When does automation of GUI testing pay off?

Martin Moser, Gregor Schmid

Quality First Software GmbH
qfs@qfs.de
Tel: +49 8171 919870

Overview

- **Background**
- Motivation
- Phases of GUI test automation
- Profits through GUI test automation
- How to gain profits

Quality First Software GmbH

- Established 2001

- Based near Munich

- Primary product: **QF-Test** – The Java GUI Testtool

- Committed to quality

- Focus on Java and test automation

- Over 300 customers worldwide in all kinds of business categories

References

Overview

- Background
- **Motivation**
- Phases of GUI test automation
- Profits through GUI test automation
- How to gain profits

Why Testing?

Why Testing?

Overview

- Background
- Motivation
- **Phases of GUI test automation**
- Profits through GUI test automation
- How to gain profits

GUI Testing in General

- **Unit tests**
  - very important, but test isolated subsystems
  - typically at class level
- **Integration tests**
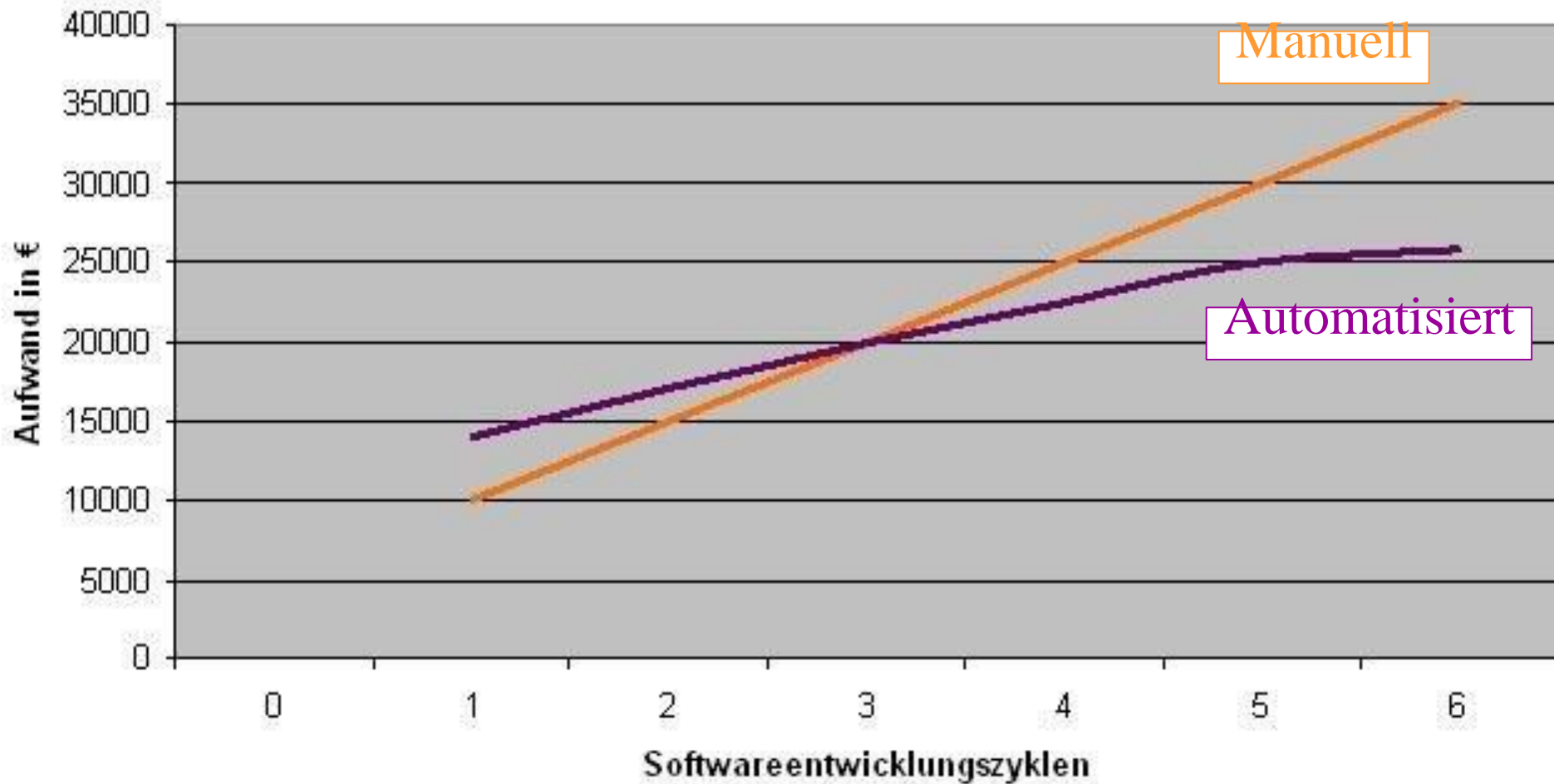 - difficult to set up for subsystems in combination
- **GUI tests**
  - are no substitute for system tests
  - tests the system as a whole through the GUI, not the GUI itself -> a common misconception
  - operate from the point of view of the end user
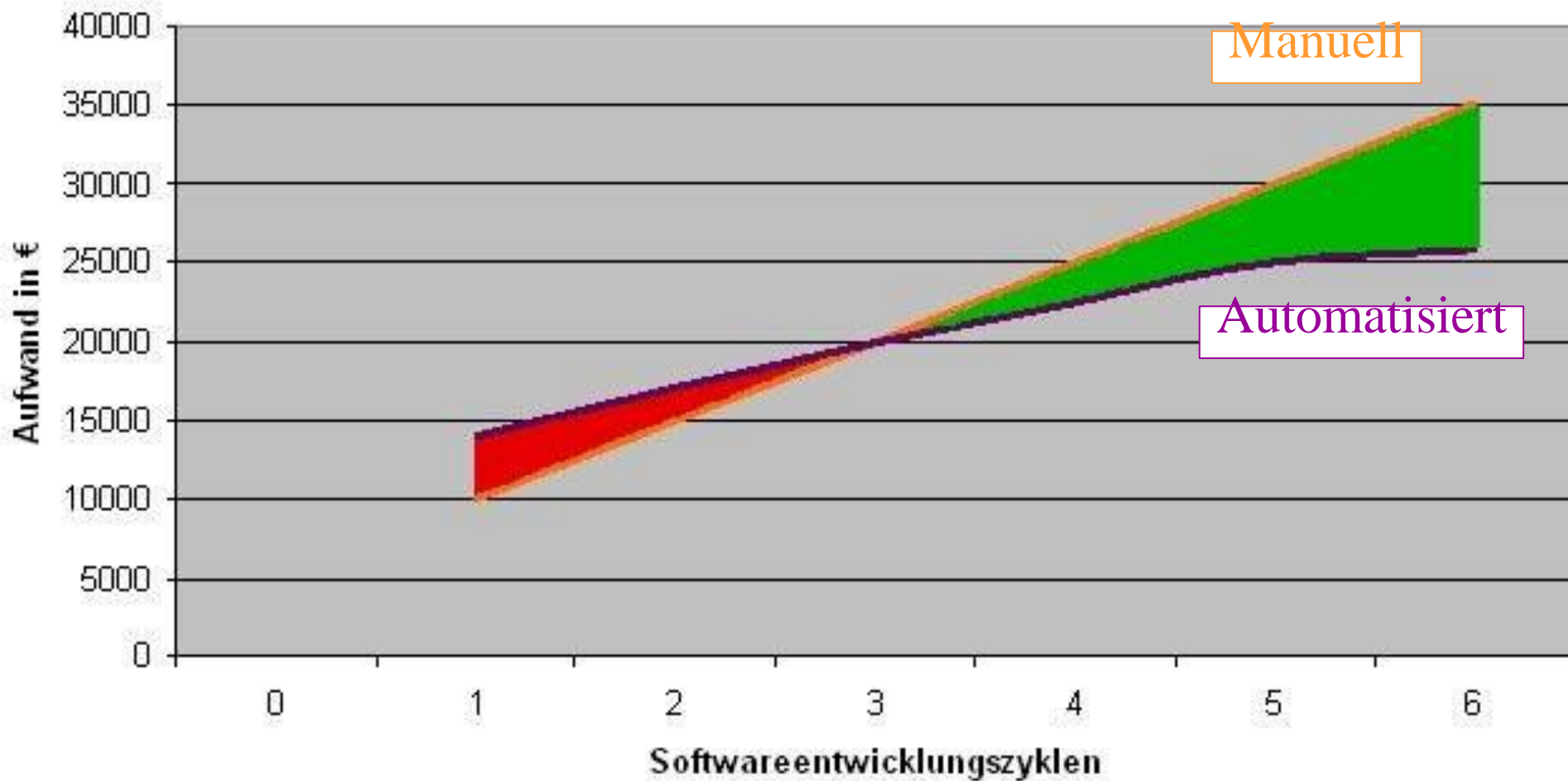
Test Phases for GUI Testing

- Integration testing
- System testing

- Functional testing
- Regression testing
- Load testing
- Stress testing
- I18N testing (Internationalization testing)
- L10N testing (Localization testing)

Return in Investment (ROI)

Return in Investment (ROI)

Phases of the Testing Process

Preparations

Determining test-cases

Test Development

Test-case Documentation

Test-case Management

Test Execution

Management of Results

Maintenance of Tests
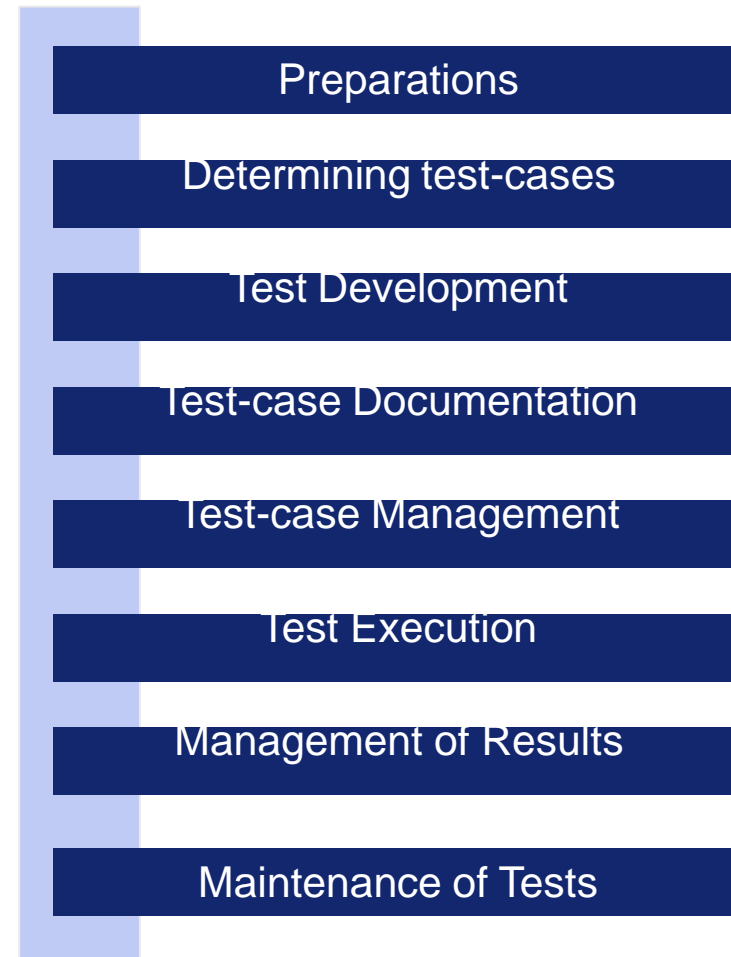
Phases with Little Influence on ROI

	Manual	Automated	Influencing factors
Preparations	Test planning Provisions for testing environment		
Determining Test-cases	Analysis of business cases		
Test-case Documentation	Test-plans correlate with test instructions	Generated from test-cases	
Test-case Management	Maintenance of documents	Management of test-suites, scripts and data	Format of test-suites, scripts and data
Management of Results	Manual entries of test-results	Automatic report generation	Quality of reports

Phases with Strong Influence on ROI

	Manual	Automated	Influencing factors
Test Development	Preparations of instructions for testers	Development of test-cases with test-tool	Complexity, ease of use of the tool, possibilities for reuse
Test Execution			
Maintenance of Tests			

Phases with Strong Influence on ROI

	Manual	**Automated**	**Influencing factors**
Test Development	Preparations of instructions for testers	Development of test-cases with test-tool	Complexity, ease of use of the tool, possibilities for reuse
Test Execution	Slow, high costs for multiple testers and associated hardware	Automatic, fast, optimal use of available hardware	Quality and reliability of test execution engine
Maintenance of Tests			

Phases with Strong Influence on ROI

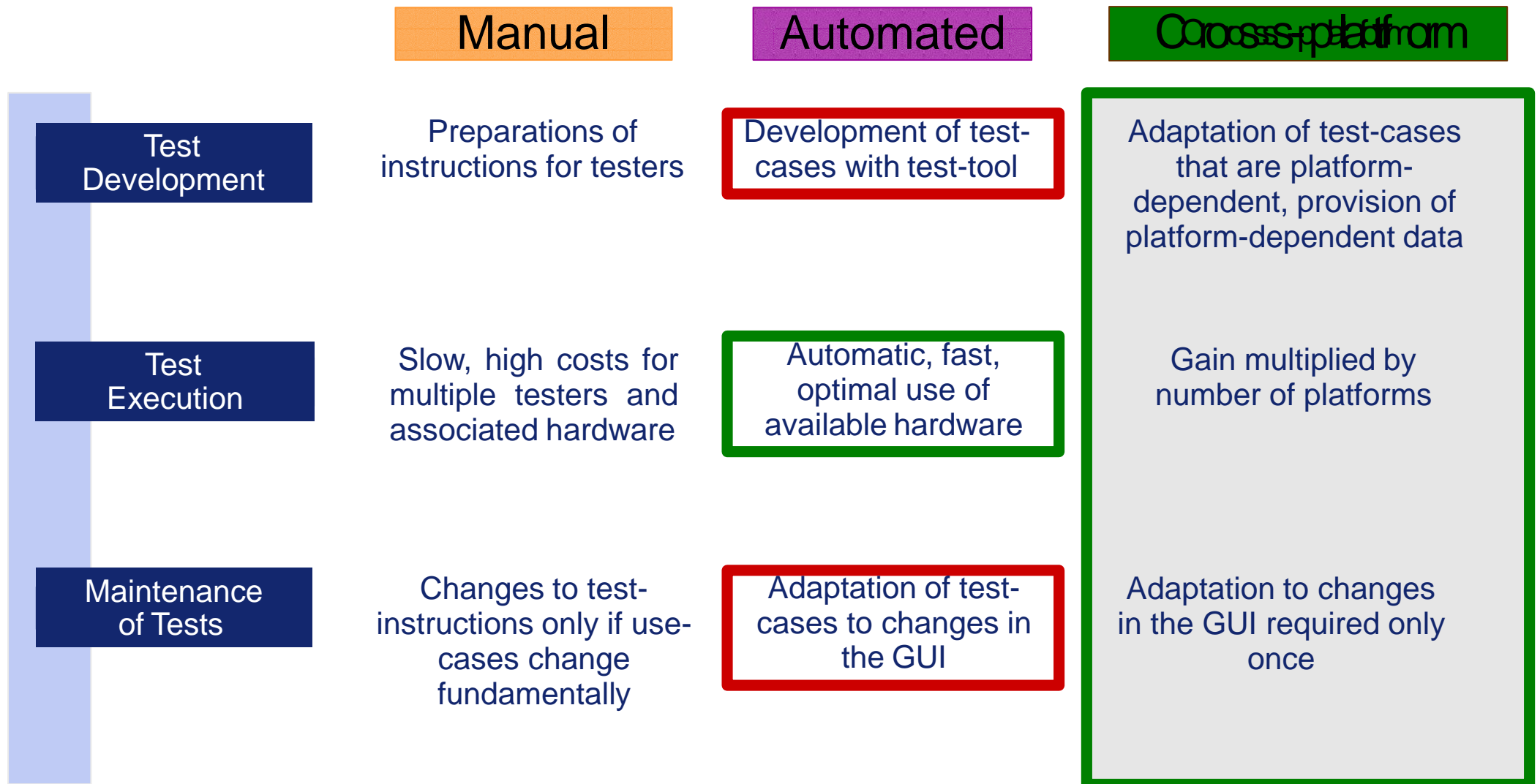|	Manual	Automated	Influencing factors
Test Development	Preparations of instructions for testers	Development of test-cases with test-tool	Complexity, ease of use of the tool, possibilities for reuse
Test Execution	Slow, high costs for multiple testers and associated hardware	Automatic, fast, optimal use of available hardware	Quality and reliability of test execution engine
Maintenance of Tests	Changes to test-instructions only if use-cases change fundamentally	Adaptation of test-cases to changes in the GUI	Quality of component recognition, adaptability to GUI changes, support for modularization

Cross-Platform Test Automation

	Manual	Automated	Cross-platform
Test Development	Preparations of instructions for testers	Development of test-cases with test-tool	Adaptation of test-cases that are platform-dependent, provision of platform-dependent data
Test Execution	Slow, high costs for multiple testers and associated hardware	Automatic, fast, optimal use of available hardware	Gain multiplied by number of platforms
Maintenance of Tests	Changes to test-instructions only if use-cases change fundamentally	Adaptation of test-cases to changes in the GUI	Adaptation to changes in the GUI required only once

Overview

- Background
- Motivation
- Phases of GUI test automation
- **Profits through GUI test automation**
- How to gain profits

Crucial Factor for ROI

Reuse on All Levels

- Reuse within tests through modularization
- Frequency of regression test execution
- Stability of tests during system changes
- Cross-platform testing
- Reuse of functional tests in other scenarios like load testing or L10N testing

Profits from Automation

- Enables regression testing -> more test-runs
- Higher reliability (human factor)
- Reproducible results
- Unattended test-runs without user interaction
- Humans motivation: energy for difficult scenarios, get rid of boring manual routines

**shorter time-to-market
higher product quality
higher reliability**

Overview

- Background

- Motivation

- Phases of GUI Test Automation

- Profits Through GUI Test Automation

- **How to gain profits**

Tool Selection

- Stable and reliable Capture/Replay?

- Recognition of all kinds of GUI-elements (also for complex components, e.g. Trees or Tables?)

- Modularization of tests?

- Parametrization of test-procedures?

- Cross-platform? Are target test platforms supported?

- Integration mechanism with existing test execution/management tools?

- L10N testing?

How to Make GUI-Tests Robust

- Establish communication between test engineers and software developers

- Create reusable test-procedures

- Create component based test-procedure libraries

- Development cycle for test automation parallel or nearby software development cycle

- Establish nightly testing

Available GUI Test Automation Tools

Windows

- *QuickTest Professional (*Mercury*, aka* WinRunner), *XDE Functional Tester* (IBM Rational, aka Robot), *Silktest* (Segue), *TestPartner, QARun* (Compuware) etc.

Unix

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

Web

- Several commercial Capture/Replay Tools in all pricing categories as well as OS tools

Java/Swing

- Open Source: Abbot, JFCUnit, Marathon – very developer oriented

- Windows based test tools offer Java plugins

- QF-Test

Java/SWT

- Abbot, Windows based tools with problems at object recognition

- QF-Test

When does automation of GUI testing pay off?

- *Use in the right areas*
- *Modularisation and reuse*
- *Use of a proper test tool*

Martin Moser, Gregor Schmid

Quality First Software GmbH
qfs@qfs.de
Tel: +49 8171 919870