buschmais

Beratung . Technologie . Innovation

Java User Group
Saxony

Database Change Management mit Liquibase

Dirk Mahler
buschmais GbR

Inhaber
Torsten Busch, Frank Schwarz,
Dirk Mahler und Tobias Israel

dirk.mahler@buschmais.com
http://www.buschmais.de/

Dresden, 06.12.2012

Database Change Management mit Liquibase - Inhalt

buschmais
Beratung . Technologie . Innovation

- Agenda
  - Das Problem
  - Das Werkzeug im Überblick
  - Demo #1
  - Erweiterte Konzepte
  - Demo #2
  - Praxiserfahrungen

buschmais

Beratung . Technologie . Innovation

Database Change Management mit Liquibase

Das Problem

Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation

- Fragen an das Auditorium

Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation

- Fragen an das Auditorium
  - Wer arbeitet mit relationalen Datenbanken?

Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation

- Fragen an das Auditorium
  - Wer arbeitet mit relationalen Datenbanken?
  - Wer muss regelmäßig Schemata – bzw. Daten migrieren?

Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation

- Fragen an das Auditorium
  - Wer arbeitet mit relationalen Datenbanken?
  - Wer muss regelmäßig Schemata – bzw. Daten migrieren?
  - Wer verwendet dafür ein Werkzeug?

Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation

- Fragen an das Auditorium
  - Wer arbeitet mit relationalen Datenbanken?
  - Wer muss regelmäßig Schemata – bzw. Daten migrieren?
  - Wer verwendet dafür ein Werkzeug?
  - Wer verwendet dafür ein ***eigenes*** Werkzeug?

Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation

- Fragen an das Auditorium
  - Wer arbeitet mit relationalen Datenbanken?
  - Wer muss regelmäßig Schemata – bzw. Daten migrieren?
  - Wer verwendet dafür ein Werkzeug?
  - Wer verwendet dafür ein ***eigenes*** Werkzeug?
  - Wer hat schon einmal die Übersicht über seine Datenbank-Skripte verloren?

Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation

- Fragen an das Auditorium
  - Wer arbeitet mit relationalen Datenbanken?
  - Wer muss regelmäßig Schemata – bzw. Daten migrieren?
  - Wer verwendet dafür ein Werkzeug?
  - Wer verwendet dafür ein ***eigenes*** Werkzeug?
  - Wer hat schon einmal die Übersicht über seine Datenbank-Skripte verloren?
  - Wer vermeidet es daher grundsätzlich, Schemata bzw. Daten zu migrieren?

Database Change Management mit Liquibase – Das Problem

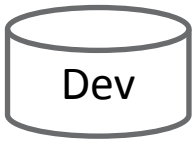
buschmais
Beratung . Technologie . Innovation

- Anwendungen unterliegen stetigen Veränderungen
  - Verwaltung des jeweiligen Entwicklungsstandes in SCM-Systemen (SVN, Git) gehört zum Standard-Vorgehen
    - Java-Klassen
    - Ressourcen
  - Arbeit in Zweigen
    - Head/Trunk, Release-/Patch-Branches
- Datenbankstrukturen sind an den jeweiligen Entwicklungsstand der Anwendung gekoppelt, z.B.
  - Tabellen, Constraints, Sequenzen, etc.
  - Stammdaten, Testdaten
- SCM-Verwaltung von Datenbankskripten ist unumgänglich
- Frage: Wie kann das effizient organisiert werden?

Database Change Management mit Liquibase – Das Problem

buschmais

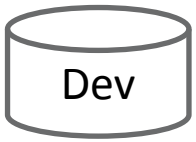
Beratung . Technologie . Innovation



Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation

create table A

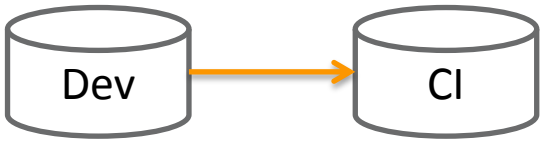


Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation

create table A

create table A



Database Change Management mit Liquibase – Das Problem

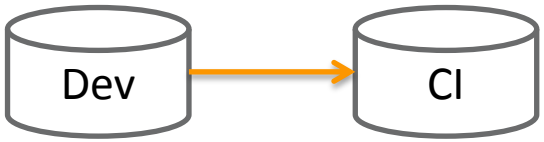
buschmais
Beratung . Technologie . Innovation

create table A

alter table B

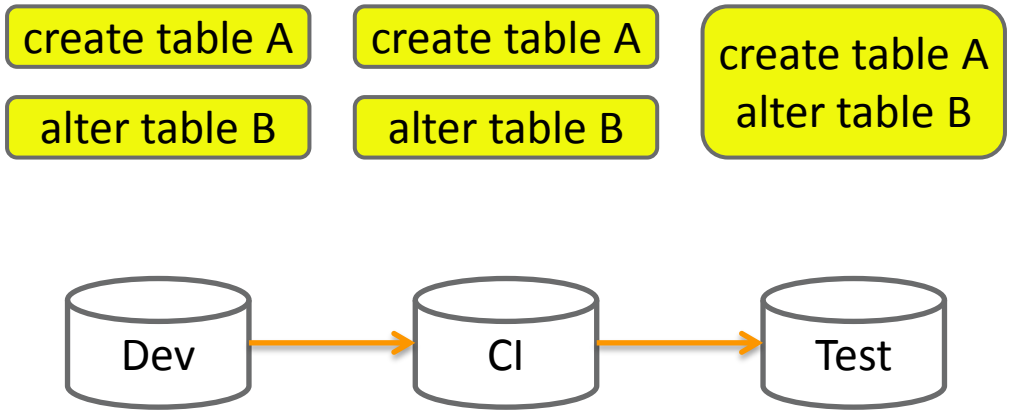
create table A

alter table B



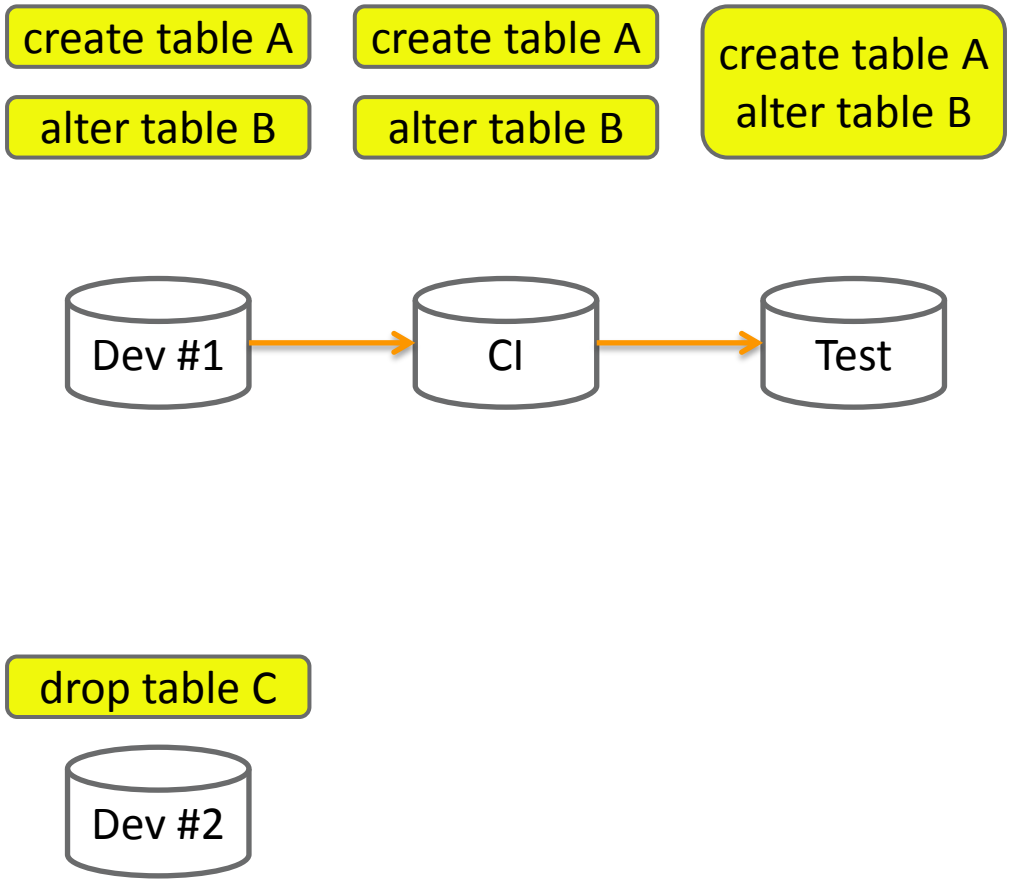
Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation



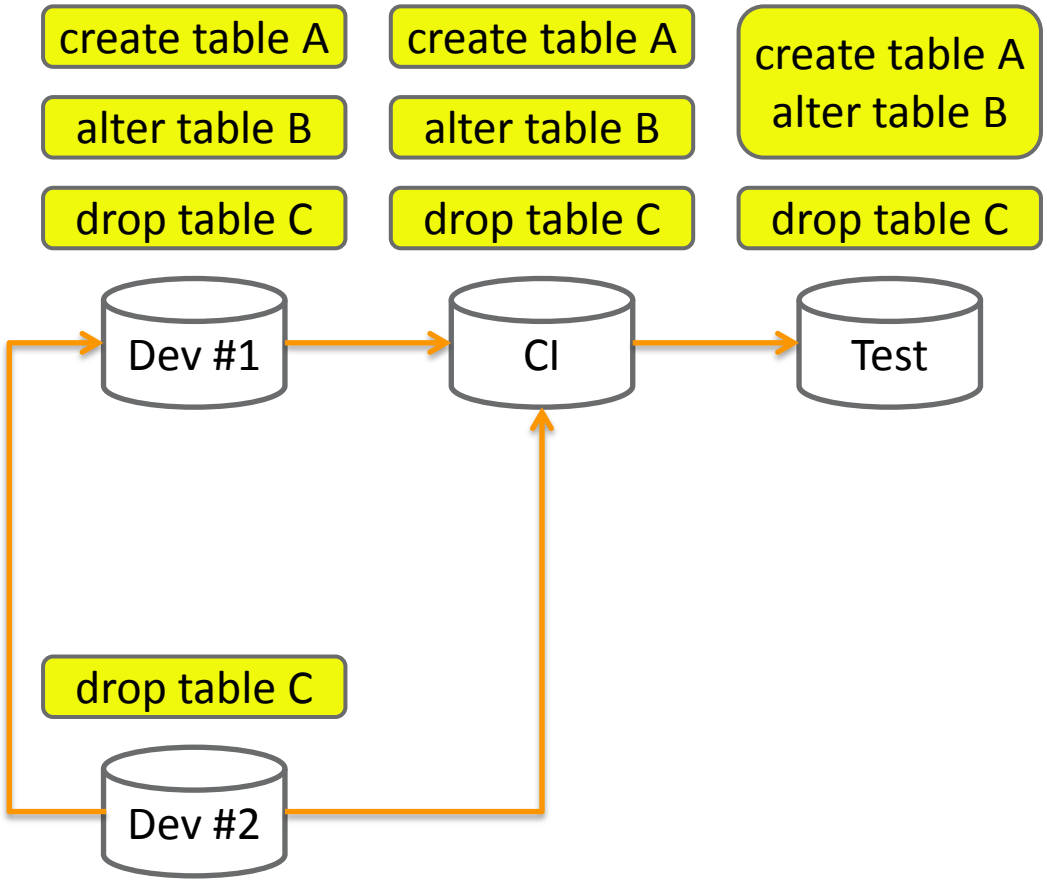
Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation



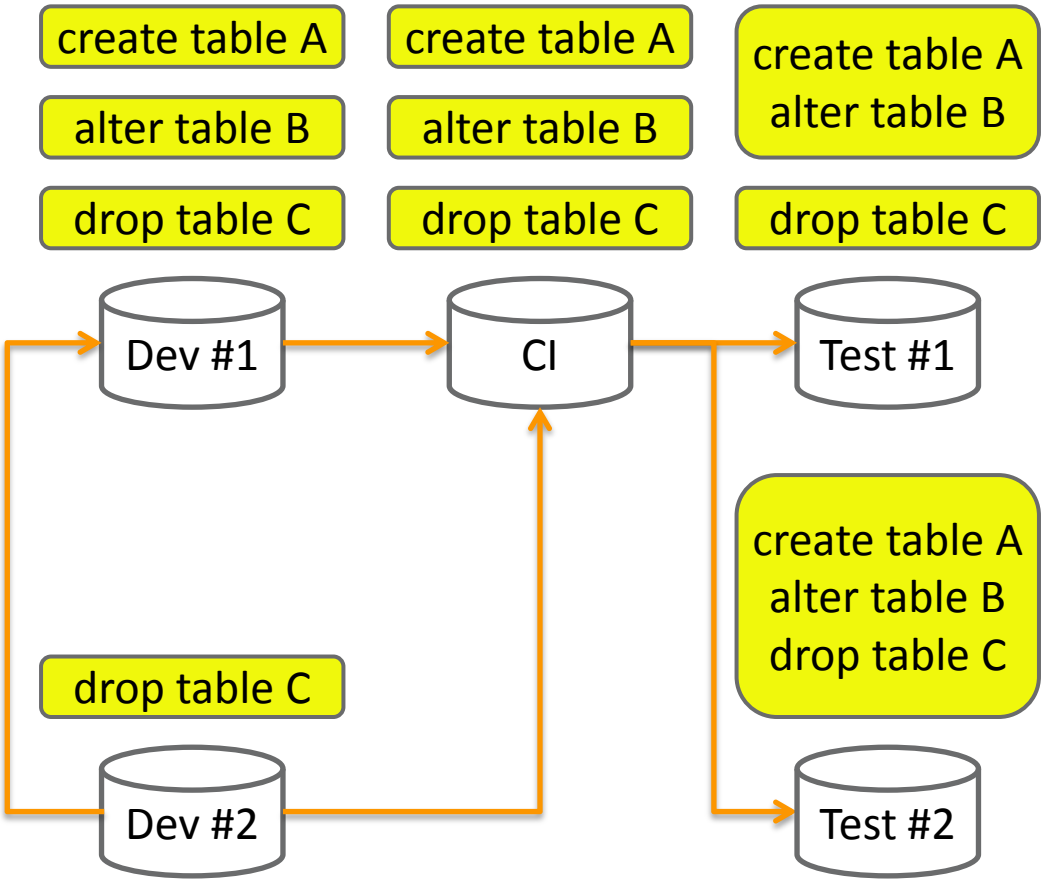
Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation



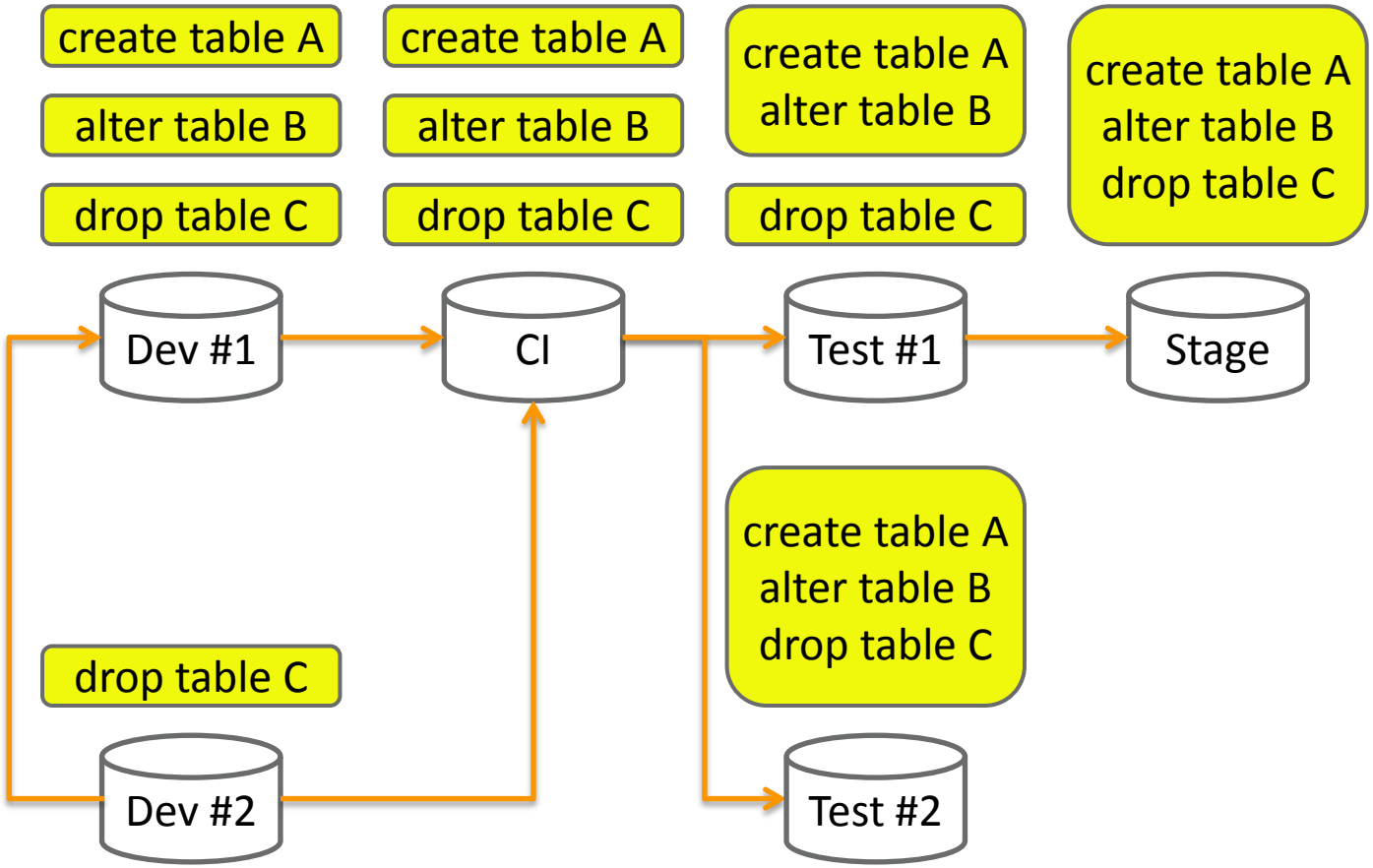
Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation



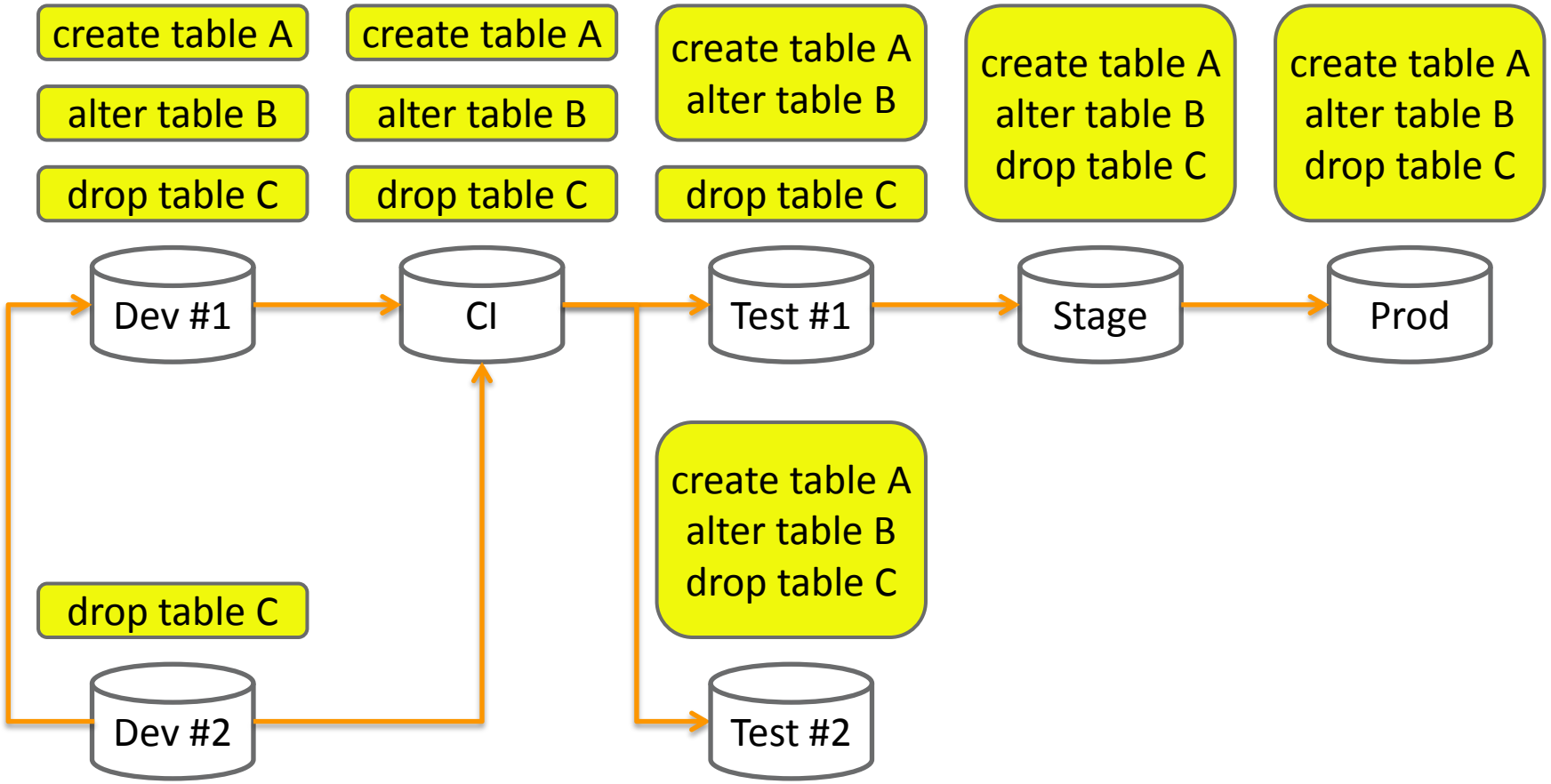
Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation



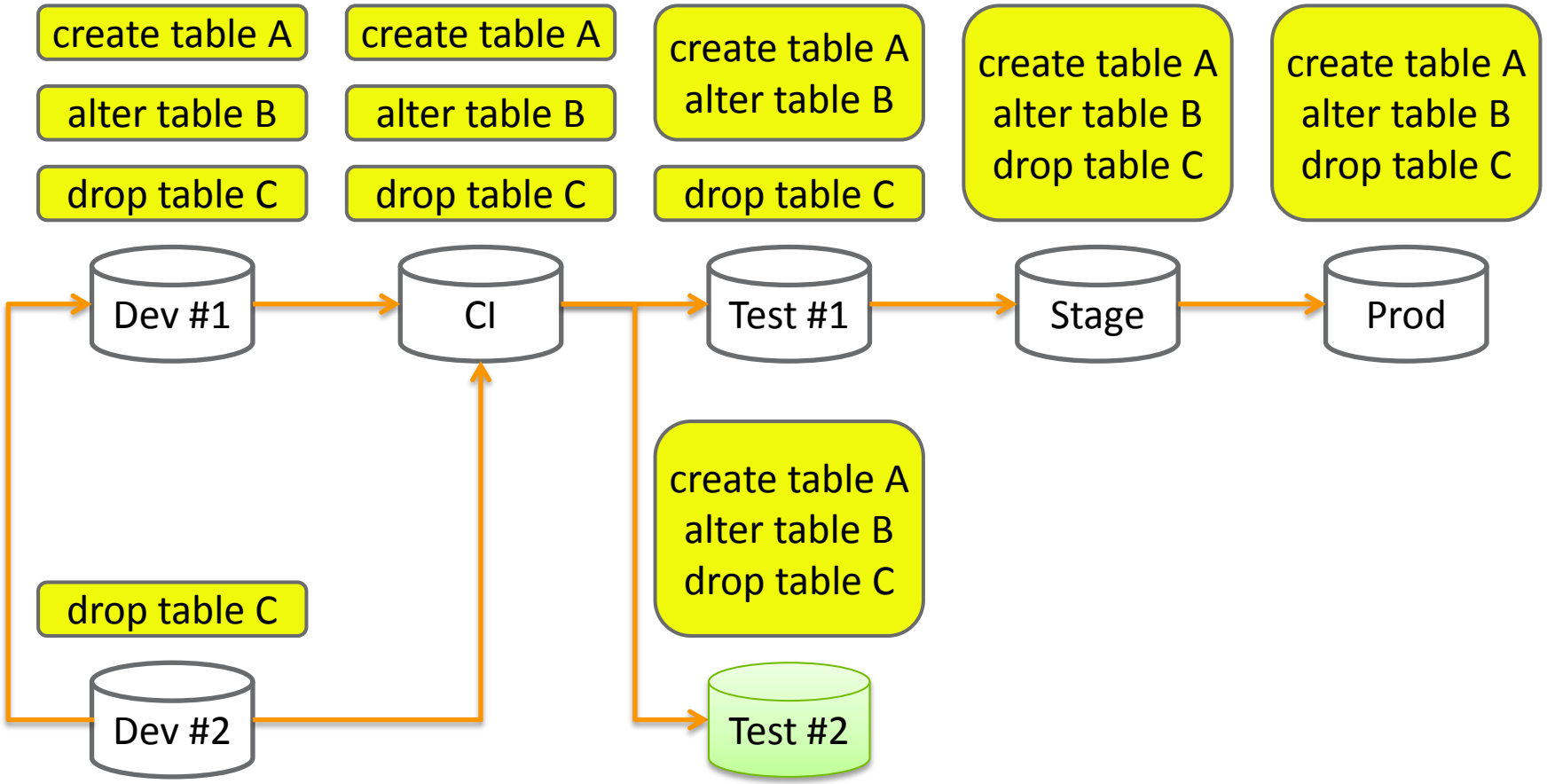
Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation



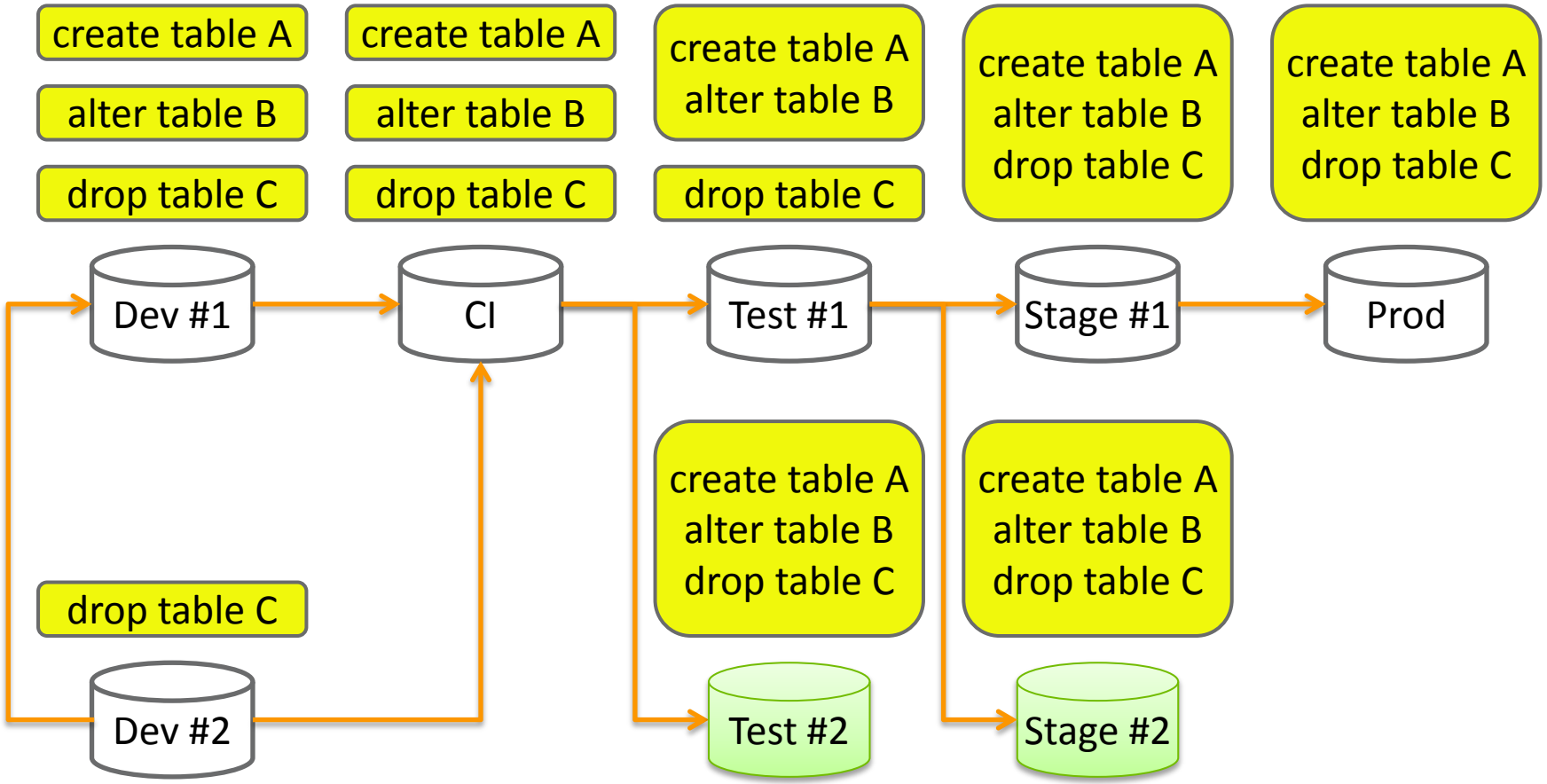
Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation



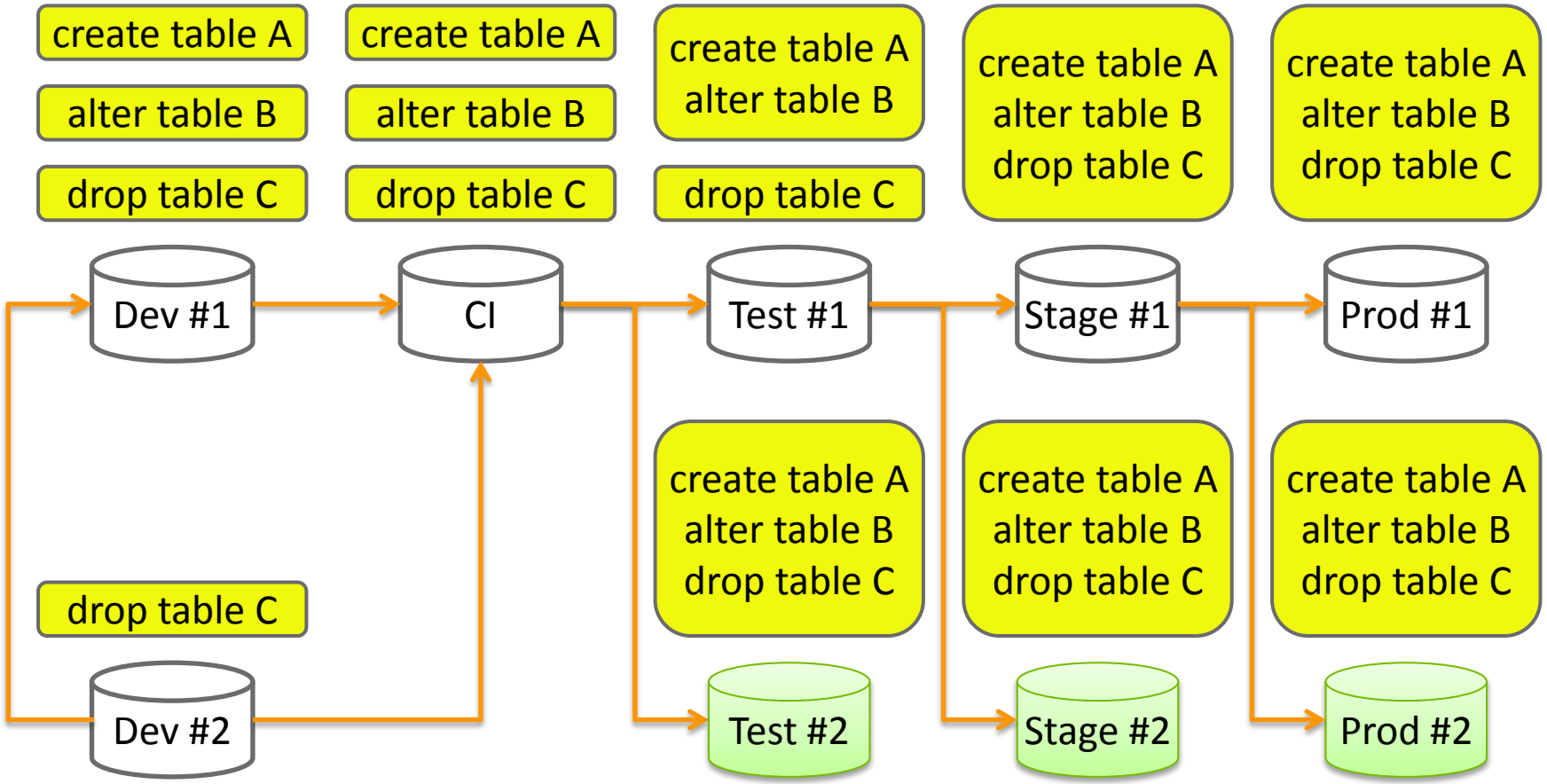
Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation



Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation



Database Change Management mit Liquibase – Das Problem

buschmais
Beratung . Technologie . Innovation

- Fragen
  - Welche Änderungen (Inkremente)…
    - …müssen für einen neuen Stand der Anwendung…
    - …auf eine konkrete Datenbankumgebung…
    - …angewendet werden?
  - Wer…
    - …ermittelt auf welchem Wege die Änderungen…
    - …und führt diese…
    - …in der jeweiligen Umgebung aus?

buschmais

Beratung . Technologie . Innovation

Database Change Management mit Liquibase

Das Werkzeug im Überblick

Database Change Management mit Liquibase – Das Werkzeug im Überblick

buschmais
Beratung . Technologie . Innovation

- Homepage: http://www.liquibase.org
- Source Code: https://github.com/liquibase/liquibase
- Lizenz: Apache License, Version 2.0

- Aktuelle Version: 2.0.5
- Initiale Veröffentlichung: Herbst 2006

Database Change Management mit Liquibase – Das Werkzeug im Überblick

buschmais
Beratung . Technologie . Innovation

- Java-basiertes Werkzeug zur automatisierten Migration von relationalen Datenbank-Schemata
- Integration in den Entwicklungs-, Build- und Auslieferungsprozess via…
  - Maven
  - Ant
  - Grails
  - Servlet Listener
    - Startup der Anwendung
  - Java API
  - Kommandozeile
- Unterstützung gängiger Datenbanken: Oracle, MySQL, MSSQL, PostgreSQL, DB2, Apache Derby, H2, etc.

Database Change Management mit Liquibase – Das Werkzeug im Überblick

buschmais
Beratung . Technologie . Innovation

- Beispiel: Maven

```
<plugin>

</plugin>
```

Database Change Management mit Liquibase – Das Werkzeug im Überblick

buschmais
Beratung . Technologie . Innovation

- Beispiel: Maven

```
<plugin>
  <groupId>org.liquibase</groupId>
  <artifactId>liquibase-maven-plugin</artifactId>
  <version>2.0.5</version>

</plugin>
```

Database Change Management mit Liquibase – Das Werkzeug im Überblick

buschmais
Beratung . Technologie . Innovation

- Beispiel: Maven

```
<plugin>
  <groupId>org.liquibase</groupId>
  <artifactId>liquibase-maven-plugin</artifactId>
  <version>2.0.5</version>
  <configuration>

    <driver>com.mysql.jdbc.Driver</driver>
    <url>jdbc:mysql://localhost:3306/demo</url>
    <username>demo</username>
    <password>demo</password>
  </configuration>
  <dependencies>
    <dependency>
      …JDBC driver…
    </dependency>
  </dependencies>
</plugin>
```

Database Change Management mit Liquibase – Das Werkzeug im Überblick

buschmais
Beratung . Technologie . Innovation

- Beispiel: Maven

```
<plugin>
  <groupId>org.liquibase</groupId>
  <artifactId>liquibase-maven-plugin</artifactId>
  <version>2.0.5</version>
  <configuration>
    <changeLogFile>liquibase/changelog.xml</changeLogFile>
    <driver>com.mysql.jdbc.Driver</driver>
    <url>jdbc:mysql://localhost:3306/demo</url>
    <username>demo</username>
    <password>demo</password>
  </configuration>
  <dependencies>
    <dependency>
      …JDBC driver…
    </dependency>
  </dependencies>
</plugin>
```

Database Change Management mit Liquibase – Das Werkzeug im Überblick

buschmais

Beratung . Technologie . Innovation

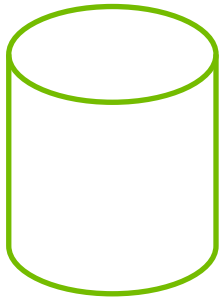
SCM

Database Change Management mit Liquibase – Das Werkzeug im Überblick

buschmais

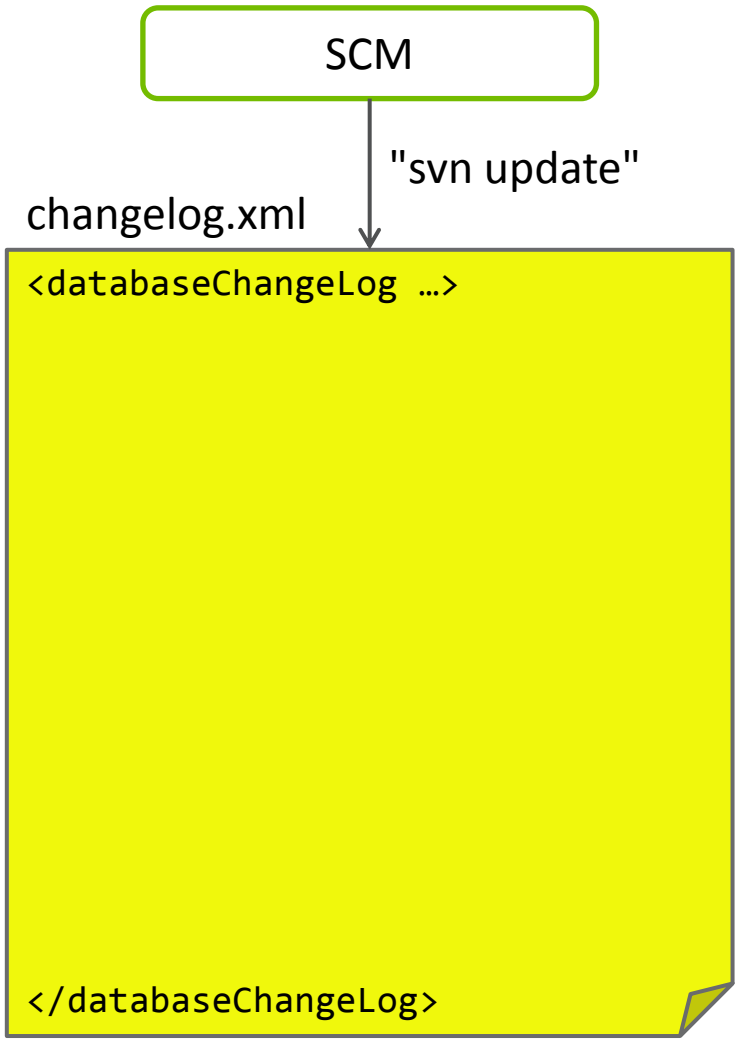
Beratung . Technologie . Innovation

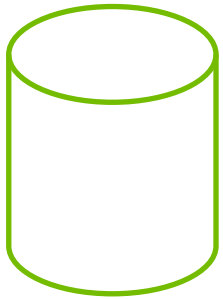
SCM



Database Change Management mit Liquibase – Das Werkzeug im Überblick

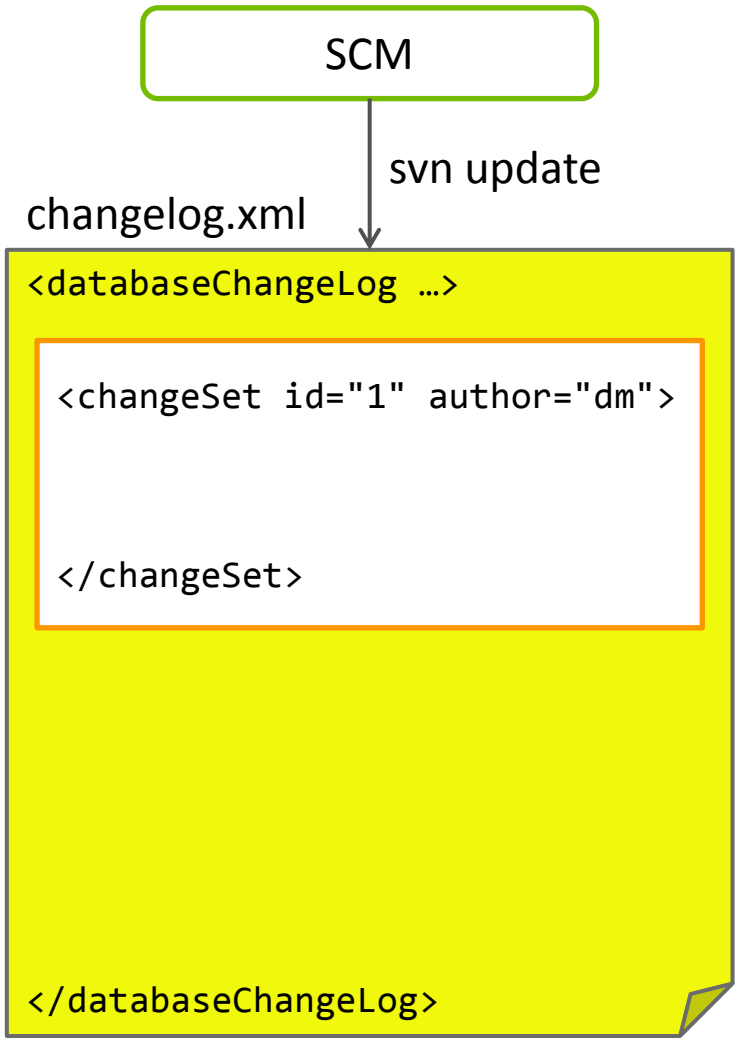
buschmais
Beratung . Technologie . Innovation

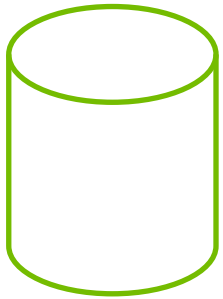




Database Change Management mit Liquibase – Das Werkzeug im Überblick

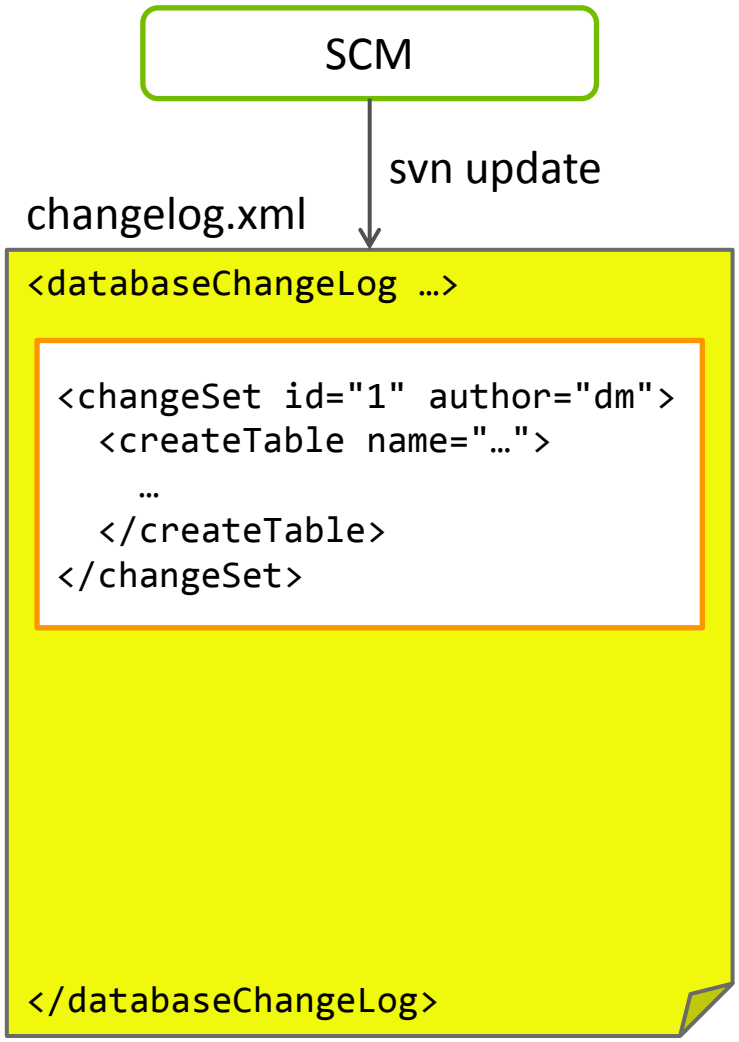
buschmais
Beratung . Technologie . Innovation

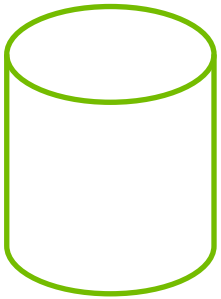




Database Change Management mit Liquibase – Das Werkzeug im Überblick

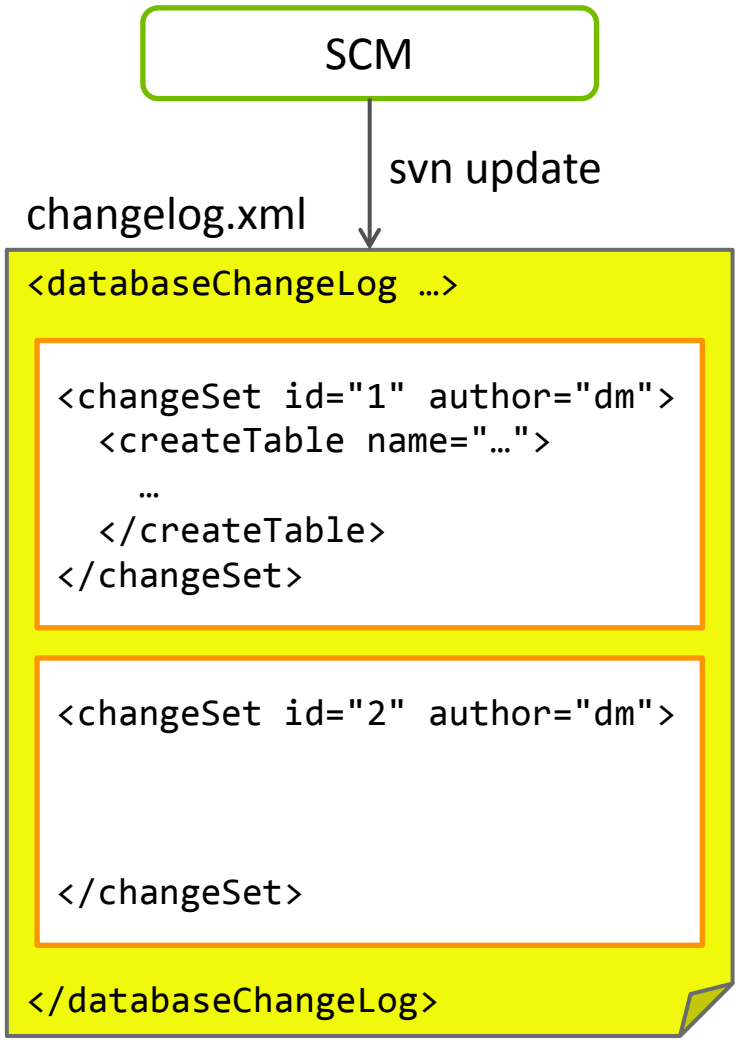
buschmais
Beratung . Technologie . Innovation

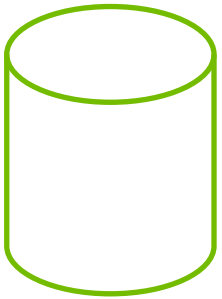




Database Change Management mit Liquibase – Das Werkzeug im Überblick

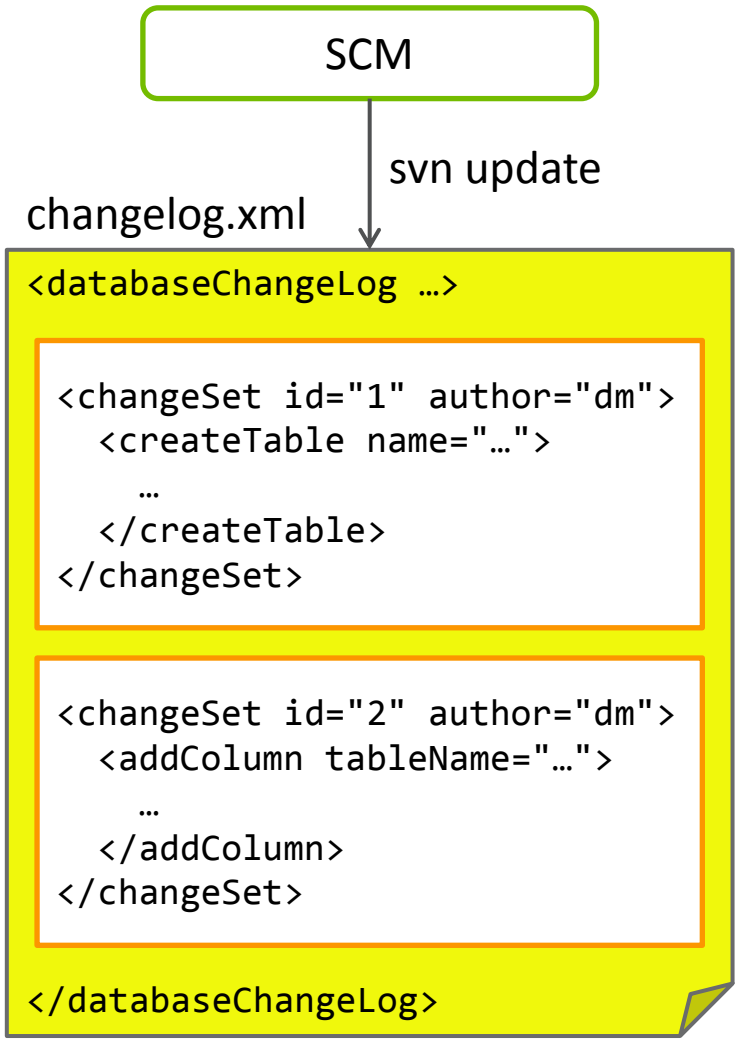
buschmais
Beratung . Technologie . Innovation

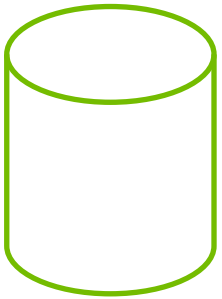




Database Change Management mit Liquibase – Das Werkzeug im Überblick

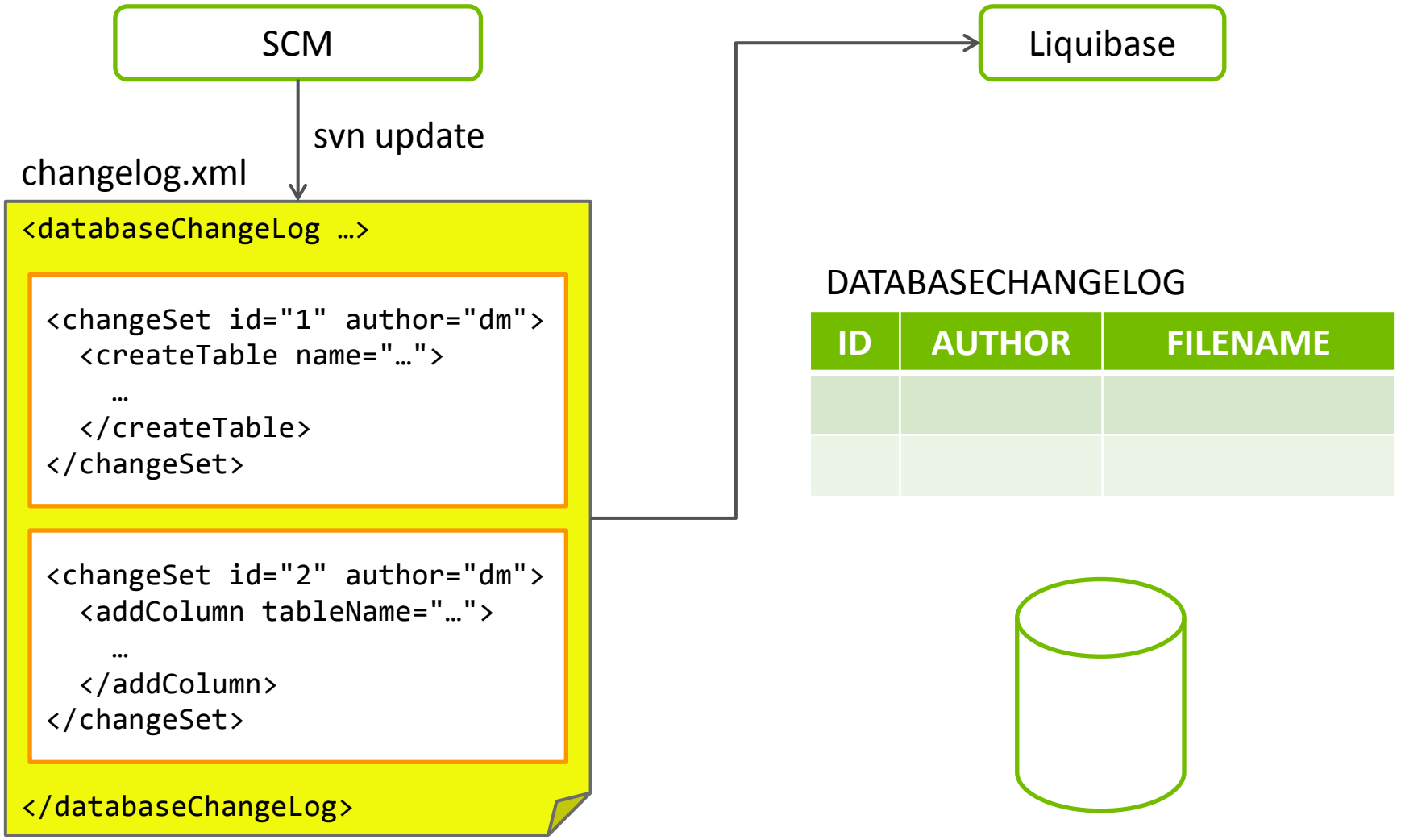
buschmais
Beratung . Technologie . Innovation





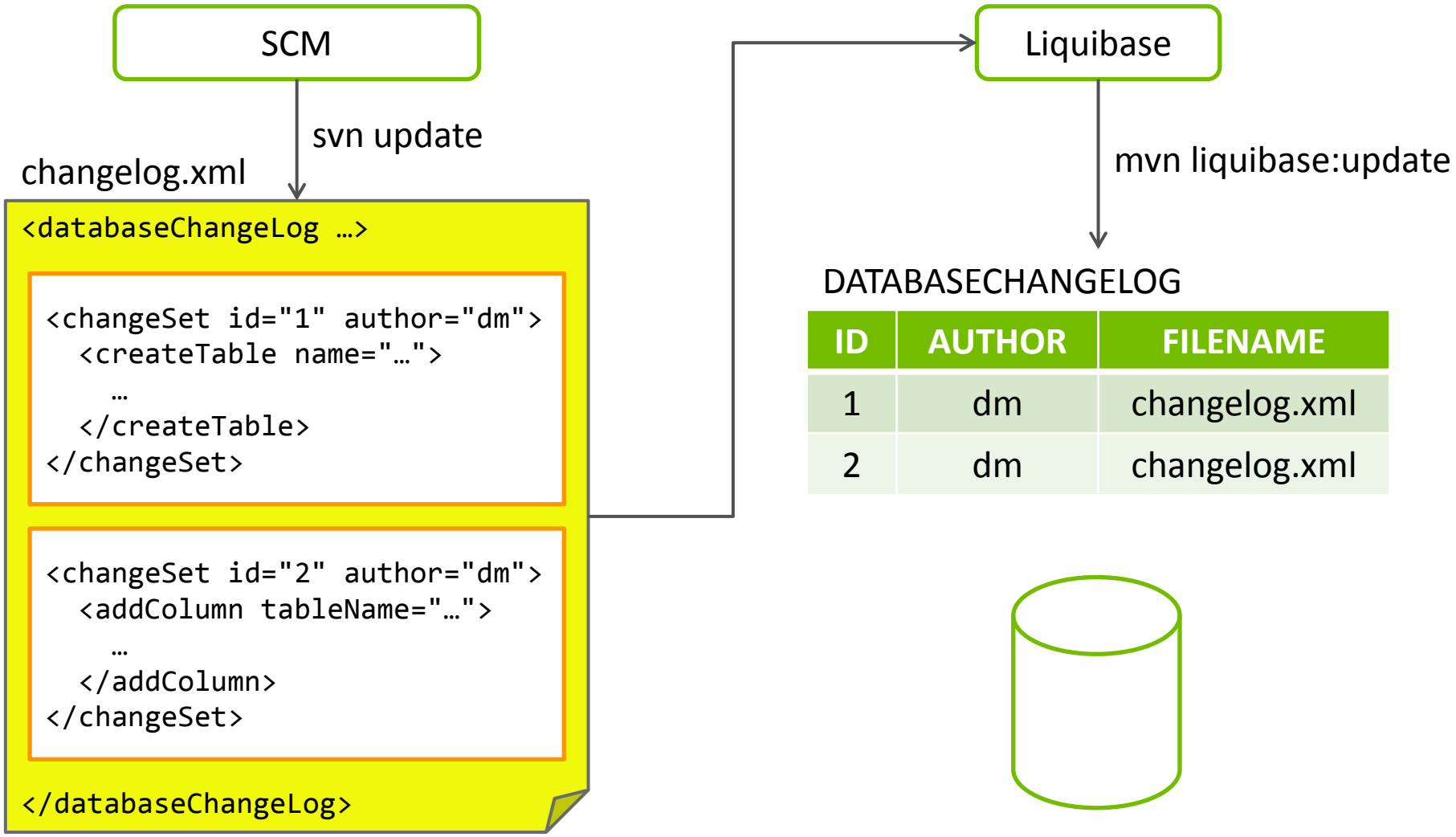
Database Change Management mit Liquibase – Das Werkzeug im Überblick

buschmais
Beratung . Technologie . Innovation



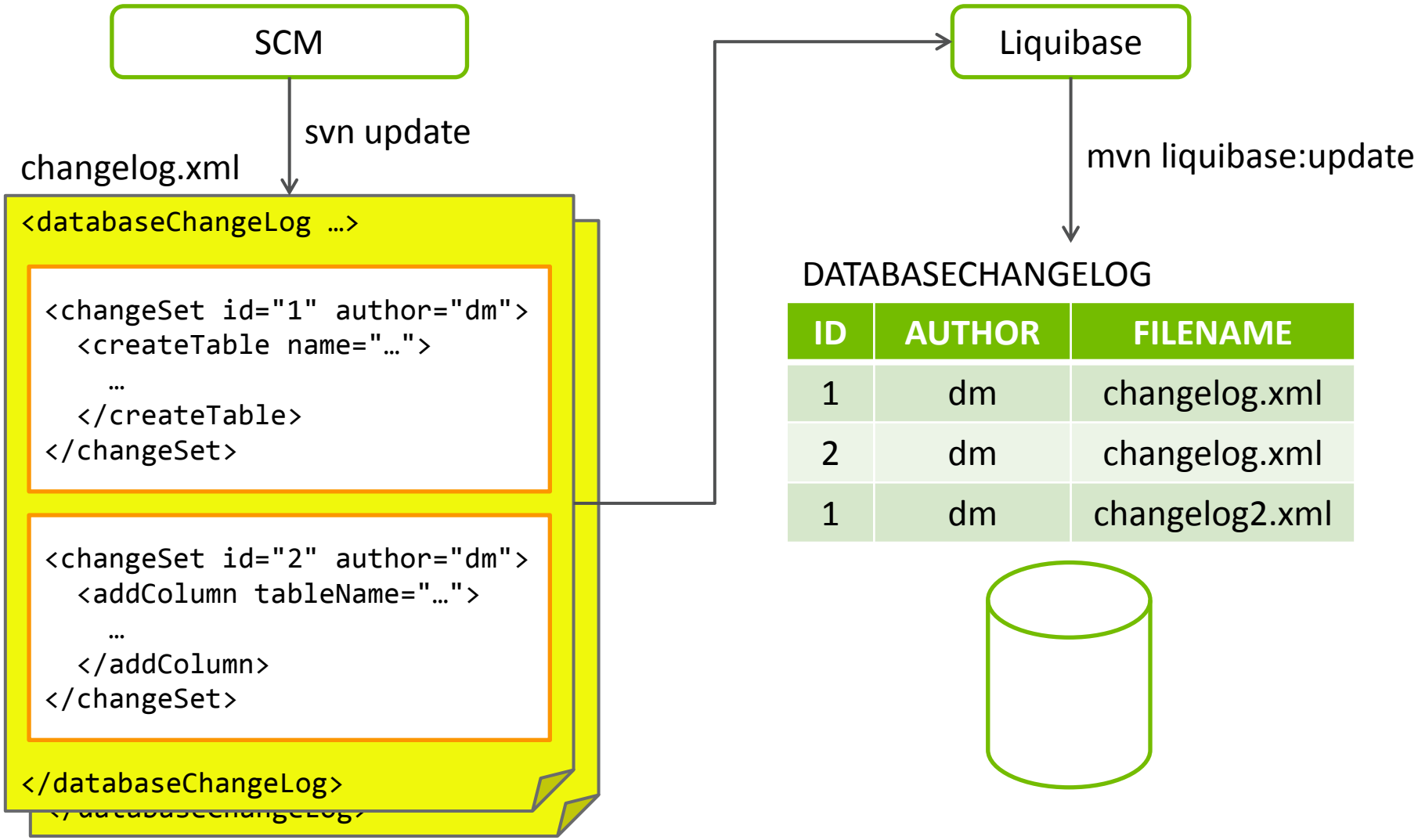
Database Change Management mit Liquibase – Das Werkzeug im Überblick

buschmais
Beratung . Technologie . Innovation



Database Change Management mit Liquibase – Das Werkzeug im Überblick

buschmais
Beratung . Technologie . Innovation



Database Change Management mit Liquibase – Das Werkzeug im Überblick

buschmais
Beratung . Technologie . Innovation

- Refactorings
  - Anweisungen zur Migration des Datenbankschemas
  - "Liquibase-DSL" = DBMS-unabhängige XML-Tags
    - Structural Refactorings, z.B. "Create Table"
    - Data Quality Refactorings, z.B. "Add Unique Constraint"
    - Referential Integrity Refactorings, z.B. "Add Foreign Key"
    - Non-Refactoring Transformations, z.B. "Insert Data"
    - Architectural Refactorings, z.B. "Create Index"
  - Custom SQL
    - Erlaubt DBMS-spezifische DDL bzw. DML
    - Statements in XML-Dateien oder
    - Import von SQL-Skripten

Refactorings (Fortsetzung)

Structural Refactorings Add Column, Rename Column, Modify Column, Drop Column, Alter Sequence, Create Table, Rename Table, Drop Table, Create View, Rename View, Drop View, Merge Columns, Create Stored Procedure

Data Quality Refactorings Add Lookup Table, Add Not-Null Constraint, Remove Not-Null Constraint, Add Unique Constraint, Drop Unique Constraint, Create Sequence, Drop Sequence, Add Auto-Increment, Add Default Value, Drop Default Value

Referential Integrity Refactorings Add Foreign Key Constraint, Drop Foreign Key Constraint, Drop All Foreign Key Constraints, Add Primary Key Constraint, Drop Primary Key Constraint

Non-Refactoring Transformations Insert Data, Load Data, Load Update Data, Update Data, Delete Data, Tag Database, Stop

Architectural Refactorings Create Index, Drop Index

Custom Refactorings Modifying Generated SQL, Custom SQL, Custom SQL File, Custom Refactoring Class, Execute Shell Command

Database Change Management mit Liquibase – Das Werkzeug im Überblick

buschmais
Beratung . Technologie . Innovation

- Beispiel: changelog.xml

```
<databaseChangeLog xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog
    http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-2.0.xsd">

  <changeSet id="1" author="dm">
    <createTable tableName="PERSON">
      <column name="ID" type="BIGINT" autoIncrement="true">
        <constraints primaryKey="true" nullable="false" />
      </column>
      <column name="FIRSTNAME" type="VARCHAR(255)" />
      <column name="LASTNAME" type="VARCHAR(255)" />
    </createTable>
  </changeSet>
  …
</databaseChangeLog>
```

- Integrität von Änderungen
  - Was passiert, wenn ein ChangeSet im Nachhinein geändert ("korrigiert") wird?
    - Ein Entwickler hat im Normalfall nach einem Check-In des Quellcodes keine Information darüber, in welchen Umgebungen das ChangeSet bereits ausgeführt wurde
    - Nachträgliche Änderungen können zu Inkonsistenzen führen
  - Schutz ausgeführter ChangeSets mit einer MD5-Summe
    - MD5-Summe wird in der Tabelle DATABASECHANGELOG hinterlegt
    - Prüfung bei jeder Ausführung und Abbruch der Migration bei erkannten Änderungen
  - Korrekturen sind im Normalfall nur über kompensierende ChangeSets möglich

- Konkurrierende Migrationen
  - In einer Umgebung darf nur eine Migration ausgeführt werden
  - Liquibase realisiert Sperren unter Nutzung einer weiteren Tabelle:

DATABASECHANGELOGLOCK

ID	LOCKED	LOCKGRANTED	LOCKEDBY
1	true	2012-06-24 23:00:22.0	Mythenmetz (169.254.32.251)

- DATABASECHANGELOG und DATABASECHANGELOGLOCK
  - Exklusive Verwaltung durch Liquibase (inkl. Erzeugung)
  - dürfen nicht verändert werden

buschmais

Beratung . Technologie . Innovation

Database Change Management mit Liquibase

Demo #1

buschmais

Beratung . Technologie . Innovation

Database Change Management mit Liquibase

Erweiterte Konzepte

Database Change Management mit Liquibase – Erweiterte Konzepte

buschmais
Beratung . Technologie . Innovation

- ChangeSets können neben Refactorings weitere Anweisungen beinhalten
  - Ausführung
    - Vorbedingungen
    - Wiederholte Ausführung
    - Fehlerverhalten
  - Rollback-Strategien
  - Kommentare für Change-Dokumentation

Database Change Management mit Liquibase – Erweiterte Konzepte

buschmais
Beratung . Technologie . Innovation

- ChangeSet-Parameter
  - runOnChange
    - das ChangeSet wird bei erkannter Änderung (MD5SUM) erneut ausgeführt
  - runAlways
    - das ChangeSet wird immer ausgeführt
    - sinnvoll für Stored Procedures
      - Definition in speziellem DatabaseChangeLog
      - Änderungshistorie kann im SCM gut nachvollzogen werden
  - failOnError
    - Bestimmt, ob die Ausführung der Migration bei Auftreten eines Fehlers abgebrochen werden sollen
    - Standardwert: true

Database Change Management mit Liquibase – Erweiterte Konzepte

buschmais
Beratung . Technologie . Innovation

- ChangeSet-Parameter (Fortsetzung)
  - dbms
    - Das ChangeSet wird nur auf dem angegebenen Datenbanksystem ausgeführt, z.B. dbms="mysql"
  - context
    - Das ChangeSet wird nur ausgeführt, wenn einer der angegebenen Kontexte aktiviert wurde, z.B. context="test"
    - Aktivierung über Parameter bei der Ausführung, z.B. Maven

```
<plugin>
  <groupId>org.liquibase</groupId>
  <artifactId>liquibase-maven-plugin</artifactId>
  <version>2.0.5</version>
  <configuration>
    <contexts>test</contexts>
    …
```

- ChangeSet-Parameter (Fortsetzung)
  - Beispiel

```
<changeSet id="1" author="dm"

  runOnChange="true"
  runAlways="false"
  failOnError="false"
  dbms="oracle"
  context="test">

  <-- Refactorings -->

</changeSet>
```

Database Change Management mit Liquibase – Erweiterte Konzepte

buschmais
Beratung . Technologie . Innovation

- ChangeSet-Elemente
  - PreConditions
    - Bedingte Ausführung eines ChangeSets, Beispiel:

```
<changeSet id="3" author="dm">
  <preConditions onFail="CONTINUE">
    <and>
      <runningAs username="demo"/>
      <changeSetExecuted changeLogFile="changelog.xml"
        author="dm" id="1"/>
      <tableExists tableName="SOME_TABLE"/>
      <sqlCheck expectedResult="0">SELECT COUNT(*) from
        SOME_TABLE</sqlCheck>
    </and>
  </preConditions>
  <addUniqueConstraint …
```

- ChangeSet-Elemente (Fortsetzung)
  - comment
    - Angabe eines Kommentars
    - Wird in der Tabelle DATABASECHANGELOG abgelegt
    - Auswertung durch DBDoc-Operation

```
<changeSet id="3" author="dm">
  <comment>Uniqueness of first name and last name must be
  guaranteed.</comment>

  <addUniqueConstraint …

</changeSet>
```

Database Change Management mit Liquibase – Erweiterte Konzepte

buschmais
Beratung . Technologie . Innovation

- ChangeSet-Elemente (Fortsetzung)
  - Rollback
    - Liquibase unterstützt Rollback von ChangeSets
    - Achtung: nicht alle Refactorings lassen sich zurückrollen
      - dropTable, dropColumn, …
    - Vordefinierte Rollback-Strategien
      - z.B. createTable -> dropTable
    - "rollback"-Element dient der Angabe einer nutzerdefinierten Strategie für ein ChangeSet:
      - Liquibase-Refactoring (z.B. createTable)
      - SQL
      - Referenz auf ein ChangeSet

Database Change Management mit Liquibase – Erweiterte Konzepte

buschmais
Beratung . Technologie . Innovation

- ChangeSet-Elemente (Fortsetzung)
  - Beispiel für Rollback-Strategien

```
<changeSet id="3" author="dm">
  <rollback>
    <dropTable tableName="PERSON" />
    <rollback changeSetAuthor="dm" changeSetId="1" />
    <sql>DROP TABLE PERSON</sql>
  </rollback>
 <createTable tableName="PERSON">
  …
```

Database Change Management mit Liquibase – Erweiterte Konzepte

buschmais
Beratung . Technologie . Innovation

- Operationen
  - tag
    - Setzen eines Tags in der aktuellen Umgebung
    - Ermöglicht späteres Rollback

```
mvn liquibase:tag -Dliquibase.tag=1.0
```

  - rollback
    - Zurückrollen von Änderungen in der aktuellen Umgebung
    - Unterstützte Modi: Tag, Count und Date

```
mvn liquibase:rollback -Dliquibase.rollbackTag=1.0
mvn liquibase:rollback -Dliquibase.rollbackCount=1
mvn liquibase:rollback -Dliquibase.rollbackDate=05.07.2012
```

Database Change Management mit Liquibase – Erweiterte Konzepte

buschmais
Beratung . Technologie . Innovation

- Operationen (Fortsetzung)
  - dropAll
    - Entfernt alle Datenbank-Objekte aus dem aktuellen Schema
    - Ausnahme: Functions, Procedures, Packages

```
mvn liquibase:dropAll
```

  - updateSQL bzw. rollbackSQL
    - Ausgabe der SQL-Statements, welche ausgeführt werden müßten, um die Umgebung zu migrieren bzw. zurückzurollen
    - Ggf. notwendig für Staging- bzw. Produktiv-Umgebungen, in denen Migrationsskripte einem Review unterzogen werden
    - Lesender Zugriff auf die Datenbank ist trotzdem nötig!

```
mvn liquibase:updateSQL bzw. mvn liquibase:rollbackSQL
```

Database Change Management mit Liquibase – Erweiterte Konzepte

buschmais
Beratung . Technologie . Innovation

- Operationen (Fortsetzung)
  - status
    - Ausgabe aller auszuführenden ChangeSets

```
mvn liquibase:status
```

  - dbDoc
    - Erzeugung einer Javadoc-ähnlichen Dokumentation aller bereits ausgeführten und noch auszuführenden ChangeSets

```
mvn liquibase:dbDoc
```

Database Change Management mit Liquibase – Erweiterte Konzepte

buschmais
Beratung . Technologie . Innovation

- Operationen (Fortsetzung)
  - generateChangeLog
    - Erzeugung einer initialen DatabaseChangeLog-Datei
      - Bootstrapping für Liquibase
      - Erstellen einer neuen Baseline
    - Nur auf der Kommandozeile verfügbar
  - diff
    - Vergleich der aktuellen Umgebung mit einer Referenzumgebung
    - Ausgabe der Differenzen als ChangeSets oder Reports

buschmais

Beratung . Technologie . Innovation

Database Change Managment mit Liquibase

Demo #2

buschmais

Beratung . Technologie . Innovation

Database Change Management mit Liquibase

Praxiserfahrungen

Database Change Management mit Liquibase – Praxiserfahrungen

buschmais
Beratung . Technologie . Innovation

- Erfahrungen mit Liquibase
  - Einsatz in eigenen Projekten seit mehr als 3 Jahren
    - Abdeckung der gesamten Auslieferungskette von der Entwicklung bis hin zur Produktion
    - Bewährtes und robustes Werkzeug
  - Anfangs etwas gewöhnungsbedürftig für Entwickler
    - "Warum darf ich das ChangeSet nicht mehr ändern?"
    - Probleme durch Quellcode-Formatierungen ("Strg-Shift-F")
    - Alle Änderungen an Datenbankstrukturen müssen konsequent über Liquibase umgesetzt werden
  - Entwicklung in Branches ist ohne Probleme möglich
    - Empfehlung: Verwendung dedizierter Entwickler-Schemata, d.h. ein Schema pro Branch und Entwickler
    - Rollback ist ein Feature, das wir nie genutzt haben…

Database Change Management mit Liquibase – Praxiserfahrungen

buschmais
Beratung . Technologie . Innovation

- Erfahrungen mit Liquibase (Fortsetzung)
  - Empfehlung: Aufteilen der DatabaseChangeLogs anhand der jeweils aktuellen Anwendungsversion:

```
liquibase/master.xml
liquibase/1.0.0.xml
liquibase/1.1.0.xml
liquibase/1.2.0.xml
```

  - Inhalt von master.xml

```
<databaseChangeLog …>
  <include file="liquibase/1.0.0.xml" />
  <include file="liquibase/1.1.0.xml" />
  <include file="liquibase/1.2.0.xml" />
</databaseChangeLog>
```

buschmais
Beratung . Technologie . Innovation

Vielen Dank!


buschmais.de


facebook.com/buschmais


twitter.com/buschmais