Data Mining im praktischen Einsatz: Prüfung von Softwareentwicklungsprozessdaten mittels Assoziationsverfahren bei der Continental Automotive GmbH

Verena Solleder, M. Sc.
Professor Dr.-Ing. Frank Herrmann
Ostbayerische Technische Hochschule Regensburg
Innovationszentrum für Produktionslogistik und Fabrikplanung (IPF)
E-Mail: vsolleder@aol.com
E-Mail: Frank.Herrmann@OTH-Regensburg.de

SCHLÜSSELWÖRTER

Data Mining, KNIME, Assoziationsverfahren, Apriori Algorithmus, Regel

ABSTRACT

Die Continental Automotive GmbH sammelt schon über einen längeren Zeitraum große Datenmengen aus laufenden IT-Kundenprojekten. Dabei handelt es sich um Engineering Daten, welche aus unterschiedlichen Anforderungen, Testfällen, Problem Reports und Change Requests bestehen. Diese liegen strukturiert und konsistent in einem Data Warehouse vor. Um einen Mehrwert aus den Daten ziehen zu können, werden diese für den Einsatz von Data Mining genutzt. Dabei wird ganz klar das Ziel verfolgt, durch Anwendung eines geeigneten Data Mining Verfahrens die aus dem Data Warehouse zur Verfügung gestellten Daten zu untersuchen und somit den aktuell bestehenden Softwareentwicklungsprozess zu prüfen. Zur Durchführung wird die freie Software KNIME verwendet. Diese dient nach sorgfältiger Datenanalyse zur Aufbereitung der Datengrundlage, Modellierung und Anwendung des ausgewählten Data Mining Verfahrens, des Assoziationsverfahrens. Die daraus resultierenden Ergebnisse werden im letzten Schritt quantifiziert und liefern Aufschlüsse über den Softwareentwicklungsprozess mit zugehöriger Datengrundlage.

Die hier vorliegende Arbeit wurde bei der Continental Automotive GmbH Regensburg durchgeführt und stammt aus dem Geschäftsfeld Body & Security aus der Division Interior.

DATA MINING

BEGRIFF

Gemäß der GARTNER GROUP wird der Begriff Data Mining folgendermaßen definiert:
„The process of discovering meaningful correlations, patterns and trends by sifting through large amounts of data stored in repositories. Data mining employs pattern recog-nition technologies, as well as statistical and mathematical techniques" [GART17].

Ein weit in der Literatur verbreitetes Synonym für den Ausdruck Data Mining ist der Begriff Knowledge Discovery in Databases (KDD).

PROZESS

Nach Fayyad, Piatetsky-Shapiro und Smyth ist der KDD Prozess folgendermaßen aufgebaut (vgl. Abb.1).

1. Selektion: Das Prozessziel wird festgelegt und dazugehörig relevante Daten ausgesucht.
2. Vorverarbeitung: Dieser Schritt befasst sich mit der Datenreinigung und –vorbereitung. Es geht darum, Daten in guter Qualität bereitzustellen. Darunter zählt beispielsweise das Erkennen von Duplikaten, die Behandlung von fehlenden Werten, die Identifikation von Ausreißern oder die Korrektur fehlerhafter Werte.
3. Transformation: Die Daten werden für die bevorstehenden Verfahren in den passenden Datentyp transformiert.

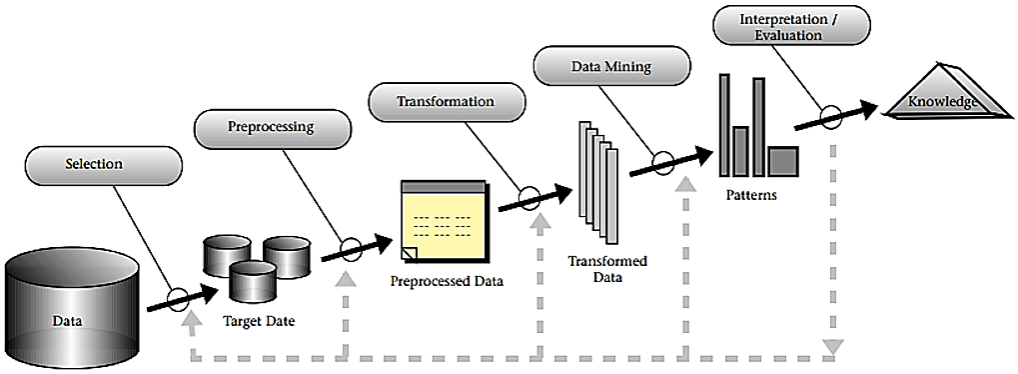

Abb. 1: Data Mining Prozess [FAPISM96]

4. Data Mining: Der für das festgelegte Ziel geeignete Data Mining Algorithmus wird ausgewählt. Dieser muss entsprechende Methoden zum Suchen der gewünschten Datenmuster enthalten.
5. Interpretation und Evaluation: Der Algorithmus und dessen Ergebnisse werden bewertet und interpretiert. Gefundene Muster werden für die Entscheidungsfindung genutzt und je nach Nutzen visuell aufbereitet.

AUFGABEN

Basierend auf den verschiedenen Zielsetzungen des Data Mining kann man zwischen diversen Data Mining Aufgaben differenzieren: [MÜLE13]

- Segmentierung: Bildung von in sich homogenen Clustern, die sich möglichst unähnlich zu anderen Gruppen unterscheiden.
- Klassifikation: Es soll auf Basis von bekannten Klassenzuordnungen mit Hilfe von Trainingsdaten ein Klassifizierungsmodell erlernt werden. Dieses Modell wird dann auf Daten (Testdaten) unbekannter Klassenzuordnung angewandt.
- Abweichungsanalyse: Auffinden von Datensätzen, welche untypisch im Vergleich zur Gesamtdatenbasis sind.
- Prognose: Voraussagung eines numerischen Wertes auf Grundlage von Vergangenheitswerten.
- Assoziationsanalyse: Hier werden regelbasierte Abhängigkeiten in der gegebenen Datenmenge gesucht.
- Sequenzanalyse: Ähnlich zur Assoziationsanalyse, jedoch werden hier typische Sequenzmuster gesucht.

ASSOZIATIONSVERFAHREN

Das in dieser Arbeit verwendete Data Mining Verfahren ist die Assoziationsanalyse. Dieses ist eine leistungsstarke Methode für die so genannte Marktkorb-Analyse, die darauf abzielt, Regelmäßigkeiten im Einkaufsverhalten von Kunden von beispielsweise Supermärkten, Versandhäusern oder Online-Shops zu finden.
Eine Assoziationsregel ist zum Beispiel: "Wenn ein Kunde Wein und Brot kauft, kauft er oft auch Käse." Es handelt sich um eine Vereinigung zwischen Sets von Gegenständen, bei denen es sich beispielsweise um Produkte eines Supermarktes oder optionale Dienstleistungen von Telekommunikationsunternehmen handelt.
Eine Assoziationsregel besagt, dass bei einem zufällig ausgewählten Kunden, welcher bestimmte Gegenstände auswählt, zuversichtlich durch einen Prozentsatz quantifiziert werden kann, dass dieser auch bestimmte andere Gegenstände gewählt hat. [BORG12]
Im Folgenden werden einige grundlegende Begriffe vorgestellt, die erforderlich sind, um das Assoziationsverfahren zu verstehen.
Diese werden anhand von Beschreibungen von Christian Borgelt, dessen Assoziationsverfahren in KNIME verwendet wird, erklärt. [BORG12]

Items und Transactions

Der Input für eine Assoziationsregel besteht abstrakt aus einem Multiset von Transaktionen, die über einen Satz von Elementen definiert sind, manchmal auch als Item Base bezeichnet. Jedes Item benötigt zur Unterscheidung eine eindeutige Identifikationsnummer. Die Item Base ist der Satz aller betrachteten Gegenstände. Jede Teilmenge der Item Base wird als Item Set bezeichnet.
Eine Transaction ist einfach ein Item Set und stellt zum Beispiel die von einem Kunden gekauften Produkte dar.

Support eines Item Sets

Sei S ein Item Set und T das Multiset aller Transaktionen, dann ist der absolute Support (oder einfach Support) des Item Sets S die Anzahl der Transaktionen in T, die S enthalten. Ebenso ist der relative Support von S der Bruchteil (oder Prozentsatz) der Transaktionen in T, die S enthalten.
In formaler Weise sei S ein Item Set und $U = \{X \in T \mid S \subseteq t\}$ das Multiset aller Transaktionen in T, die S als Teilmenge haben (d.h. alle Elemente in S und eventuell einige andere enthalten), dann ist der absolute Support

$$\text{Supp}_{abs}(S) = |U| = |\{X \in T \mid S \subseteq t\}| \text{ und}$$
$$\text{Supp}_{rel}(S) = (|U| / |T|) * 100\%$$

der relative Support von S. Hier sind $|U|$ und $|T|$ die Anzahl der Elemente in U und T.

Confidence einer Assoziationsregel

Zur Messung der Qualität von Assoziationsregeln wird die Confidence verwendet. Die Confidence einer Assoziationsregel R = "X → Y" (mit Item Sets X und Y) ist der Support der Menge aller Items, die in der Regel erscheinen (hier: Support von S = X ∪ Y) geteilt durch den Support des Antecedents (auch "if-part" oder "body" genannt) der Regel (hier X).

$$\text{Conf}(R) = \frac{\text{Supp}(X \cup Y)}{\text{Supp}(X)}$$

Die Confidence einer Regel ist die Anzahl der Fälle, in denen die Regel korrekt ist, bezogen auf die Anzahl der Fälle, in denen sie anwendbar ist. Sie gibt also den Prozentsatz der Fälle an, in denen die Regel korrekt ist.

Lift

Der sogenannte Liftwert ist der Quotient aus dem hinteren und dem vorherigen Support einer Assoziationsregel. In formaler Weise ist der Lift der Regel R = X → Y

$$\text{Lift}(R) = \frac{\text{conf}(X \to Y)}{\text{conf}(\emptyset \to Y)} = \frac{\text{supp}(X \cup Y)/\text{supp}(X)}{\text{supp}(Y)/\text{supp}(\emptyset)}$$

wo $\text{supp}(\emptyset) = |T|$, die Größe der Transaktionsdatenbank (Anzahl der Transaktionen).

Ein Lift mit einem Wert kleiner 1 liefert folglich also keine zusätzlichen Erkenntnisse, wohingegen ein Lift mit einem Wert größer 1 auf eine positive Korrelation hindeutet. [BAVO08]

APRIORI ALGORITHMUS

Der Apriori Algorithmus verwendet eine iterative Methode, um alle häufigen Item Sets zu finden und dann aus ihnen Regeln zu erzeugen. Der Algorithmus reduziert die Anzahl der zu untersuchenden Items, indem er nur die Item Sets behandelt, deren Support größer als der Minimum Support ist. Alle selten auftretenden Items werden ignoriert und nicht zur Generierung einer Regel verwendet. [DAS16]
Der Ablauf des Apriori Algorithmus gliedert sich in mehrere Schritte: [DAS16]

Bevor mit dem ersten Schritt gestartet werden kann, müssen Parameter für Support, Confidence und Minimum Set Size festgelegt werden.

Schritt 1: Für jedes Item wird der Support berechnet.
Schritt 2: Alle Items mit einem geringeren Support als dem Minimum Support werden ausgeschlossen. Die übrigen Items werden für die weitere Analyse beibehalten.
Schritt 3: Aus den Items werden Item Sets generiert.
Schritt 4: Für alle Item Sets wird der Support berechnet.
Schritt 5: Es werden die häufigsten Item Sets durch Vergleich ihres Supports mit dem Minimum Support gefunden. Dabei muss der Support höher als der Minimum Support sein.
Schritt 6: Es werden für alle häufigen Item Sets die Confidence der Regel berechnet.
Schritt 7: Es werden diejenigen Regeln, welche eine höhere Confidence als die Minimum Confidence besitzen, beibehalten. Diese bilden den finalen Regelsatz.
Die generierten Regeln sehen dabei beispielsweise so aus: A → B
Der linke Teil der Regel „A“ wird als Antecedent bezeichnet. Der rechte Regelteil „B“ heißt Consequent. Zur Bewertung einer Regel werden die Bewertungskriterien Support, Confidence und Lift verwendet. Somit lassen sich gute und schlechte Regeln identifizieren.

BETRIEBLICHE AUSGANGSSITUATION

V-MODELL

Das zur Durchführung der gegebenen IT-Kundenprojekte verwendete Vorgehensmodell ist das V-Modell, welches hier kurz beschrieben wird (vgl. Abb.2).

Das V-Modell berücksichtigt sowohl die Schritte der Software-Entwicklung als auch die Phasen der Qualitätssicherung. Die Entwicklung beginnt auf der linken Seite mit der Anforderungsdefinition, welche die Anforderungen an das zu erstellende Softwareprodukt festlegt. Daraufhin folgt der funktionale Systementwurf, der die Schritte des Grobdesigns enthält. Der technische Systementwurf bestimmt, auf welchen Systemen die Software laufen soll. Zusätzlich kann hierbei über eine notwendige Entwicklung neuer Hardware nachgedacht werden. Die letzte Phase ist die Komponentenspezifikation, welche das Feindesign beinhaltet. Qualitätssichernde Prüfmaßnahmen, welche in der Abbildung als gestrichelte Pfeile dargestellt werden, prüfen, ob die Ergebnisse zur vorherigen Phase passen und ob Verbesserungen notwendig sind. Die rechte qualitätssichernde Seite des V-Modells unterscheidet sich in verschiedenen Testarten. Der Komponententest überprüft einzelne Software-Bausteine und soll beweisen, dass die Ergebnisse mit der Komponentenspezifikation übereinstimmen. Wenn die einzelnen Komponenten zusammengebaut sind, ist das System sozusagen integriert. Hierbei wird beim Integrationstest die Lauffähigkeit des Systems mittels des technischen Systementwurfs geprüft. Beim Systemtest wird das System als Ganzes überprüft, d.h. ob es die vom Kunden gewünschte Funktionalität bietet. Zuletzt folgt der Abnahmetest, der zusammen mit dem Kunden das System und die geforderte Funktionalität testet. [KLEU13]

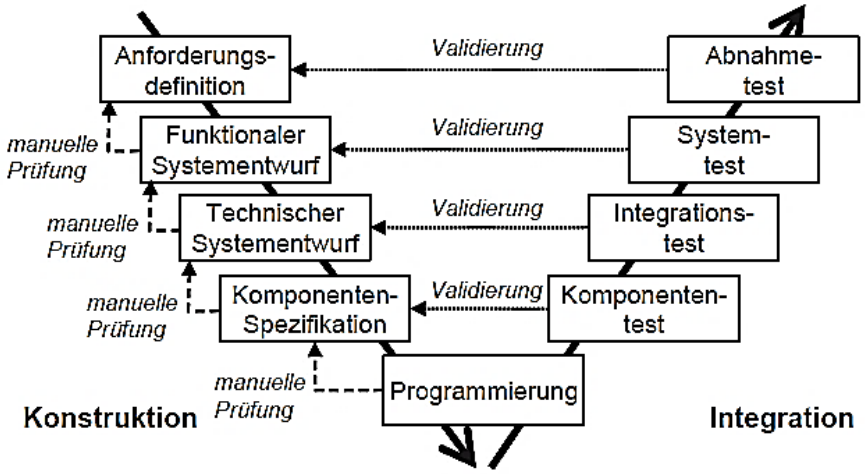

Abb. 2: V-Modell [KLEU13]

DATENGRUNDLAGE

Die für unsere Untersuchung relevanten Daten sind diverse Anforderungen, Architektur Elemente und Testfälle in Verbindung mit gewünschten Projektänderungen.
Diese werden hier kurz vorgestellt:

- Stakeholder Requirement (STR): Eine vom Auftraggeber geforderte Anforderung.
- System Requirement (SYR): Anforderung an das Zielsystem.
- System Architecture Element (SYA): Gibt Vorgaben über die Architektur eines Systems.
- Software Requirement (SWR): Anforderung an die Software.
- Software Architecture Element (SWA): Gibt Vorgaben über Softwarekomponenten.
- Test Case (TCS): Testet ein System oder dessen Software auf die anfangs spezifizierten Anforderungen.
- Problem Report (PBR): Dokumentierter Wunsch, ein Projektproblem zu lösen.
- Change Request (CHR): Dokumentierter Wunsch, eine Projektänderung durchzuführen.

Dabei können Anforderungen untereinander oder Anforderungen mit einem oder mehreren Testfällen in Beziehung stehen. Dafür gib es die zwei Relationsarten vertikale und horzontale Traceability (vgl. Abb. 4).
Die vertikale Traceability wird mit „satisfies“ ausgedrückt. Diese besteht, wenn sich eine Anforderung auf eine andere Anforderung bezieht. So kann sich beispielsweise ein Software Requirement auf ein System Requirement beziehen. Die Relation wird als vertikal bezeichnet, da diese nur einseitig auf der linken Seite des V-Modells vorkommt und sich Anforderungen aus unteren Phasen auf Anforderungen höherer Phasen beziehen, wodurch eine vertikale Richtung der Beziehung entsteht. Die zweite Relationsart ist die horizontale Traceability und wird mit „tests“ ausgedrückt. Diese besteht, wenn sich ein Test Case auf eine oder mehrere Anforderung/ -en bezieht, diese also testet. Man spricht hier von einer horizontalen Beziehung, da sich diese von einem Test Case auf der rechten Seite des V-Modells auf eine oder mehrere Anforderung/-en auf der linken Seite des V-Modells erstreckt.

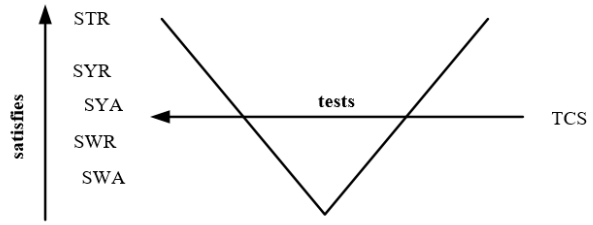

Abb. 4: Relationsarten

ARCHITEKTURAUSWAHL

Für die praktische Umsetzung stehen zwei Möglichkeiten an Architekturen zur Verfügung. Der erste Gedanke ist, das Data Mining Verfahren systemintegriert in SQUORE zu modellieren und auszuführen. Die zweite Idee ist die Verwendung des Analysetools KNIME zur Modellierung und Anwendung des geplanten Data Mining Verfahrens.
Aufgrund bevorstehender Updates in SQUORE und zusätzlich durchzuführenden Arbeiten an der Benutzeroberfläche ist die Entscheidung der praktischen Umsetzung auf das Analysetool KNIME gefallen. Die in SQUORE enthaltenen Daten können trotz Bearbeitungen einfach exportiert werden und so dem Tool KNIME als Input zur Verfügung gestellt werden. Somit ist ein ungestörtes Arbeiten mit den vorhandenen, zu untersuchenden Daten in KNIME unabhängig von SQUORE möglich.
Um einen Überblick über beide Architekturen zu geben, werden diese nachfolgend vorgestellt.

DATA WAREHOUSE SQUORE

Die SQUORING Technologien wurden 2010 von einer Gruppe aus Software Engineering Experten gegründet und sind auf die Evaluierung und Überwachung von Software- und Systementwicklungsprojekten spezialisiert. [SQUOoJ]

Die Hauptbausteine bestehen aus [SQOR17]:

- SQUORE-Parser und andere Datenprovider sind die Eingänge für den Prozess und stellen Basismetriken für das Analysemodell bereit.
- Analysemodelle definieren die Transformation zwischen den Basismetriken, die von Datenprovidern und anderen Metriken abgerufen werden.

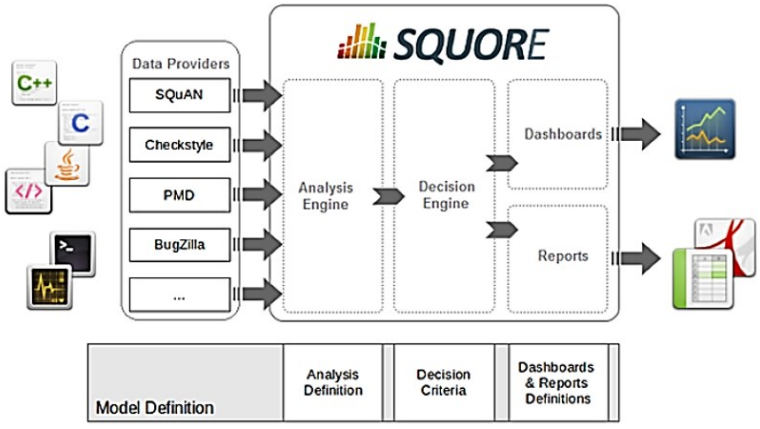

Abb. 3: SQUORE Prozess [SQOR17]

- Entscheidungsmodelle definieren, wie man Roh- und Analysedaten verarbeitet. Dabei werden sogenannte action items erhoben, welche die To-Dos definieren, die befolgt werden können, um die Qualität eines Projekts zu verbessern.
- Dashboards präsentieren die Gesamtergebnisse anschaulich. Sie sind gut anpassbar und können alle Informationen zeigen, die im täglichen Gebrauch von SQUORE benötigt werden.
- Berichte extrahieren Informationen und präsentieren sie in einem Dokument (PDF, Powerpoint oder Tabellenkalkulation). Sie können für die externe Berichterstattung verwendet werden, z.B. Wenn kein Zugriff auf die SQUORE-Schnittstelle besteht.

ANALYSETOOL KNIME

KNIME ist eine modulare Datenexplorationsplattform und wurde unter der Leitung von Prof. Berthold an der Universität Konstanz entwickelt. Die Abkürzung KNIME steht für Konstanz Information Miner. Die KNIME Analytics Plattform ist eine open-source Lösung für datengesteuerte Innovationen und ermöglicht, aus großen Datenmengen neue Erkenntnisse zu ziehen oder Zukunftsprognosen zu erstellen. [KNIM17]
Im Workflow Explorer sind alle Projekte aufgelistet. Hier können auch weitere Workflows (deutsch: Arbeitsabläufe) implementiert oder bereits bestehende Workflows exportiert werden. Die Workflow Coach enthält die favorisierten Knoten. Diese können ganz einfach aus dem Node Repository durch Drag&Drop hinzugefügt werden. Das Node Repository enthält alle Knoten, welche geordnet in Kategorien vorliegen. Outline bietet eine Navigation und ist sinnvoll bei größeren Workflows. Die Console gibt Status Informationen, Warnungen und Fehlermeldungen, welche ebenfalls in ein Log geschrieben werden. Die Node Description liefert eine genaue Beschreibung des jeweils ausgewählten Knoten. In der Mitte befindet sich der Workflow Editor, in welchem Worklows modelliert, konfiguriert und ausgeführt werden.
Ein Workflow bildet sich aus einer Reihe von Knoten, welche untereinander verbunden sind. Ein Knoten hat einen von drei Status, welche durch ein Ampelsymbol visualisiert sind. Rot bedeutet, dass der Knoten noch nicht konfiguriert wurde. Gelb steht für eine erfolgreiche Konfiguration. Grün bedeutet, dass der Knoten erfolgreich ausgeführt wurde.

EMPIRISCHER TEIL

DATENANALYSE

Die zu untersuchenden Daten werden wöchentlich aus dem Data Warehouse exportiert und als Dateien bereitgestellt. Dabei handelt es sich um insgesamt zwei Dateien im csv-Format, getrennt durch Strichpunkt.
Die darin enthaltenen Daten sind objektorientierte Artefakte. Ein Artefakt kann dabei ein Stakeholder, ein System Requirement, ein Software Architecture Element, ein System Architecture Element, ein Problem Report oder ein Change Request sein.
Dargestellt werden die Artefakte im Key-Value Format, welches so von SQUORE vorgegeben ist.
Die Dateien haben den Namen Objects2Artefacts.mtr.csv und Objects2Artefacts.lnk.csv. Die erste Datei beinhaltet diverse Informationen zu den einzelnen Artefakten. Die zweite Datei gibt an, wie die Artefakte miteinander verlinkt sind, also deren Traceability.
Die Datei Objects2Artefacts.mtr.csv besteht aus zwei unterschiedlichen Datensätzen. Diese sind der sogenannte IMS und DOORS Datensatz.

Der IMS Datensatz besteht aus den folgenden Informationen:

CHR/PBR	Abkürzung für Change Request oder Problem Report.
DWH_PATH	Pfad mit ID aus dem Data Warehouse.
IMS_ID	ID des Artefakts mit Präfix und eindeutiger Identifikationsnummer.
IMS_SUMMARY	Kurzbeschreibung über den CHR oder PBR.
IMS_STATE	Status. Dieser kann new, analyzed, classified, accepted, realized, approved oder closed sein.
IMS_LAST_MODIFIED_ DATE	Datum, an dem das Artefakt zuletzt bearbeitet wurde.
IMS_DUE_DATE	Das Fälligkeitsdatum.
IMS_PLANNED_TGT_REL	Geplantes Release, in dem das Artefakt umgesetzt werden soll.
IMS_FOUND_IN_REL	Angabe der Release Nummer, in welcher das Problem gefunden wurde. Information wird nur bei PBR angegeben.
IMS_IMPORTANCE_STR	Gibt die Wichtigkeit des Artefakts an. Diese kann in den Formen high, medium und low vorkommen.
IMS_CAUSED_BY	Diese Information bezieht sich nur auf das Artefakt PBR und gibt an, durch welche Prozessaktion ein Problem ausgelöst wurde.
IMS_DETECTED_BY	Diese Information bezieht sich nur auf das Artefakt PBR und gibt an durch wen oder was ein Fehler entdeckt wurde.
IMS_CREATED_DATE	Datum, an dem das Artefakt erstellt worden ist.
IMS_ANALYSIS_ACT_ DATE	Datum, an dem die Analyse des Artefakts abgeschlossen wurde.
IMS_DECISION_ACT_DATE	Datum, an dem beschlossen wurde, das Artefakt zu realisieren.

IMS_REALIZATION_ACT_DATE	Datum, an dem die Realisierung des Artefakts abgeschlossen wurde.
IMS_CLOSURE_ACT_DATE	Datum, an dem das Artefakt als abgeschlossen gekennzeichnet wurde.
IMS_DAYS_IN_NEW	Anzahl der Tage, bis das Artefakt vom Status new in den Status analyzed übergegangen ist. Also die Differenz zwischen IMS_ANALYSIS_ACT_DATE und IMS_CREATED_DATE.
IMS_DAYS_IN_ANALYZED	Anzahl der Tage, bis das Artefakt vom Status analyzed in den Status accepted übergegangen ist. Also die Differenz zwischen IMS_DECISION_ACT_DATE und IMS_ANALYSIS_ACT_DATE.
IMS_DAYS_IN_ACCEPTED	Anzahl der Tage, bis das Artefakt vom Status accepted in den Status realized übergegangen ist. Also die Differenz zwischen IMS_REALIZATION_ACT_DATE und IMS_DECISION_ACT_DATE.
IMS_DAYS_IN_REALIZED	Anzahl der Tage, bis das Artefakt vom Status realized in den Status closed übergegangen ist. Also die Differenz zwischen IMS_CLOSURE_ACT_DATE und IMS_REALIZATION_ACT_DATE.

Tab. 1: IMS Datensatz

Auch die Inhalte des DOORS Datensatzes werden im Folgenden kurz vorgestellt:

STR/SYR/SYA/SWR/SWA/TCS	Abkürzung für die o.g. Artefakte.
DWH_PATH	Pfad des Artefakts aus dem Data Warehouse.
DRS_NAME	ID des Artefakts, bestehend aus Präfix und eindeutiger Identifikationsnummer.
DRS_PATH	Pfad des Artefakts aus dem Tool DOORS.
DRS_DESCRIPTION	Kurzbeschreibung des Artefakts.
DRS_LAST_MODIFIED_DATE	Datum, an dem das Artefakt zuletzt bearbeitet wurde.
DRS_NON_FUNC_STR	Angabe, ob eine Anforderung funktional oder nicht funktional ist
DRS_COMPLIANCE_STR	Nur für Stakeholder Requirements relevant. Hier wird angegeben ob und inwieweit Kundenwünsche eingehalten werden.
DRS_MATURITY_STR	Gibt den momentanen Status im Projektlebenszyklus an.
DRS_DISCIPLINE_STR	Informationswert für Anforderungen. Angabe, um welche Disziplin es sich handelt
DRS_RELEASE_STR	Angabe der Release Nummer.
DRS_IMPL_STATUS	Bezieht sich nur auf Anforderungen. Gibt an, ob eine Anforderung bereits implementiert ist oder nicht
DRS_VERIFIED_METHOD	Dieser Wert legt fest, welches Level das entsprechende Arte-fakt zu verifizieren hat. Beispielsweise bedeutet Syt Systemtest oder SWIT Softwareintegra-tionstest.
DRS_ASIL_STR	Mit dem Kürzel QM wird angegeben, dass eine Qualitätssicherung durch das Quality Management gesichert ist.
DRS_SEVERITY_STR	Gibt mit den Werten top, high, medium und low an, inwieweit eine Anforderung, die Kundenzufriedenheit und Sicherheit bei sicherheitskritischen Systemen erfüllt sind.
DRS_RESULT_TEST_RUN	Bezieht sich nur auf das Artefakt Test Case und gibt an, ob dieser bestanden oder nicht bestanden hat, ob er gar nicht oder nicht vollständig ausgeführt wurde.
DRS_AUTO_MAN_STR	Bezieht sich nur auf das Artefakt Test Case. Angabe über dessen Durchführung mit den Werten manual, auotmatic, to_automate oder in_implementation. Manual gibt an, dass ein Test Case manuell durchgeführt wird, wohingegen automatic bedeutet, dass dieser automatisiert abläuft. Wenn ein Test Case noch automatisiert werden muss, wird to_automate angegeben. In_implementation sagt aus, dass ein Test Case bereits in der Implementierung ist.

Tab. 2: DOORS Datensatz

Bei Betrachtung des gesamten Datensatzes ist aufgefallen, dass alle angegebenen Datumswerte nicht in einem typischen Datumsformat wie beispielsweise im Format JJ-MM-TT angezeigt werden. Wie sich herausgestellt hat, handelt es sich dabei um UNIX Zeitstempel. Zudem gibt es ungültige Datumswerte, welche man am Wert -1 erkennt. Dies hat auch Auswirkungen auf die Spalten, welche eine Anzahl an Tagen angeben, die sich aus den gegebenen Datumswerten errechnen. Die Angabe #TBD# zeigt an, dass ein Wert nicht gepflegt wurde. Oft gibt es auch leere Strings. Bei diesen muss aber zukünftig beachtet werden, ob diese eine Bedeutung haben oder nicht und eventuell durch einen anderen Wert ersetzt werden müssen.

Die Datei Objects2Artefacts.lnk.csv enthält die Relation zweier Artefakte, jedoch handelt es sich hierbei nur um DOORS Artefakte. Anhand eines kleinen Datenausschnittes wird der Inhalt der Datei erklärt. Die Pfadagabe wird durch Auslassung bestimmter Pfadteile unkenntlich gemacht, da sonst der Projektname herausgelesen werden kann, welcher aus Datenschutzgründen nicht veröffentlicht werden darf. Ebenso wurden die IDs durch andere Zahlenwerte ersetzt, um diese unkenntlich zu machen.

SYR	System Require-ments/.../ID_1	STR	Stakeholder Require-ments/…/ID_6	SATIS FIES

Tab. 3: Relation zweier DOORS Artefakte

In der ersten Spalte steht die Abkürzung des Artefakts. In diesem Fall bedeutet das Kürzel SYR ausgeschrieben System Requirement. Der zum Artefakt zugehörige Pfad ist in Spalte zwei angegeben. Spalte drei und vier sind wie eins und zwei aufgebaut. Das Kürzel STR in Spalte drei steht für das Artefakt Stakeholder Requirement. Die letzte Spalte gibt die Traceability an, hier „satisifies“. Auf das Beispiel bezogen bedeutet das, dass eine Systemanforderung mit zugehörigem Pfad eine satisfy, also somit eine vertikale Traceability zu einer Stakeholderanforderung mit entsprechendem Pfad hat. Das bedeutet also, dass sich eine Systemanforderung mit eindeutigem Pfad auf eine Stakeholderanforderung mit eindeutigem Pfad bezieht

Bei Betrachtung der Daten ist aufgefallen, dass nur die Relationen der DOORS Artefakte angegeben sind. Die Beziehungen der IMS Artefakte werden nicht berücksichtigt.

DATENAUFBEREITUNG

IMS DATENSATZ

Für die Aufbereitung des IMS Datensatzes sind zwei Workflows notwendig. Einer für Change Request Artefakte und einer für Problem Report Artefakte.
Hier wird exemplarisch der Workflow zur Aufbereitung der CHR Artefakte gezeigt (vgl. Abb. 5) und erklärt.

Die csv-Datei Objects2Artefacts.csv wurde in eine Excel konvertiert, um vorhandene Formatschwierigkeiten aufzuheben. Diese lässt sich über den Knoten „Excel Reader“ unter Pfadangabe einlesen. Um nur die IMS Daten aus den gesamten Datensatz zu erhalten, wird der Knoten „Rule based row filter verwendet“. In dessen Konfiguration wird Option „Include TRUE Matches“ ausgewählt und die Regel „$Col2$ = "IMS_ID" => TRUE“ formuliert. Mit dieser Regel wird also Spalte zwei nach dem Inhalt „IMS_ID“ durchsucht, welche nur im IMS Datensatz vorkommt, wodurch dann eben nur IMS Daten rausgefiltert werden. Im nächsten Schritt wird der „column filter“ eingesetzt. Mit diesem lassen sich mit Hilfe der include und exclude Optionen die vorhandenen Spalten beibehalten oder entfernen. Hier werden die Key-Spalten entfernt und die Value-Spalten beibehalten. Die Value Spalten werden mit Hilfe des Knoten „column rename“ beschriftet und bekommen so die Keys als Überschriften. Der nachfolgende graue Knoten ist ein sogenannter Metanode. Ein Metanode kann mehrere Knoten oder auch weitere Metanodes enthalten. Somit lassen sich Teile des Workflows in einen Knoten zusammenfassen, wodurch sich der Gesamtworkflow optisch verkleinert. Dieser Metanode enthält sieben Mal den Knoten „Rule Engine“, welche ungültige Datumsangaben mit dem Wert -1 entfernt. Sieben Knoten sind notwendig, da der IMS Datensatz sieben Datumsspalten enthält. Bei den Datumsangaben gibt es jedoch noch ein weiteres Problem. Das Datum ist nämlich als UNIX Timestamp angegeben. Um es in das Datumsformat YY-MM-DD zu konvertieren, wird der Knoten „R Snippet“ verwendet. Dieses enthält das R-Skript zur Datumstransformation. Nach Codeausführung wird das Datum im Format YYYY-MM-DD angegeben, jedoch als Datentyp String. Mit dem Metanode „String to Date/Time“ kann auch dieses Problem behoben werden, sodass alle Datumsangaben in den Datentyp Date konvertiert werden. Da sich die DAYS_IN Spalten aus den Datumsspalten errechnen, können die bisher enthaltenen Angaben, welche fehlerhaft erscheinen, durch eine Neuberechnung mit dem Knoten „Time Difference“ ersetzt werden. In diesem Knoten müssen nur die Datumswerte, die berechnet werden sollen, angegeben werden. Ergebnis ist eine Zahl, welche in diesem Fahl die Differenz zweier Datumswerte als Anzahl an Tagen angibt. Diese Tagesanzahl wird jedoch im Datentyp Double angegeben.
Da in diesem Fall ein integer erwünscht ist, kann der Datentyp mit dem Knoten „Double to Int“ konvertiert werden. Im nächsten Schritt werden mit dem „Column Filter“ die für die in dieser Arbeit durchzuführenden Untersuchung unnütze Spalten entfernt. Diese werden im Folgenden mit einer kurzen Begründung für deren Entfernung aufgezeigt:

- IMS_SUMMARY: Unrelevant, da die Kurzbeschreibung über das Artefakt nur dem User dient und keine Auswirkung auf den Prozess hat.
- IMS_LAST_MODIFIED_DATE: Es ist für den Prozess unrelevant, wann das Artefakt zuletzt bearbeitet wurde.

- IMS_PLANNED_TGT_REL: Unrelevant, da es keine vollständig gepflegte Übersicht aller Releases gibt.
- IMS_FOUND_IN_REL: Unrelevant, da es keine vollständig gepflegte Übersicht aller Releases gibt.

Mit dem Knoten „Column Rename“ werden alle DAYS_IN Spalten umbenannt. Dabei wird nur die Zeichenfolge (#1) entfernt, welche nach Neuberechnung automatisch hinzugefügt wurde. Der nächste Knoten „Rule-based Row Filter“ filtert CHR Artefakte mit Auswahl der Option „Include TRUE matches“ und der Regel „$ITEM$ = "CHR" => TRUE“. Mit dem „Column Filter“ können dann im nächsten Schritt die Spalten, welche nur bei Problem Reports vorkommen, entfernt werden. Diese sind „IMS_CAUSED_BY“ und „IMS_DETECTED_BY“.

zusammengefügt. Der „Column Filter“ entfernt die nun überflüssigen Original Datumsspalten. Mit „Column Resort“ werden alle Quartalsangaben jeweils neben die neu zusammengefügten Datumsspalten platziert. Die Anzahl der Tage wird in allen DAYS_IN Spalten in Tagesbereiche untergliedert. Dies erfolgt über den Knoten „Nummeric Binner“. In dessen Konfiguration können pro Spalte die einzelnen nummerischen Intervalle, sogenannte Bins, festgelegt werden. Hier wird dann auch der Bereich für den 1.000.000 Wert festgelegt und als missing bezeichnet, um diesen nachher von der Bewertung einfach ausschließen zu können. Im nächsten Schritt werden zu den Spaltenwerten die zugehörigen Spaltennamen als Präfix hinzugefügt. Dies ist notwendig, da am Ende alle Spalten in einem Listenformat pro Zeile zusammengefügt werden.

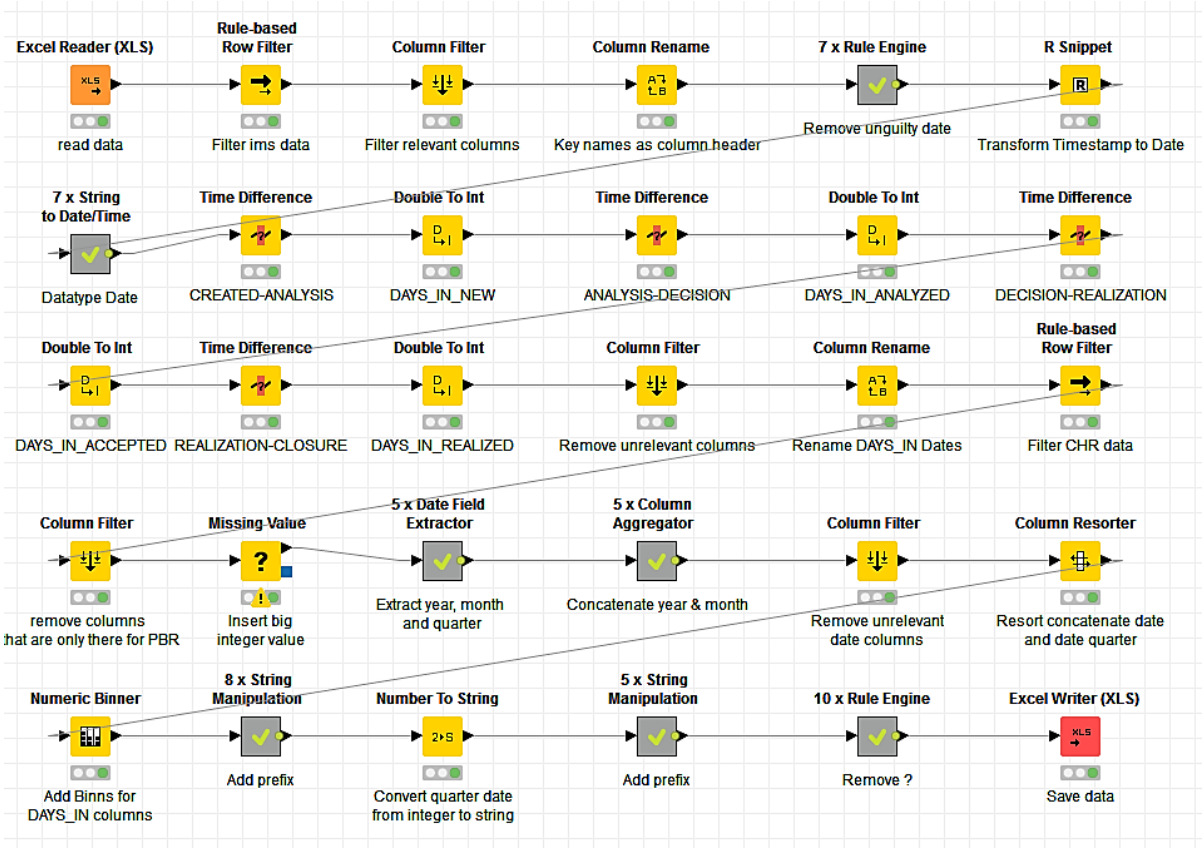

Abb. 5: Workflow zur Datenaufbereitung der IMS Change Request Artefakte

Anschließend wird mit dem Knoten „Missing Value” für alle fehlenden Werte, die den Datentyp Integer besitzen, der Wert 1.000.000 eingetragen. Dies hat den Sinn, dass alle ursprünglich fehlenden Werte sofort an dieser sehr hohen Zahl erkannt und bei der Ergebnisauswertung rausgefiltert werden können. Mit den Knoten „Date Field Extractor“, welche in einem Metanode zusammengefasst sind, werden aus allen Datumspalten die Informationen Jahr, Monat und Quartal gefiltert. Jahr und Monat werden dann mit den Knoten „Column Aggregator“, welche sich ebenfalls in einem Metanode befinden, durch Auswahl der Funktion concatenate, welche zwei Strings verbindet, in einer neuen Spalte

Um den Spaltennamen hinzuzufügen wird der Knoten „String Manipulation“ verwendet. Mit den Knoten „Rule Engine“ können letztendlich fehlende Werte, die mit „?-?“ oder „?“ gekennzeichnet sind, durch einen leeren String ersetzt werden. Dies erfolgt bei den Spalten, welche eine Zeitangabe beinhalten. Denn diese fehlenden Werte geben nur an, dass das Artefakt noch nicht den nächsten oder letzten Status erreicht hat und somit keine Zeitangabe bezüglich des Statuswechsels vorliegt. Im letzten Schritt werden die Daten per Excel Writer unter Angabe des gewünschten Zielordners abgesichert.

Hier wird exemplarisch ein aufbereiteter Change Request gezeigt:

IMS_ID	ID_1000
IMS_STATE	Closed
IMS_IMPORTANCE	High
IMS_DAYS_IN_NEW	NEW_[duration]
IMS_DAYS_IN_ANALYZED	ANALYZED_[duration]
IMS_DAYS_IN_ACCEPTED	ACCEPTED_[duration]
IMS_DAYS_IN_REALIZED	REALIZED_[duration]
CREATED_DATE	CREATED_2017-month
CREATED_DATE_QUARTER	CREATED_1
ANALYSIS_DATE	ANALYSIS_2017-month
ANALYSIS_DATE_QUARTER	ANALYSIS_Q
DECISION_DATE	DECISION_2017-month
DECISION_DATE_QUARTER	DECISION_Q
REALIZATION_DATE	REALIZATION_2017-month
REALIZATION_DATE_QUARTER	REALIZATION_Q
CLOSURE_DATE	CLOSURE_2017-month
CLOSURE_DATE_QUARTER	CLOSURE-Q

Tab. 4: Datenausschnitt aus der Datenaufbereitung der IMS Change Request Artefakte

Im Folgenden wird der erste Workflow beschrieben (Abb. 6).
Im Gegensatz zu einem IMS Artefakt wird der Status bei einem DOORS Artefakt überschrieben ohne Informationen über den vorherigen Status in anderen Spalten zu hinterlassen. Somit benötigen wir die Datenaufbereitung mit Schleife, um die zeitliche Veränderung des Status eines Artefakts verfolgen zu können.
Mehrere Dateien lassen sich dabei mit dem Knoten „List Files“ einlesen. In dessen Konfiguration wird über eine Pfadangabe auf den Ordner, welcher alle wöchentlichen Dateien enthält, verwiesen. Nach dessen Einlesen folgt der Beginn einer Schleife mit dem Knoten „Table Row To Variable Loop Start“. Beim Knoten „Excel Reader“ muss die Konfiguration so abgeändert werden, dass eine Pfadangabe mit Variable angegeben wird, damit die wöchentlichen Dateien mit verschiedenen Namen eingelesen werden können. Um den DOORS Datensatz zu filtern wird beim nächsten Schritt im Knoten „Rule based Row Filter“ die Regel „$Col2$ = "DRS_NAME" => TRUE“ formuliert und die Option „Include TRUE matches“ ausgewählt.

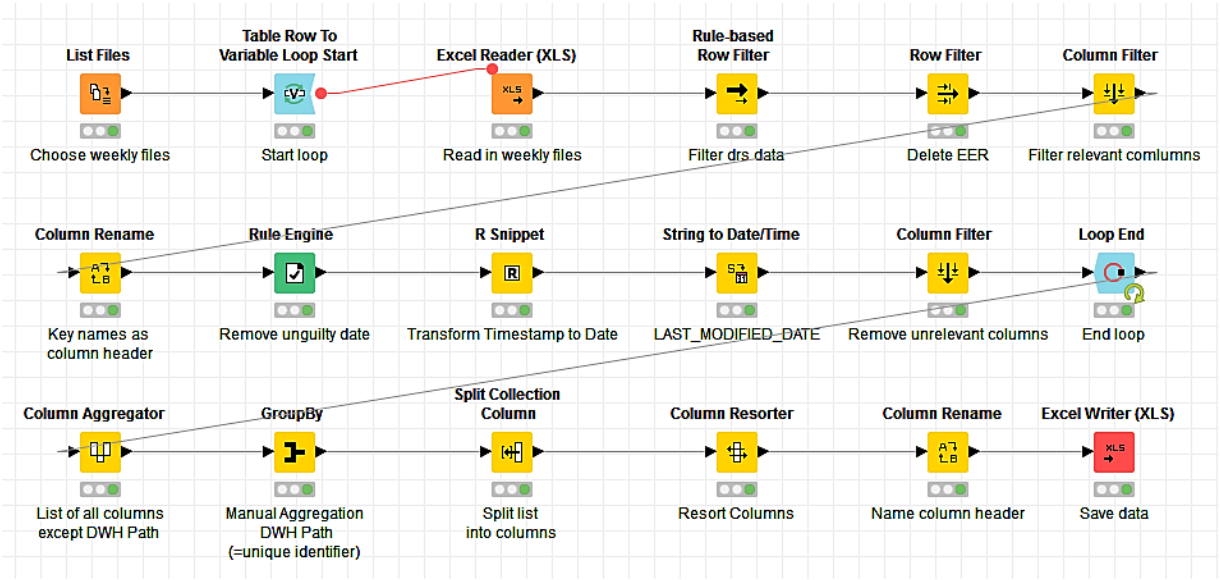

Abb. 6: Workflow zur Aufbereitung des DOORS Datensatzes mit zeitlicher Berücksichtigung

Die ID und Datums- bzw. Zeitangaben wurden unkenntlich gemacht, da diese dem Datenschutz unterliegen und nicht veröffentlicht werden dürfen.
Ein aufbereitetes PBR Artefakt und dessen Workflow zur Datenaufbereitung sind der Aufbereitung und dem Inhalt eines CHR Artefakts sehr ähnlich. Unterschiede sind hier, dass im Workflow nach PBR gefiltert wird und das aufbereitete PBR Artefakt zusätzlich noch die Informationen „IMS_CAUSED_BY“ und „IMS_DETECTED_BY“ enthält.

DOORS DATENSATZ

Für die Aufbereitung des DOORS Datensatzes sind zwei Workflows notwendig, welche nacheinander beschrieben werden.

Da keine Hardware Anforderungen, welche mit dem Kürzel EER bezeichnet werden, ausgewertet werden sollen, werden diese mit dem Knoten „Row Filter“ entfernt. Dabei wählt man die Option „Exclude rows by attribute values“ und gibt als value EER an. Mit den Knoten „Column Filter“ und „Column Rename“ werden ebenso die Key-Spalten entfernt und die Value-Spalten entsprechend ihrer Keys benannt. Mit dem Koten „Rule Engine“ wird auch hier der ungültige Datumswert -1 entfernt. Jedoch gibt es im DOORS Datensatz nur die einzige Datumsangabe DRS_LAST_MODIFIED_DATE.
Dieses enthält ebenso einen UNIX Timestamp und wird gleichermaßen mit dem Knoten „R-Snippet“ bearbeitet, um das Zielformat YYYY-MM-DD zu erreichen. Nach erfolgreicher Transformation wird die Datumsspalte,

welche den Datentyp String hat in den Datentyp Date mit Hilfe des Knotens „String to DATE/TIME“ konvertiert. Ebenso werden auch hier für die Analyse irrelevante Informationen entfernt. Diese werden mit Begründung der Entfernung aufgelistet:

- DRS_DESCRIPTION: Unrelevant, da es sich hierbei nur um eine Information für den Benutzer handelt.
- DRS_LAST_MODIFIED_DATE: Für den Prozess unrelevant, wann das Artefakt zuletzt bearbeitet wurde.
- DRS_NON_FUNC_STR: Da der Datensatz nur funktionale Anforderungen enthält, ist diese Spalte überflüssig.
- DRS_RELEASE: Unrelevant, da es keine vollständig gepflegte Übersicht aller Releases gibt.
- DRS_VERIF_METHOD: Unrelevant, da unzureichend gepflegt.
- DRS_ASIL: Unrelevant, da unzureichend gepflegt.
- DRS_SEVERITY: Unrelevant, da unzureichend gepflegt.

wendung des Data Mining Verfahrens berücksichtigt werden können.
Um dies zu realisieren, muss die Datei „Objects2Artefacts.lnk.csv“, welche die Relation der DOORS Artefakte angibt, um den Inhalt des bisher aufbereiteten DOORS Datensatzes erweitert werden.
Beide Dateien werden mit Hilfe von „Excel Reader“ und „CSV Reader“ eingelesen. Mit dem Knoten „Column Rename“ werden die Spaltennamen der Datei „Objects2Artefacts.lnk.csv“ eindeutig benannt. Diese Benennung ist notwendig, da im darauffolgenden Knoten „JOIN“ die Spaltennamen angegeben werden müssen, anhand denen eine Zusammenführung der Daten vollzogen werden soll. Dabei handelt es sich um den sogenannten „Left Outer Join“. Dieser füllt die Spalten, die aus der unteren Tabelle kommen, mit fehlenden Werten aus, wenn keine passende Zeile in der unteren Tabelle existiert. Die untere Tabelle ist in diesem Fall die Datei, welche die Relation der Artefakte angibt. Diese wird also um Informationen aus dem bisher aufbereiteten DOORS-Datensatz ergänzt. Notwendig sind

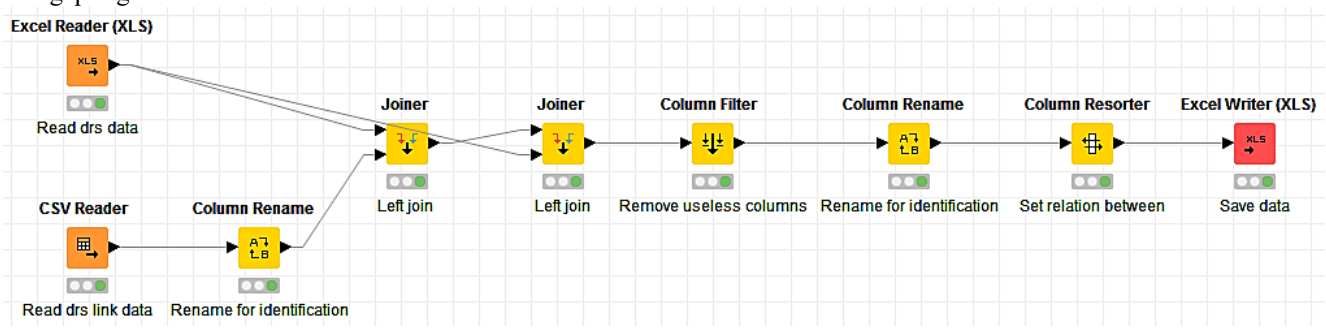

Abb. 7: Workflow zur Aufbereitung des DOORS Datensatzes mit Berücksichtigung der Relation

Um das Ende der Schleife zu markieren, benötigt man den Knoten „Loop End“. Nach diesem folgen die Knoten „Column Aggregator“, „Group By“, „Split Collection Columns“, „Column Resorter” und „Column Rename”, mit denen alle Duplikate entfernt werden. Im „Column Aggregator“ werden alle Attribute bis auf das Attribut „DWH Path“ zu einer Liste zusammengefügt. Der DWH Pfad ist dabei der eindeutige Identifikator, anhand dessen sich die Liste im Knoten „Group By“ sortieren lässt. Dabei wird pro Identifikator jeweils die erste Liste aus einer Menge gleicher Listen gefiltert, wodurch Duplikate entfernt werden. Um wieder Spalten zu erhalten, wird der Knoten „Split Collection Columns“ benutzt, der für jedes Attribut in der Liste eine Spalte erstellt. Mit dem „Column Resorter“ wird die Reihenfolge der Attribute bestimmt. Zuletzt müssen den Spalten mit „Column Rename“ Überschriften hinzugefügt werden, da diese durch die Umformung in eine Liste verloren gegangen sind. Anschließend werden die Daten wieder mit dem „Excel Reader“ gesichert.

Da beim DOORS Datensatz die Relation der Artefakte berücksichtigt wird, ist ein weiterer Workflow zur Datenaufbereitung notwendig (siehe Abb. 7), damit die Beziehungen der Artefakte untereinander bei der Anin diesem Fall zwei „Left Joins“.
Somit wird das Artefakt links neben der Relationsangabe anhand der Spalte DWH-Path und das Artefakt rechts neben der Relationsangabe anhand der Spalte DWH-Path um weitere Spalten ergänzt.
Nach der Zusammenführung der Daten werden zwei neu entstandene, jedoch unnütze Spalten entfernt. Mit „Column Rename“ werden die Spaltennamen links neben der Relation mit „left“ und die Spaltennamen rechts neben der Relation mit „right“ ergänzt. Die Spalte Relation wird mit dem Knoten „Column Resorter“ genau zwischen den Artefakten platziert. Im letzten Schritt werden die Daten mit „Excel Writer“ gesichert.
Ein Datenausschnitt wird im Kapitel ASSOZIATIONSVERFAHREN gezeigt, da der Datensatz vor dem Verfahren noch aufgeteilt werden muss.

ASSOZIATIONSVERFAHREN

IMS DATENSATZ

Für den IMS Datensatz wird das Assoziationsverfahren für Change Request Artefakte (siehe Abb. 8) gezeigt und erklärt, da im vorherigen Kapitel bereits dessen Aufbereitung beschrieben wurde.

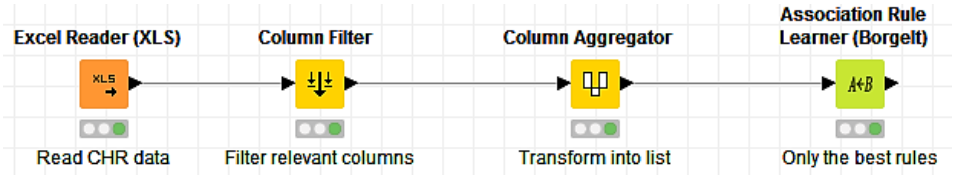

Abb. 8: Workflow des Assoziationsverfahrens für IMS Change Request Artefakte

Zur Modellierung des Assoziationsverfahrens wird der aufbereitete Datensatz der CHR Artefakte per Excel Reader eingelesen. Mit dem „Column Filter“ werden die Spalten „Item“ und „DWH Path“ entfernt, da diese unbrauchbar für die Analyse sind.
Der „Column Aggregator“ formt nun alle zuletzt vorhandenen Spalten in eine Liste um, wodurch dann der Datensatz mit dem Assoziationsverfahren verbunden werden kann. Dafür wird der Knoten „Association Rule Learner (Borgelt)“ (siehe Abb. 10) verwendet.
In dessen Konfiguration muss ein Wert für die Option „Minimum Support“ und ein Wert für die Option „Minimum Rule Confidence“ angegeben werden. Diese Werte sind die Mindestzahl der jeweiligen Option, welche erreicht werden muss, damit diese ausgewertet und angezeigt wird.
Um vorab keine minderwertigen Regeln zu erhalten wird die Konfiguration entsprechend wie in Abb. 9 dargestellt, eingestellt. Für die „Minimum set size“ wird der Wert zwei festgelegt, damit nur Item Sets mit einer minimalen Länge von zwei, also aus zwei Artefakt bestehenden Sets, ausgegeben werden. Der „Minimum support“ ist auf 10 % eingestellt. Somit werden nur Sets aus Artefakten ausgegeben, welche mindestens zu 10% in der Datengrundlage vorhanden sind. In diesem Fall sind nämlich Regeln, die auf eine Mehrzahl der Artefakte zugreifen, erwünscht. Unter einem Support von 10% wird die Datengrundlage als zu gering eingeschätzt. Die Minimum Rule Confidence bestimmt die Richtigkeit einer Regel. Um aussagekräftige Regeln zu erhalten, stellt man die „Minimum rule confidence“ auf 35% ein, damit gesichert ist, dass die Regeln mit einem Prozentsatz von mindestens 35% zutreffen.
Das Assoziationsverfahren für PBR Artefakte besteht aus dem selbem Workflow, nur dass beim Knoten „Association Rule Learner (Borgelt)“ ein Minimum support von 15 % eingestellt wird. Dies hat den Grund, dass es sich durch die zwei zusätzlichen Informationen „IMS_CAUSED_BY“ und „IMS_DETECTED_BY“ um eine größere Datengrundlage handelt, weswegen stärker gefiltert werden muss.

DOORS DATENSATZ

Da es beim DOORS Datensatz Spalten gibt, welche nur bei Testfällen oder nur bei Stakeholder Requirements vorkommen, wird der Datensatz in drei Teile aufgeteilt. Zum einen in einen Datensatz ohne die Artefaktobjekte Test Case und Stakeholder Requirement, zum anderen in einen Datensatz nur mit dem Artefakt Test Case und zuletzt in einen Datensatz nur mit dem Artefakt Stakeholder Requirement.
Hier wird das Assoziationsverfahren für Test Case Artefakte vorgestellt, wofür der in Abb. 10 gezeigte Workflow modelliert wurde.
Nach Einlesen des aufbereiteten DOORS Datensatzes über den Knoten „Excel Reader“ werden möglicherweiße fehlende Werte in den Spalten „DRS_IMPL_STATUS_left“ und „DRS_IMPL_STATUS_right“ durch den Wert „not_implemented“ ersetzt. Beim Knoten „Rule-based Row Filter“ werden mit der Option „Include TRUE matches“ und dem Befehl „$ITEM_left$ = "TCS" => TRUE“ nur die Artefakt Objekte Testfälle beibehalten. Im Anschluss wird wie beim ersten Datensatz die Beziehung zweier Artefakte zusammengefügt, dessen Originalspalten entfernt und die Relationsangabe an erster Stelle gesetzt. Ebenso werden Spalten, welche für Testfälle keine Bedeutung haben entfernt. Diese sind die Spalten „DRS_DISCIPLINE_left“, „DRS_DISCIPLINE_right“ und „DRS_COMPLIANCE_right“. Zusätzlich werden auch hier die Spalten „DWH_PATH_left“, „DRS_PATH_left“, „DWH_PATH_right“ und „DRS_PATH_right“ entfernt, da diese für die bevorstehende Untersuchung nicht brauchbar sind.

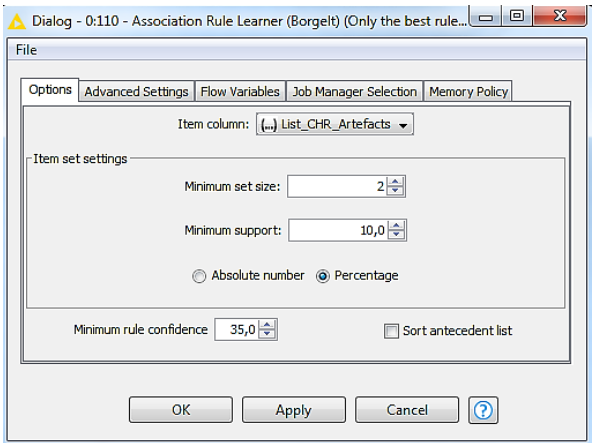

Abb. 9: Association Rule Learner (Borgelt)

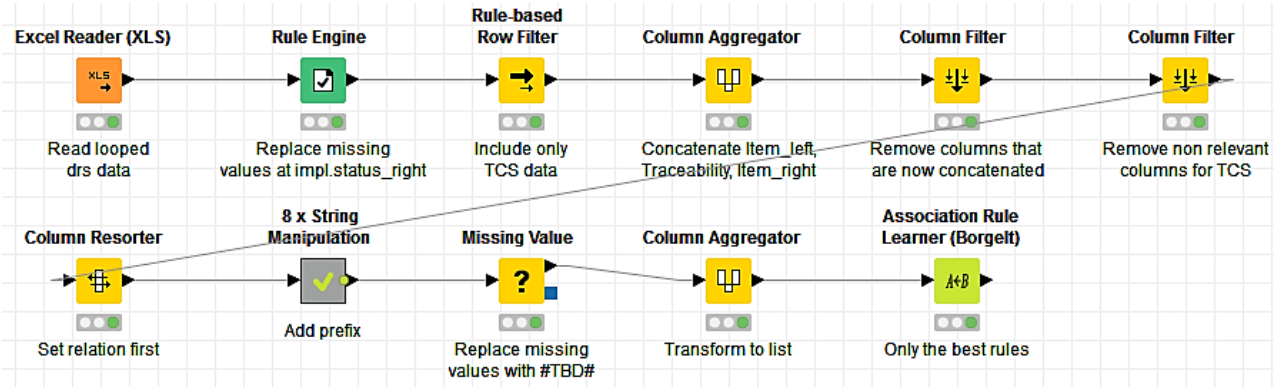

Abb. 10: Workflow des Assoziationsverfahrens für DOORS Test Case Artefakte

In den nächsten Schritten werden auch die Spaltennamen als Präfix für die Spaltenwerte mit Hilfe des Knotens „String Manipulation“ gesetzt. Schließlich werden noch alle im Datensatz möglich auftauchenden fehlende Werte mit #TBD# ersetzt, was für die Bezeichnung „nicht gepflegt“ steht, bevor die Spalten einer Liste zusammengefasst werden, was mit dem „Column Aggregator“ geschieht.

Folgender Datenausschnitt soll den aufbereiteten Datensatz näher bringen:

Relation	TCS_TESTS_SYR
ITEM_ID_left	left_id_348
DRS_MATURITY_left	left_maturity_project_accepted
DRS_RESULT_TEST_RUN_left	left_result_test_not_done
DRS_AUTO_MAN_left	left_auto_to_automate
ITEM_ID_right	right_id_312
DRS_MATURITY_right	right_maturity_project_accepted
DRS_DISCIPLINE_right	right_discipline_SW
DRS_IMPL_STATUS_right	right_impl_implemented

Tab. 5: Datenausschnitt aus der Datenaufbereitung der DOORS Test Case Artefakte

In Tab. 3 wurden beide IDs durch beliebige Zahlen ersetzt, um das tatsächliche Projekt und dessen Artefakt ID unkenntlich zu machen.

Der letzte Knoten ist der „Association Rule Learner (Borgelt)“. Die Einstellungen in der Konfiguration des Knotens betragen ebenfalls für die „Minimum set size“ zwei, dem „Minimum support“ 10% und die „Minimum rule confidence“ 35%.

Das Assoziationsverfahren für die anderen DOORS Artefakte ist dasselbe, nur das diese eben ihre zugehörigen Spalten aufweisen.

ERGEBNISSE

Das Assoziationsverfahren liefert zahlreiche Regeln. Diese können auf Grund ihrer Eigenschaften in drei verschiedene Bereiche aufgeteilt werden. Diese Bereiche werden hier als Regeltypen bezeichnet und in den nachfolgenden Kapiteln erklärt.

REGELTYP 1: DATENQUALITÄT

Regeln, die diesem Regeltyp zugeordnet werden, besitzen eine RuleConfidence nahe der 100%. Solche Regeln weisen auf Datenqualitätsprobleme hin. Was hierbei interessiert ist der Prozentsatz, der zur Erfüllung der 100%, fehlt. Dies wird anhand eines Beispiels verdeutlicht.

Consequent	Antecedent	RelativeItemSetSupport(%)	RuleConfidence(%)	RuleLift(%)
closed	[ACCEPTED_[duration]], [REALIZED_[duration]]	28,49	99	155,84

Tab. 6: Datenausschnitt aus der Datenaufbereitung der IMS Change Request Artefakte

Die Zeitangaben wurden in beiden Statusangaben aus Datenschutzgründen durch den Wert duration ersetzt
Diese Regel gibt Abhängigkeiten zwischen den Informationen eines Change Requests wieder. Der linke Teil der Regel, Antecedent, enthält die Werte [ACCEPTED_[duration]] und [REALIZED_[duration]]. Der rechte Teil der Regel, Consequent, enthält den Wert closed. Die RuleConfidence liegt bei 99%. Ein RelativeItemSetSupport mit 28,49% verdeutlicht eine gute Datengrundlage, da das Item Set häufig vorkommt. Die RuleLift ist bei diesem Regeltyp irrelevant, da sie das Aufdecken von Datenqualitätsproblemen nicht beeinflusst und deswegen nicht beachtet wird.
Aussage der Regel ist, dass ein Change Request, welcher mit einer bestimmten Dauer im Status accepted und einer bestimmten Dauer im Status realized war, sich zu 99% im Status closed befindet.
Jedoch wird der Status realized aus den Angaben IMS_REALIZATION_ACT_DATE und IMS_CLOSURE_ACT_DATE berechnet. Hier liegt also ein Fehler in der Datenqualität vor. Denn wenn es ein IMS_CLOSURE_ACT_DATE gibt, welches zur Berechnung verwendet wurde, muss der Status closed sein.
Um herauszufinden, weswegen dies nicht erfüllt ist, wird der Original Datensatz, also die Datei Objects2Artefact.mtr.csv untersucht.

Dort stößt man auf folgenden Change Request:

ID	ID-2654
State	realized
DAYS_IN_NEW	NEW_[duration]
DAYS_IN_ANALYZED	ANALYZED_[duration]
DAYS_IN_ACCEPTED	ACCEPTED_[duration]
DAYS_IN_REALIZED	REALIZED_[duration]
CREATED_DATE	CREATED_2017-month
ANALYSIS_DATE	ANALYSIS_2017-month
DECISION_DATE	DECISION_2017-month
REALIZATION_DATE	REALIZATION_2017-month
CLOSURE_DATE	CLOSURE_2017-month

Tab. 7: Change Request mit fehlerhafter Statusangabe

Die ID, Dauer in Tagen eines Status und Monatsangaben wurden aus Datenschutzgründen unkenntlich gemacht.
Wie sich deutlich erkennen lässt, beinhaltet dieser Change Request eine fehlerhafte Angabe im Status. Es handelt sich also um eine Schwäche in der Datenqualität, welche aber erkannt und identifiziert wurde.

REGELTYP 2: REDUNDANZ

Regeln aus diesem Regeltyp besitzen eine RuleConfidence von 100% und decken dadurch Redundanzen in der Datengrundlage auf. Auch hier wird ein Beispielregel gezeigt.

Consequent	Antecedent	RelativeItemSetSupport(%)	RuleConfidence(%)	RuleLift(%)
ANALYSIS_Q	[ANALYSIS_2017-month, closed]	17,969	100	272,34

Tab. 8: Beispielregel eines Problem Reports für den Regeltyp Redundanz

Bei der aufgezeigten Regel wurde aus Datenschutzgründen die Statusdauer- und Quartalsangabe unkenntlich gemacht.
Diese Regel gibt Abhängigkeiten zwischen Informationswerten eines Problem Reports wieder. Die Häufigkeit des ItemSets von 17,969% steht für eine gute Datenbasis. Die RuleLift ist bei diesem Regeltyp ebenso irrelevant, da sie die Aufdeckung von Redundanzen nicht beeinflusst, weswegen sie nicht beachtet wird. Die RuleConfidence liegt bei diesem Regeltyp immer bei 100%. Der linke Teil der Regel enthält den Wert [ANALYSIS_2017-month] und der rechte Teil ANALYSIS_Q. Dies sagt aus, dass ein im gegebenen Monat 2017 analysierter Problem Report zu 100% in einem gewissen Jahresquartal analysiert wurde. Jedoch besitzt diese Regel keine Aussagekraft, da Consequent mit einer logischen Schlussfolgerung gleichgesetzt werden kann.

REGELTYP 3: INHALTLICHE ERKENNTNIS

Dieser Regeltyp beinhaltet Regeln, welcher neue inhaltliche Erkenntnisse aus der Datengrundlage generiert. Hier kann die RuleConfidence zwischen 35% und 89% liegen.
Bei diesen Regeln müssen alle drei Faktoren RuleConfidence, ItemSetSupport und RuleLift beachtet werden, da solch eine Regel nach deren Auffinden auch noch nach Bedeutung bewertet werden muss.

Consequent	Antecedent	RelativeItemSetSupport(%)	RuleConfidence(%)	RuleLift(%)
left_maturity_proposed	TCS_TESTS_SWR	6,5722	40	122,31
left_maturity_project_accepted	right_maturity_proposed	11,558	51,4	78,03
left_maturity_project_accepted	TCS_TESTS_SYR, right_impl_implemented, right_maturity_project_accepted	12,629	85,7	130,1

Tab. 9: Beispielregeln eines Test Cases für den Regeltyp inhaltliche Erkenntnis

Die aufgezeigten Regeln werden nacheinander beschrieben.
Die erste Regel zeigt die Abhängigkeit zwischen der Relation eines Test Cases mit einem Software Requirement und der Maturity eines Test Cases. Hier fällt jedoch auf, dass der RelativeItemSetSupport einen geringen Wert unter 10% besitzt, weswegen die Regel wenig bedeutsam ist, da eine sehr geringe Datengrundlage vorliegt. Zudem ist die RuleConfidence unter 50%. Diese Regel wird also als weniger bedeutsam und deswegen als schlechte Regel angesehen. Eine RuleLift über 100% hat hier keinen Wert.
Die nächste Regel betrifft den Zusammenhang zwischen der Maturity eines linken und rechten Artefakts. Der RelativeItemSupport beträgt über 10% und steht somit für eine ausreichende Datenbasis. Die RuleConfidence liegt bei 51,4%. Somit sagt die Regel aus, dass wenn das rechte Artefakt die Maturity proposed hat, das linke Artefakt die Maturity project_accepted zu 51,4% besitzt. Die RuleLift liegt jedoch nur bei 78%. Somit besteht keine Abhängigkeit zwischen linkem und rechtem Teil der Regel. Diese Regel ist also nicht bedeutsam, also eine schlechte Regel.
Die dritte Regel zeigt Abhängigkeiten zwischen der Relation eines Test Cases mit einem System Requirement, dem Implementierungszustand und der Maturity des System Requirements und der Maturity des Test Cases. Die RelativeItemSetSupport ist mit einem Wert von 12,629% gut. Die RuleConfidence liegt bei 85,7 % und steht somit für eine hohe Richtigkeit der Regel. Diese sagt nämlich aus, dass bei einer Relation zwischen einem Test Case und einem System Requirement, bei der das System Requirement implementiert ist und sich in der Maturity proposed befindet, sich der Test

Case zu 85,7% in der Maturity project_accepted befindet. Die RuleLift liegt bei 130%. Da somit alle drei Bewertungsfaktoren sehr gute Werte haben, handelt es sich hierbei um eine gute Regel.

Consequent	Antecedent	RelativeItemSetSupport(%)	RuleConfidence(%)	RuleLift(%)
High	[CREATED_Q]	12,251	78,2	88,522
ANALYZED_[duration]	[CLOSURE_Q, High]	10,826	55,1	137,1
REALIZED_[duration]	[CLOSURE_Q, High]	14,815	75,4	157,45

Tab. 10: Beispielregeln eines Change Requests für den Regeltyp inhaltliche Erkenntnis

Bei den aufgezeigten Regeln wurde aus Datenschutzgründen die Statusdauer- und Quartalsangabe unkenntlich gemacht.
Die erste Regel zeigt den Zusammenhang zwischen Quartalsangabe und Importance eines Change Requests. Die RelativeItemSetSupport ist mit 12,251 % ausreichend. Die Regel sagt mit einer RuleConfidence von 78,2 % aus, dass ein in einem gegebenen Quartal erstellter Change Request die Importance high zu 78,2 % hat. Jedoch liegt hier die RuleLift unter 100%, wodurch keine Abhängigkeit zwischen beiden Teilen der Regel besteht und die Regel als schlecht eingestuft wird.
Die zweite und dritte Regel haben denselben Antecedent. Der RelativeItemSetSupport liegt bei beiden über 10% und ist somit ausreichend. Ebenso besitzten alle eine RuleConfidence über 50% und eine RuleLift über 100%. Bei Regel zwei tritt die Consequent bei selbem Antecedent zu 55,1 % ein und sagt aus, dass dieser Change Request eine gewisse Dauer im Status accepted war. Die dritte Regel sagt aus, dass sich der Change Request zu 75,4 % mit einer gewissen Dauer im Status realized befand. Beide Regeln sind gut, jedoch ist Regel drei durch eine höhere RuleConfidence aussagekräftiger, da sie eine höhere Richtigkeit besitzt.
Anhand der beiden Regeln kann man sehen, dass ein Antecedent mehrere Consequents mit unterschiedlichem RelativeItemSetSupport, RuleConfidence und RuleLift haben kann.

Consequent	Antecedent	RelativeItemSetSupport(%)	RuleConfidence(%)	RuleLift(%)
left_impl_status_not_implemented	left_discipline_SW	77,393	84,9	101,93

Tab. 11: Beispielregel eines Stakeholder Requirements für den Regeltyp inhaltliche Erkenntnis

Die dritte Regel zeigt eine Abhängigkeit zwischen der Disciplince und dem Implementierungsstatus eines linken Artefakts einer Relation. Regelaussage ist, dass wenn das linke Artefakt die Discipline SW hat, es zu 84,9 % nicht implementiert ist. Es liegt hier eine RuleLift über 100% vor, jedoch knapp über 100%. Darum handelt es sich um eine mittelmäßig gute Regel.

SCHLUSSBETRACHTUNG

ERGEBNIS DER ARBEIT

Aus dieser Arbeit gehen erfolgreich abgeschlossene und für das Data Mining notwendige Projektschritte sowie zahlreich erarbeitete Resultate, insbesondere die Entwicklung von drei Regeltypen, mit dem Analysetool KNIME hervor.
Durch die sorgfältige Analyse der Datengrundlage wurde ein sehr gutes Datenverständnis erzeugt, mit welchem die vorhandenen Daten genauestens dokumentiert wurden. Dazu gehört auch die geprüfte Datenqualität, welche erkannte Unstimmigkeiten im Datensatz aufzeigt. Es ist also nun ein gesamtheitlicher Überblick über den gesamten Datensatz mit dessen Relation und Qualität vorhanden, welcher zum Verständnis einzelner Daten und zur Durchführung von Verbesserungen oder Änderungen im Datensatz genutzt werden kann. Für die Aufbereitung der Daten bestehen wiederverwendbare Workflows aus dem Analysetool KNIME. Ebenso wurde in KNIME das Assoziationsverfahren modelliert und durchgeführt. Für das Einlesen und Aufbereiten sind insgesamt sechs Workflows und für das Assoziationsverfahren fünf Workflows modelliert worden. Hierbei besteht ein großer Nutzen, da diese zukünftig weiter verwendet, erweitert oder auch auf andere Daten angepasst werden können. Zudem geht hier auch die Einführung des Analysetools KNIME in der Abteilung hervor.
Mit dem Assoziationsverfahren wurden zahlreiche Regeln aufgefunden. Darunter fallen Regeln zur Aufdeckung von Schwächen in der Datenqualität, Regeln zur Aufdeckung von Redundanzen in den Daten und Regeln zur Generierung neuer inhaltlicher Erkenntnisse.
Diese Regeln erlauben es, gezielt Verbesserung im Datenbestand und in der Datenverwaltung durchzuführen und neue Zusammenhänge zwischen den Daten zu erkennen. Somit können eine höhere Datenqualität und redundanzfreie Datenbestände erzielt werden. Mit neuen inhaltlichen Erkenntnissen können Stärken und Schwächen des Prozesses erkannt werden. Dies führt dauerhaft zur Optimierung des Softwareentwicklungsprozesses.
Durch die Quantifizierung der erzielten Ergebnisse und Erklärung der Bewertung von Assoziationsregeln können weiterhin Regeln erzeugt, bewertet und genutzt werden.

FAZIT UND AUSBLICK

Mit dieser Arbeit kam es zur ersten Data Mining Anwendung in der Abteilung I B&S RD EEX aus dem Geschäftsfeld Continental Automotive GmbH. Die ausführliche Dokumentation des Vorgehens und der Umsetzung mit zugehörig modellierten Workflows kann als Einführung des Themas für ein neues Aufgabengebiet in der Abteilung angesehen werden.

Es besteht starkes Interesse aus Abteilungssicht dieses Thema weiterzuverfolgen.
Durch Erkennung und Aufzeigen von Datenqualitätsproblemen können zukünftig Unstimmigkeiten im Datensatz reduzieren und so für ein höheres Niveau der Datengrundlage sorgen. Hierbei müssen insbesondere Pflichtfelder und mögliche Ausfülloptionen aller Felder beachtet werden, um fehlerhafte Wertangaben zu vermeiden.
Ebenso können Redundanzen im Datensatz reduziert werden. Dafür muss die Datenverwaltung des Data Warehouses geprüft werden und Möglichkeiten zur Reduzierung von Redundanzen aufgezeigt werden. Eine Möglichkeit ist beispielsweise die Anwendung der dritten Normalform.
Um einen langfristigen Nutzen und eine dauerhafte Verbesserung des Prozesses zu erhalten, sind Regeln, welche inhaltliche Erkenntnisse generieren, unbedingt zu berücksichtigen. Dabei müssen zusammen mit der fachlichen Seite im ersten Schritt die Regeln herausgefiltert werden, welche für den bestehenden Softwareentwicklungsprozess als wertvoll und aussagekräftig gelten. Anhand dieser müssen dann im nächsten Schritt Verbesserungsvorschläge ausgearbeitet und ausformuliert werden. Diese können dann für eine geplante Umsetzung verwendet werden.
Ratsam ist weiterhin eine produktive Einsetzung der Ergebnisse. Da es sich um wiederverwendbare Workflows handelt, sollten diese in Zukunft weiterhin auf wöchentliche Datensätze angewendet werden. Somit können aus den sich bedingt durch den zeitlichen Verlauf ändernden Daten neue Regeln generiert werden oder schon vorhandene Regeln durch häufiges Auftreten an Bedeutung gewinnen.
Ein weiterer möglicher Schritt ist die Anfügung weiterer Datenquellen wie beispielsweise Textdaten aus Projektschritten. Durch die Vergrößerung der Datengrundlage können bereits bestehende Ergebnisse verfeinert oder auch neue Ergebnisse generiert werden.
Für den weiteren Einsatz von Data Mining empfiehlt es sich, zuerst eine Datengrundlage mit sehr hoher Datenqualität zu haben. Denn dies ist die Grundvoraussetzung für einen sicheren Erfolg mit Data Mining, da dieses mit falschen oder fehlenden Werten nicht umgehen kann. Hier taucht das in der Informatik bekannte Problem „Garbage in Garbage out“ auf, was mit „unsinnige Eingaben unsinnige Ausgaben“ übersetzt werden kann. Dies bedeutet, dass bei qualitativ minderwertigen Eingaben lediglich qualitativ minderwertige Ausgaben produziert werden.
Weitere Herausforderungen sind beispielsweise in Bezug auf Datenerfassung, Datenvolumen und Datenmanagement gegeben. Aber ebenso spielen Datenschutz und Sicherheit eine wichtige Rolle.
Neben der Reihe an Herausforderungen bietet das Data Mining aber ein enormes Potential für die Wertschöpfung. Durch immer größer und günstiger werdende Speicherkapazitäten empfiehlt es sich sehr, Informationen aufzuzeichnen, zu speichern und mit Hilfe von Data Mining auszuwerten.

Literaturverzeichnis

[BAVO08]
Bankhofer, Udo; Vogel, Jürgen (2008): Datenanalyse und Statistik. Eine Einführung für Ökonomen im Bachelor. Wiesbaden: Betriebswirtschaftlicher Verlag Dr. Th. Gabler / GWV Fachverlage GmbH Wiesbaden. Online verfügbar unter http://dx.doi.org/10.1007/978-3-8349-9654-1.

[BORG12]
Borgelt, Christian (2012): Frequent item set mining. In: *WIREs Data Mining Knowl Discov* 2012 (6), S. 437–456. DOI: 10.1002/widm.1074.

[DAS16]
Das, Sibanjan (2016): Data Science Using Oracle Data Miner and Oracle R Enterprise. Transform Your Business Systems into an Analytical Powerhouse. Online verfügbar unter http://dx.doi.org/10.1007/978-1-4842-2614-8.

[FAPISM96]
Fayyad, U.Piatetsky-Shapiro, G.Smyth, P. (1996): From data mining to knowledge discovery in databases. In: *AI Magazine 17* 1996, S. 37–54, zuletzt geprüft am 23.06.2017.

[GART17]
Gartner (Hg.) (2017): Gartner IT Glossary - Search Results. Online verfügbar unter http://www.gartner.com/it-glossary/?s=data+mining, zuletzt aktualisiert am 19.06.2017, zuletzt geprüft am 19.06.2017.

[KLEU13]
Kleuker, Stephan (2013): Grundkurs Software-Engineering mit UML. Der pragmatische Weg zu erfolgreichen Softwareprojekten. 3., korr. und erw. Aufl. 2013. Wiesbaden, s.l.: Springer Fachmedien Wiesbaden. Online verfügbar unter http://dx.doi.org/10.1007/978-3-658-00642-6.

[KNIM17]
KNIME.COM AG (Hg.) (2017): KNIME | KNIME Analytics Platform. Online verfügbar unter https://www.knime.org/knime-analytics-platform, zuletzt aktualisiert am 12.06.2017, zuletzt geprüft am 12.06.2017.

[MÜLE13]
Müller, Roland M.; Lenz, Hans-Joachim (2013): Business Intelligence. Berlin Heidelberg: Springer Berlin Heidelberg (eXamen.press). Online verfügbar unter http://dx.doi.org/10.1007/978-3-642-35560-8.

[SQUO17]
Sqoring Technologies (Hg.) (2017): Configuration Guide. 16.2.4. Aufl. Online verfügbar unter http://support.squoring.com/documentation/16.2.4/, zuletzt geprüft am 12.06.2017.

[SQUOoJ]
SQUORING Technologies (Hg.) (oJ): SQUORING Technologies | Valuing and Improving Software and SystemsCapital. Online verfügbar unter http://www.squoring.com/en/, zuletzt aktualisiert am oJ, zuletzt geprüft am 12.06.2017.