

# Dokumentation DMN

Team: Lampert, Reyer

## Inhaltsverzeichnis

Beschreibung des BPMN Prozess- Kundenbewertung .....	3
Beschreibung des DMN Prozess .....	3
DRD .....	3
Decison Tables .....	4
Verwendete Hit-Policies .....	6
Testing .....	6
Zu testende Output-Fälle .....	6
Tests .....	7

## Beschreibung des BPMN Prozess- Kundenbewertung

Die Kundenbewertung einer Bank ist ein entscheidender Faktor bei der Beurteilung des finanziellen Erfolgs des Kunden, da sie eine verlässliche Aussage zum Zahlungsverhalten bzw. zur Kreditwürdigkeit von Privatpersonen macht. Ist die Kundenbewertung eines Kunden gut bzw. sehr gut ist der persönliche Score entsprechend hoch. Ist diese schlecht ist der Score entsprechend niedrig. Der persönliche Score ist bspw. bei der Entscheidung, ob ein Kredit an den Kunden vergeben wird, relevant.

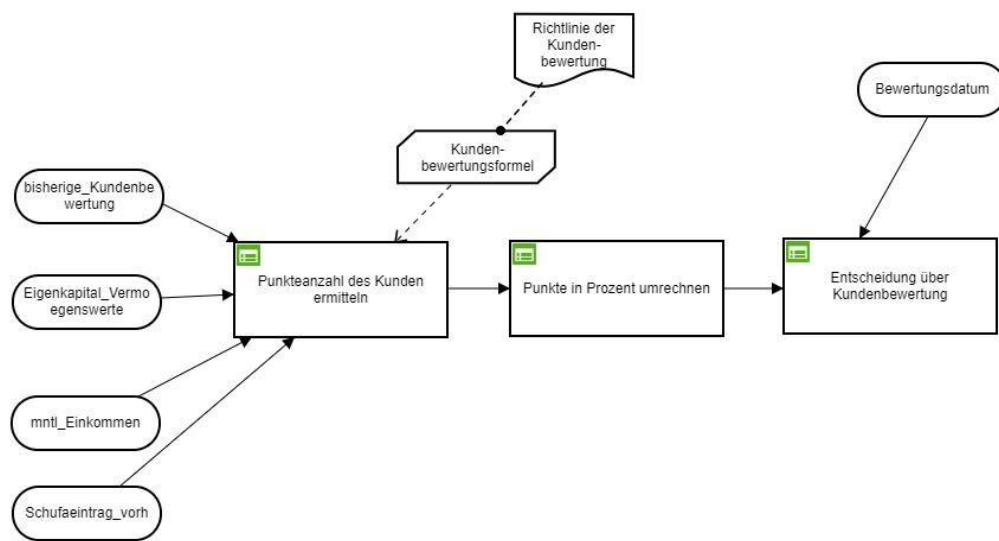
Der folgende BPMN Prozess zeigt den Prozess der Kundenbewertung. Der Prozess startet. Mittels einer User Task werden die Kundendaten eingepflegt, folglich wird mittels einer Business Task, welche auf ein DMN-Modell zugreift, über die Kundenbewertung entschieden. Danach wird über die User Task „Kundenbewertung einsehen“, die Kundenbewertung ausgegeben. Der Prozess ist beendet.



## Beschreibung des DMN Prozess

### DRD

In dem folgenden DRD sind für die Business Task relevante Decisions implementiert. Der Dateninput wird an die erste Decision „Punktzahl der Kunden ermitteln“ kommuniziert. Dabei wird mittels der Richtlinie der Kundenbewertung der Knowledge Source eingebettet. Die Richtlinie der Kundenbewertung im Business Knowledge Model gibt ein mathematisches Modell an, in dem entschieden wird.



## Decision Tables

### Punktzahl des Kunden ermitteln

Betrachtet man nun die Table „Punktzahl des Kunden ermitteln“, stellt man fest, dass die Input Daten der bisherigen Kundenbewertung, Eigenkapital Vermögenswerte, monatliches Einkommen und Schufaeintrag vorhanden eingetragen sind. Dabei wurde diese Decision Table mit der Hit Policy Collect festgelegt. Dabei sind mehrere Regeln schaltbar. Diese wurden in String, Double, Double und Boolean deklariert. Der Gedankenstrich bedeutet somit, dass dies immer gültig ist. Hat ein Kunde eine gute, mittlere, schlechte oder keine bisherige Kundenbewertung, kann dies nun in der Camunda Tasklist durch die Usertask, mittels eines Dropdowns (Enumeration) eingetragen werden. Ist das Eigenkapital größer als 500, liegt das Eigenkapital zwischen 500 bis 1000 (exkludiert), liegt das Eigenkapital zwischen 1000 und 10000 oder ist das Eigenkapital größer gleich 10000 (exkludiert) kann auch dieses ausgewählt werden. Dabei wurden die Werte mit der Edit Number Comparison verknüpft. Auch das monatliche Einkommen sowie ob ein Schufaeintrag vorhanden ist, kann ausgewählt werden.

### Punkteanzahl des Kunden ermitteln

View DRD



Punkteanzahl_ermitteln						
C+	Input +				Output +	Annotation
	bisherige_Kundenbewertung	Eigenkapital_Vermögenswerte	monit_Einkommen	Schufaeintrag_vorh	Punkte_absolut	
	string	double	double	boolean	integer	
1	"gut"	-	-	-	100	Maximal 900 Punkte möglich
2	"mittel"	-	-	-	50	-
3	"schlecht"	-	-	-	-100	-
4	"keine"	-	-	-	0	-
5	-	<500	-	-	0	-
6	-	[500..1000[	-	-	50	-
7	-	[1000..10000[	-	-	100	-
8	-	>=10000	-	-	300	-
9	-	-	<1600	-	0	-
10	-	-	[1600..2000[	-	200	-
11	-	-	[2000..4000[	-	300	-
12	-	-	>=4000	-	500	-
13	-	-	-	false	0	-
14	-	-	-	true	-10000	-
+	-	-	-	-	-	-

### Punkte zu Prozent umrechnen

Die nachfolgende Decision Table „Punkte zu Prozent umrechnen“ wird betrachtet. Die Punkte in Prozent (Output) werden mittels double deklariert und rechnen mittels der Formel  $(\text{Punkte\_absolut} / 900) * 100$ .

### Punkte in Prozent umrechnen

View DRD



Punkte_zu_Prozent			
U	Input +		Output +
	Punkte_absolut		Punkte_in_Prozent
	integer		double
1	-	-	$(\text{Punkte\_absolut}/900)*100$
+	-	-	-

### Entscheidung über Kundenbewertung

Somit erfolgt in der Entscheidung über die Kundenbewertung (Unique) der Input aus Punkte in Prozent (double) und dem Bewertungsdatum -(date).

## Entscheidung über Kundenbewertung

View DRD



Entscheidung_Kundenbewertung				
U	Input +		Output +	Annotation
	Punkte_in_Prozent	Bewertungsdatum	Kundenbewertung_Ergebnis	
	double	date	string	
1	>=50	[date and time("2019-01-01T00:00:00")..date and time("2020-01-01T00:00:00")]	"Guter Kunde"	-
2	<50	[date and time("2019-01-01T00:00:00")..date and time("2020-01-01T00:00:00")]	"Schlechter Kunde"	-
3	-	not([date and time("2019-01-01T00:00:00")..date and time("2020-01-01T00:00:00")])	"k.A. EntsMod nicht aktuell"	-
+	-	-	-	-

## Verwendete Hit-Policies

Die Decision Table „Punktzahl des Kunden ermitteln“ verwendet die Hit Policy COLLECT zusammen mit der Aggregation SUM. D.h., mehrere Decision Rules können schalten und die Output-Werte der einzelnen Decision Rules werden zusammenaddiert.

C+	Hit Policy: COLLECT
	Aggregation: SUM
1	"gut"

Die Decision Tables „Punkte\_zu\_Prozent“ sowie „Entscheidung über Kundenbewertung“ verwenden die Hit Policy UNIQUE. D.h. genau eine Decision Rule schaltet und die Menge der Decision Rules deckt alle möglichen Fälle ab.

Punkte_zu_Prozent	
U	Hit Policy: UNIQUE
	integer
1	-
+	-

## Testing

### Zu testende Output-Fälle

Es gibt 4 mögliche Output-Fälle, in denen der Prozess enden kann.

Der Score liegt über 50 % oder ist gleich 50 %:

- I. Guter Kunde, Datum korrekt
- II. Guter Kunde, Datum nicht korrekt

Der Score liegt unter 50 %:

- III. Schlechter Kunde, Datum korrekt
- IV. Schlechter Kunde, Datum nicht korrekt.

Erklärung:

Es gibt in der Decision Table „Punktzahl des Kunden ermitteln“ maximal 900 Punkte zu vergeben. Davon müssen mindestens 450 erreicht werden, damit der Kunde das Attribut „Guter Kunde“ erhält. Das Entscheidungsmodell (DRD) bzgl. der Kundenbewertung soll nur für das Jahr 2019 gültig sein. Wenn also ein Datum außerhalb von 2019 verwendet wird, ist die Kundenbewertung nicht gültig und der Prozess endet mit der Ausgabe „k.A. EntsMod nicht aktuell“.

## Tests

Exemplarisch werden die 4 oben beschriebenen Fälle in Camunda BPM getestet.

**I: Der Kunde erhält eine Punktzahl von 550 Punkten ( $\geq 50\%$ ), das Datum ist innerhalb 2019. Erwartetes Ergebnis: „Guter Kunde“**

### Daten des Kunden eingeben

kundenbewertung

 Set follow-up date

 Set due date

 Add groups

 Demo Demo ×

Form

History

Diagram

Description

bisherige\_Kundenbewertung

Eigenkapital\_Vermögenswerte  z.B.: 1000.0

mntl\_Einkommen  z.B.: 2000.0

Schufaeintrag\_vorh ☐

Bewertungsdatum  z.B.: 2019-01-01T00:00:00

Save

Complete


-> Es ergeben sich 550 Punkte für den Kunden.

Attribut	Attributwert	Punkte
bisherige_Kundenbewertung	mittel	50
Eigenkapital_Vermoegenswerte	10000	300
mntl_Einkommen	1600	200
Schufaeintrag_vorh	false	0
		Gesamtpunktzahl: 550

## Kunden-bewertung einsehen

kundenbewertung

 Set follow-up date

 Set due date

 Add groups

 Demo Demo ×

Form

History

Diagram

Description

Kundenbewertung

Guter Kunde

Save

Complete

-> Die Kundenbewertung ergibt „Guter Kunde“.

**II: Der Kunde erhält eine Punktzahl von 550 Punkten ( $\geq 50\%$ ), das Datum ist außerhalb 2019. Erwartetes Ergebnis: „k.A. EntsMod nicht aktuell“**



# Daten des Kunden eingeben

kundenbewertung

 Set follow-up date

 Set due date

 Add groups

 Demo Demo ×

Form

History

Diagram

Description

bisherige\_Kundenbewertung mittel ▾

Eigenkapital\_Vermögenswerte 10000 z.B.: 1000.0

mntl\_Einkommen 1600 z.B.: 2000.0

Schufaeintrag\_vorh ☐

Bewertungsdatum 2020-01-01T00:00:00 z.B.: 2019-01-01T00:00:00

Save

Complete

-> Es ergeben sich wieder 550 Punkte für den Kunden, aber das Bewertungsdatum liegt außerhalb von 2019.

# Kunden-bewertung einsehen

kundenbewertung

 Set follow-up date

 Set due date

 Add groups

 Demo Demo ×

Form

History

Diagram

Description

Kundenbewertung k.A. EntsMod nicht aktuell

Save

Complete

-> Die Kundenbewertung ergibt „k.A. EntsMod nicht aktuell“.

**III: Der Kunde erhält eine Punktzahl von 300 Punkten (<50%), das Datum ist innerhalb 2019. Erwartetes Ergebnis: „Schlechter Kunde“**

# Daten des Kunden eingeben

kundenbewertung

 Set follow-up date

 Set due date

 Add groups

 Demo Demo 

Form

History

Diagram

Description

bisherige\_Kundenbewertung

Eigenkapital\_Vermögenswerte  z.B.: 1000.0

mntl\_Einkommen  z.B.: 2000.0

Schufaeintrag\_vorh ☐

Bewertungsdatum  z.B.: 2019-01-01T00:00:00

Save

Complete

-> Es ergeben sich 300 Punkte für den Kunden.

Attribut	Attributwert	Punkte
bisherige_Kundenbewertung	schlecht	-100
Eigenkapital_Vermögenswerte	1000	100
mntl_Einkommen	2000	300
Schufaeintrag_vorh	false	0
		Gesamtpunktzahl: 300

## Kunden-bewertung einsehen

kundenbewertung

 Set follow-up date

 Set due date

 Add groups

 Demo Demo 

Form

History

Diagram

Description

Kundenbewertung

Save

Complete


-> Die Kundenbewertung ergibt „Schlechter Kunde“.

**IV: Der Kunde erhält eine Punktzahl von 400 Punkten (<50%), das Datum ist außerhalb 2019 Erwartetes Ergebnis: „k.A. EntsMod nicht aktuell“**

# Daten des Kunden eingeben

kundenbewertung

 Set follow-up date

 Set due date

 Add groups

 Demo Demo ×

Form

History

Diagram

Description

bisherige\_Kundenbewertung

Eigenkapital\_Vermögenswerte  z.B.: 1000.0

mntl\_Einkommen  z.B.: 2000.0

Schufaeintrag\_vorh ☐

Bewertungsdatum  z.B.: 2019-01-01T00:00:00

Save

Complete

-> Es ergeben sich wieder 300 Punkte für den Kunden, aber das Bewertungsdatum liegt außerhalb von 2019.

# Kunden-bewertung einsehen

kundenbewertung

 Set follow-up date

 Set due date

 Add groups

 Demo Demo ×

Form

History

Diagram

Description

Kundenbewertung

Save

Complete

-> Die Kundenbewertung ergibt „k.A. EntsMod nicht aktuell“.