

STRATEGIE

11. April 2024

Testautomatisierung? Das sollten Sie vor der Einführung unbedingt wissen ...

Immer mehr Unternehmen nutzen bei der Einführung oder dem Release neuer Software-Versionen die Testautomatisierung.

Automatisierungsquoten von durchschnittlich 60 bis 70 Prozent lassen sich damit erzielen. Das spart Zeit und Ressourcen beim Testen und erhöht insgesamt die Softwarequalität.

Allerdings entstehen beim Einsatz der Testautomatisierung häufig Fehler, die vermeidbar wären. Die Testautomatisierung verspricht viele Vorteile, ganz trivial ist ihre Einführung allerdings nicht. Warum und welche Stolpersteine es gibt, erfahren Sie im folgenden Beitrag.



Quelle: Innobis

von Marco Nowak, Senior Consultant Consulting & Testing Services bei Innobis

In einer Welt, in der die Agilität und Schnelligkeit der Softwareentwicklung immer mehr an Bedeutung gewinnen, hat die Testautomatisierung einen festen Platz erobert. Von der Auswahl der richtigen Testfälle bis hin zur Unterschätzung der Komplexität – Fünf Schlüsselemente, die für eine effektive Testautomatisierung entscheidend sind:

1. Falsch gewählte zu automatisierende Testfälle

Die Auswahl der Testfälle, die zu automatisieren sind, ist einer der wichtigsten Aspekte bei der Umsetzung der Testautomatisierung. So ist es nicht hilfreich, Testautomatisierungsskripte für Oberflächen anzufertigen, die einem permanenten Wandel unterliegen, dies kann zum Beispiel eine Seite mit sich ständig ändernden Inhalten wie ein News- oder Aktienticker sein. Dies würde dazu führen, dass die Testautomatisierung permanent angepasst werden müsste und dadurch der Aufwand höher wäre als das manuelle Testen.

“*Das Ziel sollte sein, sich auf wiederkehrende gleiche Testfälle zu konzentrieren – sogenannte Regressionstests.*”

Diese Regressionstests sind nötig, um zu prüfen, ob die Tests nach einer Änderung in der Software noch immer erfolgreich durchlaufen.

Darüber hinaus ist es zu Beginn der Einführung einer Testautomatisierung förderlich, wenn sich die Softwaretester auf einen kleinen Teil von Testfällen konzentrieren, mit denen sie einen Prototyp schaffen. Mit solchen Prototypen lassen sich erste Erfolge ableiten wie z. B. die erwartbare Zeitersparnis und auch der Aufwand für die weitere Umsetzung ist besser planbar.

2. Kein strukturiertes Vorgehen

Wie in vielen Bereichen der heutigen IT ist es auch für die erfolgreiche Einführung der Testautomatisierung wichtig, das Vorgehen detailliert zu strukturieren. In unserem Fall heißt das:

“*Es ist nicht zielführend, die Testautomatisierungsskripte in einem Stück herunterzuschreiben. Vielmehr hat es sich bewährt, eine Funktionsbibliothek zu etablieren.*”

Dabei werden sich wiederholende Bausteine in einer Funktion hinterlegt und im eigentlichen Testskript nur die Funktion aufgerufen: Dies kann zum Beispiel bei Anmeldedaten der Fall sein oder bei der Eingabe von personenbezogenen Daten.

Zu einem strukturierten Vorgehen gehört außerdem die vorausschauende Planung eines ausführlichen Reportings. So ist es bereits bei der Erstellung von Skripten sinnvoll, definierte Prüfungen oder Screenshots für ein Reporting festzuhalten, da ansonsten

Mehraufwand bei einer späteren Umsetzung entsteht. Das Reporting sollte neben der eigentlichen Testausführung auch automatisch erzeugt werden.

3. Fehlendes Konzept für die Testdaten

Die Softwaretester sollten sich rechtzeitig Gedanken über die Verwendung der Testdaten machen. Wer ist der Abnehmer dieser Testdaten und für welchen Zweck werden diese genutzt? Es ist wichtig darauf zu achten, keine Testdaten wie beispielsweise Vor- und Nachname direkt in den Skriptcode zu schreiben.

“ *Um im Test flexibel zu bleiben, ist es notwendig, die Testdaten von außen mitzugeben, etwa über eine Exceltabelle oder eine angebundene Datenbank.* ”

Dafür muss das Testskript so geschrieben werden, dass es in der Lage ist, die Daten auch von außen aufzunehmen. Hier ist es ebenfalls sinnvoll, Testdaten dynamisch zu generieren, um eine möglichst breite Abdeckung unterschiedlicher Konstellationen zu erhalten.

Zu einem guten Konzept gehört weiterhin die Planung der Ausgabe von Testdaten. Ist ein Testskript durchgelaufen und hat auf dem zu testenden System etwas erzeugt, wie beispielsweise eine neue Vertragsnummer, ist es wichtig, die Ausgabe und Weiterverarbeitung der Werte zu definieren, um im Anschluss mit diesen weiterarbeiten zu können.

Autor Marco Nowak, Innobis



Marco Nowak ist Senior Consultant im Team Consulting & Testing Services bei Innobis ([Website \(https://www.innobis.de/\)](https://www.innobis.de/)), leitet dort seit über 10 Jahren große Testmanagement-Vorhaben im Umfeld von SAP-Anwendungen bei Banken. Er ist Experte für die Entwicklung und Praxiseinführung von Methoden und Verfahren in diesem Kontext. Seine berufliche Laufbahn startete Marco Nowak als Bankkaufmann und später Privatkundenberater bei der Nord-Ostsee Sparkasse. Danach absolvierte er seinen Bachelor of Science in Wirtschaftsinformatik an der Fachhochschule Flensburg und später den Master of Science in Wirtschaftsinformatik an der Universität Hamburg.

4. Vernachlässigung automatisierter Testfälle

Es kommt immer wieder vor, dass ein automatisierter Testfall nach längerer Zeit einer nochmaligen Ausführung bedarf, diese aber nicht mehr funktionsfähig ist. Kontinuierliche Änderungen an den Testsystemen können ursächlich dafür sein. Um das zu vermeiden, empfiehlt es sich, die automatisierten Testfälle regelmäßig, z. B. monatlich, auszuführen – auch wenn der konkrete Testfall in dem Moment gar nicht benötigt wird. Dies kann zum Beispiel voll automatisiert durch eine Batch-Einplanung erfolgen, dessen Ergebnis an definierte Abnehmer geschickt wird. Dank der Regelmäßigkeit können die Softwaretester sofort auf Veränderung am testenden System reagieren und die Testfälle bleiben stets einsatzbereit.

5. Falsche Zielvorstellungen und unterschätzte Komplexität

Die Erfahrung zeigt, dass Softwaretester manchmal mit falschen Erwartungen an die Testautomatisierung herangehen. Sie können nicht davon ausgehen, dass alle bisherigen Testfälle auch automatisiert laufen. Grund dafür können technische oder fachliche Restriktionen sein.



Eine Quote von 70 bis 80 Prozent an Automatisierungsgrad ist ein sehr gutes Ergebnis.”

Ein Beispiel aus der Praxis: Bei einer deutschen Förderbank ließen sich bei einem großen Releasewechsel eines SAP-basierten Add-ons von 1.055 Testfällen 600 automatisieren. Die Ausführungsdauer pro automatisiertem Testfall verkürzte sich von 20 auf sechs Minuten. Insgesamt eine immense Zeitersparnis für das Institut.



Weiterhin sind auch die Wartung und Komplexität beim Einsatz der Testautomatisierung nicht zu unterschätzen.”

IT-Landschaften unterliegen einem stetigen Wandel. Dies hat auch Einfluss auf die Testautomatisierung. Änderungen sind stets zeitnah und zentral nachzuziehen.

Abgesehen davon sollte Zeit und Energie in den Ausbau des Know-hows bei den Softwaretestern fließen, sodass alle beteiligten Personen (Ersteller und Nutzer der Tests) stets auf dem gleichen Stand sind. Dies kann durch regelmäßige Abstimmungen, in denen die Änderungen vorgestellt werden, oder eine ausführliche Dokumentation der

Skripte erfolgen. Es kommt durchaus vor, dass die Ersteller automatisierter Skripte nicht die gleichen Personen sind, die die Testfälle am Ende auch ausführen und die Ergebnisse prüfen.

Fazit

Unterm Strich bleibt festzuhalten: Der Einsatz von Testautomatisierung ist in jedem Fall gewinnbringend. Aber:

“*Der Start will durchdacht, das Vorgehen klar strukturiert und die anschließende Wartung gut organisiert sein.*”

Wenn Softwaretester diese Aspekte berücksichtigen, ist das Potenzial der Testautomatisierung enorm. Testaufwand und Testzeitraum lassen sich erheblich reduzieren, kritische Fehler schneller aufdecken und beheben. Die Qualität der Software steigt – und auch die Zufriedenheit der Tester.

Marco Nowak, Innobis ■

Sie finden diesen Artikel im Internet auf der Website:

<https://itfm.link/209006>

