Datenqualität im DWH
Ist Automatisch
auch gleich besser??

Edgar Kaemper (AA-AS/EIS3-EU)
Robert Bosch GmbH
Plochingen

Schlüsselworte

Data Warehouse, Datenbewirtschaftung & ETL, Datenmodellierung, Datenqualität, DWH & Datenintegration

Einleitung

Führt eine Automatisierung von Testaktivitäten auch zu einer besseren Datenqualität? Wie kann man auch mit wenig Aufwand Testfälle in einem DWH Projekt automatisieren? Welche Prozesse sind nötig, um ein Testframework mit Leben zu füllen?

In einem DWH Projekt haben wir die Ausführung von Testfällen automatisiert. Dazu wurde ein Datenmodell entwickelt und ein PL/SQL Framework für die Ausführung und Bewertung von Testfällen implementiert.

Der Vortrag stellt das Vorgehen und unsere Erfahrungen dabei dar. Wir erläutern das Framework und das Datenmodell für die Testautomatisierung. Auch die Prozesse für die Definition der Testfälle und die Aufbereitung und Auswertung der Testergebnisse im DWH Umfeld stellen wir dar und zeigen auf, wie das Zusammenspiel von Technik und Prozessen zu einer Verbesserung der Datenqualität führt.

Umfeld

Der Bosch Geschäftsbereich Automotive Aftermarket (AA) bietet Handel und Werkstätten weltweit die komplette Diagnose- und Werkstatttechnik sowie ein umfassendes Kfz- und Nfz-Ersatzteilsortiment - vom Neuteil über instandgesetzte Austauschteile bis hin zur Reparaturlösung. Das Produktportfolio von AA besteht aus Erzeugnissen der Bosch Erstausrüstung sowie aus eigenentwickelten und -gefertigten Aftermarket-spezifischen Produkten und Dienstleistungen. Über 18.000 Mitarbeiter in 150 Ländern sowie ein weltweiter Logistikverbund stellen sicher, dass mehr als 650.000 verschiedene Ersatzteile schnell und termingerecht zum Kunden kommen.

AA bietet unter der Bezeichnung "Automotive Service Solutions" Prüf- und Werkstatttechnik, Software für Diagnose, Service-Training sowie technische Informationen und Serviceleistungen.

Der Geschäftsbereich ist auch verantwortlich für die Werkstattkonzepte Bosch Service, eine der größten unabhängigen Werkstattketten weltweit mit rund 16.500 Betrieben, und AutoCrew mit über 800 Betrieben.

Die wachsende Anzahl und die steigende Komplexität der im Fahrzeug installierten Systeme und Komponenten bedeutet, dass Service-Werkstätten einen Zugang zu breitem Wissen haben müssen. Informationssysteme in der Werkstatt (z.B. ESI[tronic]) müssen praktisch jedes Fahrzeugmodell erkennen und umfassende Informationen für die Werkstätten liefern.

Ausgangslage

Mit dem Fahrzeug-Diagnose und Werkstattinformationssystem ESI[tronic] werden für Werkstätten u.A. folgende Informationen und Funktionen bereitgestellt:

- Steuergeräte-Diagnose mit neuesten Daten für Pkw-, Transporter- und Lkw-Systeme
- Fehlersuche mit geführten Suchanleitungen
- Daten für Inspektion und Service
- Komfortschaltpläne, um Fehler im System schnell lokalisieren zu können
- Schnellzugang zu bekannten Fehlern mit den Technischen Service Informationen

Für die dargestellte Datenarchitektur ist von Bedeutung, dass neben strukturierten Daten auch Dokumente (z.B. Fehlersuchanleitungen, Ein- und Ausbaubeschreibungen, …) und Medieninhalte (z.B. interaktive Schaltpläne, Bilder zur Einbaulage von Fahrzeugkomponenten, ...) von großer Bedeutung sind und große Teile des Datenvolumens auf diese Daten entfallen.

Architektur des CDW

Ziel der Architektur des Central Diagnostic Warehouses (CDW) bei Bosch ist es, alle diagnoserelevanten Daten in einer Datenbank und in einem Datenmodell zu konsolidieren und online und offline Applikationen zur Verfügung zu stellen. Neben den Applikationen sollen die Daten auch für Online Services, Datenexporte und Reporting/Analysen verwendet werden können. Die CDW Architektur ist in die für DWH typischen Layer Staging, Cleansing und Core DWH aufgeteilt. Besonderheiten sind:

- **Internal Loadstore**: Eine Archivierung der Quelldatenlieferungen als Snapshots, jede Anlieferung ist in der Regel ein Full Load und wird komplett zu Analysezwecken im Internal Loadstore gesichert.

- **CDW release**: Die Bildung der Datenhistorie erfolgt nicht beim Übergang von Cleansing in den CDW Core Bereich sondern innerhalb des Core nach einer Freigabe der Daten.

- **Feedback**: Aus den Anwendungen bei den Kunden kommen Daten über die Nutzung und Fehlermeldungen zurück.

- **ODS**: Im Operational Datastore werden Daten gespeichert, die auf Grund Ihrer Beschaffenheit (noch) nicht korrekt in das CDW Datenmodell integriert werden können, für einzelne Analysen jedoch Zusatznutzen liefern.

- **Document Generator:** Das CDW enthält neben den relationalen Daten auch Dokumente (z.B. Fehlersuchanleitungen), die aus den Quellinformationen in mehreren Sprachen generiert werden.

- **Dump Generator:** Das CDW verfügt über ein einfaches Configuration Management, um jedem Empfänger von Daten mit Hilfe von Einträgen in Konfigurationstabellen nur die Daten im Datenbank dump zur Verfügung zu stellen, die von diesem Empfänger benötigt werden.

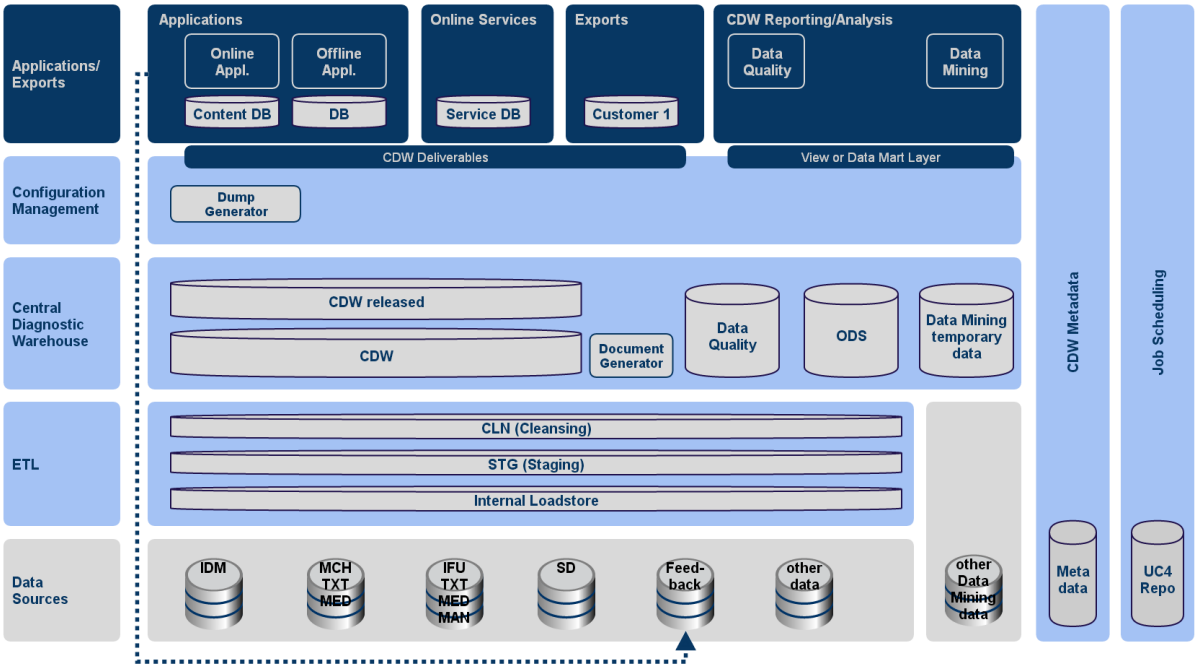

Abb. 1: Central Diagnostic Warehouse (CDW) Data Architecture

Datenqualitätskonzept

Um das Datenqualitätskonzept zu erläutern, zeigen wir die Architektur eines DWH in vereinfachter Form in Abbildung 2. Aus mehreren Quellen werden Daten durch ETL Prozesse in eine Datenbank konsolidiert. Nach dem Konsolidierungsprozess ist eine wichtige Frage, ob die Datenqualität der konsolidierten Daten ausreicht, um ausgeliefert zu werden. Der rein technisch erfolgreiche Lauf der ETL Prozesse muss nicht zwangsläufig zu korrekten Daten führen. Z.B. können geänderte Masterdaten (z.B. Mapping von Informationen auf Fahrzeugkomponenten) den ETL Lauf technisch nicht beeinflussen, jedoch zu falschen Daten führen.

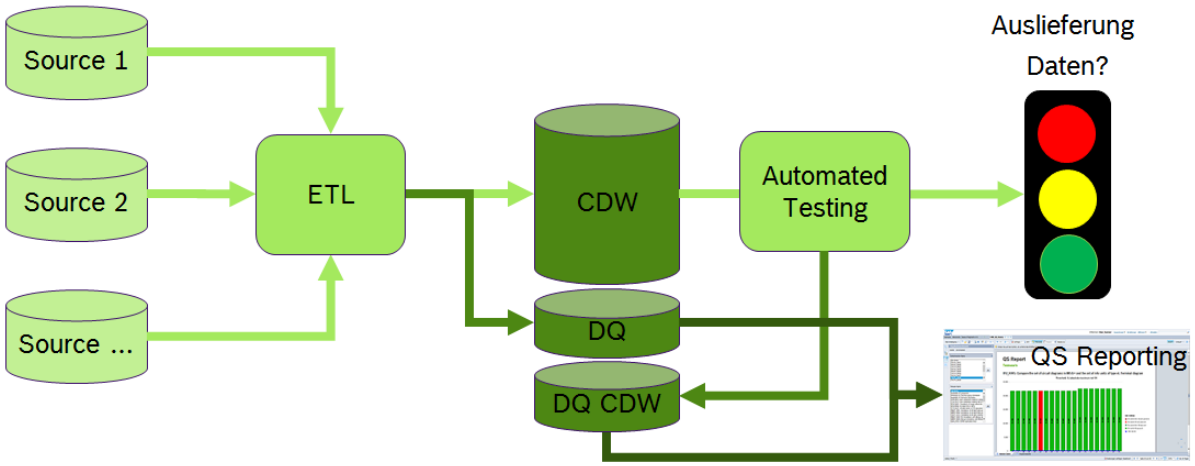

Abb. 2: Datenqualitätskonzept

Eine erste Komponente in unserem Datenqualitätskonzept ist die **Eingangskontrolle.** Hier prüfen Domain Experten mit Hilfe der QS-Reports der internen und externen Datenquellen, ob wesentliche Gründe vorliegen, die Daten nicht zu laden. Die QS-Reports dienen auch dazu, die Qualität der Daten langfristig in den regelmäßigen Gesprächen mit den Datenlieferanten weiter zu verbessern.

Eine weitere Komponente in unserem Datenqualitätskonzept ist die Einführung von **Reject-Tabellen** in einem separaten Schema (DQ) der Datenbank. Hier werden alle Datensätze gespeichert, die nicht in das CDW geladen werden konnten, weil z.B. Datenformate, Not-Null Constraints oder Foreign Key Relations verletzt wurden. Damit sind die Rejects ein Mittel um die technische Datenqualität zu unterstützen.

Um die inhaltliche Datenqualität zu steuern, haben wir sogenannte Domain Experten eingeführt. Diese kümmern sich um die inhaltliche Datenqualität eines Bereichs der Daten. Neben den bereits erwähnten Kontakten zu den Datenquellen oder Datenlieferanten gehört dazu auch die Definition von Testfällen für das automatisierte Testen von Daten

Für das automatisierte Testen von Daten haben wir ein Testframework eingeführt. Dieses überprüft komplexe Regeln, die durch die Constraints der Datenbank nicht abgebildet werden können. Zusätzlich hilft das Framework, den Ablauf der Test zu automatisieren. Ziel ist es, möglichst zeitnah nach den ETL Läufen eine technische Freigabe der Daten automatisiert zu erreichen.

Dazu wird nach dem ETL Lauf ein Testresultset angelegt, in dem alle Testfall-Ausführungen des Tests gebündelt werden. Anschließend werden alle Testfälle ausgeführt und die Ergebnisse in einer Result Tabelle gespeichert. Wenn alle Testfälle abgearbeitet sind, wird das Testresultset geschlossen, und für jeden Testfall mit seinen Grenzwerten ermittelt, ob er erfolgreich durchgeführt wurde. Die Ergebnisse werden in einem Testreport zusammengefasst.

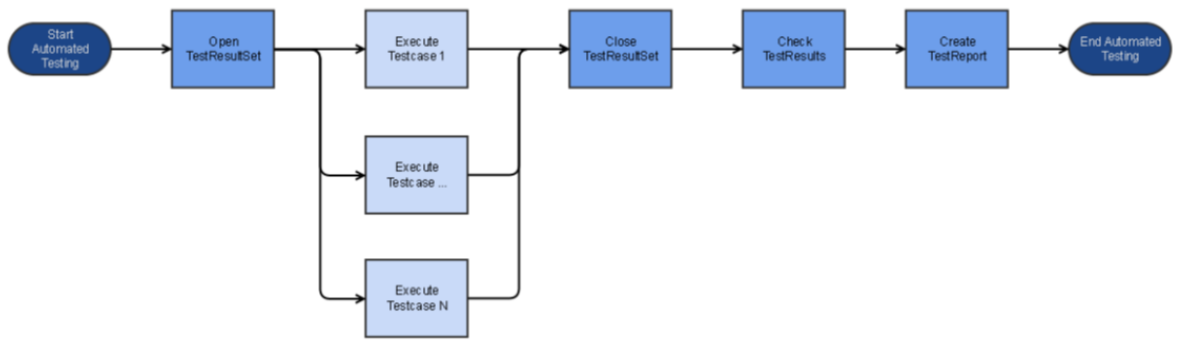

Abb. 3: Prozess für das automatisierte Testen

Datenmodell

Für das Testframework haben wir ein einfaches Datenmodell entwickelt. Neben den schon angesprochenen Tabellen für Testcase, Testresult und Testresultset spielen noch folgende 2 Tabellen eine besondere Rolle:

Signifikanz: Diese soll in der Zukunft als Basis für ein Berechnungsmodel zur Freigabe der Daten dienen. Die Testfälle sind zurzeit 2 Signifikanzen zugeordnet:

- Daten sind falsch, d.h. wir würden bei einem Testcase dieser Art, der nicht erfüllt wird, Daten freigeben, die nicht korrekt sind.
- Es fehlen Daten, d.h. wir würden bei einem Testcase dieser Art, der nicht erfüllt wird, nicht alle vorhandenen Daten freigeben.

Testcase Typen: Diese dienen der automatische Bewertung der Testcases und zeigen an, wie der zu einem Testfall gehörende Grenzwerten zu interpretieren ist.

- absolutes Maximum not OK (Wenn x Datensätze nicht OK sind, dann ist der Testcase nicht bestanden.)
- relatives Maximum not OK (Wenn x % der Datensätze nicht OK sind, dann ist der Testcase nicht bestanden.)
- absolutes Minimum OK (Wenn x Datensätze OK sind, dann ist der Testcase bestanden.)
- relatives Minimum OK (Wenn x % der Datensätze OK sind, dann ist der Testcase bestanden.)
- Vergleich Anzahl zur letztem Datenstand: (Wenn min. x % mehr Datensätze, dann ist der Testcase bestanden.)

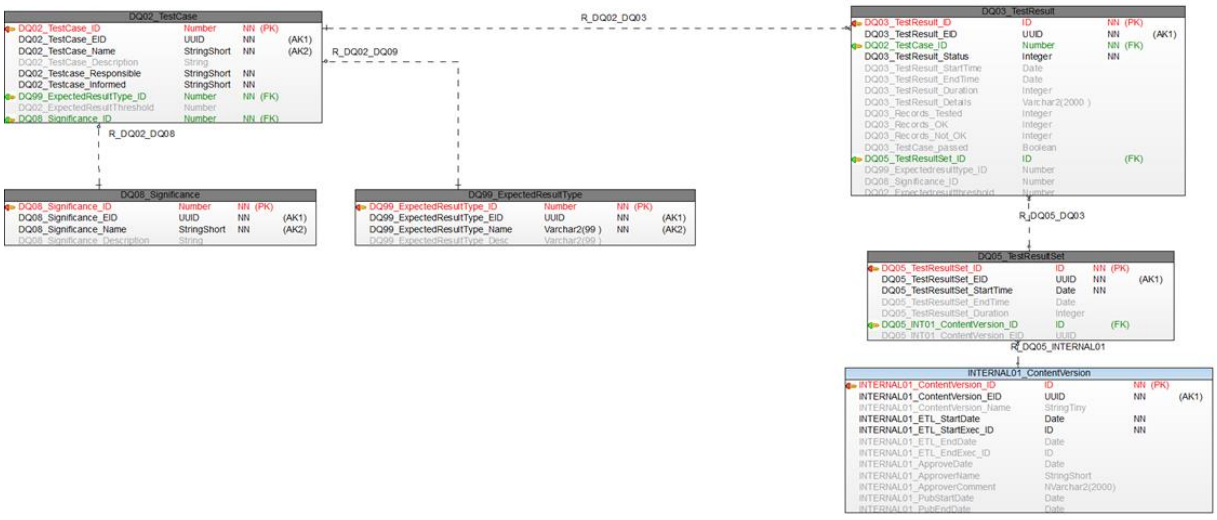

Abb. 4: Datenmodell für das automatisierte Testen

Funktionen für das Testframework

Für die Testcases wird im Wesentlichen eine Funktion benötigt, die die Testscases aus einer Datei mit einer External Table einliest

Für die Testresults gibt es eine Funktion, die diese anlegt und eine Funktion, die nach Ausführung des Testcases die Felder für Records getestet, Records OK, Records nicht OK, Details zu Fehlern, Ende und Dauer updatet. Ein weitere Funktion dient der Bewertung der Testcases und berechnet auf Basis der Grenzwerte ob der Testfall "passed" oder „not passed" abgeschlossen wurde.

Für die Testresultsets gibt es eine Funktion für das Anlegen und eine für den Update der Timestamps und der Dauer nach Ablauf aller Testfälle in einem Testresultset.

Für die Testfälle gibt es zusätzlich eine Procedure, die den Testcase abarbeitet.

Weiterentwicklung des Testframeworks

Für die Weiterentwicklung des Testframeworks wollen wir in 3 Schritten vorgehen:

Close the gap: Auch wenn wir mit den jetzt vorhandenen ca. 50 Testfällen einige Fortschritte bei der Datenqualität erreicht haben, sind 50 Testfälle für ein Datenmodell mit > 200 Tabellen noch nicht ausreichend. Weitere Testfälle zu definieren ist Aufgabe der Domain Experten. Allerdings ist auch die Frage berechtigt, wie viele Testfälle den genug sind um die Qualität von Daten zu beurteilen.

Get the lead: Wir gehen davon aus, dass wir nach der Definition einer ausreichenden Anzahl von Testfällen, die Grenzwerte der Testfälle justieren müssen. Aber auch den Zusammenhang der Ergebnisse aller Testfälle zu einer möglichen technischen Freigabe der Daten wollen wir analysieren und eine Kennzahle im Sine einer Qualitätsampel definieren, die uns anzeigt, ob die Daten freigeben werden können oder nicht.

What else?: Unser Ziel ist die automatische technische Freigabe der Daten, d.h. nach Ablauf der ETL Prozesse und Durchführung der automatischen Tests soll auch automatisch die technische Freigabe der Daten erfolgen.

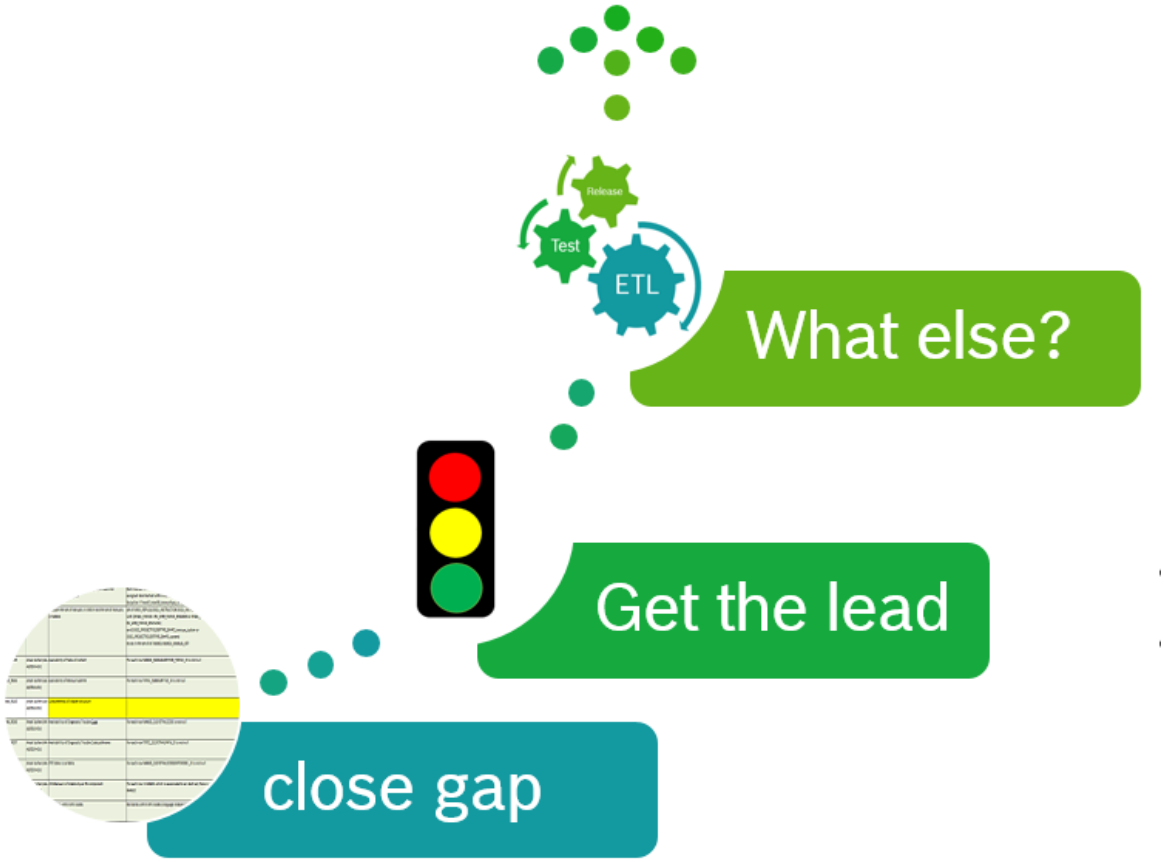

Abb. 5: Weiterentwicklung des Testframeworks

Beispiele für die Ergebnisse von Testcases

Die Ergebnisse werden als Ausschnitte aus dem QS Reporting dargestellt. Auf der X-Achse sind die verschiedenen Datenstände nach den ETL Läufen im Zeitverlauf aufgetragen. Die Y-Achse gibt die Anzahl der Datensätze an. Der untere Säulenteil zeigt die Datensätze, die nicht OK sind, der obere Säulenteil die Datensätze die OK sind. Eine rote Farbe der Säule zeigt an, dass der Testcase nicht bestanden wurde, eine grüne Farbe zeigt einen erfolgreichen Testfall. Die blaue Linie markiert die Grenzwerte.

Bei Testcase 1 sehen wir, das der Testfall nie erfolgreich war. Die Qualität wurde deutlich verbessert, es sind deutlich weniger Datensätze als nicht OK markiert. Allerdings erlaubt der Testfall überhaupt keine falschen Datensätze. Die weitere Analyse wird auch zeigen, ob dieser harte Grenzwert tatsächlich erforderlich ist.

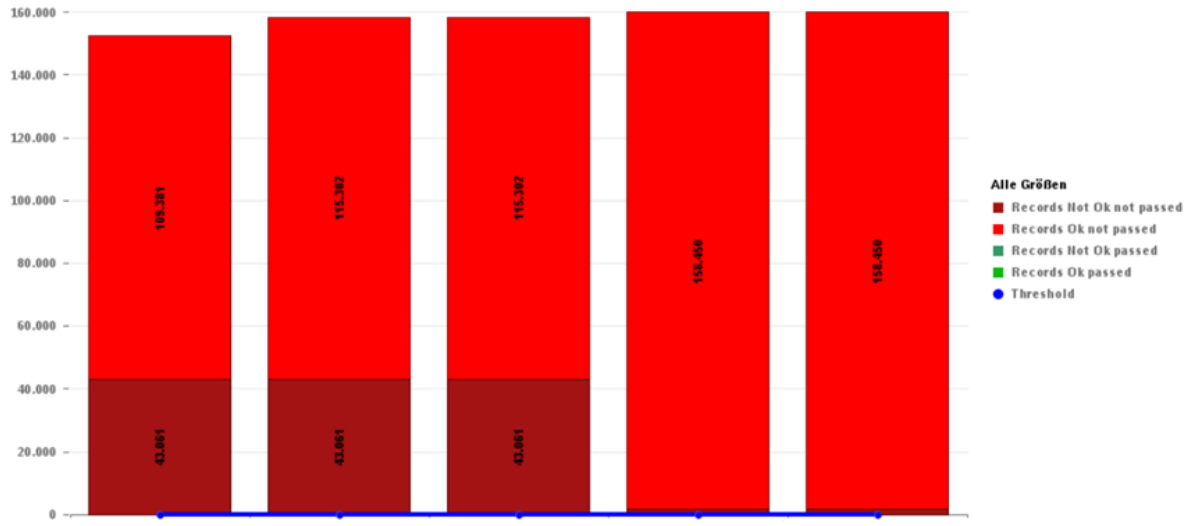

Abb. 5: Ergebnisse Testcase 1

Bei Testcase 2 sehen wir einen Fall, der nie immer bestanden wurde, auch wenn der Grenzwert von 0 falschen Datensätzen anspruchsvoll definiert ist. Auch ein Anstieg der Datenmengen über die Zeit führt nicht zu falschen Datensätzen.

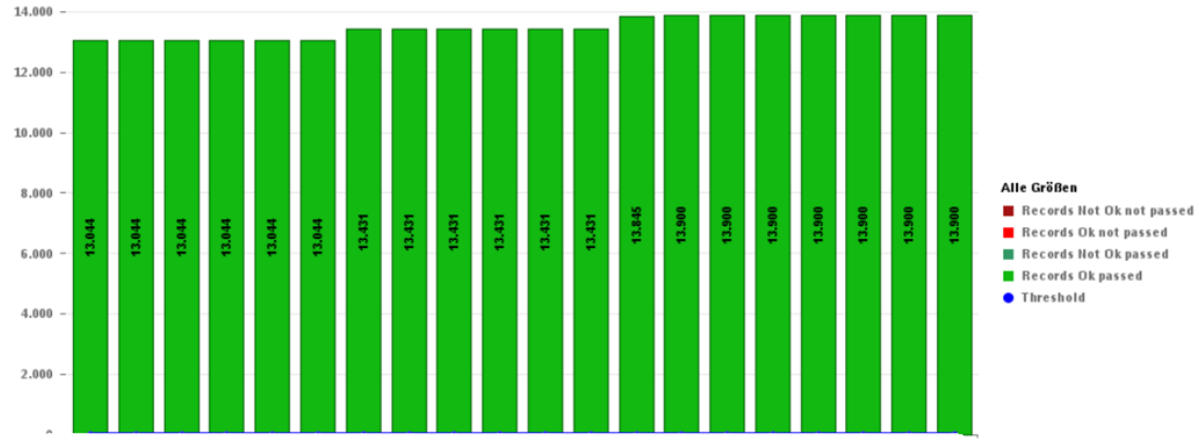

Abb. 6: Ergebnisse Testcase 2

Testcase 3 ist inzwischen erfolgreich absolviert worden. An der Datenqualität wurde gearbeitet und der Grenzwert wird inzwischen immer erreicht. Es gibt weiterhin falsche Datensätze und in der weiteren Analyse ist auch die Frage zu beantworten, ob der Grenzwert zu einfach definiert wurde.

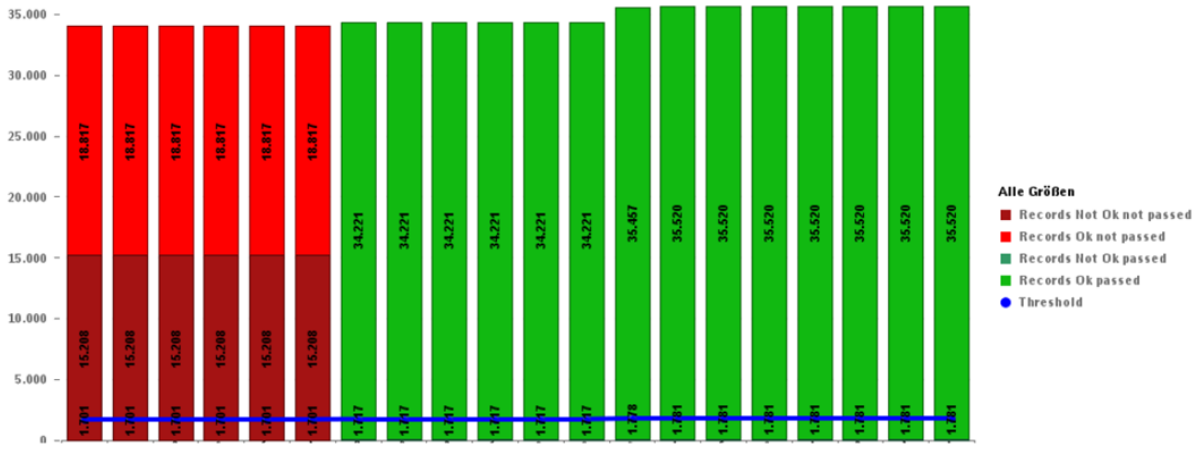

Abb. 7: Ergebnisse Testcase 3

Testcase 4 ist ein Paradebeispiel für automatisches Testen. Zur Absicherung der erreichten Qualität der Daten muss bei jedem ETL Lauf getestet werden, ob Änderungen an Daten, Masterdaten oder den ETL Prozessen nicht wieder einen bereits behobenen Fehler zurückbringt.

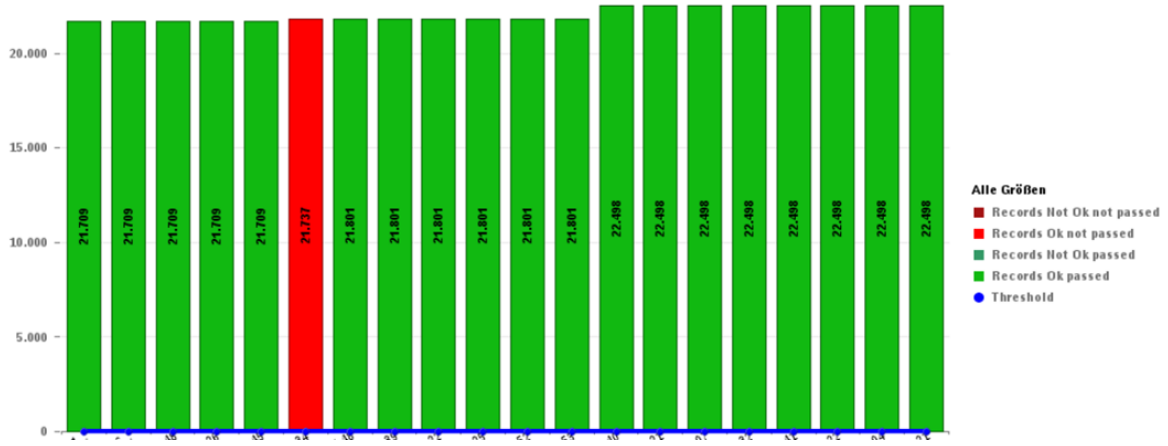

Abb. 8: Ergebnisse Testcase 4

Testcase 5 zeigt, dass auch ein immer erfolgreich bestandener Testcase weiter verbessert werden kann. Hier wurde die Anzahl der falschen Datensätze noch einmal deutlich nach unten korrigiert. Allerdings ist auch die Frage berechtigt, ob der Aufwand dafür berechtigt ist

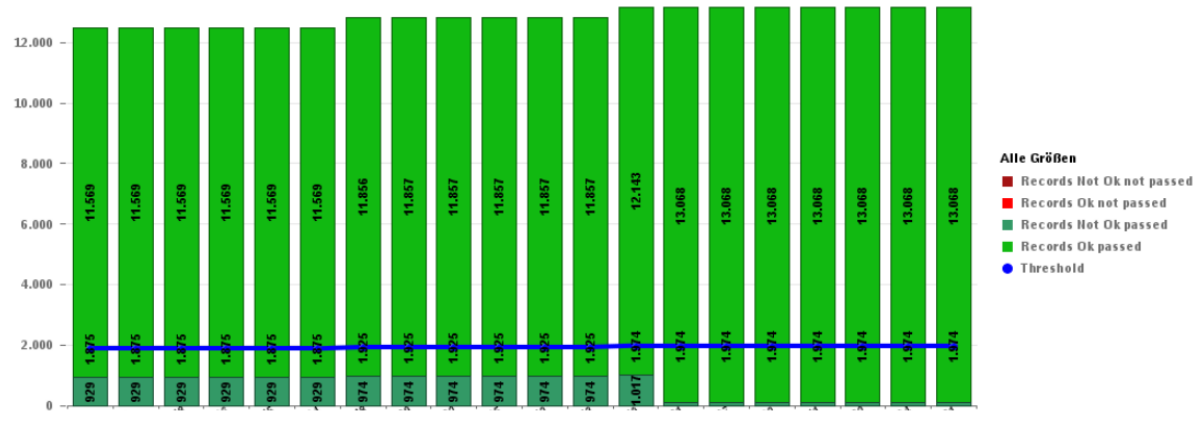

Abb. 9: Ergebnisse Testcase 5

Fazit

Mit dem Testframework für automatische Testen haben wir ein einfaches Framework etabliert. Das Testframework ist offen und flexibel für Erweiterungen. Die gewählte Technologie PL/SQL erlaubt es jedem ETL Entwickler mit dem Framework zu arbeiten und Erweiterungen daran vorzunehmen.

Mit der Technologie PL/SQL sind wir auch nah an der CDW Datenbank, dies erlaubt eine performante Abarbeitung der Testfälle.

Bezüglich der Visualisierung der Ergebnisse haben wir als Startpunkt einen einfachem ASCII Report für den Product Owner erstellt und diesem zur Verfügung gestellt. Mit der Einführung einer Reporting Plattform auf Basis der CDW, haben wir auch das QS Reporting auf diese Plattform umgestellt. Damit haben wir deutlich mehr Transparenz für die Datenqualität erreicht. Es wird auch sichtbar, dass sich die Datenqualität nicht von alleine verbessert, sondern das es dazu gelebte Prozesse braucht.

Diese Prozesse zur Verbesserung der Datenqualität leben im Wesentlichen von der Arbeit der Domain Experten. Diese haben wir für erste Datenbereiche gefunden und sind dabei deren Rolle zu etablieren.

Datenqualität ist kein Selbstläufer sondern braucht kontinuierliche Arbeit. Deswegen führt ein automatisches Testframework nicht automatisch zu besseren Daten, aber es hilft sehr auf dem Weg zu dahin.

Kontaktadresse:
Edgar Kaemper
Robert Bosch GmbH
AA-AS/EIS3-EU
Franz-Oechsle-Straße 4
D-73207 Plochingen

E-Mail: edgar.kaemper@de.bosch.com
Internet: http://bosch-automotive-aftermarket.com/de/home/