

WINCOR NIXDORF

Test nichtfunktionaler Anforderungen in der Praxis am Beispiel einer netzzentrierten Anwendung

Februar 2011

**WINCOR
NIXDORF**
EXPERIENCE MEETS VISION.

Test nichtfunktionaler Anforderungen Agenda

1. Architektur und Produkte
2. Nichtfunktionale Anforderungen
3. Testarten
4. Testdurchführung
5. Werkzeuge
6. Testautomatisierung

**Architektur und
Produkte**

Nichtfunktionale
Anforderungen

Testarten

Testdurchführung

Werkzeuge

Testautomatisierung

PC/E Server-Architektur

- Client-Server Architektur
- offene, netzzentrierte Multikanal-Architektur, die modular und service-orientiert aufgebaut ist

PC/E Self-Service

- SB-bezogene Dienste wie zum Beispiel die Kontoabfrage oder Geldausgabe

PC/E Payment Server

- Serverbasierte Lösung für Autorisierung, Switching und Routing von zahlungsbezogenen Transaktionen



**Architektur und
Produkte**

Nichtfunktionale
Anforderungen

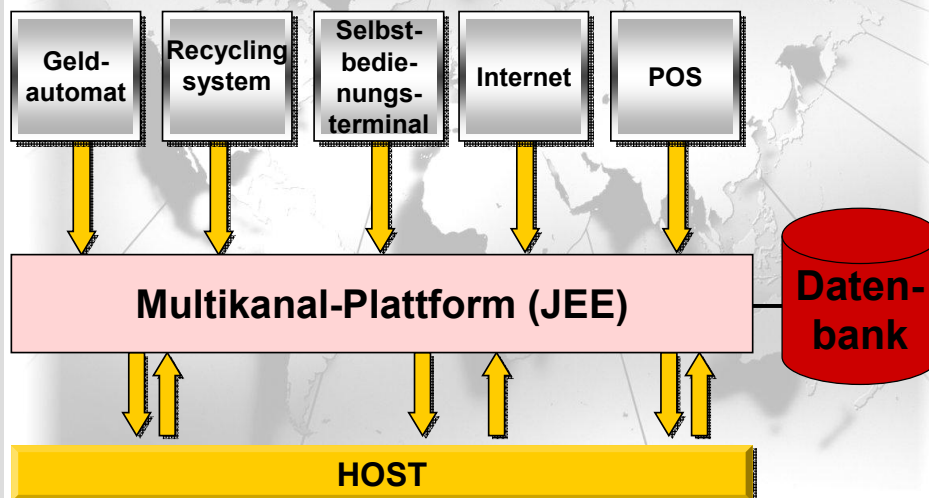
Testarten

Testdurchführung

Werkzeuge

Testautomatisierung

- Client-Server Architektur
- offene, netzzentrierte Multikanal-Architektur, die modular und service-orientiert aufgebaut ist



Definition

Eine nicht funktionale Anforderung beschreibt Attribute des Systemverhaltens, die sich nicht direkt auf die Funktionalität des Systems beziehen, sondern darauf, mit welcher Qualität das System diese Funktion erbringt.

Beispiele laut ISO 9126

- Zuverlässigkeit
- Benutzbarkeit
- Effizienz
- Änderbarkeit
- Übertragbarkeit

Anforderung	Testart
Zuverlässigkeit	Failover Test Server Ausfalltest Datenbank Ausfalltest Stresstest Dauertest
Effizienz	Performancetest
Übertragbarkeit (Skalierbarkeit)	Performancetest

Architektur und
Produkte

Nichtfunktionale
Anforderungen

Testarten

Testdurchführung

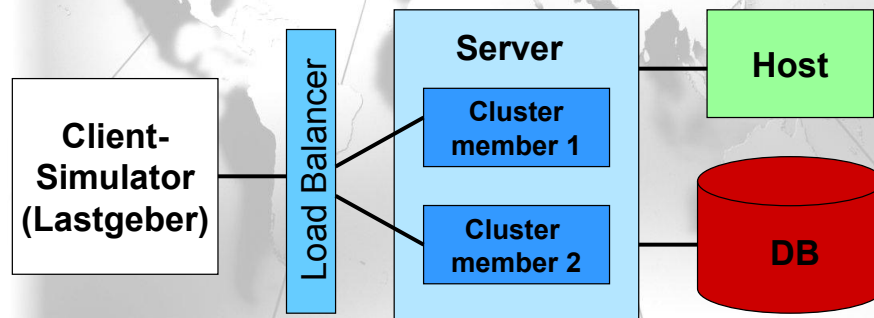
Werkzeuge

Testautomatisierung

Ausfalltest = Test auf Robustheit gegenüber Ausfall einzelner Komponenten (z.B. Server- oder Datenbank-Ausfall) → Angemessenes Verhalten

Failovertest = Test auf Ausfall einzelner Komponenten und Übernahme durch redundante Komponenten (z.B. Clustermember-Ausfall)

Stresstest = Beobachten des Systemverhaltens bei Überlastung



Architektur und
Produkte

Nichtfunktionale
Anforderungen

Testarten

Testdurchführung

Werkzeuge

Testautomatisierung

Dauertest = Test der Stabilität/Zuverlässigkeit im Dauerbetrieb

- Memory Leaks finden
- Monitoring von CPU-Auslastung und Speicherverbrauch
- Nicht an der Lastgrenze, aber ausgelastet

Realistisches Szenario vs. Analysierbares Szenario

- *Beispiel 1: Realistische Kundenumgebung*
 - Einzahler, Auszahler, Kontoauszugsdrucker, Benutzerkonsole, Datenimport laufen parallel
- *Beispiel 2: Vereinfachte Umgebung für ein neues Produkt (PC/E Payment Server)*
 - eine einzelne Transaktion
 - Szenario wird sukzessive komplexer

Architektur und
Produkte

Nichtfunktionale
Anforderungen

Testarten

Testdurchführung

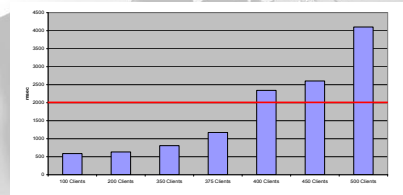
Werkzeuge

Testautomatisierung

Performancetest: Softwaretest mit dem eine zu erwartende, auch extreme Last auf dem laufenden System erzeugt und das Verhalten desselbigen beobachtet und untersucht wird.

- Antwortzeiten
- Transaktionen pro Sekunde
- CPU-Auslastung
- Speicher-Auslastung

LoadTest: Abhängig von steigender Systemlast wird das Systemverhalten untersucht



Volumen-/Massentest: Beobachtung des Systemverhaltens in Abhängigkeit von der Datenmenge

Architektur und
Produkte

Nichtfunktionale
Anforderungen

Testarten

Testdurchführung

Werkzeuge

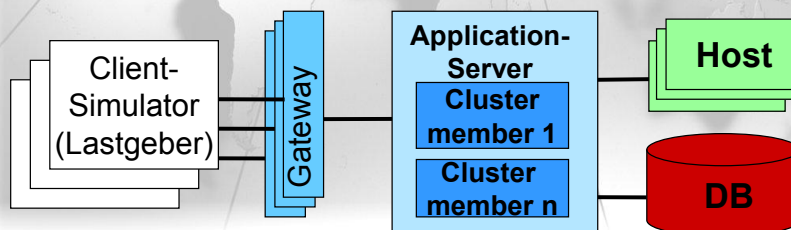
Testautomatisierung

Beispiel: Test des PC/E Payment Servers

- alle übrigen Komponenten müssen "unendlich" schnell sein → dürfen nicht Engpass werden
- Leistungsgrenze des Servers muss erreicht werden

Umgebung einrichten

- Optimierung der Datenbank
- Ausreichende Verfügbarkeit von Lastgebern und Simulatoren
- Optimale Konfiguration des Anwendungsservers
 - Abhängigkeiten zwischen den Parametern -> sehr zeitaufwändig, „Try and Error“



Architektur und
Produkte

Nichtfunktionale
Anforderungen

Testarten

Testdurchführung

Werkzeuge

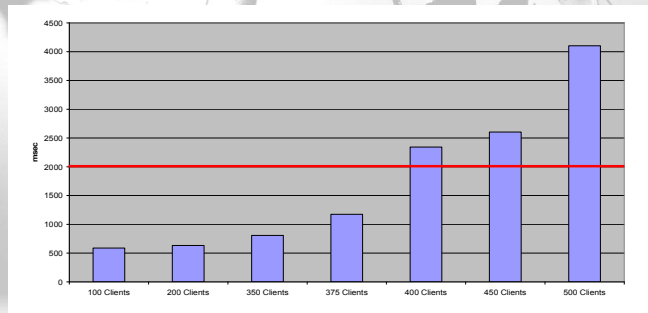
Testautomatisierung

Datenbank

- Zuwachs an Datenvolumen → linear hochrechenbar
- Einfache Analyse: Anzahl der Datenbankzugriffe (insert, updates, selects)

Server: LoadTest

- Beispiel PC/E Self Service: *Wie viele Clients können parallel betrieben werden, um die Antwortzeiten (z.B. 2 Sekunden) einzuhalten?* → nicht linear hochzurechnen



Architektur und
Produkte

Nichtfunktionale
Anforderungen

Testarten

Testdurchführung

Werkzeuge

Testautomatisierung

Motivation:

- Verfügbare Hardware ist in der Regel kleiner als die Hardware des größten Kunden.
- Ergebnisse lassen sich nicht linear hochrechnen.

Ziel:

- Beweis der Skalierbarkeit
- Hardwareempfehlungen für Kundenprojekte (Sizing)

Vorgehen/Best Practices:

- Beispiel PC/E Payment Server: *Hardware und Anzahl der Clustermember wird sukzessive erhöht bis 1000 TPS (Transaktionen pro Sekunde) erreicht werden*
- Messung in großen Szenarien finden in der Regel einmalig statt und anschließend werden Veränderungen zur Vorgängerversion überwacht

Architektur und
Produkte

Nichtfunktionale
Anforderungen

Testarten

Testdurchführung

Werkzeuge

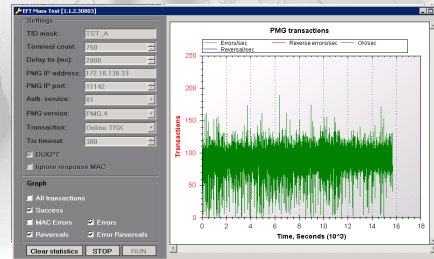
Testautomatisierung

Open Source: JMeter

- Eingeschränkter Funktionsumfang (http-Schnittstelle)
- Transaktion kann aufgenommen und bearbeitet werden → einfache Handhabung, kein Wissen über das Protokoll notwendig

Eigene Entwicklungen

- individuelle Anpassungen möglich
- Standardtools sind nicht vorhanden, z.B. WNSP Protokoll von Wincor Nixdorf
- Standardfunktionen müssen extra implementiert werden (z.B. Berichte)



Architektur und
Produkte

Nichtfunktionale
Anforderungen

Testarten

Testdurchführung

Werkzeuge

Testautomatisierung

Kommerzielle Werkzeuge

- kostenpflichtig
- decken nur Standardprotokolle ab
- Generierung von Testdaten, z.B. Karten und Terminals
- Kombination aus Transaktionen ist individuell konfigurierbar
- Funktionen zum Logging und zur Auswertung der Tests

z.B. HP Quality Center, Iliad T3

Architektur und
Produkte

Nichtfunktionale
Anforderungen

Testarten

Testdurchführung

Werkzeuge

Testautomatisierung

Testablauf

- Transaktionsvolumen erfordert immer Automatisierung
- Reproduzierbarkeit wird verbessert durch Automatisierung

Testvorbereitung und -auswertung

- Gleiche Startbedingungen automatisiert herstellen
→ vermeidet Fehler
- Sammeln der Ergebnisse und Aufbereiten der
Auswertung
→ spart Zeit (wiederholt)

Manuelle Tätigkeiten

- Optimierung der Umgebung, Problem- und
Ergebnisanalyse, Testentwurf

Vielen Dank für die Aufmerksamkeit