profi.com®
we make IT work

WHITE PAPER

METRIKEN ZUR MESSUNG DER DATENQUALITÄT

von Dr. Carsten Neise

Daten sind die Währung unserer Zeit! Was bei Datenschützern großes Unbehagen erzeugt ist für Softwareentwickler von essentieller Bedeutung. Dank offener Schnittstellen lassen sich Daten aus unzähligen verschiedenen Quellen miteinander kombinieren und in einen neuen Zusammenhang stellen. Ob die komplexe Flugbuchung inklusive Mietwagenreservierung oder die Berechnung von komplexen Versicherungsprodukten – überall werden (persönliche) Daten genutzt.

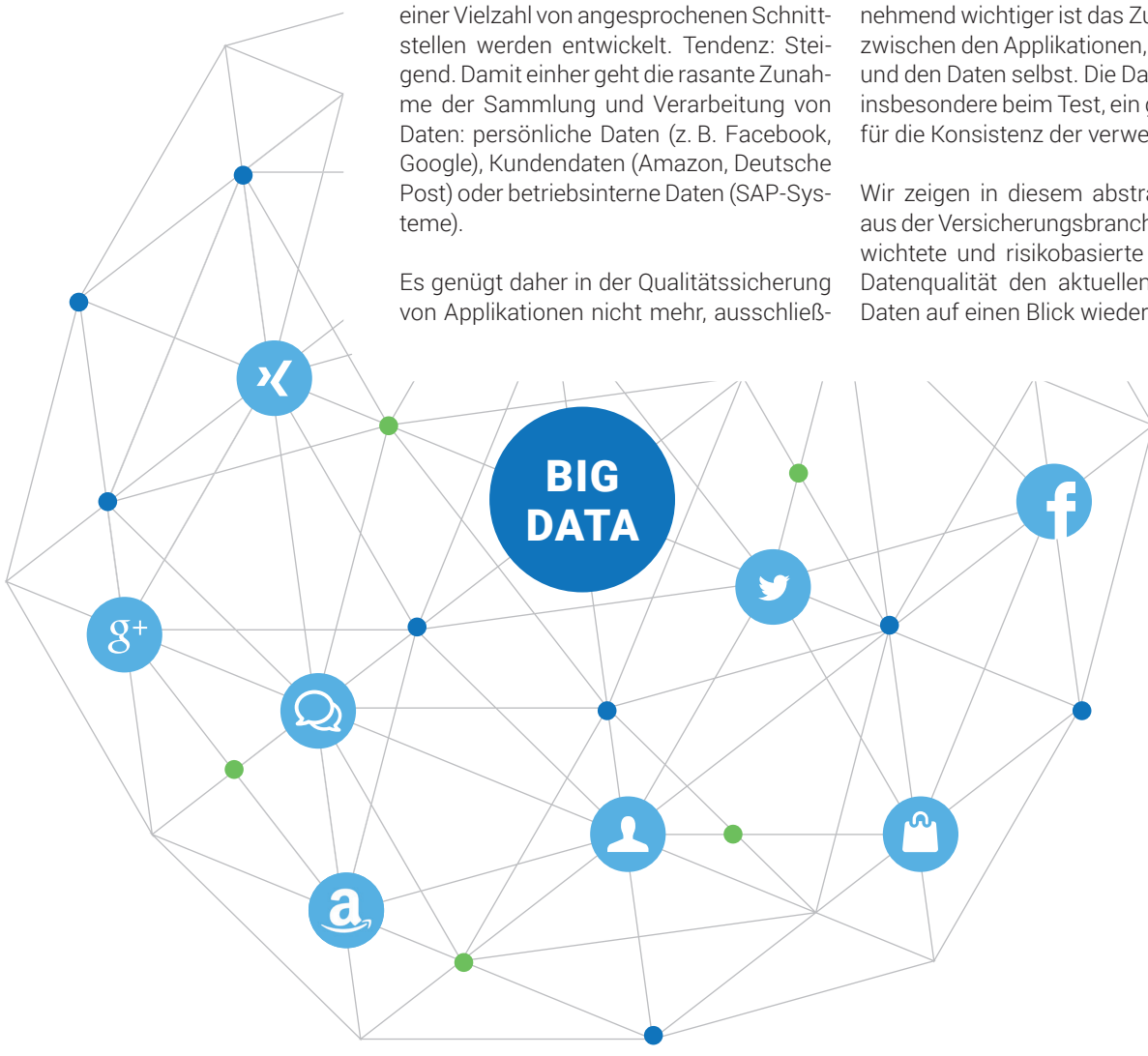
Das vorliegende White Paper beschreibt einen wissenschaftlichen Ansatz, um die Qualität der (Test-)Daten einer Softwareentwicklungsumgebung zu »messen«. Wie lässt sich schnell und mit effektiven Mitteln die Güte des aktuellen Datenbestandes ermitteln? Anhand gewichteter und risikobasierter Faktoren können Aussagen zur Güte der im System befindlichen Daten getoffen werden. Dies hilft, die Softwarequalität zu erhöhen, indem Fehleinschätzungen vermieden werden.

Einleitung

»Big Data« ist einer der wichtigsten IT-Trends unserer Zeit. Immer mehr Apps mit einer Vielzahl von angesprochenen Schnittstellen werden entwickelt. Tendenz: Steigend. Damit einher geht die rasante Zunahme der Sammlung und Verarbeitung von Daten: persönliche Daten (z. B. Facebook, Google), Kundendaten (Amazon, Deutsche Post) oder betriebsinterne Daten (SAP-Systeme).

Es genügt daher in der Qualitätssicherung von Applikationen nicht mehr, ausschließlich die reine Funktionalität der Anwendung und der Schnittstellen zu überprüfen. Zunehmend wichtiger ist das Zusammenspiel zwischen den Applikationen, Schnittstellen und den Daten selbst. Die Datenqualität ist, insbesondere beim Test, ein guter Indikator für die Konsistenz der verwendeten Daten.

Wir zeigen in diesem abstrakten Beispiel aus der Versicherungsbranche, wie eine gewichtete und risikobasierte Messung der Datenqualität den aktuellen Zustand der Daten auf einen Blick wiedergibt.



Die Herausforderung

Zur Gewährleistung der vollen Funktionalität einer Software ist es, insbesondere auf der Testumgebung, unerläßlich, die im System befindlichen Daten auf Konsistenz, Relevanz und schematische beziehungsweise syntaktische Korrektheit zu »messen«. Die Frage, die sich in diesem Zusammenhang stellt, ist: Wie kann ich schnell und mit einfachen Mitteln die Güte des aktuellen Datenbestandes prüfen? Im Folgenden wird eine Lösung vorgestellt, welche in der Praxis bei einem Kunden der Versicherungsbranche erfolgreich umgesetzt wurde.

Es soll bei dieser Auswertung weder ein Report mit einer unzählig langen Aufzählung der fehlerbehafteten Daten erstellt werden, noch soll das Resultat eine unzählig lange Auflistung der Fehler selbst sein. Vielmehr wird ein geeignetes Mittel zur Verfügung gestellt, mit dem schnell und einfach der Trend über verschiedene Messungen (z. B. Wochen-/Monatsübersicht) abgeleitet wird.

Der fortlaufende Test eines Systems oder dessen Abnahme ohne die Betrachtung der Datenqualität kann zu groben Fehleinschätzungen führen. So lässt sich anhand der summarischen Fehlerauflistung zwar eine Aussage über die Qualität der Applikation im Allgemeinen ableiten, jedoch werden diese Fehler nicht gewichtet. Ist ein Rechtschreibfehler zum Beispiel unkritischer anzusehen als die falsche Berechnung eines Versicherungsbeitrages?

Der Lösungsansatz

Der einfachste und schnellste Weg zur Messung der Datenqualität ist die Projektion der fehlerbehafteten Daten auf genau eine Zahl. Eine solche Zahl wird als Metrik – im Folgenden als Datengüte – bezeichnet. Die Datengüte ist definiert als abstraktes Maß zur Messung eines Abstands. Der Abstand in unserem Raum wird als Unterschied zwischen fehlerfreien Daten (Soll-Zustand) und den im System aktuell befindlichen Daten (Ist-Zustand) definiert.

Die Datengüte wird so ermittelt, dass diese einen Wert der Menge im Intervall [0, 1] annimmt. Im aktuellen Modell bedeuted eine Datengüte mit dem Wert 0 eine fehlerfreie Datenbank, ein Wert von 1 bedeuted eine zu 100 Prozent fehlerbehaftete Datenbank.

Es sind zwei unterschiedliche **Metriken η** und **ε** (Nomenklatur am Ende) definiert worden. Allgemein gesprochen spiegelt die Metrik die Güte der Daten in der Datenbank wider, als auch die Güte der Qualität der Datenbank an sich (Datenbankschema).

Die **relative Datengüte η** gibt an, wie gut die Datenqualität im Verhältnis zu allen auftretenden Datenfehlern ist. Währenddessen die **absolute Datengüte ε** verdeutlicht, wie gut die Datenqualität im Verhältnis zu allen Datenbankeinträgen ist.

Der Anteil, welcher auf das Datenbankschema referenziert, ist als Summe auf fehlende und fehlerhafte »Constraints« im Datenbankschema abbildbar. Diese Informationen werden zurzeit aus Datenbankkenntnissen abgeleitet. Zukünftig soll diese Information aus UML-Diagrammen gewonnen werden. Erläuterungen dazu erhalten Sie weiter unten im Abschnitt »Benefit«.

Die Datengüte wird nun als gewichtete Summe über die fehlerbehafteten Daten und Fehler im Datenbankschema geschrieben. Um **η** und **ε** im Intervall [0, 1] abzubilden, müssen die Nebenbedingungen 1 bis 3 erfüllt sein.

$$\eta = \eta_{Daten} + \eta_{Datenbankschema} \quad (1)$$

$$\varepsilon = \varepsilon_{Daten} + \varepsilon_{Datenbankschema} \quad (2)$$

$$\eta = \sum_i \eta_i = \frac{\sum_{i,n} g_{i,n} \Delta_{i,n}}{\sum_{i,n} \Delta_{i,n}} = \left(\frac{\sum_{i,n} g_{i,n} \Delta_{i,n}}{\sum_{i,n} \Delta_{i,n}}\right)_{Daten} + \left(\frac{\sum_{i,n} g_{i,n} \Delta_{i,n}}{\sum_{i,n} \Delta_{i,n}}\right)_{Datenbankschema} \quad (3)$$

$$\varepsilon = \sum_i \varepsilon_i = \frac{\sum_{i,n} g'_{i,n} \Delta_{i,n}}{\sum_{i,n} f_{i,n}} = \left(\frac{\sum_{i,n} g'_{i,n} \Delta_{i,n}}{\sum_{i,n} f_{i,n}}\right)_{Daten} + \left(\frac{\sum_{i,n} g'_{i,n} \Delta_{i,n}}{\sum_{i,n} f_{i,n}}\right)_{Datenbankschema} \quad (4)$$

$$g_{i,n} = h_i * t_n \quad (5)$$

$$\text{Nebenbedingung 1: } \sum_n t_n \leq 1 \quad (6)$$

$$\text{Nebenbedingung 2: } \sum_{i,n} h_i * t_n \leq 1 \quad (7)$$

$$\text{Nebenbedingung 3: } \eta, \varepsilon \in [0,1] \quad (8)$$

Der **Gewichtungsfaktor** $g_{i,n}$ wird als Produkt der zwei Faktoren h_i und t_n geschrieben. h_i steht für die Fehlerklasse der Fehleräquivalenzklasse und t_n steht für die Fehlerklasse der Tabelle. Zu beachten ist, dass aufgrund der Nebenbedingung 3 die **Gewichtungsfaktoren** $g_{i,n}$ und $g'_{i,n}$ nicht gleich gewählt werden dürfen. Diese werden genutzt, um zum einen η und zum anderen ε zu berechnen. Im Allgemeinen bedeutet dies, dass sich eine Renormierung zwischen den beiden Faktoren ergibt. Der Renormierungsfaktor kann aus den Gleichungen 3 bis 5 und den Nebenbedingungen 1 bis 3 ermittelt werden. Schlußendlich gewährleisten Nebenbedingung 1 und 2 die Realisierung der Nebenbedingung 3.

Die Ermittlung der **Gewichtungsfaktoren** h_i und t_n ist der zentrale Punkt in diesem Lösungsansatz. Eine richtige Bewertung der Risiken ist essentiell für die Bewertung der Datenqualität (s. Abbildung 1). Der Aufwand, welcher zum Gewinn der Faktoren benötigt wird, kann unterschiedlich hoch ausfallen. Es können Modelle für die Risikobewertung zur Bestimmung der Gewichtungsfaktoren genutzt werden oder es kann eine empirische »Best-Practice«-Ermittlung (wie in unserem Beispiel) stattfinden. Wofür man sich auch entscheidet, das Resultat der Risikobewertung muss eine Quantifizierung der einzelnen **Gewichtungsfaktoren** h_i und t_n sein. Damit wird basierend auf Formel 3 oder 4 die Datenqualität bestimmt.

Im unserem Beispiel nutzen wir die absolute **Datengüte** ε. Im Folgenden sind noch keine Prüfungen auf das Datenbankschema selbst durchgeführt, die Fehlerermittlung beläuft sich somit nur auf den ersten Teil der obigen Summe, siehe Gleichung 4. Im derzeitigen Modell werden die t_n gleichverteilt für jede Prüfung des Datenbankeintrags in verschiedenen Tabellen berechnet. Die Faktoren h_i entstammen einer Risikoabschätzung des Kunden, die von dessen Business-Analysten stammt. Hierbei werden Fehler in Äquivalenzklassen zusammengefasst. Anhand der Einschätzung der Business-Analysten erhielten grobe fachliche Verletzungen der Daten eine hohe Gewichtung (z. B. fehlende oder fehlerhafte Einträge in Tabellen für Verträge), und kleine fachliche Verletzungen (z. B. fehlende oder fehlerhafte Einträge in Kontaktdaten) eine geringe Gewichtung.

Nomenklatur	
n	Tabellenindex
i	Fehlerklassenindex (Äquivalenzklasse)
$g_{i,n}$	Gewichtungsfaktor, Gesamt (normiert)
h_i	Gewichtungsfaktor, Fehlerklasse der Fehleräquivalenzklasse
t_n	Gewichtungsfaktor, Fehlerklasse der Tabelle
$\Delta_{i,n}$	Abweichung/Fehler in der Tabelle
$f_{i,n}$	Summe aller Datenbankeinträge in der Tabelle
η	relative Datengüte
ε	absolute Datengüte

Beispiel

In unserem Beispiel rechnen wir mit 150 Tabellen in der Datenbank. Wir kalkulieren mit einer durchschnittlichen Anzahl von 10.000 Zeilen und 10 Spalten je Tabelle. Dies ergibt eine typische Größenordnung von 15 Millionen Einträgen in der Datenbank.

Typischerweise sind diese Dimensionen für Versicherungen, Telefongesellschaften, Versorgungsunternehmen, Onlinehändler oder Wirtschaftsprüfungsgesellschaften und andere Großunternehmen ein realistisches Szenario.

Nehmen wir an, eine Tabelle, welche alle Vertragsdaten bezüglich der von Kunden abgeschlosssen Versicherungen beinhaltet, wird sechs Mal überprüft. Dahingehend wird der Faktor t_n auf 1/6 bestimmt. Wird diese Tabelle nur einmal überprüft, bekommt diesen den Faktor $t_n = 1$. Die Bestimmung der h_i, also die Gewichtung der Fehlerklassen, erfolgt nun durch den Business-Analysten. So kann ein Rechschreibfehler den Faktor $h_i = 1$ und ein falscher Parameter für die Berechnung der Schadenssumme den Faktor $h_i = 10.000$ bekommen.

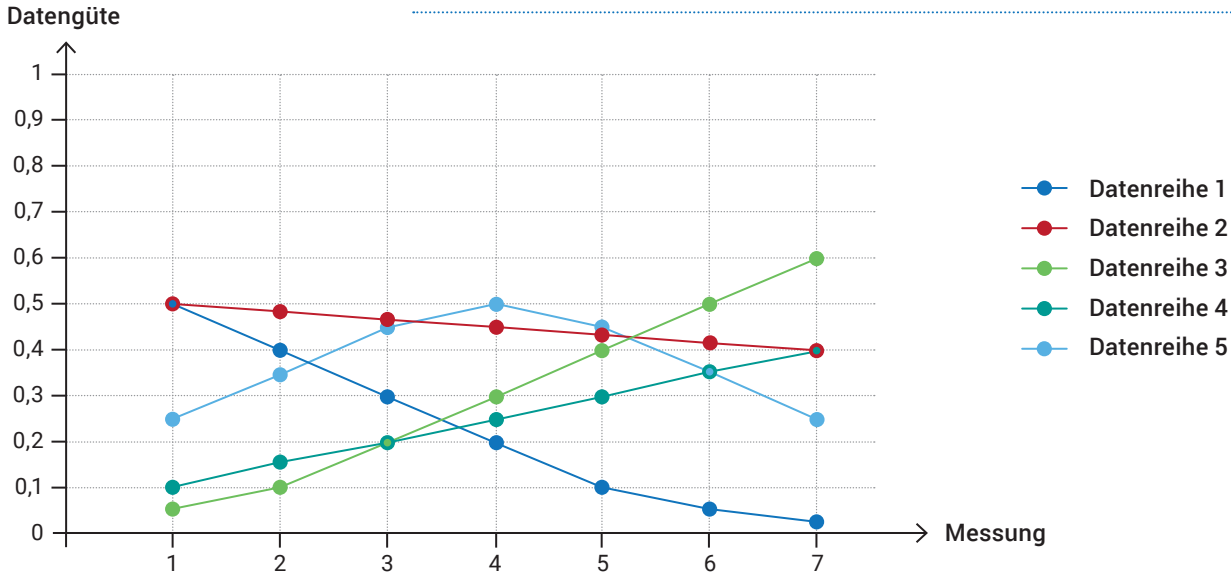


Abbildung 1:
Exemplarische Grafik zur Veraunschaulichung der Datengüte ε

Das Ergebnis dieses Vorgehens sind die bestimmten Werte t_n und h_i und die errechneten Gewichtungsfaktoren $g_{i,n}$ für die Ermittlung der Datengüte, als auch die Anzahl der gefundenen Fehler $\Delta_{i,n}$. Diese werden mit den Gewichtungsfaktoren versehen und somit die Datengüte ε bestimmt.

Als Prüfung, ob die Normierung der Wichtungsfaktoren korrekt ist, setzt man die Anzahl der gefunden Fehler pro Tabelle auf eins. Zusätzlich wird die Summe der Datensätze auf eins gesetzt. In diesem konstruierten Beispiel muss ε=1 ergeben.

Die dadurch errechnete Datengüte kann somit als Kontrollzahl jederzeit gemessen werden, um eine schnelle Aussage über die Qualität des derzeitigen Datenbestands zu erhalten. Für die alltägliche Praxis ist relevant, dass die Berechnung jederzeit ausgeführt werden kann, z. B. nach einem Datenimport aus einem Fremdsystem.

Beispielhaft zeigen wir in Abbildung 1 Berechnungen zu sieben verschiedenen Zeitpunkten, in denen fünf Datenreihen abgebildet werden.

In jeder Datenreihe sind die Gewichtungsfakoren h_i anders gewählt. So erhält man für die Datenreihe 1 und 2 eine Verbesserung der Datengüte nach jeder Messung. Anders sieht es für die Datenreihen 3 und 4 aus, hier wird eine Verschlechterung der Datengüte suggeriert. Eine Superposition beider Tendenzen vermittelt die Datenreihe 5.

Damit sei angemerkt, dass die Aussage eines Trends im Datenbestand ganz klar von der Bestimmung der einzelnen Gewichtungsfaktoren $g_{i,n}$ abhängt. Diese sollen daher entsprechend der eigenen Wichtigkeit gewählt werden, um in der Lage zu sein, die für sich richtigen Schlußfolgerungen zu ziehen.

Der Benefit

Nach einer kurzen Implementierungsphase wurde das Modell bei einem Referenzkunden aus der Versicherungsbranche praktisch umgesetzt. Die Bewertung der Datenqualität des aktuellen Releases konnte von ursprünglich zwei Stunden auf nun 15 Minuten reduziert werden. Gleichzeitig lässt sich damit auch die Qualität der einzelnen Schnittstellen beurteilen. Neben dem Vorteil der zeitlichen Effektivierung sollte beachtet werden, dass der initiale Aufwand zur Generierung der SQL-Skripte sehr hoch ist. Sind diese Skripte jedoch einmal implementiert, lässt sich die Güte der im System befindlichen Daten schnell und einfach generieren.

DAS PRO UND CONTRA DIESES VORGEHENS

Pro	Contra
Die Datengüte gibt sehr schnell und auf einen Blick den aktuellen Zustand der Datenbank wieder.	Die Umsetzung in unserem Beispiel ist aktuell nur für ein Datenbankschema in Form von SQL-Skripten realisiert. Damit ist es sehr unflexibel.
Tendenzen der Datengüte werden ebenso schnell sichtbar.	Der Aufwand, die Skripte zu generieren und für verschiedene Datenbanken (u. a. andere Projekte) anzupassen, ist sehr hoch.
Sehr sinnvoll zum Test von Schnittstellen zwischen verschiedenen Systemen.	Logischen Korrelationen verschiedener Tabellen in der Datenbank, als auch die Constraints selbst, mussten ohne Modell selbst entwickelt und in die SQL-Skripte implementiert werden.
	Die Gewichtung der einzelnen Fehlerklassen (Tabelle, Fehleräquivalenz) war nur beispielhaft implementiert.

Nach einem Initialaufwand von drei Tagen zur Erstellung der SQL-Prüfskripte konnte die Bestimmung der aktuellen Datenlage um den Faktor 8 reduziert werden. Damit sich dieser Prozess weiterhin etabliert, sehen wir folgende Punkte als unabdingbar an

- Implementierung der Prüfung des Datenbankschemas in die SQL-Skripte zur Berechnung der Datengüte
- Automatisierte Ausführung der SQL-Prüfskripte und Berechnung der Datengüte
- Zukünftig soll das Datenbank-Schema modellbasiert (UML/BPMN-Notation) erstellt werden, um daraus direkt automatisiert die Kontrollskripte zu generieren
- Implementierung und Nutzung eines generischen Risikomodells, mit dem die einzelnen Gewichtungsfaktoren ermittelt werden können