

»KÜNSTLICHE INTELLIGENZ IN DER QA FÜR BESSERE GESCHÄFTSERGEBNISSE«

Qualitätssicherung (QA) ist eine Reise, auf der es viele Herausforderungen zu bewältigen gilt, mit denen sich QA-Mitarbeiter während des gesamten Lebenszyklus der Software beschäftigen müssen. Der problemlose Zugang zum Internet führt zur Erfassung von immer mehr Daten. Durch das Internet der Dinge ist darüber hinaus überall Konnektivität erforderlich. Das bedeutet, dass wir beim Testen eines Systems aufmerksamer sein und neue Strategien für das Testen in einer vernetzten Welt definieren müssen. Da immer komplexere unternehmenskritische Anwendungen entwickelt werden, besteht die Herausforderung darin, den Kunden in kürzester Zeit eine vollständig getestete Software in höchster Qualität zu liefern. Der Einsatz von Künstlicher Intelligenz (KI) kann dies ermöglichen und für maximale Kundenzufriedenheit sorgen.



Unternehmen verwandeln sich heutzutage in intelligente, datengestützte Organisationen, die stark auf KI und digitale Technologien setzen, um ihre Vorgänge zu optimieren und ihre Geschäftsmodelle zu verändern. KI kann Softwaretests von der Testvorbereitung über die Planung bis hin zur Durchführung und zum Reporting ohne menschliches Zutun vorantreiben.

Dank KI werden genauere Ergebnisse erzielt und Continuous Testing ermöglicht. Das QA-Team und Test Engineers spielen eine neue Rolle bei der Steuerung und Überwachung der KI-Ergebnisse. Mithilfe neuer KI-Technologien wird die Zeit bis zur Markteinführung verkürzt und die Effizienz der Organisation im Rahmen komplexer Software erhöht. Durch die Nutzung von KI-Algorithmen und KI-Technologie können Unternehmen ihr Produktangebot erweitern, den Kundenservice verbessern und die Qualität der bereitgestellten Dienstleistungen erhöhen. Darüber hinaus werden die Produkte mit stabiler Software ausgestattet.

Technologischer Fortschritt mit KI

Wenn wir uns die Entwicklung der gängigen Testverfahren ansehen, sehen wir, dass wir mit manuellen Tests begonnen haben, bei denen wir dem Wasserfallmodell folgten. Dieses wurde für Software entwickelt, die über Jahre hinweg im Einsatz ist.

Anfang 2000 setzte sich agiles Testen durch, wodurch der Zeitaufwand auf Monate und Wochen verringert wurde. Mit der Einführung von DevOps implementiert jede Organisation Continuous Testing, um unmittelbares Feedback zu erhalten, wodurch der Zeitaufwand weiter auf Wochen und Tage verringert wird.

Wenn wir jedoch in die Zukunft blicken, wird deutlich, dass die Testvorgänge weiter beschleunigt und die durch Technologien wie KI/ML und Analysen gestützten Qualitätsansprüche erreicht werden müssen. Zu den Anforderungen, die zu einem Übergang führen, zählen:

- › Zunahme vernetzter Geräte und Produkte mit durch KI/ML gestützten kognitiven Funktionen,
- › Generierung von unzähligen Daten,
- › Analyse und Überwachung von großen Datenmengen,
- › zunehmender Einsatz von Agile und Dev-Ops,
- › Nachfrage nach Continuous Testing und Continuous Monitoring,
- › Bedarf nach intelligenter, self-healing End-to-End-Testautomatisierung,
- › Security Testing aufgrund der zunehmenden Komplexität,
- › steigende Nachfrage nach digitaler Transformation.

Um auf diese Unternehmensanforderungen schneller reagieren zu können, *müssen Unternehmen auf neue Technologien, wie KI, setzen*. Traditionelle Testkompetenzen werden als veraltet angesehen, da sie für eine zu starke Verlangsamung der Entwicklung verantwortlich gemacht werden. Gleichzeitig beobachten wir, dass neue Fähigkeiten und technologische Lösungen, zum Beispiel Testautomatisierung, Datenanalyse und KI, stark nachgefragt werden, da sich dadurch schneller ein geschäftlicher Mehrwert und eine verbesserte Softwarequalität erzielen lassen.

Als QA-Experten müssen wir vor allem die Art und Weise ändern, wie das gesamte Ökosystem getestet wird. Beispiele, in denen herkömmliches Testen nicht aus-

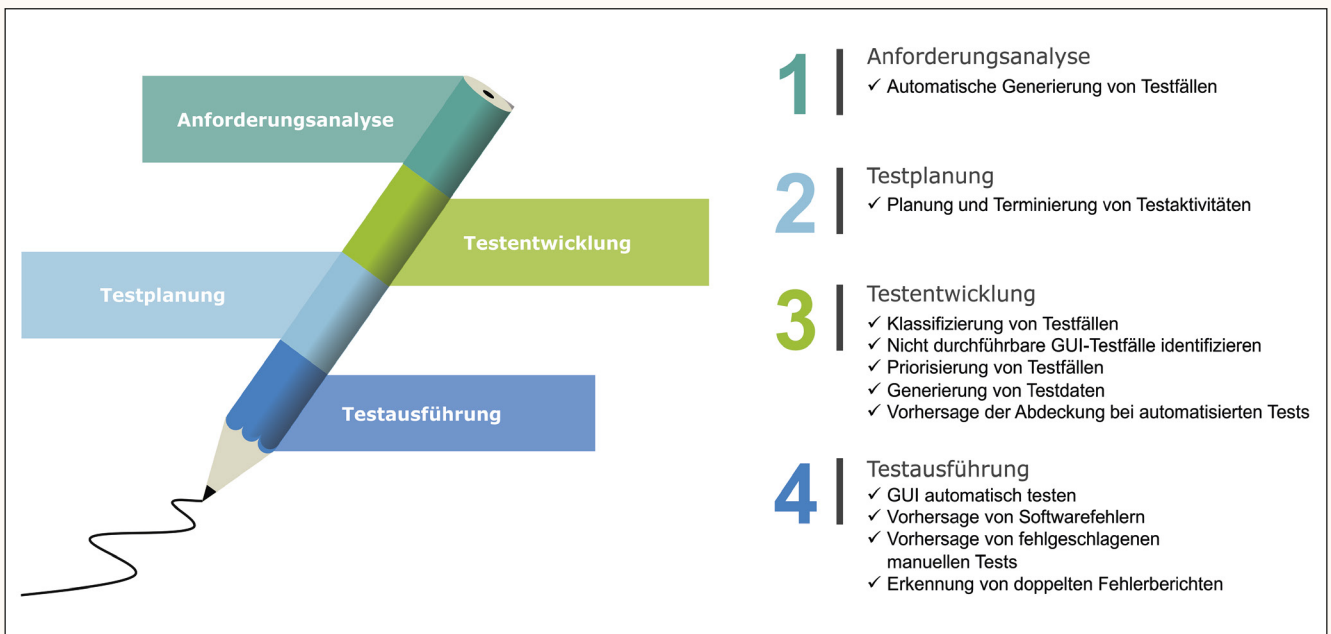


Abb. 1: KI-gesteuertes Testen – Ein neues QA-Zeitalter

reicht, um die Kundenbedürfnisse zu erfüllen, sind:

- ▶ Chatbots ersetzen Menschen im Kundendienst.
- ▶ Entsperren von Smartphones mithilfe von Gesichtserkennungstechnologien.
- ▶ Spezielles Angebot auf Basis unserer vorherigen Benutzerwahl in sozialen Medien.
- ▶ KI-gesteuerte persönliche Assistenten wie Alexa, Echo und Siri, die Sprachbefehle befolgen und die Antwort in der bevorzugten Sprache bereitstellen können.
- ▶ Apps für Fahrgemeinschaften wie Uber oder Ola führen KI-Algorithmen im Backend aus, um die Fahrten bestmöglich abzustimmen.

Wie KI den Lebenszyklus von Tests optimiert hat

Dank KI gehen Unternehmen ganz neue Wege, wenn es um das Erzielen von geschäftlichem Mehrwert geht. Erforschen wir nun im Detail, wie diese veränderten Technologien die Art und Weise, wie wir Tests durchführen, verändern werden. **Abbildung 1** zeigt KI-gesteuertes Testen: Trainieren, Lernen, Validieren und Überwachen.

Daten generieren

Mit einem KI-Modell können Testfälle automatisch anhand von Anforderungen erstellt werden. Dazu werden Werte aus verschiedenen Attributen kombiniert, die im Anforderungsdokument in einem spezifischen Format zusammen mit Unternehmensregeln bereitgestellt werden. Das Modell findet doppelte Testfälle und gruppiert sie, was als Cluster bezeichnet wird. Das Modell prüft außerdem die gegenseitigen Abhängigkeiten von Tests und Klassen und erstellt Pseudoobjekte und Scheinprozeduren, um Tests voneinander abzugrenzen.

Manuelle Tests werden mithilfe der Methode „Aufnehmen und wiedergeben“ in automatisierte Tests umgewandelt. Und zwar in jedem Durchlauf. *Das KI-Tool erkennt die Muster, erstellt Testdaten* zusammen mit untersuchten Parametern, erstellt weitere Tests und wendet erlernte Muster auf andere Tests an, wodurch die Erstellung ausgereifterer automatisierter Testszenarios unterstützt wird.

Modell trainieren – Testoptimierung

Modelle unterstützen die Testoptimierung, indem sie die Attribute und Werte in den Clustern suchen. Es werden optimale Testfallsätze erstellt, die 100 Prozent Abdeckung repräsentieren, sowie ein *Repository mit optimierten Test-Suites*, die wiederholt ver-

wendet werden können. Das KI-Modell durchläuft den Quellcode, um zu überprüfen, welche Teile des Codes nicht abgedeckt sind, und generiert dann automatisch neue Tests, um die Gesamttestabdeckung der zu testenden Anwendung zu erhöhen. Es durchläuft darüber hinaus eine Anzahl von Pfaden und lernt aus den bestehenden Daten, um Problembereiche zu ermitteln.

Testausführung und Validierung

Die tatsächliche Automatisierung wird über einen optimalen Testsatz generiert, der in ein Automatisierungs-Framework integriert ist. Das Modell bietet self-healing Funktionen zur Laufzeit an und *validiert die UI-Test-szenarien automatisch*, wodurch Herausforderungen hinsichtlich Wartungsfähigkeit und Stabilität beim Testen verringert werden. Das KI-Tool überwacht und erfasst Informationen über den Inhalt von Web-UIs, während es DOM-Elemente und deren Attribute extrahiert. Es speichert außerdem Informationen innerhalb des KI-Moduls und wird auf Basis der gelernten Inhalte aktualisiert.

Continuous Monitoring

Ein KI-Modell *überwacht und analysiert Fehler kontinuierlich*. Es identifiziert die Muster der Fehler, ermittelt den Ursprung der Fehler in Echtzeit und klassifiziert sie als neuen Fehler oder Regressionsfehler. Es übernimmt zudem das Durchsuchen von

Protokolldateien, was Zeit spart und die Korrektheit des Programms erheblich steigert. Algorithmen sagen die Wahrscheinlichkeit eines fehlgeschlagenen Builds voraus, wenn Codeänderungen an Anwendungen vorgenommen werden, und optimieren die Tests durch Selbstlernen.

So verändert KI unseren Testansatz

Statt die Aufgaben manuell auszuführen, würden wir KI-Algorithmen einsetzen. Somit liegt der Fokus stärker auf der Gewinnung von Informationen in Bezug auf das Lernen, Trainieren und Lösen von Problemen. Neben dem traditionellen Testansatz werden wir uns mehr auf die folgenden Testtypen konzentrieren.

A/B-Tests

Der Zweck von A/B-Tests oder Split-Tests ist es, Benutzern zwei Varianten derselben Webseite zu zeigen und zu vergleichen, welche Variante die meisten Treffer erzielt. Diese Tests werden verwendet, um zu analysieren, welche Variante besser abschneidet, wenn es um Design, Webseiten-Workflows, Seitenladezeit usw. geht. Für die Vermarktung von Produkten werden Variablen getestet, um zu analysieren, welche Kombination von Variablen unter allen Möglichkeiten die besten Ergebnisse erzielt. Es werden Ausführungen durchgeführt, um Änderungen an bestimmten Elementen über

mehrere Seiten hinweg zu testen und ein gleichbleibendes Nutzungserlebnis für die Endnutzer zu bieten.

Metamorphe Tests

In komplexeren Szenarien, bei denen eine bestimmte Eingabe nicht einer bestimmten Ausgabe zugeordnet werden kann, stellen metamorphe Tests eine Alternative dar. Mit diesen Tests können Eingaben in Bezug darauf beschrieben werden, welche Beziehungen zwischen Eingaben und dem Übergang zu einer Testeingabe bestehen, und es kann beobachtet werden, wie die Ausgabe des Programms zu einem korrekten Ergebnis wird. Die Herausforderung bei Softwaretests besteht darin, die beobachtete Ausgabe mit der vorhergesagten korrekten Ausgabe zu vergleichen. Das kann mithilfe von metamorphen Tests erreicht werden.

Prädiktive Analysen

Der Schwerpunkt verlagert sich aktuell vom traditionellen Ansatz hin zu einem stärker benutzerzentrierten Ansatz. Das QA-Team hat einen Überblick über das Gesamtszenario und kann die Daten in Bezug auf Kundenzufriedenheit analysieren, was beim Testen der Anwendung im richtigen Kontext hilfreich ist. *Testdaten von Unternehmen werden erfasst, analysiert und wirksam eingesetzt, um die Tests zu optimieren.* Auf der Grundlage der Analyse von Fehlern, die

beim Testen aufgetreten sind, prognostiziert das QA-Team die umsetzbaren Ergebnisse für die weitere Optimierung sowie Schwachstellen in der zu testenden Anwendung.

Kognitive QA-Tests

Mit kognitiven Tests kann der effektivste auszuführende Testsatz bestimmt werden. Dieser ist zwar fehleranfälliger, bietet aber die beste Abdeckung. Im Rahmen dieser Tests werden wiederkehrende Aufgaben bewertet und der Testfallsatz erfasst, der automatisiert werden kann. Mithilfe von intelligenter QA-Automatisierung und RPA (Robotic Process Automation) werden Testskripte automatisch mithilfe von self-healing Funktionen generiert. QA-Dashboards rufen Echtzeitdaten ab, geben rechtzeitig Alarme aus und sind bei Continuous Monitoring hilfreich. Durch automatisierte Tests können dieselben Schritte ordnungsgemäß über bestimmte Zeiträume durchgeführt werden. Darüber hinaus wird Zeit gespart und Genauigkeit gewährleistet.

Continuous Testing

Damit die Häufigkeit der durchgeführten Tests mit dem Tempo von Produkteinführungen mithalten kann, ist man in der QA zu kontinuierlichen Tests übergegangen. Wir können automatisierte Tests ohne zusätzliche Kosten schneller durchführen, da wiederholte manuelle Ausführungen fehleranfällig und kostspielig sind.

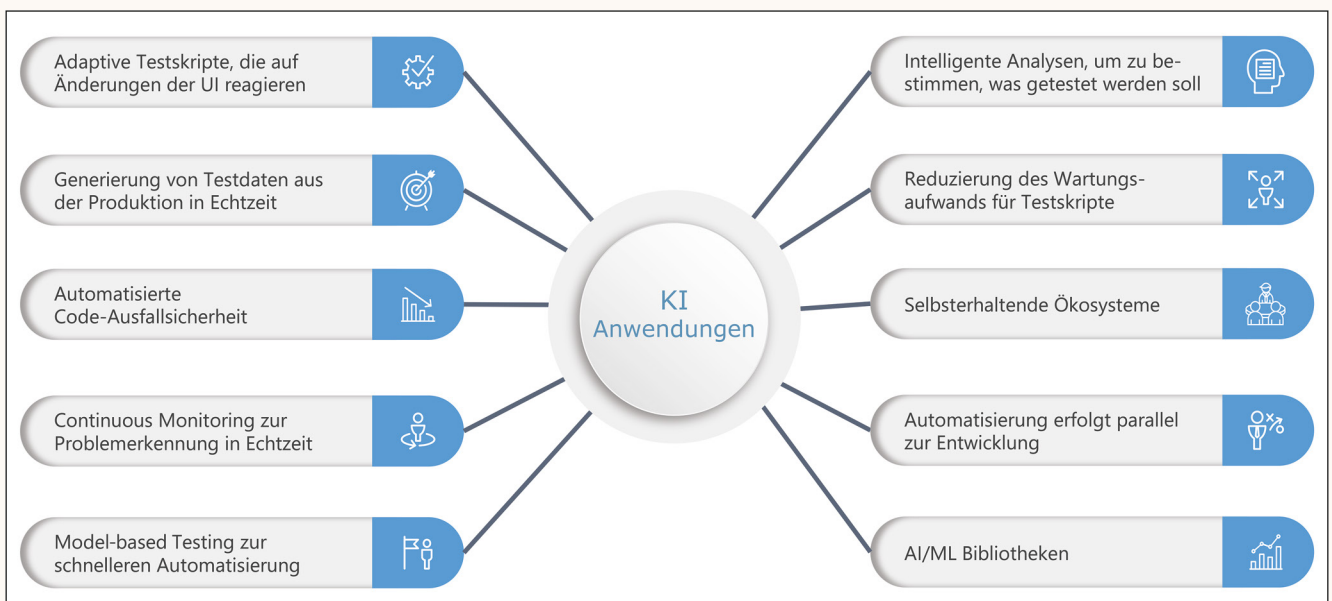


Abb. 2: Intelligente Testautomatisierungsplattform

Beim *Shift-left*-Prinzip wird ein integriertes Ökosystem mit Assets wie beispielsweise Anforderungen, Codes, Build- und Deploy-Kits, Testfällen, Automatisierungsskripten, Testumgebungen und der Bereitstellung von Testdaten geschaffen, um Tests auf eine optimierte Weise durchzuführen. Beim *Shift-right*-Prinzip können KI-Bots erstellt werden, die Produktionsbereitstellungsfehler nachvollziehen können und mit denen Abhängigkeiten erstellt und ausgeführt werden. Dank des *Shift-right*-Prinzips arbeiten Test- und Produktionsteams enger zusammen, da Implementierungs- und Release-Zyklen kürzer ausfallen.

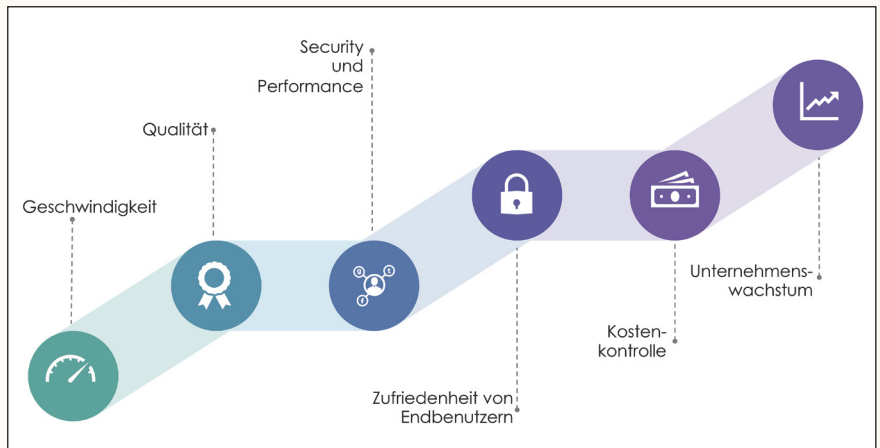


Abb. 3: Wichtige Geschäftstreiber

Wie sich KI-Automatisierungsplattformen verändert haben

Automatisierungs-Frameworks sind intelligent genug, um die in der UI vorgenommenen Änderungen zu identifizieren. Mithilfe von Frameworks können alle Exceptions analysiert und anschließend in Fehler oder Systemausnahmen kategorisiert werden. Darüber hinaus sind Analysen von Kundendaten zum besseren Verständnis der aktuellsten von Kunden benötigten Produkte und Funktionen möglich. Anschließend können die erfassten Daten durch Tests analysiert und Engpässe in den Anwendungen vorhergesagt/identifiziert werden. Algorithmen sagen die durch automatisierte Tools erreichte Abdeckung voraus. Durch Mustererkennung oder Funktionen zur visuellen Validierung werden wiederum visuelle Fehler in der Software ermittelt (siehe **Abbildung 2**).

Der Weg nach vorne und die erforderlichen Fähigkeiten zur Implementierung von KI

Das QA-Team muss Daten analysieren können, ein detailliertes Verständnis über statistische Konzepte aufweisen, über starke

mathematische Konzepte sowie Modellierungskonzepte verfügen und kognitive Prozesse verstehen können. Da diese Veränderungen rasch vor sich gehen, benötigen wir eine *self-healing Testautomatisierungsplattform, in der die Automatisierung parallel zur Entwicklung abläuft*. Mit der steigenden Nachfrage nach KI müssen Kenntnisse/Schulungen zu neuen Technologien und Tools frühestmöglich bereitgestellt werden. Zudem sind leichte Open-Source-Automatisierungstools zur raschen Erfassung und Verarbeitung von Daten gefragt. Mithilfe von KI steht die richtige Datenqualität zur Verfügung, da produktionsähnliche Testdaten durch Technologien zur Datenvirtualisierung automatisch erstellt werden.

Wichtige Geschäftstreiber

Wenn es um ein schnelleres Entwickeln und Bereitstellen der Software geht, sind *Geschwindigkeit, Qualität, Security und Performance* die wichtigsten Schlüsselfaktoren. Andere wichtige Treiber wie *Unternehmenswachstum, Zufriedenheit von Endbenutzern und Kostenkontrolle* spielen ebenfalls eine wichtige Rolle auf dem Weg zum Erfolg (siehe **Abbildung 3**).

Resümee

Um in dem sich ständig verändernden Umfeld von heute nicht an Bedeutung zu verlieren, muss unser Schwerpunkt darauf liegen, ein End-to-End-QA-Partner zu werden, der durch ein tiefgehendes Verständnis von Unternehmensprozessen und Ergebnissen überzeugt. Wir sollten das Ökosystem darüber hinaus durch kognitive und analytische Fähigkeiten mit hoher Geschwindigkeit, Agilität und Zuverlässigkeit unterstützen.

Die Anforderungen an KI-basierte Anwendungen werden die Rollen von SDETs (Software Development Engineers in Test) dahingehend verändern, dass sie mit data science/statistischen Modellen arbeiten. Zudem müssen wir unsere Fähigkeiten entsprechend anpassen, um unsere Qualitätsprozesse zu verbessern. Wir müssen KI-Algorithmen und Technologien für eine self-healing Testautomatisierung nutzen sowie Berichte und Analysen automatisieren, um unser Business voranzutreiben und die Geschäftsergebnisse zu verbessern.



Manisha Mittal

manisha.mittal@nagarro.com

verfügt über mehr als 16 Jahre Erfahrung im Bereich Software Test Engineering. Sie arbeitet als Associate Director für das Softwarehaus Nagarro und ist Co-Lead der globalen „Quality DevOps Practice“. Manisha beschäftigt sich kontinuierlich mit dem Testen von neuen Technologien und setzt sich für die Bereitstellung von qualitativ hochwertiger Software ein.