itmanagement
NOVEMBER 2018

MAGAZIN FÜR DIE ENTERPRISE IT

UNTERNEHMENSSCHUTZ AUS ZWEI PERSPEKTIVEN

KEIN ZUTRITT FÜR (CYBER-)KRIMINELLE

Milos Hrncar, General Manager DACH, Kaspersky Lab

SONDERDRUCK FÜR sogeti Part of Capgemini

APPLICATION MANAGEMENT SERVICES

Wer hält die Zügel in der Hand?

www.it-daily.net

SCHLÜSSELWORTBASIERTES

In diesem Artikel zeigen die Autoren, wie ein automatisiertes Umfeld aufgebaut werden kann, so dass manuell arbeitende Tester nach einer kurzen Einarbeitung automatisierte Testfälle schreiben und auch verwalten können.

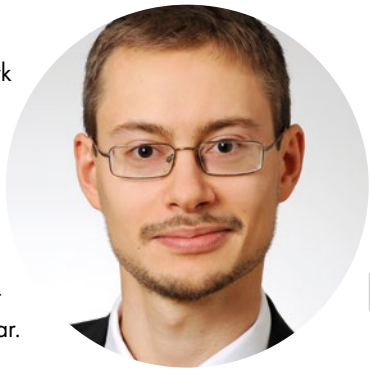
Der Bedarf an IT-Personal ist heutzutage immens und kann zur Zeit nicht mehr gedeckt werden. Gleichzeitig steigt die Anzahl der Qualifikationen, die von Testern erwartet wird[1], so dass der Fachkräftemangel noch schmerzhafter zu spüren ist. Es werden Multitalente gesucht, die sowohl einen fachlichen als auch technischen Durchblick haben müssen und dementsprechend rar und teuer sind.

Ein grundlegender Gedanke bei der Einführung von Schlüsselwörtern ist die Trennung der technischen und fachlichen Aufgaben in der Personalstruktur. So sind nur wenige, technisch sehr versierte Automatisierer erforderlich, die das Schlüsselwort-Framework pflegen, das von den fachlich bewanderten Mitarbeitern mit geringerem technischem Hintergrund genutzt werden kann.

Beim klassischen Capture-Replay-Weg von Testautomatisierung – Testfälle aufnehmen, anpassen, abspielen – geht es eher um Automatisierung als um Test. Eigentlich kann alles aufgenommen und dann abgespielt werden, nicht nur Testfälle (siehe zum Beispiel „Makro aufzeichnen"-Funktion in Office). Alle diese Aufzeichnungen müssen dann aber von Technikern betreut werden – ohne vertiefte Programmierkenntnisse hat ein Tester keine Chance, den automatisierten Testbetrieb aufrechtzuerhalten. Wer das nicht glaubt, soll den Windows-Taschenrechner starten, den Testfall „2+2=?" mit einem Testwerkzeug aufnehmen und sich dann überraschen lassen.

Arbeiten mit dem Framework

Ein großer Vorteil bei der Verwendung eines Schlüsselwort-Frameworks liegt in der besseren Skalierbarkeit. Testfälle, die mit Tools wie Selenium oder SilkTest aufgezeichnet werden, sind zwar zunächst schnell automatisiert. Jedoch rächen sich Änderungen im zu testenden Programm, da jeder bereits automatisierte Testfall wieder bearbeitet werden muss. Wenn ein Framework benutzt wird, ist zunächst ein gewisser Overhead notwendig, um die grundlegenden Funktionalitäten zu implementieren. Dafür sind Testfälle bei einer guten Frameworkumsetzung leicht um weitere Schritte erweiterbar.



Werfen wir nun einen Blick auf die Arbeitsweise mit dem Framework. Letztendlich läuft das Ganze in folgenden Schritten ab:

1. Die Testschritte werden von den Fachtestern in einem menschlich lesbaren Format mit minimalen syntaktischen Anforderungen manuell aufgeschrieben beziehungsweise bereits existierende Testfälle werden angepasst.

2. Dieser Text wird in Programmcode („Testdatei") konvertiert. Das dazu notwendige Tool wird vom Automatisierungs- und Beratungsteam zur Verfügung gestellt.

3. Die Testdatei, das heißt der Testfall, wird mittels des Testtools und der Framedateien durchgeführt und ausgewertet.

Die Trennung der Aufgaben lässt sich auch in der Personalstruktur widerspiegeln. Die syntaktischen Anforderungen der Schlüsselwort-Notation sind minimal. Jeder Fachtester, der einen Testfall in Freitext formulieren kann, ist nach einer sehr kurzen Schulung beziehungsweise Einarbeitung fähig, den Testfall in Schlüsselwörter umzuwandeln. Wenn wir nur einen Vorteil dieses Vorgehens nennen dürften, wäre es dieser Punkt: die Fachtester sind auch im automatisierten Umfeld bereits nach einer sehr kurzen Einarbeitung einsatzbereit.

Gute Testarchitektur als Basis

Die Einführung eines Schlüsselwort-Frameworks läuft wie ein kleines in sich geschlossenes Softwareprojekt ab. Nach dem Startschuss spezifizieren Techniker und Fachtester gemeinsam die Schnittstelle des Frameworks. Dazu gehören die Definition der wichtigsten Schlüsselwörter und Funktionen und die Bestimmung der minimalen syntaktischen Anforderungen. Beispielsweise kann bei einer Behörde die Funktion „Vergleiche PDF" für die Prüfung der erstellten Schriftstücke notwendig sein und auf das ansonsten ganz triviale „Bewege Maus" kann verzichtet werden. Beim Testen eines Online-Spiels sieht das aber genau umgekehrt aus.

> EIN GROSSER VORTEIL BEI DER VERWENDUNG EINES SCHLÜSSEL-WORT-FRAMEWORKS LIEGT IN DER BESSEREN SKALIERBARKEIT.
>
> Thomas Bauer, Consultant,
> Sogeti Deutschland GmbH | www.sogeti.de

Die verständliche und „fachtesterfreundliche" Struktur der gegebenenfalls deutschsprachigen Bezeichner (zum Beispiel „Ergebnis" und nicht „MainWindowResultJTextfield"...) und der Funktionen („Bewege Maus" statt „System.IO.Devices.Mouse[0].MoveTo") ist zwingend erforderlich für eine gute Testarchitektur.

Nach der Definition der wichtigsten Schlüsselwörter und Funktionen können die Fachtester schon die ersten Testfälle skizzieren. Parallel dazu können die Automatisierer mit der Implementierung beginnen. Selbst wenn erst zum Beispiel „Starte Anwendung" und „Beende Anwendung" umgesetzt sind, ist der erste Smoketest-Testfall bereits fertig! Auch der Testmanager kann loslegen und die ersten Testfallketten (Testsuiten) zusammenstellen, da eine automatisierte Durchführung von mehreren Testfällen nacheinander nun gewährleistet werden kann. Mit nur wenigen Funktionen wird schon ein

TESTEN

WIE TESTER EFFIZIENT DIGITALISIERT WERDEN KÖNNEN.

Testfall ohne Testautomatisierung	Automatisiertes Testen mit Schlüsselwörtern	
	Fachtester (Schritt 1.)	Automatisierer (Schritt 2.)
Testfall in Freitext (Drehbuch für manuellen Tester)	Testfall in SW-Notation	Konvertierung in Programmcode mit Testframework
[Fachtester]	[Fachtester]	[Automatisierer]
Starte die Anwendung	Starte Anwendung DEMO_APP	`FW.StartProgram („DEMO_APP");`
Suche auf „Max Mustermann"	Gib ein „Max Mustermann" in „Suchfeld" Drücke Schaltfläche „Suchen"	`FW.InputKeys („Suchfeld", „Max Mustermann");` `FW.ClickButton ("Suchen");`
Prüfe ob Geburtsdatum „01.01.2001" ist	Prüfe Wert: „Geburtsdatum" = „01.01.2001"	`FW.CheckValue („Geburtsdatum", „01.01.2001");`
Schließe die Anwendung	Schließe Anwendung DEMO_APP	`FW.ExitProgram („DEMO_APP");`

Die Abbildung zeigt die beispielhafte Darstellung der Arbeitsweise mit einem Framework.

überraschend großer Anteil der Testfälle abgedeckt – auch hier greift das Pareto-Prinzip. Im späteren Projektverlauf werden die Aufgaben für die Fachtester geringer. Da sie nur für die Erstellung und Verwaltung, aber nicht mehr für die Durchführung der Tests verantwortlich sind, können die meisten Mitarbeiter nach dem Go-Live – wenn kaum neue Testfälle zu erstellen sind und die Durchführung von den Maschinen übernommen wurde – in den nächsten Projekten eingesetzt werden.

Der zunächst höhere Bedarf an Automatisierern sinkt sogar noch früher – bereits nach der Umsetzung des Testframeworks. Der Betrieb eines etablierten Frameworks, also die Fehlerbehebungen und die Implementierung der neuen Bedürfnisse, ist auch mit wenigen Technikern möglich. Die meisten Testautomatisierungsexperten werden bloß in der Planungs- und in der Aufbauphase des Projektes benötigt, wenn die Schlüsselwörter und die Funktionen definiert beziehungsweise deren Interpreter implementiert werden müssen. Dieser erhöhte Bedarf an Testautomatisierern bei Projektbeginn kann am einfachsten durch die Inanspruchnahme von externen Dienstleistern abgedeckt werden - vor allem, wenn nicht ausreichend interne Automatisierungserfahrung vorhanden ist. Langfristig lässt sich das Framework mit wenigen eigenen Mitarbeitern und nur einem kleinen Anteil an Technikern betreiben.

Fazit

1. Der größte Vorteil der schlüsselwortbasierten Testautomatisierung liegt darin, dass bei den Fachtestern keine Programmierkenntnisse vorausgesetzt werden.
2. Für diesen Vorteil bezahlt man mit den im Vergleich zur Capture-Replay Automatisierung höheren Einführungsaufwänden.
3. Der hohe Bedarf an Technikern während der Einführungsphase lässt sich mit externen Mitarbeitern überbrücken.
4. Essentiell ist die sorgfältige Bestimmung der Schlüsselwörter. Diese macht sich bei der Testerstellung und besonders bei den späteren Anpassungen bezahlt.

Dr. Krisztián Vucsics, Thomas Bauer

EIN GRUNDLEGENDER GEDANKE (...) IST DIE TRENNUNG DER TECHNISCHEN UND FACHLICHEN AUFGABEN IN DER PERSONAL-STRUKTUR.

Dr. Krisztián Vucsics, Senior Consultant, Sogeti Deutschland GmbH | www.sogeti.de

[1] World Quality Report 2018-19, Seite 11, „The skills required for QA and testing have changed"

World Quality Report 2018-19

Wie wird Künstliche Intelligenz die Softwareentwicklung und Softwaretests beeinflussen?

www.worldqualityreport.com

Capgemini | MICRO FOCUS | sogeti Part of Capgemini