



Acelera los ciclos de vida del desarrollo de software AWS con la IA generativa

AWS Orientación prescriptiva



AWS Orientación prescriptiva: Acelera los ciclos de vida del desarrollo de software AWS con la IA generativa

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

Introducción	1
Objetivos	1
Destinatarios previstos	2
Experiencia de desarrollo	3
Uso de IA generativa	4
Marco 5-I	5
Descripción general del marco	6
Integración con el SDLC	8
Capacidades fundamentales	9
Administración de proyectos	16
Gestión de requisitos	19
Arquitectura y diseño	20
Colaboración	21
DevSecOps	23
Operación y mantenimiento	31
Asistentes de IA	34
Uso de análisis e información	36
Administración de conocimientos	39
Extensibilidad	40
Prácticas recomendadas	42
Cadena de herramientas integrada	42
DevSecOps oleoducto	43
Herramientas y prácticas colaborativas	43
Automatización de tareas	44
Revisión e iteración	44
Prácticas de gestión de proyectos	45
Administración de conocimientos	45
Extensibilidad y personalización	46
Optimización	46
Información basada en datos	46
Enfoque basado en plataformas	47
Medir el éxito	48
Velocidad de despliegue	49
Calidad del código	49

Eficiencia operativa	50
Productividad y satisfacción del equipo	50
Impacto empresarial	51
Conclusión	52
Recursos	52
Historial de documentos	54
.....	lv

Acelera los ciclos de vida del desarrollo de software AWS con la IA generativa

Chetan Makvana, Amazon Web Services

Abril de 2025 ([historial del documento](#))

La creciente demanda de software de alta calidad impulsa a las organizaciones a buscar constantemente formas de acelerar su ciclo de vida de desarrollo de software (SDLC). A medida que las organizaciones se esfuerzan por seguir siendo competitivas, es fundamental reducir el tiempo de comercialización y, al mismo tiempo, mantener o mejorar la calidad de los productos. Para hacer frente a estos desafíos, la experiencia de desarrollo de software debe evolucionar y utilizar tecnologías, metodologías y prácticas de vanguardia que optimicen los procesos y permitan a los equipos de desarrollo de software ser más productivos y creativos. El surgimiento de la experiencia de desarrollo de próxima generación marca un cambio significativo en la forma en que se concibe, crea, prueba e implementa el software. Integra una variedad de capacidades, que incluyen el desarrollo nativo de la nube, la automatización impulsada por la IA, la gestión avanzada de proyectos, las herramientas de colaboración y, en conjunto, mejoran la eficiencia y la DevSecOps eficacia del SDLC.

A la vanguardia de esta transformación está el auge de la IA generativa en la ingeniería de software. Según [Gartner](#), el 40% de los equipos de ingeniería de plataformas utilizarán la IA para mejorar todas las fases del SDLC de aquí a 2027, en comparación con solo el 5% en 2023. Este informe también afirma que los líderes de ingeniería de software ahora deben prepararse para adoptar la IA generativa en una gama más amplia de áreas que son fundamentales para el proceso de desarrollo. En otro informe, una [McKinsey](#) investigación muestra que las empresas con un índice de velocidad de desarrollo más alto aumentan sus ingresos entre 4 y 5 veces más rápido, obtienen una rentabilidad para los accionistas un 60% más alta y son un 55% más innovadoras. Al adoptar la IA generativa más allá de la simple generación de código, las organizaciones pueden alcanzar un nuevo nivel de eficiencia, productividad e innovación en sus flujos de trabajo de desarrollo de software. Esto puede reducir el esfuerzo manual, sacar a la luz información y aumentar la experiencia humana.

Objetivos

Este documento de estrategia describe un marco, las capacidades fundamentales, los casos de uso, las mejores prácticas y las métricas de éxito que pueden ayudarlo a acelerar su SDLC con la IA

generativa. Describe cómo integrar eficazmente la IA generativa en todas las etapas de desarrollo para mejorar la calidad y la eficiencia de los productos.

Este documento de estrategia puede ayudarle a usted y a su organización a alcanzar los siguientes objetivos:

- Implemente un marco, capacidades fundamentales, casos de uso, mejores prácticas y métricas de éxito para acelerar su SDLC con la IA generativa.
- Integre eficazmente la IA generativa en todas las etapas de desarrollo para mejorar la calidad del producto, la velocidad de lanzamiento y la eficiencia del desarrollo.
- Adáptese a la próxima generación de desarrollo de software incorporando tecnologías, metodologías y prácticas de IA de vanguardia que agilizan los procesos y capacitan a los equipos de desarrollo.

Destinatarios previstos

Este documento de estrategia está dirigido a líderes de TI, gerentes de ingeniería, directores de tecnología y equipos de desarrollo de software que desean acelerar su ciclo de vida de desarrollo de software mediante la aplicación de la IA generativa a sus prácticas de desarrollo.

Comprensión de la experiencia de desarrollo de software

La experiencia en desarrollo de software abarca el entorno, las herramientas y los procesos que utilizan sus equipos de desarrollo a lo largo del ciclo de vida del desarrollo de software (SDLC). Incluye el entorno de desarrollo integrado (IDE), las plataformas de colaboración, los marcos de pruebas, los sistemas de gestión del conocimiento, los procesos de implementación y mucho más.

Una experiencia de desarrollo bien diseñada agiliza los flujos de trabajo, reduce el esfuerzo manual y permite a sus equipos centrarse en tareas de gran valor, lo que, en última instancia, acelera su SDLC. Por ejemplo, al integrar a la perfección el IDE, el sistema de control de versiones y las herramientas de despliegue, permite a los desarrolladores escribir, probar e implementar código con mayor velocidad y eficiencia en comparación con una cadena de herramientas fragmentada que requiere transferencias manuales y cambios de contexto. Del mismo modo, la integración de un marco sólido de gestión del conocimiento ayuda a los equipos a acceder fácilmente a los conocimientos, las mejores prácticas y la documentación institucionales y a compartirlos. Esto mejora su productividad general y sus capacidades de resolución de problemas.

La experiencia de desarrollo de software tiene un impacto directo en el rendimiento general y el éxito de un equipo de desarrollo de software. Una experiencia subóptima puede llevar a lo siguiente:

- **Productividad reducida:** las herramientas ineficientes, los flujos de trabajo complejos y la falta de automatización dificultan la productividad del equipo, lo que ralentiza la entrega de funciones y actualizaciones.
- **Aumento de la deuda técnica:** las herramientas mal integradas y los procesos ad hoc pueden generar una deuda técnica, lo que dificulta el mantenimiento y la ampliación de los sistemas de software a lo largo del tiempo.
- **Disminución de la innovación:** cuando se ve empantanado por tareas manuales y repetitivas, la capacidad de su equipo para explorar nuevas tecnologías e impulsar la innovación se ve limitada.
- **Calidad comprometida:** la fragmentación de los procesos de prueba e implementación aumenta el riesgo de defectos y vulnerabilidades del software. Esto puede afectar negativamente a la calidad general del software suministrado.

Al invertir en una experiencia de desarrollo de software bien diseñada, puede obtener beneficios importantes, como una comercialización más rápida, una mejor calidad del software, una mayor satisfacción del equipo de desarrollo de software y una mayor agilidad empresarial.

Potenciando la experiencia de desarrollo de software con IA generativa

La integración de la IA generativa en el ciclo de vida del desarrollo del software (SDLC) representa un cambio de paradigma en la forma en que equipos enteros de desarrollo de software conciben, diseñan, implementan y mantienen las soluciones de software. La IA generativa tiene el potencial de revolucionar todas las fases del SDLC, incluida la gestión de proyectos, la recopilación de requisitos, el diseño, la codificación, las pruebas, el despliegue y el mantenimiento.

En esencia, una experiencia de AI-powered desarrollo generativo actúa como un colaborador inteligente para todo el equipo de desarrollo de software, incluidos los directores de producto, los diseñadores, los arquitectos de soluciones, los desarrolladores, los evaluadores y el personal de operaciones. Proporciona asistencia basada en el contexto, genera artefactos (como historias de usuarios, maquetas de diseños, fragmentos de código y casos de prueba), ofrece sugerencias prácticamente en tiempo real e incluso predice posibles problemas antes de que surjan. Este AI-augmented enfoque reduce considerablemente la carga cognitiva de los miembros del equipo. Esto les permite centrarse en las decisiones estratégicas de alto nivel y en la resolución de problemas complejos, mientras que la IA generativa se encarga de las tareas más mundanas y repetitivas.

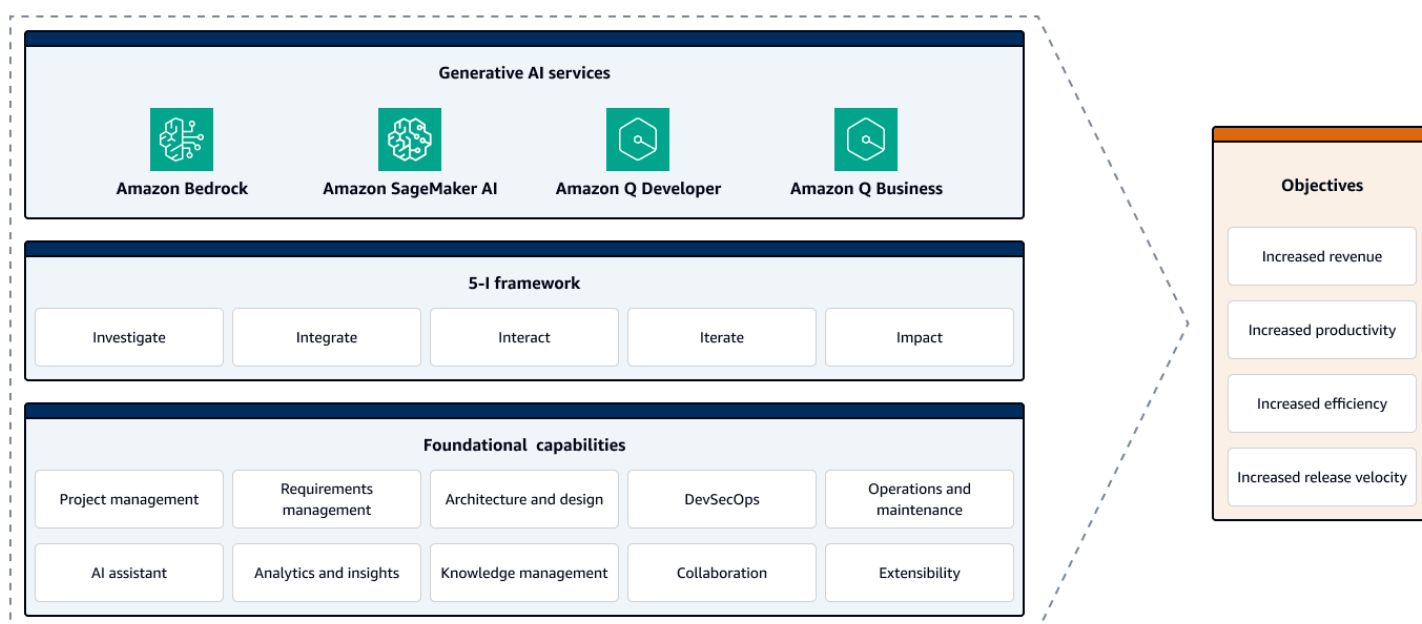
La IA generativa también sirve como amplificador del conocimiento. Ayuda a los miembros del equipo a acceder rápidamente a la información relevante, las mejores prácticas y los patrones procedentes de vastos repositorios de datos. Esto puede democratizar de manera efectiva la experiencia en toda la organización. Al integrar a la perfección las capacidades generativas de IA en toda la cadena de herramientas de desarrollo, puede crear un entorno más intuitivo, eficiente y productivo para todos sus equipos de desarrollo de software. Esta experiencia de desarrollo mejorada acelera el SDLC y mejora la calidad general. También reduce los errores y fomenta la innovación, ya que los miembros del equipo pueden explorar nuevas ideas y enfoques con mayor rapidez.

Para adoptar una experiencia de AI-powered desarrollo generativa en su organización, tenga en cuenta los siguientes elementos clave:

- [Marco 5-I](#)— El marco 5-I, que consta de cinco dimensiones, proporciona un enfoque integral para navegar por el proceso de desarrollo de software moderno. Ofrece una metodología estructurada que le ayuda a aplicar sistemáticamente la IA generativa en todas las etapas del SDLC.
- [Capacidades fundamentales](#)— Para aprovechar al máximo el poder de la IA generativa en todas las dimensiones del desarrollo de software moderno, es necesario establecer un conjunto sólido de

capacidades fundamentales. Estas capacidades forman la columna vertebral de una experiencia de AI-powered desarrollo. Estas capacidades le ayudan a integrar y utilizar la IA generativa en todo el SDLC.

En conjunto, el marco 5-I y las capacidades fundamentales forman una estrategia para reimaginar la experiencia de desarrollo de software. Las cinco dimensiones proporcionan un marco estratégico para aplicar la IA generativa, y las capacidades fundamentales preparan a su organización para respaldar este enfoque. AI-driven Servicios de AWS, como [Amazon Bedrock](#), [Amazon SageMaker AI](#), [Amazon Q Developer](#) y [Amazon Q Business](#), ofrecen funciones y funciones de IA generativa que puede integrar en su experiencia de desarrollo de software.



Marco 5-I para una AI-powered experiencia de desarrollo de software

El marco 5-I proporciona un enfoque estructurado para que los equipos de desarrollo de software integren eficazmente la IA generativa en sus prácticas de desarrollo. Le ayuda a establecer una base sólida para el uso de la IA generativa en todo el SDLC. También le ayuda a establecer las prácticas de desarrollo, los flujos de trabajo y la mentalidad adecuados para aprovechar al máximo el potencial de la IA generativa.

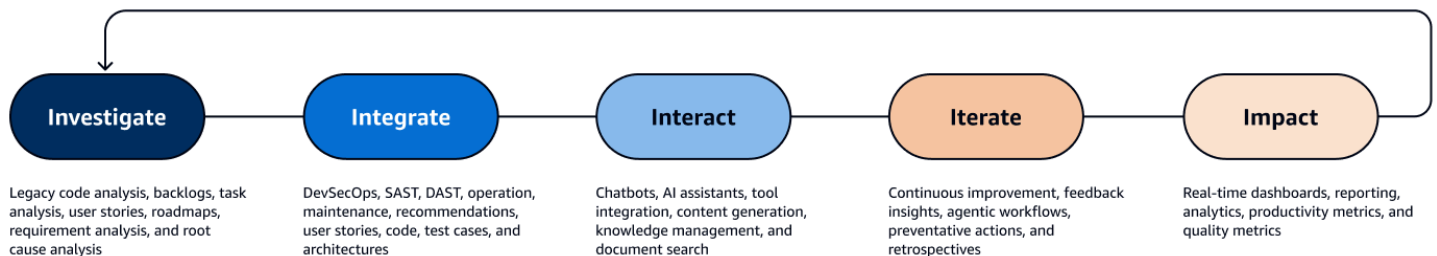
Esta sección contiene los siguientes temas:

- [Descripción general del marco](#)
- [Integración con el ciclo de vida del desarrollo de software](#)

Descripción general del marco

El marco 5-I se basa en cinco dimensiones clave: investigar, integrar, interactuar, iterar e impactar. Cada dimensión representa un área crítica en la que la IA generativa mejora significativamente el proceso de desarrollo de software. Al integrar estratégicamente la IA generativa en estas dimensiones, el marco aborda las necesidades cambiantes del desarrollo de software moderno. Puede reducir la carga cognitiva y amplificar el potencial creativo. Reconoce que la experiencia de desarrollo ideal no se basa solo en las herramientas, sino en crear un entorno en el que la IA mejore sin problemas las capacidades humanas en cada etapa.

El siguiente diagrama muestra las cinco dimensiones del desarrollo de AI-powered software. Para cada dimensión, muestra dónde se puede integrar la IA generativa para impulsar la eficiencia y la innovación.



Las cinco dimensiones del marco son las siguientes:

- **Investigue:** mejore todas las tareas analíticas de su proceso de desarrollo de software con la IA generativa. Utilice la IA generativa para comprender los requisitos, procesar grandes cantidades de datos, reconocer patrones y generar información que podría superar la capacidad humana o que tardaría mucho más en generarse. Esta información le ayuda a tomar decisiones más informadas, a identificar rápidamente las oportunidades de mejora y a ofrecer software de alta calidad de manera más eficiente. La IA generativa puede ser un socio inteligente para los procesos analíticos del SDLC. Al aprovechar la IA generativa, se realizan análisis exhaustivos en áreas críticas, como la recopilación de requisitos, el examen de la base de códigos heredada y la optimización de la cartera de productos. Por ejemplo, los propietarios de productos pueden utilizar la IA generativa para analizar las experiencias o los requisitos de los usuarios antes de crear historias de usuario. Los equipos de desarrollo pueden descubrir ineficiencias e identificar

oportunidades de optimización en las bases de código existentes. DevOps los ingenieros pueden aplicar el análisis de la causa raíz para diagnosticar rápidamente los problemas de rendimiento o las vulnerabilidades de seguridad, lo que puede mejorar la confiabilidad.

- **Integrar:** integre la IA generativa para automatizar una amplia gama de tareas y procesos en todo el SDLC. Esto incluye la generación automática de fragmentos de código, casos de prueba, diseños arquitectónicos, historias de usuarios y procesos de implementación. Al automatizar estas tareas, que suelen ser manuales, los equipos pueden centrarse en un trabajo más estratégico e innovador, lo que acelera el tiempo de comercialización y reduce la calidad de las aplicaciones. La dimensión integrada representa un cambio de paradigma en el desarrollo de software, en el que la IA se convierte en una parte integral del proceso de desarrollo. Trabaja junto con su equipo de desarrollo de software para mejorar la productividad, mejorar la calidad e impulsar la innovación. Esto se traduce en una comercialización más rápida. Reta a sus equipos de desarrollo de software a evaluar periódicamente sus procesos y flujos de trabajo preguntándose en cada paso: «¿Se puede automatizar esto?»
- **Interactúe:** utilice AI-powered asistentes generativos para proporcionar a su equipo asistencia instantánea y contextual en relación con una variedad de tareas y consultas. Estos asistentes inteligentes actúan como colaboradores expertos que se basan en un vasto repositorio de información. Pueden responder a las preguntas de codificación, ofrecer sugerencias de diseño, explicar los procedimientos operativos estándar y ayudar a solucionar problemas complejos. La integración de estos asistentes de IA en el flujo de trabajo de desarrollo aumenta la productividad y fomenta un entorno más colaborativo y de resolución de problemas.
- **Iterate:** utilice la IA generativa para realizar ajustes rápidos y basados en datos en todo el SDLC. Puede analizar continuamente los datos de fuentes como los comentarios de los clientes, los patrones de uso, las tendencias del mercado y las métricas de rendimiento del equipo para tomar decisiones informadas con rapidez. Esta adaptabilidad hace que el desarrollo de software pase de ser un proceso estático y predefinido a un enfoque fluido y receptivo. Se manifiesta de varias maneras, como la priorización dinámica de los atrasos, la asignación flexible de recursos, las estrategias de pruebas adaptables, la evolución de la documentación y los procesos de implementación adaptables. Por ejemplo, los gerentes de producto pueden utilizar la AI-generated información para reordenar sus pedidos pendientes e integrar los nuevos requisitos de los clientes y las tendencias del mercado prácticamente en tiempo real. DevOps los ingenieros pueden adaptar los planes de despliegue y las configuraciones de la infraestructura en función del análisis del rendimiento, lo que garantiza que las aplicaciones sigan siendo resilientes y estén optimizadas. Los equipos de desarrollo pueden traducir los comentarios de las retrospectivas de los sprints en

- mejoras prácticas para la siguiente iteración, lo que impulsa una cultura de mejora continua de los procesos.
- **Impacto:** aplique la IA generativa para evaluar la eficacia y el rendimiento de su proceso de desarrollo de software. Al utilizar AI-powered análisis y métricas, obtiene información más profunda sobre la eficiencia del desarrollo, la calidad del código, la participación de los usuarios y el rendimiento general de las aplicaciones. Este enfoque basado en datos le ayuda a tomar decisiones informadas, optimizar sus flujos de trabajo de desarrollo y mejorar continuamente la calidad y la experiencia de usuario de sus aplicaciones. Al evaluar la productividad del equipo de software, la IA generativa analiza varios puntos de datos, como la frecuencia de confirmación del código, los tiempos de resolución de los problemas, la velocidad de lanzamiento, las tasas de entrega de funciones y más. También puede evaluar la calidad de las revisiones del código, la eficacia de las herramientas de colaboración y el impacto de las diferentes prácticas de desarrollo en los resultados generales del equipo. Al correlacionar estas métricas con los resultados del proyecto, la IA identifica patrones y tendencias que los analistas humanos podrían pasar por alto, y puede proporcionar información útil que aumenta la productividad del equipo. Además, la IA generativa puede ayudarte a comparar el rendimiento del equipo con los estándares del sector o los datos históricos, y ofrece recomendaciones personalizadas para mejorar. También puede predecir posibles obstáculos o riesgos en el proceso de desarrollo para que puedas tomar medidas proactivas.

Integración con el ciclo de vida del desarrollo de software

El SDLC consta de múltiples fases, que pueden variar de una organización a otra. Por lo general, estas fases incluyen las siguientes: requisitos y planificación, diseño y arquitectura, implementación, pruebas, despliegue y operación y mantenimiento.

La siguiente tabla asigna las dimensiones del marco 5-I a las fases del SDLC y proporciona el nivel de integración de cada dimensión.

Dimensión del marco	Requisitos y planificación	Diseño y arquitectura	Implementación	Testeo	Implementación	Operación y mantenimiento
Investiga	Alto	Bajo	Bajo	Bajo	Bajo	Medio

Dimensión del marco	Requisitos y planificación	Diseño y arquitectura	Implementación	Testeo	Implementación	Operación y mantenimiento
Integrar	Medio	Medio	Alto	Medio	Alto	Alto
Interactúa	Alto	Alto	Alto	Medio	Medio	Alto
Iterar	Medio	Bajo	Bajo	Bajo	Bajo	Medio
Impact	Alto	Medio	Alto	Bajo	Alto	Alto

Los niveles de integración varían de altos a bajos. El mapeo revela las áreas de enfoque clave para cada dimensión. Por ejemplo, Investigate muestra una alta intensidad en la fase de requisitos y planificación. Integrate demuestra una alta intensidad en las fases de implementación, despliegue y operación y mantenimiento.

Al usar este mapeo, puede priorizar sus esfuerzos de manera efectiva. Le recomendamos que se centre en el nivel alto, luego en el medio y, por último, en el bajo. Asegúrese de adoptar un enfoque equilibrado e impactante que mejore la experiencia de desarrollo de software con la IA generativa.

Capacidades fundamentales para una AI-powered experiencia de desarrollo de software

Para implementar con éxito una experiencia de desarrollo de AI-powered software generativa, debe establecer un conjunto de capacidades fundamentales que abarquen a varias personas de su organización. Estas capacidades representan su capacidad para desplegar recursos y procesos de forma eficaz y lograr los resultados deseados en el contexto del desarrollo de AI-powered software. Al cultivar estas capacidades, usted crea una base sólida que le ayuda a integrar sin problemas la IA generativa en todas las etapas del SDLC.

AWS proporciona servicios clave para ayudarlo a implementar estas capacidades. Por ejemplo, [Amazon Q Developer](#) ayuda a acelerar el desarrollo de software al actuar como AI-powered asistente. [Amazon Q Business](#) te ayuda a obtener respuestas rápidas y relevantes a preguntas urgentes, resolver problemas y generar contenido. También puede actuar en su nombre integrando herramientas relacionadas con el desarrollo de software. [Amazon Bedrock](#) proporciona acceso a

modelos básicos y a un amplio conjunto de capacidades para personalizar requisitos y flujos de trabajo de desarrollo específicos.

Al desarrollar estas capacidades Servicios de AWS, usted crea una base sólida que le ayuda a integrar sin problemas la IA generativa en todas las etapas del SDLC.

Las siguientes son las capacidades fundamentales en las que debes centrarte:

- [Administración de proyectos](#)
- [Gestión de requisitos](#)
- [Arquitectura y diseño](#)
- [Colaboración](#)
- [DevSecOps](#)
- [Operación y mantenimiento](#)
- [Asistentes de IA](#)
- [Uso de análisis e información](#)
- [Administración de conocimientos](#)
- [Extensibilidad](#)

Cada capacidad fundamental se integra con las dimensiones del marco y las diferentes etapas del SDLC. Esta integración le ayuda a utilizar las capacidades de IA de forma eficaz en todo el proceso de desarrollo de software. Mejora la eficiencia, la calidad y la innovación en cada paso. La sinergia entre estas capacidades fundamentales, el marco y las etapas del SDLC crea un ecosistema integral para AI-powered el desarrollo de software. Esto le ayuda a aprovechar todo el potencial de la IA generativa, impulsar la mejora continua, acelerar los ciclos de desarrollo y ofrecer productos de software de calidad.

La siguiente tabla muestra cómo las capacidades y subcapacidades fundamentales se corresponden con las dimensiones del marco y las fases del SDLC.

Capacidad: subcapacidad	Investiga	Integrar	Interactuar	Itera	Impact
Gestión de proyectos:	Requisitos y planificación	Ninguno	Ninguna	Ninguna	Ninguno

Capacidad: subcapacidad	Investiga	Integrar	Interactuar	Itera	Impact
gestión de problemas					
Gestión de proyectos : gestión de sprints y tareas	Requisitos y planificación	Requisitos y planificación	Ninguno	Ninguna	Ninguno
Gestión de proyectos: gestión de la cartera de productos	Requisitos y planificación	Ninguno	Ninguno	Requisitos y planificación	Ninguno
Gestión de proyectos : mapeo de historias de usuarios	Requisitos y planificación	Ninguno	Ninguna	Ninguna	Ninguno
Gestión de proyectos: informes y análisis	Requisitos y planificación	Ninguno	Ninguna	Ninguno	Requisitos y planificación
Gestión de proyectos: gestión de la hoja de ruta del producto	Requisitos y planificación	Ninguno	Requisitos y planificación	Ninguno	Ninguno

Capacidad: subcapacidad	Investiga	Integrar	Interactuar	Itera	Impact
Gestión de proyectos : circuitos de retroalimentación	Ninguno	Ninguna	Ninguno	Requisitos y planificación	Ninguno
Gestión de proyectos : retrospectivas	Ninguno	Ninguna	Ninguno	Requisitos y planificación	Ninguno
Gestión de requisitos	Requisitos y planificación	Requisitos y planificación	Ninguno	Ninguna	Ninguno
Arquitectura y diseño: diseño de soluciones	Diseño y arquitectura	Diseño y arquitectura	Ninguno	Ninguna	Ninguno
Colaboración: gestión de la documentación	Todas las fases del SDLC	Ninguno	Todas las fases del SDLC	Ninguno	Ninguno
Colaboración: intercambio de conocimientos	Todas las fases del SDLC	Ninguno	Todas las fases del SDLC	Ninguno	Ninguno
Colaboración: gestión de activos del proyecto	Ninguno	Todas las fases del SDLC	Todas las fases del SDLC	Ninguno	Ninguno

Capacidad: subcapacidad	Investiga	Integrar	Interactuar	Itera	Impact
DevSecOps: CI/CD	Pruebas, despliegue	Implementación, pruebas, despliegue	Implementación	Ninguno	Ninguno
DevSecOps : DevOps seguridad	Implementación	Implementación, pruebas, operación y mantenimiento	Ninguno	Implementación, pruebas, operación y mantenimiento	Ninguno
DevSecOps : Supervisión del rendimiento de las aplicaciones	Ninguno	Operación y mantenimiento	Ninguno	Ninguna	Ninguno
DevSecOps : Agregación y análisis de registros	Operación y mantenimiento	Operación y mantenimiento	Ninguno	Ninguna	Ninguno
DevSecOps: AIOps	Operación y mantenimiento	Ninguno	Ninguno	Operación y mantenimiento	Ninguno
DevSecOps : Mejora continua	Ninguno	Ninguna	Ninguno	Operación y mantenimiento	Ninguno
DevSecOps : Supervisión del panel de control	Ninguno	Operación y mantenimiento	Ninguno	Ninguna	Ninguno

Capacidad: subcapacidad	Investiga	Integrar	Interactuar	Itera	Impact
DevSecOps : Información sobre el rendimiento	Operación y mantenimiento	Ninguno	Ninguno	Operación y mantenimiento	Ninguno
Operación y mantenimiento: gestión de incidentes	Ninguno	Ninguna	Ninguno	Operación y mantenimiento	Ninguno
Operación y mantenimiento: actualizaciones de código	Ninguno	Operación y mantenimiento	Ninguno	Ninguna	Ninguno
Operación y mantenimiento: optimización del código	Operación y mantenimiento	Operación y mantenimiento	Ninguno	Ninguna	Ninguno
Operación y mantenimiento: gestión técnica de la deuda	Ninguno	Operación y mantenimiento	Operación y mantenimiento	Ninguno	Ninguno
Operación y mantenimiento: gestión de cambios	Ninguno	Implementación, despliegue	Ninguno	Ninguna	Ninguno

Capacidad: subcapacidad	Investiga	Integrar	Interactuar	Itera	Impact
Operación y mantenimiento: ingeniería inversa	Operación y mantenimiento	Ninguno	Ninguna	Ninguna	Ninguno
Operación y mantenimiento: modernización del código	Ninguno	Implementación	Ninguno	Ninguna	Ninguno
Operación y mantenimiento: optimización del rendimiento	Ninguno	Operación y mantenimiento	Ninguno	Operación y mantenimiento	Ninguno
Análisis e información	Ninguno	Requisitos y planificación	Ninguno	Ninguno	Todas las fases del SDLC
Asistente de IA	Ninguno	Ninguno	Todas las fases del SDLC	Ninguno	Ninguno
Gestión del conocimiento	Ninguno	Ninguno	Todas las fases del SDLC	Ninguno	Ninguno
Extensibilidad	Ninguno	Implementación	Ninguno	Ninguna	Ninguno

Casos de uso de IA generativa para la gestión de proyectos

La gestión eficaz de los proyectos es la base del éxito del desarrollo de software. En el contexto de la IA generativa, la gestión de proyectos adquiere nuevas dimensiones. Puede ser más predictiva, adaptativa y basada en datos. AI-powered Las herramientas de gestión de proyectos analizan los datos históricos del proyecto para generar estimaciones de tiempo y recursos más precisas. Pueden priorizar automáticamente las tareas en función de los objetivos empresariales y la capacidad del equipo, e incluso pueden predecir posibles obstáculos antes de que se produzcan. Por ejemplo, un director de proyecto podría utilizar la IA generativa para crear un plan de proyecto preliminar basado en los requisitos del proyecto y en los datos históricos de proyectos similares. La IA podría entonces sugerir composiciones de equipo óptimas que tengan en cuenta las habilidades, las cargas de trabajo y las necesidades del proyecto. A lo largo del proyecto, los AI-driven paneles proporcionan información casi en tiempo real sobre el estado del proyecto al generar informes automáticamente y resaltar las áreas que requieren atención.

Este AI-augmented enfoque de la gestión de proyectos puede mejorar la eficiencia. Ayuda a los directores de proyectos a centrarse en la toma de decisiones estratégicas y en el liderazgo del equipo, en lugar de quedarse atascados en tareas administrativas rutinarias.

La siguiente tabla muestra los casos de uso de la gestión de proyectos que se pueden mejorar con la IA generativa y la persona responsable de esos casos de uso.

Subcapacidad: caso de uso	Persona
Gestión de problemas: cree y asigne problemas	Administrador de proyectos
Gestión de problemas: detecte problemas durante las pruebas y regístrelos	Ingeniero de pruebas
Gestión de problemas: priorice los problemas en función de su gravedad y asígnelos a los desarrolladores	Administrador de proyectos
Gestión de problemas: identifique y combine los problemas duplicados	Administrador de proyectos

Subcapacidad: caso de uso	Persona
Gestión de problemas: realice un seguimiento y genere informes sobre los problemas clave, las métricas y el estado general del proyecto	Administrador de proyectos
Gestión de sprints y tareas: Calcule el esfuerzo realizado para realizar las tareas y asigne argumentos en función de la capacidad del equipo	Scrum Master
Gestión del sprint y de las tareas: distribuya las tareas entre los miembros del equipo para lograr una carga de trabajo uniforme a lo largo del sprint	Scrum Master
Gestión de sprints y tareas: facilita las sesiones de planificación de los sprints para alinear los esfuerzos del equipo con los objetivos del sprint	Scrum Master
Gestión de la cartera de productos: reordene los elementos de la cartera de pedidos en función del valor empresarial, la urgencia y los comentarios de los usuarios	Propietario del producto
Gestión de la cartera de productos: integre los comentarios de los nuevos clientes y la información sobre el mercado en la cartera de productos para priorizarlos casi en tiempo real	Propietario del producto
Gestión de la cartera de productos: identifique y gestione las dependencias entre los elementos de la cartera de productos para agilizar el desarrollo	Gerente de producto

Subcapacidad: caso de uso	Persona
Mapeo de historias de usuarios: cree mapas de los viajes de los usuarios para identificar todas las funciones necesarias y sus historias de usuario correspondientes	Propietario del producto
Mapeo de historias de usuario: identifique las brechas o los pasos que faltan en el flujo de usuarios	Analista de negocios
Mapeo de historias de usuario: priorice las historias de usuario en función de su impacto en el valor empresarial	Gerente de producto
Informes y análisis: genere paneles de control prácticamente en tiempo real que visualicen las métricas clave del proyecto, como la velocidad de los sprints y las tasas de resolución de problemas	Administrador de proyectos
Informes y análisis: analice los datos históricos y prediga los resultados futuros del proyecto, como posibles retrasos o cuellos de botella	Administrador de proyectos
Informes y análisis: cree informes personalizados, como informes sobre el rendimiento del equipo o el estado del proyecto, que se adapten a las diferentes partes interesadas	Administrador de proyectos
Gestión de la hoja de ruta del producto: cree y mantenga una hoja de ruta del producto que describa los hitos principales y las fechas de lanzamiento	Administrador de proyectos

Subcapacidad: caso de uso	Persona
Gestión de la hoja de ruta del producto: actualice la hoja de ruta en función de los cambios en las prioridades o los plazos del proyecto	Gerente de producto
Gestión de la hoja de ruta del producto: comparta la hoja de ruta con las partes interesadas para proporcionar visibilidad sobre la dirección del producto	Gerente de producto
Circuitos de retroalimentación: recopile los comentarios del equipo después de cada sprint e identifique las áreas de mejora	¿Maestro de Scrum
Retrospectivas: Traduce los comentarios en elementos procesables para el próximo sprint, impulsando la mejora continua	¿Maestro de Scrum
Retrospectivas: haga un seguimiento del impacto de los cambios implementados a partir de retrospectivas anteriores para medir su eficacia	Maestro de Scrum

Casos de uso de IA generativa para la gestión de requisitos

La gestión de requisitos es un proceso fundamental que está estrechamente vinculado a la gestión de proyectos. Imagine que el propietario de un producto utiliza una herramienta de inteligencia artificial para analizar los comentarios de los clientes, las tendencias del mercado y las opiniones de las partes interesadas. La herramienta de IA podría generar un conjunto completo de historias de usuario y requisitos, clasificarlos automáticamente, detectar posibles conflictos o brechas e incluso sugerir la priorización en función del valor empresarial y la complejidad de la implementación. A medida que el proyecto avanza y los requisitos evolucionan, la IA puede actualizar y refinar continuamente los requisitos para asegurarse de que estén alineados con las cambiantes necesidades empresariales y las limitaciones técnicas. Este AI-driven enfoque dinámico de la gestión de requisitos ayuda a garantizar que los esfuerzos de desarrollo se mantengan estrechamente

alineados con las necesidades de los usuarios y los objetivos empresariales durante todo el ciclo de vida del proyecto.

En la siguiente tabla se muestran los casos de uso de la gestión de requisitos que se pueden mejorar con la IA generativa y la persona responsable de esos casos de uso.

Caso de uso	Persona
Cree requisitos empresariales	Analista de negocios
Cree epopeyas a partir de funciones	Propietario del producto
Sigue el progreso de una epopeya supervisando la finalización de las historias de usuario asociadas	Gerente de producto
Crea historias de usuario	Propietario del producto
Calcule el esfuerzo necesario para cada historia de uso y asigne puntos a la historia	Maestro de Scrum
Defina los criterios de aceptación para cada historia de usuario	Propietario del producto

Casos de uso de IA generativa para arquitectura y diseño

Con una base sólida de gestión de proyectos y requisitos bien definidos, la siguiente capacidad crítica es la arquitectura y el diseño. En este sentido, la IA generativa abre nuevas posibilidades para crear arquitecturas de software sólidas, escalables y eficientes. AI-powered las herramientas de diseño pueden analizar los requisitos y las limitaciones para sugerir patrones arquitectónicos y enfoques de diseño óptimos. Generan múltiples alternativas de diseño, y cada una está optimizada para diferentes prioridades, como el rendimiento, la escalabilidad o la facilidad de mantenimiento. Por ejemplo, un arquitecto de soluciones podría utilizar un asistente de IA para generar rápidamente varios diseños arquitectónicos de alto nivel en función de los requisitos del proyecto. Este AI-augmented enfoque acelera el proceso de diseño y ayuda a los arquitectos a tomar decisiones más informadas. Esto conduce a diseños de software más robustos y preparados para el futuro.

La siguiente tabla muestra los casos de uso de arquitectura y diseño que se pueden mejorar con la IA generativa y la persona responsable de esos casos de uso.

Caso de uso	Persona
Crea un documento de arquitectura	Arquitectura de soluciones
Cree un documento de diseño detallado	Líder técnico
Comprenda los estándares de arquitectura y diseño existentes	Arquitectura de soluciones
Desarrolle maquetas y prototipos detallados de una interfaz de usuario	UX/UI diseñador

Casos de uso de IA generativa para la colaboración

El desarrollo de software es intrínsecamente un esfuerzo colaborativo. Puede utilizar la IA generativa para mejorar la colaboración en su equipo de desarrollo de software. AI-powered las herramientas de colaboración van más allá de la simple mensajería y el intercambio de archivos. Facilitan una comunicación más eficaz al resumir largos hilos de debate, destacar las decisiones clave e incluso sugerir horarios óptimos para las reuniones en función de los horarios y los patrones de productividad de los miembros del equipo. La IA puede ayudar a revisar el código al identificar automáticamente los posibles problemas, sugerir mejoras e incluso explicar los cambios complejos a los revisores. Durante las sesiones de intercambio de ideas, la IA puede actuar como facilitadora, generar ideas, ayudar a organizar las reflexiones e incluso mediar en los debates para garantizar que se escuchen todas las voces. En el caso de los equipos distribuidos, la IA puede ayudar a superar las barreras culturales y lingüísticas. Puede ofrecer traducciones de idiomas prácticamente en tiempo real en los chats y las videollamadas, y ofrecer un contexto cultural que ayude a evitar malentendidos. Al aumentar la colaboración humana con la IA, esta capacidad ayuda a los equipos a trabajar de manera más eficiente y eficaz, lo que fomenta la innovación y mejora los resultados generales del proyecto.

En la siguiente tabla se muestra cómo se puede utilizar la IA generativa para mejorar los casos de uso de la colaboración.

Subcapacidad: caso de uso	Persona
Administración de documentos: cree y mantenga un repositorio de documentación centralizado	Redactor técnico
Gestión de documentos: permite que varios miembros del equipo colaboren en la documentación en tiempo real	Desarrollo en equipo
Intercambio de conocimientos: utilice los foros de debate como plataforma para que los desarrolladores formulen preguntas, compartan conocimientos y solucionen problemas de forma colaborativa	Desarrollo en equipo
Intercambio de conocimientos: utilice los foros de debate para documentar y realizar un seguimiento de las decisiones tomadas durante las discusiones del proyecto, asegurándose de que se capte la razón de ser de las decisiones clave y que esté accesible para consultarlas en el futuro	Director de producto
Gestión de activos del proyecto: facilite el intercambio fácil de los recursos relacionados con el proyecto	Desarrollo en equipo
Gestión de activos del proyecto: Implemente el control de versiones del contenido compartido para que los miembros del equipo puedan realizar un seguimiento de los cambios, volver a versiones anteriores y colaborar en las actualizaciones del contenido	Desarrollo en equipo

Casos de uso de IA generativa para DevSecOps

AI-powered DevSecOps las herramientas automatizan muchos aspectos del proceso de entrega de software. Por ejemplo, pueden realizar revisiones inteligentes del código, detectar posibles errores, detectar vulnerabilidades de seguridad e identificar problemas de rendimiento prácticamente en tiempo real a medida que los desarrolladores escriben el código. La IA genera y ejecuta conjuntos de pruebas integrales, y los actualiza automáticamente a medida que evoluciona la base de código. Este AI-augmented enfoque DevSecOps acelera el proceso de entrega y mejora significativamente la seguridad y la confiabilidad del software que se entrega.

En la siguiente tabla se muestran los casos de DevSecOps uso que se pueden mejorar con la IA generativa y la persona responsable de esos casos de uso.

Subcapacidad: caso de uso	Persona
DevOps y entrega continua: procesos de despliegue completos automatizados	DevOps ingeniero
DevOps y entrega continua: reciba comentarios casi en tiempo real sobre la calidad del código y los posibles problemas	Desarrollador de software
DevOps y entrega continua: reciba recomendaciones sobre problemas de seguridad y soluciones prácticamente en tiempo real	Desarrollador de software
DevOps y entrega continua: reciba sugerencias sobre códigos y mejores prácticas prácticamente en tiempo real	Desarrollador de software
DevOps y entrega continua: automatice las tareas repetitivas e integre los comandos en los scripts	DevOps ingeniero
DevOps y entrega continua: cree código y genere artefactos automáticamente después de cada confirmación de código	Desarrollador de software

Subcapacidad: caso de uso	Persona
DevOps y entrega continua: cree código de acuerdo con los estándares y el marco de la organización	Desarrollador de software
DevOps y entrega continua: ejecute automáticamente pruebas unitarias en cada commit to catch errores al principio del proceso de desarrollo	Desarrollador de software
DevOps y entrega continua: analice la cobertura de las pruebas unitarias para asegurarse de que se prueban todas las rutas de código críticas	Desarrollador de software
DevOps y entrega continua: administre las sucursales y fusione los cambios	Desarrollador de software
DevOps y entrega continua: administre el control de versiones de códigos y artefactos	Desarrollador de software
DevOps y entrega continua: almacene y gestione los artefactos de construcción y las dependencias	DevOps ingeniero
DevOps y entrega continua: resuelva y recupere las dependencias durante el proceso de creación	Desarrollador de software
DevOps y entrega continua: genere y ejecute pruebas de integración para asegurarse de que los componentes funcionan juntos según lo esperado	Ingeniero de pruebas

Subcapacidad: caso de uso	Persona
DevOps y entrega continua: utilice servicios simulados durante las pruebas de integración para simular interacciones con sistemas externos	Ingeniero de pruebas
DevOps y entrega continua: compare el rendimiento de las aplicaciones bajo diferentes cargas	Ingeniero de rendimiento
DevOps y entrega continua: simule escenarios de alto tráfico para probar la escalabilidad y los tiempos de respuesta de la aplicación	Ingeniero de rendimiento
DevOps y entrega continua: pruebe la capacidad del sistema para recuperarse de fallas, como caídas del servidor o interrupciones de la red	Ingeniero de confiabilidad del sitio
DevOps y entrega continua: realice una ingeniería de caos	Ingeniero de confiabilidad del sitio
DevOps y entrega continua: ejecute pruebas para verificar que la aplicación cumpla con los requisitos empresariales	Un ingeniero
DevOps y entrega continua: realice pruebas de aceptación por parte de los usuarios	Propietario del producto
DevOps y entrega continua: escanee las dependencias en busca de vulnerabilidades y problemas de conformidad con las licencias	Ingeniero de seguridad
DevOps y entrega continua: supervise y administre las dependencias del código abierto para asegurarse de que estén actualizadas y seguras	Ingeniero de seguridad

Subcapacidad: caso de uso	Persona
DevOps y entrega continua: genere y mantenga una lista de materiales de software (SBOM) para realizar un seguimiento de todos los componentes y dependencias	Ingeniero de seguridad
DevOps y entrega continua: utilice la SBOM para realizar auditorías a fin de garantizar el cumplimiento de las normas	Oficial de cumplimiento
DevOps y entrega continua: cree notas de lanzamiento	Gestor de versiones
DevOps y entrega continua: planifique y coordine los lanzamientos	Gestor de lanzamientos
DevOps y entrega continua: Implemente procedimientos operativos estándar para la gestión de la reversión y las versiones	Gestor de versiones
DevOps y entrega continua: utilice indicadores de funciones para habilitar o deshabilitar las funciones en producción sin implementar código nuevo	Gerente de producto
DevOps y entrega continua: ejecute A/B pruebas utilizando indicadores de características para medir el impacto de las diferentes funciones en el comportamiento de los usuarios	Gerente de producto
DevOps y entrega continua: analice y supervise las fallas en las tuberías	DevOps ingeniero
DevOps y entrega continua: cree y gestione los recursos de infraestructura	DevOps ingeniero

Subcapacidad: caso de uso	Persona
DevOps y seguridad: escanee los repositorios de códigos en busca de secretos codificados	DevOps ingeniero
DevOps y seguridad: Implemente una detección casi en tiempo real para alertar inmediatamente a los desarrolladores si hay secretos guardados en el repositorio	DevOps ingeniero
DevOps y seguridad: aplique un monitoreo continuo de la calidad del código	Desarrollador de software
DevOps y seguridad: detecte y señale los indicadores de posibles vulnerabilidades de seguridad en el código	Desarrollador de software
DevOps y seguridad: Implemente pruebas automatizadas para detectar los 10 principales riesgos de seguridad del Open Worldwide Application Security Project (OWASP) para asegurarse de que la aplicación cumple con las prácticas de seguridad estándar del sector	Ingeniero de seguridad
DevOps y seguridad: actualice y eduque periódicamente a los desarrolladores sobre los riesgos del OWASP integrando las comprobaciones en el proceso de desarrollo	Ingeniero de seguridad
DevOps y seguridad: escanee bibliotecas y dependencias de terceros para detectar vulnerabilidades de seguridad conocidas	DevOps ingeniero
DevOps y seguridad: escanee el código y la infraestructura de la aplicación para detectar vulnerabilidades	DevOps ingeniero

Subcapacidad: caso de uso	Persona
DevOps y seguridad: analice el código en busca de vulnerabilidades antes del despliegue	Ingeniero de seguridad
DevOps y seguridad: aplique las políticas de seguridad evitando que se fusione el código con vulnerabilidades críticas	Ingeniero de seguridad
DevOps y seguridad: Implemente un control de acceso basado en roles (RBAC) para restringir el acceso a los sistemas y datos confidenciales y garantizar que solo el personal autorizado pueda acceder a los recursos críticos	Ingeniero de seguridad
DevOps y seguridad: ajuste los controles de acceso en función de las funciones y responsabilidades adaptándolos a los cambios en la estructura del equipo	DevOps ingeniero
DevOps y seguridad: pruebe las aplicaciones en ejecución para detectar vulnerabilidades de seguridad casi en tiempo real simulando ataques al entorno de producción	Ingeniero de seguridad
DevOps y seguridad: supervise continuamente las aplicaciones implementadas para detectar vulnerabilidades de seguridad	DevOps ingeniero
DevOps y seguridad: programe escaneos de vulnerabilidades periódicos en todos los entornos para identificar y abordar las debilidades de seguridad	Ingeniero de seguridad
DevOps y seguridad: aplique parches y actualizaciones en función de los resultados del análisis de vulnerabilidades para ayudar a mantener los sistemas seguros	DevOps ingeniero

Subcapacidad: caso de uso	Persona
Supervisión del rendimiento de las aplicaciones: supervise continuamente el rendimiento de las aplicaciones casi en tiempo real para detectar y diagnosticar los problemas de rendimiento antes de que afecten a los usuarios	Ingeniero de confiabilidad del sitio
Supervisión del rendimiento de las aplicaciones: detecte anomalías en el rendimiento, como picos repentinos en los tiempos de respuesta o un aumento de las tasas de error, e inicie alertas	DevOps ingeniero
Supervisión del rendimiento de las aplicaciones: rastrea las solicitudes a medida que se propagan por un sistema distribuido para identificar los cuellos de botella en el rendimiento y los problemas de latencia	DevOps ingeniero
Supervisión del rendimiento de las aplicaciones: utilice el rastreo distribuido para identificar el servicio o componente exacto responsable de los fallos o la degradación del rendimiento	DevOps ingeniero
Agregación y análisis de registros: agregue registros de múltiples fuentes en un sistema centralizado para facilitar la búsqueda y el análisis a fin de identificar tendencias y problemas	Ingeniero de confiabilidad del sitio
Análisis y agregación de registros: Implemente el análisis automático de registros para extraer información relevante y detectar patrones o anomalías que puedan indicar problemas	DevOps ingeniero

Subcapacidad: caso de uso	Persona
Agregación y análisis de registros: recopile y visualice las métricas clave de rendimiento	Ingeniero de confiabilidad del sitio
Análisis y agregación de registros: supervise las métricas comparándolas con los acuerdos de nivel de servicio (SLA) predefinidos	Gerente de producto
Operaciones de IA: detecte incidentes, analice las causas fundamentales e inicie acciones correctivas sin intervención humana	DevOps ingeniero
Operaciones de IA: prediga las demandas futuras de recursos y optimice la planificación de la capacidad para evitar interrupciones	Ingeniero de confiabilidad del sitio
Mejora continua: supervise las interacciones reales de los usuarios con la aplicación para recopilar información sobre el rendimiento e identificar las áreas de mejora	Diseñador de UX
Mejora continua: realice un seguimiento del rendimiento de las aplicaciones en diferentes regiones geográficas para garantizar una experiencia de usuario uniforme a nivel mundial	Gerente de producto
Supervisión del panel de control: cree paneles personalizables para visualizar las métricas, los registros y los rastreos críticos casi en tiempo real a fin de ofrecer una visión completa del estado del sistema	Ingeniero de confiabilidad del sitio
Supervisión de paneles: cree paneles para diferentes equipos (como los equipos de desarrollo, operaciones y productos) a fin de proporcionar información relevante en función de sus áreas de interés	DevOps ingeniero

Subcapacidad: caso de uso	Persona
Información sobre el rendimiento: realice un análisis detallado del rendimiento de las aplicaciones para identificar las ineficiencias y optimizar el código o la infraestructura	Desarrollador de software
Información sobre el rendimiento: utilice la información sobre el rendimiento para mejorar de forma iterativa el rendimiento de las aplicaciones y optimizar la experiencia del usuario a lo largo del tiempo	Gerente de producto

Casos de uso de IA generativa para operación y mantenimiento

Una vez implementado el software, la atención se centra en la operación y el mantenimiento. La IA generativa puede mejorar los enfoques tradicionales al proporcionar una gestión del sistema más proactiva y eficiente. Al-powered las herramientas de operaciones supervisan continuamente el rendimiento del sistema y predicen posibles problemas antes de que afecten a los usuarios. Realizan un análisis automatizado de la causa raíz cuando se producen problemas, lo que reduce significativamente el tiempo medio de resolución. La IA también optimiza el rendimiento del sistema prácticamente en tiempo real. Ajusta automáticamente las configuraciones en función de los cambios en los patrones de carga y el comportamiento de los usuarios. Por ejemplo, un equipo de operaciones podría usar un asistente de inteligencia artificial para generar programas de mantenimiento predictivos, identificar automáticamente los componentes que tienen probabilidades de fallar y sugerir acciones preventivas. La IA también podría ayudar a planificar la capacidad al analizar las tendencias de uso y predecir las necesidades futuras de recursos con gran precisión.

La siguiente tabla muestra los casos de uso de operación y mantenimiento que se pueden mejorar con la IA generativa y la persona responsable de esos casos de uso.

Subcapacidad: caso de uso	Persona
Gestión de incidentes: gestione los incidentes casi en tiempo real mediante la integración de las herramientas de supervisión con	Ingeniero de confiabilidad del sitio

Subcapacidad: caso de uso	Persona
las plataformas de chat para que los equipos puedan detectar, debatir y resolver los problemas directamente en el entorno de chat	
Gestión de incidentes: permita a los equipos iniciar las implementaciones, ejecutar scripts y ejecutar comandos directamente desde la interfaz de chat, lo que agiliza las operaciones	DevOps ingeniero
Actualizaciones de código: actualice las dependencias y bibliotecas del código para reducir el esfuerzo manual y asegúrese de que la base de código esté actualizada con las últimas versiones	Desarrollador de software
Optimización del código: revise el código para conocer las oportunidades de optimización	Desarrollador de software
Optimización del código: identifique los cuellos de botella en el código y refactorice u optimice el código para mejorar el rendimiento	Desarrollador de software
Gestión de la deuda técnica: registre la deuda técnica como parte del proceso de desarrollo	Gerente de producto
Gestión de la deuda técnica: priorice y aborde la deuda técnica en función del impacto, el riesgo y el costo, e intégreala en el proceso habitual de planificación de los sprints	Desarrollador de software
Gestión de la deuda técnica: reduzca la deuda técnica en el código de la aplicación existente	Desarrollador de software

Subcapacidad: caso de uso	Persona
Gestión de cambios: Implemente un proceso de aprobación de cambios que garantice que todas las partes interesadas pertinentes revisen, prueben y aprueben todos los cambios en el código antes de su implementación	Gestor de cambios
Gestión de cambios: realice un análisis de impacto de los cambios propuestos	DevOps ingeniero
Ingeniería inversa: analice y comprenda la estructura y el comportamiento del código heredado	Arquitectura de soluciones
Ingeniería inversa: explique el código existente y genere documentación	Desarrollador de software
Modernización del código: Translate código de un lenguaje de programación a otro	Desarrollador de software
Modernización del código: moderniza el código heredado al lenguaje de programación más reciente	Desarrollador de software
Optimización del rendimiento: supervise y ajuste continuamente el rendimiento del sistema optimizando la asignación de recursos, el equilibrio de carga y la reconfiguración de la aplicación	Ingeniero de confiabilidad del sitio
Optimización del rendimiento: identifique y refactorice el código que está causando la degradación del rendimiento para mejorar la velocidad y la capacidad de respuesta del sistema	Desarrollador de software

Casos de uso de asistentes de IA generativa en el desarrollo de software

La capacidad de asistente de IA es la base de la experiencia de AI-powered desarrollo generativo. Este sistema inteligente y sensible al contexto sirve de colaborador virtual para todos los miembros del equipo en todo el SDLC. Imagine a un desarrollador trabajando en una pieza de código compleja. Solo tienen que pedir ayuda al asistente de IA, que puede proporcionarles fragmentos de código relevantes, explicar algoritmos complejos o incluso sugerir optimizaciones en función del contexto actual y de las mejores prácticas. El asistente de IA puede ayudar a un gerente de ITOps a comprender un procedimiento operativo estándar basado en documentos internos. Al proporcionar un apoyo contextual instantáneo, los asistentes de IA reducen significativamente la carga cognitiva de los miembros del equipo. Esto les ayuda a centrarse en tareas creativas y de resolución de problemas de alto nivel. Esta capacidad actúa como un multiplicador de fuerzas que mejora la productividad y la calidad en todas las etapas del desarrollo del software.

En la siguiente tabla se muestran los casos de uso que se pueden mejorar con los asistentes de IA y con la persona beneficiada.

Caso de uso	Persona
Brinde asistencia inmediata al equipo de desarrollo respondiendo preguntas, como las relacionadas con los requisitos, las arquitecturas y los procedimientos operativos estándar	Equipo de desarrollo de software
Busque o recupere extractos de una amplia documentación o genere resúmenes mediante consultas en lenguaje natural	Equipo de desarrollo de software
Resuma los documentos técnicos extensos, como los documentos de requisitos, la documentación de diseño de la arquitectura y los procesos internos	Equipo de desarrollo de software
Mantenga una biblioteca de indicaciones que el equipo pueda usar para tareas comunes	Equipo de desarrollo de software
Integre sin problemas la IA generativa en las herramientas y sistemas existentes	Equipo de desarrollo de software

Caso de uso	Persona
Automatice las tareas en diversas plataformas, herramientas y sistemas internos	Equipo de desarrollo de software
Cree un repositorio centralizado de conocimientos, que incluya las mejores prácticas, la información específica del proyecto y el conocimiento del equipo, al que puedan acceder todos los miembros del equipo	Equipo de desarrollo de software
Recupere el conocimiento relevante del repositorio en función del contexto de la tarea	Equipo de desarrollo de software
Realice revisiones automatizadas del código, analice la causa raíz, sugiera mejoras, detecte posibles errores y resuelva problemas	Desarrollador de software, DevOps ingeniero e ingeniero de confiabilidad del sitio
Analice los datos de rendimiento para identificar tendencias y patrones que puedan informar las decisiones sobre la optimización del rendimiento	Ingeniero de confiabilidad del sitio
Proporcione recomendaciones para mejorar la eficiencia, reducir la complejidad y mejorar la seguridad	Desarrollador de software
Sugiera optimizaciones para el uso de los recursos de la nube, como recomendaciones de escalado o estrategias de ahorro de costos	Desarrollador de software, DevOps ingeniero, ingeniero de confiabilidad de sitios y arquitecto de soluciones
Genere contenido nuevo, como documentación basada en código, guías de usuario o versiones de funciones del producto	Equipo de desarrollo de software

Casos de uso de IA generativa para análisis e información

La capacidad de análisis e información ayuda a convertir grandes cantidades de datos en información procesable que impulsa la toma de decisiones y la mejora continua. Al utilizar la IA generativa, esta capacidad procesa datos de diversas fuentes, incluidos los repositorios de código, las herramientas de gestión de proyectos y las plataformas de colaboración en equipo, para ofrecer una visión holística del proceso de desarrollo y la productividad del equipo. La IA generativa va más allá de las métricas tradicionales para ofrecer análisis predictivos y prescriptivos. Puede pronosticar posibles problemas y sugerir mejoras específicas. Por ejemplo, puede analizar los patrones en las confirmaciones de código, las tasas de resolución de errores y la velocidad de entrega de las funciones para identificar los equipos con alto rendimiento, identificar los cuellos de botella y sugerir optimizaciones de procesos. Además, puede proporcionar información sobre la dinámica del equipo y el rendimiento individual. Estos conocimientos ayudan a los líderes a tomar decisiones basadas en datos sobre la distribución de la carga de trabajo, las necesidades de formación y la composición del equipo. Al presentar estos conocimientos a través de paneles interactivos, la función permite a las partes interesadas de todos los niveles tomar decisiones informadas, optimizar los procesos y mejorar continuamente la productividad del equipo, lo que se traduce en una entrega más rápida de software de alta calidad.

En la siguiente tabla se muestran los casos de uso de la analítica que se pueden mejorar con la IA generativa y la persona responsable de esos casos de uso.

Caso de uso	Persona
Supervise la productividad individual y del equipo	Gerente de desarrollo
Analiza las tendencias de productividad para detectar posibles casos de agotamiento y tomar medidas proactivas para mantener el bienestar y la productividad del equipo	Director de desarrollo
Realice un seguimiento de la frecuencia con la que se implementan los cambios de código en la producción para medir la velocidad y la agilidad del proceso de desarrollo	Gerente de producto

Caso de uso	Persona
Analice los datos de frecuencia de implementación para identificar los períodos de baja actividad de implementación que puedan indicar ineficiencias del proceso o limitaciones de recursos	Gerente de producto
Mida el tiempo transcurrido entre la confirmación del código y la implementación para identificar oportunidades de agilizar los procesos de desarrollo e implementación	Director de desarrollo
Realice un seguimiento del porcentaje de implementaciones que provocan fallas que requieren una reparación inmediata para evaluar la confiabilidad del proceso de lanzamiento	Ingeniero de confiabilidad del sitio
Utilice las métricas de la tasa de errores de cambio para identificar las áreas del código que con frecuencia causan problemas a fin de guiar los esfuerzos de refactorización y prueba específicos	Desarrollador de software
Supervise el tiempo que se tarda en restablecer el servicio después de una interrupción o incidente para reducir el tiempo de inactividad y mejorar la resiliencia general del sistema	Ingeniero de confiabilidad del sitio
Analice las tendencias en los tiempos de restauración para mejorar los procesos de respuesta a incidentes e impulsar una recuperación más rápida en caso de averías del sistema	DevOps ingeniero

Caso de uso	Persona
Cree un panel de control personalizado que agrupe las métricas clave, como la frecuencia de implementación, el tiempo de entrega y la tasa de errores de cambio, a fin de ofrecer una visión integral del desarrollo y el estado operativo	Gerente de producto
Cree cuadros de mando que se adapten a las necesidades de los diferentes equipos a fin de proporcionar información específica sobre sus áreas de responsabilidad específicas, como el desarrollo, las operaciones o los negocios	Gerente de producto
Realice un seguimiento de los indicadores clave de rendimiento (KPI) de la empresa, como el impacto en los ingresos, la satisfacción de los clientes y la cuota de mercado, a fin de alinear los esfuerzos de desarrollo con los objetivos empresariales más amplios	Gerente de producto
Analice el impacto de las nuevas funciones en los KPI empresariales para evaluar su éxito y guiar el futuro desarrollo de productos	Analista de negocios
Supervise las métricas de calidad del código, como la complejidad del código, la cobertura de las pruebas y la densidad de errores, para asegurarse de que la base de código siga siendo mantenible y segura	Desarrollador de software
Identifique las áreas del código base que requieren refactorización para impulsar la sostenibilidad a largo plazo y reducir la deuda técnica	Arquitectura de soluciones

Casos de uso de la IA generativa para la gestión del conocimiento

En cualquier organización de desarrollo de software, el conocimiento es un activo fundamental. La capacidad de gestión del conocimiento, impulsada por la IA generativa, mejora la forma en que se captura, organiza y utiliza este activo. Los sistemas de gestión del conocimiento tradicionales suelen contener demasiada información, contenido desactualizado o son difíciles de buscar para encontrar rápidamente la información relevante.

La IA generativa aborda estos desafíos sin rodeos. Genera y actualiza automáticamente la documentación en función de los cambios en el código, las conversaciones y los artefactos del proyecto. Esto garantiza que las bases de conocimiento se mantengan actualizadas sin requerir un esfuerzo manual por parte de los miembros del equipo. Y lo que es más importante, la IA hace que este conocimiento sea accesible de forma intuitiva. Los miembros del equipo pueden hacer preguntas en un lenguaje natural y la IA puede proporcionar respuestas relevantes. La IA puede basarse en una variedad de fuentes, como documentación oficial, comentarios de código, hilos de discusión e incluso recursos externos. Por ejemplo, un nuevo miembro del equipo que intente entender un componente específico podría preguntar a la IA: «¿Cómo funciona el módulo de autenticación?» Luego, la IA proporcionaría una explicación concisa y enlaces a las secciones de código relevantes, los diagramas de arquitectura y los cambios recientes. Incluso podría adaptar esta información en función del rol y el nivel de experiencia del miembro del equipo.

Esta capacidad acelera la incorporación, reduce las preguntas repetitivas y promueve el intercambio de conocimientos en toda la organización. Ayuda a preservar el conocimiento institucional, lo que facilita a los equipos el mantenimiento y la evolución de sistemas complejos a lo largo del tiempo.

La siguiente tabla muestra los casos de uso de la gestión del conocimiento que se pueden mejorar con la IA generativa y la persona responsable de esos casos de uso.

Caso de uso	Persona
Cree una plataforma unificada que facilite el acceso a todos los conocimientos relacionados con el proyecto	Equipo de desarrollo de software
Capture el conocimiento de diversas actividades de desarrollo	Equipo de desarrollo de software

Caso de uso	Persona
Proporcione una funcionalidad de búsqueda avanzada para encontrar rápidamente el conocimiento relevante en un repositorio	Equipo de desarrollo de software
Personalice los módulos y las rutas de aprendizaje para el equipo	Equipo de desarrollo de software

Casos de uso de IA generativa para la extensibilidad

La extensibilidad permite una integración perfecta con las herramientas y los flujos de trabajo existentes, al tiempo que permite a las organizaciones adaptar el sistema de IA a sus necesidades específicas. Esta capacidad proporciona API, SDK e interfaces personalizables sólidas que facilitan la integración de las funcionalidades de la IA en las herramientas populares de desarrollo y gestión de proyectos. Por ejemplo, las organizaciones pueden mejorar Jira con AI-powered funciones para la priorización automática de los tickets, la estimación del esfuerzo y la planificación de los sprints. Puedes aumentar los procesos de trabajo de Jenkins con la IA para optimizar de forma inteligente las compilaciones y seleccionar pruebas predictivas.

Además, la extensibilidad permite una integración profunda con los entornos de desarrollo integrados (IDE), los sistemas de control de versiones y las plataformas de revisión de código. La IA puede ayudar a codificar, automatizar las revisiones de código y generar documentación contextual.

La capacidad también permite entrenar y ajustar los modelos de IA a partir de datos específicos de la organización. Esto ayuda a la IA a comprender los patrones de codificación específicos de la empresa, las preferencias arquitectónicas y el conocimiento del dominio. El resultado es una asistencia más relevante y adaptada al contexto en todas las herramientas integradas. Al proporcionar este nivel de flexibilidad e integración, la extensibilidad garantiza que la experiencia de AI-powered desarrollo evolucione con la organización. Puede adaptarse a las cambiantes tecnologías y necesidades empresariales y, al mismo tiempo, mejorar sin problemas las cadenas de herramientas y los flujos de trabajo existentes.

En la siguiente tabla se muestran los casos de uso de la extensibilidad que se pueden mejorar con la IA generativa y la persona responsable de esos casos de uso.

Caso de uso	Persona
Integre herramientas de terceros en el entorno de desarrollo	DevOps ingeniero
Cree flujos de trabajo de automatización personalizados que se adapten al proceso de desarrollo único del equipo	DevOps ingeniero
Conéctese a varias API y servicios	DevOps ingeniero
Cree conectores para herramientas multiplataforma	DevOps ingeniero

Mejores prácticas para el uso de la IA generativa en el desarrollo de software

En esta sección, se describen las mejores prácticas para integrar la IA generativa en el ciclo de vida del desarrollo de software (SDLC). Desde la implementación de cadenas de herramientas y DevSecOps procesos continuos hasta el fomento de la colaboración y la automatización de las tareas repetitivas, estas directrices le ayudan a aprovechar el potencial de la IA para mejorar sus procesos y experiencias de desarrollo. Al seguir estas mejores prácticas, los equipos de desarrollo de software pueden alcanzar nuevos niveles de eficiencia, innovación y calidad en su trabajo.

En esta sección se analizan las siguientes prácticas recomendadas:

- [Implementación de una cadena de herramientas end-to-end integrada y sin fisuras](#)
- [Implementar una end-to-end CI/CD canalización para DevSecOps](#)
- [Adoptar herramientas y prácticas colaborativas](#)
- [Automatizar las tareas repetitivas](#)
- [Revisar e iterar periódicamente la experiencia de desarrollo](#)
- [Adoptar prácticas de gestión de proyectos eficaces](#)
- [Implementar la gestión del conocimiento](#)
- [Proporciona extensibilidad y personalización](#)
- [Optimización para las operaciones](#)
- [Uso de información basada en datos](#)
- [Adoptar un enfoque basado en plataformas](#)

Implementación de una cadena de herramientas end-to-end integrada y sin fisuras

La implementación de una cadena de herramientas end-to-end integrada y sin fisuras es una de las mejores prácticas fundamentales para crear una experiencia de desarrollo generativa impulsada por la IA. La idea principal es establecer un ecosistema cohesivo de herramientas y plataformas que sus equipos de software puedan utilizar en todo el SDLC. El equipo puede utilizar la cadena de herramientas para planificar, idear, codificar, crear, probar, implementar y gestionar las operaciones

en curso. Al integrar las capacidades generativas de IA en esta cadena de herramientas, te aseguras de que la asistencia de la IA esté disponible en todas las etapas. Esta integración reduce o elimina las transferencias manuales, reduce el cambio de contexto y ayuda a que los datos y los artefactos fluyan sin problemas entre las diferentes fases de desarrollo. Por ejemplo, los fragmentos de código generados por la IA desde su entorno de desarrollo integrado (IDE) pueden fluir sin problemas a su sistema de control de versiones, y los análisis basados en la IA desde su plataforma de implementación pueden servir de base para sus herramientas de gestión de proyectos. Esto crea un circuito de retroalimentación continuo que mejora el proceso de desarrollo.

Implementar una end-to-end CI/CD canalización para DevSecOps

Para aprovechar esta cadena de herramientas integrada e implementar una integración end-to-end continua y un despliegue continuo (CI/CD) pipeline for DevSecOps. This AI-powered pipeline is a critical component that streamlines your software delivery processes. It helps you release new applications and updates more quickly and reliably. By embedding security practices throughout the entire SDLC, you can identify and address vulnerabilities much earlier, which reduces the overall cost and risk. The pipeline should incorporate AI at every stage, from continuous integration and testing to security checks and deployment. For instance, you can use AI to analyze code commits in near real time so that you can predict potential integration issues before they occur. In the CI/CD canalización), también puede utilizar la IA generativa para actualizar automáticamente las políticas de seguridad en función de la información sobre amenazas más reciente.

Adoptar herramientas y prácticas colaborativas

A medida que mejore su infraestructura de desarrollo, no olvide el elemento humano. El desarrollo de software es intrínsecamente un esfuerzo colaborativo. Involucra equipos multifuncionales compuestos por desarrolladores, diseñadores, gerentes de producto, expertos en Scrum, analistas de negocios y otras partes interesadas. Estas personas trabajan de forma colectiva para hacer realidad las ideas. Al utilizar herramientas de colaboración modernas y fomentar una cultura de comunicación abierta e intercambio de conocimientos, puede mejorar significativamente la productividad y la eficacia de sus equipos de desarrollo de software. En su experiencia de desarrollo de software basada en la IA, estas herramientas adquieren nuevas dimensiones. Puede integrar la IA en las plataformas de colaboración para facilitar una comunicación y un intercambio de conocimientos más eficaces entre los miembros del equipo. Los asistentes de IA pueden responder a preguntas frecuentes, resumir los debates o incluso mediar en conflictos. La IA generativa puede mejorar los procesos de revisión de código al sugerir automáticamente mejoras o identificar posibles

problemas. Además, puedes usar la IA para crear documentación dinámica y contextual que se actualice casi en tiempo real a medida que el proyecto evoluciona, de modo que todos los miembros del equipo tengan acceso a la información más actual y relevante.

Automatizar las tareas repetitivas

Al utilizar la IA generativa para gestionar las actividades rutinarias que consumen mucho tiempo, sus equipos de software pueden centrarse en trabajos creativos y de gran valor que impulsen la innovación y generen un impacto empresarial. Algunos ejemplos de tareas repetitivas incluyen la generación de código repetitivo, la creación de datos de prueba, la redacción de documentación o incluso la redacción de los planes iniciales del proyecto. Al delegar estas tareas en manos de la IA, los miembros del equipo pueden centrarse en un trabajo más creativo y estratégico. Por ejemplo, las herramientas de finalización de código impulsadas por la IA pueden acelerar considerablemente el proceso de codificación al sugerir fragmentos de código relevantes en función del contexto y los patrones de codificación. Del mismo modo, la IA generativa puede crear y actualizar automáticamente la documentación técnica a medida que cambia el código. Esto mantiene la documentación actualizada y reduce el esfuerzo manual que normalmente se requiere para esta tarea. En las pruebas, la IA puede generar casos de prueba exhaustivos basados en los requisitos y el análisis del código, lo que mejora la cobertura de las pruebas y reduce la probabilidad de que se pasen por alto los casos extremos. Al automatizar de forma inteligente estas tareas repetitivas, la IA generativa acelera los plazos de desarrollo, mejora la coherencia y reduce los errores humanos. El resultado son resultados de software de mayor calidad.

Revisar e iterar periódicamente la experiencia de desarrollo

Su experiencia de desarrollo de software en sí misma debe tratarse como un producto que requiere un perfeccionamiento continuo. Esto implica establecer un proceso sistemático para revisar e iterar periódicamente todos los aspectos del ciclo de vida, las herramientas y las prácticas del desarrollo. Realice evaluaciones periódicas de toda la cadena de herramientas, los flujos de trabajo y los procesos. Recopile los comentarios de todos los miembros del equipo en sus distintas funciones, incluidos los gerentes de producto, los diseñadores, los arquitectos, los desarrolladores, los evaluadores y el personal de operaciones. Pídeles que identifiquen los puntos débiles, los obstáculos y las oportunidades de mejora. Por ejemplo, los equipos podrían realizar revisiones trimestrales del rendimiento de CI/CD sus proyectos y analizar métricas como los tiempos de construcción, la frecuencia de implementación y las tasas de error para identificar las áreas de optimización. Dado que las capacidades de la IA generativa siguen evolucionando rápidamente, es fundamental evaluar

de forma coherente las nuevas herramientas y funciones impulsadas por la IA que puedan agilizar aún más los flujos de trabajo o aumentar las capacidades en todas las funciones del SDLC.

Adoptar prácticas de gestión de proyectos eficaces

Para organizar sus complejos esfuerzos de desarrollo de software de forma eficaz, adopte prácticas de gestión de proyectos aumentadas por la IA. En este contexto, la gestión eficaz de proyectos va más allá de las metodologías tradicionales. Adopta enfoques aumentados por la IA que mejoran la planificación, la ejecución y la supervisión en todo el SDLC. Los marcos ágiles promueven la flexibilidad, la colaboración y la iteración rápida, y puede utilizar la IA generativa para optimizar estos procesos. Por ejemplo, la IA generativa puede analizar los datos históricos de los proyectos para obtener estimaciones más precisas, generar y priorizar automáticamente las historias de los usuarios en función de los objetivos empresariales y los comentarios de los clientes, y proporcionar información inteligente sobre el rendimiento del equipo. Las herramientas de gestión de proyectos impulsadas por la IA pueden predecir posibles obstáculos y sugerir la asignación óptima de tareas en función de las habilidades y la carga de trabajo de los miembros del equipo. Al integrar las capacidades impulsadas por la IA en las prácticas de gestión de proyectos, puede lograr una mayor visibilidad, tomar decisiones basadas en los datos con mayor rapidez y asegurarse de que los miembros del equipo estén alineados y trabajen de manera eficiente para alcanzar los objetivos comunes.

Implementar la gestión del conocimiento

A medida que vaya madurando su experiencia de desarrollo de software basada en la IA, implemente un sistema de gestión del conocimiento sólido. Un sistema de gestión del conocimiento sólido le ayuda a capturar, organizar y conceder acceso a información valiosa, mejores prácticas y soluciones. Todos los miembros del equipo del SDLC deberían tener fácil acceso al sistema. Utilice la IA generativa para crear bases de conocimiento dinámicas e inteligentes que evolucionen con su organización. Por ejemplo, la IA puede generar y actualizar automáticamente la documentación en función de los cambios en el código, las conversaciones y los artefactos del proyecto, de modo que la información se mantenga actualizada sin intervención manual. La IA generativa también puede potenciar las capacidades de búsqueda inteligente y ayudar a los miembros del equipo a encontrar rápidamente la información relevante mediante consultas en lenguaje natural, incluso si no conocen la terminología exacta. Además, la IA generativa puede mostrar de forma proactiva información relevante a los miembros del equipo en función de sus tareas o desafíos actuales. Actúa como un mentor virtual que mejora la toma de decisiones y la resolución de problemas en todos los

roles. Al implementar un sistema de gestión del conocimiento basado en la IA, puede eliminar los compartimentos aislados, acelerar la incorporación, reducir el trabajo redundante y fomentar una cultura de aprendizaje e innovación continuos en todo su equipo de desarrollo de software.

Proporciona extensibilidad y personalización

Para maximizar los beneficios de la IA generativa en el desarrollo de software, asegúrate de que tus herramientas y plataformas basadas en la IA sean ampliables y personalizables. Esto le ayuda a adaptar las capacidades de IA a sus necesidades, flujos de trabajo y conjuntos tecnológicos específicos. Por ejemplo, puede ajustar los modelos de IA a partir de sus propias bases de código y documentación, crear herramientas personalizadas basadas en la IA para tareas específicas o integrar las capacidades de la IA en las herramientas y los procesos existentes. Esta extensibilidad le ayuda a desarrollar la experiencia de desarrollo basada en la IA para adaptarla a las necesidades cambiantes de la organización. También le ayuda a optimizar la experiencia para dominios o tipos de proyectos específicos.

Optimización para las operaciones

La IA generativa desempeña un papel crucial en la optimización de las operaciones y el mantenimiento del software. Optimice las operaciones integrando las capacidades de la IA en sus herramientas y procesos operativos. Por ejemplo, utilice la IA generativa para analizar los datos de registro prácticamente en tiempo real, predecir posibles fallos del sistema y automatizar las tareas de mantenimiento rutinarias. La IA generativa también puede ayudar a analizar la causa raíz al correlacionar eventos en sistemas distribuidos complejos. Esto mejora la confiabilidad del sistema, reduce el tiempo de inactividad y permite que sus equipos de operaciones se centren en iniciativas más estratégicas.

Uso de información basada en datos

Utilice la información basada en datos a lo largo de su proceso de desarrollo basado en la IA. Implemente sistemas para recopilar, analizar y actuar en función de los datos de todas las etapas del SDLC. Esto incluye las métricas del código, los resultados de las pruebas, los datos de implementación, los comentarios de los usuarios y el rendimiento operativo. Utilice la IA generativa para descubrir patrones e información que podrían no ser evidentes para los observadores humanos. Luego, incorpore estos conocimientos a su proceso de desarrollo para fundamentar todo, desde las decisiones arquitectónicas hasta la priorización de las características.

Adoptar un enfoque basado en plataformas

Para aprovechar al máximo los beneficios de la IA generativa en el desarrollo de software, adopte un enfoque basado en plataformas. Cree una plataforma integral e integrada que incorpore las capacidades de IA en todos los aspectos del SDLC. La plataforma debe proporcionar una experiencia de usuario coherente, una gestión y datos centralizados y una integración perfecta entre las diferentes herramientas y procesos. Esto hace que los beneficios de la IA estén disponibles de manera uniforme en toda la organización, reduce la sobrecarga que supone gestionar múltiples y dispares herramientas de IA y proporciona una base para la mejora y expansión continuas de las capacidades de la IA.

Medición del éxito de la IA generativa en el desarrollo de software

Para medir de forma eficaz el efecto de implementar una experiencia de desarrollo de software generativa basada en la IA, es necesario establecer un conjunto integral de métricas que abarquen diversas dimensiones del ciclo de vida del desarrollo de software (SDLC). Estas métricas deberían reflejar las mejoras inmediatas en la eficiencia y la productividad, y también reflejar las ganancias a largo plazo en la calidad del software, la satisfacción del equipo y el valor empresarial.

Haga lo siguiente para utilizar de forma eficaz las métricas recomendadas en esta sección:

1. Establezca bases de referencia: antes de sumergirse en la implementación de su experiencia de desarrollo basada en la IA, dedique un tiempo a recopilar datos exhaustivos sobre su rendimiento actual en función de estas métricas. Esto proporciona un punto de partida claro y le ayuda a realizar comparaciones significativas más adelante.
2. Establezca objetivos realistas: con sus puntos de referencia a mano, establezca objetivos de mejora alcanzables para cada métrica. Sea ambicioso pero realista. Recuerde que el progreso sostenible suele ser gradual.
3. Implemente un monitoreo continuo: utilice herramientas automatizadas para recopilar y analizar constantemente los datos de estas métricas en su entorno. La supervisión prácticamente en tiempo real le ayuda a supervisar el progreso e identificar rápidamente cualquier problema u oportunidad.
4. Realice revisiones periódicas: programe sesiones de revisión trimestrales o semestrales en las que usted y su equipo evalúen minuciosamente su progreso en relación con los objetivos. Usa estas sesiones para identificar áreas en las que puedas seguir mejorando y celebrar tus éxitos.
5. Repite y ajústelo: en función de la información que haya obtenido, perfeccione continuamente su implementación de la IA generativa y ajuste los objetivos según sea necesario.

En esta sección se describen las siguientes categorías de métricas:

- [Velocidad de despliegue](#)
- [Calidad del código](#)
- [Eficiencia operativa](#)
- [Productividad y satisfacción del equipo](#)

- [Impacto empresarial](#)

Velocidad de despliegue

Considere la posibilidad de medir las siguientes métricas de velocidad de despliegue.

Métrica	Description (Descripción)
Hora de comercialización	Mida la reducción del tiempo desde la concepción de la idea hasta la implementación en producción
Velocidad de sprint	Realiza un seguimiento del aumento de los puntos de historia completados por cada sprint por tus equipos
Frecuencia de confirmación de código	Supervise el aumento de las confirmaciones de código, lo que indica una aceleración de los ciclos de desarrollo
Tiempo de resolución de solicitudes de extracción	Evalúe la reducción del tiempo necesario para revisar y combinar los cambios de código en sus repositorios
Velocidad de lanzamiento	Mida el aumento en el número de lanzamientos por trimestre o año

Calidad del código

Considere la posibilidad de medir las siguientes métricas de calidad del código.

Métrica	Description (Descripción)
Densidad de defectos	Mida la reducción de los errores de software

Métrica	Description (Descripción)
Cobertura del código	Realice un seguimiento del aumento del porcentaje de cobertura de las pruebas en todo su código base
Deuda técnica	Supervise la disminución de la deuda técnica identificada a lo largo del tiempo
Puntuaciones del análisis de código estático	Evalúe las mejoras en la calidad del código en función de sus herramientas de análisis automatizadas

Eficiencia operativa

Considere la posibilidad de medir las siguientes métricas de eficiencia operativa.

Métrica	Description (Descripción)
Frecuencia de despliegue	Mida el aumento en el número de despliegues exitosos
Tiempo medio de recuperación (MTTR)	Realice un seguimiento de la reducción del tiempo que se tarda en recuperarse de los fallos del sistema
Cambie la tasa de fallas	Supervise la disminución del porcentaje de cambios que provocan errores en sus implementaciones

Productividad y satisfacción del equipo

Considera la posibilidad de medir las siguientes métricas de productividad y satisfacción del equipo.

Métrica	Description (Descripción)
Mejora de la productividad	Supervise el aumento del porcentaje de productividad de cada tarea
Puntuación de satisfacción	Realiza encuestas periódicas para evaluar la mejora en la moral y la satisfacción laboral de tu equipo
Eficiencia en el intercambio de conocimientos	Mide la reducción del tiempo que tu equipo dedica a buscar información o a hacer preguntas repetitivas
Tiempo de incorporación	Realice un seguimiento de la disminución del tiempo necesario para que los nuevos miembros del equipo sean productivos

Impacto empresarial

Considere la posibilidad de medir las siguientes métricas de impacto empresarial.

Métrica	Description (Descripción)
Tasa de adopción de funciones	Mide el aumento de la participación de los usuarios con las nuevas funciones que hayas lanzado
Puntuación de satisfacción de los clientes	Realiza un seguimiento de las mejoras en los comentarios y valoraciones de tus usuarios
Impacto en los ingresos (directo e indirecto)	Evalúe el aumento de los ingresos atribuido al aumento de la velocidad de lanzamiento o al aumento de la productividad

Conclusión

Este documento de estrategia proporciona una visión general de una experiencia de desarrollo de software basada en la IA generativa. Explora las cinco dimensiones del [marco 5-I](#): investigar, integrar, interactuar, iterar e impactar. Estas dimensiones proporcionan una hoja de ruta estratégica para integrar la IA generativa en todo el ciclo de vida del desarrollo de software (SDLC). También describe las [capacidades fundamentales](#) que se requieren para implementar con éxito este marco. Las capacidades abarcan áreas como la gestión de proyectos DevSecOps, los asistentes de IA, la gestión del conocimiento y más. Proporciona [las mejores prácticas](#) que se deben tener en cuenta a la hora de integrar la IA generativa y le ayuda a utilizar [métricas](#) para medir el impacto que la IA generativa tiene en su experiencia de desarrollo de software.

La integración de la IA generativa en los procesos de desarrollo de software representa un cambio de paradigma que tiene el potencial de acelerar la innovación, mejorar la calidad y aumentar la productividad. Sin embargo, es importante reconocer que no se trata de una implementación única. Se trata de una evolución continua que requiere un esfuerzo sostenido y un perfeccionamiento continuo.

Al embarcarse en este viaje, le recomendamos que comience con una evaluación exhaustiva de las capacidades y la preparación actuales de su organización. La herramienta de [AWS evaluación es una herramienta](#) de evaluación del desarrollo de software basada en la inteligencia artificial que puede ayudarle a identificar las áreas prioritarias y a crear una hoja de ruta de implementación personalizada.

Recursos

Una vez que haya identificado las áreas prioritarias clave, los siguientes recursos pueden ayudarlo a implementar su hoja de ruta:

AWS documentación

- [Automatice las operaciones de AWS infraestructura mediante Amazon Bedrock](#) (Guía AWS prescriptiva)
- [Prácticas recomendadas con Amazon Q Developer para la generación de código en línea y asistente](#) (orientación AWS prescriptiva)
- [Desarrolle un asistente totalmente automatizado basado en el chat mediante los agentes y las bases de conocimiento de Amazon Bedrock](#) (Guía AWS prescriptiva)

- [Transforme los modelos operativos de desarrollo y mantenimiento de aplicaciones AWS con la IA generativa](#) (orientación prescriptiva)AWS
- [Utilice Amazon Q Developer como asistente de programación para aumentar su productividad](#) (orientación AWS prescriptiva)

AWS publicaciones de blog y tutoriales

- [Publicaciones del blog de Amazon Q](#)
- [Acelere su ciclo de vida de desarrollo de software con Amazon Q](#) (AWS entrada del blog)
- [Creación de un agente de IA para arquitectos de AWS soluciones: aprovechar Amazon Bedrock para la arquitectura y el despliegue automatizados \(vídeo\)](#)AWS
- Operaciones de [tecnología generativa impulsadas por](#) la IA (entrada del blog)AWS
- [Modernice su aplicación Java con Amazon Q Developer](#) (AWS entrada del blog)
- [Utilice Amazon Bedrock para generar, evaluar y comprender el código de su proceso de desarrollo de software](#) (entrada AWS del blog)

Historial de documentos

En la siguiente tabla, se describen cambios significativos de esta guía. Si quiere recibir notificaciones de futuras actualizaciones, puede suscribirse a las [notificaciones RSS](#).

Cambio	Descripción	Fecha
Publicación inicial	No aplicable	18 de abril de 2025

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.