

Wann lohnt sich GUI- Testautomatisierung?

Martin Moser, Gregor Schmid

Quality First Software GmbH

qfs@qfs.de

Tel: +49 8171 919870

Überblick

Hintergrund

Motivation

Phasen der GUI-Testautomatisierung

Vorteile der GUI-Testautomatisierung

Vorteile sichern

Quality First Software GmbH

Gegründet 2001

Hauptprodukt: **QF-Test** – Das Java GUI Testtool

Mitarbeiter: 7

Sitz nahe München

Qualität steht im Vordergrund

Fokus auf Java und Testautomatisierung

Mehr als 350 Kunden weltweit in allen Wirtschaftszweigen

Referenzen



Überblick

Hintergrund

Motivation

Phasen der GUI-Testautomatisierung

Vorteile der GUI-Testautomatisierung

Vorteile sichern



Warum Testen?

Warum Testen?



Überblick

Hintergrund

Motivation

Phasen der GUI-Testautomatisierung

Vorteile der GUI-Testautomatisierung

Vorteile sichern

Definition GUI-Tests

Unit-Tests

sehr wichtig, aber isolierte Subsysteme
auf Klassenebene

Integrationstests

testen das Zusammenspiel von Subsystemen
schwierig aufzusetzen

GUI-Tests

testen nicht das GUI, sondern das System *als Ganzes über
das GUI* -> ein häufiges Missverständnis

werden aus Sicht des Endanwenders an einem „lebenden“
System ausgeführt

Testphasen für GUI-Tests

Integrationstests

Systemtests

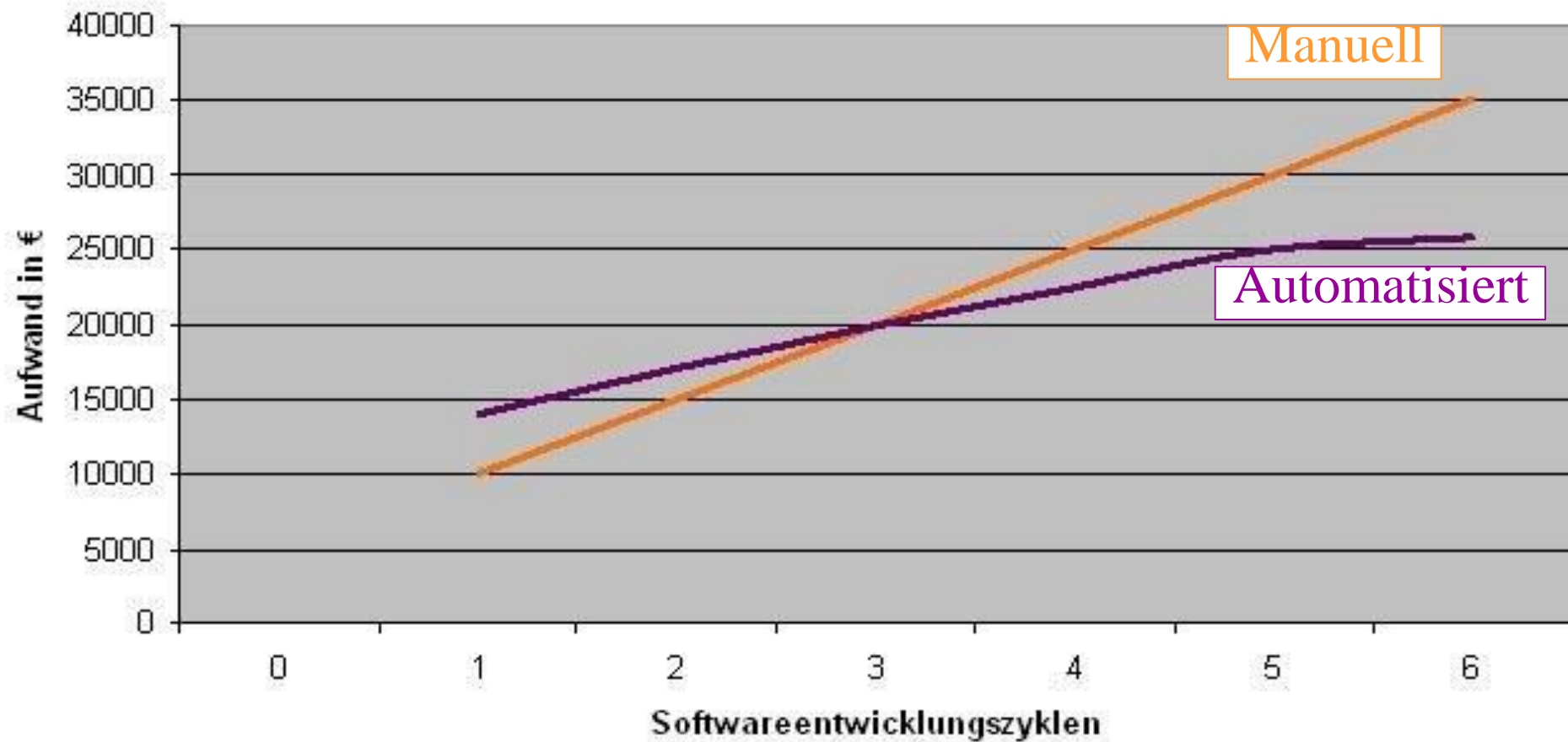
Funktionale Tests

Regressionstests

Load/Stress-Tests

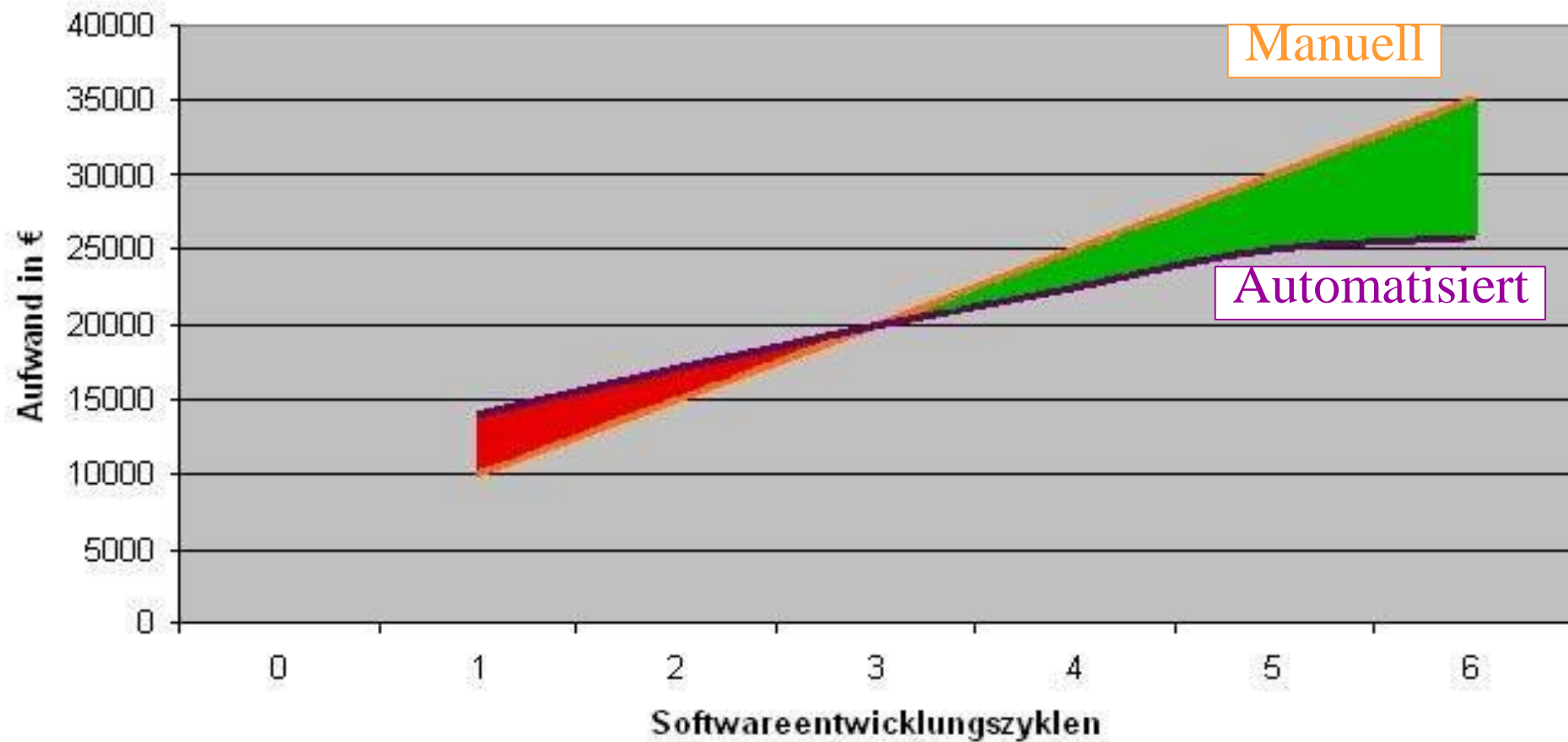
I18N/L10N-Tests (Internationalization/Localization testing)

Return On Investment (ROI)



© Imbus AG, www.imbus.de

Return On Investment (ROI)

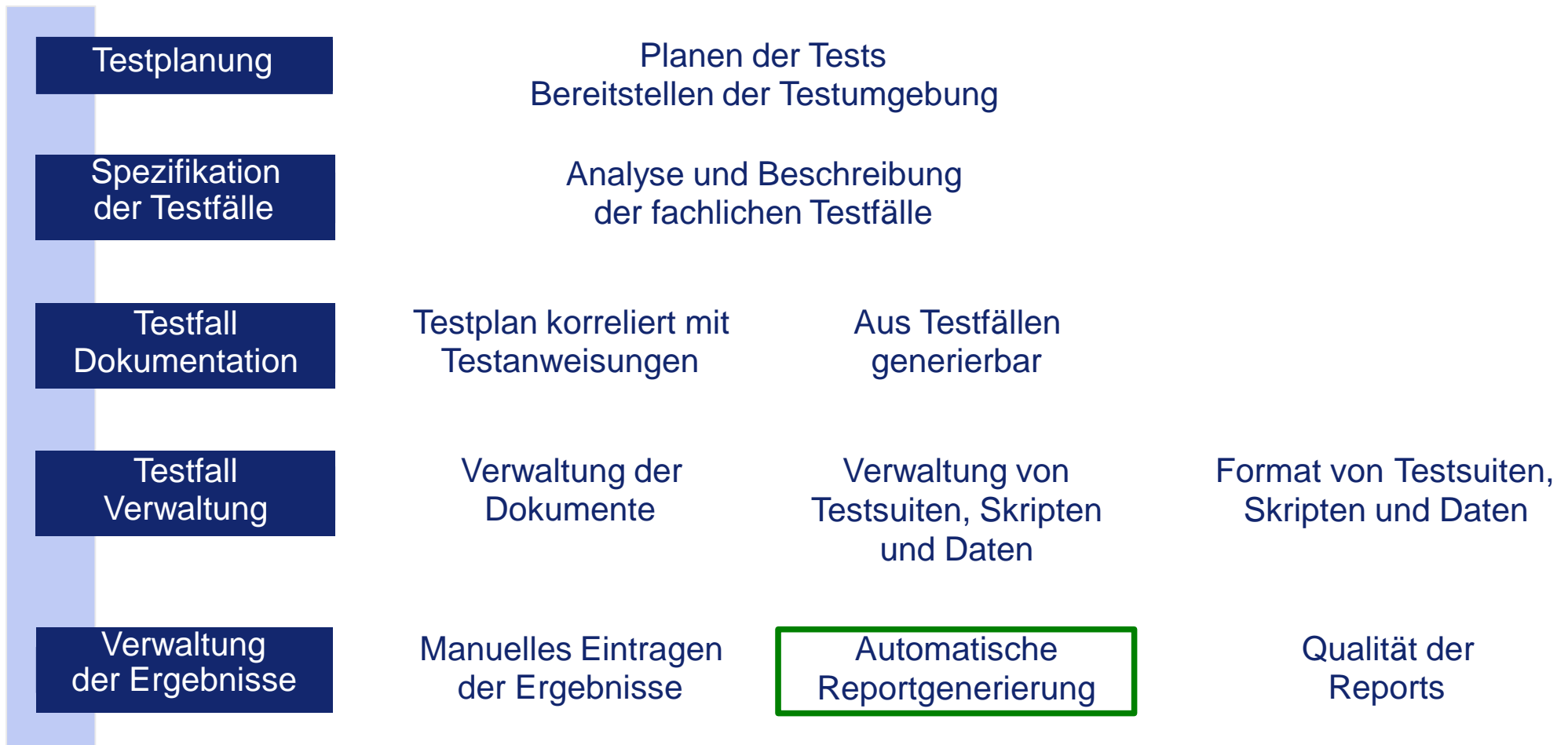


© Imbus AG, www.imbus.de

Phasen des Testprozesses



Phasen mit wenig Einfluss auf den ROI



Phasen mit hohem Einfluss auf den ROI



Testfall
Entwicklung

Erstellen der
Anweisungen für die
Tester

Implementierung der
Testfälle mit dem
Testtool

Komplexität,
Bedienbarkeit des Tools,
Möglichkeiten zur
Wiederverwendung

Test
Ausführung

Pflege der
Testfälle

Phasen mit hohem Einfluss auf den ROI



Testfall
Entwicklung

Erstellen der
Anweisungen für die
Tester

Implementierung der
Testfälle mit dem
Testtool

Komplexität,
Bedienbarkeit des Tools,
Möglichkeiten zur
Wiederverwendung

Test
Ausführung

Langsam, hohe Kosten
für Personal und
Hardware

Automatisch, schnell,
optimale Ausnutzung
der Hardware

Zuverlässigkeit des
Testtools bei der
Testdurchführung

Phasen mit hohem Einfluss auf den ROI



Erstellen der Anweisungen für die Tester

Implementierung der Testfälle mit dem Testtool

Komplexität, Bedienbarkeit des Tools, Möglichkeiten zur Wiederverwendung

Test Ausführung

Langsam, hohe Kosten für Personal und Hardware

Automatisch, schnell, optimale Ausnutzung der Hardware

Zuverlässigkeit des Testtools bei der Testdurchführung

Anpassung der Anweisungen nur nach fundamentalen Änderungen

Anpassung der Testfälle an die Veränderungen im GUI

Qualität der Wiedererkennung, Anpassungsfähigkeit an verändertes GUI, Modularisierung

Plattformübergreifende Testautomatisierung



Testfall
Entwicklung

Erstellen der
Anweisungen für die
Tester

Implementierung der
Testfälle mit dem
Testtool

Anpassung der Testfälle,
die plattformabhängig
sind. Bereitstellung
plattformspezifischer
Testdaten

Test
Ausführung

Langsam, hohe Kosten
für Personal und
Hardware

Automatisch, schnell,
optimale Ausnutzung
der Hardware

Abdeckung mehrerer
Plattformen

Anpassung der
Anweisungen nur nach
fundamentalen
Änderungen

Anpassung der
Testfälle an die
Veränderungen im GUI

Änderungen im GUI nur
einmal nachziehen

Überblick

Hintergrund

Motivation

Phasen der GUI-Testautomatisierung

Vorteile der GUI-Testautomatisierung

Vorteile sichern

Entscheidend für den ROI

Wiederverwendbarkeit

Wiederverwendbarkeit innerhalb mehrerer Tests durch Modularisierung

Häufigkeit der Regressionstests

Stabilität der Tests bei Systemveränderung

Einsatz auf mehreren Plattformen

Tests verschiedener Produktversionen, -linien

Wiederverwendung der funktionalen Tests, z.B. für Lasttests oder zur Systemüberwachung

Vorteile von Automatisierung

Ermöglicht Regressionstests -> häufigere und schnellere Testausführung -> kürzere Entwicklungszyklen

Höhere Zuverlässigkeit (menschlicher Faktor)

Reproduzierbare Ergebnisse

Unbeaufsichtigte Testausführung ohne Benutzerinteraktion

Motivation für manuelle Tester: Konzentration auf schwierige Testszenarios anstatt langweiliger Routinetests

schnellerer Markteintritt
höhere Produktqualität
höhere Zuverlässigkeit

Überblick

Hintergrund

Motivation

Phasen der GUI-Testautomatisierung

Vorteile der GUI-Testautomatisierung

Vorteile sichern

Auswahl des Testtools

- Stabiles und verlässliches Capture/Replay?
- Erkennung aller Arten von GUI-Elementen (auch komplexe Elemente wie Bäume oder Tabellen)?
- Modularisierung von Tests möglich?
- Parametrisierung von Testprozeduren möglich?
- Plattformübergreifend? Werden alle Test-Plattformen unterstützt?
- Integrationsmechanismus mit existierenden Test-Execution/Management-Tools?
- L10N-Tests möglich?

Wie werden GUI-Tests robust?

Kommunikation zwischen Testern und Entwicklern

Erstellung wiederverwendbarer Testprozeduren

Erstellung spezifischer Testbibliotheken für
Softwarekomponenten

Entwicklungszyklus der Testautomatisierung parallel oder
ähnlich dem Softwareentwicklungszyklus

Verfügbare Automatisierungstools

Windows

QuickTest Professional (Mercury/HP, aka WinRunner), XDE Functional Tester (IBM Rational, aka Robot), Silktest (Borland), TestPartner, QARun (Compuware) etc.

Unix

XRunner (Mercury/HP), XDE Functional Tester, Silktest, Squish (Froglogic for QT and XView).

Web

Diverse kommerzielle Capture/Replay Tools in allen Preiskategorien, ebenso diverse Open Source Tools

Java/Swing

Open Source: Abbot, JFCUnit, Marathon – sehr entwicklerlastig
Windows-basierte Testtools bieten inzwischen Java Plugins für Swing.

QF-Test (Quality First Software GmbH)

java/SWT

Abbot, Windows-basierte Tools mit Einschränkungen bei Objekterkennung.

QF-Test (Quality First Software GmbH)

Wann lohnt sich GUI-Testautomatisierung?

**Einsatz in richtigen Phasen
Modularisierung und
Wiederverwendung
Einsatz eines geeigneten Tools**

Martin Moser, Gregor Schmid

Quality First Software GmbH

qfs@qfs.de

Tel: +49 8171 919870