

Mobile Testing mit Perfecto Mobile

Kategorie: Perfecto Mobile bietet für den Test von Software für mobile Endgeräte eine Lösung, die aus Software, Hardware und Services besteht. Von daher die Lösung Perfecto Mobile schwer in eine bestimmte Kategorie einsortiert werden.

Autor: Dirk O. Schweier

- Zweck:** Perfecto Mobile bietet eine Dienstleistung an, die sowohl die Bereitstellung von mobilen Endgeräten als auch verschiedene Applikationen umfasst, mit denen die Kunden von Perfecto Mobile ihre eigenen Applikationen auf echten mobilen Endgeräten manuell und automatisiert testen können.
- Highlights:**
1. Tester müssen sich nicht um Beschaffung und Bewirtschaftung von Mobilien Testgeräten kümmern.
 2. Für unterschiedliche Sicherheitsanforderungen gibt es unterschiedliche Dienstleistungsmodelle.
 3. Manuelle Tests, Testautomatisierung und exploratives Testen werden komfortabel unterstützt.
 4. Offene Schnittstellen erlauben die Integration in bestehende Entwicklungs- und Testumgebungen (wie z.B. Entwicklung mit Eclipse, Continuous Integration mit Jenkins, Lasttest mit NeoLoad).
- Hersteller:** Perfecto Mobile
- Webseite:** <http://www.perfectomobile.com/>

Inhaltsverzeichnis

- 1 Einleitung..... 5
 - 1.1 Wer sollte das Dokument lesen?..... 5
 - 1.2 Begrifflichkeiten in dem Dokument 5
- 2 Management Summary..... 6
- 3 Aufgabenstellung: Automatisierte Tests von Q-Chess Mobile 7
 - 3.1 Testobjekt..... 8
 - 3.2 Testumgebung..... 8
 - 3.3 Bereitstellung der Hardware durch Perfecto Mobile 9
 - 3.4 Perfecto Mobile als Testausführungswerkzeug 12
- 4 Erkenntnisse aus dem Einsatz von Perfecto Mobile 12
 - 4.1 Flexibel in der Wahl der Teststrategie..... 12
 - 4.1.1 Manuelles Testen 12
 - 4.1.2 Automatisierte Tests 14
 - 4.1.3 Exploratives Testen 16
 - 4.2 Anbindung an andere Testwerkzeuge..... 17
 - 4.2.1 Entwicklungsumgebung Eclipse 17
 - 4.2.2 Performance- und Lasttests mit NeoLoad 17
 - 4.2.3 Continuous Integration mit Jenkins 18
 - 4.3 Erstellung von wartbaren Skripten..... 19

Abbildungsverzeichnis

Abbildung 1 Prinzipskizze der Testumgebung 9
Abbildung 2 Webapplikation von Perfecto Mobile mit Sicht auf zwei Endgeräte 13
Abbildung 3 Entwicklung von Skripten mit Perfecto Mobile 14
Abbildung 4 Konfigurationsdialog für benutzerdefinierte Funktionen 20

Tabellenverzeichnis

Tabelle 1 Aufstellung der verfügbaren Endgeräte für den Test 11

1 Einleitung

1.1 Wer sollte das Dokument lesen?

Die Zielgruppe für dieses Dokument sind in erster Linie Tester, die in unterschiedlichen Rollen an dem Thema „Mobile Testing“ interessiert sind oder ganz konkret vor der Herausforderung stehen, Tests mit mobilen Endgeräten durchführen zu müssen.

Im Kapitel 3 wird eine Aufgabenstellung mit Lösung zum Thema „Mobile Testing“ beschrieben, die sich der Firma G. Muth Partners GmbH gestellt hat. Ein bestehendes Softwaresystem wurde um eine App für das Betriebssystem Android erweitert und musste getestet werden.

Im Kapitel 4 werden die Ergebnisse aus einem Pilotprojekt unter drei zentralen Schlagworten beleuchtet:

- **Konsequenzen für die Wahl der Teststrategie**
Dieser Abschnitt richtet sich in erster Linie an Testmanager und zeigt auf, wie Perfecto Mobile die in diesem Dokument angesprochenen Teststrategien unterstützen kann.
- **Möglichkeiten der Anbindung an andere Testwerkzeuge**
Testmanager, die Perfecto Mobile in eine bestehende Werkzeuglandschaft des Unternehmens integrieren wollen, bekommen in dem Teil konkrete Werkzeuge genannt, die Schnittstellen zu Perfecto Mobile haben. Die Aufstellung der Werkzeuge ist nicht vollständig, und der Schwerpunkt liegt auf der Aufgabenstellung des genannten Werkzeugs.
Dieser Teil ist aber auch für Testautomatisierungsexperten und Softwareentwickler von Interesse, da diese eventuell heute einige der angesprochenen Werkzeuge nutzen. Diese Gruppe wird Sie finden hier Anregungen zur Nutzung von Synergieeffekten finden.
- **Wartbarkeit der entstandenen Testskripte**
Dieser Abschnitt wendet sich in erster Linie an Testautomatisierungsexperten, da hier Hinweise gegeben werden, wie Testskripte unter dem Aspekt der Wartbarkeit und Wiederverwendung auf unterschiedlichen Betriebssystemen zu entwickeln sind.

Das Dokument erhebt nicht den Anspruch, systematisch die Grundlagen des „Mobile Testings“ zu definieren, sondern zeigt einen pragmatischen Lösungsansatz auf.

1.2 Begrifflichkeiten in dem Dokument

Die in diesem Dokument verwendeten Fachbegriffe orientieren sich an dem Glossar des ISTQB.

2 Management Summary

Mit der Vorstellung des Apple iPhone im Jahre 2007 ist es den Smartphones und Tablets gelungen, einen signifikanten Marktanteil zu gewinnen der seitdem auch stetig gestiegen ist. Allein in Deutschland wurden in 2014 laut Statista¹ 24,2 Millionen Smartphones verkauft. Hinter dieser Zahl steht somit auch eine große Anzahl an Nutzern dieser mobilen Endgeräte.

Für Unternehmen aus fast allen Branchen sind die Nutzer der mobilen Endgeräte Zielgruppe für die Produkte oder Dienstleistungen des Unternehmens. Also werden Anwendungen, im Folgenden Apps genannt, in diesem Kontext entwickelt um der Zielgruppe über das mobile Endgerät Produkte, Dienstleistungen oder ganz allgemein Informationen bereitzustellen, die dann durch Testabteilungen getestet werden müssen.

Die Tatsache, dass man in diesem Zusammenhang mit „Mobile Testing“ einen neuen Fachbegriff nutzt, macht deutlich, dass das Testen in diesem Umfeld neuen Regeln folgt.

Anhand eines konkreten Beispiels aus unserem Unternehmen, bei dem Perfecto Mobile zum Einsatz kam, werden einige der Herausforderungen formuliert, die sich dem Unternehmen mit dem Test einer App für Android gestellt haben.

Dabei wird sehr schnell deutlich, dass das Produkt von Perfecto Mobile sich nicht einer bestimmten Kategorie zuordnen lässt, da die einzelnen Bestandteile Software, Hardware und auch Dienstleistungen sind.

Das Bereitstellen von mobilen Endgeräten für die Tester ist die wesentliche Dienstleistung, die Perfecto Mobile anbietet. Dabei kann der Kunde wählen, ob er die Geräte stundenweise mieten und mit einer Nutzergemeinde teilen möchte oder eine bestimmte Anzahl von Geräten exklusiv zur Benutzung bereitgestellt bekommen möchte. Damit entfällt eine aufwändige Beschaffung und Bewirtschaftung von mobilen Endgeräten im eigenen Haus. Zusätzlich kümmert sich Perfecto Mobile darum, dass in der sogenannten Public Cloud auch immer die Geräte verfügbar sind, die einen hohen Marktanteil haben. Diese Aktualisierung ist für einzelne Firmen angesichts der hohen Anzahl an Neuerscheinungen eine kostspielige Angelegenheit.

Die Perfecto-Webapplikation, mit der Tester auf die Geräte zugreifen können und auch Skripte zur Testautomatisierung entwickeln können, erlaubt dem Testmanager unverändert den Freiheitsgrad, manuell oder automatisiert zu testen. Damit wird der Einstieg in das Thema erleichtert, da Kenntnisse in der Testautomatisierung nicht zwingend erforderlich sind.

¹ siehe <http://de.statista.com/statistik/daten/studie/77637/umfrage/absatzmenge-fuer-smartphones-in-deutschland-seit-2008/>

Wer schnell einsteigen möchte, wird parallel zur Entwicklung der Testskripte manuell testen und nach und nach den Grad der Automatisierung erhöhen.

Der Nutzen der Testautomatisierung mit Perfecto Mobile wird zusätzlich durch Synergien mit anderen Werkzeugen unterstützt. Besonders hervorzuheben ist hier eine Anbindung an den Continuous Integration Prozess, der den Buildprozess um automatisierte Tests ergänzt und damit kontinuierlich die Softwareentwicklung und die Testabteilung mit Aussagen zur Softwarequalität versorgt .

3 Aufgabenstellung: Automatisierte Tests von Q-Chess Mobile

Die Firma G. Muth Partners GmbH hat mit Q-Chess eine Standardsoftware, die die Durchführung von Reviews unterstützt. Die Software erlaubt das Anlegen und Bearbeiten von Checklisten, die in der Software unter Versionskontrolle stehen. Diese Checklisten können für Reviews über einen Arbeitsplatzrechner oder ein mobiles Endgerät bereitgestellt werden. Die Protokollierung findet rechnergestützt statt und die Ergebnisse werden in einer zentralen Datenbank auf dem Q-Chess Server gespeichert.

Da der Einsatz von Tablets von Smartphones auch im betrieblichen Umfeld immer stärker zunimmt, wurde von G. Muth Partners GmbH mit Q-Chess Mobile eine App für Android entwickelt.

Q-Chess Mobile stellt die Testabteilung vor die Aufgabe, die App in den Systemtest von Q-Chess zu integrieren. Aus Sicht des Testmanagers stellte sich seine Aufgabenstellung wie folgt dar:

- Durchführung der Tests auf einer möglichst unterschiedlichen mobilen Endgeräten
- Testdurchführung soll automatisierbar sein
- Nutzung von Synergien in Verbindung mit bestehenden Test- und Entwicklungsprozessen
- Realisierung von Schnittstellen der zur Integration der Tests in eine bereits bestehende Werkzeuglandschaft

Mit Q-Chess Mobile steht dem Kunden ein Client zur Verfügung, der unter dem Betriebssystem Android lauffähig ist. Über den Q-Chess Sync Server findet die Kommunikation zwischen dem Q-Chess Server und Q-Chess Mobile statt. Auf dem mobilen Endgerät lassen sich einmal übertragene Checklisten auch ohne Verbindung zum Server für Reviews nutzen. Die Ergebnisse werden ebenfalls lokal gehalten und der Anwender kann bei bestehender Verbindung die Übertragung vom und zum Server selbst veranlassen.

3.1 Testobjekt

Bei Q-Chess handelt es sich um ein System, das aus unterschiedlichen Softwaremodulen besteht, die teilweise mit unterschiedlichen Programmiersprachen und für unterschiedliche Umgebungen entwickelt wird.

Ein Modul ist die Android App Q-Chess Mobile. Sie wird für mobile Endgeräte mit Betriebssystem Android entwickelt und soll für die Version 4.0.3 und neuer freigegeben werden.

Der Client Q-Chess Mobile hat eine Schnittstelle zum System Q-Chess über den Q-Chess Sync Server. Der Q-Chess Sync Server übergibt Verwaltungsinformationen und Checklisten an den Client und nimmt die Reviewdaten vom Client entgegen. Diese Kommunikation wird vom Anwender gezielt angestoßen. Damit ist Q-Chess Mobile auch für den Einsatz an Orten ohne Netzwerk einsetzbar.

Als Testobjekt für den durchzuführenden Mobile Test wird Q-Chess Mobile definiert.

Der Q-Chess Sync Server ist zwar Bestandteil der Testumgebung, wird jedoch in dieser Aufgabenstellung nicht als Testobjekt definiert.

3.2 Testumgebung

Mobile Endgeräte mit Betriebssystem Android 4.0.3 und neuer definieren den Kern der Testumgebung für die Software Q-Chess Mobile.

Der Q-Chess Sync Server ist ein weiterer Bestandteil der Testumgebung, da nach der Installation von Q-Chess Mobile eine initiale Datenübertragung notwendig ist, damit Verwaltungsdaten und Checklisten initial übertragen werden können. Der Q-Chess Sync Server dient zusätzlich auch als Stub wenn es darum geht, die Übertragung in Richtung Q-Chess Server auszuführen.

Der Q-Chess Client wird aus Sicht des Tests genutzt, um die Checklisten zu bearbeiten und die Daten für die mobilen Endgeräte zu disponieren.

Die Abbildung 1 liefert einen grafischen Überblick über die Testumgebung und die enthaltenen Bestandteile.

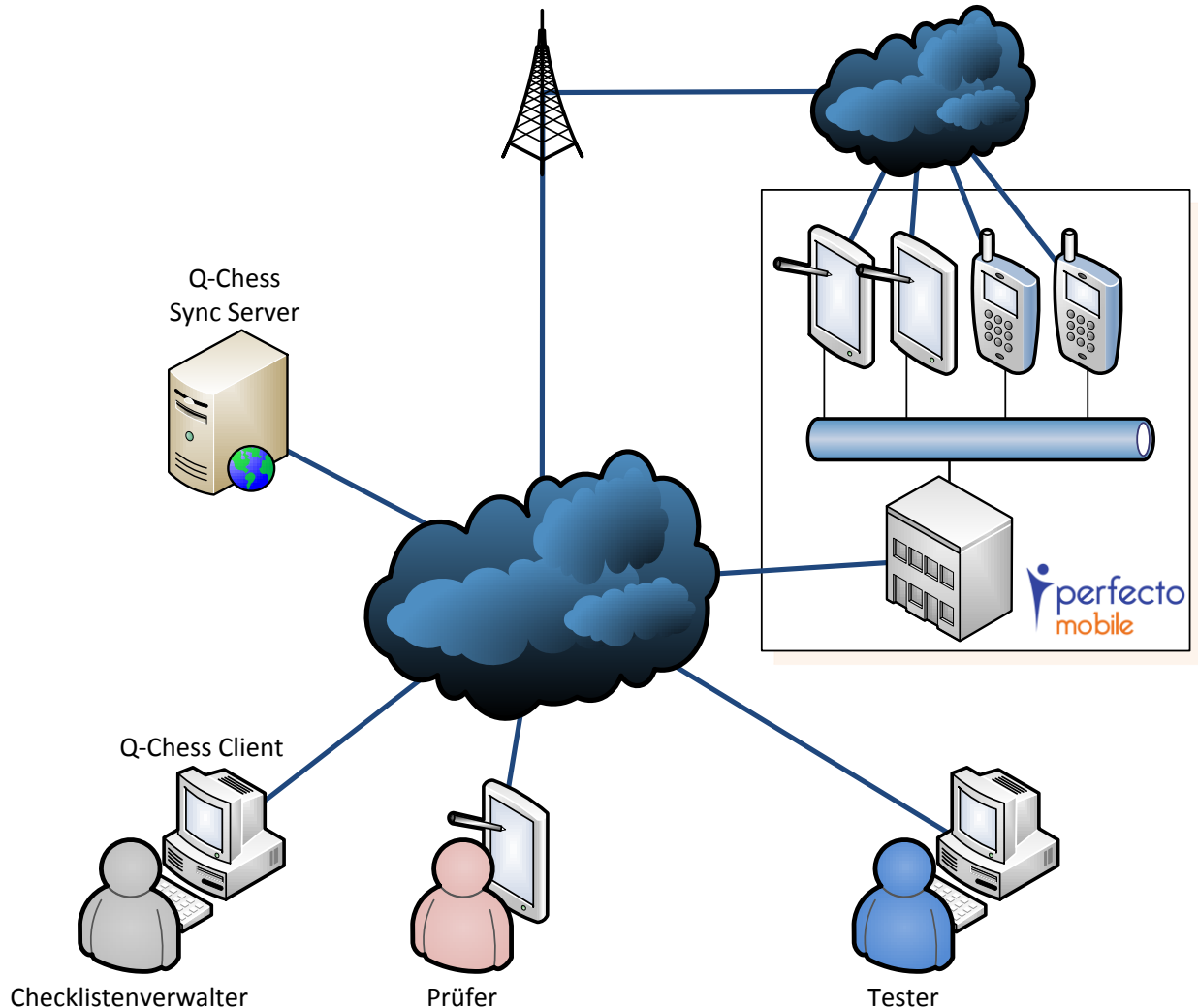


Abbildung 1 Skizze der Testumgebung

3.3 Bereitstellung der Hardware durch Perfecto Mobile

Bereits bei den ersten Vorüberlegungen zur Testabdeckung haben sich zwei wesentliche Herausforderungen in Bezug auf die einzusetzenden mobilen Endgeräte gezeigt:

- Bewirtschaftung der angeschafften mobilen Endgeräte
- Aktualisierung des angeschafften Gerätepools

Wenn mit mobilen Endgeräten getestet werden soll, dann stellen sich nach der Anschaffung Fragen hinsichtlich der Aufbewahrung der Geräte und Verteilung der Geräte auf die Tester.

Wenn der Test beginnt, dann sollten die Geräte aufgeladen sein, damit nicht die Länge des Ladekabels Auswirkungen auf den Aktionsradius des Testers hat.

Ein weiterer Aspekt ist die Wartung der mobilen Endgeräte in Bezug auf die installierte Software. Hier stellt sich zum Beispiel die Frage, wie man das mobile Endgerät nach Installation von Software wieder in den Zustand versetzen kann, den das mobile Endgerät vor dem Test hatte?

Alle diese Themen machen deutlich, dass beim Mobile Testing neue Fragestellungen auftauchen, die sich dem Testteam bisher nicht gestellt haben, da die „normalen“ Tests mit dem Arbeitsplatzrechner und einer lokalen Testumgebung durchgeführt werden konnten.

Das Geschäftsmodell von G. Muth Partners GmbH sieht vor, dass sich der Kunde die Hardware für Q-Chess selbst beschafft. Damit ist im Vorfeld nicht klar, auf welchen mobilen Endgeräten Q-Chess Mobile durch den Kunden zum Einsatz kommen wird. Daher müssen die für den Test anzuschaffenden Geräte stets einen repräsentativen Durchschnitt vom Markt widerspiegeln.

Die Testgeräte müssen entweder von einem Mitglied des Testteams bzw. einer Person innerhalb der Unternehmung besorgt werden oder die Beschaffung wird ausgelagert. Im konkreten Fall wurde entschieden, diese Aufgabe als Dienstleister an Perfecto Mobile anzugeben.

Perfecto Mobile bietet in Bezug auf die Bereitstellung von Hardware grundsätzlich drei Modelle an, die auch kombiniert werden können:

- Public MobileCloud

Die Public MobileCloud ist eine große Auswahl an mobilen Endgeräten, die in bestimmten Rechenzentren liegen, die von Perfecto Mobile oder seinen Partnern betrieben werden. Die Kunden können auf den Bestand zugreifen und teilen sich die Geräte untereinander. Dieses Modell deckt die Top 100 Geräte am Markt ab und der Bestand wird regelmäßig an die neuesten Verkaufszahlen angepasst.

Dieses Angebot ist geeignet für Kunden, die die Dienstleistung von Perfecto Mobile kennenlernen wollen. Ebenfalls interessant ist das Angebot für Kunden, die eine große Anzahl an Geräten und Varianten im Test abdecken wollen und die Geräte nicht dauerhaft benötigen.

- Private MobileCloud

Eine Private MobileCloud wird speziell für einen Kunden von Perfecto Mobile aufgebaut. Die Geräte stehen exklusiv dem Kunden zur Verfügung und der Kunde legt auch fest, welche Geräte bereitgestellt werden sollen.

Dieses Geschäftsmodell ist für Kunden geeignet, die Apps testen, die zum Beispiel in Bezug auf Datenschutz oder Sicherheit sensibel ist und man Bedenken hat, die App vorab auf einem mobilen Endgerät zu installieren, das anschließend von unbekanntem Dritten genutzt wird.

Eine solche Lösung garantiert dem Kunden allerdings auch eine exklusive Bereitstellung der Hardware, was höhere Kosten für die Dienstleistung zur Folge hat.

- Local MobileCloud

Perfecto Mobile lagert die mobilen Endgeräte in den Rechenzentren in einer speziellen Hardware, die bei „Craddle“ genannt wird. Diese Craddles sind eine Entwicklung von Perfecto Mobile.

Entscheidet sich ein Kunde für eine Local MobileCloud, dann werden diese Craddles bei dem Kunden von Perfecto Mobile aufgebaut und betrieben. Dieses sichert dem Kunden ebenfalls einen exklusiven Zugriff auf die mobilen Endgeräte zu und schafft die Sicherheit, dass das Testobjekt innerhalb eines geschlossenen Netzwerkes ohne Nutzung öffentlicher Netze genutzt werden kann.

Bei diesem Ansatz muss der Kunde dafür sorgen, dass er einen geeigneten Platz für die Hardware bereitstellt. Angesichts des direkten Zugriffs auf die physikalische Hardware kann er auch einige Veränderung selbst vornehmen, wie zum Beispiel einen Austausch von mobilen Endgeräten in den Craddles.

- Hybrider Ansatz

Die beschriebenen Ansätze lassen sich laut Perfecto Mobile auch mischen, um eine Testumgebung zu erhalten, die sich genau den Anforderungen des jeweiligen Kunden anpasst.

Hersteller	Modell	Betriebssystem	Version	SIM
Google	Nexus 5	Android	4.4.2	
HTC	One	Android	4.4.2	X
HTC	One X	Android	4.1.1	X
Samsung	Galaxy S5	Android	4.4.4	X
Samsung	Galaxy S5 SM-G900A	Android	4.4.2	
Samsung	SCH-I705 Galaxy Tab 2	Android	4.1.2	X
Samsung	Galaxy S6	Android	5.0.2	X

Tabelle 1 Aufstellung der verfügbaren Endgeräte für den Test

Im Fall von Q-Chess Mobile hat sich der Testmanager mit der Partner Cloud für eine Public Mobile Cloud entschieden, die Perfecto Mobile seinen Partnern bereitstellt.

Die Geräte auf die für den Test in der Partner Cloud Mitte 2015 zur Verfügung standen, sind in Tabelle 1 aufgeführt.

3.4 Perfecto Mobile als Testausführungswerkzeug

Auf die mobilen Endgeräte, die von Perfecto Mobile bereitgestellt werden, kann der Kunde über eine Webapplikation zugreifen. Diese Webapplikation bietet auch die Funktionen, wie sie für ein Testausführungswerkzeug charakteristisch sind:

- Ausführung des Testobjekts unter Verwendung von Skripten
- Protokollierung der durchgeführten Tests
- Aufnahme von manuellen Aktionen, die anschließend abgespielt werden können
- Skriptschnittstelle zur Entwicklung individueller Testabläufe.

4 Erkenntnisse aus dem Einsatz von Perfecto Mobile

4.1 Flexibel in der Wahl der Teststrategie

Die Effizienz eines Werkzeugs ist auch davon abhängig, wie gut das Werkzeug in der Lage ist einen Prozess zu unterstützen.

In diesem Dokument sollen drei Teststrategien behandelt werden, die von Perfecto Mobile gut unterstützt werden.

4.1.1 Manuelles Testen

Ein wesentlicher Bestandteil der Dienstleistung von Perfecto Mobile ist das Bereitstellen von mobilen Endgeräten für Tester. Über eine Webapplikation lassen sich die zur Verfügung gestellten Geräte auswählen, bedienen und auch reservieren.

Ein Gerät, wie man es zum Beispiel in Abbildung 2 sehen kann, lässt sich mit der Maus am PC bedienen. Eine Berührung auf dem Display entspricht einem Klick auf dem Bildschirm. Auch Wischen und andere Gesten lassen sich mit Hilfe der Maus abbilden.

Von dem Gerät selbst ist neben dem sichtbaren Bereich immer auch das Gehäuse als Bild eingeblendet. Die Schaltflächen, die am Gerät selbst angebracht sind, kann der Tester drücken, in dem er auf die entsprechende Taste in der Abbildung des Gehäuses klickt.

Auf der linken Seite sind Funktionen angeordnet, die der Anwender durch Klicken auf den Geräten ausführen kann.



Abbildung 2 Webapplikation von Perfecto Mobile mit Sicht auf zwei Endgeräte

Damit hat der Tester eine gute Umgebung, in der er manuelle Tests durchführen kann. Um die Verfügbarkeit von ganz bestimmten Modellen sicherzustellen, lassen sich die Geräte reservieren. Wenn die Tests frühzeitig geplant werden steht dann auch in der „Public Cloud“ das gewünschte Modell zur Testdurchführung bereit.

Mit Perfecto Mobile kann ein Anwender fast wie gewohnt manuell testen. Der Unterschied zu eigenen Geräten ist, dass er sich um das Bewirtschaften der Geräte, das z.B. das Aufladen des Akkus, das Einspielen von Updates, die Ausgabe an Tester, usw. umfasst, keine Gedanken machen muss. Somit kann sich ein Kunde von Perfecto Mobile auf das Wesentliche, die Benutzung der Geräte, konzentrieren.

Speziell für den Test ist ein Vorteil die Mitschnittfunktion von Perfecto Mobile. Der Tester kann die einzelnen Schritte aufnehmen und den Mitschnittanschließend als Video herunterladen, das z.B. der Entwicklung zur Verfügung gestellt werden kann.

4.1.2 Automatisierte Tests

Die Webapplikation von Perfecto Mobile ist auch ein Testausführungswerkzeug. Der Anwender bekommt eine Entwicklungsumgebung, in der er Skripte in Verbindung mit Geräten entwickeln und verwalten kann.

In der einfachsten Form können Skripte mittels Capture and Replay erstellt werden. Diese Skripte können eine gute Grundlage sein, um das Gerüst für ein Testskript zu bekommen. Bei genauerer Betrachtung zeigt sich häufig, dass die Skripte überarbeitet werden müssen, da in den aufgezeichneten Skripten häufig konstante Koordinaten aufgezeichnet werden.

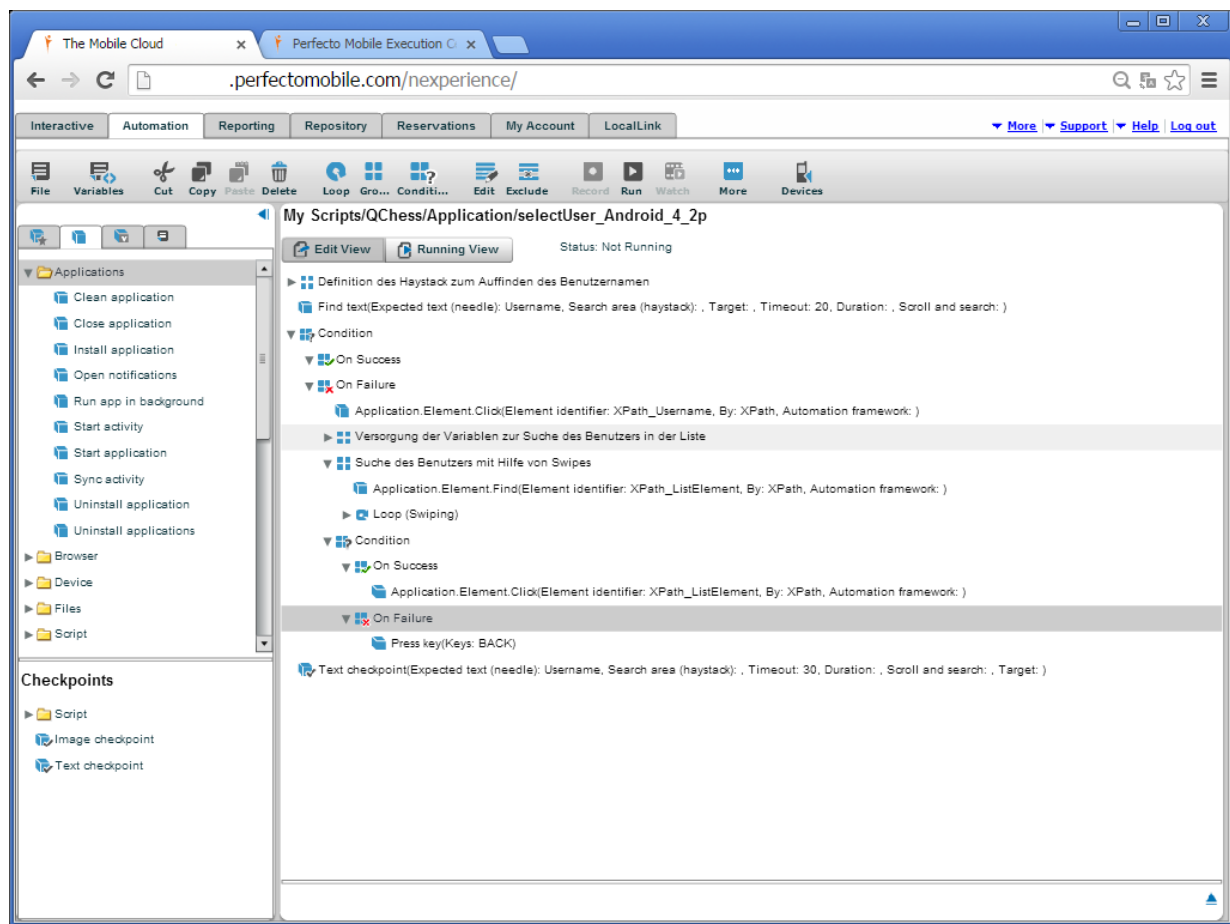


Abbildung 3 Entwicklung von Skripten mit Perfecto Mobile

Durch Capture and Replay entstehen Skripte, die Sequenzen von Aktionen enthalten. In der Testautomatisierung möchte man bestimmte Sequenzen mehrfach wiederholen oder situationsabhängig reagieren. Somit besteht die Anforderung an Kontrollstrukturen, die derartiges erlauben.

Beide Arten von Kontrollstrukturen werden von der Skriptsprache, die Perfecto Mobile bereitstellt, unterstützt. In der oberen Icon-Leiste steht „Loop“ für das Schleifen und „Condition“ für Verzweigungen mit einem Wahr-Zweig („On Success“) und einem Falsch-Zweig („On Failure“).

Der Befehlsvorrat, den Perfecto Mobile zur Verfügung stellt, sind Funktionen, die im linken Bereich in der Abbildung 3 zu erkennen sind. Die Funktionen sind nach fachlichen Aspekten gruppiert und lassen sich auf diese Weise von dem Anwender zügig auffinden.

Parallel zur Entwicklung von Skripten kann der Anwender auch mobile Endgeräte öffnen und die entwickelten Skripte testen.

Damit steht in der Webapplikation eine gute Entwicklungsumgebung zur Verfügung, mit der Testskripte entwickelt werden können.

Die Ausführung von Skripten kann, wie bereits erwähnt, jederzeit ad hoc durch den Anwender veranlasst werden. Während der Ausführung eines Skripts kann der Anwender weder das aktuelle noch ein anderes Skript bearbeiten. Der Arbeitsplatz ist durch die Testausführung in gewisser Weise blockiert. Um eine solche Blockierung zu vermeiden, stellt Perfecto Mobile den „Execution Center“ bereit.

Der Execution Center erlaubt es dem Anwender die Ausführung unterschiedlicher Skripte mit zeitlich zu planen. Ein Skript kann in dem Execution Center entweder direkt gestartet werden oder mit einer Startbedingung versehen werden. Bei diesen Startbedingungen können Zeiten und Wiederholungen hinterlegt werden. Damit muss der Anwender die Durchführung der Tests nicht manuell starten. Während der Testdurchführung kann der Anwender auch an bestehenden Skripten arbeiten, da die Skripte „im Hintergrund“ ablaufen und die Entwicklungsumgebung nicht beeinträchtigen. Die mobilen Endgeräte, auf denen der Test läuft, sind in dem Augenblick natürlich im Zugriff des Execution Center und stehen nicht zur Verfügung.

Die Bereitstellung wird durch den Execution Center nicht gewährleistet. Hier muss der Anwender durch geeignete Strategien selbst tätig werden. Wenn in dem Test bestimmte Geräte verwendet werden sollen, dann kann er diese über das Reservierungsmanagement, das in Kapitel 4.1.1 beschrieben ist, reservieren.

Eine zweite Möglichkeit ist die Verwendung der Funktion „Select Device“. Die Funktion liefert eine zufällige Id aus einer Menge von verfügbaren Geräten, auf die Eigenschaften zutreffen, die der Anwender in Form von Parametern übergeben hat. Es kann z.B. als Betriebssystem „Android“ und als Version „5.*“ übergeben werden. Wenn mindestens ein Gerät dazu passt, dann kann der Rückgabewert als DUT (Device under Test) verwendet werden.

Der Einsatz von Testautomatisierung ist in der Einführungsphase im Vergleich zu manuellen Tests in aller Regel aufwändiger. Aber da anschließend die Tests automatisch und auch ohne

direkte Aktion eines Testers erfolgen können, sind Testwiederholungen mit verhältnismäßig wenig Aufwand möglich.

Eine funktionierende Testautomatisierung ist mit Perfecto Mobile sehr gut umsetzbar, da in der Entwicklungsumgebung in Form einer Webapplikation alle wichtigen Funktionen zur Entwicklung von Skripten vorhanden sind und die Ausführung von Skripten über den Execution Center gesteuert werden kann. Da auch die Bereitstellung der Hardware durch Perfecto Mobile erfolgen kann, muss sich der Anwender wenig Gedanken machen, wie die Hardware mit dem Testausführungswerkzeug zusammengebracht werden muss. Das Reservierungsmanagement unterstützt den Anwender zusätzlich dabei, dass die gewünschte Hardware zum gewünschten Zeitpunkt bereitgestellt werden kann.

Der Nutzen einer Testautomatisierung wird nicht allein durch den Einsatz einer geeigneten Automatisierungslösung erzielt. Es spielen zusätzlich noch Aspekte eine Rolle, die nicht vom Testausführungswerkzeug beeinflusst werden können:

- Die Applikation, die es zu testen gilt, sollte so entwickelt worden sein, dass die GUI-Elemente auf der Oberfläche mit Hilfe guter Merkmale, wie zum Beispiel über Ids, zweifelsfrei durch das Testskript identifiziert und angesprochen werden können.
- Die Wartbarkeit der Testskripte selbst ist ebenfalls ein wesentlicher Schlüssel zum Erfolg in der Testautomatisierung. Es gilt die Skripte möglichst so zu entwickeln, dass sie nicht mit jeder Änderung des Testobjekts neu entwickelt oder überarbeitet werden müssen. Hier kommt es auf geeignete Architektur der Testautomatisierungslösung und Anwendung von Grundsätzen der Softwareentwicklung, wie zum Beispiel Modularisierung, an.

Perfecto Mobile unterstützt die Wartbarkeit von Skripten mit dem Konzept ScriptOnce™, auf das im Kapitel 4.2.1 noch näher eingegangen wird.

4.1.3 Exploratives Testen

In Projekten mit hohem Zeitdruck oder unvollständigen Anforderungen steht dem Testmanager mit dem explorativen Testen eine reaktive Teststrategie zur Verfügung. Die Tester bekommen für die Durchführung eine Testcharta, die keine Testfälle sondern Testziele formuliert. Parallel zur Testdurchführung entwerfen die Tester Ideen, wie sie das Testobjekt testen.

In Kapitel 4.1.1 wurde das Thema manuelles Testen mit Perfecto Mobile bereits thematisiert. Die Aussagen treffen auch auf das explorative Testen zu, da manuelles Testen für diese Teststrategie charakteristisch ist.

Beim explorativen Testen wird die Testdurchführung so protokolliert, dass Testfälle im Nachgang dokumentiert werden können. Im Falle von Zeitdruck ist es von Vorteil, wenn das

Testausführungswerkzeug die Protokollierung übernehmen kann, damit sich der Tester auf die Durchführung konzentrieren kann.

Mit Perfecto Mobile kann der Tester seine Tests aufzeichnen und das Ergebnis ist sowohl ein Video als auch eine Aufstellung aller Aktionen in Form eines Skripts. Die Mitschnitte eignen sich somit sowohl für die Erstellung von manuellen als auch automatisierten Testfällen.

4.2 Anbindung an andere Testwerkzeuge

4.2.1 Entwicklungsumgebung Eclipse

Perfecto Mobile bietet neben dem Automatisierungsframework in der Webapplikation mit dem MobileCloud™ WebDriver eine Schnittstelle, die die Entwicklung von Skripten unter Nutzung von der quelloffenen integrierten Entwicklungsumgebung Eclipse erlaubt.

Mit Selenium 2.0 wurde in der freien Software zum Test von Webapplikationen das WebDriver-Konzept eingeführt. Es handelt sich dabei um eine ganz allgemeine Schnittstelle, die einen grundlegenden Satz von Befehlen definiert, die benötigt werden, um Webapplikationen bedienen zu können. Drittanbieter können diesen Befehlsumfang implementieren und damit dem Testautomatisierer unterschiedliche Techniken bereitstellen.

Das Eclipse-Plugin MobileCloud™ erlaubt den Zugriff auf mobile Endgeräte und das Ausführen von Skripten aus Eclipse heraus und ist damit ein vollwertiger Ersatz für die in diesem Dokument beschriebene Webapplikation.

Sollte in der Testabteilung bereits Erfahrung mit Selenium vorhanden sein, kann man diesen Weg nur empfehlen. Da in der Programmiersprache mehr Kontrollstrukturen zur Verfügung stehen, lassen sich in dieser Umgebung Skripte manchmal effizienter und auch leichter wartbarer entwickeln als das in der Webapplikation möglich ist.

Die Webapplikation eignet sich gerade wegen des überschaubaren Umfangs an Befehlen und Kontrollstrukturen zur Einarbeitung oder für Tester mit weniger Erfahrung im Umgang mit Programmiersprachen.

4.2.2 Performance- und Lasttests mit NeoLoad

Neben den funktionalen Anforderungen an eine Software gibt es weitere Faktoren, die vor allem bei der Akzeptanz einer Software durch den Kunden entscheidend sind. Ein Aspekt in diesem Zusammenhang ist die Performance des Gesamtsystems. In dem hier vorgestellten Beispiel stellt sich die Frage, wie der Q-Chess Sync Server darauf reagiert, wenn gleichzeitig Daten von mehreren mobilen Endgeräten angefordert oder übergeben werden. Diese Antwort lässt sich mit Lasttests beantworten, bei denen dann der Q-Chess Sync Server das Testobjekt ist.

Die Durchführung von derartigen Tests macht in aller Regel auch den Einsatz spezialisierter Werkzeuge, wie zum Beispiel NeoLoad, erforderlich. Eine Stärke von NeoLoad von der Firma Neotys ist das Simulieren von mehreren Anwendern, die gleichzeitig auf den Server zugreifen, und das Überwachen der Parameter, wie CPU-Auslastung, Speicherverbrauch, Datendurchsatz, Übertragungszeiten, usw. Die Simulation der Anwender durch Lastprofile, die zuvor entwickelt werden müssen, basiert meist rein auf dem Absetzen von Requests und dem Empfangen von Daten. Die Antworten werden in aller Regel nicht ausgewertet und auch nicht weiterverarbeitet, da der Schwerpunkt auf der Zeitmessung liegt.

Ein Anwender interessiert sich in Bezug auf das Zeitverhalten für den Zeitpunkt zu dem die Daten dargestellt werden. Dieser Aspekt lässt sich nur prüfen, wenn die empfangenen Daten ausgewertet und entsprechend verarbeitet werden. Dafür werden in aller Regel Testausführungswerkzeuge eingesetzt, die auf Oberflächentests spezialisiert sind, wie das Testautomatisierungsframework von Perfecto Mobile.

Da in aller Regel die Testskripte für den Test der funktionalen Anforderungen entwickelt sind, stellt sich die Frage, ob diese Skripte nicht in den Performance- und Lasttest integriert werden können? Das Werkzeug NeoLoad ermöglicht genau diese Anbindung der Skripte, die unter der Webapplikation von Perfecto Mobile entwickelt wurden. Damit lassen sich Lasttests in NeoLoad entwickeln, die zu einem großen Teil auf Lastprofile zur reinen Übertragung von Daten setzen; und ein Teil der Last wird durch reale mobile Endgeräte erzeugt, die von Perfecto Mobile bereitgestellt werden.

Vitaldaten, wie z.B. CPU-Auslastung des mobilen Endgerätes oder andere Parameter zum Zeitverhalten werden zwischen Perfecto Mobile und NeoLoad über das Data Exchange API von NeoLoad ausgetauscht. Damit stehen dem Anwender zur Auswertung der Lasttests alle Informationen in NeoLoad zur Verfügung.

4.2.3 Continuous Integration mit Jenkins

Der Begriff Continuous Integration wurde von der Softwareentwicklung definiert und meint eine permanente oder kontinuierliche Integration einzelner Komponenten zu einer Anwendung. Werkzeuge, die diesen Prozess unterstützen, übersetzen automatisiert Quellcode und erzeugen daraus zumindest ausführbare Dateien oder Installationsroutinen zur Installation der Anwendung auf geeigneten Zielsystemen. Dieser Prozess soll einerseits manuelle Prozesse automatisieren und zum anderen auch die Qualität der Anwendung steigern, da zu jedem Durchlauf, der auch Build genannt wird, eine Erfolgsaussage gemacht wird. Jenkins ist ein webbasiertes Werkzeug, das quelloffen ist und lizenzkostenfrei genutzt werden kann.

Ein wesentliches Merkmal von Jenkins ist die Erweiterbarkeit über Plug-Ins und eine REST-basierte Programmierschnittstelle. Damit kann Jenkins neben dem reinen Übersetzen von

Quellcode auch andere Aufgaben übernehmen. Eine naheliegende Aufgabe ist die Ausführung von automatisierten Tests, die mit deren Ergebnis dann auch in auf die Erfolgsaussage Einfluss nehmen.

Mit dem „MobileCloud for Jenkins Plugin“ können in Perfecto Mobile hinterlegte Skripte zur Ausführung gebracht werden. Eine zweite wesentliche Funktion ist das Übertragen von Daten von Jenkins zu Perfecto Mobile. Damit lassen sich dann die kompilierten Testobjekte auf den zum Test eingeplanten mobilen Endgeräten installieren und ausführen.

Die Kombination von Jenkins und Perfecto Mobile liefert der Softwareentwicklung stets zeitnahe Rückmeldung über die Folgen einer Änderung im Sourcecode.

Für die Testabteilung ist die Erfolgsaussage, die mit jedem Durchlauf in Jenkins festgehalten wird, ebenfalls eine interessante Information. Wenn der Durchlauf erfolgreich war, dann eignet sich die bereitgestellte Version der Anwendung ganz grundsätzlich zur Ausführung von weiteren Tests.

Ein Fehlschlagen der automatisierten Tests kann ein Hinweis darauf sein, dass es in der Anwendung Änderungen gab, die ein Anpassen der Testfälle zur Folge haben. Diese Information benötigt der Testmanager grundsätzlich bevor Änderungen gemacht werden. Wenn dieses versäumt wurde, kommt die Information über Jenkins unmittelbar nach Umsetzung der Änderung und damit hoffentlich noch vor einem Feature Freeze und einer anschließenden Testphase.

Ganz allgemein steigt der Nutzen der Testautomatisierung, da die Skripte im Idealfall jeden Tag ausgeführt werden und eine Qualitätsaussage liefern. Mit höherem Nutzen lassen sich dann auch die Kosten und Aufwände für die Entwicklung der Testskripte argumentieren.

4.3 Erstellung von wartbaren Skripten

Perfecto Mobile verfolgt die Philosophie ScriptOnce™, was bedeutet, dass Skripte grundsätzlich betriebssystemunabhängig angelegt werden sollten. Die Umsetzung der Philosophie wird sehr deutlich, wenn man sich den Konfigurationsdialog für benutzerdefinierte Funktionen ansieht. In der Abbildung 4 kann man erkennen, dass eine Funktion zum einen durch die Übergabeparameter und zum zweiten durch Angabe eines oder mehrerer Skripte im Bereich „Matching rules“ definiert ist. Der Anwender kann eine Regel erzeugen, die z.B. auf ein bestimmtes Betriebssystem oder auch auf ein ganz bestimmtes Modell zutreffen kann. Bei der Testdurchführung werden die Regeln zeilenweise mit dem aktuellen „Device under Test“ (DUT) abgeglichen und sobald eine Übereinstimmung gefunden wird, wird das Skript aufgerufen, das in der Zeile hinterlegt ist.

Das konkrete Beispiel in Abbildung 4 zeigt eine Funktion für eine Applikation, die derzeit nur für Android entwickelt wurde. Mit Android 4.2.2 ist es möglich, den Object Spy von Perfecto

Mobile zu nutzen, der eine Identifizierung von GUI-Elementen über XPath-Ausdrücke erlaubt. Da diese Methode in aller Regel schneller und zuverlässiger ist, soll sie auch wann immer möglich zur Anwendung kommen.

Die Funktion „answerDialog“ wurde zweimal implementiert. Die eine Version (PRIVATE:QChess\Application\Dialog\clickAnswer\clickAnswer_Android_4_2p.xml) findet die GUI-Elemente über den Objektbaum mit XPath-Ausdrücken und im anderen Skript (PRIVATE:QChess\Application\Dialog\clickAnswer\clickAnswer_Android.xml) wird mittels Texterkennung gearbeitet.

Aufgrund der Tatsache, dass der erste Treffer über die Auswahl des zu startenden Skripts entscheidet, muss die Regel für die speziellen Skripte am Anfang stehen und die allgemeineren Skripte weiter unten.

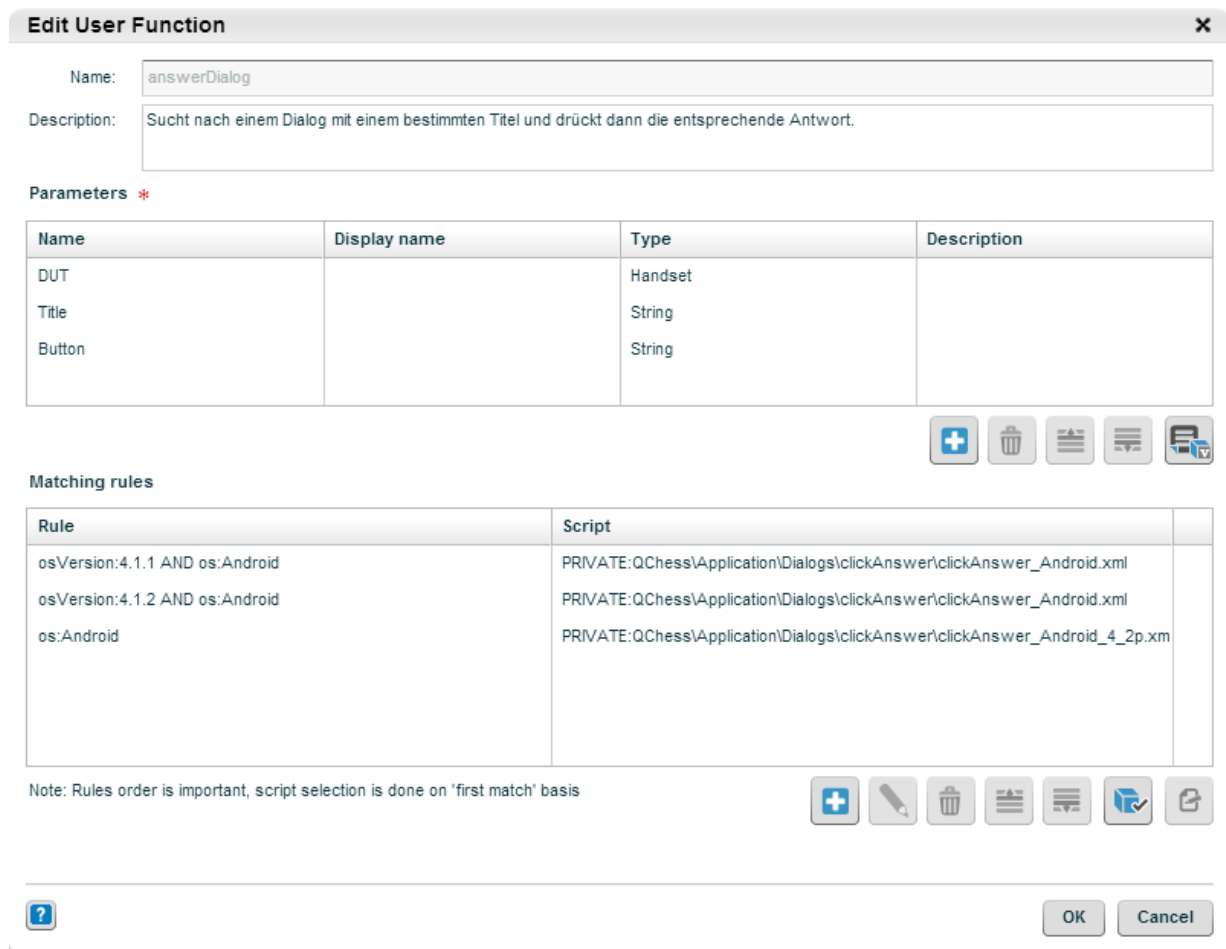


Abbildung 4 Konfigurationsdialog für benutzerdefinierte Funktionen

Die Funktion „answerDialog“ kann jetzt in Skripten verwendet werden und der Nutzer dieser Funktion muss sich keine Gedanken darum machen, ob es bei dem jeweiligen DUT irgendwelche Besonderheiten zu beachten gilt.

Wenn eine Funktion immer in genau einem Skript abgebildet werden würde, dann müssen die unterschiedlichen Varianten im Skript durch Verzweigungen realisiert werden, was zu einer höheren Komplexität der Skripte und damit zu einer schlechteren Wartbarkeit führt. Zusätzlich begünstigt das Aufteilen in unterschiedliche Skripte auch eine arbeitsteilige Entwicklung der Funktionen.

Mit Hilfe des „Execution Center“ bietet Perfecto Mobile die Möglichkeit, Skripte für einen sofortigen Start oder gemäß einem Zeitplan mit Wiederholungen einzuplanen. Damit lassen sich auch regelmäßige Regressionstests ausführen. Die Anzahl der Wiederholungen und die Vollständigkeit der Regressionstests ist in aller Regel ein Argument für Testautomatisierung und dadurch amortisiert sich dann auch der hohe initiale Aufwand, der beim Einstieg in die Testautomatisierung anfällt.