

QF-Test

Evaluation eines GUI-Testtool

Sebastian Kühlewind



Abstract—In dieser Arbeit geht es um Probleme und Lösungsansätze, mit denen sich GUI-Testtools beschäftigen. Dabei werden nicht nur Definitionen und Stichpunkte aufgezählt, sondern auch aktuelle Entwicklungen in der Softwaretechnik aufgezeigt. Diesbezüglich werden auch andere Beispiele herangezogen, um sie mit QF-Test zu vergleichen. Abschließend sollen QF-Tests Merkmale anhand der vorab definierten Kriterien hervorgehoben und mittels eines praktischen Beispiel gezeigt werden.

Index Terms—GUI, Testtool, Software, GUI-Test, Qualitätsmanagement.

DANKSAGUNG

Ich möchte mich bei der Firma Quality First Software für die freundliche und hilfreiche Kommunikation mit den Mitarbeitern und die Bereitstellung einer Lizenz bedanken.

1 EINLEITUNG

Heutzutage hat fast jede Anwendung eine graphische Oberfläche, mit der ein Benutzer mit der Logik interagieren kann. Dies erhöht die Benutzerfreundlichkeit. Kommandozeilen und Terminals werden damit für Otto Normalverbraucher überflüssig, sodass eine größere Reichweite an Benutzern erreicht werden kann. In Zeiten des technischen Fortschritts und der exponentiell steigenden Verwendung von graphischen Oberflächen z.B. im Rahmen von mobilen Anwendungen sollte sichergestellt werden, dass die sogenannte GUI (engl. "Graphical User Interface", graphische Benutzeroberfläche) korrekt funktioniert. Funktioniert die GUI nicht einwandfrei oder im Sinne des Kunden, kann es für einen Softwareproduzenten schnell zu Kunden- und Reputationsverlust kommen. Um dies zu vermeiden, kann man mithilfe unterschiedlicher Methoden und GUI-Testtools die eigene GUI auf Korrektheit hin überprüfen und sicherstellen, dass sie wie gewünscht funktioniert.

1.1 Was sind GUI-Testtools?

Bei GUI-Testtools handelt es sich um Programme, die graphische Oberflächen von anderen Softwares auf Ihre Korrektheit hin überprüft. Dabei wird jede Komponenten der GUI einzeln und mit anderen Komponenten zusammen getestet. Für die Testverfahren wird je nach Testtool eine andere Methode oder ein anderes Vorgehen verwendet. Meistens besitzt das GUI-Testtool eine eigene graphische

Benutzeroberfläche, über die es verwendet werden kann. Für eine bessere Bedienbarkeit ermöglichen viele Testtools das Starten der zu testenden Anwendung, sodass die zu prüfende GUI direkt zum Erstellen der Tests herangezogen werden kann.

1.2 Das Ziel

Eine GUI ist wie in der Einleitung erwähnt eine Benutzerschnittstelle. Sie ermöglicht es einem Normalverbraucher ohne große Programmiererfahrungen oder Kenntnisse über die vorliegende Software mit dem Programm zu interagieren. Dies kann durch allgemein bekannte Zeichen oder Funktionen geschehen. Damit ein Benutzer den Zweck einer Software mithilfe einer GUI ausführen kann, muss diese möglichst einwandfrei funktionieren. Diese Fehlerfreiheit kann durch das Testen mit GUI-Testtools garantiert oder zumindest angenähert werden. Eine Anwendung mit einwandfrei funktionierender GUI macht den Benutzer glücklicher und wirkt sich positiv auf den Ruf und damit die Wettbewerbsfähigkeit einer Firma aus.

2 GRUNDBEGRIFFE DES GUI-TESTEN

2.1 Funktionaler und Nicht-Funktionaler Test

Bei funktionalem Testen überprüft man die implementierten Funktionen auf eine korrekte beziehungsweise gewünschte Funktionalität. Dabei beschränkt man sich lediglich auf die eigene Software und auf die vorhanden Features. Anders als die Funktionalität wird bei nicht funktionalen Tests das Verhalten der Software unter bestimmten Bedingungen getestet. Diese können beispielsweise eine schnelle Folge von Abfragen oder häufige Wiederholungen der gleichen Arbeitsschritte sein. Die GUI wird also auf "stressige" Bedingungen hin untersucht.

2.2 Lasttest

Bei einem Lasttest handelt es sich um einen nicht-funktionalen Test. Er prüft wie das Testobjekt auf einen andauernden Strom von Eingaben oder Anfragen reagiert. Schnelle Eingaben und Bestätigungen können bei einem System zu Fehlern führen, wenn dieses mit der Verarbeitung nicht nach kommt. So können Daten aufgenommen werden, die noch gar nicht überprüft wurden.

2.3 Regressionstest

Ein Regressionstest beschreibt einen Test, der bereits getestete Bereiche überprüft. Er soll sicherstellen, dass eine Anwendung nach Änderungen im Quell-Code noch wie gewünscht und vom vorherigen Test bestätigt, funktioniert. Diese Testart soll das berühmte Beispiel "Ein Fehler behoben, zwei neue erschienen" vermeiden. [1]

3 HERAUSFORDERUNG

In der Software-Entwicklung geht es bei den heutigen gesellschaftlichen und technischen Entwicklungen um Geschwindigkeit und Kundenzufriedenheit. Dies erreicht man durch die Bereiche der Softwaretechnik Qualitäts- und Zeitmanagement. Das allgemeine Ziel ist es hierbei immer schneller und kosteneffizienter zu entwickeln, sodass ein Kunde eher an sein Produkt kommt, die Firma schneller Umsatz macht und der Ruf dieser verbessert wird.

3.1 Qualitäts- und Zeitmanagement

In dem Qualitätsmanagement, o.a. Qualitätssicherung, geht es um die Überprüfung der Korrektheit von Modulen und ihren Beziehungen. Allgemein überprüft man das sogenannte Front- und Backend einer Software. Da es sich bei QF-Test um ein GUI-Testtool handelt, liegt der Schwerpunkt dieser Software auf dem Frontend, sodass die Thematik des Backend hier nicht weiter ausgeführt werden soll. Bei dem Frontend geht es um die Korrektheit der graphischen Benutzeroberfläche. Mittels unterschiedlicher Testkonzepte und stetig laufender Testphasen kann man viele Probleme vermeiden und sich so auf die eigentliche Entwicklung konzentrieren. Für diese verfolgten Funktionen werden Tests geschrieben, die repetitiv die Anwendung auf die gewünschte Funktionalität überprüft. So kann fehlerhaftem Verhalten vorgebeugt werden.

Durch kürzere Entwicklungsphasen und sich unterbietende Entwicklungsfirmen müssen GUI-Testtools kurze Einarbeitungsphasen und schnelle Resultate ermöglichen. Kurze Entwicklungszeiten ermöglichen einen früheren Markteintritt, eine höhere Kundenzufriedenheit und mehr Profit für den Produzenten. Diese Punkte werden aber nur unter der Voraussetzung korrekter Funktionalität gewährleistet. [3]

4 ANFORDERUNGEN

Vorab soll erwähnt werden, dass die hier aufgeführten Anforderungen keine Vollständigkeit erfüllen. Einige der unten aufgeführten Kriterien wurden aus bereits bestehenden Listen mit weitaus mehr Kriterien ausgewählt und übersetzt. [2]

Mit diesem Hinweis sollen nun drei Anforderungsbereiche angeguckt werden. In Bezug auf..

.. allgemeine Anforderungen

Nicht jedes Programm funktioniert auf jedem System und nicht jede Programmiersprache wird unterstützt. In diesem Sinne ist das Werkzeug auf Kompatibilität mit anderen

Komponenten zu prüfen. Welche Betriebssysteme werden unterstützt, welche Programmiersprachen können getestet werden oder in welchem Rahmen sind Erweiterungen möglich? Auch möchte ein Benutzer nicht unkontrollierbar Aktualisierungen des Werkzeugs durchführen müssen, so dass eine Kompatibilität mit älteren Versionen gegeben sein sollte.

Müssen Benutzer eine neue Skript-Sprache lernen, können sie die der Test-Anwendung verwenden oder sind keinerlei Programmiererfahrungen erforderlich, wie es beispielsweise bei einfachen "Capture&Replay"-Funktionen der Fall wäre ("Capture&Replay" bedeutet hier eine Testumgebung, bei der Abläufe durch Anklicken von GUI-Elementen aufgezeichnet und anschließend wiedergegeben werden können).

Innerhalb der Softwaretechnik wird zwischen zwei Bereichen von Testarten unterschieden: Funktionale und nicht-funktionale Tests. In welchem Ausmaß unterstützt das GUI-Testtool nun die diese Testarten? Um die Möglichkeiten der Testreichweite aus zu reizen sollte das Werkzeug im Idealfall beide Kategorien unterstützen.

Wurde nun mit einem Tool ein Testablauf erstellt, steht als nächstes die Ausführung an. Um bei Fehlern oder großen Projekten nicht rätselnd warten zu müssen, welcher Fortschritt bereits erreicht wurde, sollte der aktuelle Stand der Ausführung leicht nachvollziehbar dargestellt sein.

Als abschließendes Kriterium für diesen Abschnitt soll noch die Dokumentation genannt werden. Testfälle sollen nicht nur erstellt und ausgeführt werden, sondern auch dokumentiert. Die Dokumentation soll beispielsweise für Kundengespräche oder Planungen in der Entwicklungsphase verwendet werden. Stellt das Testtool solch eine Funktionalität zur Verfügung?

.. den Tester

Mitarbeiter der Qualitätssicherung müssen effizient und effektiv Fehler finden und Testen, daher sind Kriterien für diesen Bereich zu beachten. Ein GUI-Testtool sollte eine möglichst ideale Lernkurve haben, damit eine kurze Einarbeitungszeit gewährleistet werden kann. Dafür sollte ein Tool beispielsweise übersichtlich, konstant gestaltet sein und über eine ausführliche Dokumentation über Funktionen verfügen.

Häufige De- und Installationen, z.B. bei Mitarbeiter eigenen Computern, sollten ohne großen Aufwand durchführbar sein. Ein komplizierter De-/Installationsvorgang kann zu verringerter Benutzerzufriedenheit führen.

Für die Wartung und Verständlichkeit sollten Testfortschritte eindeutig dargestellt werden, sodass beispielsweise Fehler im Detail lokalisierbar sind. Eine lange Ursprungssuche des Fehlers hilft dem Tester nicht, seiner Arbeit nachzukommen. Und ist der Auslöser ein Test, sollte dieser auch ohne großen Aufwand änderbar sein.

Benutzereingaben sollten durch das GUI-Testtool selbst übernommen oder simuliert werden, damit eine kontinuierliche Anwesenheit nicht nötig ist.

Diese Kriterien kommen vor allem dem Tester zugute. Sie sollen zu einem reibungslosen Testverlauf führen

.. die Firma

Für die Firma sollen nur kurz zwei ausschlaggebende Punkte erwähnt werden: Preis und Updates. Wie viel kostet das GUI-Testtool und welche Finanzierungspläne sind vorgesehen. Damit das Werkzeug nach Erwerb noch lange für den Kunden von Nutzen sein kann, sollten Updates bereitgestellt werden. Neue Eigenschaften eines Betriebssystem oder technischen Neuerungen können so nach Kauf noch ergänzt werden.

5 QF-TEST

QF-Test wird im Bereich der Qualitätssicherung verwendet. Es wurde 2001 von Gregor Schmidt mit der Absicht hochwertiger Software herzustellen begonnen. Seither arbeitet die Firma QFS, Quality First Software, an der Software.

5.1 Technische Daten

Es unterstützt die Betriebssysteme Windows, Linux, Mac OS und Unix. Damit sind ein Großteil der gängigsten Computersysteme verfügbar. Beispielsweise Chrome OS, ein Betriebssystem für beliebte Studenten- und Akademikercomputer, wird nicht unterstützt. Auf diesen Systemen kann dann QF-Test für Anwendungen, die in Java oder im Web-Bereich geschrieben sind, verwendet werden. Auch sind sogenannte hybride Systeme unterstützt (e.g. JavaFX WebView, JXBrowser, SWT-Browser). Damit ist es browserübergreifend (Chrome, Firefox, Microsoft Edge, Safari etc.). Die Unterstützung der gewünschten Programmiersprache ist ausschlaggebend für die Auswahl des Testwerkzeuges. QFS arbeitet zur Zeit an einer Möglichkeit, auch mobile Anwendungen mit QF-Test zu testen. Mit diesem Bereich wäre ein exponentiell wachsender Bereich abgedeckt. [4]

5.2 Oberfläche und Struktur

In Abschnitt 4 bei den Anforderungen für den Tester wurde ein möglichst effizientes und effektives GUI-Verhalten zusammengefasst. Das Testtool sollte über ein intuitives Design verfügen. Diesbezüglich besitzt QF-Test eine intuitive Benutzeroberfläche mit Baumstrukturen und modularisierten Bereichen. Für Tester ohne Programmiererfahrungen sind Funktionen wie das bereits erwähnte "Capture&Replay" integriert. Erfahrenere Benutzer können mithilfe von den integrierten Skriptingsprachen Jython und Groovy mehr Flexibilität und Testkomplexität realisieren. Dadurch können Tester direkt auf die Java API zugreifen. [4]

5.3 Testarten

QFS Philosophie beim Testen liegt bei den Regressions- und Lasttests. Wie in Kapitel 2 eingeleitet, beschäftigen sie sich mit dem nicht funktionalen Testen. QF-Test unterstützt dies beispielsweise durch beliebig oft und schnell ausführbare Testfälle. In der Anforderung Dokumentation bietet das GUI-Testtool die Möglichkeit Testberichte zu erstellen, in denen Einsichten über aktuelle Testfälle, Ergebnisse des aktuellen Fortschritts und Fehlschläge sind. Die Dokumentation kann auf Knopfdruck oder mithilfe von Kommandozeilen generiert werden. [4]

5.4 Erwerb einer Lizenz

Es können eine Entwicklerlizenz (1.995Euro) oder eine Runtime-Lizenz (995Euro) gekauft werden. Zusätzlich können für weitere Technologien (Swing, AWT, JavaFX etc.) sogenannte GUI-Technologie-Erweiterungen (je 500Euro) gekauft werden. Ergänzend wird noch ein Pflegevertrag angeboten (480Euro), in dem alle Upgrades und eine weitere GUI-Technologie angeboten wird. [5]

6 STAND DER SOFTWARETECHNIK

Aktuell befinden sich unzählige Programme zum GUI-Testen auf dem Markt. Von kostenlos und open-source zu monatlichen Preisen und geschlossen findet sich vieles. [2] Die Schwierigkeit besteht in der Auswertung der eigenen Anforderungen an das Testtool.

Einige häufig auftretende Testtools sind Unified Functional Testing, Selenium, Ranorex und Protractor. Es sollen kurz UFT und Selenium aufgegriffen werden. Unified Functional Testing besitzt eine ähnliche Oberflächenstruktur wie QF-Test und unterstützt weitaus mehr Programmiersprachen (Oracle, Microsoft .NET, Java, SOA, Rest). [6] Auf der anderen Seite befindet sich UFT in höheren Preisklassen. So kostet eine Laufzeitlizenz beispielsweise 600Euro für drei Monate. [7]

Anders als UFT und QF-Test ist Selenium open source und kostenlos. Es finanziert sich durch Sponsoren und Spender aus der eigenen Community. [8] Wie QF-Test unterstützt es alle Betriebssysteme. Gegenüber QF-Test befindet es sich aber auch in einigen Nachteilen: Zwar unterstützt es die Sprachen C++, Ruby, PHP, dafür aber kein Java. Folgend ist es nur für reine Entwickler geschrieben. Programmiererfahrung ist also erforderlich. Auch ist es schwieriger zu installieren. Tests zu bearbeiten ist umständlicher und Selenium verfügt über kein intuitives Design. [9]

Das Marktangebot ist vielfältig. QF-Test fällt durch die starke Spezialisierung und das verhältnismäßig gute Preis/Leistungs-Verhältnis auf.

7 CAR-CONFIGURATOR

QFS stellt für Demonstrationszwecke einen sogenannten "Car-Configurator" zur Verfügung. Dabei handelt es sich um eine Java-Anwendung mit Hintergrundlogik und einer graphischen Benutzeroberfläche.

The screenshot shows a window titled "JCarConfigurator - Version 1". It has a menu bar with "Datei", "Einstellungen", "Bestellung", and "Hilfe". Below the menu is a tree view with "Fahrzeuge", "Sondermodelle", and "Zubehör". The "Fahrzeuge" section is expanded, showing a table of car models and prices:

Modell	Preis
Rolo	12.300,00 €
Minigolf	15.000,00 €
Rasant	17.000,00 €
Rasant Family	18.500,00 €
I5	29.000,00 €

Below the table, there are several price-related fields:

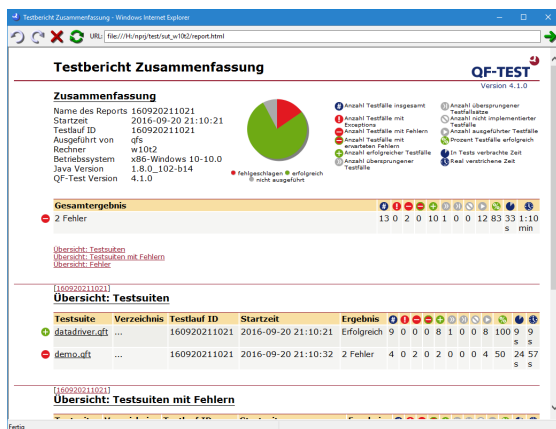
Preis Basismodell	0,00 €
Preis Sonderausstattung	0,00 €
Rabatt	0%
Endpreis	0,00 € -5%

Wie in Abbildung 1 gezeigt, verfügt das Beispiel über unterschiedliche GUI-Komponenten: Labels, Buttons, eine Menüleiste, TabPanes und Textfields. Unter Datei lässt sich ein Fehler einbauen, der später im Lösungsansatz aufgegriffen werden soll, um das Verhalten von Tests bei Fehlverhalten zu beschreiben. Dieses Beispiel zeigt, dass QF-Test eine benutzerfreundliche Möglichkeit bietet, die Ziel-Anwendung in QF-Test zu emulieren und direkt auswählbar zu machen. Somit können die gewünschten Abläufe direkt mittels Capture&Replay in der eigenen Anwendung simuliert werden.

8 QF-TEST LÖSUNGSANSATZ

Wie bereits erwähnt wurde ist eines der wichtigsten Features von QF-Test das Capture&Replay-Feature. Es ermöglicht schnelles und intuitives Testen der Anwendung durch einfaches Anklicken der jeweiligen Komponenten. Es ist weniger zeitaufwendig und benötigt keine Programmiererfahrung. Ob Java-AWT, -Swing oder -Fx, QF-Test ermöglicht eine solide Abdeckung dieser Programmiersprachen. Webanwendungen können auch über einen Assistenten importiert werden. Die Skriptingsprachen Jython und Groovy können genutzt werden um weitere Methoden in QF-Test zu deklarieren, die beispielsweise initialisieren, die GUI mit Daten füllen oder kleinere Überprüfungen durchführen. Auch können Sequenzen als unterschiedliche Module geschrieben werden: QF-Test erbt einige Aspekte der Programmierung von Java. So können Schleifen und Wenn-Bedingungen formuliert werden, die auf der GUI ausgeführt werden sollen. [12]

Nachdem die GUI also gestartet und eingebunden wurde, kann mithilfe eines Aufnahme-Buttons eine Sequenz gestartet werden. Es wird eine Klick-Abfolge in der Test-GUI vorgenommen, wie sie beispielsweise vom Benutzer erwartet wird oder die auf Fehler hin untersucht werden soll. In der Sequenz können Punkte markiert werden, die das Tool testen soll. Abschließend ist die Sequenz fertig. [13] Nach dem Ausführen eines Tests, welches durch Rechtsklick auf die Testsuite und "Wiedergabe starten..." erfolgt, kann Dokumentation generiert werden. Der Test läuft je nach Größe der Testsuite schnell durch.



Quelle: https://www.qfs.de/qf-test-handbuch/lc/manual-de-user_report.html

Diese Dokumentation kann für Besprechungen und den

Austausch mit dem Kunden, aber auch zur Evaluierung des eigenen Fortschritts verwendet werden. Da sie in HTML-Format generiert wird ist sie leicht in Webseiten einzubetten.

9 EVALUATION

Ziel war es, die graphische Oberfläche einer Anwendung effektiv und effizient zu testen. QF-Test ist für alle gängigen Betriebssysteme verfügbar, mit und ohne Programmiererfahrung benutzbar, integrier- und erweiterbar und durch Dokumentation/Support unterstützt. Als Flaschenhals besitzt QF-Test einmal die Programmiersprache und zum anderen den Preis. Nicht jede Anwendung mit einer graphischen Oberfläche ist in Java geschrieben und nicht jede Firma besitzt das Geld, um sich ein kostenpflichtiges Testwerkzeug zu finanzieren. Allerdings, stellt Dominik Mohilo heraus, sei Java neben C++ und Python einer der beliebtesten Programmiersprachen auf dem Markt. [14] Julia Schmidt bestätigt die Führung und den weiteren Aufstieg der Sprache Java. So sei Java weiterhin eine Sprache mit wachsender Gemeinschaft und Anwendung in der Praxis. [15] Mobile Anwendungen beispielsweise werden auch mit der Unterstützung der Programmiersprache Java geschrieben. QFS entwickelt derweilen auch eine Lösung, zum Testen von mobilen Anwendungen. Ist die Finanzierung kein Aspekt, der zu einem anderem Tool führen würde, und befindet sich die Anwendung im Bereich Java und/oder Web, so ist QF-Test die perfekte Lösung mit einem großen Angebot an Features, Dokumentation und Unterstützung.

REFERENCES

- [1] Ralf Nörenberg, *Effizienter Regressionstest von E/E-Systemen nach ISO 26262*, 2012, Seite: 62
- [2] Unbekannt, *Software Test Automation Tool Evaluation*, http://sqa.fyicenter.com/FAQ/Testing-Tools/Software_Test_Automation_Tool_Evaluation_Criteri.html, Stand: 15.07.2017
- [3] Georg Herzwurm, *Kundenorientierte Softwareentwicklung*, Springer Verlag, 2013, Seite: 62ff.
- [4] QFS, *GUI-Testautomatisierung für Java und Web*, https://www.qfs.de/qf-test/testautomatisierung-mit-qf-test.html?pk_campaign=db&pk_kwd=qftest, Stand: 15.07.2017
- [5] QFS, *Preise für Lizenzen und Pflegevertrag*, <https://www.qfs.de/qf-test/preise.html>, Stand: 15.07.2017
- [6] Unified Functional Testing, *Features*, <https://saas.hpe.com/en-us/software/uft/features>, Stand: 15.07.2017
- [7] Unified Functional Testing, *Prices*, <https://saas.hpe.com/en-us/buy/uft>, Stand: 15.07.2017
- [8] Selenium, *Selenium HQ - What is Selenium?*, <http://www.seleniumhq.org>, Stand: 15.07.2017
- [9] Selenium, *Selenium Documentation*, <http://www.seleniumhq.org/docs/>, Stand: 15.07.2017
- [10] Ranorex, *Ranorex - How it works*, <https://www.ranorex.com/how-test-automation-works.html>, Stand: 15.07.2017
- [11] Ranorex, *Ranorex - Pricing & Licensing*, <https://www.ranorex.com/pricing.html>, Stand: 15.07.2017
- [12] QF-Test, *QF-Test Handbuch*, <https://www.qfs.de/qf-test-handbuch/lc/manual-de-control.html>, Stand: 15.07.2017
- [13] QF-Test, *QF-Test Handbuch*, https://www.qfs.de/qf-test-tutorial/lc/manual-en-user_capture.html, Stand: 15.07.2017
- [14] Dominik Mohilo, *Das sind die populärsten Programmiersprachen 2016*, <https://jaxenter.de/programmiersprachen-rankings-49399>, Stand: 15.07.2017
- [15] Julia Schmidt, *Programmiersprachen Ranking: Februar-Ausgabe des FIOBE-Index legt Kurs auf Java-Entwicklung*, <https://www.heise.de/developer/meldung/Programmiersprachen-Ranking-Februar-Ausgabe-des-FIOBE-Index-legt-Fokus-auf-Java-Entwicklung-3091879.html>, Stand: 15.07.2017