

Fachvortrag BPMN

Business Process Model und Notation

Dragonfly

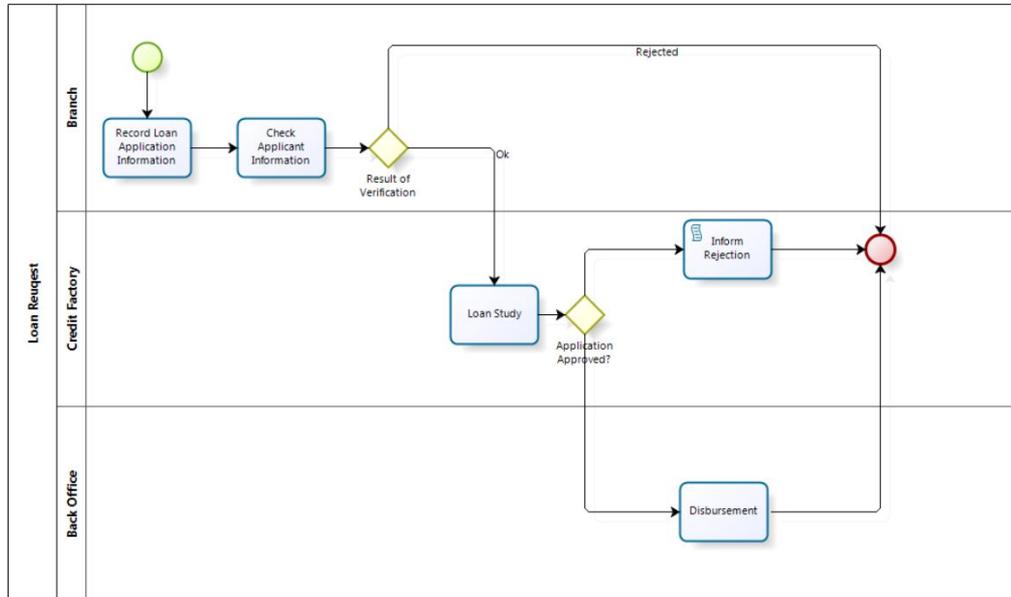
Deniz Arslan 848578
Benno Wende 852210
Jens Richter 851992

Gliederung

- Einführung zu BPMN
 - Unterschiede zu UML
 - Geschichte
 - Zielgruppen/Levels
- Notationen
 - Sub-Prozesse
 - Flow Objects
 - Connecting Objects
 - Pools and Swimlanes
 - Artifacts
- Beispiel: Pizza

Was ist BPMN?

- grafische Darstellung und Notation der Schritte im Geschäftsprozess
- detaillierte Aktivitäten und Informationsflüsse



Unterschiede zu UML

- UML ist Objekt-Orientiert
- BPMN ist Prozess-Orientiert
- UML ist primär für Software-Design
- BPMN ist primär für Geschäftsprozesse
- BPMN und UML sind kompatibel miteinander

Geschichte der BPMN

- 2001 von IBM-Mitarbeiter Stephen A. White erschaffen
- 2004 von der Business Process Management Initiative (BPMI) veröffentlicht
- Juni 2005 wurde es von der Object Management Group (OMG) übernommen
- Gleichzeitig haben BPMI und OMG fusioniert, wodurch es zum OMG standard wurde
- Seit 2006 ist BPMN 1.0 offizieller OMG standard.
- 2008 erschien version 1.1
- 2009 erschien version 1.2
- Januar 2011 BPMN 2.0

Warum gibt es BPMN?

- große Änderungen in der Geschäftsprozess-Welt
- Prozesse werden immer komplexer
- vor BPMN kein Standard
- keine eindeutige Kommunikation möglich

→ BPMN wurde entwickelt um Nutzern eine lizenzfreie Notation zur Verfügung zu stellen.

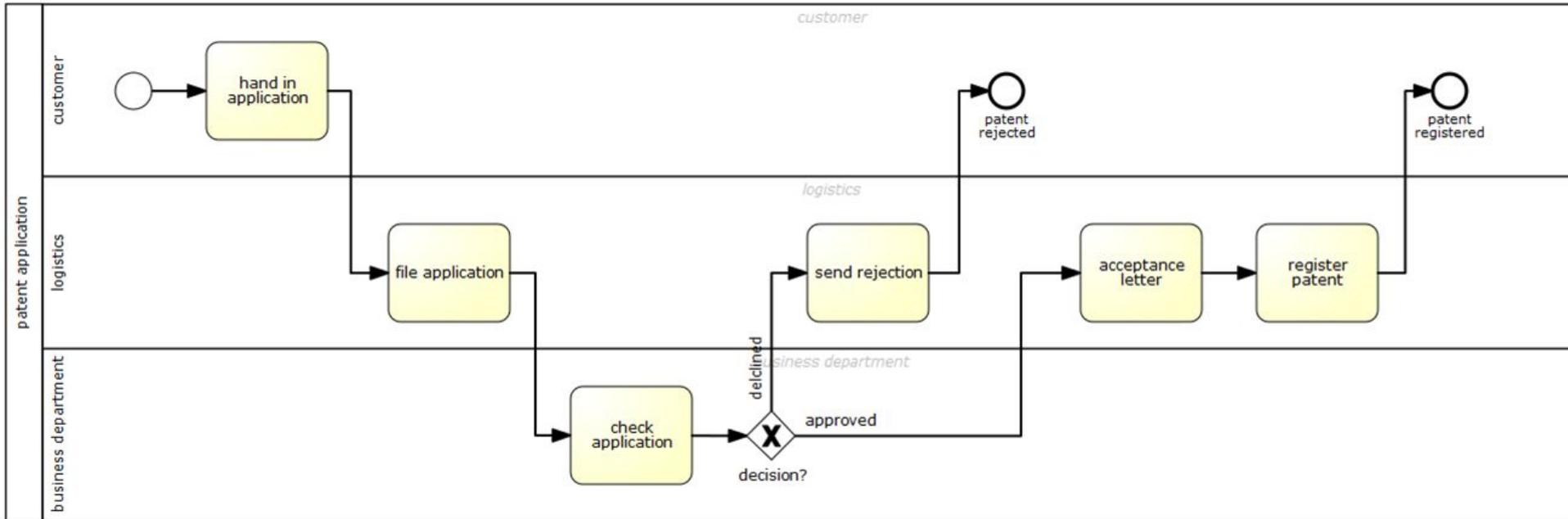
Zielgruppe des BPMN - High Level

Mehrere Level der Komplexität -> mehr Zielgruppen

High Level: Business-User

- leicht verständlich
- Basis-Kenntnisse über swim-lanes, task, subprocess

Zielgruppe des BPMN - High Level

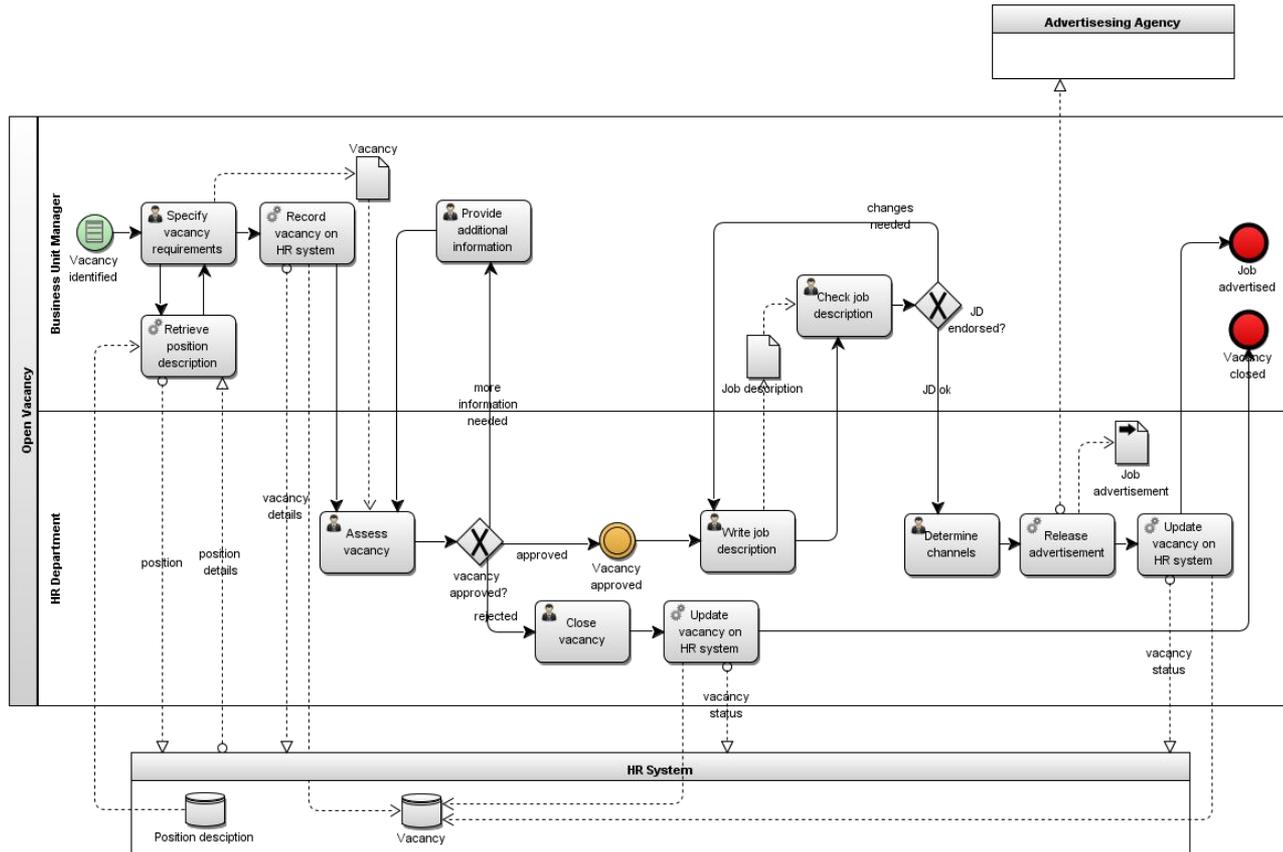


Zielgruppe des BPMN - Lower Level

Lower Level: Process-Implementierer

- macht High-Level detaillierter
- high-level Kenntnisse + decision & merge patterns, events, exception handling notwendig
- viel komplexer

Zielgruppe des BPMN - Lower Level



Activities



A **Task** is a unit of work, the job to be performed. When marked with a symbol it indicates a **Sub-Process**, an activity that can be refined.



A **Transaction** is a set of activities that logically belong together; it might follow a specified transaction protocol.



An **Event Sub-Process** is placed into a Process or Sub-Process. It is activated when its start event gets triggered and can interrupt the higher level process context or run in parallel (non-interrupting) depending on the start event.



A **Call Activity** is a wrapper for a globally defined Task or Process reused in the current Process. A call to a Process is marked with a .

Activity Markers

Markers indicate execution behavior of activities:

- Sub-Process Marker
- Loop Marker
- Parallel MI Marker
- Sequential MI Marker
- Ad Hoc Marker
- Compensation Marker

Sequence Flow
defines the execution order of activities.

Default Flow
is the default branch to be chosen if all other conditions evaluate to false.

Conditional Flow
has a condition assigned that defines whether or not the flow is used.

Task Types

Types specify the nature of the action to be performed:

- Send Task
- Receive Task
- User Task
- Manual Task
- Service Task
- Script Task

Conversations



A **Conversation** defines a set of logically related message exchanges. When marked with a it indicates a **Sub-Conversation**, a compound conversation element.

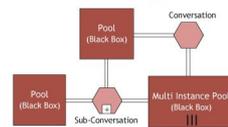


A **Call Conversation** is a wrapper for a globally defined Conversation or Sub-Conversation. A call to a Conversation is marked with a .

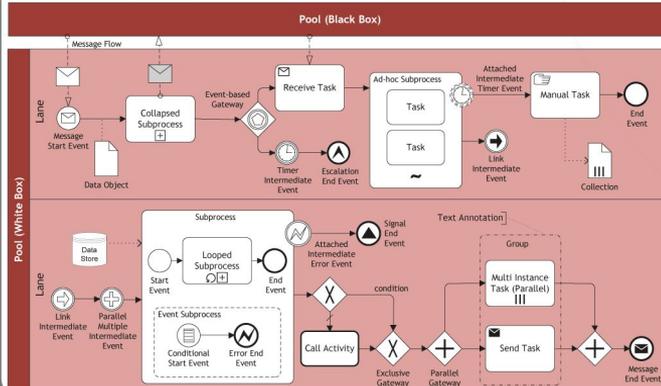


A **Conversation Link** connects Conversations and Participants.

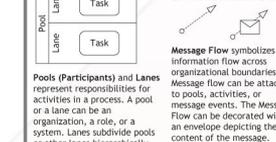
Conversation Diagram



Collaboration Diagram



Swimlanes



Choreographies



A **Choreography Task** represents an Interaction (Message Exchange) between two Participants.

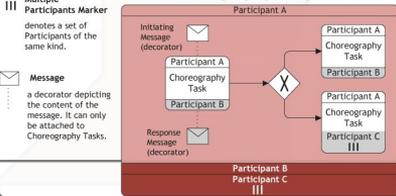


A **Sub-Choreography** contains a refined choreography with several interactions.



A **Call Choreography** is a wrapper for a globally defined Choreography Task or Sub-Choreography. A call to a Sub-Choreography is marked with a .

Choreography Diagram



Events

	Start	Intermediate	End
Standard			
Event Sub-Process Interrupting			
Event Sub-Process Non-Interrupting			
Catching			
Boundary Interrupting			
Boundary Non-Interrupting			
Throwing			
Standard			

None: Untyped events, indicate start point, state changes or final states.

Message: Receiving and sending messages.

Timer: Cyclic timer events, points in time, time spans or timeouts.

Escalation: Escalating to an higher level of responsibility.

Conditional: Reacting to changed business conditions or integrating business rules.

Link: Off-page connectors. Two corresponding link events equal a sequence flow.

Error: Catching or throwing named errors.

Cancel: Reacting to cancelled transactions or triggering cancellation.

Compensation: Handling or triggering compensation.

Signal: Signalling across different processes. A signal thrown can be caught multiple times.

Multiple: Catching one out of a set of events. Throwing all events defined.

Parallel Multiple: Catching all out of a set of parallel events.

Terminate: Triggering the immediate termination of a process.

Gateways



Exclusive Gateway When splitting, it routes the sequence flow to exactly one of the outgoing branches. When merging, it awaits one incoming branch to complete before triggering the outgoing flow.



Event-based Gateway Is always followed by catching events or receive tasks. Sequence flow is routed to the subsequent event/task which happens first.



Parallel Gateway When used to split the sequence flow, all outgoing branches are activated simultaneously. When merging parallel branches it waits for all incoming branches to complete before triggering the outgoing flow.



Inclusive Gateway When splitting, one or more branches are activated. All active incoming branches must complete before merging.



Exclusive Event-based Gateway (Instantane) Each occurrence of a subsequent event starts a new process instance.



Complex Gateway Complex merging and branching behavior that is not captured by other gateways.



Parallel Event-based Gateway (Instantane) The occurrence of all subsequent events starts a new process instance.

Data



A **Data Object** represents information flowing through the process, such as business documents, e-mails, or letters.



A **Collection Data Object** represents a collection of information, e.g., a list of order items.



A **Data Input** is an external input for the entire process. A kind of input parameter.



A **Data Output** is data result of the entire process. A kind of output parameter.



A **Data Association** is used to associate data elements to Activities, Processes and Global Tasks.

A **Data Store** is a place where the process can read or write data, e.g., a database or a filing cabinet. It persists beyond the lifetime of the process instance.



Notationen

- Sub-Prozesse
- Flow Objects
- Connecting Objects
- Pools and Swimlanes
- Artifacts

Sub-Prozesse

Private(interne) Geschäftsprozesse

- Die Sequenz bleibt in der “Swim lane”
- Kommunikation zu anderen prozessen kann gezeigt werden

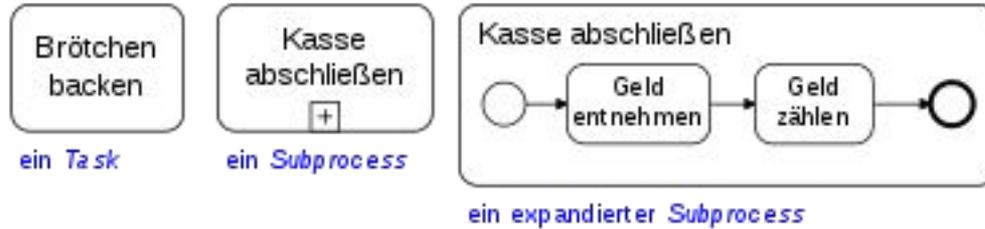
Abstrakte(Öffentliche) Prozesse

- Interaktion von mehreren Geschäftsprozessen
- Nur die prozesse die außerhalb des privaten Geschäftsprozess kommunizieren werden gezeigt

Zusammenarbeit(Globale) Prozesse

- Beschreibt die interaktion von 2 oder mehr Geschäfts Entitäten

Flow Objects: Activities



- Eine Aktivität beschreibt eine einzelne aufgabe die zu erledigen ist.
- Eine Einzelne Aktivität wird "Task" genannt.
- Eine Komplexe Aktivität wird "Sub-process" genannt.
- Ein Task und ein Sub-process unterscheiden sich durch ein +

Flow Objects: Gateway



exklusives
Gateway
(XOR)



inklusive
Gateway
(OR)



paralleles
Gateway
(AND)



Event-
basiertes
Gateway

- Ein Gateway stellt einen Entscheidungspunkt dar.
- Sie können eine Spaltung oder ein Zusammenlaufen Notieren.
- Desweiteren können sie auch von spezifischen events abhängig sein.

Flow Objects: Events



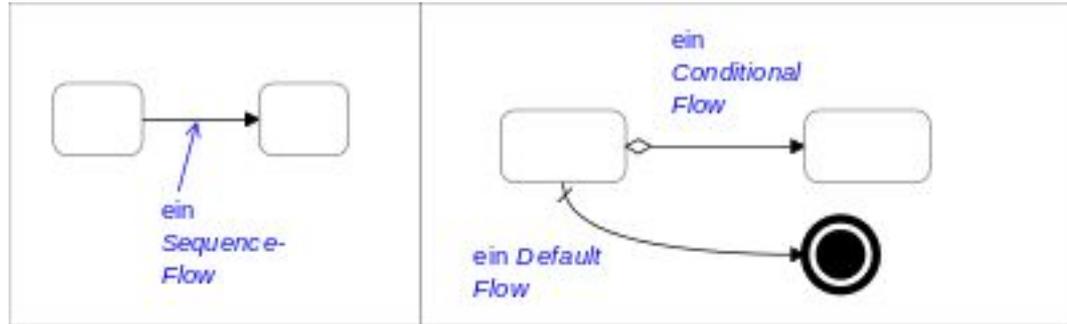
Ein Event ist ein ereignis in einem Geschäftsprozess.

Events werden in drei Klassen eingeteilt:

- Nach ihrer Position im Prozess in “Start”, “Intermediate” und “End-Event”
- Nach ihrer Wirkung im Prozess in “Catching” und “Throwing-Event”
- Nach ihrer Art in “Timer”, “Message”, “Exception-Event” etc.

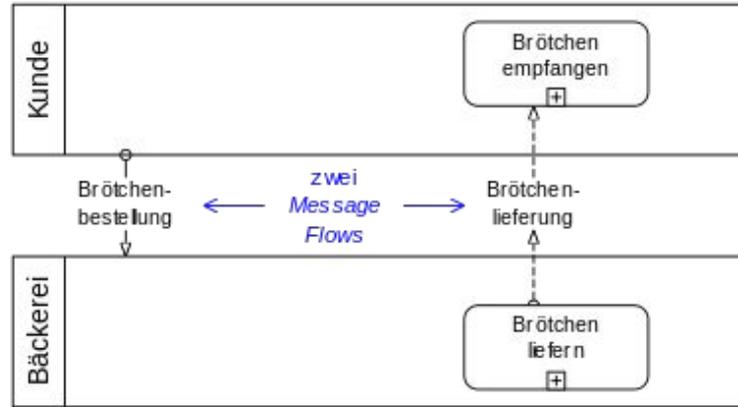
Pro Event-typ gibt es ein Symbol das im Kreis Symbols des Events angezeigt wird

Connecting Objects: Sequence Flow



- Sequence Flows verbinden Aktivitäten, Gateways und Events.
- Sie stellen die reihenfolge und den Ablauf dar.
- Conditional Flows gelten nur unter bestimmten Bedingungen.
- Default Flows werden gelten wenn kein andere Flow gültig ist.

Connecting Objects: Message Flow



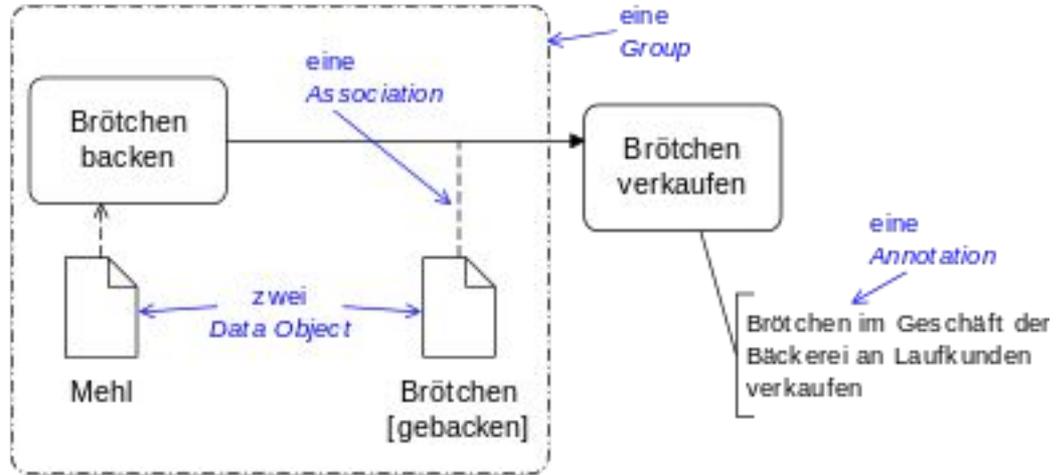
- Message Flow zeigt wie Geschäftsprozesse meldungen austauschen
- Message Flow kann nur Lanes oder Pools verbinden
- Message Flow verbindet Lanes, Pools oder Flow Objects nur Temporär

Pools And Swimlanes



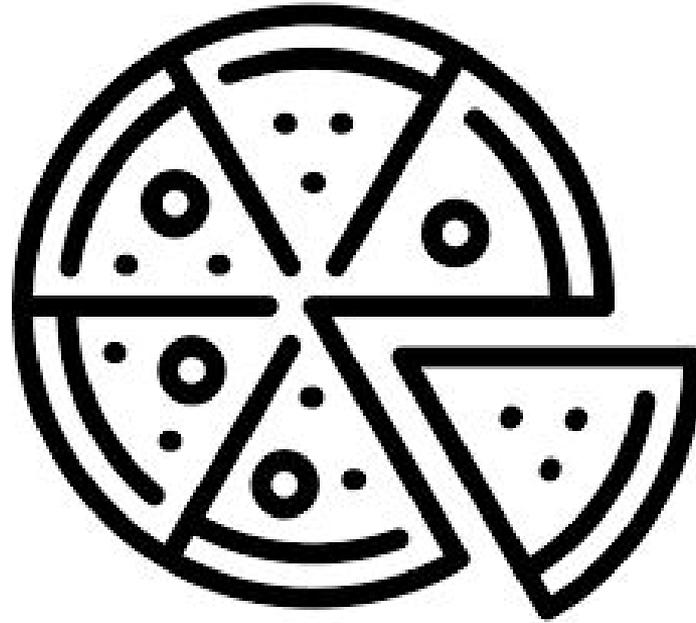
- Ein Pool repräsentiert die Prozessbeteiligten, z.B. eine Organisation
- Lanes sind Unterteilungen des Pools über die gesamte Länge.
- Lanes sind Teilnehmer, also Benutzer, Benutzerrollen oder ein System.

Artifacts



- Annotationen sind Kommentare die einer Activity zugeordnet sind
- Data Objects können digitale dokumente oder physische objekte sein
- Data Objects repräsentieren Artefakte die der Geschäftsprozess bearbeitet.
- Groups fassen Elemente eines Geschäftsprozesses visuell zusammen

Praktisches Beispiel: Pizza Bestellung



Pizzeria

Student (Pizza Kunde)

Student (Pizza Kunde)

Pizzeria

Lieferant

Pizzabäcker

Angestellter

Student (Pizza Kunde)


Heißhunger
auf
Pizza

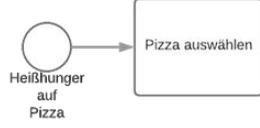
Pizzeria

Pizzabäcker

Liefrant

Angestellter

Student (Pizza Kunde)



Pizzaria

Pizzabäcker

Angestellter

Liefrant

Student (Pizza Kunde)



Pizzeria

Pizzabäcker

Liefrent

Angestellter

Student (Pizza Kunde)

Heißhunger
auf
Pizza

Pizza auswählen

Pizza bestellen
(Pizza Mozzarella)

Pizza
erhalten

Pizzeria

Pizzabäcker

Liefrent

Angestellter

Student (Pizza Kunde)



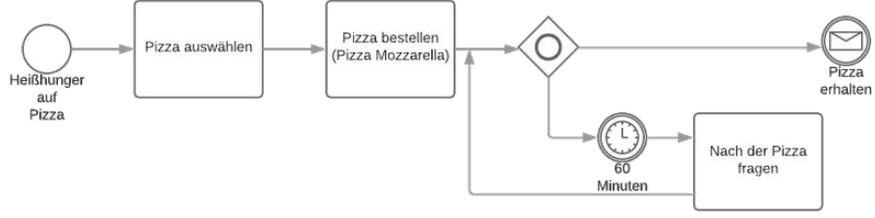
Pizzeria

Pizzabäcker

Angestellter

Liefrent

Student (Pizza Kunde)



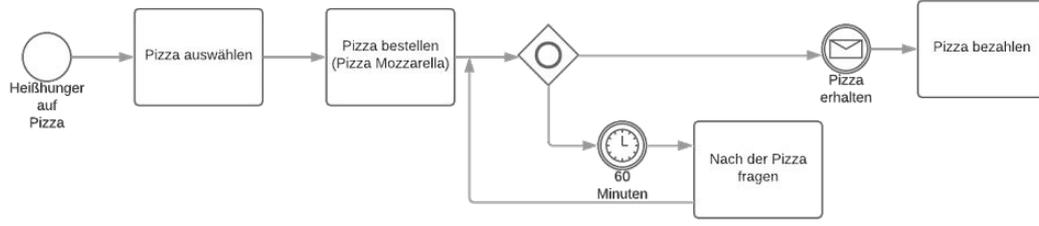
Pizzeria

Pizzabäcker

Liefrent

Angestellter

Student (Pizza Kunde)



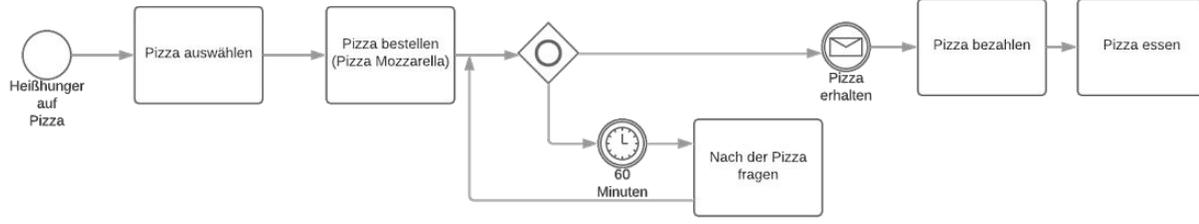
Pizzaria

Pizzabäcker

Liefrent

Angestellter

Student (Pizza Kunde)



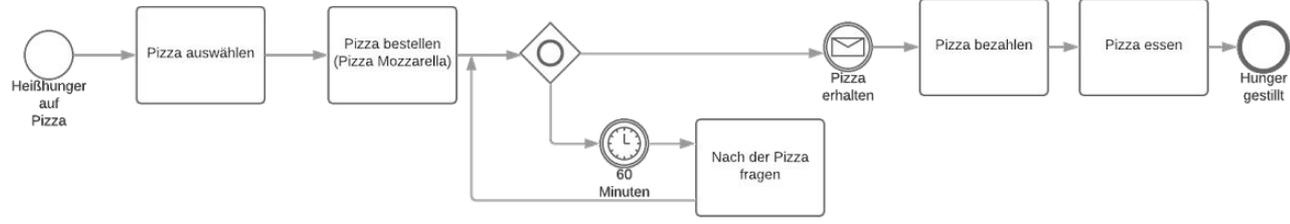
Pizzaria

Pizzabäcker

Liefrent

Angestellter

Student (Pizza Kunde)

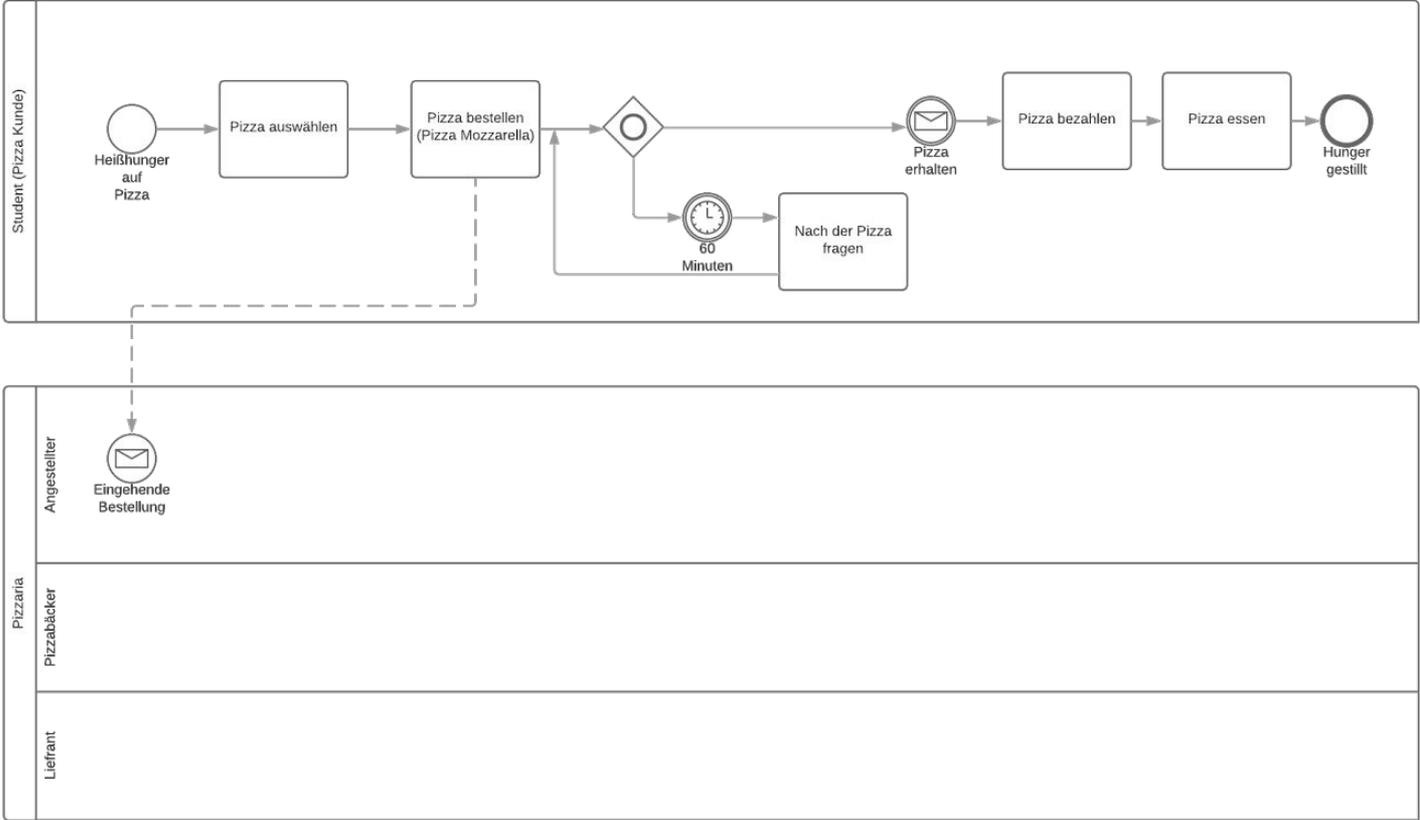


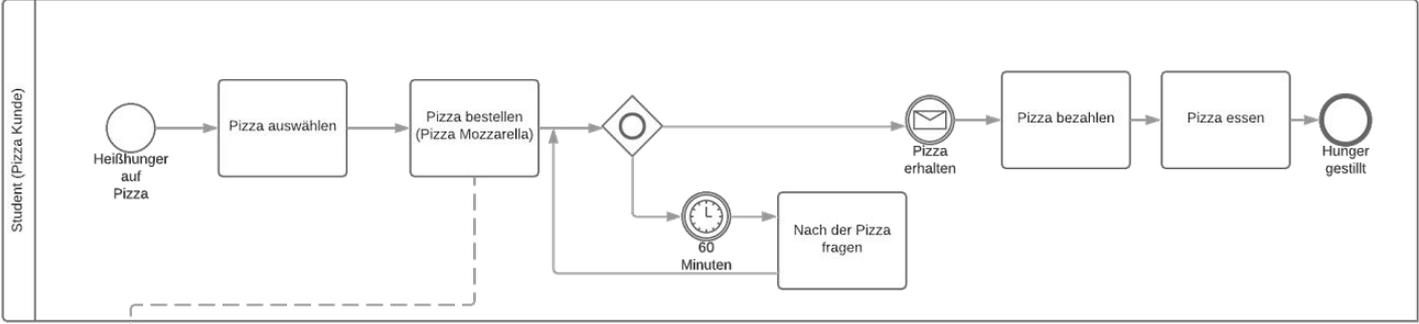
Pizzeria

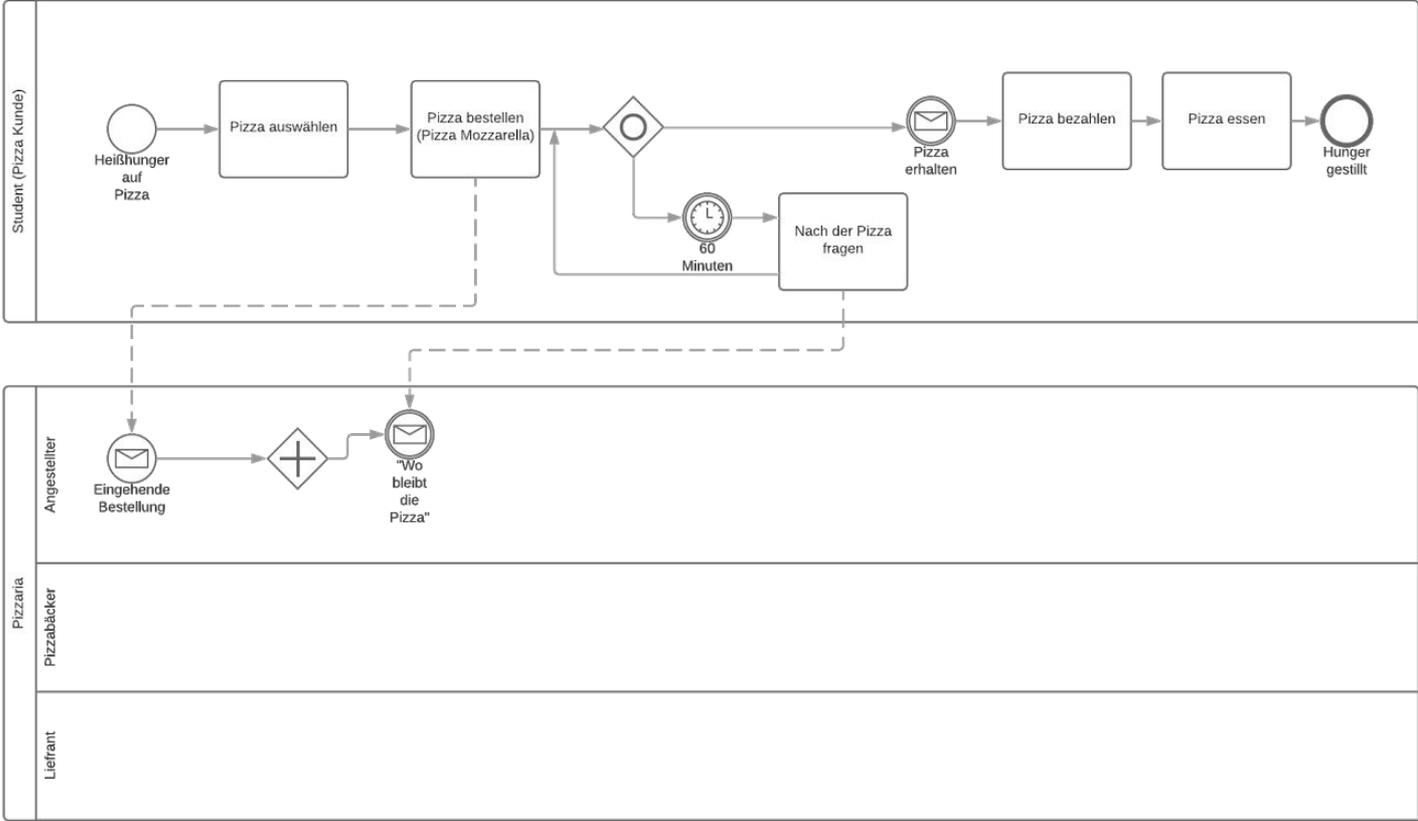
Pizzabäcker

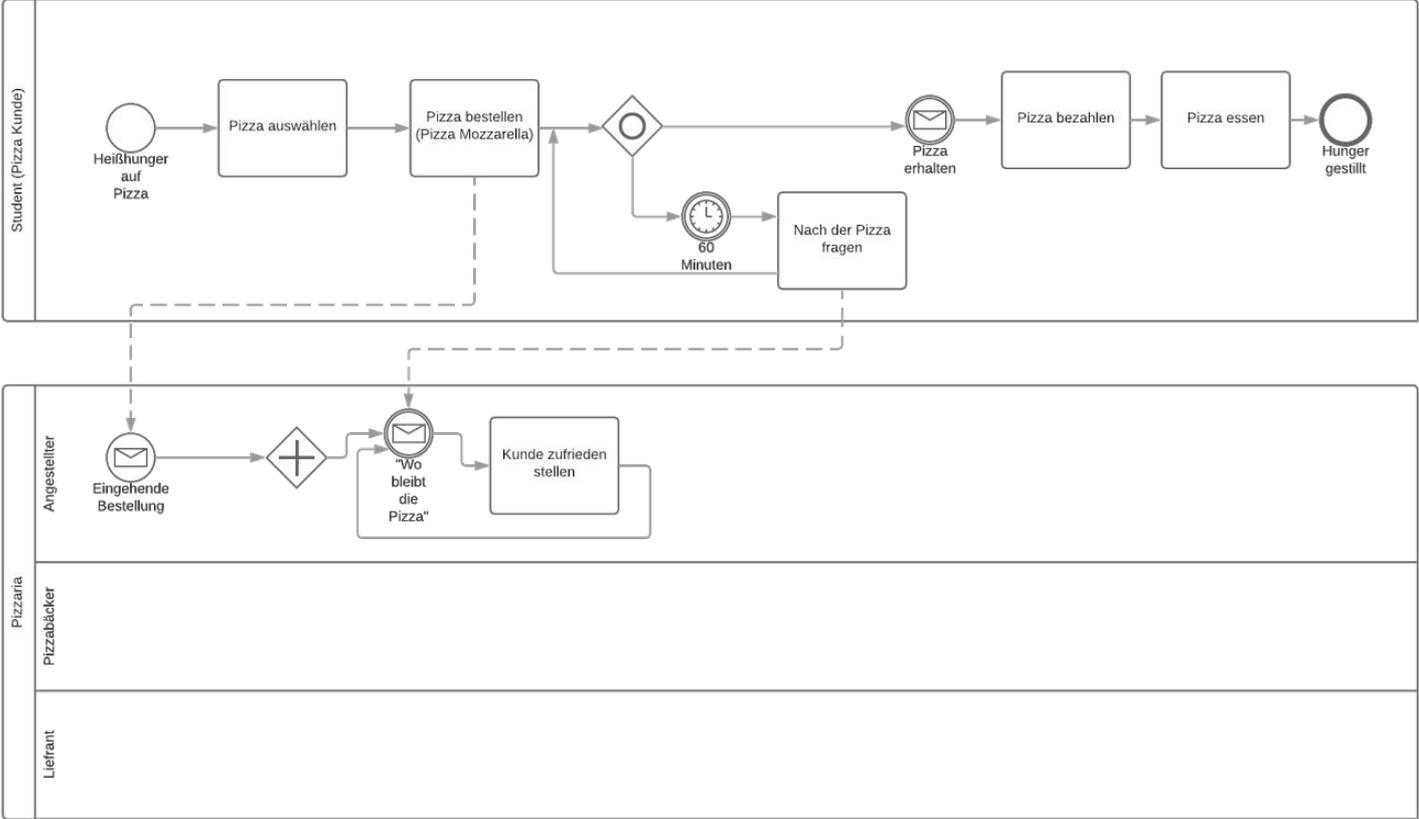
Liefrent

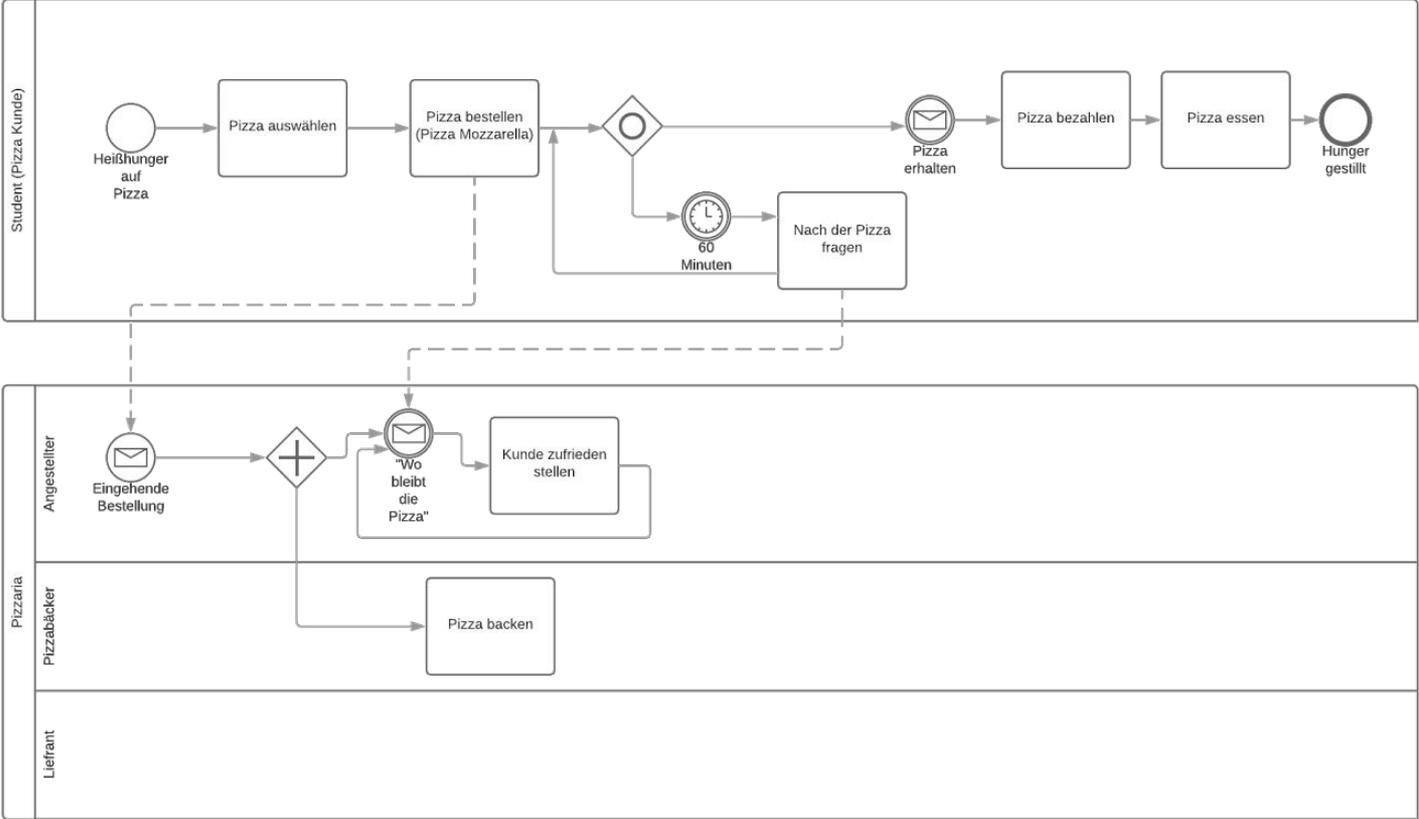
Angestellter

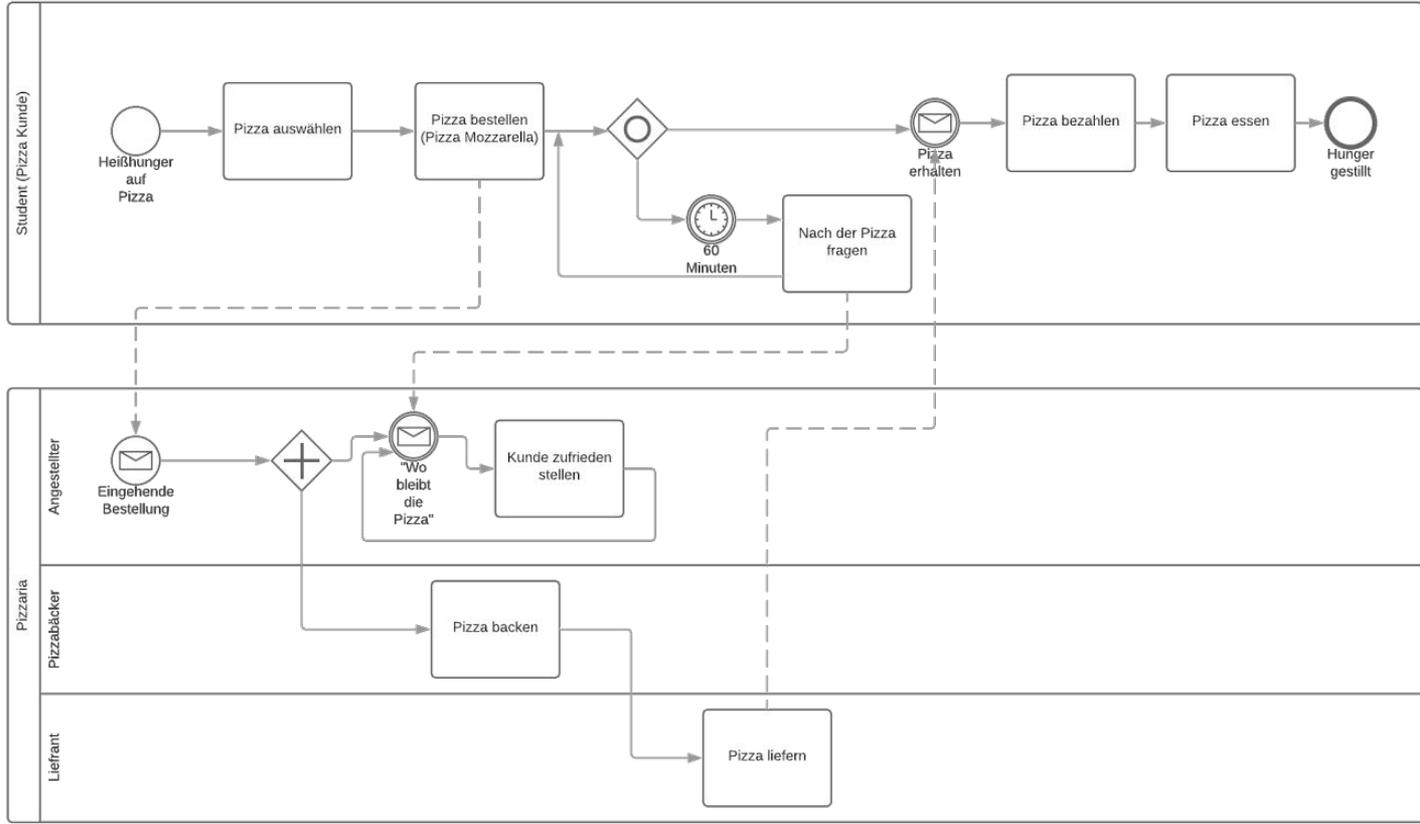


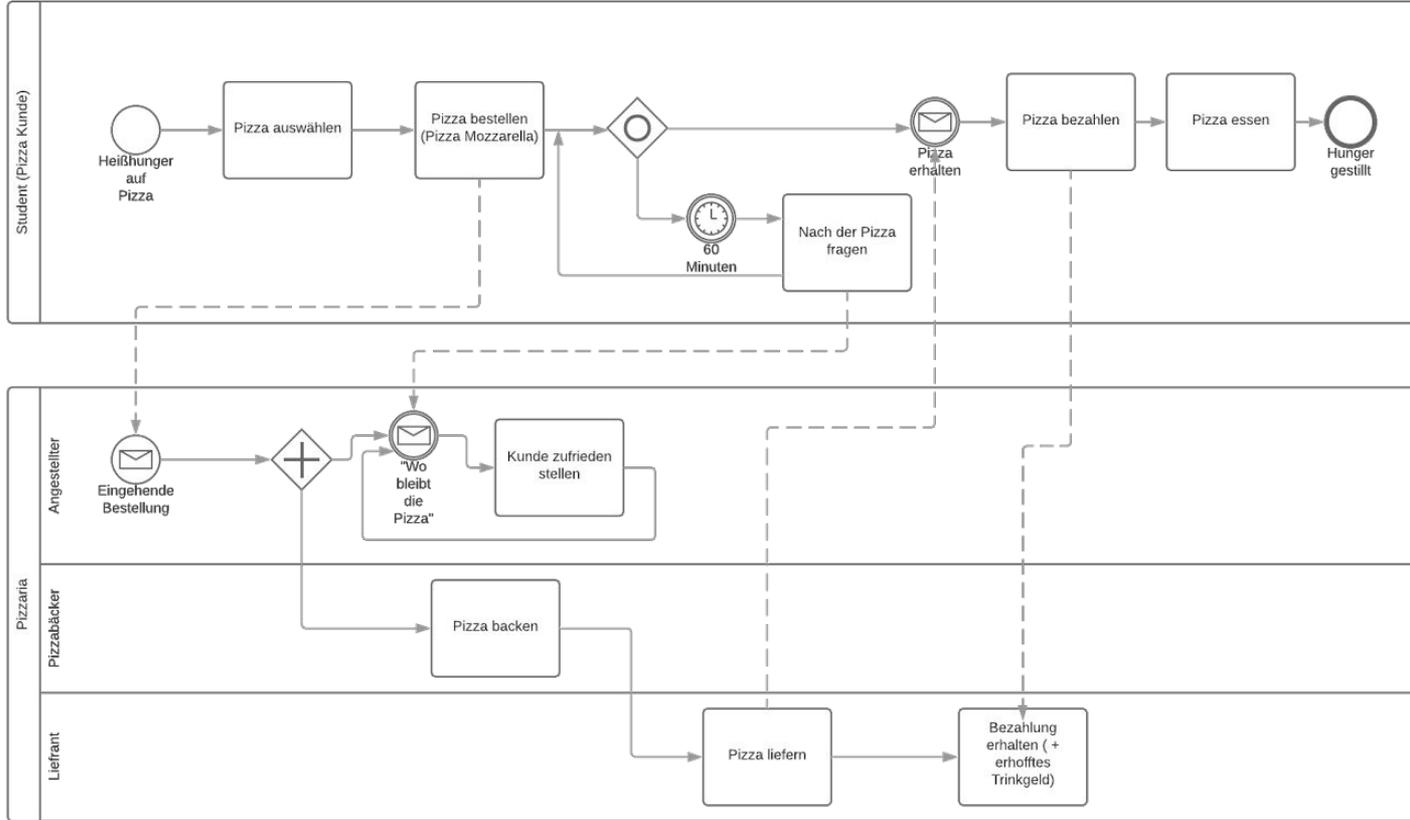


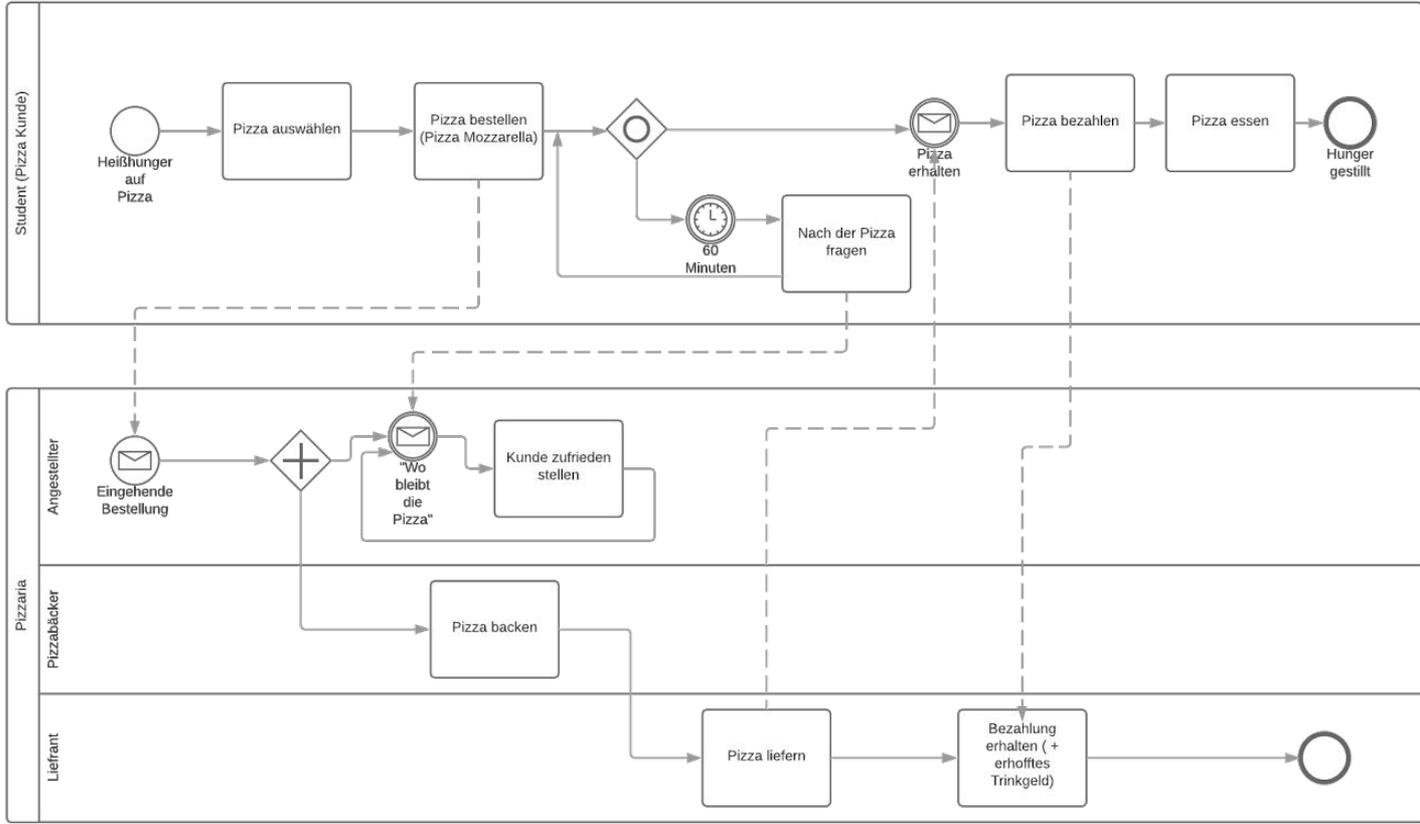


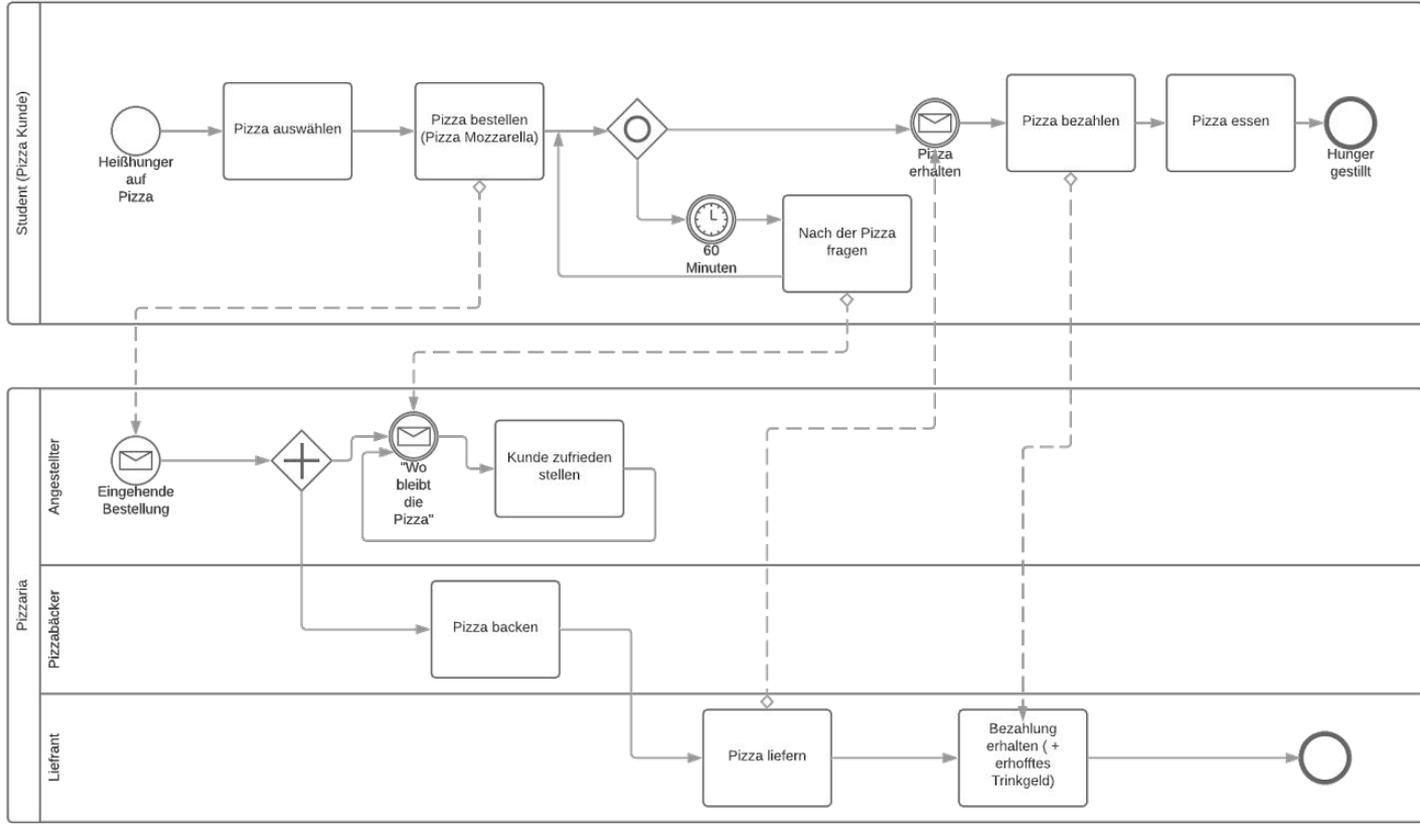












Wichtige Aspekte bei der Erstellung

- klaren Scope definieren
 - Start & Ende
- Empfehlung:
 - Geschäftsprozess modellieren
 - Ineffizienzen ermitteln
 - optimiertes BPMN
- überschaubare Größe
- Sequenzflüsse horizontal
- Assoziationen und Datenflüsse vertikal

Quellen

- https://de.wikipedia.org/wiki/Business_Process_Model_and_Notation
- <http://www.bpmn.org/>
- <http://wirtschaftslexikon.gabler.de/Definition/business-process-model-and-notation-bpmn.html>
- <http://www.omg.org/spec/BPMN/>
- <https://www.kurze-prozesse.de/BPMN/>
- <https://bpmnmatrix.github.io/>
- https://en.wikipedia.org/wiki/Business_Process_Model_and_Notation#Comparison_of_BPMN_versions
- https://de.wikipedia.org/wiki/Datei:Standards_BPM_2009_11.jpg
- <https://camunda.com/bpmn/>
- <https://www.lucidchart.com/pages/de/was-ist-business-process-modeling-notation>