testing experience de

Das Magazin für Tester, Entwickler und Manager

Nr. 6 | Juli 2014

kostenlose Ausgabe

DIE ZUKUNFT DES TESTENS

Welche Skills werden gebraucht?

Díaz Hilterscheid

Wie die Welt von morgen getestet wird

Der Autor

Derk Masselink

Derk Masselink wurde in Delft in den Niederlanden geboren. Er studierte Technische Physik an der Delft-Universität für Technologie und machte seinen Master im Computational Physics Department. Er hat als Softwareentwickler mit C++ gearbeitet, dann als Produktionsplaner im Lebensmittelbereich und danach für 15 Jahre in den Bereichen Softwaretest und Qualitätssicherung. In den letzten Jahren waren seine Hauptaktivitäten Testmanagement und Testautomatisierung sowohl für den Versicherungs- und Finanzsektor als auch für den Navigationssektor.

Letztes Jahr habe ich an einem automatisierten Test in JMeter für das neue Middleware-System einer Versicherungsgesellschaft gearbeitet. Wir hatten einige sehr gute Tests in JMeter, aber es war sehr schwer, die Testkomponenten wiederzuverwenden und aufrechtzuerhalten. Gleichzeitig experimentierte ich mit Fitnesse. Fitnesse ist ein wikiartiges Testautomatisierungstool, mit dem man das generelle Verhalten der Tests in Java-Klassen programmiert und dies dann benutzt, um auf einfache Weise Testfälle in einem benutzerfreundlichen Format in einem Wiki zu erstellen.

Das Problem war dann, dass ich nicht genug Java-Skills hatte, um einen verwertbaren Test zu erstellen, also blieb ich dabei, mit Fitnesse und JMeter zu experimentieren.

In der Zwischenzeit begann das Team, sich über Schwierigkeiten mit der Wartbarkeit von JMeter-Tests und den Mangel an Testkapazitäten zu beschweren. Also diskutierten wir die Möglichkeit, ein einfacher aufrechtzuerhaltendes Test-Tool wie Fitnesse zu benutzen.

Nach einer Weile half uns ein erfahrener Java-Architekt und -Programmierer, der Teil unseres Teams war, dabei, ein Proof of Concept für einen Fitnesse-Test zu erstellen. In wenigen Tagen hatten wir einen automatisierten Test für unsere Transaktionen. Er erstellte einige grundlegende Java-Klassen, die hilfreich waren, und erklärte, wie wir Fitnesse nutzen können. Nach ein paar Wochen konnte das ganze Team Fitnesse einwandfrei nutzen, und die Wartbarkeit erhöhte sich stark. Der Vorteil hierbei ist, dass jedes Teammitglied eine Rolle dabei spielen kann, einen Test zu erstellen, und die Tests einfach vom Auftraggeber eingesehen werden können. Tatsächlich macht es ihr Wiki-Charakter sogar möglich, die Tests und die Anforderungen auf denselben Wiki-Seiten aufzuschreiben.

Was können wir hiervon also lernen? Ich denke, die größte Lektion ist, dass es uns großen Nutzen bringen kann, mit Programmierern und anderen Menschen mit technischen oder geschäftlichen Fähigkeiten zu kooperieren.

> „WIR HABEN UNS GLEICHZEITIG IN RICHTUNG TECHNISCHER SKILLS UND BENUTZERFREUNDLICHKEIT BEWEGT“

Natürlich muss man, um sich an eine neue Art von Arbeit anzupassen, auch neue Fähigkeiten erlernen und ihnen gegenüber offen sein.

Eine weitere wichtige Lektion ist, dass wir mit der Einführung von Fitnesse zwei Veränderungen mitgemacht haben – eine geht in eine eher technische Richtung durch das Programmieren in Java und das Ausführen von Tests, die andere führt zu einer simpleren und benutzerfreundlicheren Art von Arbeit, indem der Überblick über Testfälle auf benutzerfreundliche Weise in einem Wiki ermöglicht wurde. Dies machte es extrem leicht, die Testfälle zu überprüfen und zu warten.

Das Paradoxe daran ist, dass zwei gegensätzliche Veränderungen stattgefunden haben, obwohl nur ein neues Tool eingeführt wurde. Der Prozess ist gleichzeitig technischer und weniger technisch geworden. Und die Test-Rolle ist ebenfalls gleichzeitig technisch und weniger technisch geworden! Außer natürlich man entscheidet sich, die Test-Rolle zu zerteilen, aber das ist keine gute Idee.

Eine auffallende Eigenschaft sowohl der technischen als auch der weniger technischen Seite ist die Wartbarkeit.

Natürlich ist Tooling ein Trend, den es schon seit einigen Jahren gibt, und es ist zu erwarten, dass er noch wachsen wird. Ich denke, dass wir als Tester viel von Entwicklern lernen können, die mit Tools arbeiten, um die Testabdeckung in Modultests zu prüfen, sowie auch mit Tools, mit denen sie ihre Database organisieren und nächtliche Tests ausführen.

In vielen Situationen sind Tester in der Lage, von Tools zu profitieren, die bereits von Entwicklern eingeführt wurden und auch von Testern verwendet werden können.

Extrapolation von Trends und ihr Effekt auf das Testen

Im Folgenden wird ein Blick auf mögliche Trends im IT-Bereich geworfen und welchen Effekt diese auf das Testen haben könnten. Ich kann nicht in die Zukunft sehen, aber ich hoffe, einige Behauptungen sind wiedererkennbar.

Trend	Warum	Effekt auf IT	Effekt auf Testen
Transaktionskosten und Kosten von Softwareverteilung sinken.	Apps werden heutzutage zu Preisen von unter einem Euro verkauft. Dies ist möglich, weil die Transaktionskosten des Kaufs sich aufgelöst haben und weil Software in großen Mengen verkauft wird, was die niedrigen Preise profitabel macht.	Viele Systeme werden sich nur auf eine bestimmte Aufgabe spezialisieren und Teil einer großen Kette von Systemen sein, die die Gesamtfunktionalität gewährleisten. In manchen Marktsegmenten wird die Spezialisierung unter Umständen rasant vonstattengehen.	Die Ketten und Netzwerke werden sich stark vergrößern. Funktionalitäten werden redundant, und manche Tests werden auf vielen Pfaden des Netzwerks durchgeführt werden.
Die Anzahl von Softwarelösungen in unseren Aktivitäten steigern	Besseres Tooling, um Software zu erstellen, durch das Senken von Transaktionskosten	Die Zahl der Anbieter wird wahrscheinlich steigen. Es ist nicht möglich, den Überblick über all diese Systeme zu behalten – deshalb wird die Kommunikation zwischen Systemen standardisiert werden müssen.	Testen über eine Kette bedeutet nicht, eine feste Reihe von Systemen zu testen, sondern eine flexible und sich verändernde. Gleichzeitig wird die Zahl von notwendigen Tests ansteigen. Dies kann durch das Nutzen von Tools erreicht werden, die Tests aus Prinzip-Aussagen generieren. Viele Informationskomponenten und Softwareapplikationen werden außerhalb der direkten Kontrolle der Tester sein. Qualitätssicherung der gesamten Kette wird durch gute Beziehungen und geteilte Anwendungen zwischen Anbieter und Kunden geschaffen werden.
Hardwarekosten sinken und die Nutzung von Hardware in Konsumprodukten steigt.	Dieser Trend besteht bereits seit 50 Jahren.	Gegenstände wie Schuhe können ihren eigenen Sensor sowie Computer- und Netzwerkverbindung haben. Eine enorme Menge an Information wird verfügbar sein (Big Data).	Testumgebungen werden Simulatoren für physische Objekte beinhalten müssen.
Der funktionalitätsbezogene Lebenszyklus von Software wird kürzer.	Die Gründe sind unklar.	Das Risiko, dass zwischen Systemen Kompatibilitätsprobleme entstehen, erhöht sich.	Allgemeine up to date integrierte Testumgebungen werden wichtiger.
Die Menge an Information, die für Softwarelösungen zur Verfügung steht, wird steigen.	Laufender Trend	Das Gesamtbild von Daten wird wichtig, und einzelne Datenelemente haben ohne Kontext keine Bedeutung. Um mit diesen Daten umzugehen, benötigen wir fortgeschrittenes Data-Mining und Machine-Learning-Techniken.	Manche Information ist nicht mehr wahr oder falsch, sondern wird stattdessen einen bestimmten Umfang an Nützlichkeit und Effektivität haben. Die Bedeutung und der Wert von Daten werden wichtig.
Generierte Software und Machine Learning	Obwohl vergangene Erwartungen nicht eingetroffen sind, wird dieser Trend wahrscheinlich wachsen, besonders in einigen Sub-domains.	Prinzipien festlegen und formen wird wichtiger, als tatsächlich Code zu schreiben.	Wenn die Funktionalität nicht geschrieben, sondern gelernt wurde, wird Testen ein wichtiges Tool, um die Funktionalität zu reparieren, Einschränkungen zu definieren und Qualität zu messen. Testen ist sogar ein Teil des Machine-Learning-Ansatzes.
Der Zeitpunkt des Testdesigns im Entwicklungsprozess verändert sich.	In der Vergangenheit schien Test-Driven-Development wie ein Wunder. Bei dem Arbeiten mit Fitnesse konnten wir tatsächlich Tests parallel zur Software entwickeln, und in manchen Fällen waren die Tests zuerst fertig.		In Zukunft werden Testfälle die Voraussetzung sein und realisiert werden, bevor gecodet wird. Die Verantwortung der Tester wird sich in Richtung geschäftlicher Wert statt Korrektheit verlagern.

Tabelle 1: Trends, die die Arbeit des Testers beeinflussen werden

„TESTEN WIRD IN EINEM GROSSEN NETZWERK VON SYSTEMEN STATTFINDEN."

Eine Analogie

Wenn man Tabelle 1 betrachtet, kann man für sich selbst urteilen, ob man diesen Trends und den Effekten auf Tester zustimmt. Wie sieht das allgemeine Bild aus, das sich aus diesen Daten ergibt? Kann man eine Metapher für zu erwartende Änderungen finden? Vielleicht ist es mit einem Bienenstock vergleichbar. Bienen tun natürlich nur, was sie tun müssen. Aber nehmen wir mal an, wir müssten das Verhalten von Bienen programmieren. Wie würden wir das testen? Und nehmen wir mal an, genau wie die Bienen müssten nun auch Blumen und andere Teile der Natur programmiert werden und jeder könnte diese Blumen, Bäume usw. programmieren.

Bienensimulatoren in ihren Testumgebungen. In der Praxis werden die meisten Beteiligten ihre Testumgebungen veröffentlichen, und eine weltweite Test-Acceptance-Umgebung wird entstehen. Natürlich ist die Qualität der programmierten Blumen für uns genauso wichtig wie die Qualität unserer Bienen. Wir werden uns also mit unseren direkten Zulieferern und Kunden auf eine Art einigen, die Qualität zu prüfen, und kontinuierlich die gemessene Qualität unserer Bienen veröffentlichen. Bienen können nichts richtig oder falsch machen, sondern nur mehr oder weniger effektiv.

„DURCH DIE KOMPLEXITÄT UND DIE GESCHWINDIGKEIT VON VERÄNDERUNGEN WIRD DER TESTER ENTSCHEIDUNGEN TREFFEN MÜSSEN, OHNE DEN BETRIEB ZU KONSULTIEREN."

Wir würden wahrscheinlich einige sehr spezifische Verhaltensweisen für eine Biene definieren und dafür dann einen Test machen. Allerdings könnten wir uns nicht sicher sein, ob dies das bestmögliche Verhalten ist. Also müssten wir Tests für das Gruppenverhalten von Bienen definieren, wie zum Beispiel die Weitergabe von Informationen über Nektarquellen. Und wir würden Ziele und Teilziele definieren, um beurteilen zu können, ob das getestete und definierte Verhalten wirklich zielführend ist. In der Zwischenzeit haben die Blumenentwickler das Verhalten, die Anzahl und die Wege zu den Blumen geändert. Also müssen unsere Testfälle angepasst werden. Nach einigen dieser Änderungen bauen wir ein Tool, um Tests aus diesen Teilzielen zu generieren. Um sicherzugehen, dass alles glatt läuft, brauchen wir eine Testumgebung, die Blumen und Blumensimulatoren enthält. Wiederum brauchen die Blumenentwickler

Eine neue Rolle

In naher Zukunft werden technische Skills wichtiger werden. Tester werden programmieren und werden wissen müssen, wie sie auf System Interfaces statt auf User Interfaces testen können.

Auf lange Sicht wird die Rolle das Verständnis für ein sich veränderndes Modell, abstrakten Dingen Bedeutung zuzuweisen und das Einschätzen des Wertes einer Lösung beinhalten. Alles verändert sich ständig, also wird der wichtigste Skill wahrscheinlich die Fähigkeit zu lernen und anderen dabei helfen zu lernen sein. Außerdem werden das Verstehen und Priorisieren basierend auf Einschätzungen von brauchbaren Geschäftswerten wichtig werden. Die Zeit, komplexe, abstrakte Lösungen mit dem Geschäftsbesitzer zu diskutieren, wird es nicht geben. Der Tester selbst wird entscheiden müssen, wie er weitermacht, in Kombination mit dem Tooling, welches entwickelt wurde, um diese Entscheidungen zu unterstützen.

Soziale Aspekte

Diese Veränderungen in der Umgebung, dem Entwicklungsprozess, dem Tooling und so weiter haben einen Effekt auf die sozialen Aspekte der Arbeit. Wird dies andere Skills erfordern?

Nun, wenn wir die sich kontinuierlich ändernde große Gruppe von Systemen betrachten, die von einer sich verändernden Gruppe von Anbietern gemacht wurde, dann können wir uns vorstellen, dass es hilfreich ist, eine Art Reputation aufzubauen und diese zu nutzen, um kommende Änderungen vorauszusehen. Die soziale Intelligenz, die hierfür notwendig ist, hat mit einer Mischung aus (Gruppen von) Menschen und Systemen zu tun. Manche Menschen, die in Firmen mit einem großen IT-Bereich arbeiten, mögen heute schon das Bedürfnis haben, durch einen Wald von Projekten, Verbesserungen und Bugfixes zu navigieren, wo es wichtig ist, ein gutes Gespür für Dringlichkeit und Priorität zu haben. Andere verlassen sich auf dich, und du verlässt dich auf andere. Zwischen großen und kleineren Problemen zu unterscheiden, hält die gesamte Anlage gesund. Aber um zu wissen, was heute gemacht werden muss, muss man wissen, was schiefgehen wird und was nicht, was sich ändern wird und was gleich bleiben wird.

Gleichzeitig ist man Teil eines großen, komplexen Systems. Man hat eine sehr spezialisierte Aufgabe. Die eigene Arbeit kann sich mit der Zeit sehr von der Arbeit anderer unterscheiden in Bezug auf Konzepte und Aktion. Andererseits können Automatisierung, Unterstützung durch Tooling und mögliche Standardisierungen zu einer Deckungsgleichheit in der Art von Arbeit führen. Dies ist schwer vorauszusagen. Die Gefahr, dass überspezialisiert wird und der Kontakt zu anderen Themen verloren geht, kann vermieden werden, indem aktiv Rollen oder Bereiche getauscht werden, sowie auch durch Tätigkeiten, die nur indirekt mit der täglichen Arbeit zusammenhängen. Da man das ganze Leben lang lernt, werden die Kosten eines Berufswechsels nicht so hoch sein, wie sie zuerst scheinen.

„HEUTE IST, WO SICH ZUKUNFT UND VERGANGENHEIT TREFFEN."

Und was ist mit dem Team, in dem man arbeitet? Da sich alles ständig ändert, wird sich auch das Team ändern. Und im Zuge dessen ist zu erwarten, dass man in Zukunft Teil von mehr als nur einem Team ist. Man wird also viele soziale Kontakte und Möglichkeiten haben, aber man muss auch ständig mit neuen Leuten umgehen können müssen, außer man nimmt eine abstrakte Position ein und fokussiert sich rein auf Systeme.

Fazit

Dieser Artikel mag futuristisch scheinen. Aber manche der erwähnten Trends werden jetzt schon von einigen praktiziert. In den kommenden Jahren erwarte ich, dass automatisiertes Testen immer wichtiger wird, damit die gewünschte Funktionalität in einem sich ständig verändernden System gegeben ist. Das bedeutet, dass Tester lernen werden müssen, wie man automatisiert oder wie man eng mit Entwicklern, die das für sie tun können, zusammenarbeitet.

Wenn man den automatisierten Test hat, kann man ihn in einem Wiki veröffentlichen (wie mit Fitnesse) und anderen Leuten zum Überprüfen bereitstellen. Wenn das getan ist, warum dann nicht Testfälle als Voraussetzung nehmen?

Noch weiter in der Zukunft, sagen wir in zwanzig Jahren, erwarte ich, dass die getesteten Funktionalitäten so komplex und über Netzwerke und Systeme verteilt sind, dass Bedeutung und Wert genauso wichtig werden wie Korrektheit. Genauso wie man mehrere Wege nutzen kann, um einen weit entfernten Urlaubsort zu erreichen, so wird es auch mehrere Wege geben, um eine Transaktion zu erzielen. Wie ein Navigationssystem wird der Tester wahrscheinlich einer der Ersten sein, der durch das Systemnetzwerk navigieren muss.

Letztendlich ist zu erwarten, dass Testfälle generiert werden, die auf ausgewählten Zielen basieren, die man als Tester absichert. ■

Softwaretester mit ausgeprägter rechter Gehirnhälfte

Der Autor
Frank Titulaer

Frank Titulaer ist ein höchst erfahrener Testmanager bei Salves in den Niederlanden. Er hat über 12 Jahre Erfahrung im Bereich des Testens sowie eine ISEB-Practitioner-Zertifizierung. Franks Interessen sowohl innerhalb als auch außerhalb der Tester-Szene sind breit gefächert. Seine Beiträge können auf www.salves.nl gelesen werden. @frank_titulaer

In den letzten Jahrzehnten haben Softwaretester hart dafür gearbeitet, für das Softwaretesten Anerkennung zu erlangen. Wir haben Vorgehensweisen standardisiert und Bücher über diese Ansätze veröffentlicht. Wir haben Schulungen und Zertifizierungsprogramme kreiert und das alles international organisiert. Daraus resultierend ist der Softwaretester-Beruf herangereift. Er ist standardisiert, internationalisiert und toolunterstützt. Wir sind so weit gekommen, dass Tests automatisch und über den halben Globus hinweg durchgeführt werden können.

Analytische Fähigkeiten, die der linken Gehirnhälfte zugeschrieben werden, haben uns und den Tester-Beruf dorthin gebracht, wo wir heute sind. Nun mag man einwenden, dass Softwaretesten zur Handelsware geworden ist und wir deshalb neue Wege finden müssen, um einen Mehrwert zu schaffen. Wir müssen uns auf diejenigen Bereiche konzentrieren, die nicht automatisiert oder outgesourct werden können, und uns auf sie spezialisieren. Dies zwingt uns, unseren Beruf neu zu überdenken. Wir haben den Punkt erreicht, an dem unsere exzellenten analytischen Fähigkeiten nicht länger ausreichen. Neue, zusätzliche Fähigkeiten sind vonnöten – Fähigkeiten, die