

**BPMN Konventionshandbuch für die Lehrveranstaltung  
Geschäftsprozesse**

## **Zusammenfassung**

Das Modul *Geschäftsprozesse*<sup>1</sup> der TU Berlin lehrt ausführlich die Modellierung von Geschäftsprozessen mit dem Standard *Business Process Model and Notation*<sup>2</sup> (BPMN). In diesem Handbuch werden die Konventionen der Lehrveranstaltung zur Modellierung von Geschäftsprozessen erläutert, deren Einhaltung verpflichtend für die Studenten ist.

Im Gegensatz zu anderen Konventionen ist das Ziel nicht die Einschränkung der Symbolpalette zur Steigerung der schnelleren Verständlichkeit eines Diagramms, sondern die Verbesserung der Übersichtlichkeit bei Benutzung der vollständigen BPMN Symbolpalette. Dazu werden einige Möglichkeiten der BPMN eingeschränkt. Des Weiteren sind Regeln enthalten, die nicht dem Standard widersprechen, sondern als Orientierung zur Vermeidung von häufig gemachten Anfänger-Fehlern dienen.

---

<sup>1</sup><http://www.snet.tu-berlin.de/gepit>

<sup>2</sup><http://www.omg.org/spec/BPMN/Current/>

# Inhaltsverzeichnis

<b>Synonyme</b>	<b>3</b>
<b>1 Einleitung</b>	<b>4</b>
1.1 Überblick . . . . .	4
1.2 Weiterführende Ressourcen . . . . .	5
<b>2 Generelle Konventionen</b>	<b>6</b>
2.1 Sprache . . . . .	6
2.1.1 Abkürzungen . . . . .	6
2.1.2 Beschreibungskonventionen . . . . .	6
2.2 Darstellungskonventionen . . . . .	7
2.2.1 Modellierungsrichtung . . . . .	7
2.2.2 Formatierung . . . . .	7
2.2.3 Farben . . . . .	7
<b>3 Modellierungskonventionen BPMN</b>	<b>9</b>
3.1 Aktivitäten . . . . .	9
3.1.1 Jede Aktivität hat genau einen eingehenden und einen ausgehenden Sequenzfluss . . . . .	9
3.1.2 Bedingte Sequenzflüsse nicht verwenden . . . . .	10
3.1.3 Aktivitäten normalerweise ohne eingehende Nachrichtenflüsse . . . . .	10
3.1.4 Eingebettete Teilprozesse mit einem unbestimmten Starterereignis starten . . . . .	11
3.1.5 Durchlauf-Bedingungen von Schleifen als Anmerkungen an die Aktivität anheften . . . . .	11
3.1.6 Durchläufe von Mehrfachaktivitäten durch eine vorhergehende Aktivität bestimmen . . . . .	12
3.1.7 Typisierung von Aufgaben nicht notwendig . . . . .	12
3.1.8 Transaktionsunterprozesse mit Abbruchereignissen . . . . .	13

3.2	Ereignisse . . . . .	14
3.2.1	Ereignisse stets beschriften . . . . .	14
3.2.2	Auslösende Nachrichtenereignisse nicht verwenden . . . . .	14
3.2.3	Empfangende Nachrichtenereignisse normalerweise gegenüber Aktivitäten bevorzugen . .	15
3.2.4	Startereignisse . . . . .	16
3.2.5	Zwischenereignisse . . . . .	17
3.2.6	Endereignisse . . . . .	18
3.3	Gateways . . . . .	19
3.3.1	Genau ein ausgehender bzw. eingehender Sequenzfluss bei mehreren eingehenden bzw. ausgehenden Sequenzflüssen . . . . .	19
3.3.2	Aktivität vor einem datenbasiertem Gateway . . . . .	19
3.3.3	Bedingungen an den ausgehenden Sequenzflüssen eines datenbasierten Gateways . . . . .	20
3.3.4	Komplexe Verzweigungen durch die selben Gateways andersherum wieder zusammenführen	20
3.3.5	Verzweigte Sequenzflüsse auf derselben horizontalen Linie zusammenführen . . . . .	21
3.3.6	Komplexe Gateways vermeiden . . . . .	21
3.3.7	Verwendung von instanziiierenden exklusiven ereignisbasierten Gateways verboten . . . .	22
<b>4</b>	<b>Modellierungskonventionen CMMN</b>	<b>23</b>
4.1	Allgemein . . . . .	23
4.1.1	Aufgaben stets typisieren . . . . .	23
4.1.2	Manuelle Ausführung von Aufgaben . . . . .	23
4.1.3	Beendigung einer Stage mit Autocomplete . . . . .	24
4.1.4	Beendigung eines Falls . . . . .	24
4.2	Wächter . . . . .	24
4.2.1	Verbindungen beschriften mit Lebenszyklus-Events . . . . .	24
4.2.2	Bedingung eines Eingangs-Wächters mit Textanmerkung beschreiben . . . . .	25
4.2.3	Bedingung eines Ausgangs-Wächters mit Textanmerkung beschreiben . . . . .	25

# Synonyme

**Aktivität** es gibt zwei Ausprägungen: Aufgaben und Teilprozesse.

**Aufgabe** Synonym: Task.

**Case** Synonym: Fall.

**Ereignis** Synonym: Event.

**Event** Synonym: Ereignis.

**Fall** Synonym: Case.

**Sentry** Synonym: Wächter.

**Task** Synonym: Aufgabe.

**Teilprozess** Synonym: Unterprozess.

**unbestimmtes** Ereignis, Synonym: untypisiertes E.

**Unterprozess** Synonym: Teilprozess.

**untypisiertes** Ereignis, Synonym: unbestimmtes E.

**Wächter** Synonym: Sentry.

# Kapitel 1

## Einleitung

Dieses Dokument vereinheitlicht die Anwendung des Standards *Business Process Model And Notation (BPMN)*<sup>1</sup> für die grafische Darstellung (Notation) von Prozessen in der Lehrveranstaltung *Geschäftsprozesse*<sup>2</sup>. Die Einhaltung dieser Konventionen ist für die Studenten verpflichtend.

### 1.1 Überblick

Die Lehrveranstaltung Geschäftsprozesse behandelt die vollständige Symbolpalette des BPMN-Standards. Ziel dieser Modellierungskonvention ist es, die Übersichtlichkeit der resultierenden Diagramme zu steigern, indem die verschiedenen Verwendungsmöglichkeiten bewusst eingeschränkt werden. Im Gegensatz zu anderen Konventionen wird die Symbolpalette dafür nicht verkleinert.

Teilweise sind auch Regeln enthalten, die nicht dem Standard widersprechen. Diese sind zur Vermeidung von typischen Anfängerfehlern gedacht, die wir in den Lehrveranstaltungen beobachten konnten.

Ein einheitlicher BPMN-Modellierungsstil führt dazu, dass sich die Geschäftsprozessdiagramme und Geschäftsprozessdokumentationen strukturell und semantisch ähnlich werden. Dadurch wird die Lesbarkeit erhöht und das gemeinsame Verständnis gefördert.

Diese Modellierungskonventionen legen besonderen Wert auf folgende Grundsätze:

- Geschäftsprozessmodelle sollen einfach lesbar sein.
- Die Verwendung von BPMN-Symbolen und Diagrammen soll so erklärt sein, dass sie auch von Modellierungsanfängern gemäß diesen Konventionen richtig eingesetzt werden können.
- Typische, beobachtete Modellierungsfehler sollen vermieden werden.
- Anhand von Beispielen sollen diese Punkte verständlich und allgemein anwendbar gemacht werden.

---

<sup>1</sup><http://www.omg.org/spec/BPMN/Current/>

<sup>2</sup><http://www.snet.tu-berlin.de/gepit>

## 1.2 Weiterführende Ressourcen

Diese Konvention baut auf der BPMN-Modellierungskonvention für die öffentliche Verwaltung der Schweiz (eCH-0158, veröffentlicht am 04.09.2014) auf und übernimmt einige Abschnitte. Da das Ziel jedoch anders ist (Verständlichkeit vs. Vollständigkeit), wurden viele Abschnitte verändert, entfernt oder erweitert.

Weitere einbezogene Quellen sind:

- Praxisbuch BPMN, 5. Auflage, J. Freund und B. Rücker, 2017
- Fundamentals of Business Process Management, M. Dumas, M. La Rosa, J. Mendling und H. Reijers, 2013
- <http://brsilver.com/the-rules-of-bpmn/>
- <https://camunda.org/bpmn/examples/#bpmn-modeling-styles>
- <https://www.kurze-prozesse.de/bpmn-auf-deutsch/>

# Kapitel 2

## Generelle Konventionen

### 2.1 Sprache

Manchmal werden unterschiedliche Wörter mit derselben oder ähnlicher Bedeutung (Synonyme) verwendet (beispielsweise statt „prüfen“ die Verben „begutachten“, „kontrollieren“, „mustern“, etc.). Dies kann zu Fehlinterpretationen oder unterschiedlichen Benennungen derselben Elemente führen (beispielsweise „Rechnung prüfen“, „Rechnung kontrollieren“). Deshalb empfiehlt es sich, den selben Begriff durchgängig in einem Diagramm zu verwenden.

Die Nutzung festgelegter Terminologien (Fachbegriffe) erhöht die allgemeine Verständlichkeit, außerdem ist eine Übersetzung in eine andere Sprache deutlich einfacher, da die Begriffe in den Katalogen festgelegte Übersetzungen besitzen. Über Ontologien sind Begriffe sprachwissenschaftlich in ein lexikalisch-semantisches Begriffsnetz von Oberbegriffen, Unterbegriffen und Synonyme eingeordnet, im deutschsprachigen Raum beispielsweise im GermaNet<sup>1</sup> oder ansatzweise im Wiktionary<sup>2</sup>.

#### 2.1.1 Abkürzungen

Allgemein übliche Abkürzungen und Fachbegriffe können verwendet werden.

#### 2.1.2 Beschreibungskonventionen

**Pools** Pools werden in der Regel mit dem Unternehmensnamen, Organisationseinheiten, dem Prozessnamen oder dem Namen anderer Prozessteilnehmer bezeichnet.

**Swimlanes** Lanes werden meist mit Rollenbezeichnungen im Singular beschriftet. Eine Lane darf nicht gleich wie ein Pool bezeichnet werden.

**Tasks** Atomare Aufgaben sind eine nicht weiter unterteilbare Aktivität. Sie wird durch ein vorangestelltes substantiviertes Objekt und ein infinitives Verb bezeichnet, z.B. „Rechnung versenden“. Sie müssen atomar sein, d.h. es dürfen nicht mehrere Aktivitäten in einer Aufgabe mit „und“ verbunden werden.

---

<sup>1</sup>[www.sfs.uni-tuebingen.de/GermaNet](http://www.sfs.uni-tuebingen.de/GermaNet)

<sup>2</sup>[de.wiktionary.org](http://de.wiktionary.org)



**Unterprozesse** Teilprozesse bestehen aus mehreren Teilschritten. Sie werden durch die substantivierte Verknüpfung eines Objektes und der Tätigkeit bezeichnet, z.B. „Rechnungsprüfung“.

**Ereignisse** Events beziehen sich immer auf etwas, das bereits passiert ist (unabhängig von der Kategorisierung „eintretend“ und „auslösend“). Sie werden deshalb mit substantiviertem Objekt und passiven Verb beschrieben, z.B. „Rechnung versandt“ oder „Essen eingetroffen“. Ausnahme: Zeitereignis.

## 2.2 Darstellungskonventionen

### 2.2.1 Modellierungsrichtung

Entsprechend dem Sequenzfluss sind die Modelle möglichst von links nach rechts zu modellieren. Rückflüsse (Ausnahme: explizite Rückführungen für Schleifen), Überlagerungen und Überkreuzungen sind bei Nachrichten- und Sequenzflüssen zu vermeiden.

BPMN-Diagramme können mehrere Pools als Teil einer Choreographie enthalten. Alle Pools sind vertikal untereinander anzuordnen. Zugeklappte Pools enthalten mindestens einen eingehenden oder ausgehenden Nachrichtenfluss.

### 2.2.2 Formatierung

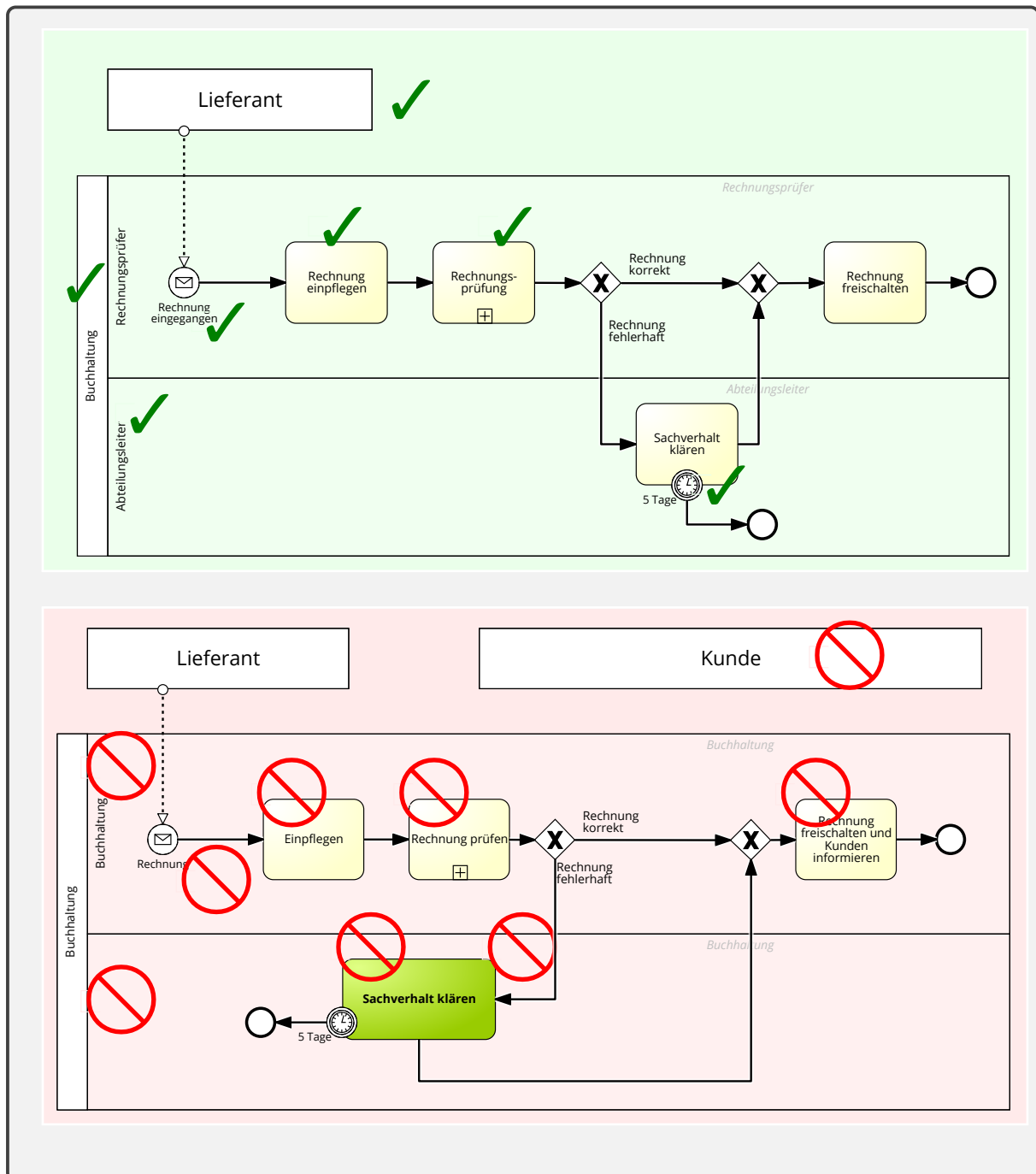
Beschriftungen und Bezeichnungen von Elementen sollen nach Möglichkeit keine Formatierungen (Kursiv, Fett, etc.) enthalten.

Sämtliche BPMN-Elemente sind pro Elementtyp in derselben Größe und Form zu halten (Beispiel: Alle Tätigkeiten sind gleich groß).

Besitzt ein Diagramm mehrere Pools, brauchen diese nicht zwangsläufig über die selbe Diagrammbreite dargestellt werden, sondern können unterschiedliche Größen besitzen. Die Höhe des Pools richtet sich nach dessen Inhalt.

### 2.2.3 Farben

Grundsätzlich sind Farben zu vermeiden. Wenn es der Lesbarkeit dient, können den Elementtypen (z.B. alle Aufgaben mit einem gelben Hintergrund) oder auch einzelnen Elementen nach einem klaren Farbkonzept zurückhaltend entsprechende Farben zugewiesen werden, wie das standardmäßig beispielsweise im Signavio-Editor erfolgt.



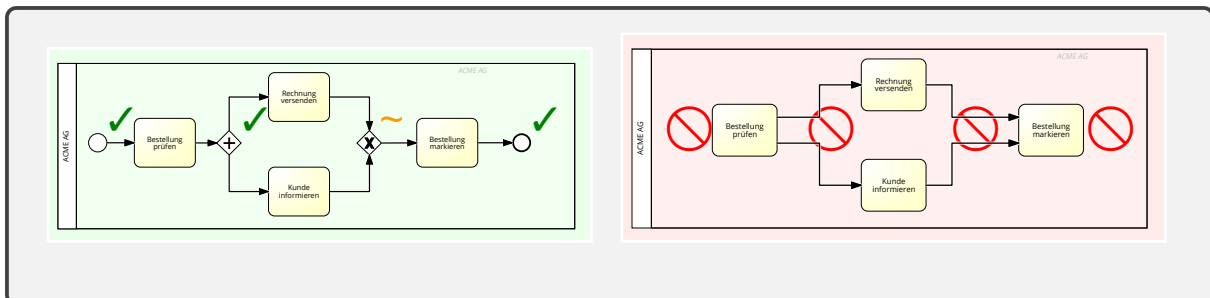
# Kapitel 3

## Modellierungskonventionen BPMN

### 3.1 Aktivitäten

#### 3.1.1 Jede Aktivität hat genau einen eingehenden und einen ausgehenden Sequenzfluss

Die BPMN-Spezifikation erlaubt, dass Aktivitäten keinen, einen oder mehrere eingehende und ausgehende Sequenzflüsse besitzen können (Abschnitt 10.3, Sequence Flow Connections). In dieser Konvention wird festgelegt, dass jegliche Aktivitäten (Aufgaben und Unterprozesse) genau einen eingehenden und einen ausgehenden Sequenzfluss besitzen.



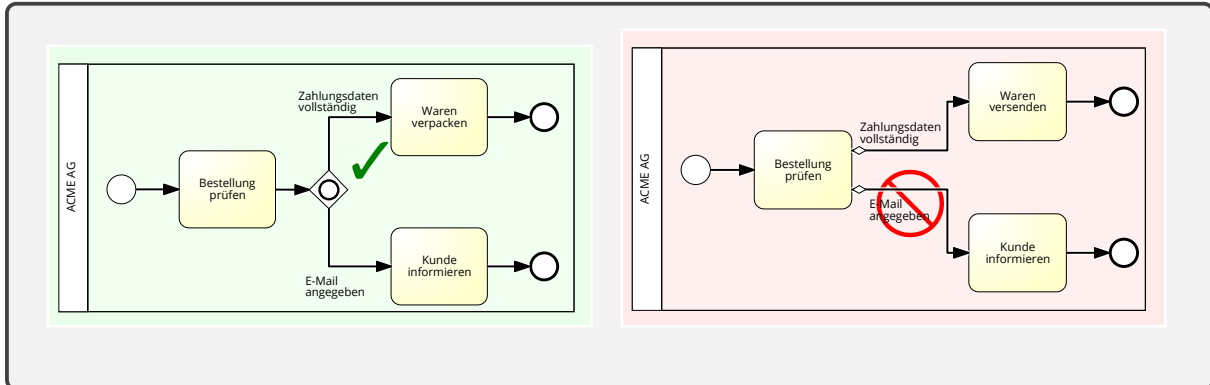
**Hinweis:** in der Darstellung des erlaubten Falles sind das öffnende und schließende Gateway unterschiedlich. Dies muss in den meisten Fällen vermieden werden (siehe 3.3.4). Es ist so dargestellt, weil es dadurch am ehesten der Semantik des nicht erlaubten Falles entspricht.

**Erklärung der nicht erlaubten Modellierung:** Im Fall keines eingehenden Sequenzflusses würde diese Aktivität bei Prozessinstanziierung gestartet werden. Im Fall mehrerer ausgehender Sequenzflüsse wird jeder Sequenzfluss nach Beendigung der Aktivität mit einem Token belegt. Die Semantik würde demnach einem Parallelen Gateway entsprechen. Im Fall mehrerer eingehender Sequenzflüsse würde jedes ankommende Token die Aktivität starten, was zu einem „unkontrollierten“ Ablauf führen würde. Die Semantik entspricht also einem XOR-Gateway, die Sequenzflüsse sind aber nicht synchronisiert. Der Fall keines ausgehenden Sequenzflusses entspricht implizit einem unbestimmten Endereignis, d.h. das Token würde nach Beendigung der Aktivität gelöscht werden.

Diese unterschiedliche Semantik der eingehenden und ausgehenden Sequenzflüsse führt oft zu Verwirrungen und ist in dieser Konvention nicht erlaubt. Durch die explizite Darstellung mittels Startereignis, Endereignis Exklusiven und Parallelen Gateway wird die Semantik bewusst modelliert.

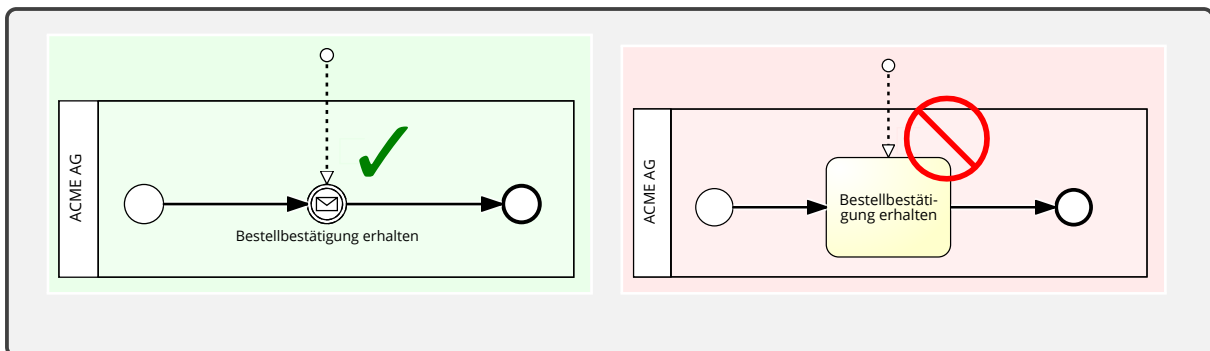
### 3.1.2 Bedingte Sequenzflüsse nicht verwenden

Die BPMN-Spezifikation erlaubt aus Aktivitäten spezielle Sequenzflüsse, die mit einer Raute beginnen, mit Bedingungen zu belegen. Es ist semantisch äquivalent zum inklusiven Gateway. Diese Darstellung der Bedingung an einem Sequenzfluss ist häufig missverständlich und wird in dieser Konvention deshalb untersagt.



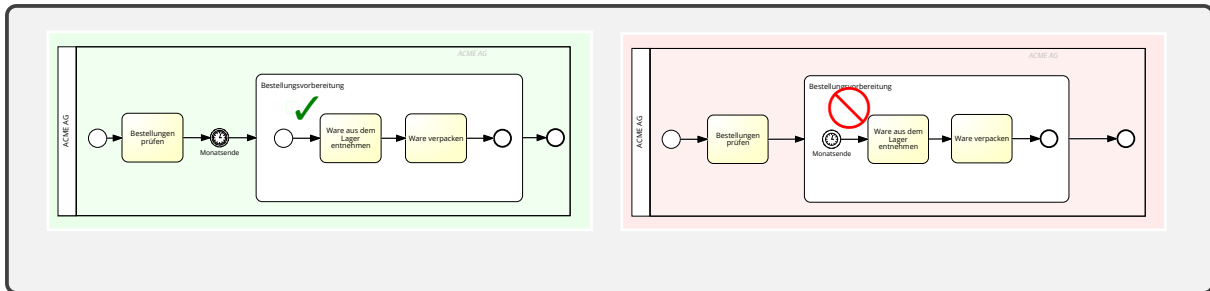
### 3.1.3 Aktivitäten normalerweise ohne eingehende Nachrichtenflüsse

Für eintreffende Nachrichten ist ein empfangendes Nachrichtenereignis zu benutzen. Nur in Ausnahmefällen darf ein eingehender Nachrichtenfluss an eine Aktivität angeheftet sein. Für Ausnahmen, siehe 3.2.3.



### 3.1.4 Eingebettete Teilprozesse mit einem unbestimmten Startereignis starten

Ein eingebetteter Unterprozess muss immer mit einem untypisiertem Ereignis gestartet werden.



**Erklärung:** Eingebettete Teilprozesse sind Teil des Elternprozesses und können nicht selbstständig existieren. Der Elternprozess startet immer den Unterprozess – dies geschieht durch Auslösen des unbestimmten Start-Ereignisses.

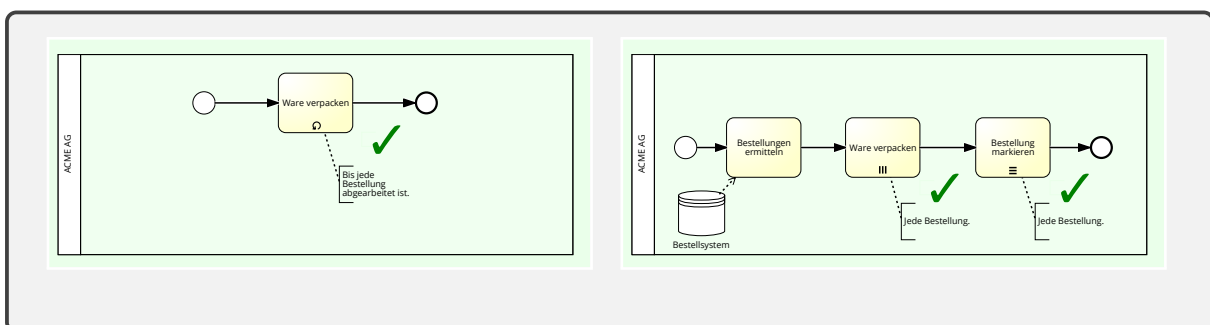
**Hinweis:** Ein wiederverwendbarer Teilprozess ruft einen global verfügbaren Prozess auf (mit einer Aufruf-Aktivität). Der aufgerufene Prozess muss somit mindestens ein untypisiertes Startereignis besitzen, damit dieser ausgeführt werden kann. Er kann jedoch außerdem noch weitere, typisierte Startereignisse besitzen, wenn der Prozess auch durch andere Ereignisse instanziiert werden soll.

### 3.1.5 Durchlauf-Bedingungen von Schleifen als Anmerkungen an die Aktivität anheften

Wenn eine Aktivität mit einer Schleifen-Markierung versehen ist, dann ist die Bedingung für den Durchlauf der Schleife als Text-Anmerkung an die Aktivität zu heften.

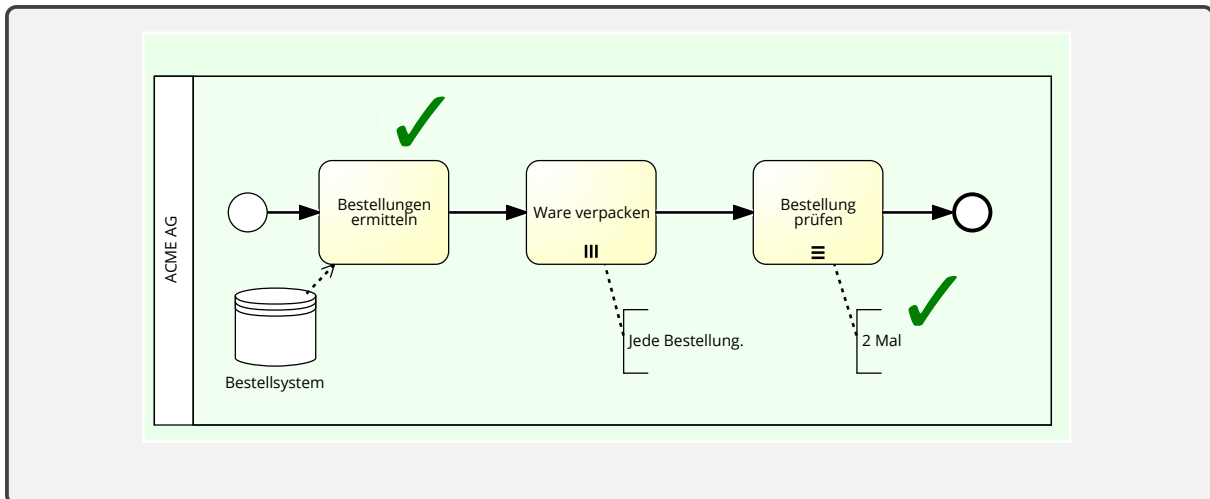
Beim normalen Schleifensymbol muss die Textanmerkung zu einer booleschen Bedingung führen, wobei nur bei true die Schleife weiter iteriert wird. Beispiel: „Bis Event X stattgefunden hat“ bedeutet, dass die Schleife bei einer Evaluation zu „true“ evaluiert, solange ein externes „Event X“ noch nicht stattgefunden hat.

Bei Mehrfachausführungs-Markierungen (Parallel und Sequentiell) ist die Anzahl der Wiederholungen bereits vor Beginn der Aktivität bekannt. Die Textanmerkung muss dies widerspiegeln.



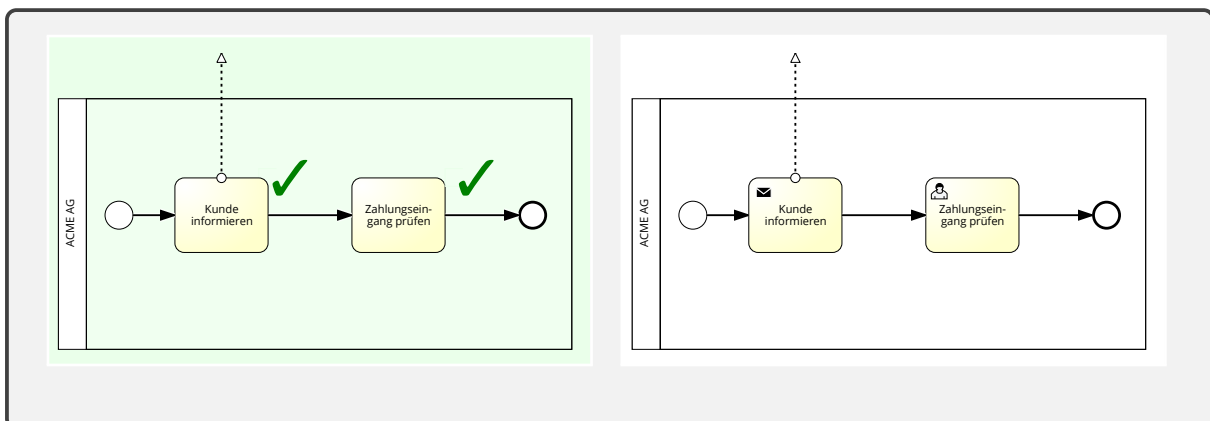
### 3.1.6 Durchläufe von Mehrfachaktivitäten durch eine vorhergehende Aktivität bestimmen

Bei Mehrfachaktivitäten steht die Anzahl der Durchführungen vor Beginn der Aktivität fest. Diese kann zur Modellierungszeit festgelegt oder dynamisch zur Prozesslaufzeit ermittelt werden. Wenn sie dynamisch bestimmt wird, dann müssen die Daten vorher im Prozess durch eine Aktivität erfasst und verfügbar sein.



### 3.1.7 Typisierung von Aufgaben nicht notwendig

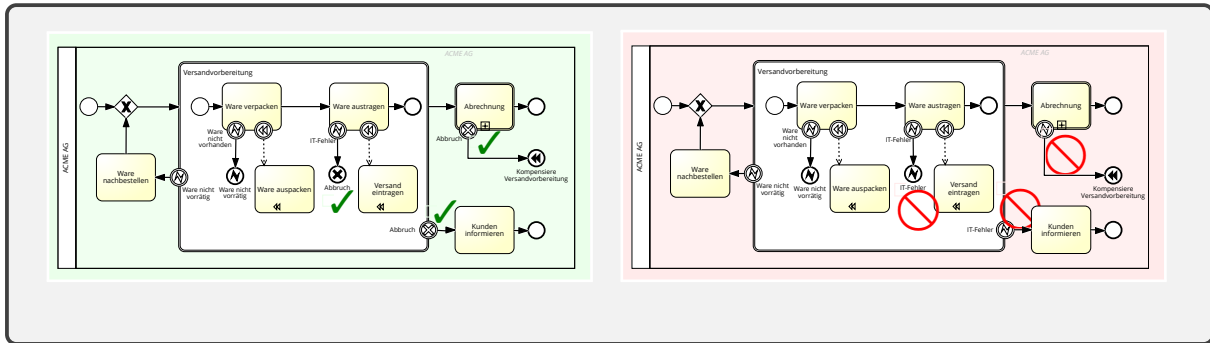
Aufgaben können typisiert werden, was insbesondere relevant für die technische Umsetzung von Prozessmodellen ist. Für die Prozessmodelle der Lehrveranstaltung Geschäftsprozesse ist es nicht wichtig die Aufgaben zu typisieren – es wird bei richtiger Verwendung aber auch nicht negativ angerechnet.



### 3.1.8 Transaktionsunterprozesse mit Abbruchereignissen

Transaktionsteilprozesse haben eine doppelte Umrandung und können auf unterschiedliche Arten enden (erfolgreich, fehlerhaft, abbrechend). Sie müssen mindestens ein angeheftetes Abbruch-Zwischenereignis besitzen.

Bei der Modellierung eines aufgeklappten Transaktionsteilprozesses muss dieser mindestens ein Abbruch-Endereignis enthalten.



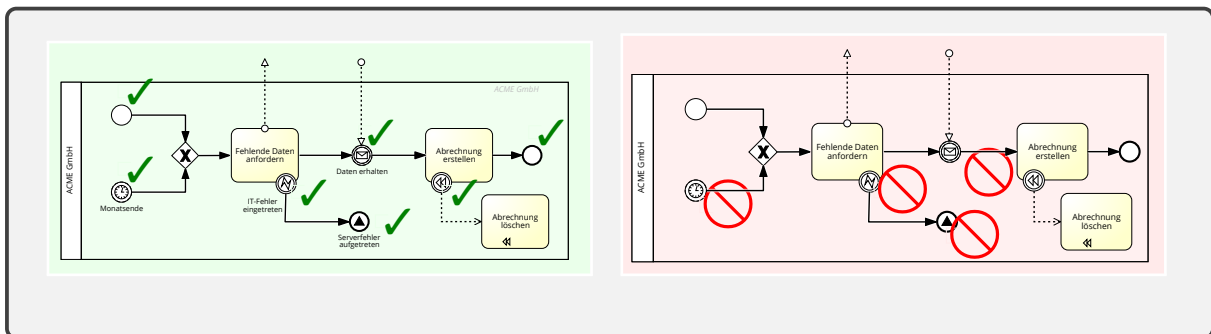
**Hinweis:** Beim Auslösen eines Abbruch-Endereignisses wird der Prozessfluss nicht sofort an das angeheftete Abbruchereignis übergeben – stattdessen werden zuerst alle Kompensationen von erfolgreich durchgeführten Aktivitäten in umgekehrter Reihenfolge des Sequenzflusses durchgeführt. Erst danach erhält das angeheftete Ereignis das Token zum Fortführen des Prozesses.

## 3.2 Ereignisse

### 3.2.1 Ereignisse stets beschriften

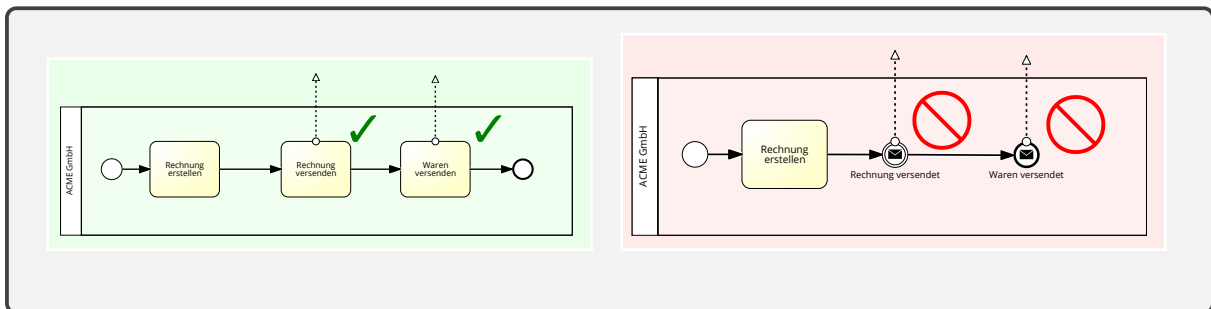
Jedes modellierte Ereignis muss eine aussagekräftige Beschriftung besitzen. Auch alle Nachrichtenergebnisse müssen der Vollständigkeit halber beschriftet werden, selbst wenn man diese implizit durch den anderen Prozesssteilnehmer herleiten könnte.

**Ausnahme 1:** untypisierte Start- und Endereignisse, sowie angeheftete eintretende Kompensationsereignisse brauchen nicht beschriftet werden.



### 3.2.2 Auslösende Nachrichtenergebnisse nicht verwenden

Da das Senden von direkten Nachrichten an andere Prozesssteilnehmer (Pools) eine Aktivität impliziert, dürfen keine auslösenden Nachrichtenergebnisse verwendet werden. Stattdessen ist eine Aufgabe mit ausgehenden Nachrichtenfluss zu modellieren und die Beschreibungskonvention für Aufgaben einzuhalten.



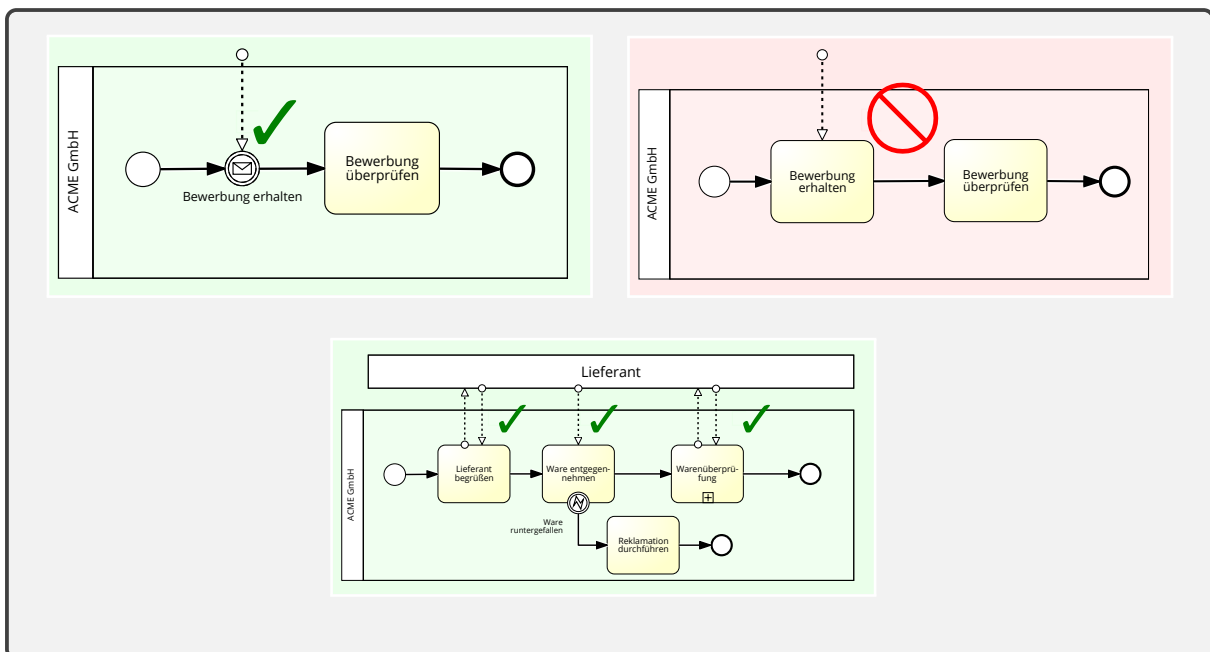


### 3.2.3 Empfangende Nachrichtenereignisse normalerweise gegenüber Aktivitäten bevorzugen

Der Nachrichtenempfang ist meistens nur sehr kurz, weshalb hierfür normalerweise keine Für den Empfang von Nachrichten sind empfangende Nachrichtenereignisse zu nutzen, da Ereignisse einen Zustand darstellen, der bereits passiert ist.

Nur in folgenden Ausnahmefällen dürfen Aktivitäten mit eingehenden Nachrichtenflüssen verwendet werden:

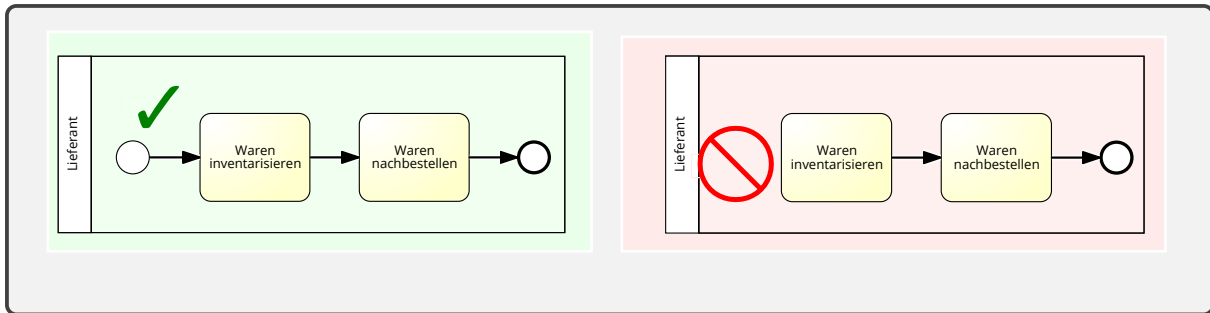
- Es soll explizit modelliert werden, dass der Nachrichtenempfang eine Aktivität ist, die mit Hilfe eines angehefteten Ereignisses abgebrochen werden kann.
- Zum Modellieren einer bilateralen Unterhaltung, z.B. bei einer Begrüßung.
- Wenn ein collabierter (zugeklappter) Teilprozess eine oder mehrere empfangende Nachrichten beinhaltet.



### 3.2.4 Startereignisse

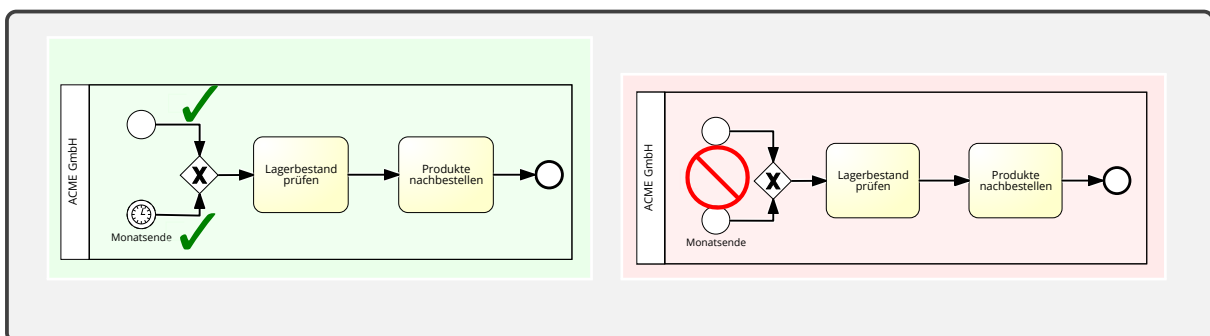
#### 3.2.4.1 Mindestens ein Startereignis pro Pool

Jeder Pool muss mindestens ein Startereignis besitzen. Das implizite Starten von Aktivitäten, indem kein eingehender Sequenzfluss modelliert wird, ist verboten. Für eine Erklärung, siehe 3.1.1.



#### 3.2.4.2 Maximal ein unbestimmtes Startereignis pro Pool

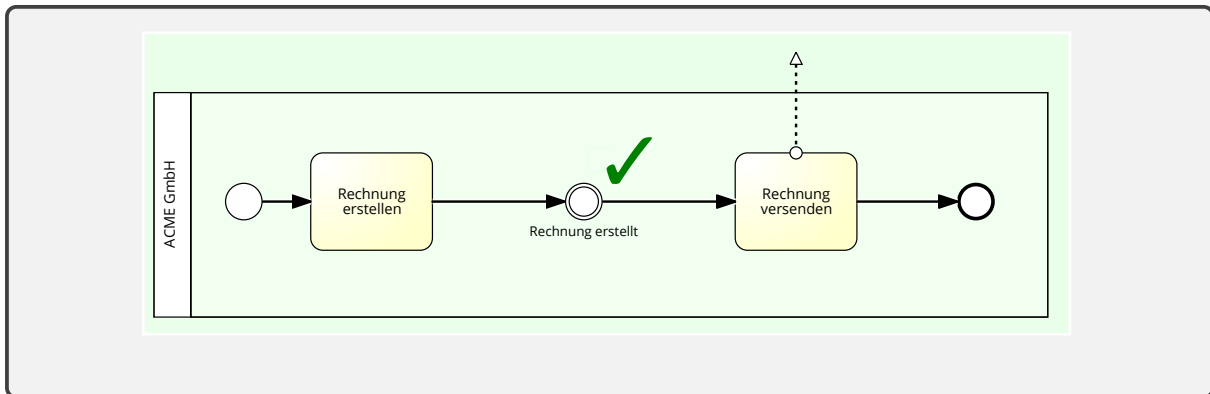
Die BPMN-Spezifikation erlaubt die Verwendung von mehreren unbestimmte Startereignissen in einem Pool. Dies wird jedoch problematisch, wenn der Prozess von einer Prozessengine oder einer Aufruf-Aktivität instanziiert wird, da nicht bestimmt werden kann, wie der Prozess gestartet wird. Deshalb erlaubt diese Konvention nur die Modellierung eines unbestimmten Ereignisses pro Pool.



### 3.2.5 Zwischenereignisse

#### 3.2.5.1 Unbestimmte Zwischenereignisse sind explizite Zustandsanzeigen

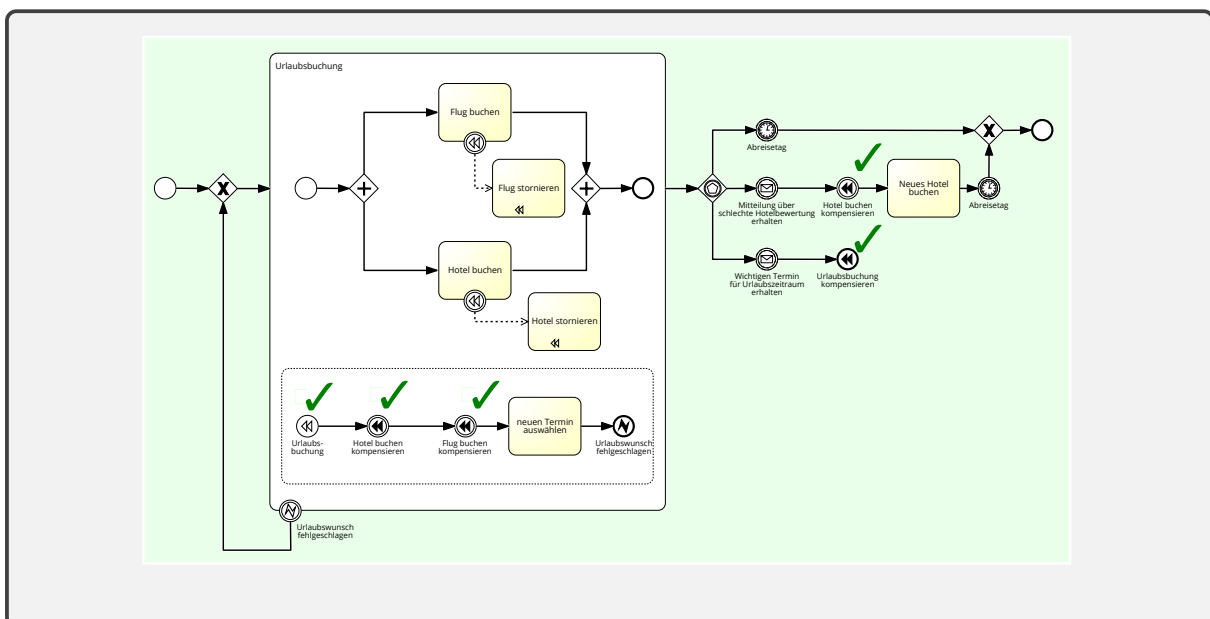
Untypisierte Zwischenereignisse stellen explizite Zustände dar, die durch den Prozessfortschritt erreicht werden. Sie dürfen wie typisierte Ereignisse im Diagramm verwendet werden – sollten aber nur verwendet werden, wenn ein erreichter Status explizit dargestellt werden soll, z.B. bei Meilensteinen.



#### 3.2.5.2 Auslösende Kompensationsereignisse referenzieren Aktivitäten

Wenn ein auslösendes Kompensations-Zwischen- oder Endereignis benutzt wird, dann muss dieses eine Bezeichnung besitzen, welche auf genau eine Aktivität verweist.

Handelt es sich dabei um eine atomare Aufgabe, so muss diese ein angeheftetes Kompensationsereignis besitzen. Handelt es sich um einen Unterprozess wird zuerst geprüft, ob ein Ereignis-Teilprozess mit einem Kompensations-Startereignis vorhanden ist. Wenn nicht, werden alle Aktivitäten des Unterprozesses mit angeheftetem Kompensationsereignis kompensiert.



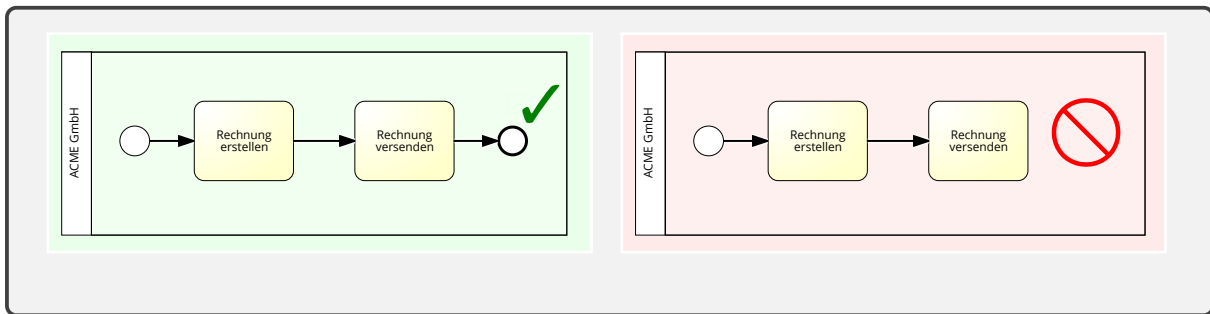
**Hinweis:** Technisch kann angegeben werden, ob ein auslösendes Kompensations-Zwischenereignis wartet bis alle Kompensationen durchgeführt wurden (synchron) oder ob der Prozessfluss sofort fortgeführt wird.

### 3.2.6 Endereignisse

#### 3.2.6.1 Mindestens ein Endereignis pro Pool

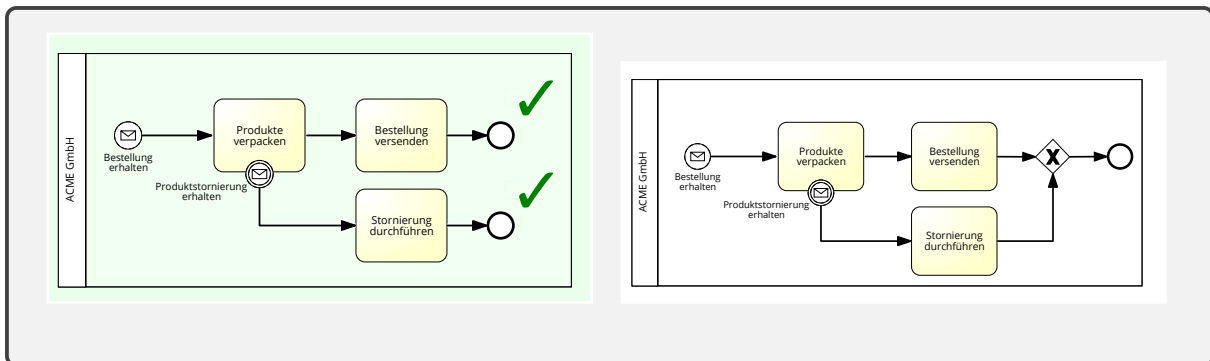
Jeder Pool muss mindestens ein Endereignis besitzen. Das implizite Verwenden von unbestimmten Endereignissen, indem aus einer Aktivität kein ausgehender Sequenzfluss modelliert wird, ist verboten.

In Regel 3.2.4.1 wird die Verwendung von Startereignissen vorgegeben. Bei einer solchen Vorgabe sagt die BPMN Spezifikation, dass Endereignisse explizit modelliert werden müssen.



#### 3.2.6.2 Mehrere Endereignisse pro Pool möglich

Es ist erlaubt in einem Pool mehrere Endereignisse zu verwenden. Der Übersichtlichkeit halber ist es oft sinnvoll einen verzweigten Prozessfluss mit einem Endereignis zu beenden anstatt alle Prozessflüsse letztendlich zusammenzuführen.

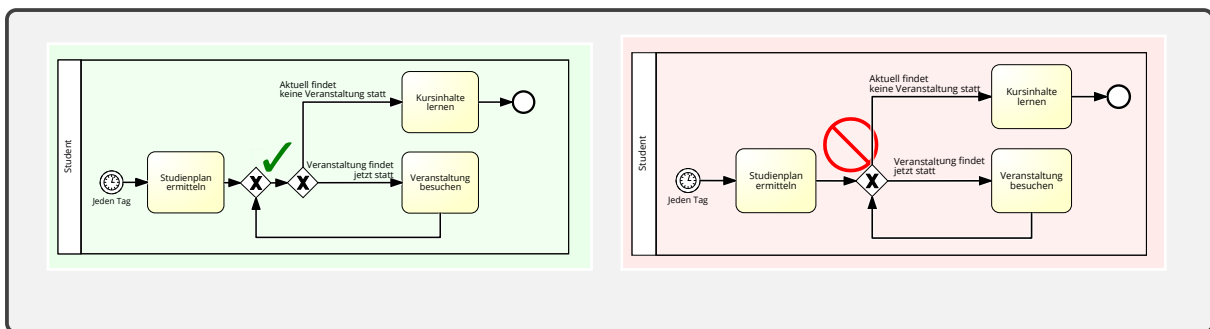


### 3.3 Gateways

#### 3.3.1 Genau ein ausgehender bzw. eingehender Sequenzfluss bei mehreren eingehenden bzw. ausgehenden Sequenzflüssen

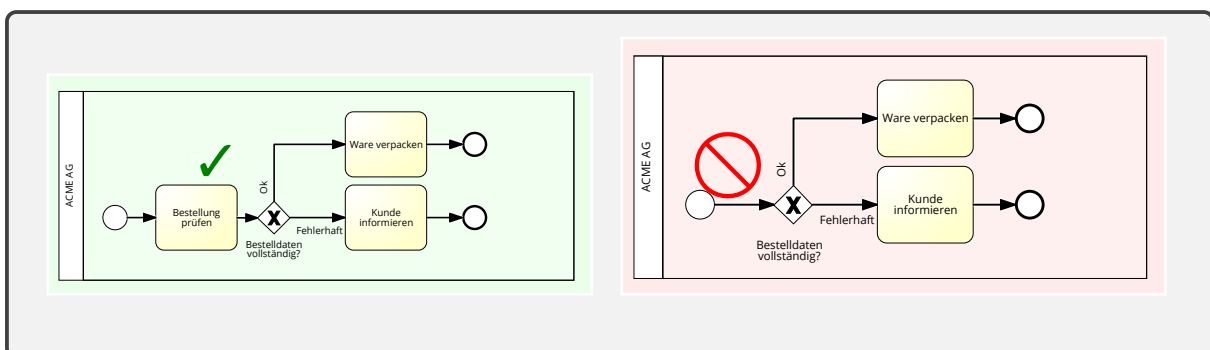
Gateways dürfen mehrere eingehende Sequenzflüsse haben. Wenn dies der Fall ist, darf es der Übersichtlichkeit halber nur einen ausgehenden Sequenzfluss geben.

Gateways dürfen auch mehrere ausgehende Sequenzflüsse haben. Wenn dies der Fall ist, darf es der Übersichtlichkeit halber nur einen eingehenden Sequenzfluss geben.



#### 3.3.2 Aktivität vor einem datenbasiertem Gateway

Exklusive und inklusive Gateways sind datenbasiert, d.h. das Aufteilen des Prozessflusses erfolgt auf Grundlage von Daten, die im Prozess verfügbar sind. Deshalb ist eine Aktivität, die die notwendigen Daten ermittelt, vor einem XOR- und OR-Split-Gateway obligatorisch. Zusätzlich kann (optional) eine Frage an das Gateway modelliert werden, welche die Entscheidung nochmals verdeutlicht.



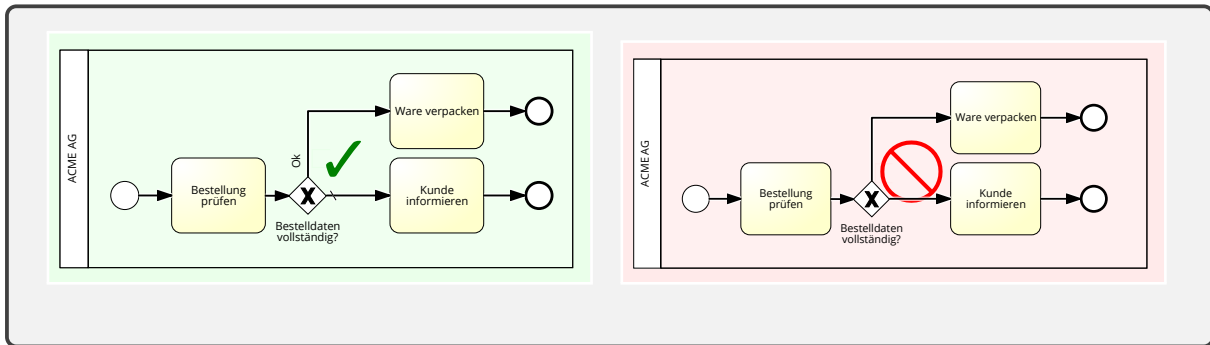
**Hinweis:** Es geht darum, dass die notwendigen Daten im Prozess verfügbar sind. Meistens wird deshalb eine Aktivität direkt vor dem Gateway modelliert. Es ist aber auch möglich, dass diese Daten-Abfrage bereits mehrere Schritte zuvor stattgefunden hat.

Z.B. dürfen mehrere Gateways aufeinander folgen (Achtung: das gilt nicht für ereignisbasierte Gateways). Wenn die Daten für die Entscheidung bereits verfügbar sind, ist das nicht problematisch. Dies ist jedoch mit Vorsicht anzuwenden, da die Lesbarkeit darunter leidet. Meistens können alternativ bei einer Verschachtelung von vielen Gateways auch Geschäftsregel-Aufgaben benutzt werden.

### 3.3.3 Bedingungen an den ausgehenden Sequenzflüssen eines datenbasierten Gateways

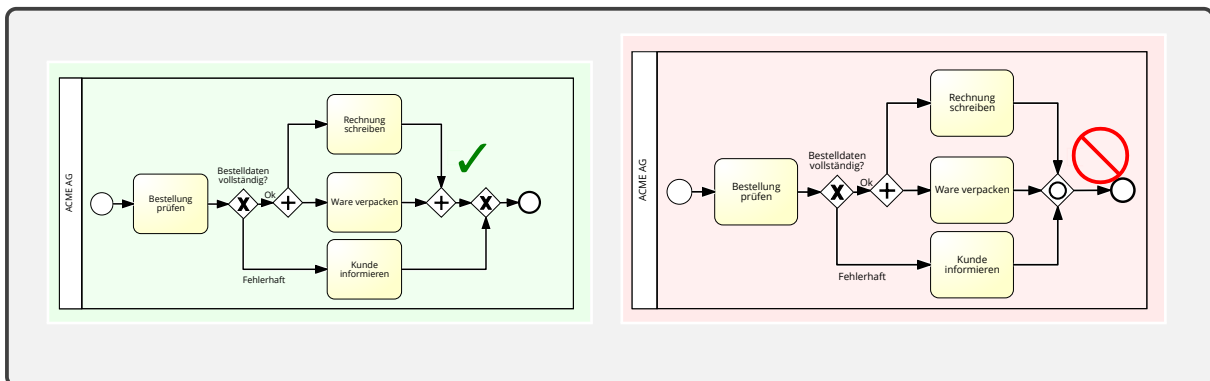
Ein datenbasiertes Split-Gateway teilt den Prozessfluss in ein oder mehrere Pfade auf. Damit ersichtlich ist, wann welcher Sequenzfluss fortgesetzt wird, muss jeder ausgehende Sequenzfluss mit einer Bedingung beschriftet sein.

**Ausnahme:** Die Verwendung des Standardfluss-Symbols bei datenbasierten Gateways entspricht einem ausgehenden Sequenzfluss mit der Bedingung „Sonstiges“. Das Symbol wird immer verwendet, wenn nicht alle Antworten mit Sicherheit im Vorfeld beantwortet werden können.



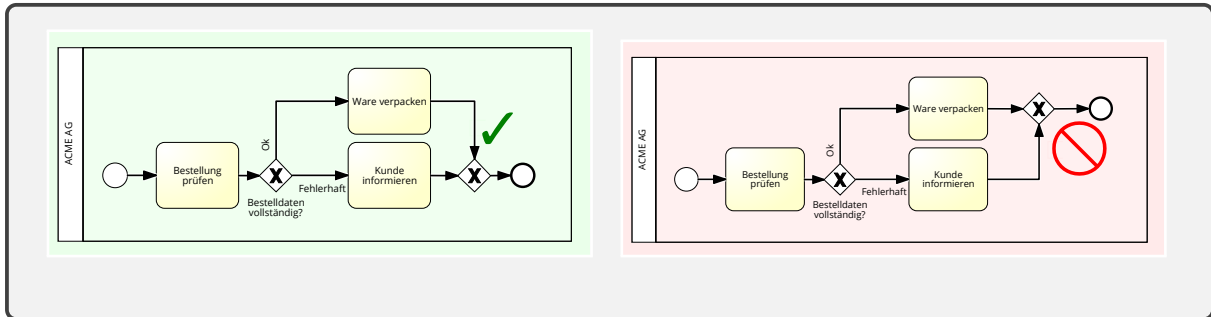
### 3.3.4 Komplexe Verzweigungen durch die selben Gateways andersherum wieder zusammenführen

Wenn ein Prozessmodell mehrere ineinander verschachtelte Verzweigungen besitzt, dann müssen die Sequenzflüsse durch die umgekehrte Reihenfolge der jeweils zuvor verwendeten Gateways wieder zusammengeführt werden.



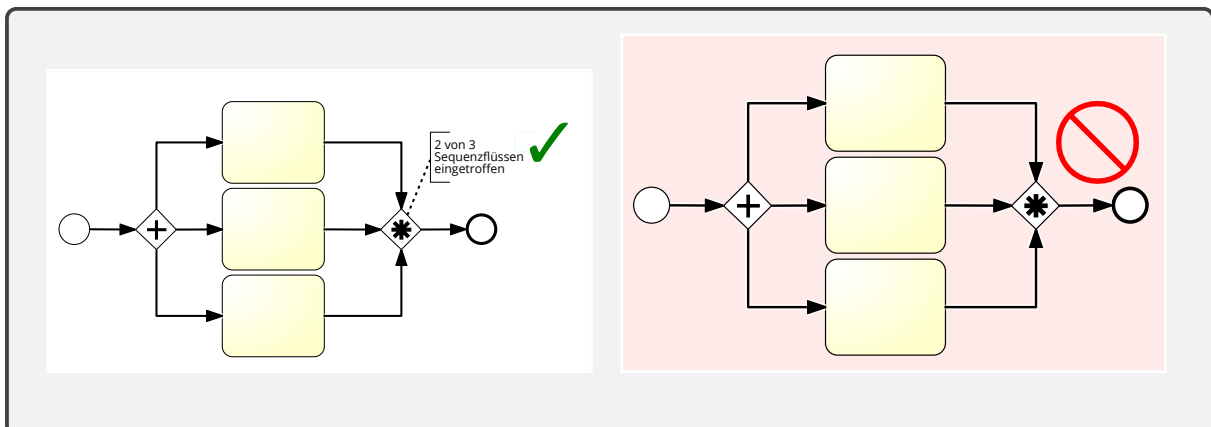
### 3.3.5 Verzweigte Sequenzflüsse auf derselben horizontalen Linie zusammenführen

Die Zusammenführung von verzweigten Sequenzflüssen erfolgt auf derselben horizontalen Linie wie die Verzweigung.



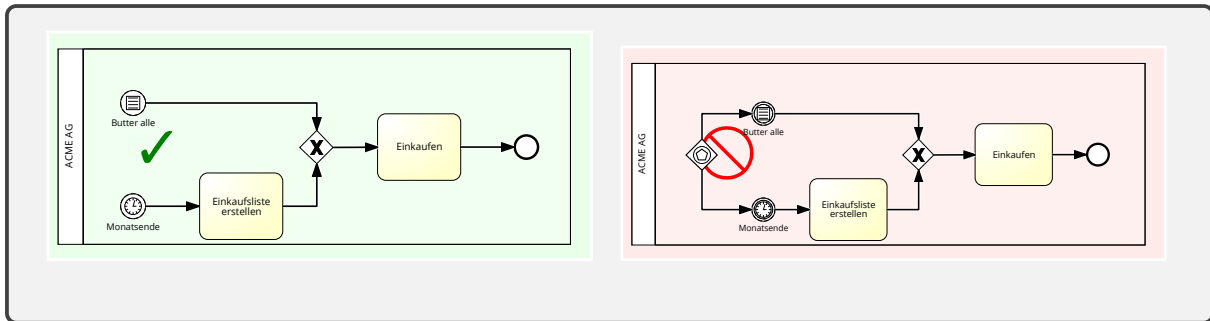
### 3.3.6 Komplexe Gateways vermeiden

Komplexe Gateways mit aussagekräftigen Text-Anmerkungen dürfen nur verwendet werden, wenn der abgebildete Sachverhalt nicht durch ein anderes Gateway dargestellt werden kann.



### 3.3.7 Verwendung von instanziiierenden exklusiven ereignisbasierten Gateways verboten

Semantisch ist die Verwendung von mehreren Startereignissen oder die Verwendung von einem instanziiierenden exklusiven ereignisbasiertem Gateway, welches mit mehreren Zwischenereignissen verbunden ist, meist gleich: es muss eines der dargestellten Ereignisse eintreten, welches den Prozess instanziiert. Es handelt sich dann um zwei verschiedene Modellierungsvarianten, weshalb in dieser Konvention die Verwendung des genannten Gateways verboten wird.



**Erklärung:** Eine Unterscheidung macht die Spezifikation (Abschnitt 13.5.1, gilt auch für das parallele instanziiierende ereignisbasierten Gateway). Wenn dieses Verhalten gewollt ist, kommt man um die Verwendung des entsprechenden Gateways nicht herum:

“It is possible to have multiple groups of Event-Based Gateways starting a Process, provided they participate in the same Conversation and hence share the same correlation information. In that case, one Event out of each group needs to arrive; the first one creates a new Process instance, while the subsequent ones are routed to the existing instance, which is identified through its correlation information.”



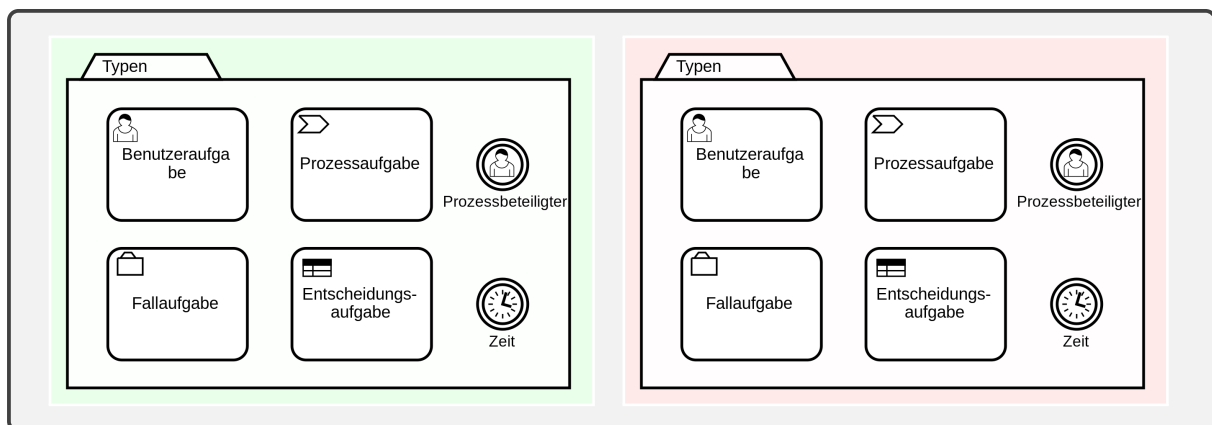
# Kapitel 4

## Modellierungskonventionen CMMN

### 4.1 Allgemein

#### 4.1.1 Aufgaben stets typisieren

Eine Aufgabe muss stets eine der vier möglichen Aufgabentypen zugeordnet sein: Benutzeraufgabe, Prozessaufgabe, Fallaufgabe oder Entscheidungsaufgabe. Ein Ereignis muss ebenfalls immer typisiert sein: Benutzer- oder Zeitevent.



#### 4.1.2 Manuelle Ausführung von Aufgaben

Es gibt Aufgaben, die manchmal in einem Fall bearbeitet werden müssen. Der Fallbearbeiter kann selbst entscheiden, ob er diese Aufgabe durchführt. Solche Aufgaben sind mit einem unausgefüllten Dreieck (Play-Symbol) versehen.

**Achtung:** An dieser Stelle ist die Spezifikation leider widersprüchlich und definiert die Semantik teilweise unterschiedlich. Wir legen fest: wenn der Marker vorhanden ist, wird die Aufgabe/Stage zuerst in den Zustand *Enabled* überführt und wartet darauf vom Akteur manuell gestartet zu werden. Wenn dieser Marker nicht vorhanden ist, wird die Aufgabe/Stage automatisch gestartet sobald sie sich im Zustand *Available* befindet und alle Bedingungen erfüllt sind.

### 4.1.3 Beendigung einer Stage mit Autocomplete

Eine Stage mit AutoComplete endet im Zustand *Completed*, wenn es keine *Active* Kind-Aufgaben/Stages mehr gibt UND alle erforderlichen Aufgaben bearbeitet wurden. Das heißt, es könnte z.B. noch normale Aufgaben mit Eingangswächter und manuelle Aufgaben geben, die sich im Zustand *Available* oder *Enabled* befinden, die nie ausgeführt werden.

Eine Stage ohne AutoComplete (schwarzes Stop-Viereck am unteren Rand) endet im Zustand *Completed*, wenn der Fallbearbeiter keine weiteren Aufgaben erledigen könnte – d.h. alle Kind-Elemente befinden sich im Zustand *Completed*, *Terminated*, *Failed* oder *Disabled* (dies betrifft sowohl manuelle, normale als auch erforderliche Aufgaben) UND es keine Planbaren Aufgaben gibt.

**Hinweis:** In der Spezifikation gibt es einen kleinen Fehler, so dass AutoComplete normalerweise keine Events beachtet. Wir definieren: eine Stage mit AutoComplete beachtet Events nicht, eine Stage ohne Autocomplete erwartet Events im Zustand *Completed* oder *Terminated* bevor sie enden kann.

### 4.1.4 Beendigung eines Falls

Ein Fall endet im Zustand *Completed*, wenn alle enthaltenen Meilensteine, Stages und Aufgaben sich in einem Endzustand befinden (*Completed*, *Terminated*, *Disabled*, *Failed*).

Außerdem kann ein Fall auch enden im Zustand *Terminated*, wenn ein Ausgangswächter am Fallrand zutrifft, oder ein Fallbearbeiter den Fall manuell schließt.

## 4.2 Wächter

Bei den Wächtern kann man unterscheiden zwischen Eingangs- (weißer Diamant) und Ausgangs-Wächter (schwarzer Diamant). Ein Wächter evaluiert ein Lebenszyklus-Event (OnPart) oder eine Bedingung (IfPart), welche sich auf Daten in einem Informationsobjekt des Falls bezieht. Ein Wächter enthält immer mindestens ein OnPart oder ein IfPart, kann aber auch beide besitzen.

### 4.2.1 Verbindungen beschriften mit Lebenszyklus-Events

Der Verbinder zwischen einem Element und einem Eingangs-Wächter visualisiert den OnPart eines Wächters, d.h. dieser überwacht das Eintreten eines definierten Lebenszyklus-Events.

Um darzustellen um welchen Lebenszyklus-Übergang es sich handelt, wird dieser normalerweise auf den Verbinder geschrieben. Da einige Übergänge häufiger vorkommen, braucht in den folgenden Standardfällen das Lebenszyklus-Event nicht auf den Verbinder geschrieben werden:

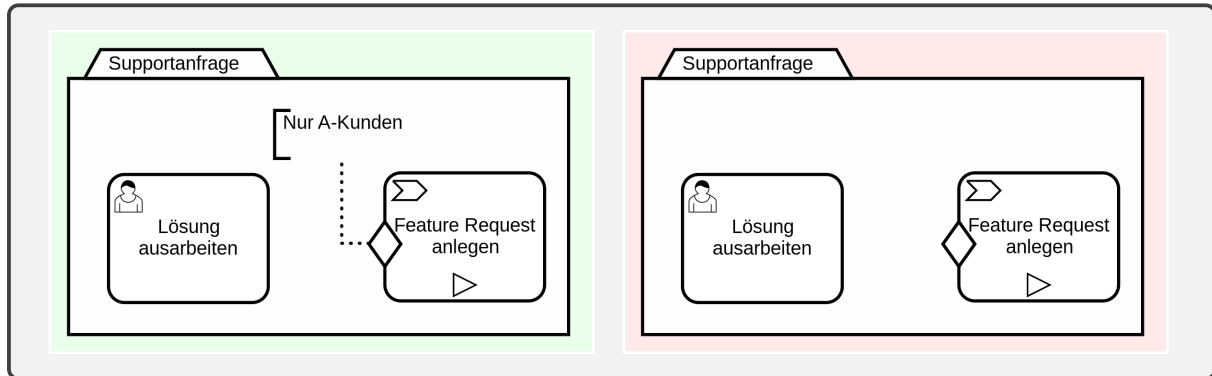
- ausgehende Verbindung aus einer Aufgabe; Standard: *complete*-Event
- ausgehende Verbindung aus einem Meilenstein; Standard: *occur*-Event
- ausgehende Verbindung aus einem Event; Standard: *occur*-Event
- ausgehende Verbindung aus einem Ausgangs-Wächter<sup>1</sup>; Standard: *exit*-Event

Wenn jedoch auf das Eintreten eines anderen Lebenszyklus-Events als der genannten geprüft wird, muss dieses explizit auf den Verbinder geschrieben werden, z.B. bei *start*, *enable*, etc.

<sup>1</sup> Tatsächlich gibt es keinen Lebenszyklus für Wächter. Hiermit ist eigentlich das *exit*-Event von Tasks/Stages gemeint, welches auftritt, wenn ein Ausgangswächter eintritt. Um dies abzugrenzen von einer normalen ausgehenden Verbindung (*complete*), legt diese Konvention fest, dass die Verbindung aus dem Ausgangswächter erfolgen muss, damit *exit* weggelassen werden kann.)

### 4.2.2 Bedingung eines Eingangs-Wächters mit Textanmerkung beschreiben

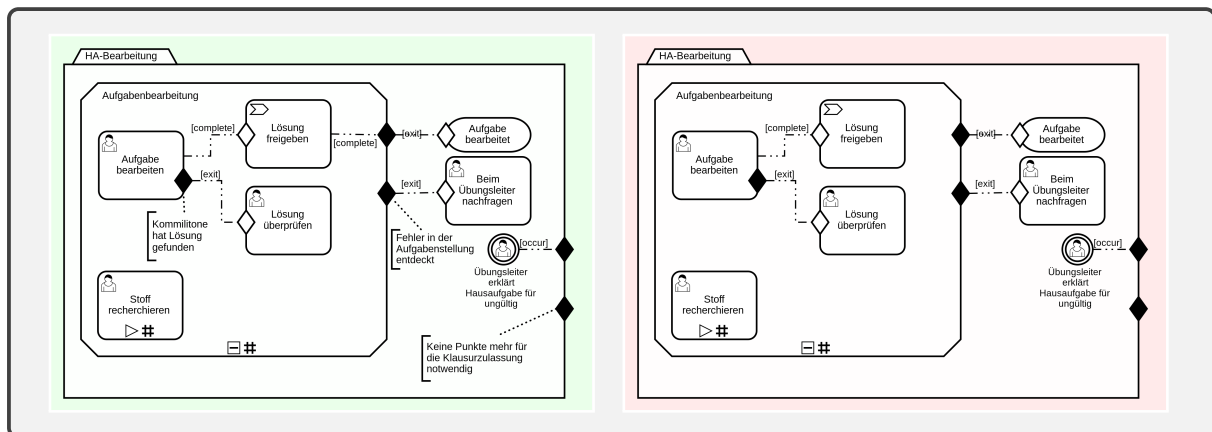
Wenn ein Eingangs-Wächter eine Bedingung (If-Part) besitzt, muss diese Bedingung mit einer Textanmerkung am Wächter beschrieben werden. Wenn keine Textanmerkung vorhanden ist, wird angenommen, dass der Wächter keinen If-Part besitzt.



### 4.2.3 Bedingung eines Ausgangs-Wächters mit Textanmerkung beschreiben

Aufgaben, Abschnitte und Fälle können Ausgangs-Wächter besitzen, die auf das Eintreten eines Lebenszyklus-Events (OnPart) (siehe dazu 4.2.1) oder einer Ausstiegsbedingung (IfPart) prüfen. Wenn es sich um letzteres handelt, muss eine Textanmerkung die Bedingung beschreiben.

RahmenNoFloat



**Erklärung:** Wenn ein Ausgangs-Wächter zutrifft, wird die betreffende Aufgabe, der Abschnitt oder der Fall beendet, d.h. in den Zustand *Terminated* überführt. Dadurch werden (bei Abschnitten und Fällen) alle enthaltenen Aufgaben und Abschnitte ebenfalls beendet und in den Zustand *Terminated* überführt. Im Gegensatz zur normalen Erledigung müssen aktive, offene oder erforderliche Aufgaben dabei nicht abgeschlossen sein, sondern werden beim Eintreffen des Ausgangs-Wächters gewissermaßen abgebrochen.