

# Continuous Database Integration – Ja! Aber wie?

## DOAG Konferenz 2011

Andrej Pashchenko  
Consultant, Trivadis GmbH

15.11.2011 Nürnberg

BASEL    BERN    LAUSANNE    ZÜRICH    DÜSSELDORF    FRANKFURT A.M.    FREIBURG I.BR.    HAMBURG    MÜNCHEN    STUTTGART    WIEN

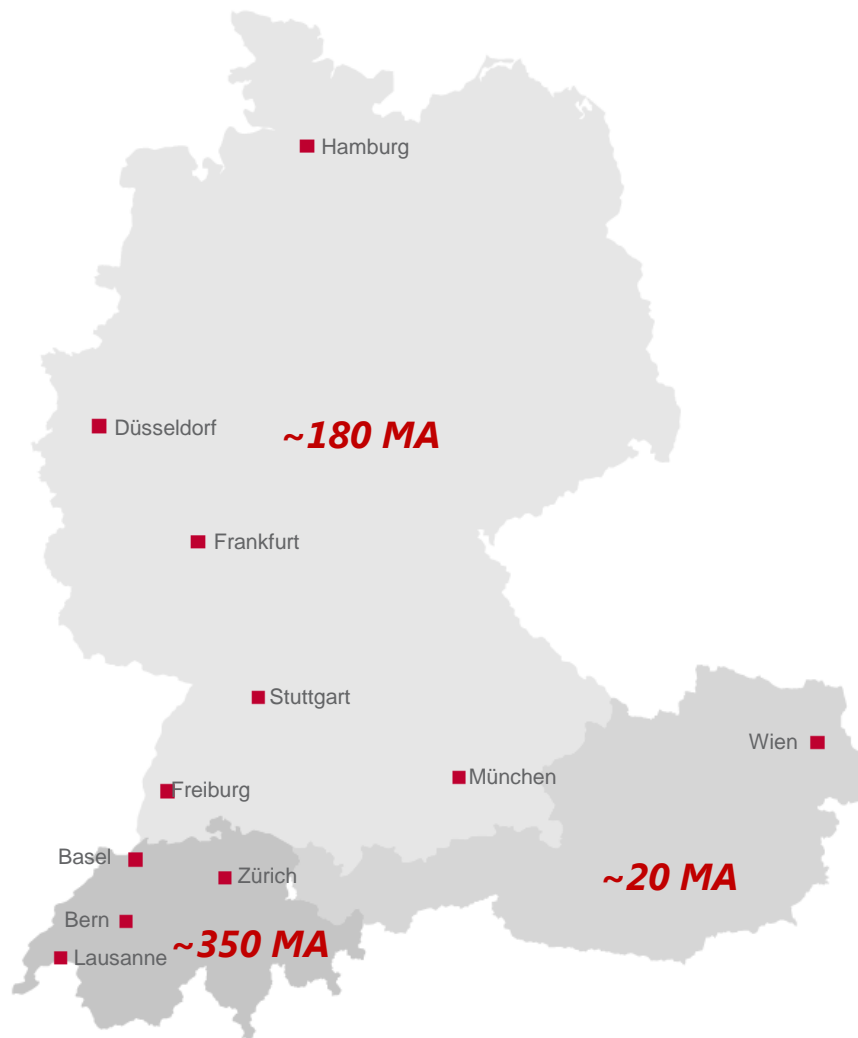
1

2011 © Trivadis

Continuous Database Integration – Ja aber wie?  
15.11.2011

**trivadis**  
makes IT easier. ■ ■ ■

# Trivadis Facts & Figures



11 Trivadis Niederlassungen mit über 550 Mitarbeitern

Finanziell unabhängig und nachhaltig profitabel

Kennzahlen 2010

- Umsatz CHF 101 / EUR 73 Mio.
- Dienstleistungen für über 700 Kunden in mehr als 1'800 Projekten
- Über 170 Service Level Agreements
- Mehr als 5'000 Trainingsteilnehmer
- Forschungs- und Entwicklungsbudget: CHF 5.0 / EUR 3.6 Mio.

# Das Besondere

## **Kundenindividuelle Lösungskompetenz und Herstellerunabhängigkeit**

- bietet fundierte Methodenkenntnisse und eigenentwickelte Vorgehensweisen
- garantiert wiederholbare Qualität und Realisierungssicherheit

## **Technologiekompetenz**

- hat über 17 Jahre Expertise in Oracle und Microsoft
- verfügt über ein eigenes Technology Center und setzt auf technologische Exzellenz

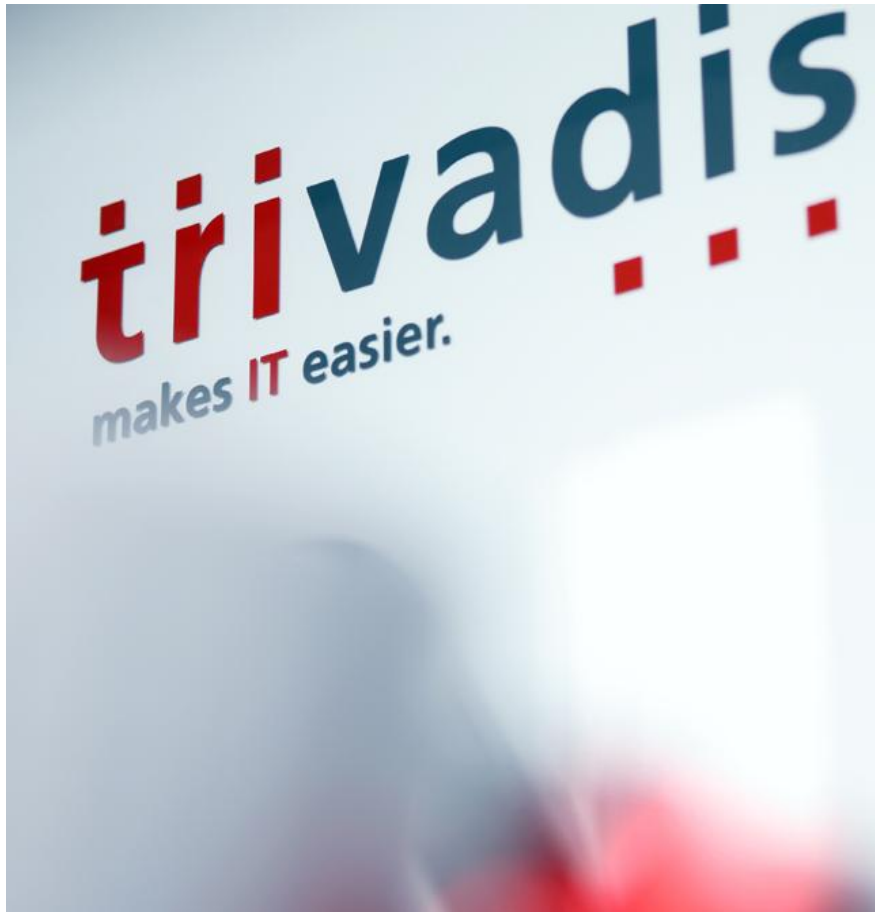
## **Lösungs- und Integrations-Know-how**

- hat eine breite, branchenübergreifende Kundenbasis und jährlich über 1800 Projekte
- verbindet technologisches Spezialistenwissen mit dem Verständnis für die Business-Spezifika des Kunden

## **Begleitung über den gesamten IT-Projekt- Lifecycle**

- begleitet den gesamten IT-Projekt-Lifecycle mit einem modularen Dienstleistungsportfolio
- bietet für jeden „Reifegrad“ die passende Dienstleistungs- und Lösungskombination

# AGENDA



Einführung: CI und CDBI

Build-Prozess

DB-Change-Tool

Verbindungsaufbau

Unit-Tests

What's next?

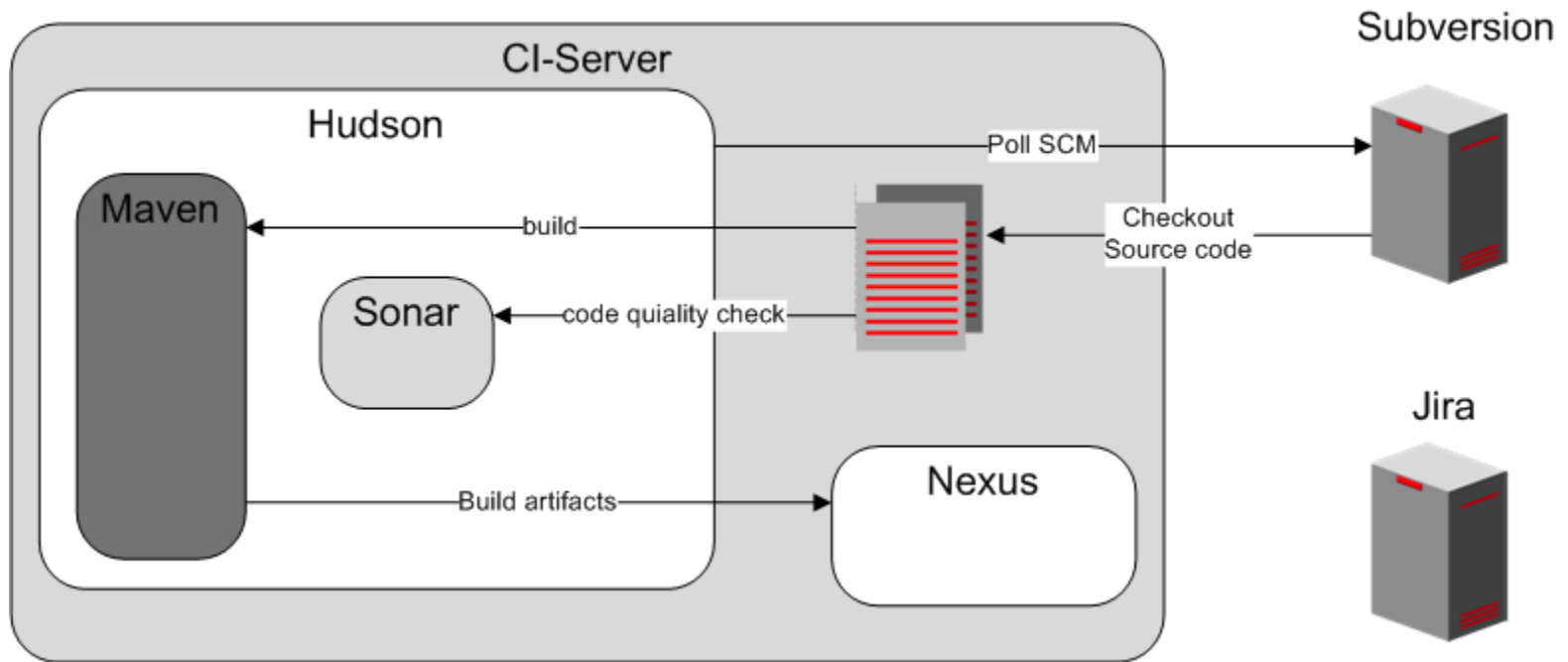
Referenzen

# Einführung

- Continuous Integration ist eine Best Practice:
  - Frühzeitig und häufig integrieren
- Vorteile:
  - Risikominimierung: nachgelagerte Integration erschwert die Sichtbarkeit des Projektstandes
  - Integrationsprobleme frühzeitig erkennen
  - Fehler sind einfacher zu beheben
  - Aktueller Stand ist immer verfügbar als Demo oder für QA
- Nachteile:
  - Initialer Aufwand
  - Test-Suite für automatisierte Tests benötigt
- Voraussetzungen für Continuous Integration:
  - Schneller, vollautomatisierter Build, der Tests beinhaltet
  - Ein zentrales Quellcode-Repository
  - Der Build wird in einem Produktionsklon durchgeführt

# Einführung

Was fehlt hier?

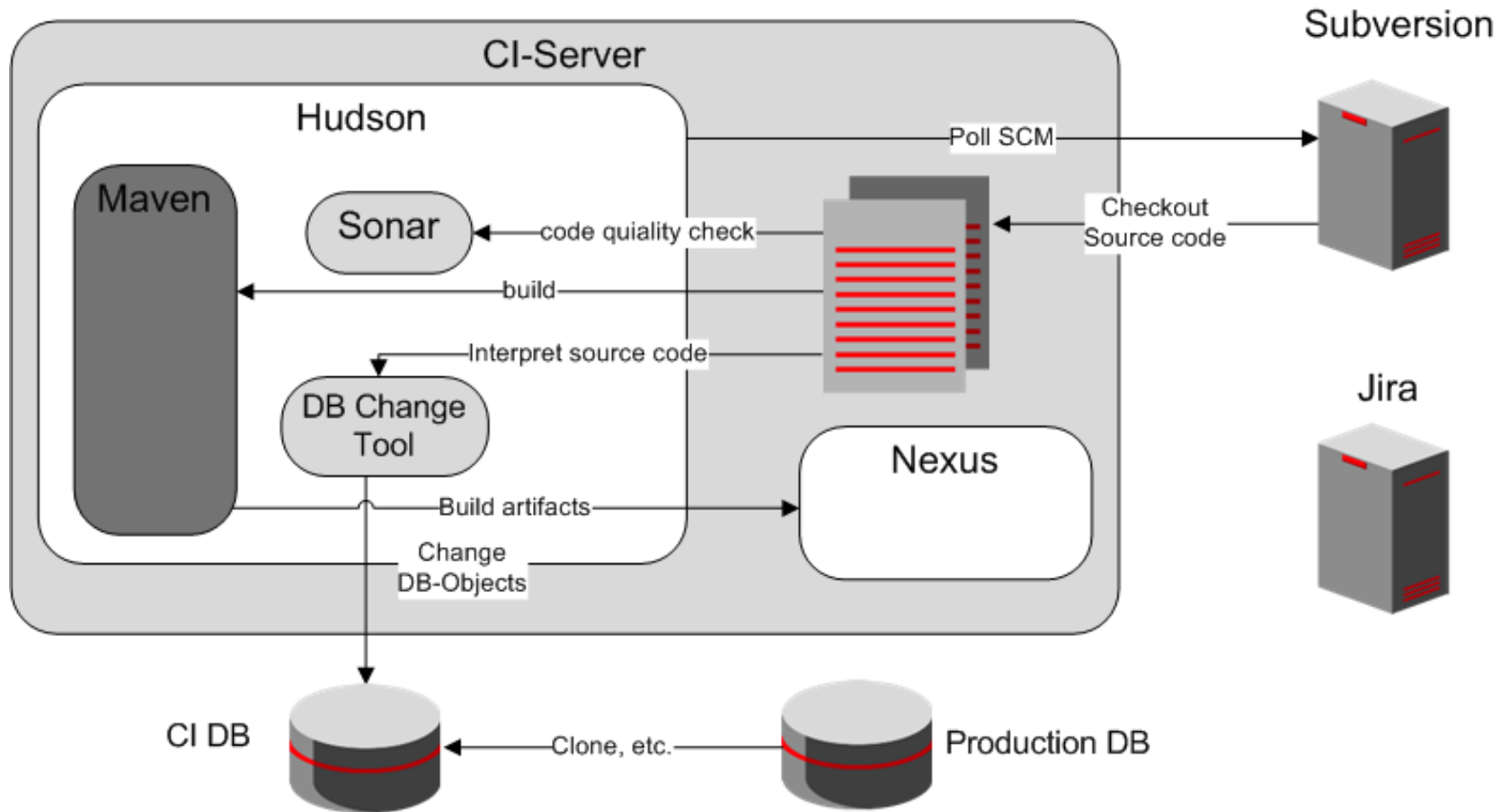


# Einführung

Warum ist die Datenbankentwicklung anders?

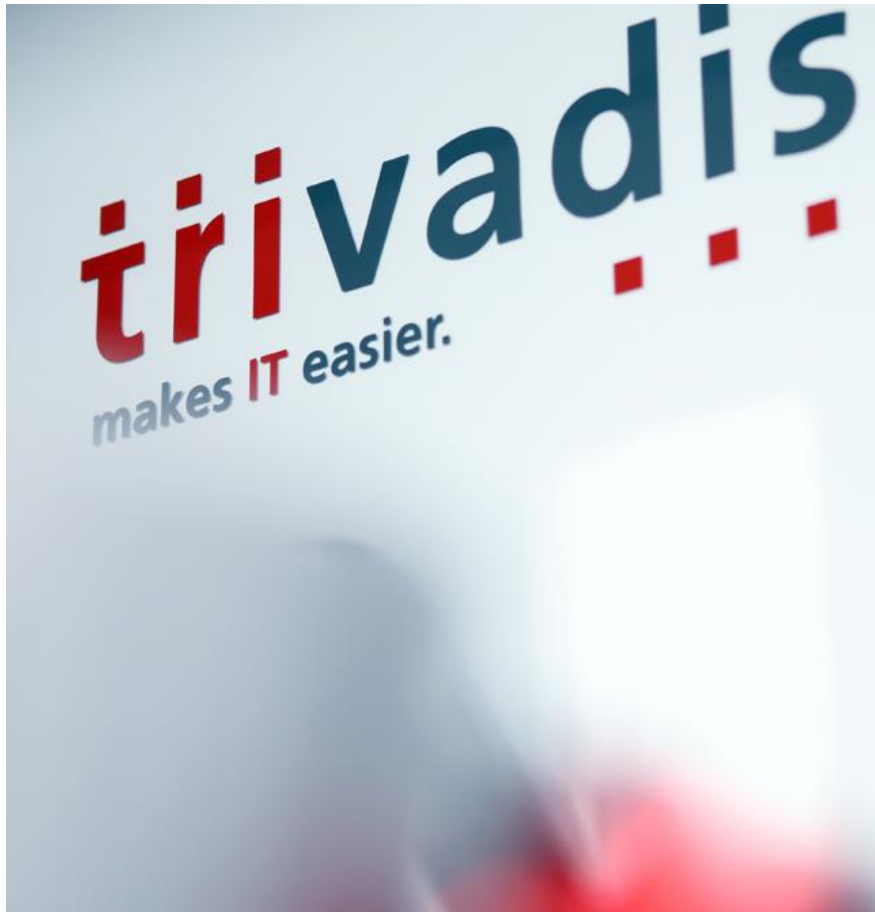
- Unmöglich komplett neu aufzubauen
- Build ist eine Änderung an einem existierenden DB-Schema
- Benötigt einen DB-Stand, auf den die Änderungen angewendet werden
- Normalerweise besteht die Änderung aus:
  - Änderungen an Datenstrukturen
  - Datenmigration
  - Änderungen am Datenzugriffscod
- Objekte wie Views oder PL/SQL können immer überschrieben werden
- Andere Objekte wie Tabellen erfordern Änderung durch ALTER-DDL-Statements und womöglich eine Datenmigration

# Einführung





# AGENDA



Einführung: CI und CDBI

Build-Prozess

DB-Change-Tool

Verbindungsaufbau

Unit-Tests

What's next?

Referenzen

# Build-Prozess: DB-Change Tool

Was leistet ein DB-Change Tool?

- Hilft die Build-Logik aus dem Quelltext auszulagern:
  - Was ist bereits ausgeliefert worden?
  - Was ist in einer Fehlersituation zu tun?
  - In welcher Reihenfolge sollen die Änderungen angewendet werden?
  - Protokollieren

Anforderungen:

- Alle benötigten Änderungsskripte in definierter Reihenfolge und nur einmalig ausführen
- Im Fehlerfall die Weiterverarbeitung stoppen
- Protokoll-Tabelle in der DB, um Ausführung und den Erfolgsstatus festzuhalten
- Abgestimmt mit der Art und Weise wie der Quellcode organisiert ist (Verzeichnisstruktur, Inhalt der Skripte, Verbindungsaufbau)

# DB-Change Tool

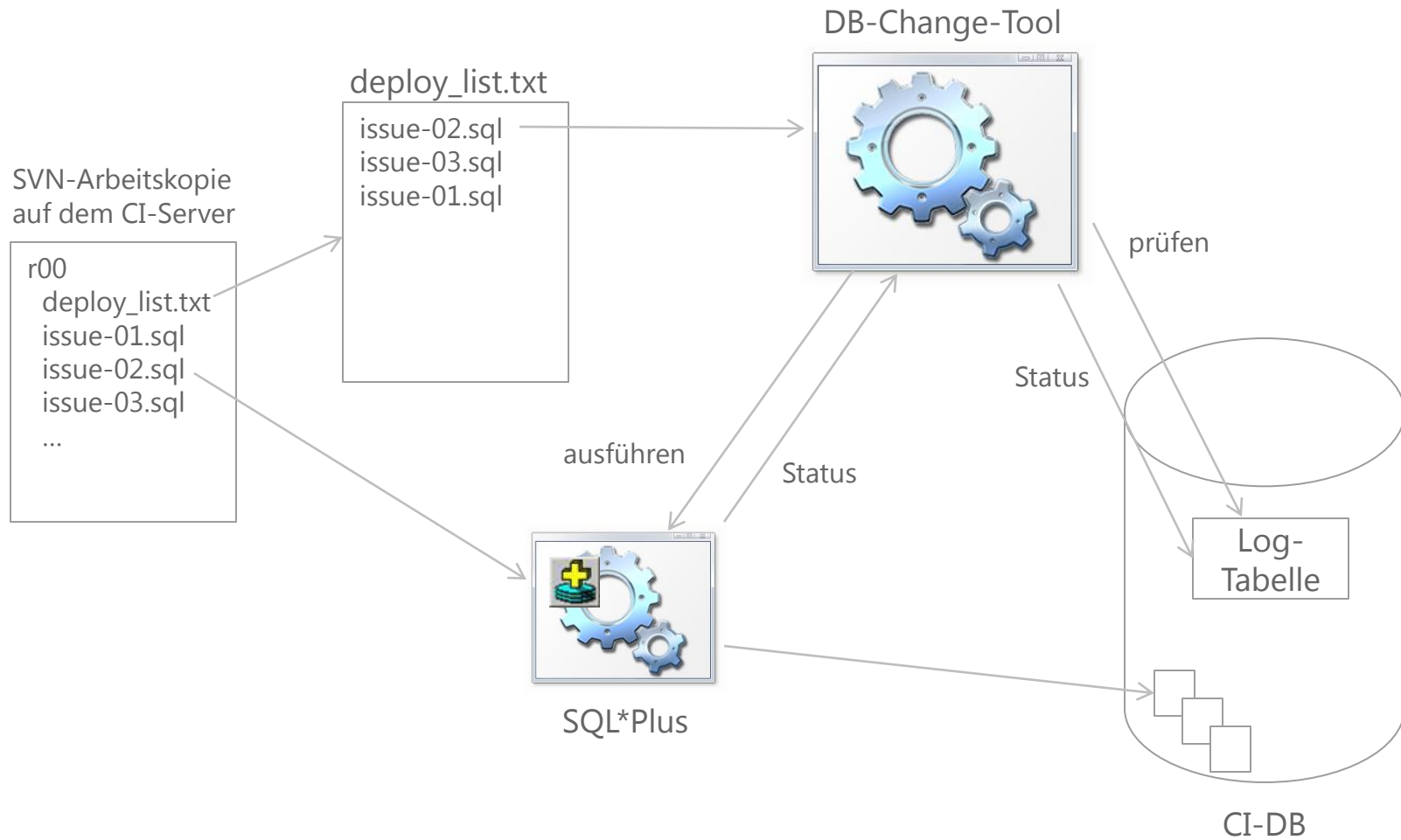
- Vorhandene Open-Source DB-Change Tools:
  - Liquibase, <http://www.liquibase.org/>
  - Flyway, <http://code.google.com/p/flyway/>
  - Dbmaintain, <http://www.dbmaintain.org/>
  - c5-db-migration, <http://code.google.com/p/c5-db-migration/>
  - Dbdeploy, <http://dbdeploy.com>
  - Autopatch, <http://autopatch.sourceforge.net/>
  - migratedB, <http://migratedb.sourceforge.net/>
- Nachteile:
  - Sehr dünne Benutzer- und Entwickler-Basis
  - Zum Teil sehr sparsame Dokumentation
  - Weiterentwicklung kann jederzeit eingestellt werden
  - Explizite Reihenfolge der Änderungen kann nur bei Liquibase (als XML) definiert werden

# DB-Change Tool

- Alternative: Trivadis-Lösung auf der Basis der Shell- und SQL-Skripte
- Vorteile:
  - Gut abgestimmt mit evtl. vorhandener Vorgehensweise bei Auslieferung
  - Ausführungsumgebung SQL\*Plus
  - Oracle Client Features nutzen
- Auszug aus POM.XML

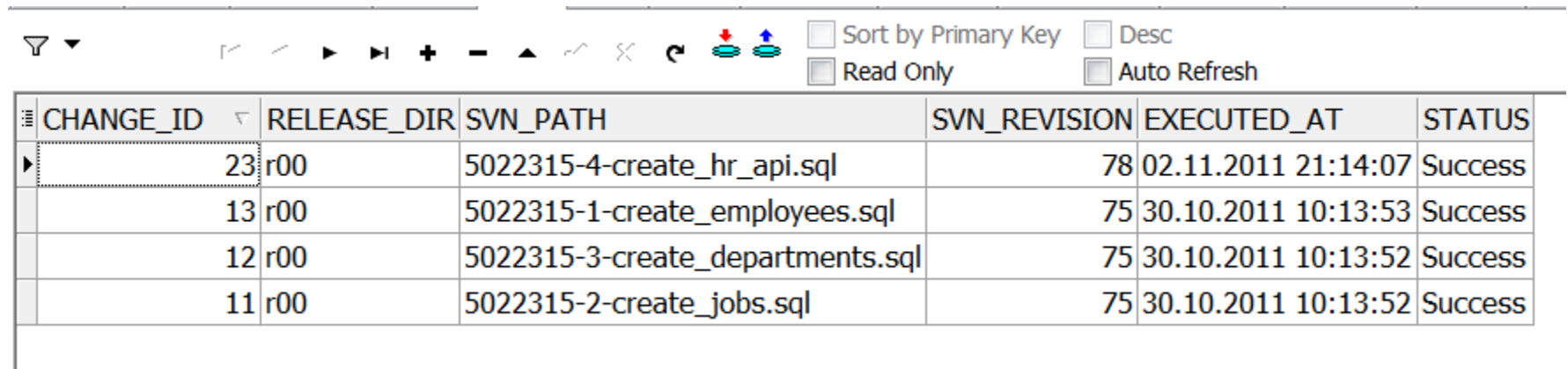
```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>exec-maven-plugin</artifactId>
  <executions>
    <execution>
      <id>sqldeploy</id>
      <phase>process-resources</phase>
      <goals> <goal>exec</goal> </goals>
      <configuration>
        <executable>dbd_start.sh</executable>
        <workingDirectory>${basedir}/src/main/db/${change_dir}</workingDirectory>
      </configuration>
    </execution>
  </executions>
</plugin>
```

# DB-Change Tool



# DB-Change Tool

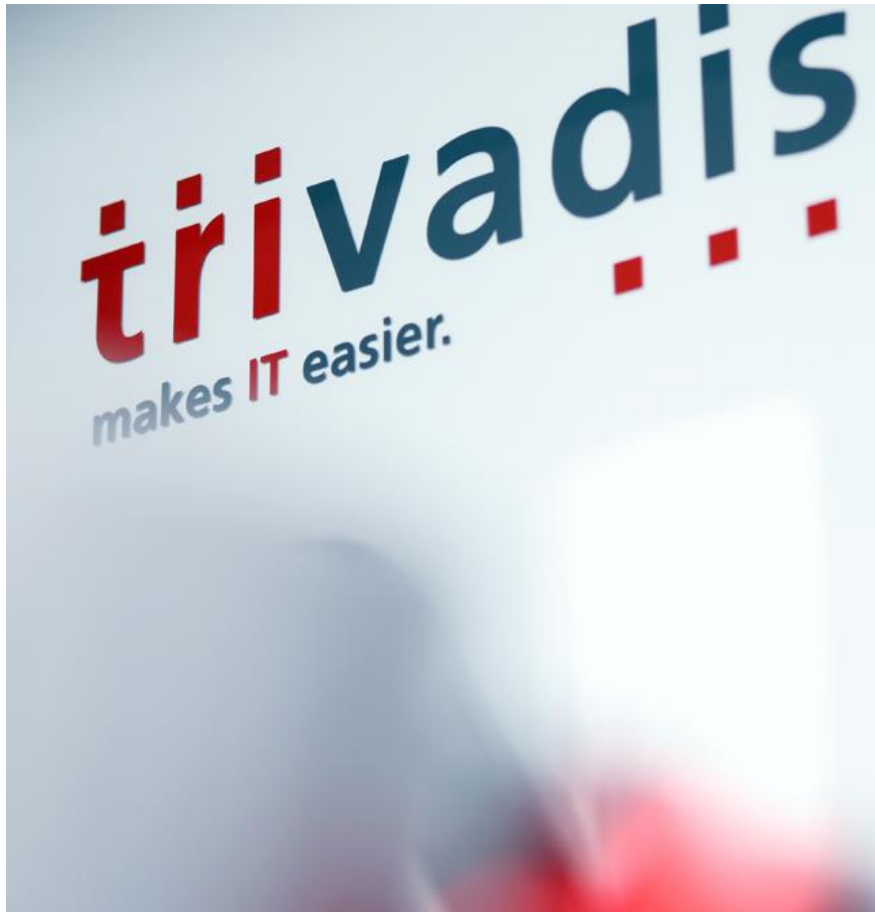
- Log-Tabelle in der Datenbank enthält folgende Informationen:
  - welche Skripte wann in der DB gelaufen sind
  - was war die Reihenfolge der Anwendung
  - was war der Erfolgsstatus
  - welcher SVN-Revisionsnummer entspricht der Stand der DB



Sort by Primary Key  Desc   
Read Only  Auto Refresh

CHANGE_ID	RELEASE_DIR	SVN_PATH	SVN_REVISION	EXECUTED_AT	STATUS
23	r00	5022315-4-create_hr_api.sql	78	02.11.2011 21:14:07	Success
13	r00	5022315-1-create_employees.sql	75	30.10.2011 10:13:53	Success
12	r00	5022315-3-create_departments.sql	75	30.10.2011 10:13:52	Success
11	r00	5022315-2-create_jobs.sql	75	30.10.2011 10:13:52	Success

# AGENDA



Einführung: CI und CDBI

Build-Prozess

DB-Change-Tool

Verbindungsaufbau

Unit-Tests

What's next?

Referenzen

# DB-Change Tool

- Verbinden mit Oracle Secure External Password Store
  - Verbindungsdaten (Benutzer, Passwort, TNS-Alias) werden verschlüsselt in einem Wallet abgespeichert

- SQLNET.ORA

```
WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY = C:\tns) ) )

SQLNET.WALLET_OVERRIDE = TRUE
SSL_CLIENT_AUTHENTICATION = FALSE
SSL_VERSION = 0
```

- TNSNAMES.ORA

```
crm_at_cldb = (DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP) (HOST = yourhost.com) (PORT = 1521))
  (CONNECT_DATA = (SERVER = DEDICATED) (SID = cldb) ) )
```

- Wallet anlegen und Verbindungsdaten speichern

```
>mkstore -wrl c:\tns -create
...
>mkstore -wrl c:\tns -createCredential crm_at_cldb crm crm
...
>sqlplus /@crm_at_cldb
```



# DB-Change Tool

- Beispiel: eine neue Spalte durch ETL-Schiene ziehen
  - Übersichtlich
  - Skripte kann man aufgabenbezogen gestalten
  - Der Entwickler muss die echten Verbindungsdaten nicht wissen

```
-- View-Definition in der Quell-DB
connect /@crm_at_db_src
@../crm_at_db_src/views/kunde.vw

-- Tabelle im Stage erweitern
connect /@stage_at_db_dwh

ALTER TABLE stg_kunde ADD (cust_category VARCHAR2(100));

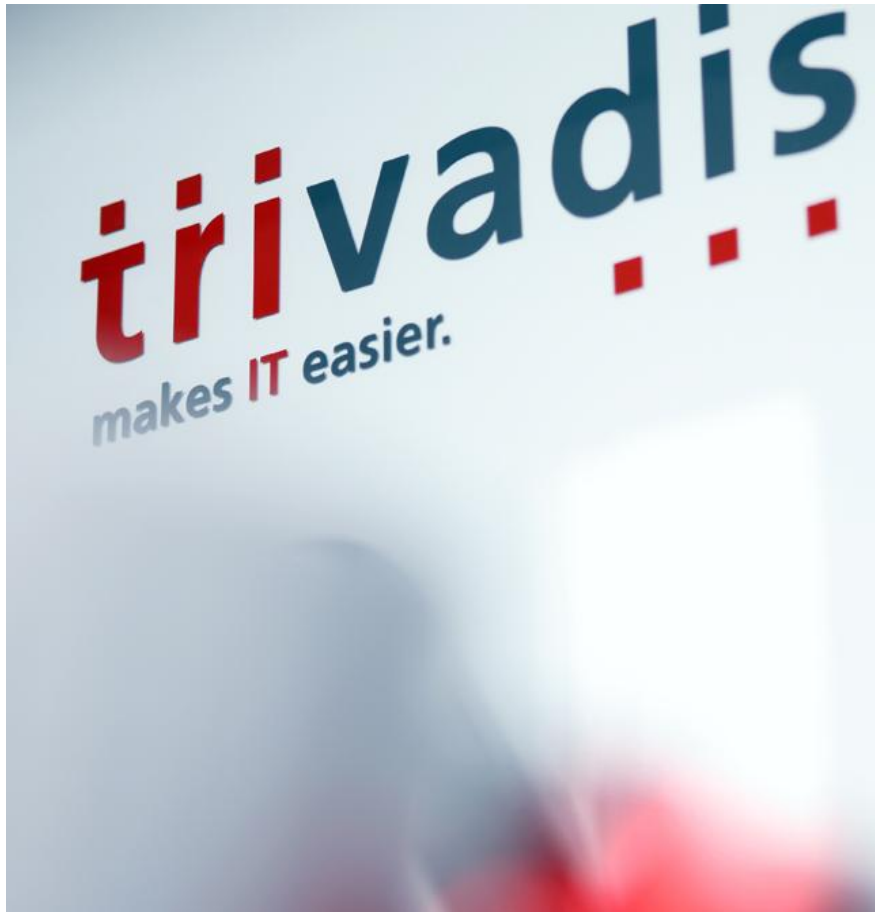
-- Tabelle im Core erweitern
connect /@core_at_db_dwh

ALTER TABLE co_kunde ADD (cust_category VARCHAR2(100));

-- Dimension im Mart erweitern
connect /@mart_at_db_dwh

ALTER TABLE dim_kunde ADD (cust_category VARCHAR2(100));
```

# AGENDA



Einführung: CI und CDBI

Build-Prozess

DB-Change-Tool

Verbindungsaufbau

Unit-Tests

What's next?

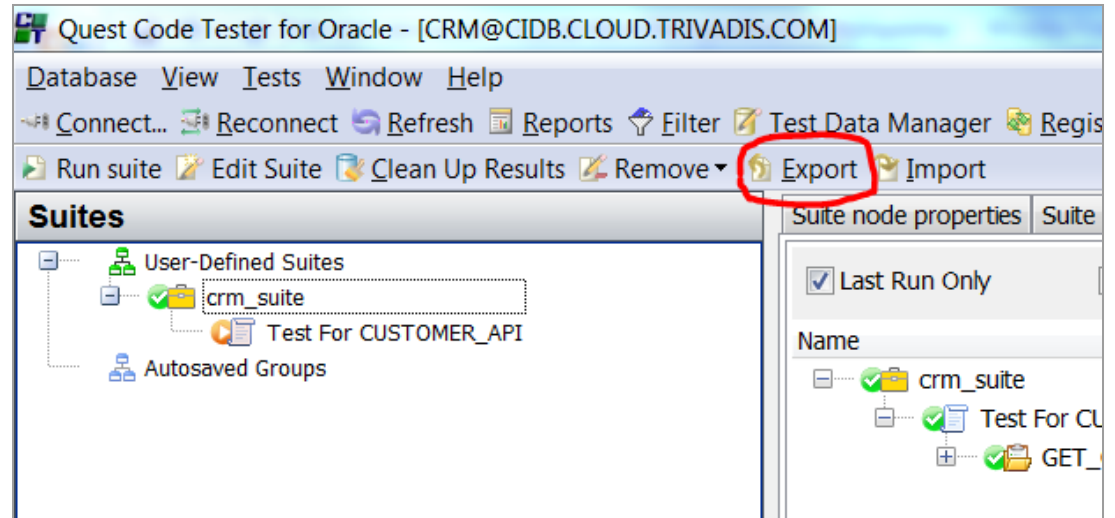
Referenzen

# Unit-Tests

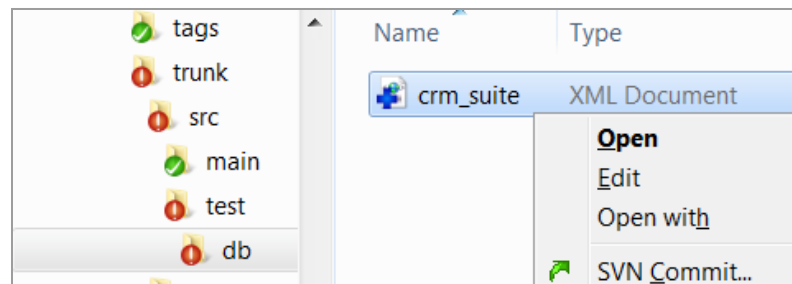
- Quest Code Tester for Oracle
  - Kostenpflichtiges aber komfortables Tool für das Testen von PL/SQL
  - Nimmt viel Arbeit durch das Generieren von Test-Code ab
  - Stellt PL/SQL API für Automatisierung zur Verfügung
  - Nachfolger von utplsql
- Anbindung an CI:
  - Testdefinitionen über VCS verteilen
  - Tests ausführen
  - Testergebnisse bereitstellen

# Unit-Tests

- Testdefinitionen über VCS verteilen
  - Export von Testdefinitionen als XML



- XML-Datei einchecken



# Unit-Tests

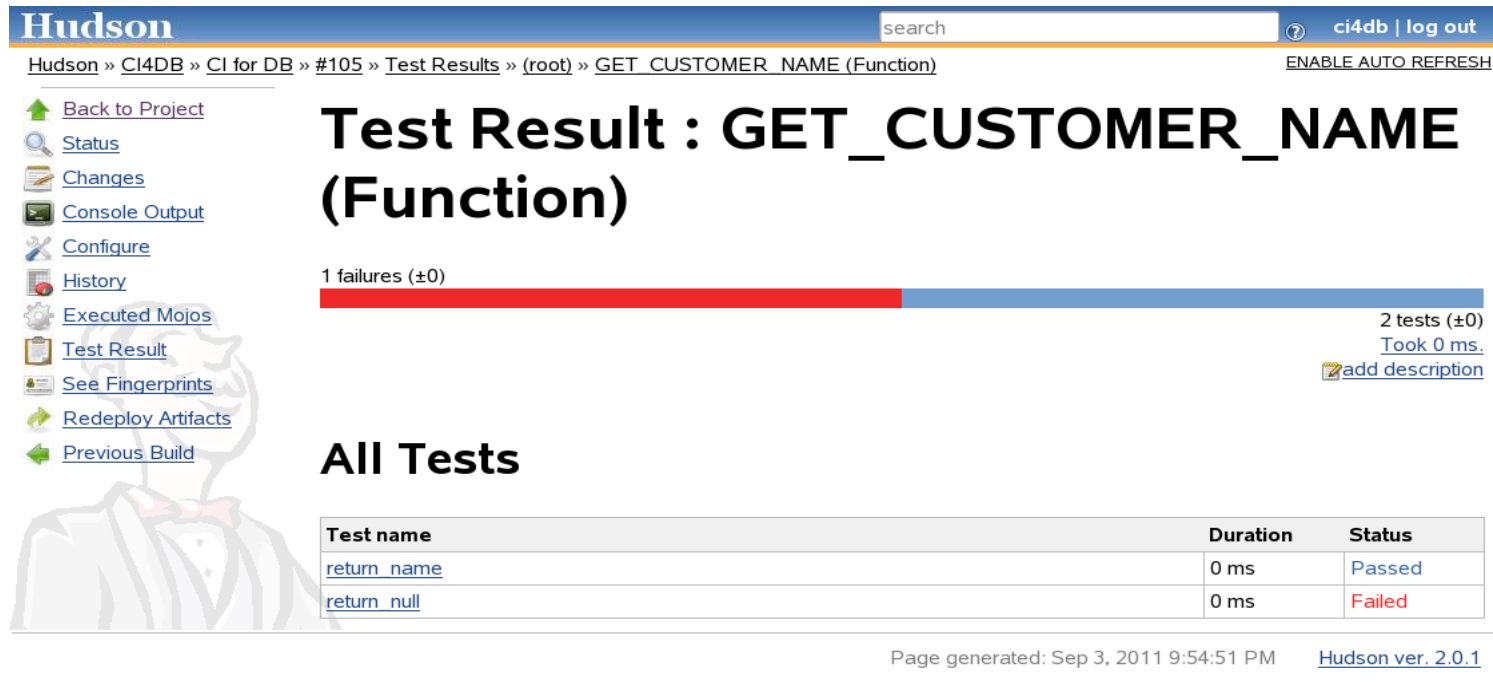
- Testdefinitionen über VCS verteilen
  - Import der XML-Definitionen mittels PL/SQL API über CI-Server
    - Kopieren der XML-Datei auf den DB-Server und Laden als BFILE:
      - Vorteil: Einfachere Implementierung
      - Nachteil: Zusätzliche Konfiguration (Zugang zum DB-Server ), Directory und Rechte in der DB
    - Importieren der XML-Datei als CLOB direkt vom Client (CI-Server) aus:
      - Vorteil: Keine zusätzliche Konfiguration
      - Nachteil: Implementierung ist nicht mehr trivial
    - Aufruf von PL/SQL API (Package QU\_XMLDOM\_IMPORT )

```
begin
  -- Load the export, preparing for import.
  qu_xmldom_import.init_xml(:xml_import);
  qu_xmldom_import.set_options( ... );
  qu_xmldom_import.set_for_mapping(prog_owner_in => 'CRM'
                                   ,harn_owner_in => 'CRM' );

  -- Perform the import.
  qu_xmldom_import.import_as_xml;
end;
/
```

# Unit-Tests

- Ausführen von Testdefinitionen
  - PL/SQL API von Code Tester, Package QU\_TEST
    - Bietet verschiedene Möglichkeiten Test-Suites und einzelne Test-Cases auszuführen
  - Über PL/SQL lassen wir die Ergebnisse im JUnit-XML-Format ausgeben (Surefire-Report), sodass sie von Hudson interpretiert werden:



**Hudson** search ci4db | log out

Hudson » CI4DB » CI for DB » #105 » Test Results » (root) » GET\_CUSTOMER\_NAME (Function) ENABLE AUTO REFRESH

[Back to Project](#)  
[Status](#)  
[Changes](#)  
[Console Output](#)  
[Configure](#)  
[History](#)  
[Executed Mojos](#)  
[Test Result](#)  
[See Fingerprints](#)  
[Redeploy Artifacts](#)  
[Previous Build](#)

## Test Result : GET\_CUSTOMER\_NAME (Function)

1 failures (±0)

2 tests (±0)  
Took 0 ms.  
[add description](#)

### All Tests

Test name	Duration	Status
<a href="#">return_name</a>	0 ms	Passed
<a href="#">return_null</a>	0 ms	Failed

Page generated: Sep 3, 2011 9:54:51 PM [Hudson ver. 2.0.1](#)

# Unit-Tests

## Hudson

Hudson » CI4DB » CI for DB » #105 » Test Results » (root) » GET\_CUSTOMER\_NAME (Function) » return\_null

- [Back to Project](#)
- [Status](#)
- [Changes](#)
- [Console Output](#)
- [Configure](#)
- [History](#)
- [Executed Mojos](#)
- [Test Result](#)
- [See Fingerprints](#)
- [Redeploy Artifacts](#)
- [Previous Build](#)

# Failed

GET\_CUSTOMER\_NAME (Function).return\_null (from crm\_suite)

## Error Message

```
From Program Value of "-1" <> Expected Value "NULL"
```

# AGENDA



Einführung: CI und CDBI

Build-Prozess

DB-Change-Tool

Verbindungsaufbau

Unit-Tests

What's next?

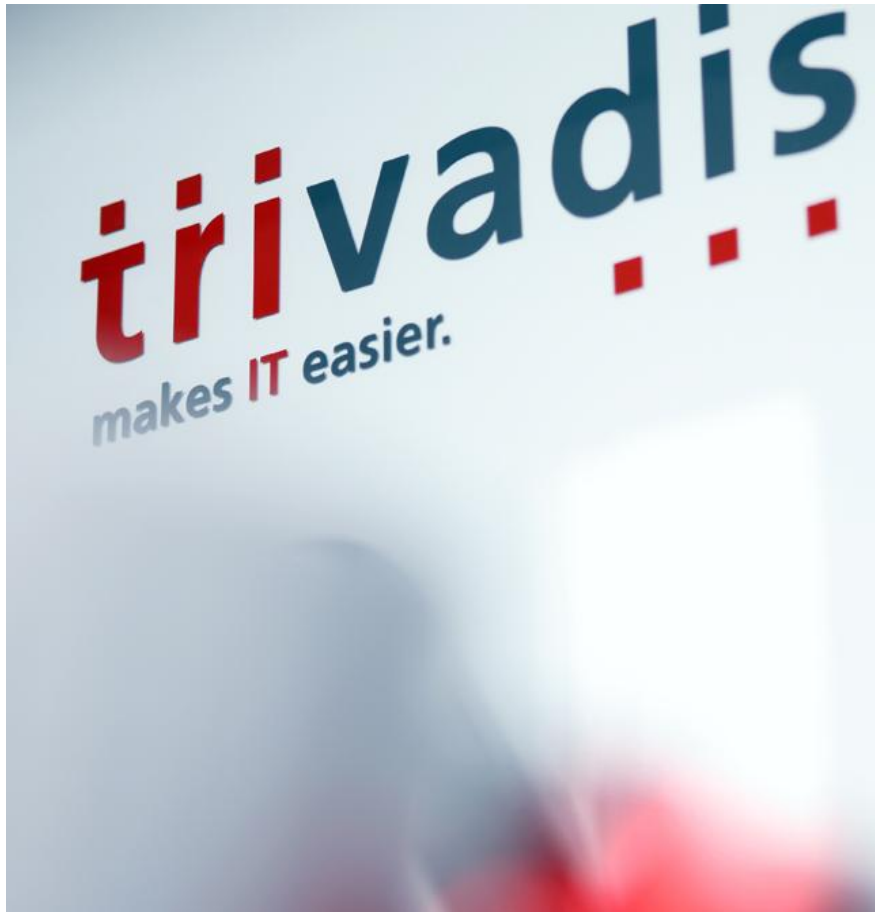
Referenzen



# What's next?

- Trivadis SQL & PL/SQL Coding Guidelines,  
<http://www.trivadis.com/de/technologie/oracle/oracle-application-development/oracle-sql-und-plsql.html>  
Integration von Trivadis Code-Checker
- Code Qualitätsmetriken: Sonar Plugin für PL/SQL
- PL/SQL Inline-Dokumentation, z.B. PLDoc
- APEX
- Data Warehouse (OWB, ODI)

# AGENDA



Einführung: CI und CDBI

Build-Prozess

DB-Change-Tool

Verbindungsaufbau

Unit-Tests

What's next?

Referenzen

# Referenzen

- Scott Ambler, Pramod Sadalage: Refactoring Databases – evolutionary database design, Addison Wesley, 2006
- Paul M. Duvall: Continuous Integration , Addison Wesley, 2007
- Gunter Popp: Konfigurationsmanagement mit Subversion, Maven und redmine, 3. Auflage, dpunkt.verlag, 2009
- [QCTO Community Blog](http://toadworld.com/BLOGS/tabid/67/EntryId/493/Continuous-Integration.aspx)  
<http://toadworld.com/BLOGS/tabid/67/EntryId/493/Continuous-Integration.aspx>

# VIELEN DANK.

Trivadis GmbH

Andrej Pashchenko

Werdener Str. 4  
40227 Düsseldorf

Tel. +49 211 58 66 64 70  
Fax + 49 211 58 66 64 71

info@trivadis.com  
www.trivadis.com

BASEL    BERN    LAUSANNE    ZÜRICH    DÜSSELDORF    FRANKFURT A.M.    FREIBURG I.BR.    HAMBURG    MÜNCHEN    STUTTGART    WIEN