

Testautomatisierung umgekehrt

Wie man mit dem entsprechenden Aufbau der IT-Organisation wertvolle Testautomatisierer sparen kann

Testautomatisierung ist ein fachliches Stiefkind für IT-Fachleute: Für die Fachtester ist die Automatisierung viel zu technisch, für technikorientierte Entwickler aber nicht besonders attraktiv. Die Folge ist, was jeder Ökonom weiß: Raritäten, also Testautomatisierer, sind teuer.

Der klassische Capture-Replay-Weg von Testautomatisierung – Testfälle aufnehmen, anpassen, abspielen – ist aber definitiv technikerbedürftig, das bezweifelt keiner, der sich schon einmal die generierten Sourcedateien eines sogar ganz einfachen Testfalles angesehen hat. (Wer das nicht glaubt, soll den Windows-Standardrechner starten, den Testfall „2+2“ mit dem Testwerkzeug aufnehmen und sich dann überraschen lassen.)

Muss aber ein so einfacher Testfall wie „2+2“ nach der Automatisierung wirklich so unverständlich aussehen? Was wäre, wenn die Fachtester auch in der automatisierten Umgebung die bekannten Testschritte etwa wie `Gib ein: 2+2` und `Prüfe Wert: Ergebnis=4` eingeben könnten?

Die Automatisierungswerkzeuge bieten i.d.R. die Trennung der fachlichen und der technischen Funktionen an. Diese Tren-

nung der beiden Funktionalitäten kann sich aber auch im Aufbau der Personalstruktur widerspiegeln.

Die Fachtester bleiben auch in der automatisierten Umgebung bloß für den fachlichen Inhalt der Testdateien verantwortlich. Der technische Hintergrund wird von den Automatisierern in gut lesbare Bezeichner eingekapselt, diese Bezeichnerstruktur dient als Schnittstelle für die Fachtester. Die verständliche und „fachtesterfreundliche“ Struktur der ggf. deutschsprachigen Bezeichner (im oberen Beispiel z.B. ganz konkret „Ergebnis“ und nicht „MainWindow-ResultJTextField“) ist ein Schwerpunkt der Testarchitektur.

Die Trennung der Aufgaben kann noch mit einem einfachen Übersetzungstool (Parser) abgerundet werden, mit welchem dann die gut lesbaren und schreibbaren Testschritte wie `Gib ein: 2+2` in ein Programmiersprachenformat wie zum Beispiel `Main.Keyboard.Input („2+2“)`; umgewandelt werden können.

Zusammenfassend besteht der Testautomatisierungsprozess also aus den folgenden Schritten:



1. Die Testschritte werden von den Fachtestern in einem menschlich lesbaren Format mit minimalen syntaktischen Anforderungen manuell abgeschrieben.
2. Dieser Text wird in Programmzeilen („Testdatei“) konvertiert, das Tool wird vom Automatisierungsteam zur Verfügung gestellt.
3. Die Testdatei, d.h. der Testfall, wird mittels des Testtools und der Framedateien durchgeführt.

Nicht nur die Fachtester können sich auf eine solche Struktur freuen: Die Testautomatisierer können sich von dem fachlichen Inhalt der Testfälle fernhalten, ihnen bleiben die Entwicklungsaufgaben und die Verwaltung der Architektur.

Erfahrungsgemäß ist während der Aufbauphase eine stärkere technische Unterstützung des Testteams notwendig. In den etablierten Phasen reichen aber bloß 5-8% Testautomatisierer innerhalb des Testteams aus.

Dr. Krisztián Vuocsics
Senior Consultant
Sogeti Deutschland GmbH

