



□ Dr. Frank Simon

(frank.simon@german-testing-board.info)

befasst sich seit über 15 Jahren mit Software-Qualitätssicherung. Er leitet den Bereich „Research“ der SQS Software Quality Systems AG und ist Mitglied des German Testing Board e. V. (GTB) sowie des BITKOM, wo er den Arbeitskreis „Software-Entwicklungsprozesse und Tools“ leitet.

Effizientes „Mobile Software Testing“

„Mobile Software“ scheint Software-Tester vor neue Herausforderungen zu stellen. Dabei bleibt eigentlich vieles beim Alten. Denn: Für effizientes Testen ist nach wie vor nicht allein spezifisches Know-how entscheidend, sondern Systematik im Vorgehen und die Fähigkeit, Testprozesse zu integrieren. Bewährte Grundlagen behalten also auch in Zeiten von „Mobile“ ihre Gültigkeit – und Weiterbildungen wie die zum ISTQB® Certified Tester ihre Berechtigung. Eine Begründung.

Die Märkte für Smartphones und Tablet-Computer ziehen kräftig an: BITKOM zufolge wuchsen beide allein im vergangenen Jahr um 13 bzw. 14 Prozent (vgl. [Bit12] und [Bit11]); der Branchenverband beziffert den Umsatz, den die deutsche Industrie mit den mobilen Geräten 2011 erwirtschaftet hat, auf zusammen rund 5,2 Mrd. Euro. Mit der Nachfrage nach mobilen Geräten wächst auch das Interesse an mobiler Software, also Betriebssystemen und Anwendungen, die speziell für Smartphones und Tablet-Computer entwickelt werden. Vor allem der neue Teilmarkt für Apps wächst stark: Bis 2013 sollen mit den Kleinanwendungen weltweit mehr als 12 Mrd. Euro erwirtschaftet werden – 2009 waren es noch 1,6 Mrd. Euro (vgl. [Res11]).

Der weltweite Siegeszug mobiler Software hat also begonnen. Und er erstreckt sich nicht mehr nur auf „kleine Spielchen“ oder nette Gimmicks, sondern bezieht immer häufiger Geschäftsanwendungen mit ein, bei denen es um Geld und Risiko geht: 40 % aller Smartphone-User verwenden ihr Smartphone etwa bereits für Finanzservices wie Konteneinsicht (52 %) und Finanztransaktionen (42 %) (vgl. [Syn11]). Damit eröffnet dieser Trend nicht nur Wachstumschancen für Entwickler. Auch Software-Tester, die sich auf die Prüfung mobiler Anwendungen verstehen und deren Einsatz gerade bei geschäftskritischen Anwendungen wichtig ist, werden zunehmend gebraucht – schließlich müssen Tablets und Smartphones im Business

genauso sicher und zuverlässig funktionieren wie Desktop-Rechner.

Dass das bisweilen eine Herausforderung sein kann, haben spektakuläre System- oder Software-Fehler wie das Antennen-Setup eines neuen Smartphones oder die Fehler in der Speicheradressierung eines neuen mobilen Betriebssystems gezeigt: Beide Fehler waren erst nach der Markteinführung bekannt geworden, mit zumindest kurzzeitig negativen Folgen für die jeweiligen Unternehmen (die z. T. durch Entschädigungszahlungen an enttäuschte Anwender minimiert wurden).

Auch die zunehmenden Sicherheitsrisiken für mobile Endgeräte – Juniper MTC hat z. B. knapp 800.000 Anwendungen auf Malware untersucht und fast 29.000 Befunde identifiziert (vgl. [Jun12]) – sorgen für Handlungsbedarf hinsichtlich Qualität.

Fehler und Risiken wie diese zu vermeiden, ist keineswegs trivial. Das Testen von mobilen Geräten und der entsprechenden Software (Mobile Software Testing) stellt Unternehmen und Software-Tester vor große Herausforderungen – sowohl handwerklicher als auch wirtschaftlicher Art.

Hohe Komplexität, großer Zeitdruck

Das Testen mobiler Software ist komplex. Die von den Testern zu berücksichtigenden „Eco Systems“ bestehen eben nicht nur aus der eigentlichen Hard- und Software, sondern auch aus zahlreichen Drittsystemen wie etwa Software-Infrastrukturen der Geräte-

hersteller – z. B. Shop-Systeme wie Apple App Store oder Google Play Store – und dem Funknetz mit seinen zahlreichen Standards und Bandbreiten. Auch die Schnittstellen eines mobilen Endgeräts wie Bluetooth, Wi-Fi oder NFC sind vielfältiger als solche von festen Endgeräten. Entsprechend hoch ist die Anzahl der Testartefakte, die untersucht, und die Zahl der Tests, die geplant und umgesetzt werden müssen.

Für viel Aufwand sorgt dabei die Vielzahl der Hardware- und Betriebssystemvarianten: Smartphones und Tablets unterscheiden sich bisweilen selbst bei Geräten aus ein und derselben Baureihe, weil OEM-Hersteller, Netzbetreiber und Händler länderspezifische oder markenspezifische Anpassungen an Hard- oder Firmware vornehmen. Das Gleiche gilt für die zahlreichen gängigen Betriebssysteme und deren Versionsvielfalt sowie OEM-Varianten. Eine Software, die auf allen diesen Systemen laufen soll, muss in der Regel individuell entwickelt und getestet werden. Im Zweifel bedeutet das: ein Entwicklungs- und ein Testverfahren für jedes Betriebssystem, jede Version, jede Hardware usw. Eine kosten- und zeitintensive Tätigkeit.

Viel Zeit steht dafür jedoch meist nicht zur Verfügung. Die Entwicklungs- und Release-Zyklen für mobile Geräte sind sehr kurz; zwischen zwei Generationen eines Geräts bzw. einer Betriebssystemversion liegen in der Regel nur wenige Monate. Entsprechend hoch ist der Zeitdruck, unter dem die Entwickler und Tester arbeiten.

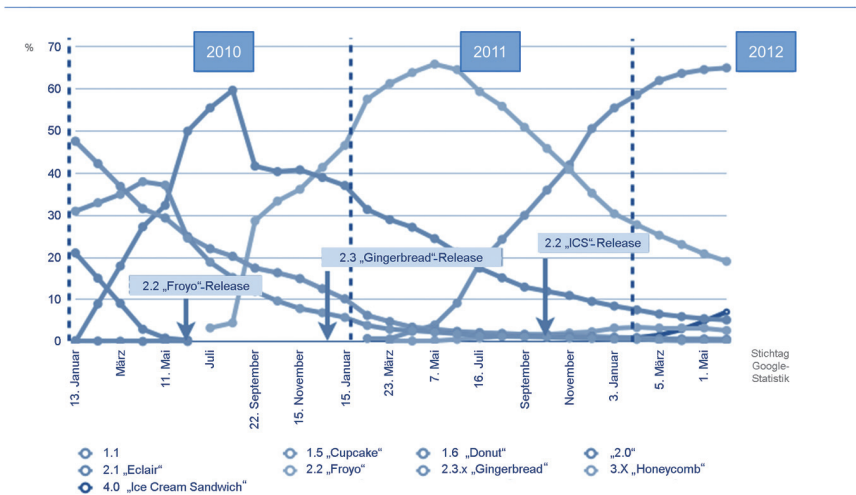


Abb. 1: Android-Versionen im Zeitverlauf (Anmeldungen im Google Play Store).

Spielräume für die Abstimmung der verschiedenen Verfahren und Teams gibt es kaum. **Abbildung 1** zeigt die daraus resultierende Dynamik am Beispiel des Android-Betriebssystems (vgl. [Gua12]), basierend auf Zahlen von Google, welche Android-Versionen den Google Play Store innerhalb einer 14-Tage-Frist besuchen.

Effektivität und Effizienz

Wer diese beiden Herausforderungen bewältigen will – d.h. die Komplexität beherrschen und gleichzeitig knappe Zeitvorgaben einhalten – muss systematisch arbeiten, um nicht für jedes neue Zielgerät einen kompletten Testprozess einrichten zu müssen. Das setzt entsprechende Strukturen und Abläufe für das Software Testing voraus beziehungsweise die auf höchste Wirksamkeit und Effizienz gerichtete Systematisierung, Standardisierung und Integration von Testentwicklungs- und Testprozessen. Der Grundsatz hierfür lautet: So viel Effektivität wie möglich (z.B. Testabdeckung), so wenig Aufwand bzw. Zeit wie nötig (z.B. möglichst viele Schritte trotz unterschiedlicher Zielsysteme nur einmal durchführen).

Unternehmen, die das nicht leisten können, werden mit dem Testen mobiler Software Schwierigkeiten bekommen – auch wenn sie über noch so viele Spezialisten mit herausragendem Mobilfunk-Know-how verfügen. Welche Schwierigkeiten das sind, mag ein (sicher überspitztes, aber keineswegs ganz wirklichkeitsfernes) Beispiel zeigen:

Ein Software-Unternehmen hat entlang klar umrissener Anforderungen eine App

entwickelt, die auf allen fünf gängigen Smartphone-Betriebssystemen laufen soll. Diese soll jetzt für jedes der besagten fünf Betriebssysteme getestet werden – und zwar so, dass bei Auslieferung kein kritischer Fehler mehr in der Software vorhanden ist. Die verantwortlichen Testmanager entscheiden sich, fünf Testteams aufzustellen, eines für jedes Betriebssystem. Jedes Team soll dabei auf Basis der Anforderungen risikobasierte Tests für das jeweilige Zielsystem planen, entwerfen und durchführen.

Fünf Systeme, fünf Teams, fünf spezifische Testreihen: Das scheint auf den ersten Blick schlüssig; so sollte sich das angestrebte Null-Fehler-Ziel erreichen lassen. Auf den zweiten Blick wird allerdings erkennbar: Sehr effizient ist diese Vorgehensweise nicht. Mehrere Teams, die jeweils Testfälle gemäß den Anforderungen an die Software entwerfen – da werden sicherlich viele inhaltsgleiche Aufgaben mehrfach bearbeitet. Zudem würde dieses Duplikationsphänomen innerhalb der Teams fortgesetzt: Da gäbe es ein Testteam für Android 2.1, eines für 2.3. und eines für 4.0. Selbiger Ansatz könnte auch für die unterschiedlichen iOS-Versionen Verwendung finden. Insgesamt nur ein bedingt funktionierender Ansatz, der heute häufig mittels „Mut zur Lücke“ geklärt wird: „Wer hat schon Betriebssystem X in der Version Y auf dem Gerät Z? Das tut's sicher schon.“

Das dargestellte Vorgehen „n Geräte = n Testprozesse“ findet sich in der wirklichen Praxis natürlich kaum: Meist finden die eingesetzten Testteams zügig einige Aktivitäten, deren Ergebnisse für mehrere

Geräte und somit die jeweils anderen Teams relevant sind. Aber: Die Identifikation solcher Schnittmengen verläuft meist ungesteuert, was zu fehlenden Synergien (immer noch viel Redundanz) sowie zu falschen Synergien (Spezifika werden gar nicht mehr berücksichtigt) führen kann.

Systematisieren und Integrieren

Unternehmen und Testspezialisten, die wirklich effizient vorgehen, arbeiten daher in der Regel anders: Sie entwickeln systematisch standardisierte Prozesse für den Test, die darauf abzielen, unterschiedliche Testvorgänge – im Beispiel die für die fünf Betriebssysteme – weitestgehend in einen gemeinsamen Vorgang zu integrieren, d.h. Arbeitsschritte, die für alle Testvorgänge einheitlich bzw. zumindest ähnlich sind, nur einmal vorzunehmen und individuelle Anpassungen als Add-on-Variation zu bearbeiten. Übertragen auf unser Beispiel könnte dies heißen: Das Testteam führt eine strukturierte Qualitätssicherung auf Grundlage der zuvor festgelegten Anforderungen durch (Early Error Detection, EED). Fehler in den Anforderungen haben für alle Zielsysteme negative Auswirkungen und müssen nur einmal entfernt werden. Auch die Risikoanalyse auf Basis der Anforderungen kann größtenteils universal durchgeführt werden. Gleiches gilt für die Testobjektabgrenzung, die Testfallerstellung („high-level“ und „low-level“) usw. Spezifika in den Anforderungen (etwa die nicht auf allen Geräten vorhandene Multi-Touch-Eingabe) werden separat bearbeitet, machen aber in der Praxis i.d.R. weniger als 10 % aus. Die Variantenvielfalt beschränkt sich damit nur noch auf die Testausführung auf dem jeweiligen Zielgerät (entweder über Emulatoren, Roboter oder Automatisierungen auf dem Gerät selbst). Ein überwiegender Teil der Testware selbst kann hierbei jeweils einfach wiederverwendet werden. Der Unterschied zwischen diesen beiden Vorgehensweisen ist in **Abbildung 2** schematisch dargestellt.

Diese Vorgehensweise ist sowohl effektiv (gemessen in Testabdeckung der Anforderungen) als auch effizient (gemessen in Zeit, Aufwand und Ressourcen), setzt aber voraus, dass das Testen nicht als chaotische Ad-hoc-Tätigkeit verstanden wird, sondern als disziplinierte Ingenieursdisziplin. Tester müssen nämlich zusätzlich zum Ad-hoc-Testen in der Lage sein, systematisch vorzugehen und Testentwicklungsprozesse zu

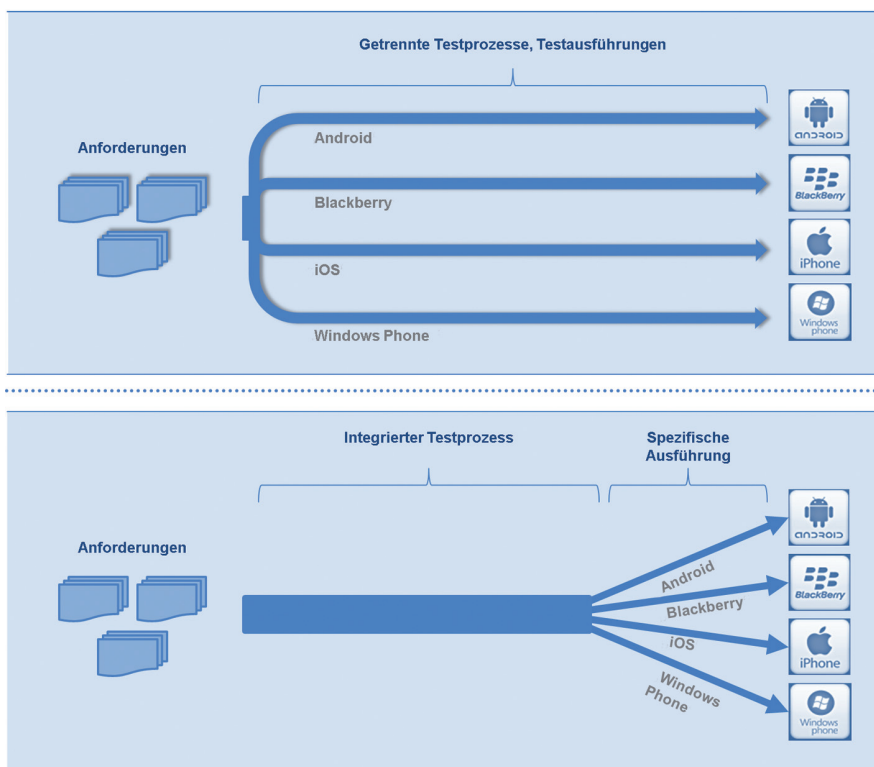


Abb. 2: Testprozess und Testausführung.

entwerfen, auszuführen, zu überwachen und gegebenenfalls zu verbessern. Und dies alles nicht jeweils nur unter der Fragestellung des grundsätzlich Machbaren, sondern unter Berücksichtigung auch betriebswirtschaftlicher Gesichtspunkte wie Zeit- und Kostenrahmen.

Gute Ausbildung als Schlüssel

Darin wird das Mobile Software Testing nun anderen Arten des Softwaretestens sehr ähnlich: Effektives und effizientes Vorgehen erfordert zwar viel spezifisches Test-Know-how, in diesem Fall z. B. Wissen um die unterschiedlichen Smartphone-Betriebssysteme. Es erfordert aber auch – und vor allem – die Fähigkeit, systematisch zu denken, Prozesse zu planen, diese auf wirtschaftliche Art und Weise umzusetzen und, falls nötig, laufend zu verbessern. Diesbezüglich unterscheidet sich das Mobile Software Testing also nicht vom klassischen Testen. Neu ist lediglich die explodierende Vielfalt von Endgeräten, Betriebssystemen, mobilen Zugängen, etc., die dieses systematische Vorgehen nahezu unumgänglich macht.

Unternehmen, die effizientes Mobile Software Testing betreiben wollen, sollten also darauf achten, dass die von ihnen eingesetzten Software-Tester über beide Fähig-

keiten verfügen: Grundlagen systematischen Testens plus Systemspezifika mobiler Endgeräte. Der Schlüssel hierzu ist eine gute Ausbildung der jeweiligen Fachleute.

Das erstgenannte „Skill Set“ erwerben viele Tester in der Regel als Erfahrungswissen im Rahmen ihrer Arbeit oder bei entsprechenden Einarbeitungen bei ihrem Arbeit- oder Auftraggeber. Allerdings kann dieses Wissen evtl. schon zu proprietär und zu wenig generalisierbar sein. Für das auf systematisches Arbeiten bezogene Skill Set absolvieren daher viele zusätzlich eine grundlegende, breiter aufge-

stellte Ausbildung im Bereich Testen. Die branchenweit gängigste ist der international anerkannte „Certified Tester“ des unabhängigen Software-Tester-Verbands ISTQB® (International Software Testing Qualifications Board). Über drei Weiterbildungs-Level („Foundation“, „Advanced“, „Expert“) lernen Software-Tester in unterschiedlichen Bereichen (z.B. Testmanagement) alles über die effektive sowie effiziente Planung und Steuerung von Tests aller Art. Aktuell können bereits über 200.000 Tester eine dieser Zertifizierungen ihr Eigen nennen und damit systematisches Testwissen vorweisen. Das für Deutschland zuständige nationale Board GTB (German Testing Board) wacht hierbei permanent über die Aktualität und Angemessenheit der verwendeten Trainingskurse und Prüfungen.

Fazit

Das Testen von Software für mobile Endgeräte weist einige Besonderheiten auf, die das effektive und effiziente Software-Testen auf den ersten Blick erschweren: Es gibt gänzlich neue „Eco Systems“, in denen die Software laufen kann, wodurch sich der Fokus eines Testmanagers neu ausrichten sollte. Kurze Entwicklungs- und Release-Zyklen sorgen zudem für erheblichen Zeitdruck, was hocheffizientes Testen unabdingbar macht. Diese Besonderheiten erfordern ein systematisches, strukturiertes und integriertes Vorgehen, wenn sowohl wirksam, schnell und ohne unnötige Mehrfacharbeiten getestet werden soll. Software-Tester, die eine entsprechende gute Ausbildung mitbringen, können dieses Vorgehen leisten – und sind letztlich der Schlüssel zu effektivem und effizientem Mobile Software Testing. ■

Referenzen

[Bit11] BITKOM: Tablet Computer erobern den Massenmarkt. Pressemitteilung vom 14. Dezember 2011.
 [Bit12] BITKOM: Smartphone-Absatz steigt rasant. Pressemitteilung vom 09. Januar 2012.
 [Gua12] Guardian: <http://www.guardian.co.uk/technology/appsblog/2012/jun/03/google-play-android-ics-share>.
 [Jun12] Juniper Networks: 2011 Mobile Threats Report, published February 2012, siehe http://www.juniper.net/us/en/local/pdf/additional-resources/jnpr-2011-mobile-threats-report.pdf?utm_source=promo&utm_medium=right_promo&utm_campaign=mobile_threat_report_0212.
 [Res11] Research2Guidance: Global Smartphone Application Market Report 2010. Berlin, 2011; eigene Berechnungen.
 [Syn11] Synergistics Research: Aware of availability of mobile banking at financial institutions used, siehe <http://rpmconsultingllc.blogspot.de/2011/01/mobile-banking-trends-from-synergistics.html>.