



DIE ZUKUNFT DES TESTENS

Welche Skills
werden gebraucht?

Fähigkeiten eines Testers –
Abstraktion mit Leidenschaft
Johannes Widmann Seite 20

Was haben Koala und Ninja mit
dem Softwaretest zu tun?
Kay Grebenstein Seite 29

Die Zukunft des Testens – Qualifizierte
(Crowd-) Tester braucht das Land
Veronika Wasza Seite 32



Skillsets moderner Softwaretester in aktuellen Qualitätssicherungsprozessen

Software muss heutzutage höchsten Anforderungen genügen, sei es aufgrund der immer weiter voranschreitenden Durchdringung unseres Lebens durch Computer, der stetig wachsenden Komplexität der zugrunde liegenden IT-Umgebungen, steigender Qualitätsansprüche oder der immer größer werdenden Abhängigkeit von Wirtschaftsbetrieben in ihren Produktionsprozessen. Fehlerhafte Software kann so nicht nur wirtschaftliche, sondern auch gesundheitsgefährdende Risiken beinhalten. In dem gleichen Maße, wie die zu Anfang aufgezählten Faktoren die Anforderungen an Software erhöhen, steigt auch der Bedarf an professioneller Qualitätssicherung. Diese soll dabei helfen, Probleme wie überschrittene Kostenbudgets oder nicht eingehaltene Projektlaufzeiten zu vermeiden und die korrekte Funktionsweise der Software sicherzustellen.

Als Mitglieder und Tester in unterschiedlichen Projekten, in denen sowohl agile Entwicklungsmethoden als auch „traditionelle“ (Wasserfall, V-Modell) eingesetzt werden, ist es für uns immer wieder spannend zu sehen, wie die Art und Weise der Qualitätssicherung in den unterschiedlichen Prozessen gelebt wird und wo Unterschiede und Gemeinsamkeiten liegen. Dabei haben wir uns gefragt, wie es möglich ist, trotz aller Unterschiede in den Prozessabläufen die Zusammenarbeit unter den einzelnen unterschiedlichen Teammitgliedern zu optimieren und die Projekte zum Erfolg zu führen. Welche Testing-Skills jenseits von Tool- und Programmiersprachenkenntnissen ermöglichen unabhängig vom gelebten Softwareentwicklungsprozess eine

produktive Zusammenarbeit mit allen beteiligten Personen?

Zur Beantwortung dieser Frage werden wir im Laufe dieses Artikels die essenziellen Skillsets für Softwaretester in zwei Bereiche einteilen, die wir anhand von Mnemonics erläutern werden: Kommunikationskompetenz (SCCIL) und persönliche Kompetenzen (THE MOCCA PUB). Hierzu möchten wir klarstellen, dass die zu erarbeitenden Kompetenzen nicht allein auf das (Nach-)Testen beschränkt sind, sondern ebenso in Requirements Engineering und Diskussionen von Nutzen sind. In den Bereich der Kommunikationskompetenz fallen hierbei:

S
C
C
I
L

Stubborn
(Hartnäckigkeit)

Conflict Solver
(Konfliktlösungsfähigkeit)

Constructive Criticism
(Kritikfähigkeit)

Interpreter between different Views
(Fähigkeit verschiedener Sichten)

Language
(Fähigkeit der gemeinsamen Sprache)

Die Autoren



Ajantha Yogamoorthy

Ajantha Yogamoorthy hat einen Abschluss als Diplom-Wirtschaftsinformatikerin von der Technischen Universität Braunschweig. Bei

BREDEX GmbH ist sie angestellt als Test Consultant und für den Marketing-Bereich. Zu ihren Aufgaben gehören das Schreiben von automatisierten Softwaretests, die Pflege der Homepage und die Öffentlichkeitsarbeit.



Oliver Goetz

Oliver Goetz hat einen Abschluss als Magister der Soziologie von der Technischen Universität Braunschweig und ist seit April 2011 als Test

Consultant bei der BREDEX GmbH angestellt. Zu seinen Aufgaben gehören neben der Qualitätssicherung im hauseigenen Jubula/GUIDancer-Projekt die Testautomatisierung in einem Kundenprojekt, das Halten von GUIDancer-Schulungen sowie Vortragstätigkeiten auf diversen Konferenzen.

Die notwendigen persönlichen Kompetenzen sehen wir in den Fähigkeiten:

T H E M E S O C C A P U B

- Time management**
(Zeitmanagement)
- Healthy balance**
(Balance realistischer, optimistischer und negativer Sicht)
- Exactness**
(Genauigkeit)
- Methods**
(Methodenkenntnisse)
- Organisation**
(Organisationstalent)
- Curiosity**
(Neugier)
- Courage**
(Mut)
- Analytical**
(Analytisches Denken)
- Prioritisation**
(Fähigkeit zum Setzen von Prioritäten)
- Understand software**
(Verständnis der Software und des Entwicklungsprozesses)
- stay on the Ball** (Lernfähigkeit)

Kommunikationskompetenzen (SCCIL)

Hartnäckigkeit (Stubborn)

Hiermit sind konsequentes Auftreten und Handeln gemeint. Gerade in heißen Phasen des Entwicklungsprozesses, wenn die Anspannung innerhalb des Teams wächst, ist es wichtig, dass Tester auf die Einhaltung der angestrebten Qualitätssicherungsmaßnahmen drängen und auf Verletzungen hinweisen, wenn nötig immer wieder. Hier ist es die Aufgabe der Tester, dem Team in Erinnerung zu rufen, dass der Entwicklungsprozess erst mit der erfolgreichen Abnahme durch die Qualitätssicherung abgeschlossen ist, auch wenn dies bedeutet, einen Bug zum fünften Mal wieder zu öffnen.

Konfliktlösungsfähigkeit (Conflict Solver)

Konflikte treten dort auf, wo Menschen miteinander kommunizieren und interagieren. Dabei müssen Konflikte nicht per se schlecht sein, sondern beinhalten in den meisten Fällen eine konstruktive Komponente. Konfliktlösungsfähigkeit bedeutet jedoch nicht nur das Geschick, Konflikte zu lösen, sondern auch, diese zuallererst zu erkennen. Dabei spielt der Tester unserer Meinung nach eine gewichtige Rolle, wenn es darum geht, zwischen den einzelnen Teammitgliedern zu vermitteln. Probleme im Softwareentwicklungsprozess, die die Qualität des gesamten Projektes in Frage stellen könnten, sollen so frühzeitig erkannt und angesprochen und dadurch soll ein Beitrag zu deren Lösung geleistet werden.

Kritikfähigkeit (Constructive Criticism)

Keiner hört gerne, dass etwas Selbstkreiertes nicht gut genug ist. Unter Kritikfähigkeit verstehen wir das Talent, Kritik auf eine Art und Weise zu äußern, die den anderen in seiner Arbeit motiviert statt demotiviert. Dies ist gerade im Bereich der Softwarequalitätssicherung eine entscheidende Eigenschaft, denn hier kontrollieren wir Tester die Erzeugnisse der Entwickler, die bei den von ihnen entwickelten Features davon ausgehen, dass diese einwandfrei auf die gewünschte Art und Weise funktionieren. Bugreports z.B. sollten deshalb stets rein fachlich das Problem beschreiben, und wenn möglich sollte eine kurze Rücksprache mit dem verantwortlichen Entwickler gehalten werden. Persönliche Angriffe

fe, egal ob offen oder verdeckt, sind Gift für das Projekt und müssen unter allen Umständen vermieden werden.

Fähigkeit verschiedener Sichten (Interpreter)

Tester besetzen aus unserer Sicht eine Vermittlerrolle zwischen den einzelnen Teammitgliedern und müssen in der Lage sein, die Erwartungen und Probleme jedes Einzelnen nachvollziehen zu können. Im Qualitätssicherungsprozess ist diese Fähigkeit besonders dann von hohem Nutzen, wenn z.B. ein neues Feature abgenommen werden muss. Ist der Tester in der Lage, die verschiedenen Sichten (Kunde, Projektleiter/Product Owner, Entwickler) auf das Feature einzunehmen, steigert dies sein Verständnis von dem Testobjekt und erhöht das Niveau der Qualitätssicherung zu allen Zeitpunkten im Entwicklungsprozess.

Fähigkeit der gemeinsamen Sprache (Language)

Tester sind nicht nur Dolmetscher, sondern müssen auch Sprachlehrer sein. Fehlt das gemeinsame Verständnis für die Anforderungen an die zu entwickelnde Software, dann lauern zukünftig Probleme in der Funktionsweise und Qualität. Hier sollte ein Tester frühzeitig erkennen, wenn die unterschiedlichen Teammitglieder aneinander vorbeireden, auf diesen Umstand hinweisen und an der Erarbeitung einer gemeinsamen Sprache mitwirken.

Nachdem wir im vorhergehenden Abschnitt die bedeutendsten kommunikativen Kompetenzen von Testern in Softwareprojekten erläutert haben, wenden wir uns im nächsten Abschnitt den essenziellen persönlichen Kompetenzen zu.

Persönliche Kompetenzen (THE MOCCA PUB)

Zeitmanagement (Time Management)

Mit Zeitmanagement meinen wir die Fähigkeit, sowohl die eigenen als auch gegebenenfalls fremde Zeitressourcen effektiv einplanen, organisieren und kontrollieren zu können. Unsere Erfahrung zeigt uns, dass ein effektives Zeitmanagement allen beteiligten Personen Stress ersparen und Chaos verhindern kann. Hierzu gehört zum Beispiel auch die Anfertigung eines sorgfältig ausgearbeiteten Testplans, der neben den zu testenden

Features vor allem die Einplanung der zur Verfügung stehenden Testressourcen enthält.

Balance zwischen realistischer, positiver und negativer Sicht (Healthy Balance)

Healthy Balance bedeutet, dass wir verschiedene Blickwinkel einnehmen können, aber nie eine Perspektive überwiegen lassen. Die realistische Perspektive beinhaltet vor allem die Betrachtung des aktuellen Projektstatus anhand von zu messenden Parametern wie zum Beispiel verbliebene zeitliche Ressourcen. Die positive sowie die negative Perspektive betrachten anschließend diese Fakten aus dem jeweils gegenteiligen Blickwinkel. Dabei soll die positive Perspektive der Motivation des Teams dienen, indem zum Beispiel die bisher erreichten Fortschritte hervorgehoben werden. Die negative Sicht hingegen schärft den Blick für Engpässe, Versäumnisse und potenzielle Gefahrenquellen für das Projekt.

Genauigkeit (Exactness)

Software lässt sich grundsätzlich nicht nur auf eine Art und Weise bedienen. Vielmehr gilt die Regel, dass viele Wege nach Rom führen (können) und somit das Potenzial für Fehlerquellen dementsprechend hoch ist. Aus diesem Grund ist Genauigkeit beim Planen, Durchführen und Dokumentieren von Tests eine unabdingbare Fähigkeit von Softwaretestern, um so Verwirrung bei der Bestimmung des Qualitätszustandes zu vermeiden.

Methodenkenntnisse (Methods)

Das Testen von Software verlangt neben einer Affinität zu Technologie, Neugier, Kreativität und anderen Dingen natürlich auch spezifische Kenntnisse über das methodische Vorgehen in der Qualitätssicherung von Softwareprojekten. Die beste Möglichkeit, diese zu erlangen, sehen wir in der kontinuierlichen Weiterbildung der Tester zum Beispiel durch regelmäßige Fachlektüre sowie durch die Teilnahme an Schulungen und das Absolvieren von Zertifizierungen. Auf diese Weise soll sichergestellt werden, dass das Testen nicht zu einer zufälligen Folge von Aktivitäten verkommt, sondern bestimmten Regeln und Vorgehensweisen folgt. Hierzu kann auch die Kenntnis von Testautomatisierungstools und Managementsystemen gehören.

Organisationstalent (Organisation)

Organisationstalent bedeutet hier die Fähigkeit, Prioritäten setzen und deren Erledigung koordinieren zu können, möglichst unter optimaler Ausnutzung aller zur Verfügung stehenden Ressourcen. Hierdurch sollen zeitliche, personelle und finanzielle Engpässe vermieden werden.

Neugier (Curiosity)

Neugier stellt für uns den Schlüssel zu kreativem Arbeiten dar. Dies bedeutet, dass vorhandene Problemstellungen und/oder Lösungswege nicht einfach hin- und übernommen, sondern stets hinterfragt werden. Gerade in einem so komplexen Bereich wie der Softwareentwicklung stellt eine natürliche Neugier ein gutes Hilfsmittel zur Erarbeitung vielfältiger Herangehensweisen bei der Problemaufdeckung und -lösung dar. So sollten Softwaretests in der Regel nicht nur den „Happy Path“ abdecken, sondern auch komplexere Bedienmethoden beinhalten, um so mögliche Fehlerquellen aufzudecken.

Mut (Courage)

Das Ansprechen von Fehlern, seien es die von anderen oder die eigenen, fällt niemandem leicht. Doch genau dieses Aufdecken von Qualitätsmängeln oder Problemen ist eine Aufgabe von Testern in einem Projekt. Der Mut, eben diese Mängel anzusprechen und so dem Team vor Augen zu führen, stellt deshalb eine wichtige Eigenschaft dar. Das Verheimlichen von Missständen hilft niemandem und gefährdet den erfolgreichen Abschluss eines Projekts. Klar ist allerdings auch, dass dieser Mut durch das Team und besonders die verantwortlichen Personen gefördert werden muss: Das Arbeitsklima im Team muss Mut unterstützen.

Analytisches Denken (Analytical)

Hiermit meinen wir die Fähigkeit, Sachverhalte logisch auf eine „Wenn-dann“-Art zu betrachten und zu durchleuchten. Zusammenhänge werden erkannt und gedeutet und daran anschließend wird die passende Vorgehensweise geplant. Ein guter Tester muss analytisch und logisch denken und seine Denk- und Arbeitsweise nachvollziehbar beschreiben können.

Fähigkeit zum Setzen von Prioritäten (Prioritisation)

Mit diesem Soft Skill meinen wir die Fähigkeit, Prioritäten zu setzen für das Testen an sich. Eine vollständige Testfallüberdeckung ist beim dynamischen Testen einer Software unmöglich, deshalb muss der Tester Testdaten-Kombinationen auswählen, die einen großen Teil der Anwendungsfälle repräsentieren. Da Zeit stets ein rares Gut innerhalb eines Projekts darstellt, muss der Tester in der Lage sein, aus der Vielfalt der zu testenden Funktionen die wichtigsten auszuwählen, um deren Funktionstüchtigkeit zu überprüfen. Die Fähigkeit, Prioritäten zu setzen, ist hierbei von entscheidender Bedeutung.

Verständnis der Software und des Entwicklungsprozesses (Understand Software)

Hiermit ist gemeint, dass ein Tester die Fähigkeit besitzen muss, die einzelnen Phasen des Softwareentwicklungsprozesses nachvollziehen zu können, um daraus die passenden Schritte zur Qualitätssicherung herzuleiten. Dies kann sich zum Beispiel in der Auswahl bestimmter Testtools, die für den aktuellen Status des Projekts am geeignetsten sind, äußern. Weiterhin kann das Verständnis der Funktionsweise von Software dabei hilfreich sein, Fehler aufzudecken, die unter Umständen von anderen Fehlern verdeckt werden.

Lernfähigkeit (Stay on the Ball)

Gerade in einem so schnelllebigem Bereich wie der Softwareentwicklung ist es wichtig, stets auf dem Laufenden zu sein und sich mit neuen Entwicklungen zu befassen. Dazu gehört nicht nur die Kenntnis aktueller Testtools und -verfahren, sondern auch das Wissen über Prozessorganisation und neue Technologien in

der Softwareentwicklung. Hierbei helfen die regelmäßige Lektüre von Fachliteratur sowie der Besuch von Konferenzen und Seminaren. Doch Lernfähigkeit beschränkt sich nicht auf das Testen im Allgemeinen. Ein Großteil aktueller Softwareprojekte wird durch eine Vielzahl unterschiedlicher Stakeholder beeinflusst. Eine ausgeprägte Lernfähigkeit hilft dem Tester dabei, die verschiedenen Sichten und Interessen zu verstehen und nachzuvollziehen und somit die Qualitätssicherung aus den unterschiedlichen Blickwinkeln der Stakeholder zu betreiben.

Fazit

Die aus unserer Sicht essenziellen Fähigkeiten moderner Softwaretester lassen sich wie erläutert in die Bereiche Kommunikations- und persönliche Kompetenzen einteilen. Was bei einer genaueren Betrachtung der aufgezählten Fähigkeiten auffällt, ist, dass diese Kompetenzen an sich nichts Neues darstellen und vom Tester auch nichts Unmögliches verlangen. Natürlich verlangt jedes Projekt nach weiteren ganz spezifischen Kenntnissen und Fähigkeiten, doch nach unserer Erfahrung stellen die zuvor ausgearbeiteten Kompetenzen ein solides Grundset für eine erfolgreiche Mit- und Zusammenarbeit dar. Dabei wollen wir an dieser Stelle klarstellen, dass diese Kompetenzen nicht nur exklusiv für Tester hilfreich sind, sondern auch für alle anderen Teammitglieder ein nützliches Skillset zur produktiven Mit- und Zusammenarbeit darstellen. Eine mögliche Zukunft des Testens wäre, dass jeder im Team lernt, auch auf diese Art und Weise zu denken. Weiterhin sollte klar sein, dass wir mit den von uns erarbeiteten Kompetenzen keinerlei Anspruch auf Vollständigkeit erheben wollen. Wer sich über diesen Artikel hinausgehend mit in der Qualitätssicherung hilfreichen Kompetenzen auseinandersetzen möchte, sei deshalb an dieser Stelle an die einschlägige Fachliteratur zu Soft Skills und Softwaretests verwiesen. ■