legacy systems. *Proceedings - 2018 44th Latin American Computing Conference, CLEI 2018*, pages 139–148, 2018. doi: 10.1109/CLEI.2018.00025.

J. Davis, K. Daniels, and R. Daniels. *Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale*, volume 53. O'Reilly Media, Inc., 1st edition, 2016. ISBN 1491926309.

R. de Feijter, S. Overbeek, R. van Vliet, E. Jagroep, and S. Brinkkemper. *DevOps competences and maturity for software producing organizations*, volume 318. Springer International Publishing, 2018. ISBN 9783319917030. doi: 10.1007/978-3-319-91704-7_16. URL https://link.springer.com/chapter/10.1007/978-3-319-91704-7_16.

Devopsgroup. The CALMS Model of DevOps. URL https://www.devopsgroup.com/insights/resources/diagrams/all/calms-model-of-devops.

R. P. DORA's. Devops tech: Version control, a. URL https://cloud.google.com/architecture/devops/devops-tech-version-control.

R. P. DORA's. DevOps tech: Cloud infrastructure, b. URL https://cloud.google.com/architecture/devops/devops-tech-cloud-infrastructure.

R. P. DORA's. DevOps tech: Test data management, c. URL https://cloud.google.com/architecture/devops/devops-tech-test-data-management.

R. P. DORA's. DevOps culture: Transformational leadership, d. URL https://cloud.google.com/architecture/devops/devops-culture-transformational-leadership.

R. P. DORA's. DevOps measurement: Work in process limits, e. URL https://cloud.google.com/architecture/devops/devops-measurement-wip-limits.

R. P. DORA's. DevOps measurement: Proactive failure notification, f. URL ciatnet.ciat.cgiar.org/sites/it/ExternalInformation/Forms/AllItems.aspx?RootFolder={%}2Fsites{%}2Fit{%}2FExternalInformation.

R. P. DORA's. DevOps culture: Westrum organizational culture, g. URL https://cloud.google.com/architecture/devops/devops-culture-westrum-organizational-culture.

R. P. DORA's. DevOps process: Working in small batches, h. URL https://cloud.google.com/architecture/devops/devops-process-working-in-small-batches.

R. P. DORA's. DevOps culture: Learning culture, i. URL https://cloud.google.com/architecture/devops/devops-culture-learning-culture.

R. P. DORA's. DevOps measurement: Monitoring systems to inform business decisions, j. URL https://cloud.google.com/architecture/devops/devops-measurement-monitoring-systems.

R. P. DORA's. DevOps measurement: Visual management capabilities, k. URL https://cloud.google.com/architecture/devops/devops-measurement-visual-management.

R. P. DORA's. DevOps tech: Trunk-based development, l. URL https://cloud.google.com/architecture/devops/devops-tech-trunk-based-development.

R. P. DORA's. DevOps tech: Continuous integration, m. URL https://cloud.google.com/architecture/devops/devops-tech-continuous-integration.

R. P. DORA's. Devops process: Streamlining change approval, n. URL https://cloud.google.com/architecture/devops/devops-process-streamlining-change-approval.

R. P. DORA's. DevOps tech: Continuous testing, o. URL https://cloud.google.com/architecture/devops/devops-tech-test-automation.

R. P. DORA's. DevOps tech: Continuous delivery, p. URL https://cloud.google.com/architecture/devops/devops-tech-continuous-delivery.

R. P. DORA's. DevOps process: Customer feedback, q. URL https://cloud.google.com/architecture/devops/devops-process-customer-feedback.

R. P. DORA's. DevOps tech: Shifting left on security, r. URL https://cloud.google.com/architecture/devops/devops-tech-shifting-left-on-security.

R. P. DORA's. DevOps culture: Job satisfaction, s. URL https://cloud.google.com/architecture/devops/devops-culture-job-satisfaction.

R. P. DORA's. DevOps process: Visibility of work in the value stream, t. URL https://cloud.google.com/architecture/devops/devops-process-work-visibility-in-value-stream.

R. P. DORA's. Devops tech: Empowering teams to choose tools, u. URL https://cloud.google.com/architecture/devops/devops-tech-teams-empowered-to-choose-tools.

R. P. DORA's. 2021 accelerate state of devops report. 2021. ISSN 0145-2134. URL https://cloud.google.com/devops/state-of-devops.

Euroforum. ¿sabes lo que son las organizaciones teal? URL https://www.euroforum.es/blog/sabes-lo-que-son-las-organizaciones-teal/.

N. Forsgren, J. Humble, and G. Kim. *Accelerate: The Science of Lean Software and DevOps Building and Scaling High Performing Technology Organizations*. IT Revolution Press, 1st edition, 2018. ISBN 978-1942788331. URL https://itrevolution.com/book/accelerate/.

I. Gartner. DevOps. URL https://www.gartner.com/en/information-technology/glossary/devops.

D. Gonzalez. *Implementing Modern DevOps*. Packt Publishing Ltd, 2017. ISBN 9781786466877.

L. Google. Devops research and assessment (dora). URL https://www.devops-research.com/research.html.

P. Hammant's. Trunk Based Development. URL https://trunkbaseddevelopment.com/.

Kanbantool. Limitando el Trabajo en Progreso. URL https://kanbantool.com/es/guia-kanban/fundamentos-de-kanban/limitando-el-trabajo-en-progreso.

G. Kim, P. Debois, J. Willis, and J. Humble. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press, first edition, 2016. ISBN 1942788002.

G. Kim, J. Humble, P. Debois, J. Willis, and N. Forsgren. *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations*. IT Revolution Press, 2021. ISBN 9781950508433.

F. Laloux and A. Maturana. *Reinventar las organizaciones : cómo crear organizaciones inspiradas en el siguiente estadio de la conciencia humana*. 2016. ISBN 8416601054.

K. Martin and M. Osterling. Value Stream Mapping : How to Visualize Work and Align Leadership for Organizational Tranformation, 2014. ISSN 1098-6596.

A. Mishra and Z. Otaiwi. DevOps and software quality: A systematic mapping. *Computer Science Review*, 38:100308, 2020. ISSN 15740137. doi: 10.1016/j.cosrev.2020.100308. URL https://doi.org/10.1016/j.cosrev.2020.100308.

H. A. Ordóñez and V. Buchelli. Lineamientos para la implementación del modelo calms de devops en mipymes desarrolladoras de software en el contexto sur colombiano. *Revista Guillermo de Ockham*, 18(1):81–91, 2020. ISSN 1794-192X. doi: 10.21500/22563202.4270.

OWASP Foundation. OWASP Devsecops Maturity Model. URL https://owasp.org/www-project-devsecops-maturity-model/.

P. Perera, R. Silva, and I. Perera. Improve software quality through practicing DevOps. *17th International Conference on Advances in ICT for Emerging Regions, ICTer 2017 - Proceedings*, 2018-Janua:13–18, 2017. doi: 10.1109/ICTER.2017.8257807.

Project Management Institute. *PMBOK guide + Agile practice guide : Agile practice guide bundle*. 2017. ISBN 9781628253825.

I. Red Hat. ¿qué es la infraestructura como código (iac)? URL https://www.redhat.com/es/topics/automation/what-is-infrastructure-as-code-iac.

Redhat. ¿qué son la integración y la distribución continuas (ci/cd)?, 2018. URL https://www.redhat.com/es/topics/devops/what-is-ci-cd.

S. Sharma. *The DevOps Adoption Playbook*. 2017. ISBN 9781119308744. doi: 10.1002/9781119310778.

J. Smith. *Operations Anti-Patterns, DevOps Solutions*. 2020. ISBN 978-1-61729-698-7. URL https://learning.oreilly.com/library/view/-/9781617296987/?ar.

I. Sommerville. *Ingeniería de Software*. PEARSON EDUCACIÓN, México, novena edi edition, 2011. ISBN 978-607-97357-1-5.

L. S. ThePowerMBA. Organizaciones Teal. URL https://www.thepowermba.com/es/blog/organizaciones-teal/.

Udemy. DevOps - Introducción y Fundamentos no técnicos. URL https://www.udemy.com/course/devops-en-espanol.

J. Verona, M. Duffy, and P. Swartout. *Learning DevOps: Continuously Deliver Better Software*. 2016. ISBN 9781787126619.