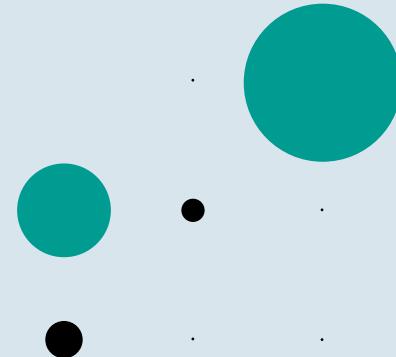




Workflow2Go

Geschäftsprozessautomatisierung durch Workflow Management Systeme



Vorstellung

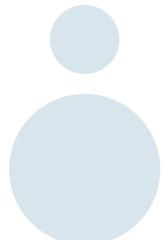
Team Digiboyz:

- Furkan Yücel
- Marc Eberle
- Luca Merz
- Jakub Rzasa
- Felix Peter
- Eduard Lang

Wichtiger Link

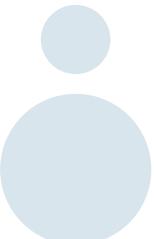
Sie finden alle benötigten Unterlagen, sowie
Links auf GitHub:

- <https://github.com/furkanyuecel/Workflow2Go>



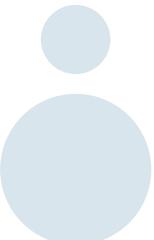
Motivation

- Geschäftsprozesse besser verstehen
- Kunden- und Serviceorientierung
- Wirtschaftlichkeit
- Neue/s Fähigkeiten/Wissen
- Unbenotete Prüfungsleistung 😊



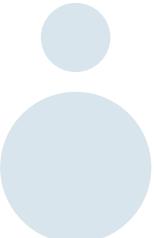
Lernziele

- BPMN 2.0 anwenden
- Workflows und Workflow-Management verstehen
- Modellieren mit Camunda



Roadmap

	Tag 1 – Einführung	Tag 2 – Camunda Workshop	Camundathon
Morgen	BPMN - Theorie	Camunda Basics	<ul style="list-style-type: none">• Komplexe Modellierung• Probleme aus realen Szenarien• Jung, kreativ, Bier, Modellieren, Spaß• Freiwilliger Abend• Nacht des Workflow-Managements
Vormittag	BPMN - Aufgaben	Workflowmodellierung	
Mittag	Workflow-Management	Workflowmodellierung	
Nachmittag	Camunda Einstieg	Besprechung der Aufgaben	

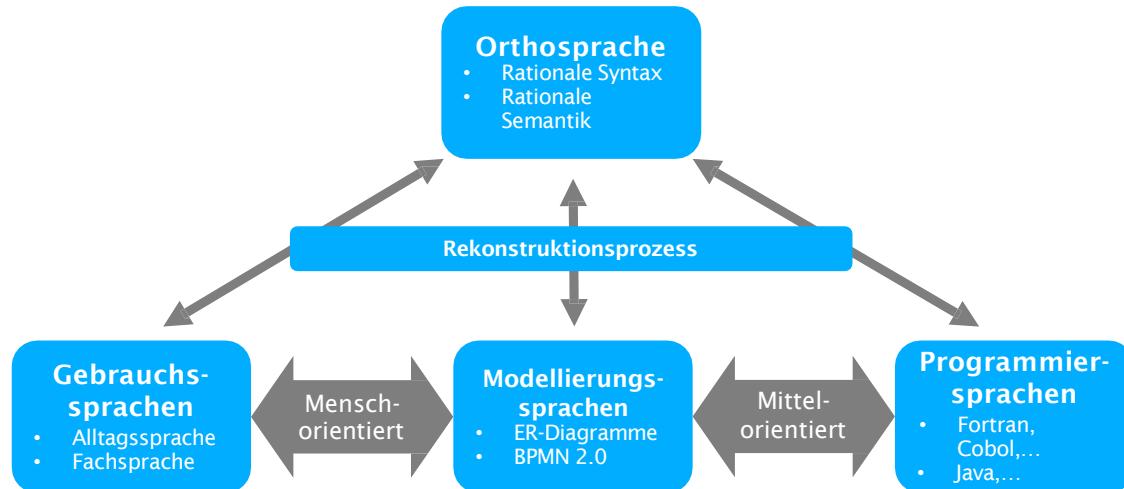


BPMN 2.0

Wiederholung

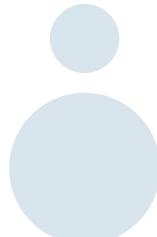
Geschäftsprozessmanagement

Sprachen

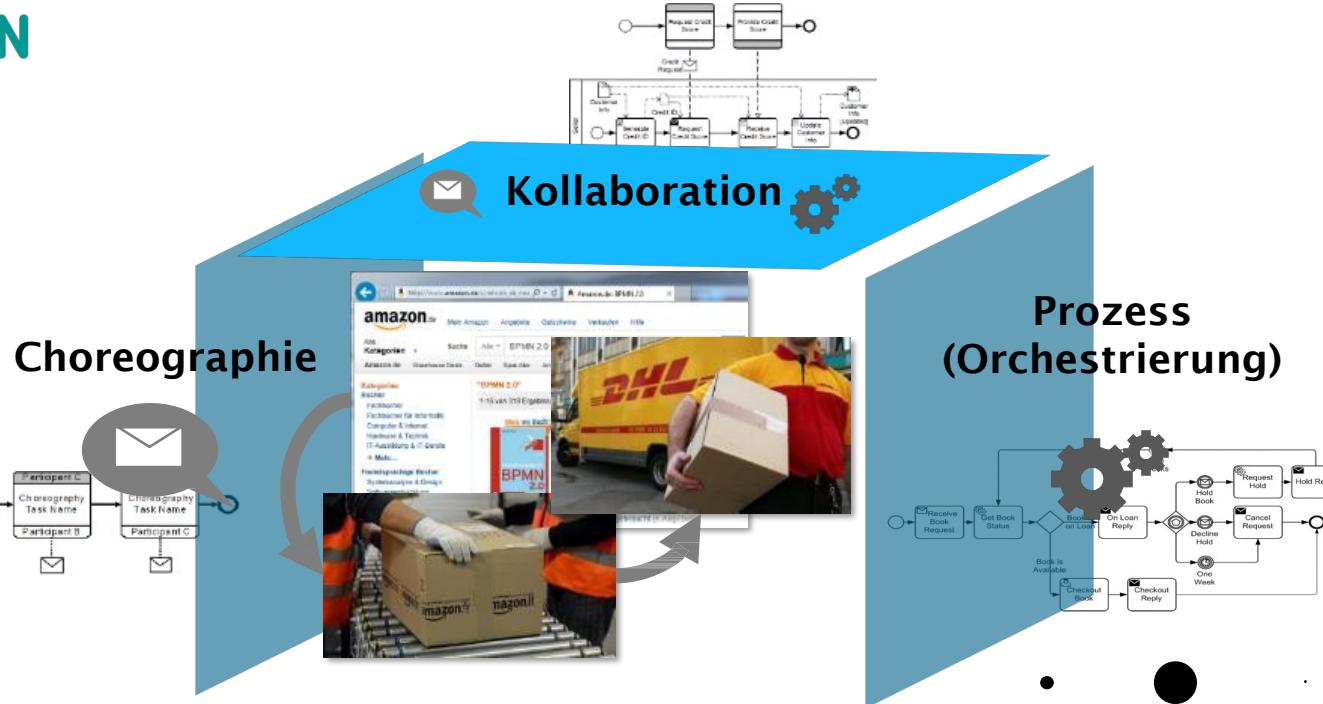


Was ist BPMN 2.0?

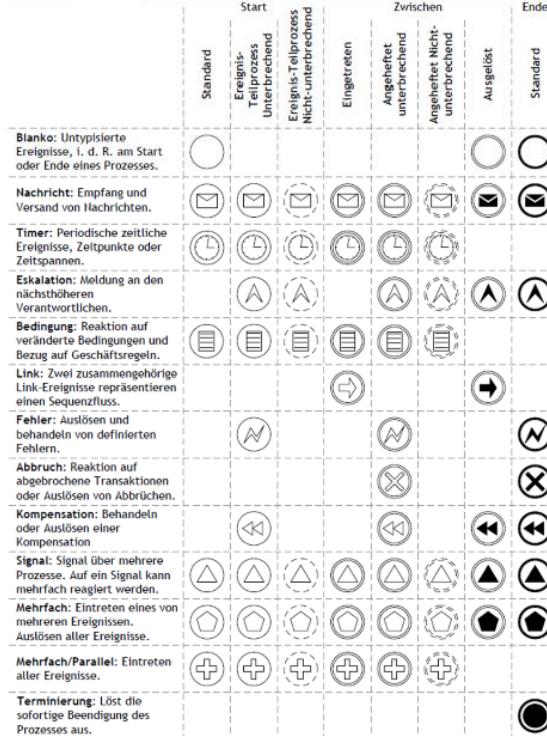
- Business Process Model and Notation
- Standard der Object Management Group (OMG)
- Seit Dezember 2013 *ISO/IEC 19510* Standard
- Ziele von BPMN:
 - Verständliche Notation für alle Prozessbeteiligten (Business-Analysts, Anwender, Entwickler)
 - Standardisierte Brücke zwischen Geschäftsprozessdesign und -implementierung

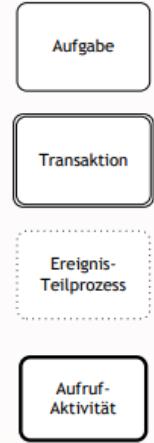


Diagrammtypen und Blickwinkel



Events





Aufgaben

Markierungen

Sie beschreiben das Ausführungsverhalten von Aktivitäten:

- + Teilprozess
- ⟳ Schleife
- ☰ Parallel Mehrfachausführung
- ☰ Sequentialie Mehrfachausführung
- Ad-Hoc
- ◀ Kompensation

Aufgaben-Typen

Sie beschreiben den Charakter einer Aufgabe:

- ✉ Senden
- ✉ Empfangen
- 👤 Benutzer
- 📁 Manuell
- 📝 Geschäftsregel
- ⚙️ Service
- scriptId Skript

Sequenzfluss

Gateways



Exklusives
Gateway



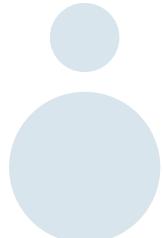
Paralleles
Gateway



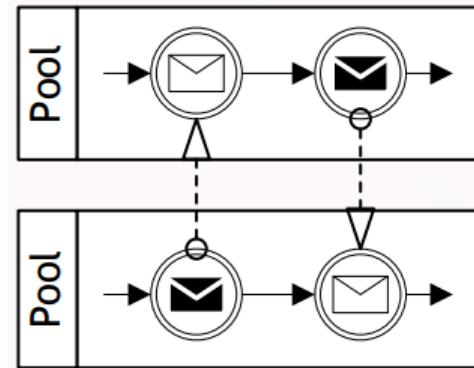
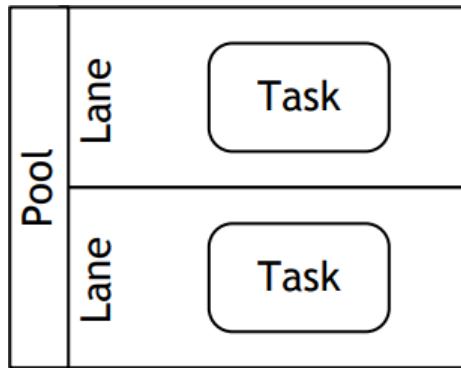
Ereignis-
basiertes
Gateway



Inklusives
Gateway

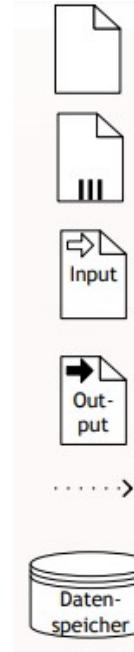


Pools & Lanes



Nachrichtenfluss

Daten



Ein **Datenobjekt** repräsentiert Informationen, die durch den Prozess fließen, wie z.B. Dokumente, Emails, Briefe oder Datensätze.

Ein **Listen-Datenobjekt** repräsentiert eine Gruppe von Informationen, wie z.B. eine Liste mit Bestellpositionen.

Ein **Dateninput** ist ein externer Input für den ganzen Prozess, der von einer Aktivität gelesen wird, und die nach außen sichtbare Eingangs-Prozessschnittstelle.

Ein **Datenoutput** ist eine Variable, die als Ergebnis eines ganzen Prozesses erzeugt wird, und die nach außen sichtbare Ausgangs-Prozessschnittstelle.

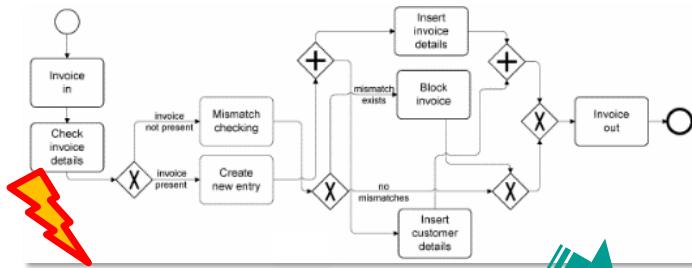
Eine **Daten-Assoziation** verknüpft Datenobjekte mit Aktivitäten, Prozessen und Aufruf-Aktivitäten.

Ein **Datenspeicher** ist ein Ort, auf den der Prozess lesend und schreibend zugreifen kann, z.B. eine Datenbank oder ein Aktenenschrank. Er existiert unabhängig von der Lebensdauer der Prozessinstanz.

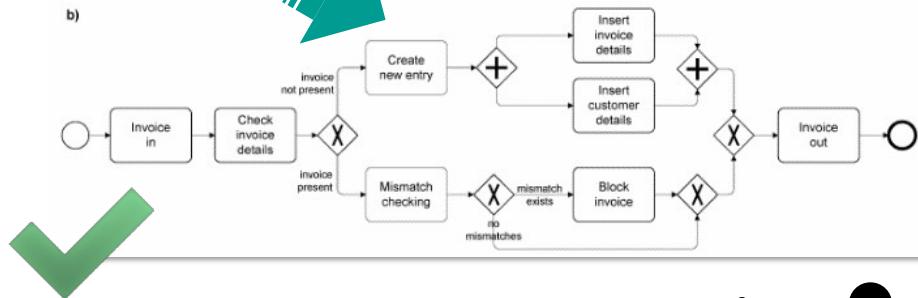
„Best Practices“

Do's und Don'ts in der Geschäftsprozessmodellierung

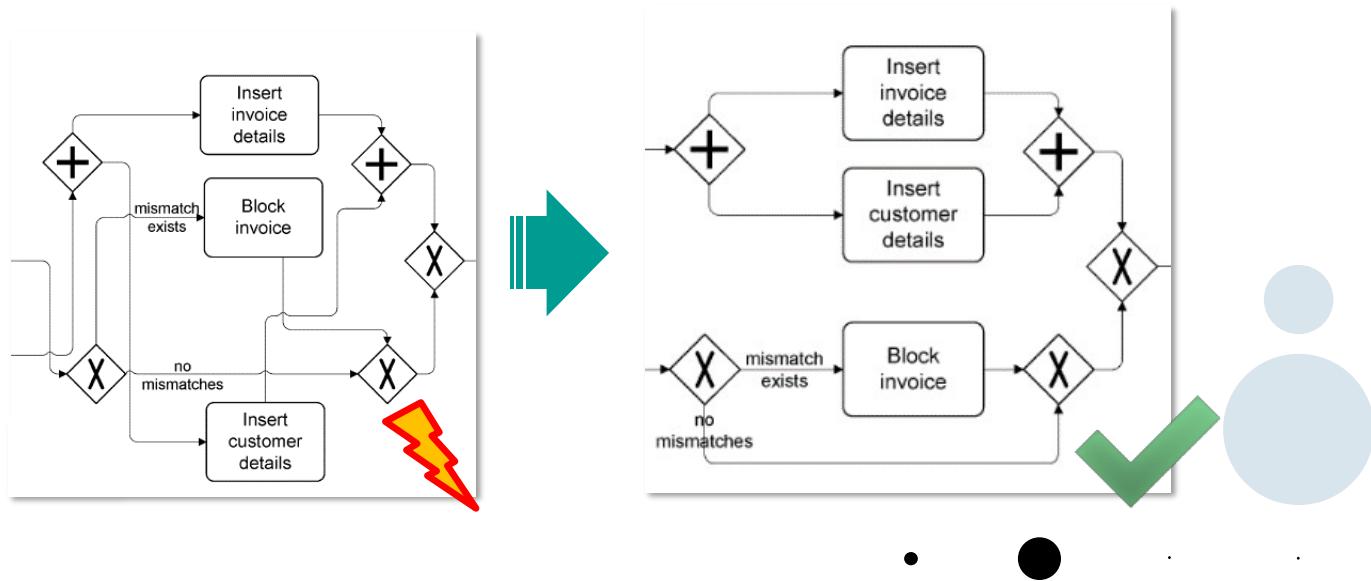
Modellierung in Leserichtung



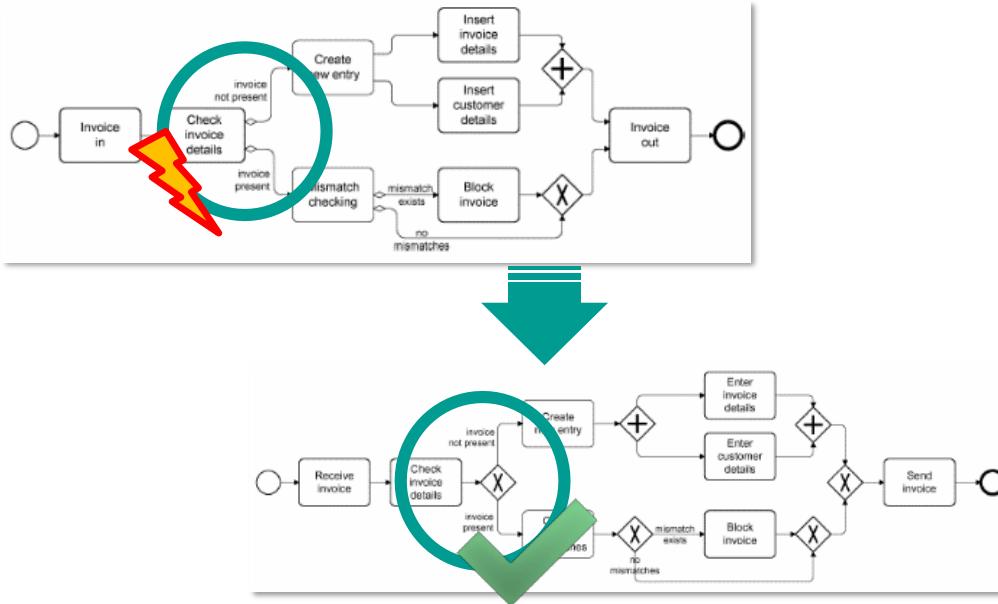
b)



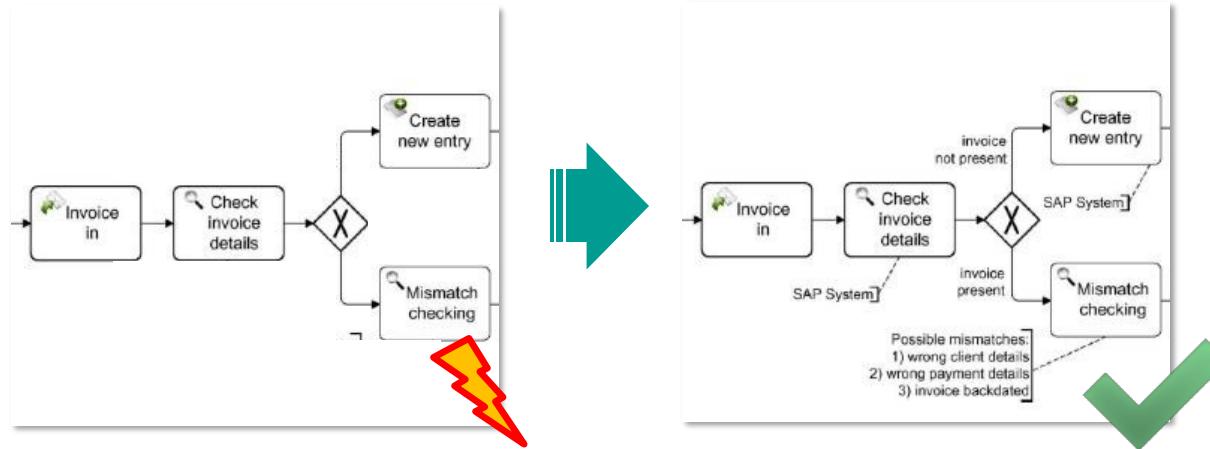
Visuell nachvollziehbarer Sequenzfluss



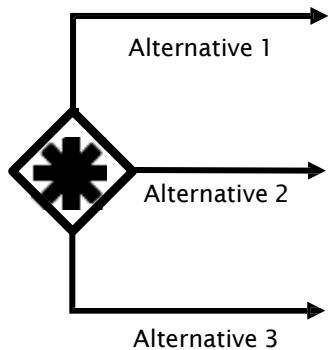
Synonyme Darstellung von Sachverhalten



Gebrauchssprachliche Komplexität kompensation



Synonyme Darstellung von Sachverhalten



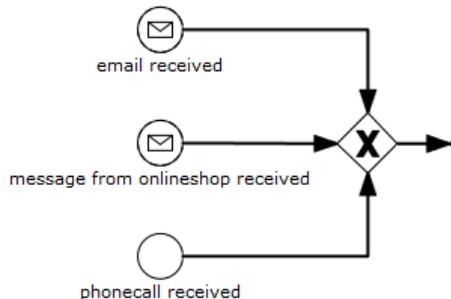
Komplexes Gateway mit zwei Alternativen.



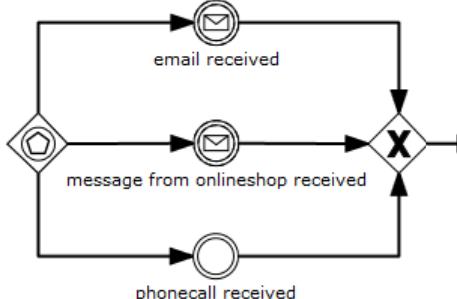
Inklusives Gateway mit zwei Bedingungen und Defaultsequenzfluss.

Synonyme Darstellung von Sachverhalten

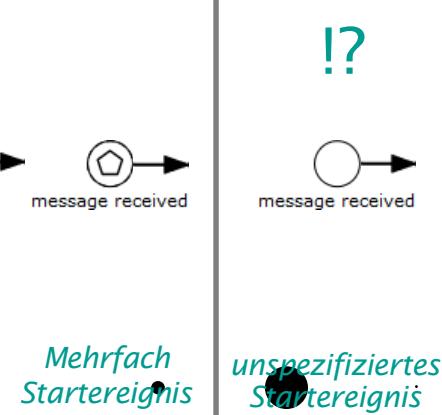
- Der Geschäftsprozess „*Reklamation*“ kann durch ein eingehendes Telefongespräch **ODER** durch eine eingehende E-Mail-Nachricht **ODER** durch eine eingehende Systemnachricht ausgelöst werden.



Unterschiedliche Startevents



Ereignis-basiertes Gateway



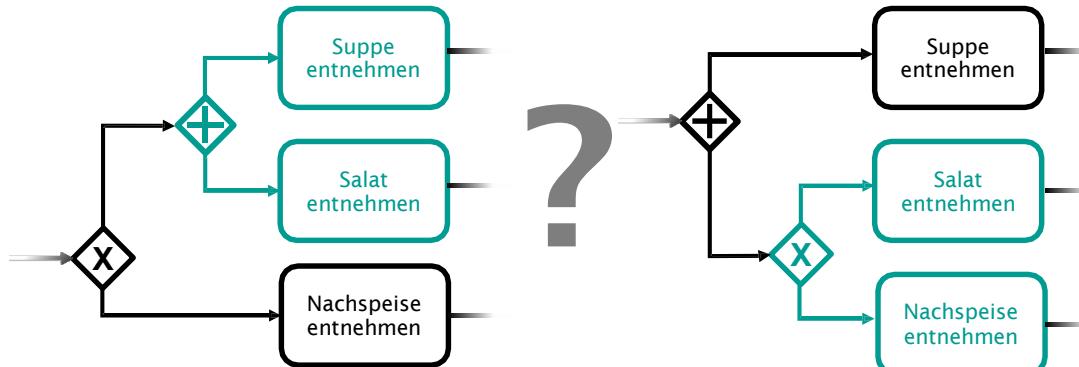
Mehrfach Startereignis

unspezifiziertes Startereignis

Problem der Mehrdeutigkeit

Interpretationsspielräume der Gebrauchssprache

...entnehme Suppe **und** Salat **oder** Nachspeise.

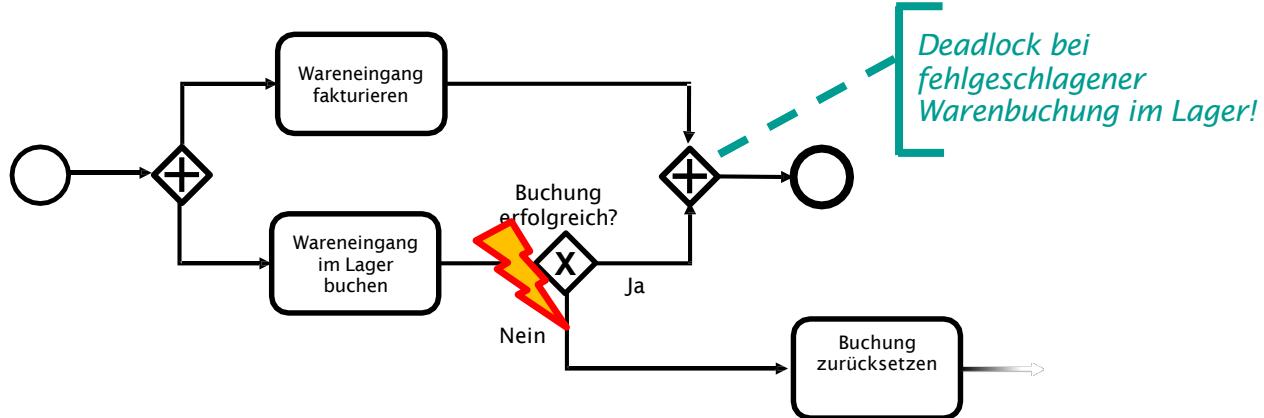


...**(Suppe und Salat) oder** Nachspeise.

...Suppe **und** **(Salat oder Nachspeise)**.

Verwendung von Gateways

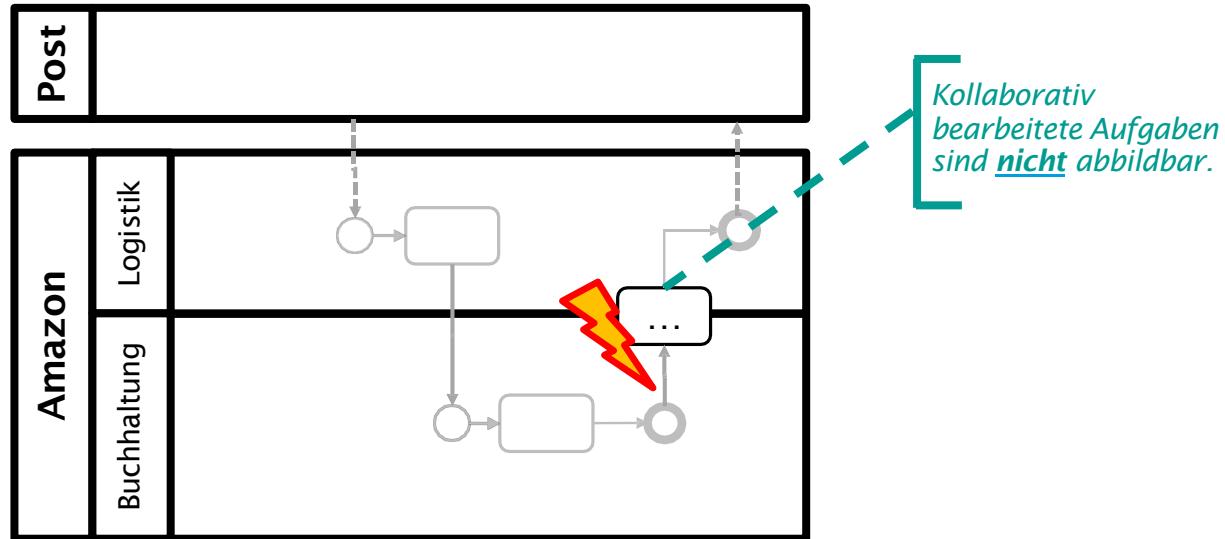
Vermeidung von „Deadlocks“ durch angeheftete Kompensationen



Pools und Lanes

Abilden von Verantwortlichkeiten

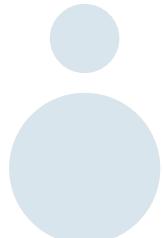
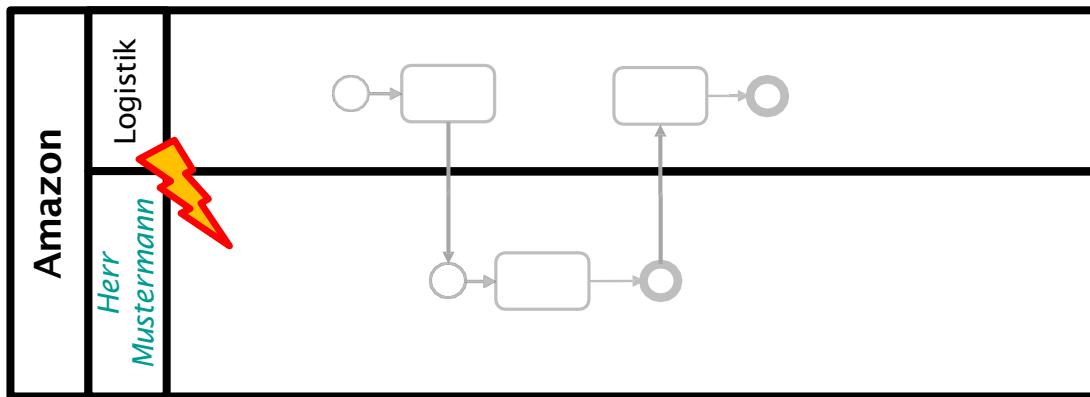
- Task **muss** eindeutig einer Lane zugeordnet werden!



Pools und Lanes

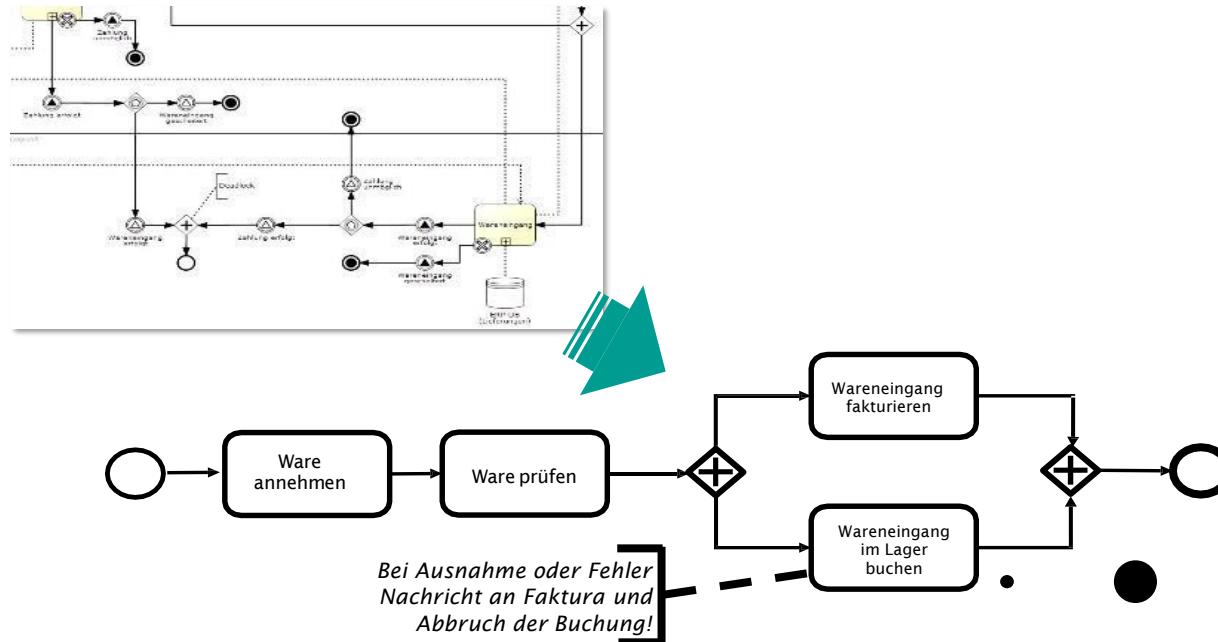
Abstraktionsebenen

- Abstraktionsebenen der Lanes nicht vorgegeben!

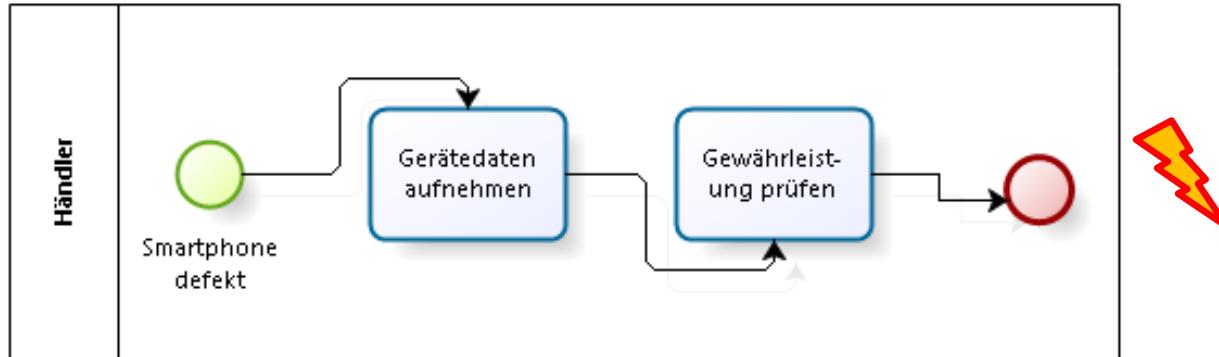


Zielgruppenorientierte Modellierung

Modellverifikation der Anwendergruppe

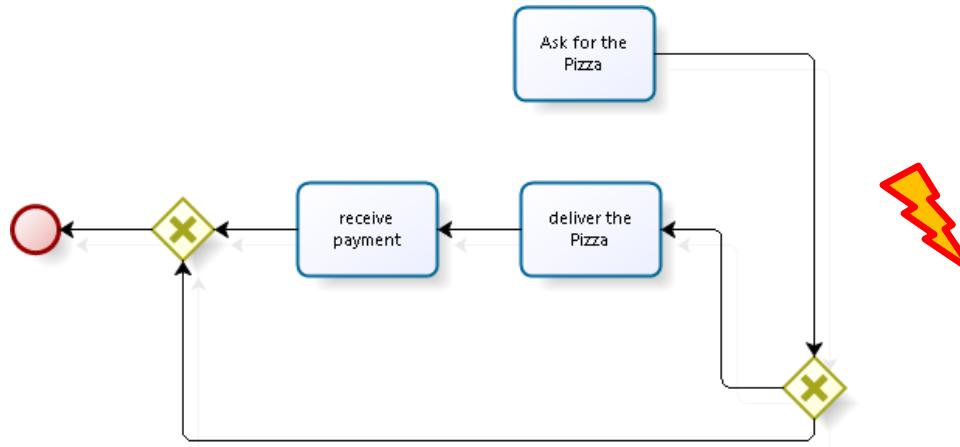


Einheitliches Eingangs- und Ausgangsverhalten von Sequenzflüssen



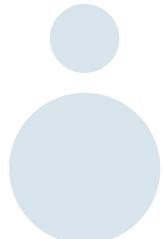
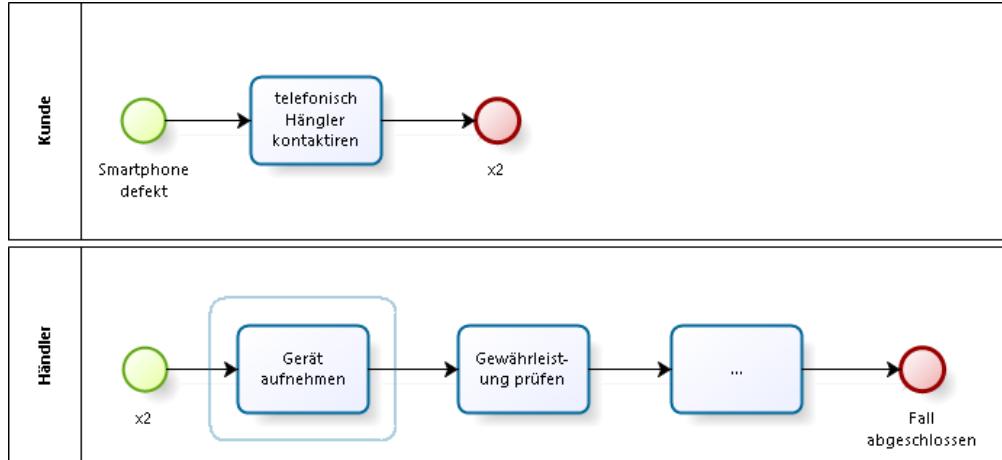
- Das einheitliche Ein- und Ausgangsverhalten von Sequenzflüssen dient der Übersichtlichkeit und zum Verständnis des Prozesses.

Einhaltung der Modellierungsrichtung bei Sequenzflüssen

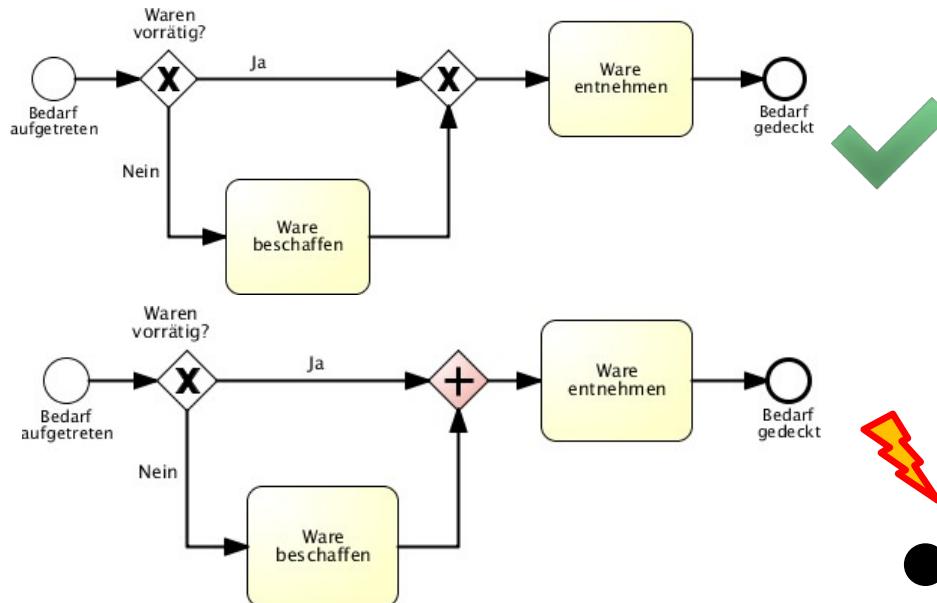


- Die Modellierung von Sequenzflüssen darf nicht entgegen der einheitlichen Richtung (definiert als Diagrammattribut) laufen. Es wird empfohlen, keine unnötigen Sequenzflüsse zu modellieren, damit das Prozessmodell so einfach wie möglich gehalten wird.

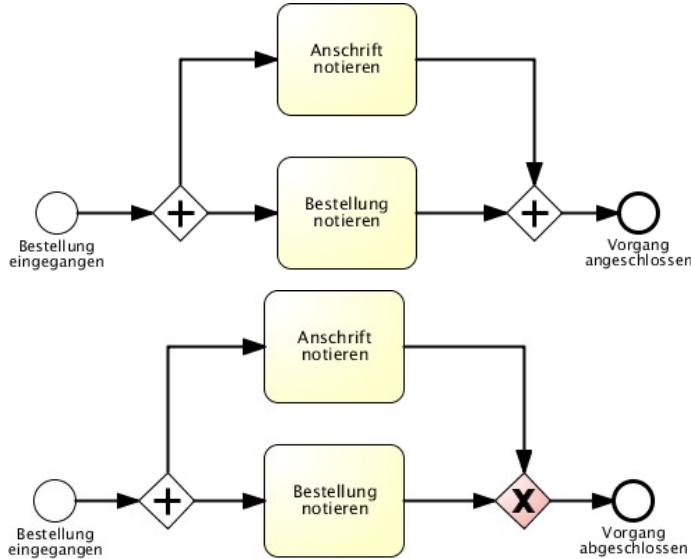
Häufige Fehler



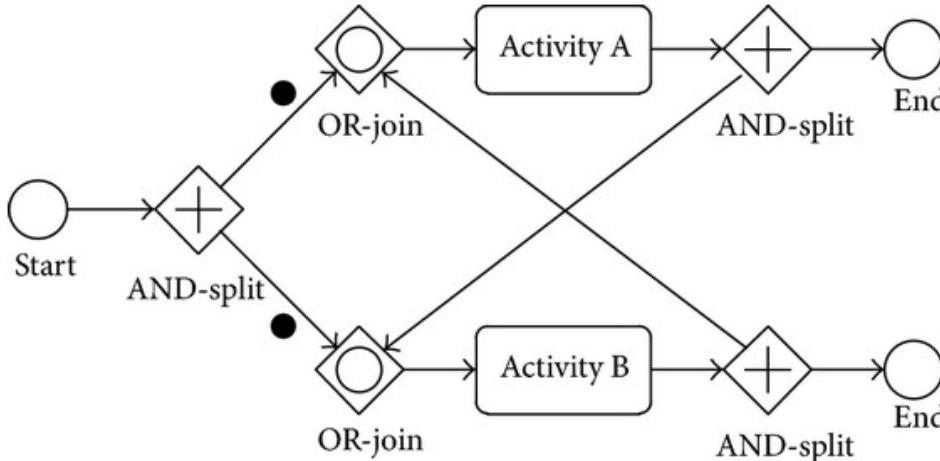
Abwesenheit von Verklemmungen (Deadlocks)



Abwesenheit von Multi-Merges



Vicious Circle

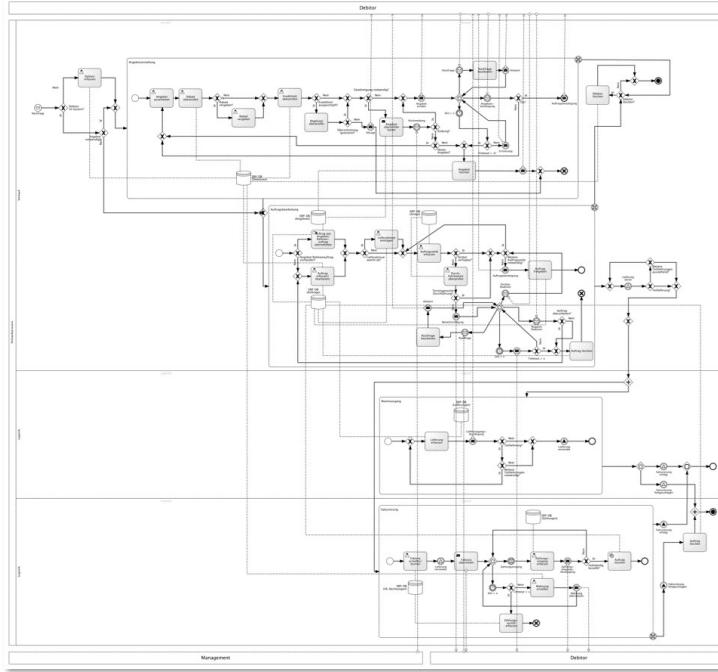


Quelle: <http://www.hindawi.com/journals/isrn/2013/507984.fig.0018.jpg> (Abruf 18.06.2015); Theoretische Grundlage siehe [7]

Aussagenorientierte Modellierung

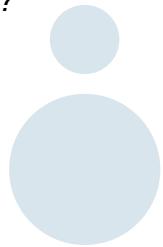
Reduzierung der Elemente

Debitor
Verkauf
Logistik 1
Logistik 2
Management



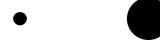
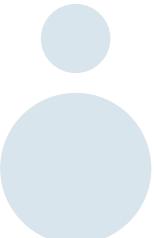
Probleme:

- *Verifikation?*
- *Optimierung?*
- *Anpassung?*
- *Abstraktion?*



Wiederholung – Wie gut hast du es verstanden?

BPMN 2.0 Quiz



Übung 1 – Prozessmodellierung

Akteure

Kunde, Händler, Werkstatt, Post

Szenario

Ein Kunde stellt fest, dass sein kürzlich erworbenes Smartphone defekt ist. Aus diesem Grund ruft der Kunde den Händler an und meldet das defekte Smartphone. Der Händler nimmt die Gerätedaten des Smartphones auf und prüft die Gewährleistung.

Falls eine Reparatur möglich ist, kontaktiert der Händler die Werkstatt und vereinbart einen Reparaturtermin. Die Werkstatt erzeugt eine RMA-Nummer mit einem Abholauftrag für das Smartphone. Sobald das Smartphone in die Werkstatt eintrifft, wird die Reparatur (als Subprozess) durchgeführt. Daraufhin wird das reparierte Smartphone mit der Post wieder zum Kunden zurückgesendet.

Falls eine Reparatur nicht möglich ist, nimmt der Händler das Smartphone entgegen und erstattet dem Kunden den gezahlten Betrag.

Übung 2 – Prozessmodellierung

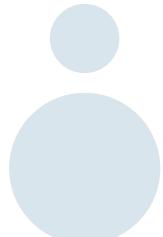
Akteure:

Fitnessstudio, Kunden

Szenario

Eine Fitnessstudio Kette, möchte im Rahmen eines Digitalisierungsprojektes den Anmeldeprozess mit Hilfe eines Workflowmanagementsystems digitalisieren. Der Kunde kann sich dann mit einem Tablet an der Theke anmelden.

Um das Anmeldeformular valide ausfüllen zu können, muss eine Kopie des Personalausweises hinzugefügt werden. Je nach Wahl von einem Abo wird im Anschluss berechnet wie hoch der monatliche Beitrag ist.



Übung 2 – Prozessmodellierung

Szenario Fortsetzung

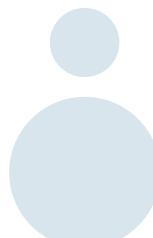
Folgende Möglichkeiten gibt es bei den Abos:

- Student (30€)
- Normal (50€)
- Premium (70€)

Da bei dem Premium Packet eine Dampfsauna und ein Solarium enthalten sind, müssen weitere Angaben zur gesundheitlichen Lage der Person getroffen werden. Hierfür füllt der Kunde ein anderes Formular aus.

Bei einem Studententarif wird eine Einverständniserklärung ausgefüllt, die das halbjährliche Vorzeigen des Studentenausweises verlangt.

Nach dem Absenden des Anmeldeformulars bekommt das Fitness eine Benachrichtigung.



Workflows

Definition und Abgrenzung

Definition Workflow



https://mind-force.de/files/2018/05/Fotolia_115511688_Subscription_XXL-660x425.jpg

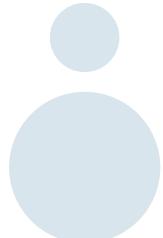
Definition Geschäftsprozess

Prozessebene:

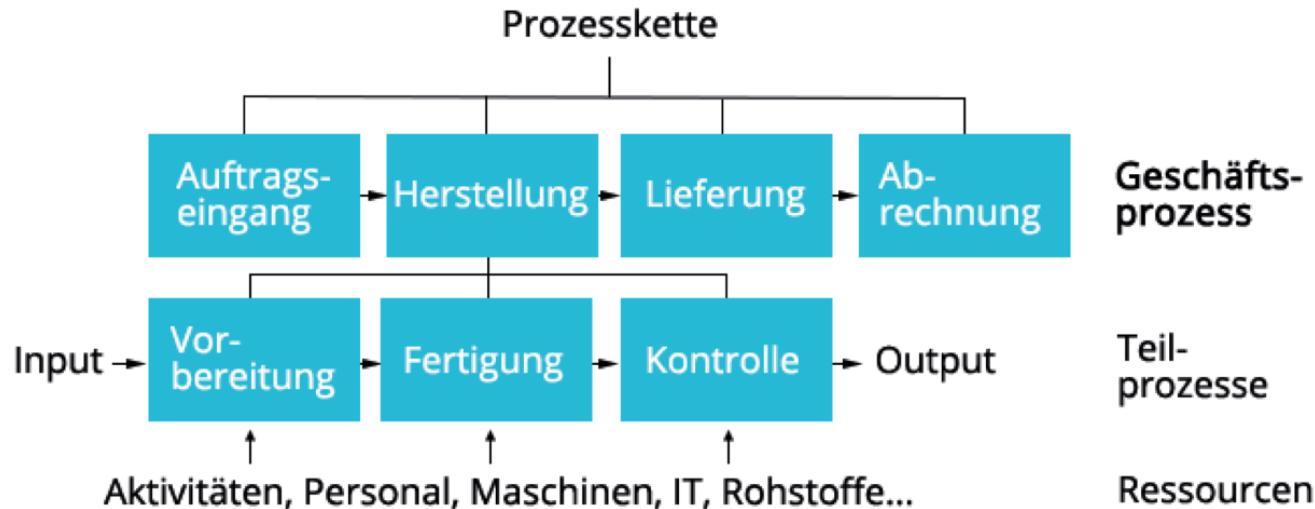
- Unterprozesse
- Start- und Endzeitpunkt
- Liefern bestimmtes Ergebnis

Aktivitätsebene:

- Unterschiedliche Aktivitäten
- Festlegung der Aktivitäten

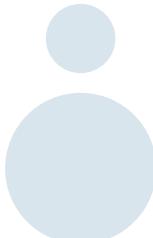


Definition Geschäftsprozess



Abgrenzung von Geschäftsprozessen und Workflows

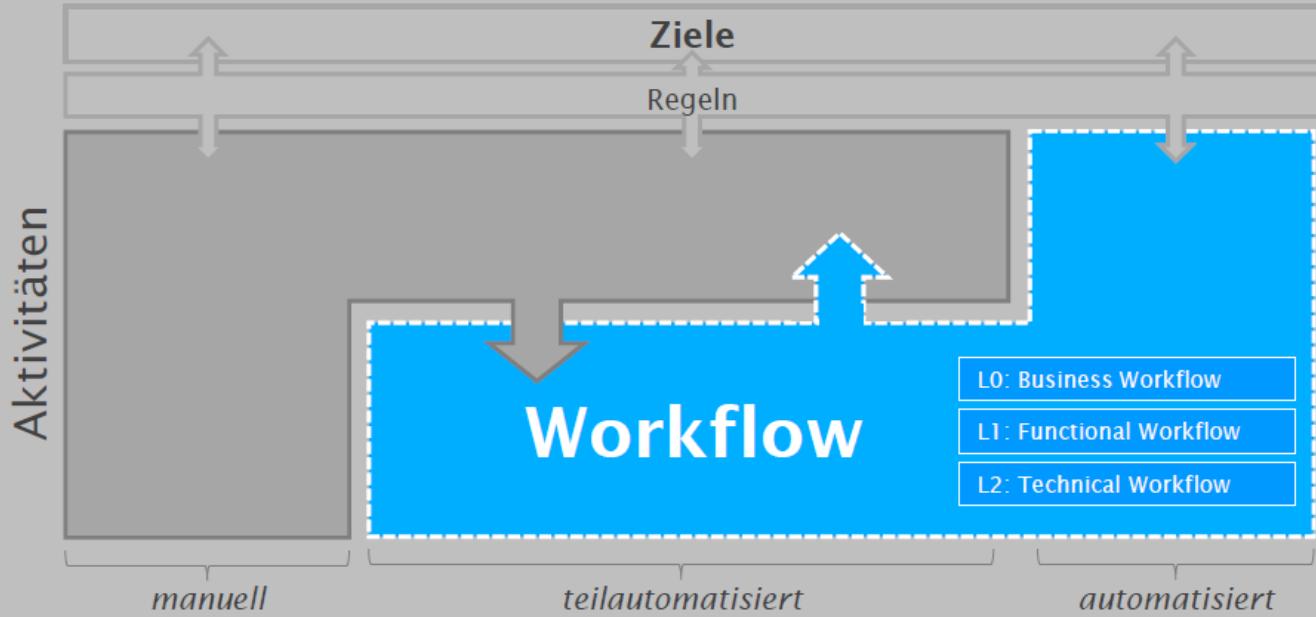
- Workflow und Prozess häufig verwechselt
- Workflow = Koordinierung einzelner Prozesse
- Workflow Teil des Geschäftsprozessmanagements
- Schwerpunkt auf Verteilung von Aufgaben



„Projekt“

„Prozess“

G e s c h ä f t s p r o z e s s

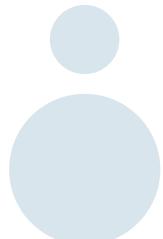


Was beschreibt die Workflow Ebene?

- Prozesse und Aktivitäten in Geschäftsprozess automatisieren
- Aktivitäten in kleinstmögliche Arbeitsschritte zerlegen
- Arbeitsschritte abbilden und abarbeiten



- Informationstechnische Unterstützung bei Geschäftsprozessen
- Workflow nur automatisierter Ablauf
- Workflow-Management initiieren, steuern und überwachen
- Dafür werden Workflow-Management Systeme genutzt
 - Axon.ivy, Camunda, etc...

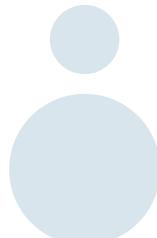


Workflow Management System

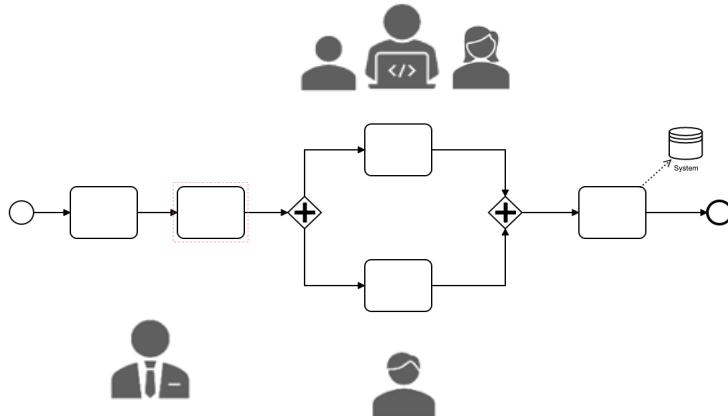
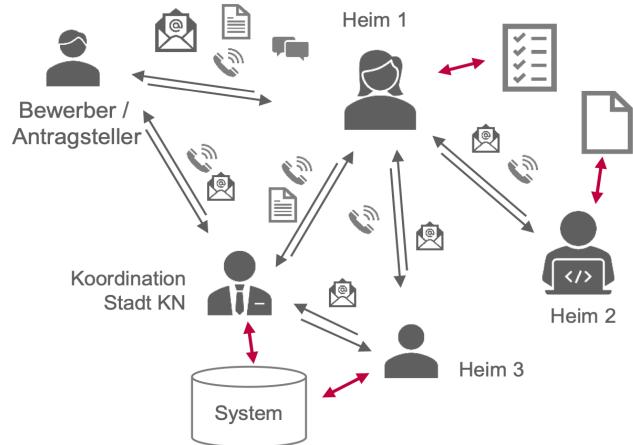


Workflow Management System

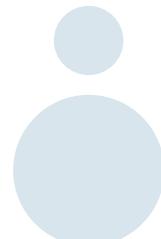
- Man hat keinen festen Ablauf
 - Variables Reagieren auf bestimmte Sachlagen
- Durch Standardfunktionen eines WfMS wird viel Arbeit und Zeit gespart
- Schaffen von klar zugeordneten Aufgaben und Entscheidungen
- Durch WfMS kann man beispielweise Papierflut verhindern und einfache Schnittstellen entwerfen
 - Bearbeitungszeit und die damit verbundenen Kosten werden verringert



Klassischer Prozess vs. Digitalisierter Prozess

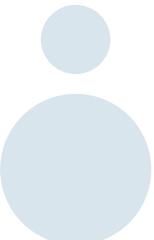


Kriterien zur Wahl des WfMS



Kriterien zur Wahl des WfMS

- Wahl Anhang von:
 - Benutzer Technologie
 - Stärke der Mitarbeiter
 - Schnittstellenanbindung
 - Benötigter Knowledge Base
 - Lizenzkosten



Vorteile von Workflows



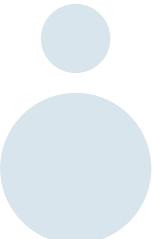
- Sicherstellung der Arbeitsabläufe
- Optimierung der Arbeitsabläufe
- Erhöhung der Transparenz
- Verbesserung der Kommunikation
- Aktuelle Informationen für bessere Entscheidungen

- Starre Vorgabe
- Ändernde Geschäftswelt
- Sinkende Motivation
- Abweichungen
- Fehler werden zugeordnet
- Mitarbeiter austauschbar



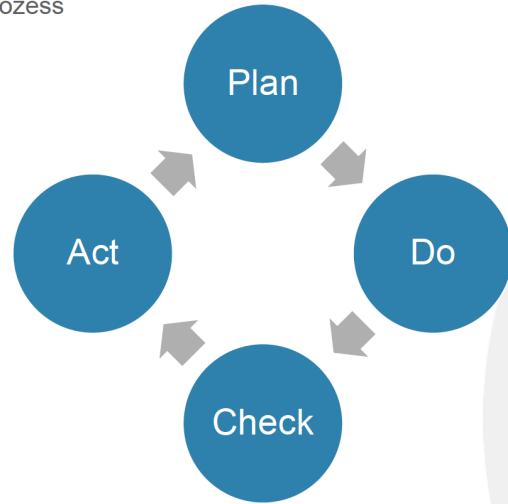
Wie erstelle ich einen Workflow?

1. Überblick verschaffen (Skizze)
2. Geschäftsprozess-Modellierung
3. Workflow-Modellierung
4. Workflow regelmäßig überprüfen und anpassen



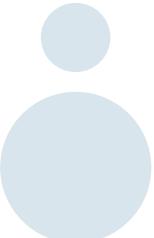
...und danach?

Der Managementprozess



Wiederholung – Wie gut hast du es verstanden?

Workflowmanagement Quiz



Camunda

Einführung

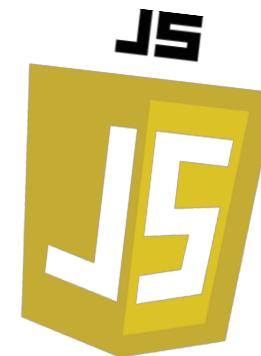
Warum Camunda?

- Open source Programm
- Vielseitiges Tool zum Modellieren und Automatisieren von Prozessen
- Umfasst BPMN und DMN Tasks
- Camunda vereint verschiedene Engines
- Lässt sich in Java implementieren
- Bekanntestes Automatisierungstool aktuell (State of the Art)

H T
W I
G N

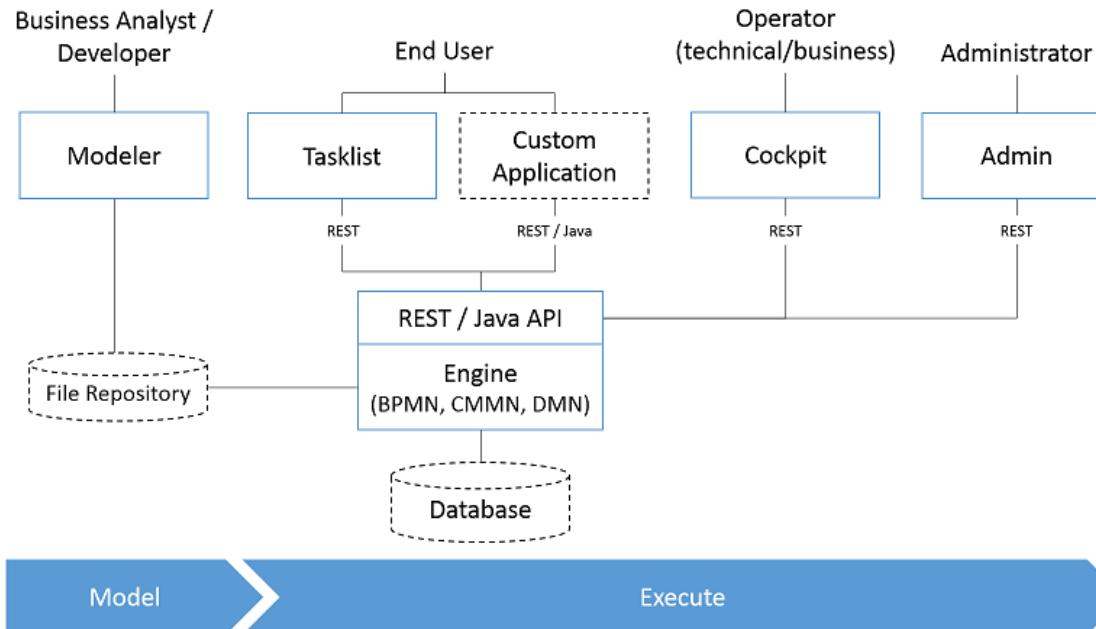
Camunda Languages

Hochschule Konstanz
Fakultät Informatik



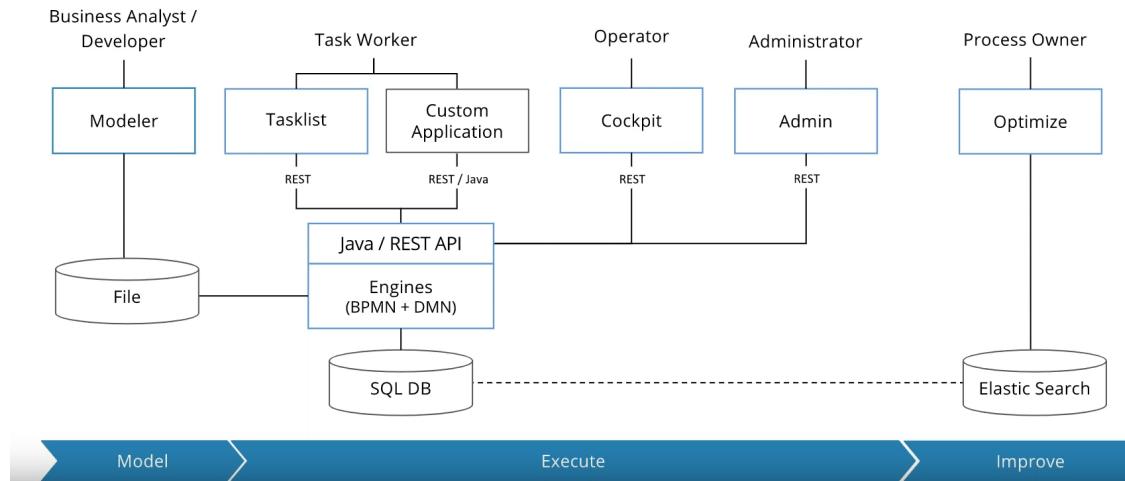
- Apache Tomcat 6.0 / 7.0 / 8.0
- JBoss Application Server 7.2 and JBoss EAP 6.1 / 6.2 / 6.3 / 6.4
- Wildfly Application Server 8.2 / 10.0
- IBM WebSphere Application Server 8.0 / 8.5 ([Enterprise Edition only](#))
- Oracle WebLogic Server 12c (12R1,12R2)

Funktionsweise von Camunda



Funktionsweise von Camunda

Camunda Stack

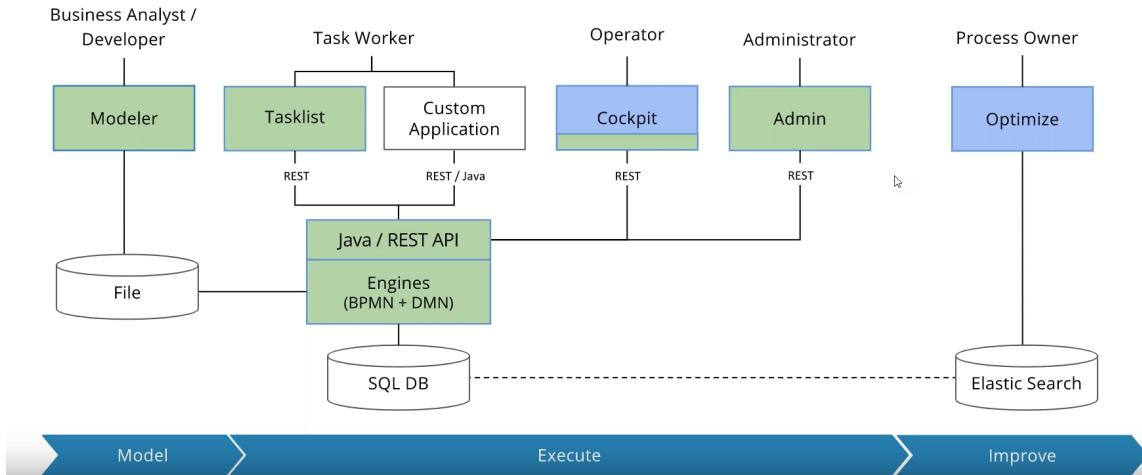


Inhalte der kostenfreien Version

Community vs. Enterprise

Community
Open Source

Enterprise
Commercial



- Die kostenfreie Version reicht für unser Vorhaben aus!

Camunda Modeler

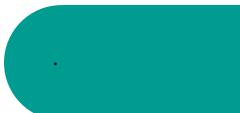
Erste Schritte

Installation Guide

1. Gehe auf die Camunda Seite
 2. Klicke auf Downloads
 3. Wähle den Camunda Modeler aus
 4. Klicke auf Download
-
- Alternativ: <https://camunda.com/download/modeler/>

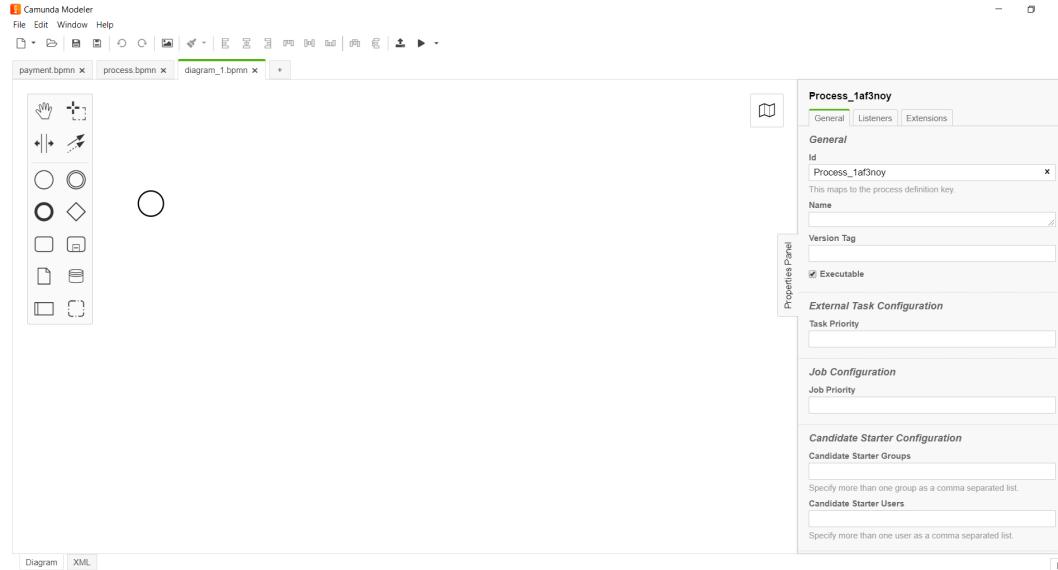
Implemented Standards

1. BPMN 2.0
2. CMMN – Case Management Model and Notation
3. DNS – Decision Model and Notation

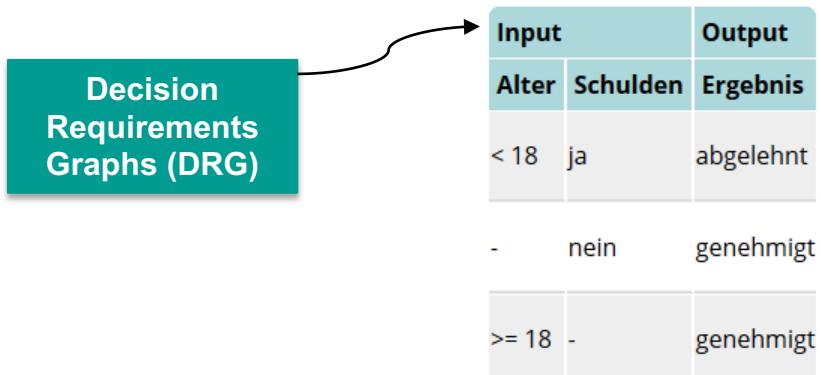


Startseite Camunda

Um das erste BPMN Model zu erstellen schaut
euch [Quickstart](#) auf der Camunda Seite an.



<https://docs.camunda.org/manual/7.5/modeler/camunda-modeler/bpmn/#quickstart>

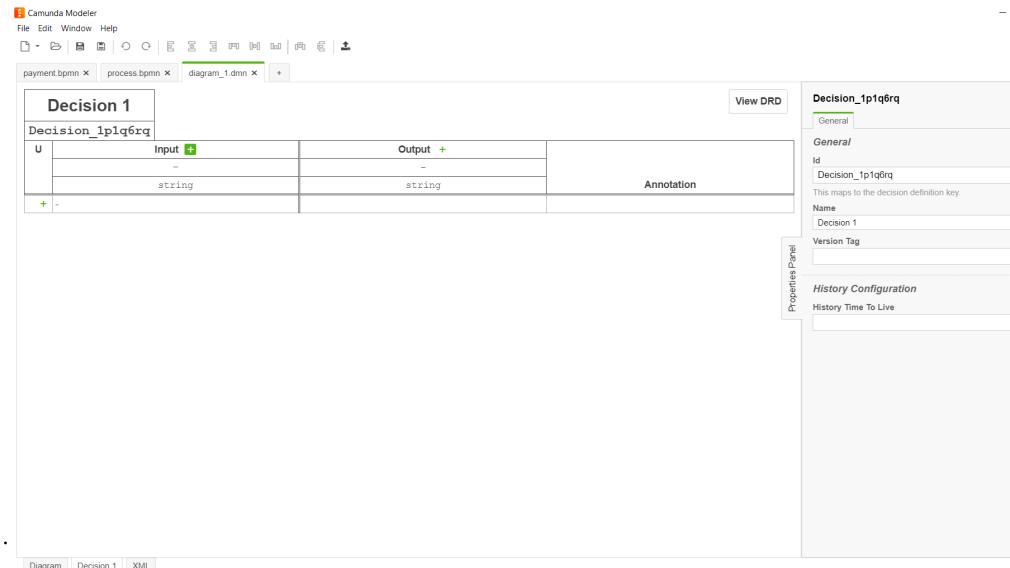


Decision Requirements Graphs (DRG)

Input		Output
Alter	Schulden	Ergebnis
< 18	ja	abgelehnt
-	nein	genehmigt
>= 18	-	genehmigt

Camunda DMN-Table Startseite

Um das erste DMN-Table zu erstellen schaut
euch [Quickstart](#) auf der Camunda Seite an.

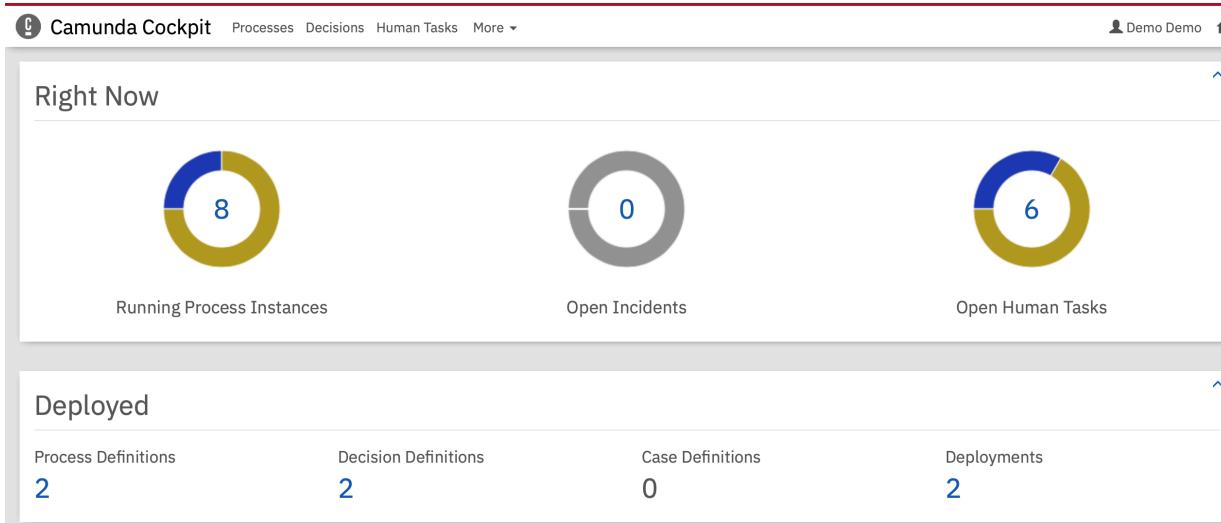


Camunda Cockpit

Komponenten und Funktionen

Camunda Cockpit

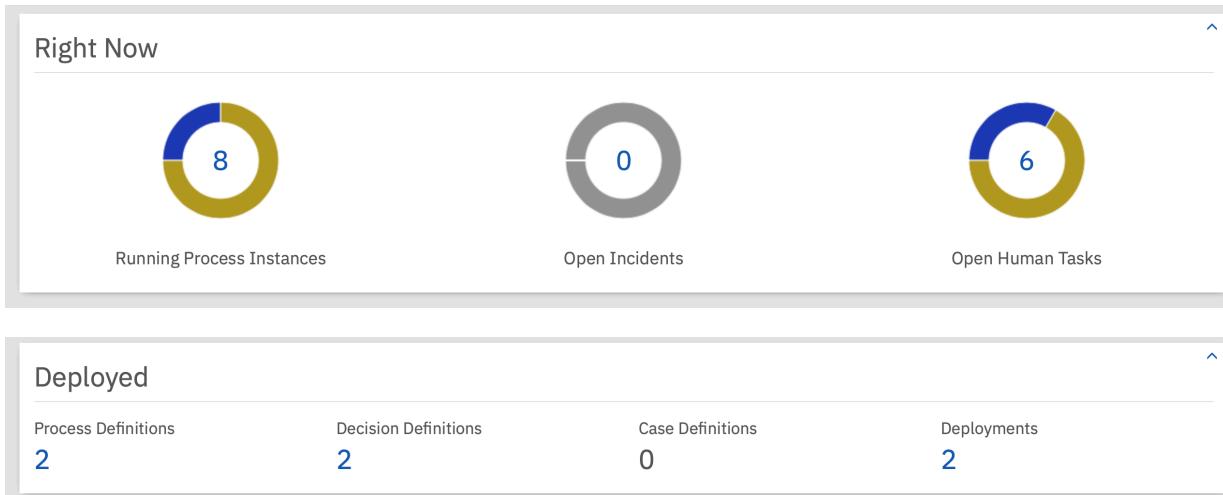
- Camunda Cockpit ist eine Web-Applikation für die Beobachtung und den Betrieb von Prozessen
- Bietet Zugriff auf BPMN Prozesse und DMN Entscheidungen
- Erlaubt es, durch laufende oder beendete Instanzen zu suchen und Operationen an diesen durchzuführen



Dashboard

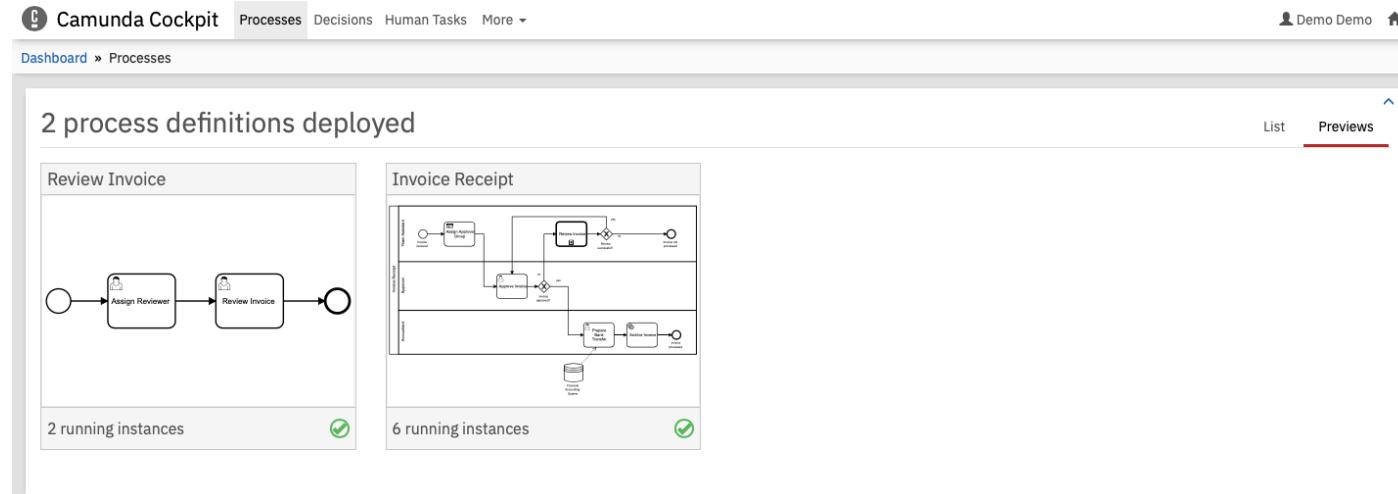
Das Dashboard ist der Einstiegspunkt für die Kontrolle der Prozesse.

Es lässt sich durch Plug-Ins individualisieren, z.B. durch eine Übersicht über alle bereitgestellten Prozesse.



Processes

Übersicht über alle bereitgestellten Prozessdefinitionen &
Beobachtung der Zustände und Anzahl der Instanzen



Process Definition View

Diese Sicht bietet Information über die Definition und den Status eines Prozesses

The screenshot shows the Camunda Cockpit interface for the "Invoice Receipt" process. On the left, a sidebar displays process metadata: Definition Version (1), Version Tag (null), Definition ID (invoice:1:7ee58b1b-8ea2-11ea-85...), Definition Key (invoice), Definition Name (Invoice Receipt), History Time To Live (null), Tenant ID (null), Deployment ID (7ead15c3-8ea2-11ea-855b-0242...), Instances Running (current version: 8, all versions: 8), and Related (Reports, Migration).

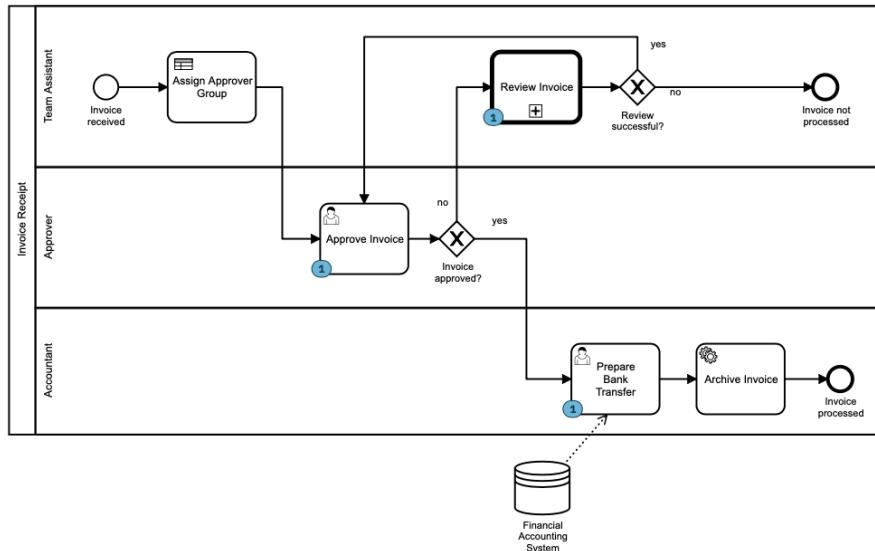
The main area features a process diagram titled "Invoice Receipt". It starts with an "Approver" activity, which leads to a decision diamond "Invoice approved?". If "no", it goes to an "Approve invoice" task. If "yes", it goes to a "Prepare Bank Transfer" task. Both lead to an "Archive invoice" task, which then ends the process. There are two counter boxes: "Instance Counter" and "Incident Counter". Below the diagram is a table of "Process Instances" with the following data:

State	ID	Start Time	Business Key
Running	89954aeb-8ea2-11ea-855b-02426f7bbacd	2020-05-05T09:32:13	
Running	8991a14c-8ea2-11ea-855b-02426f7bbacd	2020-05-05T09:32:13	
Running	898e45cd-8ea2-11ea-855b-02426f7bbacd	2020-05-05T09:32:13	
Running	898aea4e-8ea2-11ea-855b-02426f7bbacd	2020-05-05T09:32:13	
Terminated	89656160-8ea2-11ea-855b-02426f7bbacd	2020-05-05T09:32:12	
Running	896190ad-8ea2-11ea-855b-02426f7bbacd	2020-05-05T09:32:12	

Annotations: A callout box labeled "Laufende Instanzen" points to the "Instances Running" section in the sidebar. An arrow points from the "Incident Counter" box to the "Incident Counter" column in the table.

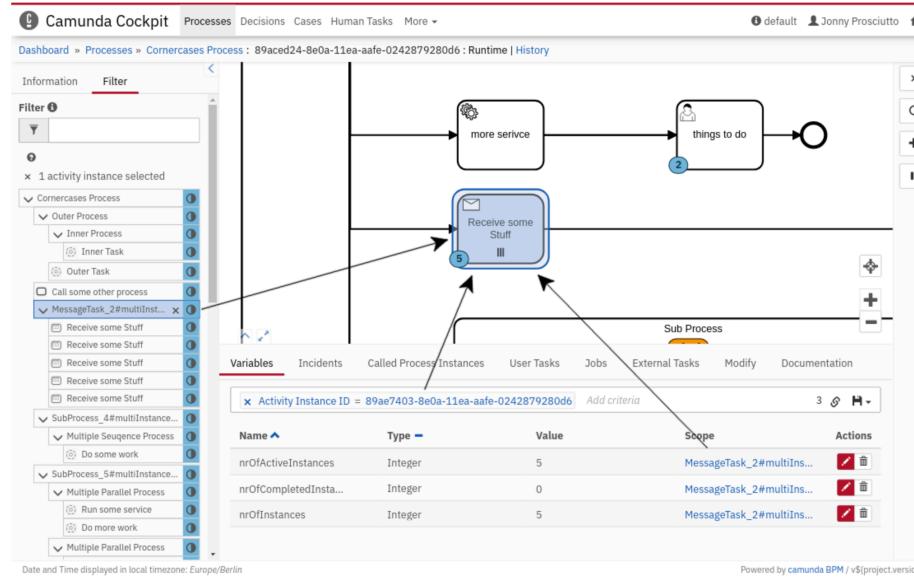
Process Definition View

- Auf der linken Seite kann man die Versionen des Prozesses auswerten und sehen, wie viele Instanzen der Versionen momentan laufen
- Vorfälle in den laufenden Prozessinstanzen werden durch einen Instanz-Zähler im zugehörigen Diagramm dargestellt, sodass es einfach ist, fehlgeschlagene Aktivitäten zu lokalisieren



Process Instance View

Diese Sicht erlaubt es, eine einzelne Prozessinstanz aufzufächern und dessen laufende Aktivitäten, sowie Variablen, Aufgaben, etc. zu erkunden



Add, Edit and Filter Variables

The screenshot shows a user interface for managing variables in a process instance. It consists of two main sections and a modal dialog.

Top Section: This section displays a table of variables for a specific process instance. A button labeled "Variablen editieren" (Edit Variables) is highlighted with a red box and a black arrow pointing to it from the top right.

Name	Type	Value	Scope	Actions
amount	Double	20.5	Invoice Receipt	
approverGroups	Object	java.util.ArrayList	Invoice Receipt	
creditor	String	Fruits Inc.	Invoice Receipt	
invoiceCategory	String	Travel Expenses	Invoice Receipt	
invoiceDocument	File	Download	... Invoice Receipt	
invoiceNumber	String	GREEN-14492	Invoice Receipt	

Bottom Section: This section displays a table of variables filtered by the search term "amount". A button labeled "Variablen filtern" (Filter Variables) is highlighted with a red box and a black arrow pointing to it from the bottom right.

Name	Value	Type	Scope	Actions
invoiceNumber	GREEN-14492	String	Invoice Receipt	
invoiceDocument	...	File	Invoice Receipt	
amount	20.5	Double	Invoice Receipt	
creditor	Fruits Inc.	String	Invoice Receipt	
invoiceCategory	Travel Expenses	String	Invoice Receipt	
approverGroups	java.util.ArrayList	Object	Invoice Receipt	

Modal Dialog: This dialog is titled "Add Variable to Process Instance". It contains fields for "Variable Name*", "Variable Type*", and "Variable Value*". The "Variable Name*" field has "amount" typed into it. A "Close" button and an "Add" button are at the bottom right.

Process History

In der History-Sicht der Prozessdefinition kann eine Übersicht über alle laufenden und beendeten Prozessinstanzen eingesehen werden

Camunda Cockpit Processes Decisions Cases Human Tasks More ▾ default Jonny Prosciutto Home ▾

Dashboard > Processes Invoice Receipt : History | Runtime

Definition Version:
1

Definition ID:
invoice:1:83b97d18-8e0a-11ea-aafe...

Definition Key:
invoice

Definition Name:
Invoice Receipt

History Time To Live:
null

Tenant ID:
null

Deployment ID:
838c7970-8e0a-11ea-aafe-0242879...

Instances Running:

- current version: 8
- all versions: 8

Related:

- Reports
- Migration

Time Period: Today
Heatmap: on

State	ID	Start Time	End Time	Business Key
running	8c8847c1-8e0a-11ea-aafe-0242879280d6	2020-05-04T15:24:14		
running	8c849e22-8e0a-11ea-aafe-0242879280d6	2020-05-04T15:24:14		
running	8c8190c3-8e0a-11ea-aafe-0242879280d6	2020-05-04T15:24:14		
running	8c7e5c54-8e0a-11ea-aafe-0242879280d6	2020-05-04T15:24:14		
completed	8c753554-8e0a-11ea-aafe-0242879280d6	2020-05-04T15:24:14	2020-04-20T15:24:14	
has	8c604d76-8e0a-11ea-aafe-0242879280d6	2020-05-04T15:24:14		

Date and Time displayed in local timezone: Europe/Berlin Powered by camunda BPM / v\${project.version}

Failed Jobs

Ungelöste Vorfälle von Prozessinstanzen oder Unterprozessinstanzen werden im Cockpit als failed jobs angegeben

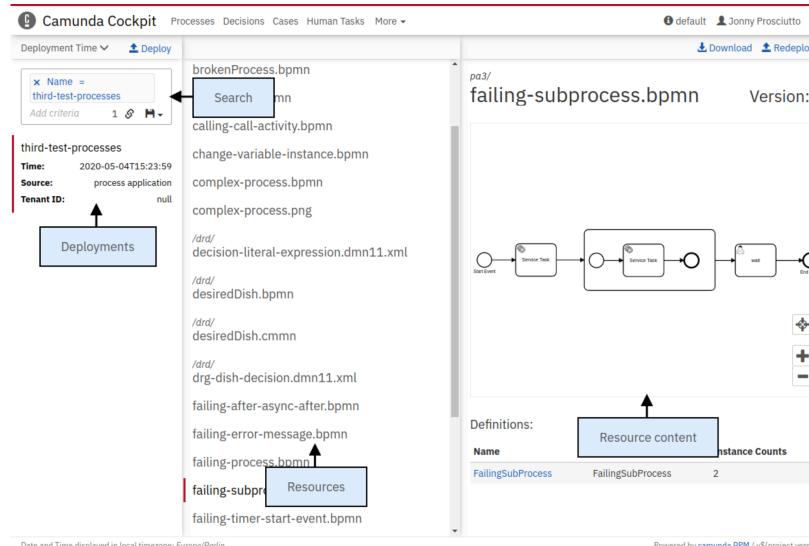
The screenshot shows the Camunda Cockpit interface. On the left, a sidebar displays process definitions and runtime statistics. The main area shows a table of failed process instances. One instance, 'FailingProcess' (Definition ID: FailingProcess:1:83b955fe-8e0a-1...), is highlighted with a red border. A tooltip 'FailingSubProcess' points to its status. The table has columns for Message, Timestamp, Activity, Failing Activity, Cause Process Instance ID, Root Cause Process Instance ID, Type, and Action. One row in the table is also highlighted with a red border, showing the message 'I am failing!', timestamp '2020-04-15T21:05:51', activity 'Service Task', failing activity 'Service Task', cause process instance ID '8971919f-8e0a-11ea-aafe-0242879280d6', root cause process instance ID '89733f61-8e0a-11ea-aafe-0242879280d6', type 'Failed Job', and action 'C'. Arrows point from the text 'Ungelöste Vorfälle von Prozessinstanzen oder Unterprozessinstanzen werden im Cockpit als failed jobs angegeben' to the highlighted rows in the table.

Message	Timestamp	Activity	Failing Activity	Cause Process Instance ID	Root Cause Process Instance ID	Type	Action
I am failing!	2020-04-15T21:05:51	Service Task	Service Task	8971919f-8e0a-11ea-aafe-0242879280d6	89733f61-8e0a-11ea-aafe-0242879280d6	Failed Job	C
				89749f06-8e0a-11ea-aafe-0242879280d6			
				8976c201-8e0a-11ea-aafe-0242879280d6			
				8976e918-8e0a-11ea-aafe-0242879280d6			

Deployment View

Übersicht über alle Deployments, dessen Ressourcen und dem Inhalt dieser Ressourcen.

Es erlaubt einem das Löschen existierender und dem Kreieren neuer Deployments,
aber auch das Redeployment alter Ressourcen,



Open Tasks Dashboard

Das Open Tasks Dashboard kann genutzt werden,
um zu sehen wie die offenen Tasks durch individuelle Faktoren verteilt sind

The screenshot shows the Camunda Cockpit Human Tasks dashboard. At the top, there is a navigation bar with links for Processes, Decisions, Cases, Human Tasks (which is highlighted with a red box), and More. Below the navigation is a breadcrumb trail: Dashboard > Human Tasks.

Assignments by type: This section contains a summary table with the following data:

Tasks Types
135 assigned to a user
10 assigned to 1 or more groups
879 unassigned
1024 Total

A callout box with the text "Wie sind die Tasks allgemein verteilt?" points to this section.

Assignments by group: This section contains a summary table with the following data:

Tasks Group
879 without group
7 Accounting
3 Management
5 Sales

A callout box with the text "Wie sind die Tasks innerhalb der Gruppen verteilt?" points to this section.

Search Human Tasks: This section features a search bar with the placeholder text "Filtern nach spezifischen Tasks." and a link "Add criteria". Below the search bar is a table listing tasks:

Task ID	Task Name	Start Time	End Time	Instance ID	Definition Key
8ca593cf-8e0a-11ea-aafe-0242...	make dish	2020-05-04T15:24...		8ca4a954-8e0a-11ea-aafe...	desiredDish
8ca3e603-8e0a-11ea-aafe-0242...	make dish	2020-05-04T15:24...		8ca2fb88-8e0a-11ea-aafe...	desiredDish
8ca25f47-8e0a-11ea-aafe-0242...	make dish	2020-05-04T15:24...		8ca126ac-8e0a-11ea-aafe...	desiredDish
8c9ae5f2-8e0a-11ea-aafe-0242...	Approve Invoice	2020-05-04T15:24...		8ca9a4996-8e0a-11ea-aafe...	invoice

Callout boxes with the text "Search Human Tasks" point to the search bar and the table header.

Camunda Tasklist

Komponenten und Funktionen

Tasklist

Auf dem Dashboard der Tasklist kann eine Übersicht über die ausstehenden Tasks erhalten werden

The screenshot displays the Camunda Tasklist application interface. On the left, a sidebar titled "Camunda Tasklist" shows a list of tasks under "My Tasks" and "Soon due tasks (30)". A blue box labeled "Filters" with an upward arrow points to the sidebar. A blue box labeled "Filter Results" with a downward arrow points to a dropdown menu in the top right. The main area is titled "Prepare Bank Transfer" and contains a sub-section "Invoice Receipt". This section includes a "Set follow-up date" button (7 days ago), a "Claim" button, and a "Task View" button, all enclosed in a blue box with a downward arrow. Below this is a text input field: "Please prepare the bank transfer for the following invoice". To the right, there is a table with fields: "Invoice Document" (invoice.pdf), "Creditor" (Bobby's Office Supplies), "Amount" (900), "Invoice Number" (BOS-43934), and "Approved by" (demo). At the bottom are "Save" and "Complete" buttons.

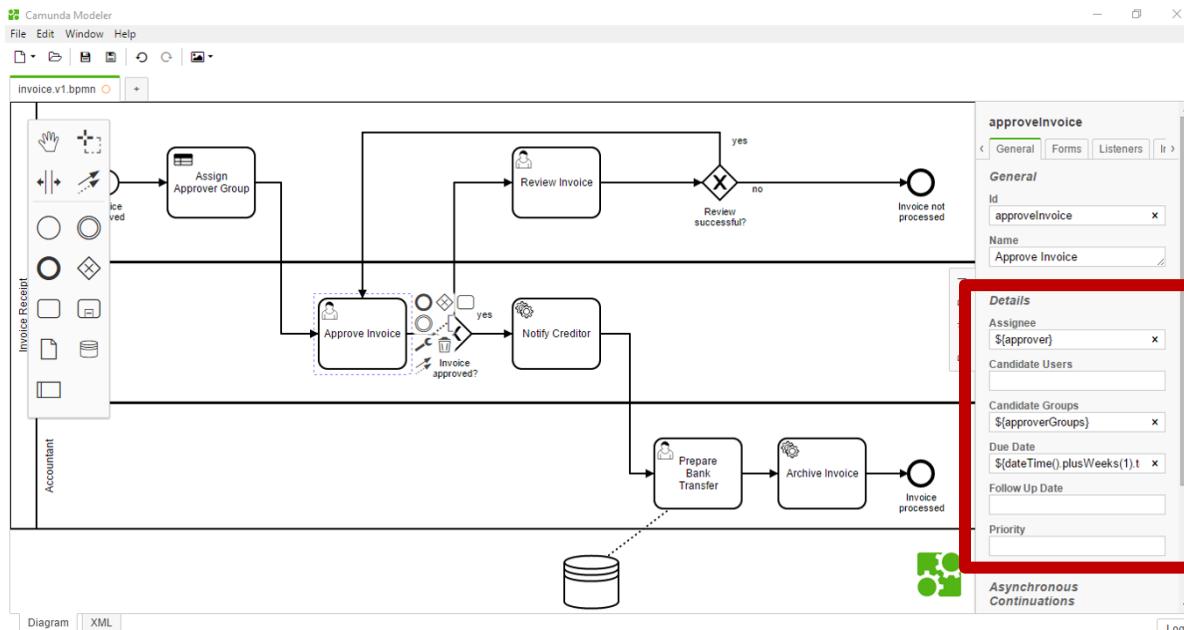
Wichtige Komponenten der Tasklist

The diagram illustrates four key components of the Camunda Tasklist:

- Start process**: A screenshot of the "Start process" screen. It shows a search bar "Search by process name." and a list of available processes: Another Failing Process, Call Case Process, CallActivity, CallingCallActivity, Failing Error Message, Failing Timer Start Event, FailingProcess, FailingSubProcess, and Invoker recent Generated Firms. A large black arrow points from this screen to the "Prozess starten" box.
- Create task**: A screenshot of the "Create task" dialog. It includes fields for Name (Send Invoice), Assignee (John), Tenant ID, and Description (Invoice to be sent by mail). A large black arrow points from this screen to the "Task erstellen" box.
- Ergebnisse filtern**: A screenshot of the "Camunda Tasklist" interface showing a filtered list of tasks under "My Tasks". The filter criteria are "Created > + 1 day" and "Priority: High". A large black arrow points from this screen to the "Ergebnisse filtern" box.
- Suche, Task View und Task Details**: A screenshot of the "Camunda Tasklist" interface showing a detailed view of a task titled "Approve Invoice". It includes fields for "Invoice Receipt" (checkbox), "Set follow-up ...", "Creditor" (Fruits Inc.), "Amount" (20.5), "Invoice Category" (Travel Expenses), and "Invoice Number" (GREEN-14492). A large black arrow points from this screen to the "Suche, Task View und Task Details" box.

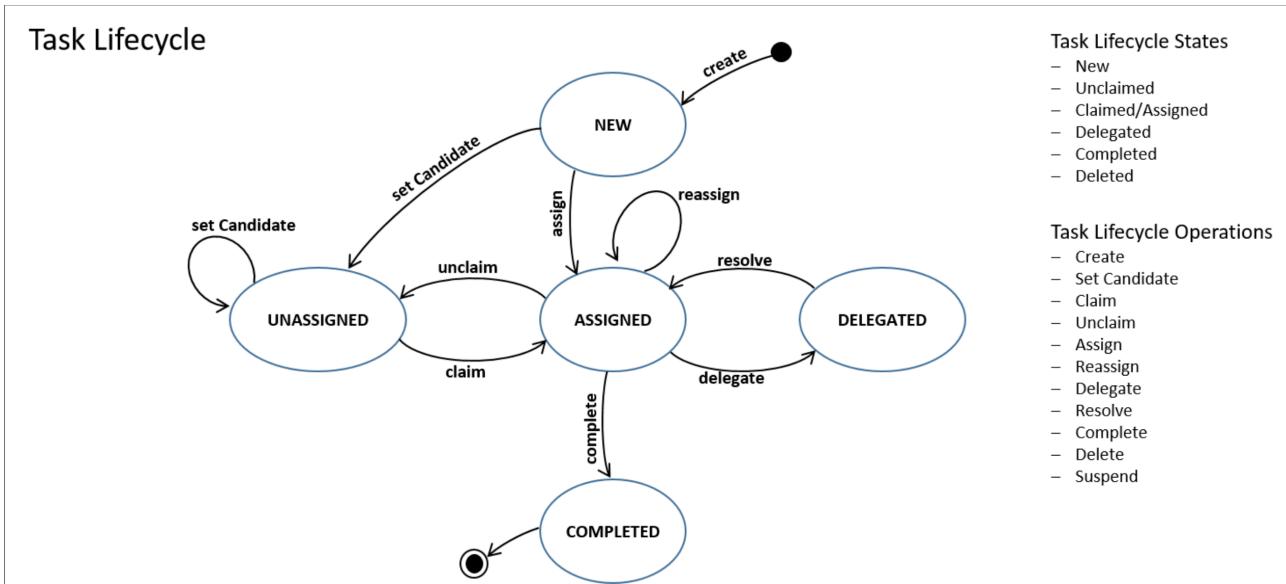
User Assignment

Damit die User wissen, an welchen Tasks sie arbeiten müssen, müssen diese entsprechend zugewiesen werden, damit diese einfacher auffindbar sind



Task Lifecycle

Task Lifecycle zeigt an, welche Übergänge von Camunda unterstützt werden

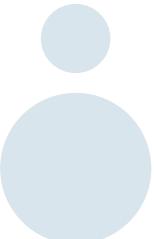


Was ist in der Praxis zu beachten?

Clean Code und Unit Tests

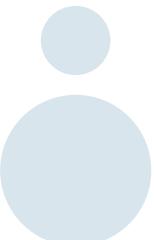
Clean Code

- Durch bestimmte Prinzipien schafft man:
 - Klare Lesbarkeit von geschulten Entwicklern
 - Ersichtlicher Zusammenhang
 - Verständlichkeit von Aufgaben der einzelnen Klassen



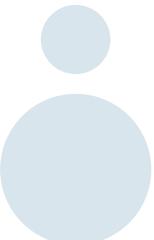
Prinzipien von Clean Code

- Klassen und Methoden werden klein geschrieben
- Klassen und Methoden haben eine eindeutige Aufgabe, welche durch die Benennung vorhersehbar ist
- Benennung von Klassen, Methoden und Variablen soll verständlich sein
- Klassen werden groß geschrieben
- Der Code wurde ausreichend getestet (JUnit Tests sind vorhanden)
- Strings und Nummern, die häufig benutzt werden als Konstanten deklarieren



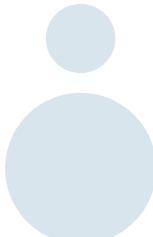
Weitere Clean Code Prinzipien

- „Keep it simple“ -> Der Code soll so einfach wie möglich sein
 - Unnötige Komplexität soll vermieden werden
- Es sollen keine Dopplungen im Code auftauchen
- Alles was nicht gebraucht wird -> Löschen



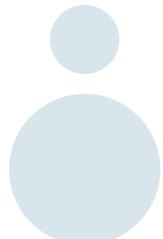
Continuous Integration

- Durch “Continuous Integration“ schafft man es die nur funktionierende Bausteine in sein Programm einzubauen
- Sehr wichtiger Schritt in der Agilen Arbeitsweise
 - Agil = inkrementell, iterativ, empirisch
- Vor allem um iterativ und in Inkrementen zu arbeiten hilft die „Continuous Integration“



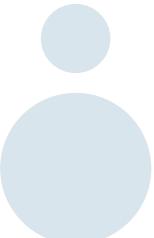
Ablauf der Continuous Integration

1. Es erfolgt die lokale Entwicklung eines “Bausteins”
2. Versionierung durch Git
3. Git mit webbasiertem Onlinedienst zur Versionsverwaltung verbinden (Bitbucket, Github, etc.)
4. Durch lokales Git Änderungen auf Versionsverwaltung „pushen“
5. Dann erfolgt ein automatischer Build über Jenkins
 - Alle Tests und Änderungen werden überprüft
6. Nach erfolgreichem Build + Analyse erfolgt das Deployment



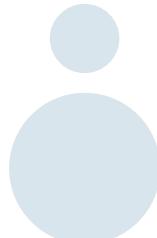
Unit Tests

- Was machen Unit Tests? -> Sie testen Spezialfälle
- Die einzelnen Unit Tests sind isoliert und unabhängig voneinander
- Diese Tests sind voll automatisiert und testen immer nur eine Funktion
- Ergebnis: Erfolgreich oder Fehlgeschlagen
- Frameworks: [Junit](#), AssertJ, Mockito, PowerMock, Hamcrest



Ablauf beim Schreiben von Tests

- Der Test wird zur “passenden” Funktion geschrieben
- Ausführen des Tests
- Bei fehlschlagen: Code überarbeiten und erneutes Testen
- Führe dies für jede Funktion aus, bis alle Tests erfolgreich sind

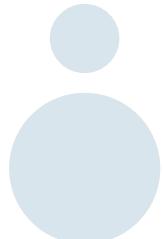


Ab in die Praxis

Das Erstellen einer Camunda Applikation

Einrichten von Camunda mit Eclipse

1. Eclipse herunterladen (Version: 2020-06)
2. Vorgefertigtes Camunda Project in Eclipse importieren
 - GitHub: <https://github.com/furkanyuecel/Workflow2Go>
 - Alternativ: <https://start.camunda.com>
3. Im Projekt Rechtsklick auf pom.xml
 - Run As
 - 5 Maven Install
4. Im Projekt Run Applikation.java
5. Auf <http://localhost:8080/> einloggen
 - Im vorgefertigten Projekt Username und Passwort: demo/demo



Alternative: Camunda BPM Initializr

Camunda BPM Initializr

Group com.example.workflow	Artifact my-project
Camunda BPM Version 7.13.0	Spring Boot Version 2.2.5.RELEASE
H2 Database On-Disk	Java Version 8
Camunda BPM Modules	Spring Boot Modules
<input checked="" type="checkbox"/> REST API 	<input type="checkbox"/> Security
<input checked="" type="checkbox"/> Webapps 	<input type="checkbox"/> Web
<input type="checkbox"/> Assert 	
Admin Username demo	Admin Password ****

GENERATE PROJECT **EXPLORE PROJECT**

Privacy Statement | Imprint | Camunda Services GmbH © 2019 - 2020

The screenshot shows the Eclipse IDE interface with a dark theme. The left side features the Package Explorer view, which lists the project structure: my-project, src/main/java (containing com.example.workflow), src/main/resources, JRE System Library [JavaSE-1.8], Maven Dependencies, bin, src, target, and pom.xml. The Application.java file is selected in the Package Explorer.

The central area contains the code editor with the following content:

```
1 package com.example.workflow;
2
3+ import org.springframework.boot.SpringApplication;[]
4
5 @SpringBootApplication
6 public class Application {
7
8     public static void main(String[] args) {
9         SpringApplication.run(Application.class);
10    }
11 }
12
13 }
```

The bottom right corner of the code editor shows a small icon of a person with a gear, indicating a Mylyn task.

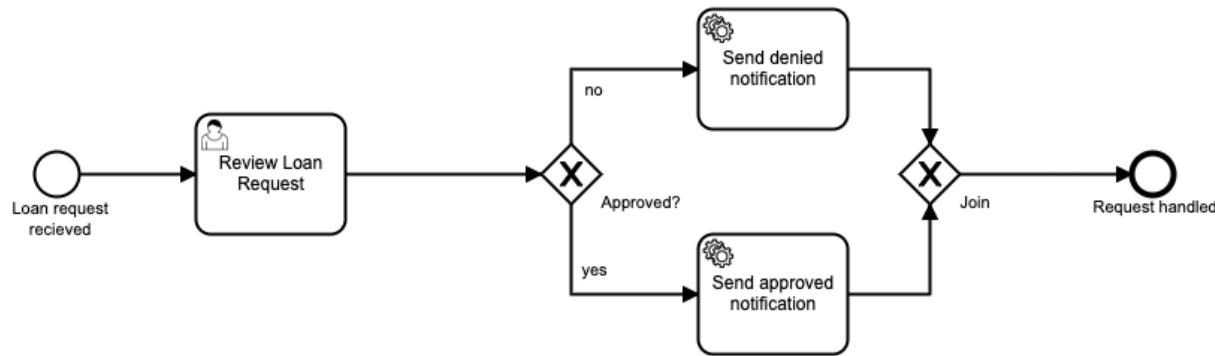
The bottom of the screen displays the Eclipse status bar with the text "com.example.workflow.Application.java - my-project/src/main/java".

The right side of the interface includes several toolbars and views:

- Task List**: Shows a single entry: "Connect Mylyn" with the sub-instruction "Connect to your task and ALM tools or [create](#) a local task."
- Outline**: Shows the class structure: com.example.workflow, Application, and main(String[]) : void.

The bottom of the interface has a toolbar with various icons for navigating and managing the workspace.

Beispielprozess: Das Bearbeiten einer Leihanfrage



Folgendes ist zu beachten

Den User Task der richtigen Person zuordnen

Details

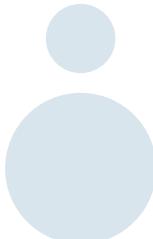
Assignee

x

Candidate Users

Candidate Groups

Due Date



Folgendes ist zu beachten

Die HTML-Formen an das Startereignis und den User Task binden

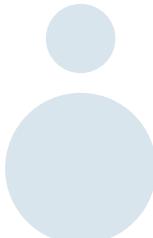
Forms

Form Key

embedded:app:forms/reviewLoanRequest.html x

Form Fields

x +



Folgendes ist zu beachten

Ein Element der HTML-Form hat folgenden Aufbau

```
<form role="form" class="form-horizontal">
  <div class="control-group">
    <label class="control-label">Name</label>
    <div class="controls">
      <input type="text" class="form-control"
        cam-variable-name="name"
        cam-variable-type="String"
        required
      />
    </div>
  </div>
```

Folgendes ist zu beachten

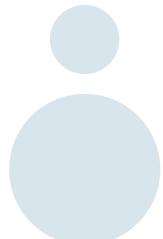
Die Java-Klassen per Delegate Expression an die Service Tasks binden

Details

Implementation

Delegate Expression

#{deniedAdapter}



Folgendes ist zu beachten

Beispiel für die Java Klasse

```
@Service("approvedAdapter")
public class LoanRequestApprovedDelegate implements JavaDelegate {

    public void execute(DelegateExecution execution) throws Exception {
        String name = (String) execution.getVariable("name");
        String comments = (String) execution.getVariable("comments");

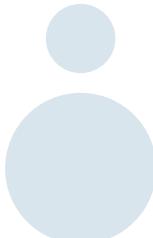
        System.out.println("\n\n#####\n");
        System.out.println("Name: " + name + "\n");
        System.out.println("Your loan has been approved: " + "\n");
        System.out.println("Comment: " + comments + "\n");
        System.out.println("\n\n#####\n");
    }
}
```

Camundathon

Die Nacht der Modellierung

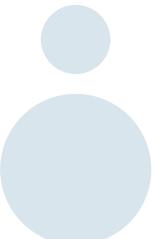
Was ist Camundathon – Nacht der Modellierung ?

- kreativer Abend von Studenten für Studenten
- Aufgaben der anderen Studierenden lösen mit Realitätsbezug
- Austausch mit anderen Studenten und Professoren in angenehmer Atmosphäre
- Netzwerken
- Wissensanwendung und dem anschließenden sicheren Umgang mit der Modellierungssprache BPMN 2.0



Rahmenbedingungen:

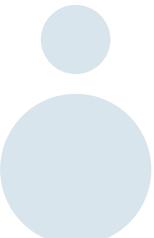
- Gruppen von maximal vier Teilnehmern
- Die Teilnahme ist **FREIWILLIG**
- Workflow mithilfe von Camunda
- Themenvorschläge nach der ersten Woche über Moodle o.Ä. an die leitende Person kommuniziert
- Erfolgreiches Einreichen einer Aufgabe wirkt sich positiv auf Ihre Klausur aus (Bonuspunkte oder andere Anreize)
- Diskussionen und Austausch erwünscht



Aufgabenstellung Camundathon

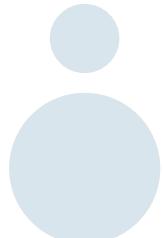
Jedes Modell sollte folgende Teile beinhalten:

- Start Event
- Service Task
- User Task
- Script Task (Bspl.: Validierung des Alters, Berechnung,...)
- Gateway
- JUnit Test
- End Event



Szenario - Camundathon

Sie erstellen einen Workflow mithilfe von Camunda. Dieser soll eine angebotene Leistung der Hochschule automatisieren bzw. digitalisieren. Die erste Woche wird dafür genutzt, das passende Thema zu finden und gegebenenfalls auch schon erste Prototypen/MVP's zu generieren.



H T
W I
G N

Hochschule Konstanz
Fakultät Informatik