Betriebswirtschaftslehre II
Vorlesung 12: Prozessmodellimplementierung

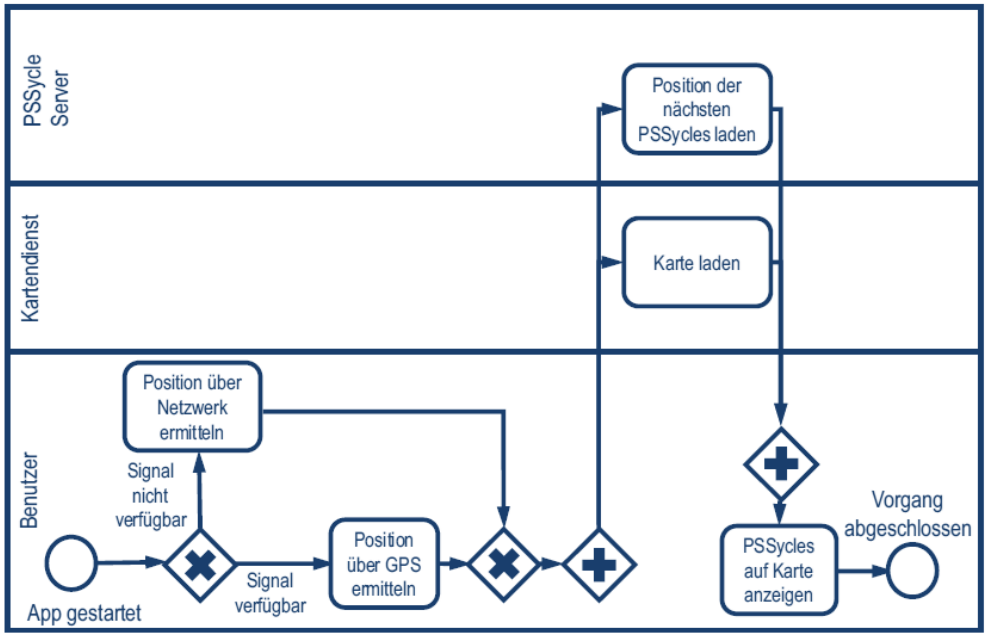
Wintersemester 2018/19

Prof. Dr. Martin Schultz
martin.schultz@haw-hamburg.de

Bisherige Sicht auf Prozessmodelle

- Beschreibung von Geschäftsprozessen mit Modellen unter Verwendung geeigneter Prozessmodellierungssprachen (z.B. EPK, BPMN)

→ die erstellten Modelle dienen dann als **Spezifikationsteil** für die Entwicklung von **Individualsoftware** zur Unterstützung der Geschäftsprozesse bzw. zur Unterstützung bei der Einführung/ Customizing von **Standardsoftware**



Auführung von BPMN-Modellen

- Der BPMN-Standard sieht explizit ausführbare Modelle vor, für die geeignete Modellelemente (z.B. Script Task) mit entsprechender **Anreicherung** (insbesondere formale Beschreibung der Semantik, z.B. JAVA-Code) verwendet werden müssen

- **Idee also**: BPMN-Modell in ausführbarer Form erstellen und direkt an eine **Execution Engine** übergeben

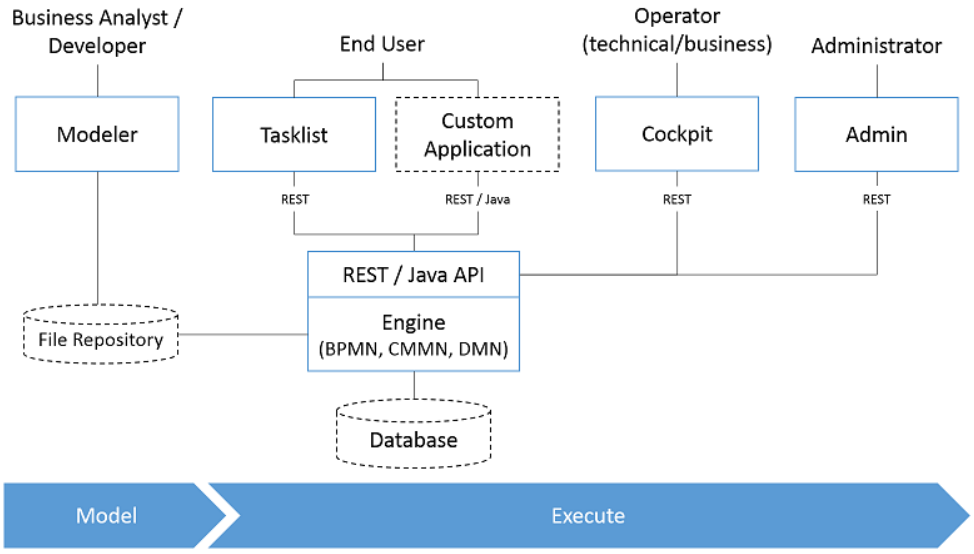
- **Mittel zur Übergabe:** Der BPMN-Standard definiert ein XML-Format (Austauschformat) zur textuellen Beschreibung von BPMN-Modellen

```
<bpmn:startEvent id="StartEvent_1">
  <bpmn:outgoing>SequenceFlow_1jyna88</bpmn:outgoing>
</bpmn:startEvent>
<bpmn:task id="Task_1ukd8pt" name="Angebotsanfrage erstellen">
  <bpmn:incoming>SequenceFlow_1jyna88</bpmn:incoming>
  <bpmn:outgoing>SequenceFlow_0jfm0ve</bpmn:outgoing>
</bpmn:task>
<bpmn:sequenceFlow id="SequenceFlow_1jyna88" sourceRef="StartE
<bpmn:task id="Task_0gikaw3" name="Auftrag bestätigen">
  <bpmn:incoming>SequenceFlow_0jfm0ve</bpmn:incoming>
  <bpmn:outgoing>SequenceFlow_0hbw3kw</bpmn:outgoing>
</bpmn:task>
<bpmn:sequenceFlow id="SequenceFlow_0jfm0ve" sourceRef="Task_1
<bpmn:endEvent id="EndEvent_01cvgl7">
  <bpmn:incoming>SequenceFlow_0hbw3kw</bpmn:incoming>
</bpmn:endEvent>
```

Execution Engine (Workflow-Engine)

- Die Model Execution Engine liest Modelle als XML-Datei ein und führt sie „direkt“ aus.
- Die Modelle sind somit als Quellcode einer Softwarelosung anzusehen.
  Zur Workflow- oder Prozessautomatisierung benötigt die Workflow Engine den Geschäftsprozess in einem Modell, das alle technischen Details enthält, die zur Ausführung benötigt werden.
  → Die Modelle müssen exakt und detailliert definiert sein
- Zur **Laufzeit** werden dann Prozessinstanzen für jeden individuellen Prozessdurchlauf erzeugt, wobei die Workflow Engine den **Kontrollfluss** berechnet und immer „weiss“, was als Nächstes zu tun ist.

Beispiel camunda



Execution Engine (Workflow-Engine) - Gartner

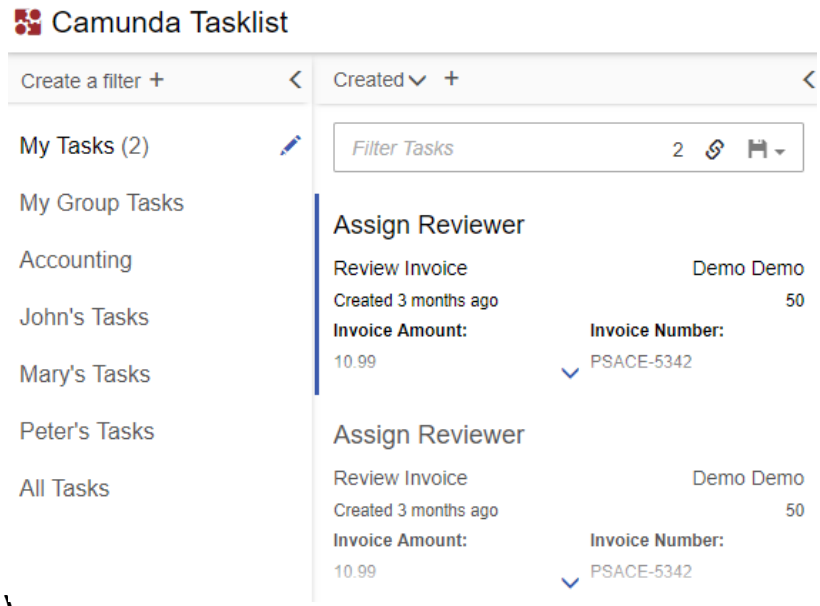
Figure 1. Magic Quadrant for Intelligent Business Process Management Suites

Source: Gartner (October 2017)

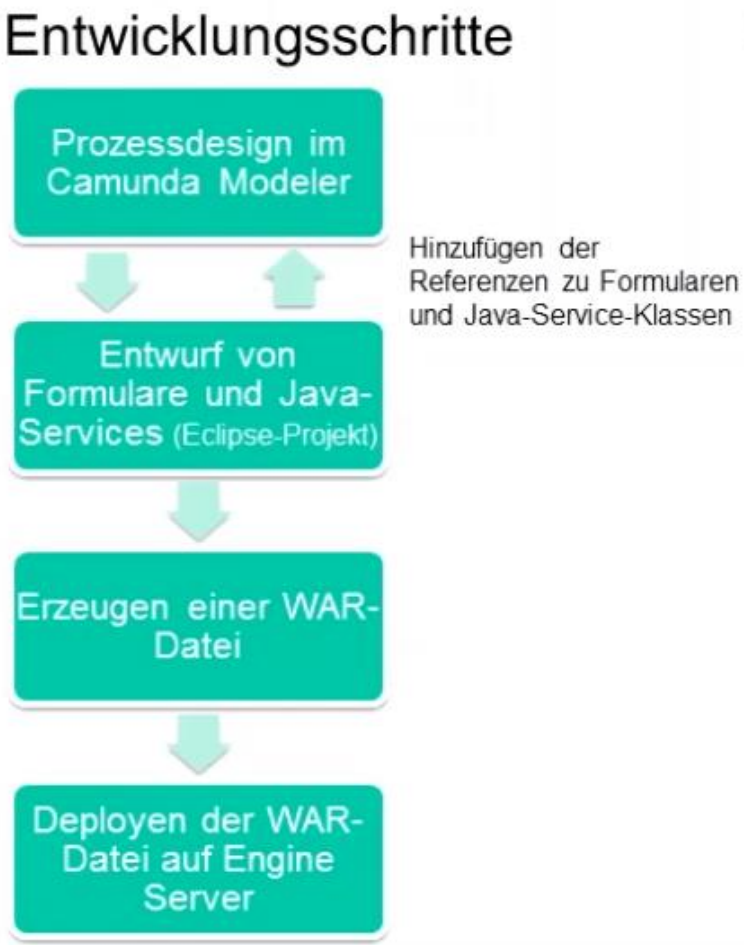
Ausgangslage

- Eine Engine kennt zwei grundsätzlich unterschiedliche Arten von Aktivitäten:
  - die, bei denen eine **menschliche Interaktion** notwendig ist,
  - und alle anderen, die **automatisiert** ablaufen können (z.B. Serviceaufrufe, die Auswertung von Gateways und Ereignissen.

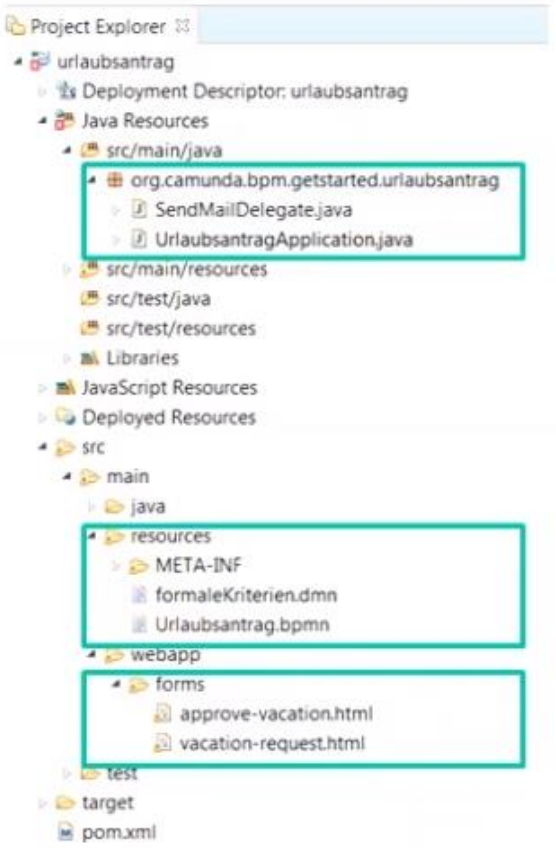
- Für **menschliche Interaktionen** werden die Benutzer-Aufgaben eingesetzt. Engines bringen üblicherweise eine **Aufgabenliste** mit, anhand derer der Benutzer weiß, welche Aufgaben noch zu erledigen sind. Je Aufgabe stehen vorkonfigurierte **Bildschirmmasken** zur Verfügung, mit der man Daten einsehen und bearbeiten oder Entscheidungen treffen kann.
- Für **automatisierbare Aufgaben** kann jeweils die Ausführungssemantik (z.B. in JAVA) implementiert werden



Notwendige Schritte (camunda)



Hinzufügen der Referenzen zu Formularen und Java-Service-Klassen

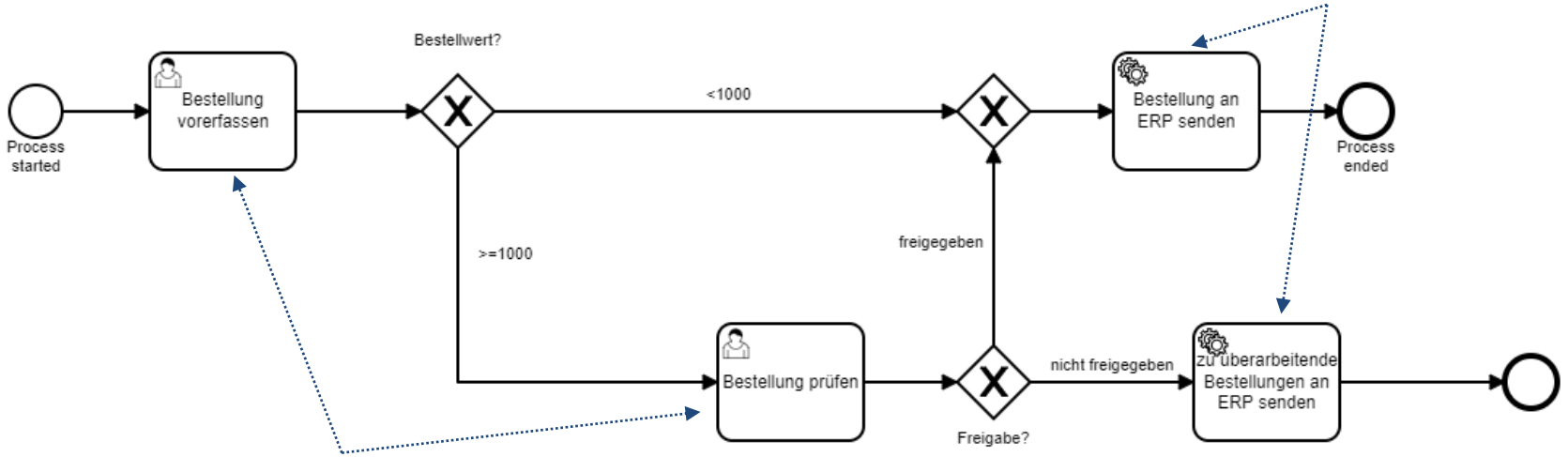


Eclipse-Projekt

Beispiel Dirt Bikes

- Bei Dirt Bikes soll die Bearbeitung von Bestellungen und deren Freigabe automatisiert werden
- Ein entsprechendes BPMN-Modell liegt bereits vor

Automatisierbare Aufgabe (Service Task)

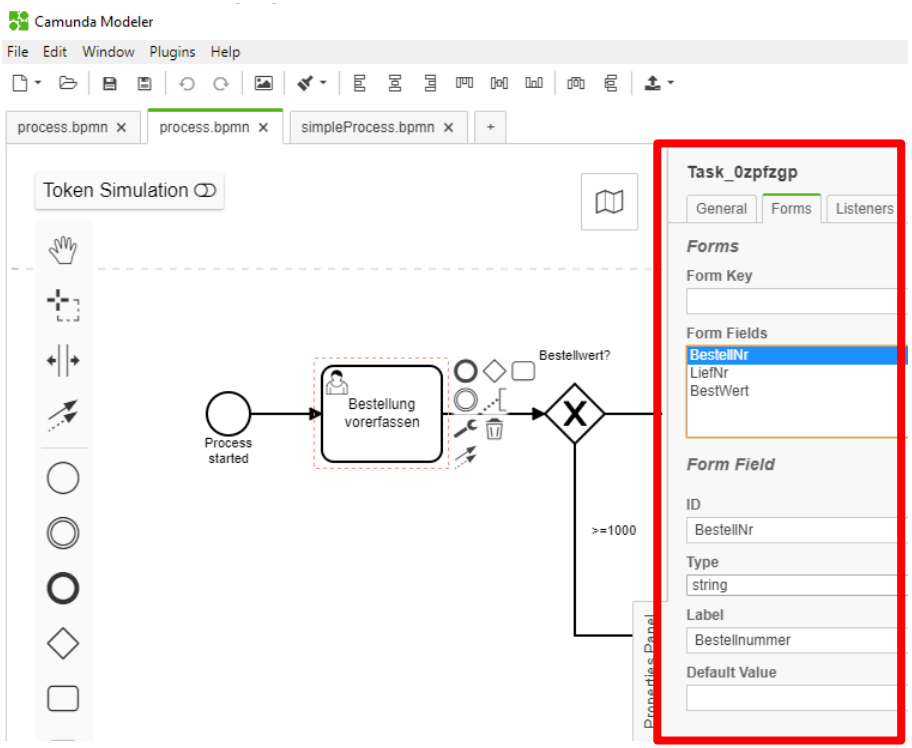


menschliche Interaktion (User Task)

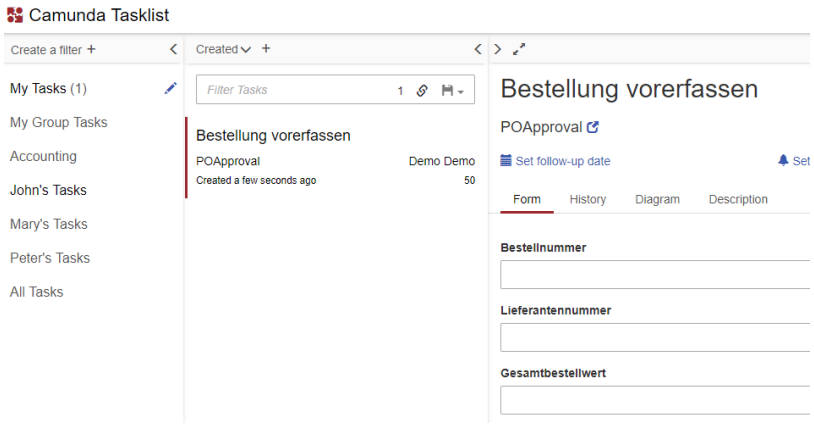
Beispiel Dirt Bikes: User Tasks

- Erstellung der Formulare (Bildschirm-Masken) für die User-Tasks

BPMN Modeler



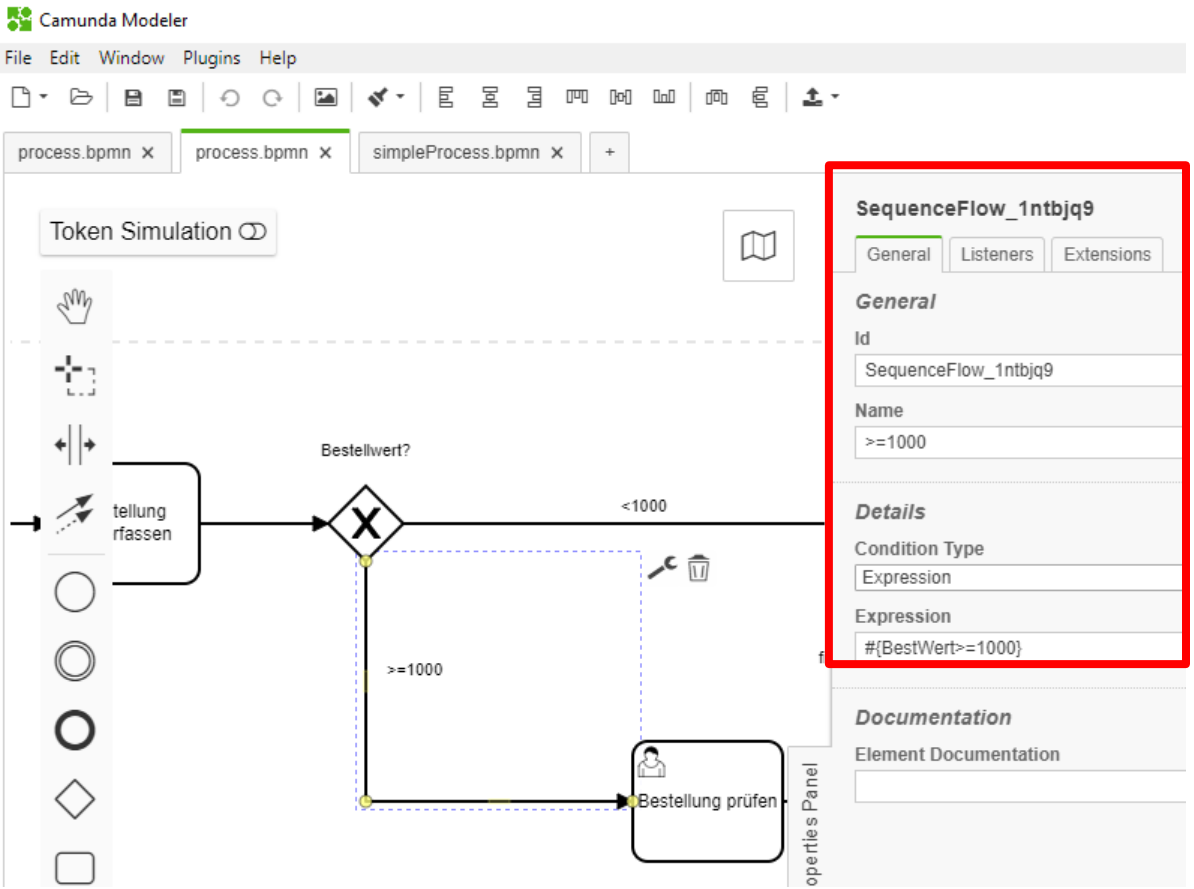
Task List



Beispiel Dirt Bikes: Gateways

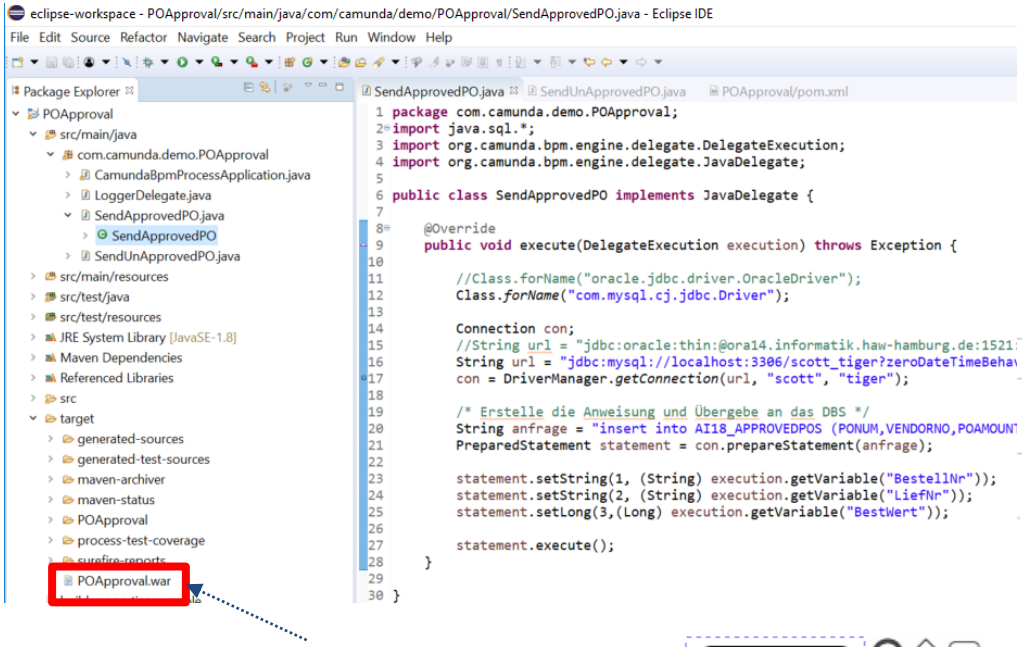
- Festlegung von Bedingungen zur Steuerung des Kontrollflusses entlang der Gateways

BPMN Modeler

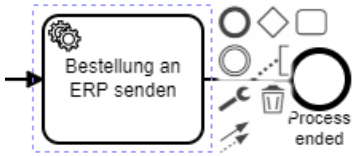


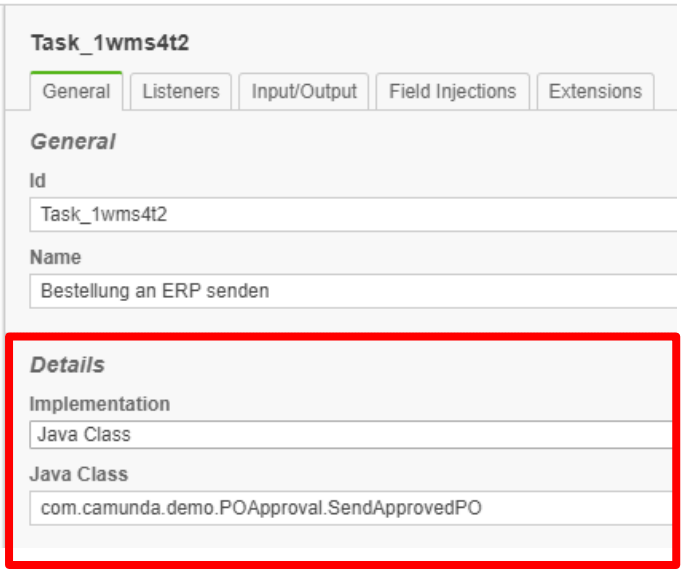
Beispiel Dirt Bikes: Service Tasks

- Implementierung der Verarbeitungslogik für die automatisierbaren Aktivitäten im Prozessmodell
- Verknüpfung der implementierten Logik mit der Task Prozessmodell



- Erzeugen einer war-Datei
- Deployment der war-Datei in der Execution Engine





Wann lohnt sich eine Execution Engine

Hohe Wiederholungszahl: Der Aufwand für die Automatisierung lohnt sich natürlich nur, wenn entsprechend viele Instanzen zur Ausführung kommen, da sonst die Entwicklungskosten eventuell eingesparte Prozesskosten bei Weitem übersteigen.

Standardisierung: Sind Prozesse nur schwach strukturiert und laufen ständig anders ab, ist die Engine fehl am Platz. Ein Großteil der Instanzen muss also dem gleichen Muster folgen.

Informationslastig: Prinzipiell eignen sich informationslastige Prozesse besser zur Automatisierung. Werden oft physische Gegenstände bewegt, ist die Automatisierung meist schwierig und weniger spannend.

Hoher potenzieller Automatisierungsgrad: Effizienz im Prozessablauf kann man natürlich durch Automatisierung von Aufgaben steigern. Manche Aufgaben wie zum Beispiel das Buchen in einem ERP-System eignen sich sehr gut, um automatisiert aus der Engine heraus angesprochen zu werden. Die Daten müssen nicht mehr manuell in eine Maske eingegeben werden. Manche manuellen Aufgaben eignen sich nicht zur Automatisierung, beispielsweise das Anrufen eines Kunden.