



Verkürzung der Lebenszyklen der Softwareentwicklung mit generativer KI AWS

AWS Präskriptive Leitlinien



AWS Präskriptive Leitlinien: Verkürzung der Lebenszyklen der Softwareentwicklung mit generativer KI AWS

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

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irreführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Einführung	1
Ziele	2
Zielgruppe	2
Erfahrung in der Entwicklung	3
Einsatz generativer KI	5
5-I-Framework	6
Überblick über das Framework	7
Integration mit dem SDLC	9
Grundlegende Funktionen	10
Projektmanagement	16
Anforderungsmanagement	20
Architektur und Design	21
Zusammenarbeit	22
DevSecOps	23
Betrieb und Wartung	32
KI-Assistenten	35
Analytik und Einblicke	37
Wissensmanagement	40
Erweiterbarkeit	41
Best Practices	43
Integrierte Toolchain	43
DevSecOps Pipeline	44
Tools und Praktiken für die Zusammenarbeit	44
Automatisierung von Aufgaben	45
Überprüfung und Iteration	45
Praktiken des Projektmanagements	46
Wissensmanagement	46
Erweiterbarkeit und Anpassung	47
Optimierung	47
Datengestützte Erkenntnisse	47
Plattformbasierter Ansatz	48
Messung des Erfolgs	49
Geschwindigkeit der Bereitstellung	50
Qualität des Codes	50

Operationelle Effizienz	51
Produktivität und Zufriedenheit des Teams	51
Auswirkungen auf das Geschäft	52
Schlussfolgerung	53
Ressourcen	53
Dokumentverlauf	55
Glossar	56
#	56
A	57
B	60
C	62
D	66
E	70
F	72
G	74
H	75
I	77
L	80
M	81
O	85
P	88
Q	91
R	92
S	95
T	99
U	101
V	101
W	102
Z	103
.....	civ

Verkürzung der Lebenszyklen der Softwareentwicklung mit generativer KI AWS

Chetan Makvana, Amazon Web Services

April 2025 (Geschichte [der Dokumente](#))

Die wachsende Nachfrage nach qualitativ hochwertiger Software veranlasst Unternehmen, ständig nach Möglichkeiten zu suchen, ihren Softwareentwicklungszyklus (SDLC) zu beschleunigen. Da Unternehmen bestrebt sind, wettbewerbsfähig zu bleiben, ist es von entscheidender Bedeutung, die Markteinführungszeit zu verkürzen und gleichzeitig die Produktqualität aufrechtzuerhalten oder zu verbessern. Um diesen Herausforderungen zu begegnen, muss sich die Softwareentwicklung weiterentwickeln und modernste Technologien, Methoden und Verfahren einsetzen, die Prozesse rationalisieren und Softwareentwicklungsteams in die Lage versetzen, produktiver und kreativer zu arbeiten. Das Aufkommen der Entwicklungserfahrung der nächsten Generation markiert einen bedeutenden Wandel in der Art und Weise, wie Software konzipiert, gebaut, getestet und eingesetzt wird. Es integriert eine Vielzahl von Funktionen — darunter Cloud-native Entwicklung, KI-gestützte Automatisierung, fortschrittliches Projektmanagement, Tools für die Zusammenarbeit usw. DevSecOps —, die zusammen die Effizienz und Effektivität des SDLC verbessern.

Im Vordergrund dieser Transformation steht der Aufstieg der generativen KI in der Softwareentwicklung. Laut [Gartner](#) werden bis 2027 40% der Plattformentwicklungsteams KI einsetzen, um jede Phase des SDLC zu erweitern, verglichen mit nur 5% im Jahr 2023. In diesem Bericht heißt es auch, dass führende Softwareentwickler sich nun darauf vorbereiten müssen, generative KI in einem breiteren Spektrum von Bereichen einzuführen, die für den Entwicklungsprozess von entscheidender Bedeutung sind. In einem anderen Bericht zeigen [McKinsey](#) Untersuchungen, dass Unternehmen mit einem höheren Developer Velocity Index ihren Umsatz vier- bis fünfmal schneller steigern, 60% höhere Aktionärsrenditen erzielen und 55% innovativer sind. Durch die Einführung generativer KI, die über die bloße Codegenerierung hinausgeht, können Unternehmen ein neues Maß an Effizienz, Produktivität und Innovation in ihren Softwareentwicklungsworkflows erreichen. Dies kann den manuellen Aufwand reduzieren, Erkenntnisse gewinnen und das menschliche Fachwissen erweitern.

Ziele

In diesem Strategiedokument werden ein Framework, grundlegende Funktionen, Anwendungsfälle, bewährte Verfahren und Erfolgsmetriken beschrieben, mit denen Sie Ihr SDLC mit generativer KI beschleunigen können. Es beschreibt, wie generative KI effektiv in alle Entwicklungsphasen integriert werden kann, um die Produktqualität und Effizienz zu verbessern.

Dieses Strategiedokument kann Ihnen und Ihrer Organisation helfen, die folgenden Ziele zu erreichen:

- Implementieren Sie ein Framework, grundlegende Funktionen, Anwendungsfälle, Best Practices und Erfolgsmetriken, um Ihr SDLC mit generativer KI zu beschleunigen.
- Integrieren Sie generative KI effektiv in alle Entwicklungsphasen, um die Produktqualität, die Veröffentlichungsgeschwindigkeit und die Entwicklungseffizienz zu verbessern.
- Passen Sie sich an die nächste Generation der Softwareentwicklung an, indem Sie modernste KI-Technologien, -Methoden und -Praktiken integrieren, die Prozesse rationalisieren und Entwicklungsteams stärken.

Zielgruppe

Dieses Strategiedokument richtet sich an IT-Führungskräfte, technische Manager, Chief Technology Officers und Softwareentwicklungsteams, die ihren Softwareentwicklungszyklus beschleunigen möchten, indem sie generative KI auf ihre Entwicklungspraktiken anwenden.

Die Erfahrung in der Softwareentwicklung verstehen

Die Erfahrung in der Softwareentwicklung umfasst die Umgebung, Tools und Prozesse, die Ihre Entwicklungsteams während des gesamten Softwareentwicklungszyklus (SDLC) verwenden. Es umfasst die integrierte Entwicklungsumgebung (IDE), Kollaborationsplattformen, Test-Frameworks, Wissensmanagementsysteme, Bereitstellungspipelines und mehr.

Ein gut durchdachtes Entwicklungserlebnis optimiert Arbeitsabläufe, reduziert den manuellen Aufwand und ermöglicht es Ihren Teams, sich auf wichtige Aufgaben zu konzentrieren, was letztendlich Ihren SDLC beschleunigt. Durch die nahtlose Integration Ihrer IDE, Ihres Versionskontrollsystems und Ihrer Bereitstellungstools ermöglichen Sie es Entwicklern beispielsweise, Code schneller und effizienter zu schreiben, zu testen und bereitzustellen als mit einer fragmentierten Toolchain, die manuelle Übergaben und Kontextwechsel erfordert. In ähnlicher Weise hilft die Integration eines robusten Wissensmanagement-Frameworks den Teams dabei, auf institutionelles Wissen, bewährte Verfahren und Dokumentation zuzugreifen und diese auszutauschen. Dies verbessert ihre allgemeine Produktivität und ihre Fähigkeiten zur Problemlösung.

Die Erfahrung in der Softwareentwicklung wirkt sich direkt auf die Gesamtleistung und den Erfolg eines Softwareentwicklungsteams aus. Eine suboptimale Erfahrung kann zu Folgendem führen:

- Geringere Produktivität — Ineffiziente Tools, komplexe Workflows und mangelnde Automatisierung beeinträchtigen die Teamproduktivität, was die Bereitstellung von Funktionen und Updates verlangsamt.
- Zunehmende technische Verschuldung — Schlecht integrierte Tools und Ad-hoc-Prozesse können zu technischen Schulden führen, was die Wartung und Skalierung Ihrer Softwaresysteme im Laufe der Zeit schwieriger macht.
- Verminderte Innovation — Wenn Ihr Team mit manuellen, sich wiederholenden Aufgaben feststeckt, ist die Fähigkeit Ihres Teams, neue Technologien zu erforschen und Innovationen voranzutreiben, eingeschränkt.
- Beeinträchtigte Qualität — Fragmentierte Test- und Bereitstellungsprozesse erhöhen das Risiko von Softwarefehlern und Sicherheitslücken. Dies kann sich negativ auf die Gesamtqualität der gelieferten Software auswirken.

Wenn Sie in ein gut durchdachtes Softwareentwicklungserlebnis investieren, können Sie erhebliche Vorteile erzielen, wie z. B. kürzere Markteinführungszeiten, verbesserte Softwarequalität, höhere Zufriedenheit des Softwareentwicklungsteams und größere geschäftliche Flexibilität.

Förderung der Softwareentwicklungserfahrung mit generativer KI

Die Integration generativer KI in den Softwareentwicklungszyklus (SDLC) stellt einen Paradigmenwechsel in der Art und Weise dar, wie ganze Softwareentwicklungsteams Softwarelösungen konzipieren, entwerfen, implementieren und warten. Generative KI hat das Potenzial, jede Phase des SDLC zu revolutionieren, einschließlich Projektmanagement, Erfassung von Anforderungen, Design, Codierung, Test, Bereitstellung und Wartung.

Im Kern fungiert eine generative KI-gestützte Entwicklungserfahrung als intelligenter Partner für Ihr gesamtes Softwareentwicklungsteam, einschließlich Produktmanagern, Designern, Lösungsarchitekten, Entwicklern, Testern und Betriebspersonal. Es bietet kontextsensitive Unterstützung, generiert Artefakte (wie User Stories, Designmodelle, Codefragmente und Testfälle), bietet Vorschläge nahezu in Echtzeit und prognostiziert sogar potenzielle Probleme, bevor sie auftreten. Dieser KI-gestützte Ansatz reduziert die kognitive Belastung der Teammitglieder erheblich. Auf diese Weise können sie sich auf strategische Entscheidungen auf hoher Ebene und komplexe Problemlösungen konzentrieren, während die generative KI die alltäglicheren, sich wiederholenden Aufgaben erledigt.

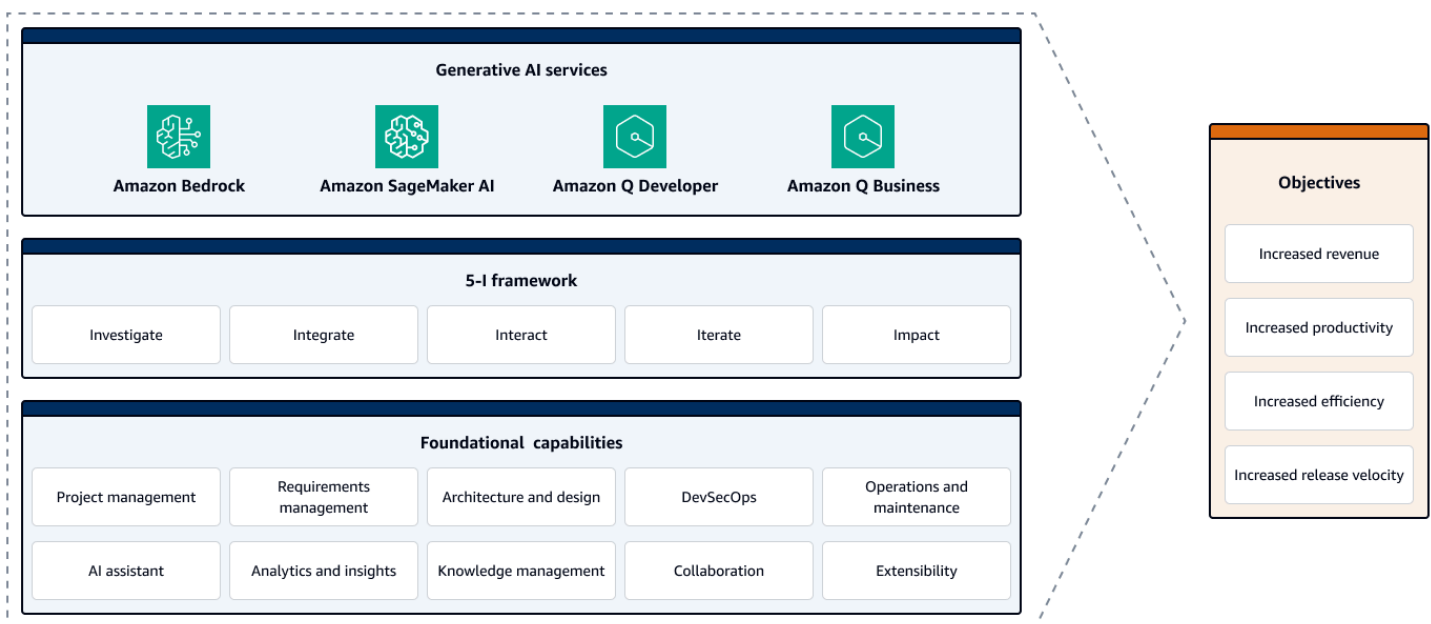
Generative KI dient auch als Wissensverstärker. Es hilft Teammitgliedern, schnell auf relevante Informationen, bewährte Verfahren und Muster aus riesigen Datenrepositorien zuzugreifen. Dadurch kann Fachwissen im gesamten Unternehmen effektiv demokratisiert werden. Durch die nahtlose Integration generativer KI-Funktionen in die gesamte Entwicklungs-Toolchain können Sie eine intuitivere, effizientere und produktivere Umgebung für Ihre gesamten Softwareentwicklungsteams schaffen. Diese verbesserte Entwicklungserfahrung beschleunigt den SDLC und verbessert die Gesamtqualität. Es reduziert auch Fehler und fördert Innovation, da die Teammitglieder neue Ideen und Ansätze schneller entwickeln können.

Um ein generatives KI-gestütztes Entwicklungserlebnis in Ihrem Unternehmen einzuführen, sollten Sie die folgenden Schlüsselemente berücksichtigen:

- [5-I-Framework](#)— Das 5-I-Framework besteht aus fünf Dimensionen und bietet einen umfassenden Ansatz zur Steuerung des Prozesses der modernen Softwareentwicklung. Es bietet eine strukturierte Methodik, mit der Sie generative KI systematisch in allen Phasen des SDLC anwenden können.

- **Grundlegende Funktionen**— Um das Potenzial der generativen KI in allen Bereichen der modernen Softwareentwicklung voll auszuschöpfen, müssen Sie eine Reihe robuster grundlegender Funktionen einrichten. Diese Funktionen bilden das Rückgrat einer KI-gestützten Entwicklungserfahrung. Diese Funktionen helfen Ihnen dabei, generative KI im gesamten SDLC zu integrieren und zu nutzen.

Zusammen bilden das 5-I-Framework und die grundlegenden Funktionen eine Strategie zur Neugestaltung der Softwareentwicklungserfahrung. Die fünf Dimensionen bieten einen strategischen Rahmen für die Anwendung generativer KI, und die grundlegenden Funktionen bereiten Ihr Unternehmen darauf vor, diesen KI-gestützten Ansatz zu unterstützen. AWS-Services, wie [Amazon Bedrock](#), [Amazon SageMaker AI](#), [Amazon Q Developer](#) und [Amazon Q Business](#), bieten generative KI-Funktionen und -Features, die Sie in Ihre Softwareentwicklungserfahrung integrieren können.



5-I-Framework für eine KI-gestützte Softwareentwicklungserfahrung

Das 5-I-Framework bietet Softwareentwicklungsteams einen strukturierten Ansatz, um generative KI effektiv in ihre Entwicklungspraktiken zu integrieren. Es hilft Ihnen dabei, eine solide Grundlage für den Einsatz generativer KI im gesamten SDLC zu schaffen. Es hilft Ihnen auch dabei, die richtigen Entwicklungspraktiken, Workflows und Denkweisen einzurichten, um das Potenzial der generativen KI voll auszuschöpfen.

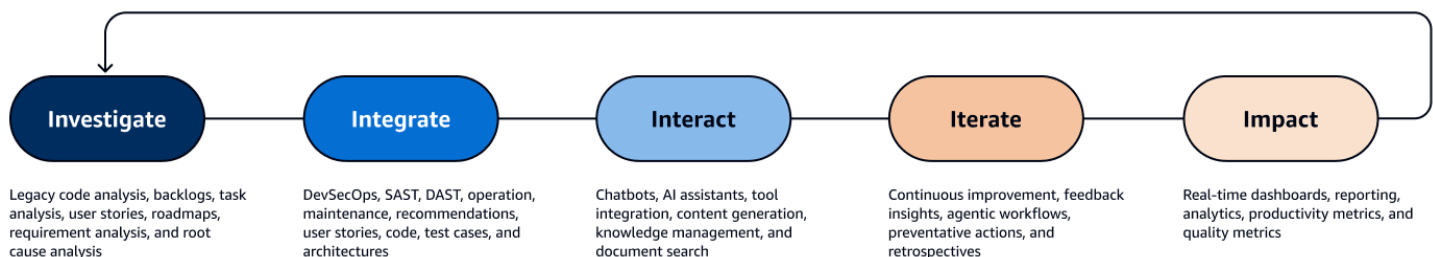
In diesem Abschnitt werden folgende Themen behandelt:

- [Überblick über das Framework](#)
- [Integration in den Lebenszyklus der Softwareentwicklung](#)

Überblick über das Framework

Das 5-I-Framework basiert auf fünf Schlüsseldimensionen: Untersuchen, Integrieren, Interagieren, Iterieren und Auswirkungen. Jede Dimension stellt einen kritischen Bereich dar, in dem generative KI den Softwareentwicklungsprozess erheblich verbessert. Durch die strategische Integration generativer KI in all diesen Dimensionen trägt das Framework den sich wandelnden Bedürfnissen der modernen Softwareentwicklung Rechnung. Es kann die kognitive Belastung reduzieren und das kreative Potenzial steigern. Es wird anerkannt, dass es bei der idealen Entwicklungserfahrung nicht nur um Tools geht, sondern auch darum, eine Umgebung zu schaffen, in der KI die menschlichen Fähigkeiten in jeder Phase nahtlos verbessert.

Das folgende Diagramm zeigt die fünf Dimensionen der KI-gestützten Softwareentwicklung. Für jede Dimension wird aufgezeigt, wo Sie generative KI integrieren können, um Effizienz und Innovation zu fördern.



Im Folgenden sind die fünf Dimensionen des Frameworks aufgeführt:

- **Untersuchen** — Verbessern Sie jede analytische Aufgabe in Ihrem Softwareentwicklungsprozess mit generativer KI. Nutzen Sie generative KI, um Anforderungen zu verstehen, riesige Datenmengen zu verarbeiten, Muster zu erkennen und Erkenntnisse zu gewinnen, deren Erstellung möglicherweise über menschliche Fähigkeiten hinausgeht oder deren Erstellung erheblich länger dauern würde. Diese Erkenntnisse helfen Ihnen dabei, fundiertere Entscheidungen zu treffen, Verbesserungsmöglichkeiten schnell zu erkennen und qualitativ hochwertige Software effizienter bereitzustellen. Generative KI kann ein intelligenter Partner für die Analyseprozesse im gesamten SDLC sein. Durch die Nutzung generativer KI wenden Sie eingehende Analysen auf kritische Bereiche an, wie z. B. die Erfassung von Anforderungen, die Prüfung älterer Codebasen und die Optimierung des Produktrückstands. Produkteigentümer können generative

KI beispielsweise verwenden, um Nutzererlebnisse oder Anforderungen zu analysieren, bevor sie User Stories erstellen. Entwicklungsteams können Ineffizienzen aufdecken und Optimierungsmöglichkeiten in bestehenden Codebasen identifizieren. DevOps Ingenieure können Ursachenanalysen anwenden, um Leistungsprobleme oder Sicherheitslücken schnell zu diagnostizieren, was die Zuverlässigkeit verbessern kann.

- **Integrieren** — Integrieren Sie generative KI, um eine Vielzahl von Aufgaben und Prozessen im gesamten SDLC zu automatisieren. Dazu gehört die automatische Generierung von Codefragmenten, Testfällen, Architekturentwürfen, Anwenderberichten und Bereitstellungs Pipelines. Durch die Automatisierung dieser typischerweise manuellen Aufgaben können sich Teams auf strategischere und innovativere Aufgaben konzentrieren, was zu einer schnelleren Markteinführung und zu qualitativ hochwertigen Anwendungen führt. Die Integrate-Dimension stellt einen Paradigmenwechsel in der Softwareentwicklung dar, bei dem KI zu einem integralen Bestandteil des Entwicklungsprozesses wird. Sie arbeitet mit Ihrem Softwareentwicklungsteam zusammen, um die Produktivität zu steigern, die Qualität zu verbessern und Innovationen voranzutreiben. Dies führt zu einer schnelleren Markteinführung. Es stellt Ihre Softwareentwicklungsteams vor die Herausforderung, ihre Prozesse und Workflows regelmäßig zu überprüfen, indem sie bei jedem Schritt fragen: „Kann das automatisiert werden?“
- **Interagieren** — Verwenden Sie generative KI-gestützte Assistenten, um Ihrem Team sofortige, kontextbezogene Unterstützung bei einer Reihe von Aufgaben und Fragen zu bieten. Diese intelligenten Assistenten agieren als sachkundige Mitarbeiter, die auf eine riesige Menge an Informationen zurückgreifen. Sie können Fragen zur Programmierung beantworten, Designvorschläge unterbreiten, Standardarbeitsanweisungen erklären und bei der Behebung komplexer Probleme helfen. Die Integration dieser KI-Assistenten in den Entwicklungsablauf steigert die Produktivität und fördert eine kollaborativere Umgebung zur Problemlösung.
- **Iterieren** — Verwenden Sie generative KI, um schnelle, datengestützte Anpassungen im gesamten SDLC zu ermöglichen. Sie können kontinuierlich Daten aus Quellen wie Kundenfeedback, Nutzungsmustern, Markttrends und Kennzahlen zur Teamleistung analysieren, um schnell fundierte Entscheidungen zu treffen. Diese Anpassungsfähigkeit verfeinert Ihre Softwareentwicklung von einem statischen, vordefinierten Prozess zu einem flüssigen, responsiven Ansatz. Sie zeigt sich auf verschiedene Weise, unter anderem durch dynamische Priorisierung von Backlogs, flexible Ressourcenzuweisung, anpassungsfähige Teststrategien, sich entwickelnde Dokumentation und reaktionsschnelle Bereitstellungsprozesse. Produktmanager können beispielsweise KI-generierte Erkenntnisse nutzen, um ihre Backlogs neu zu ordnen und so neue Kundenanforderungen und Markttrends nahezu in Echtzeit zu integrieren. DevOps Ingenieure können Bereitstellungspläne und Infrastrukturkonfigurationen auf der Grundlage von Leistungsanalysen anpassen und so

sicherstellen, dass Anwendungen robust und optimiert bleiben. Entwicklungsteams können Feedback aus Sprint-Retrospektiven in umsetzbare Verbesserungen für die nächste Iteration umsetzen und so eine Kultur der kontinuierlichen Prozessverbesserung fördern.

- **Wirkung** — Wenden Sie generative KI an, um die Effektivität und Leistung Ihres Softwareentwicklungsprozesses zu bewerten. Durch den Einsatz von KI-gestützten Analysen und Metriken erhalten Sie tiefere Einblicke in die Entwicklungseffizienz, die Codequalität, die Benutzerinteraktion und die allgemeine Anwendungsleistung. Dieser datengesteuerte Ansatz hilft Ihnen, fundierte Entscheidungen zu treffen, Ihre Entwicklungsabläufe zu optimieren und die Qualität und Benutzererfahrung Ihrer Anwendungen kontinuierlich zu verbessern. Bei der Bewertung der Produktivität von Softwareteams analysiert generative KI verschiedene Datenpunkte, z. B. die Häufigkeit von Code-Commits, Problemlösungszeiten, Release-Geschwindigkeit, Feature-Bereitstellungsraten und mehr. Es kann auch die Qualität von Code-Reviews, die Effektivität von Tools für die Zusammenarbeit und die Auswirkungen verschiedener Entwicklungspraktiken auf die Gesamtleistung des Teams bewerten. Durch die Korrelation dieser Kennzahlen mit den Projektergebnissen identifiziert die KI Muster und Trends, die menschliche Analysten möglicherweise übersehen, und sie kann umsetzbare Erkenntnisse liefern, die die Teamproduktivität steigern. Darüber hinaus kann Ihnen generative KI dabei helfen, die Teamleistung anhand von Industriestandards oder historischen Daten zu vergleichen und personalisierte Verbesserungsempfehlungen zu geben. Sie kann auch potenzielle Engpässe oder Risiken im Entwicklungsprozess vorhersagen, sodass Sie proaktive Maßnahmen ergreifen können.

Integration in den Lebenszyklus der Softwareentwicklung

Das SDLC besteht aus mehreren Phasen, die sich von Organisation zu Organisation unterscheiden können. In der Regel umfassen diese Phasen Folgendes: Anforderungen und Planung, Design und Architektur, Implementierung, Tests, Bereitstellung sowie Betrieb und Wartung.

In der folgenden Tabelle werden die Dimensionen des 5-I-Frameworks den SDLC-Phasen zugeordnet und der Integrationsgrad für jede Dimension angegeben.

Framework-Dimension	Anforderungen und Planung	Design und Architektur	Implementierung	Testen	Bereitstellung	Betrieb und Wartung
Untersuchen	Hoch	Niedrig	Niedrig	Niedrig	Niedrig	Mittel

Framework-Dimension	Anforderungen und Planung	Design und Architektur	Implementierung	Testen	Bereitstellung	Betrieb und Wartung
Integrieren	Mittel	Medium	Hoch	Medium	Hoch	Hoch
Interagieren	Hoch	Hoch	Hoch	Mittel	Medium	Hoch
Iterieren	Mittel	Niedrig	Niedrig	Niedrig	Niedrig	Mittel
Auswirkung	Hoch	Medium	Hoch	Niedrig	Hoch	Hoch

Der Integrationsgrad variiert von hoch bis niedrig. Die Kartierung zeigt die wichtigsten Schwerpunktbereiche für jede Dimension. Investigate weist beispielsweise eine hohe Intensität in der Anforderungs- und Planungsphase auf. Integrate weist eine hohe Intensität in der Implementierungs-, Bereitstellungs-, Betriebs- und Wartungsphase auf.

Mithilfe dieser Zuordnung können Sie Ihre Bemühungen effektiv priorisieren. Wir empfehlen, dass Sie sich auf hoch, dann mittel und dann niedrig konzentrieren. Stellen Sie sicher, dass Sie einen ausgewogenen und wirkungsvollen Ansatz verfolgen, der die Softwareentwicklung mit generativer KI verbessert.

Grundlegende Funktionen für eine KI-gestützte Softwareentwicklungserfahrung

Um ein generatives KI-gestütztes Softwareentwicklungserlebnis erfolgreich zu implementieren, müssen Sie eine Reihe grundlegender Funktionen einrichten, die sich auf mehrere Personas in Ihrem Unternehmen erstrecken. Diese Fähigkeiten stehen für Ihre Fähigkeit, Ressourcen effektiv einzusetzen, Prozesse zu implementieren und die gewünschten Ergebnisse im Kontext der KI-gestützten Softwareentwicklung zu erzielen. Durch die Nutzung dieser Fähigkeiten schaffen Sie eine solide Grundlage, auf der Sie generative KI nahtlos in alle Phasen des SDLC integrieren können.

AWS bietet wichtige Dienste, die Sie bei der Implementierung dieser Funktionen unterstützen. [Amazon Q Developer](#) hilft beispielsweise dabei, die Softwareentwicklung zu beschleunigen, indem es als KI-gestützter Assistent fungiert. [Amazon Q Business](#) hilft Ihnen dabei, schnelle, relevante

Antworten auf dringende Fragen zu erhalten, Probleme zu lösen und Inhalte zu generieren. Es kann auch in Ihrem Namen handeln, indem es Tools für die Softwareentwicklung integriert. [Amazon Bedrock](#) bietet Zugriff auf grundlegende Modelle und eine breite Palette von Funktionen zur Anpassung spezifischer Entwicklungsabläufe und -anforderungen.

Indem Sie diese Funktionen nutzen AWS-Services, schaffen Sie eine solide Grundlage, die Ihnen hilft, generative KI nahtlos in alle Phasen des SDLC zu integrieren.

Im Folgenden sind die grundlegenden Funktionen aufgeführt, auf die Sie sich konzentrieren sollten:

- [Projektmanagement](#)
- [Anforderungsmanagement](#)
- [Architektur und Design](#)
- [Zusammenarbeit](#)
- [DevSecOps](#)
- [Betrieb und Wartung](#)
- [KI-Assistenten](#)
- [Analytik und Einblicke](#)
- [Wissensmanagement](#)
- [Erweiterbarkeit](#)

Jede grundlegende Funktion lässt sich in die Framework-Dimensionen und die verschiedenen Phasen des SDLC integrieren. Diese Integration hilft Ihnen, KI-Funktionen während Ihres gesamten Softwareentwicklungsprozesses effektiv zu nutzen. Sie verbessert die Effizienz, Qualität und Innovation bei jedem Schritt. Die Synergie zwischen diesen grundlegenden Funktionen, dem Framework und den SDLC-Phasen schafft ein umfassendes Ökosystem für die KI-gestützte Softwareentwicklung. Dies hilft Ihnen, das volle Potenzial generativer KI auszuschöpfen, kontinuierliche Verbesserungen voranzutreiben, Entwicklungszyklen zu beschleunigen und hochwertige Softwareprodukte zu liefern.

Die folgende Tabelle zeigt, wie die grundlegenden Funktionen und Unterfunktionen den Framework-Dimensionen und den SDLC-Phasen zugeordnet sind.

Fähigkeit: Unterfähigkeit	Untersuchen	Integrieren	Interagieren	Iterieren	Auswirkung
Projektma- nagement: Problemma- nagement	Anforderu- ngen und Planung	Keine	Keine	Keine	Keine
Projektma- nagement: Sprint- und Aufgabenm- anagement	Anforderu- ngen und Planung	Anforderu- ngen und Planung	Keine	Keine	Keine
Projektma- nagement: Verwaltung des Produkt- Backlogs	Anforderu- ngen und Planung	Keine	Keine	Anforderu- ngen und Planung	Keine
Projektma- nagement: Zuordnung von User Stories	Anforderu- ngen und Planung	Keine	Keine	Keine	Keine
Projektma- nagement: Berichter- stattung und Analytik	Anforderu- ngen und Planung	Keine	Keine	Keine	Anforderu- ngen und Planung
Projektma- nagement: Verwaltung der Produkt- Roadmap	Anforderu- ngen und Planung	Keine	Anforderu- ngen und Planung	Keine	Keine

Fähigkeit: Unterfähigkeit	Untersuchen	Integrieren	Interagieren	Iterieren	Auswirkung
Projektma- nagement: Feedbacks- chleifen	Keine	Keine	Keine	Anforderu- ngen und Planung	Keine
Projektma- nagement: Rückblicke	Keine	Keine	Keine	Anforderu- ngen und Planung	Keine
Anforderu- ngsmanage- ment	Anforderu- ngen und Planung	Anforderu- ngen und Planung	Keine	Keine	Keine
Architektur und Design: Lösungsde- sign	Design und Architektur	Design und Architektur	Keine	Keine	Keine
Zusammena- rbeit: Dokumenta- tionsmana- gement	Alle SDLC- Phasen	Keine	Alle SDLC- Phasen	Keine	Keine
Zusammena- rbeit: Wissensau- stausch	Alle SDLC- Phasen	Keine	Alle SDLC- Phasen	Keine	Keine
Zusammena- rbeit: Verwaltung von Projektan- lagen	Keine	Alle SDLC- Phasen	Alle SDLC- Phasen	Keine	Keine

Fähigkeit: Unterfähigkeit	Untersuchen	Integrieren	Interagieren	Iterieren	Auswirkung
DevSecOps: CI/CD	Testen, Einsatz	Implementierung, Testen, Bereitstellung	Bereitstellung	Keine	Keine
DevSecOps : DevOps Sicherheit	Implementierung	Implementierung, Testen, Betrieb und Wartung	Keine	Implementierung, Testen, Betrieb und Wartung	Keine
DevSecOps : Überwachung der Anwendungsleistung	Keine	Betrieb und Wartung	Keine	Keine	Keine
DevSecOps: Aggregation und Analyse von Protokollen	Betrieb und Wartung	Betrieb und Wartung	Keine	Keine	Keine
DevSecOps: AIOps	Betrieb und Wartung	Keine	Keine	Betrieb und Wartung	Keine
DevSecOps : Kontinuierliche Verbesserung	Keine	Keine	Keine	Betrieb und Wartung	Keine

Fähigkeit: Unterfähigkeit	Untersuchen	Integrieren	Interagieren	Iterieren	Auswirkung
DevSecOps : Überwachung über das Dashboard	Keine	Betrieb und Wartung	Keine	Keine	Keine
DevSecOps : Einblicke in die Leistung	Betrieb und Wartung	Keine	Keine	Betrieb und Wartung	Keine
Betrieb und Wartung: Störfallmanagement	Keine	Keine	Keine	Betrieb und Wartung	Keine
Betrieb und Wartung: Code-Upgrades	Keine	Betrieb und Wartung	Keine	Keine	Keine
Betrieb und Wartung: Code-Optimierung	Betrieb und Wartung	Betrieb und Wartung	Keine	Keine	Keine
Betrieb und Wartung: Technisches Schuldenmanagement	Keine	Betrieb und Wartung	Betrieb und Wartung	Keine	Keine
Betrieb und Wartung: Change Management	Keine	Implementierung, Bereitstellung	Keine	Keine	Keine

Fähigkeit: Unterfähigkeit	Untersuchen	Integrieren	Interagieren	Iterieren	Auswirkung
Betrieb und Wartung: Reverse Engineering	Betrieb und Wartung	Keine	Keine	Keine	Keine
Betrieb und Wartung: Code-Mode rnisierung	Keine	Implement ierung	Keine	Keine	Keine
Betrieb und Wartung: Leistungs optimierung	Keine	Betrieb und Wartung	Keine	Betrieb und Wartung	Keine
Analysen und Einblicke	Keine	Anforderu ngen und Planung	Keine	Keine	Alle SDLC- Phasen
KI-Assistent	Keine	Keine	Alle SDLC- Phasen	Keine	Keine
Wissensma nagement	Keine	Keine	Alle SDLC- Phasen	Keine	Keine
Erweiterb arkeit	Keine	Bereitstellung	Keine	Keine	Keine

Generative KI-Anwendungsfälle für das Projektmanagement

Effektives Projektmanagement ist das Herzstück einer erfolgreichen Softwareentwicklung. Im Kontext der generativen KI nimmt das Projektmanagement neue Dimensionen an. Es kann prädiktiver, anpassungsfähiger und datengesteuerter werden. KI-gestützte Projektmanagement-Tools analysieren historische Projektdaten, um genauere Zeit- und Ressourcenschätzungen zu erstellen.

Sie können Aufgaben auf der Grundlage von Geschäftszielen und Teamkapazitäten automatisch priorisieren und sogar potenzielle Hindernisse vorhersagen, bevor sie auftreten. Ein Projektmanager könnte beispielsweise generative KI verwenden, um einen vorläufigen Projektplan zu erstellen, der auf den Projektanforderungen und historischen Daten aus ähnlichen Projekten basiert. Die KI könnte dann optimale Teamzusammensetzungen vorschlagen, die Fähigkeiten, Arbeitsbelastungen und Projektanforderungen berücksichtigen. Während des gesamten Projekts bieten KI-gesteuerte Dashboards nahezu in Echtzeit Einblicke in den Projektstatus, indem sie automatisch Berichte generieren und Bereiche hervorheben, die Aufmerksamkeit erfordern.

Dieser KI-gestützte Ansatz für das Projektmanagement kann die Effizienz steigern. Es hilft Projektmanagern, sich auf strategische Entscheidungen und Teamführung zu konzentrieren, anstatt sich in routinemäßigen Verwaltungsaufgaben zu verzetteln.

Die folgende Tabelle zeigt Anwendungsfälle für das Projektmanagement, die Sie mit generativer KI erweitern können, sowie die Person, die für diese Anwendungsfälle verantwortlich ist.

Unterfähigkeit: Anwendungsfall	Persona
Problemmanagement: Probleme erstellen und zuweisen	Projektmanager
Problemmanagement: Erkennen Sie Probleme beim Testen und protokollieren Sie sie	Testingenieur
Problemmanagement: Priorisieren Sie Probleme nach Schweregrad und weisen Sie sie Entwicklern zu	Projektmanager
Problemmanagement: Identifizieren Sie doppelte Probleme und führen Sie sie zusammen	Projektmanager
Problemmanagement: Verfolgen und erstellen Sie Berichte über wichtige Probleme, Kennzahlen und den allgemeinen Status des Projekts	Projektmanager
Sprint- und Aufgabenmanagement: Schätzen Sie den Aufwand für Aufgaben ab und	Scrum Master

Unterfähigkeit: Anwendungsfall	Persona
weisen Sie Storypoints auf der Grundlage der Teamkapazität zu	
Sprint- und Aufgabenmanagement: Verteilen Sie die Aufgaben auf die Teammitglieder, um eine gleichmäßige Arbeitsbelastung während des gesamten Sprints zu erreichen	Scrum Master
Sprint- und Aufgabenmanagement: Erleichtern Sie Sprint-Planungssitzungen, die die Teambemühungen an den Sprintzielen ausrichten	Scrum Master
Produkt-Backlog-Management: Ordnen Sie Backlog-Elemente auf der Grundlage von Geschäftswert, Dringlichkeit und Benutzerfeedback neu an	Eigentümer des Produkts
Verwaltung des Produkt-Backlogs: Integrieren Sie neues Kundenfeedback und Markteinblicke in den Produkt-Backlog, um Prioritäten nahezu in Echtzeit zu setzen	Eigentümer des Produkts
Produkt-Backlog-Management: Identifizieren und verwalten Sie Abhängigkeiten zwischen Backlog-Elementen, um die Entwicklung zu optimieren	Produktmanager
Zuordnung von User Stories: Erstellen Sie Karten der Nutzererlebnisse, um alle erforderlichen Funktionen und die entsprechenden User Stories zu identifizieren	Eigentümer des Produkts
Zuordnung von User Stories: Identifizieren Sie Lücken oder fehlende Schritte im Nutzerablauf	Geschäftsanalyst

Unterfähigkeit: Anwendungsfall	Persona
Zuordnung von User Stories: Priorisieren Sie User Stories auf der Grundlage ihrer Auswirkungen auf den Geschäftswert	Produktmanager
Berichterstattung und Analyse: Generieren Sie Dashboards nahezu in Echtzeit, die wichtige Projektkennzahlen wie Sprint-Geschwindigkeit und Problemlösungsraten visualisieren	Projektmanager
Berichterstattung und Analyse: Analysieren Sie historische Daten und prognostizieren Sie future Projektergebnisse, wie z. B. mögliche Verzögerungen oder Engpässe	Projektmanager
Berichterstattung und Analyse: Erstellen Sie benutzerdefinierte Berichte, z. B. Berichte zur Teamleistung oder zum Projektstatus, die auf verschiedene Stakeholder zugeschnitten sind	Projektmanager
Produkt-Roadmap-Management: Erstellen und pflegen Sie eine Produkt-Roadmap, in der die wichtigsten Meilensteine und Veröffentlichungstermine aufgeführt sind	Projektmanager
Produkt-Roadmap-Management: Aktualisieren Sie die Roadmap auf der Grundlage von Änderungen der Projektprioritäten oder Zeitpläne	Produktmanager
Verwaltung der Produkt-Roadmap: Teilen Sie die Roadmap den Stakeholdern mit, um einen Überblick über die Ausrichtung des Produkts zu erhalten	Produktmanager

Unterfähigkeit: Anwendungsfall	Persona
Feedback-Schleifen: Sammeln Sie nach jedem Sprint Feedback vom Team und identifizieren Sie Bereiche mit Verbesserungspotenzial	Scrum Master
Rückblicke: Setzen Sie Feedback in umsetzbare Elemente für den nächsten Sprint um und treiben Sie so die kontinuierliche Verbesserung voran	Scrum Master
Rückblicke: Verfolgen Sie die Auswirkungen von Änderungen, die im Rahmen früherer Retrospektiven vorgenommen wurden, um deren Wirksamkeit zu messen	Scrum Master

Generative KI-Anwendungsfälle für das Anforderungsmanagement

Das Anforderungsmanagement ist ein wichtiger Prozess, der eng mit dem Projektmanagement verknüpft ist. Stellen Sie sich vor, ein Product Owner verwendet ein KI-Tool, um Kundenfeedback, Markttrends und Beiträge von Stakeholdern zu analysieren. Das KI-Tool könnte eine umfassende Sammlung von Anwenderberichten und Anforderungen generieren, diese automatisch kategorisieren, potenzielle Konflikte oder Lücken erkennen und sogar eine Priorisierung vorschlagen, die auf dem Geschäftswert und der Komplexität der Implementierung basiert. Wenn das Projekt voranschreitet und sich die Anforderungen weiterentwickeln, kann die KI die Anforderungen kontinuierlich aktualisieren und verfeinern, um sicherzustellen, dass sie den sich ändernden Geschäftsanforderungen und technischen Einschränkungen entsprechen. Dieser dynamische, KI-gestützte Ansatz für das Anforderungsmanagement trägt dazu bei, dass die Entwicklungsbemühungen während des gesamten Projektlebenszyklus eng an den Benutzerbedürfnissen und Geschäftszielen ausgerichtet bleiben.

Die folgende Tabelle zeigt Anwendungsfälle für das Anforderungsmanagement, die Sie mit generativer KI erweitern können, sowie die Person, die für diese Anwendungsfälle verantwortlich ist.

Anwendungsfall	Persona
Erstellen Sie Geschäftsanforderungen	Geschäftsanalyst
Erstellen Sie Epen aus Features	Eigentümer des Produkts
Verfolge den Fortschritt eines Epos, indem du die Fertigstellung der zugehörigen User Stories überwachst	Produktmanager
Erstellen Sie Anwenderberichte	Eigentümer des Produkts
Schätzen Sie den Aufwand für jede Use Story ab und weisen Sie Story Points zu	Scrum Master
Definieren Sie Akzeptanzkriterien für jede User Story	Eigentümer des Produkts

Generative KI-Anwendungsfälle für Architektur und Design

Mit einer soliden Grundlage für Projektmanagement und klar definierten Anforderungen sind Architektur und Design die nächste wichtige Fähigkeit. Hier eröffnet generative KI neue Möglichkeiten zur Schaffung robuster, skalierbarer und effizienter Softwarearchitekturen. KI-gestützte Designtools können Anforderungen und Einschränkungen analysieren, um optimale Architekturmuster und Entwurfsansätze vorzuschlagen. Sie generieren mehrere Entwurfsalternativen, und jede ist für unterschiedliche Prioritäten wie Leistung, Skalierbarkeit oder Wartbarkeit optimiert. Ein Lösungsarchitekt könnte beispielsweise einen KI-Assistenten verwenden, um schnell mehrere grundlegende Architekturentwürfe auf der Grundlage der Projektanforderungen zu erstellen. Dieser KI-gestützte Ansatz beschleunigt den Entwurfsprozess und hilft Architekten, fundiertere Entscheidungen zu treffen. Dies führt zu robusteren und zukunftssichereren Softwaredesigns.

Die folgende Tabelle zeigt Anwendungsfälle für Architektur und Design, die Sie mit generativer KI erweitern können, sowie die Person, die für diese Anwendungsfälle verantwortlich ist.

Anwendungsfall	Persona
Erstellen Sie ein Architekturdokument	Lösungsarchitekt

Anwendungsfall	Persona
Erstellen Sie ein detailliertes Entwurfsdokument	Technischer Leiter
Verstehen Sie bestehende Architektur- und Designstandards	Lösungsarchitekt
Entwickeln Sie detaillierte Modelle und Prototypen einer Benutzeroberfläche	UX/UI-Designer

Generative KI-Anwendungsfälle für die Zusammenarbeit

Softwareentwicklung ist von Natur aus ein kollaboratives Unterfangen. Sie können generative KI verwenden, um die Zusammenarbeit in Ihrem Softwareentwicklungsteam zu verbessern. KI-gestützte Tools für die Zusammenarbeit gehen über einfaches Messaging und Filesharing hinaus. Sie ermöglichen eine effektivere Kommunikation, indem sie lange Diskussionsstränge zusammenfassen, wichtige Entscheidungen hervorheben und sogar optimale Zeiten für Besprechungen vorschlagen, die auf den Zeitplänen und Produktivitätsmustern der Teammitglieder basieren. KI kann bei Codeüberprüfungen helfen, indem sie potenzielle Probleme automatisch identifiziert, Verbesserungen vorschlägt und den Prüfern sogar komplexe Änderungen erklärt. Bei Brainstorming-Sitzungen kann KI als Moderator fungieren, Ideen generieren, bei der Organisation von Gedanken helfen und sogar Diskussionen vermitteln, um sicherzustellen, dass alle Stimmen gehört werden. Für verteilte Teams kann KI helfen, kulturelle und sprachliche Barrieren zu überbrücken. Sie kann fast in Echtzeit Sprachübersetzungen in Chat- und Videoanrufen ermöglichen und einen kulturellen Kontext bieten, um Missverständnissen vorzubeugen. Durch die Erweiterung der menschlichen Zusammenarbeit mit KI hilft diese Funktion Teams dabei, effizienter und effektiver zu arbeiten, was Innovationen fördert und die Gesamtergebnisse des Projekts verbessert.

Die folgende Tabelle zeigt, wie Sie generative KI verwenden können, um Anwendungsfälle für die Zusammenarbeit zu verbessern.

Unterfähigkeit: Anwendungsfall	Persona
Dokumentenmanagement: Erstellen und verwalten Sie ein zentrales Dokumentationsarchiv	Technischer Redakteur

Unterfähigkeit: Anwendungsfall	Persona
Dokumentenmanagement: Ermöglichen Sie mehreren Teammitgliedern, in Echtzeit gemeinsam an der Dokumentation zu arbeiten	Entwicklungsteam
Wissensaustausch: Nutzen Sie Diskussionsforen als Plattform für Entwickler, um Fragen zu stellen, Wissen auszutauschen und Probleme gemeinsam zu beheben	Entwicklungsteam
Wissensaustausch: Verwenden Sie Diskussionsforen, um Entscheidungen zu dokumentieren und nachzuverfolgen, die während der Projektdiskussionen getroffen wurden, und stellen Sie sicher, dass die Gründe für wichtige Entscheidungen erfasst und für future Nachschlagewerke zugänglich sind	Produktmanager
Projekt-Asset-Management: Erleichtern Sie die einfache gemeinsame Nutzung projektbezogener Ressourcen	Entwicklungsteam
Projekt-Asset-Management: Implementieren Sie die Versionskontrolle für gemeinsam genutzte Inhalte, sodass Teammitglieder Änderungen verfolgen, zu früheren Versionen zurückkehren und gemeinsam an Inhaltsaktualisierungen arbeiten können	Entwicklungsteam

Generative KI-Anwendungsfälle für DevSecOps

KI-gestützte DevSecOps Tools automatisieren viele Aspekte der Softwarebereitstellungspipeline. Sie können beispielsweise intelligente Codeprüfungen durchführen, potenzielle Fehler erkennen, Sicherheitslücken erkennen und Leistungsprobleme nahezu in Echtzeit identifizieren, während Entwickler Code schreiben. KI generiert und führt umfassende Testsuiten aus und aktualisiert sie automatisch, wenn sich die Codebasis weiterentwickelt. Dieser KI-gestützte Ansatz DevSecOps

beschleunigt die Bereitstellungspipeline und verbessert die Sicherheit und Zuverlässigkeit der bereitgestellten Software erheblich.

Die folgende Tabelle zeigt DevSecOps Anwendungsfälle, die Sie mit generativer KI erweitern können, sowie die Person, die für diese Anwendungsfälle verantwortlich ist.

Unterfähigkeit: Anwendungsfall	Persona
DevOps und kontinuierliche Bereitstellung: Automatisierte gesamte Bereitstellungspipeline	DevOps Ingenieur
DevOps und kontinuierliche Bereitstellung: Erhalten Sie nahezu in Echtzeit Feedback zur Codequalität und zu potenziellen Problemen	Softwareentwickler
DevOps und kontinuierliche Bereitstellung: Erhalten Sie nahezu in Echtzeit Sicherheitsprobleme und Empfehlungen zu deren Behebung	Softwareentwickler
DevOps und kontinuierliche Bereitstellung: Sie erhalten Code- und Best-Practice-Vorschläge fast in Echtzeit	Softwareentwickler
DevOps und kontinuierliche Bereitstellung: Automatisieren Sie sich wiederholende Aufgaben und integrieren Sie Befehle in Skripte	DevOps Ingenieur
DevOps und kontinuierliche Bereitstellung: Code erstellen und Artefakte nach jedem Code-Commit automatisch generieren	Softwareentwickler
DevOps und kontinuierliche Bereitstellung: Erstellen Sie Code gemäß den Standards und Rahmenbedingungen der Organisation	Softwareentwickler
DevOps und Continuous Delivery: Führen Sie bei jedem Commit automatisch Unit-Tests	Softwareentwickler

Unterfähigkeit: Anwendungsfall	Persona
durch catch um Fehler schon früh im Entwicklungsprozess zu erkennen	
DevOps und kontinuierliche Bereitstellung: Analysieren Sie den Umfang der Komponententests, um sicherzustellen, dass alle kritischen Codepfade getestet werden	Softwareentwickler
DevOps und kontinuierliche Bereitstellung: Filialen verwalten und Änderungen zusammenführen	Softwareentwickler
DevOps und kontinuierliche Bereitstellung: Verwalten Sie die Versionierung von Code und Artefakten	Softwareentwickler
DevOps und kontinuierliche Bereitstellung: Speichern und verwalten Sie Build-Artefakte und Abhängigkeiten	DevOps Ingenieur
DevOps und kontinuierliche Bereitstellung: Auflösen und Abrufen von Abhängigkeiten während des Build-Prozesses	Softwareentwickler
DevOps und kontinuierliche Bereitstellung: Generieren und führen Sie Integrationstests durch, um sicherzustellen, dass die Komponenten erwartungsgemäß zusammenarbeiten	Testingenieur
DevOps und kontinuierliche Bereitstellung: Verwenden Sie bei Integrationstests Scheindienste, um Interaktionen mit externen Systemen zu simulieren	Testingenieur

Unterfähigkeit: Anwendungsfall	Persona
DevOps und kontinuierliche Bereitstellung: Vergleichen Sie die Anwendungsleistung unter verschiedenen Lasten	Leistungsingenieur
DevOps und kontinuierliche Bereitstellung: Simulieren Sie Szenarien mit hohem Datenaufkommen, um die Skalierbarkeit und die Reaktionszeiten der Anwendung zu testen	Leistungsingenieur
DevOps und kontinuierliche Bereitstellung: Testen Sie die Fähigkeit des Systems, sich nach Ausfällen wie Serverabstürzen oder Netzwerkausfällen zu erholen	Techniker für Zuverlässigkeit vor Ort
DevOps und kontinuierliche Bereitstellung: Führen Sie Chaos Engineering durch	Ingenieur für Zuverlässigkeit vor Ort
DevOps und kontinuierliche Bereitstellung: Führen Sie Tests durch, um zu überprüfen, ob die Anwendung die Geschäftsanforderungen erfüllt	QA-Ingenieur
DevOps und kontinuierliche Bereitstellung: Führen Sie Benutzerakzeptanztests durch	Eigentümer des Produkts
DevOps und kontinuierliche Bereitstellung: Untersuchen Sie Abhängigkeiten auf Sicherheitslücken und Probleme mit der Einhaltung von Lizenzbestimmungen	Sicherheitsingenieur
DevOps und kontinuierliche Bereitstellung: Überwachen und verwalten Sie Open-Source-Abhängigkeiten, um sicherzustellen, dass sie aktuell und sicher sind	Sicherheitsingenieur

Unterfähigkeit: Anwendungsfall	Persona
DevOps und kontinuierliche Bereitstellung: Generieren und verwalten Sie eine Software-Stückliste (SBOM), um alle Komponenten und Abhängigkeiten nachzuverfolgen	Sicherheitsingenieur
DevOps und kontinuierliche Bereitstellung: Verwenden Sie die SBOM, um Audits zur Einhaltung gesetzlicher Vorschriften durchzuführen	Compliance-Beauftragter
DevOps und kontinuierliche Bereitstellung: Erstellen Sie Versionshinweise	Versionsmanager
DevOps und kontinuierliche Bereitstellung: Planen und koordinieren Sie Releases	Versionsmanager
DevOps und kontinuierliche Bereitstellung: Implementieren Sie Standardarbeitsanweisungen für Rollback und Release-Management	Versionsmanager
DevOps und kontinuierliche Bereitstellung: Verwenden Sie Feature-Flags, um Funktionen in der Produktion zu aktivieren oder zu deaktivieren, ohne neuen Code bereitzustellen	Produktmanager
DevOps und kontinuierliche Bereitstellung: Führen Sie A/B Tests mithilfe von Feature-Flags durch, um die Auswirkungen verschiedener Funktionen auf das Benutzerverhalten zu messen	Produktmanager
DevOps und kontinuierliche Bereitstellung: Analysieren und überwachen Sie Pipeline-Ausfälle	DevOps Ingenieur

Unterfähigkeit: Anwendungsfall	Persona
DevOps und kontinuierliche Bereitstellung: Infrastrukturressourcen erstellen und verwalten	DevOps Ingenieur
DevOps und Sicherheit: Code-Repositorys nach fest codierten Geheimnissen scannen	DevOps Ingenieur
DevOps und Sicherheit: Implementieren Sie eine Erkennung nahezu in Echtzeit, um Entwickler sofort zu benachrichtigen, wenn geheime Daten in das Repository übertragen werden	DevOps Ingenieur
DevOps und Sicherheit: Erzwingen Sie eine kontinuierliche Überwachung der Codequalität	Softwareentwickler
DevOps und Sicherheit: Erkennen und kennzeichnen Sie Indikatoren für potenzielle Sicherheitslücken im Code	Softwareentwickler
DevOps und Sicherheit: Implementieren Sie automatisierte Tests für die 10 wichtigsten Sicherheitsrisiken des Open Worldwide Application Security Project (OWASP), um sicherzustellen, dass die Anwendung den branchenüblichen Sicherheitspraktiken entspricht	Sicherheitsingenieur
DevOps und Sicherheit: Informiere Entwickler regelmäßig und informiere sie über OWASP-Risiken, indem du Prüfungen in den Entwicklungsprozess integrierst	Sicherheitsingenieur
DevOps und Sicherheit: Durchsuchen Sie Bibliotheken und Abhängigkeiten von Drittanbietern auf bekannte Sicherheitslücken	DevOps Ingenieur

Unterfähigkeit: Anwendungsfall	Persona
DevOps und Sicherheit: Scannen Sie den Anwendungscode und die Infrastruktur, um Sicherheitslücken zu erkennen	DevOps Ingenieur
DevOps und Sicherheit: Analysieren Sie den Code vor der Bereitstellung auf Sicherheitslücken	Sicherheitsingenieur
DevOps und Sicherheit: Setzen Sie Sicherheitsrichtlinien durch, indem Sie verhindern, dass Code mit kritischen Sicherheitslücken zusammengeführt wird	Sicherheitsingenieur
DevOps und Sicherheit: Implementieren Sie eine rollenbasierte Zugriffskontrolle (RBAC), um den Zugriff auf sensible Systeme und Daten einzuschränken und sicherzustellen, dass nur autorisiertes Personal auf kritische Ressourcen zugreifen kann	Sicherheitsingenieur
DevOps und Sicherheit: Passen Sie die Zugriffskontrollen an die Rollen und Verantwortlichkeiten an, indem Sie sie an Änderungen in der Teamstruktur anpassen	DevOps Ingenieur
DevOps und Sicherheit: Testen Sie laufende Anwendungen nahezu in Echtzeit auf Sicherheitslücken, indem Sie Angriffe auf die Produktionsumgebung simulieren	Sicherheitsingenieur
DevOps und Sicherheit: Überwachen Sie bereitgestellte Anwendungen kontinuierlich auf Sicherheitslücken	DevOps Ingenieur

Unterfähigkeit: Anwendungsfall	Persona
DevOps und Sicherheit: Planen Sie regelmäßige Schwachstellenscans in allen Umgebungen ein, um Sicherheitslücken zu identifizieren und zu beheben	Sicherheitsingenieur
DevOps und Sicherheit: Wenden Sie Patches und Updates auf der Grundlage der Ergebnisse von Schwachstellenscans an, um die Sicherheit Ihrer Systeme aufrechtzuerhalten	DevOps Ingenieur
Überwachung der Anwendungsleistung: Kontinuierliche Überwachung der Anwendungsleistung nahezu in Echtzeit, um Leistungsprobleme zu erkennen und zu diagnostizieren, bevor sie sich auf Benutzer auswirken	Ingenieur für Zuverlässigkeit vor Ort
Überwachung der Anwendungsleistung: Erkennen Sie Leistungsanomalien, wie z. B. plötzliche Spitzen bei den Antwortzeiten oder erhöhte Fehlerraten, und lösen Sie Warnmeldungen aus	DevOps Ingenieur
Überwachung der Anwendungsleistung: Verfolgen Sie Anfragen, während sie sich in einem verteilten System ausbreiten, um Leistungsentpässe und Latenzprobleme zu identifizieren	DevOps Ingenieur
Überwachung der Anwendungsleistung: Verwenden Sie verteiltes Tracing, um genau den Dienst oder die Komponente zu ermitteln, die für Ausfälle oder Leistungseinbußen verantwortlich ist	DevOps Ingenieur

Unterfähigkeit: Anwendungsfall	Persona
Aggregation und Analyse von Protokollen: Aggregieren Sie Protokolle aus mehreren Quellen in einem zentralen System, um sie einfach durchsuchen und analysieren zu können, um Trends und Probleme zu identifizieren	Ingenieur für Zuverlässigkeit vor Ort
Aggregation und Analyse von Protokollen: Implementieren Sie eine automatisierte Protokollanalyse, um relevante Informationen zu extrahieren und Muster oder Anomalien zu erkennen, die auf Probleme hinweisen könnten	DevOps Ingenieur
Aggregation und Analyse von Protokollen: Erfassung und Visualisierung wichtiger Leistungskennzahlen	Ingenieur für Standortzuverlässigkeit
Aggregation und Analyse von Protokollen: Überwachen Sie Metriken anhand vordefinierter Service Level Agreements () SLAs	Produktmanager
KI-Operationen: Erkennen Sie Vorfälle, analysieren Sie die Ursachen und leiten Sie Korrekturmaßnahmen ein, ohne dass ein menschliches Eingreifen erforderlich ist	DevOps Ingenieur
KI-Betrieb: Prognostizieren Sie den future Ressourcenbedarf und optimieren Sie die Kapazitätsplanung, um Ausfälle zu vermeiden	Ingenieur für Zuverlässigkeit vor Ort
Kontinuierliche Verbesserung: Überwachen Sie die tatsächlichen Benutzerinteraktionen mit der Anwendung, um Erkenntnisse über die Leistung zu gewinnen und Bereiche mit Verbesserungspotenzial zu identifizieren	UX-Designer

Unterfähigkeit: Anwendungsfall	Persona
Kontinuierliche Verbesserung: Verfolgen Sie die Anwendungsleistung in verschiedenen geografischen Regionen, um weltweit ein einheitliches Benutzererlebnis zu gewährleisten	Produktmanager
Dashboard-Überwachung: Erstellen Sie anpassbare Dashboards, um wichtige Kennzahlen, Protokolle und Traces nahezu in Echtzeit zu visualisieren und so einen umfassenden Überblick über den Systemzustand zu erhalten	Ingenieur für Zuverlässigkeit vor Ort
Dashboard-Überwachung: Erstellen Sie Dashboards für verschiedene Teams (z. B. Entwicklungs-, Betriebs- und Produktteams), um relevante Einblicke auf der Grundlage ihrer Schwerpunktbereiche zu erhalten	DevOps Ingenieur
Einblicke in die Leistung: Führen Sie eine detaillierte Analyse der Anwendungsleistung durch, um Ineffizienzen zu identifizieren und Code oder Infrastruktur zu optimieren	Softwareentwickler
Einblicke in die Leistung: Nutzen Sie Leistungseinblicke, um die Anwendungsleistung schrittweise zu verbessern und die Benutzererfahrung im Laufe der Zeit zu optimieren	Produktmanager

Generative KI-Anwendungsfälle für Betrieb und Wartung

Nach der Bereitstellung der Software verlagert sich der Schwerpunkt auf Betrieb und Wartung. Generative KI kann traditionelle Ansätze verbessern, indem sie ein proaktiveres und effizienteres Systemmanagement ermöglicht. KI-gestützte Betriebstools überwachen kontinuierlich die Systemleistung und prognostizieren potenzielle Probleme, bevor sie sich auf Benutzer auswirken. Sie führen automatische Ursachenanalysen durch, wenn Probleme auftreten, wodurch die

durchschnittliche Zeit bis zur Lösung erheblich verkürzt wird. KI optimiert auch die Systemleistung nahezu in Echtzeit. Sie passt Konfigurationen automatisch an sich ändernde Lastmuster und Benutzerverhalten an. Beispielsweise könnte ein Betriebsteam einen KI-Assistenten verwenden, um vorausschauende Wartungspläne zu erstellen, automatisch Komponenten zu identifizieren, bei denen ein Ausfall wahrscheinlich ist, und präventive Maßnahmen vorzuschlagen. Die KI könnte auch bei der Kapazitätsplanung helfen, indem sie Nutzungstrends analysiert und den future Ressourcenbedarf mit hoher Genauigkeit vorhersagt.

Die folgende Tabelle zeigt Anwendungsfälle für Betrieb und Wartung, die Sie mit generativer KI verbessern können, sowie die Person, die für diese Anwendungsfälle verantwortlich ist.

Unterfähigkeit: Anwendungsfall	Persona
Incident Management: Managen Sie Vorfälle nahezu in Echtzeit, indem Sie Überwachungstools in Chat-Plattformen integrieren, sodass Teams Probleme direkt in der Chat-Umgebung erkennen, besprechen und lösen können	Ingenieur für Zuverlässigkeit vor Ort
Vorfallmanagement: Ermöglichen Sie es Teams, Bereitstellungen zu initiieren, Skripte auszuführen und Befehle direkt von der Chat-Oberfläche aus auszuführen, was den Betrieb rationalisiert	DevOps Ingenieur
Code-Upgrades: Aktualisieren Sie Codeabhängigkeiten und Bibliotheken, um den manuellen Aufwand zu reduzieren und sicherzustellen, dass die Codebasis mit den neuesten Versionen auf dem neuesten Stand bleibt	Softwareentwickler
Code-Optimierung: Überprüfen Sie den Code auf Optimierungsmöglichkeiten	Softwareentwickler
Codeoptimierung: Identifizieren Sie Engpässe im Code und refaktorisieren oder optimieren Sie den Code, um die Leistung zu verbessern	Softwareentwickler

Unterfähigkeit: Anwendungsfall	Persona
Technisches Schuldenmanagement: Protokollieren Sie technische Schulden als Teil des Entwicklungsprozesses	Produktmanager
Technisches Schuldenmanagement: Priorisieren und behandeln Sie technische Schulden auf der Grundlage von Auswirkungen, Risiken und Kosten und integrieren Sie sie in den regulären Sprint-Planungsprozess	Softwareentwickler
Technisches Schuldenmanagement: Reduzieren Sie technische Schulden im vorhandenen Anwendungscode	Softwareentwickler
Änderungsmanagement: Implementieren Sie einen Prozess zur Genehmigung von Änderungen, der sicherstellt, dass alle Codeänderungen vor der Implementierung von den erforderlichen Beteiligten überprüft, getestet und genehmigt werden	Manager ändern
Änderungsmanagement: Führen Sie eine Folgenabschätzung der vorgeschlagenen Änderungen durch	DevOps Ingenieur
Reverse Engineering: Analysieren und verstehen Sie die Struktur und das Verhalten von Legacy-Code	Lösungsarchitekt
Reverse Engineering: Erläutern Sie den vorhandenen Code und erstellen Sie eine Dokumentation	Softwareentwickler
Code-Modernisierung: Translate Sie Code von einer Programmiersprache in eine andere	Softwareentwickler

Unterfähigkeit: Anwendungsfall	Persona
Code-Modernisierung: Modernisieren Sie älteren Code in die neueste Programmiersprache	Softwareentwickler
Leistungsoptimierung: Überwachen und optimieren Sie kontinuierlich die Systemleistung, indem Sie die Ressourcenzuweisung, den Lastenausgleich und die Neukonfiguration der Anwendung optimieren	Ingenieur für Zuverlässigkeit vor Ort
Leistungsoptimierung: Identifizieren und überarbeiten Sie den Code, der zu Leistungseinbußen führt, um die Geschwindigkeit und die Reaktionsfähigkeit des Systems zu verbessern	Softwareentwickler

Anwendungsfälle für generative KI-Assistenten in der Softwareentwicklung

Die KI-Assistentenfunktion ist das Herzstück der generativen KI-gestützten Entwicklungserfahrung. Dieses intelligente, kontextsensitive System dient als virtueller Kollaborateur für alle Teammitglieder im gesamten SDLC. Stellen Sie sich einen Entwickler vor, der an einem komplexen Code arbeitet. Sie können den KI-Assistenten einfach um Hilfe bitten und er kann relevante Codefragmente bereitstellen, komplizierte Algorithmen erklären oder sogar Optimierungen vorschlagen, die auf dem aktuellen Kontext und den Best Practices basieren. Der KI-Assistent kann einem ITOps Manager helfen, eine Standardarbeitsanweisung zu verstehen, die auf internen Dokumenten basiert. Durch die sofortige, kontextuelle Unterstützung reduzieren KI-Assistenten die kognitive Belastung der Teammitglieder erheblich. Dies hilft ihnen, sich auf übergeordnete Problemlösungs- und kreative Aufgaben zu konzentrieren. Diese Fähigkeit wirkt wie ein Multiplikator, der die Produktivität und Qualität in allen Phasen der Softwareentwicklung steigert.

Die folgende Tabelle zeigt Anwendungsfälle, die Sie mit KI-Assistenten und der Benefited Persona verbessern können.

Anwendungsfall	Persona
Bieten Sie dem Entwicklungsteam sofortige Unterstützung, indem Sie Fragen zu Anforderungen, Architekturen und Standardarbeitsanweisungen beantworten	Softwareentwicklungsteam
Suchen oder rufen Sie Auszüge aus umfangreicher Dokumentation ab oder generieren Sie Zusammenfassungen mithilfe von Abfragen in natürlicher Sprache	Softwareentwicklungsteam
Fassen Sie umfangreiche technische Dokumente wie Anforderungsdokumente, Architekturentwurfsdokumentationen und interne Prozesse zusammen	Softwareentwicklungsteam
Pflegen Sie eine Bibliothek mit Eingabeaufforderungen, die das Team für allgemeine Aufgaben verwenden kann	Softwareentwicklungsteam
Integrieren Sie generative KI nahtlos in bestehende Tools und Systeme	Softwareentwicklungsteam
Automatisieren Sie Aufgaben auf verschiedenen Plattformen, Tools und internen Systemen	Softwareentwicklungsteam
Schaffen Sie einen zentralen Wissensspeicher, der für alle Teammitglieder zugänglich ist, einschließlich bewährter Verfahren, projektspezifischer Informationen und Teamwissen	Softwareentwicklungsteam
Rufen Sie je nach Kontext der Aufgabe relevantes Wissen aus dem Repository ab	Softwareentwicklungsteam
Führen Sie automatisierte Codeprüfungen und Ursachenanalysen durch, schlagen Sie Verbesserungen vor, erkennen Sie potenziell	Softwareentwickler, DevOps Ingenieur und Techniker für die Zuverlässigkeit von Websites

Anwendungsfall	Persona
le Fehler und führen Sie Problembehebungen durch	
Analysieren Sie Leistungsdaten, um Trends und Muster zu identifizieren, die als Grundlage für Entscheidungen zur Leistungsoptimierung dienen können	Ingenieur für Standortzuverlässigkeit
Geben Sie Empfehlungen zur Verbesserung der Effizienz, zur Verringerung der Komplexität und zur Verbesserung der Sicherheit	Softwareentwickler
Schlagen Sie Optimierungen für die Nutzung von Cloud-Ressourcen vor, z. B. Skalierungsempfehlungen oder Strategien zur Kosteneinsparung	Softwareentwickler, DevOps Ingenieur, Ingenieur für Standortzuverlässigkeit und Lösungsarchitekt
Generieren Sie neue Inhalte, wie z. B. Dokumentation, die auf Code, Benutzerhandbüchern oder Produktversionen basiert	Softwareentwicklungsteam

Generative KI-Anwendungsfälle für Analysen und Erkenntnisse

Die Analyse- und Erkenntnisfunktionen helfen dabei, riesige Datenmengen in umsetzbare Erkenntnisse umzuwandeln, die die Entscheidungsfindung und kontinuierliche Verbesserung fördern. Durch den Einsatz generativer KI verarbeitet diese Funktion Daten aus verschiedenen Quellen, darunter Code-Repositorys, Projektmanagement-Tools und Plattformen für die Teamzusammenarbeit, um einen ganzheitlichen Überblick über den Entwicklungsprozess und die Teamproduktivität zu erhalten. Generative KI geht über herkömmliche Kennzahlen hinaus und bietet prädiktive und präskriptive Analysen. Sie kann potenzielle Probleme vorhersagen und gezielte Verbesserungen vorschlagen. Es kann beispielsweise Muster bei Code-Commits, Fehlerbehebungsrate und die Geschwindigkeit der Bereitstellung von Funktionen analysieren, um leistungsstarke Teams zu identifizieren, Engpässe zu lokalisieren und Prozessoptimierungen vorzuschlagen. Darüber hinaus kann es Einblicke in die Teamdynamik und die individuelle Leistung geben. Diese Erkenntnisse helfen Führungskräften, datengestützte Entscheidungen über die

Verteilung der Arbeitslast, den Schulungsbedarf und die Teamzusammensetzung zu treffen. Durch die Präsentation dieser Erkenntnisse in interaktiven Dashboards können Stakeholder auf allen Ebenen fundierte Entscheidungen treffen, Prozesse optimieren und die Teamproduktivität kontinuierlich steigern, was zu einer schnelleren Bereitstellung hochwertiger Software führt.

Die folgende Tabelle zeigt Anwendungsfälle für Analysen, die Sie mit generativer KI verbessern können, sowie die Person, die für diese Anwendungsfälle verantwortlich ist.

Anwendungsfall	Persona
Überwachen Sie die Produktivität von Einzelpersonen und Teams	Entwicklungsleiter
Analysieren Sie Produktivitätstrends, um potenziellen Burnout zu erkennen, sodass Sie proaktive Maßnahmen ergreifen können, um das Wohlbefinden und die Produktivität Ihres Teams aufrechtzuerhalten	Entwicklungsleiter
Verfolgen Sie, wie oft Codeänderungen in der Produktion implementiert werden, um die Geschwindigkeit und Agilität des Entwicklungsprozesses einzuschätzen	Produktmanager
Analysieren Sie Daten zur Bereitstellungshäufigkeit, um Zeiträume mit geringer Bereitstellungsaktivität zu identifizieren, die auf Prozessineffizienzen oder Ressourcenbeschränkungen hinweisen könnten	Produktmanager
Messen Sie die Zeit zwischen der Code-Commit und der Implementierung, um Möglichkeiten zur Rationalisierung der Entwicklungs- und Bereitstellungsprozesse zu ermitteln	Leiter der Entwicklung
Verfolgen Sie den Prozentsatz der Bereitstellungen, die zu Ausfällen führen, die sofort	Ingenieur für Zuverlässigkeit vor Ort

Anwendungsfall	Persona
<p>behoben werden müssen, um die Zuverlässigkeit des Release-Prozesses zu beurteilen</p>	
<p>Identifizieren Sie anhand von Kennzahlen zur Change-Failure-Rate Codebereiche, in denen häufig Probleme auftreten, und leiten Sie so gezielte Refactoring- und Testmaßnahmen ein</p>	Softwareentwickler
<p>Überwachen Sie, wie lange es dauert, den Service nach einem Ausfall oder Vorfall wiederherzustellen, um Ausfallzeiten zu reduzieren und die allgemeine Systemstabilität zu verbessern</p>	Ingenieur für Zuverlässigkeit vor Ort
<p>Analysieren Sie Trends bei den Wiederherstellungszeiten, um die Reaktionsprozesse bei Vorfällen zu verbessern und die Wiederherstellung nach Systemausfällen zu beschleunigen</p>	DevOps Ingenieur
<p>Erstellen Sie ein maßgeschneidertes Dashboard, das wichtige Kennzahlen wie Bereitstellungshäufigkeit, Vorlaufzeit und Fehlerquote bei Änderungen zusammenfasst, um einen umfassenden Überblick über die Entwicklung und den Betriebsstatus zu erhalten</p>	Produktmanager
<p>Erstellen Sie Dashboards, die auf die Bedürfnisse verschiedener Teams zugeschnitten sind, um gezielte Einblicke in ihre jeweiligen Verantwortungsbereiche wie Entwicklung, Betrieb oder Geschäft zu bieten</p>	Produktmanager

Anwendungsfall	Persona
Verfolgen Sie wichtige Leistungsindikatoren (KPIs) des Unternehmens, wie Umsatzauswirkungen, Kundenzufriedenheit und Marktanteil, um die Entwicklungsbemühungen auf umfassendere Geschäftsziele auszurichten	Produktmanager
Analysieren Sie die Auswirkungen neuer Funktionen auf das Geschäft KPIs, um deren Erfolg zu beurteilen und die future Produktentwicklung zu steuern	Geschäftsanalyst
Überwachen Sie Kennzahlen zur Codequalität wie Codekomplexität, Testabdeckung und Fehlerdichte, um sicherzustellen, dass die Codebasis wartbar und sicher bleibt	Softwareentwickler
Identifizieren Sie Bereiche der Codebasis, die überarbeitet werden müssen, um die langfristige Nachhaltigkeit zu fördern und technische Schulden zu reduzieren	Lösungsarchitekt

Generative KI-Anwendungsfälle für Wissensmanagement

In jeder Softwareentwicklungsorganisation ist Wissen ein entscheidendes Gut. Die Wissensmanagement-Funktion, die auf generativer KI basiert, verbessert die Art und Weise, wie diese Ressource erfasst, organisiert und genutzt wird. Herkömmliche Wissensmanagementsysteme enthalten oft zu viele Informationen, enthalten veraltete Inhalte oder sind schwer zu durchsuchen, um schnell relevante Informationen zu finden.

Generative KI geht diese Herausforderungen direkt an. Sie generiert und aktualisiert automatisch die Dokumentation auf der Grundlage von Codeänderungen, Konversationen und Projektartefakten. Dadurch wird sichergestellt, dass die Wissensdatenbanken auf dem neuesten Stand bleiben, ohne dass die Teammitglieder manuelle Anstrengungen unternehmen müssen. Noch wichtiger ist, dass KI dieses Wissen auf intuitive Weise zugänglich macht. Teammitglieder können Fragen in natürlicher Sprache stellen, und die KI kann relevante Antworten geben. Die KI kann aus

einer Vielzahl von Quellen schöpfen, z. B. aus offizieller Dokumentation, Codekommentaren, Diskussionsfäden und sogar externen Ressourcen. Beispielsweise könnte ein neues Teammitglied, das versucht, eine bestimmte Komponente zu verstehen, die KI fragen: „Wie funktioniert das Authentifizierungsmodul?“ Die KI würde dann eine kurze Erklärung und Links zu relevanten Codeabschnitten, Architekturdiagrammen und aktuellen Änderungen bereitstellen. Sie könnte diese Informationen sogar auf die Rolle und das Fachwissen des Teammitglieds zuschneiden.

Diese Funktion beschleunigt das Onboarding, reduziert sich wiederholende Fragen und fördert den Wissensaustausch im gesamten Unternehmen. Sie trägt dazu bei, institutionelles Wissen zu bewahren, und erleichtert es Teams, komplexe Systeme im Laufe der Zeit zu pflegen und weiterzuentwickeln.

Die folgende Tabelle zeigt Anwendungsfälle für Wissensmanagement, die Sie mit generativer KI erweitern können, sowie die Person, die für diese Anwendungsfälle verantwortlich ist.

Anwendungsfall	Persona
Schaffen Sie eine einheitliche Plattform, die den Zugriff auf das gesamte projektbezogene Wissen erleichtert	Softwareentwicklungsteam
Erfassen Sie Wissen aus verschiedenen Entwicklungsaktivitäten	Softwareentwicklungsteam
Stellen Sie erweiterte Suchfunktionen bereit, um schnell relevantes Wissen in einem Repository zu finden	Softwareentwicklungsteam
Personalize Sie Lernmodule und Pfade für das Team	Softwareentwicklungsteam

Generative KI-Anwendungsfälle für Erweiterbarkeit

Die Erweiterbarkeit ermöglicht eine nahtlose Integration in bestehende Tools und Workflows und ermöglicht es Unternehmen gleichzeitig, das KI-System an ihre spezifischen Bedürfnisse anzupassen. Diese Funktion bietet robuste und anpassbare Schnittstellen APIs SDKs, die die Integration von KI-Funktionen in beliebige Entwicklungs- und Projektmanagement-Tools erleichtern.

Unternehmen können Jira beispielsweise um KI-gestützte Funktionen für die automatisierte Ticketpriorisierung, Aufwandsschätzung und Sprint-Planung erweitern. Sie können die Jenkins-Pipelines mit KI für intelligente Build-Optimierung und prädiktive Testauswahl erweitern.

Darüber hinaus ermöglicht die Erweiterbarkeit eine tiefe Integration mit integrierten Entwicklungsumgebungen (IDEs), Versionskontrollsystemen und Code-Review-Plattformen. Die KI kann beim Programmieren helfen, Codeüberprüfungen automatisieren und kontextbezogene Dokumentation generieren.

Die Funktion unterstützt auch das Training und die Feinabstimmung von KI-Modellen anhand unternehmensspezifischer Daten. Dies hilft der KI, unternehmensspezifische Codierungsmuster, architektonische Präferenzen und Fachwissen zu verstehen. Das Ergebnis ist eine relevantere und kontextsensivere Unterstützung für alle integrierten Tools. Durch dieses Maß an Flexibilität und Integration stellt die Erweiterbarkeit sicher, dass sich die KI-gestützte Entwicklungserfahrung mit der Organisation weiterentwickelt. Es kann sich an sich ändernde Technologien und Geschäftsanforderungen anpassen und gleichzeitig bestehende Toolchains und Workflows nahtlos verbessern.

Die folgende Tabelle zeigt Anwendungsfälle zur Erweiterbarkeit, die Sie mit generativer KI verbessern können, sowie die Person, die für diese Anwendungsfälle verantwortlich ist.

Anwendungsfall	Persona
Integrieren Sie Tools von Drittanbietern in die Entwicklungsumgebung	DevOps Ingenieur
Erstellen Sie benutzerdefinierte Automatisierungsworkflows, die auf den einzigartigen Entwicklungsprozess des Teams zugeschnitten sind	DevOps Ingenieur
Connect zu verschiedenen APIs Diensten her	DevOps Ingenieur
Erstellen Sie Konnektoren für plattformübergreifende Tools	DevOps Ingenieur

Bewährte Methoden für den Einsatz generativer KI in der Softwareentwicklung

In diesem Abschnitt werden bewährte Methoden für die Integration generativer KI in den Softwareentwicklungszyklus (SDLC) beschrieben. Von der Implementierung nahtloser Toolchains und DevSecOps Pipelines bis hin zur Förderung der Zusammenarbeit und Automatisierung sich wiederholender Aufgaben helfen Ihnen diese Richtlinien dabei, das Potenzial der KI zur Verbesserung Ihrer Entwicklungsprozesse und -erfahrungen zu nutzen. Durch die Befolgung dieser Best Practices können Softwareentwicklungsteams ein neues Maß an Effizienz, Innovation und Qualität ihrer Arbeit erreichen.

In diesem Abschnitt werden die folgenden bewährten Methoden erörtert:

- [Implementierung einer nahtlosen, end-to-end integrierten Toolchain](#)
- [Implementierung einer end-to-end CI/CD Pipeline für DevSecOps](#)
- [Einführung von Tools und Praktiken für die Zusammenarbeit](#)
- [Automatisierung sich wiederholender Aufgaben](#)
- [Regelmäßige Überprüfung und Weiterentwicklung der Entwicklungserfahrungen](#)
- [Einführung effektiver Projektmanagementpraktiken](#)
- [Implementierung von Wissensmanagement](#)
- [Bereitstellung von Erweiterbarkeit und Anpassung](#)
- [Optimierung für den Betrieb](#)
- [Nutzung datengestützter Erkenntnisse](#)
- [Einführung eines plattformbasierten Ansatzes](#)

Implementierung einer nahtlosen, end-to-end integrierten Toolchain

Die Implementierung einer nahtlosen, end-to-end integrierten Toolchain ist eine grundlegende bewährte Methode für die Schaffung eines generativen KI-gestützten Entwicklungserlebnisses. Die Kernidee besteht darin, ein zusammenhängendes Ökosystem von Tools und Plattformen aufzubauen, das Ihre Softwareteams im gesamten SDLC nutzen können. Das Team kann die Toolchain verwenden, um den laufenden Betrieb zu planen, zu entwickeln, zu entwickeln, zu testen, bereitzustellen und zu verwalten. Durch die Integration generativer KI-Funktionen in diese

Toolchain stellen Sie sicher, dass KI-Unterstützung in jeder Phase verfügbar ist. Diese Integration reduziert oder eliminiert manuelle Übergaben, reduziert den Kontextwechsel und trägt dazu bei, dass Daten und Artefakte reibungslos zwischen verschiedenen Entwicklungsphasen fließen können. Beispielsweise können KI-generierte Codefragmente aus Ihrer integrierten Entwicklungsumgebung (IDE) nahtlos in Ihr Versionskontrollsystem einfließen, und KI-gestützte Analysen aus Ihrer Bereitstellungsplattform können als Grundlage für Ihre Projektmanagement-Tools dienen. Dadurch entsteht eine kontinuierliche Feedback-Schleife, die Ihren Entwicklungsprozess verbessert.

Implementierung einer end-to-end CI/CD Pipeline für DevSecOps

Um auf dieser integrierten Toolchain aufzubauen und eine end-to-end kontinuierliche Integration und Bereitstellung (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 Pipeline) zu implementieren, können Sie auch generative KI verwenden, um Sicherheitsrichtlinien auf der Grundlage der neuesten Bedrohungsinformationen automatisch zu aktualisieren.

Einführung von Tools und Praktiken für die Zusammenarbeit

Vergessen Sie bei der Verbesserung Ihrer Entwicklungsinfrastruktur nicht die menschliche Komponente. Softwareentwicklung ist von Natur aus ein gemeinschaftliches Unterfangen. Es umfasst funktionsübergreifende Teams, die sich aus Entwicklern, Designern, Produktmanagern, Scrum Mastern, Geschäftsanalysten und anderen Interessengruppen zusammensetzen. Diese Personen arbeiten gemeinsam daran, Ideen zu verwirklichen. Durch den Einsatz moderner Tools für die Zusammenarbeit und die Förderung einer Kultur der offenen Kommunikation und des Wissensaustauschs können Sie die Produktivität und Effektivität Ihrer Softwareentwicklungsteams erheblich steigern. In Ihrer Erfahrung mit KI-gestützter Softwareentwicklung nehmen diese Tools neue Dimensionen an. Sie können KI in Kollaborationsplattformen integrieren, um eine effektivere Kommunikation und einen effektiveren Wissensaustausch zwischen den Teammitgliedern zu ermöglichen. KI-Assistenten können häufig gestellte Fragen beantworten, Diskussionen zusammenfassen oder sogar Konflikte schlichten. Generative KI kann Prozesse zur Codeüberprüfung verbessern, indem sie automatisch Verbesserungen vorschlägt oder

potenzielle Probleme identifiziert. Darüber hinaus können Sie KI verwenden, um dynamische, kontextsensitive Dokumentation zu erstellen, die nahezu in Echtzeit aktualisiert wird, wenn sich das Projekt weiterentwickelt, sodass alle Teammitglieder Zugriff auf die aktuellsten und relevantesten Informationen haben.

Automatisierung sich wiederholender Aufgaben

Durch den Einsatz generativer KI für routinemäßige, zeitaufwändige Aktivitäten können sich Ihre Softwareteams auf hochwertige, kreative Arbeit konzentrieren, die Innovationen vorantreibt und geschäftliche Auswirkungen hat. Beispiele für sich wiederholende Aufgaben sind die Generierung von Standardcode, die Erstellung von Testdaten, das Schreiben von Dokumentation oder sogar die Erstellung erster Projektpläne. Durch die Auslagerung dieser Aufgaben an KI können sich die Teammitglieder auf kreativere und strategischere Arbeiten konzentrieren. KI-gestützte Tools zur Codevervollständigung können beispielsweise den Codierungsprozess erheblich beschleunigen, indem sie relevante Codefragmente auf der Grundlage von Kontext und Codierungsmustern vorschlagen. In ähnlicher Weise kann generative KI automatisch technische Dokumentationen erstellen und aktualisieren, wenn sich der Code ändert. Dadurch bleibt die Dokumentation auf dem neuesten Stand und der manuelle Aufwand, der normalerweise für diese Aufgabe erforderlich ist, wird reduziert. Beim Testen kann KI auf der Grundlage von Anforderungen und Codeanalysen umfassende Testfälle generieren, wodurch die Testabdeckung verbessert und die Wahrscheinlichkeit von übersehenen Grenzfällen verringert wird. Durch die intelligente Automatisierung dieser sich wiederholenden Aufgaben beschleunigt die generative KI die Entwicklungszeiten, verbessert die Konsistenz und reduziert menschliche Fehler. Das Ergebnis sind qualitativ hochwertigere Softwareausgaben.

Regelmäßige Überprüfung und Weiterentwicklung der Entwicklungserfahrungen

Ihre Erfahrung in der Softwareentwicklung selbst sollte als ein Produkt behandelt werden, das ständig weiterentwickelt werden muss. Dazu gehört die Einrichtung eines systematischen Prozesses zur regelmäßigen Überprüfung und Iteration aller Aspekte des Entwicklungslebenszyklus, der Tools und Verfahren. Führen Sie regelmäßige Bewertungen der gesamten Toolchain, der Workflows und Prozesse durch. Sammeln Sie Feedback von allen Teammitgliedern aus verschiedenen Rollen, einschließlich Produktmanagern, Designern, Architekten, Entwicklern, Testern und Betriebspersonal. Bitten Sie sie, Schwachstellen, Engpässe und Verbesserungsmöglichkeiten zu identifizieren. Teams könnten beispielsweise vierteljährliche Überprüfungen der Leistung ihrer CI/CD Pipeline durchführen

und Kennzahlen wie Erstellungszeiten, Bereitstellungshäufigkeit und Fehlerquoten analysieren, um Bereiche mit Optimierungsbedarf zu identifizieren. Da sich die generativen KI-Funktionen weiterhin rasant weiterentwickeln, ist es von entscheidender Bedeutung, ständig neue KI-gestützte Tools und Funktionen zu evaluieren, die Arbeitsabläufe weiter rationalisieren oder die Fähigkeiten aller Rollen im SDLC erweitern könnten.

Einführung effektiver Projektmanagementpraktiken

Um Ihre komplexen Softwareentwicklungsbemühungen effektiv zu orchestrieren, sollten Sie KI-gestützte Projektmanagementpraktiken einführen. In diesem Zusammenhang geht effektives Projektmanagement über traditionelle Methoden hinaus. Es umfasst KI-gestützte Ansätze, die die Planung, Ausführung und Überwachung im gesamten SDLC verbessern. Agile Frameworks fördern Flexibilität, Zusammenarbeit und schnelle Iteration, und Sie können generative KI verwenden, um diese Prozesse zu optimieren. Generative KI kann beispielsweise historische Projektdaten analysieren, um genauere Schätzungen zu erhalten, Benutzerberichte auf der Grundlage von Geschäftszielen und Kundenfeedback automatisch generieren und priorisieren und intelligente Einblicke in die Teamleistung bieten. KI-gestützte Projektmanagement-Tools können potenzielle Hindernisse vorhersagen und optimale Aufgabenzuweisungen vorschlagen, die auf den Fähigkeiten und der Arbeitsbelastung der Teammitglieder basieren. Durch die Integration von KI-gestützten Funktionen in die Projektmanagementpraktiken können Sie für mehr Transparenz sorgen, datengestützte Entscheidungen schneller treffen und sicherstellen, dass die Teammitglieder aufeinander abgestimmt sind und effizient auf gemeinsame Ziele hinarbeiten.

Implementierung von Wissensmanagement

Implementieren Sie ein robustes Wissensmanagementsystem, wenn Ihre Erfahrung in der KI-gestützten Softwareentwicklung reift. Ein robustes Wissensmanagementsystem hilft Ihnen dabei, wertvolle Erkenntnisse, bewährte Verfahren und Lösungen zu erfassen, zu organisieren und Zugriff darauf zu gewähren. Alle Teammitglieder im SDLC sollten einfachen Zugriff auf das System haben. Verwenden Sie generative KI, um dynamische, intelligente Wissensdatenbanken zu erstellen, die sich mit Ihrem Unternehmen weiterentwickeln. KI kann beispielsweise automatisch Dokumentationen auf der Grundlage von Codeänderungen, Konversationen und Projektartefakten generieren und aktualisieren, sodass die Informationen ohne manuelles Eingreifen aktuell bleiben. Generative KI kann auch intelligente Suchfunktionen unterstützen und Teammitgliedern helfen, mithilfe von Abfragen in natürlicher Sprache schnell relevante Informationen zu finden, auch wenn sie die genaue Terminologie nicht kennen. Darüber hinaus kann generative KI den Teammitgliedern proaktiv

relevante Informationen auf der Grundlage ihrer aktuellen Aufgaben oder Herausforderungen zur Verfügung stellen. Sie fungiert als virtueller Mentor, der die Entscheidungsfindung und Problemlösung in allen Rollen verbessert. Durch die Implementierung eines KI-gestützten Wissensmanagementsystems können Sie Silos aufbrechen, das Onboarding beschleunigen, redundante Arbeit reduzieren und eine Kultur des kontinuierlichen Lernens und der Innovation in Ihrem gesamten Softwareentwicklungsteam fördern.

Bereitstellung von Erweiterbarkeit und Anpassung

Um die Vorteile generativer KI in der Softwareentwicklung zu maximieren, stellen Sie sicher, dass Ihre KI-gestützten Tools und Plattformen erweiterbar und anpassbar sind. Dies hilft Ihnen dabei, die KI-Funktionen an Ihre spezifischen Bedürfnisse, Workflows und Technologie-Stacks anzupassen. Sie können beispielsweise KI-Modelle anhand Ihrer eigenen Codebasen und Dokumentationen verfeinern, benutzerdefinierte KI-gestützte Tools für bestimmte Aufgaben erstellen oder KI-Funktionen in bestehende Tools und Prozesse integrieren. Diese Erweiterbarkeit hilft Ihnen dabei, die KI-gestützte Entwicklungserfahrung weiterzuentwickeln, um den sich ändernden Anforderungen des Unternehmens gerecht zu werden. Sie hilft Ihnen auch dabei, das Erlebnis für bestimmte Bereiche oder Projekttypen zu optimieren.

Optimierung für den Betrieb

Generative KI spielt eine entscheidende Rolle bei der Optimierung des Betriebs und der Wartung von Software. Optimieren Sie den Betrieb, indem Sie KI-Funktionen in Ihre betrieblichen Tools und Prozesse integrieren. Verwenden Sie beispielsweise generative KI, um Protokolldaten nahezu in Echtzeit zu analysieren, potenzielle Systemausfälle vorherzusagen und routinemäßige Wartungsaufgaben zu automatisieren. Generative KI kann auch bei der Ursachenanalyse helfen, indem sie Ereignisse in komplexen verteilten Systemen korreliert. Dies verbessert die Systemzuverlässigkeit, reduziert Ausfallzeiten und gibt Ihren Betriebsteams die Möglichkeit, sich auf strategischere Initiativen zu konzentrieren.

Nutzung datengestützter Erkenntnisse

Nutzen Sie während Ihrer gesamten KI-gestützten Entwicklungsreise datengestützte Erkenntnisse. Implementieren Sie Systeme zur Erfassung, Analyse und Auswertung von Daten aus allen Phasen des SDLC. Dazu gehören Code-Metriken, Testergebnisse, Bereitstellungsdaten, Benutzerfeedback und Betriebsleistung. Verwenden Sie generative KI, um Muster und Erkenntnisse aufzudecken, die

für menschliche Beobachter möglicherweise nicht offensichtlich sind. Lassen Sie diese Erkenntnisse dann wieder in Ihren Entwicklungsprozess einfließen, um alles zu beeinflussen, von architektonischen Entscheidungen bis hin zur Priorisierung von Funktionen.

Einführung eines plattformbasierten Ansatzes

Um die Vorteile generativer KI in der Softwareentwicklung voll auszuschöpfen, sollten Sie einen plattformbasierten Ansatz verfolgen. Schaffen Sie eine umfassende, integrierte Plattform, die KI-Funktionen in allen Aspekten des SDLC integriert. Die Plattform sollte eine konsistente Benutzererfahrung, zentrale Verwaltung und Daten sowie eine nahtlose Integration zwischen verschiedenen Tools und Prozessen bieten. Dadurch sind KI-Vorteile einheitlich in Ihrem gesamten Unternehmen verfügbar, der Aufwand für die Verwaltung mehrerer und unterschiedlicher KI-Tools wird reduziert und eine Grundlage für die kontinuierliche Verbesserung und Erweiterung der KI-Funktionen geschaffen.

Messung des Erfolgs generativer KI in der Softwareentwicklung

Um die Auswirkungen der Implementierung einer generativen KI-gestützten Softwareentwicklungserfahrung effektiv messen zu können, müssen Sie einen umfassenden Satz von Kennzahlen erstellen, die sich über verschiedene Dimensionen Ihres Softwareentwicklungszyklus (SDLC) erstrecken. Diese Kennzahlen sollten unmittelbare Verbesserungen der Effizienz und Produktivität erfassen und auch langfristige Verbesserungen der Softwarequalität, der Teamzufriedenheit und des Geschäftswerts widerspiegeln.

Gehen Sie wie folgt vor, um die empfohlenen Kennzahlen in diesem Abschnitt effektiv zu nutzen:

1. Legen Sie Basiswerte fest — Bevor Sie mit der Implementierung Ihrer KI-gestützten Entwicklungserfahrung beginnen, sollten Sie sich Zeit nehmen, umfassende Daten über Ihre aktuelle Leistung anhand dieser Kennzahlen zu sammeln. Dies bietet einen klaren Ausgangspunkt und hilft Ihnen, später aussagekräftige Vergleiche anzustellen.
2. Setzen Sie realistische Ziele — Legen Sie mit Ihren Ausgangswerten für jede Kennzahl erreichbare Verbesserungsziele fest. Seien Sie ehrgeizig, aber realistisch. Denken Sie daran, dass nachhaltiger Fortschritt oft schrittweise erfolgt.
3. Implementieren Sie eine kontinuierliche Überwachung — Verwenden Sie automatisierte Tools, um kontinuierlich Daten für diese Kennzahlen in Ihrer Umgebung zu sammeln und zu analysieren. Die Überwachung nahezu in Echtzeit hilft Ihnen dabei, den Fortschritt zu überwachen und Probleme oder Chancen schnell zu erkennen.
4. Führen Sie regelmäßige Überprüfungen durch — Planen Sie vierteljährliche oder halbjährliche Überprüfungssitzungen, in denen Sie und Ihr Team Ihre Fortschritte anhand der Ziele gründlich bewerten. Nutzen Sie diese Sitzungen, um Bereiche mit Verbesserungspotenzial zu identifizieren und Ihre Erfolge zu feiern.
5. Iterieren und anpassen — Verfeinern Sie Ihre generative KI-Implementierung auf der Grundlage der gewonnenen Erkenntnisse kontinuierlich und passen Sie die Ziele nach Bedarf an.

In diesem Abschnitt werden die folgenden Kategorien von Kennzahlen beschrieben:

- [Geschwindigkeit der Bereitstellung](#)
- [Qualität des Codes](#)

- [Operationelle Effizienz](#)
- [Produktivität und Zufriedenheit des Teams](#)
- [Auswirkungen auf das Geschäft](#)

Geschwindigkeit der Bereitstellung

Erwägen Sie, die folgenden Kennzahlen zur Bereitstellungsgeschwindigkeit zu messen.

Metrik	Description
Zeit bis zur Markteinführung	Messen Sie die Verkürzung der Zeit von der Ideenkonzeption bis zur Serieneinführung
Geschwindigkeit des Sprints	Verfolge den Anstieg der Story Points, die deine Teams pro Sprint abgeschlossen haben
Häufigkeit der Code-Commits	Überwachen Sie die Zunahme von Code-Commits, was auf beschleunigte Entwicklungszyklen hindeutet
Zeit für die Lösung von Pull-Anfragen	Beurteilen Sie, wie viel Zeit für die Überprüfung und Zusammenführung von Codeänderungen in Ihren Repositories benötigt wird
Geschwindigkeit der Veröffentlichung	Messen Sie den Anstieg der Anzahl der Veröffentlichungen pro Quartal oder Jahr

Qualität des Codes

Erwägen Sie, die folgenden Kennzahlen zur Codequalität zu messen.

Metrik	Description
Dichte der Defekte	Messen Sie die Reduzierung von Softwarefehlern

Metrik	Description
Codeabdeckung	Verfolgen Sie den Anstieg der prozentualen Testabdeckung in Ihrer gesamten Codebasis
Technische Schulden	Überwachen Sie den Rückgang der identifizierten technischen Schulden im Laufe der Zeit
Ergebnisse der statischen Codeanalyse	Beurteilen Sie Verbesserungen der Codequalität auf der Grundlage Ihrer automatisierten Analysetools

Operationelle Effizienz

Erwägen Sie die Messung der folgenden Kennzahlen zur betrieblichen Effizienz.

Metrik	Description
Häufigkeit der Bereitstellung	Messen Sie den Anstieg der Anzahl erfolgreicher Bereitstellungen
Mittlere Zeit bis zur Wiederherstellung (MTTR)	Verfolgen Sie die Verkürzung der Zeit, die für die Wiederherstellung nach Systemausfällen benötigt wird
Ausfallrate ändern	Überwachen Sie den Rückgang des Prozentsatzes der Änderungen, die zu Fehlern in Ihren Bereitstellungen führen

Produktivität und Zufriedenheit des Teams

Erwägen Sie, die folgenden Kennzahlen zur Teamproduktivität und -zufriedenheit zu messen.

Metrik	Description
Verbesserung der Produktivität	Überwachen Sie den prozentualen Anstieg der Produktivität für jede Aufgabe
Zufriedenheitswert	Führen Sie regelmäßige Umfragen durch, um die Verbesserung der Arbeitsmoral und der Arbeitszufriedenheit Ihres Teams zu messen
Effizienz des Wissensaustauschs	Messen Sie, wie viel Zeit Ihr Team damit verbringt, nach Informationen zu suchen oder sich wiederholende Fragen zu stellen
Einführungszeit	Verfolgen Sie die Verkürzung der Zeit, die neue Teammitglieder benötigen, um produktiv zu werden

Auswirkungen auf das Geschäft

Erwägen Sie, die folgenden Kennzahlen zu den Auswirkungen auf das Geschäft zu messen.

Metrik	Description
Akzeptanzrate der Funktionen	Messen Sie die Zunahme der Benutzerinteraktion mit neuen Funktionen, die Sie veröffentlicht haben
Bewertung der Kundenzufriedenheit	Verfolgen Sie die Verbesserungen Ihres Benutzerfeedbacks und Ihrer Bewertungen
Auswirkungen auf den Umsatz (direkt und indirekt)	Beurteilen Sie den Umsatzanstieg, der auf eine höhere Veröffentlichungsgeschwindigkeit oder eine höhere Produktivität zurückzuführen ist

Schlussfolgerung

Dieses Strategiedokument bietet einen Überblick über eine generative KI-gestützte Softwareentwicklungserfahrung. Es untersucht die fünf Dimensionen des [5-I-Frameworks](#) — Investigate, Integrate, Interact, Iterate und Impact. Diese Dimensionen bieten eine strategische Roadmap für die Integration generativer KI über den gesamten Softwareentwicklungszyklus (SDLC). Es beschreibt auch die [grundlegenden Funktionen](#), die für die erfolgreiche Implementierung dieses Frameworks erforderlich sind. Die Funktionen umfassen Bereiche wie Projektmanagement, DevSecOps, KI-Assistenten, Wissensmanagement und mehr. Es bietet [bewährte Verfahren](#), die Sie bei der Integration generativer KI berücksichtigen sollten, und hilft Ihnen dabei, anhand von [Metriken](#) zu messen, wie sich generative KI auf Ihre Erfahrung in der Softwareentwicklung auswirkt.

Die Integration generativer KI in Softwareentwicklungsprozesse stellt einen Paradigmenwechsel dar, der das Potenzial hat, Innovationen zu beschleunigen, die Qualität zu verbessern und die Produktivität zu steigern. Es ist jedoch wichtig zu erkennen, dass es sich nicht um eine einmalige Implementierung handelt. Es handelt sich um eine kontinuierliche Weiterentwicklung, die nachhaltige Anstrengungen und kontinuierliche Weiterentwicklung erfordert.

Wenn Sie sich auf diese Reise begeben, empfehlen wir Ihnen, mit einer gründlichen Bewertung der aktuellen Fähigkeiten und Einsatzbereitschaft Ihres Unternehmens zu beginnen. Das [AWS Assessment Tool](#) ist ein KI-gestütztes Bewertungstool für die Softwareentwicklung, das Ihnen helfen kann, Schwerpunktbereiche zu identifizieren und einen maßgeschneiderten Implementierungsplan zu erstellen.

Ressourcen

Sobald Sie die wichtigsten Schwerpunktbereiche identifiziert haben, können Ihnen die folgenden Ressourcen bei der Umsetzung Ihrer Roadmap helfen:

AWS Dokumentation

- [Automatisieren Sie den AWS Infrastrukturbetrieb mithilfe von Amazon Bedrock](#) (AWS Prescriptive Guidance)
- [Bewährte Methoden mit Amazon Q Developer für die Inline- und Assistentengenerierung](#) (AWS Prescriptive Guidance)
- [Entwickeln Sie mithilfe von Amazon Bedrock-Agenten und Wissensdatenbanken einen vollautomatischen chatbasierten Assistenten](#) (AWS Prescriptive Guidance)

- [Transformieren Sie die Betriebsmodelle für Anwendungsentwicklung und Wartung AWS mit generativer KI \(Prescriptive Guidance\)](#)AWS
- [Verwenden Sie Amazon Q Developer als Programmierassistenten, um Ihre Produktivität zu steigern](#) (AWS Prescriptive Guidance)

AWS Blogbeiträge und Tutorials

- [Amazon Q-Blogbeiträge](#)
- [Beschleunigen Sie Ihren Softwareentwicklungszyklus mit Amazon Q](#) (AWS Blogbeitrag)
- [Aufbau eines KI-Agenten für AWS Solutions Architect: Nutzung von Amazon Bedrock für automatisierte Architektur und Bereitstellung \(Video\)](#)AWS
- [Generativer KI-gestützter Technologiebetrieb \(Blogbeitrag\)](#)AWS
- [Modernisieren Sie Ihre Java-Anwendung mit Amazon Q Developer](#) (AWS Blogbeitrag)
- [Verwenden Sie Amazon Bedrock, um Code in Ihrer Softwareentwicklungspipeline zu generieren, zu bewerten und zu verstehen](#) (AWS Blogbeitrag)

Dokumentverlauf

In der folgenden Tabelle werden wichtige Änderungen in diesem Leitfaden beschrieben. Um Benachrichtigungen über zukünftige Aktualisierungen zu erhalten, können Sie einen [RSS-Feed](#) abonnieren.

Änderung	Beschreibung	Datum
Erste Veröffentlichung	Nicht zutreffend	18. April 2025

AWS Glossar zu präskriptiven Leitlinien

Die folgenden Begriffe werden häufig in Strategien, Leitfäden und Mustern von AWS Prescriptive Guidance verwendet. Um Einträge vorzuschlagen, verwenden Sie bitte den Link Feedback geben am Ende des Glossars.

Zahlen

7 Rs

Sieben gängige Migrationsstrategien für die Verlagerung von Anwendungen in die Cloud. Diese Strategien bauen auf den 5 Rs auf, die Gartner 2011 identifiziert hat, und bestehen aus folgenden Elementen:

- Faktorwechsel/Architekturwechsel – Verschieben Sie eine Anwendung und ändern Sie ihre Architektur, indem Sie alle Vorteile cloudnativer Feature nutzen, um Agilität, Leistung und Skalierbarkeit zu verbessern. Dies beinhaltet in der Regel die Portierung des Betriebssystems und der Datenbank. Beispiel: Migrieren Sie Ihre lokale Oracle-Datenbank auf die Amazon Aurora PostgreSQL-kompatible Edition.
- Plattformwechsel (Lift and Reshape) – Verschieben Sie eine Anwendung in die Cloud und führen Sie ein gewisses Maß an Optimierung ein, um die Cloud-Funktionen zu nutzen. Beispiel: Migrieren Sie Ihre lokale Oracle-Datenbank zu Amazon Relational Database Service (Amazon RDS) für Oracle in der AWS Cloud
- Neukauf (Drop and Shop) – Wechseln Sie zu einem anderen Produkt, indem Sie typischerweise von einer herkömmlichen Lizenz zu einem SaaS-Modell wechseln. Beispiel: Migrieren Sie Ihr CRM-System (Customer Relationship Management) zu Salesforce.com.
- Hostwechsel (Lift and Shift) – Verschieben Sie eine Anwendung in die Cloud, ohne Änderungen vorzunehmen, um die Cloud-Funktionen zu nutzen. Beispiel: Migrieren Sie Ihre lokale Oracle-Datenbank zu Oracle auf einer EC2-Instanz in der AWS Cloud
- Verschieben (Lift and Shift auf Hypervisor-Ebene) – Verlagern Sie die Infrastruktur in die Cloud, ohne neue Hardware kaufen, Anwendungen umschreiben oder Ihre bestehenden Abläufe ändern zu müssen. Sie migrieren Server von einer lokalen Plattform zu einem Cloud-Dienst für dieselbe Plattform. Beispiel: Migrieren Sie eine Microsoft Hyper-V Anwendung zu AWS.
- Beibehaltung (Wiederaufgreifen) – Bewahren Sie Anwendungen in Ihrer Quellumgebung auf. Dazu können Anwendungen gehören, die einen umfangreichen Faktorwechsel erfordern und

die Sie auf einen späteren Zeitpunkt verschieben möchten, sowie ältere Anwendungen, die Sie beibehalten möchten, da es keine geschäftliche Rechtfertigung für ihre Migration gibt.

- Außerbetriebnahme – Dekommissionierung oder Entfernung von Anwendungen, die in Ihrer Quellumgebung nicht mehr benötigt werden.

A

ABAC

Siehe [attributbasierte](#) Zugriffskontrolle.

abstrahierte Dienste

Siehe [Managed Services](#).

ACID

Siehe [Atomarität, Konsistenz, Isolierung und Haltbarkeit](#).

Aktiv-Aktiv-Migration

Eine Datenbankmigrationsmethode, bei der die Quell- und Zieldatenbanken synchron gehalten werden (mithilfe eines bidirektionalen Replikationstools oder dualer Schreibvorgänge) und beide Datenbanken Transaktionen von miteinander verbundenen Anwendungen während der Migration verarbeiten. Diese Methode unterstützt die Migration in kleinen, kontrollierten Batches, anstatt einen einmaligen Cutover zu erfordern. Es ist flexibler, erfordert aber mehr Arbeit als eine [aktiv-passive](#) Migration.

Aktiv-Passiv-Migration

Eine Datenbankmigrationsmethode, bei der die Quell- und Zieldatenbanken synchron gehalten werden, aber nur die Quelldatenbank verarbeitet Transaktionen von verbindenden Anwendungen, während Daten in die Zieldatenbank repliziert werden. Die Zieldatenbank akzeptiert während der Migration keine Transaktionen.

Aggregatfunktion

Eine SQL-Funktion, die mit einer Gruppe von Zeilen arbeitet und einen einzelnen Rückgabewert für die Gruppe berechnet. Beispiele für Aggregatfunktionen sind SUM und MAX.

AI

Siehe [künstliche Intelligenz](#).

AIOps

Siehe [Operationen im Bereich künstliche Intelligenz](#).

Anonymisierung

Der Prozess des dauerhaften Löschens personenbezogener Daten in einem Datensatz. Anonymisierung kann zum Schutz der Privatsphäre beitragen. Anonymisierte Daten gelten nicht mehr als personenbezogene Daten.

Anti-Muster

Eine häufig verwendete Lösung für ein wiederkehrendes Problem, bei dem die Lösung kontraproduktiv, ineffektiv oder weniger wirksam als eine Alternative ist.

Anwendungssteuerung

Ein Sicherheitsansatz, bei dem nur zugelassene Anwendungen verwendet werden können, um ein System vor Schadsoftware zu schützen.

Anwendungsportfolio

Eine Sammlung detaillierter Informationen zu jeder Anwendung, die von einer Organisation verwendet wird, einschließlich der Kosten für die Erstellung und Wartung der Anwendung und ihres Geschäftswerts. Diese Informationen sind entscheidend für [den Prozess der Portfoliofindung und -analyse](#) und hilft bei der Identifizierung und Priorisierung der Anwendungen, die migriert, modernisiert und optimiert werden sollen.

künstliche Intelligenz (KI)

Das Gebiet der Datenverarbeitungswissenschaft, das sich der Nutzung von Computertechnologien zur Ausführung kognitiver Funktionen widmet, die typischerweise mit Menschen in Verbindung gebracht werden, wie Lernen, Problemlösen und Erkennen von Mustern. Weitere Informationen finden Sie unter [Was ist künstliche Intelligenz?](#)

Operationen mit künstlicher Intelligenz (AIOps)

Der Prozess des Einsatzes von Techniken des Machine Learning zur Lösung betrieblicher Probleme, zur Reduzierung betrieblicher Zwischenfälle und menschlicher Eingriffe sowie zur Steigerung der Servicequalität. Weitere Informationen zur Verwendung in der AWS Migrationsstrategie finden Sie im [Operations Integration Guide](#). AIOps

Asymmetrische Verschlüsselung

Ein Verschlüsselungsalgorithmus, der ein Schlüsselpaar, einen öffentlichen Schlüssel für die Verschlüsselung und einen privaten Schlüssel für die Entschlüsselung verwendet. Sie können den öffentlichen Schlüssel teilen, da er nicht für die Entschlüsselung verwendet wird. Der Zugriff auf den privaten Schlüssel sollte jedoch stark eingeschränkt sein.

Atomizität, Konsistenz, Isolierung, Haltbarkeit (ACID)

Eine Reihe von Softwareeigenschaften, die die Datenvalidität und betriebliche Zuverlässigkeit einer Datenbank auch bei Fehlern, Stromausfällen oder anderen Problemen gewährleisten.

Attributbasierte Zugriffskontrolle (ABAC)

Die Praxis, detaillierte Berechtigungen auf der Grundlage von Benutzerattributen wie Abteilung, Aufgabenrolle und Teamname zu erstellen. Weitere Informationen finden Sie unter [ABAC AWS](#) in der AWS Identity and Access Management (IAM-) Dokumentation.

autoritative Datenquelle

Ein Ort, an dem Sie die primäre Version der Daten speichern, die als die zuverlässigste Informationsquelle angesehen wird. Sie können Daten aus der maßgeblichen Datenquelle an andere Speicherorte kopieren, um die Daten zu verarbeiten oder zu ändern, z. B. zu anonymisieren, zu redigieren oder zu pseudonymisieren.

Availability Zone

Ein bestimmter Standort innerhalb einer AWS-Region, der vor Ausfällen in anderen Availability Zones geschützt ist und kostengünstige Netzwerkkonnektivität mit niedriger Latenz zu anderen Availability Zones in derselben Region bietet.

AWS Framework für die Einführung der Cloud (AWS CAF)

Ein Framework mit Richtlinien und bewährten Verfahren, das Unternehmen bei der Entwicklung eines effizienten und effektiven Plans für die erfolgreiche Umstellung auf die Cloud unterstützt. AWS CAF unterteilt die Leitlinien in sechs Schwerpunktbereiche, die als Perspektiven bezeichnet werden: Unternehmen, Mitarbeiter, Unternehmensführung, Plattform, Sicherheit und Betrieb. Die Perspektiven Geschäft, Mitarbeiter und Unternehmensführung konzentrieren sich auf Geschäftskompetenzen und -prozesse, während sich die Perspektiven Plattform, Sicherheit und Betriebsabläufe auf technische Fähigkeiten und Prozesse konzentrieren. Die Personalperspektive zielt beispielsweise auf Stakeholder ab, die sich mit Personalwesen (HR), Personalfunktionen und Personalmanagement befassen. Aus dieser Perspektive bietet AWS CAF Leitlinien für Personalentwicklung, Schulung und Kommunikation, um das Unternehmen auf eine erfolgreiche

Cloud-Einführung vorzubereiten. Weitere Informationen finden Sie auf der [AWS -CAF-Webseite](#) und dem [AWS -CAF-Whitepaper](#).

AWS Workload-Qualifizierungsrahmen (AWS WQF)

Ein Tool, das Workloads bei der Datenbankmigration bewertet, Migrationsstrategien empfiehlt und Arbeitsschätzungen bereitstellt. AWS WQF ist in () enthalten. AWS Schema Conversion Tool AWS SCT Es analysiert Datenbankschemas und Codeobjekte, Anwendungscode, Abhängigkeiten und Leistungsmerkmale und stellt Bewertungsberichte bereit.

B

schlechter Bot

Ein [Bot](#), der Einzelpersonen oder Organisationen stören oder ihnen Schaden zufügen soll.

BCP

Siehe [Planung der Geschäftskontinuität](#).

Verhaltensdiagramm

Eine einheitliche, interaktive Ansicht des Ressourcenverhaltens und der Interaktionen im Laufe der Zeit. Sie können ein Verhaltensdiagramm mit Amazon Detective verwenden, um fehlgeschlagene Anmeldeversuche, verdächtige API-Aufrufe und ähnliche Vorgänge zu untersuchen. Weitere Informationen finden Sie unter [Daten in einem Verhaltensdiagramm](#) in der Detective-Dokumentation.

Big-Endian-System

Ein System, welches das höchstwertige Byte zuerst speichert. Siehe auch [Endianness](#).

Binäre Klassifikation

Ein Prozess, der ein binäres Ergebnis vorhersagt (eine von zwei möglichen Klassen). Beispielsweise könnte Ihr ML-Modell möglicherweise Probleme wie „Handelt es sich bei dieser E-Mail um Spam oder nicht?“ vorhersagen müssen oder „Ist dieses Produkt ein Buch oder ein Auto?“

Bloom-Filter

Eine probabilistische, speichereffiziente Datenstruktur, mit der getestet wird, ob ein Element Teil einer Menge ist.

Blau/Grün-Bereitstellung

Eine Bereitstellungsstrategie, bei der Sie zwei separate, aber identische Umgebungen erstellen. Sie führen die aktuelle Anwendungsversion in einer Umgebung (blau) und die neue Anwendungsversion in der anderen Umgebung (grün) aus. Mit dieser Strategie können Sie schnell und mit minimalen Auswirkungen ein Rollback durchführen.

Bot

Eine Softwareanwendung, die automatisierte Aufgaben über das Internet ausführt und menschliche Aktivitäten oder Interaktionen simuliert. Manche Bots sind nützlich oder nützlich, wie z. B. Webcrawler, die Informationen im Internet indexieren. Einige andere Bots, sogenannte bösartige Bots, sollen Einzelpersonen oder Organisationen stören oder ihnen Schaden zufügen.

Botnetz

Netzwerke von [Bots](#), die mit [Malware](#) infiziert sind und unter der Kontrolle einer einzigen Partei stehen, die als Bot-Herder oder Bot-Operator bezeichnet wird. Botnetze sind der bekannteste Mechanismus zur Skalierung von Bots und ihrer Wirkung.

branch

Ein containerisierter Bereich eines Code-Repositorys. Der erste Zweig, der in einem Repository erstellt wurde, ist der Hauptzweig. Sie können einen neuen Zweig aus einem vorhandenen Zweig erstellen und dann Feature entwickeln oder Fehler in dem neuen Zweig beheben. Ein Zweig, den Sie erstellen, um ein Feature zu erstellen, wird allgemein als Feature-Zweig bezeichnet. Wenn das Feature zur Veröffentlichung bereit ist, führen Sie den Feature-Zweig wieder mit dem Hauptzweig zusammen. Weitere Informationen finden Sie unter [Über Branches](#) (GitHub Dokumentation).

Zugang durch Glasbruch

Unter außergewöhnlichen Umständen und im Rahmen eines genehmigten Verfahrens ist dies eine schnelle Methode für einen Benutzer, auf einen Bereich zuzugreifen AWS-Konto, für den er normalerweise keine Zugriffsrechte besitzt. Weitere Informationen finden Sie unter dem Indikator [Implementation break-glass procedures](#) in den AWS Well-Architected-Leitlinien.

Brownfield-Strategie

Die bestehende Infrastruktur in Ihrer Umgebung. Wenn Sie eine Brownfield-Strategie für eine Systemarchitektur anwenden, richten Sie sich bei der Gestaltung der Architektur nach den Einschränkungen der aktuellen Systeme und Infrastruktur. Wenn Sie die bestehende Infrastruktur erweitern, könnten Sie Brownfield- und [Greenfield](#)-Strategien mischen.

Puffer-Cache

Der Speicherbereich, in dem die am häufigsten abgerufenen Daten gespeichert werden.

Geschäftsfähigkeit

Was ein Unternehmen tut, um Wert zu generieren (z. B. Vertrieb, Kundenservice oder Marketing). Microservices-Architekturen und Entwicklungsentscheidungen können von den Geschäftskapazitäten beeinflusst werden. Weitere Informationen finden Sie im Abschnitt [Organisiert nach Geschäftskapazitäten](#) des Whitepapers [Ausführen von containerisierten Microservices in AWS](#).

Planung der Geschäftskontinuität (BCP)

Ein Plan, der die potenziellen Auswirkungen eines störenden Ereignisses, wie z. B. einer groß angelegten Migration, auf den Betrieb berücksichtigt und es einem Unternehmen ermöglicht, den Betrieb schnell wieder aufzunehmen.

C

CAF

[Weitere Informationen finden Sie unter Framework AWS für die Cloud-Einführung.](#)

Bereitstellung auf Kanaren

Die langsame und schrittweise Veröffentlichung einer Version für Endbenutzer. Wenn Sie sich sicher sind, stellen Sie die neue Version bereit und ersetzen die aktuelle Version vollständig.

CCoE

Weitere Informationen finden Sie [im Cloud Center of Excellence](#).

CDC

Siehe [Erfassung von Änderungsdaten](#).

Erfassung von Datenänderungen (CDC)

Der Prozess der Nachverfolgung von Änderungen an einer Datenquelle, z. B. einer Datenbanktabelle, und der Aufzeichnung von Metadaten zu der Änderung. Sie können CDC für verschiedene Zwecke verwenden, z. B. für die Prüfung oder Replikation von Änderungen in einem Zielsystem, um die Synchronisation aufrechtzuerhalten.

Chaos-Technik

Absichtliches Einführen von Ausfällen oder Störungsereignissen, um die Widerstandsfähigkeit eines Systems zu testen. Sie können [AWS Fault Injection Service \(AWS FIS\)](#) verwenden, um Experimente durchzuführen, die Ihre AWS Workloads stress, und deren Reaktion zu bewerten.

CI/CD

Siehe [Continuous Integration und Continuous Delivery](#).

Klassifizierung

Ein Kategorisierungsprozess, der bei der Erstellung von Vorhersagen hilft. ML-Modelle für Klassifikationsprobleme sagen einen diskreten Wert voraus. Diskrete Werte unterscheiden sich immer voneinander. Beispielsweise muss ein Modell möglicherweise auswerten, ob auf einem Bild ein Auto zu sehen ist oder nicht.

clientseitige Verschlüsselung

Lokale Verschlüsselung von Daten, bevor das Ziel sie AWS-Service empfängt.

Cloud-Exzellenzzentrum (CCoE)

Ein multidisziplinäres Team, das die Cloud-Einführung in der gesamten Organisation vorantreibt, einschließlich der Entwicklung bewährter Cloud-Methoden, der Mobilisierung von Ressourcen, der Festlegung von Migrationszeitplänen und der Begleitung der Organisation durch groß angelegte Transformationen. Weitere Informationen finden Sie in den [CCoE-Beiträgen](#) im AWS Cloud Enterprise Strategy Blog.

Cloud Computing

Die Cloud-Technologie, die typischerweise für die Ferndatenspeicherung und das IoT-Gerätemanagement verwendet wird. Cloud Computing ist häufig mit [Edge-Computing-Technologie](#) verbunden.

Cloud-Betriebsmodell

In einer IT-Organisation das Betriebsmodell, das zum Aufbau, zur Weiterentwicklung und Optimierung einer oder mehrerer Cloud-Umgebungen verwendet wird. Weitere Informationen finden Sie unter [Aufbau Ihres Cloud-Betriebsmodells](#).

Phasen der Einführung der Cloud

Die vier Phasen, die Unternehmen bei der Migration in der Regel durchlaufen AWS Cloud:

- Projekt – Durchführung einiger Cloud-bezogener Projekte zu Machbarkeitsnachweisen und zu Lernzwecken
- Fundament — Tätigen Sie grundlegende Investitionen, um Ihre Cloud-Einführung zu skalieren (z. B. Einrichtung einer landing zone, Definition eines CCo E, Einrichtung eines Betriebsmodells)
- Migration – Migrieren einzelner Anwendungen
- Neuentwicklung – Optimierung von Produkten und Services und Innovation in der Cloud

Diese Phasen wurden von Stephen Orban im Blogbeitrag [The Journey Toward Cloud-First & the Stages of Adoption](#) im AWS Cloud Enterprise Strategy-Blog definiert. Informationen darüber, wie sie mit der AWS Migrationsstrategie zusammenhängen, finden Sie im Leitfaden zur Vorbereitung der [Migration](#).

CMDB

Siehe [Datenbank für das Konfigurationsmanagement](#).

Code-Repository

Ein Ort, an dem Quellcode und andere Komponenten wie Dokumentation, Beispiele und Skripts gespeichert und im Rahmen von Versionskontrollprozessen aktualisiert werden. Zu den gängigen Cloud-Repositorys gehören GitHub oder Bitbucket Cloud. Jede Version des Codes wird Zweig genannt. In einer Microservice-Struktur ist jedes Repository einer einzelnen Funktionalität gewidmet. Eine einzelne CI/CD-Pipeline kann mehrere Repositorien verwenden.

Kalter Cache

Ein Puffer-Cache, der leer oder nicht gut gefüllt ist oder veraltete oder irrelevante Daten enthält. Dies beeinträchtigt die Leistung, da die Datenbank-Instance aus dem Hauptspeicher oder der Festplatte lesen muss, was langsamer ist als das Lesen aus dem Puffercache.

Kalte Daten

Daten, auf die selten zugegriffen wird und die in der Regel historisch sind. Bei der Abfrage dieser Art von Daten sind langsame Abfragen in der Regel akzeptabel. Durch die Verlagerung dieser Daten auf leistungsschwächere und kostengünstigere Speicherstufen oder -klassen können Kosten gesenkt werden.

Computer Vision (CV)

Ein Bereich der [KI](#), der maschinelles Lernen nutzt, um Informationen aus visuellen Formaten wie digitalen Bildern und Videos zu analysieren und zu extrahieren. Amazon SageMaker AI bietet beispielsweise Bildverarbeitungsalgorithmen für CV.

Drift in der Konfiguration

Bei einer Arbeitslast eine Änderung der Konfiguration gegenüber dem erwarteten Zustand. Dies kann dazu führen, dass der Workload nicht mehr richtlinienkonform wird, und zwar in der Regel schrittweise und unbeabsichtigt.

Verwaltung der Datenbankkonfiguration (CMDB)

Ein Repository, das Informationen über eine Datenbank und ihre IT-Umgebung speichert und verwaltet, inklusive Hardware- und Softwarekomponenten und deren Konfigurationen. In der Regel verwenden Sie Daten aus einer CMDB in der Phase der Portfolioerkennung und -analyse der Migration.

Konformitätspaket

Eine Sammlung von AWS Config Regeln und Abhilfemaßnahmen, die Sie zusammenstellen können, um Ihre Konformitäts- und Sicherheitsprüfungen individuell anzupassen. Mithilfe einer YAML-Vorlage können Sie ein Conformance Pack als einzelne Entität in einer AWS-Konto AND-Region oder unternehmensweit bereitstellen. Weitere Informationen finden Sie in der Dokumentation unter [Conformance Packs](#). AWS Config

Kontinuierliche Bereitstellung und kontinuierliche Integration (CI/CD)

Der Prozess der Automatisierung der Quell-, Build-, Test-, Staging- und Produktionsphasen des Softwareveröffentlichungsprozesses. CI/CD wird allgemein als Pipeline beschrieben. CI/CD kann Ihnen helfen, Prozesse zu automatisieren, die Produktivität zu steigern, die Codequalität zu verbessern und schneller zu liefern. Weitere Informationen finden Sie unter [Vorteile der kontinuierlichen Auslieferung](#). CD kann auch für kontinuierliche Bereitstellung stehen. Weitere Informationen finden Sie unter [Kontinuierliche Auslieferung im Vergleich zu kontinuierlicher Bereitstellung](#).

CV

Siehe [Computer Vision](#).

D

Daten im Ruhezustand

Daten, die in Ihrem Netzwerk stationär sind, z. B. Daten, die sich im Speicher befinden.

Datenklassifizierung

Ein Prozess zur Identifizierung und Kategorisierung der Daten in Ihrem Netzwerk auf der Grundlage ihrer Kritikalität und Sensitivität. Sie ist eine wichtige Komponente jeder Strategie für das Management von Cybersecurity-Risiken, da sie Ihnen hilft, die geeigneten Schutz- und Aufbewahrungskontrollen für die Daten zu bestimmen. Die Datenklassifizierung ist ein Bestandteil der Sicherheitssäule im AWS Well-Architected Framework. Weitere Informationen finden Sie unter [Datenklassifizierung](#).

Datendrift

Eine signifikante Abweichung zwischen den Produktionsdaten und den Daten, die zum Trainieren eines ML-Modells verwendet wurden, oder eine signifikante Änderung der Eingabedaten im Laufe der Zeit. Datendrift kann die Gesamtqualität, Genauigkeit und Fairness von ML-Modellvorhersagen beeinträchtigen.

Daten während der Übertragung

Daten, die sich aktiv durch Ihr Netzwerk bewegen, z. B. zwischen Netzwerkressourcen.

Datennetz

Ein architektonisches Framework, das verteilte, dezentrale Dateneigentum mit zentraler Verwaltung und Steuerung ermöglicht.

Datenminimierung

Das Prinzip, nur die Daten zu sammeln und zu verarbeiten, die unbedingt erforderlich sind. Durch Datenminimierung im AWS Cloud können Datenschutzrisiken, Kosten und der CO2-Fußabdruck Ihrer Analysen reduziert werden.

Datenperimeter

Eine Reihe präventiver Schutzmaßnahmen in Ihrer AWS Umgebung, die sicherstellen, dass nur vertrauenswürdige Identitäten auf vertrauenswürdige Ressourcen von erwarteten Netzwerken zugreifen. Weitere Informationen finden Sie unter [Aufbau eines Datenperimeters](#) auf AWS

Vorverarbeitung der Daten

Rohdaten in ein Format umzuwandeln, das von Ihrem ML-Modell problemlos verarbeitet werden kann. Die Vorverarbeitung von Daten kann bedeuten, dass bestimmte Spalten oder Zeilen entfernt und fehlende, inkonsistente oder doppelte Werte behoben werden.

Herkunft der Daten

Der Prozess der Nachverfolgung des Ursprungs und der Geschichte von Daten während ihres gesamten Lebenszyklus, z. B. wie die Daten generiert, übertragen und gespeichert wurden.

betreffene Person

Eine Person, deren Daten gesammelt und verarbeitet werden.

Data Warehouse

Ein Datenverwaltungssystem, das Business Intelligence wie Analysen unterstützt. Data Warehouses enthalten in der Regel große Mengen historischer Daten und werden in der Regel für Abfragen und Analysen verwendet.

Datenbankdefinitionssprache (DDL)

Anweisungen oder Befehle zum Erstellen oder Ändern der Struktur von Tabellen und Objekten in einer Datenbank.

Datenbankmanipulationssprache (DML)

Anweisungen oder Befehle zum Ändern (Einfügen, Aktualisieren und Löschen) von Informationen in einer Datenbank.

DDL

Siehe [Datenbankdefinitionssprache](#).

Deep-Ensemble

Mehrere Deep-Learning-Modelle zur Vorhersage kombinieren. Sie können Deep-Ensembles verwenden, um eine genauere Vorhersage zu erhalten oder um die Unsicherheit von Vorhersagen abzuschätzen.

Deep Learning

Ein ML-Teilbereich, der mehrere Schichten künstlicher neuronaler Netzwerke verwendet, um die Zuordnung zwischen Eingabedaten und Zielvariablen von Interesse zu ermitteln.

defense-in-depth

Ein Ansatz zur Informationssicherheit, bei dem eine Reihe von Sicherheitsmechanismen und -kontrollen sorgfältig in einem Computernetzwerk verteilt werden, um die Vertraulichkeit, Integrität und Verfügbarkeit des Netzwerks und der darin enthaltenen Daten zu schützen. Wenn Sie diese Strategie anwenden AWS, fügen Sie mehrere Steuerelemente auf verschiedenen Ebenen der AWS Organizations Struktur hinzu, um die Ressourcen zu schützen. Ein defense-in-depth Ansatz könnte beispielsweise Multi-Faktor-Authentifizierung, Netzwerksegmentierung und Verschlüsselung kombinieren.

delegierter Administrator

In AWS Organizations kann ein kompatibler Dienst ein AWS Mitgliedskonto registrieren, um die Konten der Organisation und die Berechtigungen für diesen Dienst zu verwalten. Dieses Konto wird als delegierter Administrator für diesen Service bezeichnet. Weitere Informationen und eine Liste kompatibler Services finden Sie unter [Services, die mit AWS Organizations funktionieren](#) in der AWS Organizations -Dokumentation.

Einsatz

Der Prozess, bei dem eine Anwendung, neue Feature oder Codekorrekturen in der Zielumgebung verfügbar gemacht werden. Die Bereitstellung umfasst das Implementieren von Änderungen an einer Codebasis und das anschließende Erstellen und Ausführen dieser Codebasis in den Anwendungsumgebungen.

Entwicklungsumgebung

Siehe [Umgebung](#).

Detektivische Kontrolle

Eine Sicherheitskontrolle, die darauf ausgelegt ist, ein Ereignis zu erkennen, zu protokollieren und zu warnen, nachdem ein Ereignis eingetreten ist. Diese Kontrollen stellen eine zweite Verteidigungslinie dar und warnen Sie vor Sicherheitsereignissen, bei denen die vorhandenen präventiven Kontrollen umgangen wurden. Weitere Informationen finden Sie unter [Detektivische Kontrolle](#) in Implementierung von Sicherheitskontrollen in AWS.

Abbildung des Wertstroms in der Entwicklung (DVSM)

Ein Prozess zur Identifizierung und Priorisierung von Einschränkungen, die sich negativ auf Geschwindigkeit und Qualität im Lebenszyklus der Softwareentwicklung auswirken. DVSM erweitert den Prozess der Wertstromanalyse, der ursprünglich für Lean-Manufacturing-Praktiken

konzipiert wurde. Es konzentriert sich auf die Schritte und Teams, die erforderlich sind, um durch den Softwareentwicklungsprozess Mehrwert zu schaffen und zu steigern.

digitaler Zwilling

Eine virtuelle Darstellung eines realen Systems, z. B. eines Gebäudes, einer Fabrik, einer Industrieanlage oder einer Produktionslinie. Digitale Zwillinge unterstützen vorausschauende Wartung, Fernüberwachung und Produktionsoptimierung.

Maßtabelle

In einem [Sternschema](#) eine kleinere Tabelle, die Datenattribute zu quantitativen Daten in einer Faktentabelle enthält. Bei Attributen von Dimensionstabellen handelt es sich in der Regel um Textfelder oder diskrete Zahlen, die sich wie Text verhalten. Diese Attribute werden häufig zum Einschränken von Abfragen, zum Filtern und zur Kennzeichnung von Ergebnismengen verwendet.

Katastrophe

Ein Ereignis, das verhindert, dass ein Workload oder ein System seine Geschäftsziele an seinem primären Einsatzort erfüllt. Diese Ereignisse können Naturkatastrophen, technische Ausfälle oder das Ergebnis menschlichen Handelns sein, wie z. B. unbeabsichtigte Fehlkonfigurationen oder ein Malware-Angriff.

Notfallwiederherstellung (DR)

Die Strategie und der Prozess, mit denen Sie Ausfallzeiten und Datenverluste aufgrund einer [Katastrophe](#) minimieren. Weitere Informationen finden Sie unter [Disaster Recovery von Workloads unter AWS: Wiederherstellung in der Cloud im](#) AWS Well-Architected Framework.

DML

Siehe Sprache zur [Datenbankmanipulation](#).

Domainorientiertes Design

Ein Ansatz zur Entwicklung eines komplexen Softwaresystems, bei dem seine Komponenten mit sich entwickelnden Domains oder Kerngeschäftsziele verknüpft werden, denen jede Komponente dient. Dieses Konzept wurde von Eric Evans in seinem Buch *Domaingesteuertes Design: Bewältigen der Komplexität im Herzen der Software* (Boston: Addison-Wesley Professional, 2003) vorgestellt. Informationen darüber, wie Sie domaingesteuertes Design mit dem Strangler-Fig-Muster verwenden können, finden Sie unter [Schrittweises Modernisieren älterer Microsoft ASP.NET \(ASMX\)-Webservices mithilfe von Containern und Amazon API Gateway](#).

DR

Siehe [Disaster Recovery](#).

Erkennung von Driften

Verfolgung von Abweichungen von einer Basiskonfiguration. Sie können es beispielsweise verwenden, AWS CloudFormation um [Abweichungen bei den Systemressourcen zu erkennen](#), oder Sie können AWS Control Tower damit [Änderungen in Ihrer landing zone erkennen](#), die sich auf die Einhaltung von Governance-Anforderungen auswirken könnten.

DVSM

Siehe [Abbildung des Wertstroms in der Entwicklung](#).

E

EDA

Siehe [explorative Datenanalyse](#).

EDI

Siehe [elektronischer Datenaustausch](#).

Edge-Computing

Die Technologie, die die Rechenleistung für intelligente Geräte an den Rändern eines IoT-Netzwerks erhöht. Im Vergleich zu [Cloud Computing](#) kann Edge Computing die Kommunikationslatenz reduzieren und die Reaktionszeit verbessern.

elektronischer Datenaustausch (EDI)

Der automatisierte Austausch von Geschäftsdokumenten zwischen Organisationen. Weitere Informationen finden Sie unter [Was ist elektronischer Datenaustausch](#).

Verschlüsselung

Ein Rechenprozess, der Klartextdaten, die für Menschen lesbar sind, in Chiffretext umwandelt.

Verschlüsselungsschlüssel

Eine kryptografische Zeichenfolge aus zufälligen Bits, die von einem Verschlüsselungsalgorithmus generiert wird. Schlüssel können unterschiedlich lang sein, und jeder Schlüssel ist so konzipiert, dass er unvorhersehbar und einzigartig ist.

Endianismus

Die Reihenfolge, in der Bytes im Computerspeicher gespeichert werden. Big-Endian-Systeme speichern das höchstwertige Byte zuerst. Little-Endian-Systeme speichern das niedrigwertigste Byte zuerst.

Endpunkt

[Siehe](#) Service-Endpunkt.

Endpunkt-Services

Ein Service, den Sie in einer Virtual Private Cloud (VPC) hosten können, um ihn mit anderen Benutzern zu teilen. Sie können einen Endpunktdienst mit anderen AWS-Konten oder AWS Identity and Access Management (IAM AWS PrivateLink -) Prinzipalen erstellen und diesen Berechtigungen gewähren. Diese Konten oder Prinzipale können sich privat mit Ihrem Endpunktservice verbinden, indem sie Schnittstellen-VPC-Endpunkte erstellen. Weitere Informationen finden Sie unter [Einen Endpunkt-Service erstellen](#) in der Amazon Virtual Private Cloud (Amazon VPC)-Dokumentation.

Unternehmensressourcenplanung (ERP)

Ein System, das wichtige Geschäftsprozesse (wie Buchhaltung, [MES](#) und Projektmanagement) für ein Unternehmen automatisiert und verwaltet.

Envelope-Verschlüsselung

Der Prozess der Verschlüsselung eines Verschlüsselungsschlüssels mit einem anderen Verschlüsselungsschlüssel. Weitere Informationen finden Sie unter [Envelope-Verschlüsselung](#) in der AWS Key Management Service (AWS KMS) -Dokumentation.

Umgebung

Eine Instance einer laufenden Anwendung. Die folgenden Arten von Umgebungen sind beim Cloud-Computing üblich:

- **Entwicklungsumgebung** – Eine Instance einer laufenden Anwendung, die nur dem Kernteam zur Verfügung steht, das für die Wartung der Anwendung verantwortlich ist. Entwicklungsumgebungen werden verwendet, um Änderungen zu testen, bevor sie in höhere Umgebungen übertragen werden. Diese Art von Umgebung wird manchmal als Testumgebung bezeichnet.
- **Niedrigere Umgebungen** – Alle Entwicklungsumgebungen für eine Anwendung, z. B. solche, die für erste Builds und Tests verwendet wurden.

- Produktionsumgebung – Eine Instance einer laufenden Anwendung, auf die Endbenutzer zugreifen können. In einer CI/CD Pipeline ist die Produktionsumgebung die letzte Bereitstellungsumgebung.
- Höhere Umgebungen – Alle Umgebungen, auf die auch andere Benutzer als das Kernentwicklungsteam zugreifen können. Dies kann eine Produktionsumgebung, Vorproduktionsumgebungen und Umgebungen für Benutzerakzeptanztests umfassen.

Epics

In der agilen Methodik sind dies funktionale Kategorien, die Ihnen helfen, Ihre Arbeit zu organisieren und zu priorisieren. Epics bieten eine allgemeine Beschreibung der Anforderungen und Implementierungsaufgaben. Zu den Sicherheitsepen AWS von CAF gehören beispielsweise Identitäts- und Zugriffsmanagement, Detektivkontrollen, Infrastruktursicherheit, Datenschutz und Reaktion auf Vorfälle. Weitere Informationen zu Epics in der AWS -Migrationsstrategie finden Sie im [Leitfaden zur Programm-Implementierung](#).

ERP

Siehe [Enterprise Resource Planning](#).

Explorative Datenanalyse (EDA)

Der Prozess der Analyse eines Datensatzes, um seine Hauptmerkmale zu verstehen. Sie sammeln oder aggregieren Daten und führen dann erste Untersuchungen durch, um Muster zu finden, Anomalien zu erkennen und Annahmen zu überprüfen. EDA wird durchgeführt, indem zusammenfassende Statistiken berechnet und Datenvisualisierungen erstellt werden.

F

Faktentabelle

Die zentrale Tabelle in einem [Sternschema](#). Sie speichert quantitative Daten über den Geschäftsbetrieb. In der Regel enthält eine Faktentabelle zwei Arten von Spalten: Spalten, die Kennzahlen enthalten, und Spalten, die einen Fremdschlüssel für eine Dimensionstabelle enthalten.

schnell scheitern

Eine Philosophie, die häufige und inkrementelle Tests verwendet, um den Entwicklungslebenszyklus zu verkürzen. Dies ist ein wichtiger Bestandteil eines agilen Ansatzes.

Grenze zur Fehlerisolierung

Dabei handelt es sich um eine Grenze AWS Cloud, z. B. eine Availability Zone AWS-Region, eine Steuerungsebene oder eine Datenebene, die die Auswirkungen eines Fehlers begrenzt und die Widerstandsfähigkeit von Workloads verbessert. Weitere Informationen finden Sie unter [Grenzen zur AWS Fehlerisolierung](#).

Feature-Zweig

Siehe [Zweig](#).

Features

Die Eingabedaten, die Sie verwenden, um eine Vorhersage zu treffen. In einem Fertigungskontext könnten Feature beispielsweise Bilder sein, die regelmäßig von der Fertigungslinie aus aufgenommen werden.

Bedeutung der Feature

Wie wichtig ein Feature für die Vorhersagen eines Modells ist. Dies wird in der Regel als numerischer Wert ausgedrückt, der mit verschiedenen Techniken wie Shapley Additive Explanations (SHAP) und integrierten Gradienten berechnet werden kann. Weitere Informationen finden Sie unter [Interpretierbarkeit von Modellen für maschinelles Lernen mit AWS](#).

Featuretransformation

Daten für den ML-Prozess optimieren, einschließlich der Anreicherung von Daten mit zusätzlichen Quellen, der Skalierung von Werten oder der Extraktion mehrerer Informationssätze aus einem einzigen Datenfeld. Das ermöglicht dem ML-Modell, von den Daten profitieren. Wenn Sie beispielsweise das Datum „27.05.2021 00:15:37“ in „2021“, „Mai“, „Donnerstag“ und „15“ aufschlüsseln, können Sie dem Lernalgorithmus helfen, nuancierte Muster zu erlernen, die mit verschiedenen Datenkomponenten verknüpft sind.

Eingabeaufforderung mit wenigen Klicks

Bereitstellung einer kleinen Anzahl von Beispielen, die die Aufgabe und das gewünschte Ergebnis veranschaulichen, bevor das [LLM](#) aufgefordert wird, eine ähnliche Aufgabe auszuführen. Bei dieser Technik handelt es sich um eine Anwendung des kontextbezogenen Lernens, bei der Modelle anhand von Beispielen (Aufnahmen) lernen, die in Eingabeaufforderungen eingebettet sind. Bei Aufgaben, die spezifische Formatierungs-, Argumentations- oder Fachkenntnisse erfordern, kann die Eingabeaufforderung mit wenigen Handgriffen effektiv sein. [Siehe auch Zero-Shot Prompting](#).

FGAC

Siehe [detaillierte Zugriffskontrolle](#).

Feinkörnige Zugriffskontrolle (FGAC)

Die Verwendung mehrerer Bedingungen, um eine Zugriffsanfrage zuzulassen oder abzulehnen.

Flash-Cut-Migration

Eine Datenbankmigrationsmethode, bei der eine kontinuierliche Datenreplikation durch [Erfassung von Änderungsdaten](#) verwendet wird, um Daten in kürzester Zeit zu migrieren, anstatt einen schrittweisen Ansatz zu verwenden. Ziel ist es, Ausfallzeiten auf ein Minimum zu beschränken.

FM

Siehe [Fundamentmodell](#).

Fundamentmodell (FM)

Ein großes neuronales Deep-Learning-Netzwerk, das mit riesigen Datensätzen generalisierter und unbeschrifteter Daten trainiert wurde. FMs sind in der Lage, eine Vielzahl allgemeiner Aufgaben zu erfüllen, z. B. Sprache zu verstehen, Text und Bilder zu generieren und Konversationen in natürlicher Sprache zu führen. Weitere Informationen finden Sie unter [Was sind Foundation-Modelle](#).

G

Generative KI

Eine Untergruppe von [KI-Modellen](#), die mit großen Datenmengen trainiert wurden und mit einer einfachen Textaufforderung neue Inhalte und Artefakte wie Bilder, Videos, Text und Audio erstellen können. Weitere Informationen finden Sie unter [Was ist Generative KI](#).

Geoblocking

Siehe [geografische Einschränkungen](#).

Geografische Einschränkungen (Geoblocking)

Bei Amazon eine Option CloudFront, um zu verhindern, dass Benutzer in bestimmten Ländern auf Inhaltsverteilungen zugreifen. Sie können eine Zulassungsliste oder eine Sperrliste verwenden,

um zugelassene und gesperrte Länder anzugeben. Weitere Informationen finden Sie in [der Dokumentation unter Beschränkung der geografischen Verteilung Ihrer Inhalte](#). CloudFront

Gitflow-Workflow

Ein Ansatz, bei dem niedrigere und höhere Umgebungen unterschiedliche Zweige in einem Quellcode-Repository verwenden. Der Gitflow-Workflow gilt als veraltet, und der [Trunk-basierte Workflow](#) ist der moderne, bevorzugte Ansatz.

goldenes Bild

Ein Snapshot eines Systems oder einer Software, der als Vorlage für die Bereitstellung neuer Instanzen dieses Systems oder dieser Software verwendet wird. In der Fertigung kann ein Golden Image beispielsweise zur Bereitstellung von Software auf mehreren Geräten verwendet werden und trägt zur Verbesserung der Geschwindigkeit, Skalierbarkeit und Produktivität bei der Geräteherstellung bei.

Greenfield-Strategie

Das Fehlen vorhandener Infrastruktur in einer neuen Umgebung. Bei der Einführung einer Neuausrichtung einer Systemarchitektur können Sie alle neuen Technologien ohne Einschränkung der Kompatibilität mit der vorhandenen Infrastruktur auswählen, auch bekannt als [Brownfield](#). Wenn Sie die bestehende Infrastruktur erweitern, könnten Sie Brownfield- und Greenfield-Strategien mischen.

Integritätsschutz

Eine allgemeine Regel, die dazu beiträgt, Ressourcen, Richtlinien und die Einhaltung von Vorschriften in allen Unternehmenseinheiten zu regeln (OUs). Präventiver Integritätsschutz setzt Richtlinien durch, um die Einhaltung von Standards zu gewährleisten. Sie werden mithilfe von Service-Kontrollrichtlinien und IAM-Berechtigungsgrößen implementiert. Detektivischer Integritätsschutz erkennt Richtlinienverstöße und Compliance-Probleme und generiert Warnmeldungen zur Abhilfe. Sie werden mithilfe von AWS Config, AWS Security Hub CSPM, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector und benutzerdefinierten AWS Lambda Prüfungen implementiert.

H

HEKTAR

Siehe [Hochverfügbarkeit](#).

Heterogene Datenbankmigration

Migrieren Sie Ihre Quelldatenbank in eine Zieldatenbank, die eine andere Datenbank-Engine verwendet (z. B. Oracle zu Amazon Aurora). Eine heterogene Migration ist in der Regel Teil einer Neuarchitektur, und die Konvertierung des Schemas kann eine komplexe Aufgabe sein. [AWS bietet AWS SCT](#), welches bei Schemakonvertierungen hilft.

hohe Verfügbarkeit (HA)

Die Fähigkeit eines Workloads, im Falle von Herausforderungen oder Katastrophen kontinuierlich und ohne Eingreifen zu arbeiten. HA-Systeme sind so konzipiert, dass sie automatisch ein Failover durchführen, gleichbleibend hohe Leistung bieten und unterschiedliche Lasten und Ausfälle mit minimalen Leistungseinbußen bewältigen.

historische Modernisierung

Ein Ansatz zur Modernisierung und Aufrüstung von Betriebstechnologiesystemen (OT), um den Bedürfnissen der Fertigungsindustrie besser gerecht zu werden. Ein Historian ist eine Art von Datenbank, die verwendet wird, um Daten aus verschiedenen Quellen in einer Fabrik zu sammeln und zu speichern.

Daten zurückhalten

Ein Teil historischer, beschrifteter Daten, der aus einem Datensatz zurückgehalten wird, der zum Trainieren eines Modells für [maschinelles](#) Lernen verwendet wird. Sie können Holdout-Daten verwenden, um die Modellleistung zu bewerten, indem Sie die Modellvorhersagen mit den Holdout-Daten vergleichen.

Homogene Datenbankmigration

Migrieren Sie Ihre Quelldatenbank zu einer Zieldatenbank, die dieselbe Datenbank-Engine verwendet (z. B. Microsoft SQL Server zu Amazon RDS für SQL Server). Eine homogene Migration ist in der Regel Teil eines Hostwechsels oder eines Plattformwechsels. Sie können native Datenbankserviceprogramme verwenden, um das Schema zu migrieren.

heiße Daten

Daten, auf die häufig zugegriffen wird, z. B. Echtzeitdaten oder aktuelle Transaktionsdaten. Für diese Daten ist in der Regel eine leistungsstarke Speicherebene oder -klasse erforderlich, um schnelle Abfrageantworten zu ermöglichen.

Hotfix

Eine dringende Lösung für ein kritisches Problem in einer Produktionsumgebung. Aufgrund seiner Dringlichkeit wird ein Hotfix normalerweise außerhalb des typischen DevOps Release-Workflows erstellt.

Hypercare-Phase

Unmittelbar nach dem Cutover, der Zeitraum, in dem ein Migrationsteam die migrierten Anwendungen in der Cloud verwaltet und überwacht, um etwaige Probleme zu beheben. In der Regel dauert dieser Zeitraum 1–4 Tage. Am Ende der Hypercare-Phase überträgt das Migrationsteam in der Regel die Verantwortung für die Anwendungen an das Cloud-Betriebsteam.

I

IaC

Sehen Sie [Infrastruktur als Code](#).

Identitätsbasierte Richtlinie

Eine Richtlinie, die einem oder mehreren IAM-Prinzipalen zugeordnet ist und deren Berechtigungen innerhalb der AWS Cloud Umgebung definiert.

Leerlaufanwendung

Eine Anwendung mit einer durchschnittlichen CPU- und Arbeitsspeicherauslastung zwischen 5 und 20 Prozent über einen Zeitraum von 90 Tagen. In einem Migrationsprojekt ist es üblich, diese Anwendungen außer Betrieb zu nehmen oder sie On-Premises beizubehalten.

IloT

Siehe [Industrielles Internet der Dinge](#).

unveränderliche Infrastruktur

Ein Modell, das eine neue Infrastruktur für Produktionsworkloads bereitstellt, anstatt die bestehende Infrastruktur zu aktualisieren, zu patchen oder zu modifizieren. [Unveränderliche Infrastrukturen sind von Natur aus konsistenter, zuverlässiger und vorhersehbarer als veränderliche Infrastrukturen](#). Weitere Informationen finden Sie in der Best Practice [Deploy using immutable infrastructure](#) im AWS Well-Architected Framework.

Eingehende (ingress) VPC

In einer Architektur AWS mit mehreren Konten ist dies eine VPC, die Netzwerkverbindungen von außerhalb einer Anwendung akzeptiert, überprüft und weiterleitet. Die [AWS Security Reference Architecture](#) empfiehlt, Ihr Netzwerkkonto mit eingehendem und ausgehendem Datenverkehr und Inspektion einzurichten, VPCs um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet im weiteren Sinne zu schützen.

Inkrementelle Migration

Eine Cutover-Strategie, bei der Sie Ihre Anwendung in kleinen Teilen migrieren, anstatt eine einziges vollständiges Cutover durchzuführen. Beispielsweise könnten Sie zunächst nur einige Microservices oder Benutzer auf das neue System umstellen. Nachdem Sie sich vergewissert haben, dass alles ordnungsgemäß funktioniert, können Sie weitere Microservices oder Benutzer schrittweise verschieben, bis Sie Ihr Legacy-System außer Betrieb nehmen können. Diese Strategie reduziert die mit großen Migrationen verbundenen Risiken.

Industrie 4.0

Ein Begriff, der 2016 von [Klaus Schwab](#) eingeführt wurde und sich auf die Modernisierung von Fertigungsprozessen durch Fortschritte in den Bereichen Konnektivität, Echtzeitdaten, Automatisierung, Analytik und KI/ML bezieht.

Infrastruktur

Alle Ressourcen und Komponenten, die in der Umgebung einer Anwendung enthalten sind.

Infrastructure as Code (IaC)

Der Prozess der Bereitstellung und Verwaltung der Infrastruktur einer Anwendung mithilfe einer Reihe von Konfigurationsdateien. IaC soll Ihnen helfen, das Infrastrukturmanagement zu zentralisieren, Ressourcen zu standardisieren und schnell zu skalieren, sodass neue Umgebungen wiederholbar, zuverlässig und konsistent sind.

industrielles Internet der Dinge (T) Ilo

Einsatz von mit dem Internet verbundenen Sensoren und Geräten in Industriesektoren wie Fertigung, Energie, Automobilindustrie, Gesundheitswesen, Biowissenschaften und Landwirtschaft. Weitere Informationen finden Sie unter [Aufbau einer digitalen Transformationsstrategie für das industrielle Internet der Dinge \(IIoT\)](#).

Inspektions-VPC

In einer Architektur AWS mit mehreren Konten eine zentralisierte VPC, die Inspektionen des Netzwerkverkehrs zwischen VPCs (in demselben oder unterschiedlichen AWS-Regionen), dem Internet und lokalen Netzwerken verwaltet. In der [AWS Security Reference Architecture](#) wird empfohlen, Ihr Netzwerkkonto mit eingehendem und ausgehendem Datenverkehr sowie Inspektionen einzurichten, VPCs um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet im weiteren Sinne zu schützen.

Internet of Things (IoT)

Das Netzwerk verbundener physischer Objekte mit eingebetteten Sensoren oder Prozessoren, das über das Internet oder über ein lokales Kommunikationsnetzwerk mit anderen Geräten und Systemen kommuniziert. Weitere Informationen finden Sie unter [Was ist IoT?](#)

Interpretierbarkeit

Ein Merkmal eines Modells für Machine Learning, das beschreibt, inwieweit ein Mensch verstehen kann, wie die Vorhersagen des Modells von seinen Eingaben abhängen. Weitere Informationen finden Sie unter Interpretierbarkeit des [Modells für maschinelles Lernen](#) mit AWS

IoT

Siehe [Internet der Dinge](#).

IT information library (ITIL, IT-Informationsbibliothek)

Eine Reihe von bewährten Methoden für die Bereitstellung von IT-Services und die Abstimmung dieser Services auf die Geschäftsanforderungen. ITIL bietet die Grundlage für ITSM.

T service management (ITSM, IT-Servicemanagement)

Aktivitäten im Zusammenhang mit der Gestaltung, Implementierung, Verwaltung und Unterstützung von IT-Services für eine Organisation. Informationen zur Integration von Cloud-Vorgängen mit ITSM-Tools finden Sie im [Leitfaden zur Betriebsintegration](#).

BIS

Siehe [IT-Informationsbibliothek](#).

ITSM

Siehe [IT-Servicemanagement](#).

L

Labelbasierte Zugangskontrolle (LBAC)

Eine Implementierung der Mandatory Access Control (MAC), bei der den Benutzern und den Daten selbst jeweils explizit ein Sicherheitslabelwert zugewiesen wird. Die Schnittmenge zwischen der Benutzersicherheitsbeschriftung und der Datensicherheitsbeschriftung bestimmt, welche Zeilen und Spalten für den Benutzer sichtbar sind.

Landing Zone

Eine landing zone ist eine gut strukturierte AWS Umgebung mit mehreren Konten, die skalierbar und sicher ist. Dies ist ein Ausgangspunkt, von dem aus Ihre Organisationen Workloads und Anwendungen schnell und mit Vertrauen in ihre Sicherheits- und Infrastrukturmgebung starten und bereitstellen können. Weitere Informationen zu Landing Zones finden Sie unter [Einrichtung einer sicheren und skalierbaren AWS -Umgebung mit mehreren Konten..](#)

großes Sprachmodell (LLM)

Ein [Deep-Learning-KI-Modell](#), das anhand einer riesigen Datenmenge vorab trainiert wurde. Ein LLM kann mehrere Aufgaben ausführen, z. B. Fragen beantworten, Dokumente zusammenfassen, Text in andere Sprachen übersetzen und Sätze vervollständigen. [Weitere Informationen finden Sie unter Was sind LLMs](#)

Große Migration

Eine Migration von 300 oder mehr Servern.

SCHWARZ

Siehe [Labelbasierte Zugriffskontrolle](#).

Geringste Berechtigung

Die bewährte Sicherheitsmethode, bei der nur die für die Durchführung einer Aufgabe erforderlichen Mindestberechtigungen erteilt werden. Weitere Informationen finden Sie unter [Geringste Berechtigungen anwenden](#) in der IAM-Dokumentation.

Lift and Shift

Siehe [7 Rs](#).

Little-Endian-System

Ein System, welches das niedrigwertigste Byte zuerst speichert. Siehe auch [Endianness](#).

LLM

Siehe [großes Sprachmodell](#).

Niedrigere Umgebungen

Siehe [Umgebung](#).

M

Machine Learning (ML)

Eine Art künstlicher Intelligenz, die Algorithmen und Techniken zur Mustererkennung und zum Lernen verwendet. ML analysiert aufgezeichnete Daten, wie z. B. Daten aus dem Internet der Dinge (IoT), und lernt daraus, um ein statistisches Modell auf der Grundlage von Mustern zu erstellen. Weitere Informationen finden Sie unter [Machine Learning](#).

Hauptzweig

Siehe [Filiale](#).

Malware

Software, die entwickelt wurde, um die Computersicherheit oder den Datenschutz zu gefährden. Malware kann Computersysteme stören, vertrauliche Informationen durchsickern lassen oder sich unbefugten Zugriff verschaffen. Beispiele für Malware sind Viren, Würmer, Ransomware, Trojaner, Spyware und Keylogger.

verwaltete Dienste

AWS-Services für die die Infrastrukturebene, das Betriebssystem und die Plattformen AWS betrieben werden, und Sie greifen auf die Endgeräte zu, um Daten zu speichern und abzurufen. Amazon Simple Storage Service (Amazon S3) und Amazon DynamoDB sind Beispiele für Managed Services. Diese werden auch als abstrakte Dienste bezeichnet.

Manufacturing Execution System (MES)

Ein Softwaresystem zur Verfolgung, Überwachung, Dokumentation und Steuerung von Produktionsprozessen, bei denen Rohstoffe in der Fertigung zu fertigen Produkten umgewandelt werden.

MAP

Siehe [Migration Acceleration Program](#).

Mechanismus

Ein vollständiger Prozess, bei dem Sie ein Tool erstellen, die Akzeptanz des Tools vorantreiben und anschließend die Ergebnisse überprüfen, um Anpassungen vorzunehmen. Ein Mechanismus ist ein Zyklus, der sich im Laufe seiner Tätigkeit selbst verstärkt und verbessert. Weitere Informationen finden Sie unter [Aufbau von Mechanismen](#) im AWS Well-Architected Framework.

Mitgliedskonto

Alle AWS-Konten außer dem Verwaltungskonto, die Teil einer Organisation sind. AWS Organizations Ein Konto kann jeweils nur Mitglied einer Organisation sein.

MES

Siehe [Manufacturing Execution System](#).

Message Queuing-Telemetrietransport (MQTT)

[Ein leichtes machine-to-machine \(M2M\) -Kommunikationsprotokoll, das auf dem Publish/Subscribe-Muster für IoT-Geräte mit beschränkten Ressourcen basiert.](#)

Microservice

Ein kleiner, unabhängiger Dienst, der über genau definierte Kanäle kommuniziert APIs und in der Regel kleinen, eigenständigen Teams gehört. Ein Versicherungssystem kann beispielsweise Microservices beinhalten, die Geschäftsfunktionen wie Vertrieb oder Marketing oder Subdomains wie Einkauf, Schadenersatz oder Analytik zugeordnet sind. Zu den Vorteilen von Microservices gehören Agilität, flexible Skalierung, einfache Bereitstellung, wiederverwendbarer Code und Ausfallsicherheit. Weitere Informationen finden Sie unter [Integration von Microservices mithilfe serverloser Dienste](#). AWS

Microservices-Architekturen

Ein Ansatz zur Erstellung einer Anwendung mit unabhängigen Komponenten, die jeden Anwendungsprozess als Microservice ausführen. Diese Microservices kommunizieren mithilfe von Lightweight über eine klar definierte Schnittstelle. APIs Jeder Microservice in dieser Architektur kann aktualisiert, bereitgestellt und skaliert werden, um den Bedarf an bestimmten Funktionen einer Anwendung zu decken. Weitere Informationen finden Sie unter [Implementierung von Microservices](#) auf. AWS

Migration Acceleration Program (MAP)

Ein AWS Programm, das Beratung, Unterstützung, Schulungen und Services bietet, um Unternehmen dabei zu unterstützen, eine solide betriebliche Grundlage für die Umstellung auf

die Cloud zu schaffen und die anfänglichen Kosten von Migrationen auszugleichen. MAP umfasst eine Migrationsmethode für die methodische Durchführung von Legacy-Migrationen sowie eine Reihe von Tools zur Automatisierung und Beschleunigung gängiger Migrationsszenarien.

Migration in großem Maßstab

Der Prozess, bei dem der Großteil des Anwendungsportfolios in Wellen in die Cloud verlagert wird, wobei in jeder Welle mehr Anwendungen schneller migriert werden. In dieser Phase werden die bewährten Verfahren und Erkenntnisse aus den früheren Phasen zur Implementierung einer Migrationsfabrik von Teams, Tools und Prozessen zur Optimierung der Migration von Workloads durch Automatisierung und agile Bereitstellung verwendet. Dies ist die dritte Phase der [AWS - Migrationsstrategie](#).

Migrationsfabrik

Funktionsübergreifende Teams, die die Migration von Workloads durch automatisierte, agile Ansätze optimieren. Zu den Teams in der Migrationsabteilung gehören in der Regel Betriebsabläufe, Geschäftsanalysten und Eigentümer, Migrationsingenieure, Entwickler und DevOps Experten, die in Sprints arbeiten. Zwischen 20 und 50 Prozent eines Unternehmensanwendungsportfolios bestehen aus sich wiederholenden Mustern, die durch einen Fabrik-Ansatz optimiert werden können. Weitere Informationen finden Sie in [Diskussion über Migrationsfabriken](#) und den [Leitfaden zur Cloud-Migration-Fabrik](#) in diesem Inhaltssatz.

Migrationsmetadaten

Die Informationen über die Anwendung und den Server, die für den Abschluss der Migration benötigt werden. Für jedes Migrationsmuster ist ein anderer Satz von Migrationsmetadaten erforderlich. Beispiele für Migrationsmetadaten sind das Zielsubnetz, die Sicherheitsgruppe und AWS das Konto.

Migrationsmuster

Eine wiederholbare Migrationsaufgabe, in der die Migrationsstrategie, das Migrationsziel und die verwendete Migrationsanwendung oder der verwendete Migrationsservice detailliert beschrieben werden. Beispiel: Rehost-Migration zu Amazon EC2 mit AWS Application Migration Service.

Migration Portfolio Assessment (MPA)

Ein Online-Tool, das Informationen zur Validierung des Geschäftsszenarios für die Migration auf das bereitstellt. AWS Cloud MPA bietet eine detaillierte Portfoliobewertung (richtige Servergröße, Preisgestaltung, Gesamtbetriebskostenanalyse, Migrationskostenanalyse) sowie Migrationsplanung (Anwendungsdatenanalyse und Datenerfassung, Anwendungsgruppierung,

Migrationspriorisierung und Wellenplanung). Das [MPA-Tool](#) (Anmeldung erforderlich) steht allen AWS Beratern und APN-Partnerberatern kostenlos zur Verfügung.

Migration Readiness Assessment (MRA)

Der Prozess, bei dem mithilfe des AWS CAF Erkenntnisse über den Cloud-Bereitschaftsstatus eines Unternehmens gewonnen, Stärken und Schwächen identifiziert und ein Aktionsplan zur Schließung festgestellter Lücken erstellt wird. Weitere Informationen finden Sie im [Benutzerhandbuch für Migration Readiness](#). MRA ist die erste Phase der [AWS - Migrationsstrategie](#).

Migrationsstrategie

Der Ansatz, der verwendet wurde, um einen Workload auf den AWS Cloud zu migrieren. Weitere Informationen finden Sie im Eintrag [7 Rs](#) in diesem Glossar und unter [Mobilisieren Sie Ihr Unternehmen, um umfangreiche Migrationen zu beschleunigen](#).

ML

[Siehe maschinelles Lernen.](#)

Modernisierung

Umwandlung einer veralteten (veralteten oder monolithischen) Anwendung und ihrer Infrastruktur in ein agiles, elastisches und hochverfügbares System in der Cloud, um Kosten zu senken, die Effizienz zu steigern und Innovationen zu nutzen. Weitere Informationen finden Sie unter [Strategie zur Modernisierung von Anwendungen in der AWS Cloud](#).

Bewertung der Modernisierungsfähigkeit

Eine Bewertung, anhand derer festgestellt werden kann, ob die Anwendungen einer Organisation für die Modernisierung bereit sind, Vorteile, Risiken und Abhängigkeiten identifiziert und ermittelt wird, wie gut die Organisation den zukünftigen Status dieser Anwendungen unterstützen kann. Das Ergebnis der Bewertung ist eine Vorlage der Zielarchitektur, eine Roadmap, in der die Entwicklungsphasen und Meilensteine des Modernisierungsprozesses detailliert beschrieben werden, sowie ein Aktionsplan zur Behebung festgestellter Lücken. Weitere Informationen finden Sie unter [Evaluierung der Modernisierungsbereitschaft von Anwendungen in der AWS Cloud](#).

Monolithische Anwendungen (Monolithen)

Anwendungen, die als ein einziger Service mit eng gekoppelten Prozessen ausgeführt werden. Monolithische Anwendungen haben verschiedene Nachteile. Wenn ein Anwendungs-Feature stark nachgefragt wird, muss die gesamte Architektur skaliert werden. Das Hinzufügen oder

Verbessern der Feature einer monolithischen Anwendung wird ebenfalls komplexer, wenn die Codebasis wächst. Um diese Probleme zu beheben, können Sie eine Microservices-Architektur verwenden. Weitere Informationen finden Sie unter [Zerlegen von Monolithen in Microservices](#).

MPA

Siehe [Bewertung des Migrationsportfolios](#).

MQTT

Siehe [Message Queuing-Telemetrietransport](#).

Mehrklassen-Klassifizierung

Ein Prozess, der dabei hilft, Vorhersagen für mehrere Klassen zu generieren (wobei eines von mehr als zwei Ergebnissen vorhergesagt wird). Ein ML-Modell könnte beispielsweise fragen: „Ist dieses Produkt ein Buch, ein Auto oder ein Telefon?“ oder „Welche Kategorie von Produkten ist für diesen Kunden am interessantesten?“

veränderbare Infrastruktur

Ein Modell, das die bestehende Infrastruktur für Produktionsworkloads aktualisiert und modifiziert. Für eine verbesserte Konsistenz, Zuverlässigkeit und Vorhersagbarkeit empfiehlt das AWS Well-Architected Framework die Verwendung einer [unveränderlichen Infrastruktur](#) als bewährte Methode.

O

OAC

Siehe [Origin Access Control](#).

EICHE

Siehe [Zugriffsidentität von Origin](#).

COM

Siehe [organisatorisches Change-Management](#).

Offline-Migration

Eine Migrationsmethode, bei der der Quell-Workload während des Migrationsprozesses heruntergefahren wird. Diese Methode ist mit längeren Ausfallzeiten verbunden und wird in der Regel für kleine, unkritische Workloads verwendet.

OI

Siehe [Betriebsintegration](#).

OLA

Siehe Vereinbarung auf [operativer Ebene](#).

Online-Migration

Eine Migrationsmethode, bei der der Quell-Workload auf das Zielsystem kopiert wird, ohne offline genommen zu werden. Anwendungen, die mit dem Workload verbunden sind, können während der Migration weiterhin funktionieren. Diese Methode beinhaltet keine bis minimale Ausfallzeit und wird in der Regel für kritische Produktionsworkloads verwendet.

OPC-UA

Siehe [Open Process Communications — Unified Architecture](#).

Offene Prozesskommunikation — Einheitliche Architektur (OPC-UA)

Ein machine-to-machine (M2M) -Kommunikationsprotokoll für die industrielle Automatisierung. OPC-UA bietet einen Interoperabilitätsstandard mit Datenverschlüsselungs-, Authentifizierungs- und Autorisierungsschemata.

Vereinbarung auf Betriebsebene (OLA)

Eine Vereinbarung, in der klargestellt wird, welche funktionalen IT-Gruppen sich gegenseitig versprechen zu liefern, um ein Service Level Agreement (SLA) zu unterstützen.

Überprüfung der Betriebsbereitschaft (ORR)

Eine Checkliste mit Fragen und zugehörigen bewährten Methoden, die Ihnen helfen, Vorfälle und mögliche Ausfälle zu verstehen, zu bewerten, zu verhindern oder deren Umfang zu reduzieren. Weitere Informationen finden Sie unter [Operational Readiness Reviews \(ORR\)](#) im AWS Well-Architected Framework.

Betriebstechnologie (OT)

Hardware- und Softwaresysteme, die mit der physischen Umgebung zusammenarbeiten, um industrielle Abläufe, Ausrüstung und Infrastruktur zu steuern. In der Fertigung ist die Integration von OT- und Informationstechnologie (IT) -Systemen ein zentraler Schwerpunkt der [Industrie 4.0-Transformationen](#).

Betriebsintegration (OI)

Der Prozess der Modernisierung von Abläufen in der Cloud, der Bereitschaftsplanung, Automatisierung und Integration umfasst. Weitere Informationen finden Sie im [Leitfaden zur Betriebsintegration](#).

Organisationspfad

Ein Pfad, der von erstellt wird und in AWS CloudTrail dem alle Ereignisse für alle AWS-Konten in einer Organisation protokolliert werden. AWS Organizations Diese Spur wird in jedem AWS-Konto , der Teil der Organisation ist, erstellt und verfolgt die Aktivität in jedem Konto. Weitere Informationen finden Sie in der CloudTrail Dokumentation unter [Einen Trail für eine Organisation erstellen](#).

Organisatorisches Veränderungsmanagement (OCM)

Ein Framework für das Management wichtiger, disruptiver Geschäftstransformationen aus Sicht der Mitarbeiter, der Kultur und der Führung. OCM hilft Organisationen dabei, sich auf neue Systeme und Strategien vorzubereiten und auf diese umzustellen, indem es die Akzeptanz von Veränderungen beschleunigt, Übergangsprobleme angeht und kulturelle und organisatorische Veränderungen vorantreibt. In der AWS Migrationsstrategie wird dieses Framework aufgrund der Geschwindigkeit des Wandels, der bei Projekten zur Cloud-Einführung erforderlich ist, als Mitarbeiterbeschleunigung bezeichnet. Weitere Informationen finden Sie im [OCM-Handbuch](#).

Ursprungszugriffskontrolle (OAC)

In CloudFront, eine erweiterte Option zur Zugriffsbeschränkung, um Ihre Amazon Simple Storage Service (Amazon S3) -Inhalte zu sichern. OAC unterstützt alle S3-Buckets insgesamt AWS-Regionen, serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) sowie dynamische PUT und DELETE Anfragen an den S3-Bucket.

Ursprungszugriffsidentität (OAI)

In CloudFront, eine Option zur Zugriffsbeschränkung, um Ihre Amazon S3 S3-Inhalte zu sichern. Wenn Sie OAI verwenden, CloudFront erstellt es einen Principal, mit dem sich Amazon S3 authentifizieren kann. Authentifizierte Principals können nur über eine bestimmte Distribution auf Inhalte in einem S3-Bucket zugreifen. CloudFront Siehe auch [OAC](#), das eine detailliertere und verbesserte Zugriffskontrolle bietet.

ORR

Weitere Informationen finden Sie unter [Überprüfung der Betriebsbereitschaft](#).

NICHT

Siehe [Betriebstechnologie](#).

Ausgehende (egress) VPC

In einer Architektur AWS mit mehreren Konten eine VPC, die Netzwerkverbindungen verarbeitet, die von einer Anwendung aus initiiert werden. Die [AWS Security Reference Architecture](#) empfiehlt die Einrichtung Ihres Netzwerkkontos mit eingehendem und ausgehendem Datenverkehr sowie Inspektion, VPCs um die bidirektionale Schnittstelle zwischen Ihrer Anwendung und dem Internet im weiteren Sinne zu schützen.

P

Berechtigungsgrenze

Eine IAM-Verwaltungsrichtlinie, die den IAM-Prinzipalen zugeordnet ist, um die maximalen Berechtigungen festzulegen, die der Benutzer oder die Rolle haben kann. Weitere Informationen finden Sie unter [Berechtigungsgrenzen](#) für IAM-Entitys in der IAM-Dokumentation.

persönlich identifizierbare Informationen (PII)

Informationen, die, wenn sie direkt betrachtet oder mit anderen verwandten Daten kombiniert werden, verwendet werden können, um vernünftige Rückschlüsse auf die Identität einer Person zu ziehen. Beispiele für personenbezogene Daten sind Namen, Adressen und Kontaktinformationen.

Personenbezogene Daten

Siehe [persönlich identifizierbare Informationen](#).

Playbook

Eine Reihe vordefinierter Schritte, die die mit Migrationen verbundenen Aufgaben erfassen, z. B. die Bereitstellung zentraler Betriebsfunktionen in der Cloud. Ein Playbook kann die Form von Skripten, automatisierten Runbooks oder einer Zusammenfassung der Prozesse oder Schritte annehmen, die für den Betrieb Ihrer modernisierten Umgebung erforderlich sind.

PLC

Siehe [programmierbare Logiksteuerung](#).

PLM

Siehe [Produktlebenszyklusmanagement](#).

policy

Ein Objekt, das Berechtigungen definieren (siehe [identitätsbasierte Richtlinie](#)), Zugriffsbedingungen spezifizieren (siehe [ressourcenbasierte Richtlinie](#)) oder die maximalen Berechtigungen für alle Konten in einer Organisation definieren kann AWS Organizations (siehe [Dienststeuerungsrichtlinie](#)).

Polyglotte Beharrlichkeit

Unabhängige Auswahl der Datenspeichertechnologie eines Microservices auf der Grundlage von Datenzugriffsmustern und anderen Anforderungen. Wenn Ihre Microservices über dieselbe Datenspeichertechnologie verfügen, kann dies zu Implementierungsproblemen oder zu Leistungseinbußen führen. Microservices lassen sich leichter implementieren und erzielen eine bessere Leistung und Skalierbarkeit, wenn sie den Datenspeicher verwenden, der ihren Anforderungen am besten entspricht.

Portfoliobewertung

Ein Prozess, bei dem das Anwendungsportfolio ermittelt, analysiert und priorisiert wird, um die Migration zu planen. Weitere Informationen finden Sie in [Bewerten der Migrationsbereitschaft](#).

predicate

Eine Abfragebedingung, die `true` oder zurückgibt `false`, was üblicherweise in einer Klausel vorkommt. WHERE

Prädikat Pushdown

Eine Technik zur Optimierung von Datenbankabfragen, bei der die Daten in der Abfrage vor der Übertragung gefiltert werden. Dadurch wird die Datenmenge reduziert, die aus der relationalen Datenbank abgerufen und verarbeitet werden muss, und die Abfrageleistung wird verbessert.

Präventive Kontrolle

Eine Sicherheitskontrolle, die verhindern soll, dass ein Ereignis eintritt. Diese Kontrollen stellen eine erste Verteidigungslinie dar, um unbefugten Zugriff oder unerwünschte Änderungen an Ihrem Netzwerk zu verhindern. Weitere Informationen finden Sie unter [Präventive Kontrolle](#) in Implementierung von Sicherheitskontrollen in AWS.

Prinzipal

Eine Entität AWS, die Aktionen ausführen und auf Ressourcen zugreifen kann. Diese Entität ist in der Regel ein Root-Benutzer für eine AWS-Konto, eine IAM-Rolle oder einen Benutzer. Weitere Informationen finden Sie unter Prinzipal in [Rollenbegriffe und -konzepte](#) in der IAM-Dokumentation.

Datenschutz von Natur aus

Ein systemtechnischer Ansatz, der den Datenschutz während des gesamten Entwicklungsprozesses berücksichtigt.

Privat gehostete Zonen

Ein Container, der Informationen darüber enthält, wie Amazon Route 53 auf DNS-Abfragen für eine Domain und deren Subdomains innerhalb einer oder mehrerer VPCs Domains antworten soll. Weitere Informationen finden Sie unter [Arbeiten mit privat gehosteten Zonen](#) in der Route-53-Dokumentation.

proaktive Steuerung

Eine [Sicherheitskontrolle](#), die den Einsatz nicht richtlinienkonformer Ressourcen verhindern soll. Diese Steuerelemente scannen Ressourcen, bevor sie bereitgestellt werden. Wenn die Ressource nicht der Kontrolle entspricht, wird sie nicht bereitgestellt. Weitere Informationen finden Sie im [Referenzhandbuch zu Kontrollen](#) in der AWS Control Tower Dokumentation und unter [Proaktive Kontrollen](#) unter Implementierung von Sicherheitskontrollen am AWS.

Produktlebenszyklusmanagement (PLM)

Das Management von Daten und Prozessen für ein Produkt während seines gesamten Lebenszyklus, vom Design, der Entwicklung und Markteinführung über Wachstum und Reife bis hin zur Markteinführung und Markteinführung.

Produktionsumgebung

Siehe [Umgebung](#).

Speicherprogrammierbare Steuerung (SPS)

In der Fertigung ein äußerst zuverlässiger, anpassungsfähiger Computer, der Maschinen überwacht und Fertigungsprozesse automatisiert.

schnelle Verkettung

Verwendung der Ausgabe einer [LLM-Eingabeaufforderung](#) als Eingabe für die nächste Aufforderung, um bessere Antworten zu generieren. Diese Technik wird verwendet, um eine komplexe Aufgabe in Unteraufgaben zu unterteilen oder um eine vorläufige Antwort iterativ zu verfeinern oder zu erweitern. Sie trägt dazu bei, die Genauigkeit und Relevanz der Antworten eines Modells zu verbessern und ermöglicht detailliertere, personalisierte Ergebnisse.

Pseudonymisierung

Der Prozess, bei dem persönliche Identifikatoren in einem Datensatz durch Platzhalterwerte ersetzt werden. Pseudonymisierung kann zum Schutz der Privatsphäre beitragen. Pseudonymisierte Daten gelten weiterhin als personenbezogene Daten.

publish/subscribe (pub/sub)

Ein Muster, das asynchrone Kommunikation zwischen Microservices ermöglicht, um die Skalierbarkeit und Reaktionsfähigkeit zu verbessern. In einem auf Microservices basierenden [MES](#) kann ein Microservice beispielsweise Ereignismeldungen in einem Kanal veröffentlichen, den andere Microservices abonnieren können. Das System kann neue Microservices hinzufügen, ohne den Veröffentlichungsservice zu ändern.

Q

Abfrageplan

Eine Reihe von Schritten, wie Anweisungen, die für den Zugriff auf die Daten in einem relationalen SQL-Datenbanksystem verwendet werden.

Abfrageplanregression

Wenn ein Datenbankserviceoptimierer einen weniger optimalen Plan wählt als vor einer bestimmten Änderung der Datenbankumgebung. Dies kann durch Änderungen an Statistiken, Beschränkungen, Umgebungseinstellungen, Abfrageparameter-Bindungen und Aktualisierungen der Datenbank-Engine verursacht werden.

R

RACI-Matrix

Siehe [verantwortlich, rechenschaftspflichtig, konsultiert, informiert \(RACI\)](#).

RAG

Siehe Erweiterte [Generierung beim Abrufen](#).

Ransomware

Eine bösartige Software, die entwickelt wurde, um den Zugriff auf ein Computersystem oder Daten zu blockieren, bis eine Zahlung erfolgt ist.

RASCI-Matrix

Siehe [verantwortlich, rechenschaftspflichtig, konsultiert, informiert \(RACI\)](#).

RCAC

Siehe [Zugriffskontrolle für Zeilen und Spalten](#).

Read Replica

Eine Kopie einer Datenbank, die nur für Lesezwecke verwendet wird. Sie können Abfragen an das Lesereplikat weiterleiten, um die Belastung auf Ihrer Primärdatenbank zu reduzieren.

neu strukturieren

Siehe [7 Rs](#).

Recovery Point Objective (RPO)

Die maximal zulässige Zeitspanne seit dem letzten Datenwiederherstellungspunkt. Damit wird festgelegt, was als akzeptabler Datenverlust zwischen dem letzten Wiederherstellungspunkt und der Serviceunterbrechung gilt.

Wiederherstellungszeitziel (RTO)

Die maximal zulässige Verzögerung zwischen der Betriebsunterbrechung und der Wiederherstellung des Dienstes.

Refaktorisierung

Siehe [7 Rs](#).

Region

Eine Sammlung von AWS Ressourcen in einem geografischen Gebiet. Jeder AWS-Region ist isoliert und unabhängig von den anderen, um Fehlertoleranz, Stabilität und Belastbarkeit zu gewährleisten. Weitere Informationen finden [Sie unter Geben Sie an, was AWS-Regionen Ihr Konto verwenden kann.](#)

Regression

Eine ML-Technik, die einen numerischen Wert vorhersagt. Zum Beispiel, um das Problem „Zu welchem Preis wird dieses Haus verkauft werden?“ zu lösen Ein ML-Modell könnte ein lineares Regressionsmodell verwenden, um den Verkaufspreis eines Hauses auf der Grundlage bekannter Fakten über das Haus (z. B. die Quadratmeterzahl) vorherzusagen.

rehosten

Siehe [7 Rs.](#)

Veröffentlichung

In einem Bereitstellungsprozess der Akt der Förderung von Änderungen an einer Produktionsumgebung.

umziehen

Siehe [7 Rs.](#)

neue Plattform

Siehe [7 Rs.](#)

Rückkauf

Siehe [7 Rs.](#)

Ausfallsicherheit

Die Fähigkeit einer Anwendung, Störungen zu widerstehen oder sich von ihnen zu erholen. [Hochverfügbarkeit](#) und [Notfallwiederherstellung](#) sind häufig Überlegungen bei der Planung der Ausfallsicherheit in der. AWS Cloud Weitere Informationen finden Sie unter [AWS Cloud Resilienz](#).

Ressourcenbasierte Richtlinie

Eine mit einer Ressource verknüpfte Richtlinie, z. B. ein Amazon-S3-Bucket, ein Endpunkt oder ein Verschlüsselungsschlüssel. Diese Art von Richtlinie legt fest, welchen Prinzipalen der Zugriff gewährt wird, welche Aktionen unterstützt werden und welche anderen Bedingungen erfüllt sein müssen.

RACI-Matrix (verantwortlich, rechenschaftspflichtig, konsultiert, informiert)

Eine Matrix, die die Rollen und Verantwortlichkeiten aller an Migrationsaktivitäten und Cloud-Operationen beteiligten Parteien definiert. Der Matrixname leitet sich von den in der Matrix definierten Zuständigkeitstypen ab: verantwortlich (R), rechenschaftspflichtig (A), konsultiert (C) und informiert (I). Der Unterstützungstyp (S) ist optional. Wenn Sie Unterstützung einbeziehen, wird die Matrix als RASCI-Matrix bezeichnet, und wenn Sie sie ausschließen, wird sie als RACI-Matrix bezeichnet.

Reaktive Kontrolle

Eine Sicherheitskontrolle, die darauf ausgelegt ist, die Behebung unerwünschter Ereignisse oder Abweichungen von Ihren Sicherheitsstandards voranzutreiben. Weitere Informationen finden Sie unter [Reaktive Kontrolle](#) in Implementieren von Sicherheitskontrollen in AWS.

Beibehaltung

Siehe [7 Rs](#).

zurückziehen

Siehe [7 Rs](#).

Retrieval Augmented Generation (RAG)

Eine [generative KI-Technologie](#), bei der ein [LLM](#) auf eine maßgebliche Datenquelle verweist, die sich außerhalb seiner Trainingsdatenquellen befindet, bevor eine Antwort generiert wird. Ein RAG-Modell könnte beispielsweise eine semantische Suche in der Wissensdatenbank oder in benutzerdefinierten Daten einer Organisation durchführen. Weitere Informationen finden Sie unter [Was ist RAG](#).

Drehung

Der Vorgang, bei dem ein [Geheimnis](#) regelmäßig aktualisiert wird, um es einem Angreifer zu erschweren, auf die Anmeldeinformationen zuzugreifen.

Zugriffskontrolle für Zeilen und Spalten (RCAC)

Die Verwendung einfacher, flexibler SQL-Ausdrücke mit definierten Zugriffsregeln. RCAC besteht aus Zeilenberechtigungen und Spaltenmasken.

RPO

Siehe [Recovery Point Objective](#).

RTO

Siehe [Ziel für die Erholungszeit](#).

Runbook

Eine Reihe manueller oder automatisierter Verfahren, die zur Ausführung einer bestimmten Aufgabe erforderlich sind. Diese sind in der Regel darauf ausgelegt, sich wiederholende Operationen oder Verfahren mit hohen Fehlerquoten zu rationalisieren.

S

SAML 2.0

Ein offener Standard, den viele Identitätsanbieter (IdPs) verwenden. Diese Funktion ermöglicht föderiertes Single Sign-On (SSO), sodass sich Benutzer bei den API-Vorgängen anmelden AWS-Managementkonsole oder die AWS API-Operationen aufrufen können, ohne dass Sie einen Benutzer in IAM für alle in Ihrer Organisation erstellen müssen. Weitere Informationen zum SAML-2.0.-basierten Verbund finden Sie unter [Über den SAML-2.0-basierten Verbund](#) in der IAM-Dokumentation.

SCADA

Siehe [Aufsichtskontrolle und Datenerfassung](#).

SCP

Siehe [Richtlinie zur Dienstkontrolle](#).

Secret

Interne AWS Secrets Manager, vertrauliche oder eingeschränkte Informationen, wie z. B. ein Passwort oder Benutzeranmeldeinformationen, die Sie in verschlüsselter Form speichern. Es besteht aus dem geheimen Wert und seinen Metadaten. Der geheime Wert kann binär, eine einzelne Zeichenfolge oder mehrere Zeichenketten sein. Weitere Informationen finden Sie unter [Was ist in einem Secrets Manager Manager-Geheimnis?](#) in der Secrets Manager Manager-Dokumentation.

Sicherheit durch Design

Ein systemtechnischer Ansatz, der die Sicherheit während des gesamten Entwicklungsprozesses berücksichtigt.

Sicherheitskontrolle

Ein technischer oder administrativer Integritätsschutz, der die Fähigkeit eines Bedrohungsakteurs, eine Schwachstelle auszunutzen, verhindert, erkennt oder einschränkt. Es gibt vier Haupttypen von Sicherheitskontrollen: [präventiv](#), [detektiv](#), [reaktionsschnell](#) und [proaktiv](#).

Härtung der Sicherheit

Der Prozess, bei dem die Angriffsfläche reduziert wird, um sie widerstandsfähiger gegen Angriffe zu machen. Dies kann Aktionen wie das Entfernen von Ressourcen, die nicht mehr benötigt werden, die Implementierung der bewährten Sicherheitsmethode der Gewährung geringster Berechtigungen oder die Deaktivierung unnötiger Feature in Konfigurationsdateien umfassen.

System zur Verwaltung von Sicherheitsinformationen und Ereignissen (security information and event management – SIEM)

Tools und Services, die Systeme für das Sicherheitsinformationsmanagement (SIM) und das Management von Sicherheitsereignissen (SEM) kombinieren. Ein SIEM-System sammelt, überwacht und analysiert Daten von Servern, Netzwerken, Geräten und anderen Quellen, um Bedrohungen und Sicherheitsverletzungen zu erkennen und Warnmeldungen zu generieren.

Automatisierung von Sicherheitsreaktionen

Eine vordefinierte und programmierte Aktion, die darauf ausgelegt ist, automatisch auf ein Sicherheitsereignis zu reagieren oder es zu beheben. Diese Automatisierungen dienen als [detektive](#) oder [reaktionsschnelle](#) Sicherheitskontrollen, die Sie bei der Implementierung bewährter AWS Sicherheitsmethoden unterstützen. Beispiele für automatisierte Antwortaktionen sind das Ändern einer VPC-Sicherheitsgruppe, das Patchen einer Amazon EC2 EC2-Instance oder das Rotieren von Anmeldeinformationen.

Serverseitige Verschlüsselung

Verschlüsselung von Daten am Zielort durch denjenigen AWS-Service, der sie empfängt.

Service-Kontrollrichtlinie (SCP)

Eine Richtlinie, die eine zentrale Steuerung der Berechtigungen für alle Konten in einer Organisation in ermöglicht AWS Organizations. SCPs Definieren Sie Leitplanken oder legen Sie Grenzwerte für Aktionen fest, die ein Administrator an Benutzer oder Rollen delegieren kann. Sie können sie SCPs als Zulassungs- oder Ablehnungslisten verwenden, um festzulegen, welche Dienste oder Aktionen zulässig oder verboten sind. Weitere Informationen finden Sie in der AWS Organizations Dokumentation unter [Richtlinien zur Dienststeuerung](#).

Service-Endpoint

Die URL des Einstiegspunkts für einen AWS-Service. Sie können den Endpunkt verwenden, um programmgesteuert eine Verbindung zum Zielservice herzustellen. Weitere Informationen finden Sie unter [AWS-Service -Endpunkte](#) in der Allgemeine AWS-Referenz.

Service Level Agreement (SLA)

Eine Vereinbarung, in der klargestellt wird, was ein IT-Team seinen Kunden zu bieten verspricht, z. B. in Bezug auf Verfügbarkeit und Leistung der Services.

Service-Level-Indikator (SLI)

Eine Messung eines Leistungsaspekts eines Dienstes, z. B. seiner Fehlerrate, Verfügbarkeit oder Durchsatz.

Service-Level-Ziel (SLO)

Eine Zielkennzahl, die den Zustand eines Dienstes darstellt, gemessen anhand eines [Service-Level-Indikators](#).

Modell der geteilten Verantwortung

Ein Modell, das die Verantwortung beschreibt, mit der Sie gemeinsam AWS für Cloud-Sicherheit und Compliance verantwortlich sind. AWS ist für die Sicherheit der Cloud verantwortlich, während Sie für die Sicherheit in der Cloud verantwortlich sind. Weitere Informationen finden Sie unter [Modell der geteilten Verantwortung](#).

SIEM

Siehe [Sicherheitsinformations- und Event-Management-System](#).

Single Point of Failure (SPOF)

Ein Fehler in einer einzelnen, kritischen Komponente einer Anwendung, der das System stören kann.

SLA

Siehe [Service Level Agreement](#).

SLI

Siehe [Service-Level-Indikator](#).

ALSO

Siehe [Service-Level-Ziel](#).

split-and-seed Modell

Ein Muster für die Skalierung und Beschleunigung von Modernisierungsprojekten. Sobald neue Features und Produktversionen definiert werden, teilt sich das Kernteam auf, um neue Produktteams zu bilden. Dies trägt zur Skalierung der Fähigkeiten und Services Ihrer Organisation bei, verbessert die Produktivität der Entwickler und unterstützt schnelle Innovationen. Weitere Informationen finden Sie unter [Schrittweiser Ansatz zur Modernisierung von Anwendungen in der AWS Cloud](#)

SPOTTEN

Siehe [Single Point of Failure](#).

Sternschema

Eine Datenbank-Organisationsstruktur, die eine große Faktentabelle zum Speichern von Transaktions- oder Messdaten und eine oder mehrere kleinere dimensionale Tabellen zum Speichern von Datenattributen verwendet. Diese Struktur ist für die Verwendung in einem [Data Warehouse](#) oder für Business Intelligence-Zwecke konzipiert.

Strangler-Fig-Muster

Ein Ansatz zur Modernisierung monolithischer Systeme, bei dem die Systemfunktionen schrittweise umgeschrieben und ersetzt werden, bis das Legacy-System außer Betrieb genommen werden kann. Dieses Muster verwendet die Analogie einer Feigenrebe, die zu einem etablierten Baum heranwächst und schließlich ihren Wirt überwindet und ersetzt. Das Muster wurde [eingeführt von Martin Fowler](#) als Möglichkeit, Risiken beim Umschreiben monolithischer Systeme zu managen. Ein Beispiel für die Anwendung dieses Musters finden Sie unter [Schrittweises Modernisieren älterer Microsoft ASP.NET \(ASMX\)-Webservices mithilfe von Containern und Amazon API Gateway](#).

Subnetz

Ein Bereich von IP-Adressen in Ihrer VPC. Ein Subnetz muss sich in einer einzigen Availability Zone befinden.

Aufsichtskontrolle und Datenerfassung (SCADA)

In der Fertigung ein System, das Hardware und Software zur Überwachung von Sachanlagen und Produktionsabläufen verwendet.

Symmetrische Verschlüsselung

Ein Verschlüsselungsalgorithmus, der denselben Schlüssel zum Verschlüsseln und Entschlüsseln der Daten verwendet.

synthetisches Testen

Testen eines Systems auf eine Weise, die Benutzerinteraktionen simuliert, um potenzielle Probleme zu erkennen oder die Leistung zu überwachen. Sie können [Amazon CloudWatch Synthetics](#) verwenden, um diese Tests zu erstellen.

Systemaufforderung

Eine Technik, mit der einem [LLM](#) Kontext, Anweisungen oder Richtlinien zur Verfügung gestellt werden, um sein Verhalten zu steuern. Systemaufforderungen helfen dabei, den Kontext festzulegen und Regeln für Interaktionen mit Benutzern festzulegen.

T

tags

Schlüssel-Wert-Paare, die als Metadaten für die Organisation Ihrer Ressourcen dienen. AWS Mit Tags können Sie Ressourcen verwalten, identifizieren, organisieren, suchen und filtern. Weitere Informationen finden Sie unter [Markieren Ihrer AWS -Ressourcen](#).

Zielvariable

Der Wert, den Sie in überwachtem ML vorhersagen möchten. Dies wird auch als Ergebnisvariable bezeichnet. In einer Fertigungsumgebung könnte die Zielvariable beispielsweise ein Produktfehler sein.

Aufgabenliste

Ein Tool, das verwendet wird, um den Fortschritt anhand eines Runbooks zu verfolgen. Eine Aufgabenliste enthält eine Übersicht über das Runbook und eine Liste mit allgemeinen Aufgaben, die erledigt werden müssen. Für jede allgemeine Aufgabe werden der geschätzte Zeitaufwand, der Eigentümer und der Fortschritt angegeben.

Testumgebungen

[Siehe Umgebung.](#)

Training

Daten für Ihr ML-Modell bereitstellen, aus denen es lernen kann. Die Trainingsdaten müssen die richtige Antwort enthalten. Der Lernalgorithmus findet Muster in den Trainingsdaten, die die Attribute der Input-Daten dem Ziel (die Antwort, die Sie voraussagen möchten) zuordnen. Es gibt ein ML-Modell aus, das diese Muster erfasst. Sie können dann das ML-Modell verwenden, um Voraussagen für neue Daten zu erhalten, bei denen Sie das Ziel nicht kennen.

Transit-Gateway

Ein Netzwerk-Transit-Hub, über den Sie Ihre Netzwerke VPCs und Ihre lokalen Netzwerke miteinander verbinden können. Weitere Informationen finden Sie in der Dokumentation unter [Was ist ein Transit-Gateway](#). AWS Transit Gateway

Stammbasierter Workflow

Ein Ansatz, bei dem Entwickler Feature lokal in einem Feature-Zweig erstellen und testen und diese Änderungen dann im Hauptzweig zusammenführen. Der Hauptzweig wird dann sequentiell für die Entwicklungs-, Vorproduktions- und Produktionsumgebungen erstellt.

Vertrauenswürdiger Zugriff

Gewährung von Berechtigungen für einen Dienst, den Sie angeben, um Aufgaben in Ihrer Organisation AWS Organizations und in deren Konten in Ihrem Namen auszuführen. Der vertrauenswürdige Service erstellt in jedem Konto eine mit dem Service verknüpfte Rolle, wenn diese Rolle benötigt wird, um Verwaltungsaufgaben für Sie auszuführen. Weitere Informationen finden Sie in der AWS Organizations Dokumentation [unter Verwendung AWS Organizations mit anderen AWS Diensten](#).

Optimieren

Aspekte Ihres Trainingsprozesses ändern, um die Genauigkeit des ML-Modells zu verbessern. Sie können das ML-Modell z. B. trainieren, indem Sie einen Beschriftungssatz generieren, Beschriftungen hinzufügen und diese Schritte dann mehrmals unter verschiedenen Einstellungen wiederholen, um das Modell zu optimieren.

Zwei-Pizzen-Team

Ein kleines DevOps Team, das Sie mit zwei Pizzen ernähren können. Eine Teamgröße von zwei Pizzen gewährleistet die bestmögliche Gelegenheit zur Zusammenarbeit bei der Softwareentwicklung.

U

Unsicherheit

Ein Konzept, das sich auf ungenaue, unvollständige oder unbekannte Informationen bezieht, die die Zuverlässigkeit von prädiktiven ML-Modellen untergraben können. Es gibt zwei Arten von Unsicherheit: Epistemische Unsicherheit wird durch begrenzte, unvollständige Daten verursacht, wohingegen aleatorische Unsicherheit durch Rauschen und Randomisierung verursacht wird, die in den Daten liegt. Weitere Informationen finden Sie im Leitfaden [Quantifizieren der Unsicherheit in Deep-Learning-Systemen](#).

undifferenzierte Aufgaben

Diese Arbeit wird auch als Schwerstarbeit bezeichnet. Dabei handelt es sich um Arbeiten, die zwar für die Erstellung und den Betrieb einer Anwendung erforderlich sind, aber dem Endbenutzer keinen direkten Mehrwert bieten oder keinen Wettbewerbsvorteil bieten. Beispiele für undifferenzierte Aufgaben sind Beschaffung, Wartung und Kapazitätsplanung.

höhere Umgebungen

Siehe [Umgebung](#).

V

Vacuuming

Ein Vorgang zur Datenbankwartung, bei dem die Datenbank nach inkrementellen Aktualisierungen bereinigt wird, um Speicherplatz zurückzugewinnen und die Leistung zu verbessern.

Versionskontrolle

Prozesse und Tools zur Nachverfolgung von Änderungen, z. B. Änderungen am Quellcode in einem Repository.

VPC-Peering

Eine Verbindung zwischen zwei VPCs, die es Ihnen ermöglicht, den Verkehr mithilfe privater IP-Adressen weiterzuleiten. Weitere Informationen finden Sie unter [Was ist VPC-Peering?](#) in der Amazon-VPC-Dokumentation.

Schwachstelle

Ein Software- oder Hardwarefehler, der die Sicherheit des Systems beeinträchtigt.

W

Warmer Cache

Ein Puffer-Cache, der aktuelle, relevante Daten enthält, auf die häufig zugegriffen wird. Die Datenbank-Instance kann aus dem Puffer-Cache lesen, was schneller ist als das Lesen aus dem Hauptspeicher oder von der Festplatte.

warme Daten

Daten, auf die selten zugegriffen wird. Bei der Abfrage dieser Art von Daten sind mäßig langsame Abfragen in der Regel akzeptabel.

Fensterfunktion

Eine SQL-Funktion, die eine Berechnung für eine Gruppe von Zeilen durchführt, die sich in irgendeiner Weise auf den aktuellen Datensatz beziehen. Fensterfunktionen sind nützlich für die Verarbeitung von Aufgaben wie die Berechnung eines gleitenden Durchschnitts oder für den Zugriff auf den Wert von Zeilen auf der Grundlage der relativen Position der aktuellen Zeile.

Workload

Ein Workload ist eine Sammlung von Ressourcen und Code, die einen Unternehmenswert bietet, wie z. B. eine kundenorientierte Anwendung oder ein Backend-Prozess.

Workstream

Funktionsgruppen in einem Migrationsprojekt, die für eine bestimmte Reihe von Aufgaben verantwortlich sind. Jeder Workstream ist unabhängig, unterstützt aber die anderen Workstreams im Projekt. Der Portfolio-Workstream ist beispielsweise für die Priorisierung von Anwendungen, die Wellenplanung und die Erfassung von Migrationsmetadaten verantwortlich. Der Portfolio-Workstream liefert diese Komponenten an den Migrations-Workstream, der dann die Server und Anwendungen migriert.

WURM

Sehen [Sie einmal schreiben, viele lesen](#).

WQF

Siehe [AWS Workload-Qualifizierungsrahmen](#).

einmal schreiben, viele lesen (WORM)

Ein Speichermodell, das Daten ein einziges Mal schreibt und verhindert, dass die Daten gelöscht oder geändert werden. Autorisierte Benutzer können die Daten so oft wie nötig lesen, aber sie können sie nicht ändern. Diese Datenspeicherinfrastruktur gilt als [unveränderlich](#).

Z

Zero-Day-Exploit

Ein Angriff, in der Regel Malware, der eine [Zero-Day-Sicherheitslücke](#) ausnutzt.

Zero-Day-Sicherheitslücke

Ein unfehlbarer Fehler oder eine Sicherheitslücke in einem Produktionssystem. Bedrohungsakteure können diese Art von Sicherheitslücke nutzen, um das System anzugreifen. Entwickler werden aufgrund des Angriffs häufig auf die Sicherheitsanfälligkeit aufmerksam.

Eingabeaufforderung ohne Zwischenfälle

Bereitstellung von Anweisungen für die Ausführung einer Aufgabe an einen [LLM](#), jedoch ohne Beispiele (Schnappschüsse), die ihm als Orientierungshilfe dienen könnten. Der LLM muss sein vortrainiertes Wissen einsetzen, um die Aufgabe zu bewältigen. Die Effektivität von Zero-Shot Prompting hängt von der Komplexität der Aufgabe und der Qualität der Aufforderung ab. [Siehe auch Few-Shot-Prompting](#).

Zombie-Anwendung

Eine Anwendung, deren durchschnittliche CPU- und Arbeitsspeichernutzung unter 5 Prozent liegt. In einem Migrationsprojekt ist es üblich, diese Anwendungen außer Betrieb zu nehmen.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.