



Objekt-orientiertes Programmieren

OOP 2006

The essence of modern software engineering

Mi 18

Wann Lohnt sich die Automatisierung von Tests?

Gregor Schmid
Frank Schmeißner

Organized by:

SIGS DATACOM

Lindlastr. 2c, 53842 Troisdorf,
Tel.: +49 (0)2241 2341-100, Fax.: +49 (0)2241 2341-199
www.oopconference.com



Intelligenter testen ...
... bessere Produkte!



imbus Kurzprofil

- **Fokus auf Softwarequalitätssicherung und -Test**
 - imbus Softwaretest Services, imbus Testlabor
 - imbus Akademie (**ISTQB Certified Tester - Foundation Level bis Expert Level**)
 - imbus Beratung: SPICE, TPI, ..
 - imbus TestBench - der komplette Testcycle in einer WorkBench
- 85 Mitarbeiter - drei Standorte
- Internationales Partnernetz (UK, NL, USA, India, China)
- Technologieführer



www.imbus.de

2

QFS
Quality First Software

Technologies

Wann lohnt sich die Automatisierung von GUI Tests?

Gregor Schmid
Quality First Software GmbH
gregor.schmid@qfs.de
Tel: +49 8171 919870

© 2005 Quality First Software GmbH 18.01.06 1

QFS
Quality First Software

Technologies

Quality First Software GmbH


- . Gegründet 2001
- . Hauptprodukt: *qftestJUI* – Das Java GUI Testtool
- . Mitarbeiter: 5
- . Sitz nahe München
- . Qualität steht im Vordergrund
- . Fokus auf Java und Testautomatisierung
- . Mehr als 240 Kunden weltweit in allen Wirtschaftszweigen

© 2005 Quality First Software GmbH 18.01.06 2

QFS
Quality First Software

Technology

Warum Testen?

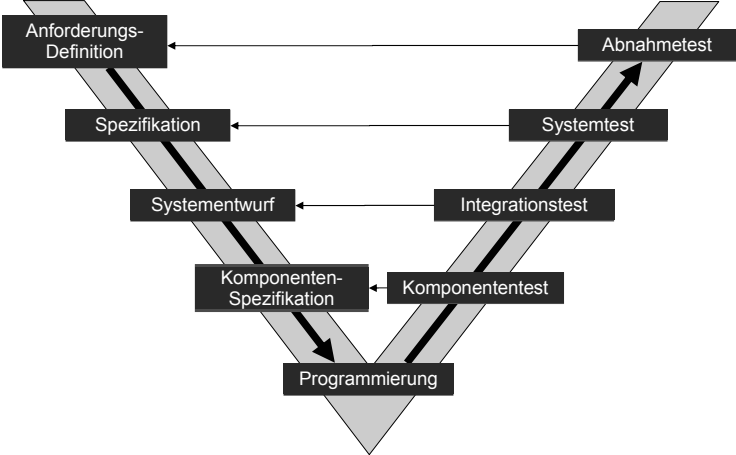


© 2005 Quality First Software GmbH 18.01.06 3

QFS
Quality First Software

Technology

Testen: Das V-Modell



© 2005 Quality First Software GmbH 18.01.06 4

Die Teststufen

- Unit Tests testen isolierte Subsysteme, typischerweise auf Klassenebene.
- Integrationstests testen das Zusammenspiel von Subsystemen. Sie sind oft schwer aufzusetzen.
- Beide sind kein Ersatz für Systemtests.
- GUI Tests testen nicht *das GUI*, sondern das System als Ganzes *über das GUI*.
- Systemtests werden aus Sicht des Endanwenders an einem „lebenden“ System ausgeführt.
- Abnahmetests werden durch den Kunden durchgeführt.

GUI Testautomatisierung

- Manuelle GUI Tests sind zeitraubend und monoton
->Automatisierung hat ein hohes Einsparpotential

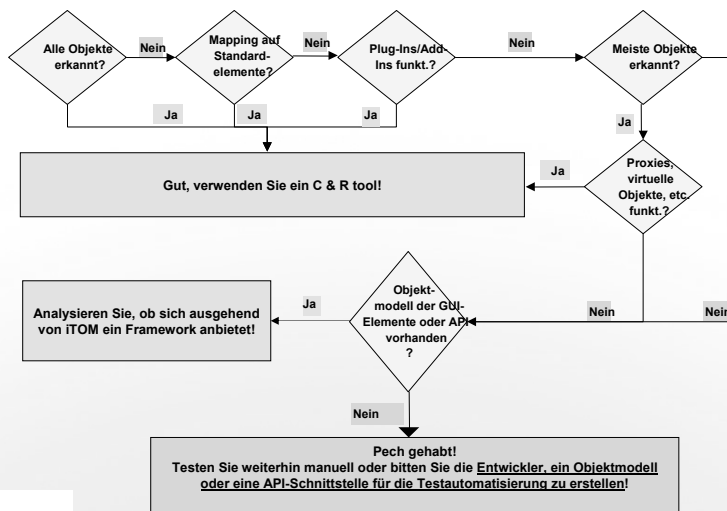
Kurze Demo:
Capture/Replay und darüber hinaus

GUI Testautomatisierung

- Manuelle GUI Tests sind zeitraubend und monoton
->Automatisierung hat ein hohes Einsparpotential

Wann lohnt sich die Automatisierung
von GUI Tests?

Pattern 2: Was tun bei Problemen mit der Objekterkennung?



www.imbus.de

3

Pattern 3: Design for Testability

Was machen Entwickler aus Sicht der Tester?

- Sie machen Fehler.
- Sie entwickeln Software, die schwer zu testen ist.

→ Sind Entwickler die natürlichen Feinde der Tester?



Nein!
Redet miteinander!



Entwickler sollten beim Testen beteiligt sein!

Tester sollten frühzeitig beim Design hinzugezogen werden, insbesondere im Hinblick auf eine Automatisierung von Tests!



www.imbus.de

4

Wir wollen Geld sparen ⇒ Testautomation! (IST-Zahlen aus der Praxis)

Input	Phase	Activity		Effort	
Specification	Test Specification	Creation	64,00 h	20 %	6400,00 €
		Maintenance Effort per Year	12,80 h		
Automation	Test Automation	Creation	70,00 h	50 %	3500,00 €
		Maintenance Effort per Year	14,00 h		
Execution	Test Execution	manual	Effort	0,15 h	22,50 €
			hourly Rate		
		automatic	Effort	0,50 h	100,00 €/h
			hourly Rate		
Logging	Test Logging	manual	0,50 h		
		automatic	0,15 h		
miscellaneous Parameter	Number of Releases per Year				2
	Number of Servicepacks per Release				2
	Number of initial Test Cycles per Year (calculated)				6
	Number of Test Cycles per Release or SP respect.				3
	Number of Platforms to be tested per Test Cycles				1,25

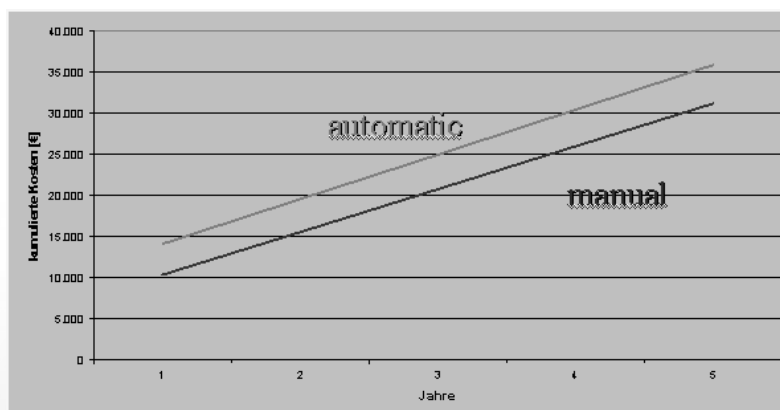
Results	Timeframe		Effort [h]		accumulated Costs [€]			
			Nr. of Test Cycles	manual	automatic	manual	automatic	
1st Year		Test Specification		64,00	64,00	6400,00	6400,00	
		Test Automation		0,00	70,00	0,00	7000,00	
		Test Execution	22,5	56,25	3,38	2812,50	337,50	
		Logging	22,5	11,25	3,38	1125,00	337,50	
		Maintenance		12,80	12,80	1280,00	1280,00	
each additional Year		Maintenance Test Specification		0,00	35,00	0,00	3500,00	
		Maintenance Test Automation		22,5	56,25	3,38	2812,50	337,50
		Test Execution	22,5	56,25	3,38	2812,50	337,50	
		Logging	22,5	11,25	3,38	1125,00	337,50	



www.imbus.de

5

Pitfall 1: Wirtschaftlichkeit von Automatisierung bei 50 % jährlichem Wartungsaufwand



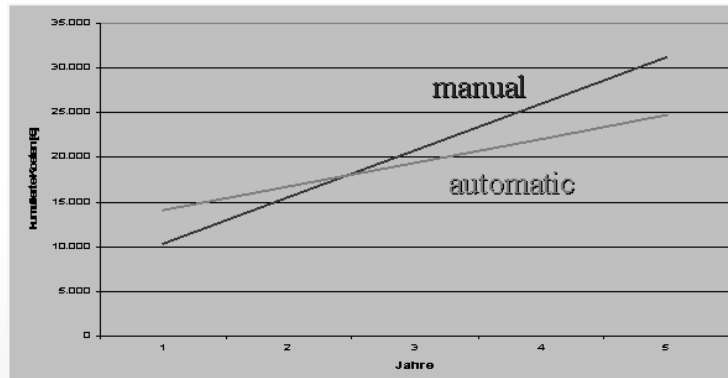
- Zusätzliche Investitionen: 3.740 €
- Break Even: nie!



www.imbus.de

6

Pitfall 1: Wirtschaftlichkeit von Automatisierung bei 10 % jährlichem Wartungsaufwand



	Zeitraumen	
	3	5
IRR	24%	57%
ROI	37%	174%
NPV	978,32	5.094,28

- **Zusätzliche Investitionen: 3.740 €**
- **Break Even: nach 2,5 Jahren**



www.imbus.de

7

Pitfall 1: Zusammenfassung und Erkenntnisse: Wirtschaftlichkeit von Testautomatisierung

Testautomatisierung lohnt, sofern Beschränkung auf geeignete Testthemen und Kombination mit manuellen Tests

- Neue Funktionen werden überwiegend manuell getestet; nur eine Teilmenge davon wird in die Regressions-Testsuite aufgenommen
- Kompatibilitätstests eignen sich für die Automatisierung
- Ein übergreifendes Testmanagement ist erforderlich



Zusätzliche wesentliche Einsparungen (schwer quantifizierbar)

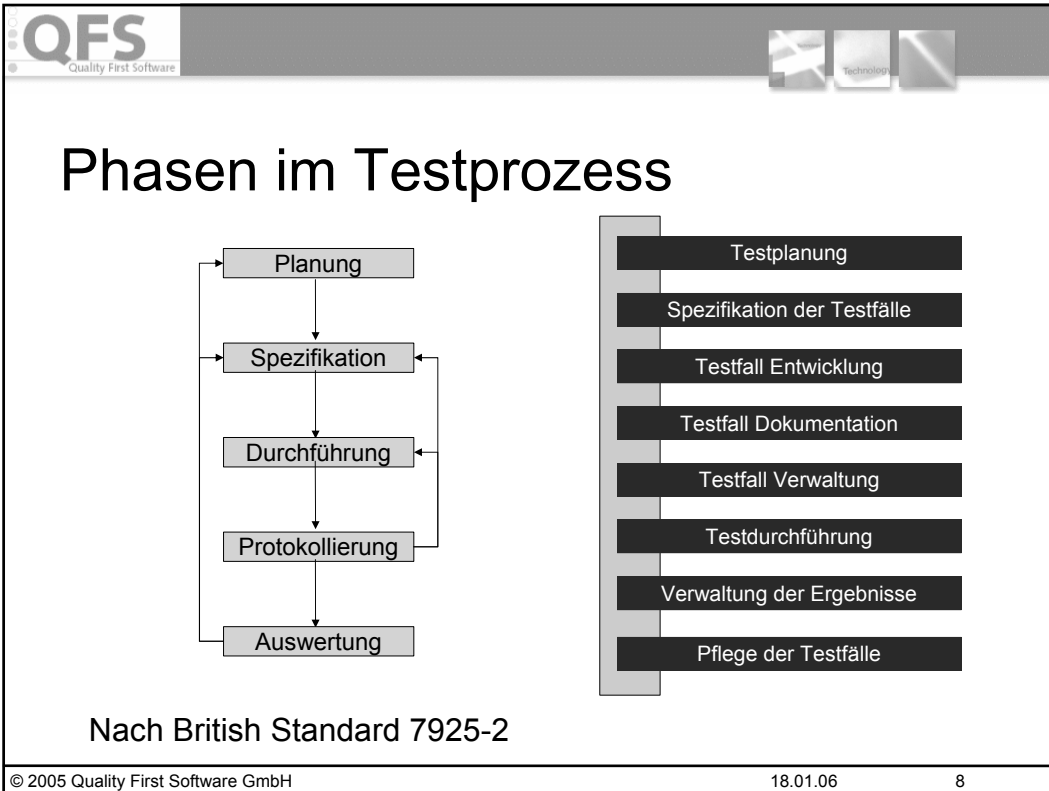


- Schnelleres Time to Market - die Testdurchführungszeiten werden um mindestens 50% reduziert.
- Höhere Qualität und daher größere Kundenzufriedenheit und geringere Fehlerfolgekosten (Support-Aufwände, Erstellung von Patches) aufgrund schnellerer und preiswerterer Regressionstests



www.imbus.de

8



QFS
Quality First Software

Phasen mit geringem Einfluss auf ROI

	Manuell	Automatisch	Einflussfaktoren
Testplanung	Planen der Tests Bereitstellen der Testumgebung		
Spezifikation der Testfälle	Analyse und Beschreibung der fachlichen Testfälle		
Testfall Dokumentation	Testplan korreliert mit Testanweisungen	Aus Testfällen generierbar	
Testfall Verwaltung	Verwaltung der Dokumente	Verwaltung von Testsuiten, Skripten und Daten	Format von Testsuiten, Skripten und Daten
Verwaltung der Ergebnisse	Manuelles Eintragen der Ergebnisse	Automatische Report Generierung	Qualität der Reports

© 2005 Quality First Software GmbH 18.01.06 9

	Manuell	Automatisch	Einflussfaktoren
Testfall Entwicklung	Erstellen der Anweisungen für die Tester	Implementierung der Testfälle mit dem Testtool	Komplexität, Bedienbarkeit des Tools, Möglichkeiten zur Wiederverwendung
Test-durchführung	Langsam, hohe Kosten für Personal und Hardware	Automatisch, schnell, optimale Ausnutzung der Hardware	Zuverlässigkeit des Testtools bei der Testdurchführung
Pflege der Testfälle	Anpassung der Anweisungen nur nach fundamentalen Änderungen	Anpassung der Testfälle an die Veränderungen im GUI	Qualität der Wiederkennung, Anpassungsfähigkeit an verändertes GUI, Modularisierung

© 2005 Quality First Software GmbH 18.01.06 10

	Manuell	Automatisch	Einflussfaktoren
Testfall Entwicklung	Erstellen der Anweisungen für die Tester	Implementierung der Testfälle mit dem Testtool	Komplexität, Bedienbarkeit des Tools, Möglichkeiten zur Wiederverwendung
Test-durchführung	Langsam, hohe Kosten für Personal und Hardware	Automatisch, schnell, optimale Ausnutzung der Hardware	Zuverlässigkeit des Testtools bei der Testdurchführung
Pflege der Testfälle	Anpassung der Anweisungen nur nach fundamentalen Änderungen	Anpassung der Testfälle an die Veränderungen im GUI	Qualität der Wiederkennung, Anpassungsfähigkeit an verändertes GUI, Modularisierung

Weitere Einsatzmöglichkeiten: Lasttests über das Client GUI

- Alternative zu Protokoll basierten Lasttests
- Höherer Ressourcenverbrauch, daher nur mit begrenzter Zahl von Clients möglich
- Einfache Erstellung komplexer Tests
- Hoher Grad an Wiederverwendbarkeit von funktionalen Tests
- Es werden End-To-End Zeiten ermittelt, im Gegensatz zu Antwortzeiten des Servers

© 2005 Quality First Software GmbH 18.01.06 11

Weitere Einsatzmöglichkeiten: Systemüberwachung über das GUI

- Alternative zum Prüfen der Server auf „Lebenszeichen“.
- Prüft aus Sicht des Endanwenders Verfügbarkeit, Antwortzeitverhalten und Korrektheit in einem.
- Hoher Grad an Wiederverwendbarkeit von funktionalen Tests.

Entscheidend für den ROI: Wiederverwendung

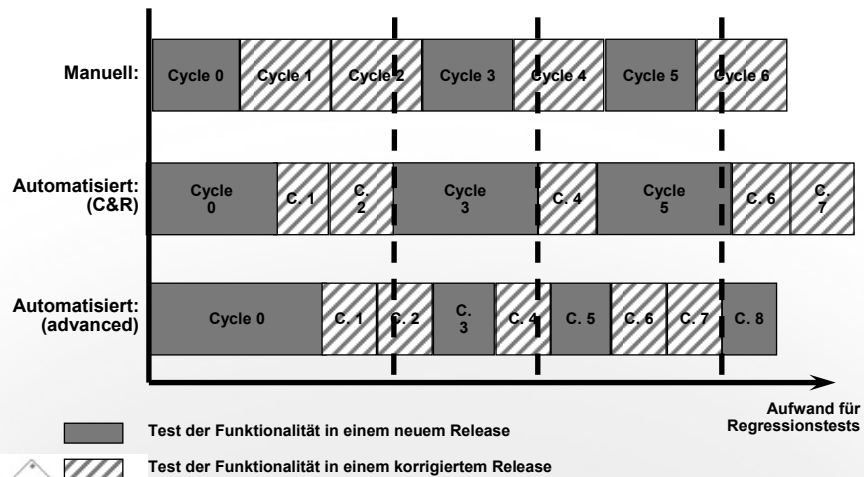
- Der Grad an Wiederverwendung auf allen Ebenen ist das zentrale Kriterium:
 - Wiederverwendung innerhalb der Tests
 - Häufigkeit der Regressionstests
 - Stabilität der Tests bei Systemveränderung
 - Einsatz auf mehreren Plattformen
 - Wiederverwendung der funktionalen Tests, z.B. für Lasttests oder zur Systemüberwachung

-> Wie gut unterstützt das jeweilige Tool diese Arten von Wiederverwendung?

Weitere Vorteile von Automatisierung

- Tests sind schneller und können häufiger komplett durchgeführt werden
 - > kürzere Entwicklungszyklen und frühere Fehlererkennung
 - Höhere Zuverlässigkeit (menschlicher Faktor)
 - Reproduzierbare Ergebnisse
- Σ schnellerer Markteintritt bei höherer Softwarequalität

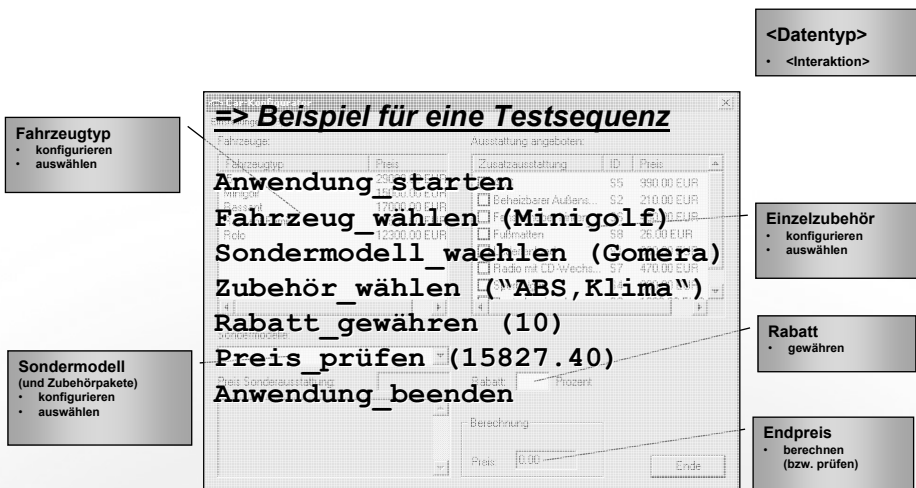
Die Regressions-Falle



www.imbus.de

9

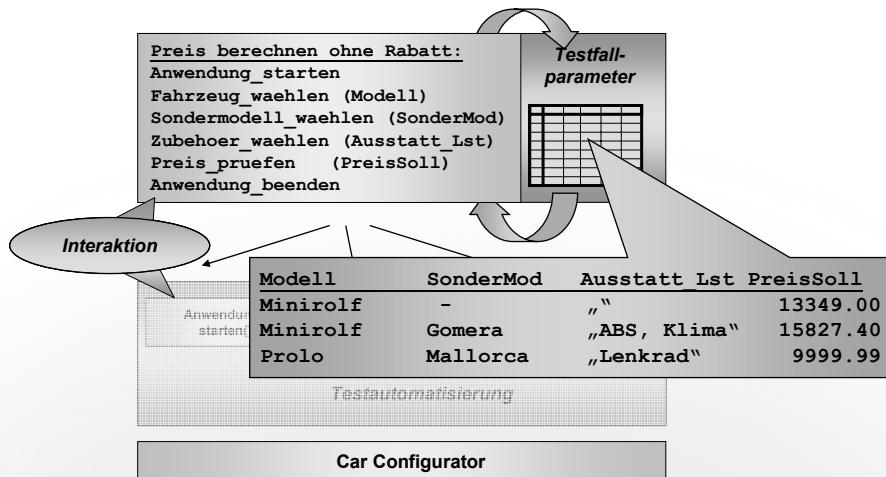
Pattern 4: Beispiel für interaktionsbasierten Testfall



www.imbus.de

10

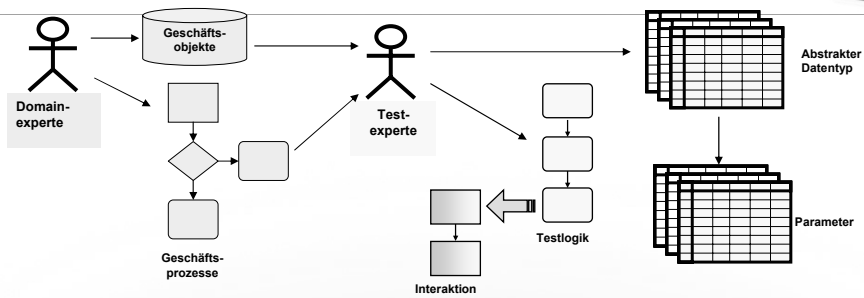
Pattern 4: Vervielfältigung durch Parametrierung



www.imbus.de

11

Pattern 4: Workflow und beteiligte Rollen



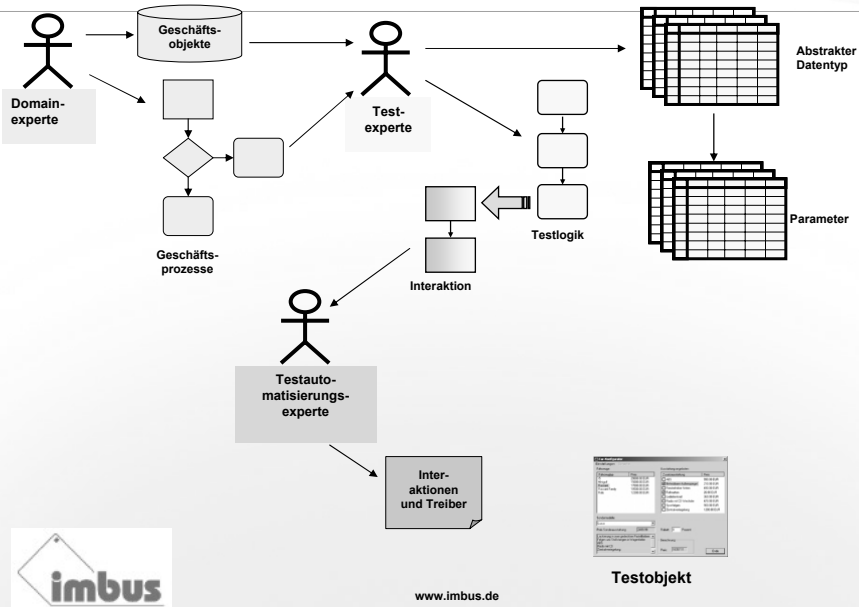
Testobjekt



www.imbus.de

12

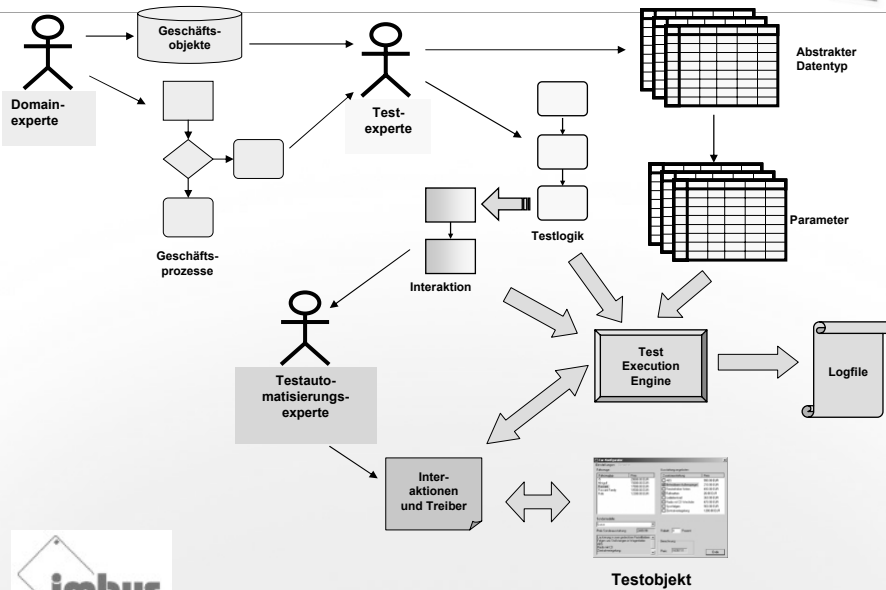
Pattern 4: TestBench – Tests automatisieren



www.imbus.de

13

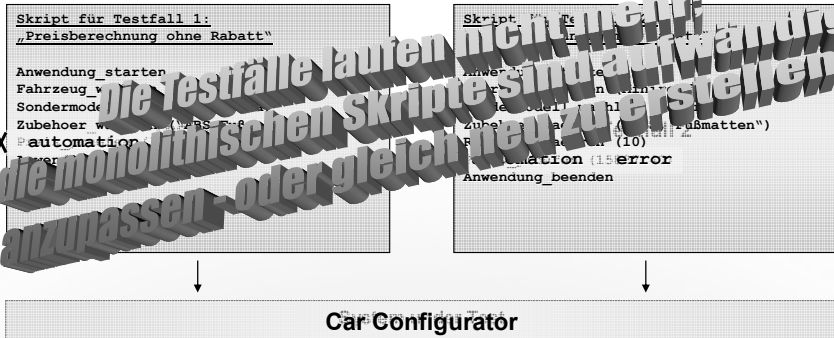
Pattern 4: TestBench – Testautomation starten



www.imbus.de

14

TestBench ~~monolithischer~~¹⁾ Aufbau



Änderung im neuen Release: Der Start der Preisberechnung erfolgt durch „Calc-Button“ statt Drücken der F9-Taste.



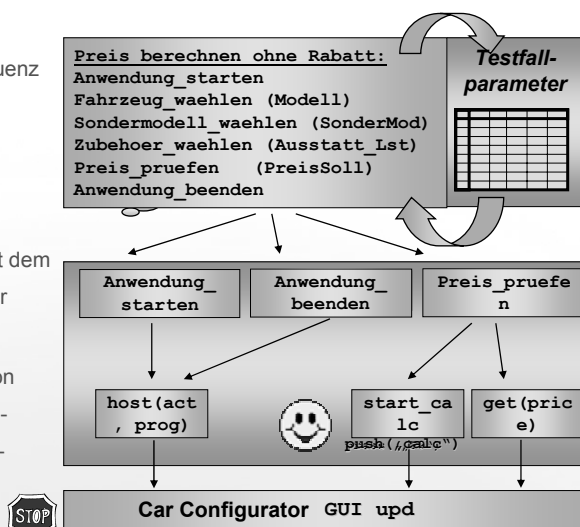
¹⁾monolithisch := „Testfall : Skript = 1 : 1“

www.imbus.de

15

TestBench - modularer Aufbau

- **Abstrakte Testlogik** := Sequenz von *Interaktionen* (Testbaustein, Metawort, Aktionswort, Key Word, ..)
- **Interaktion** := atomare Geschäftsfunktion, Aktion mit dem Testobjekt oder Prüfung einer Reaktion (*fachlich*)
- **Treiber** := get / set Werte von Forms bzw. Feldern und GUI-Controls; Auswahl von Menüeinträgen etc. (*technisch*)



www.imbus.de

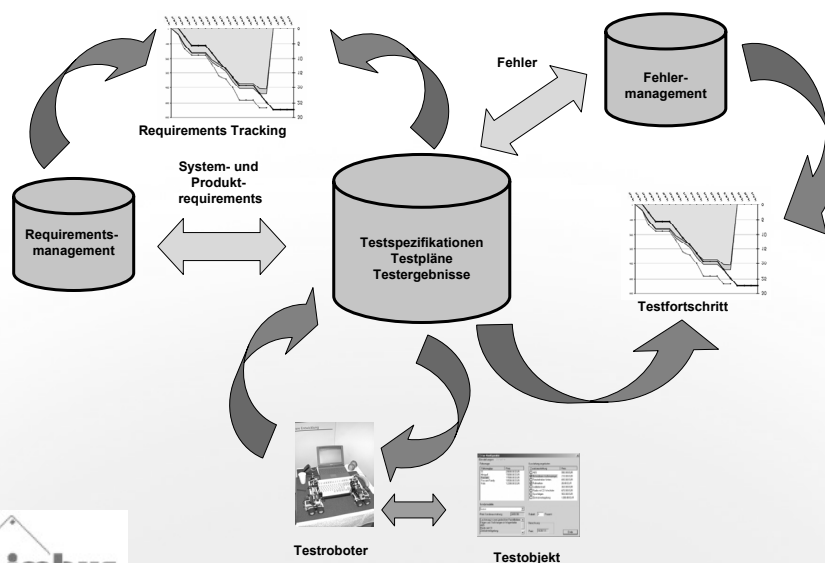
16

Sind Sie reif für die Inset-Testautomation? (auf Basis der Test Maturity Matrix von TPI®)

Key Area	Scale	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Test strategy/ plan			A												D
Life-cycle model			A												
Moment of involvement				A					B			C			D
Estimating and planning	plan, spec, exec											B			
Test specification techniques			A		B										
Static test techniques						A		B							
Metrics	Formal Test-design Techniques						A						C		D
Test tools					A				B			C			
Test environments					A										C
Office environment	Managed & Controlled				A										
Commitment and motivation			A				B								
Test functions and training					A				B				C		
Scope of methodology	Assignment of Budget & Time					A						B			C
Communication				A		B									
Reporting			A			B			C						D
Defect Management			A					B							
Testware management				A				B					C		D
Test process management							B								C
Evaluation (Structured review)	Plan, exec, monitor, adjust								A						
Low-level testing (Build testing)									B			C			

17

Durchgängige und dauerhaft reife Prozesse



www.imbus.de

18

Die Ausgangssituation – in GJ 2004



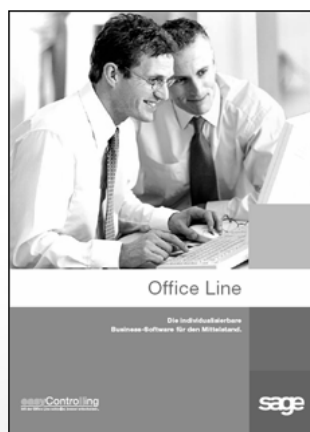
- Ein „Major Release“ p.a.
- 2-3 parallel laufende Versionen in der Wartung
- 3-4 SPs p.a. pro gewarteter Version
- Hotfixes in allen gewarteten Versionen bei Bedarf bei z.B.
 - A-Bugs beim Kunden
 - gesetzliche Änderungen
- Fest definierte Response-Zeiten auf vom Kunden gemeldete Fehler



www.imbus.de



Die Ausgangssituation – in GJ 2004



- Signifikante Abweichung zwischen den Ergebnissen der Kundenzufriedenheitsumfrage und den Sage-Zielen
- Beschwerde- / Eskalationsfälle zu Produktfehlern = 73, Anwenderfehlern = 77
- Sage-interne Fehlerfolgekosten von ca. 600 T€ p.a.
- Belastung von CS, SWE, QCC durch unproduktive Fehlerfolgearbeiten (Bugfixing)
- Hohe Kündigungsrate der Wartungsverträge (versch. Ursachen)



www.imbus.de



Die Vorgehensweise

- Definition der Projektziele
 - Deutliche Reduzierung der Kundenbeschwerden (-30%)
 - Reduktion der Fehlerfolgekosten um >15% (entspr. 64 T€)
 - Definition und Konkretisierung der Strategie
 - Teststrategie und Testplan
 - Einführung strukturierter und dokumentierter Entwicklertests
 - Novellierung des Regressionstests
 - Festlegung verbindlicher Metriken
 - Einführung unterstützender Werkzeuge
 - Umsetzungsstrategie
-
- Begleitung des Projektes durch imbus



www.imbus.de



Das Vorgehen im Detail 1/7

- Die Teststrategie
 - beschreibt welche Bestandteile in welchem Umfang getestet werden,
 - definiert Qualitätsmerkmale und Testziele,
 - gibt einen Überblick über die eingesetzten Methoden, Werkzeuge und Metriken,
 - legt die für den Test eingesetzte Hard- und Software fest und
 - koordiniert, welche Personen oder Teams sich mit welchen Testaufgaben beschäftigen.
- Der Testplan
 - stellt dar, wie der Testprozess in das Gesamtprojekt integriert ist.



www.imbus.de



Das Vorgehen im Detail 2/7

- Die Entwickler- / Low-Level-Tests
 - in ihrer aktuellen Form sind ein neuer Bestandteil der Sage Teststrategie,
 - wurden in bereits bestehende Abläufe und insbesondere Dokumente integriert,
 - werden unter Entwicklungsbedingungen ausgeführt,
 - sind White-Box-Tests gegen die Detailspezifikation,
 - werden im Rahmen der Code-Reviews verifiziert und
 - haben die Ziele: technische Sicherheit des Programms, semantische und syntaktische Korrektheit der Masken, „sauberes“ Layout der Masken sowie die Gewährleistung der Testbarkeit



www.imbus.de



Das Vorgehen im Detail 3/7

- Novellierung Regressionstest bedeutet
 - Erfassen aller implementierten Geschäftsprozesse (GP)
 - Ableiten aller resultierenden Testthemen (TT)
 - Bewertung der Relevanz/Prioritäten der Testthemen auf Basis von:
 - Komplexität
 - Anwendungshäufigkeit
 - Fehlerhäufigkeit
 - Testaufwand (inkl. Vorbereitung)
 - Feststellen der Abdeckung durch aktuelle Testspezifikation
 - Ermitteln der Wiederverwendbarkeit bestehender TestSpec
 - Ableiten des Umsetzungsaufwands und Gegenüberstellung zu „drohenden“ Fehlerfolgekosten
 - Umsetzung obiger Schritte



www.imbus.de



BB-Qualitätskennzahlen Vergleich OL 3.2 –3.3



OL 3.2 (SWE-Projekt)

Anzahl aller anerkannten Originaleinträge,
ohne Doku-Folgefehler

sage

Klassifikation	Anzahl	%-Anteil	Faktor	Gewichtung
A	212	11,95%	4	0,48
B	813	45,83%	2	0,92
C	749	42,22%	1	0,42
Gesamt	1.774	100,00%		1,82

Quantität der Fehler -45%

Verteilung der Qualität von Fehlern + 9%



OL 3.3 (SWE-Projekt)

Anzahl aller anerkannten Originaleinträge,
ohne Doku-Folgefehler

Klassifikation	Anzahl	%-Anteil	Faktor	Gewichtung
A	74	7,63%	4	0,31
B	409	42,16%	2	0,84
C	487	50,21%	1	0,50
Gesamt	970	100,00%		1,65



www.imbus.de

25

BB-Qualitätskennzahlen Vergleich OL 3.2 –3.3



OL 3.2 (CS-Projekt)

Anzahl aller anerkannten Originaleinträge,
mit Kundenbezug (12.04.2004 - 01.09.2004)

sage

Klassifikation	Anzahl	%-Anteil	Faktor	Gewichtung
A	15	8,06%	4	0,32
B	127	68,28%	2	1,37
C	44	23,66%	1	0,24
Gesamt	186	100,00%		1,92

Quantität der Fehler -45%

Verteilung der Qualität von Fehlern + 9%



OL 3.3 CS-Projekt)

Anzahl aller anerkannten Originaleinträge,
mit Kundenbezug (29.03.2005 - 18.08.2005)

Klassifikation	Anzahl	%-Anteil	Faktor	Gewichtung
A	15	11,03%	4	0,44
B	95	69,85%	2	1,40
C	26	19,12%	1	0,19
Gesamt	136	100,00%		2,03



www.imbus.de

26

Die Ist-Situation – GJ 2005 (YTD)

- Kundenzufriedenheit 2005: Befragung läuft aktuell / offen
- Beschwerde- / Eskalationsfälle
 - Produktfehler = 32 (- 56,2%)
 - Anwenderfehler = 36 (- 53,2%)
- Fehler mit Kundenbezug massiv zurückgegangen s.o.
 - (- 26,9 % => ca. -100.000 € Folgekosten p.a.)
- SWE-Zeit für neue Features statt Bugfixprozess
 - SWE von 953 für V 3.3 auf 1.306 für V 3.4
(+ 37% / ca. 10% externe SWE)
- Reduzierung der Testzeiten trotz größerem Funktionsumfang
 - QA von 940 für V 3.3 auf 832 V 3.4
(- 11,5% => ca. -30.000 € externe Kosten p.a.)
- Kündigungsrate der Wartungsverträge offen

-
- **ROI bereits im 1. Jahr nach Einführung erreicht!**
(kalkulierter Zinsfuß nach 2 Jahren > 40%)



www.imbus.de

sage

Weiterführende Informationen

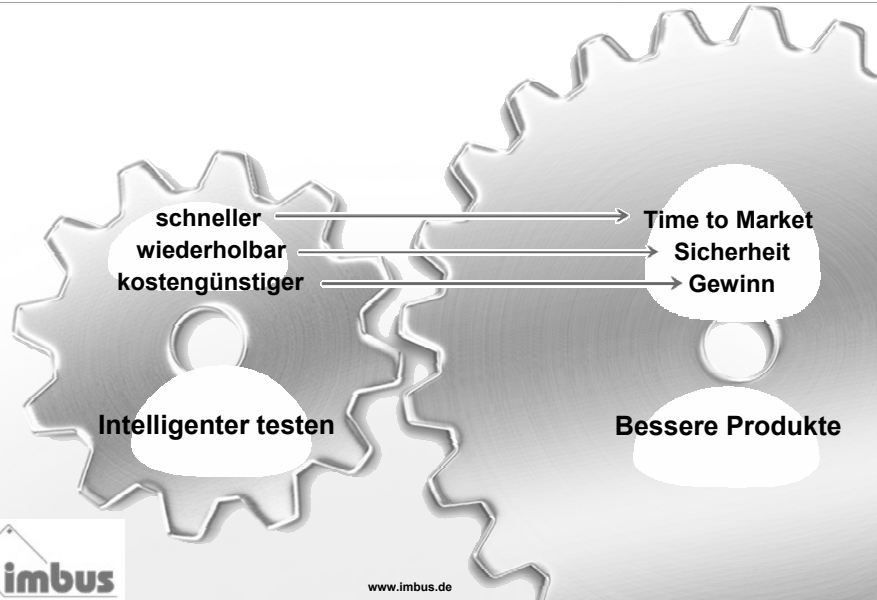
- “The Road to successful Test Automation - or - 10 Pitfalls Testmanagers should be aware of”
http://www.imbus.de/download/papers/pitfalls_doc.zip and
http://www.imbus.de/download/papers/pitfalls_ppt.zip
- iTOM Homepage: White Paper, Demo für “Northwind Application”,
Artikel im MS Access Inside Magazin
www.imbus.de/produkte/iTOM.html
- imbus TestBenchHomepage: AVIs, Flyer, ..
www.imbus.de/produkte/testbench.html
- Kontakt: info@imbus.de, Frank.Schmeissner@imbus.de



www.imbus.de

28

Ihr Nutzen



Kontakt

Hauptsitz

imbus AG
Kleinseebacher Straße 9
91096 Möhrendorf
Tel: 0 91 31 / 75 18-0
Fax: 0 91 31 / 75 18-50

Geschäftsstellen

imbus AG
Unter der Linde 16
80939 München
Tel: 0 89 / 3 21 99 09-0
Fax: 0 89 / 3 21 99 09-50

imbus Rhein-Main GmbH
Zanggasse 6
65719 Hofheim
Tel: 0 61 92 / 9 21 92-0
Fax: 0 61 92 / 9 21 92-50


info@imbus.de
Frank.Schmeissner@imbus.de



www.imbus.de

30

QFS
Quality First Software



Verfügbare Automatisierungstools

- Windows**
 - *QuickTest Professional* (von Mercury, früher WinRunner, *XDE Functional Tester* (von IBM Rational, früher Robot), *Silktest* (von Segue), *TestPartner*, *QARun* (von Compuware) und div. andere.
- Unix**
 - *XRunner* (von Mercury), *XDE Functional Tester*, *Silktest*, *Suish* (von Froglogic für QT und XView).
- Web**
 - Diverse kommerzielle Capture/Replay Tools in allen Preiskategorien, ebenso diverse Open Source Tools, allerdings Skript basiert ohne Aufnahmemöglichkeit
- Java/Swing**
 - Open Source: Abbot, JFCUnit, Marathon, entwicklerlastig, ohne oder mit stark eingeschränkten Aufnahmemöglichkeiten
 - Windows basierte Testtools bieten inzwischen Java Plugins für Swing
 - *qftestJUI* ist auf Swing spezialisiert
- Java/SWT**
 - Abbot (nur harte Events mit kritischem Timing), Windows basierte Tools mit Einschränkungen bei Objekterkennung.
 - *qftestJUI* demnächst

© 2005 Quality First Software GmbH 18.01.06 15

QFS
Quality First Software



Vielen Dank für Ihre Aufmerksamkeit!

Fragen?

© 2005 Quality First Software GmbH 18.01.06 16