

**Hochschule Wismar**

University of Applied Sciences, Technology,  
Business and Design

Fakultät für Ingenieurwissenschaften  
Bereich Elektrotechnik und Informatik

---



# Master-Thesis

Untersuchungen zu automatisierbaren Testszenarios für User  
Interfaces von kaufmännischen Softwaresystemen

Eingereicht am: 12. August 2011

von: Christian Froh  
geboren am 03. März 1986  
in Hagenow

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>5</b>
1.1	Analyse der Aufgabenstellung und Anforderungen . . . . .	6
1.2	Unternehmen SIV.AG . . . . .	7
<b>2</b>	<b>Grundlagen</b>	<b>8</b>
2.1	Agile Softwareentwicklung . . . . .	8
2.2	Automatisierung von Softwaretests . . . . .	9
2.2.1	Hintergründe zum Testen von Software . . . . .	10
2.2.2	Vor- und Nachteile der Testautomatisierung . . . . .	10
2.2.3	Testverfahren . . . . .	13
2.2.4	Testebenen . . . . .	14
2.2.5	Testmethoden . . . . .	16
2.3	Automatisierung von Tests grafischer Benutzeroberflächen . . .	17
2.3.1	Simulation von Benutzereingaben . . . . .	18
2.3.2	Skript-Techniken . . . . .	19
<b>3</b>	<b>Aufwandsanalyse</b>	<b>22</b>
3.1	Fallstudie: Manuelles und automatisiertes Testen im Vergleich .	22
3.2	Fallstudie: Bestimmung des Wertes einer Testautomatisierung .	23
3.3	Einschätzung der Kosteneinsparungen bezüglich der SIV.AG . .	25
<b>4</b>	<b>Aufbau und Architektur der Software kVASy</b>	<b>28</b>
4.1	Ist-Stand: kVASy4 - Entwicklung . . . . .	29
4.1.1	PL/SQL . . . . .	29
4.1.2	Oracle Forms . . . . .	29
4.1.3	Oracle Reports . . . . .	30
4.1.4	Testmethodik . . . . .	30
4.2	Soll-Stand: kVASy5 - Entwicklung . . . . .	31
4.2.1	Eclipse Plattform . . . . .	31
4.2.2	kVASyClipse . . . . .	32
4.2.3	Xtext . . . . .	33
4.2.4	Riena . . . . .	33
4.2.5	Testmethodik . . . . .	34
<b>5</b>	<b>Evaluierung der Testwerkzeuge</b>	<b>35</b>
5.1	Analyse der Anforderungen . . . . .	35
5.2	Bewertung und Auswahl von Testwerkzeugen . . . . .	36
5.2.1	Automated GUI Recorder . . . . .	36
5.2.2	GUIDancer . . . . .	37

5.2.3	Unified Functional Testing . . . . .	38
5.2.4	QF-Test . . . . .	39
5.2.5	Squish . . . . .	41
5.2.6	SWTBot . . . . .	42
5.2.7	Bewertungsübersicht der Testwerkzeuge . . . . .	43
5.3	QF-Test vs GUIDancer . . . . .	44
<b>6</b>	<b>Einführung in das Testwerkzeug QF-Test</b>	<b>46</b>
6.1	Die Testsuite . . . . .	46
6.2	Verbindung mit SUT . . . . .	47
6.3	Erstellung eines Testfalls . . . . .	49
6.4	Implementierung eines Checks . . . . .	50
<b>7</b>	<b>Entwicklung des Prototyps</b>	<b>53</b>
7.1	Lösungsvarianten zur Entwicklung der Tests . . . . .	53
7.2	Entwurf der ausgewählten Lösungsvariante . . . . .	54
7.2.1	Testanalyse . . . . .	54
7.2.2	Design des Testmodells . . . . .	55
7.2.3	Design des QF-Test Testsuitemodells . . . . .	58
7.3	Implementierung der ausgewählten Lösung . . . . .	59
7.3.1	Implementierung der Testmethoden in QF-Test . . . . .	60
7.3.2	Ausführung der Tests . . . . .	62
7.3.3	Reports und Testdokumentationen . . . . .	62
7.3.4	Implementierung von Abhängigkeiten . . . . .	64
7.4	Nachweis und Bewertung der Funktionalität . . . . .	65
<b>8</b>	<b>Auswertung und Ausblick</b>	<b>66</b>
	<b>Literaturverzeichnis</b>	<b>67</b>
	<b>Abbildungsverzeichnis</b>	<b>69</b>
	<b>Tabellenverzeichnis</b>	<b>71</b>
	<b>Abkürzungsverzeichnis</b>	<b>72</b>
<b>A</b>	<b>kVASy-Produktportfolio</b>	<b>73</b>
<b>B</b>	<b>Bewertung QF-Test und GUIDancer</b>	<b>74</b>
<b>C</b>	<b>Entwicklung des Prototyps</b>	<b>77</b>
	<b>Selbstständigkeitserklärung</b>	<b>81</b>
	<b>Thesen</b>	<b>82</b>

## 5 Evaluierung der Testwerkzeuge

Es sollte zunächst ein geeignetes Testwerkzeug zur Durchführung von GUI-Tests ausgewählt werden. In diesem Zusammenhang werden im Folgenden die Anforderungen analysiert, um dann hinsichtlich der Testwerkzeuge eine Bewertung vorzunehmen. Nachdem ein geeignetes Testwerkzeug gewählt wurde, soll es bei der SIV.AG eingeführt werden. Dazu sind entsprechende Tests, die im kommenden Kapitel ausführlich erläutert werden, zu erstellen, auszuwerten und zu dokumentieren.

### 5.1 Analyse der Anforderungen

Bevor ein Qualitätssicherungsteam die Anschaffung eines Testwerkzeugs in Erwägung zieht, sollte es sicherstellen, dass gewisse Rahmenbedingungen erfüllt sind, um eine Fehlinvestition zu vermeiden. Da die kVASy5-Anwendung eine Riena-Applikation ist und sich die grafische Oberfläche aus SWT-Elementen zusammensetzt, sind die Hauptkriterien die Unterstützung für Eclipse RCP und SWT. Diese Anforderungen sind für das Testwerkzeug zwingend erforderlich, da es das Fundament des neuen kVASy5-Systems ist.

Des Weiteren sollte eine intuitive Skriptaufzeichnung vorhanden sein, um die Tests später in aufgezeichneter oder modifizierter Form aufrufen zu können. Das Hauptproblem hierbei ist die Wiederfindung der Oberflächenkomponenten in unabhängigen Durchläufen. Selbst unter identischen Umständen gibt es keine Garantie, dass der Objektbaum, der die GUI repräsentiert, in den nachfolgenden Durchläufen identisch zum aufgezeichneten Zustand ist.

Damit Fehler entdeckt und beseitigt werden können, sollte das Testwerkzeug Auswertungen ausgeführter Tests in Form von Reports bereitstellen. Nicht zwingend erforderlich aber für eventuelle Lasttests sollte das Ausführen der

Tests im Batch<sup>1</sup>-Modus möglich sein. Ebenfalls nicht zwingend notwendig aber wünschenswert wäre das automatisierte Erstellen einer Testdokumentation.

## 5.2 Bewertung und Auswahl von Testwerkzeugen

Insgesamt wurden 33 Testwerkzeuge bezüglich der Hauptkriterien untersucht. In die engere Auswahl haben es letztendlich sechs Testwerkzeuge geschafft, die dann ausführlich auf die geforderten Anforderungen untersucht wurden.

### 5.2.1 Automated GUI Recorder

Der AGR (Automated GUI Recorder) ist ein Open-Source Testwerkzeug, das im Rahmen der TPTP (Test & Performance Tools Platform) entwickelt wird. Im Grunde genommen ist der AGR ein Eclipse Plugin und ist somit kein eigenständiges Programm.

#### Features

AGR unterstützt eine automatische Skriptaufzeichnung, wobei diese auf „Position Based Recording“ basiert. Dabei werden die Aktionen nicht auf die Komponente bezogen aufgezeichnet, sondern auf dem Pixel genau. Der Nachteil ist, dass nicht die Koordinaten innerhalb des Fensters ausschlaggebend sind, sondern die Bildschirmkoordinaten. Schon das Starten der Anwendung an einer anderen Position bzw. das Verändern der Auflösung bereitet schon einige Probleme beim Ablaufen der Tests.

Außerdem besitzt AGR keinerlei Analyse und Auswertungswerkzeuge. Sowohl das Ausführen der Tests im Batch-Modus als auch eine automatische Erstellung einer Testdokumentation ist nicht möglich.

#### Lizenz

Der AGR steht unter der EPL (Eclipse Public License) und gilt somit als freie Software. Die Nutzung der Software ist kostenlos und der Quelltext kann aus dem CVS-Repository des TPTP Projektes bezogen werden.

---

<sup>1</sup>**Batch-Modus** In diesem Modus wird ein Programm auf Kommandozeilenebene gestartet.

## **Fazit**

Der Automated GUI Recorder unterstützt zwar die Hauptkriterien, jedoch aufgrund der fehlenden Features bezüglich der Anforderungen kommt dieses Testwerkzeug als Kandidat nicht in Frage.

### **5.2.2 GUIDancer**

GUIDancer ist ebenfalls ein auf Eclipse basierendes Testwerkzeug der Braunschweiger Firma BreDEX GmbH. Es lässt sich als Standalone-Anwendung oder als Eclipse-Plugin ausführen. Zusammen mit der Version 5.0 wurde im Frühjahr 2011 unter dem Namen Jubula Kernteile der Software als Open Source veröffentlicht. GUIDancer mit der kompletten Funktionalität bleibt dagegen proprietär.

## **Features**

GUIDancer unterscheidet sich von anderen Testwerkzeugen insbesondere dadurch, dass nach der Aufstellung der Anforderungen mit der Testautomatisierung begonnen werden kann. Entwicklung und Integrationstests laufen synchron während des gesamten Projektverlaufs. Diese große Parallelität eignet sich besonders gut für agile Entwicklungsprozesse. Um dies bewerkstelligen zu können, muss allerdings auf das Capture-Verfahren verzichtet werden. Es werden dahingehend keine Skripte aufgezeichnet, sondern lediglich drei Komponenten vom Benutzer zusammengestellt. Es handelt sich dabei um die GUI-Komponente, Aktion und Parameter, die dann einen Testschritt repräsentieren.

Bezüglich der Anforderungen ist zumindest das proprietäre Testwerkzeug GUIDancer mit weiteren Funktionalitäten wie das Erstellen von Reports, Testdokumentationen und das Ausführen der Tests im Batch-Modus ausgestattet. Jubula unterstützt dagegen nur die Kernfunktionalitäten wie die Testfallerstellung und die Testausführung.

## Lizenz

Durch die Einführung der Open-Source Version Jubula haben sich während der Bearbeitung der Master-Thesis die Lizenzen stark verändert. Im August 2011 sind folgende Lizenzen zu erwerben:

Produkt	Beschreibung	Preis Brutto
GUIDancer 5.1	Benutzerlizenz, Updates für ein Jahr inklusive	238.00 EUR
Support 10	10 Stunden, 6 Monate	1142.40 EUR
Support 25	25 Stunden, 12 Monate	2856.00 EUR

**Tabelle 5.1:** GUIDancer Lizenzen, Quelle: [16]

## Fazit

GUIDancer erfüllt alle Bedingungen und kommt somit als potentieller Kandidat in Frage.

### 5.2.3 Unified Functional Testing

UFT (Unified Functional Testing) ist ein eigenständiges, proprietärs Testwerkzeug der Firma Hewlett-Packard. Es ist bereits seit 2001 auf dem Markt und aktuell in der Version 11.0 zu erwerben.

## Features

UFT besticht mit seiner extremen Vielfalt an den zu unterstützenden Technologien. Im Gegensatz zu allen anderen Testwerkzeugen unterstützt UFT Technologien wie Web, Java SWT, Eclipse RCP, .Net, WPF, Silverlight, Delphi, Flex, Web Services und Windows Mobile.

Testskripte können schnell und leicht über eine schlüsselwortgetriebene Technologie gepflegt werden. Auch die zuverlässige Testausführung ist mit Hilfe einer speziellen Objektidentifizierungsmethode gewährleistet. Allerdings stellt UFT keine Werkzeuge zur Analyse- und Auswertung sowie für das Ausführen der Tests im Batch-Modus bereit.

## Lizenz

Leider gibt es keinerlei Informationen zu den Lizenzen. Alle Lizenzen bezüglich des Testwerkzeugs UFT sind nur auf Anfrage<sup>2</sup> erhältlich.

## Fazit

Das von Hewlett-Packard entwickelte Werkzeug hat sehr gute Ansätze sowohl in der Testfallerstellung als auch in der Testfallausführung. Durch die eigens entwickelte Objektidentifizierungsmethode ist die Testausführung, auch bei Änderungen an der Software, sehr zuverlässig. Jedoch kommt UFT nicht in Frage, da die anderen gestellten Anforderungen nicht erfüllt werden.

### 5.2.4 QF-Test

QF-Test ist ein Softwareprojekt des Unternehmens Quality First Software GmbH (QFS) und ist wie UFT seit 2001 auf dem Markt. Es ist ein eigenständiges, proprietäres Programm, das aktuell in der Version 3.4.0 verfügbar ist.

## Features

QF-Test legt besonders viel Wert auf Einfachheit, Robustheit und Vielfältigkeit. Über die grafische Oberfläche können Testskripte schnell, einfach und intuitiv aufgenommen werden. Programmierkenntnisse sind in QF-Test nicht zwingend erforderlich. Für komplexere Tests kann auf die Skriptsprachen Jython oder Groovy zurückgegriffen werden.

Auch QF-Test enthält einen eigens entwickelten Wiedererkennungsmechanismus, der sehr robust gegenüber der zu testenden Applikation ist. Weiterhin ist QF-Test mit allen möglichen Features, wie Reporting, Erstellen von Testdokumentationen und Ausführen der Tests im Batch-Modus ausgestattet.

---

<sup>2</sup>Eine für diese Master-Thesis gestellte Anfrage wurde von HP nicht beantwortet

## Lizenz

QF-Test gibt es in unterschiedlichen Lizenztypen und -modellen. Die folgenden Abbildungen enthalten die Preise für Entwicklerlizenzen und Runtime-Lizenzen für Version 3.4 von QF-Test.

Preise für Entwicklerlizenzen (in EUR)				
Anzahl	Zeitlich unbefristete Lizenz		Optionaler Pflegevertrag (jährlicher Preis)	Gesamtpreis 5% Rabatt und 1. Jahr Pflegevertrag
	Einzelpreis	Gesamtpreis		
1 Lizenz	1.595,00	1.595,00	435	1.950,25
2 Lizenzen	1.495,00	2.990,00	740	3.580,50
3 Lizenzen	1.495,00	4.485,00	1.025,00	5.285,75
4 Lizenzen	1.495,00	5.980,00	1.285,00	6.966,00
5 Lizenzen	1.495,00	7.475,00	1.535,00	8.636,25
6 Lizenzen	1.395,00	8.370,00	1.780,00	9.731,50
7 Lizenzen	1.395,00	9.765,00	2.015,00	11.291,75
8 Lizenzen	1.395,00	11.160,00	2.250,00	12.852,00
9 Lizenzen	1.395,00	12.555,00	2.475,00	14.402,25
10 Lizenzen	1.395,00	13.950,00	2.700,00	15.952,50
Mehr als 10 Lizenzen	Individuelles Angebot			

Tabelle 5.2: QF-Test Entwicklerlizenzen

Preise für Runtime-Lizenzen (in EUR)				
Anzahl	Zeitlich unbefristete Lizenz		Optionaler Pflegevertrag (jährlicher Preis)	Gesamtpreis 5% Rabatt und 1. Jahr Pflegevertrag
	Einzelpreis	Gesamtpreis		
1 Lizenz	1.036,75	1.036,75	155,5	1.140,40
2 Lizenzen	964,00	1.928,00	289,2	2.120,80
3 Lizenzen	964,00	2.892,00	433,80	3.181,20
4 Lizenzen	964,00	3.856,00	578,40	4.241,60
5 Lizenzen	964,00	4.820,00	723,00	5.302,00
6 Lizenzen	900,00	5.400,00	810,00	5.940,00
7 Lizenzen	900,00	6.300,00	945,00	6.930,00
8 Lizenzen	900,00	7.200,00	1.080,00	7.920,00
9 Lizenzen	900,00	8.100,00	1.215,00	8.910,00
10 Lizenzen	843,00	8.435,00	1.265,25	9.278,50
20 Lizenzen	568,75	11.375,00	1.706,25	12.512,50
50 Lizenzen	337,50	16.875,00	2.531,25	18.562,50
Sonstige	Individuelles Angebot			

Tabelle 5.3: QF-Test Runtime-Lizenzen

## Fazit

QF-Test erfüllt, wie auch schon der GUIDancer, alle Anforderungen und ist somit ein weiterer möglicher Kandidat.

### 5.2.5 Squish

Das im Jahr 2003 gegründete Unternehmen froglogic GmbH ist auf GUI-Testen spezialisiert und hat das Testwerkzeug Squish in seinen Reihen. Squish ist ein „cross-platform GUI test automation tool“, das auf vielen GUI-Technologien und Plattformen basiert.

Folgende GUI-Technologien werden unterstützt:

- |  |   |
|--|---|
|  Nokia QT Software Frameworks     |  Web, HTML, AJAX             |
|  Java SWT, Eclipse RCP, AWT/Swing |  MAC OSX Carbon/Cocoa        |
|  iPhone, iPad CocoaTouch, iPod   |  Native Windows (.NET, MFC) |

Folgende Plattformen werden unterstützt:

- |  |  |
|--|--|
|  Windows                    |  Linux, MeeGo |
|  Unix                       |  MAC OSX      |
|  Diverse mobile Plattformen |  |

## Features

Die froglogic GmbH legt in Squish den Schwerpunkt auf Unabhängigkeit. Fast alle Plattformen und GUI-Technologien werden unterstützt. Squish verfügt neben einer einfachen Capture-Funktion auch über einen Wiedererkennungsmechanismus für Objekte. Für die Erstellung von Skripten wird die freie Auswahl zwischen vier verschiedenen Skriptsprachen gelassen: Python, JavaScript, Perl und Tcl. Auch ist es möglich Squish über die Kommandozeile im Batch-Modus zu betreiben. Jedoch fehlen Funktionalitäten wie die Bereitstellung von Auswertungen ausgeführter Tests sowie das Erstellen einer Testdokumentation.

## Lizenz

Die froglogic GmbH bietet Squish in zwei Lizenzmodellen an.

**Named Lizenzen** Jeder Benutzer bekommt eine eigene Lizenz und kann diese auf uneingeschränkt vielen Rechnern und Betriebssystemen nutzen. Eine Named Lizenz kostet inklusive einem Jahr Support und Updates 2.400 EUR.

**Gruppenlizenz** Bei der Gruppenlizenz kommt es auf zwei Faktoren an. Die Anzahl der Benutzer insgesamt und Anzahl der Benutzer, die Squish im Durchschnitt gleichzeitig benutzen. Bezüglich der SIV.AG wurde bei der Anzahl der Benutzer auf 15 und bei der gleichzeitigen Nutzung auf fünf bis sieben geschätzt. Eine Gruppenlizenz würde in diesem Fall 18.000 EUR kosten.

## Fazit

Squish überzeugt in der breiten Unterstützung an GUI-Technologien und Plattformen. Auch überzeugend ist die Skripterstellung mittels Capture und die Komponentenwiedererkennung. Squish weist allerdings einige Defizite beim Umgang mit komplexen Oberflächenstrukturen auf. Auch Funktionalitäten wie Reporting und das Erstellen einer Testdokumentation fehlen. Squish kommt damit als möglicher Kandidat nicht in Frage.

### 5.2.6 SWTBot

Ein weiteres freies Produkt heißt SWTBot und wurde von Ketan Padegaonkar entwickelt. SWTBot ist weder ein eigenständiges Programm noch ein Plugin. Es ist lediglich eine Erweiterung des JUnit APIs und somit ein programmier-technischer Ansatz. Aus JUnit-Tests heraus können Anwendungen ferngesteuert und systematisch abgeprüft werden. SWTBot ist als „Klick-Roboter“ realisiert, der durch Aufrufe des SWTBot-APIs das GUI bedient.

## Features

Da SWTBot eine Erweiterung des JUnit-APIs ist, setzt es Programmierkenntnisse voraus. Es ist extrem robust gegenüber der zu testenden Anwendung,

da die GUI-Elemente direkt aus dem API angesprochen werden. Allerdings verfügt SWTBot über keine der geforderten Funktionen.

### Lizenz

SWTBot steht unter der EPL und gilt somit als freie Software.

### Fazit

SWTBot kommt ebenfalls als möglicher Kandidat nicht in Frage. Es verfolgt gänzlich einen anderen Ansatz, demzufolge fehlt es an Funktionalität.

### 5.2.7 Bewertungsübersicht der Testwerkzeuge

Beschreibung	Automated GUI Recorder	GUIDancer	Unified Functional Testing	QF-Test	Squish	SWTBot
Firma	Eclipse Community	Bredex GmbH	Hewlett-Packard	Quality First Software GmbH	Froglogic GmbH	Eclipse Community
kommerziell	Nein	Ja	Ja	Ja	Ja	Nein
Kosten/Lizenz	Eclipse Public License	238.00 EUR exkl. Support	Auf Anfrage	1595.00 EUR	2.400 EUR	Eclipse Public License
Anforderungen						
SWT-RCP-Support	✓	✓	✓	✓	✓	✓
Erstellung von Makros/Scripts	✓	✓	✓	✓	✓	✓
Reporting	✗	✓	✗	✓	✗	✗
Multi-User-Support	✗	✓	✗	✓	✓	✗
Batch-Betrieb	✗	✓	✗	✓	✓	✗
Erstellung von Testdokumentation	✗	✓	✗	✓	✗	✗
Fazit						
Erste Einschätzung	Nicht geeignet	<b>Möglicher Kandidat</b>	Nicht geeignet	<b>Möglicher Kandidat</b>	Nicht geeignet	Nicht geeignet

**Tabelle 5.4:** Bewertungsübersicht der Testwerkzeuge

### 5.3 QF-Test vs GUIDancer

Nachdem eine Bewertung der Anforderungsanalyse vollzogen wurde, bleiben zwei Testwerkzeuge als mögliche Kandidaten übrig. Um herauszufinden, welches Testwerkzeug die konkreten Anforderungen am besten erfüllt, wurde auf der Grundlage der Anforderungen ein Spezifikations- und Bewertungsformular entwickelt. Damit die beiden Testwerkzeugkandidaten bewertet werden konnten, wurden sie zunächst in einer isolierten Testumgebung ausgiebig untersucht.

Das Bewertungsformular enthält verschiedene Merkmale, die wiederum mehrere Anforderungen enthalten. Folgende Merkmale fließen in die Bewertung ein:

- Einfachheit der Benutzung
- Plattformunterstützung
- Aufspüren von Fehlern
- Funktionalität
- Berichtsfähigkeit
- Leistungs- und Belastungstest
- Versionskontrolle
- Planen und Verwalten
- Preisgestaltung
- Qualifikation des Anbieters
- Dokumentation

Für jede Anforderung wurde dann eine Gewichtung im Wertebereich von 1 bis 10 festgelegt, die angibt, wie bedeutend die Anforderung insbesondere für die SIV.AG bzw. das Testteam ist. Die dann von den Testingenieuren zu vergebenen Punktzahlen im Bereich von 1 bis 5 (1 = „Anforderung überhaupt nicht erfüllt“ und 5 = „Anforderung erfüllt es exzellent“) werden mit der Gewichtung multipliziert, wodurch dann ein Wert entsteht, der maximal 50 betragen kann. Zum Schluss werden alle Werte addiert, die dann eine Gesamtpunktzahl ergeben.

Mit 83% bzw. 912 von 1095 möglichen Punkten konnte sich QF-Test gegen den GUIDancer (67%, 737 von 1095 möglichen Punkten) durchsetzen. Das komplette Bewertungsformular befindet sich im Anhang B.1, B.2 und B.3.

Ausschlaggebend für den Sieg war unter dem Strich das Gesamtpaket QF-Test. QF-Test zeichnet sich durch die Einfachheit in der Bedienung aus, wofür maßgeblich das intuitive Design verantwortlich ist. Auch können die Testskripte im Gegensatz zum GUIDancer sehr schnell und einfach aufgenommen werden. GUIDancer verfolgt diesbezüglich kein Capture-Verfahren, welches zwar den Vorteil hat, dass die Testfälle schon während der Spezifikationsphase entwickelt werden können, jedoch beansprucht GUIDancer's Verfahren dafür extrem viel Zeit. Weiterhin ausschlaggebend war, dass Skriptsprachen in QF-Test eingesetzt werden können. Für einfache Testfälle werden in der Regel keine Skriptsprachen benötigt, sind Testfälle allerdings sehr komplex, können Skriptsprachen dabei sehr hilfreich sein. Letztendlich überzeugte QF-Test bei der Funktionalität, Qualifikation und Dokumentation. Bezüglich der Anforderungen bei der Qualifikation, das heißt Ausgereiftheit des Produkts, Marktanteil des Produkts oder Kundenunterstützung wurde QF-Test durchweg mit sehr gut eingestuft.

## Literaturverzeichnis

- [1] Dustin, Elfriede; Rashka, Jeff; Paul, John: Software automatisch testen: Springer (Xpert.press), 2000 - ISBN 3-540-67639-2
- [2] Spillner, Andreas; Linz, Tilo: Basiswissen Softwaretest: 2., überarbeitete Auflage: dpunkt.verlag, 2004 - ISBN 3-89864-256-9
- [3] Thaller, Georg Erwin: Software-Test Verifikation und Validation: 2., aktualisierte und erweiterte Auflage: Heise, 2002 - ISBN 3-88229-198-2
- [4] Hoffmann, Dirk W.: Software-Qualität: Springer-Verlag, 2008 - ISBN 978-3-540-76323-9
- [5] Westphal, Frank: Testgetriebene Entwicklung mit JUnit & FIT: 1. Auflage: dpunkt.verlag, 2006 - ISBN 3-89864-220-8
- [6] Wirdemann, Ralf: SCRUM mit User Stories: Carl Hanser Verlag, 2009 - ISBN 978-3-446-41656-7
- [7] Hauer, Philipp: Modellgetriebene Softwareentwicklung - Management der Modelle während der Entwicklung und zur Laufzeit, Hochschule für Wirtschaft und Recht Berlin, Fachbereich Duales Studium Wirtschaft·Technik, Bachelor-Thesis, 2011
- [8] Moser, Martin: Wann lohnt sich GUI-Testautomatisierung?, <http://www.qfs.de/en/info/HandoutsJFS07.pdf>, Stand: 04.05.2011
- [9] Linz, Tilo, Daigl Matthias: GUI Testing Made Painless. Implementation and Results of the ESSI Project Number 24306, 1998, <http://www.imbus.de/forschung/pie-gui-test/gui-testing-made-painless>
- [10] Littlewood, Bev: How Good Are Software Reliability Predictions? Software Reliability Achievement and Assessment, Oxford: Blackwell Scientific Publications, 1987
- [11] Scrum: SCRUM Zyklusmodell, <http://www.3m5.de/scrum>, Stand: 04.05.2011

- [12] Brčan, Gregor; Hänsel, Jochen: Automatisierung von GUI-Tests, TU Berlin, 2003, Stand: 30.05.2011
- [13] <http://qse.ifs.tuwien.ac.at/courses/skriptum/download/09PFunktwid20040204.pdf>, Stand: 31.05.2011
- [14] <http://entwickler-magazin.de/zonen/magazine/psecom,id,8,online,843,p,0.html>, Stand: 25.07.2011
- [15] Oracle Reports, [Oracle Reports Tutorial](#), Stand: 27.07.2011
- [16] GUIDancer Lizenzen, [GUIDancer Lizenzen](#), Stand: 01.08.2011
- [17] Schmid, Gregor: QF-Test - Das Handbuch, [http://www.qfs.de/qftest/manual\\_de.pdf](http://www.qfs.de/qftest/manual_de.pdf), Stand: 05.08.2011
- [18] Zöld, Philipp: SIV.AG kVASy5 User Interface Guidelines, Version 0.1, Stand: 28.06.2011

## Abbildungsverzeichnis

2.1	Scrum-Zyklusmodell, Quelle: [11]	9
2.2	Kosten durch Fehlerbehebung in den verschiedenen Stadien des Entwicklungslebenszyklus	11
2.3	Testverfahren	13
2.4	White-Box und Black-Box, Quelle: [13]	14
2.5	Testebenen mit zunehmender Granularität	15
2.6	Testmethoden	16
2.7	Skript-Techniken	19
3.1	Aufwandsanalyse Best Case	25
3.2	Aufwandsanalyse Worst Case	26
3.3	Aufwandsanalyse mit Einstellparameter	27
4.1	kVASy-Produktportfolio	28
4.2	kVASy4-Entwicklungskomponenten	29
4.3	Oracle Web Report, Quelle: [15]	30
4.4	kVASy5-Entwicklungskomponenten, Quelle: [7]	31
4.5	Komponenten der Eclipse-Plattform, Quelle: [7]	32
4.6	IDE kVASyClipse mit integriertem Maskenmodell-Editor	33
6.1	QF-Test Testsuite	47
6.2	QF-Test Schnellstart-Assistent	47
6.3	QF-Test Schnellstart-Assistent	48
6.4	QF-Test Schnellstart-Assistent	48
6.5	QF-Test Schnellstart-Assistent resultierende Knoten	49
6.6	Knoten Extrasequenzen	49
6.7	SUT im Checkmodus	51
6.8	Check Text	52
7.1	Jahrestestzyklus der SIV.AG	54
7.2	Teststruktur 3-Ebenen-Modell	55
7.3	Teststruktur 2-Ebenen-Modell	56
7.4	Teststruktur 1-Ebenen-Modell	57
7.5	QF-Test Testsuitemodell	58
7.6	Geschäftsvorfall im UML Aktivitätsdiagramm	59
7.7	QF-Test Report	63
7.8	QF-Test Abhängigkeiten	64
7.9	QF-Test Abhängigkeiten	65
A.1	kVASy-Produktportfolio	73

C.1	QF-Test Geschäftslogik Implementierung . . . . .	77
C.2	kVASy5 Geschäftslogik Implementierung . . . . .	78
C.3	QF-Test Usability Implementierung . . . . .	79
C.4	QF-Test Datengetriebene Tests Implementierung . . . . .	80
C.5	QF-Test Administration-Testsuite . . . . .	80

## Tabellenverzeichnis

3.1	Manuelles und automatisiertes Testen im Vergleich, Quelle: [1]	23
3.2	Break-Even-Punkt der Automatisierung von GUI-Tests, Quelle: [9]	24
3.3	Parametereinstellungen vom Testingenieur	26
5.1	GUIDancer Lizenzen, Quelle: [16]	38
5.2	QF-Test Entwicklerlizenzen	40
5.3	QF-Test Runtimelizenzen	40
5.4	Bewertungsübersicht der Testwerkzeuge	43
6.1	Check-Typen von Check-Boolean, Quelle: [17]	50
B.1	Bewertung QF-Test und GUIDancer	74
B.2	Bewertung QF-Test und GUIDancer	75
B.3	Bewertung QF-Test und GUIDancer	76

## Abkürzungsverzeichnis

<b>AGR</b>	Automated GUI Recorder
<b>EPL</b>	Eclipse Public License
<b>ERP</b>	Enterprise Resource Planning
<b>GUI</b>	Graphical User Interface
<b>HTML</b>	Hypertext Markup Language
<b>IDE</b>	Integrated Development Environment
<b>MDS</b>	Model-Driven Software Development
<b>ODBC</b>	Open Database Connectivity
<b>PL/SQL</b>	Procedural Language Structured Query Language
<b>RCP</b>	Rich Client Platform
<b>PDF</b>	Portable Document Format
<b>RDBMS</b>	Relational Database Management System
<b>RTF</b>	Rich Text Format
<b>SOA</b>	Service-oriented Architecture
<b>SUT</b>	Software Under Test
<b>SWT</b>	Standard Widget Toolkit
<b>TPTP</b>	Test & Performance Tools Platform
<b>UFT</b>	Unified Functional Testing
<b>XML</b>	Extensible Markup Language

## B Bewertung QF-Test und GUIDancer

Merkmale des Test-Tools	Gewichtung (1-10)	Punktzahl (1-5)		Wert (1-50)	
		GUIDancer	QF-Test	GUIDancer	QF-Test
<b>Einfachheit der Benutzung</b>					
Lernkurve	7	2	5	14	35
Pflegeleichtigkeit des Tools	9	3	4	27	36
Einfache Installation	4	4	4	16	16
<b>Plattformunterstützung</b>					
Lässt es sich übertragen und unmittelbar über ein Netzwerk auf mehreren Plattformen betreiben?	5	5	5	25	25
<b>Aufspüren von Fehlern</b>					
Funktion zum Aufspüren von Fehlern?	10	3	4	30	40
<b>Funktionalität</b>					
Kann Skriptsprache eingesetzt werden?	10	1	4	10	40
Ist modulare Skriptentwicklung möglich?	9	1	5	9	45
Aufzeichnung auf Steuerelementebene	8	1	5	8	40
ODBC Unterstützung	6	2	4	12	24
Unterstützt das Anlegen einer Bibliothek wiederverwendbarer Funktionen?	8	4	4	32	32
Analyse der Testergebnisse – automatische Erstellung eines Ergebnisprotokolls	10	4	4	40	40
Unterstützung Datengetriebenes Testen	10	3	4	30	40

**Tabelle B.1:** Bewertung QF-Test und GUIDancer

Merkmale des Test-Tools	Gewichtung (1-10)	Punktzahl (1-5)		Wert (1-50)	
		GUIDancer	QF-Test	GUIDancer	QF-Test
<b>Berichtsfähigkeit</b>					
Grafische Darstellung der Ergebnisse (Diagramme und Kurven)?	6	4	1	24	6
Berichterstellung	8	4	4	32	32
Lassen sich Berichte modifizieren (Design)?	3	5	3	15	9
<b>Leistungs- und Belastungstest</b>					
Unterstützung von Belastungs-, Last- und Leistungstest?	8	2	3	16	24
Benutzersimulation ohne Einsatz physischer Arbeitsstationen möglich?	6	4	5	24	30
Unterstützung Ressourcenüberwachung (Arbeitsspeicher, Platz auf Datenträgern, Systemressourcen)	8	2	5	16	40
<b>Versionskontrolle</b>					
Besitzt das Tool eine integrierte Fähigkeit zur Versionskontrolle?	10	5	3	50	30
<b>Planen und Verwalten</b>					
Integration von Tools zum Planen und Verwalten möglich?	3	2	4	6	12
<b>Preisgestaltung</b>					
Liegt der Preis im geschätzten Bereich?	8	5	3	40	24
Welche Art von Lizenzen wird vergeben (fließend, fest)?	3	5	2	15	6

**Tabelle B.2:** Bewertung QF-Test und GUIDancer

Merkmale des Test-Tools	Gewichtung (1-10)	Punktzahl (1-5)	Punktzahl (1-5)	Wert (1-50)	Wert (1-50)
		GUIDancer	QF-Test	GUIDancer	QF-Test
<b>Qualifikation des Anbieters</b>					
Ausgereiftheit des Produkts	8	4	5	32	40
Marktanteil des Produkts	5	3	5	15	25
Werden Software-Patches geliefert, wenn sie nötig erscheinen?	10	5	5	50	50
Gibt es regelmäßige Aktualisierungen?	8	5	5	40	40
Kundenunterstützung	8	4	5	32	40
Schulung verfügbar	7	5	3	35	21
<b>Dokumentation</b>					
Besitzt das Tool eine Hilfefunktion?	7	5	5	35	35
Ist es gut dokumentiert?	7	1	5	7	35
<b>Gesamtpunkte (mögliche Gesamtpunktzahl: 1095)</b>				<b>737</b>	<b>912</b>
<b>In Prozent</b>				<b>67,31</b>	<b>83,29</b>

**Tabelle B.3:** Bewertung QF-Test und GUIDancer