



BITMARCK®



Automatisierte GUI Tests in fachlichen Teststufen

07.09.2011 – Patrick Möller



Inhaltsangabe

Vorstellung und Situation BITMARCK

- BITMARCK und iskv_21c

Testautomatisierung - warum?

- Teststufen bei BITMARCK
- Testautomatisierung mit fachlicher Spezifikation

Testautomatisierung – wie?

- Toolauswahl
- Testteams

Erfahrungen / Ergebnisse





Vorstellung BITMARCK und iskv_21c





BITMARCK

- 1.200 Mitarbeiter
- 250 Mio Jahresumsatz
- Full-Service IT-Dienstleister im Gesundheitswesen (GKV)
- Gesellschafter sind gesetzliche Krankenversicherer
- Programmierung, Consulting und Betrieb der Standard-Anwendungen
 - ISKV-Basissystem
 - iskv_21c (im Rollout – bei 27 Krankenkassen produktiv)
 - DAKIDIS – Standardanwendung der DAK





iskv_21c

iskv_21c bietet als Komplettpaket ein Instrument, das den Anforderungen eines modernen Kassenmanagements entspricht.

Damit verbundene Wettbewerbsvorteile werden erreicht durch

- dispositive Datenhaltung
- ineinander greifende Querschnittsfunktionen
- Technik, Software, Service und Beratung aus einer Hand

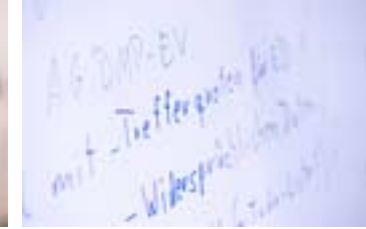
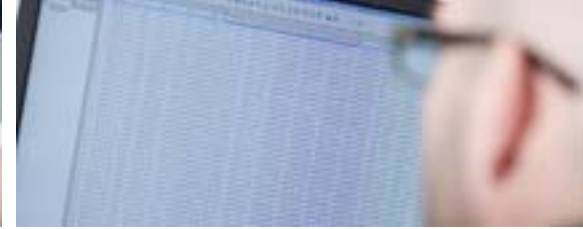
- Client/Server Architektur (Java)
- SWING Desktop Client





Testautomatisierung – warum ?





Teststufen (1/2)

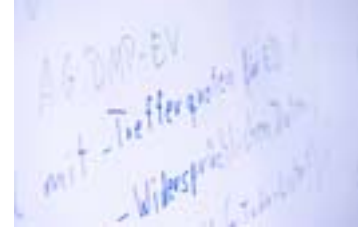
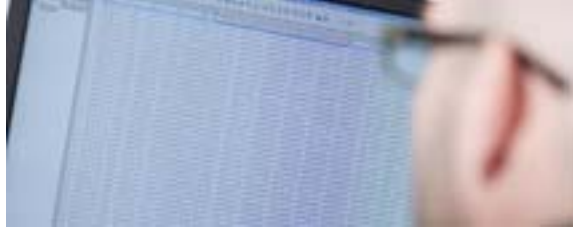
Entwicklertest

- JUnit (Server und Standalone-Tests)
- Continuous Integration
- Tester – Entwickler der Fachkomponente

Produkttest

- Erste integrative Teststufe
- Exploratives Testen – keine Automatisierung
- (interner) Tester – fachlicher Analyse der Fachkomponente





Teststufen (2/2)

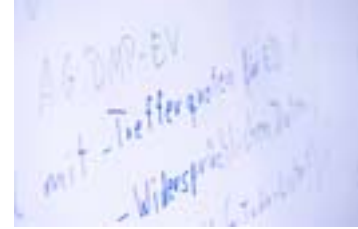
Systemtest

- Test durch externe (Krankenkassen-)Mitarbeiter
- Exploratives Testen
- Erste Automatisierung durch Marathon (Open Source)

Pilottest / Pilotanwendung

- Test bei Kunden
- Pilottest in Testumgebung, Pilotanwendung in Produktion





Aktuelle Situation in den Teststufen

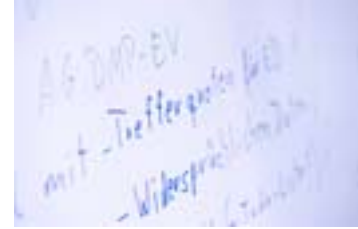
Entwicklertest

- Regressionstest nur auf Basis JUnit / Hudson (CI)

Produkt-/Systemtest

- Überwiegend fachliche Tester
- Manuelle Oberflächentest
- Manuelle Batchtests – Ergebnisvergleich in den Oberflächen
- Exploratives Testen mit unterschiedlichen Testergebnissen
- Keine Regressionstests





BITMARCK®

Ziele für Verbesserung in Teststufen

Schwerpunkt Produkt-/Systemtest

- Einführung von Testfällen (manuell/automatisiert)
 - Möglichst fachliche Beschreibungssprache für Automatisierung
 - Abstraktion von Programmierung
 - Wiederverwendung von Testfällen (Module)
- Unterstützung Oberflächentest durch Automation

BITMARCK führt agile Methoden ein

- Regressionstests





Testautomatisierung – wie ?





Toolauswahl

Anforderungen

- Möglichst leichte (fachliche) Beschreibung
- Wiederverwendung von Testfällen
- Spezifikationsmöglichkeit vor Entwicklung der GUI (nicht ausschließlich Capture/Replay)
- Unterstützung SWING GUIs und eigene Erweiterungen von SWING
- Möglichkeit CI Unterstützung





Toolauswahl

Evaluierte Tools

- QFTest, Silktest, GUIDancer
- Auswahl an Hand von Entscheidungsmatrix
- Genaue Definition der Anforderungen und Gewichtung durch Punktesystem





Testspezifikation

- Manuelle Tests werden in fachlichen Schritten erstellt
- Automatisierte Tests wurden zunächst durch einheitliche Sprache erstellt (SET, GET, VER)
 - Umstellung auf screenshotbasierte Keywords
- Einteilung in Sockettestfälle für Regressionstests





Testautomatisierungsteam

- Spezifizierte Testfälle werden technisch gemapped
- Implementierung der fachlichen Spezifikation
- Bereitstellung von Reports
- Unterstützung beim Testdatenmanagement (z.B. Rücksetzung der DB)
- Kontrolle und Wartung der Ablaufumgebung (Remote Test Execution)



Erfahrungen / Ergebnisse

- Ansatz Testfälle direkt in GUI-Dancer/Jubula zu spezifizieren durch Fachteams abgelehnt
 - Bereitstellung einheitlicher Spezifikationen zur Beschreibung der (automatisierten) Testfälle
 - Entwicklung Test-Designer – Unterstützung der Modul/Wiederverwendung auf fachlicher Ebene
- Eigenes Testautomatisierungsteam
 - Unterstützung QM
 - Implementierung und Ergebnisbereitstellung
 - Erstellung vom Test-Designer zur fachlichen Abstraktion
- Live Demo – Test-Designer