



The essence of modern software engineering

OOP 2009

Software meets Business

# Testen von Prozessszenarien

## Ein Beispiel aus der Praxis

Michael Blech

***BwFuhrparkService***

Gregor Hermann

**ex|Xcellent**  
solutions



## **x| Einordnung des Testvorgehens**

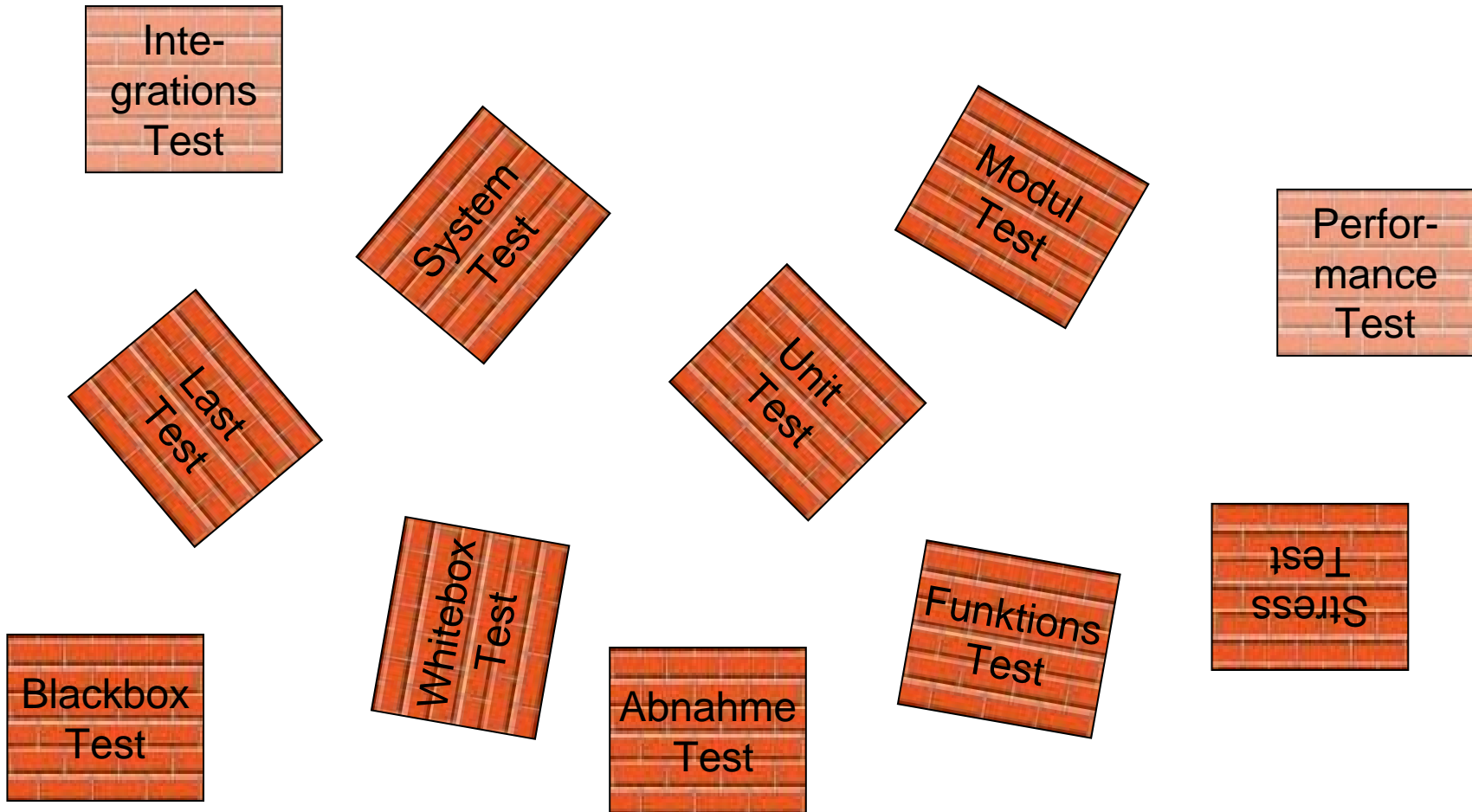
## **x| Unser Fallbeispiel**

- | Das Business im Überblick
- | Der Prozess: Fakturierung und Reklamationsbearbeitung
- | Der Systemüberblick
- | Die Testherausforderung

## **x| Die Hürden - oder was es alles zu beachten gibt**

- | Nicht zuviel und nicht zu wenig – die richtige Menge an Tests
- | Input-Daten erzeugen und aktuell halten
- | Die Abhängigkeit von der Zeit – Durchlauf im Zeitraffer
- | Die Datenmanipulationen
- | Referenz-Daten erzeugen, persistieren, aktualisieren und vergleichen
- | Tools und Integration

## **x| Das Resumée**



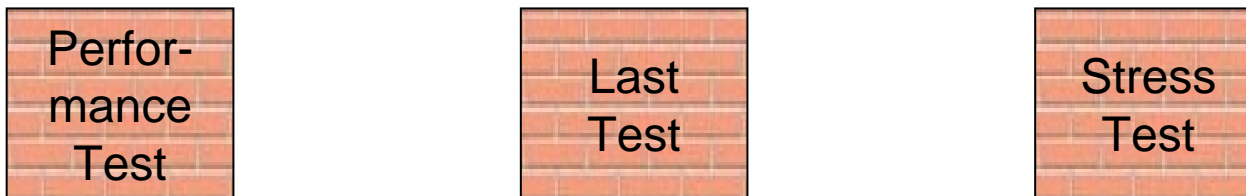
## x| Funktion auf unterschiedlichen Ebenen

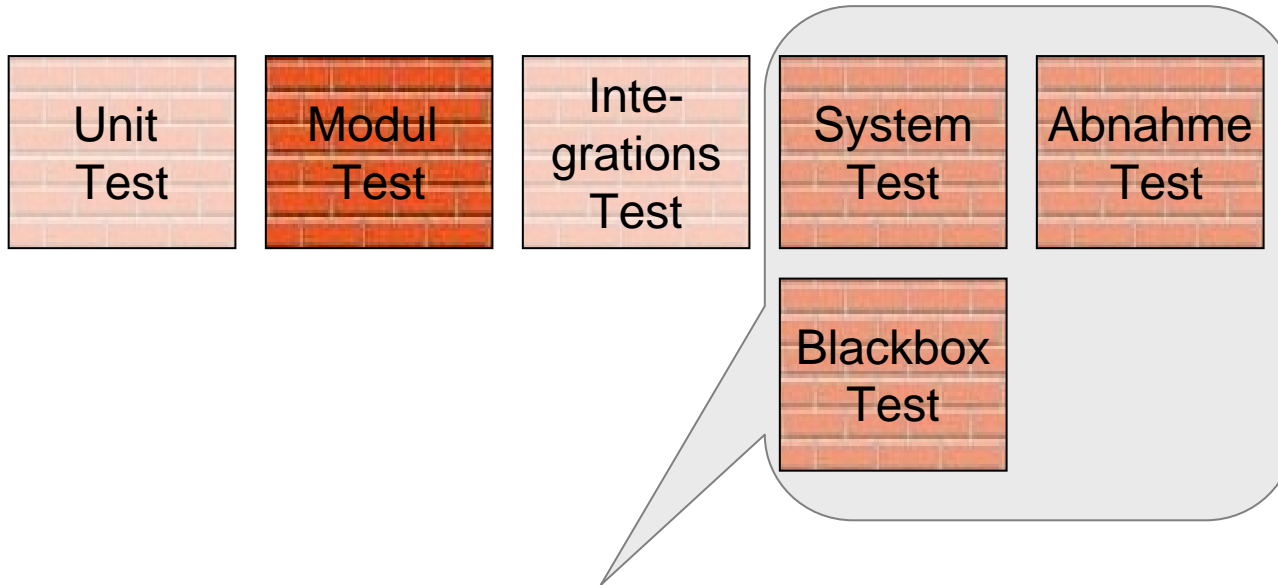


## x| Sicht auf das zu testende System



## x| Verhalten





**x|** Zur End-to-End Abdeckung von **Prozessszenarien** mit **Zeitabhängigkeit** und **zeitlichem Verlauf**

## Die BundeswehrFuhrparkService (BwFPS) seit 2002 Mobilitätsdienstleister der Bundeswehr

### x| Mission

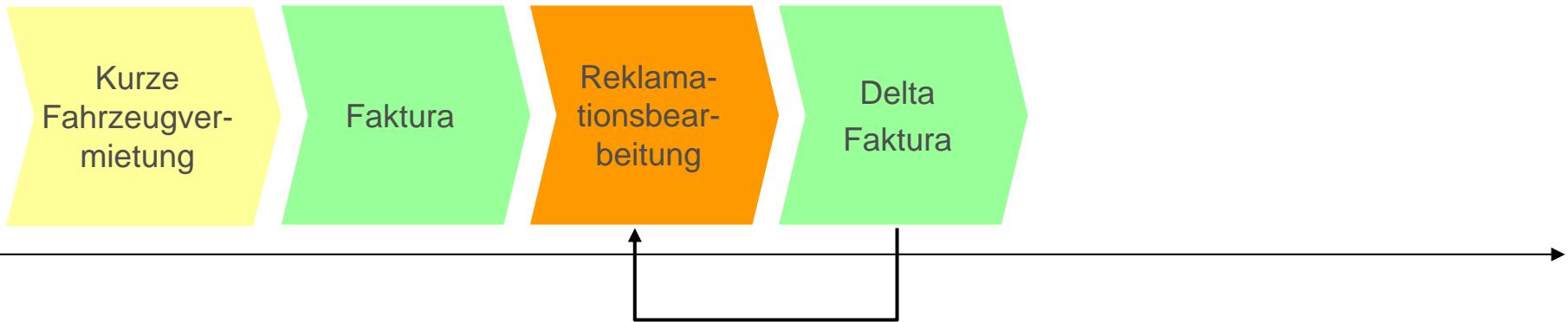
- | **Kostenreduktion** im Fuhrparkmanagement der Bundeswehr
- | **Modernisierung** des Fuhrparks der Bundeswehr

### x| Leistungsspektrum

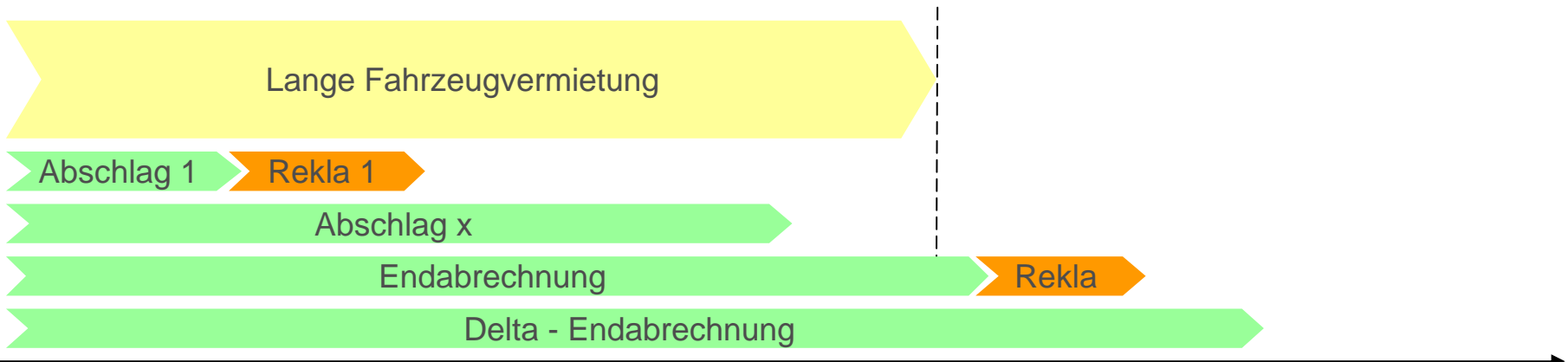
- | **Vermietung von Fahrzeugen inkl. Dienstleistungen** in unterschiedlichsten Produkten
  - Langzeitmiete – analog Leasing
  - Kurzzeitmiete – ähnlich Mietwagen
  - Carsharing
- | **Vermietung in unterschiedlichen Nutzungsszenarien**
  - Grundbedarf
  - Ausbildung/Übung
  - Einsatz



## Szenario 1: Einmalige Abrechnungen



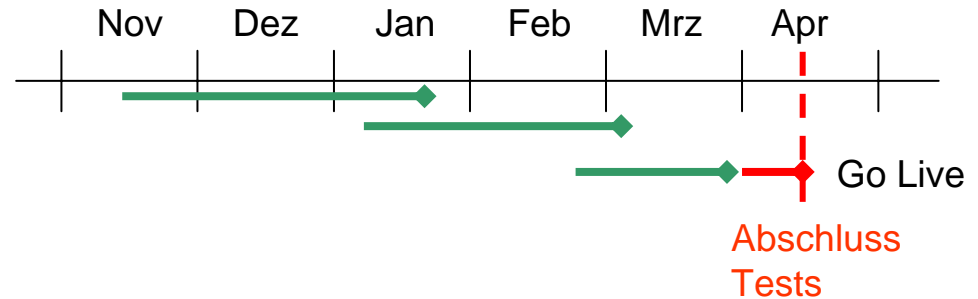
## Szenario 2: Abschlagsrechnungen mit Endabrechnung



## Soll

**x|** Ein bestehendes Fakturasytem soll schrittweise abgelöst werden

**x|** Enger zeitlicher Rahmen für die erste Version



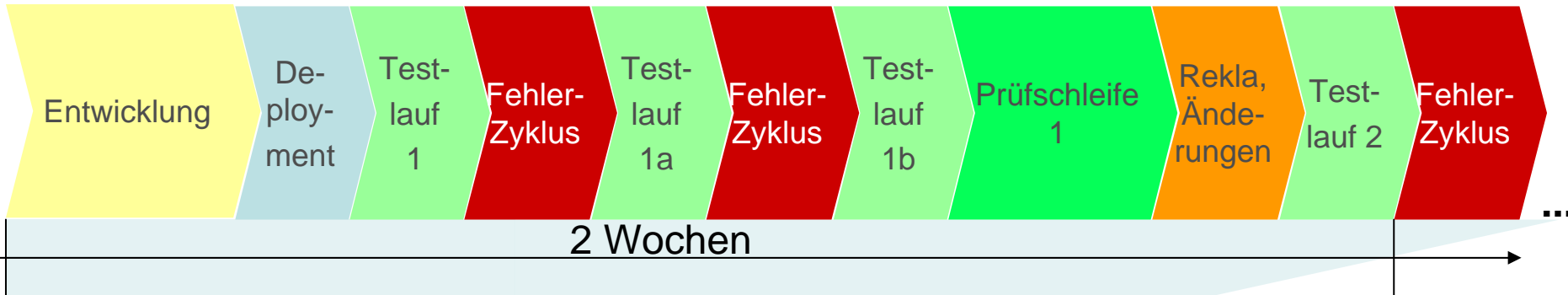
## Ist

**x|** Dauer für den Test einer neuen Faktura-Version ca. 4 Wochen

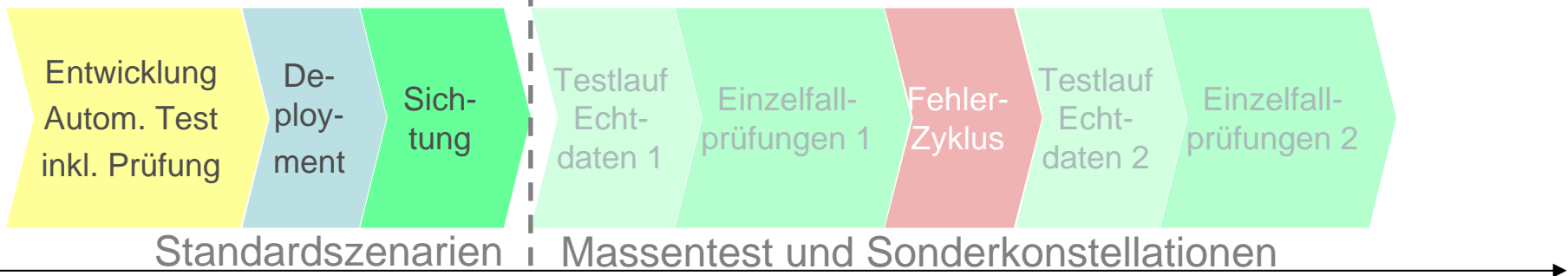
- | Testlauf vorbereiten mit Echtdaten
- | N-Testiterationen auf Grund von erkannten Fehlern
- | Fallkonstellationen aus Echtdaten identifizieren
- | Einzelfallprüfungen

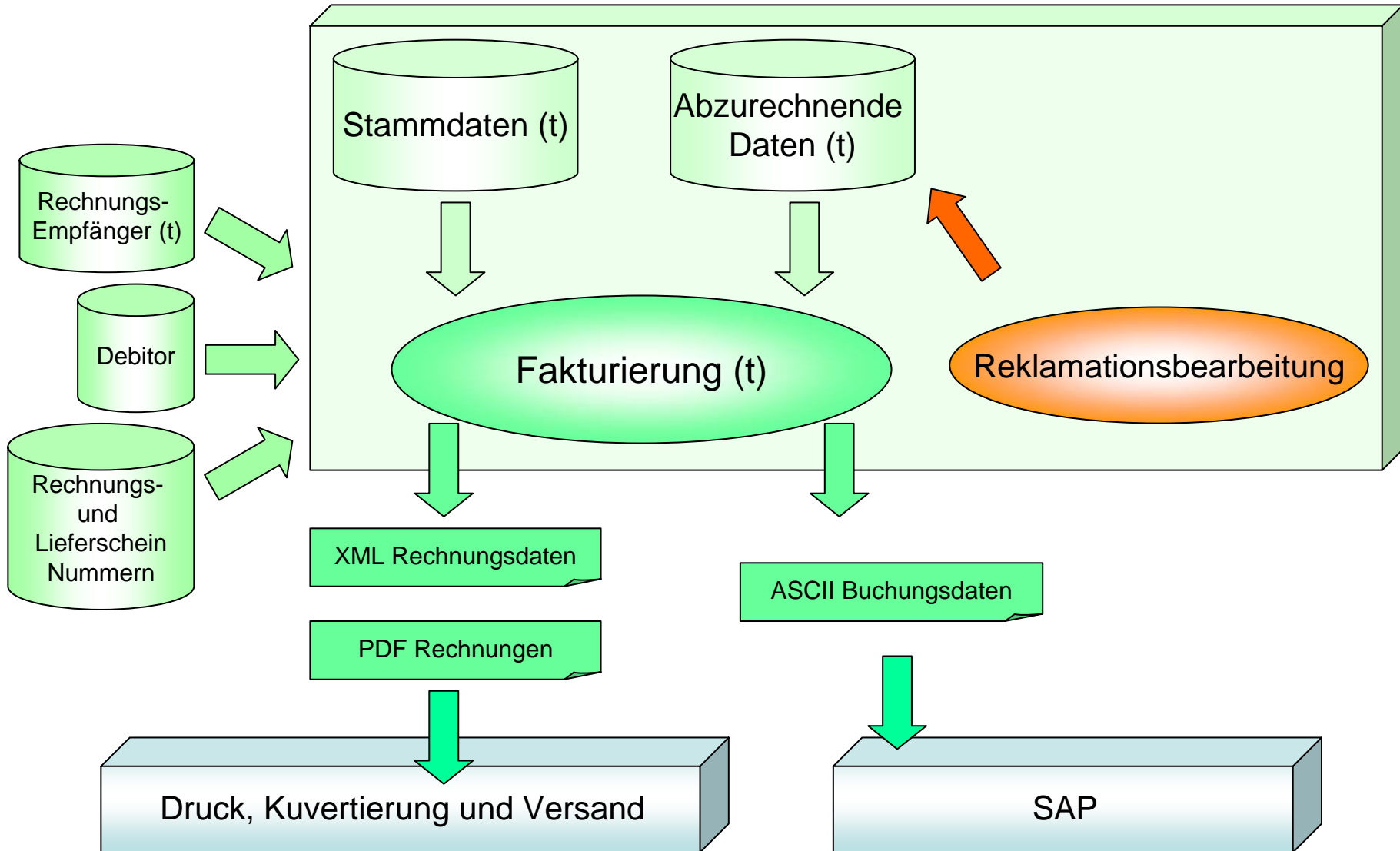
**x|** Sehr dynamische fachliche Anforderungen → häufiger Anpassungsbedarf

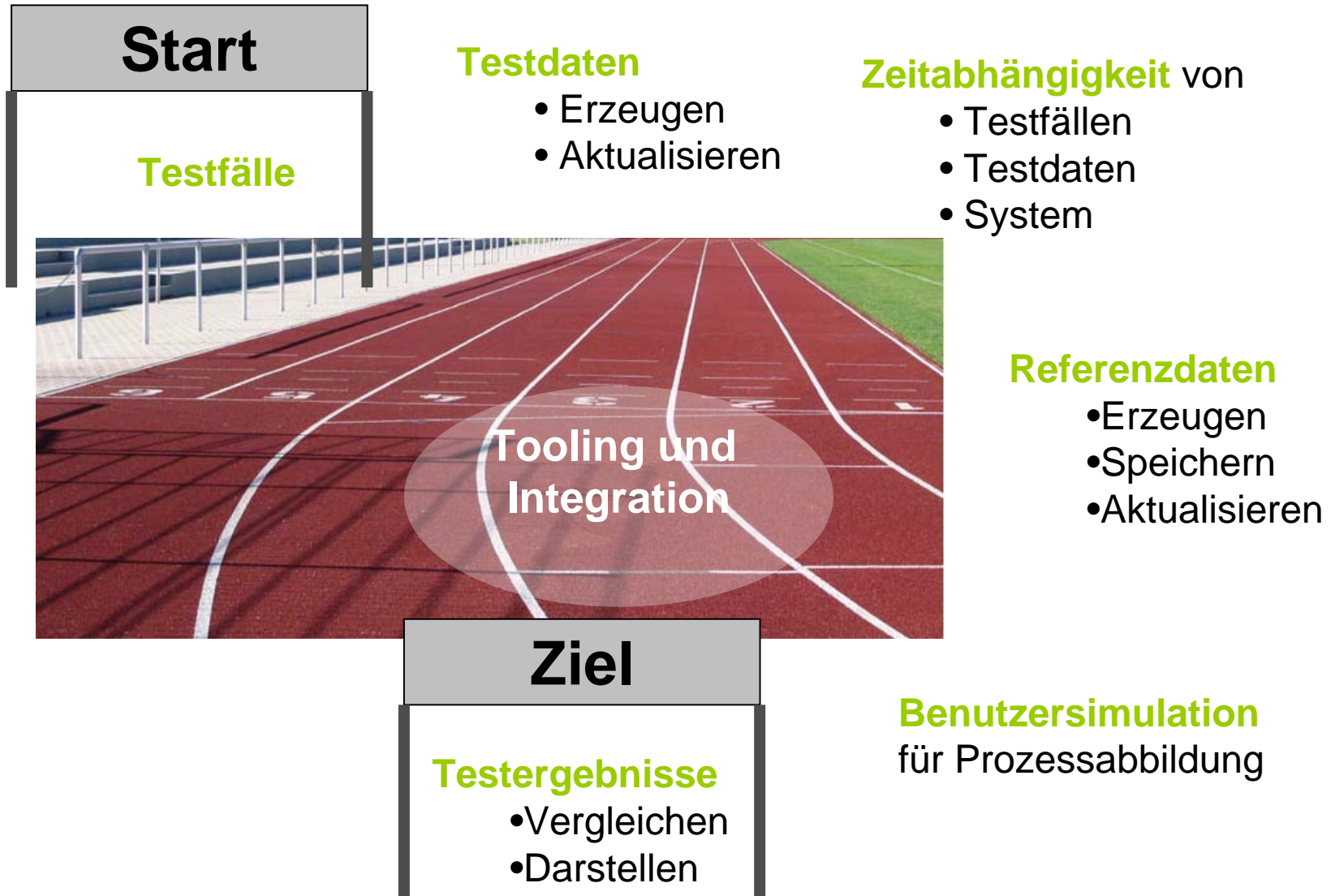
## Ausgangssituation Testprozess



## Ziel

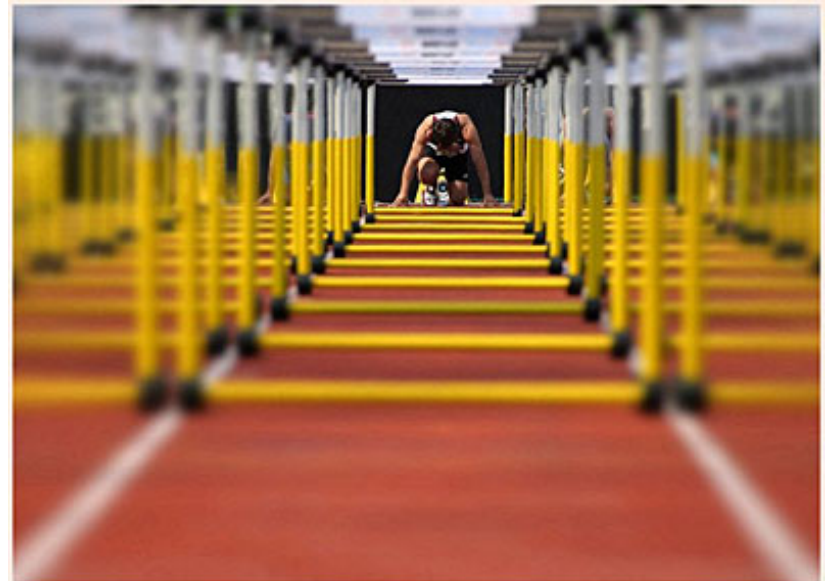






## Welche und wieviele Testfälle?

- x|** Kern: Prozessablauf unterschiedlichster Szenarien, nicht die einzelnen Varianten innerhalb der Szenarien
- x|** 80:20 gilt auch hier
- x|** In unserem Fall 25 Szenarien überschaubarer Komplexität



## Dokumentation der Testfälle

- xl** Übersicht in Excel
- xl** Einzelne, gruppierte Fakturaaspekte auf der Y-Achse
- xl** Benannte Szenarien auf der X-Achse
- xl** Detaillierte Beschreibung der Szenarien in separater Datei

	A	B	C	J	K	L	M	N
1	<b>Fakturierungs-Testfälle ohne Preisber</b>							
2	Testfall-Nummer (Fakturatest-#, Abrufnummer A99-99-99-00xx)			<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>
3	Name		Chauffeurabruf mit Lücke	Serie Hülle	Serie Einzelfakturierung	Sperre		Überbrückung LHM Schaden fremdverschuldet
4	Beschreibung		Chauffeurabruf mit Lücke und Krankheit	Detaildisposition / Serienabruf mit Fakturierung der Hülle	Komplettabruf Fahrzeug mit Zubehörserie	Gesperter Abruf		Überbrückung LZM mit Grund "Schaden fremdverschuldet"
5	Lieferscheintyp			Einzellieferschein am 1.2.	Einzellieferschein am 1.2.	Einzellieferschein am 1.2.	-	-
6	Nutzung	Normal (komplette Nutzung)	Ok	x	x	x	x	x
7		Nur ausgegeben (bei Abschlag)	Ok					
9		Spätere Abholung	Ok					
11		Frühere Rückgabe	Ok					
12		Kulanz am Beginn	Ok					
13		Kulanz am Ende	Ok					
14		Überzug	Ok					
15		Überzug innerhalb Karenzzeit	Ok	x				
16		Nichtabholung	Ok					
18		Objekttausch	Ok					
19		Objekttausch mit Lücke	Ok					
20		Upgrade	Ok					
21		Downsize	Ok					
22		Nicht geleisteter Hol-Bring-Service	Ok					
23		externes Fahrzeug	Ok					
24		Altfahrzeug	Ok					
25		Neufahrzeug	Ok		x			
27		externer Chauffeur	Ok		x			
28		beigestellter Chauffeur	Ok		x			
29		Chauffeur mit Krankheit	Ok	x				
30		Chauffeurtausch mit Lücke	Ok	x				
31	Serienabrufe	Mischung Leistung als Serie/Leistung ohne Serie	Ok		x	x		
32		Detaildispositionen	Ok		x			
33		Berechnung der Einzelelemente	Ok			x		
34		Serienabruf > 45 Tage	Ok					
35		Serienabruf < 45 Tage	Ok		x	x		
36	Leistungsvertrag	HU	Ok	x	x	x	x	x

## Welche Daten sind notwendig?

### x| Stammdaten

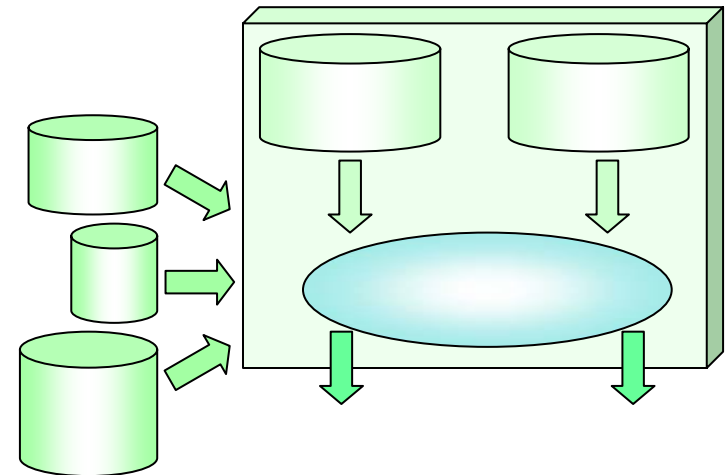
Als Basis für die Bewegungsdaten

### x| Bewegungsdaten

Fahrzeugvermietungszenarien

### x| Externe prozessspezifische Daten

externe Stammdaten für Faktura



## Wo kommen sie her?

x| Stammdaten: initial befüllter Datenbank-Snapshot

x| Bewegungsdaten: Programmierete Befüllung des Datenbank-Snapshots (über Objektmodell, konsistent und schnell)

x| Prozessspezifische externe Daten: über Dummy-Schnittstellen

## Aber wie?

### x| Datenbank-Snapshot

- | Anpassung der Struktur an neue Version
- | Hochmigrieren der Daten
- | Automatisiert mit LiquiBase

### x| Testcases zur Erzeugung der Bewegungsdaten

- | Programmanpassungen an neue Version
- | Manueller Aufwand

### x| Dummy Schnittstellen

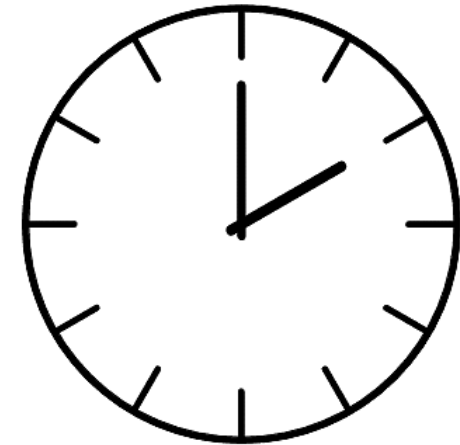
- | Manuell anpassen an geänderte Schnittstellen und neue Daten
- | Anpassungsbedarf eher selten



## Welche Abhängigkeiten sind das?

### x| Absolute Zeitabhängigkeit

Abhängigkeit von zeitabhängigen Daten, wie z.B. Preislisten und evtl. davon abhängige Logik



### x| Relative Zeitabhängigkeit

erst wird fakturiert, dann reklamiert, dann nochmal fakturiert

### x| Prozesszeitpunkte und Zeitlicher Verlauf

Fakturen finden wöchentlich statt

Abschläge werden nur am Monatsbeginn berechnet

→ Unterschied ob die Faktura am 28. eines Monats oder am 3. des Folgemonats durchgeführt wird

## Aber wie?

### x| Testfälle

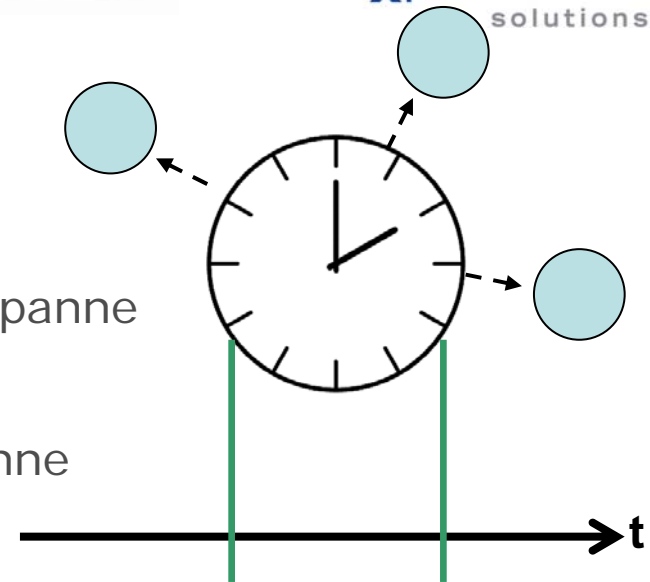
- | Genaue Definition der zu testenden Zeitspanne
- | Abstimmung der Testszenarien auf die Anforderungen der zu testenden Zeitspanne

### x| Daten

- | Definition mit fixem Zeitbezug, der in die zu testende Zeitspanne fällt

### x| Applikation / System

- | Zeitliche Parametrierung aller zeitliefernden Systeme („heute“)  
→ möglichst nur eine „Zeitquelle“ in der Applikation
- | Problem bei Erzeugung von Zeitstempeln in der Datenbank



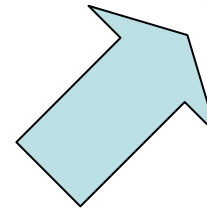
## Wodurch entstehen diese?

### x| Reguläre Änderungen

Parameter des zu fakturierenden Szenarios ändern sich, d.h. die Daten, die fakturiert werden sollen ändern sich

### x| Reklamationsbearbeitung

Falsche Leistungsdaten führen zu Reklamationen und müssen korrigiert werden



## Wie die menschlichen Eingaben ersetzen?

### x| Programmatische Manipulation der Daten

- | Meist sehr robust und schnell
- | Kein End-to-End Test

### x| User-Eingaben über GUI-Tests

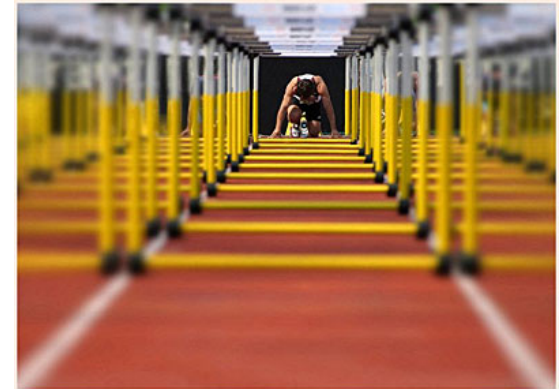
- | Am besten vergleichbar mit der Eingabe durch den Benutzer  
→ End-to-End
- | Pflegeaufwand bei Anpassungen der GUI  
→ Abhängig von der Dynamik des Systems



## Erzeugen, persistieren und aktualisieren?

### x| Erzeugen

Wie erhalte ich Ergebnis-Referenzdaten zur automatisierten Überprüfung?



### x| Persistieren

Wie können Ergebnis-Referenzdaten gespeichert werden für mehrfache Vergleiche?

### x| Aktualisieren

Wie können Ergebnis-Referenzdaten aktualisiert werden bei Logikänderungen oder neuen Testszenarien?

## Wie erzeugen, persistieren und aktualisieren?

### x| Erzeugen

- | Initiale Testläufe
- | Manuelle Prüfung bis alle Fälle OK sind

### x| Persistieren

- | Dump der internen Datenstrukturen der Faktura Daten aus der Datenbank in XML und abspeichern
- | XML-Rohdaten für Rechnungen abspeichern
- | ASCII-Datei mit Buchungsergebnissen abspeichern

### x| Aktualisieren

- | Analog der Erzeugung, nur manuelle Prüfung der von Änderungen betroffenen Fälle



## Die Hürden – Ergebnisse vergleichen

### Vergleich von Testergebnis und Referenzdaten?

#### x| Durchführung der Vergleiche

Wie können Referenzdaten und Testergebnisse verglichen werden?



#### x| Darstellung der Testergebnisse

Wurden die Referenzdaten mit den Testergebnissen verglichen, so müssen

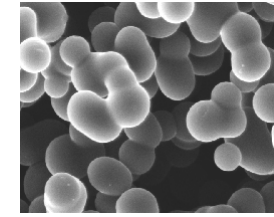
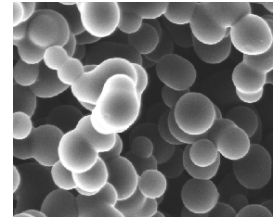
- | die Ergebnisse dargestellt werden
- | im Fehlerfall die Fehler identifiziert werden
- | die Testergebnisse in Stichproben geprüft werden

## Aber wie?

### x| Vergleichen auf zwei Ebenen

#### | Interne Datenstrukturen

- Dump aller internen Nutzdaten nach XML
- → XML-Diff mit Referenzstand



#### | Ergebnis Artefakte

- Letzte Strukturierte Outputebene
- XML-Rechnungs- und Lieferscheindaten → XML-Diff
- ASCII-Buchungsdaten → Text-Diff

```
1 <?xml version="1.0"?>
2 <!-- File: lib.xml -->
3
4 <xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/90-xsl">
5 <xsl:template match="/">
6 <tbl:library/td>
7 <table border="1" cellpadding="5">
8 <tr>
9 <td>ISEN</td>
10 <td>Author</td>
11 <td>Title</td>
12 <td>Publisher</td>
13 </tr>
14 <xsl:apply-templates select="library/book">
15 </xsl:template>
16 </xsl:template>
17
18 <xsl:template match="book">
19 <tr>
20 <td><xsl:value-of select="ISEN"/></td>
21 <td><xsl:for-each select="author">
```



```
1 <?xml version="1.0"?>
2 <!-- File: lib.xml -->
3
4 <xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/90-xsl">
5 <xsl:template match="/">
6 <tbl:library/td>
7 <table border="1" cellpadding="5">
8 <tr>
9 <td>ISEN</td>
10 <td>Author</td>
11 <td>Title</td>
12 <td>Publisher</td>
13 </tr>
14 <xsl:apply-templates select="library/book">
15 </xsl:template>
16 </xsl:template>
17
18 <xsl:template match="book">
19 <tr>
20 <td><xsl:value-of select="ISEN"/></td>
21 <td><xsl:for-each select="author">
```

# Die Hürden – Ergebnisse darstellen

## Aber wie?

### x| Ergebnisse darstellen: 3-stufig

- | Fehler ja/nein
- | Fehlerdarstellung als XML-Diff-Output  
→ meist weiterführende Betrachtung über  
visuelles Diff-Tool notwendig
- | Ergebnis-Artefakte und zugehörige DB-Snapshots für Review  
zusammengepackt



# Was haben wir betrachtet ?

**Start**

**Testfälle**

**Testdaten**

- Erzeugen
- Aktualisieren

**Zeitabhängigkeit** von

- Testfällen
- Testdaten
- System



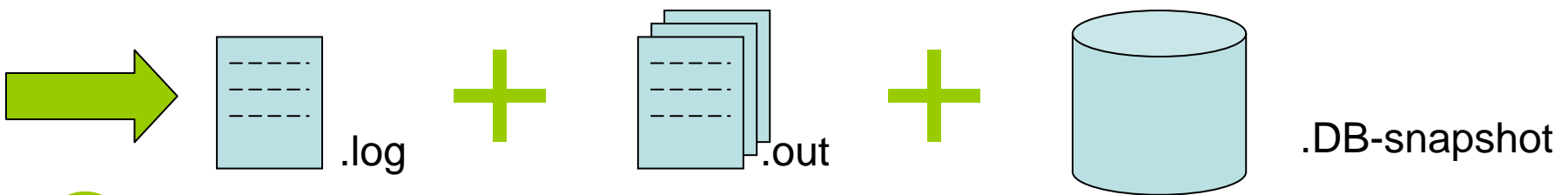
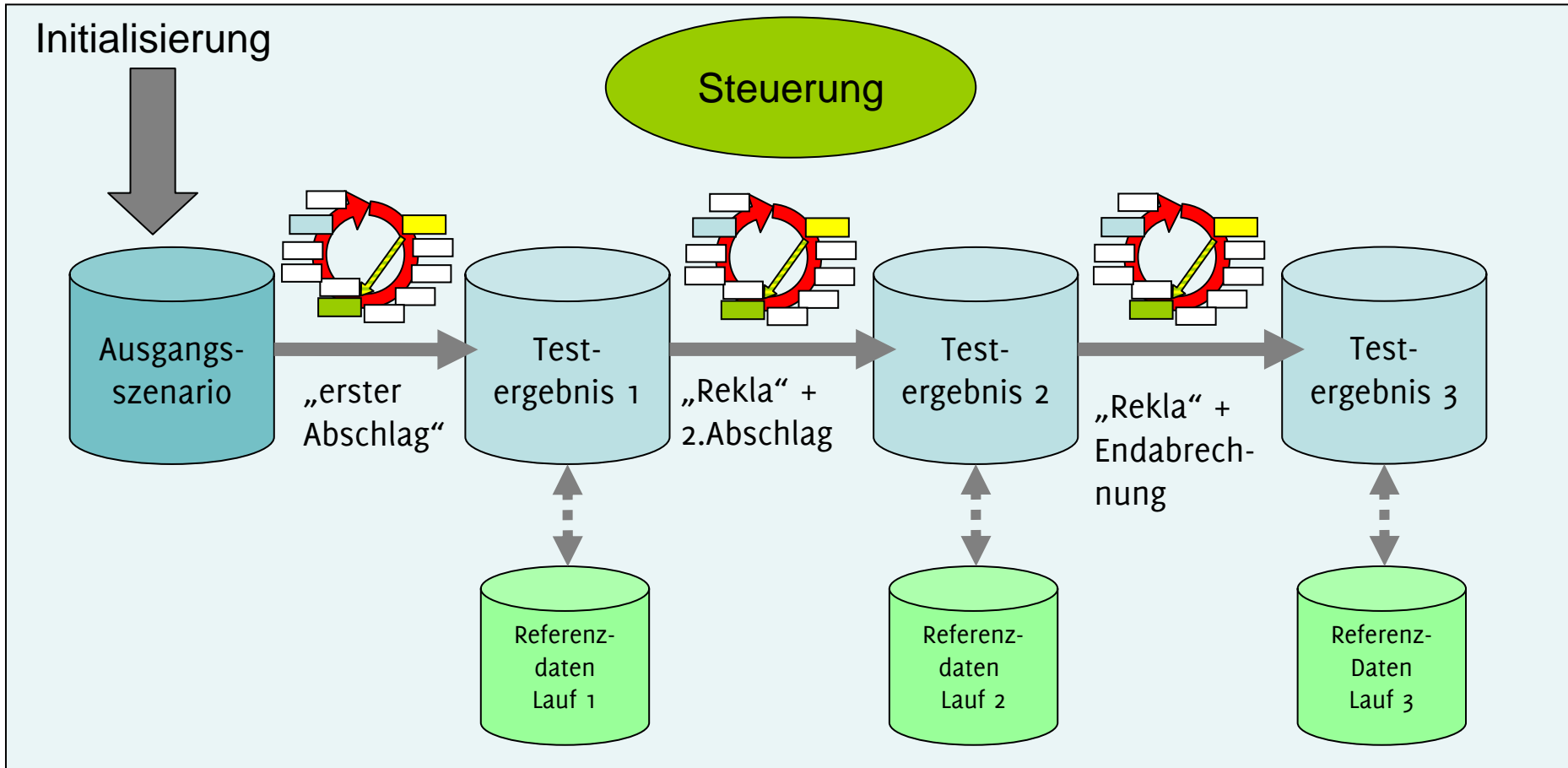
**Referenzdaten**

- Erzeugen
- Speichern
- Aktualisieren

**Testergebnisse**

- Vergleichen
- Darstellen

**Benutzersimulation**  
für Prozessabbildung



## Wie bekommt man all diese Themen unter einen Hut?

### x| Die Toollandschaft

Welche Tool-Umgebung ist notwendig um diese Art zu testen effizient durchführen zu können?

### x| Die Integration

Wie erfolgt die Integration in die Entwicklung?

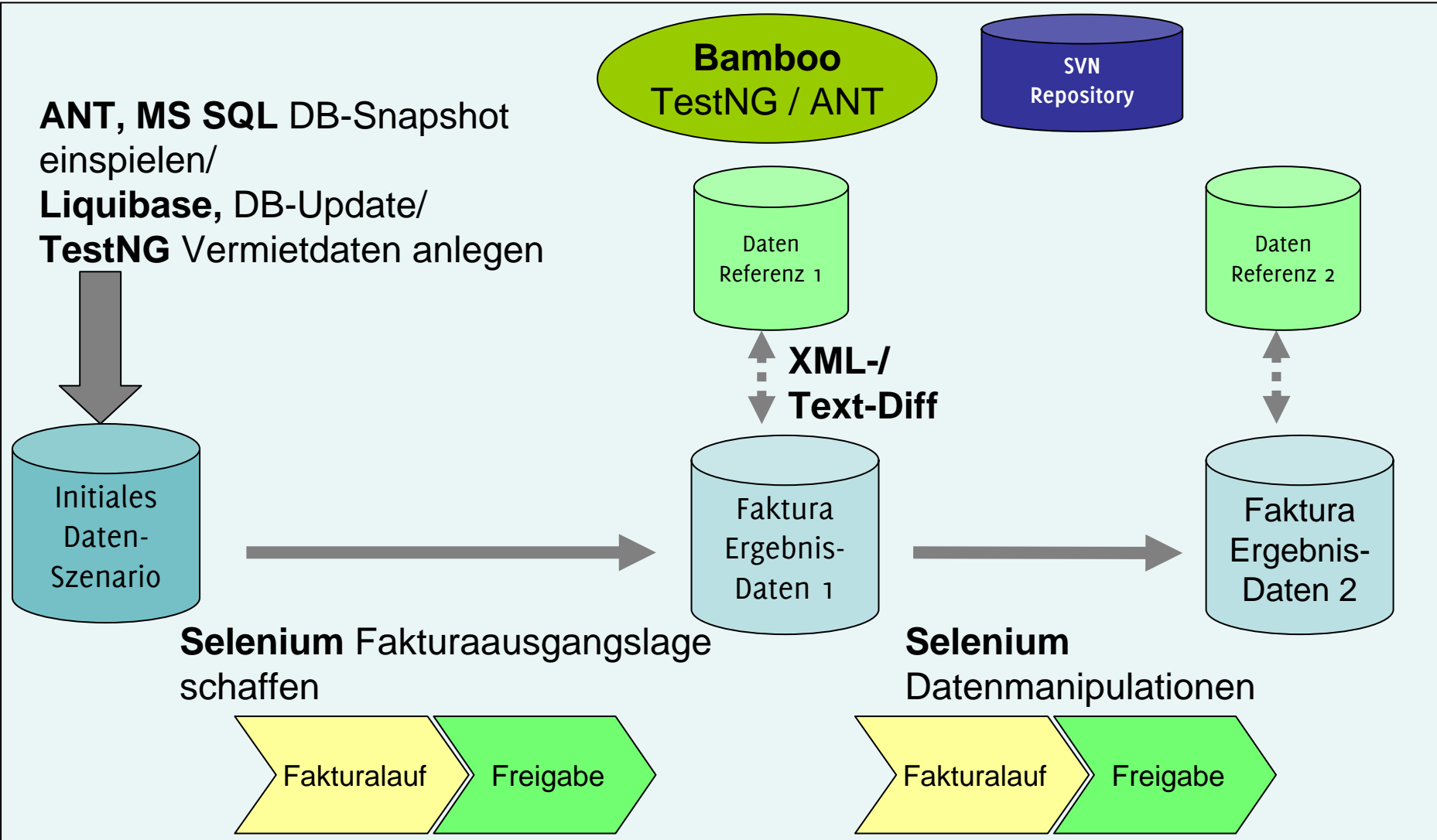


## Welche Tools wurden benutzt?

### x| Die Toollandschaft

- | Anker: Build-Management System **Bamboo**
- | Steuerung: eigene Steuerung, umgesetzt mit **TestNG** und **ANT**
- | Simulation der Userinteraktion: **Selenium** GUI-Test-Framework
- | Vergleiche über **XML- und Text-Diff**-Bibliotheken
- | Einspielen und erstellen von Datenbanksnapshots: **MS SQL Server**
- | Automatische Migration von Datenbankstrukturen und -Inhalten: **LiquiBase**
- | Verwaltung aller Ressourcen: **SVN**





## Wie erfolgt die Integration in die Entwicklung?

### x| Die Integration

- | Aus Sicht Entwicklungsprozess
  - Bei stabilem Entwicklungs (-zwischen) stand
  - Vor Auslieferung zum Test an den Kunden
  
- | Aus Toolsicht
  - Optimalerweise integriert in Continuous Integration Server
  
- | Aus Verwaltungssicht
  - Alles unter Versionskontrolle
    - Test als elementarer Bestandteil der Entwicklung

## Erkenntnisse aus verschiedensten Blickwinkeln

### x| Aufwand

- | Für die Erstellung der initialen Infrastruktur bis zur Produktionstauglichkeit ist nicht unerheblich
- | Für die initiale Erstellung von Testfällen ist nicht zu vernachlässigen
- | Für die Pflege bestehender Testfälle ist gering

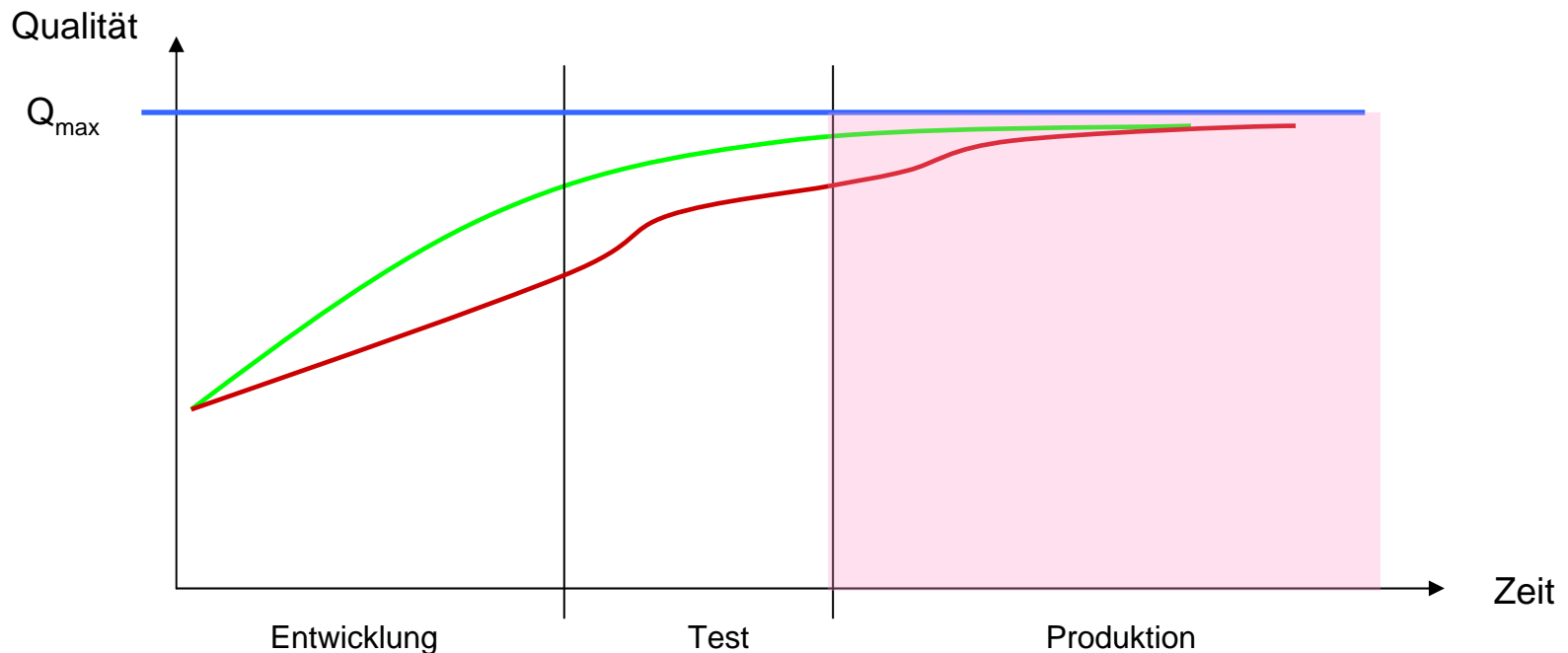
→ **Optimal bei** Systemen mit **hoher Änderungsdynamik**

(große Systeme mit lokalen Änderungen  
und hohem Bedarf an Regressionstests)

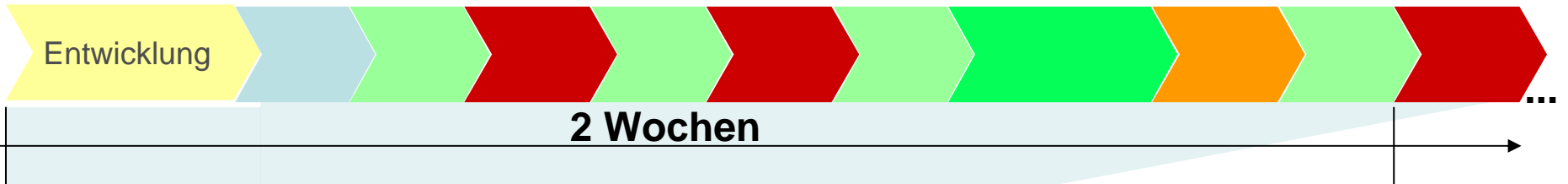
## Erkenntnisse aus verschiedensten Blickwinkeln

### x| Qualität

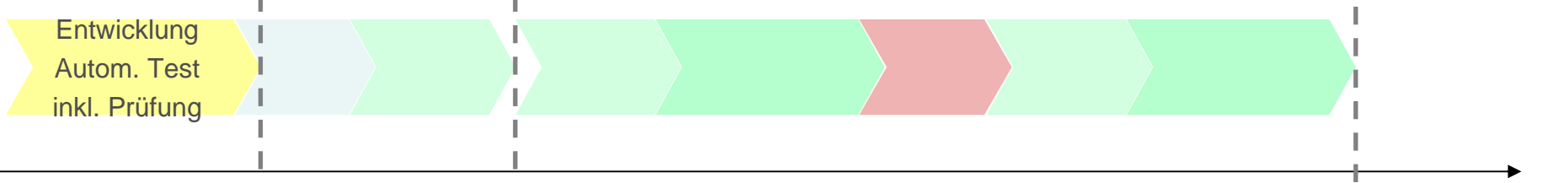
- | Hohe Qualität bereits zu Beginn der eigentlichen Testphase
- | **Höhere Qualität** in der Produktion
- | **Hohe Qualität** des Systems **über die Zeit**



## Bisher



**Jetzt: 25 min → 80.640% schneller!**



Standardszenarien      Massentest und Sonderkonstellationen

→ **Hohe Planungsgenauigkeit** für Produktivnahmen  
bei kürzerer Testphase

**xl** Es sind viele Hürden zu überwinden, aber...

**Es lohnt sich!**

**Sie werden  
schneller  
und besser!**



# Danke für Ihre Aufmerksamkeit!

[Michael.Blech@bwfuhrpark.de](mailto:Michael.Blech@bwfuhrpark.de)

[G.Hermann@excellent.de](mailto:G.Hermann@excellent.de)

Weitere Informationen zu eXXcellenten  
Softwareprojekten erhalten sie am Stand von

ex|Xcellent  
solutions

oder unter [www.excellent.de](http://www.excellent.de)