

## Geleitwort von Harry M. Sneed

Die nunmehr zweite Auflage von »Basiswissen Testautomatisierung« untermauert deutlich die Notwendigkeit des Einsatzes von automatisierten Testverfahren in der modernen Softwareentwicklung. Ohne leistungsfähige Testwerkzeuge ist der Test komplexer IT-Systeme und erst recht eingebetteter Echtzeitsysteme kaum vorstellbar. Solche Systeme mit ihren vielfachen Nutzungsvariationen stellen die Menschen vor eine manuell nicht zu bewältigende Aufgabe, vor allem nicht in der Zeit, die ihnen dafür zur Verfügung steht. Zu bedenken ist, dass diese Systeme Tausende Durchlaufpfade und Abertausende Datenzustände haben. Um nur einen repräsentativen Ausschnitt davon abzudecken, werden mehrere Zehntausend Testfälle benötigt. Testfälle, die nicht nur einmal ausgeführt werden, sondern immer wieder, jedes Mal, wenn sich das System ändert. In einer iterativen Entwicklung wird der Test ebenfalls viele Male wiederholt, bis das Produkt überhaupt freigegeben wird. Dies ohne Testautomatisierung zu bewältigen ist undenkbar. Insofern ist die Automatisierung der Testaufgaben eine unabdingbare Voraussetzung für die Entwicklung und Erhaltung größerer Softwaresysteme.

Demzufolge muss sich jeder Tester von größeren Softwaresystemen mit Testwerkzeugen auseinandersetzen. Er muss wissen, welche es gibt und welche zu seinem spezifischen Test passen. Es genügt nicht, sie nur bedienen zu können. Er muss auch in der Lage sein, sie zu evaluieren und zu beurteilen. Das setzt voraus, dass er weiß, was er von ihnen erwarten kann, was automatisierbar ist und was nicht. Die Testlandschaft ist sehr vielfältig geworden. Sie hat mindestens drei Dimensionen:

- Eine horizontale Dimension der unterschiedlichen Systemtypen wie eingebettete Systeme, Prozesssteuerungssysteme, Standalone-Systeme, vernetzte Systeme, Webapplikationen, Telekommunikationssysteme usw. Jede Systemart wird anders getestet und das bedeutet, dass verschiedene Testwerkzeuge benötigt werden.
- Eine vertikale Dimension der abzuarbeitenden Testaktivitäten, z.B. Testplanung, Testspezifikation, Testdesign, Testdatenerstellung,

Testausführung und Testauswertung. Jede Aktivität stellt eigene Aufgaben, die gelöst werden müssen. Daraus folgt, dass jede Testphase andere Anforderungen an die Testwerkzeuge stellt. Hierzu kommen auch die unterschiedlichen Teststufen wie Komponententest, Integrationstest, Systemtest und Abnahmetest.

- Eine dritte Dimension ist die technische Testumgebung, bestehend aus Hardware und Software. Für eine Mainframe-Testumgebung werden andere Werkzeuge benötigt als für eine Client-Server-Umgebung oder für eine vernetzte Webumgebung. Im Gegensatz zu den Entwicklungswerkzeugen, die von der Zielumgebung unabhängig sein können, müssen die Testwerkzeuge zu der jeweiligen Testumgebung passen.

Dies erklärt die Vielfalt des Toolangebots: Das breite Angebot ist nur eine Antwort auf die breit gefächerte Nachfrage. Tester brauchen nicht ein Tool für alles, sondern ein Tool zu jedem Zweck in jeder Situation. Da es kein allumfassendes Testwerkzeug geben kann, das wäre die sogenannte eierlegende Wollmilchsau, müssen sich Tester mit einem Testwerkzeugkasten ausrüsten, ein Kasten mit dem richtigen Tool für jeden Zweck. Es ist von Vorteil, wenn die verschiedenen Tools miteinander kompatibel sind, besser ist es, wenn sie miteinander verknüpfbar sind, und noch besser, wenn sie aufeinander aufbauen. Der Tester wird immer mit der Frage konfrontiert, ob er das »best of breed«-Tool für den jeweiligen Zweck nimmt oder ob er das Tool verwendet, das mit den anderen Tools am besten integriert oder integrierbar ist.

Manche Testwerkzeuge werden als Standalone-Lösung zu einer bestimmten Testaufgabe angeboten, andere als Teil einer Werkzeugfamilie. Es hängt von der Projektsituation ab, welche Alternative der Testverantwortliche wählt. Für einmalige, vorübergehende Projekte genügen Standalone-Werkzeuge. Für dauerhafte, wiederholende Projekte ist es preiswerter und effektiver, wenn die Werkzeuge integriert sind. Dann ist nämlich die Kontinuität des Tests am besten gewährleistet.

Die Fähigkeit zu entscheiden, welche Testwerkzeuge wo und wann einzusetzen sind, setzt voraus, dass sich der Testverantwortliche mit der Testautomatisierung auskennt. Die Fähigkeit, die gewählten Testwerkzeuge richtig einzusetzen, setzt wiederum voraus, dass der einzelne Tester mit den Möglichkeiten und Grenzen der Testautomatisierung vertraut ist. Beide Voraussetzungen werden durch dieses Buch geschaffen. Das Buch zielt sowohl auf den Testverantwortlichen als auch auf den durchführenden Tester ab.

Dem Autorenteam ist es gelungen, beiden Zielgruppen – Testmanagern und Testern – gerecht zu werden. Im ersten Kapitel erfahren Sie,

warum Sie sich mit Testautomatisierung befassen sollten. Es gibt eigentlich keine Alternative, um das Mengenproblem des Testens zu bewältigen.

Im zweiten Kapitel lernen Sie, wie Testautomatisierung in den Testprozess einzubinden ist. Ein mehr oder weniger willkürlicher, unkontrollierter Einsatz von Werkzeugen, ohne diesen einen geordneten Prozessrahmen zu geben, ist ebenso wenig zweckmäßig wie die Definition theoretischer Abläufe und Methoden, die ohne eine verfügbare Werkzeugunterstützung nicht sinnvoll umsetzbar sind. Der Diskurs zwischen den Anforderungen und deren Machbarkeiten ist in jedem Unternehmen, oft auch in jedem Projekt immer wieder aufs Neue zu führen, denn beide Dimensionen unterliegen laufenden Anpassungen und Innovationen.

Das dritte Kapitel erläutert die unterschiedlichen methodischen Ansätze zur Testfall- und Testdatenspezifikation, die wie im manuellen Test auch in der Testautomatisierung eine wichtige Voraussetzung darstellen. Das Kapitel beschreibt weiterhin verschiedene Arten der Durchführung von automatisierten Tests und welche möglichen Fehlerquellen lauern.

Das vierte Kapitel zeigt, wie ein automatisiertes Testframework aufzubauen ist. Dies ist für Tester wichtig, die verschiedene Testwerkzeuge einsetzen und miteinander verbinden oder eine eigene Testarchitektur aufbauen wollen. Sie lernen hier anhand praktischer Beispiele, wie man vorgeht, um so einen Werkzeugrahmen zu bauen, nach dem Motto »selbst ist der Mann«. Dies ist in vielen Fällen notwendig, wenn es darum geht, die Werkzeuge anzupassen. Hierfür gibt es bewährte Muster bzw. Patterns, die von den Autoren geschildert werden.

Dazu gehört auch das grundlegende Verständnis der verschiedenen Testautomatisierungsansätze, z.B. der skriptgetriebene Ansatz, der datengetriebene Ansatz, der schlüsselwortgetriebene Ansatz und der modellgetriebene Ansatz. Hier lernt der Leser, dass hinter jedem Testwerkzeug eine Testideologie steckt, eine Grundannahme, wie man testen sollte. Wer die zugrunde liegende Ideologie eines Werkzeugs nicht versteht oder nicht akzeptiert, wird sich mit dem Einsatz des Tools schwertun. Des Weiteren geht es um die Bedienung der Werkzeuge durch den Tester. Viele Werkzeuge unterscheiden sich in ihren technischen Konzepten im Handling der vielfältigen Applikationsszenarien. Es ist ein Unterschied, ob eine Webapplikation, Webservices oder Client-Server-Systeme zu testen sind. Es ist etwas anderes, Testszenarien auf GUI-Ebene zu automatisieren oder auf Protokollebene einer Systemschnittstelle. Den unterschiedlichen Konzepten ist jedoch gemeinsam, dass diese den Prinzipien der Einfachheit, Wiederverwendbarkeit und Wartbarkeit entsprechen müssen.

Im fünften Kapitel gehen die Autoren auf die diversen Einsatzgebiete für Testautomatisierung ein. Diese reichen von Desktop-Stand-

alone-Anwendungen bis zu breit angelegten, serviceorientierten Architekturen. Auf jeder Plattform und auf jeder Teststufe braucht der Tester Werkzeugunterstützung. Hier werden die Formen der Unterstützung zusammengefasst. Wichtig ist, dass die Werkzeuge zur Umgebung passen.

Im sechsten Kapitel wird eine Auswahl der auf dem Markt angebotenen Testwerkzeuge kurz beschrieben. Dabei sind sowohl Open-Source- wie auch kommerzielle Produkte im Fokus. Angesichts der Menge der weltweit angebotenen Tools ist es kaum möglich, sie alle zu erwähnen, geschweige denn zu beschreiben. Das würde den Rahmen dieses Werkes sprengen. Hier werden in erster Linie Produkte vorgestellt, mit denen die Autoren in ihrer praktischen Tätigkeit als Tester Erfahrung gesammelt haben. Das ist insofern legitim, als die Autoren nicht einfach Broschüren zitieren, sondern ihre persönlichen Erfahrungen weitergeben, und dadurch dem Leser zusätzliche Informationen und Einschätzungen zur Verfügung stellen.

Das siebente Kapitel beschreibt die organisatorische Eingliederung der Testwerkzeuge bzw. welche Auswirkung die Testautomatisierung auf die Testorganisation hat. Die Art der Testautomatisierung bestimmt, welche Rollen es im Testteam geben wird und wie diese Rollen miteinander zu vereinbaren sind. Anderseits können die im Unternehmen vorhandenen Skills die Auswahl der einzusetzenden Werkzeuge beeinflussen. So gesehen hat die Testautomatisierung einen maßgebenden Einfluss auf die Organisation eines Testprojekts und umgekehrt. Das müssen die Testverantwortlichen zur Kenntnis nehmen.

Das achte und letzte Kapitel des Buches fasst den Stand der Technologie zusammen und vermittelt einen Einblick in die künftige Weiterentwicklung. Betont wird die zunehmende Bedeutung der Testautomatisierung für alle Testbereiche. Eines steht fest, die Testautomatisierung befindet sich noch am Anfang. Viele Testaufgaben sind bereits automatisiert, aber es warten noch viele mehr. Eine Horde menschlicher Tester auf die Software loszulassen, ist teuer und ineffektiv. Gefordert sind nicht mehr »Tester«, sondern gut ausgebildete und ausgerüstete Testspezialisten, die mit ihren Werkzeugen weit aus mehr erreichen können als die Testhorden ohne Ausrüstung. Allerdings müssen wir noch viel mehr in die Toolentwicklung investieren, um den Automatisierungsstand zu erreichen, den wir brauchen. Bis dahin bleibt der Test in den Augen vieler nach wie vor ein großer Ressourcenfresser, der uferlose Kosten verursacht und unberechenbaren Nutzen bringt. Mit dem Einsatz von Testautomatisierung kann dieses Bild gewandelt werden.

*Harry M. Sneed, Wien 2011/2015*