Ranorex + TestRail + infometis

Die Testautomatisierungslücke schliessen

Ein Leitfaden zur Umsetzung von Automatisierungszielen in die Praxis

Inhalt

Einleitung

Das Testautomatisierungs-Paradoxon

Testautomatisierung ist keine neue Technologie. Viele der Frameworks und Tools, die heute genutzt werden, sind bereits vor über 20 Jahren auf den Markt gekommen. Dennoch haben fast alle, die automatisierte Tests einsetzen, weiterhin mit der Testautomatisierungslücke zu kämpfen.

Vereinfacht ausgedrückt beschreibt die Testautomatisierungslücke die Differenz zwischen der Anzahl der Tests, die man gerne automatisieren würde, und der tatsächlichen Realität der vorhandenen Automatisierungssuite. Und diese Lücke scheint sich bei den meisten Teams nicht zu schliessen.

In unserem jährlichen Software Testing and Quality Report befragen wir jedes Jahr Tausende von QA- und DevOps-Teams, um verschiedene Kennzahlen zu bewerten und zu vergleichen, darunter auch die Testautomatisierungslücke. QA-Teams verfehlen ihre Automatisierungsziele dabei regelmässig. Während sich Teams jedes Jahr ambitioniertere Ziele setzen – mit dem Anspruch, bis Ende 2025 63% aller Tests zu automatisieren, verglichen mit einem Ziel von 54% im Jahr 2020 – sind die tatsächlichen Automatisierungsraten in den letzten fünf Jahren bei rund 40% geblieben.

QA-Teams verfügen heute über fortschrittlichere Tools und effizientere Methoden als je zuvor. Warum also wird die Testautomatisierungslücke immer grösser und kostet Teams nicht nur Zeit und Testabdeckung, sondern behindert auch ihre Fähigkeit zu skalieren und mit modernen Anforderungen Schritt zu halten? Genau diese Frage wollen wir hier beantworten.

Auch wenn es beruhigend sein mag, zu wissen, dass man mit diesem Problem nicht allein ist: Nur weil die Testautomatisierungslücke weit verbreitet ist, bedeutet das nicht, dass sie weniger kritisch ist. Das dauerhafte Verfehlen von Automatisierungszielen wirkt sich direkt auf die Softwarequalität, die Release-Geschwindigkeit und die Effizienz der Teams aus. Da die Anforderungen an schnellere Releases bei gleichzeitig höherer Qualität weiter steigen, ist das Schliessen der Testautomatisierungslücke heute wichtiger denn je.

Dieses E-Book richtet sich an QA Engineers, Testautomatisierungsingenieure, QA Manager, Führungskräfte in der Softwareentwicklung sowie Softwareentwickler, die nach konkreten Erkenntnissen und Strategien suchen, um ihre Automatisierungsbemühungen zu beschleunigen. Die eigene Testautomatisierungslücke kann sich wie eine unüberwindbare Hürde anfühlen. Wir helfen dabei, die Ursachen zu identifizieren und eine Strategie zu entwickeln, um diese Kluft zu überbrücken.

Abschnitt 1

Der Stand der Testautomatisierung heute

Der Stand der Testautomatisierung heute

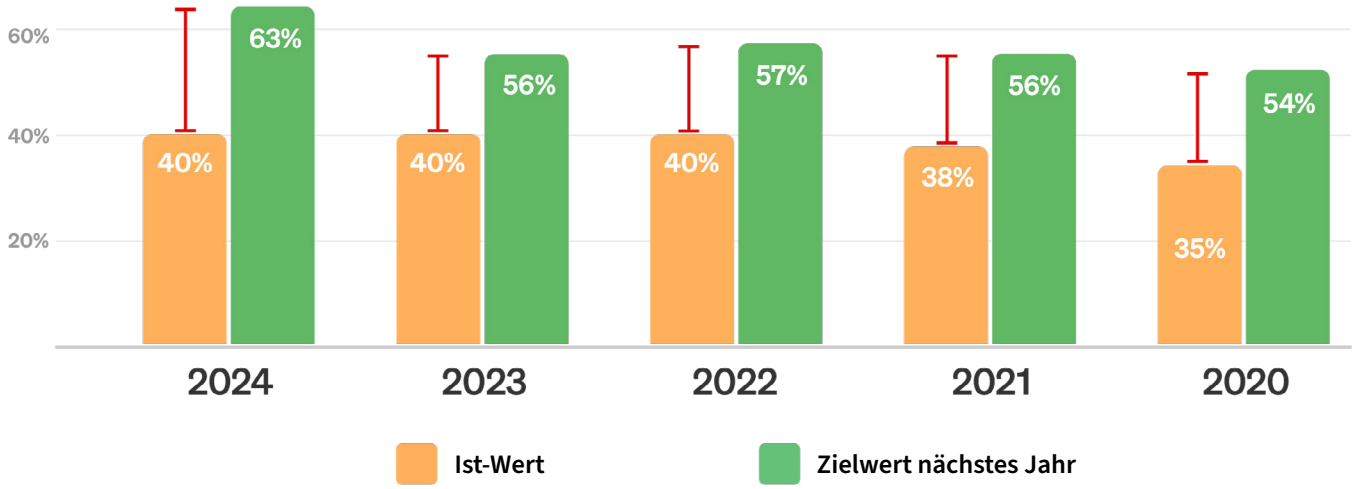
Wie bereits in der Einleitung erwähnt, veröffentlicht TestRail jedes Jahr einen Bericht zum Stand der QA-Branche mit dem Titel Software Testing and Quality Report. Dieser Bericht liefert Statistiken, Benchmarks und Einblicke in Testtools, Teams, Arbeitsweisen und Prioritäten – basierend auf dem Feedback von Tausenden von QA-Fachkräften wie dir.

Testautomatisierung war auch in diesem Jahr ein zentrales Thema im Software Testing and Quality Report, so wie in jedem Jahr seit dessen Einführung. Automatisierung taucht dabei immer wieder sowohl auf der Seite der Erfolge als auch der Herausforderungen auf: Sie ist aktuell oft eine Quelle der Frustration, trägt aber gleichzeitig viel Hoffnung für die Zukunft in sich.

Um Ihnen zu helfen, Ihr eigenes QA-Team mit anderen Teams aus der Branche zu vergleichen, finden Sie hier eine Auswahl von Ergebnissen aus der vierten Ausgabe des Software Testing and Quality Report.

Aktuelle und erwartete Automatisierungsraten

Wie entwickelt sich das Verhältnis automatisierter zu manuellen Tests im kommenden Jahr?



Nichts verdeutlicht die Testautomatisierungslücke besser als dieses Diagramm, das die jahrübergreifende Erwartung der Automatisierungsquote der tatsächlich erreichten Quote gegenüberstellt.

Das «Automatisierungsziel» ist über die Jahre hinweg kontinuierlich gestiegen, mit einem besonders deutlichen Anstieg in 2025 – möglicherweise getrieben durch die Erwartungen an KI oder als Reaktion auf den stetig wachsenden Druck, schnellere bzw. iterative Releases mit weniger Fehlern bereitzustellen.

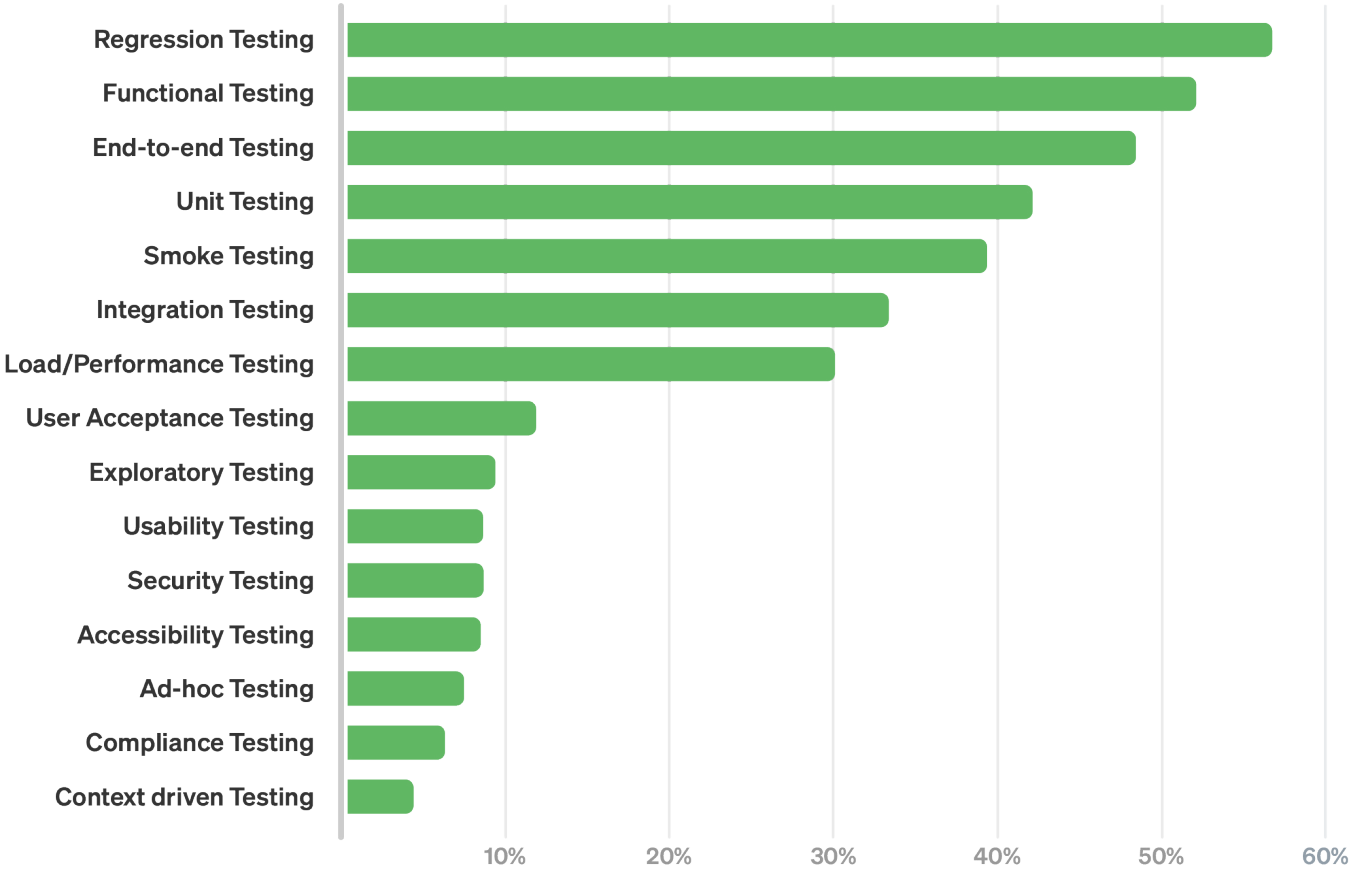
Während die tatsächlich erreichten Automatisierungsraten über die Jahre hinweg deutlich langsamer gestiegen sind, scheinen sie bei rund 40% zu stagnieren.

Dies deutet darauf hin, dass Teams – unabhängig von ihren Bemühungen zur Steigerung der Automatisierung – mit dem steigenden Bedarf bzw. der Erwartung nicht Schritt halten können.

Diese Ergebnisse zeigen, dass die Testautomatisierungslücke kein neues Phänomen ist.

Automatisierte Testarten im Team

Welche Tests werden automatisiert ausgeführt?



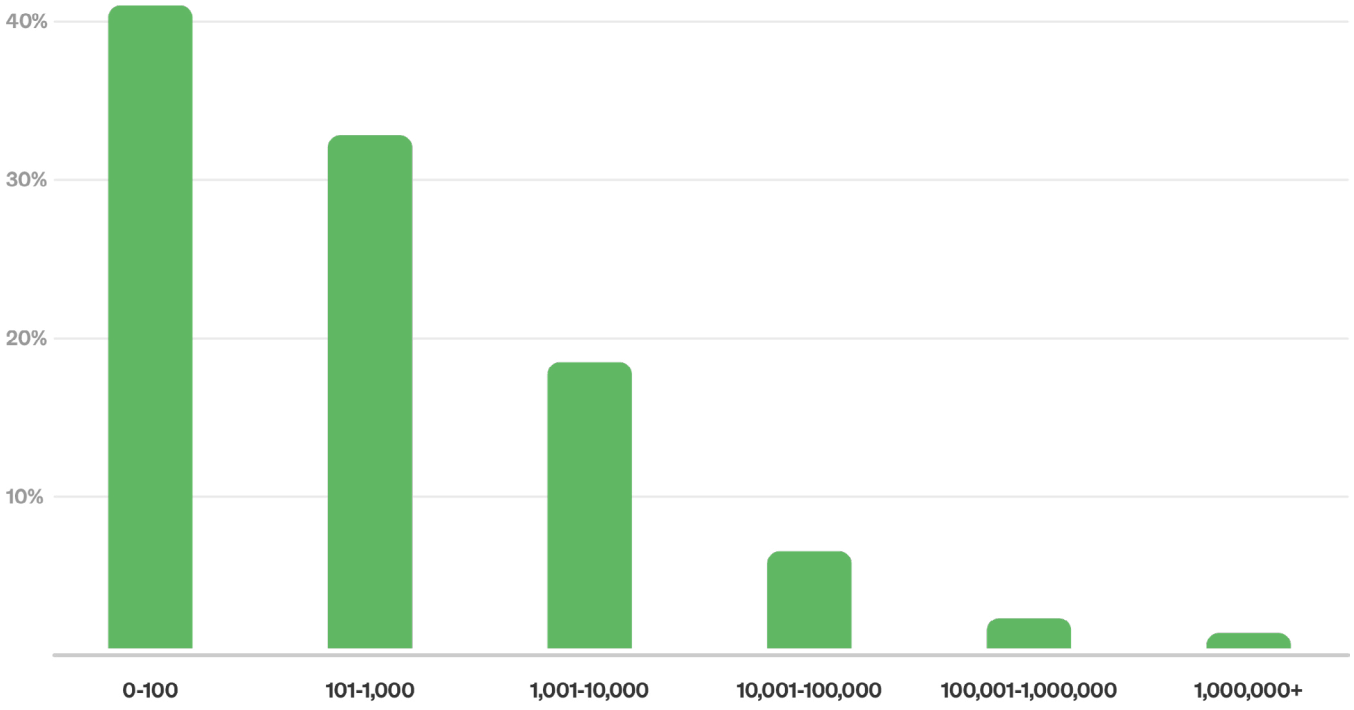
Die Umfrageergebnisse zeigen, dass die meisten Teams ihre Automatisierungsbemühungen auf Testarten konzentrieren, die einen hohen Mehrwert liefern und sich leicht wiederholen lassen, wie zum Beispiel Regressionstests und funktionale Tests. Diese Testarten sind für menschliche Tester manuell sehr zeitaufwendig, sodass ihre Automatisierung die Release-Zyklen beschleunigt und Tester entlastet. Dadurch bleibt mehr Zeit für Aufgaben, die menschliches Urteilsvermögen erfordern.

Viele dieser vorher genannten Aufgaben betreffen andere Testarten. Auch wenn es überraschend klingen mag: 100% Automatisierung ist kein Ziel und sollte es auch niemals sein. Bestimmte Tests, wie exploratives Testen, Usability-, Accessibility-, Security- und Compliance-Tests, benötigen menschliche Intuition und Einschätzungen, um wirksam zu sein. Diese Tests basieren nicht nur auf Instinkt, Wahrnehmung und Kreativität, sondern ihr Automatisierungsversuch führt häufig dazu, dass mehr manuelle Arbeit entsteht, als dadurch eingespart wird.

Ein effizientes Testautomatisierungskonzept konzentriert sich darauf, die richtigen Tests zu automatisieren – nicht jeden Test.

Täglich ausgeführte automatisierte Tests (im Durchschnitt)

Wie viele Tests führt Ihre Organisation durchschnittlich pro Tag aus?



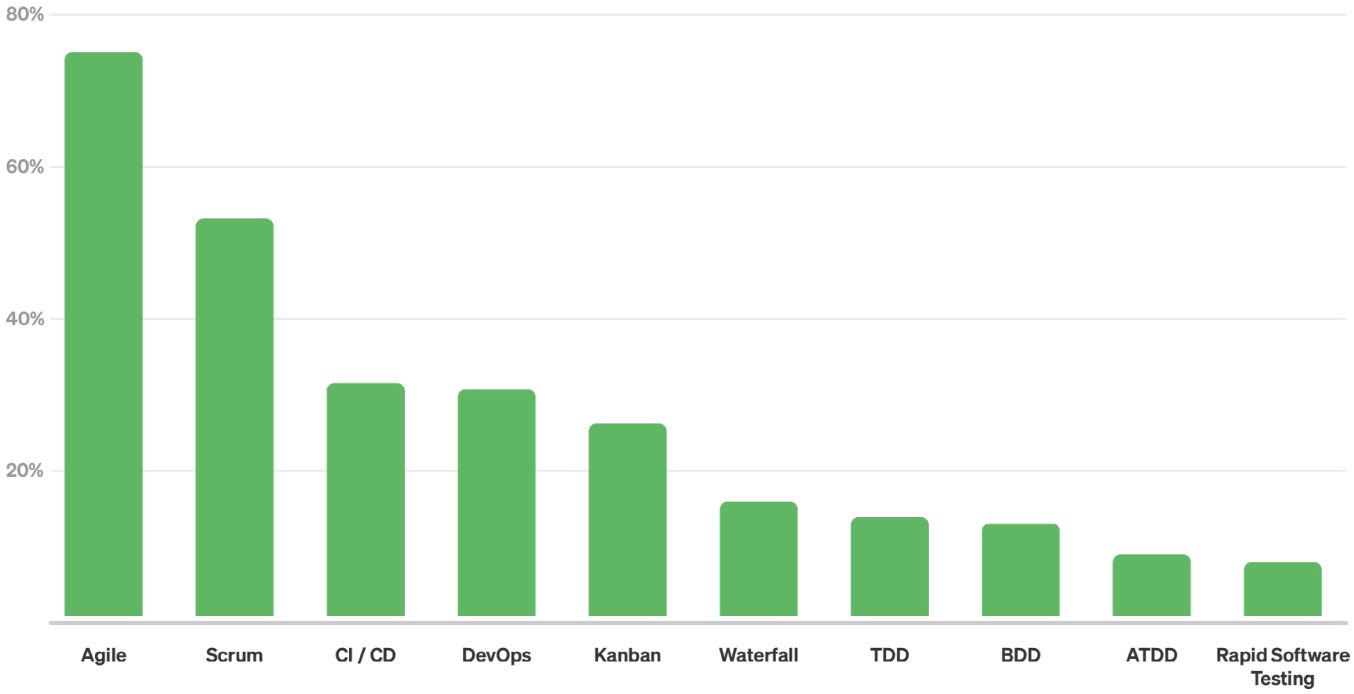
Dieses Diagramm veranschaulicht die Reifegradlücke in der Testautomatisierung. Umfrageteilnehmer berichten von täglich ausgeführten automatisierten Tests in einer Spannweite von null bis über eine Million. Dies spiegelt eine enorme Spannweite des Reifegrads der Testautomatisierung sowie der zugrunde liegenden Infrastruktur in modernen QA-Teams wider.

Die meisten Teams befinden sich weiterhin im frühen bis mittleren Bereich dieser Spannweite. Rund 41% der Teams führen täglich weniger als 100 Tests aus, während weitere 33% zwischen 101 und 1.000 Tests ausführen. Dies deutet darauf hin, dass viele QA-Teams ihre Automatisierungsfähigkeiten aufbauen, jedoch noch nicht das volle Potential erreicht haben.

Gleichzeitig zeigt die rechte Seite des Diagramms mit über 1 Mio. Ausführungen, welches Potenzial bei voller Skalierung mit der richtigen Strategie und passenden Investitionen möglich ist.

Verbreitungsgrad von Continuous Integration und Continuous Delivery (CI/CD)

Welche der folgenden Entwicklungsmethoden oder -techniken nutzt Ihr Team?



CI/CD ist ein zentraler Baustein im Gesamtbild der Testautomatisierung. Durch die Integration von Testautomatisierung in die CI/CD-Pipeline werden automatisierte Tests bei jeder Codeänderung ausgelöst. Dies ermöglicht schnellere Deployments mit höherer Qualität und spart sowohl Testern als auch Entwicklern wertvolle Zeit.

Rund ein Drittel der Teams gibt an, CI/CD zu nutzen. Dies zeigt, dass CI/CD bereits eine relevante Rolle in der Branche spielt, zugleich aber noch erhebliches Wachstumspotenzial besteht. Für die Teams, die CI/CD einsetzen, berichten die Umfrageteilnehmer von erheblichen Vorteilen:

- **86% der Teams mit hohem Automatisierungsgrad und CI/CD-Integration verzeichnen schnellere Release-Zyklen,** verglichen mit nur 42% der Teams mit niedrigem Automatisierungsgrad.
- **71% der Teams mit hoher Testautomatisierung und CI/CD-Integration erleben eine reduzierte Defect Leakage,** im Vergleich zu 35% der Teams mit geringer Automatisierung.
- **58% der Teams, die automatisierte Testausführung an CI/CD koppeln, berichten von einem messbaren Return on Investment (ROI)** innerhalb von sechs Monaten.

Der Aufbau einer CI/CD-Pipeline ist möglicherweise nicht für jedes QA-Team, jede Struktur oder jede Teamgrösse geeignet. Für Teams jedoch, die über die notwendigen Ressourcen und ein ausreichendes Testvolumen verfügen, um eine solche Pipeline zu rechtfertigen, sprechen die Vorteile für sich.

Wo möchten wir im nächsten Jahr stehen?

Neben dem schwer greifbaren Dauerziel, mehr Tests zu automatisieren, hoffen Teams darauf, eine Reihe alltäglicher Herausforderungen in der Testautomatisierung besser in den Griff zu bekommen.

- **45% möchten weniger Testabbrüche**
  UI-Änderungen und häufige Applikationsupdates führen oft dazu, dass automatisierte Tests fehlschlagen. Dies verursacht hohe Wartungskosten und reduziert die Zuverlässigkeit der Tests.
- **32% wünschen sich ein besseres Testdatenmanagement**
  Das Erstellen und Pflegen von Testdaten ist zeitaufwendig, und vielen Teams fehlen strukturierte Prozesse, um eine schnelle und konsistente Automatisierung zu unterstützen.
- **30% möchten mehr qualifiziertes Personal einstellen**
  Der Mangel an erfahrenen Automatisierungsingenieuren erschwert es Teams, stabile und wartbare Testsuiten aufzubauen.
- **29% benötigen Unterstützung bei der Auswahl geeigneter Tools**
  Angesichts der Vielzahl verfügbarer Werkzeuge fällt es Teams schwer, eine Lösung zu finden, die zu ihren spezifischen Anforderungen, dem Tech-Stack und der Komplexität der Anwendung passt.
- **26% möchten instabile und unzuverlässige Tests reduzieren**
  Inkonsistente Testergebnisse bremsen Teams aus und erschweren es, der Automatisierung als Bestandteil der CI/CD-Pipeline zu vertrauen.
- **26% streben eine bessere Testabdeckung an**
  Viele Teams geben an, noch immer nicht ausreichend automatisieren zu können, wodurch Lücken in der Qualitätssicherungsstrategie bestehen bleiben.
- **21% möchten schnellere Testausführungen**
  Lange Testläufe können Engpässe in Release-Prozessen verursachen und schnelle Feedback-Zyklen behindern.
- **19% möchten eine robustere CI/CD-Integration**
  Einige Organisationen arbeiten noch daran, ihre Testautomatisierung besser mit DevOps-Tools und -Workflows zu verzahnen.

Darüber hinaus sehen Teams grosses Potenzial im Einsatz von KI in der Testautomatisierung. 29% der Umfrageteilnehmer berichten bereits von gesteigerter Effizienz und Automatisierung im Testing durch KI. Als vielversprechendste KI-Chance im QA-Bereich für die nächsten fünf Jahre gelten KI-gestützte Testautomatisierung und selbstheilende Testscripte.

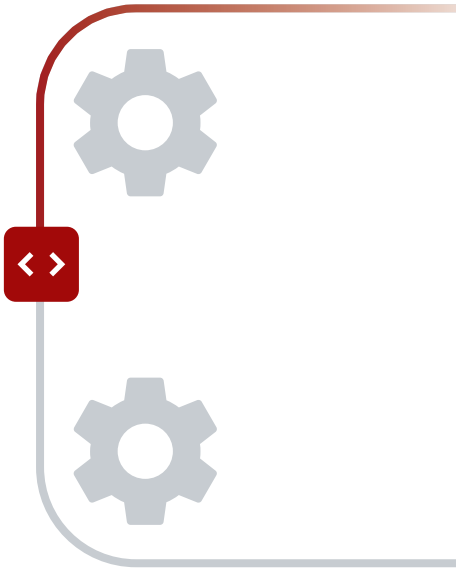
Insgesamt scheinen sich Teams ihrer Automatisierungsdefizite bewusst zu sein und investieren Zeit sowie Ressourcen in Verbesserungen. Dennoch bleibt die Frage bestehen: Warum geht die Lücke zwischen den Zielen der Testautomatisierung und der tatsächlichen Umsetzung Jahr für Jahr weiter auseinander?

Abschnitt 2

Warum QA-Teams Schwierigkeiten haben, ihre Automatisierungsziele zu erreichen

Trotz des zunehmenden Drucks, Tests zu automatisieren, stehen viele QA-Teams vor dauerhaften Hürden, die sie daran hindern, ihre Automatisierungsziele zu erreichen. Diese Herausforderungen sind häufig tief in den Fähigkeiten der Teams, technischen Komplexitäten und organisatorischen Rahmenbedingungen verankert.

Kompetenz- und Qualifikationslücken



Eine wesentliche Hürde ist der mangelnde Programmierhintergrund vieler Tester, der es erschwert, Automatisierungsframeworks effektiv aufzubauen oder zu warten.

Auch die Rekrutierung qualifizierter Automatisierungsingenieure gestaltet sich schwierig, da die Nachfrage das Angebot deutlich übersteigt. Selbst wenn Unternehmen in die Weiterqualifizierung ihres bestehenden QA-Personals investieren, ist es herausfordernd, das Erlernen neuer Fähigkeiten mit den täglichen operativen Aufgaben in Einklang zu bringen.

Dies trägt zu einer umfassenderen Talentressourcenlücke im QA-Bereich bei, insbesondere bei Rollen, die sowohl Test- als auch Programmierkompetenzen erfordern. Darüber hinaus fehlt vielen Teams schlicht das Know-how zur Konzeption und Umsetzung einer guten Testautomatisierungsstrategie, sodass sie ohne klare Ausrichtung und Zielbild agieren.

Komplexität des zu testenden Systems (SUT)

Ein weiterer zentraler Faktor ist die zunehmende Komplexität moderner Anwendungen. Verteilte Systeme mit Cloud-Infrastruktur, Multi-Tenant-Architekturen und Microservices machen das Testen deutlich anspruchsvoller.

Viele bestehende Testwerkzeuge können mit diesen Umgebungen nur schwer Schritt halten, da ihnen entweder die notwendige Flexibilität oder die erforderlichen Integrationsmöglichkeiten fehlen, um eine umfassende Automatisierung in solchen Kontexten zu unterstützen.

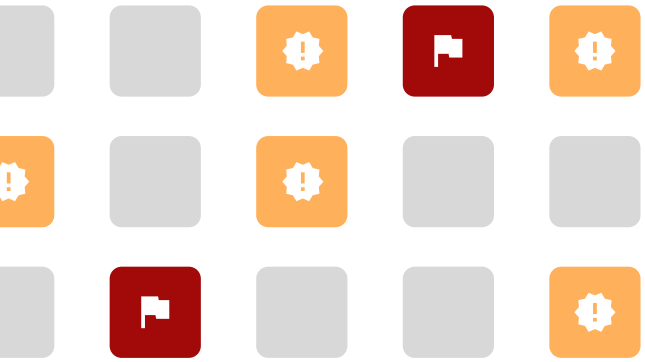
Tool-Fragmentierung und fehlendes zentrales Management

Nicht einheitliche Tool-Landschaften über verschiedene Teams hinweg führen häufig zu fragmentierten Automatisierungsansätzen. Unterschiedliche Gruppen nutzen eigene Werkzeuge und Prozesse, was zu **isolierten Automatisierungslösungen** führt, die sich nur schwer skalieren oder zentral verwalten lassen.

Zudem wird die Zusammenführung von Testergebnissen aus manuellen und automatisierten Quellen erschwert, was eine fundierte Analyse und ein aussagekräftiges Reporting verhindert. Ohne ein zentrales System zur Verwaltung und Nachverfolgung von Testausführung und -ergebnissen kämpfen Teams mit Ineffizienzen und fehlender Transparenz über ihre gesamte Testlandschaft hinweg.

Zeit und Ressourcen

Selbst wenn sich Teams klar für eine Automatisierung geeinigt haben, sind **Zeit und Ressourcen** oft begrenzt. Die Automatisierung komplexer Szenarien ist arbeitsintensiv und wird in schnelllebigen Release-Zyklen selten gegenüber kurzfristigen Lieferzielen priorisiert. Nur wenige Teams verfügen über ausreichende Kapazitäten oder dedizierte Ressourcen, um Automatisierungsskripte regelmässig zu warten und zu aktualisieren. Dies bremst den langfristigen Fortschritt und führt zu einer schleichenden Verschlechterung und Veralterung der Testsuiten.



Teststabilität und Unzuverlässigkeit

Instabile automatisierte Tests stellen einen der häufigsten Schmerzpunkte dar. Hohe Wartungskosten und regelmässige Testfehlschläge – oft verursacht durch UI-Änderungen oder dynamisch generierte Inhalte – erschweren es Teams, sich auf die Automatisierung zu verlassen. Insbesondere die Untersuchung von False-Positives und False-Negatives verursacht hohen zeitlichen Aufwand und reduziert die Verlässlichkeit der Testsuite aus Teamsicht.

Testabdeckungsstrategien

Ein weiteres häufiges Problem ist ein unausgewogener Ansatz bei der Testabdeckung. Viele Teams fokussieren sich übermässig auf UI-basierte Automatisierung und vernachlässigen dabei andere kritische Ebenen wie API-, Integrations- oder Unit-Tests. Ohne einen strukturierten Ansatz zur Priorisierung dessen, was automatisiert werden sollte, sind die Automatisierungsvorhaben oft nicht auf tatsächliche Risiko- und Wirkungsbereiche ausgerichtet, was zu einem geringen Return on Investment führt.

Fehlende Unterstützung durch das Management und die Organisation

Schliesslich scheitern Automatisierungsvorhaben häufig an fehlender Unterstützung durch das Management. Führungskräfte stehen dem Return on Investment oft skeptisch gegenüber, insbesondere wenn Vorteile wie Zeitersparnis oder erhöhte Zuverlässigkeit nicht klar belegt werden können. Ohne die Fähigkeit, den Mehrwert der Automatisierung messbar darzustellen und wirksam zu kommunizieren, gelingt es Teams nur schwer, die notwendigen Ressourcen und das erforderliche Commitment zu sichern, um ihre Automatisierungsvorhaben zu skalieren.

Abschnitt 3

Testautomatisierung beschleunigen: Strategien für den Erfolg

Wenn Ihnen einige der vorher beschriebenen Herausforderungen bekannt vorkommen, sind Sie nicht allein. Die gute Nachricht ist: Die Testautomatisierungslücke lässt sich schliessen, unabhängig davon, wie unüberwindbar sie aus der aktuellen Perspektive erscheinen mag. Bereits kleine, schrittweise Verbesserungen in einem oder mehreren der folgenden Bereiche können schnell messbare Fortschritte in der Automatisierung bewirken.

Schulung und Weiterbildung fördern

Schulungs- und Zertifizierungsprogramme bieten die Möglichkeit, wertvolle Qualifikationen zu erwerben, die dem Team Struktur verleihen. Insbesondere für weniger erfahrene Tester kann gezielte Weiterbildung eine komplexe und unübersichtliche Landschaft in einen klaren Rhythmus verwandeln: lernen, ausprobieren, reflektieren, wiederholen. Jedes abgeschlossene Modul stärkt dabei nicht nur die Fähigkeiten, sondern auch das professionelle Selbstverständnis.

Die aktive Unterstützung und Priorisierung von Weiterbildung signalisiert dem Team, dass Lernen ein fester Bestandteil der Arbeit ist und kein nachrangiges Thema. Langfristig verändert dies die Teamkultur: Wissensaustausch nimmt zu, Diskussionen verlagern sich hin zu Standards und Best Practices, und die Personalgewinnung wird durch klare Entwicklungspfade erleichtert. Auf diese Weise wird Weiterbildung mehr als nur eine kurzfristige Lösung für Qualifikationslücken – sie wird zur Grundlage kontinuierlicher Verbesserung.

Actionplan

- Relevante Zertifizierungsprogramme für das Team identifizieren und aktiv unterstützen.
- Regelmässiges Lernen als festen Bestandteil des Wochenablaufs etablieren (z. B. 1 Stunde pro Woche).
- Raum für internen Wissensaustausch schaffen (z. B. Lightning Talks, Lunch-and-Learn-Formate).
- Kompetenzentwicklung in Jahresziele und Karrierepfaden verankern.

Das richtige Tool wählen

Auf dem Markt gibt es eine Vielzahl unterschiedlicher Automatisierungstools, und jedes davon kann potenziell die beste Wahl für ein bestimmtes Vorhaben sein. Während Bewertungsplattformen eine erste Orientierung bieten können, lässt sich das passende Tool für die eigenen Anforderungen letztlich nur team- und kontextspezifisch bestimmen.

Bei der Auswahl von Automatisierungstools ist es entscheidend, die Komplexität der Anwendung, den Tech-Stack, den Reifegrad der Automatisierung, die technischen Fähigkeiten des Teams sowie selbstverständlich das verfügbare Budget zu berücksichtigen.

Für Teams mit geringem Automatisierungsreifegrad, die gerade erst mit Testautomatisierung beginnen und überschaubare Testanforderungen haben, sind Low-Code- bzw. No-Code-Lösungen in der Regel gut geeignet. Reifere Teams, die bereits weiter in der Automatisierung sind und komplexe Testanforderungen abdecken müssen, benötigen hingegen meist eine leistungsfähigere und flexiblere Plattform. Unabhängig von der gewählten Lösung ist eine nahtlose Integration in den bestehenden Tech-Stack entscheidend, damit sich das Tool sinnvoll in den bestehenden Workflow integriert.

Actionplan

- Zusammensetzung des Teams, aktuellen Automatisierungsgrad und Tech-Stack bewerten.
- Verfügbare Automatisierungstools recherchieren und mit den eigenen Anforderungen und Nutzen abgleichen.
- Testversionen nutzen, um Funktionalität und Workflow-Integration zu prüfen.
- Ein Onboarding etablieren, um das Team optimal im Umgang mit dem Tool zu schulen.

Wissensaustausch teamübergreifend fördern

Das regelmässige Zusammenbringen von manuellen Testern und Test Automation Engineers ist eine vergleichsweise niedrigschwellige Massnahme mit hohem Nutzen. Manuelle Tester bringen ein tiefes Verständnis für das Produkt, die Nutzerintention sowie für Randfälle mit, die in der Dokumentation häufig nicht abgebildet sind. Test Automation Engineers hingegen verfügen über fundierte Kenntnisse zu Tools, Frameworks und technischen Workflows. Das Zusammenbringen beider Perspektiven führt zu einer qualitativ besseren Testautomatisierung.

Wissen wird dadurch verteilt. Manuelle Tester gewinnen Sicherheit im Umgang mit technischen Werkzeugen, indem sie diese in der Praxis ausüben und theoretisches Wissen mit konkreter Arbeit verknüpfen. Test Automation Engineers profitieren wiederum von neuen Testideen und explorativen Denkansätzen, die sie zuvor möglicherweise nicht berücksichtigt hätten. So entsteht ein Umfeld, in dem automatisiertes Testen bewusst, durchdacht und strategisch umgesetzt wird.

Actionplan

- Manuelle Tester regelmässig mit Test Automation Engineers zusammenarbeiten lassen.
- Gemeinsame Testdesign Sessions vor grösseren Sprints durchführen.
- Wiederverwendbare Muster und Lessons Learned dokumentieren und teamweit teilen.
- Teammitglieder rollenübergreifend rotieren lassen, um Empathie und fachliche Tiefe aufzubauen.

Automatisierungs- und Testmanagementtools gezielt einsetzen

In vielen Organisationen ist Testautomatisierung über mehrere Orte verteilt – von lokalen Skripten über CI-Dashboards bis hin zu Tabellen. Diese zersplitterte Struktur erschwert den Blick auf das Gesamtbild. Testmanagement-Tools schaffen hier Abhilfe, indem sie manuelle und automatisierte Tests zentral zusammenführen und übersichtlich abbilden. Für Tester entsteht so mehr Klarheit über Abdeckung, Status und offene Handlungsfelder.

Werden Tools wie Ranorex und TestRail integriert, ist der Unterschied im täglichen Arbeitsablauf unmittelbar spürbar. Ein in Ranorex erstellter automatisierter Test kann direkt mit einem Testfall in TestRail verknüpft werden, wo er gemeinsam mit manuellen Schritten, Historie und dem Kontext von Testläufen verwaltet wird. Dadurch entsteht Echtzeit-Transparenz über Ausführungsergebnisse und eine deutlich verbesserte Zusammenarbeit.

Actionplan

- Alle manuellen und automatisierten Testfälle in einer einheitlichen Plattform zentralisieren.
- Tools wie Ranorex und TestRail zur Sicherstellung von Nachverfolgbarkeit und Transparenz integrieren.
- Testfälle mit User Stories sowie CI/CD-Ereignissen verknüpfen.
- Testabdeckung und Ausführungsdaten regelmässig gemeinsam im Team reviewen.

Zuverlässige und robuste Automatisierung bauen

Das Vorantreiben der Testautomatisierung ist wirkungslos, wenn die Tests instabil oder fragil sind. Zuverlässigkeit sollte von Anfang an ein zentrales Designziel sein. Anstelle willkürlicher «Delays» sollten explizite «Waits» verwendet werden, Tests sollten voneinander isoliert sein, um Probleme durch gemeinsam verwendete States zu vermeiden und in sauberen Umgebungen ausgeführt werden. Die Testsuite sollte sinnvoll geschichtet aufgebaut werden: Unit-Tests für Geschwindigkeit und Präzision, API-Tests zur Validierung der Logik und UI-Tests ausschliesslich dort, wo eine End-to-End-Abdeckung den grössten Mehrwert liefert.

Ebenso wichtig ist die Robustheit der Tests, insbesondere in der UI-Automatisierung. Änderungen an der UI-Oberfläche können dazu führen, dass automatisierte Tests bereits bei kleinen Anpassungen fehlschlagen und dadurch mehr Wartungsaufwand als Nutzen erzeugen. Der Fokus sollte daher auf der Intention der Interaktion liegen: Wird die Platzierung eines Buttons getestet oder das Nutzererlebnis? Je stärker Tests am Nutzerverhalten und an den Systemzielen ausgerichtet sind, desto besser halten sie veränderlichen UIs und wachsenden Codebasen stand.

Zusätzlich kann auch die Wahl des richtigen Tools zur Robustheit der Tests beitragen. Es empfiehlt sich, ein Automatisierungstool auszuwählen, dessen Funktionen und Fähigkeiten den Aufbau zuverlässiger und widerstandsfähiger Tests unterstützen, etwa durch Self-Healing-Mechanismen oder Objekterkennung. Auch wenn solche Funktionen das Testdesign nicht vollständig automatisieren können, reduzieren sie Flakiness erheblich und schaffen Freiräume, in denen sich Testanalysten auf andere Aufgaben konzentrieren können.

Actionplan

- Ein Tool wählen, das zuverlässiges Testdesign unterstützt, und Tests so gestalten, dass sie unabhängig in sauberen Umgebungen ausgeführt werden.
- Explizite Waits einsetzen und den Systemzustand vor Interaktionen verifizieren.
- Auf stabile Selektoren (z. B. data-test-id) und wiederverwendbare Aktionen setzen.
- Assertions auf Nutzerergebnisse ausrichten, nicht ausschliesslich auf UI-Sequenzen.

KI gezielt einsetzen, um die Zuverlässigkeit der Testautomatisierung zu verbessern

KI wird Tester weder ersetzen noch instabile Tests auf einen Schlag stabilisieren. Ihr eigentlicher Mehrwert liegt darin, menschliche Arbeit sinnvoll zu unterstützen, insbesondere bei repetitiven oder komplexen Aufgaben. Auch wenn KI grosses Potenzial bietet, hat sich KI-gestütztes QA-Tooling bislang noch nicht flächendeckend etabliert. Ein Bereich, in dem die Branche jedoch bereits verlässliche Ergebnisse sieht, ist im Testdesign und der Testoptimierung.

Tools wie DesignWise können KI-Algorithmen nutzen, um Testfälle effizienter zu entwerfen. Dabei werden Empfehlungen generiert, die unnötigen Aufwand reduzieren und gleichzeitig eine angemessene Abdeckung sowie die Priorisierung kritischer Pfade sicherstellen. Darüber hinaus kann KI helfen, Lücken in der Testabdeckung zu identifizieren, Parametrisierung zu automatisieren, Gherkin-Testscripte zu erzeugen und umfangreiche Testergebnisse so zu filtern, dass relevante Informationen hervorgehoben werden.

Gleichzeitig bringt der Einsatz von KI auch Einschränkungen mit sich. Allgemeine Modelle oder falsch eingesetzte Automatisierung können stille Fehler verursachen, wenn Teams Ergebnisse ungeprüft übernehmen, ohne sie zu verstehen. Richtig eingesetzt erweitert KI jedoch den Handlungsspielraum, sodass sich Tester auf Urteilsvermögen, Priorisierung und Testdesign konzentrieren können.

Actionplan

- KI zur Unterstützung bei der effizienten Erstellung von Testfällen einsetzen.
- KI-Tools nutzen, um Lücken in der Testabdeckung zu identifizieren.
- Flakiness-Muster analysieren, um Wartungsaufwand gezielt zu priorisieren.
- Alle KI-basierten Änderungen überprüfen – Ergebnisse nicht blind übernehmen.

Eine nachhaltige Strategie entwickeln und Rückhalt sichern

Eine starke Automatisierungsstrategie beginnt mit Klarheit: Definieren Sie, was erreicht werden soll, wie Erfolg gemessen wird und welche Voraussetzungen dafür notwendig sind. Setzen Sie realistische Ziele auf Basis bisheriger Erfahrungen und berücksichtigen Sie sowohl Risiken als auch den erwarteten Return on Investment (ROI). Nicht jeder Test sollte automatisiert werden. Kriterien wie Ausführungshäufigkeit, Kritikalität und Komplexität helfen dabei, fundierte Entscheidungen zu treffen. Orientieren Sie sich an der Testautomatisierungspyramide: Unit-Tests für Geschwindigkeit und Zuverlässigkeit priorisieren, diese durch API- und Integrationstests ergänzen und UI-Tests nur dort einsetzen, wo End-to-End-Abdeckung wirklich notwendig ist.

Ebenso wichtig wie die Planung ist das Sichern von Unterstützung. Teilen Sie Ihre Roadmap mit Stakeholdern, zeigen Sie auf, wie Automatisierung die Geschäftsziele unterstützt, und definieren Sie klare Kennzahlen zur Messung des Nutzens. Mit Rückhalt durch das Management und einer fokussierten, anpassungsfähigen Strategie sind Sie deutlich besser positioniert, um Testautomatisierung nachhaltig zu skalieren.

Actionplan

- Die Testautomatisierungspyramide nach Mike Cohn als Leitlinie nutzen.
- Erfolgskriterien definieren und den ROI von Beginn an messen.
- Automatisierung selektiv und zielgerichtet anhand des Testpyramiden-Modells einsetzen.
- Automatisierungsziele mit Produkt- und Geschäftsprioritäten abstimmen.
- Die Roadmap klar präsentieren und Stakeholder frühzeitig einbinden.

Abschnitt 4

Die Zukunft der Testautomatisierung: Vorhersagen für das kommende Jahr

Die Testautomatisierung befindet sich in einer tiefgreifenden Weiterentwicklung. Mit intelligenteren Tools, schnelleren Release-Zyklen und strategischeren QA-Rollen übernimmt Automatisierung zunehmend umfangreichere Verantwortung. In den kommenden Jahren ist zu erwarten, dass Testautomatisierung mehr Komplexität bewältigt, sich nahtloser in DevOps integriert und Shift-Left-Testing zum Standard wird. Um Schritt zu halten, müssen QA-Teams agil bleiben, neue Technologien adaptieren und ihren Beitrag zur Software Delivery überdenken.

KI-gestützte Testautomatisierung

KI entwickelt sich rasant zu einem echten Game Changer in der Testautomatisierung. Tools mit Self-Healing-Funktionen können fehlerhafte Testscripte automatisch reparieren und so den manuellen Wartungsaufwand deutlich reduzieren. Auf Basis von Machine Learning sind diese Tools in der Lage, Testfälle zu generieren, Lücken in der Testabdeckung aufzudecken und anhand historischer Daten vorherzusagen, wo Fehler mit hoher Wahrscheinlichkeit auftreten werden. Auch wenn sich diese Ansätze noch in der Entwicklung befinden, wird KI zunehmend zu einem zuverlässigen Co-Piloten, der Tester dabei unterstützt, Releases zu beschleunigen, ohne die Qualität zu beeinträchtigen.

Etablierung des Shift-Left-Testing

„Shift Left“ ist längst kein reines Schlagwort mehr, sondern hat sich als fester Bestandteil des Softwareentwicklungslebenszyklus etabliert. Indem Tests früher ansetzen – oft bereits in der Design- oder Implementierungsphase – können Teams Probleme schneller erkennen und beheben. Diese frühen Feedback-Schleifen reduzieren Nacharbeit und rücken Qualität dauerhaft in den Fokus. Werden automatisierte Tests fest in CI/CD-Pipelines integriert, wird Testen zu einem kontinuierlichen Prozess statt zu einer abschliessenden Kontrollinstanz. Mit zunehmender Geschwindigkeit der Release-Zyklen wird Shift-Left-Testing entscheidend sein, um Agilität und Stabilität sicherzustellen.

Durchgängige Automatisierung entlang der DevOps-Pipeline

DevOps und Testautomatisierung wachsen zunehmend zu einem einheitlichen, automatisierten Workflow zusammen. Tests werden mit jedem Code-Commit ausgeführt und liefern sofortiges Feedback sowie verwertbare Erkenntnisse. Diese enge Integration verbessert die Codequalität, beschleunigt Releases und stellt sicher, dass das Testing mit der Entwicklung Schritt hält. Gleichzeitig fördert sie die Zusammenarbeit zwischen Entwicklung, Testing und Betrieb und schafft einen transparenteren und reibungsloseren Delivery-Prozess.

Vom Anspruch zur Umsetzung: Die Testautomatisierungslücke schliessen

Trotz der Reife moderner Testwerkzeuge und -methoden verfehlen viele QA-Teams weiterhin ihre Automatisierungsziele. Die Ursachen sind klar und wiederkehrend: begrenzte Zeit, Qualifikationslücken, instabile Tests, uneinheitliche Tool-Landschaften und mangelnde organisatorische Unterstützung. Diese Faktoren bremsen den Fortschritt, sind jedoch keineswegs unüberwindbar.

Mit wachsender Reife und stärkerer Integration der Automatisierung können Tester zunehmend Einfluss auf den gesamten Entwicklungsprozess nehmen – von der Implementierung bis zum Deployment. Fähigkeiten wie Scripting, Datenanalyse und der Umgang mit KI-gestützten Tools werden dabei immer wichtiger. Gleichzeitig erweitern Low-Code- und No-Code-Plattformen den Kreis der Personen, die zur Automatisierung beitragen können. QA-Rollen befinden sich im Wandel – diesen aktiv anzunehmen ist entscheidend, damit Teams langfristig erfolgreich bleiben.

Die Wahl der richtigen Tools spielt dabei eine Schlüsselrolle. Lösungen wie Ranorex und TestRail unterstützen QA-Teams dabei, die Testautomatisierungslücke zu schliessen, indem sie Automatisierung effizient skalierbar machen, Komplexität reduzieren und die Zusammenarbeit über den gesamten Entwicklungslebenszyklus hinweg verbessern. Eine Testversion oder eine Demo bietet die Möglichkeit, diese Fähigkeiten direkt in der Praxis kennenzulernen.

Über Infometis

Infometis ist dein Partner für Softwarequalität.

Wir helfen dir, dein Testing zu optimieren.

Wir stehen für wertschöpfende und nachhaltige Teststrategien – vom Testmanagement bis zur Testautomatisierung. Wir erhöhen die Testeffizienz durch gezielte Automatisierung und verkürzen so deine Time-to-Market. Manuelle Tätigkeiten und Engpässe werden auf ein Minimum reduziert, Risiken in Software-Releases nachhaltig minimiert.

Als langjähriger Partner führender Testmanagement- und Testautomationslösungen verfügen wir über die Erfahrung und Expertise, innovative Technologien optimal einzusetzen. Wir stellen den risikobasierten Ansatz ins Zentrum und berücksichtigen deine Prozesse End-to-End. Funktionale und nicht funktionale Qualitätsaspekte fliessen dabei gleichermassen ein.

Als Partner etablierter Tools wie TestRail und Ranorex unterstützt Infometis dich in sämtlichen Facetten eines modernen Testmanagements und der Testautomatisierung – von der Integration in deine Toollandschaft bis zu massgeschneiderten Services und Beratung rund ums Testing und die Automatisierung.

Infometis – dein Partner für Softwarequalität & Automation

Entdecke unsere Dienstleistungen

Kontaktiere uns