



Theory

01 | Expanded Subprocess



Theory

01 | Expanded Subprocess

02 | Event Subprocess



Theory

- 01 | Expanded Subprocess
- 02 | Event Subprocess
- 03 | Transaction Subprocess



Theory

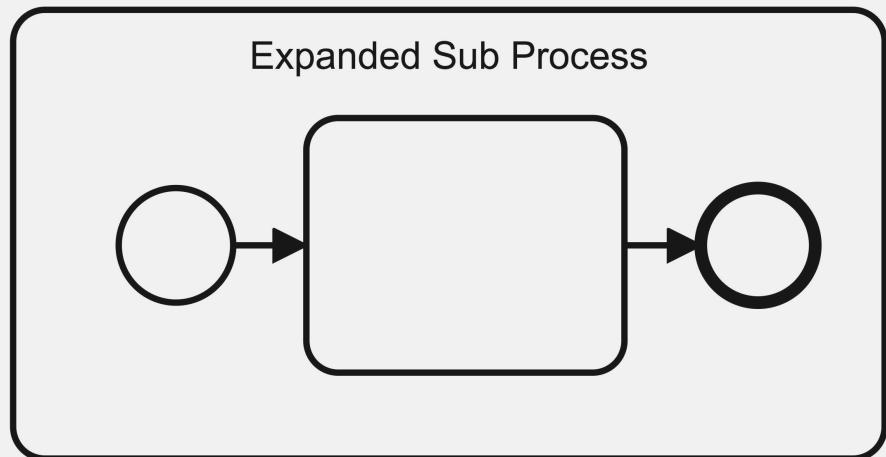
- 01 | Expanded Subprocess
- 02 | Event Subprocess
- 03 | Transaction Subprocess
- 04 | Loop Types

BPMN Theory

process camp

Subprocess

[Expanded]



- All elements are displayed instead of hidden
- Useful to apply events to multiple tasks
- Always starts with a plain start event

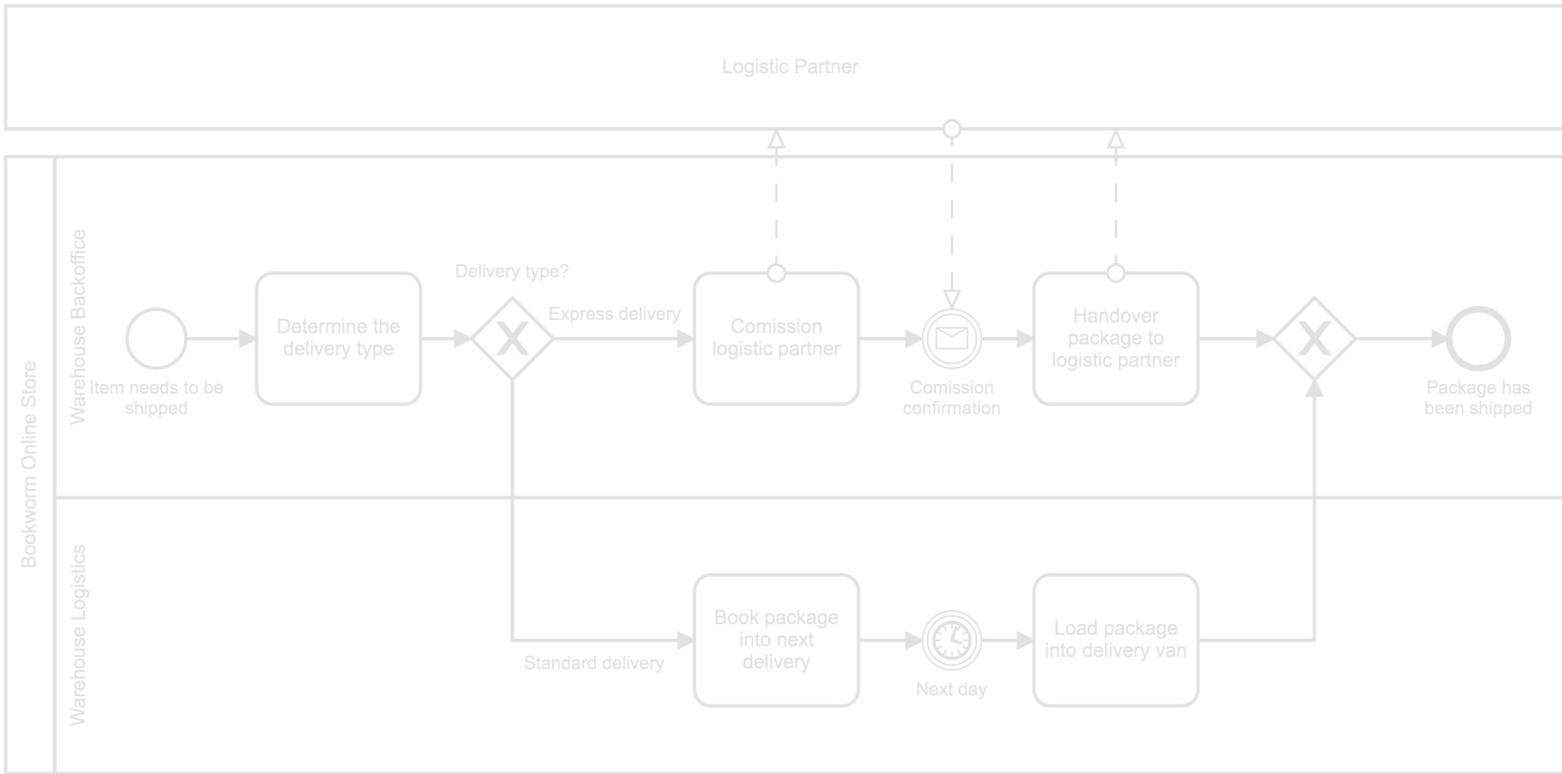
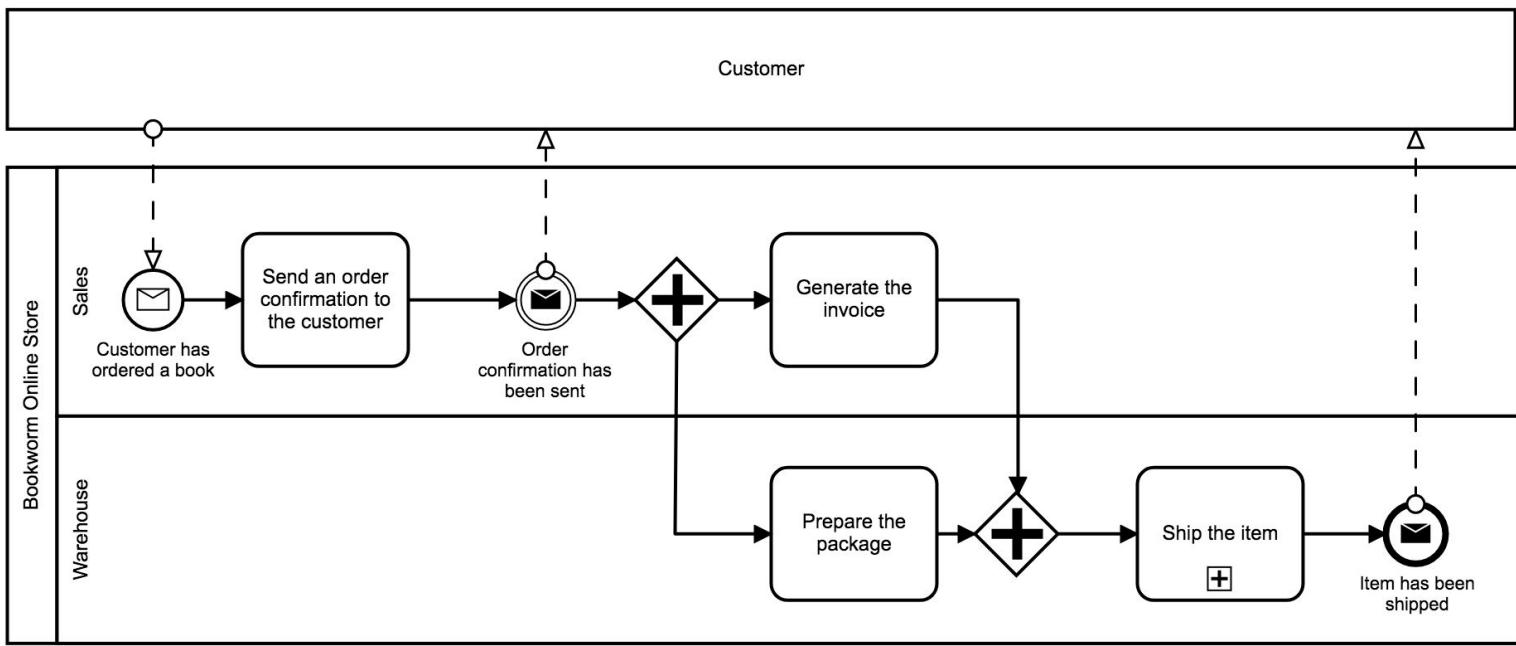


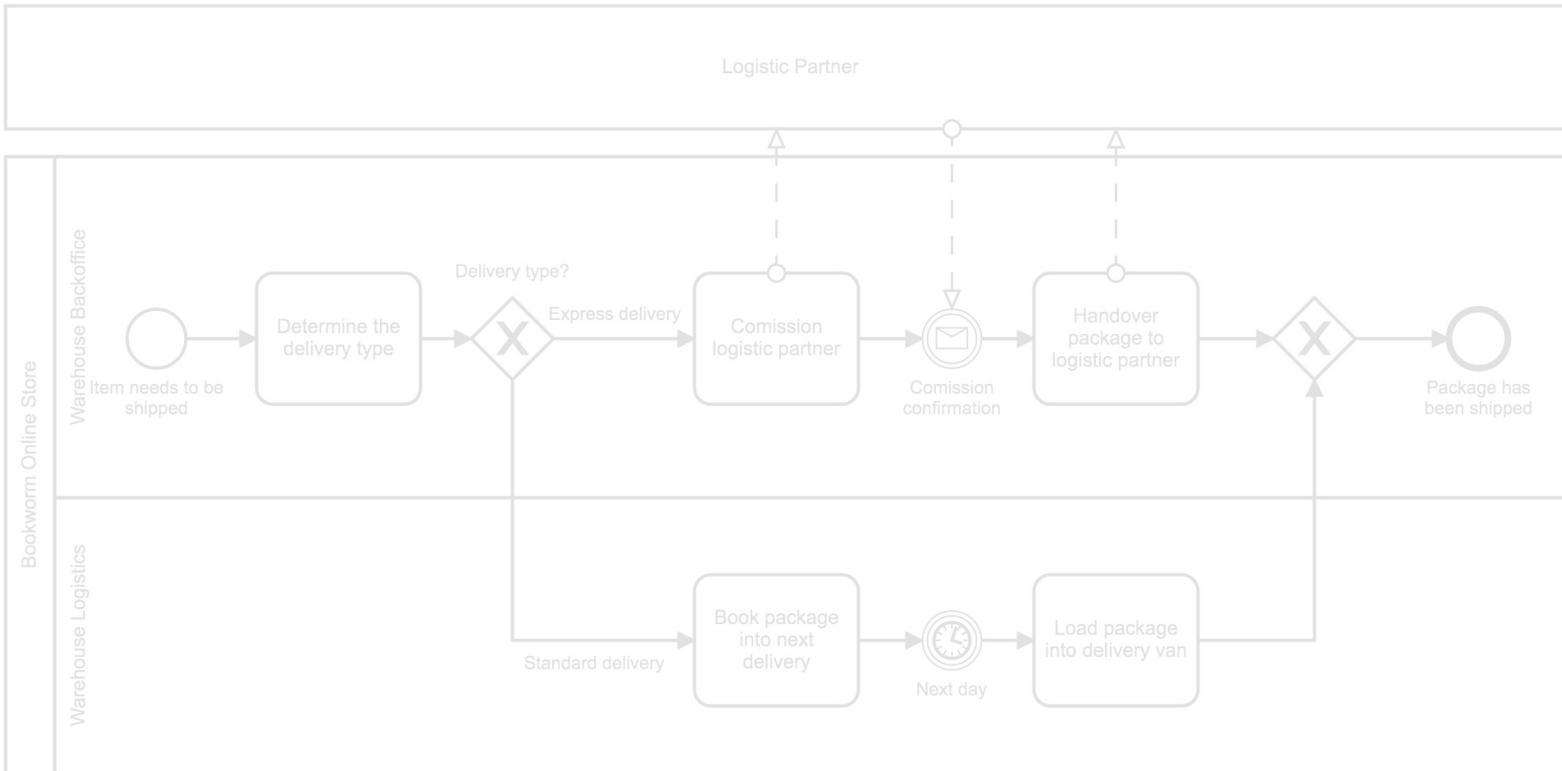
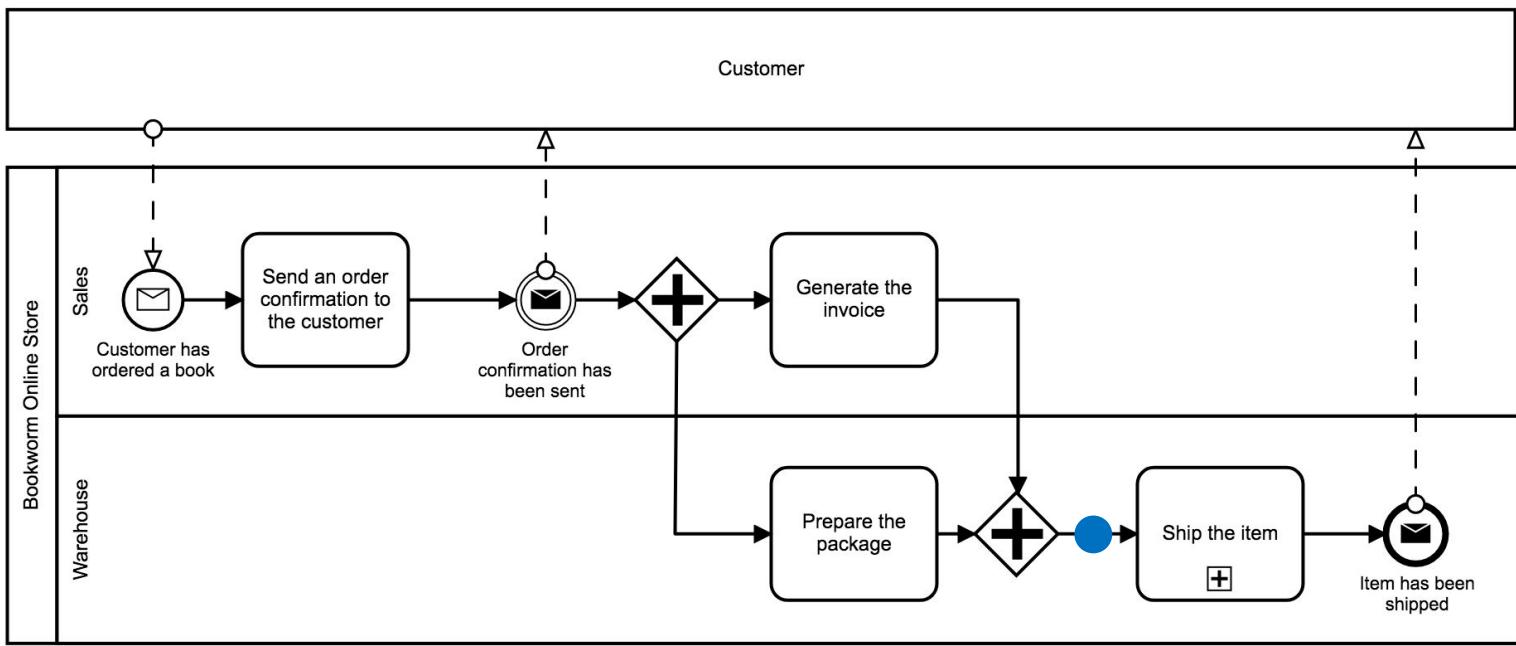
Let's recap how the regular sub process works

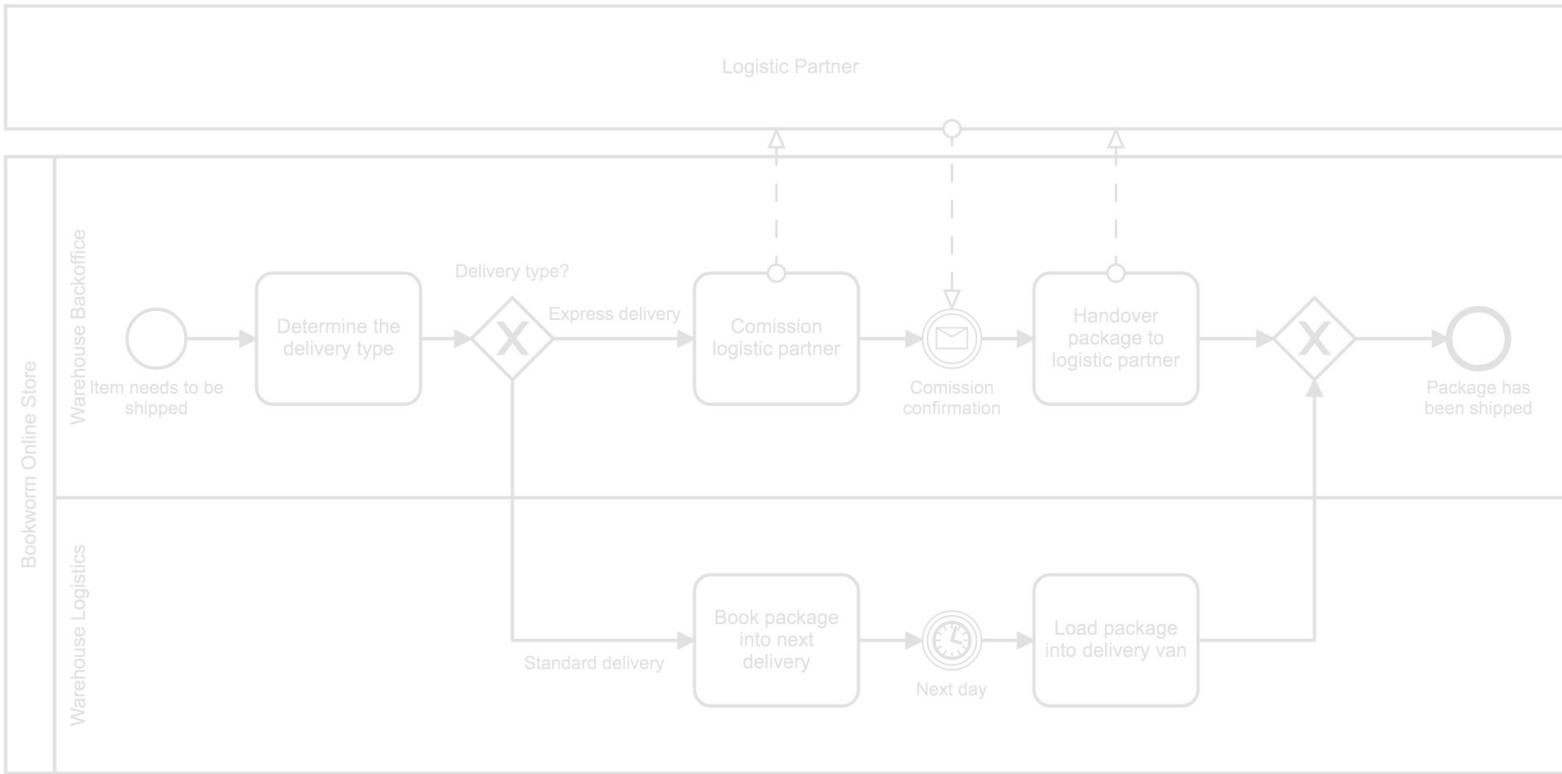
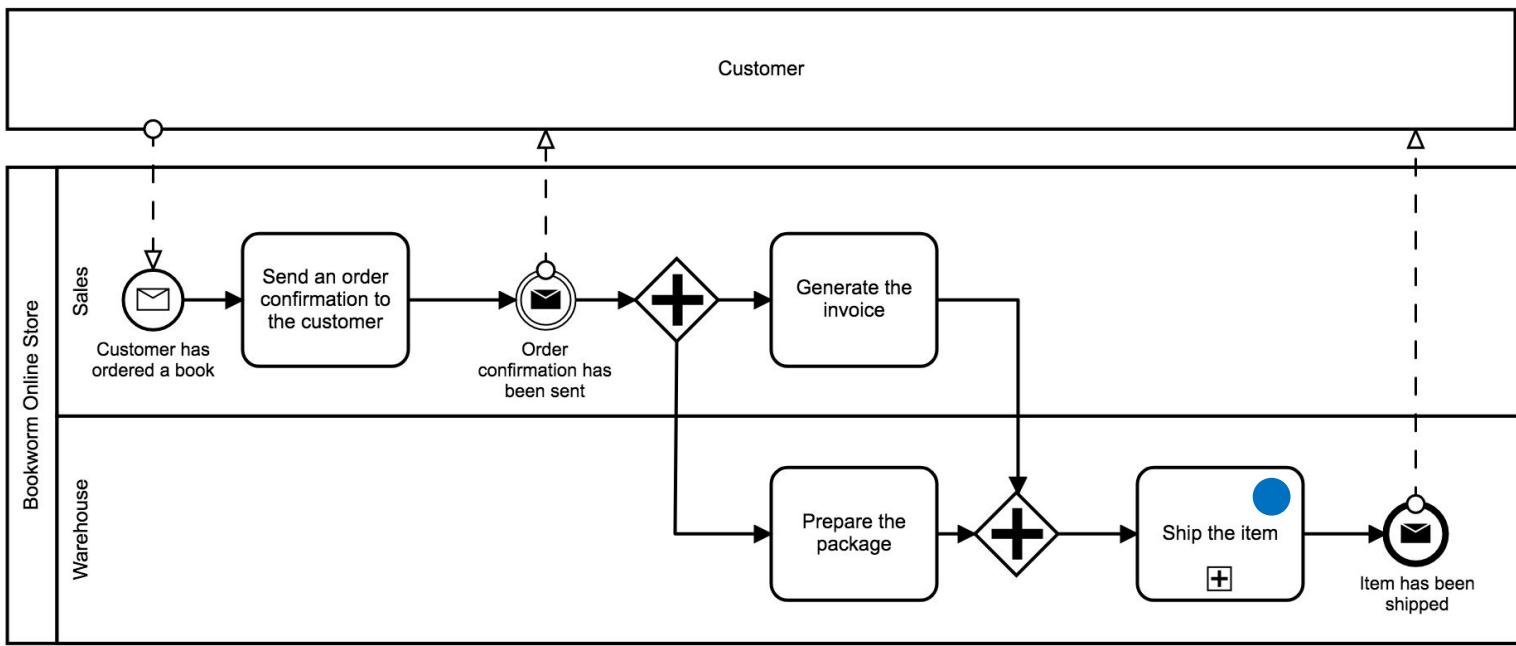


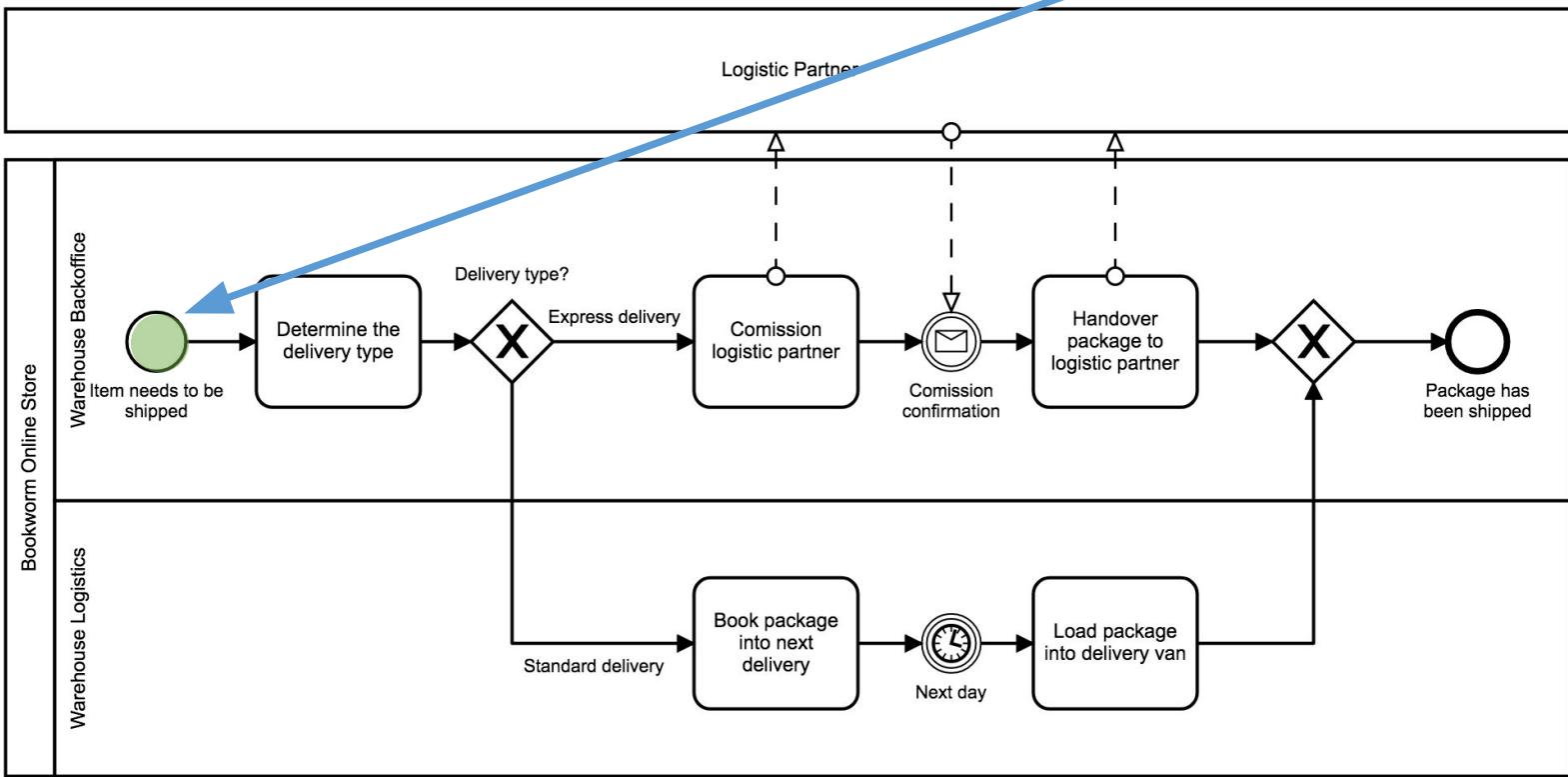
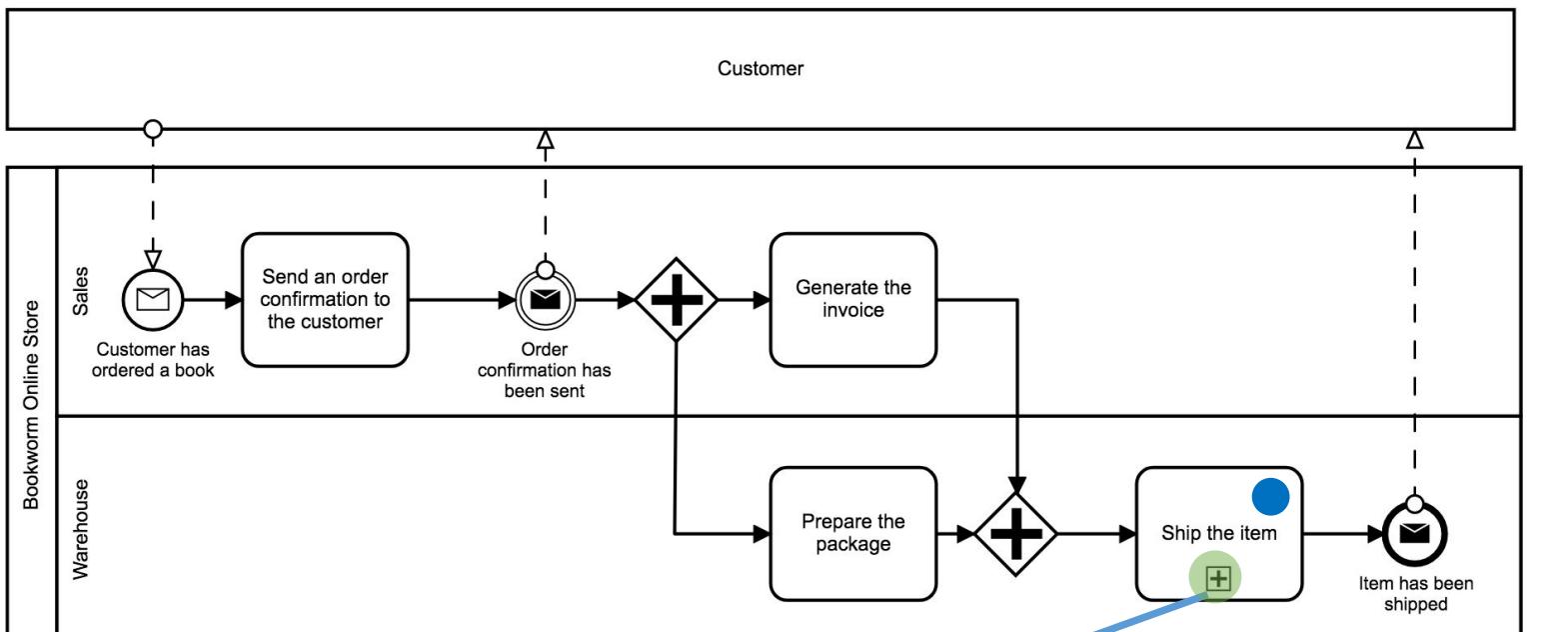
Order process at the Bookworm Online Store

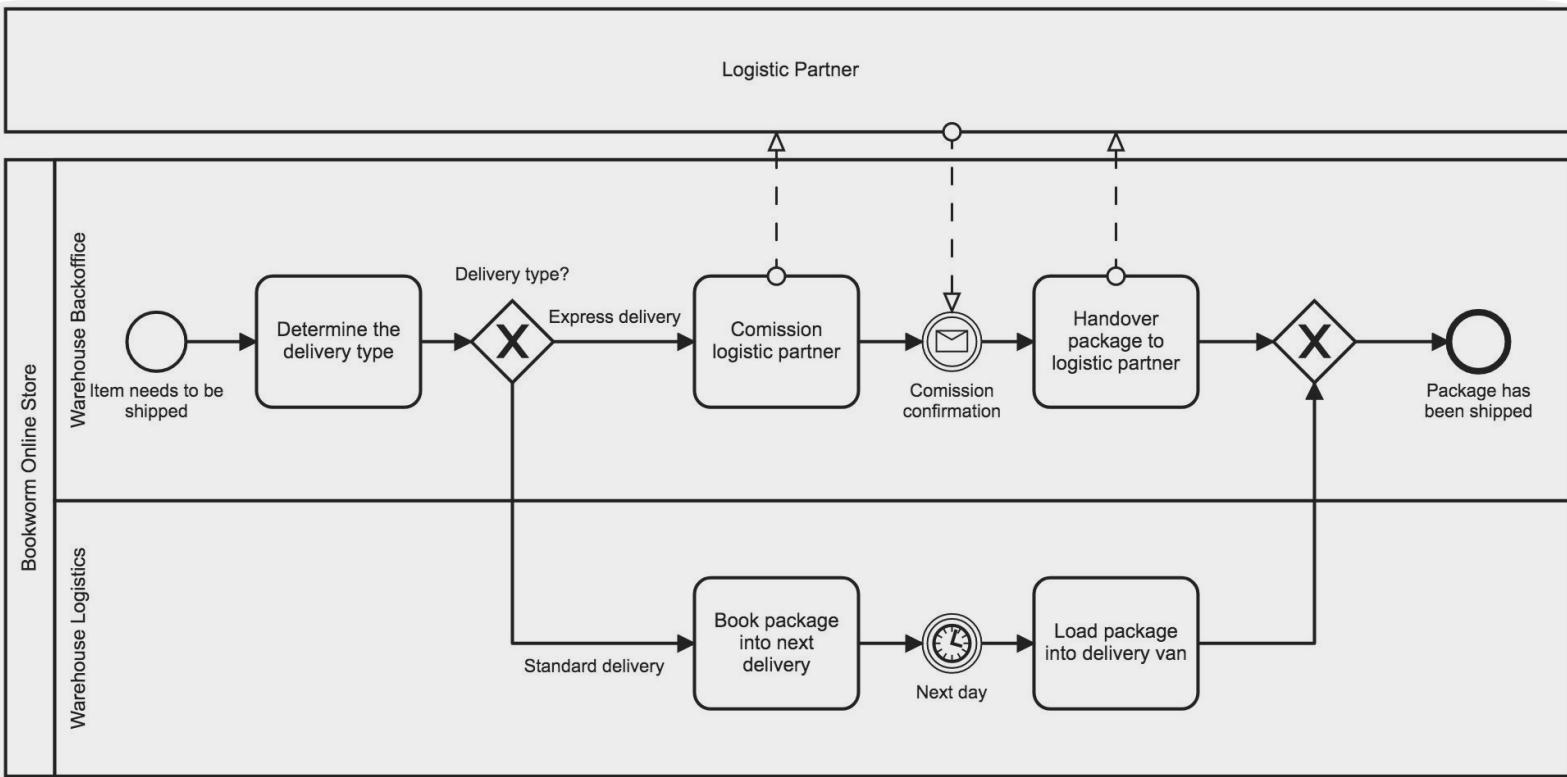
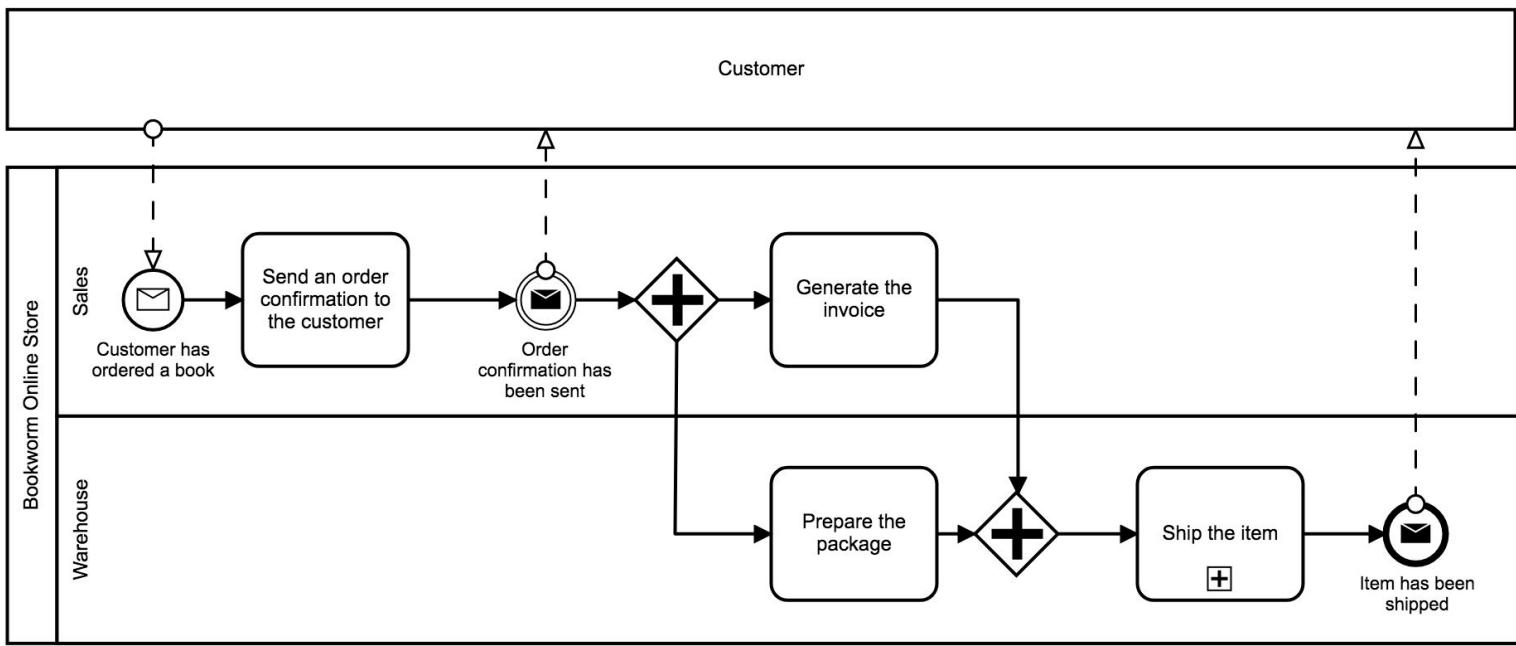
process camp

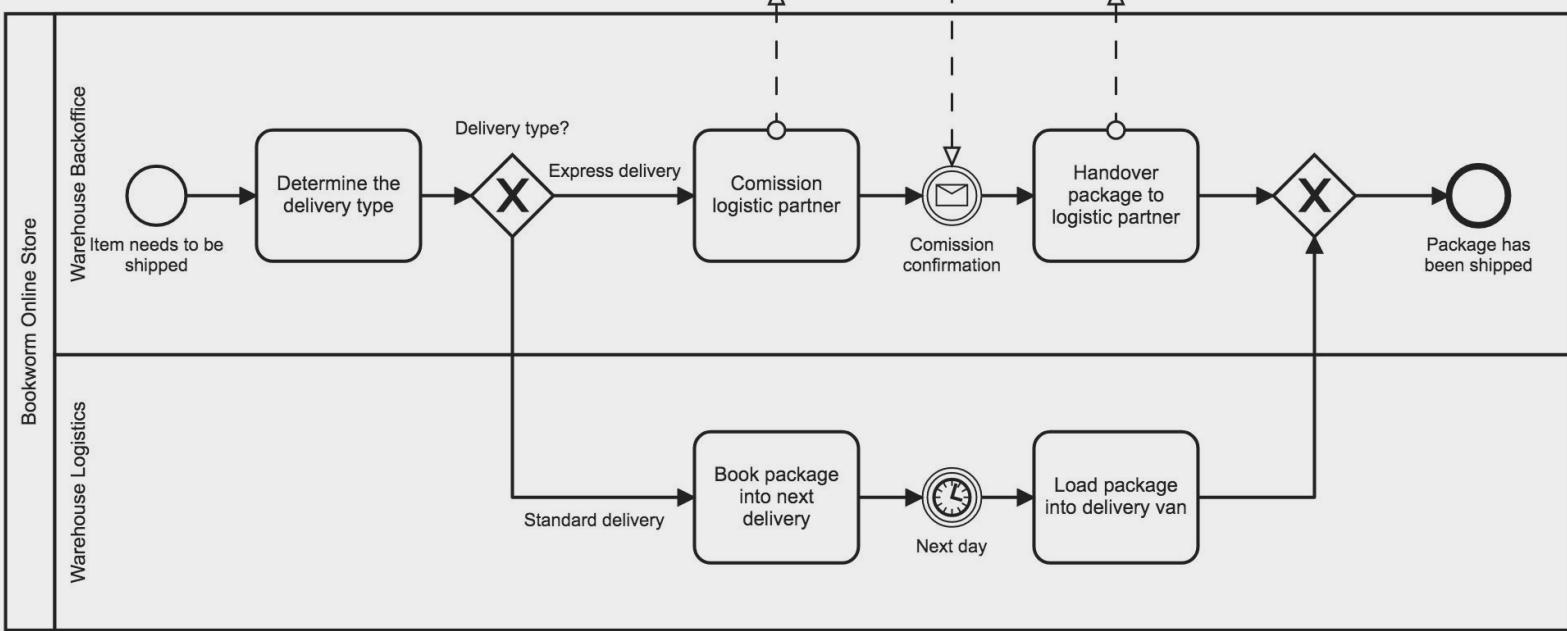
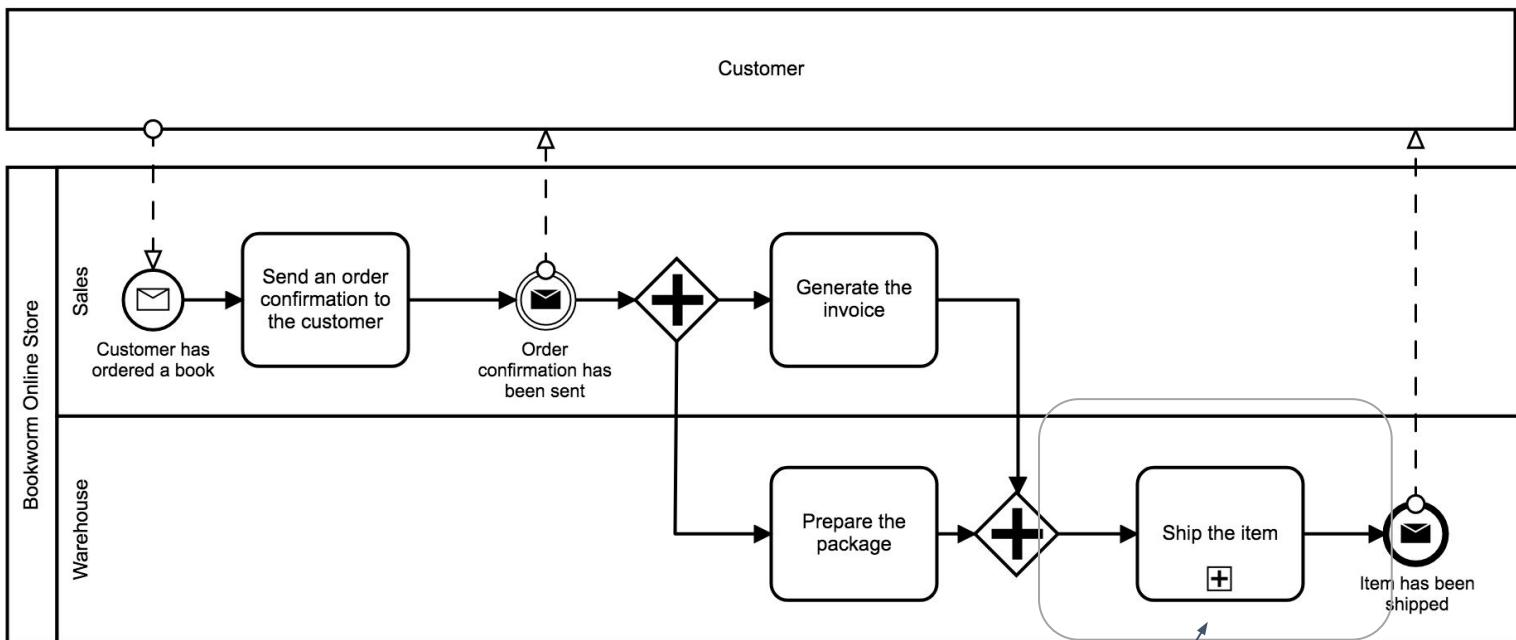


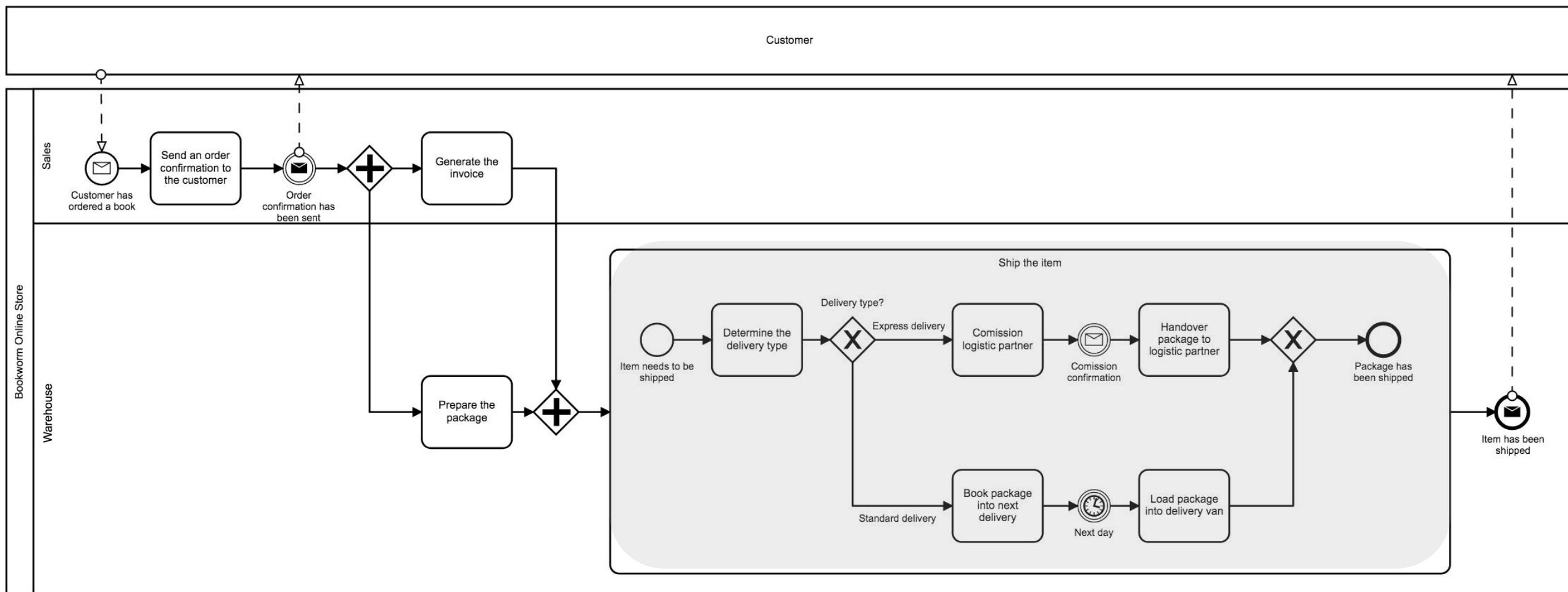


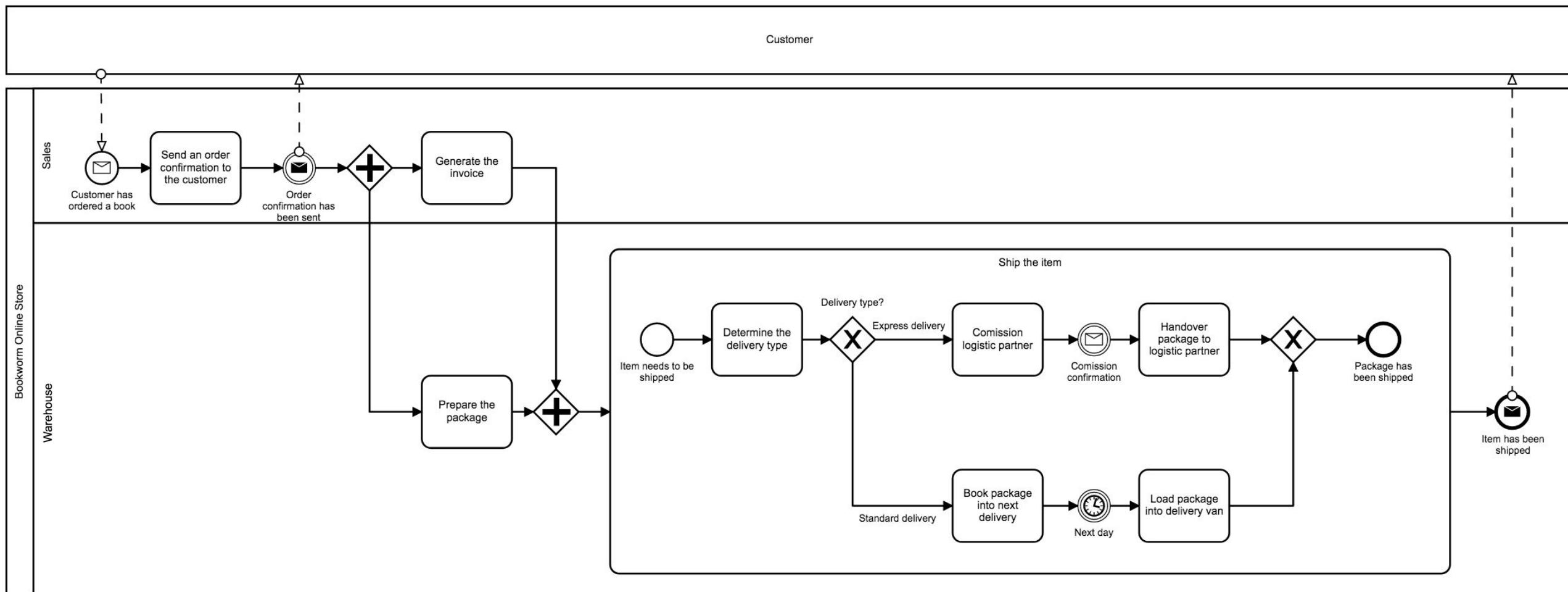


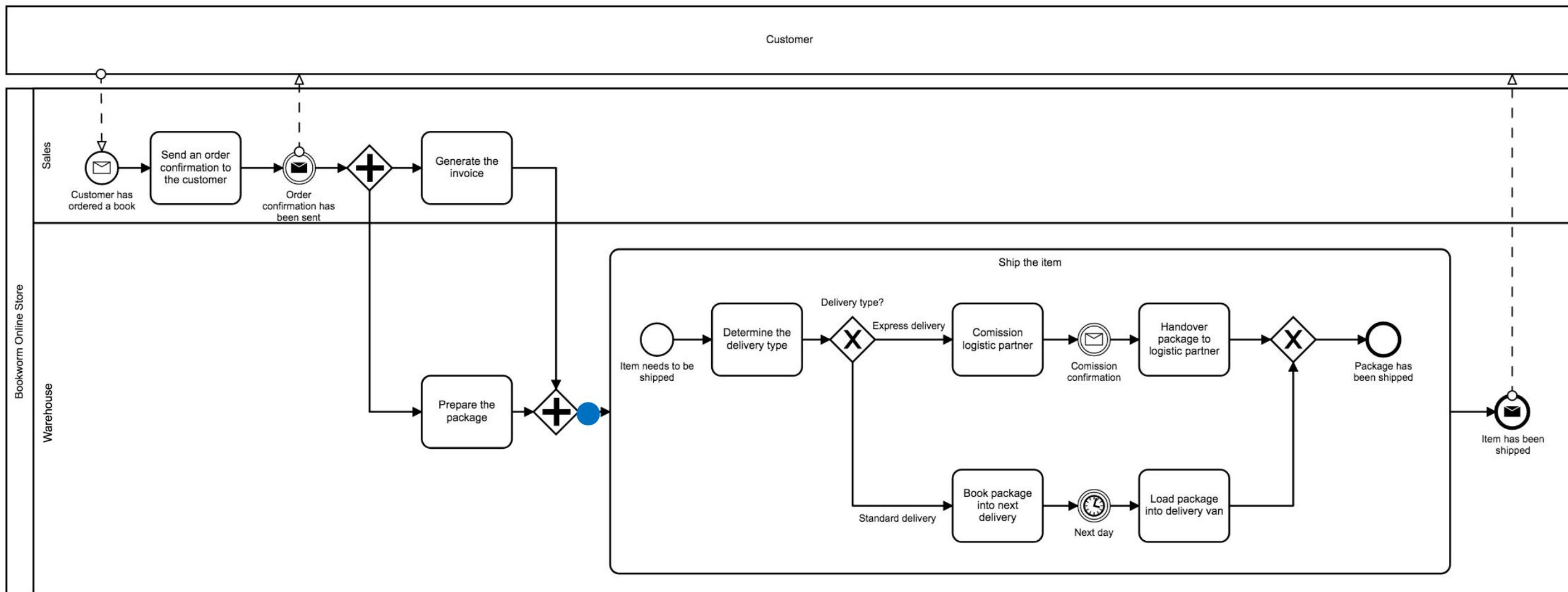


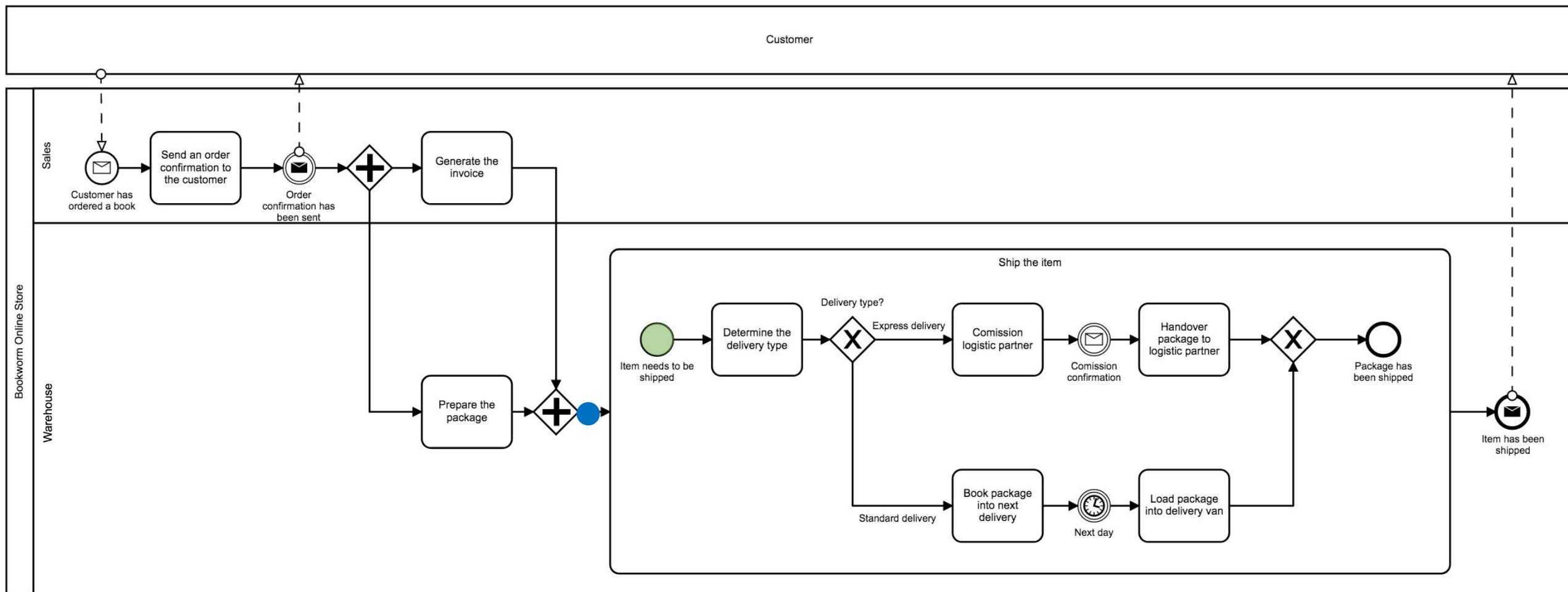


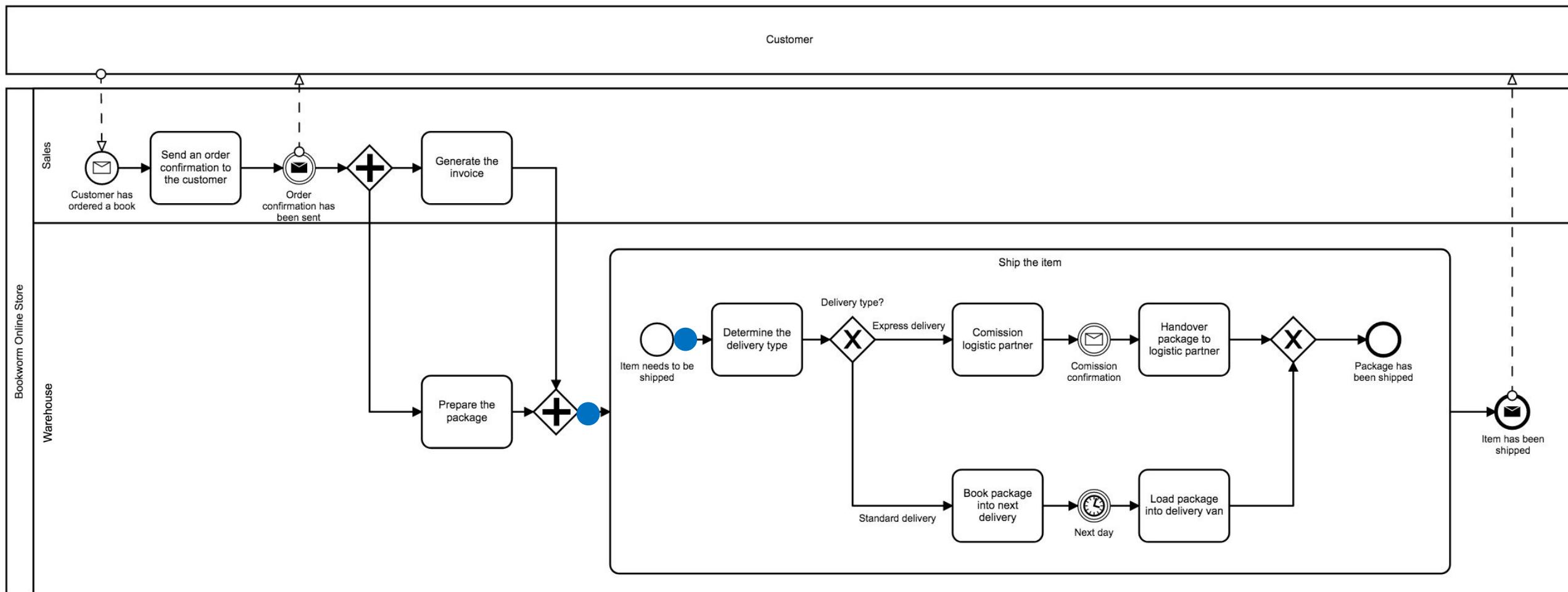


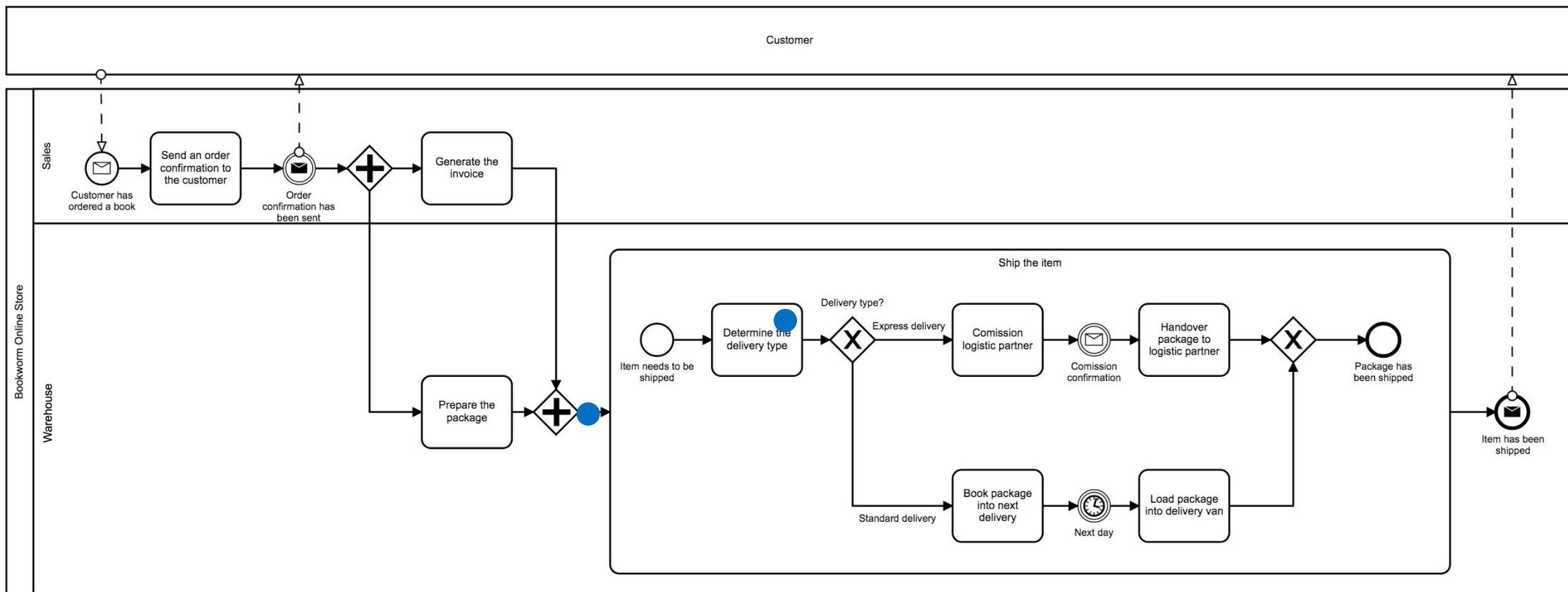


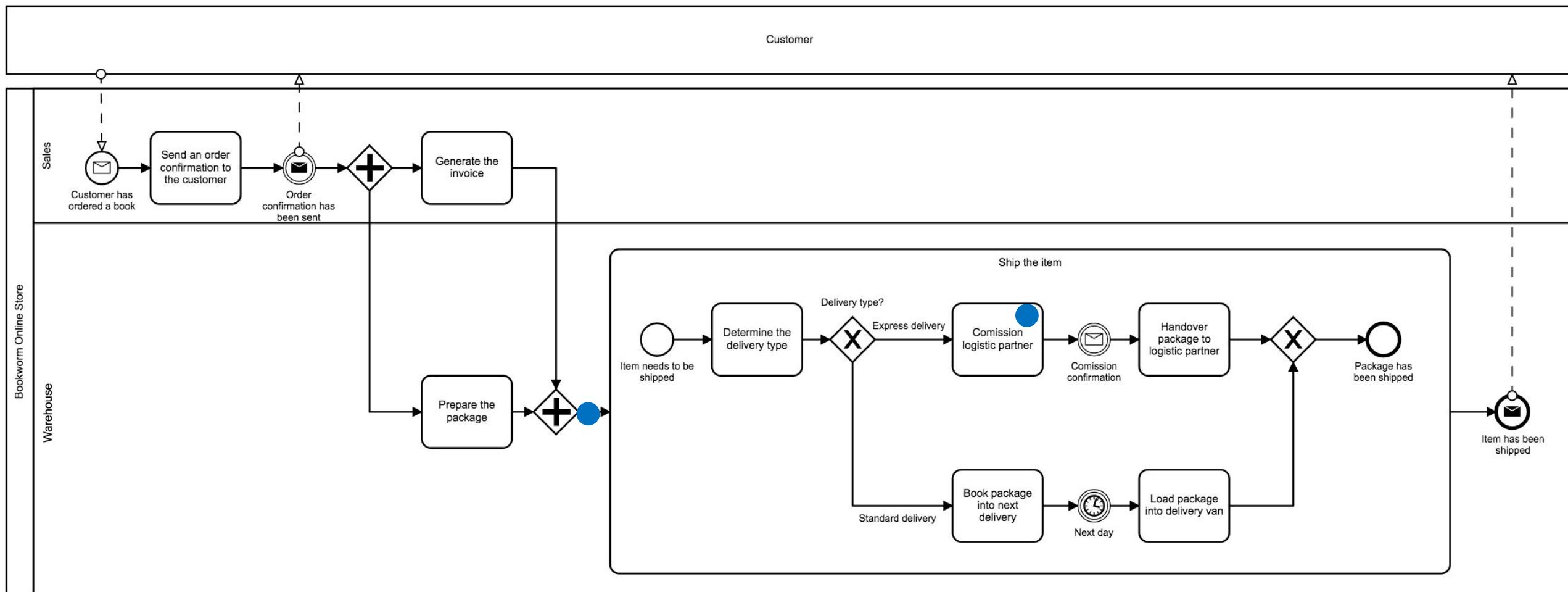


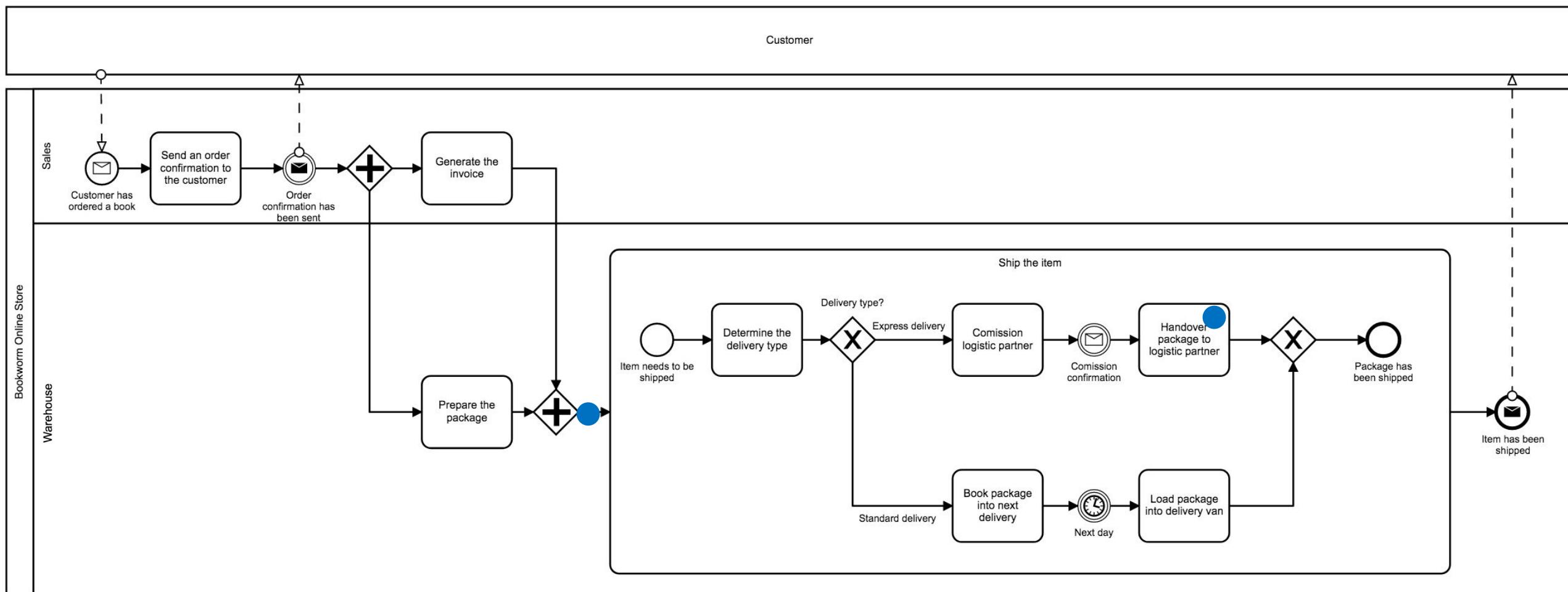


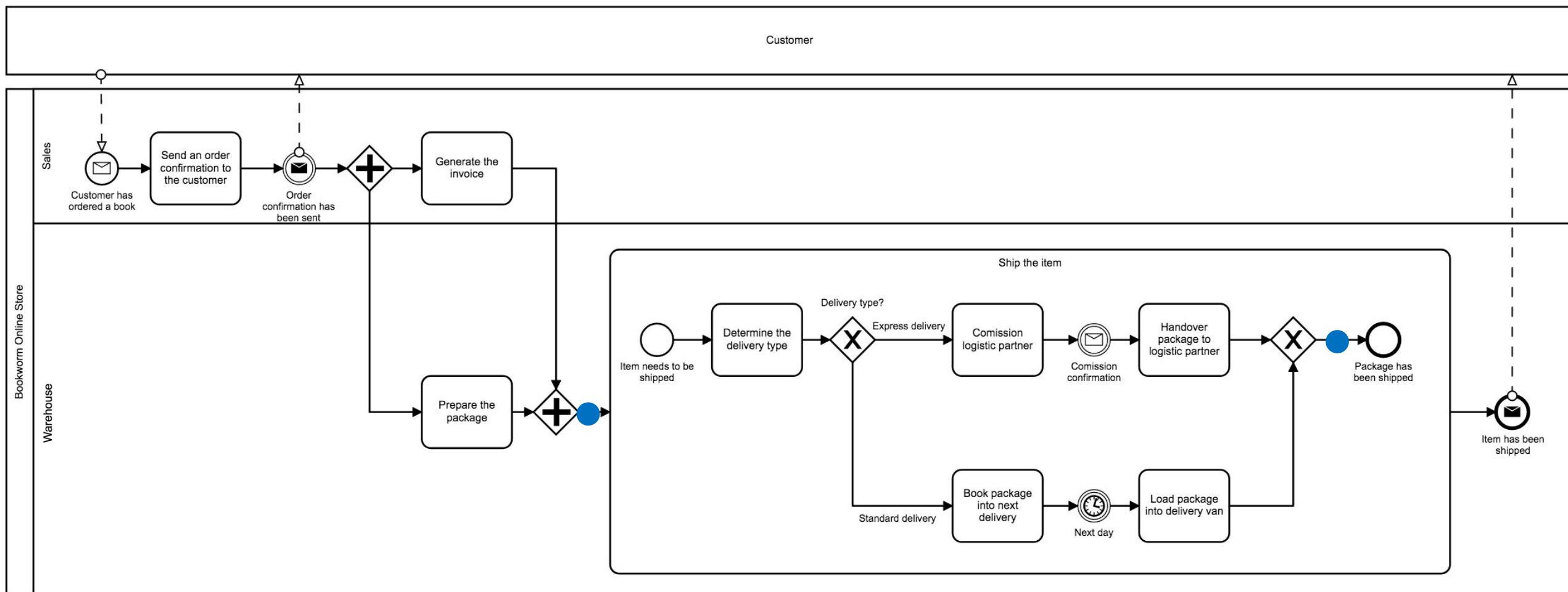


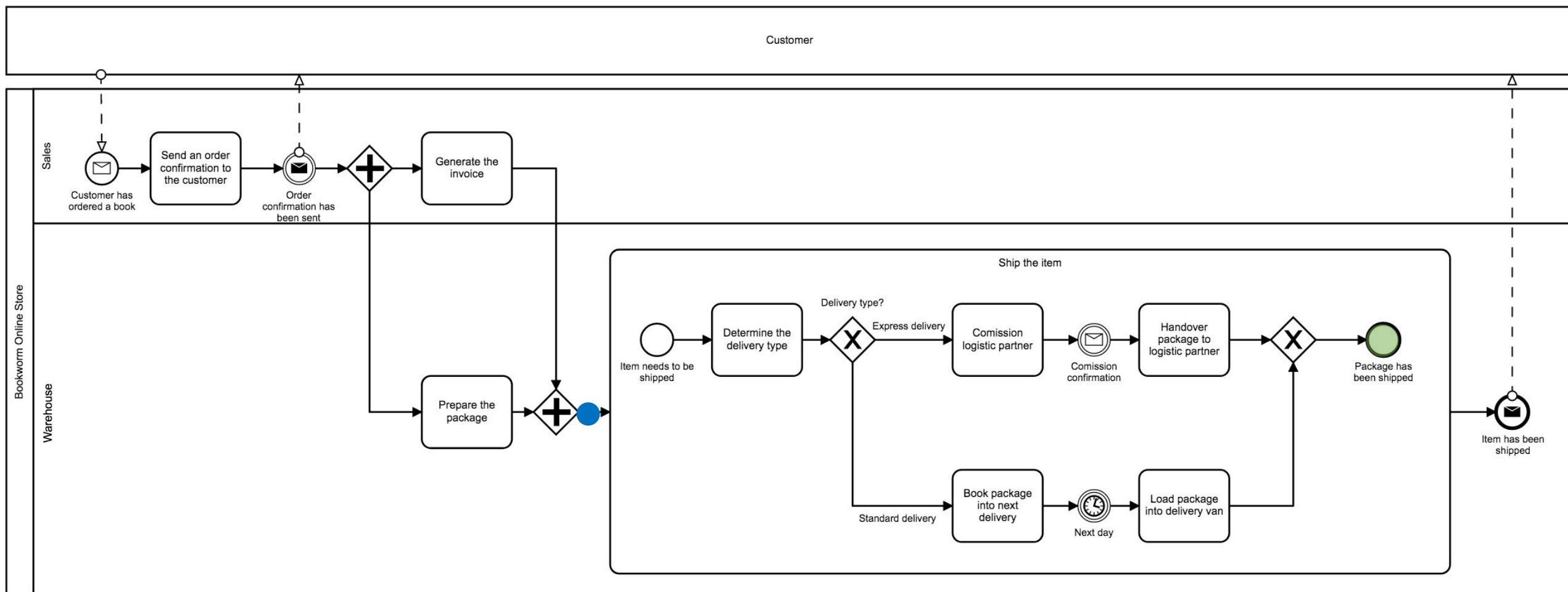


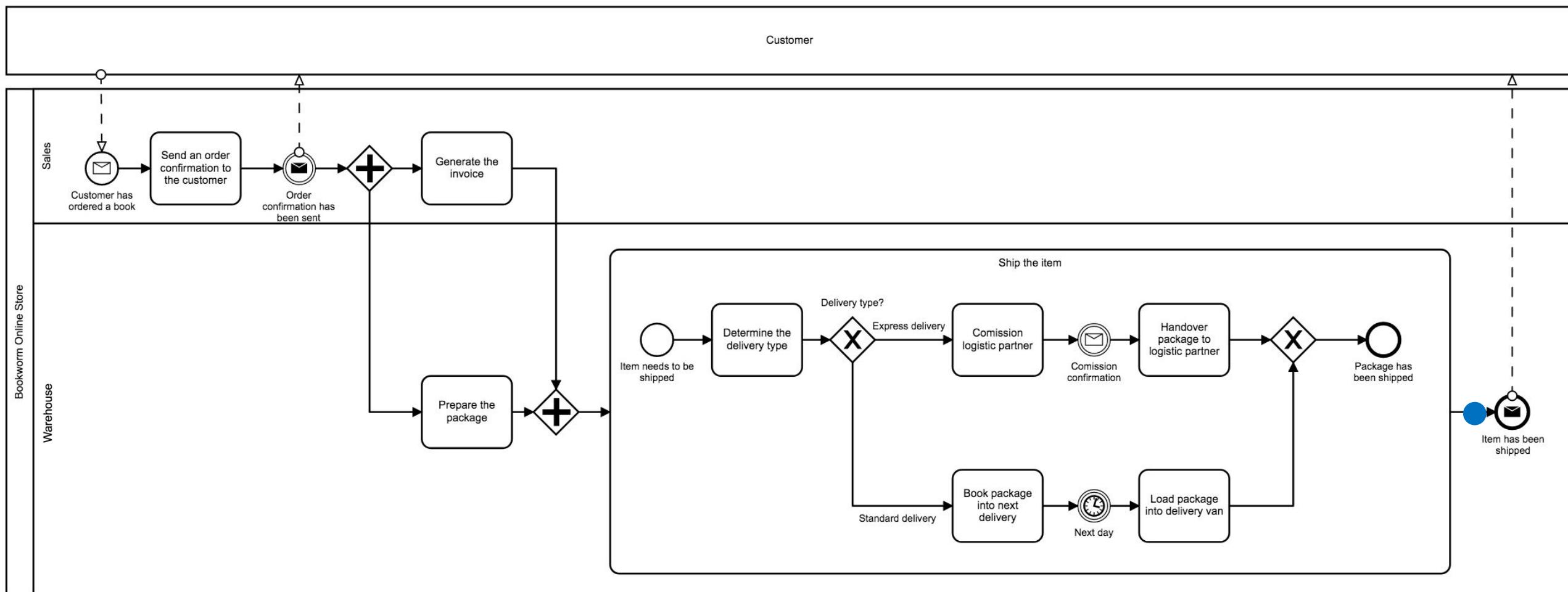








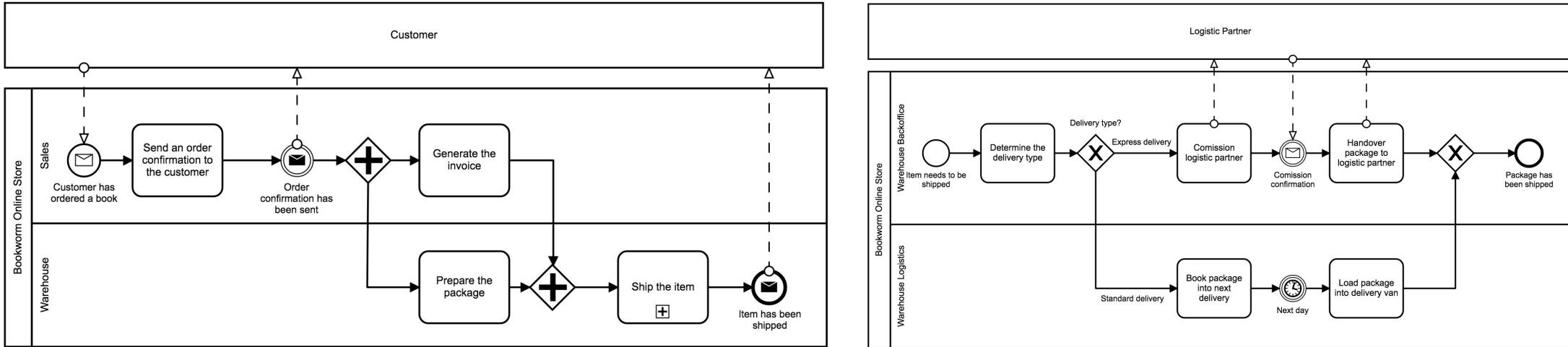




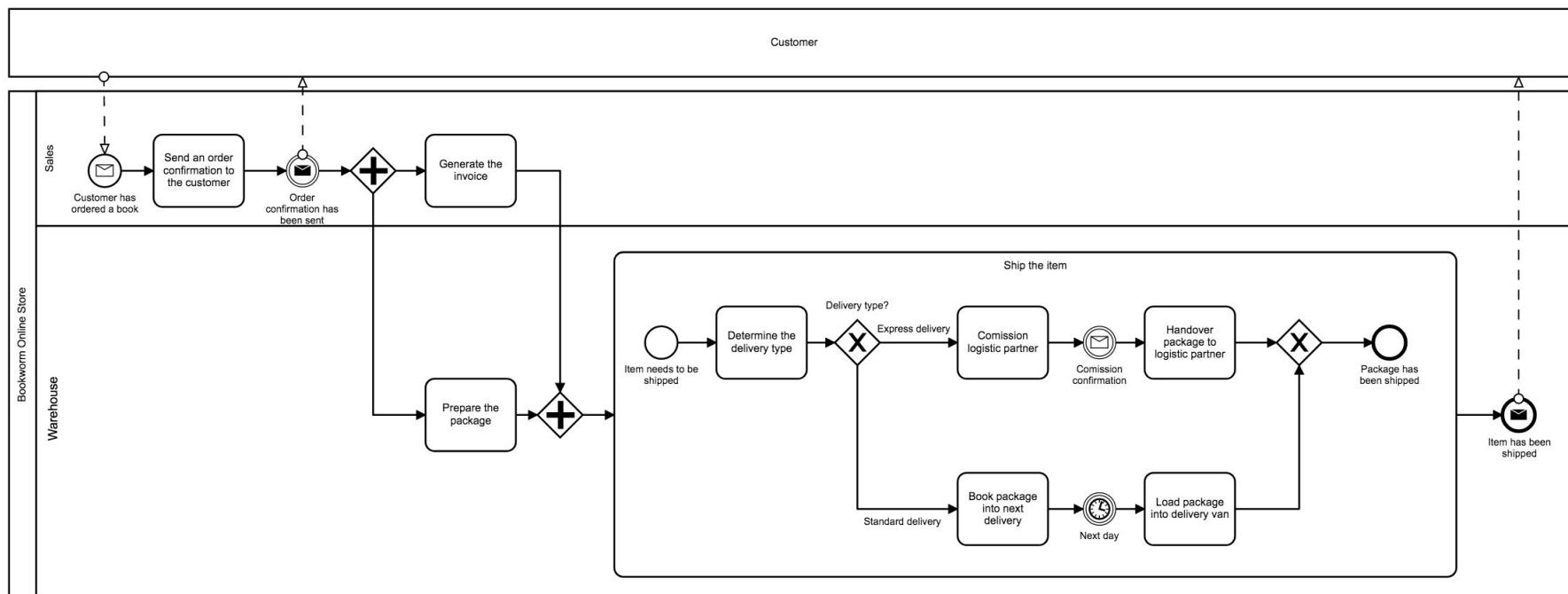


When should I use the expanded sub process?

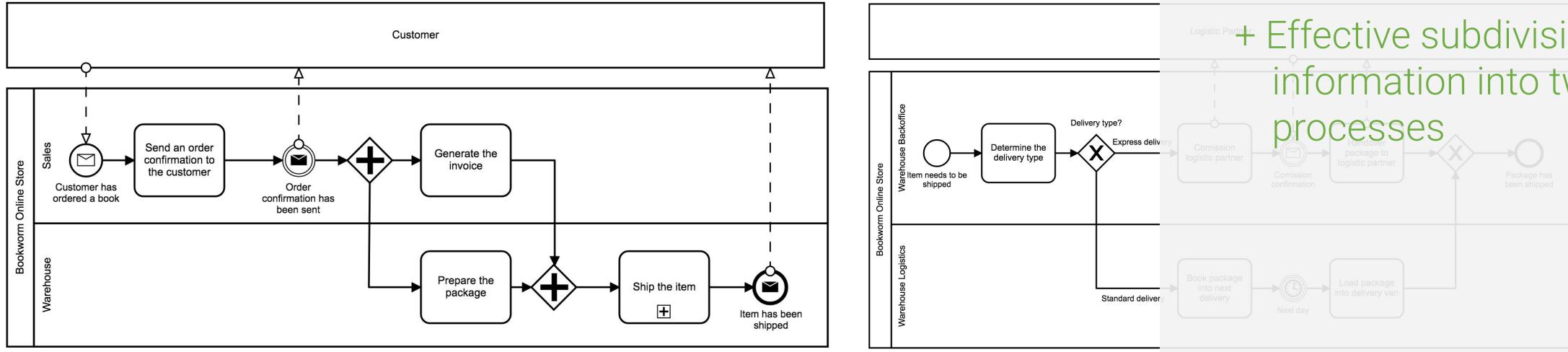
Collapsed



Expanded

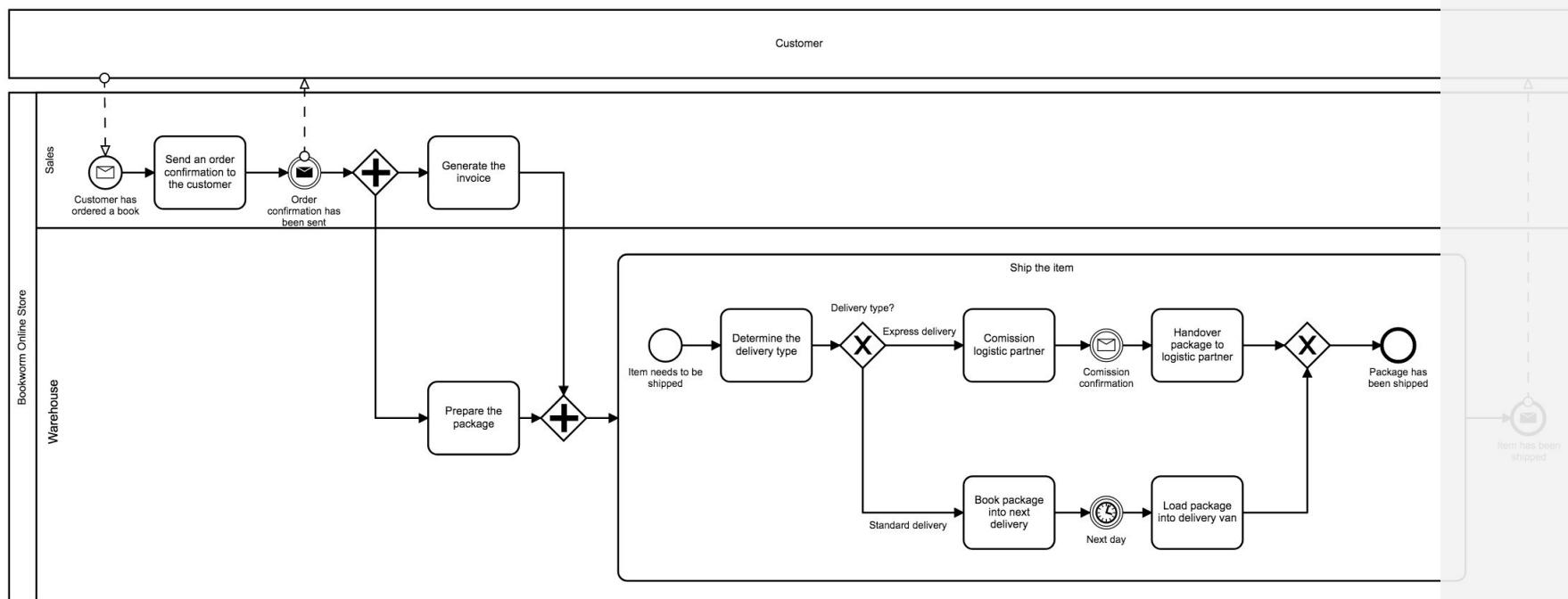


Collapsed

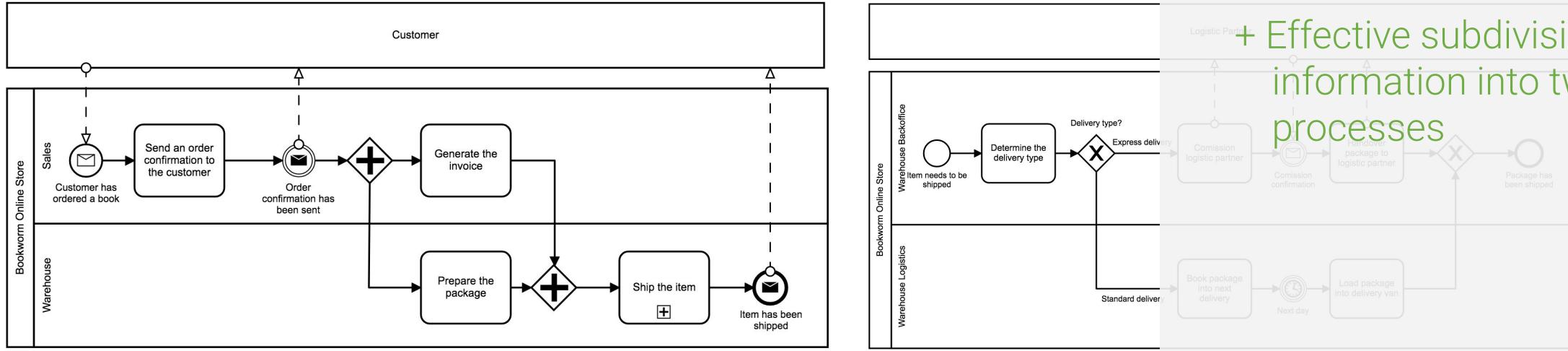


+ Effective subdivision of information into two processes

Expanded

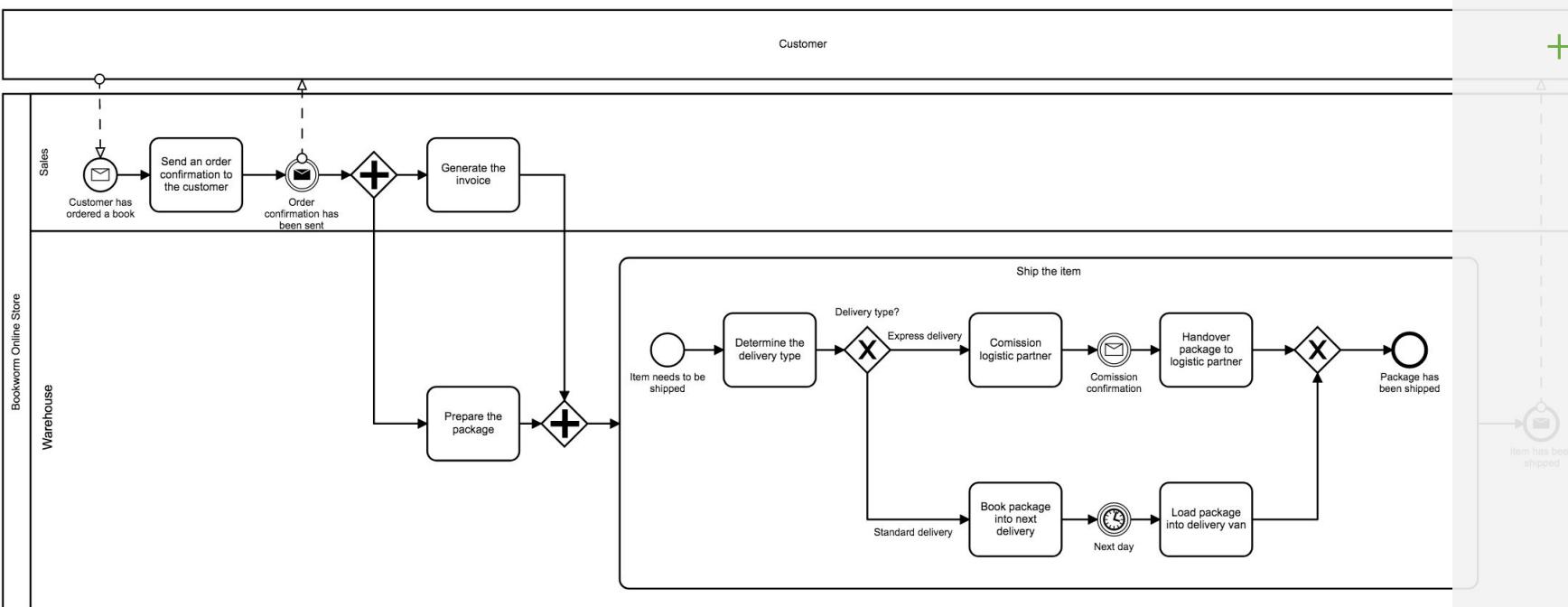


Collapsed



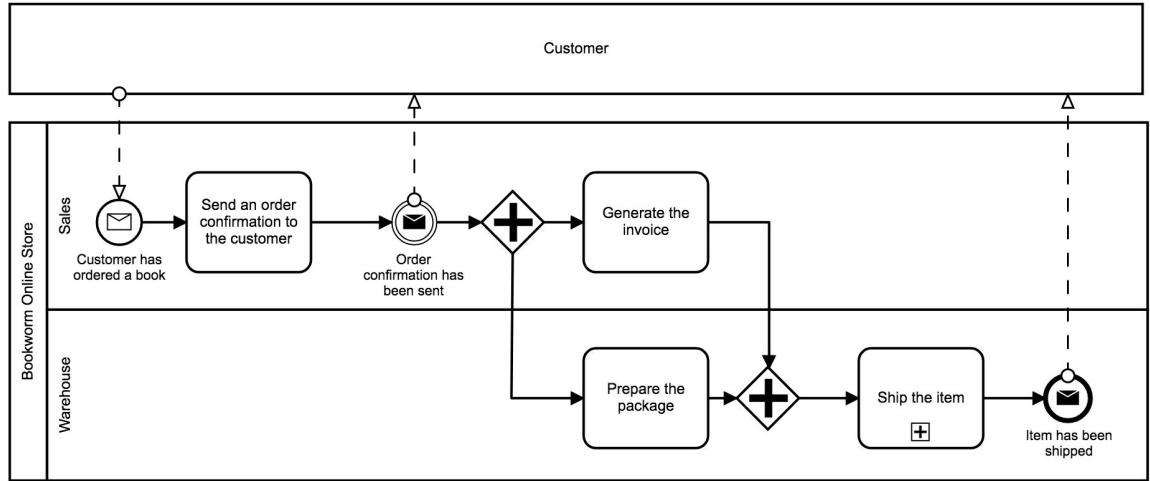
+ Effective subdivision of information into two processes

Expanded

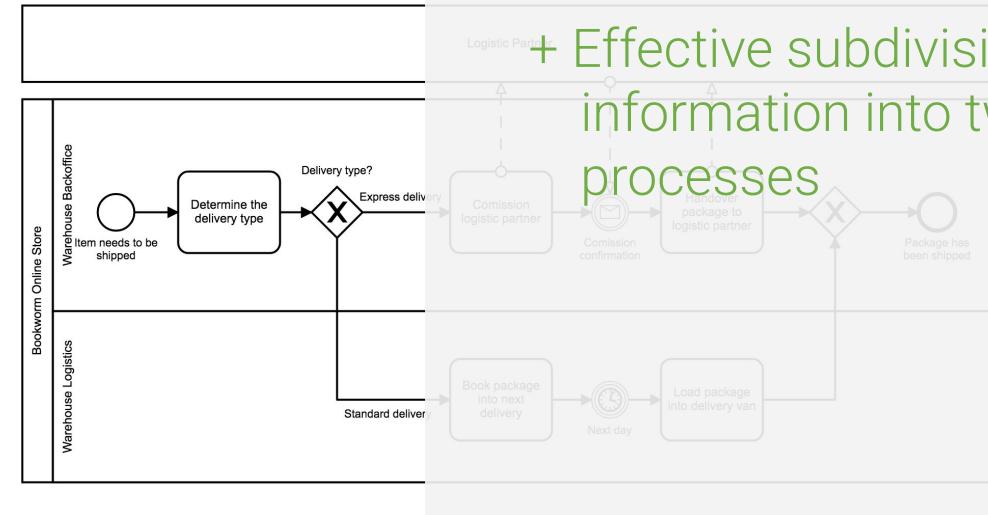


+ Subprocess and parent process information in one place

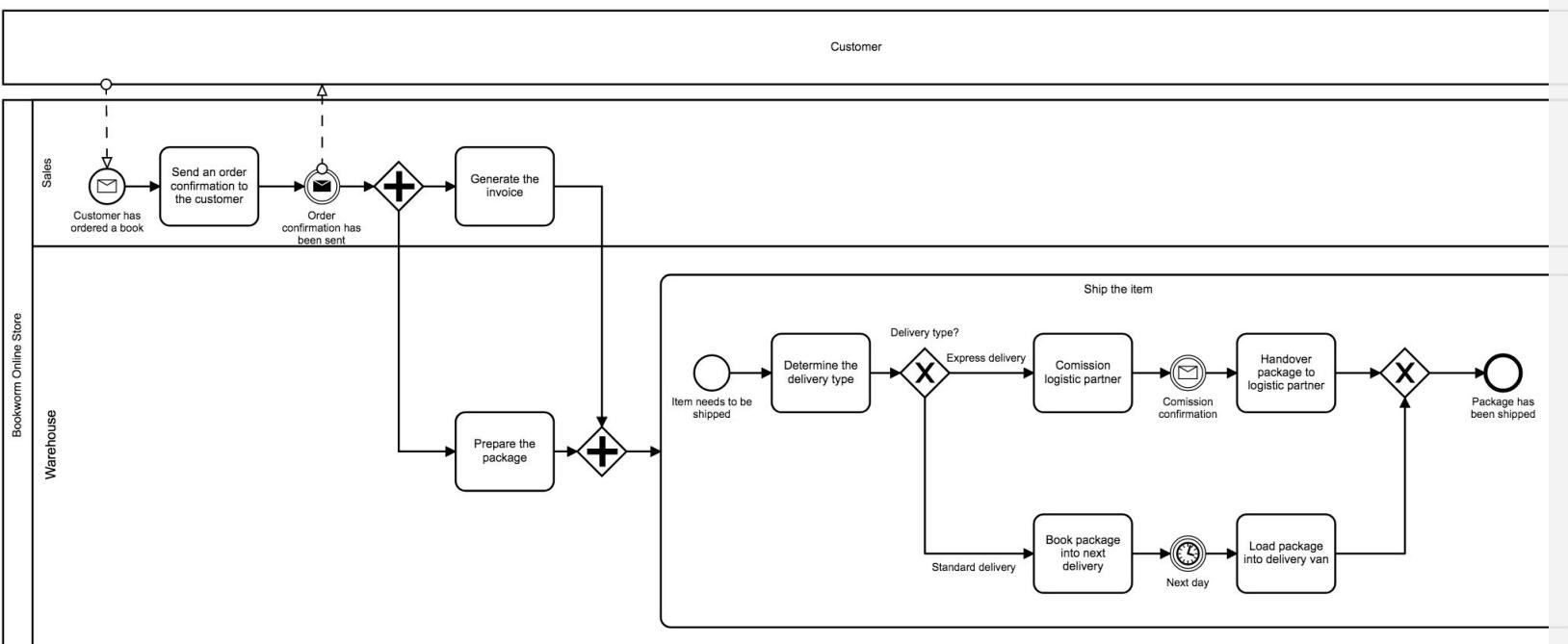
Collapsed



+ Effective subdivision of information into two processes



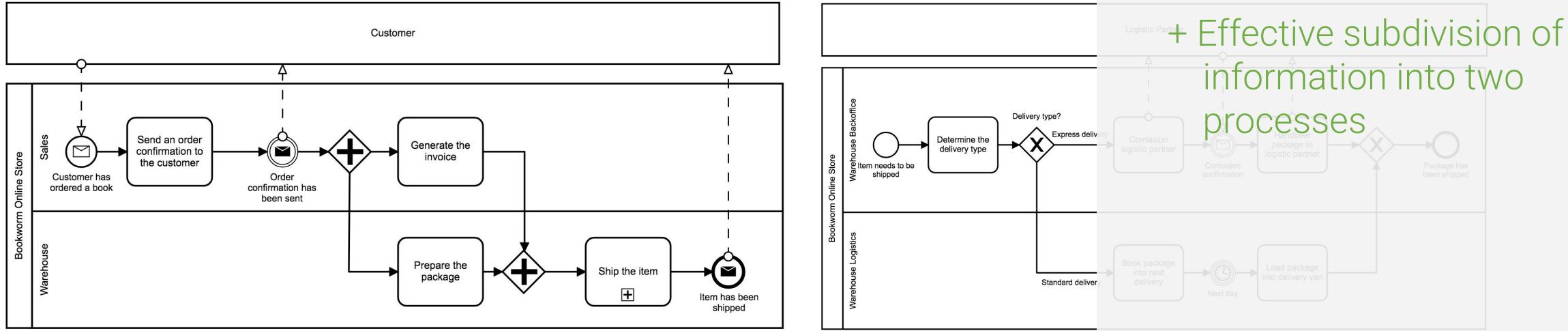
Expanded



+ Subprocess and parent process information in one place

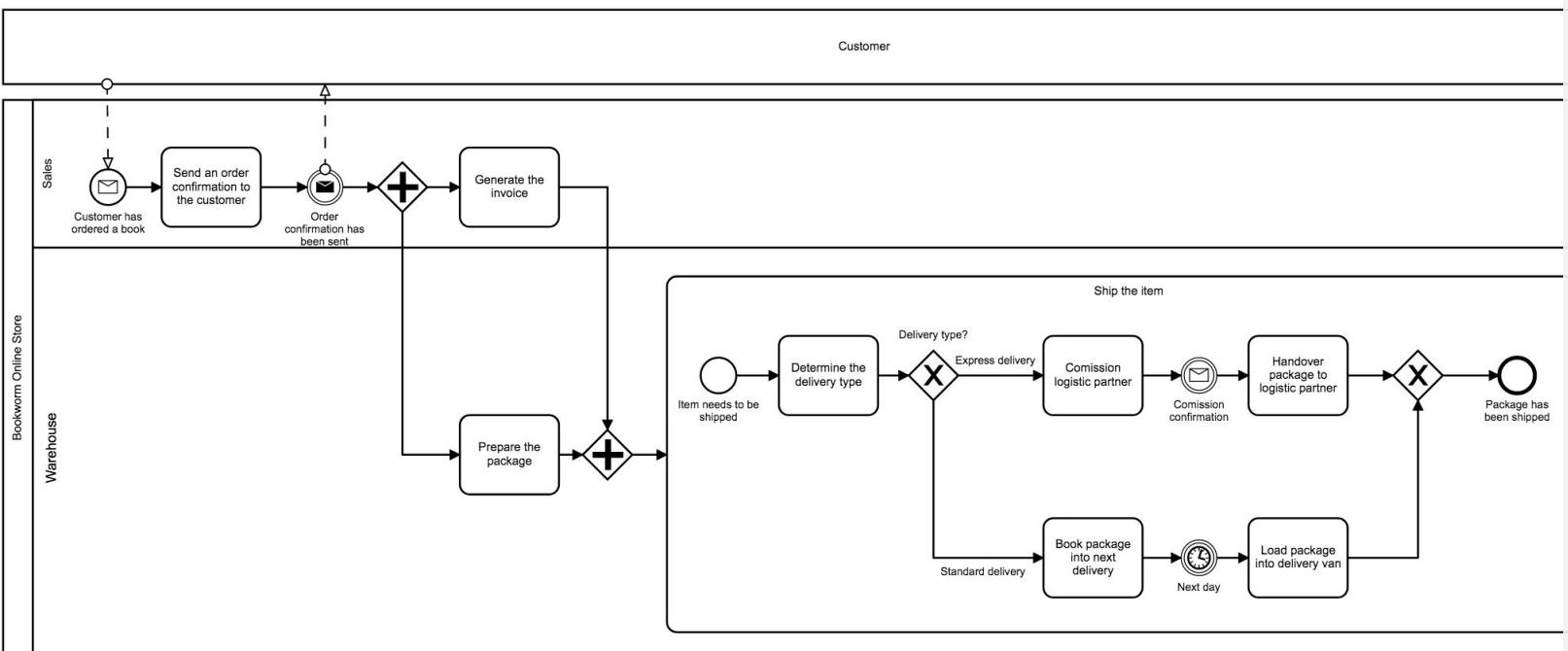
+ Apply attached events and/or loops to multiple tasks

Collapsed



+ Effective subdivision of information into two processes

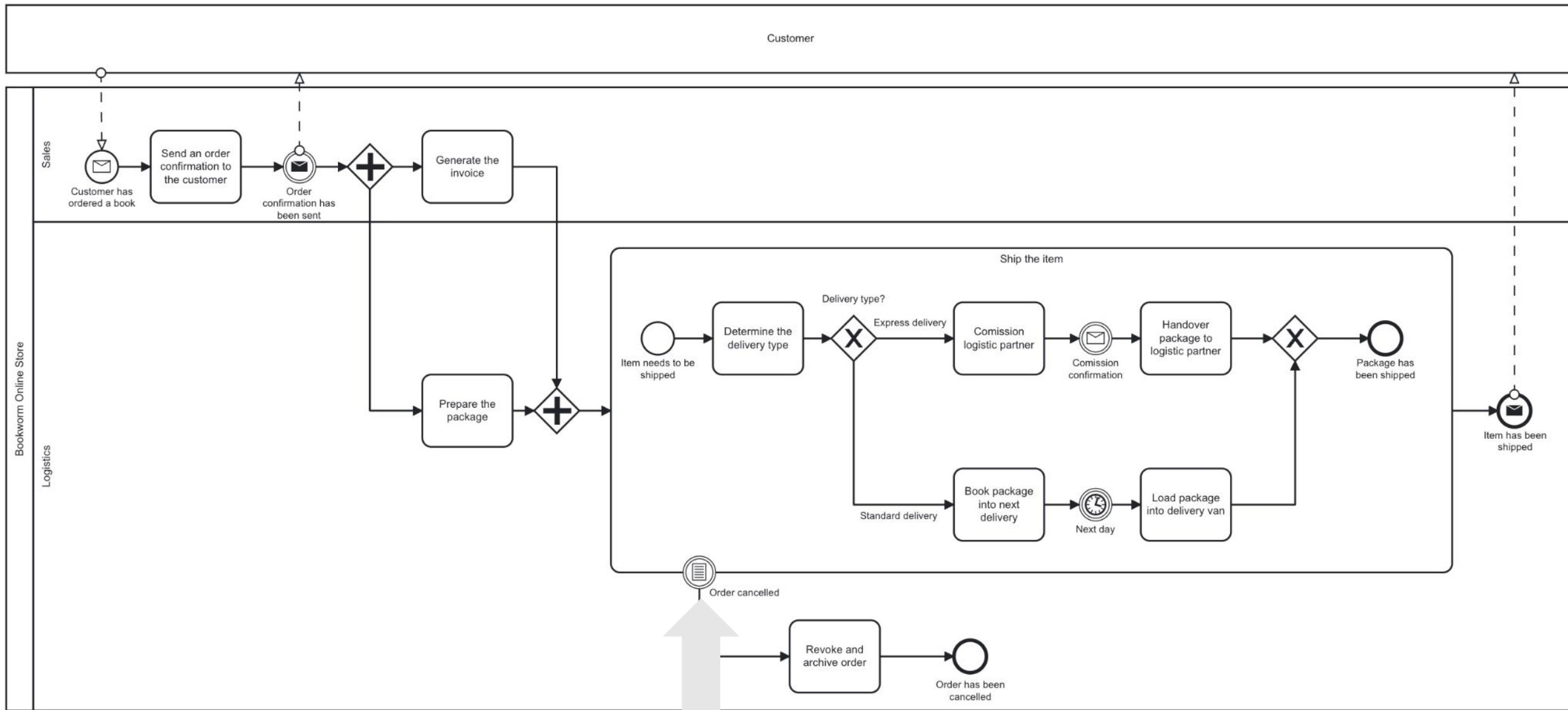
Expanded

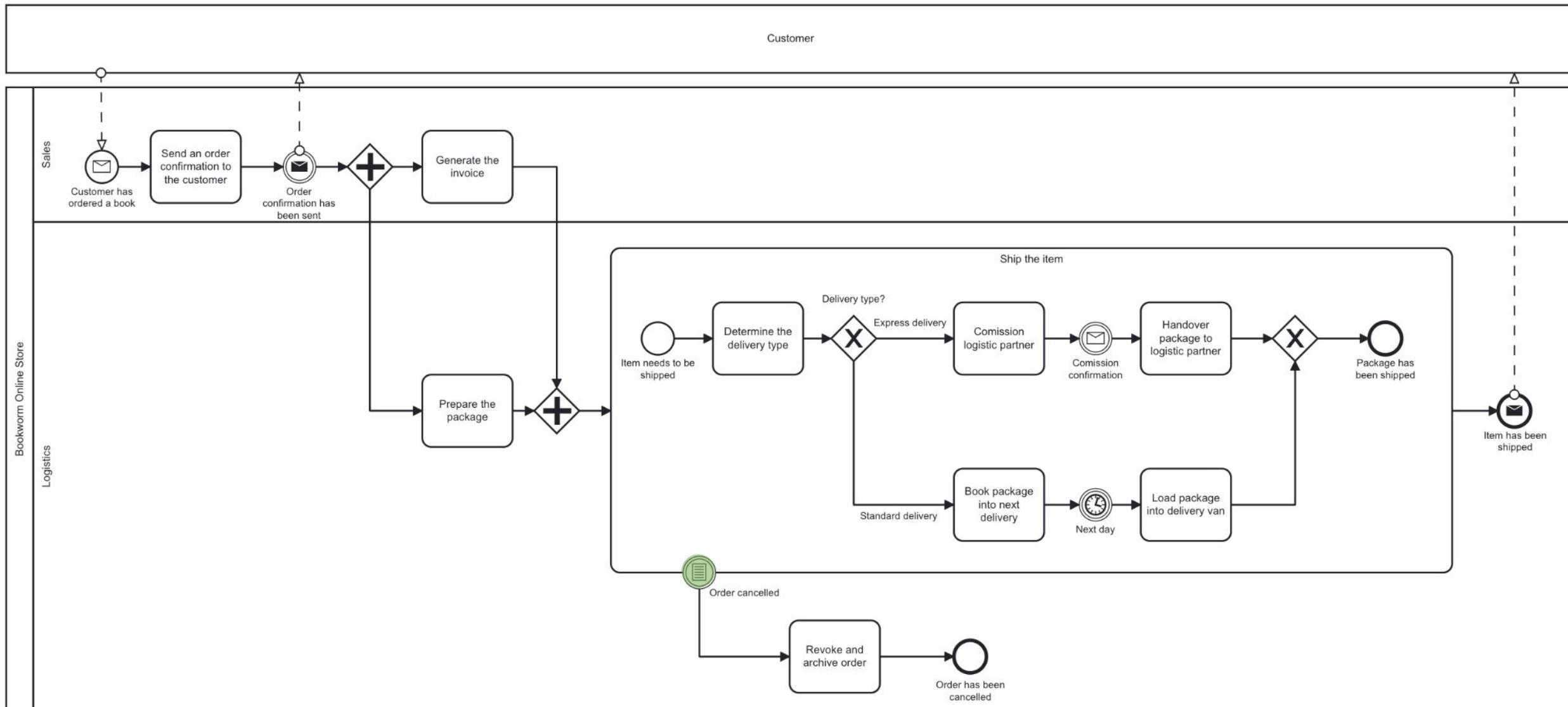


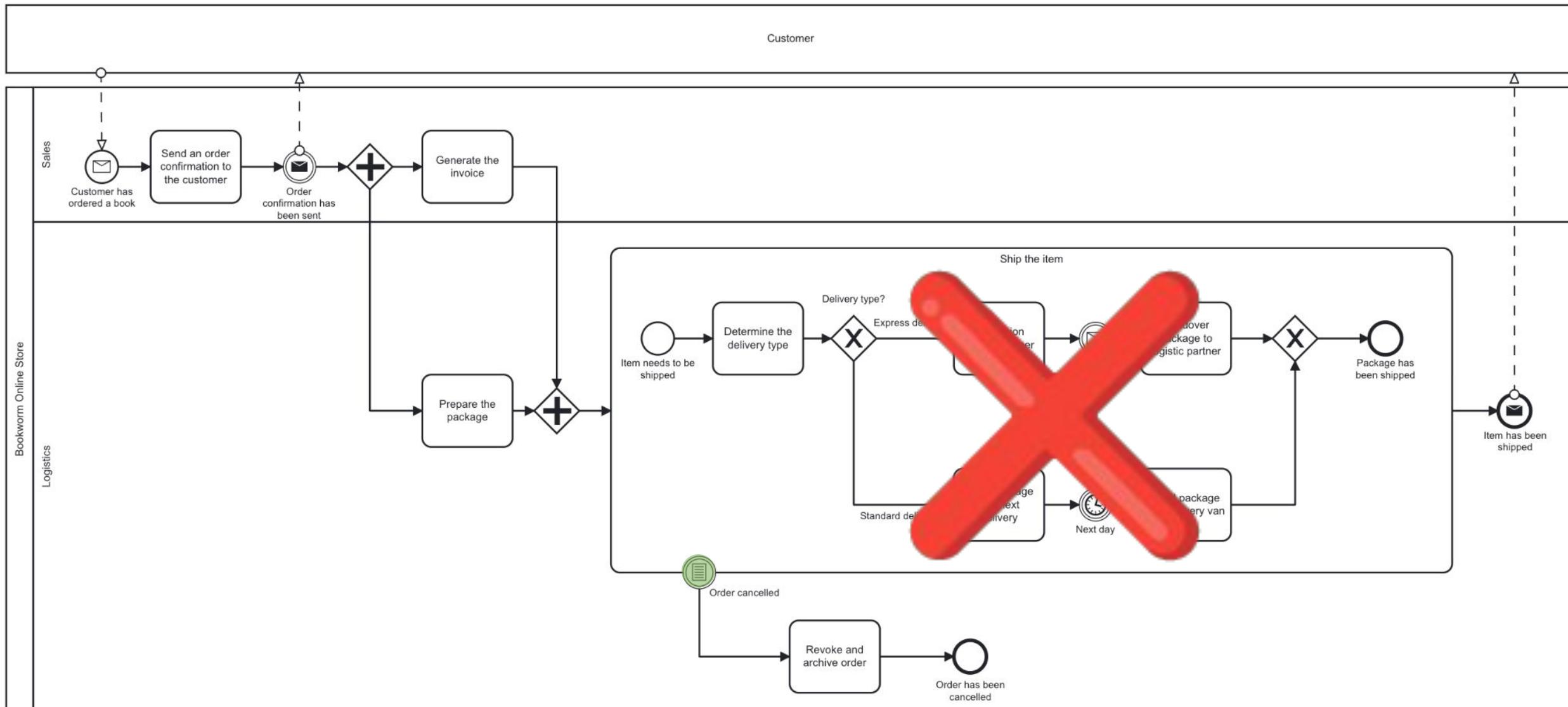
+ Subprocess and parent process information in one place

+ Apply attached events and/or loops to multiple tasks

- Over-complication of the process diagram









Best Practice Rule:

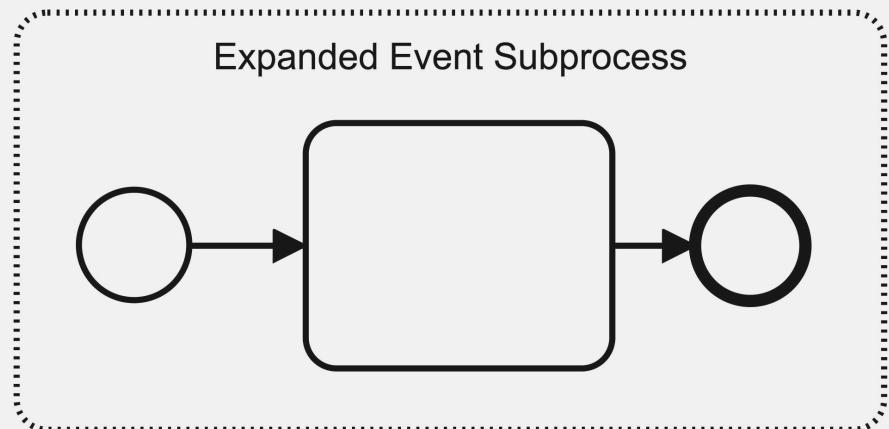
In general, try to use the collapsed subprocess.
But, if you want to apply events or loops to
multiple tasks, use the expanded one.

BPMN Theory

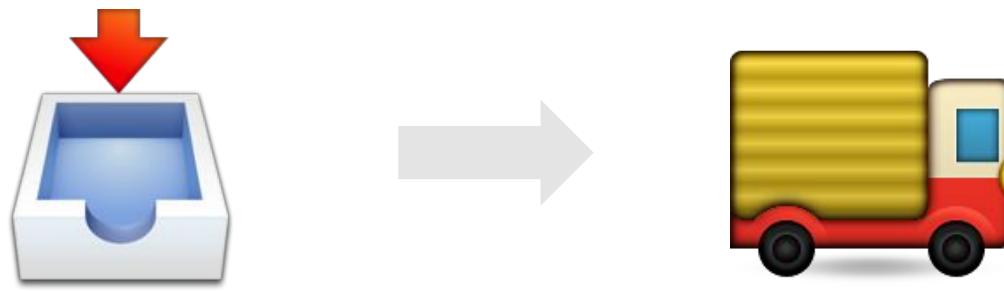
process camp

Event Subprocess

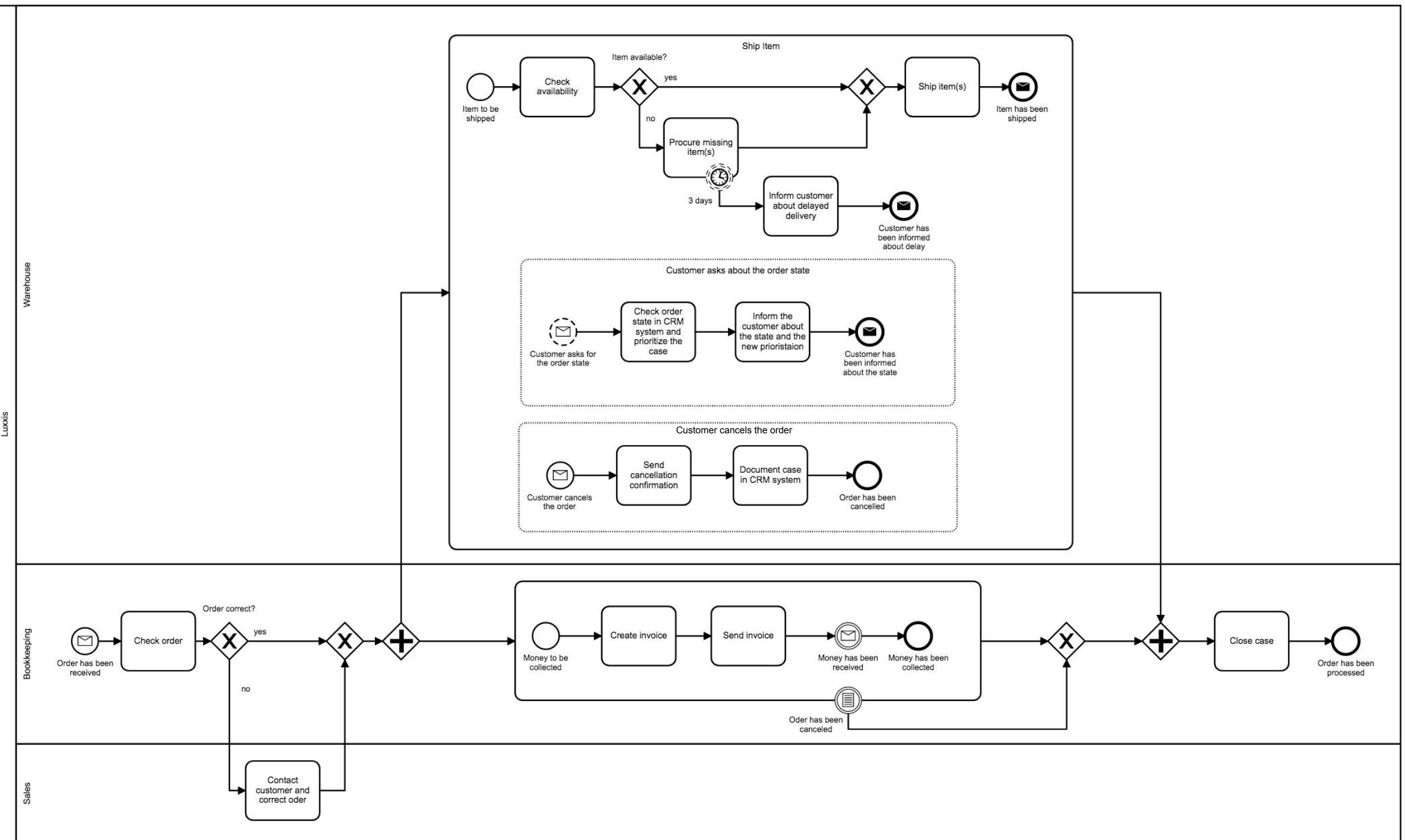
[Expanded]

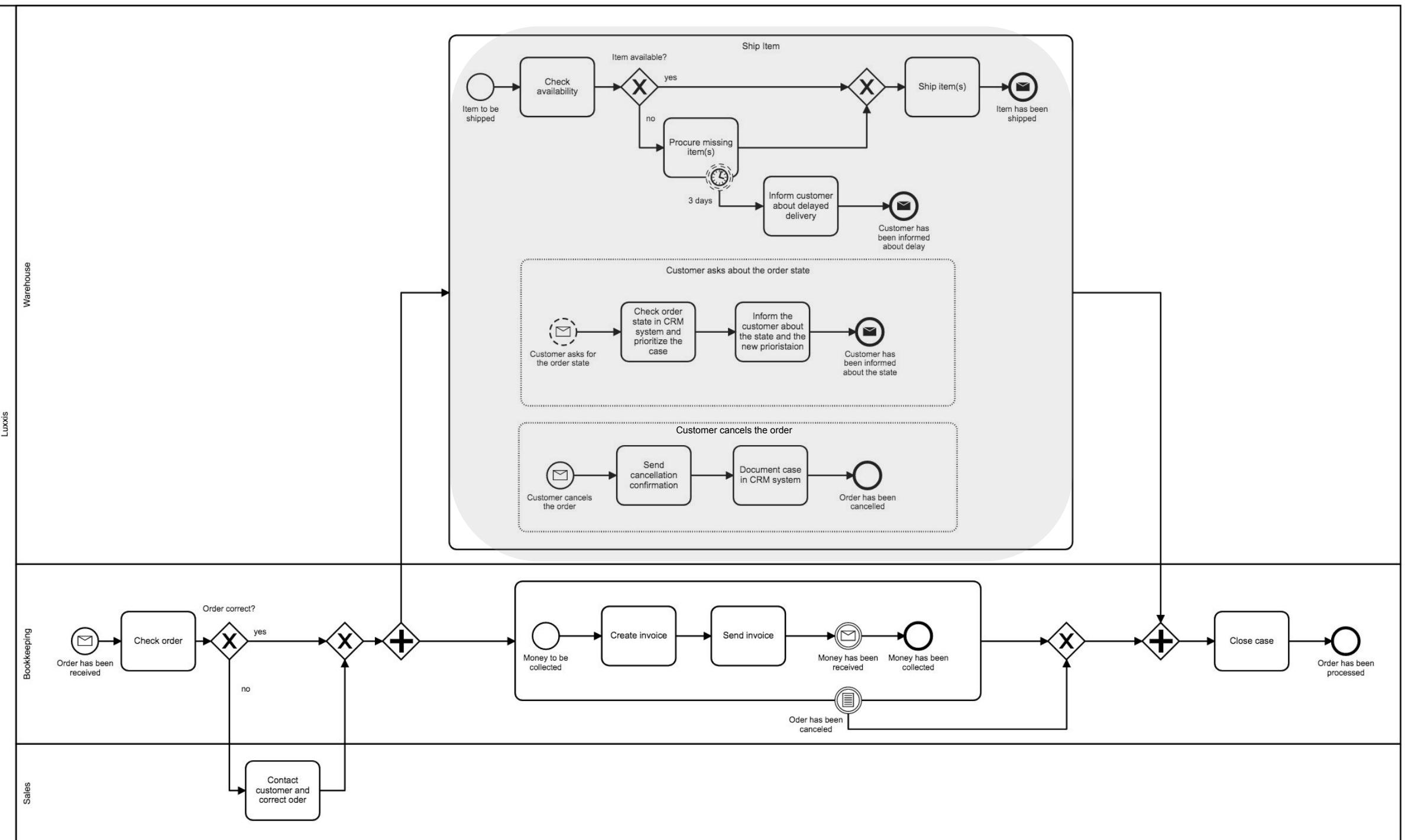


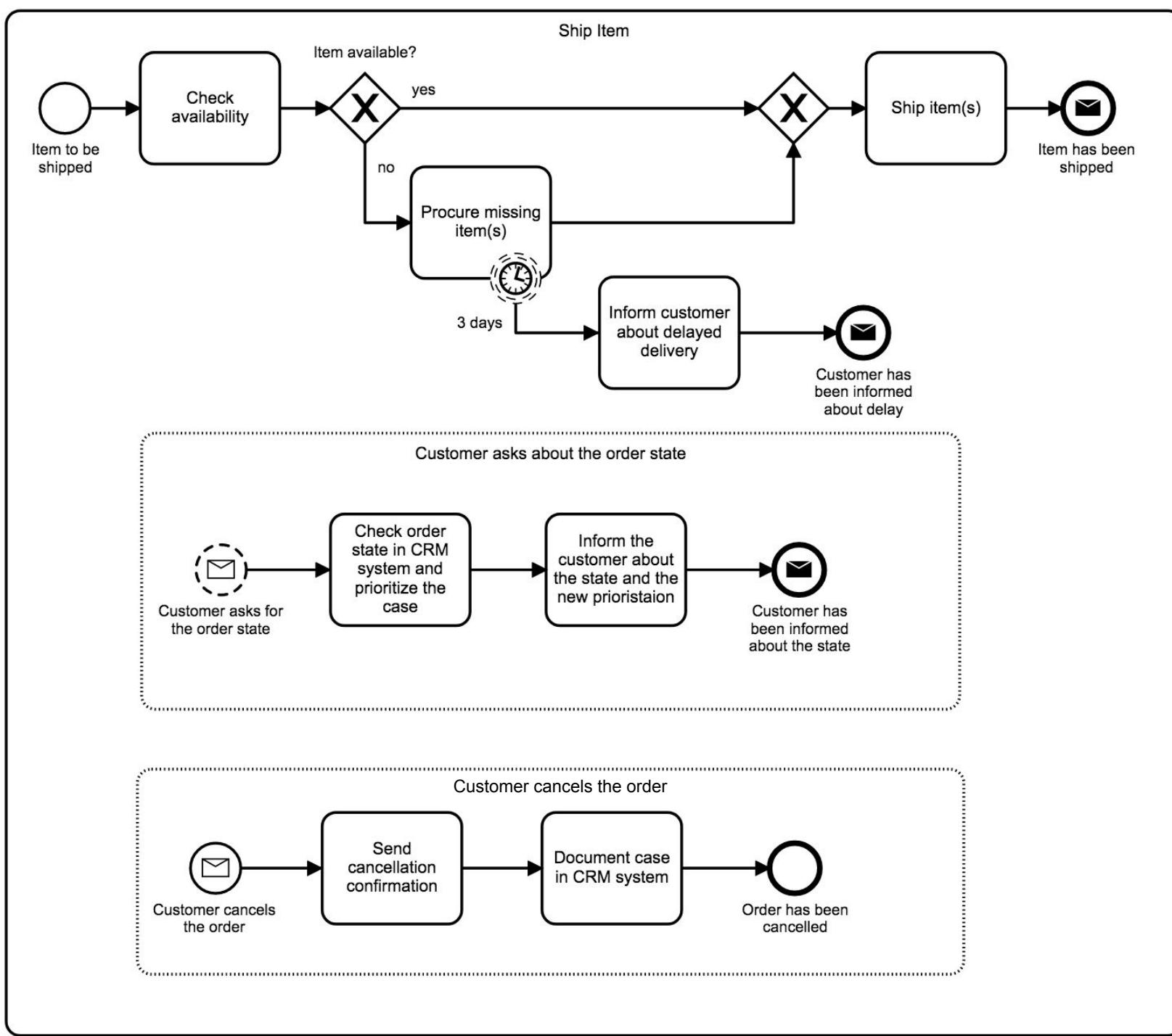
- A subprocess that handles exceptions
- Embedded in the parent process or in a subprocess

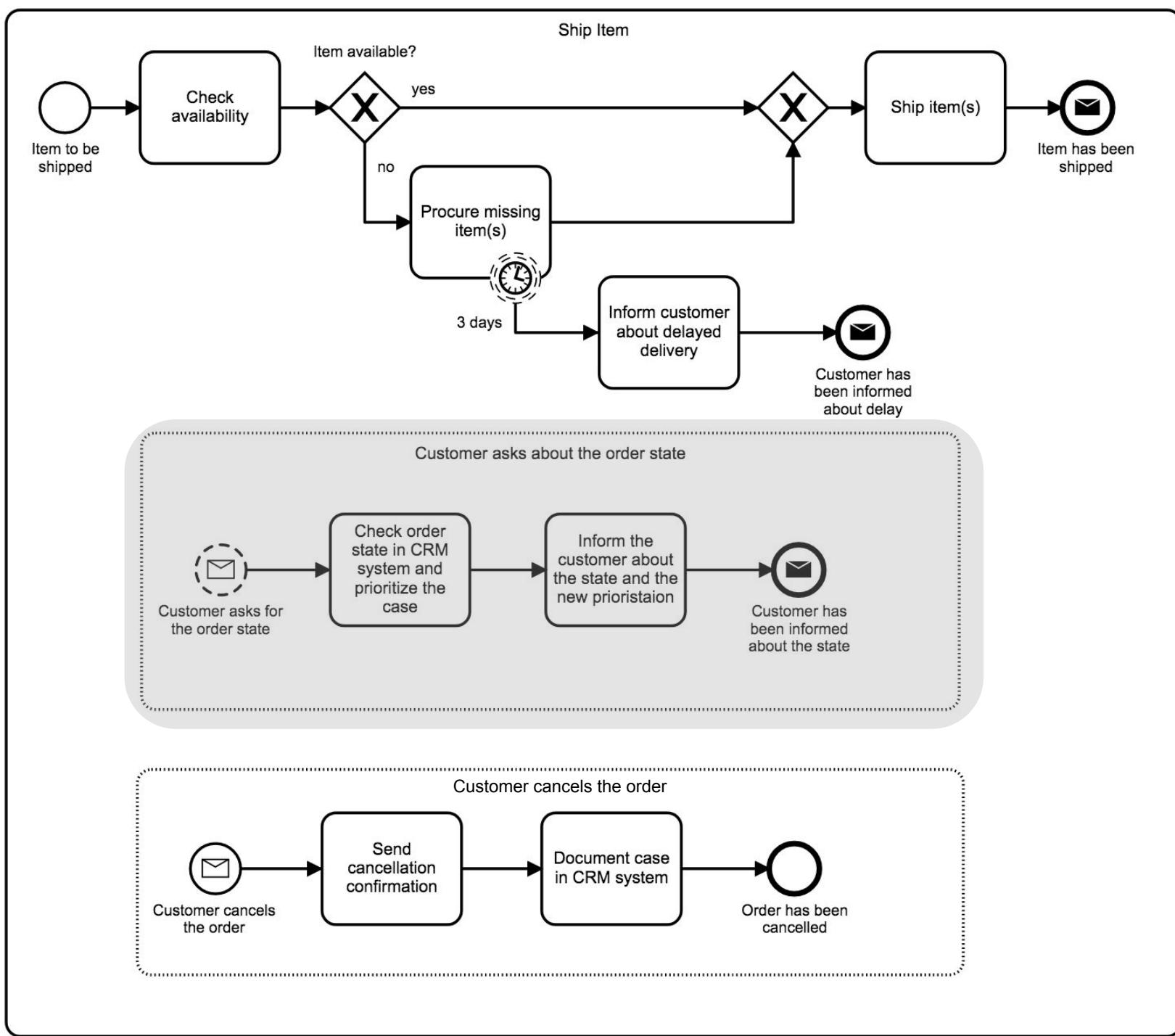


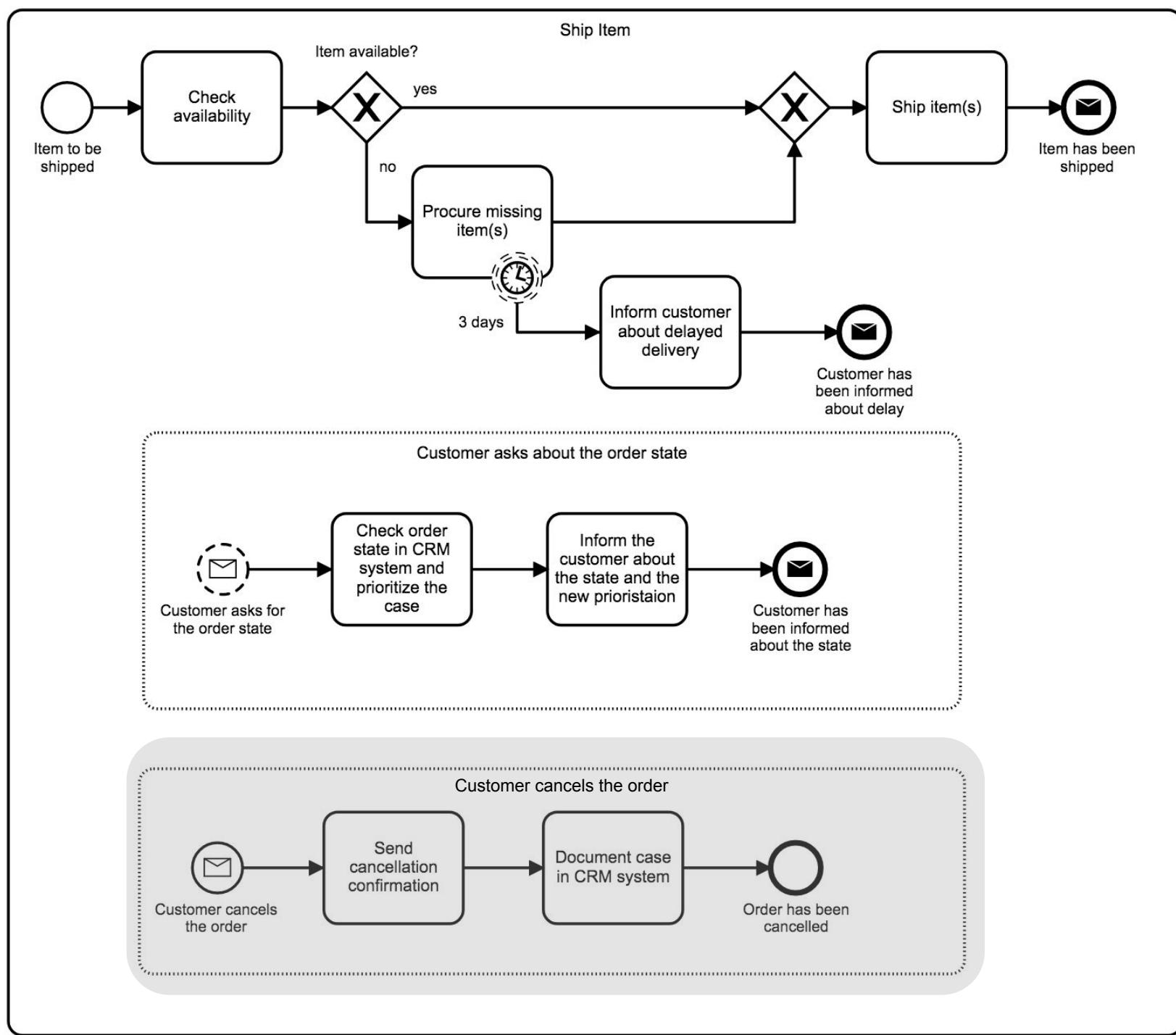
The order process at Luxxis





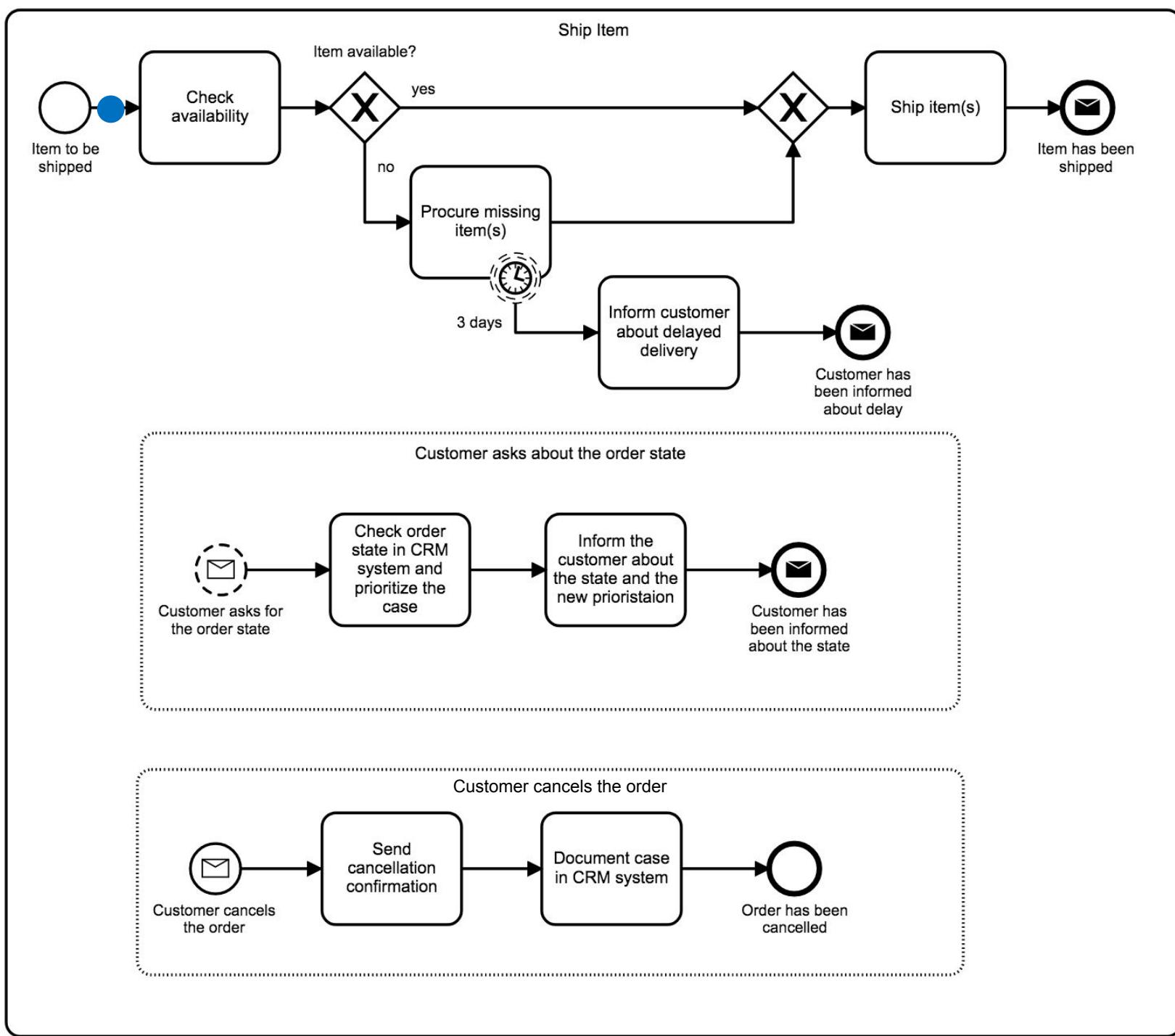


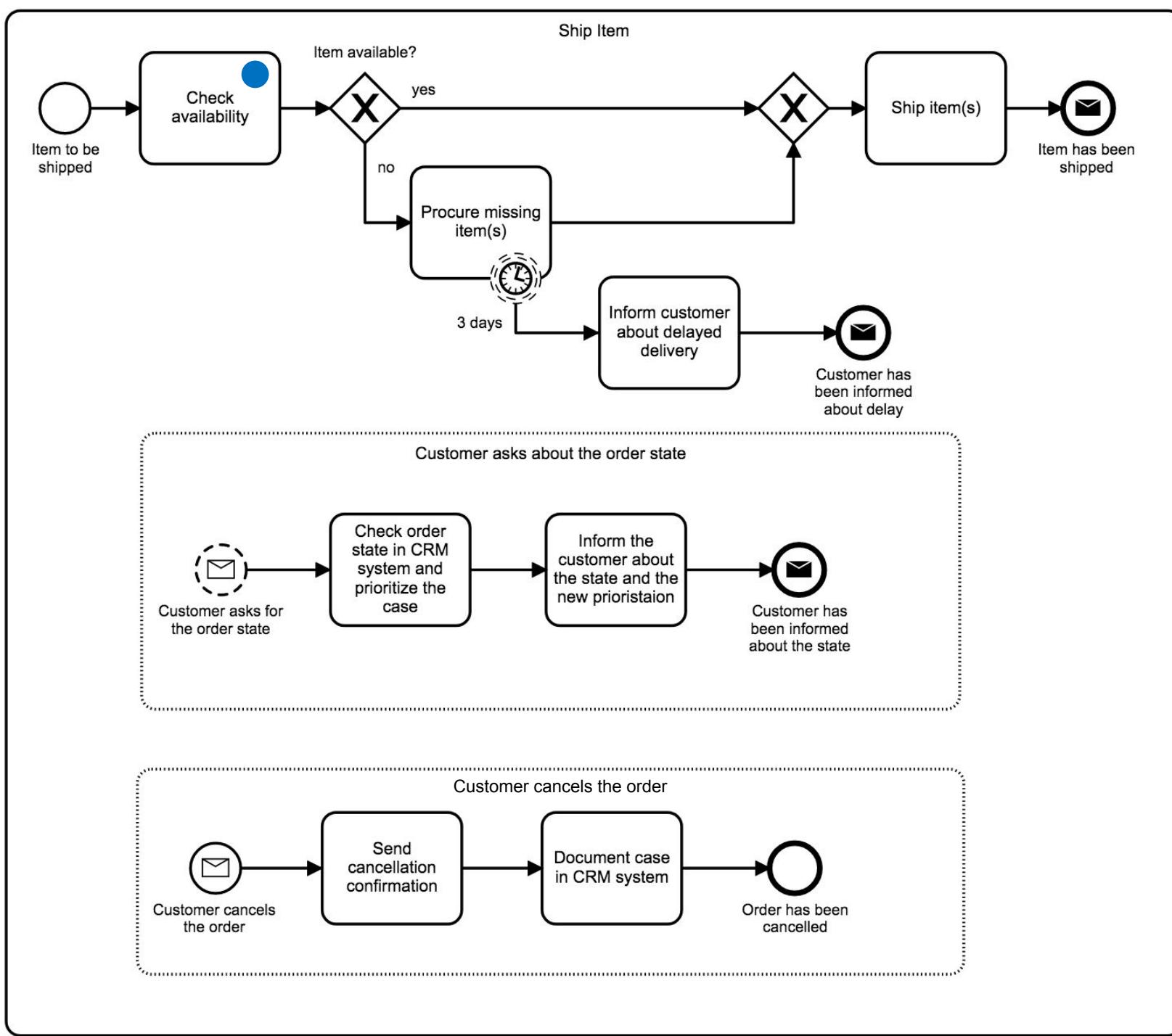


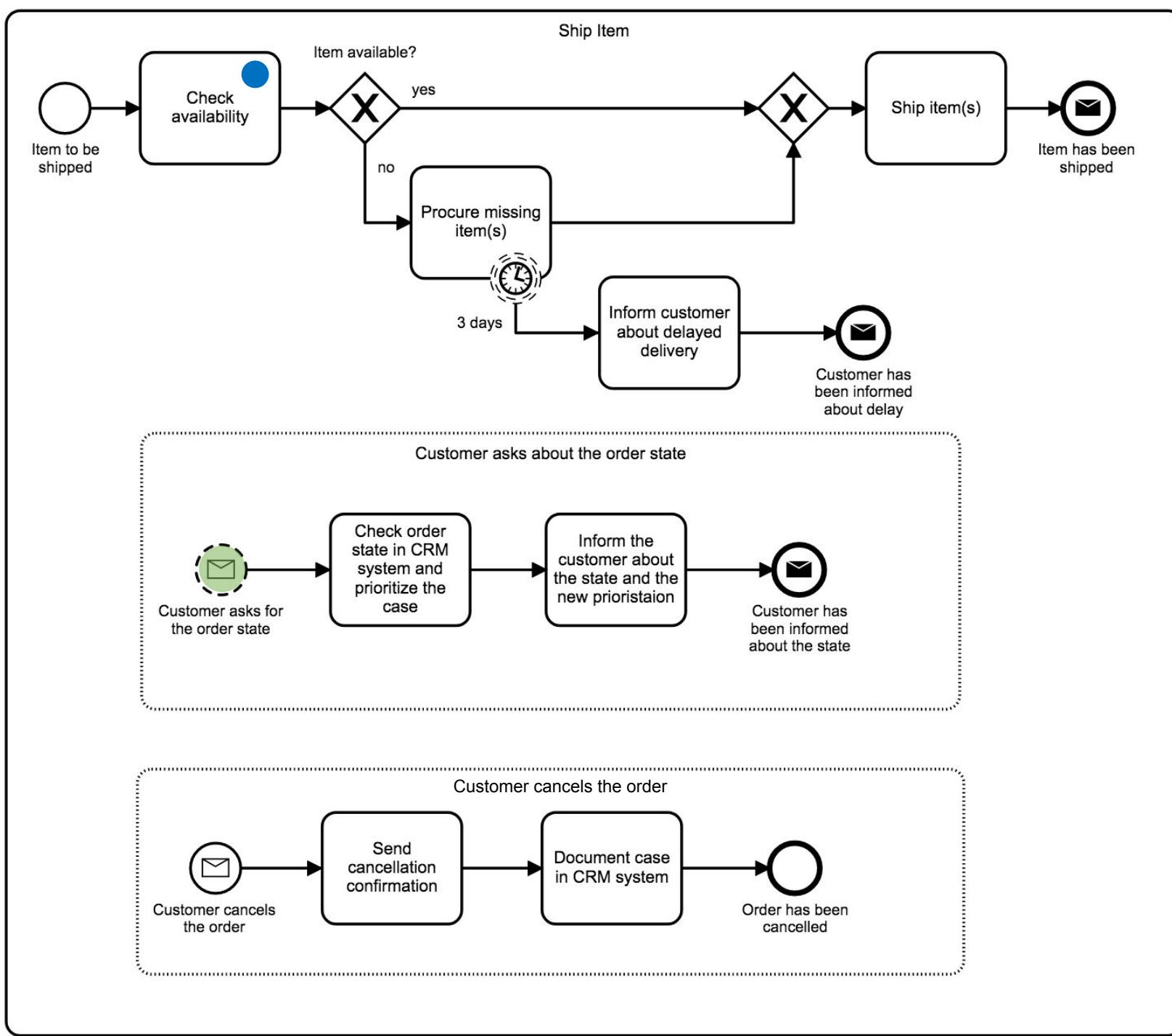


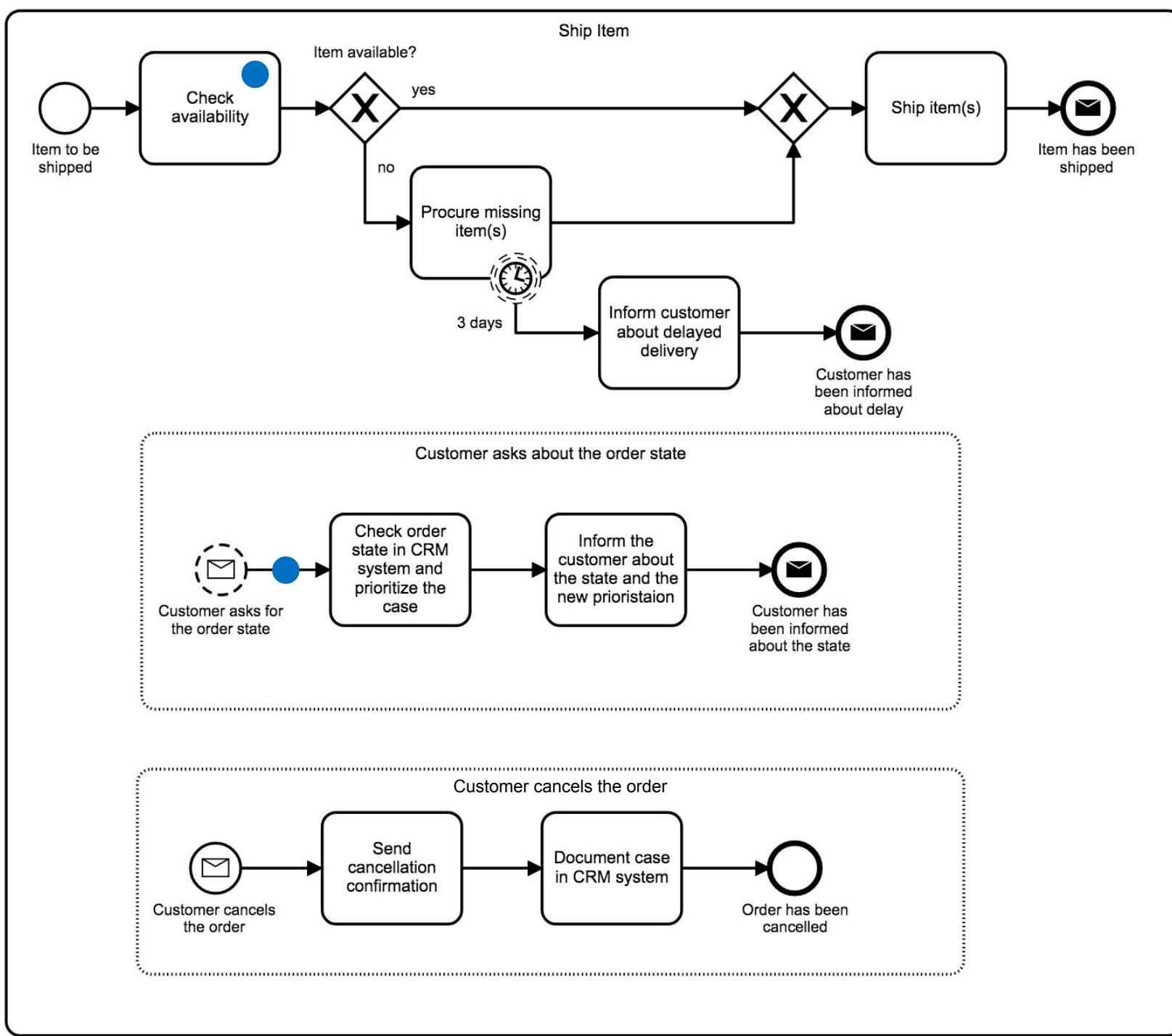


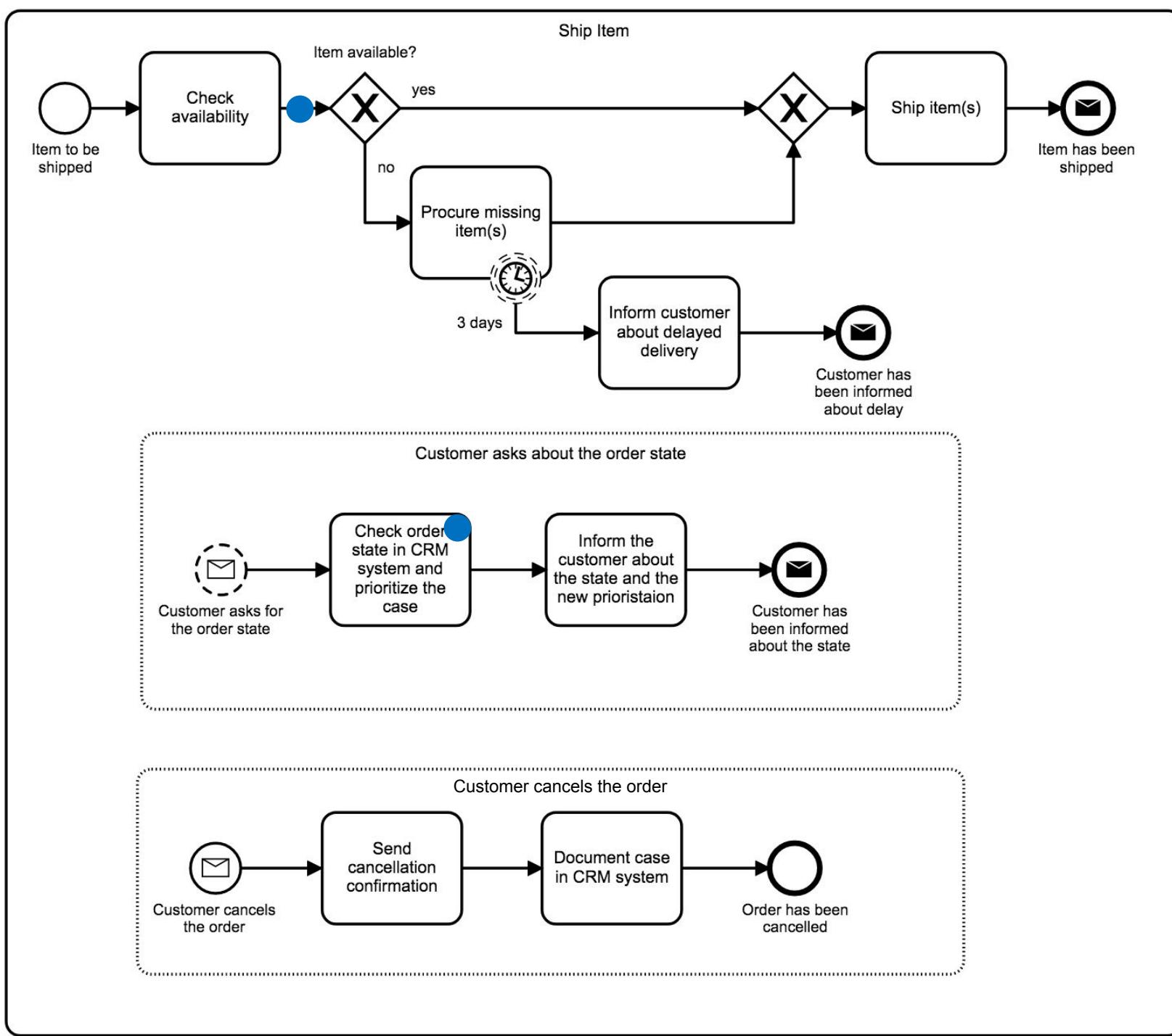
A customer wants to know about the state
of her order.

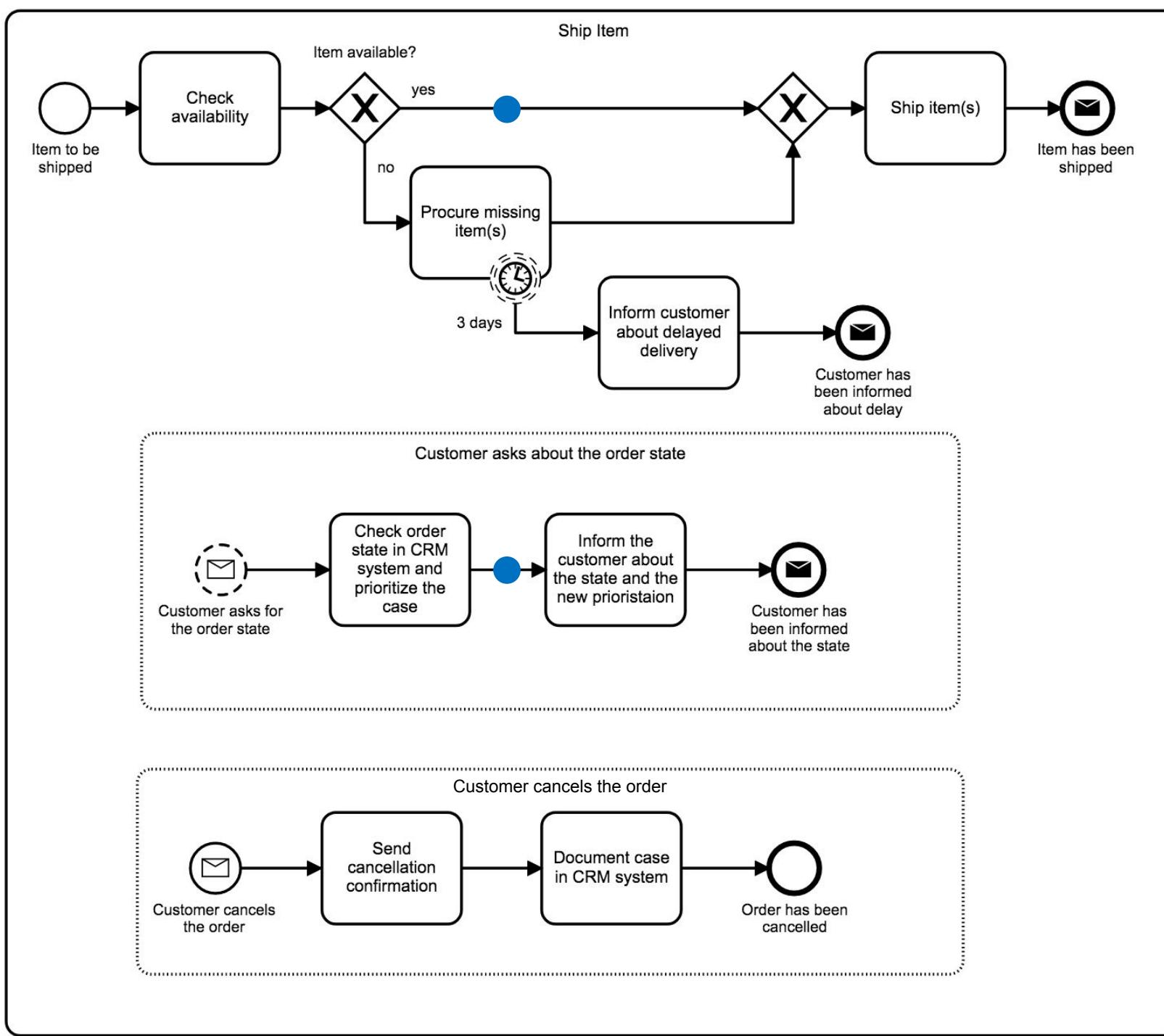


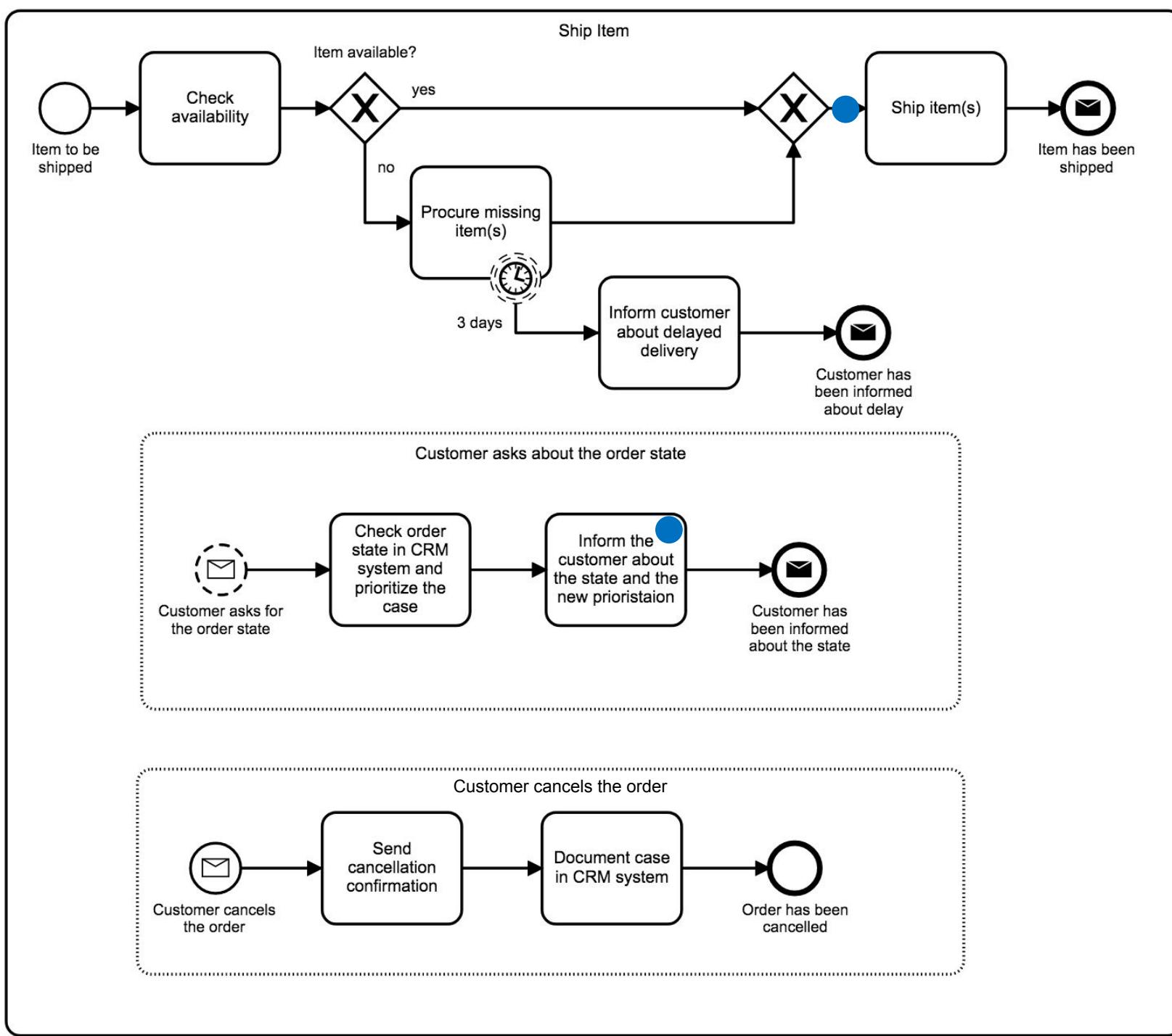


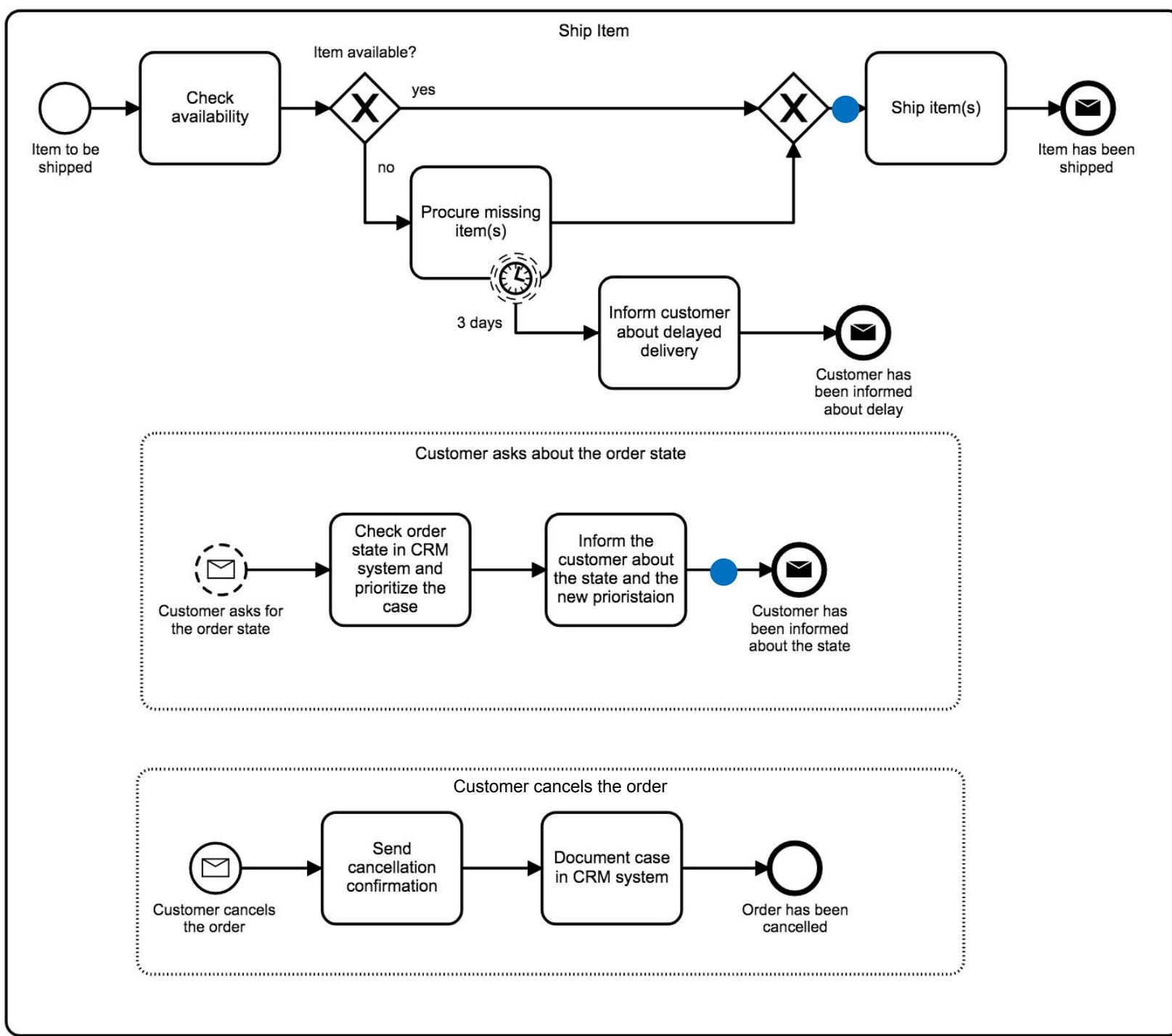


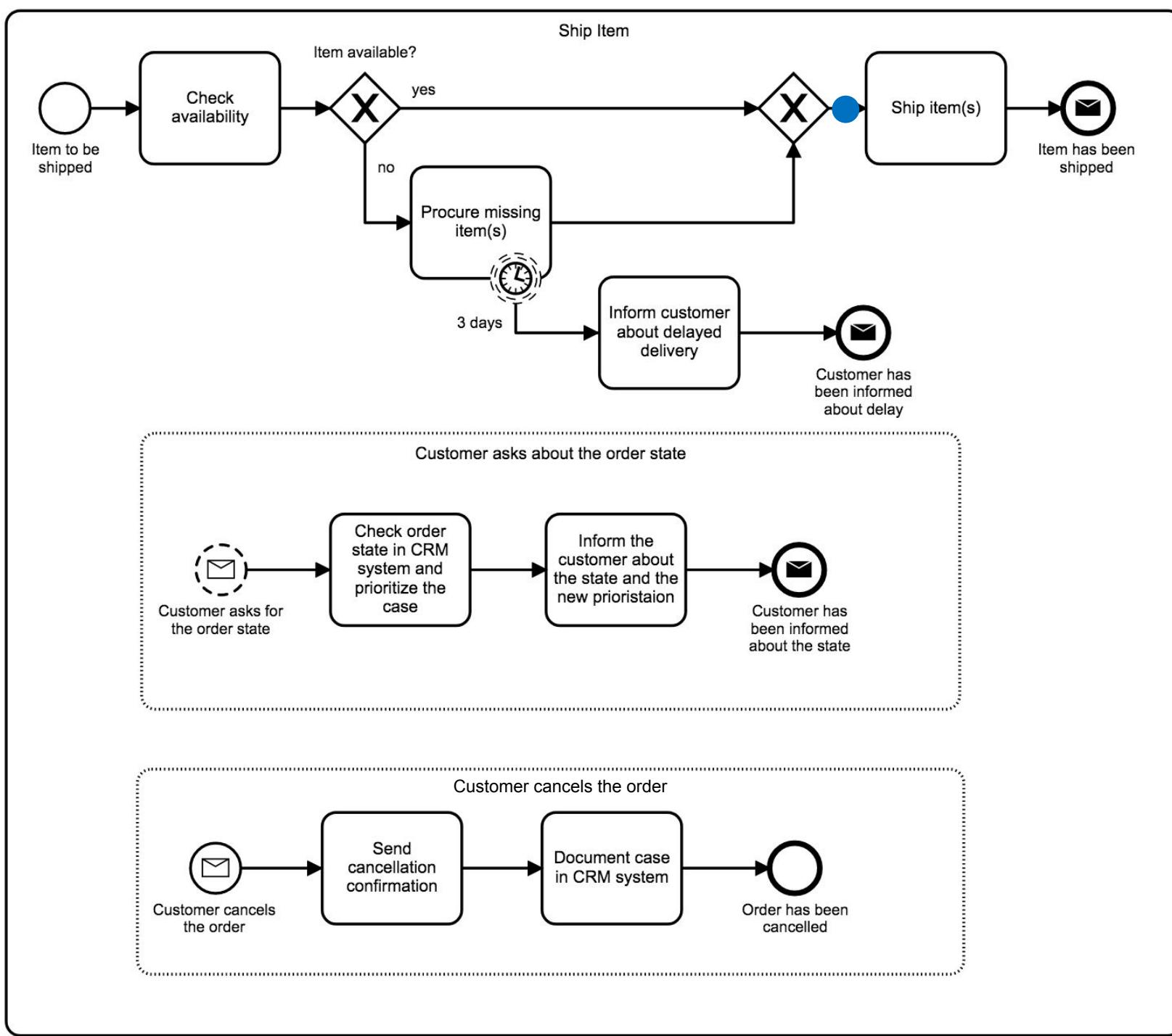






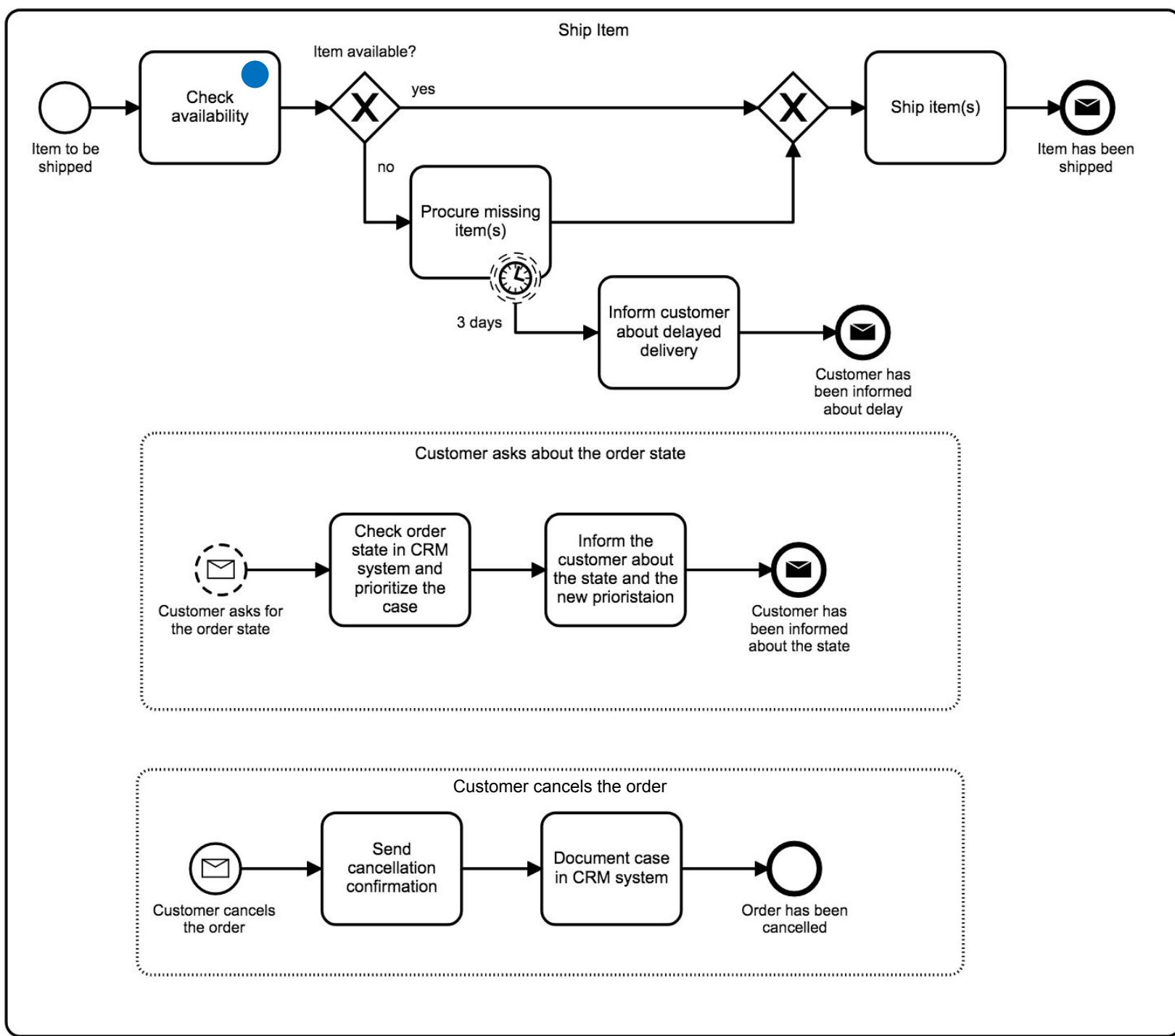


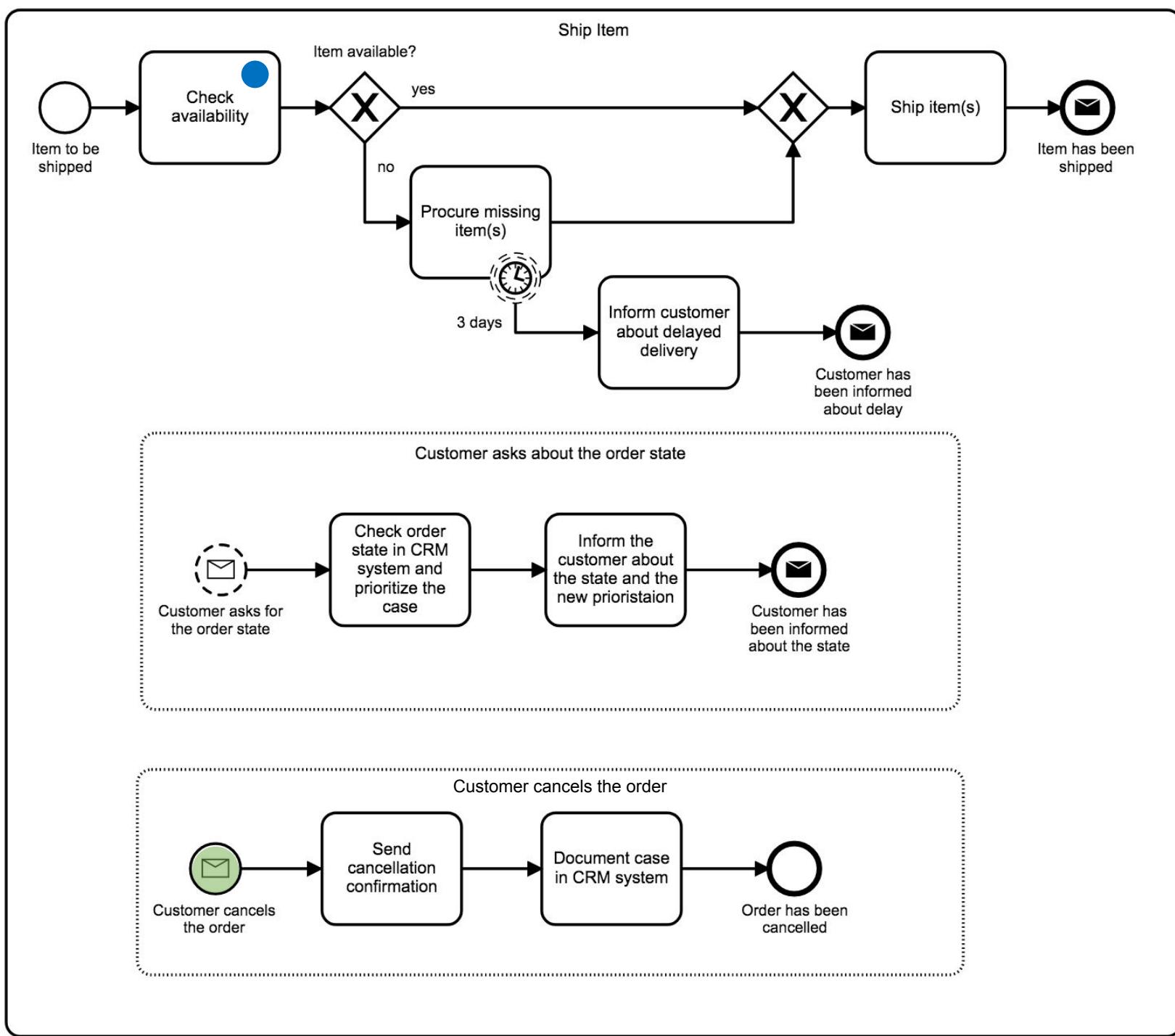


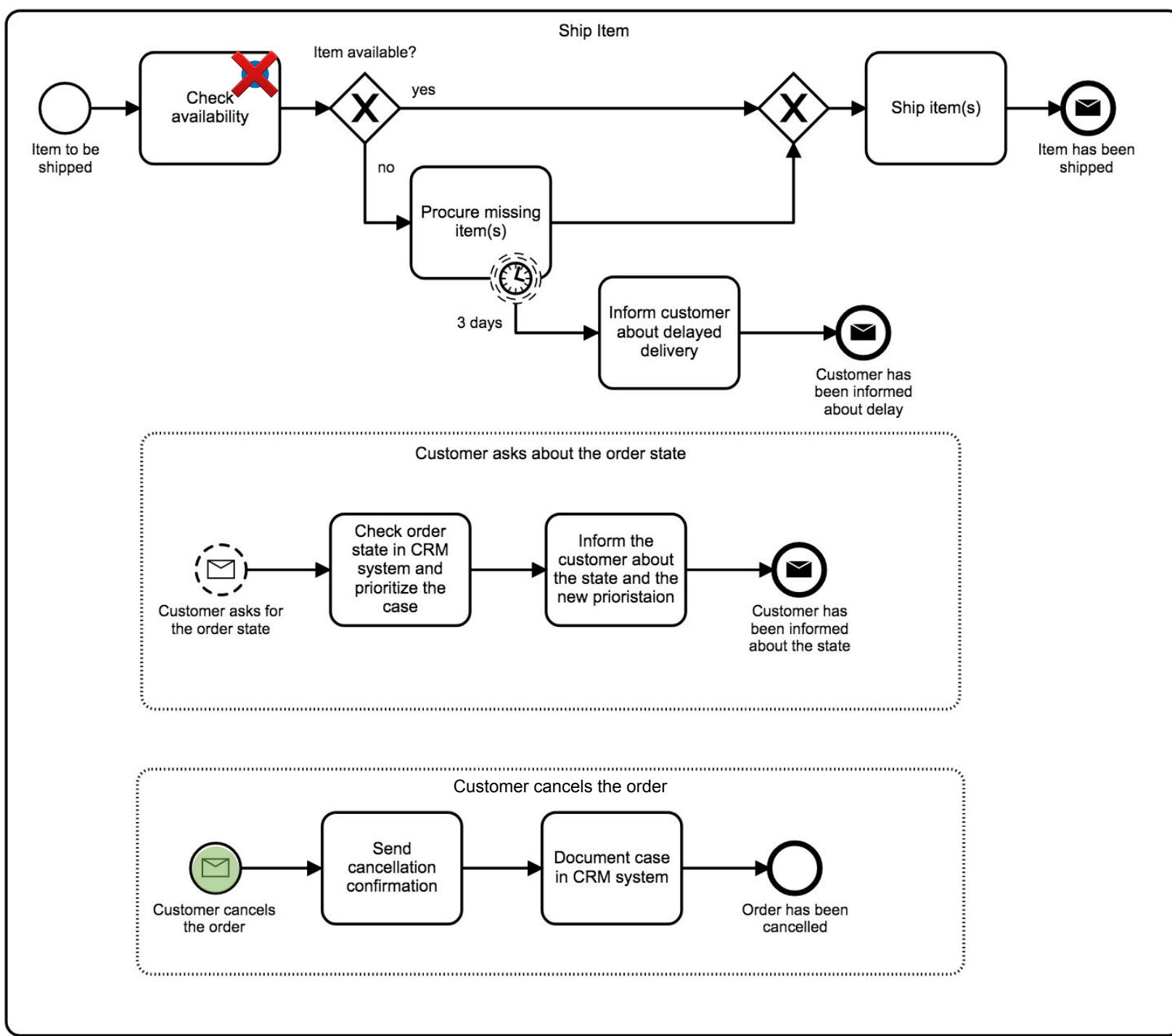


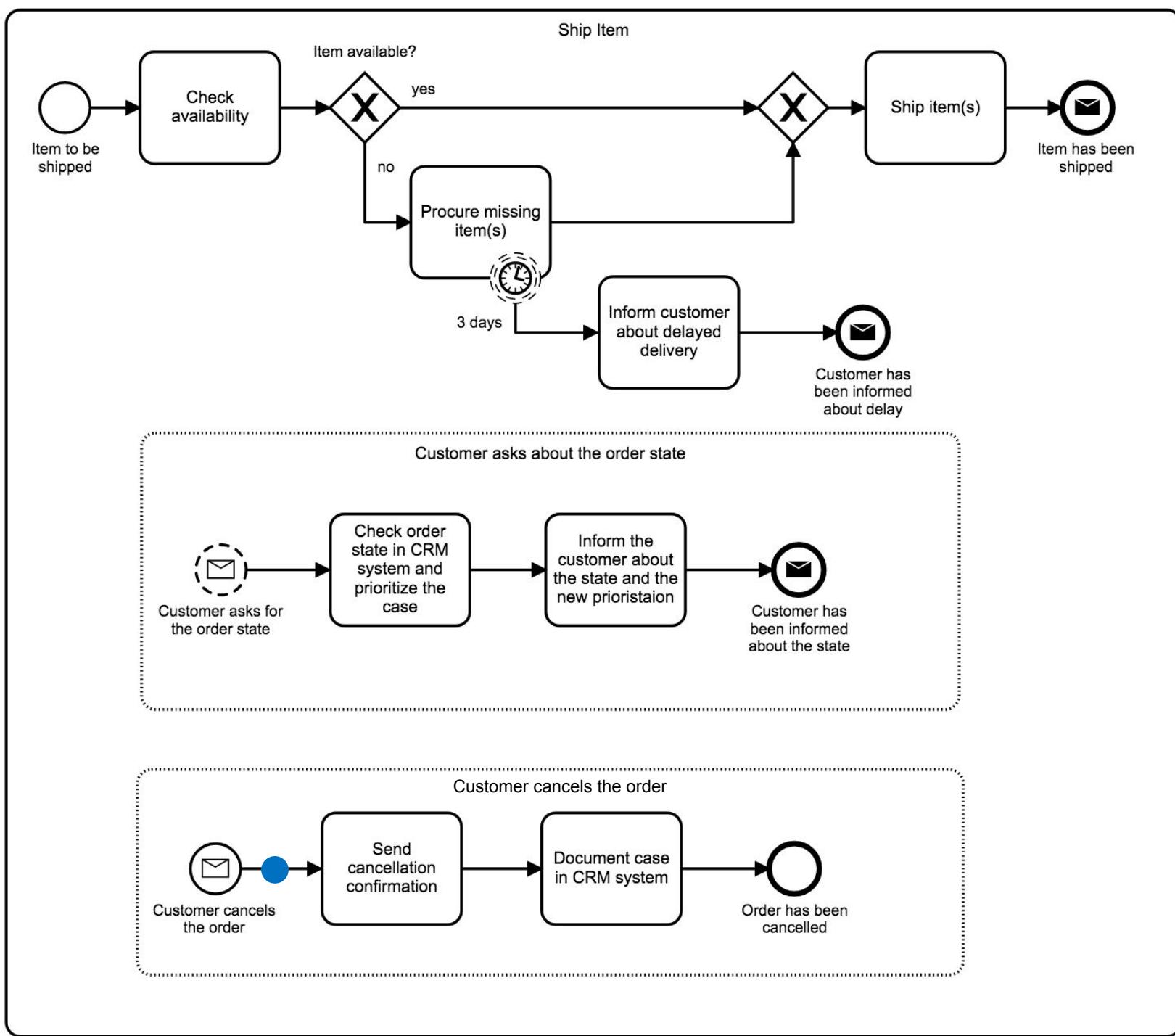


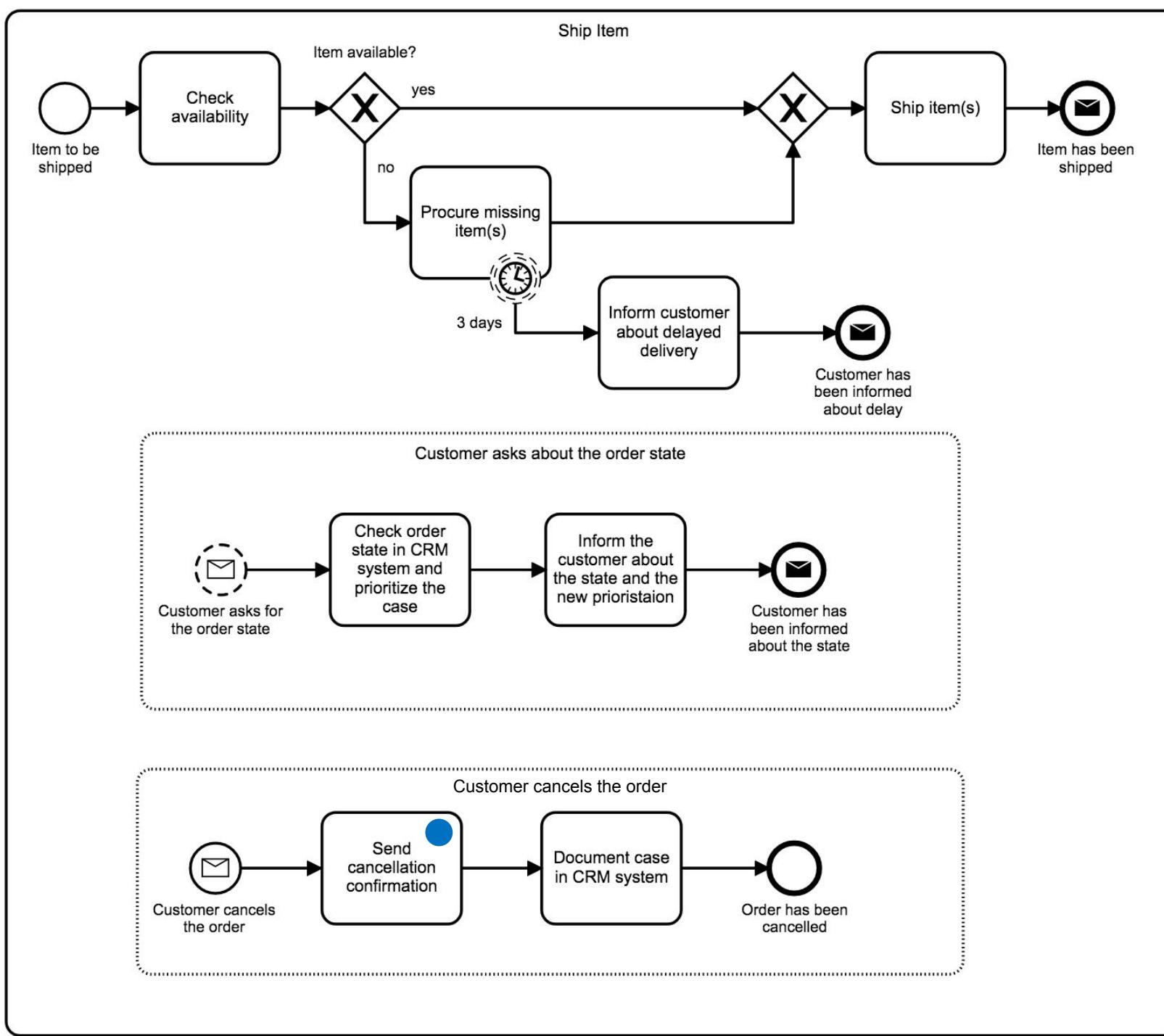
The customer cancels the order

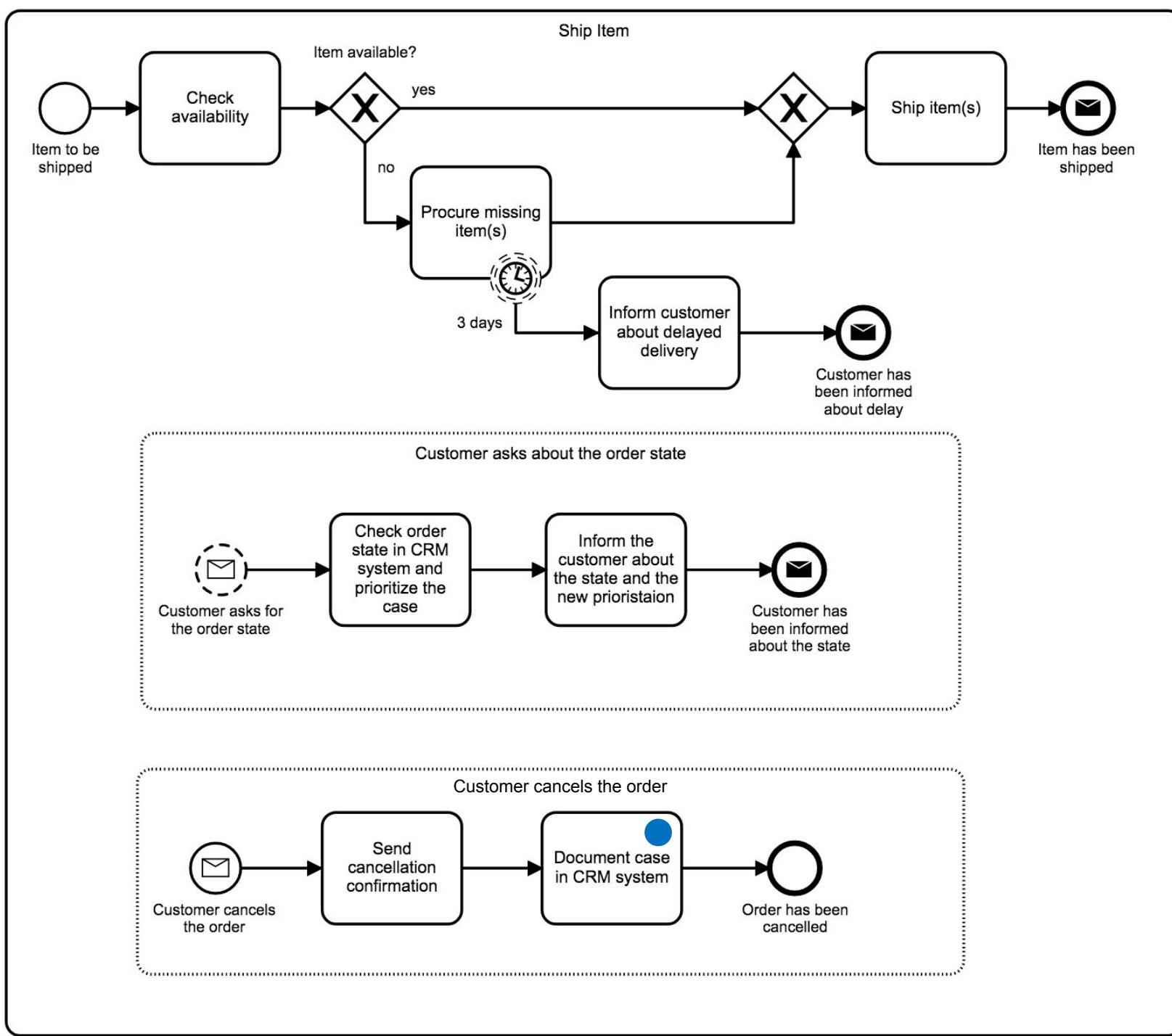


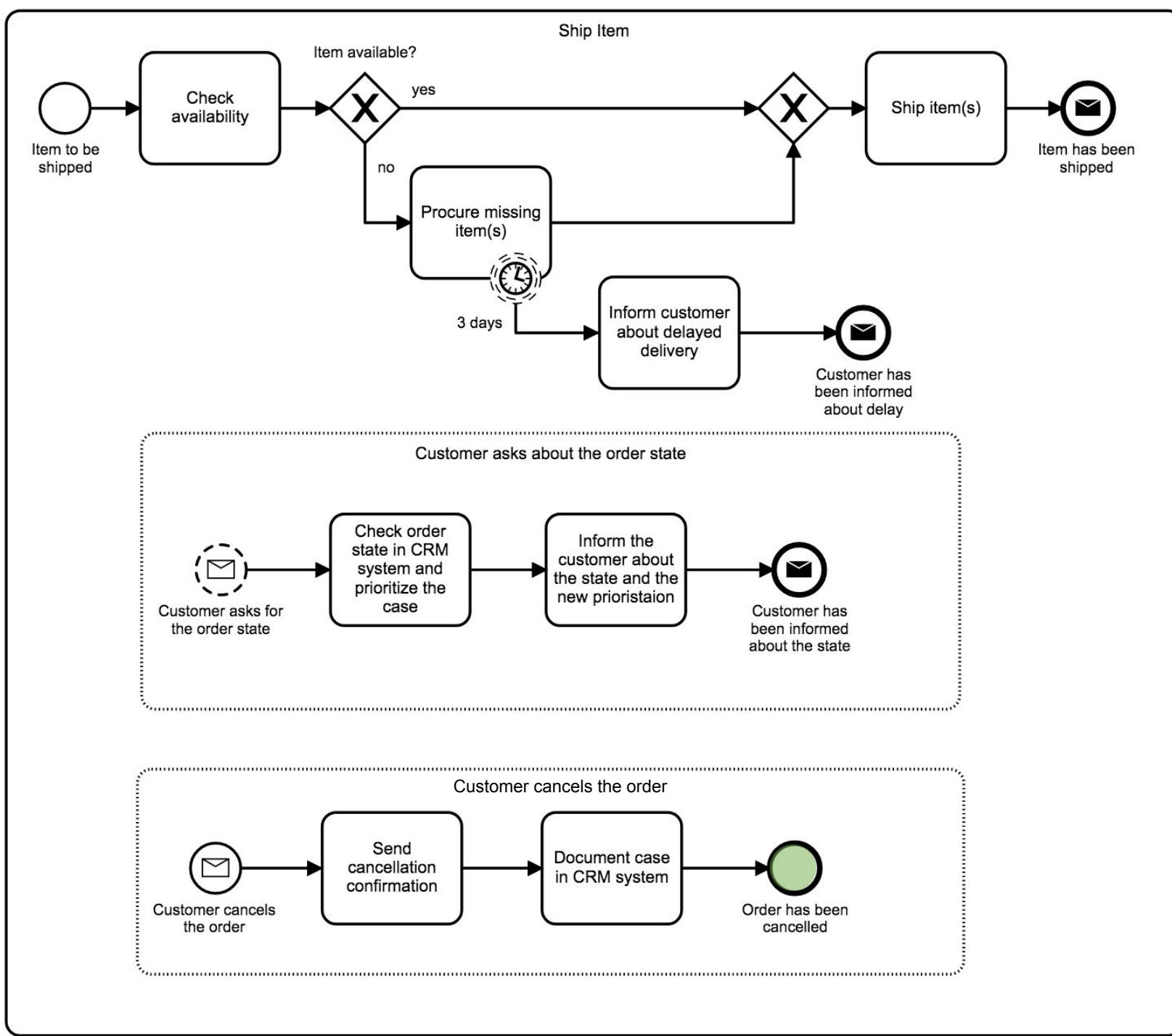


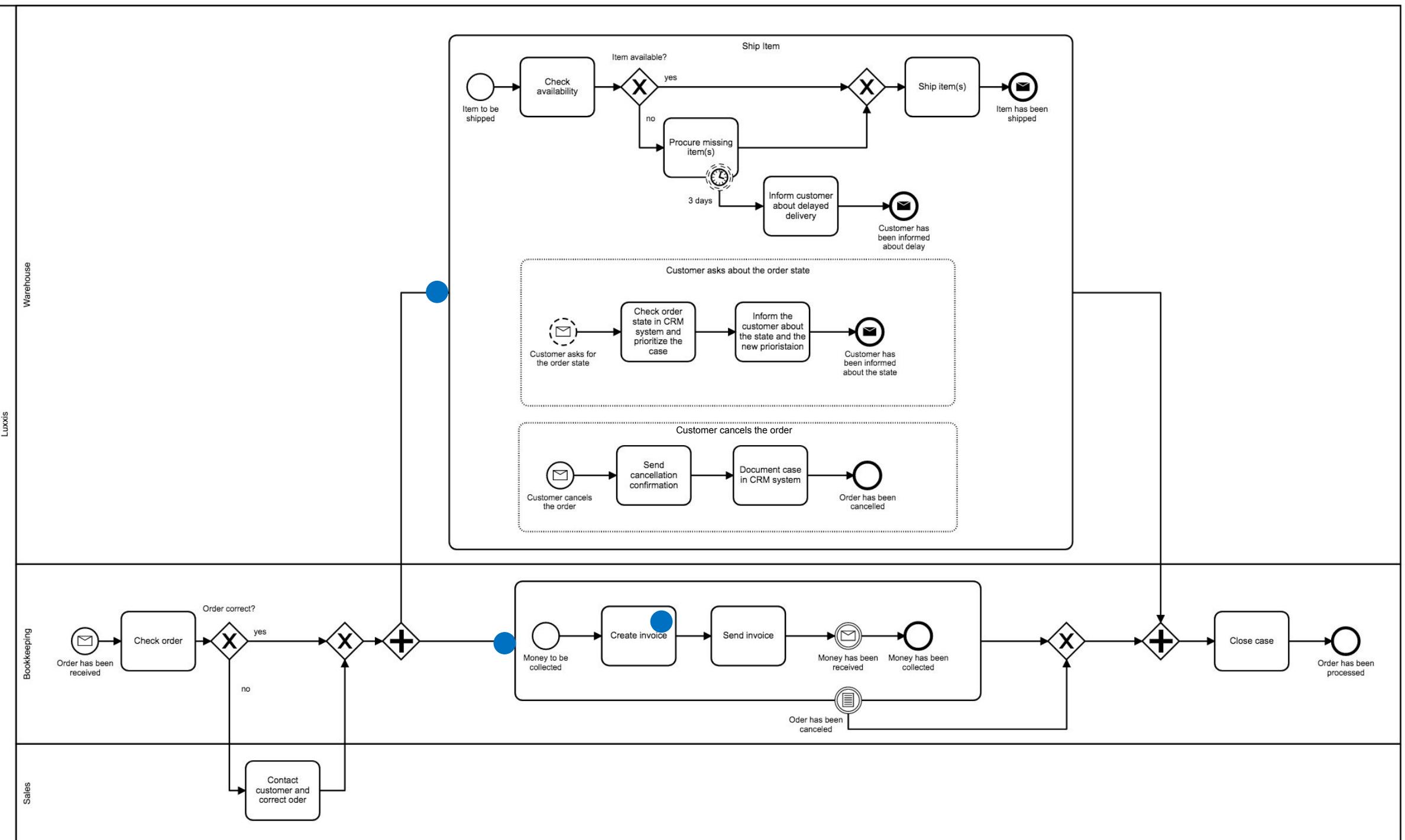


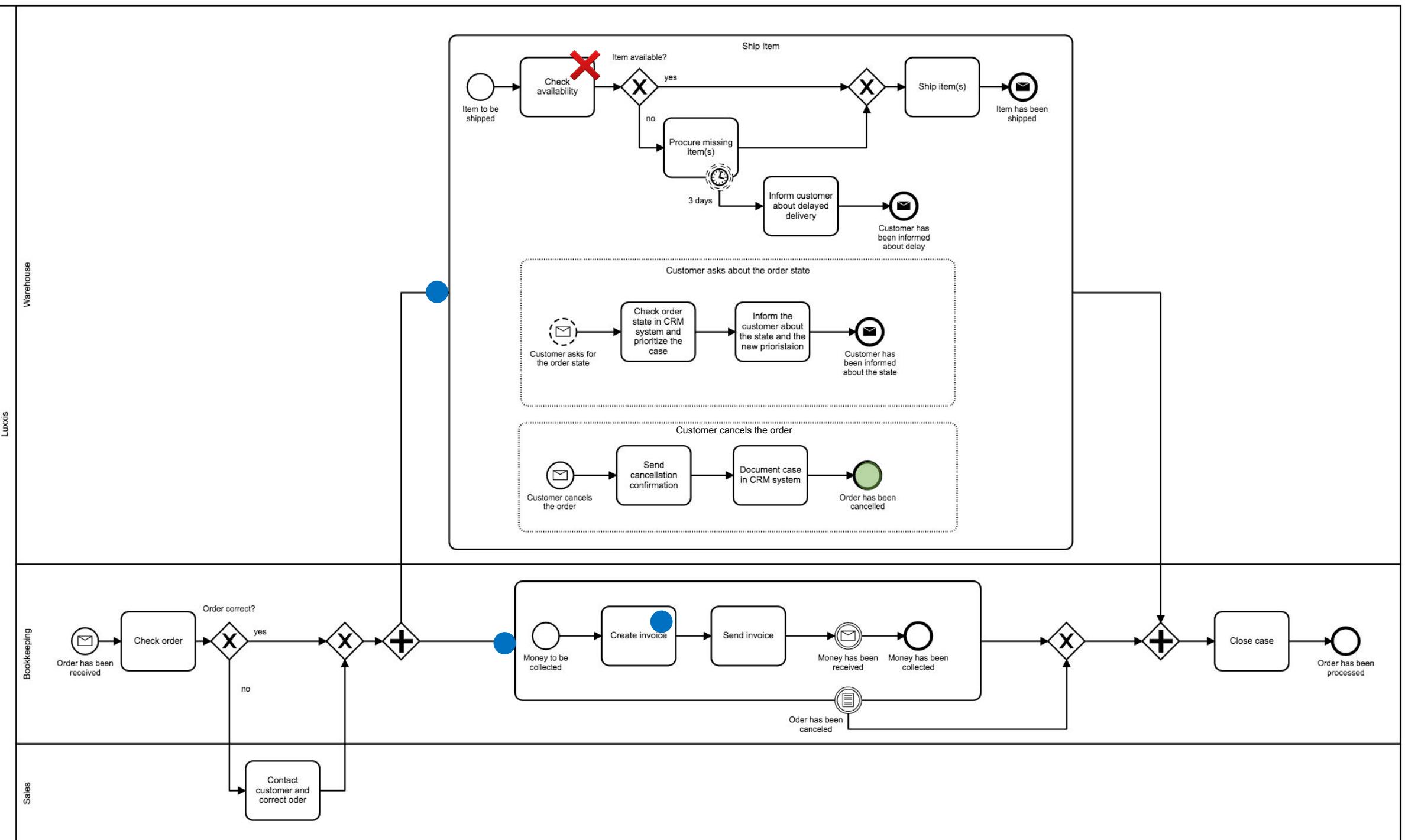


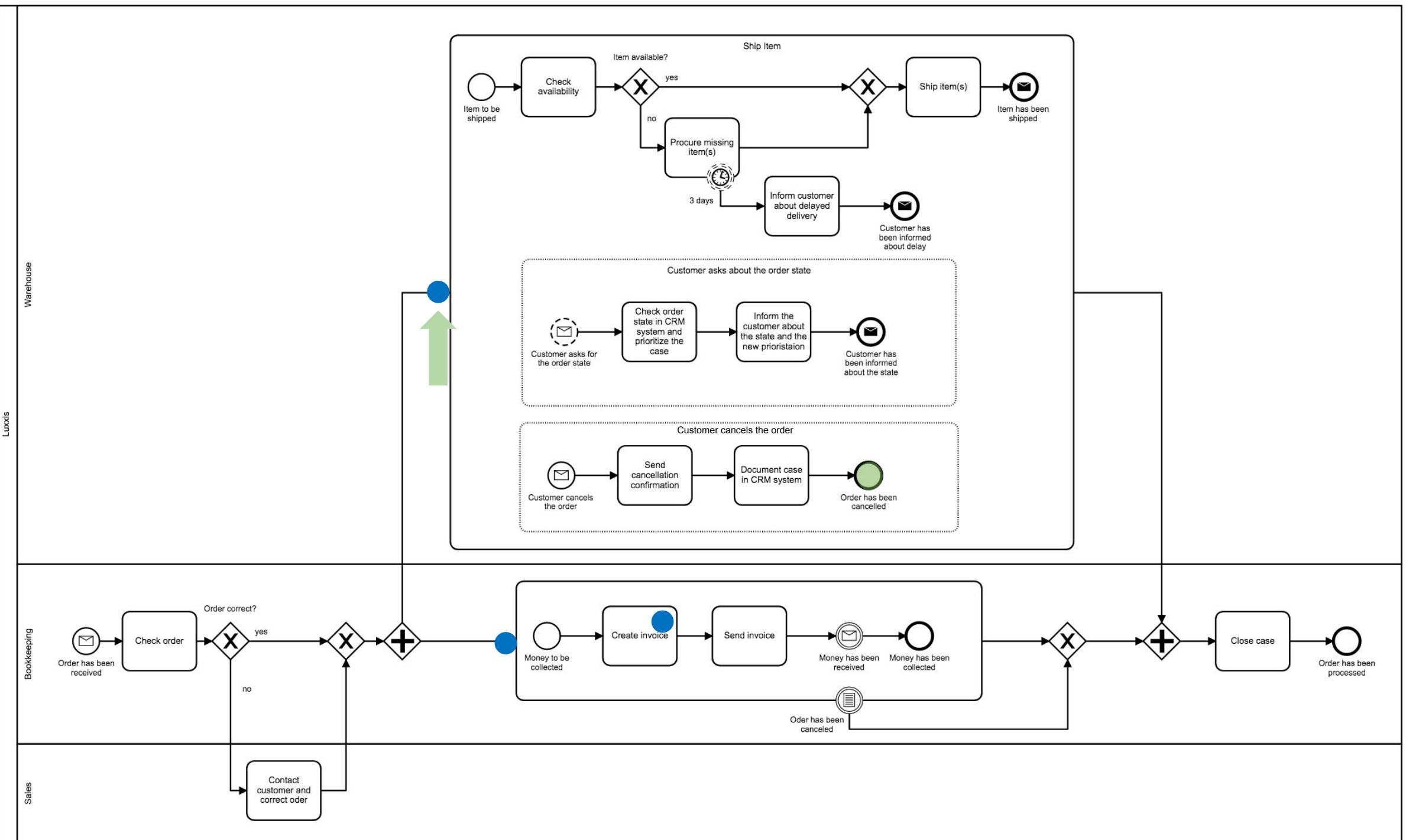


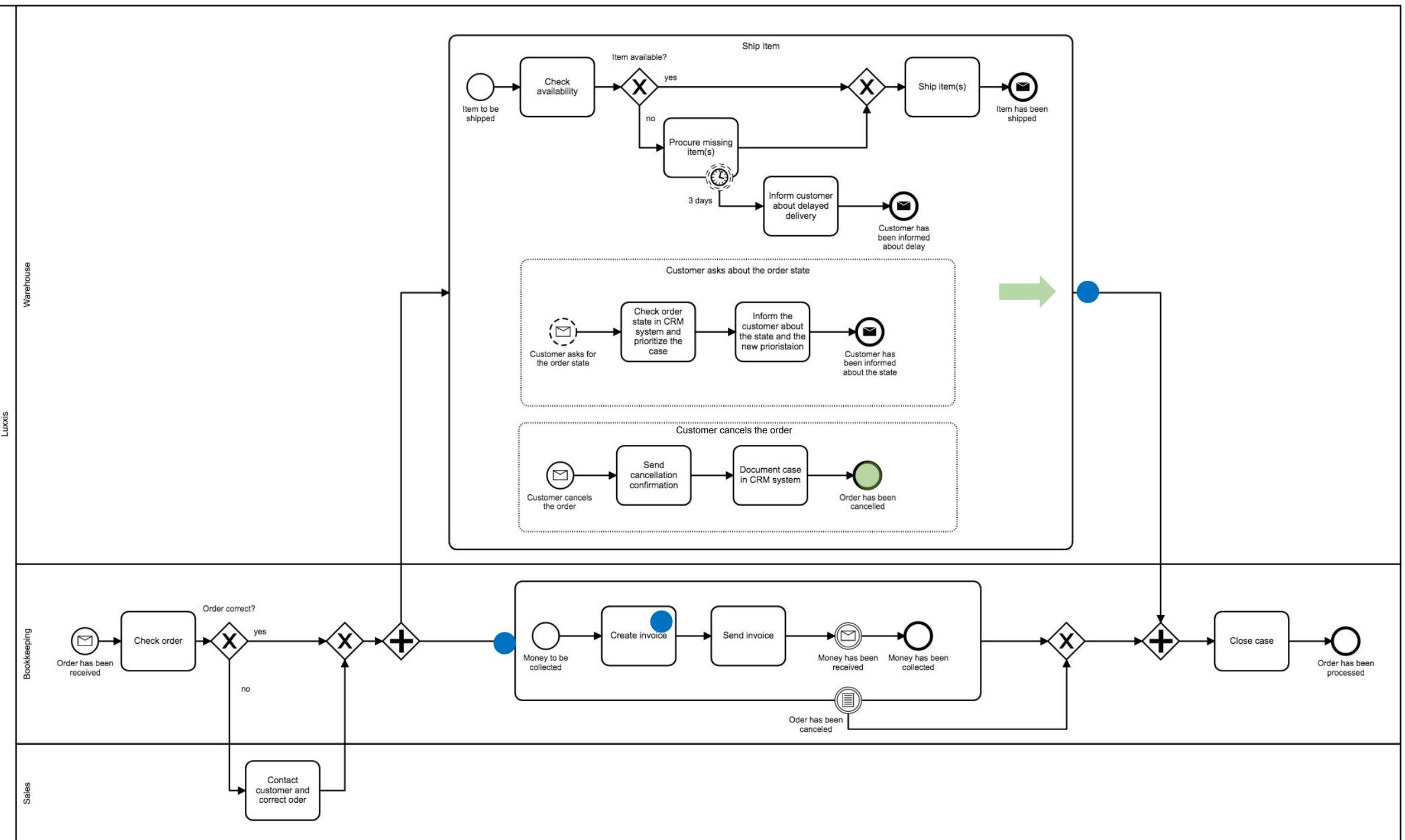


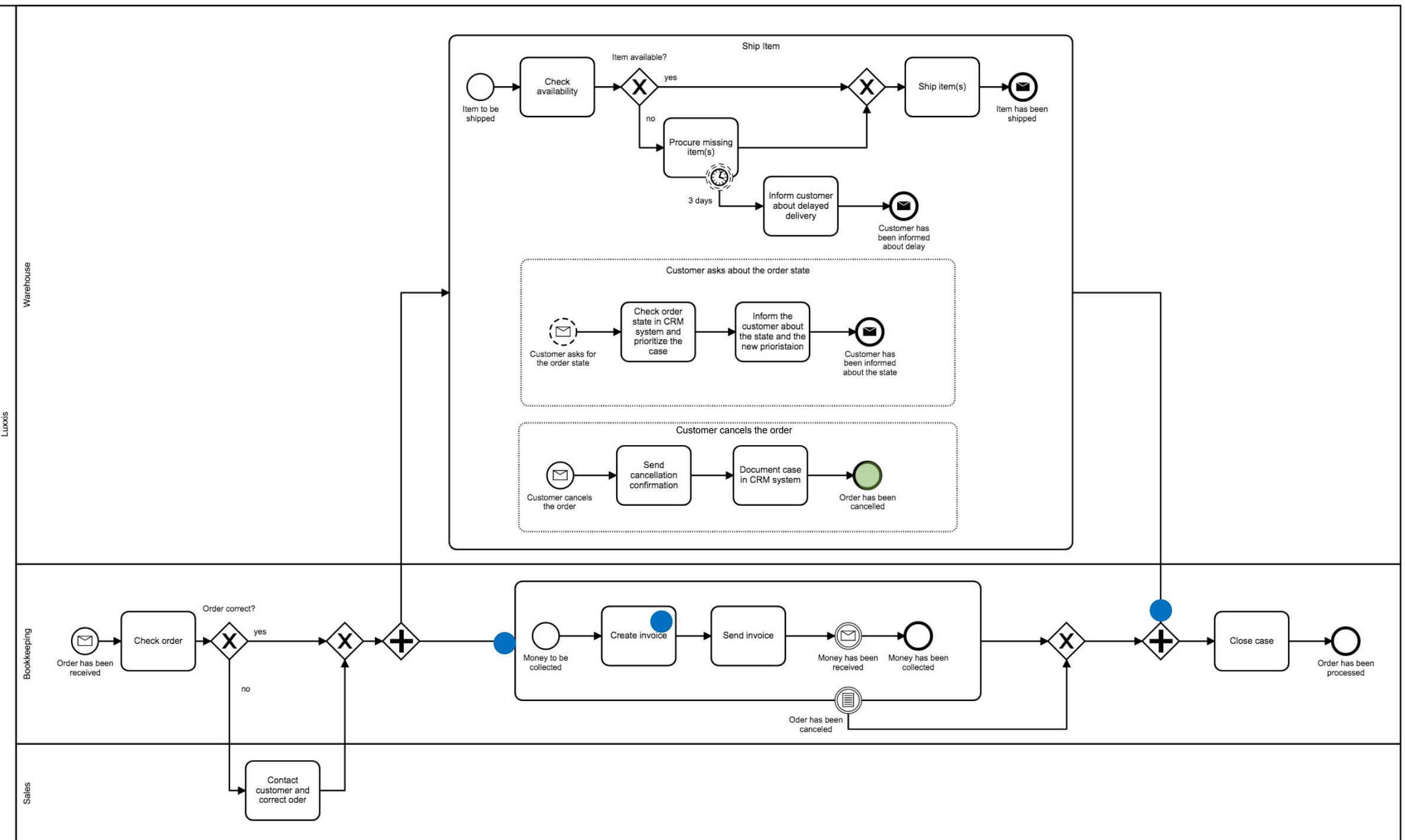


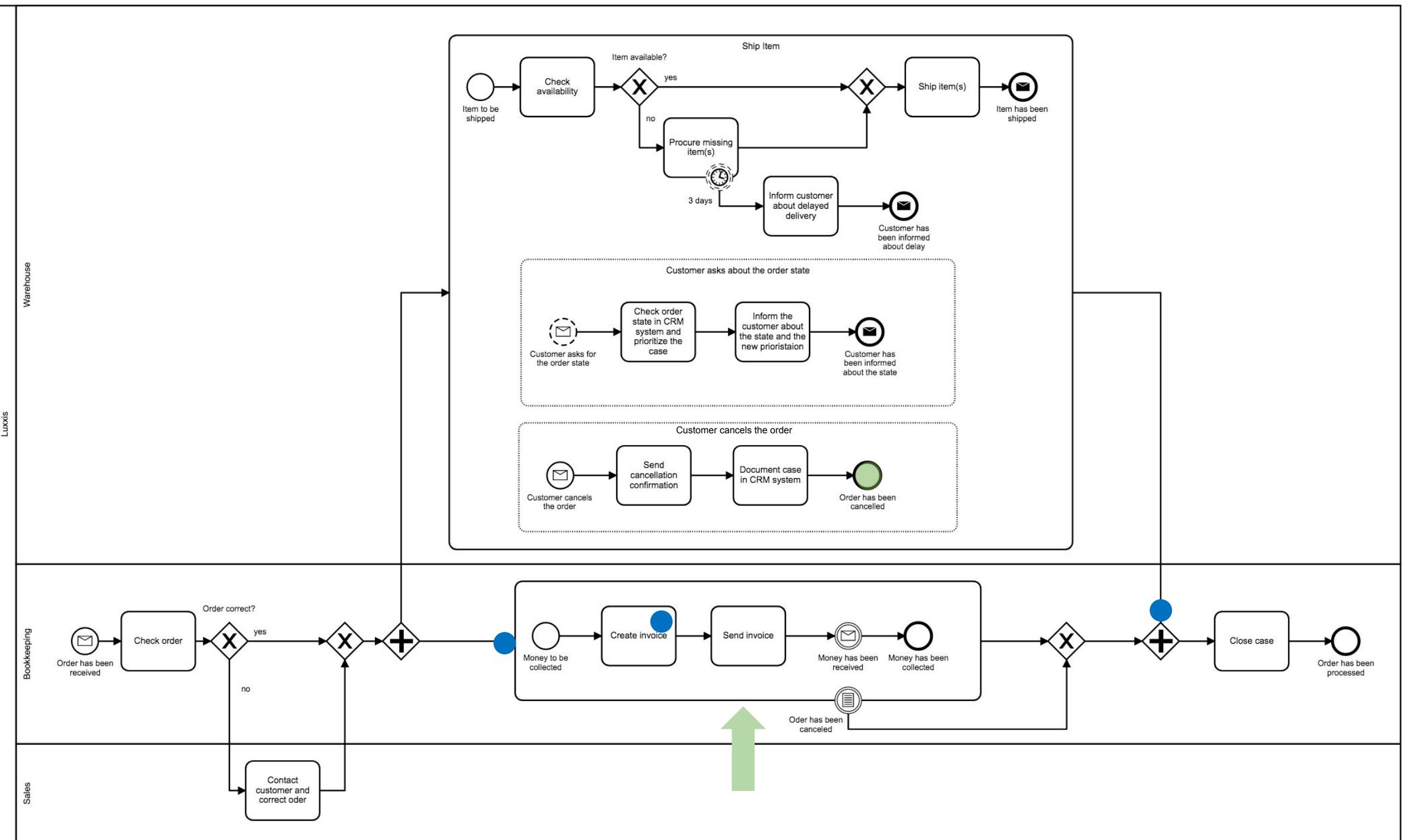


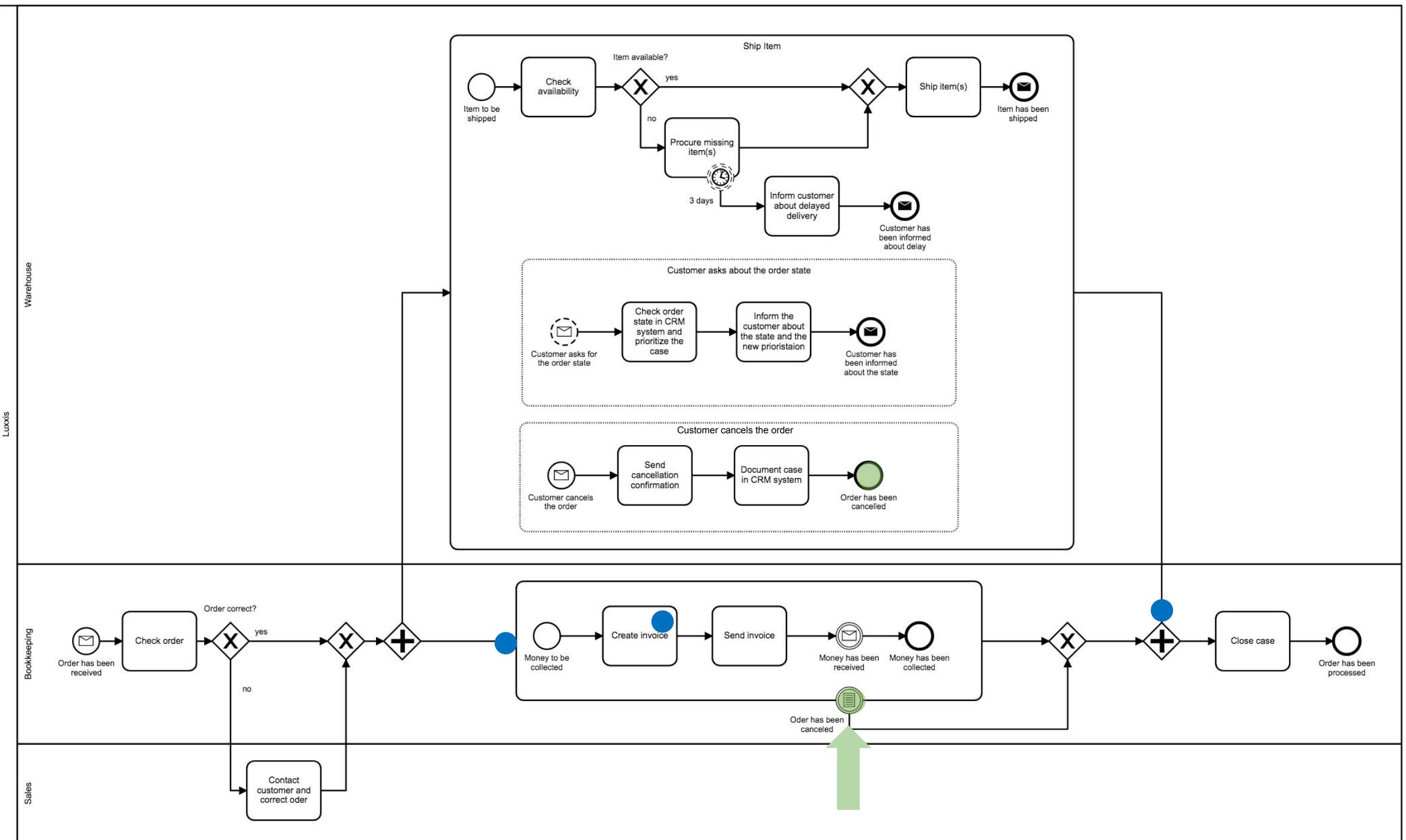


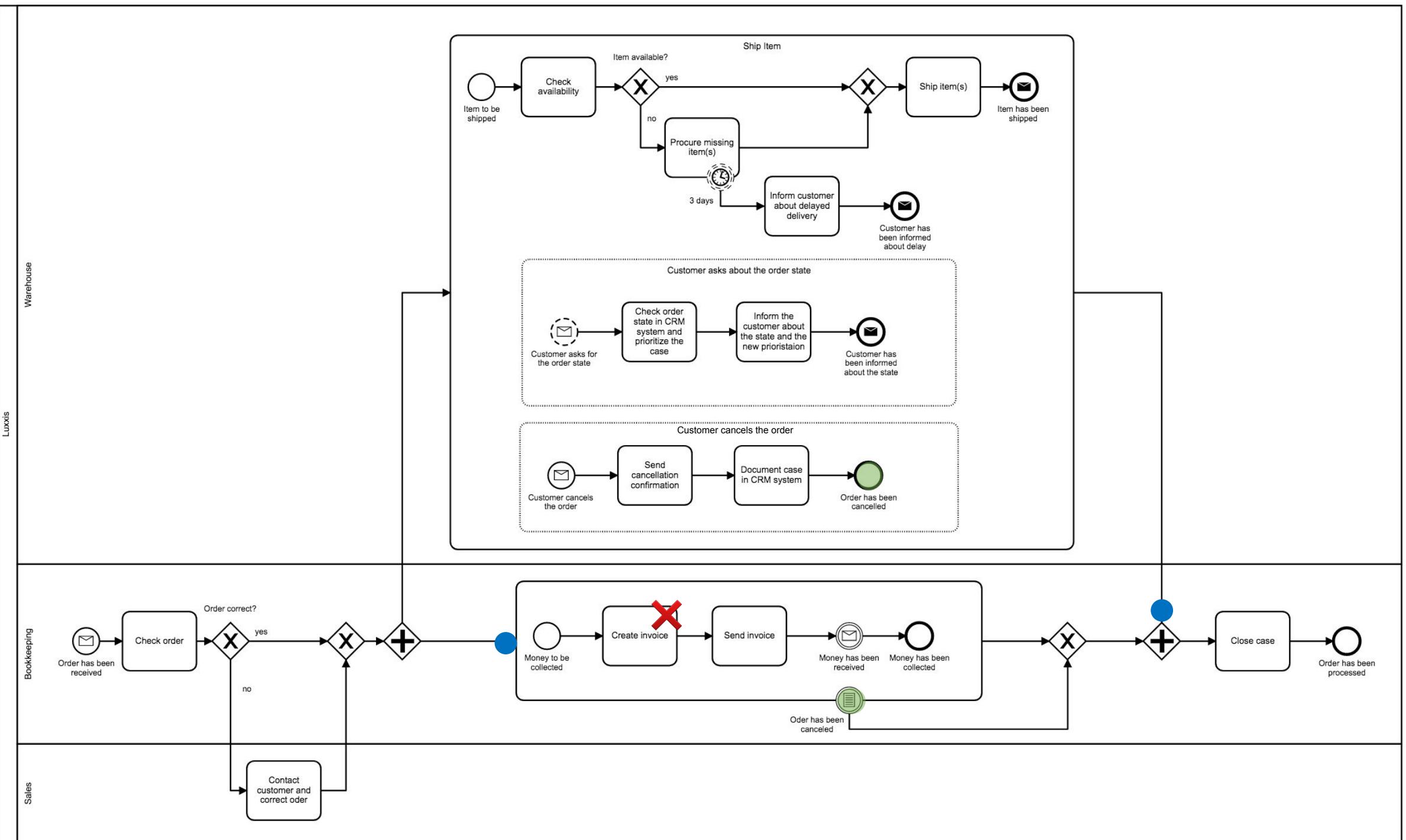


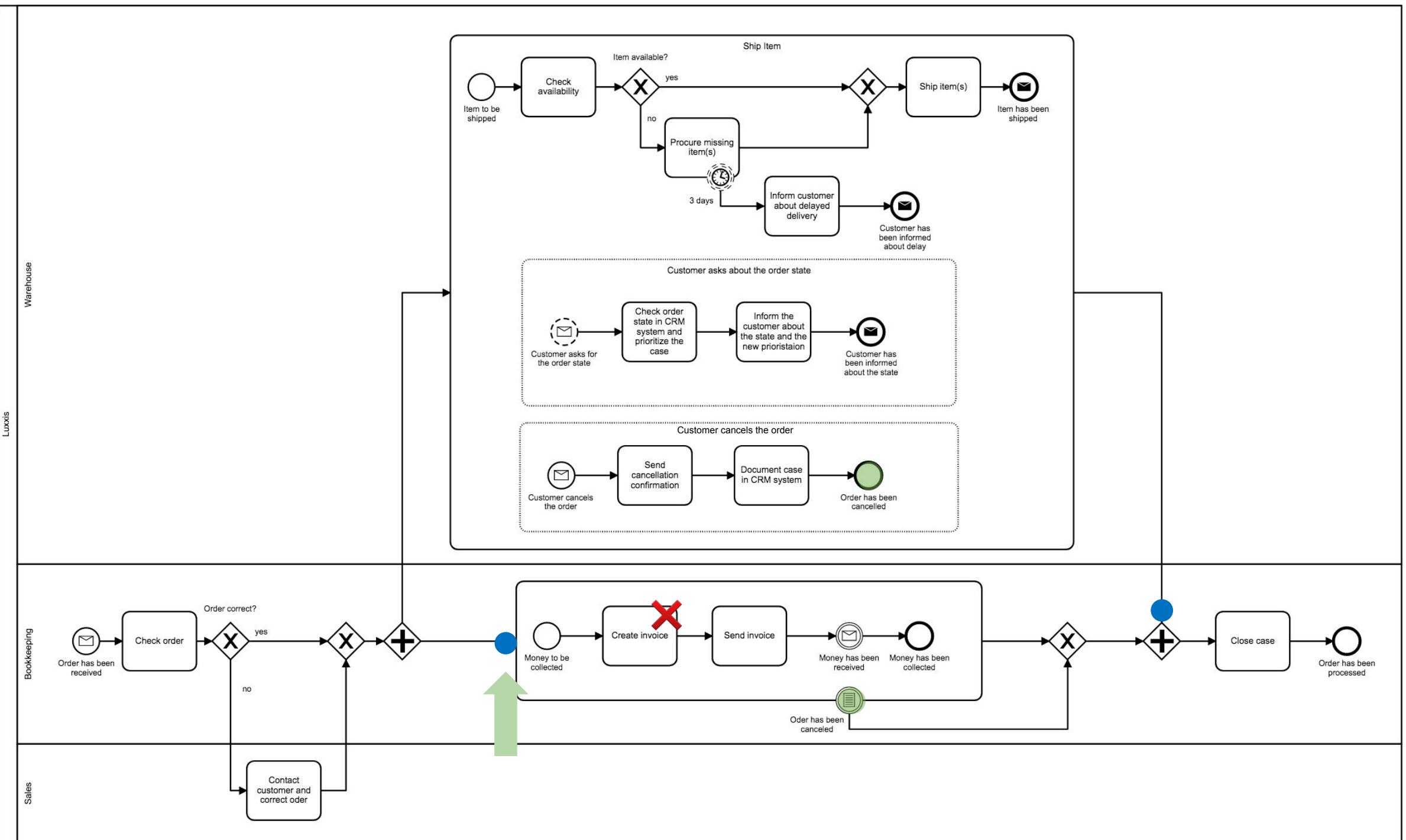


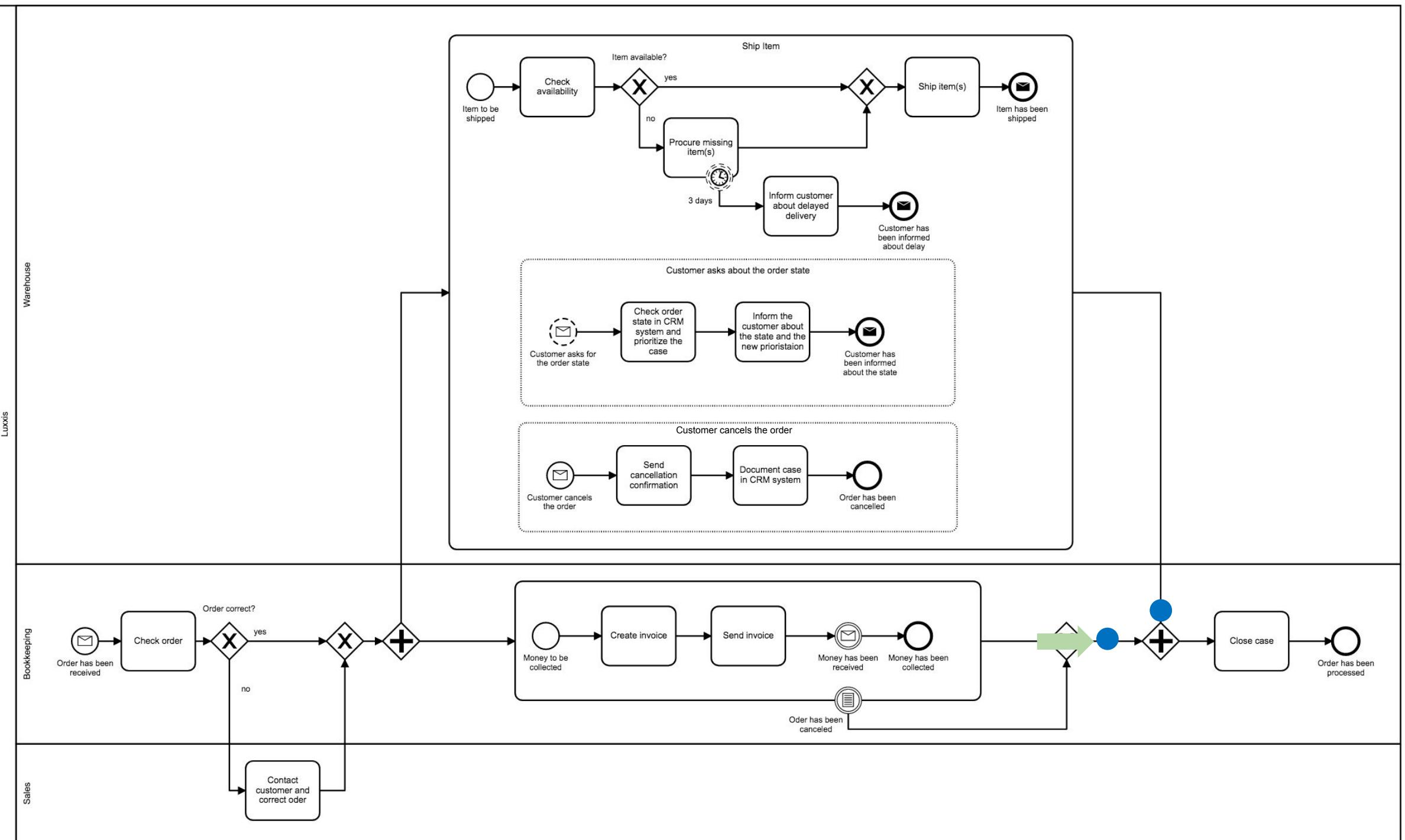


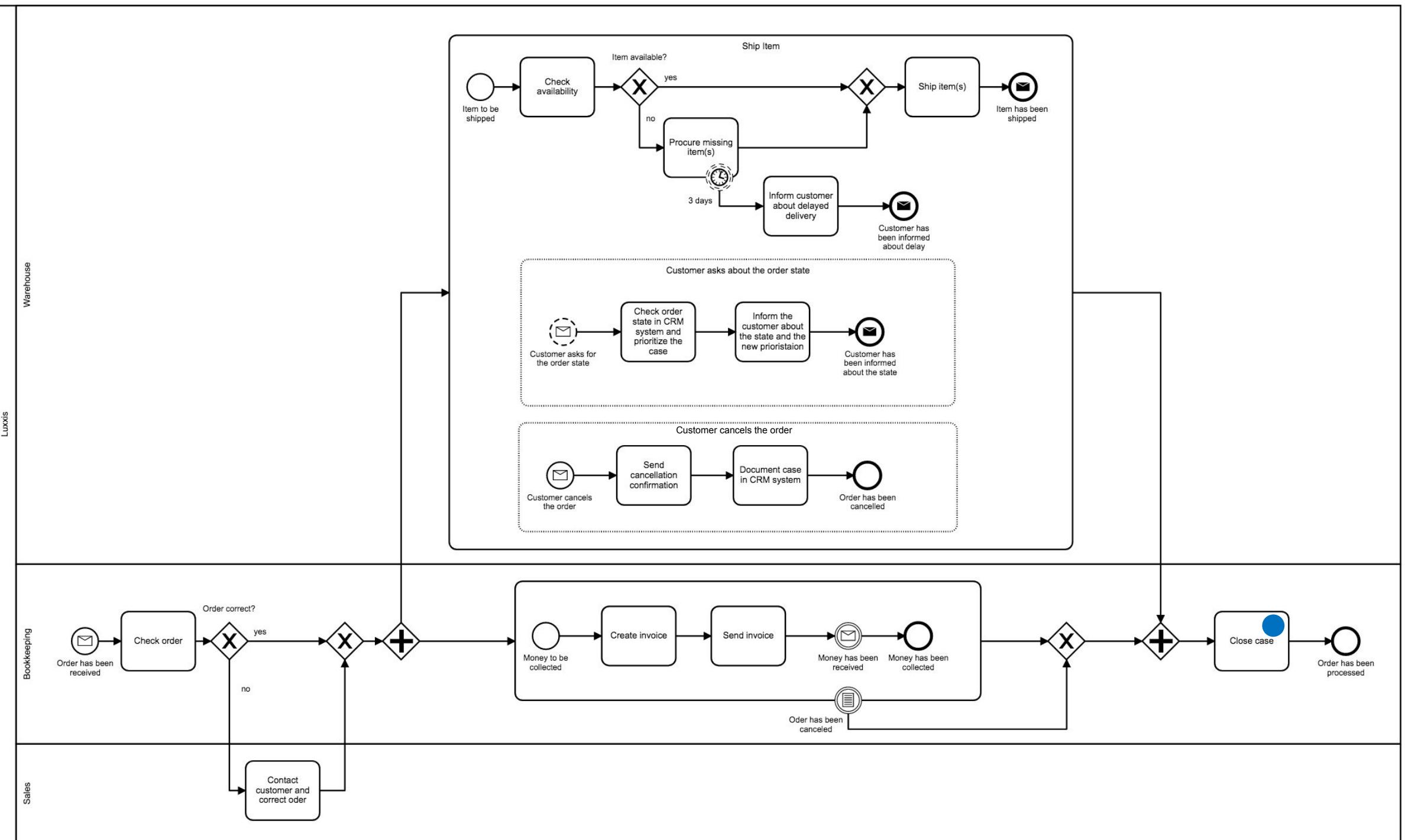




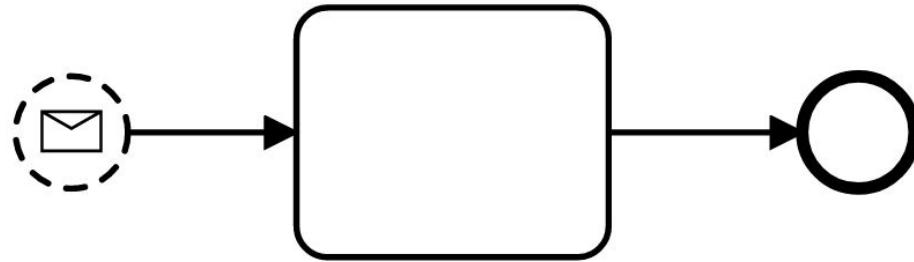




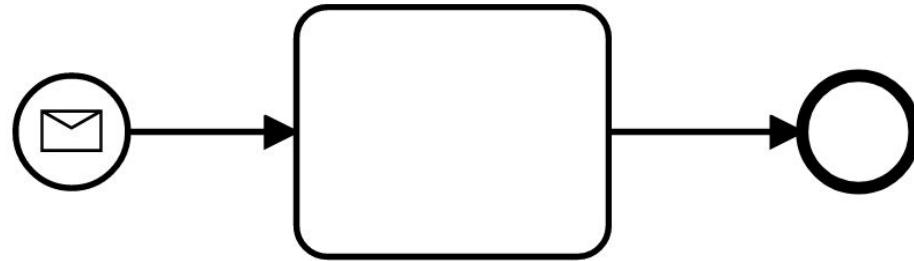




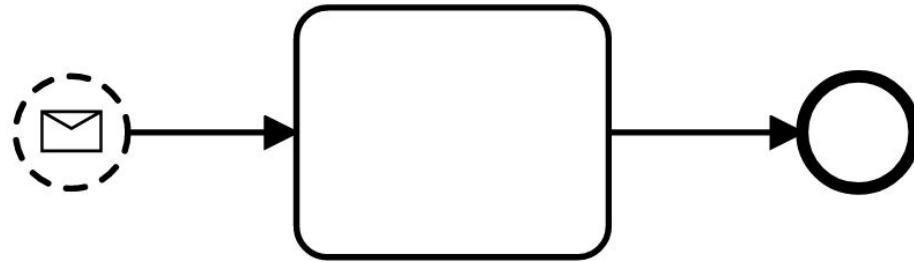
Event Sub Process (Non Cancelling)



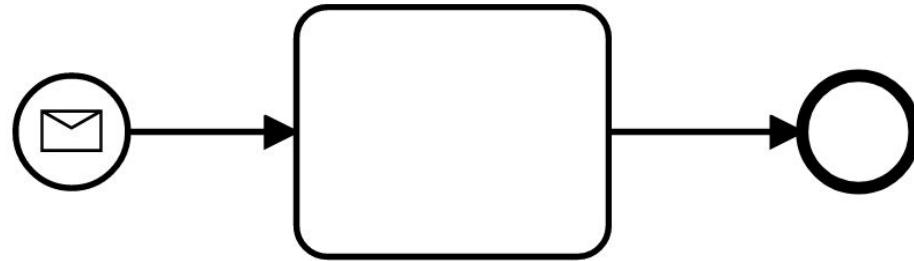
Event Sub Process (Cancelling)



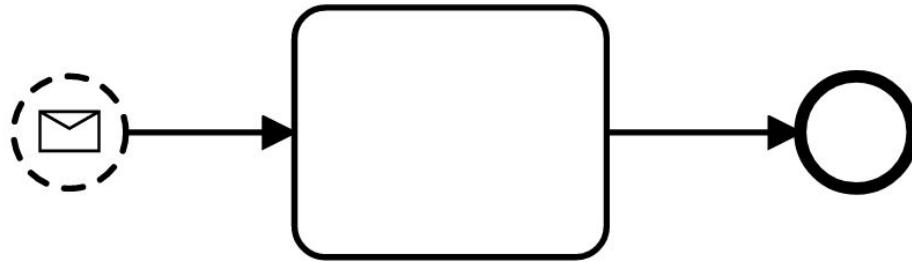
Event Sub Process (Non Cancelling)



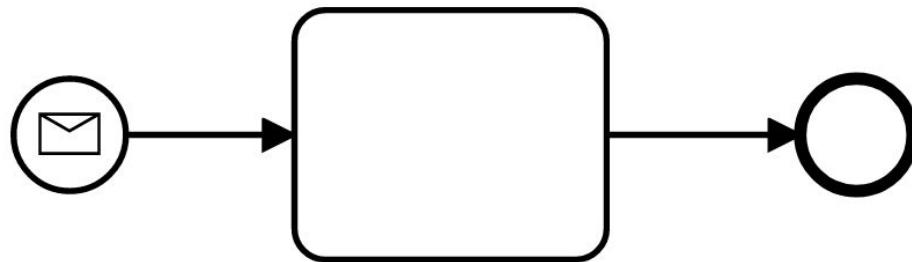
Event Sub Process (Cancelling)



Event Sub Process (Non Cancelling)



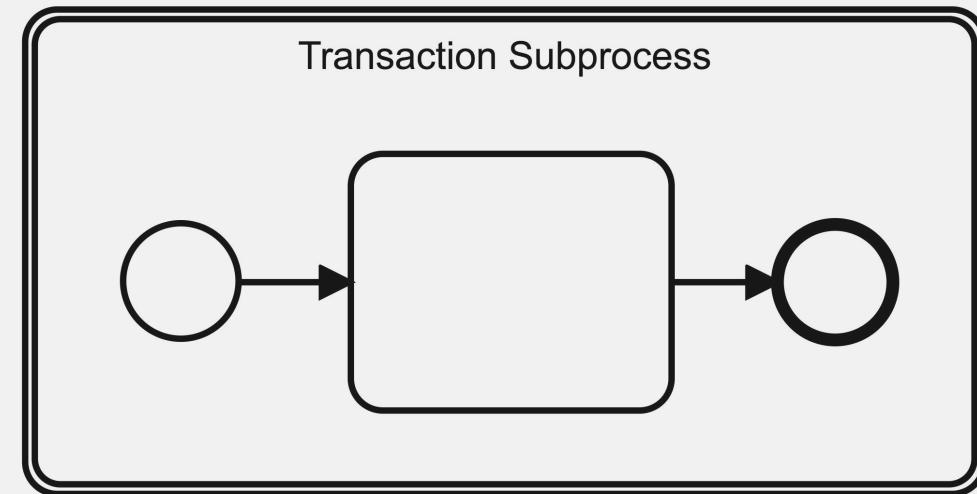
Event Sub Process (Cancelling)



BPMN Theory

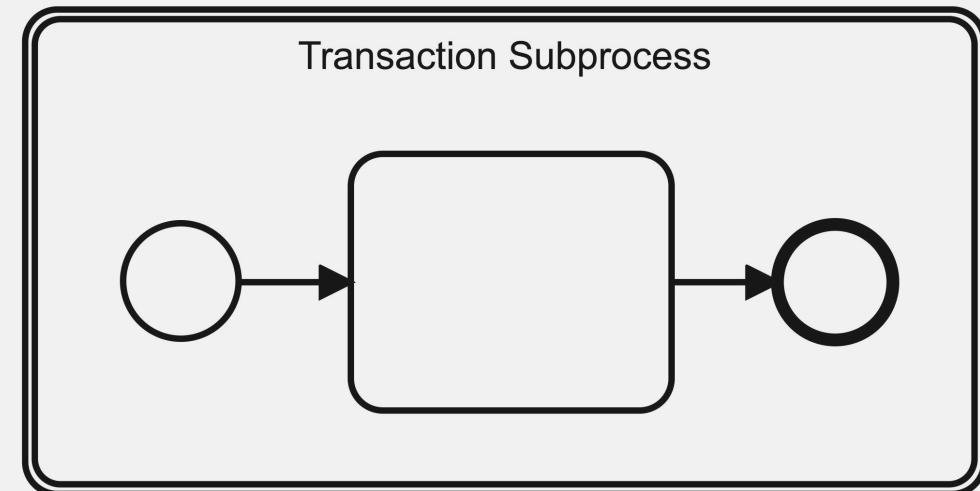
process camp

Transaction Subprocess

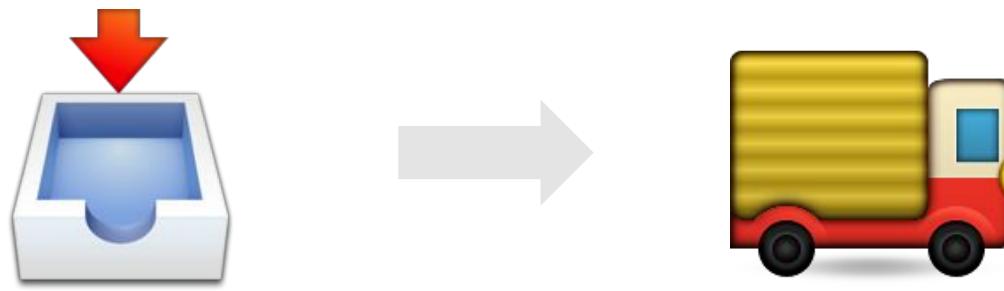


- Is used to group multiple activities to a transaction
- A logical unit of work, which either succeeds or fails collectively

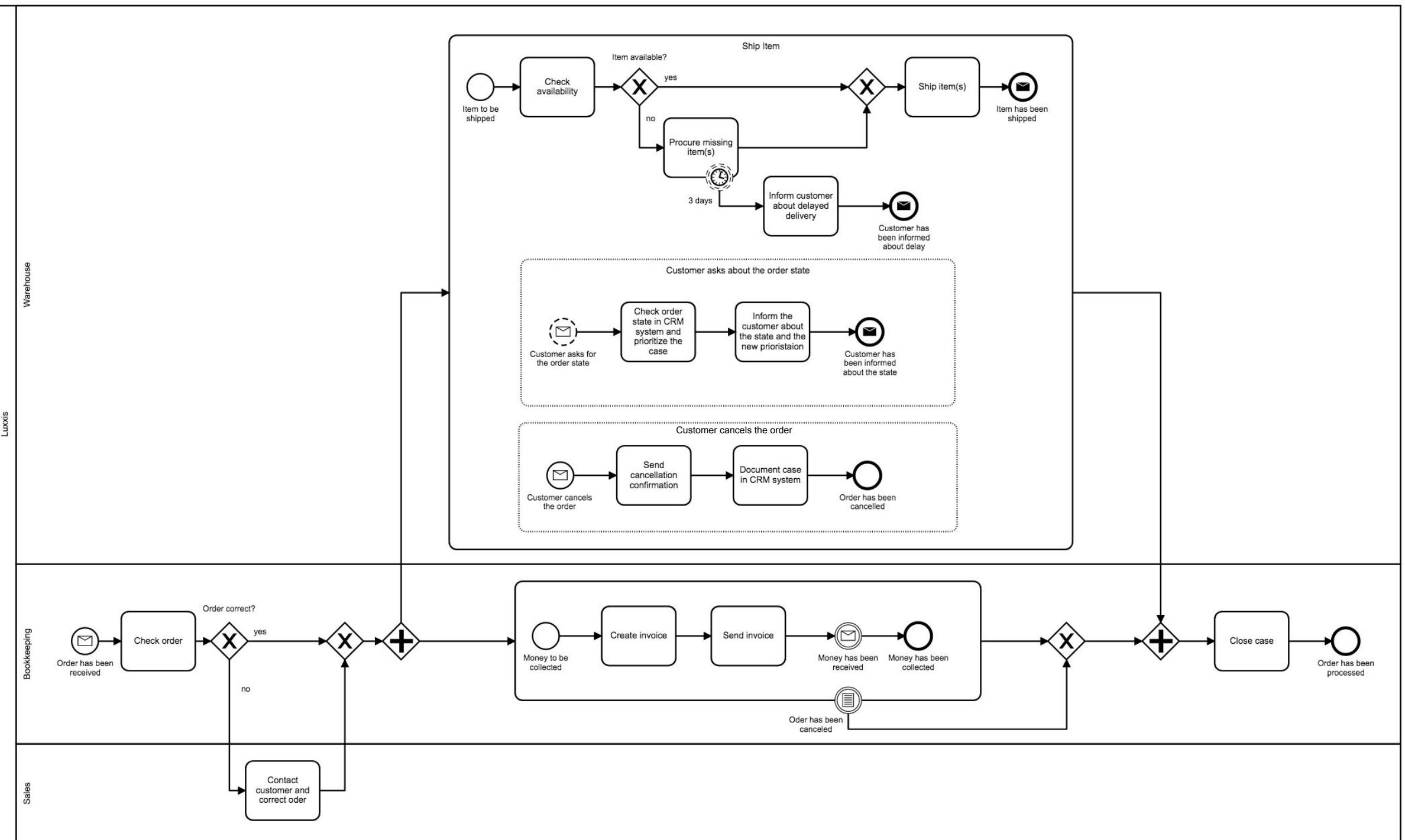
Transaction Subprocess

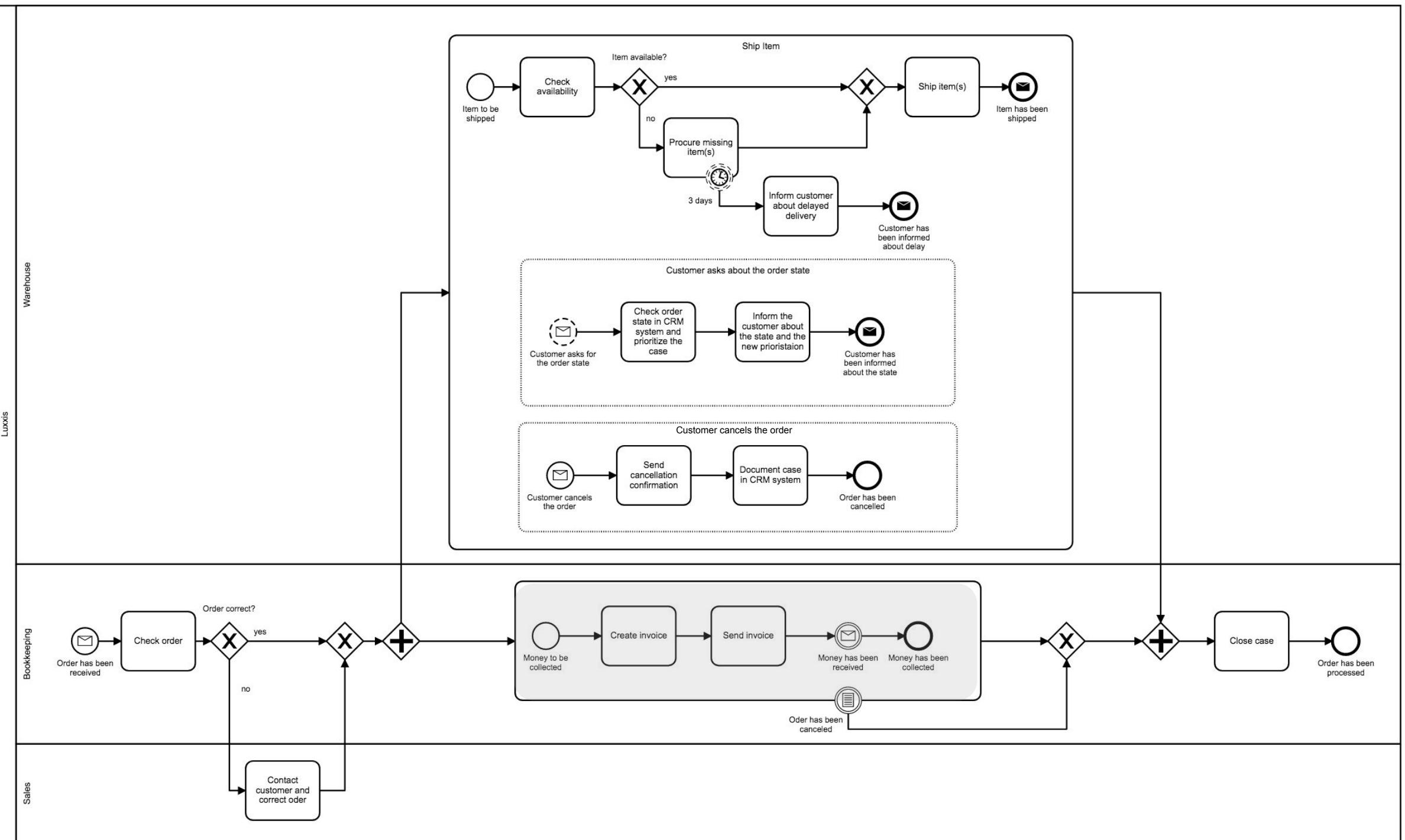


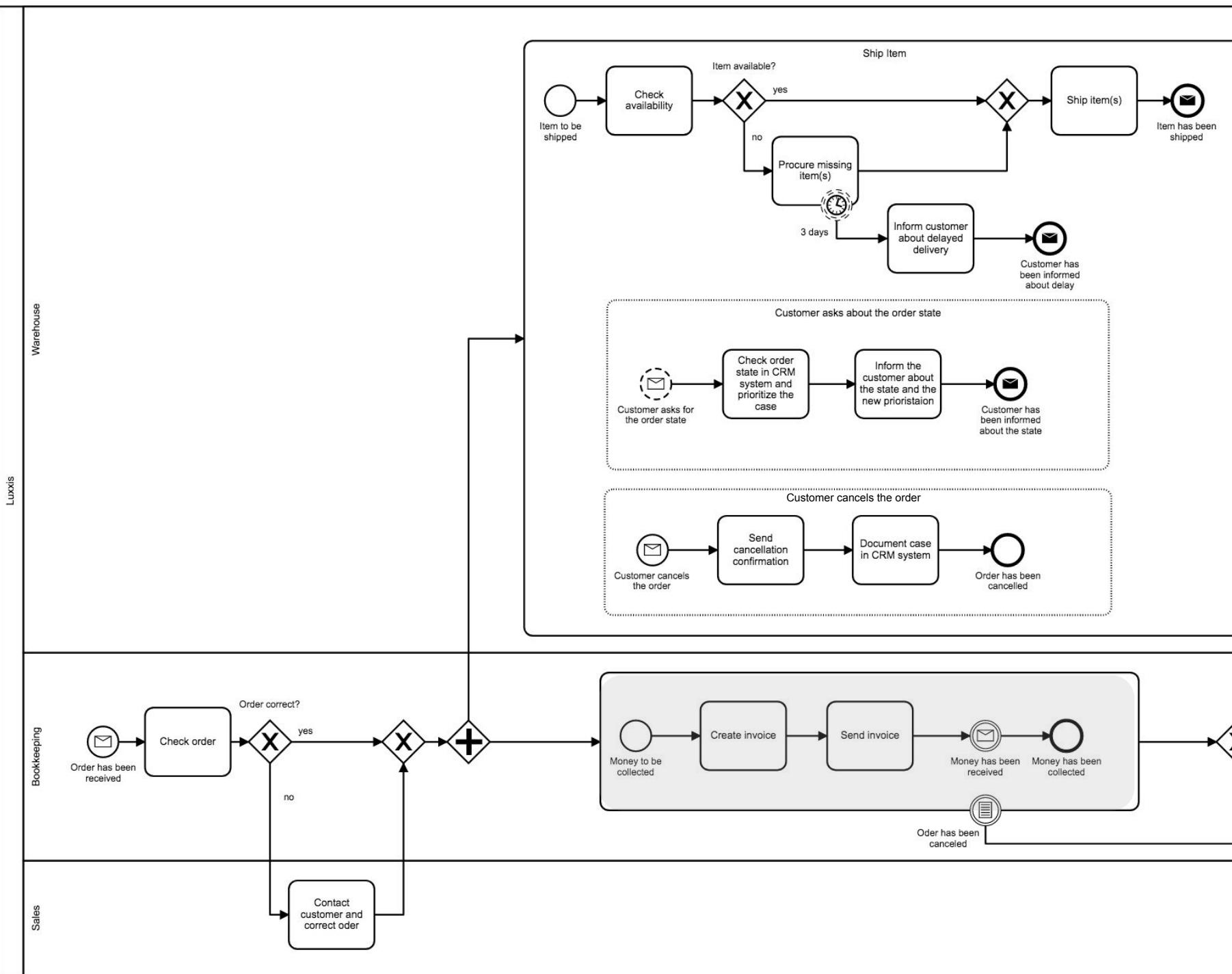
- A transaction can have three different outcomes
 - Succeeded
 - Cancelled
 - Hazard



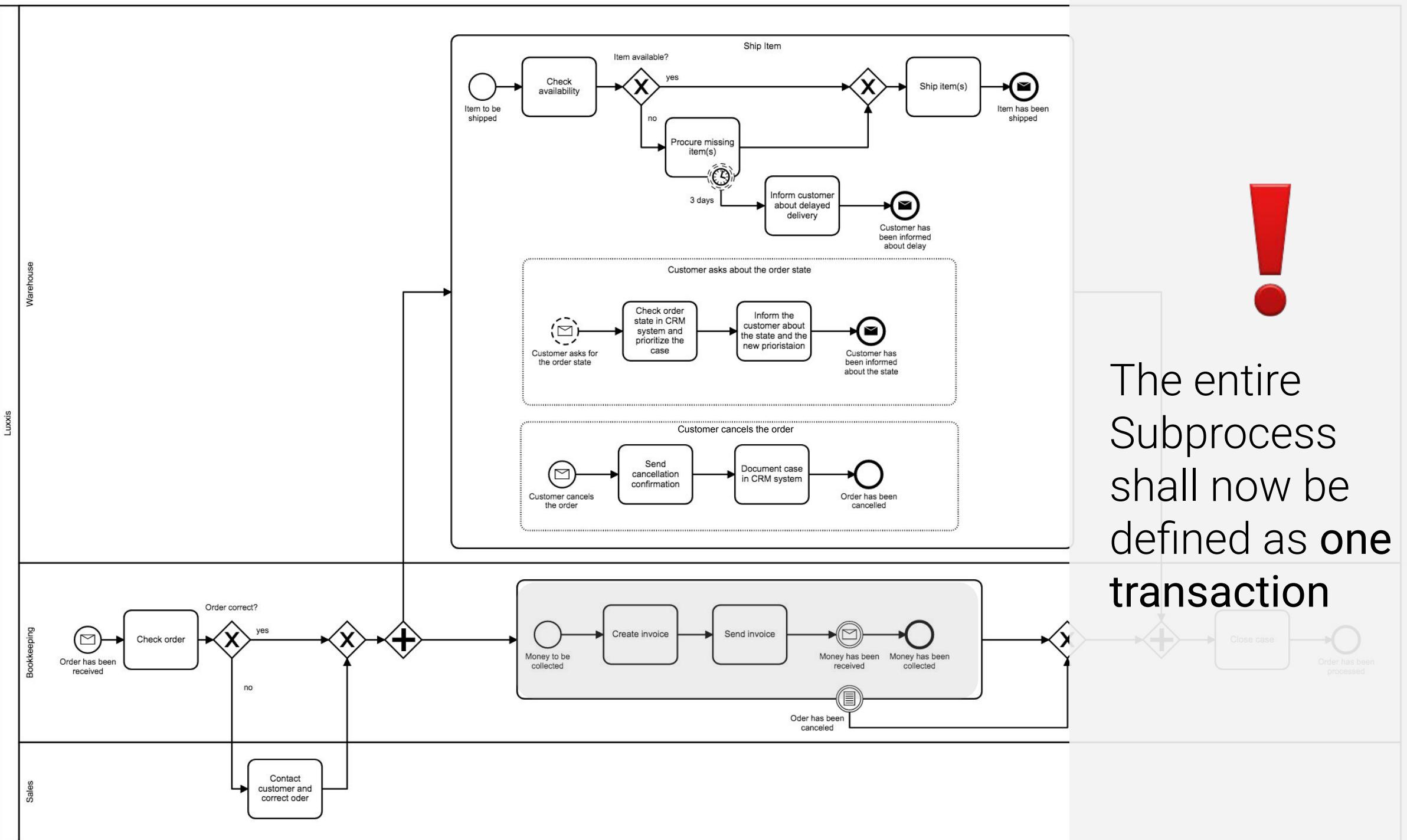
The order process at Luxxis

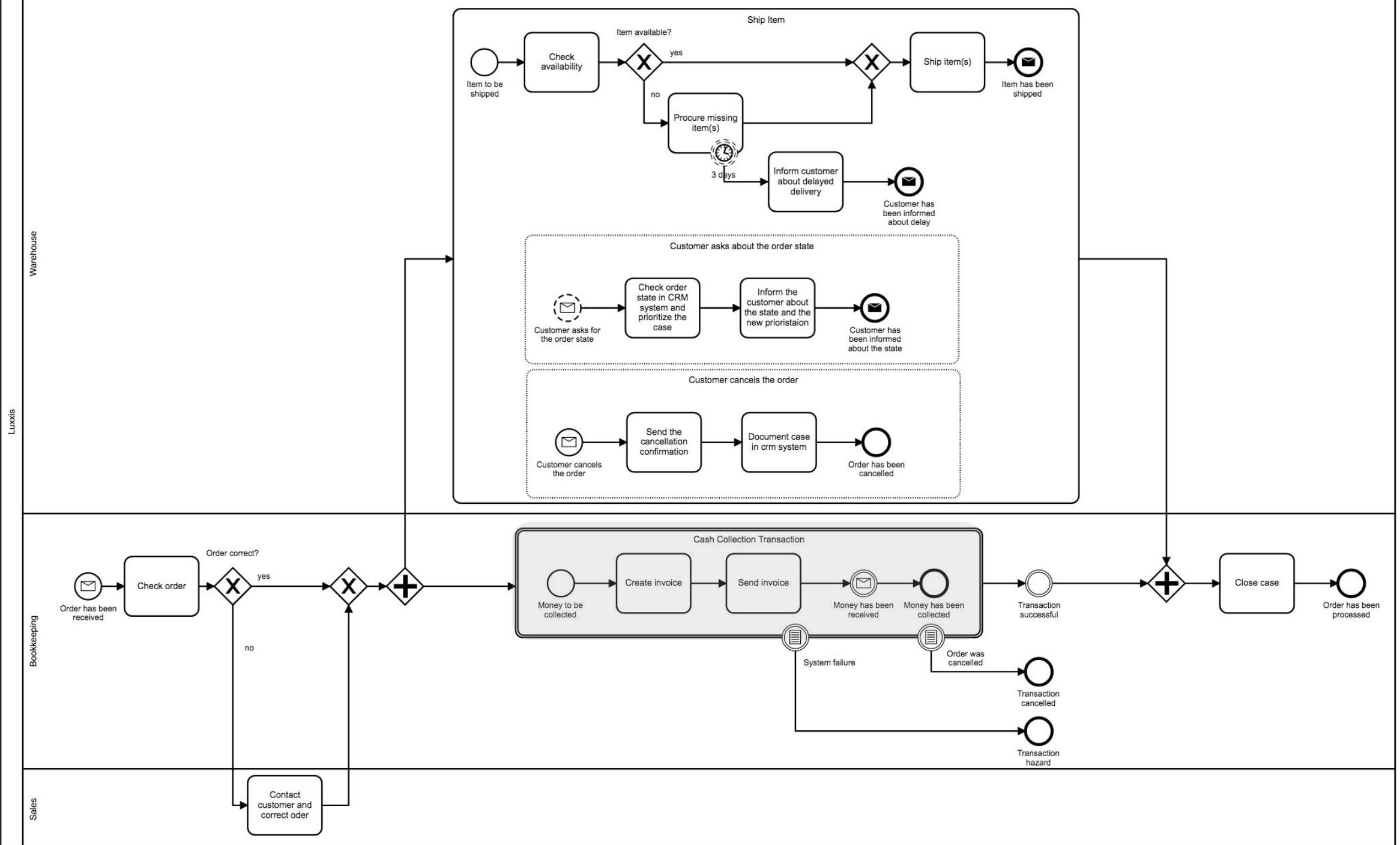


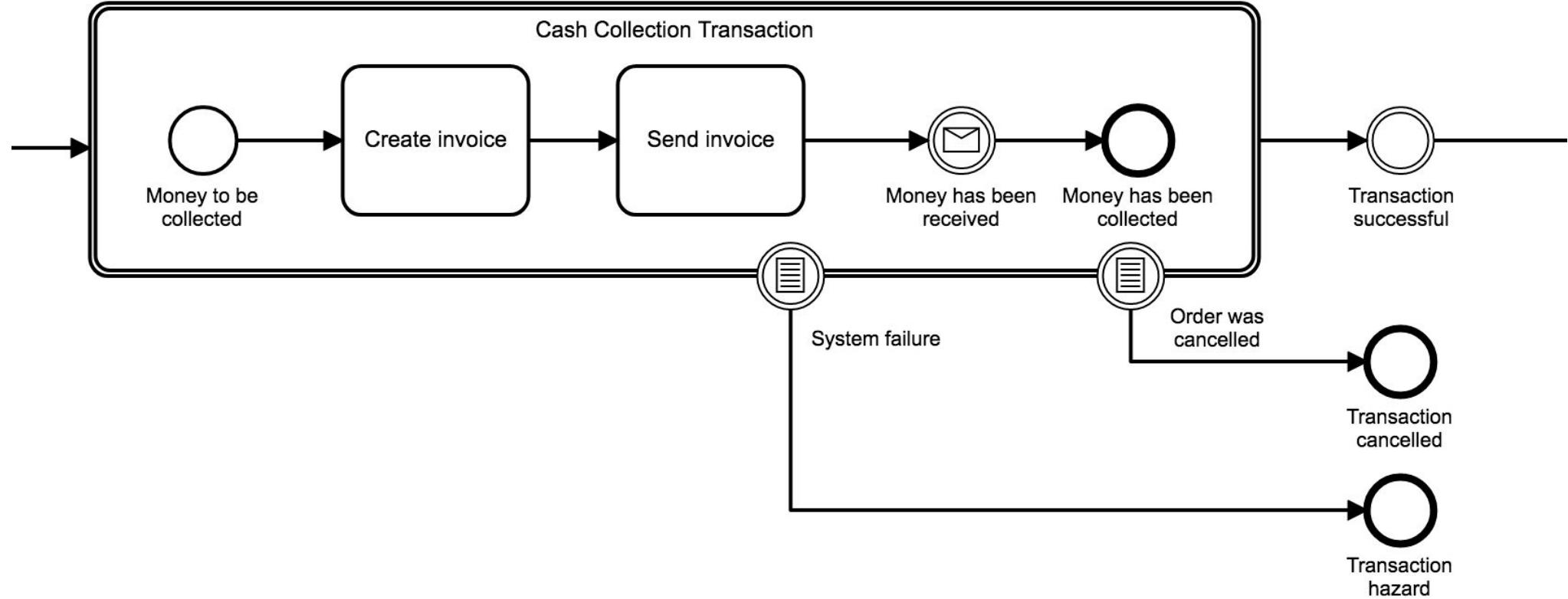




These steps
are now to be
executed by
cash collection
a system

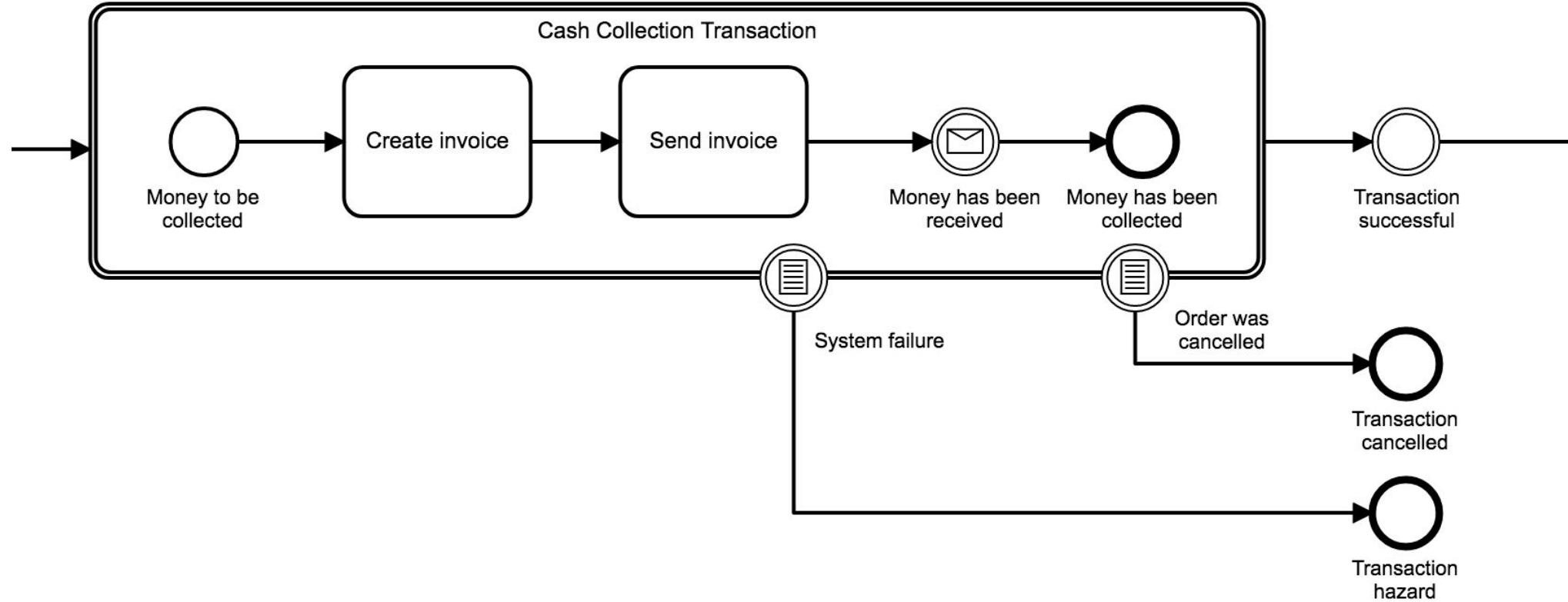




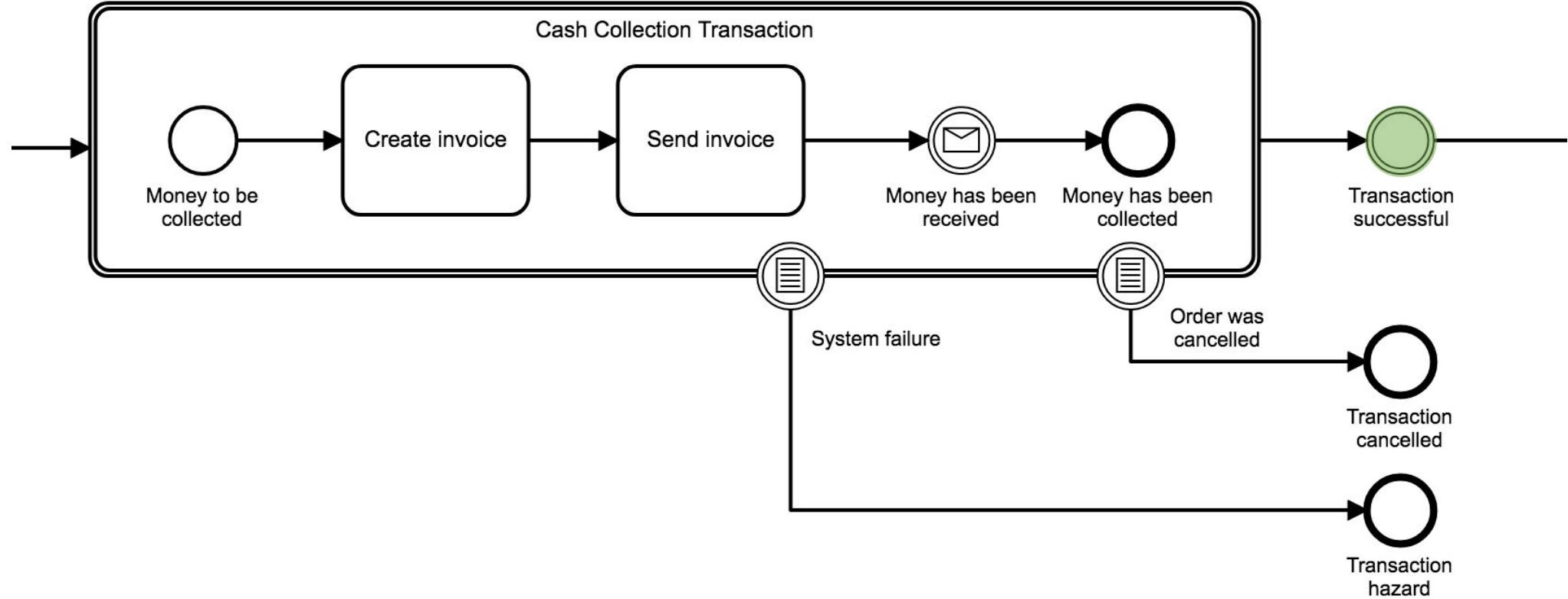


process cc

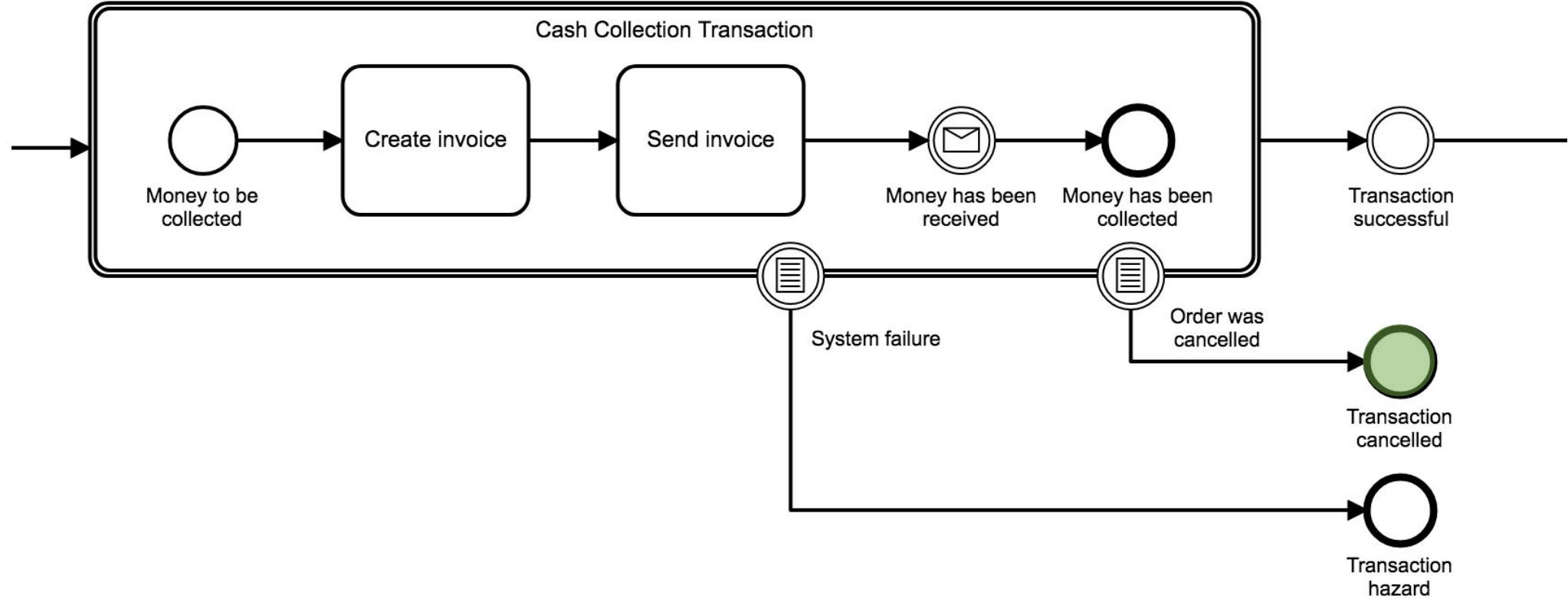
Logical Unit of Work



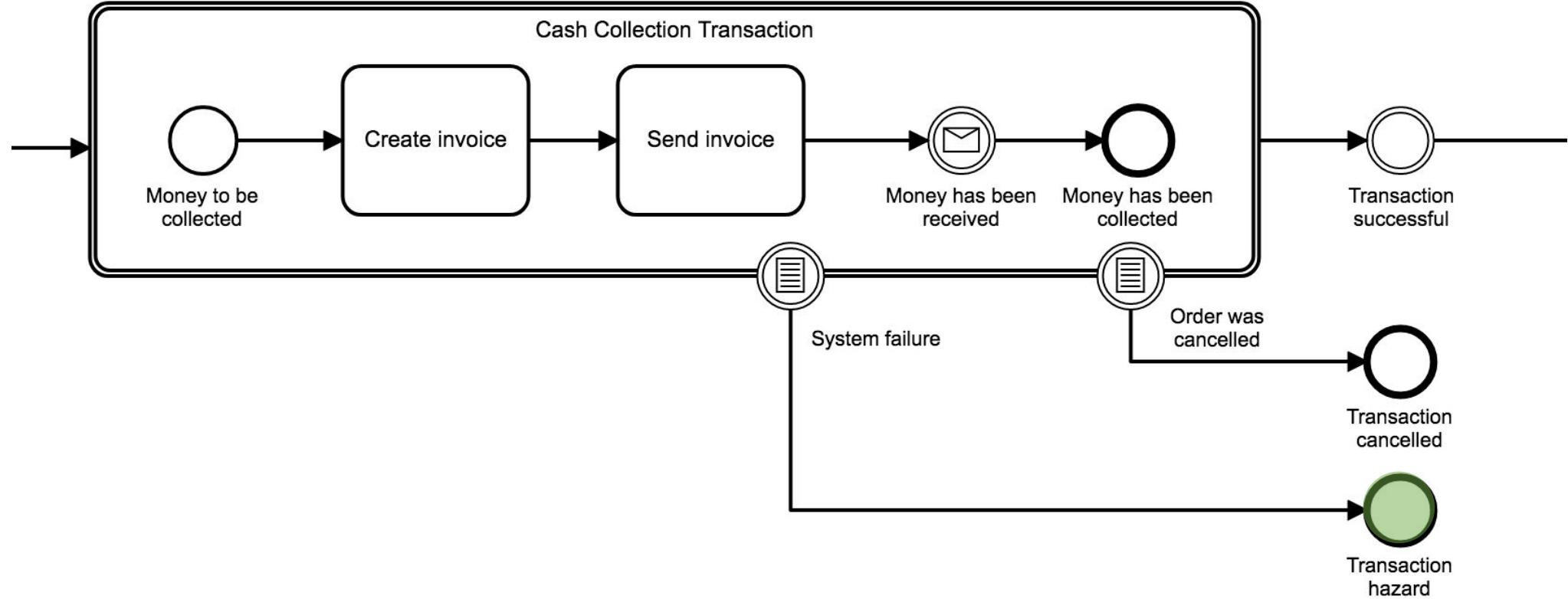
process cc



process cc

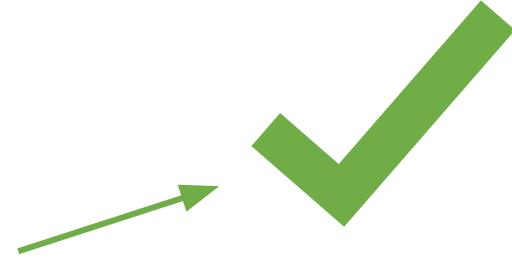
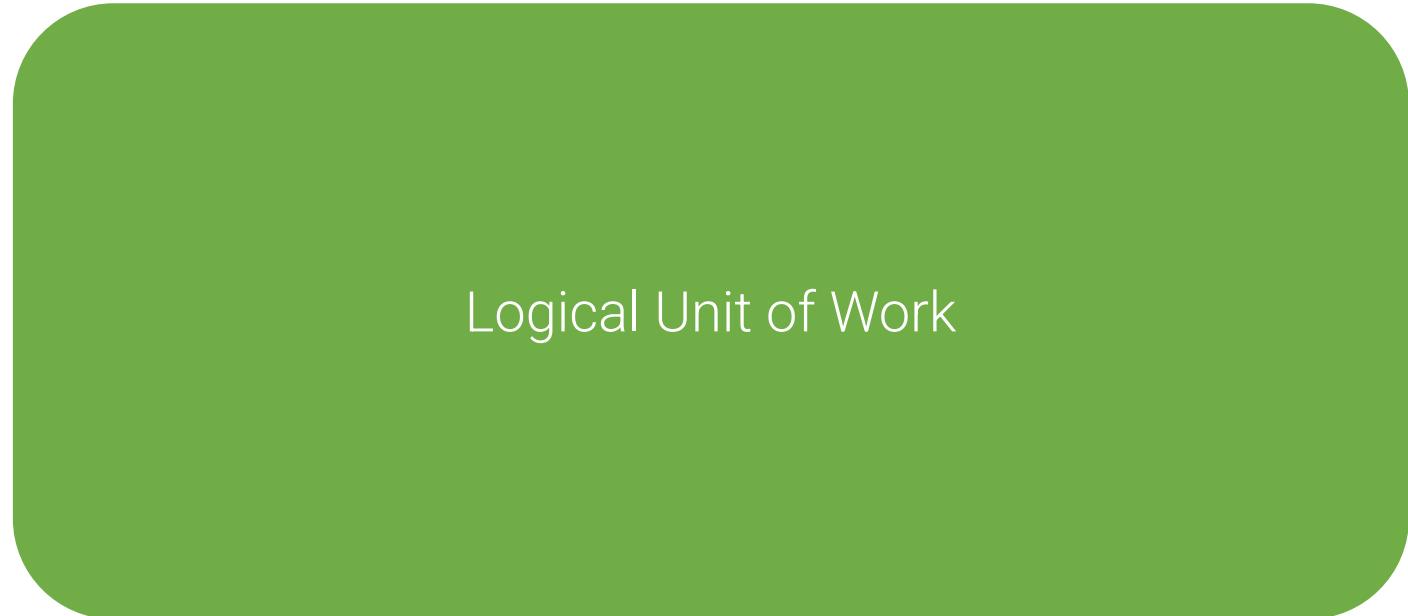


process cc

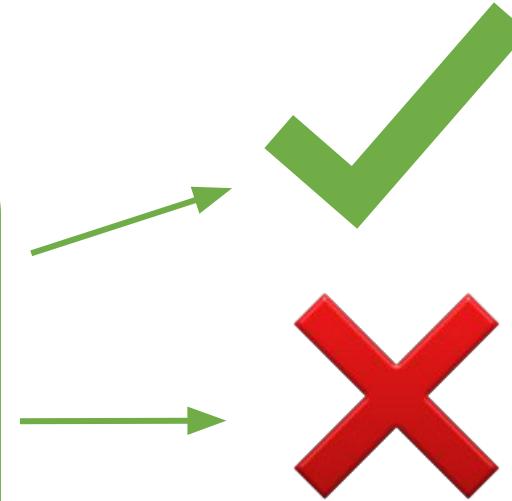
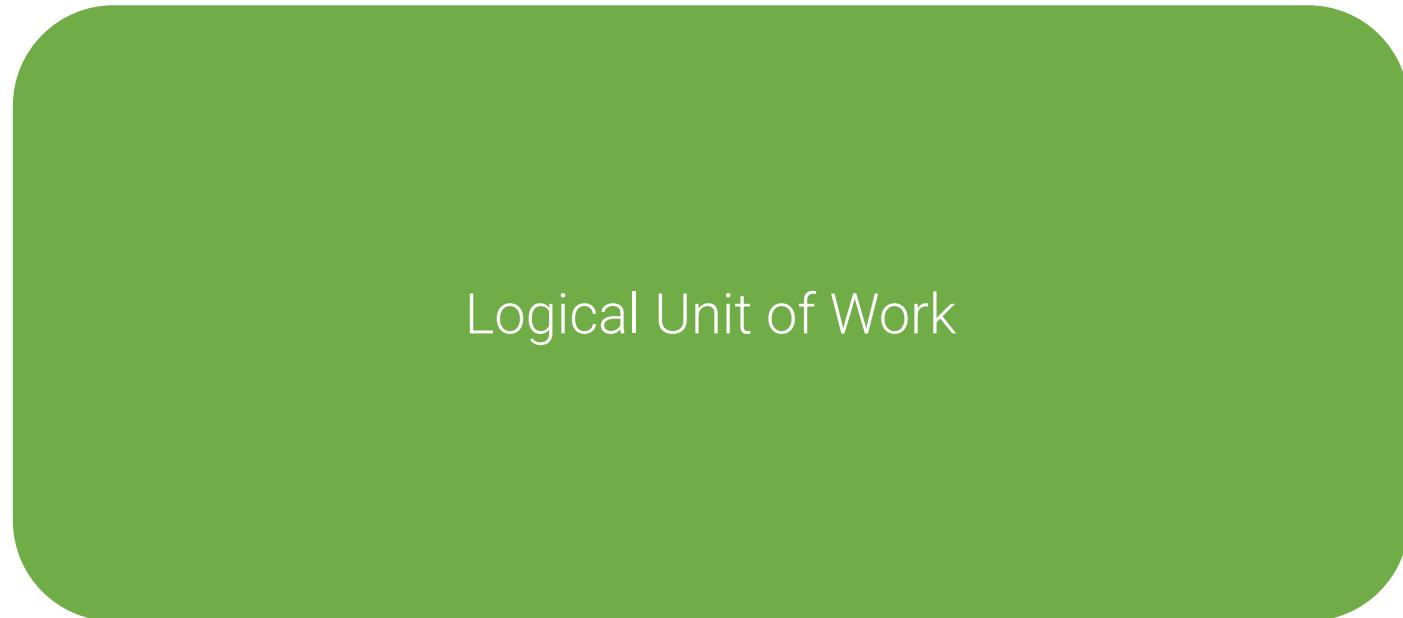


process cc

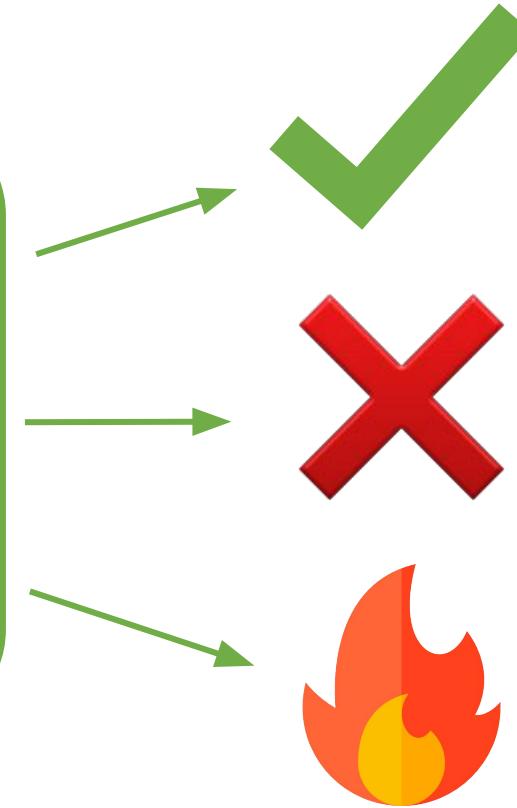
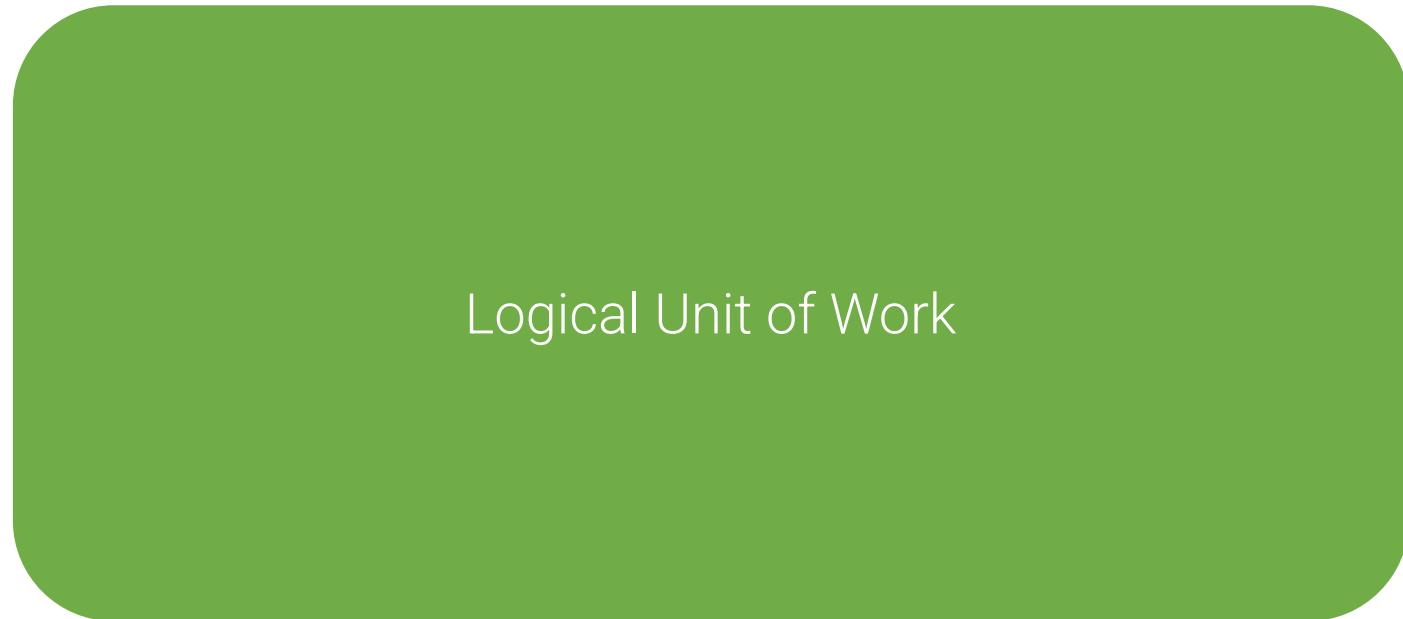
Logical Unit of Work



process cc



process cc



process cc

Loop

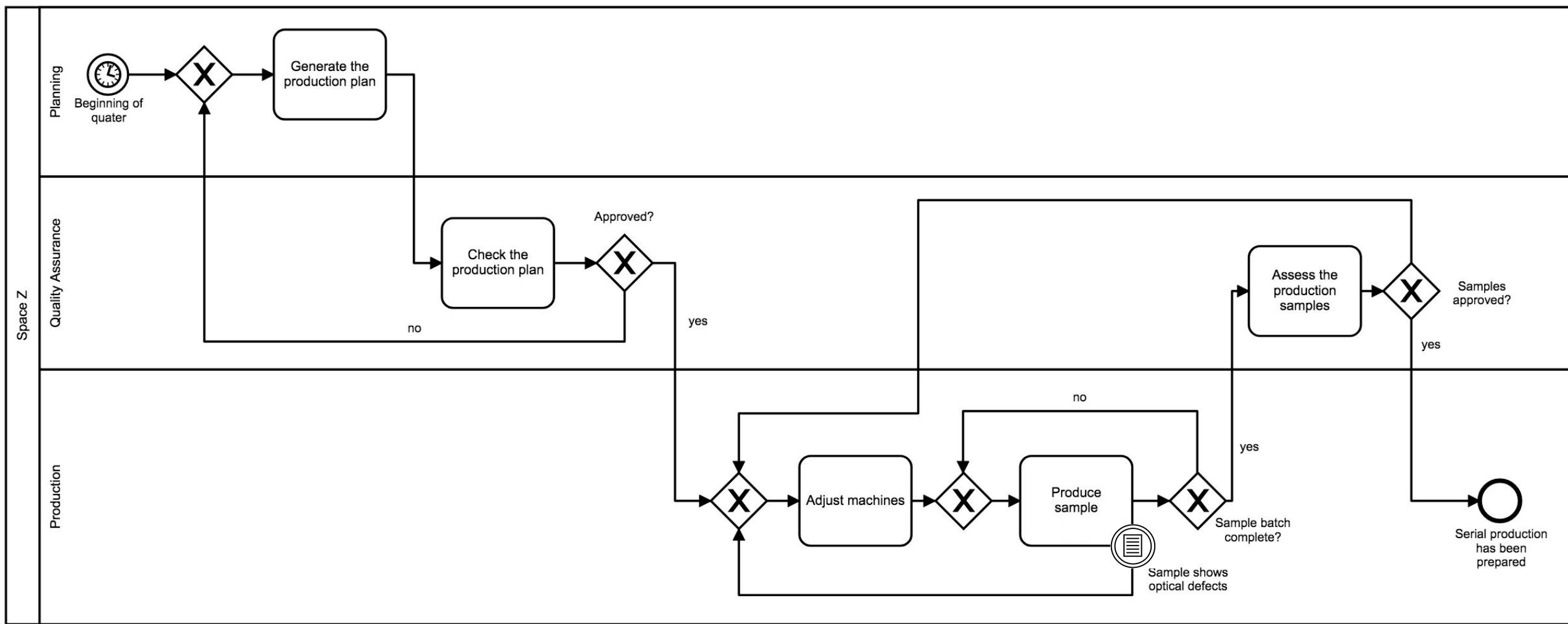
Task

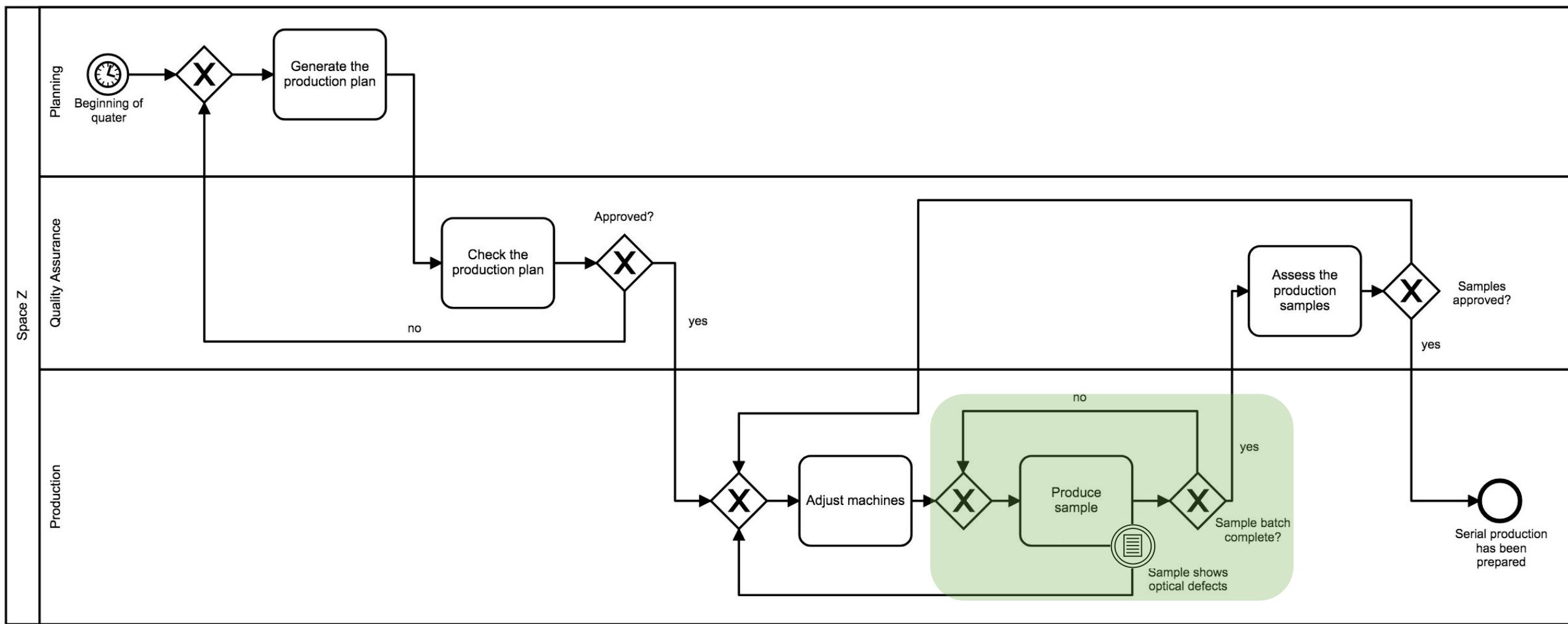


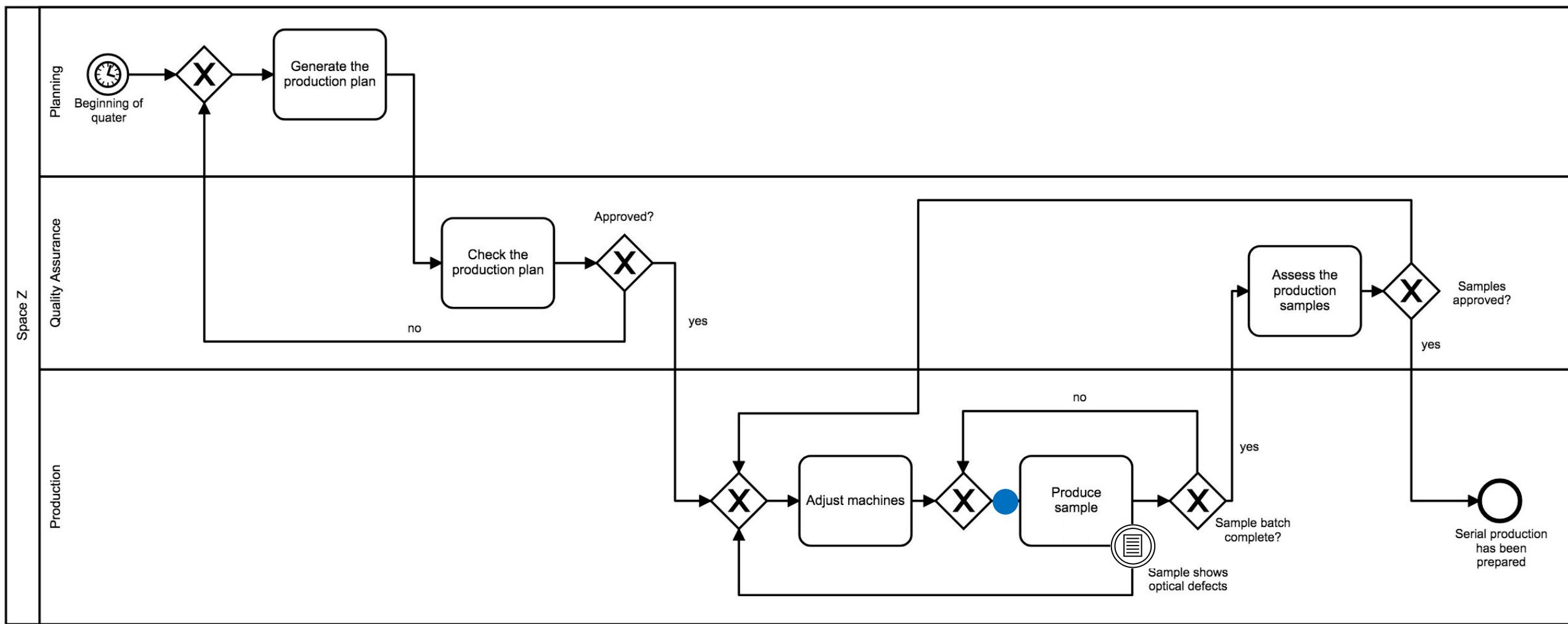
- Defines an activity that is executed multiple times until a defined condition is satisfied

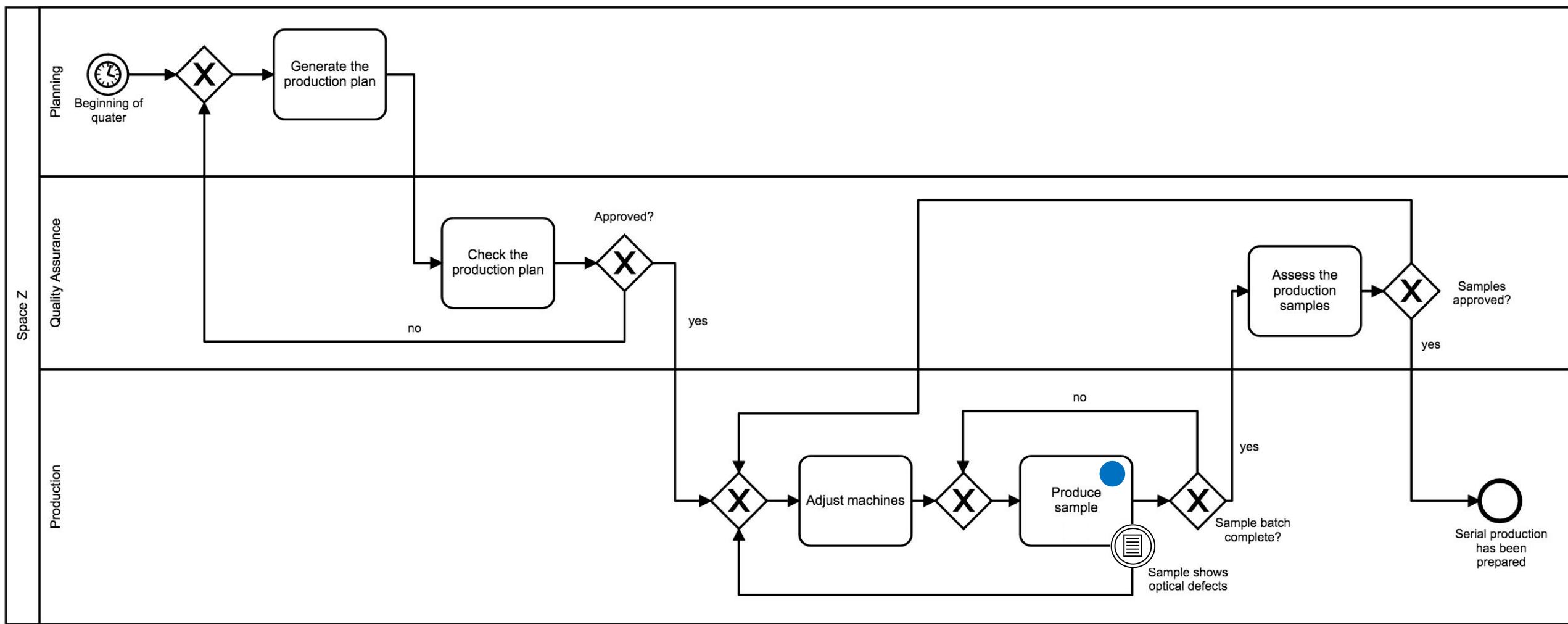


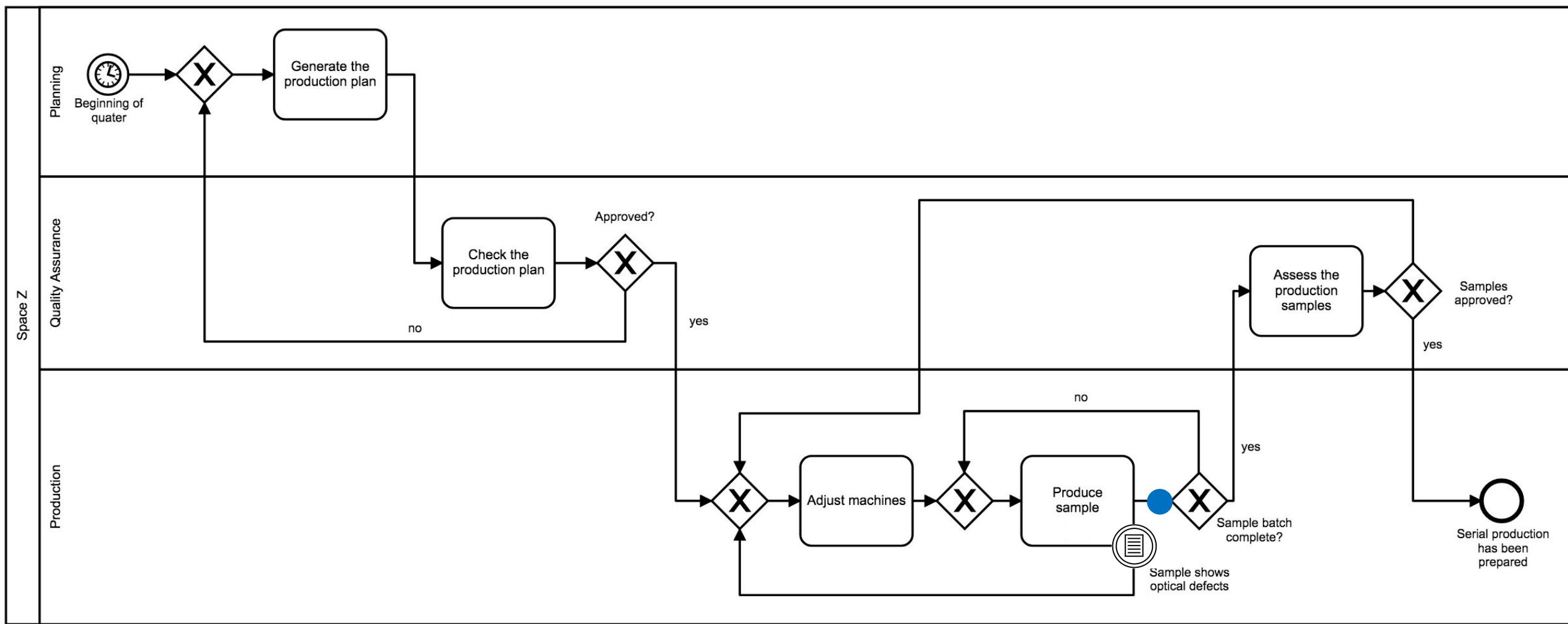
Sample production at Space Z

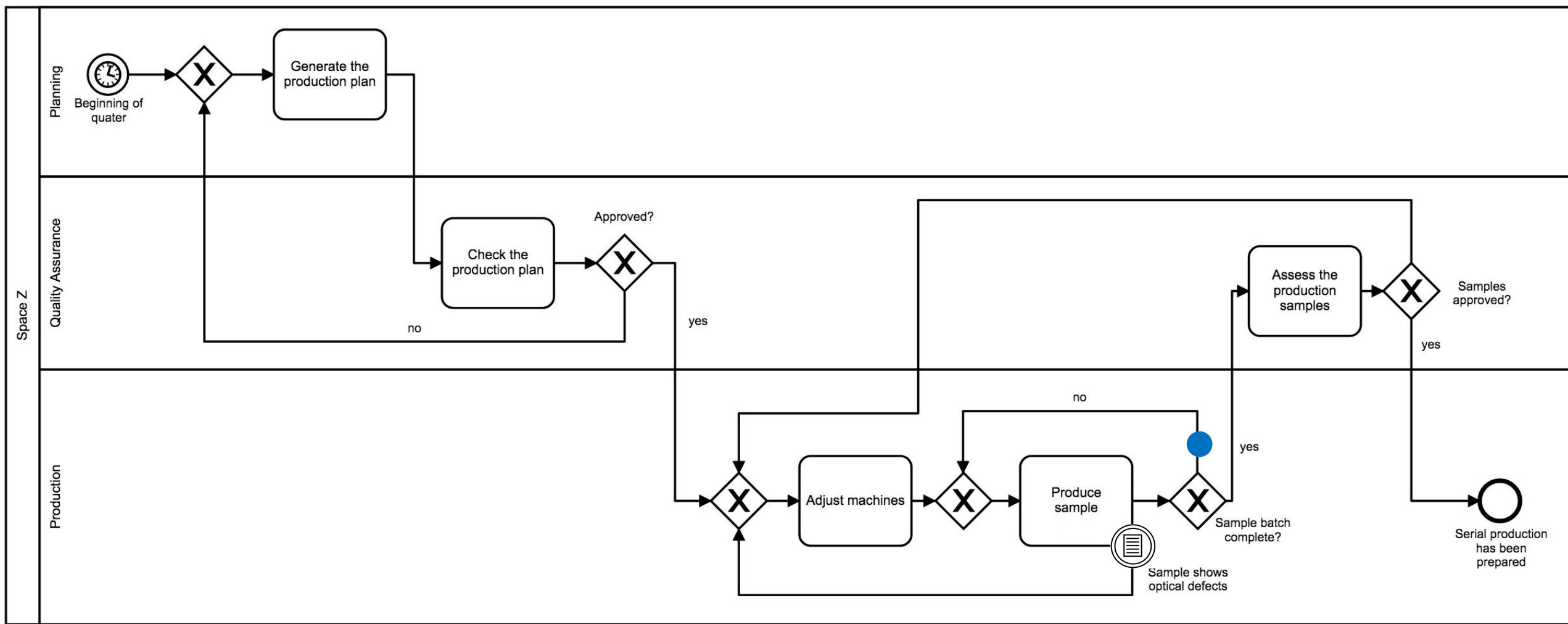


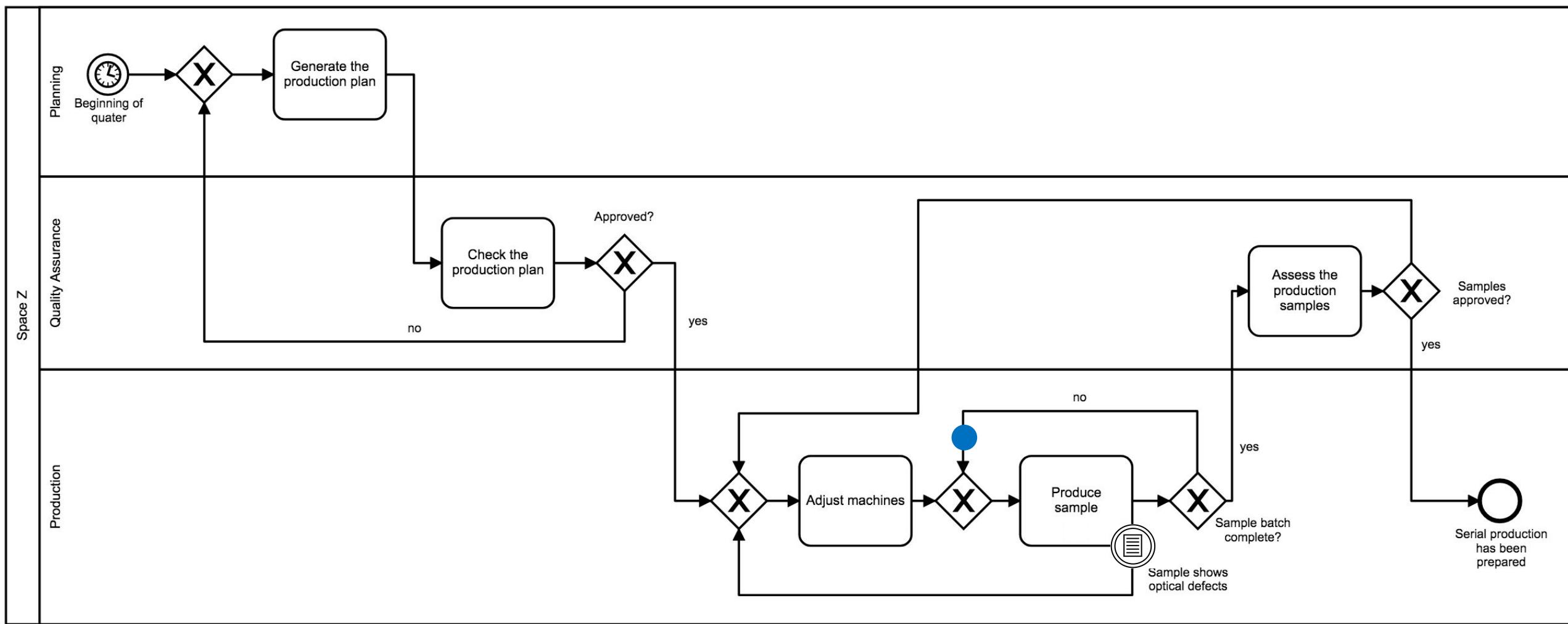


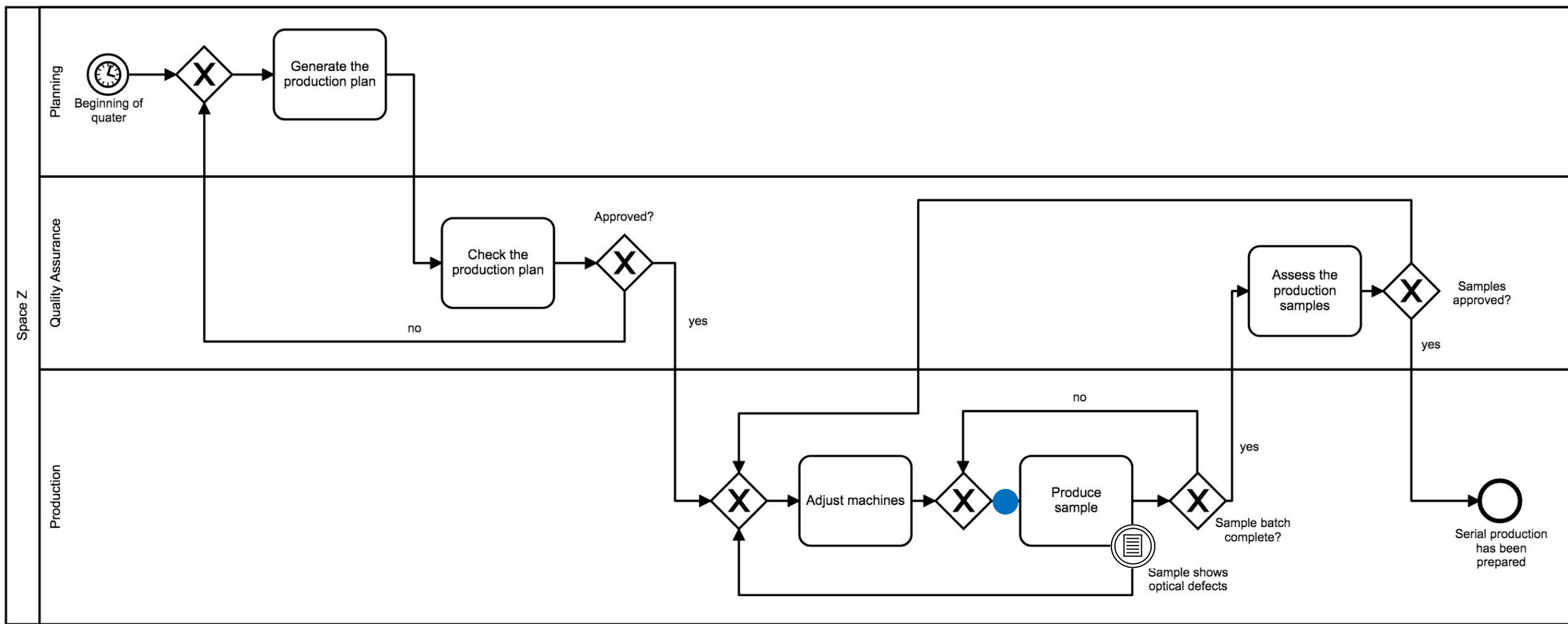


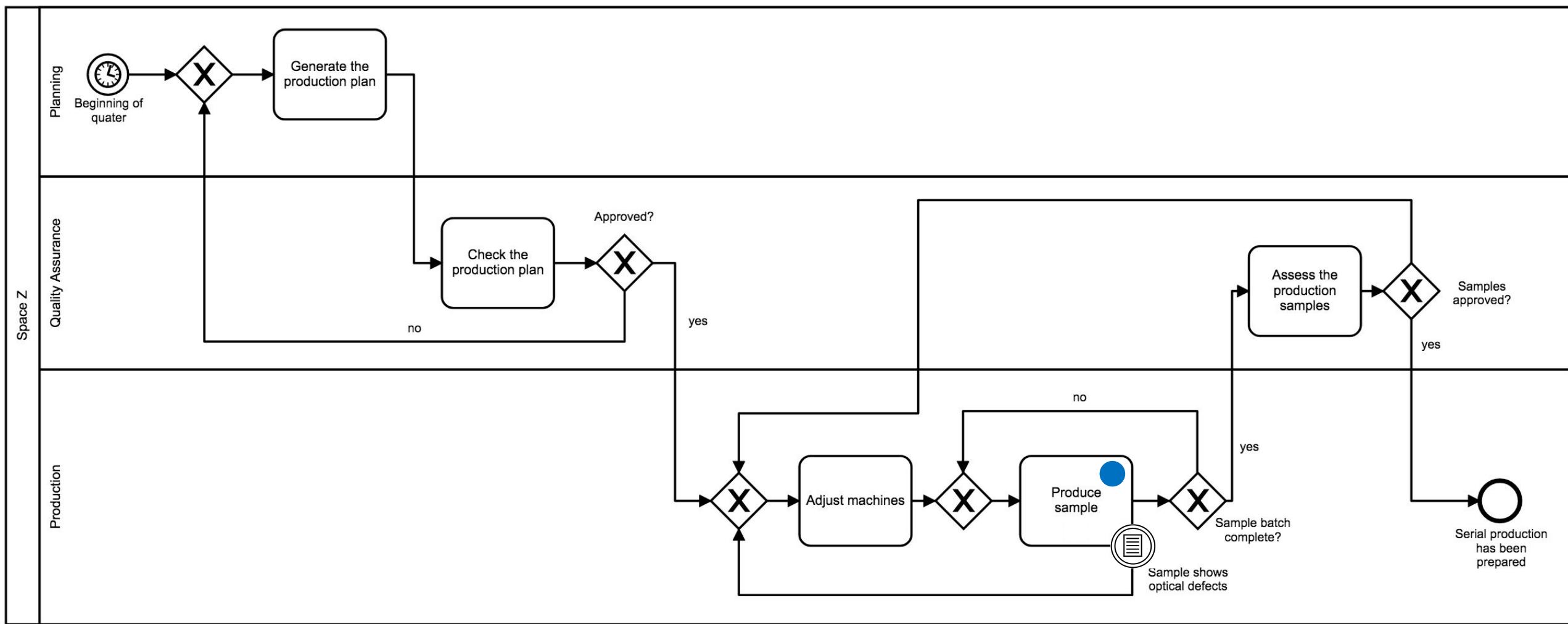


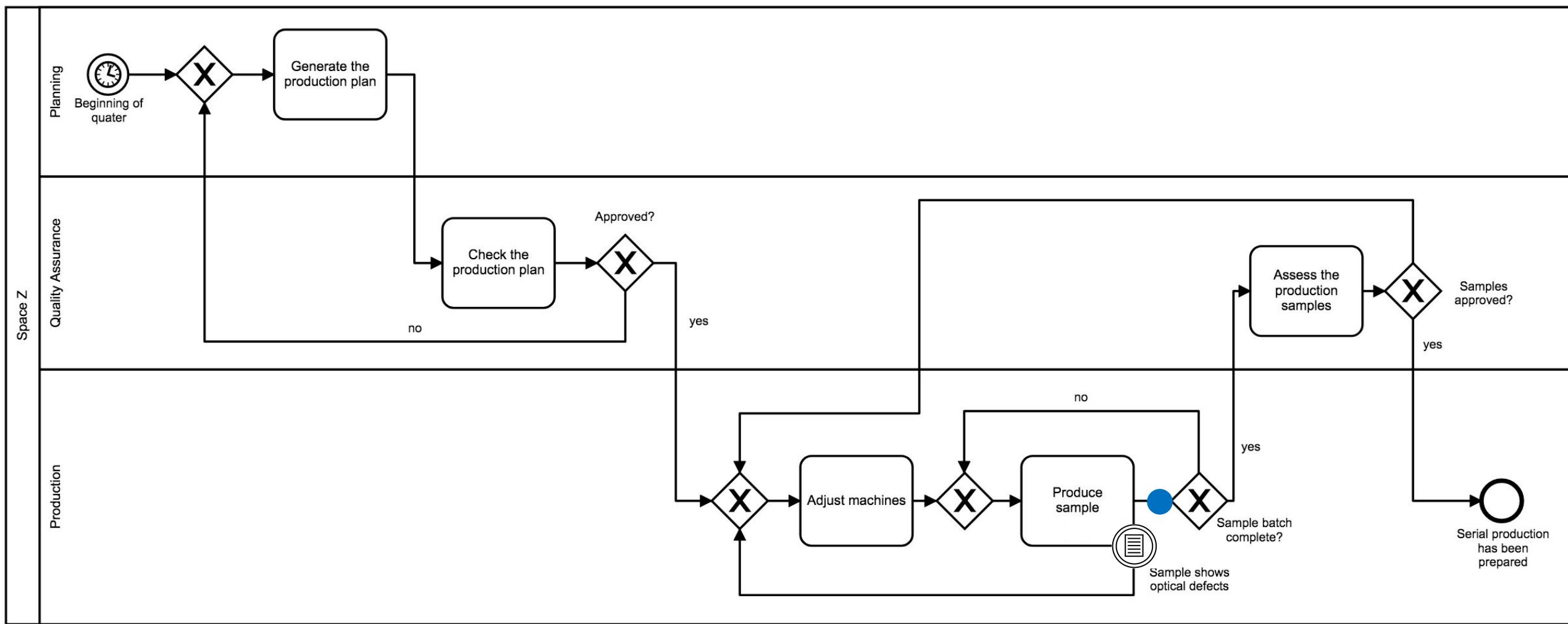


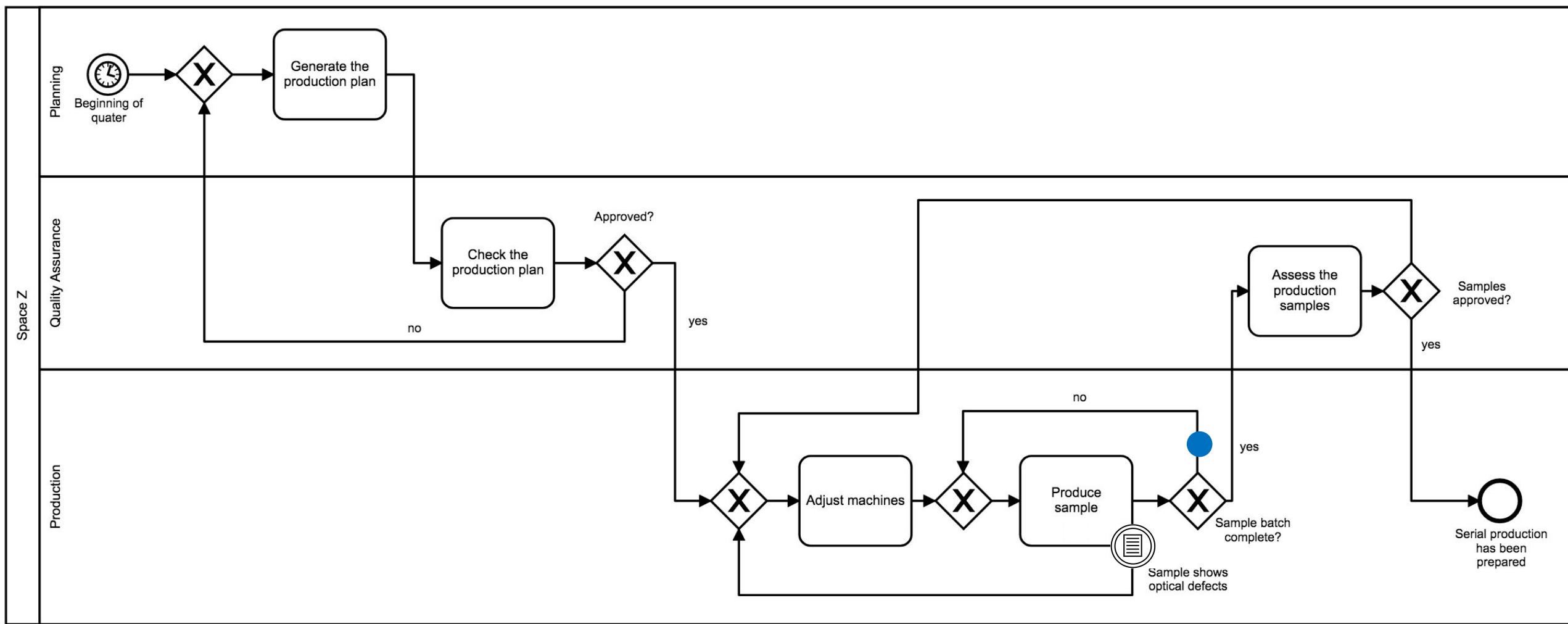


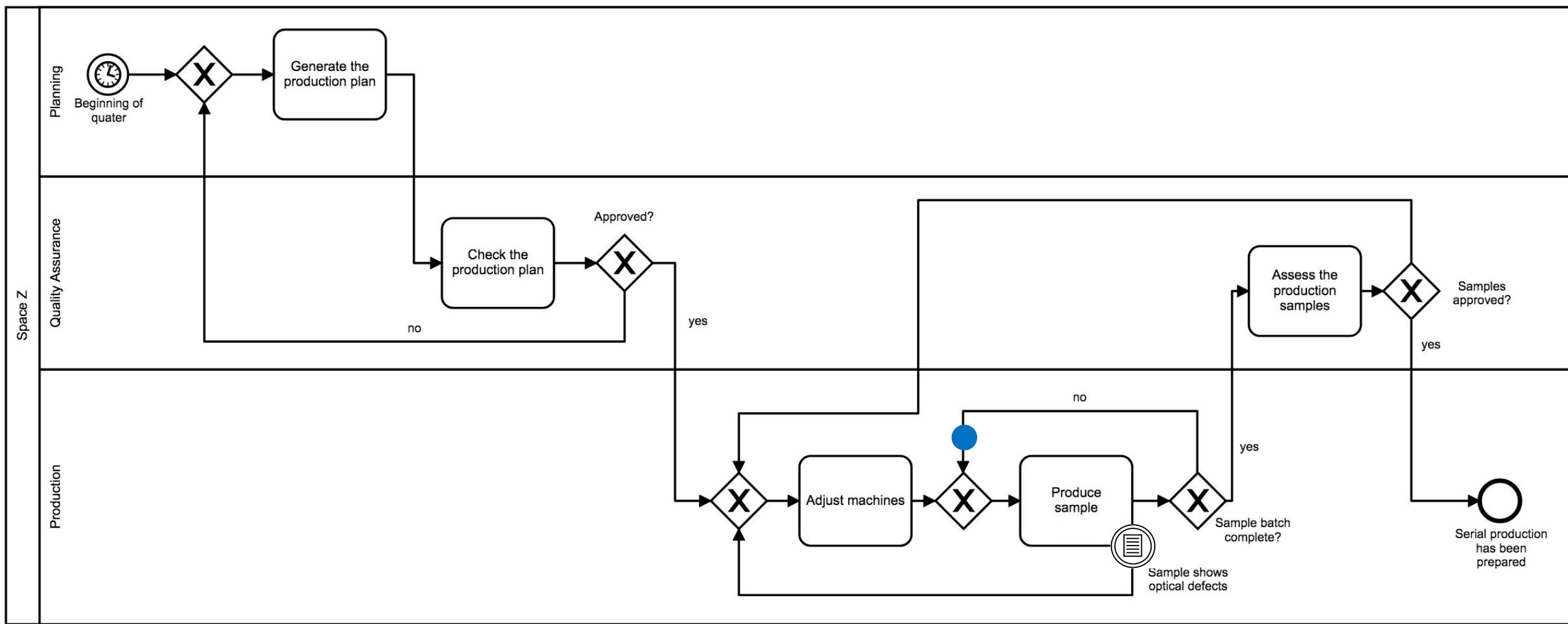


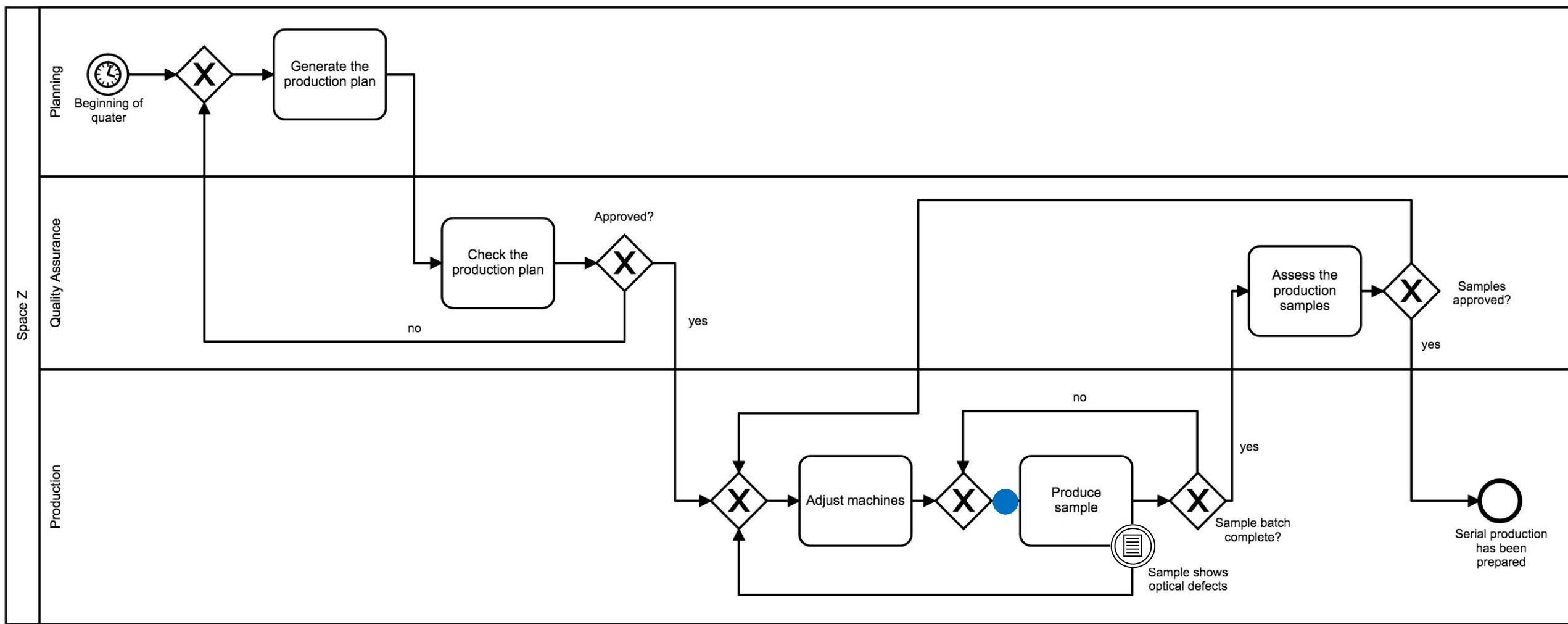


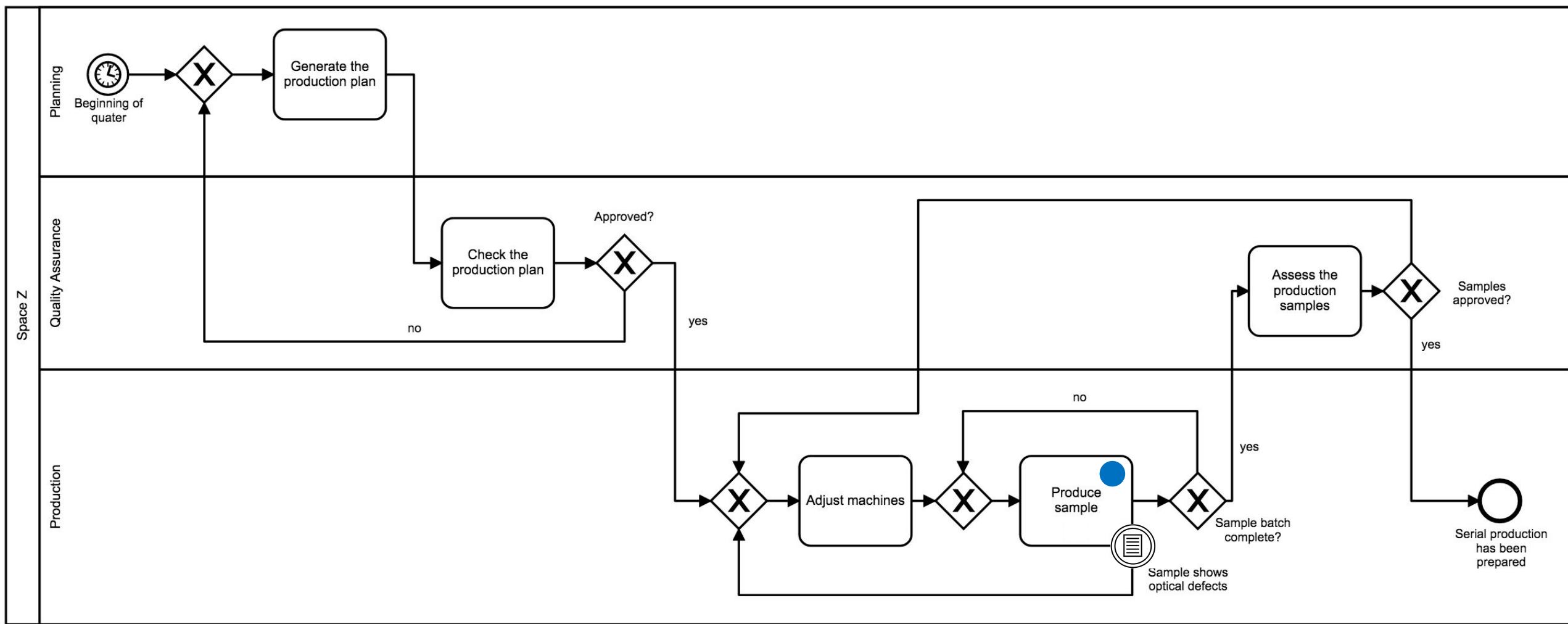


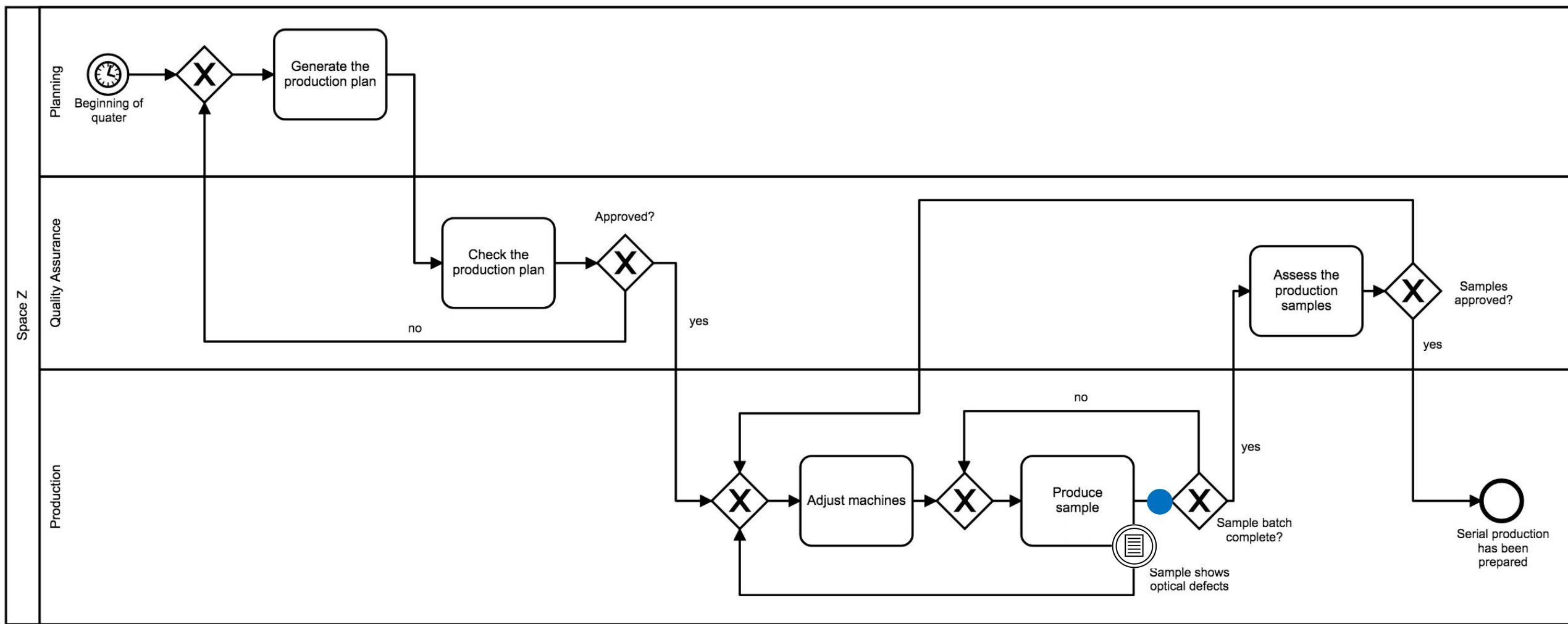


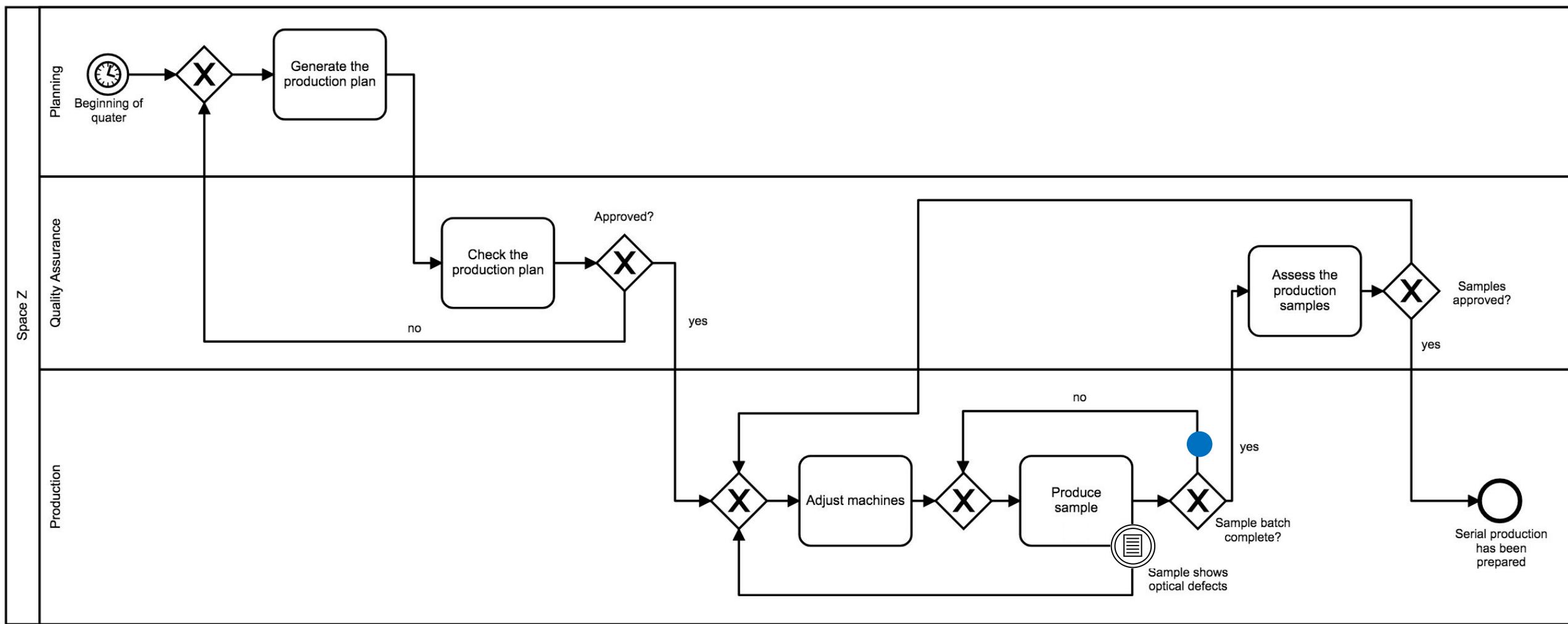


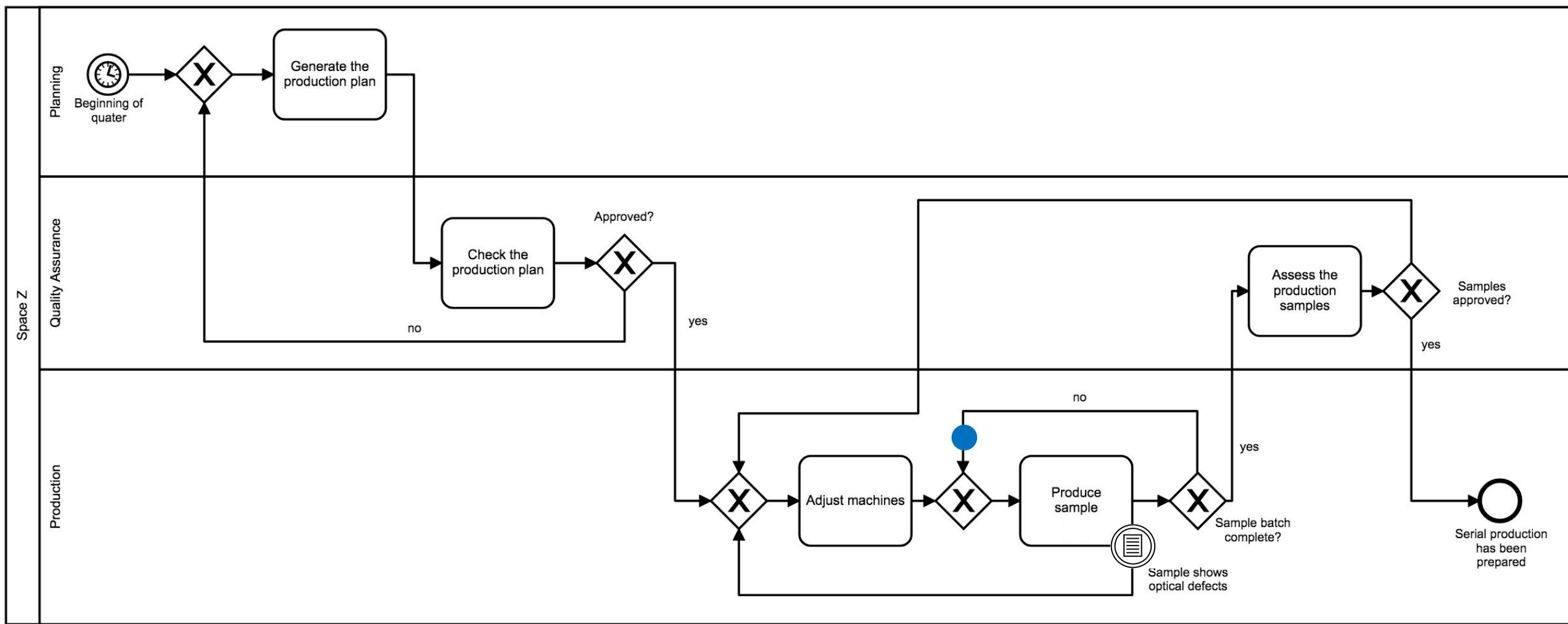


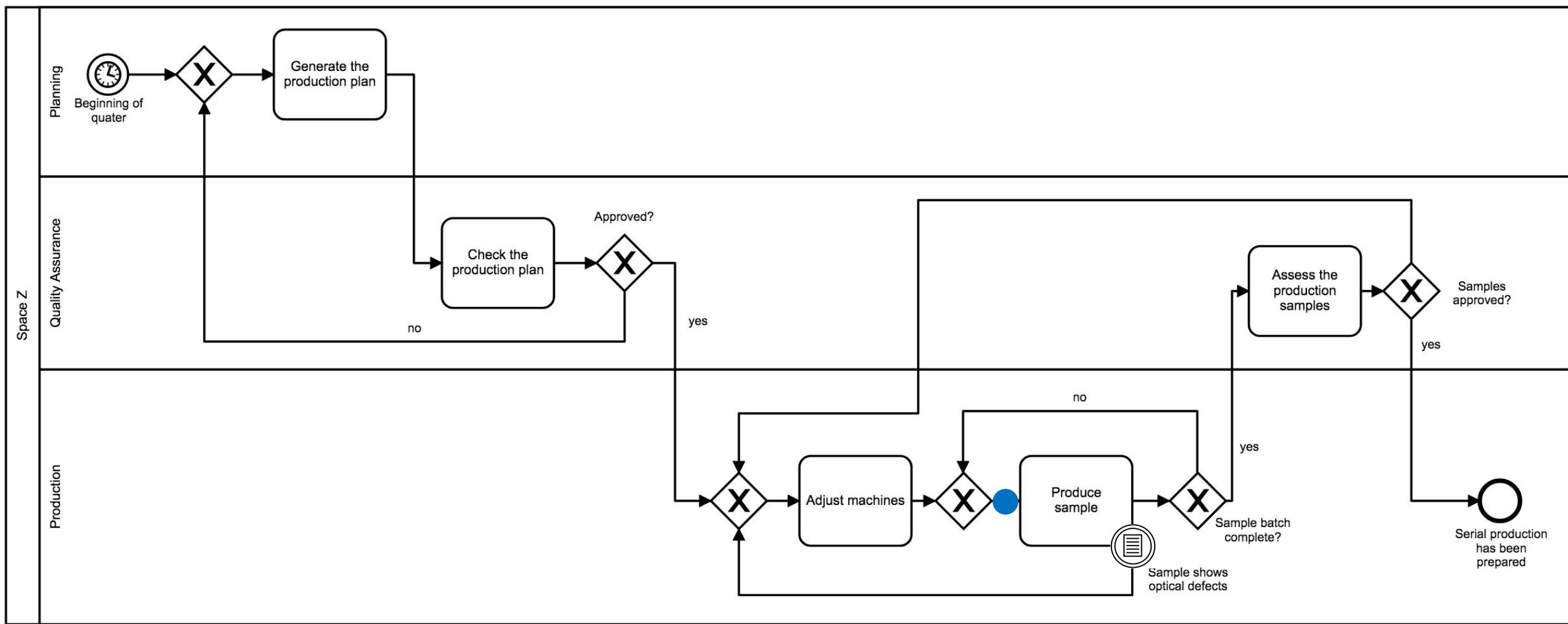


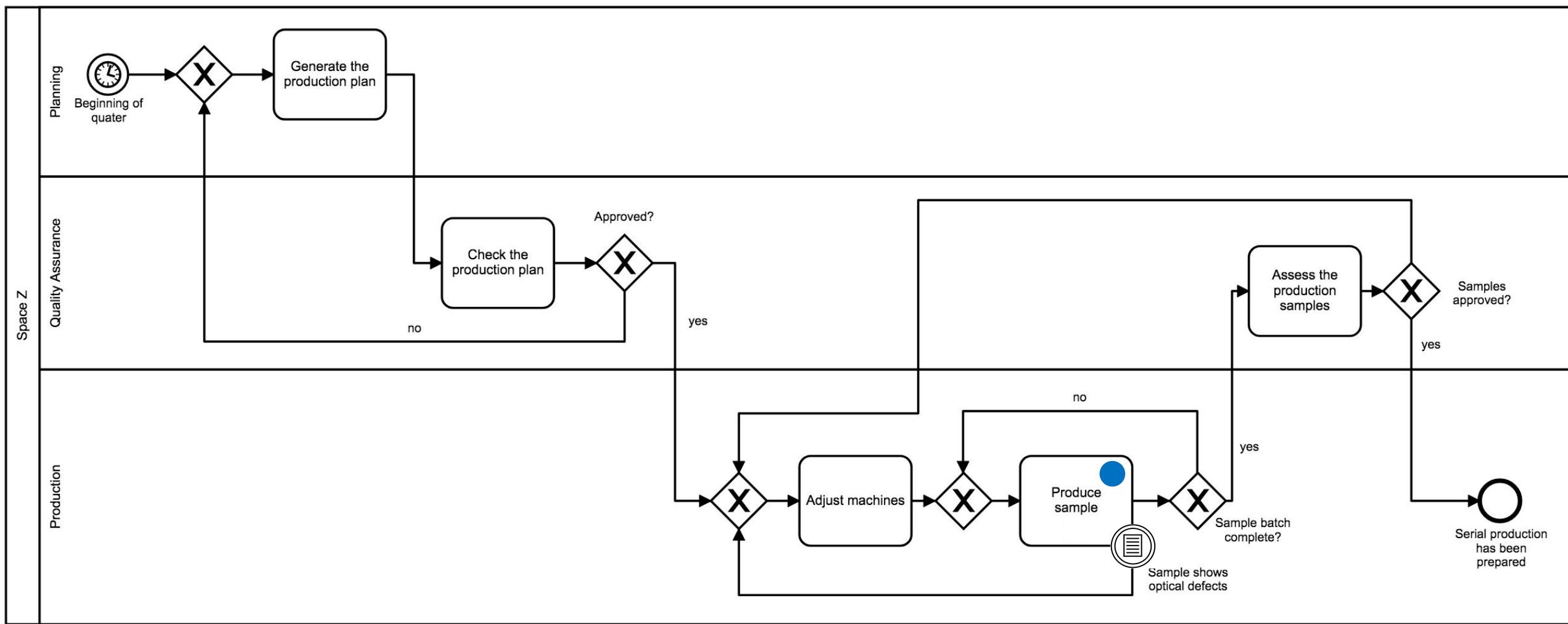


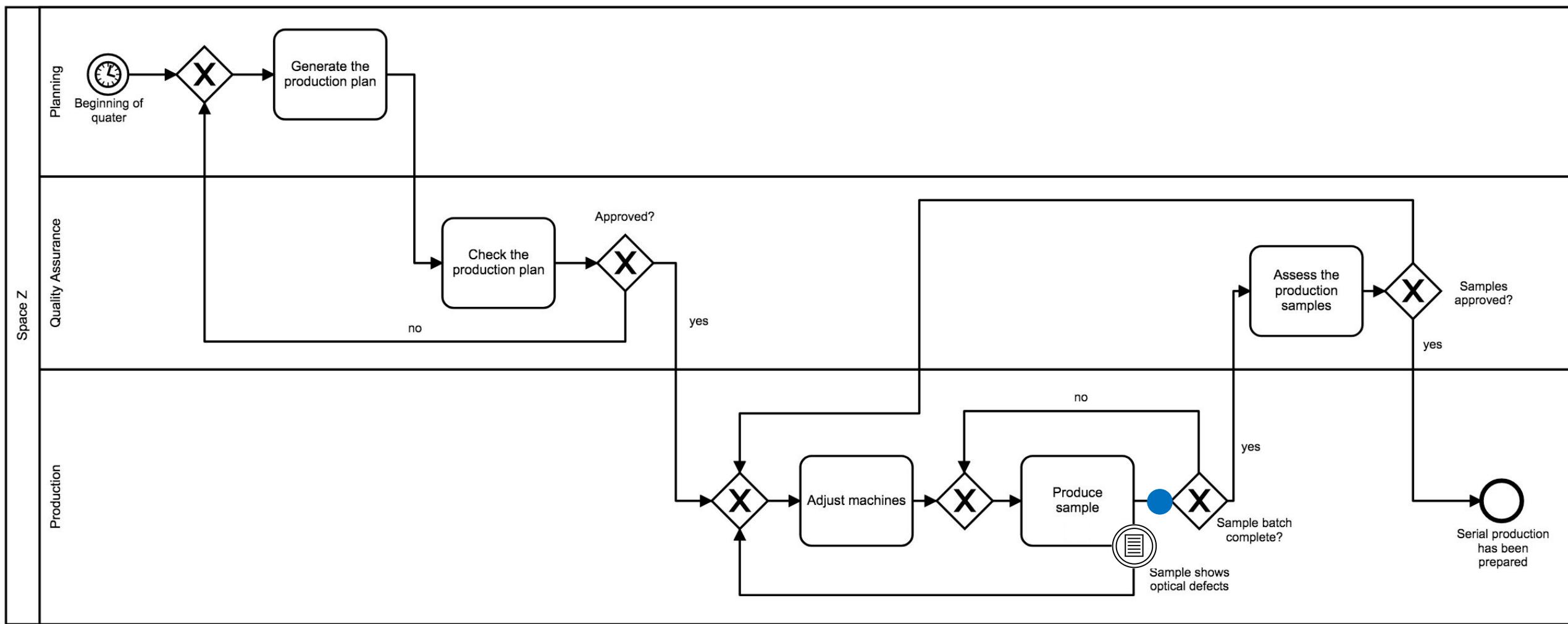


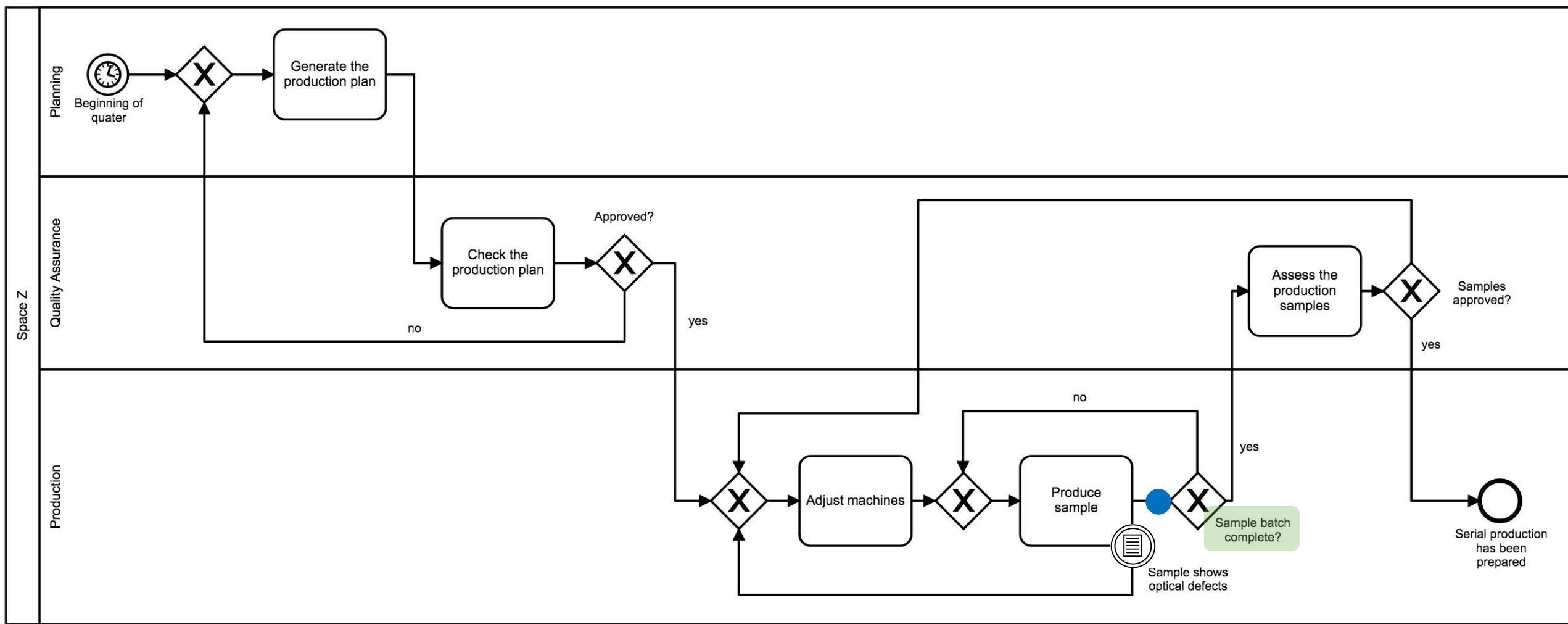


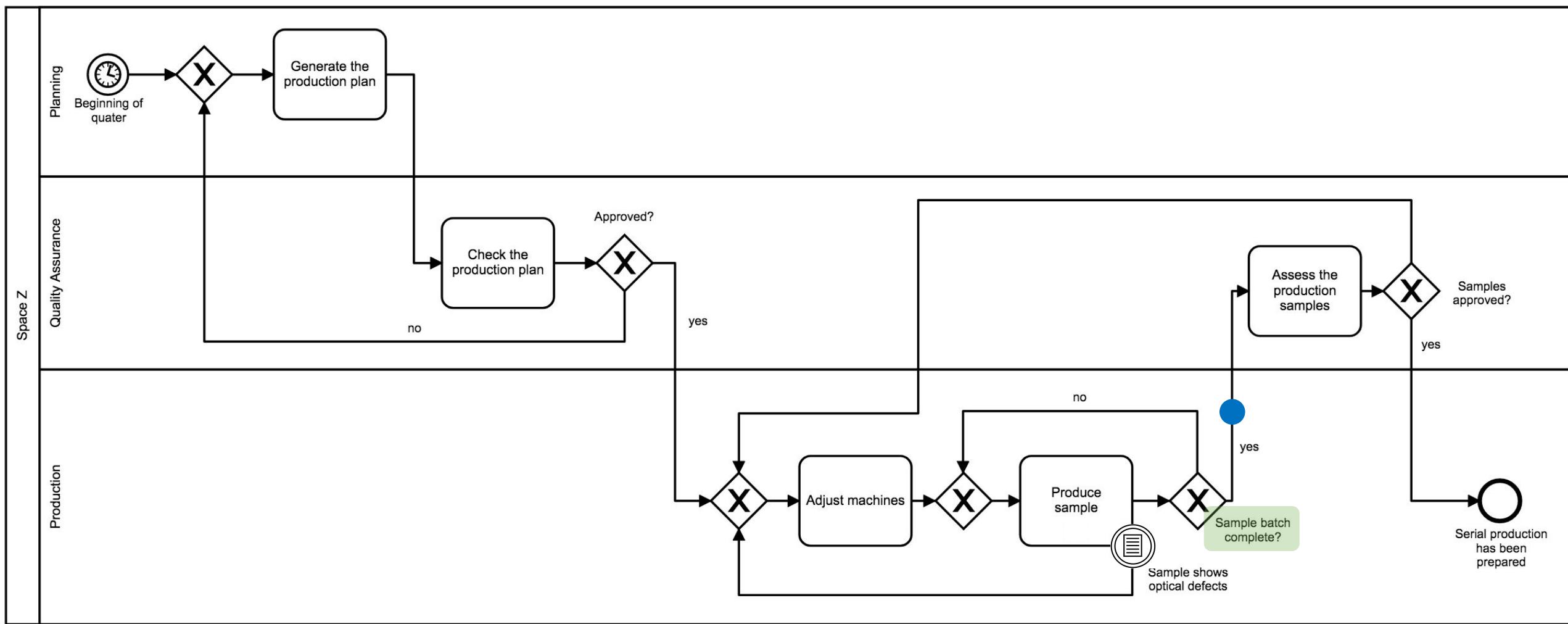


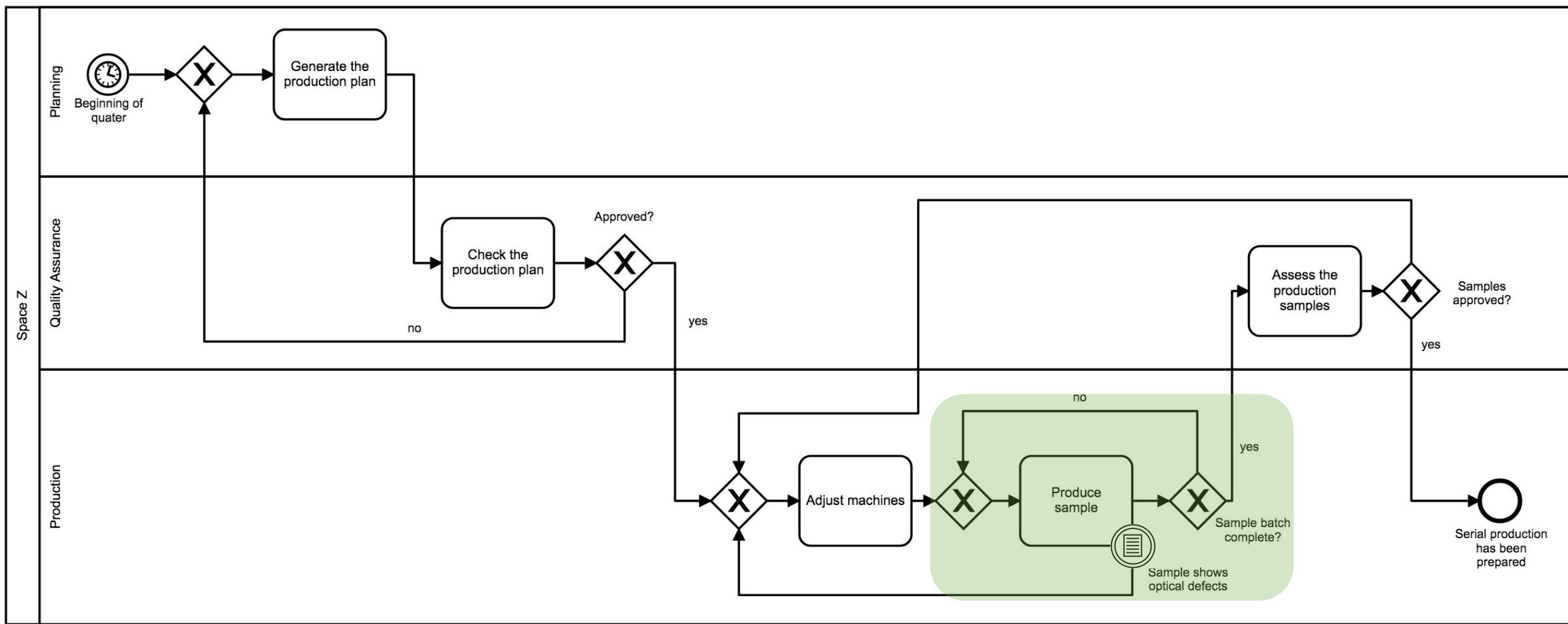






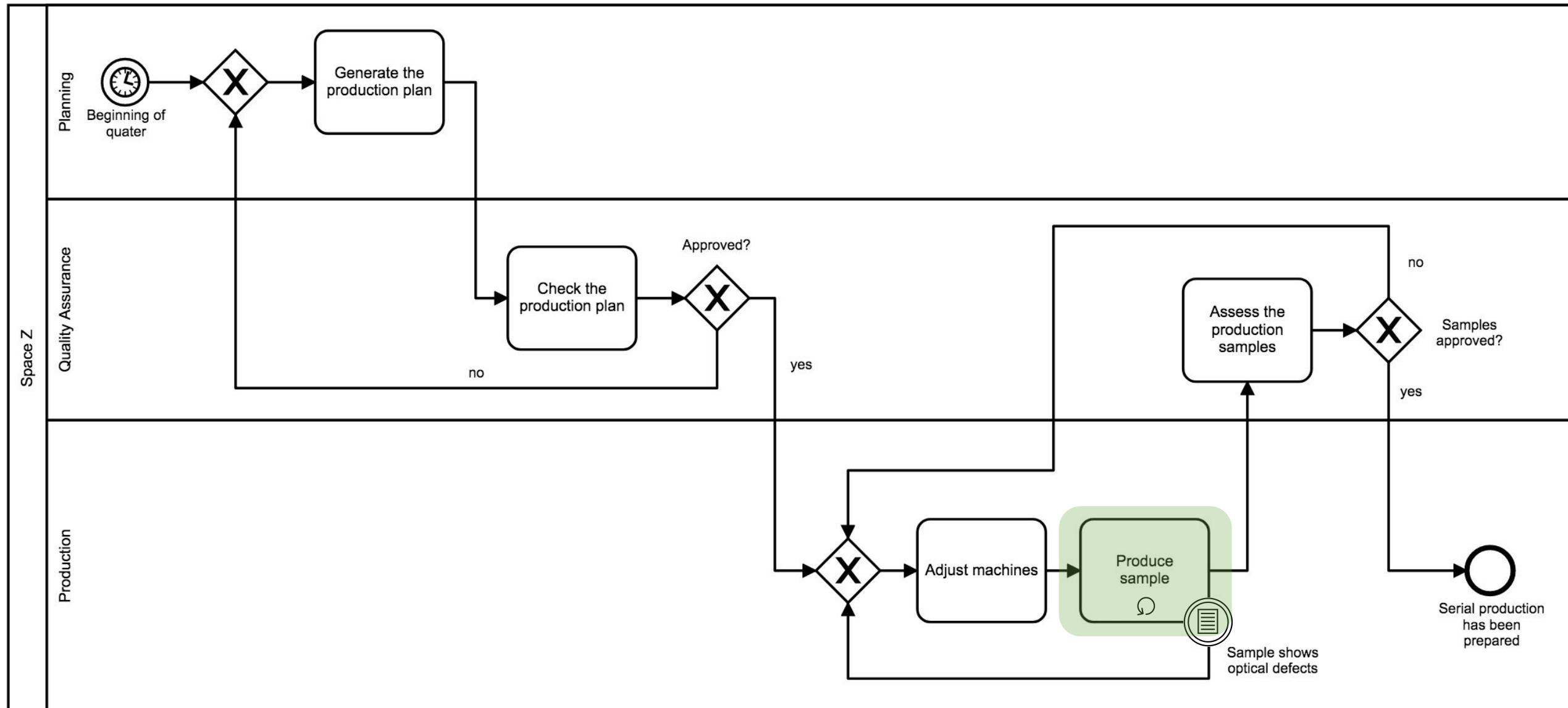


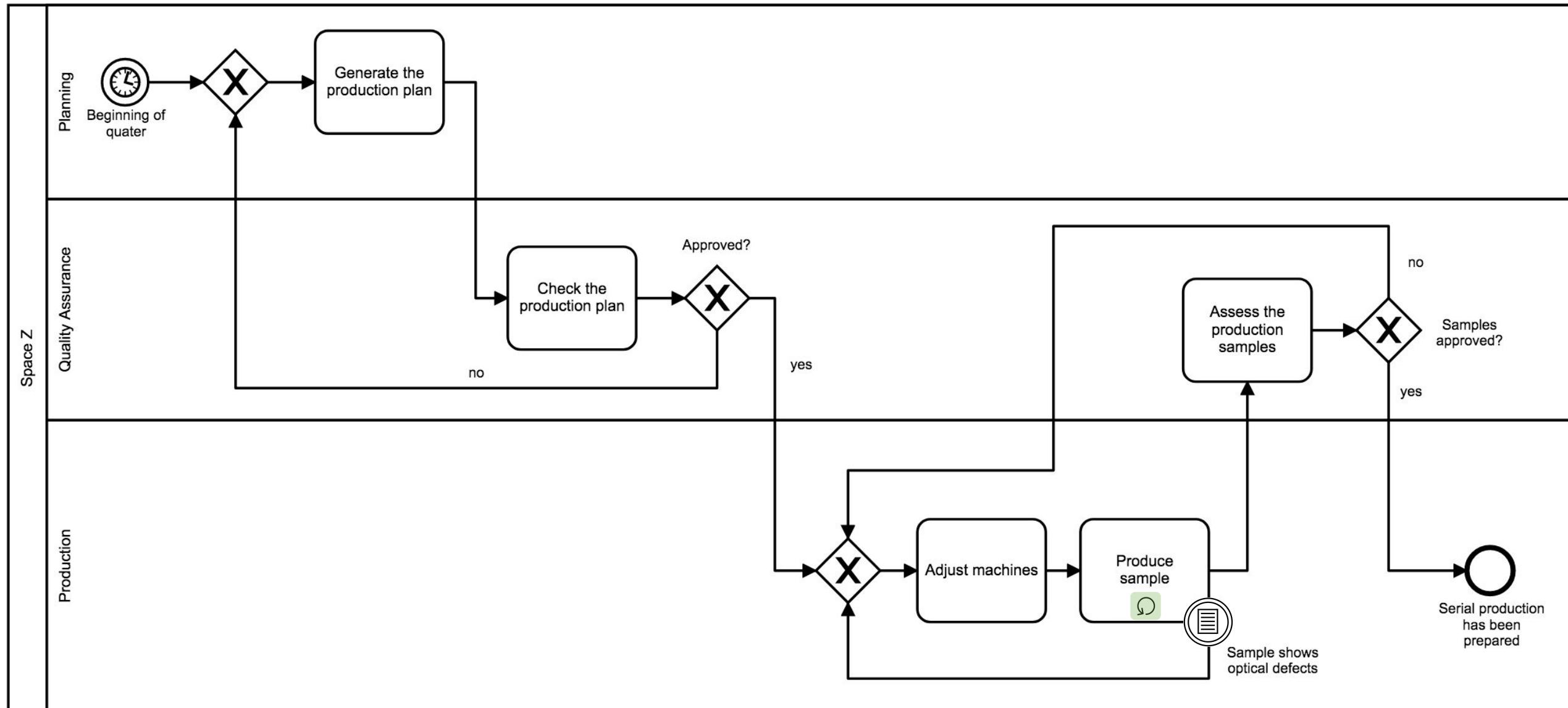


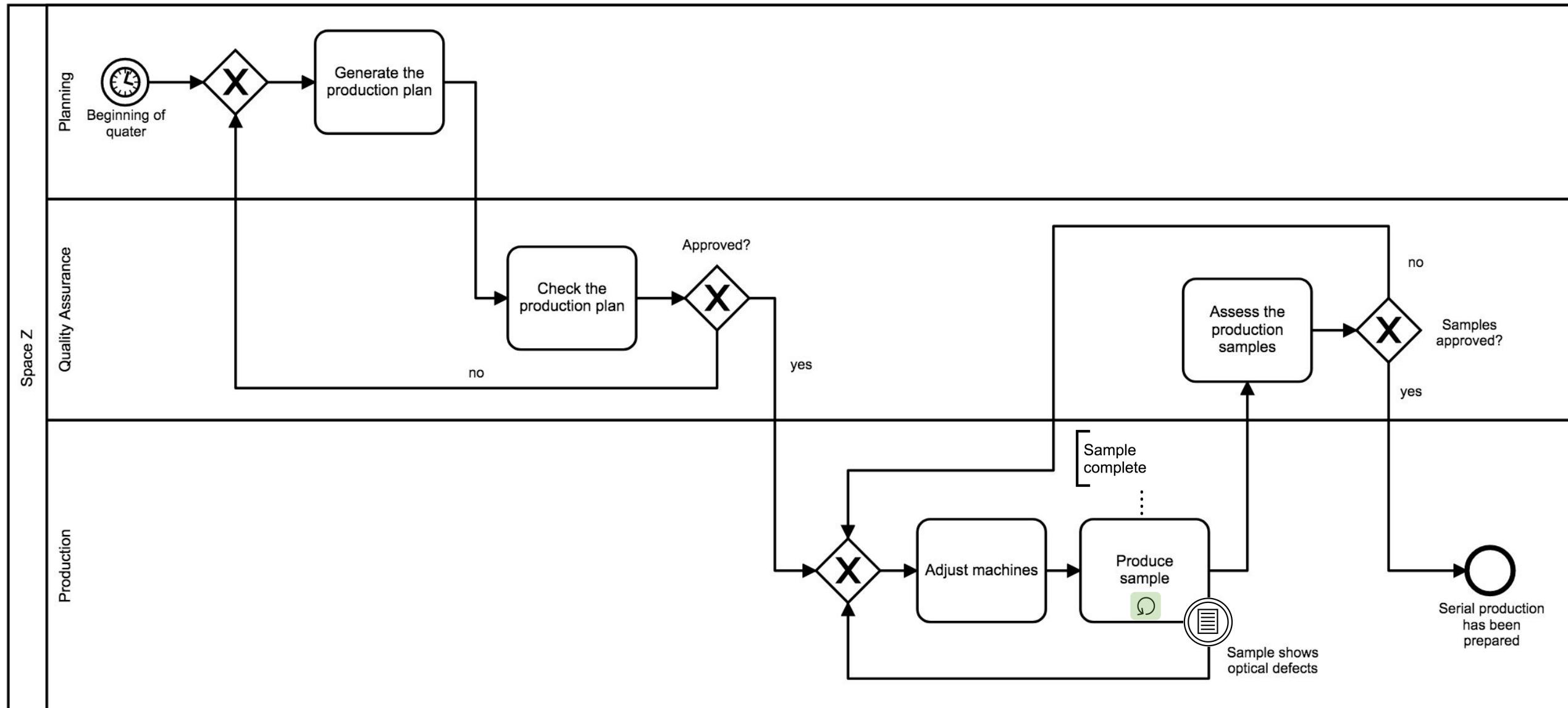


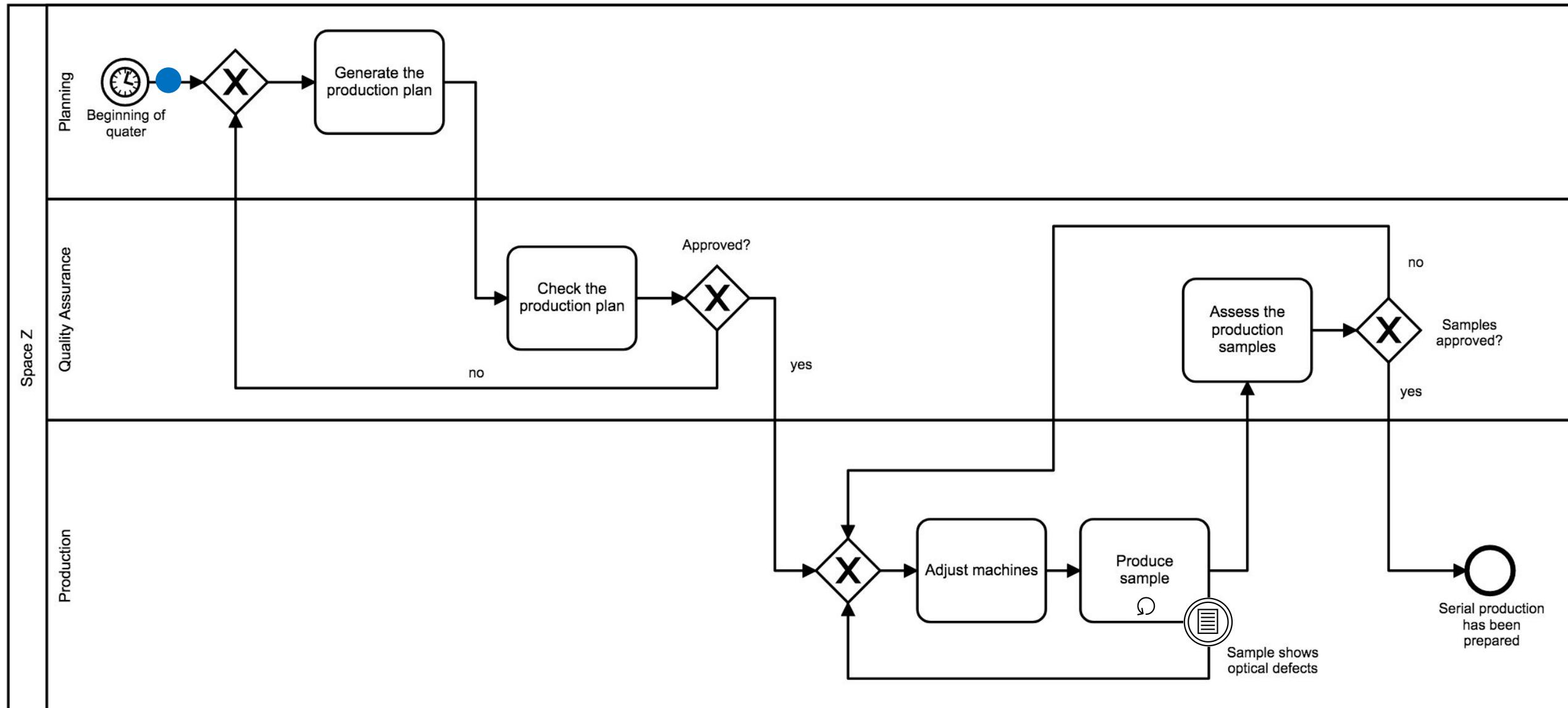


Let's use the **loop task!**

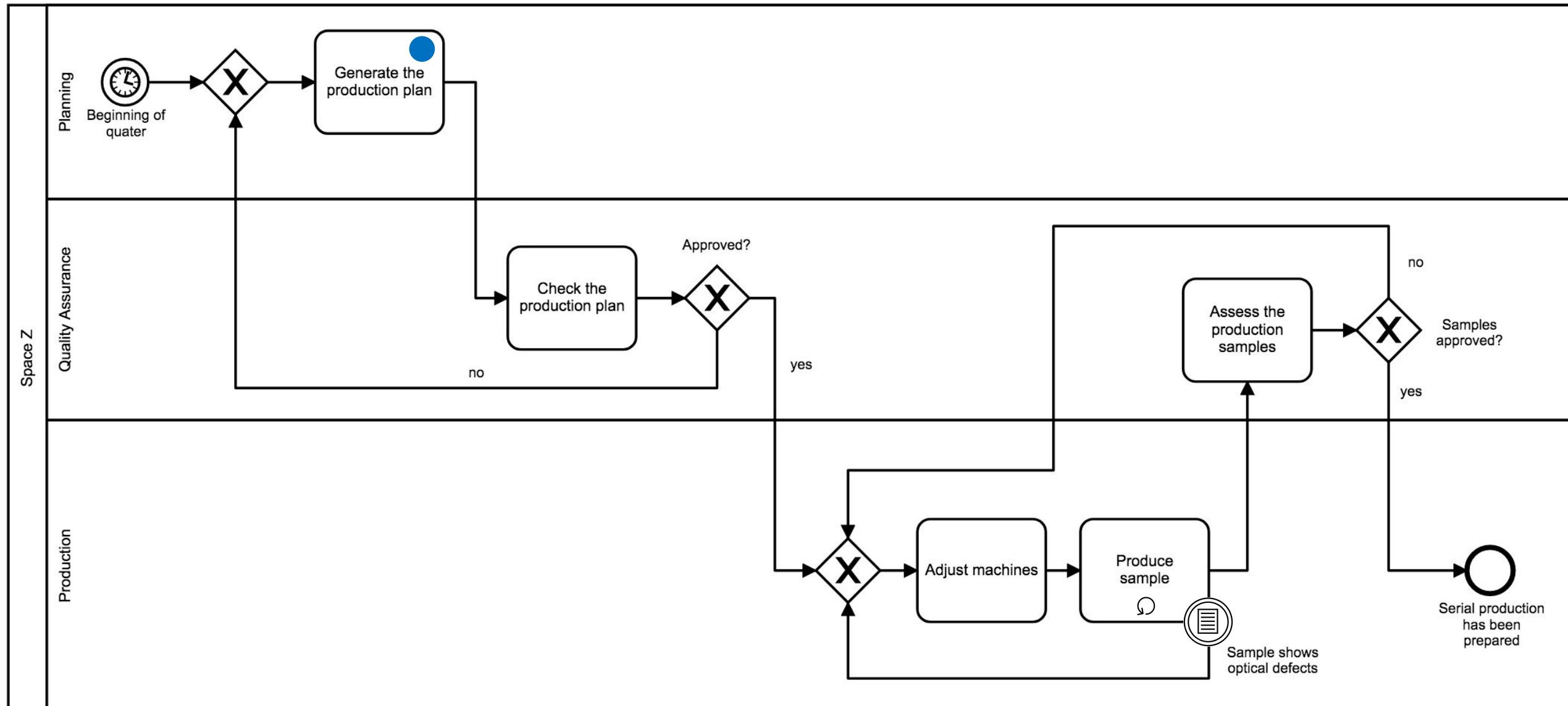


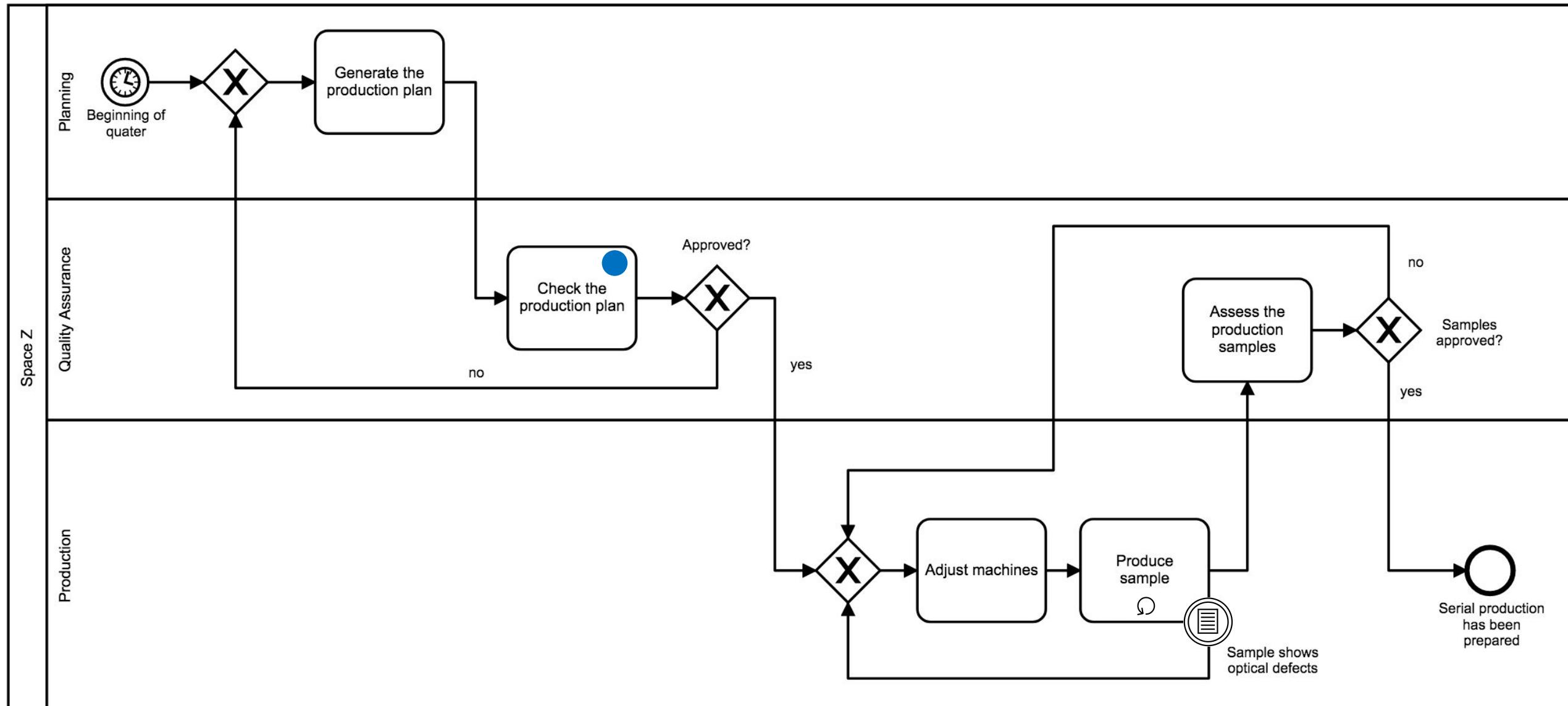


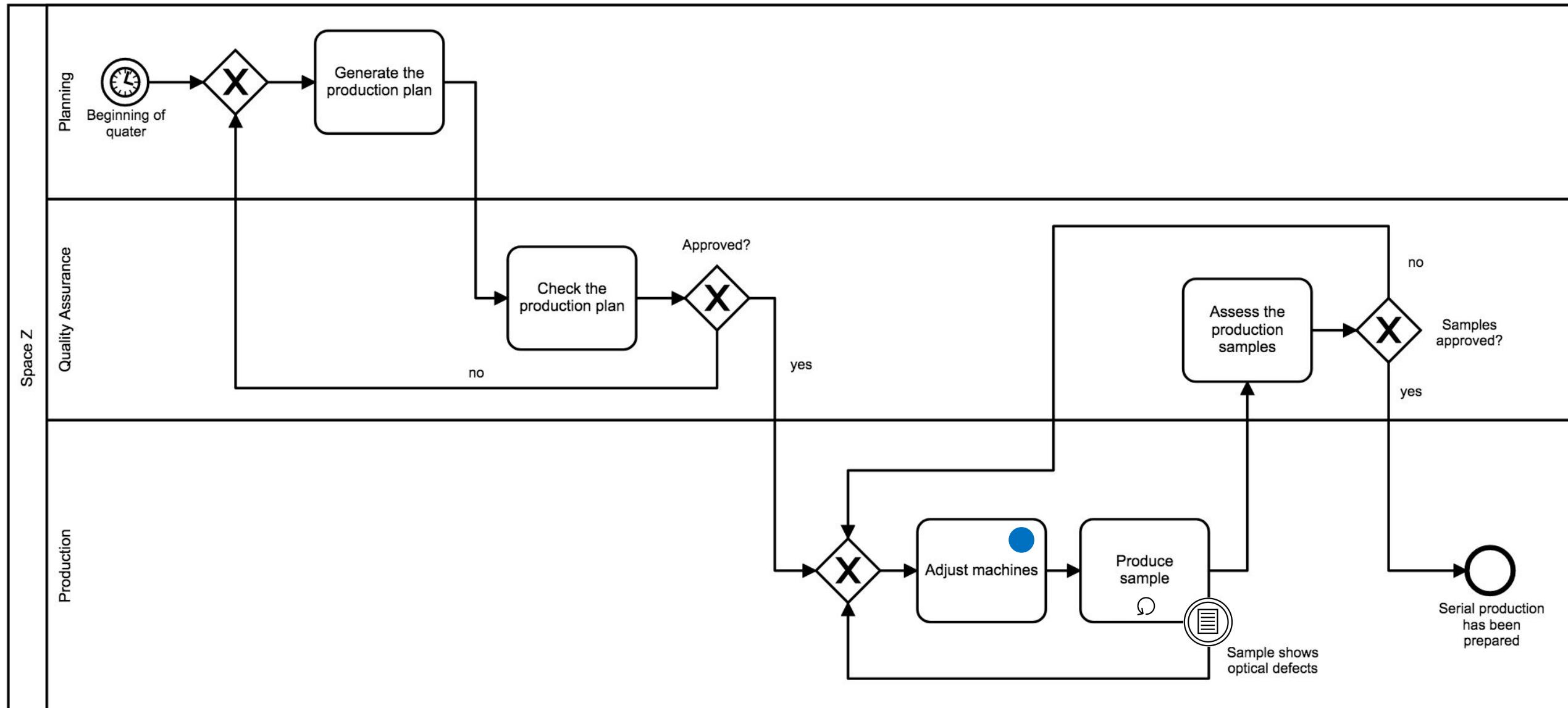


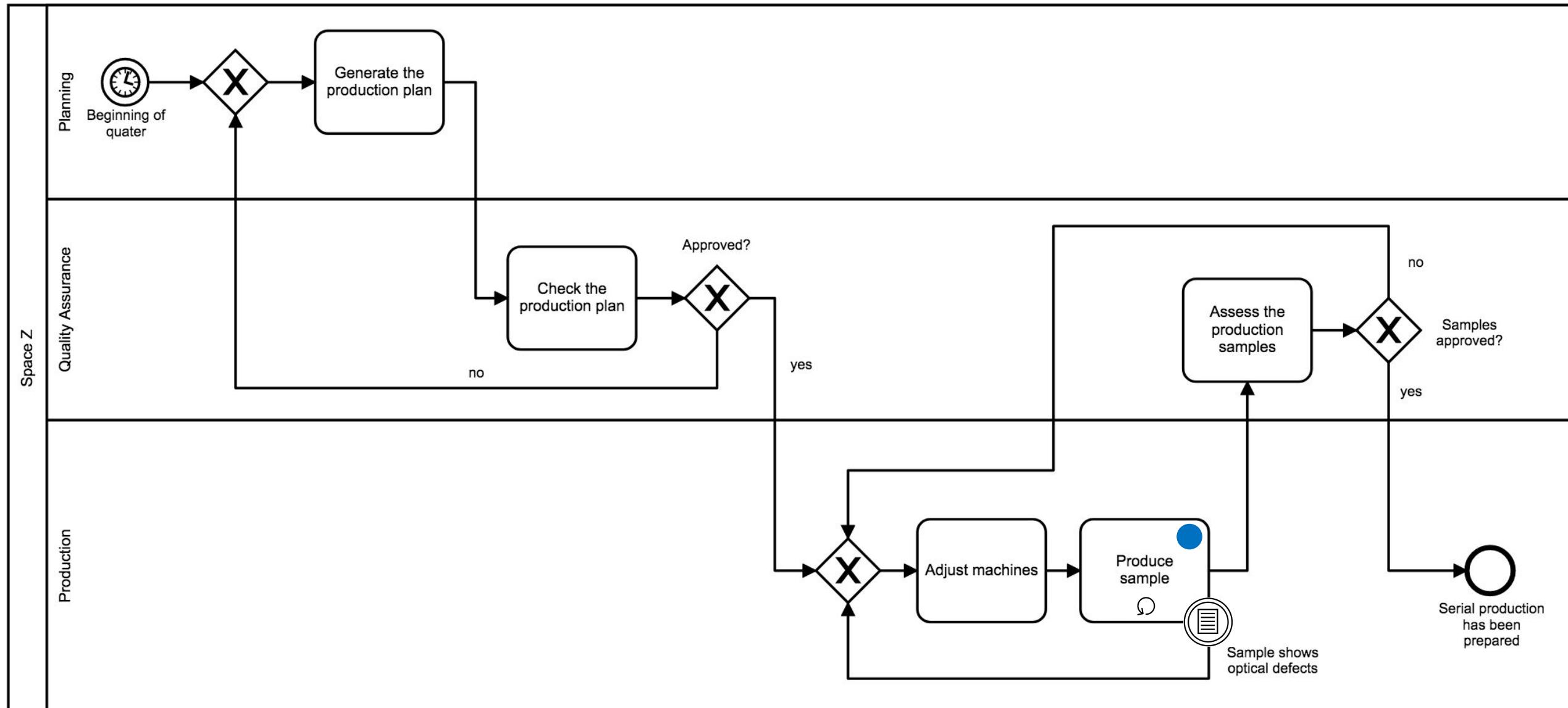


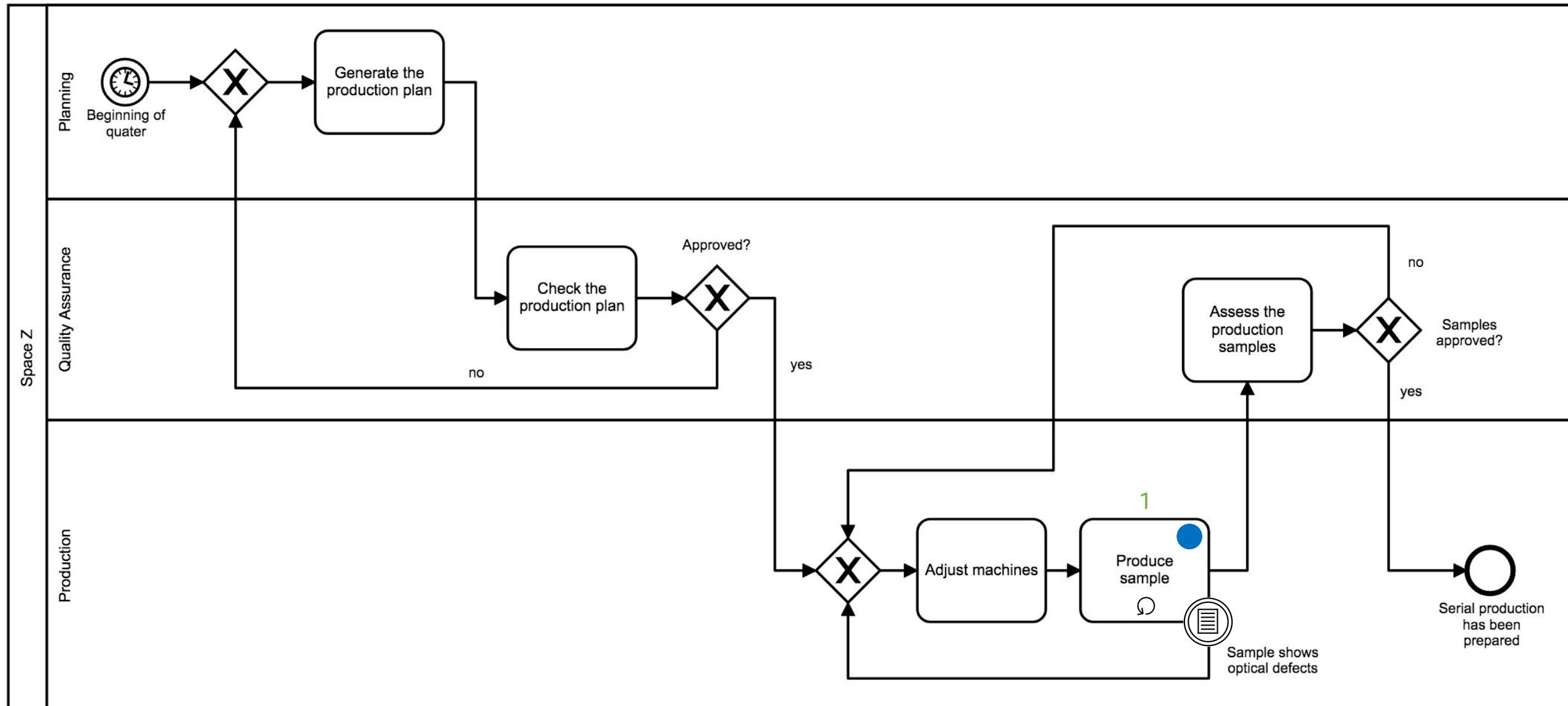
process camp

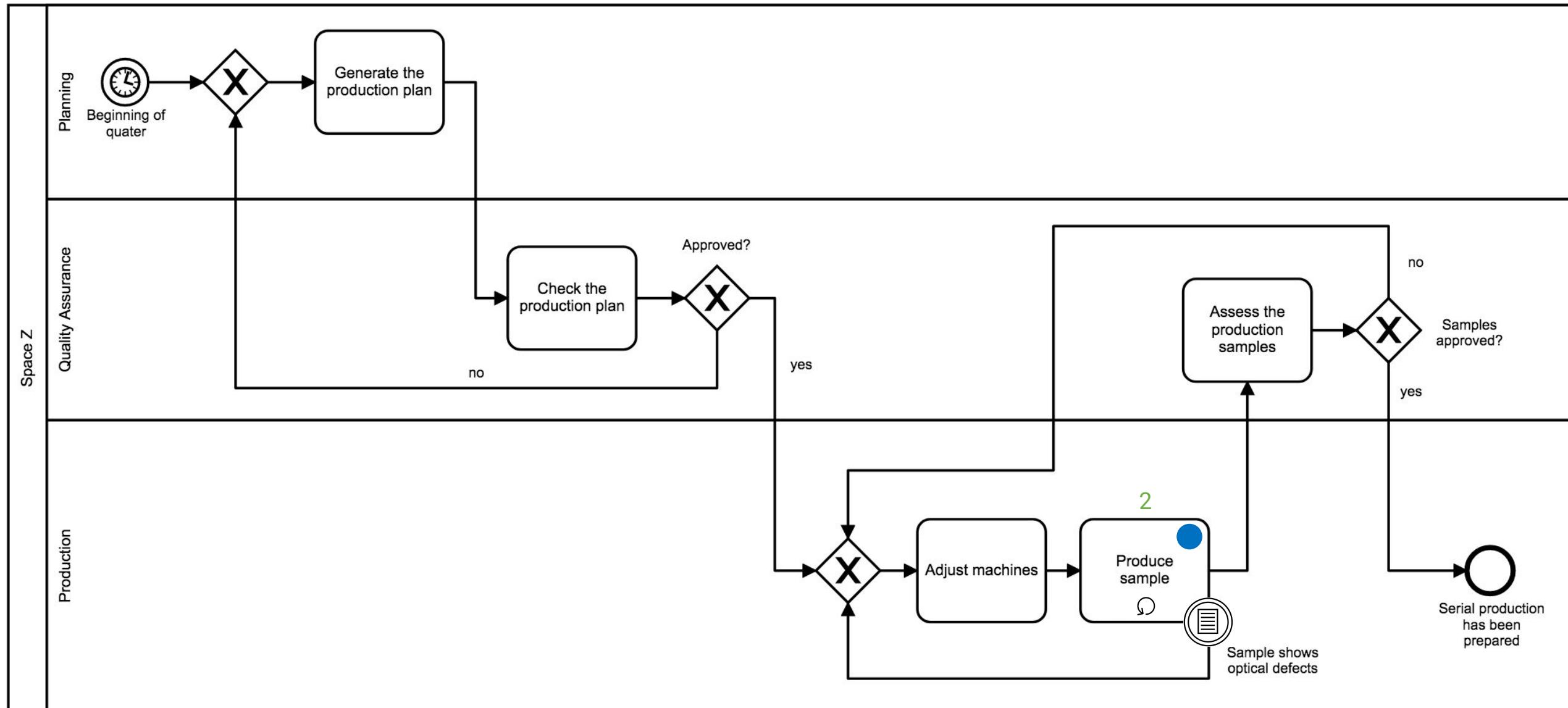


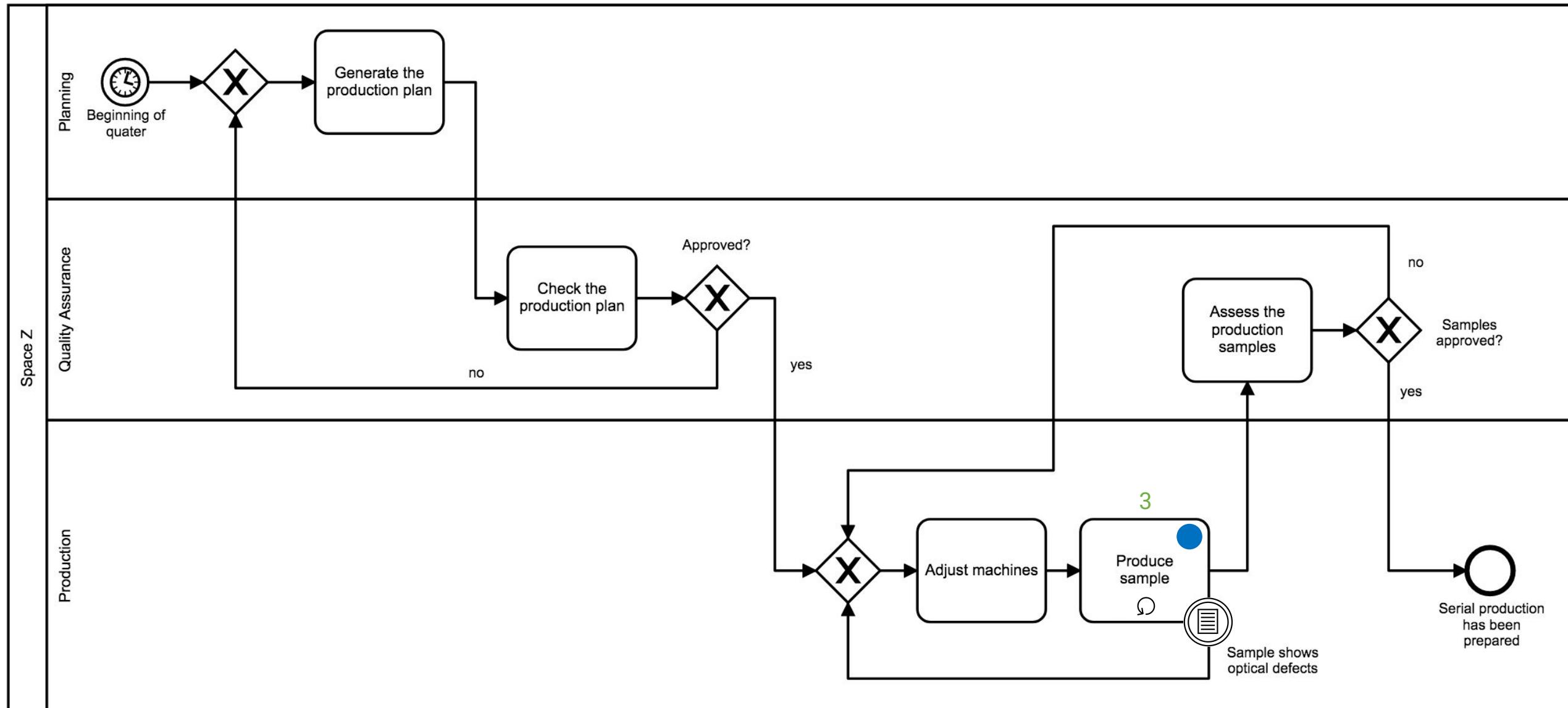


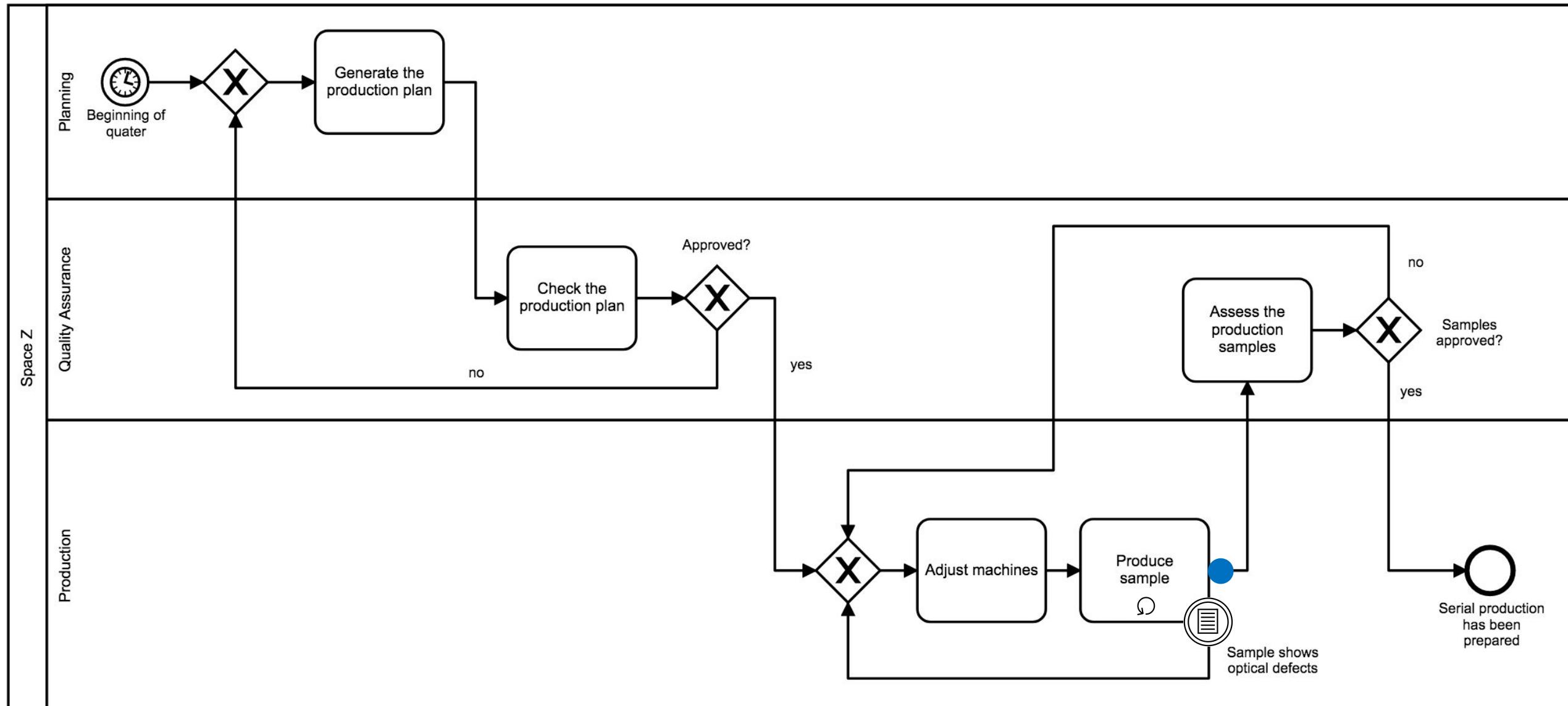












process camp



Sequential Multi Instance

- Defines an activity that is executed sequentially
- The execution ends with a defined condition



It's **payday** at the bookworm store!

The money transfer is **sequential**



The money transfer is **sequential**



The money transfer is **sequential**

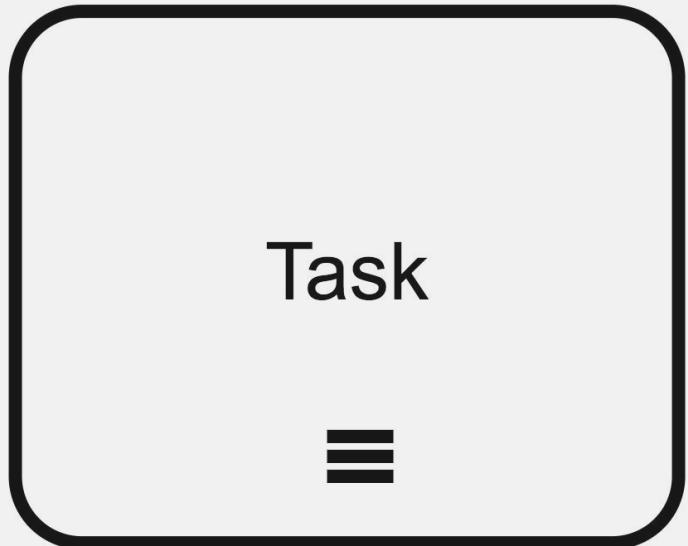


The money transfer is **sequential**

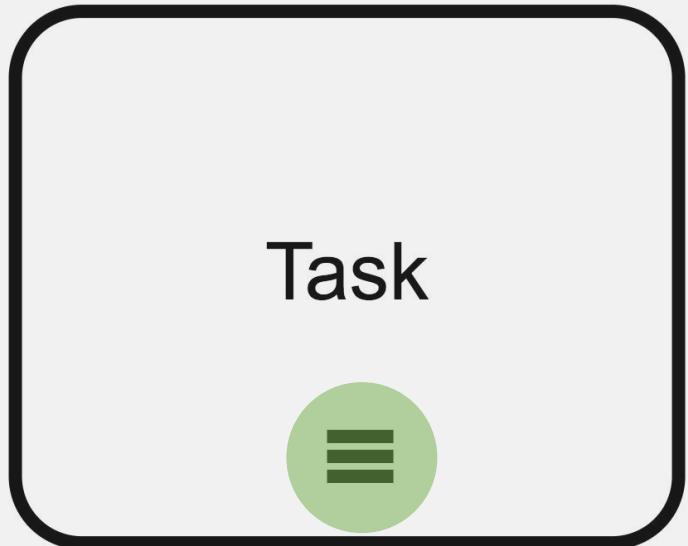
BPMN Theory

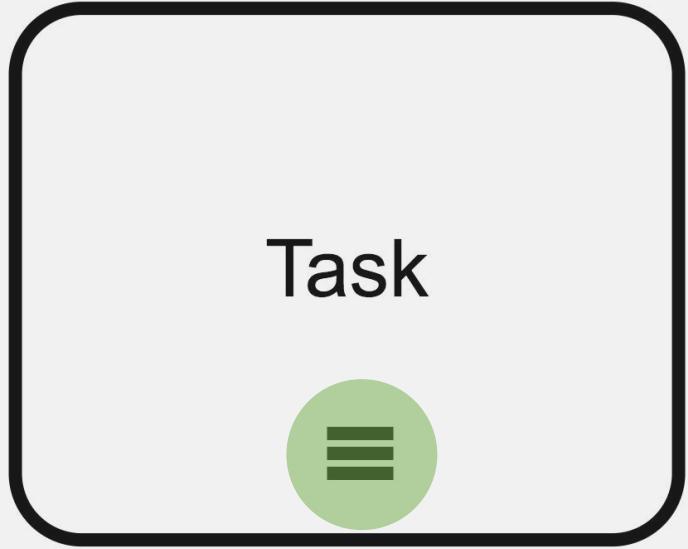
process camp

Sequential Multi Instance

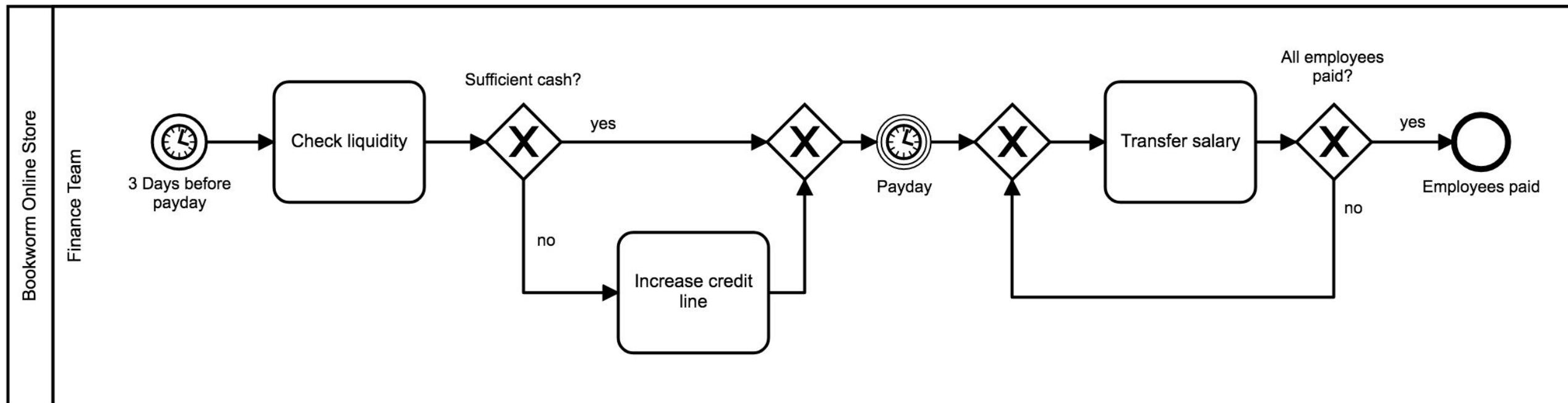


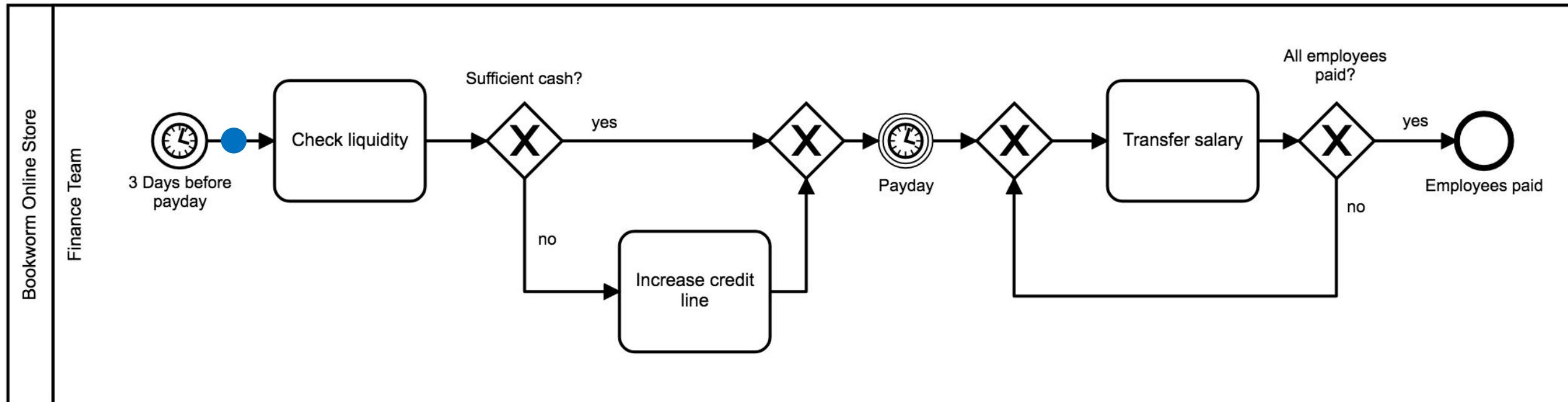
Sequential Multi Instance

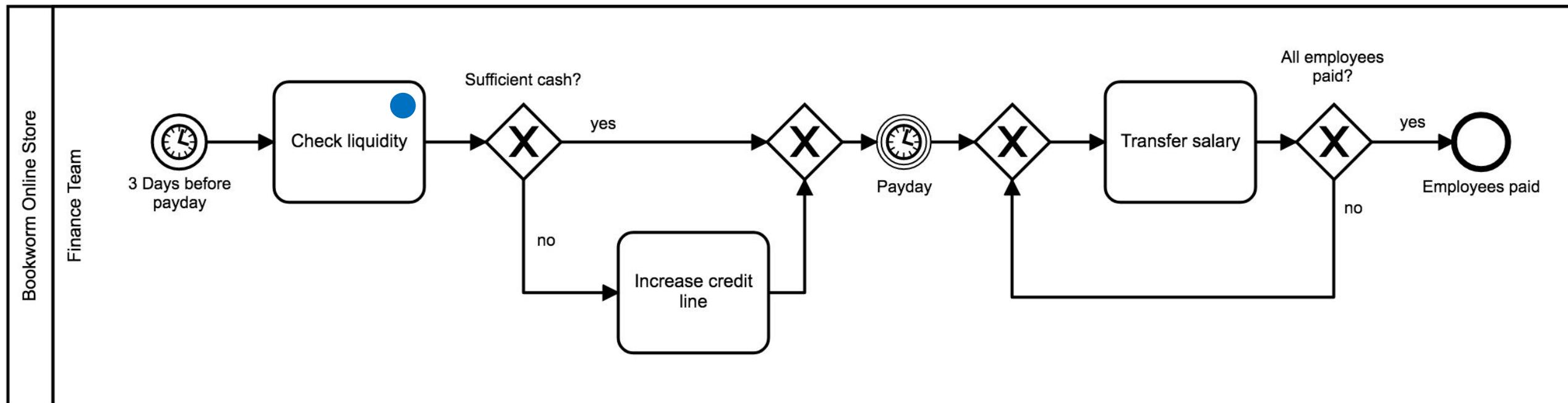


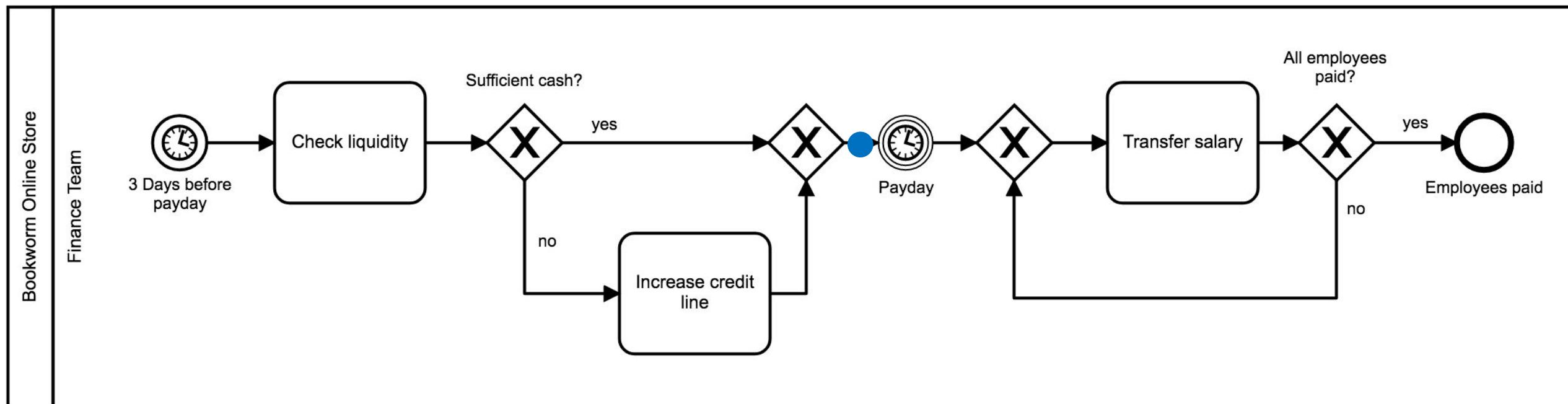


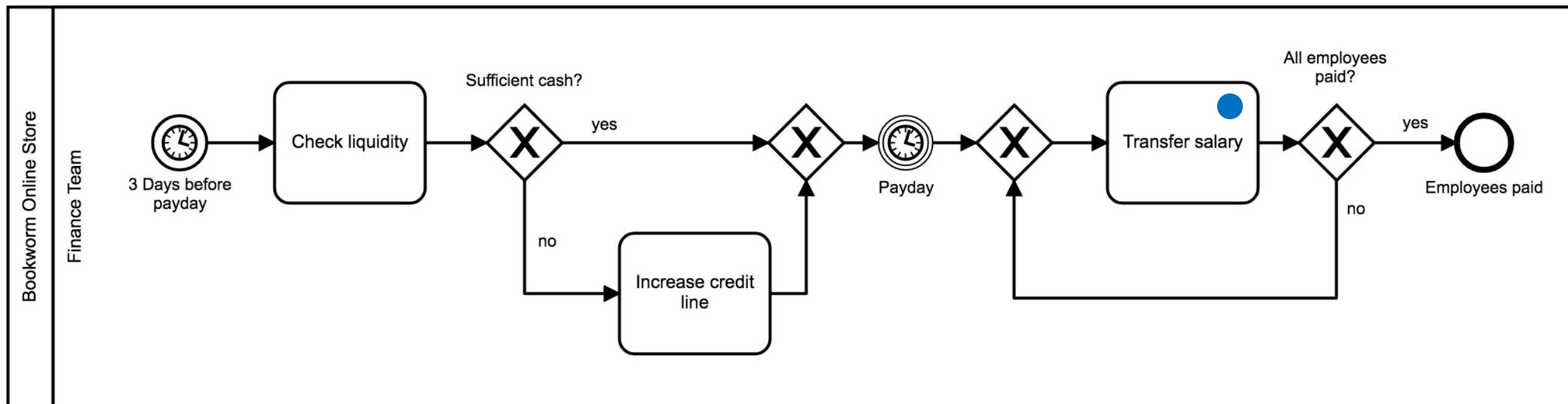
Note that a sequential task can be executed more (or less) than 3 times

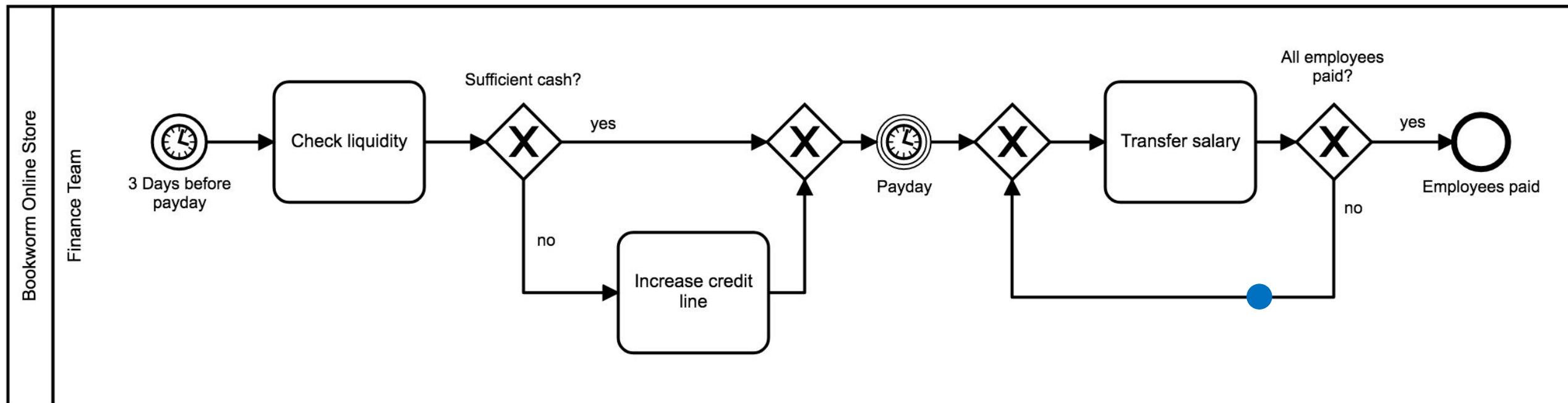


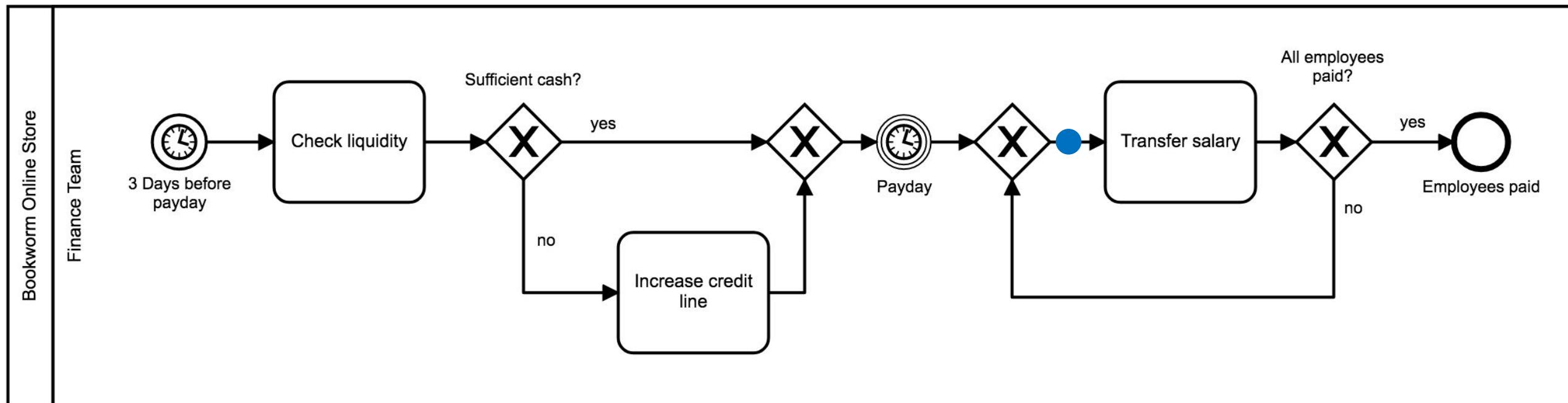


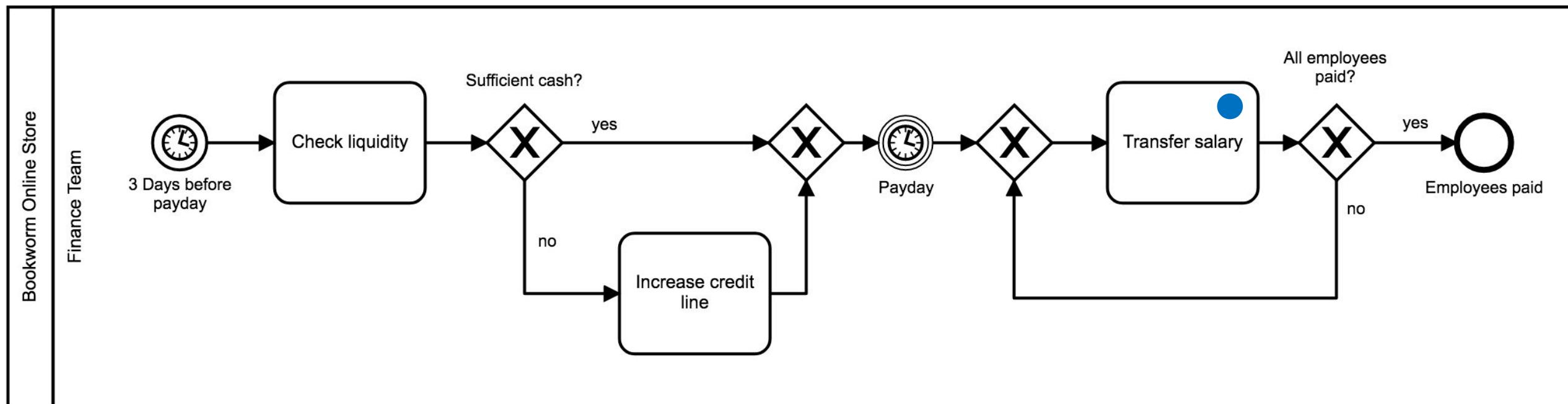


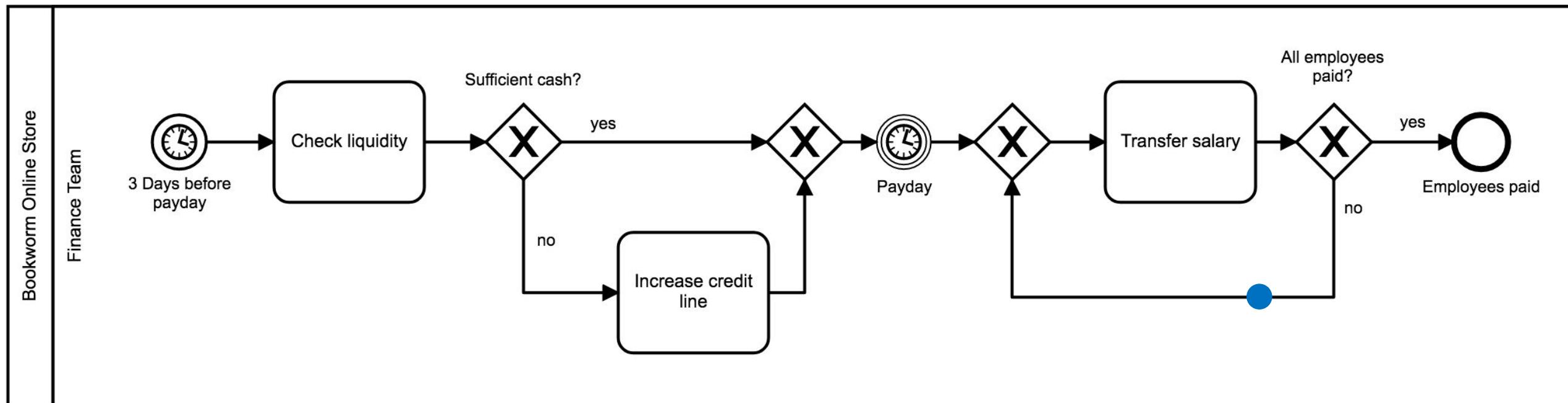


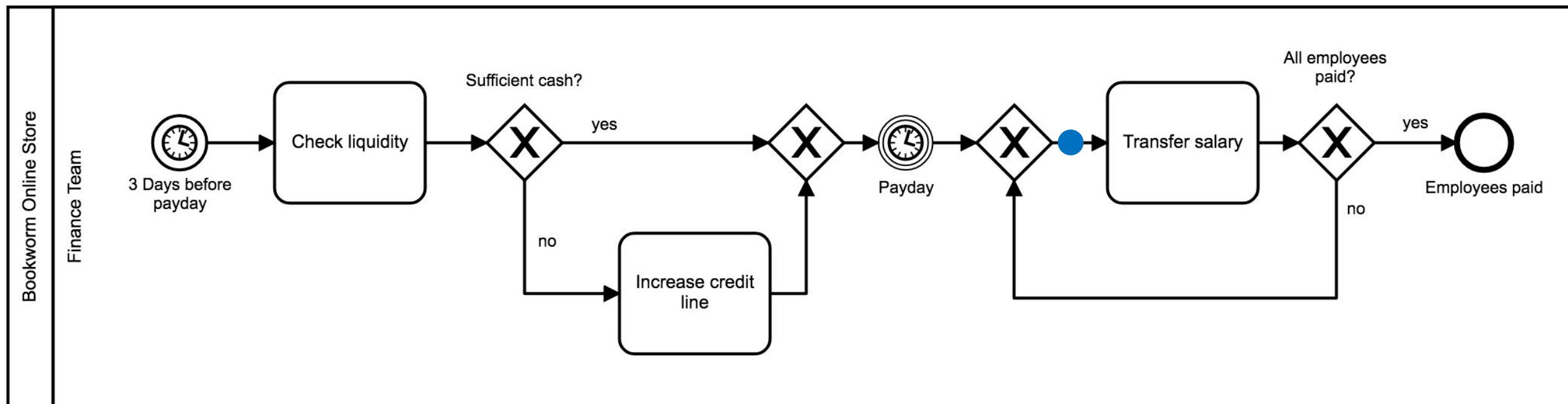


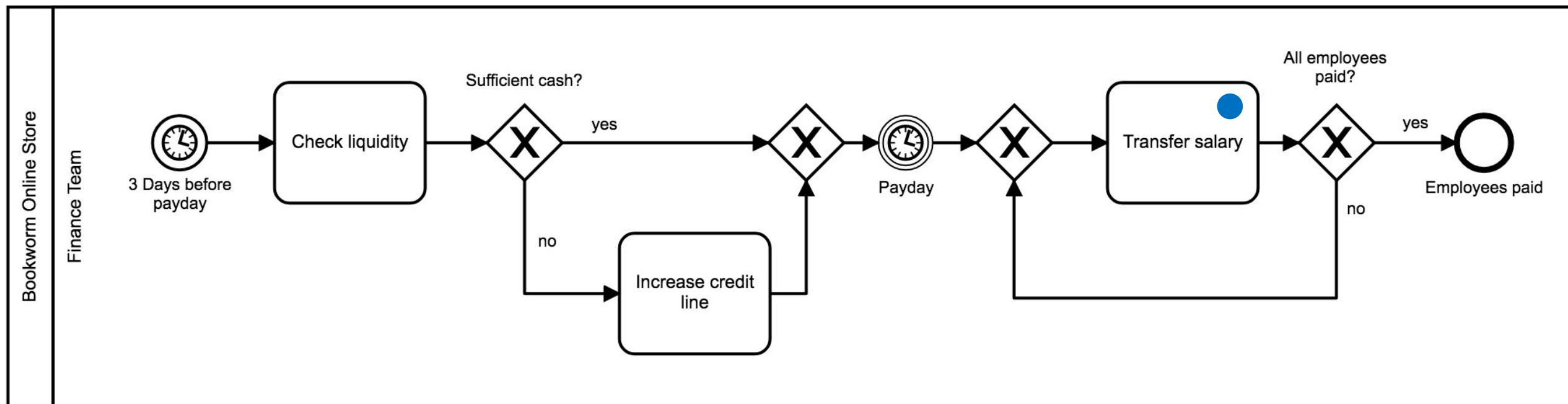


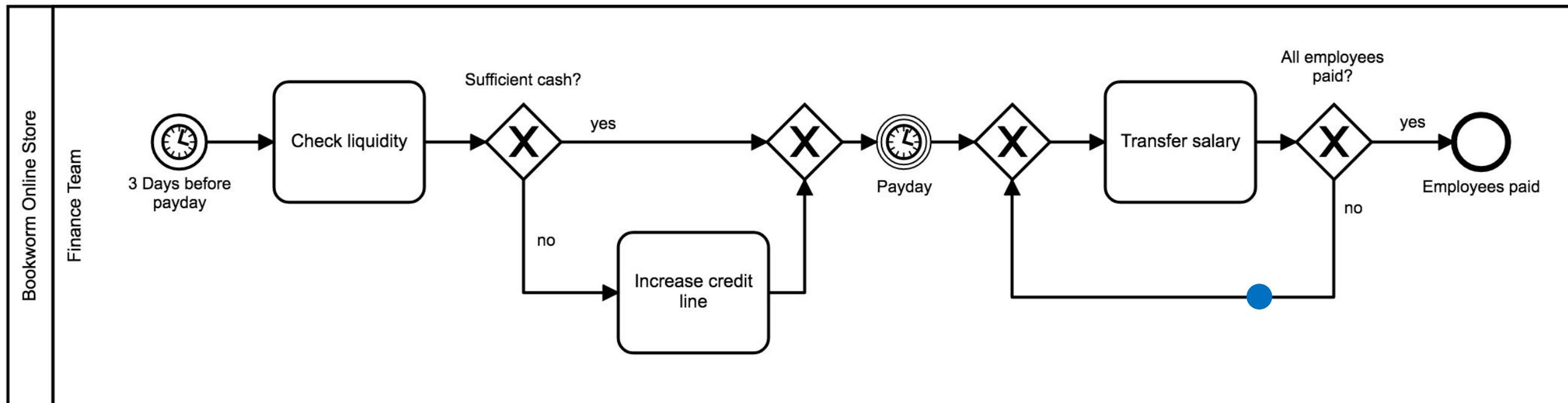


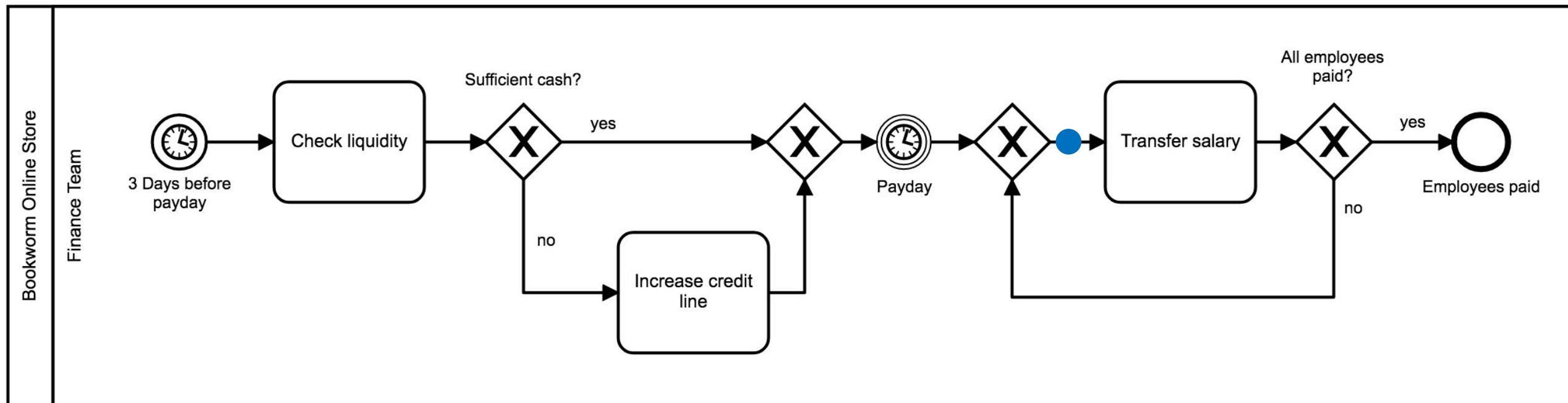


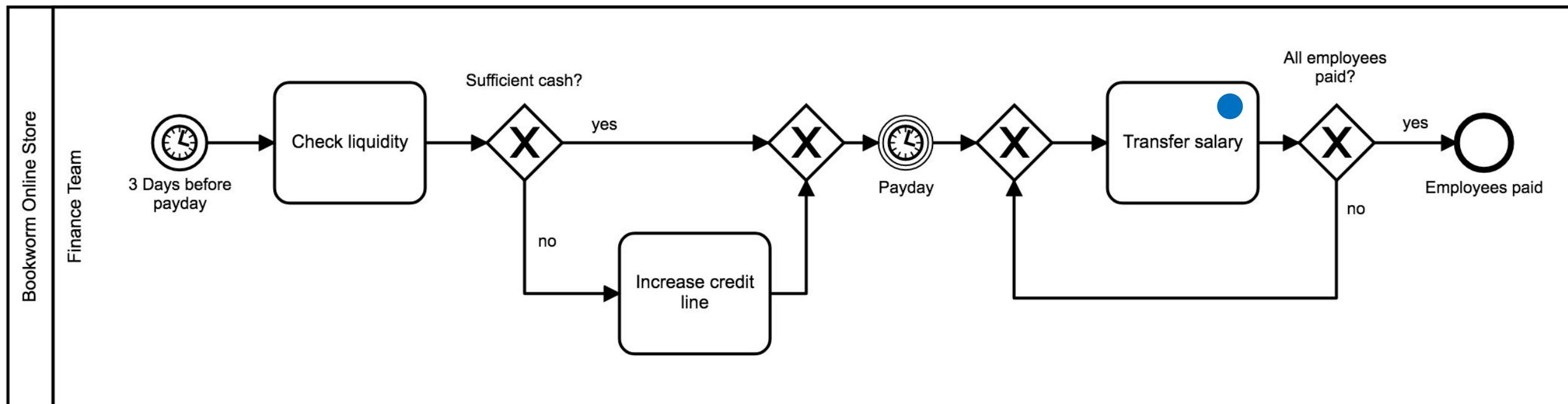


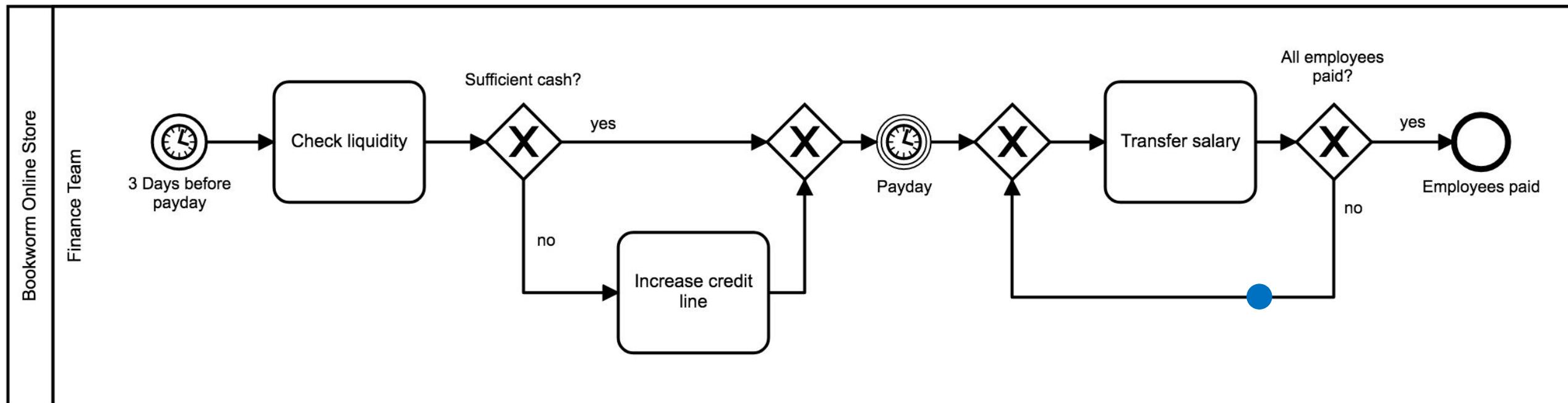


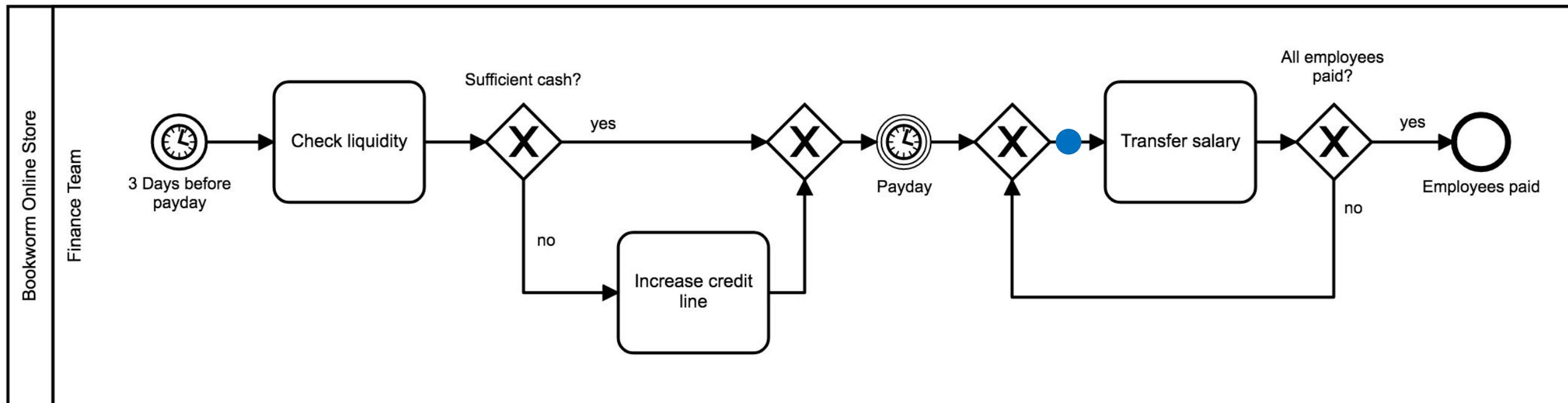


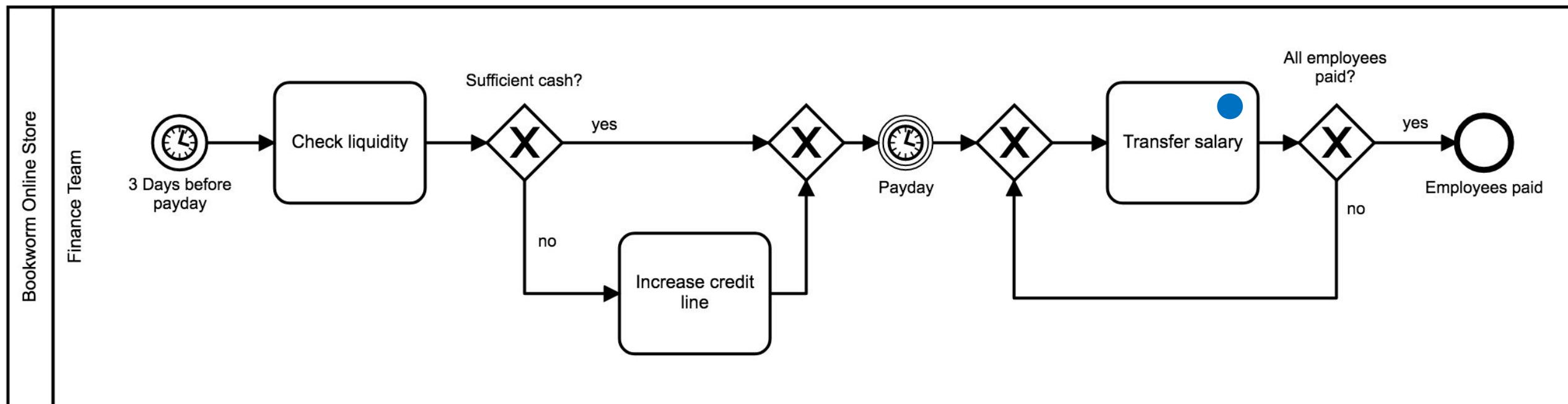


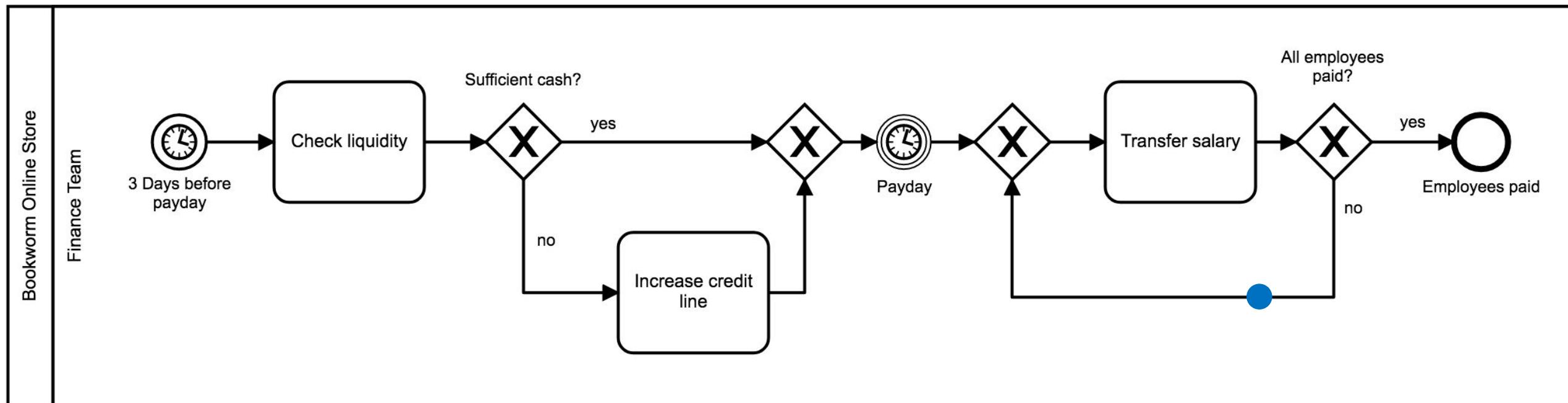


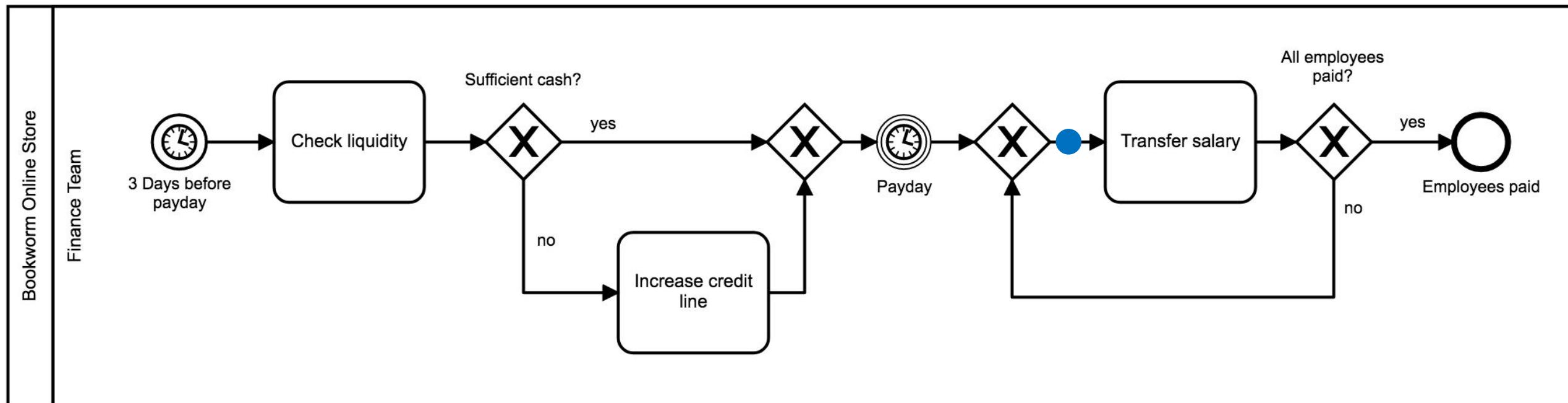


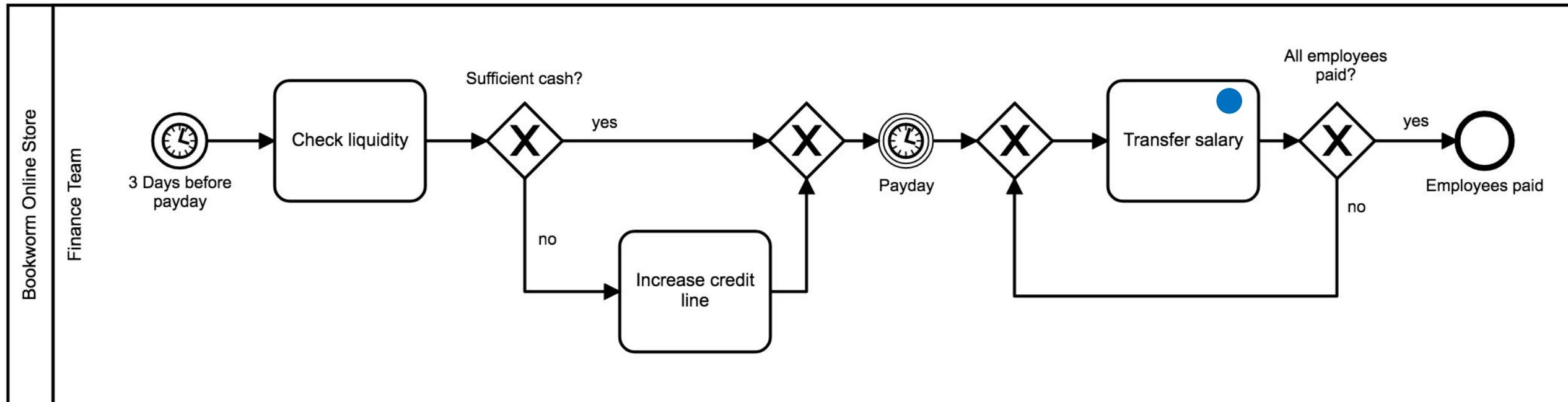


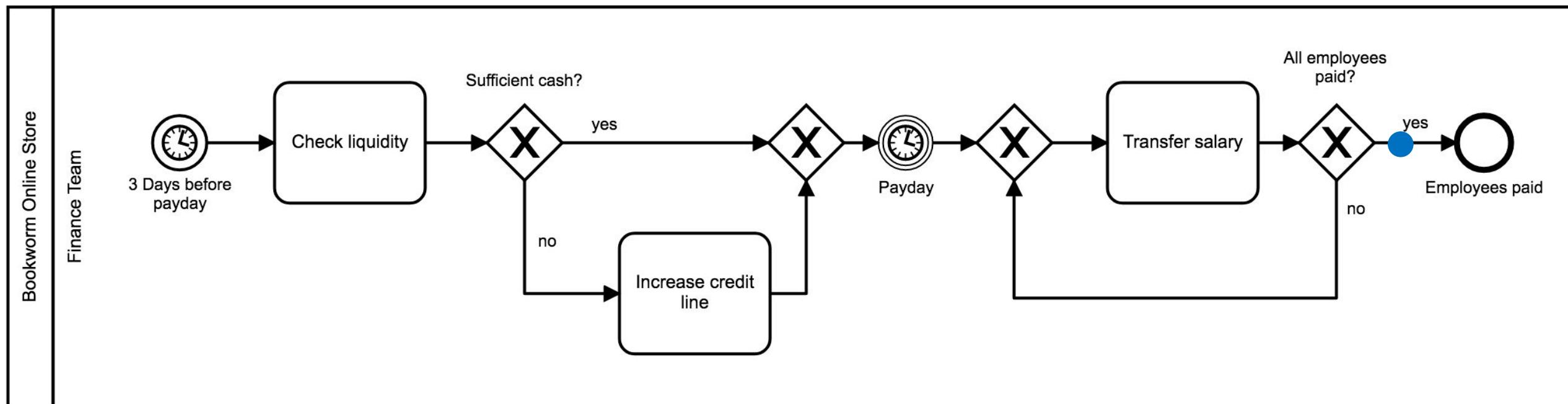


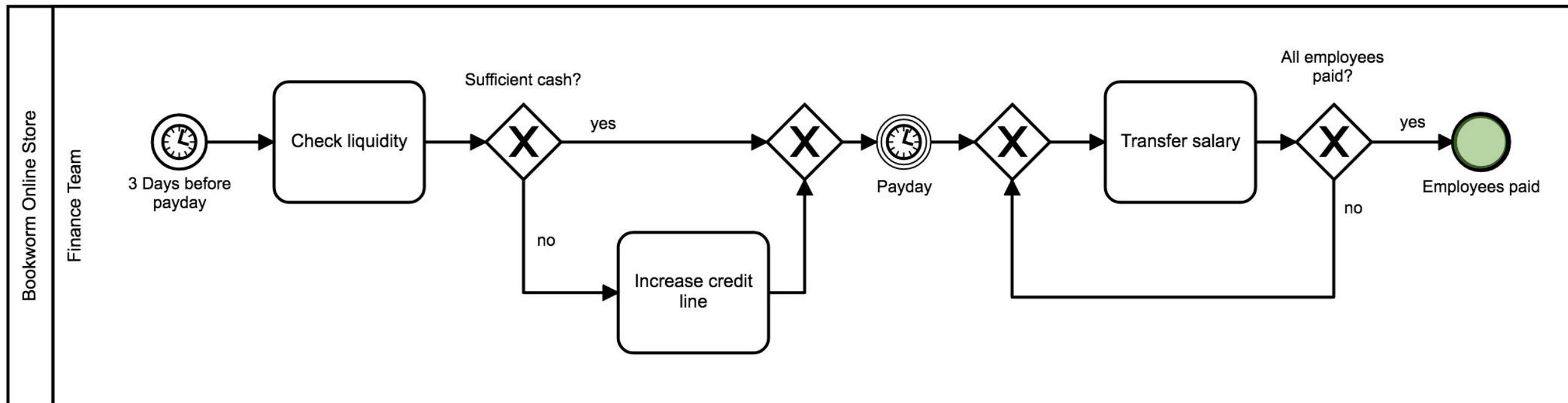


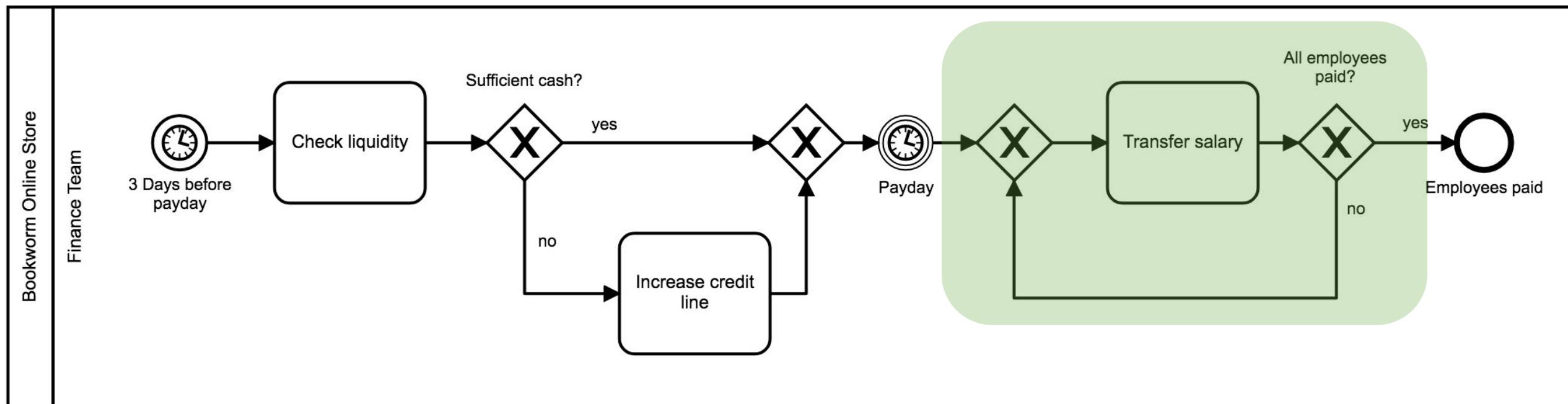


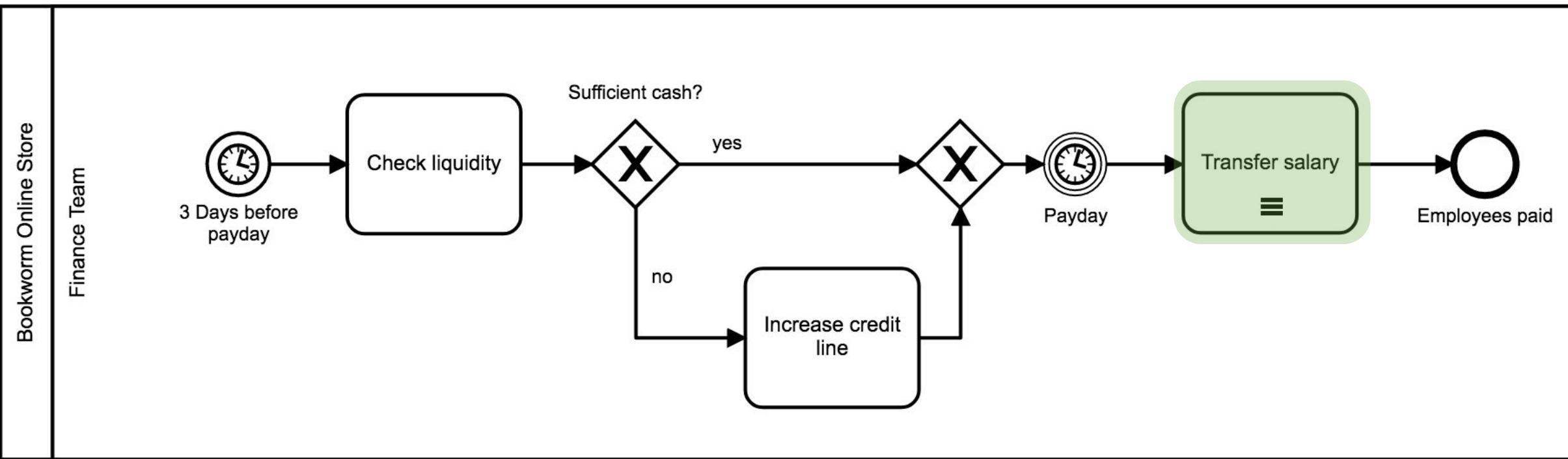


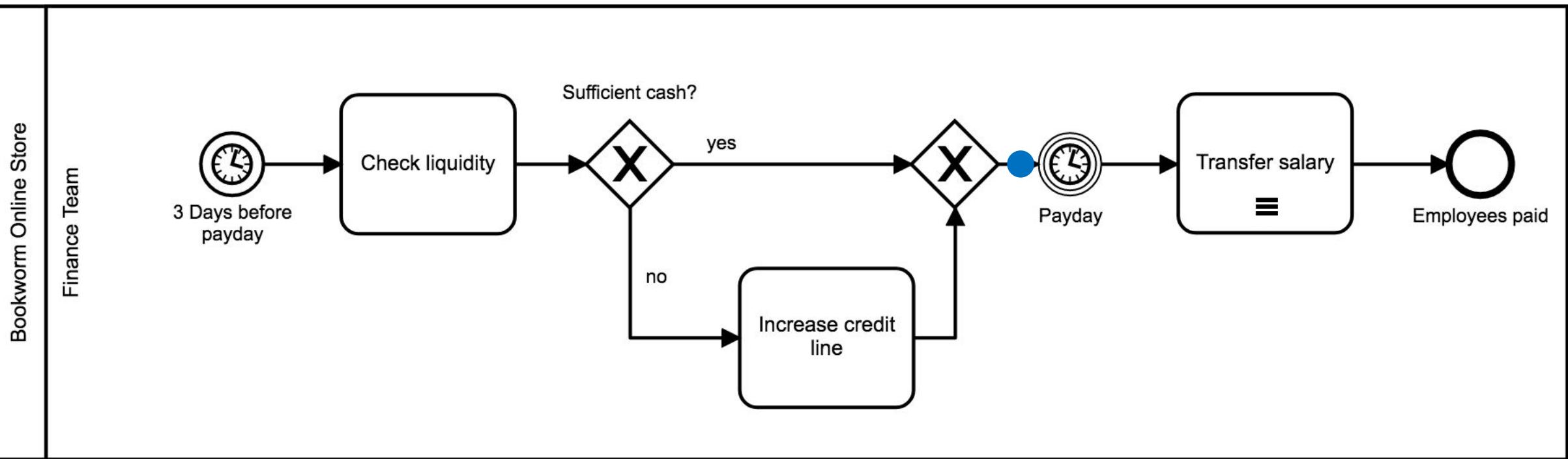


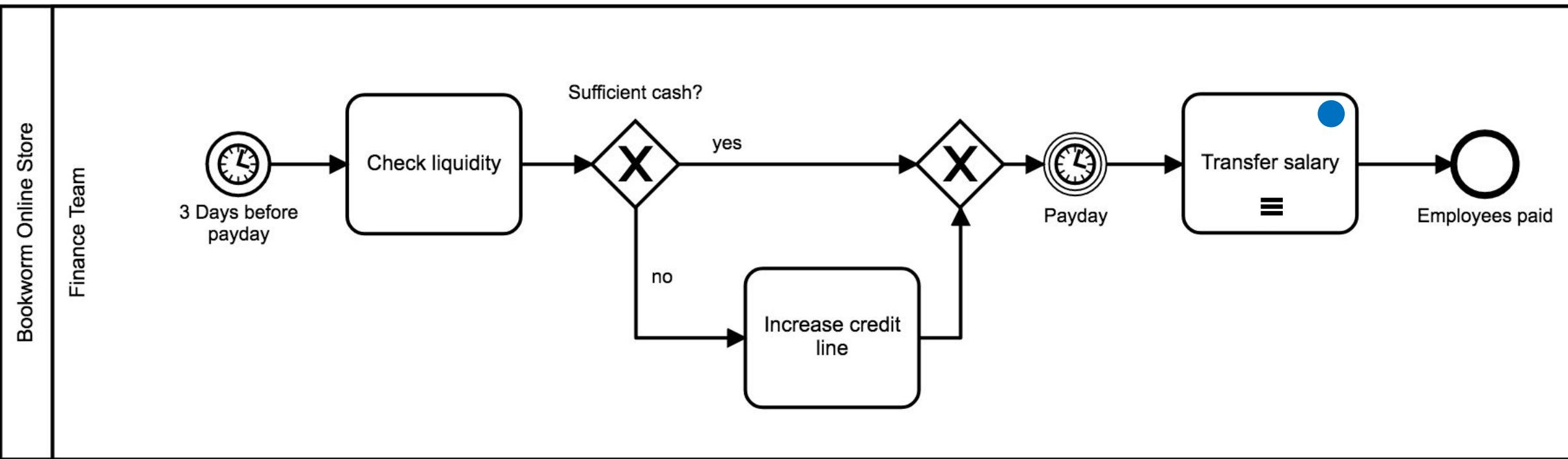


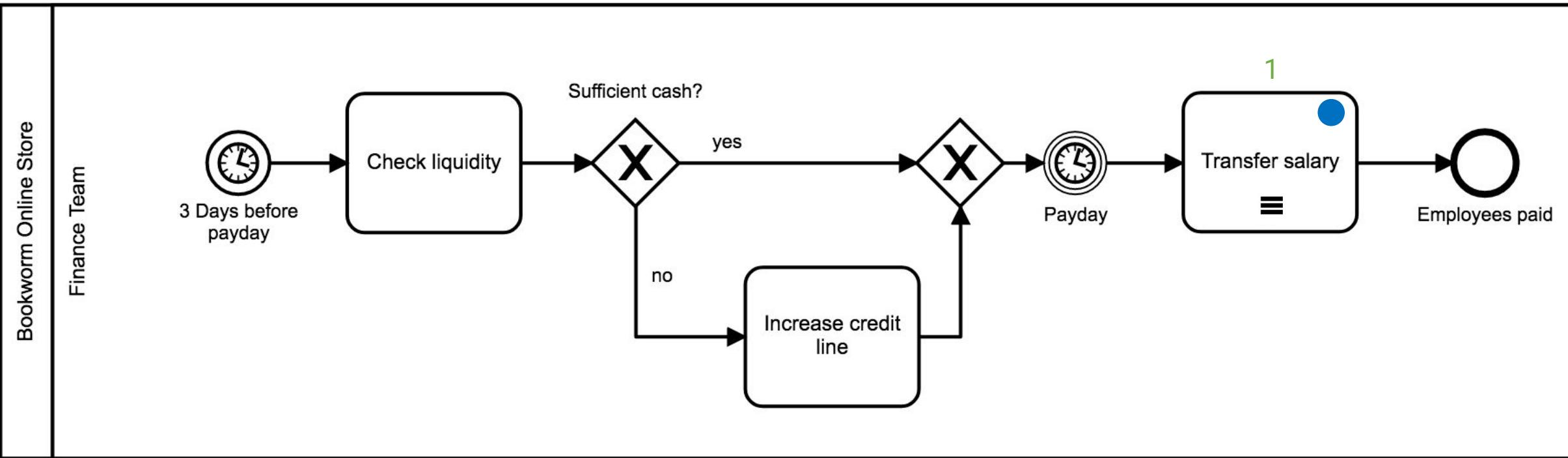


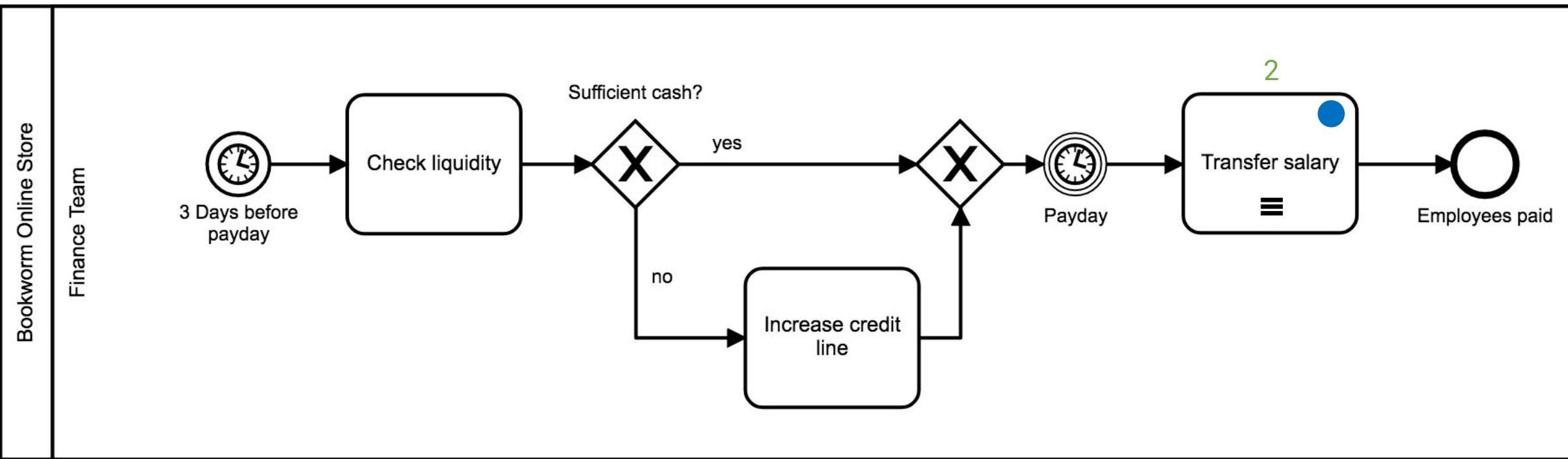


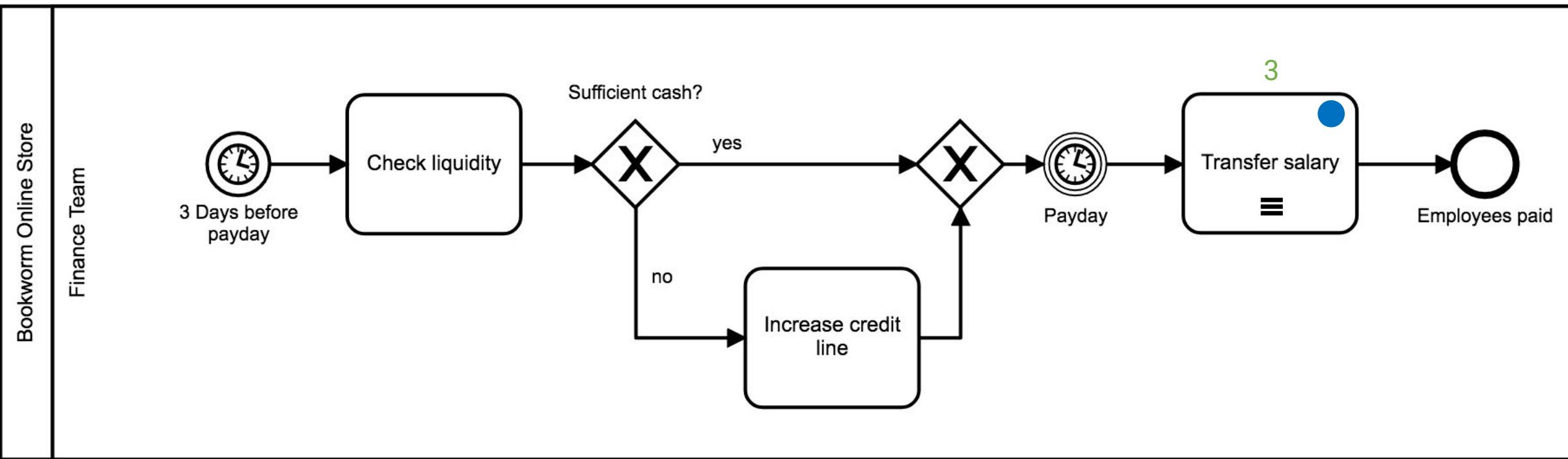


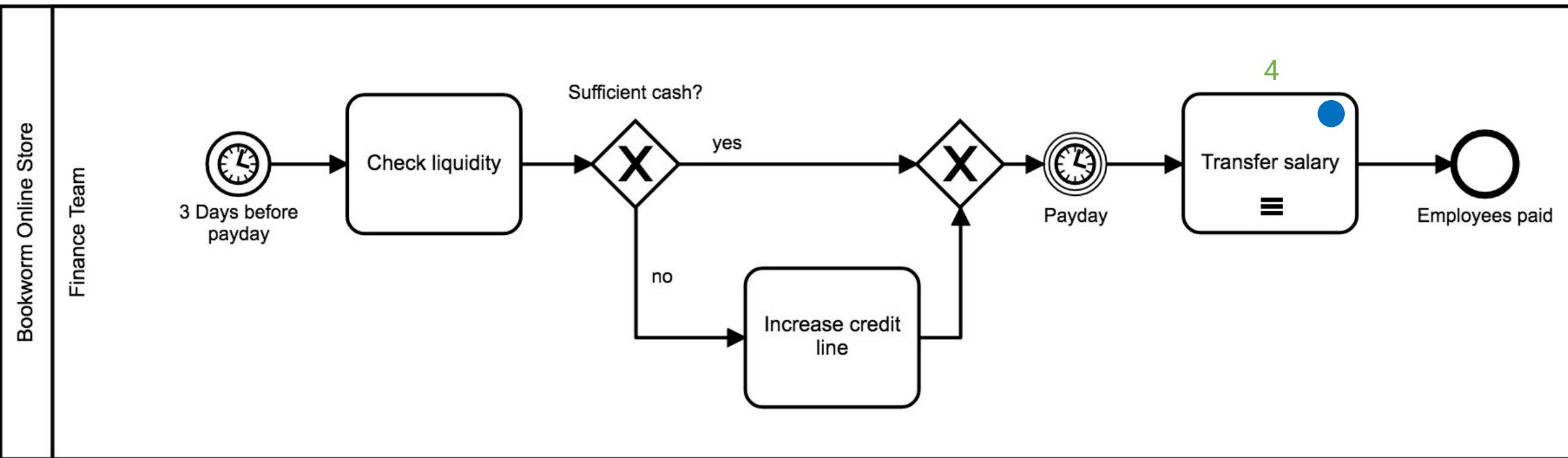


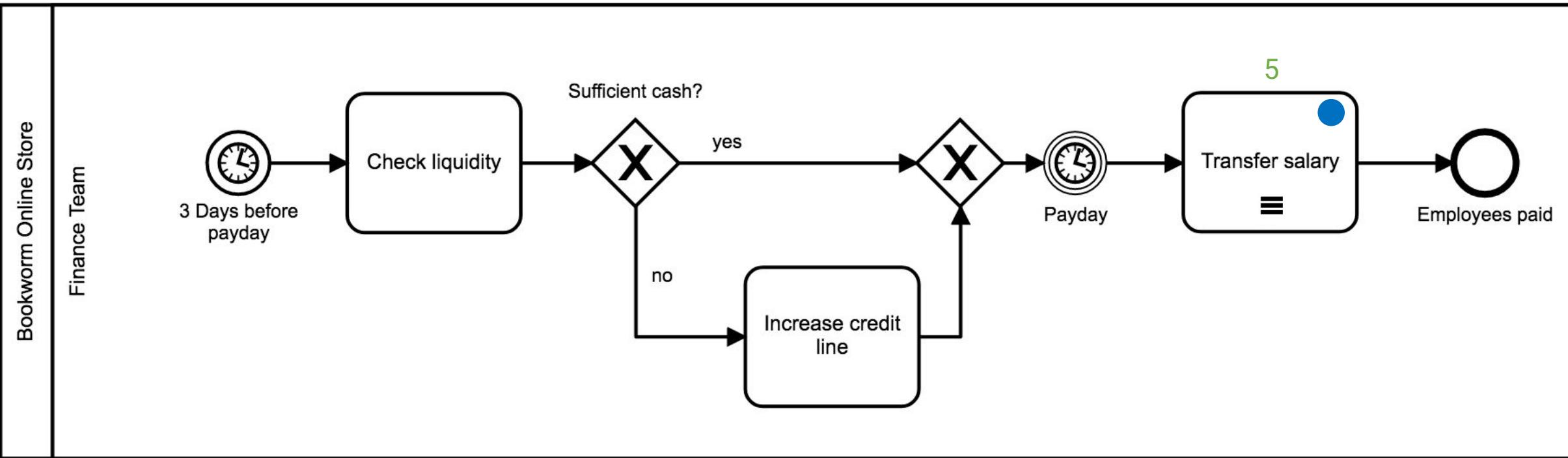


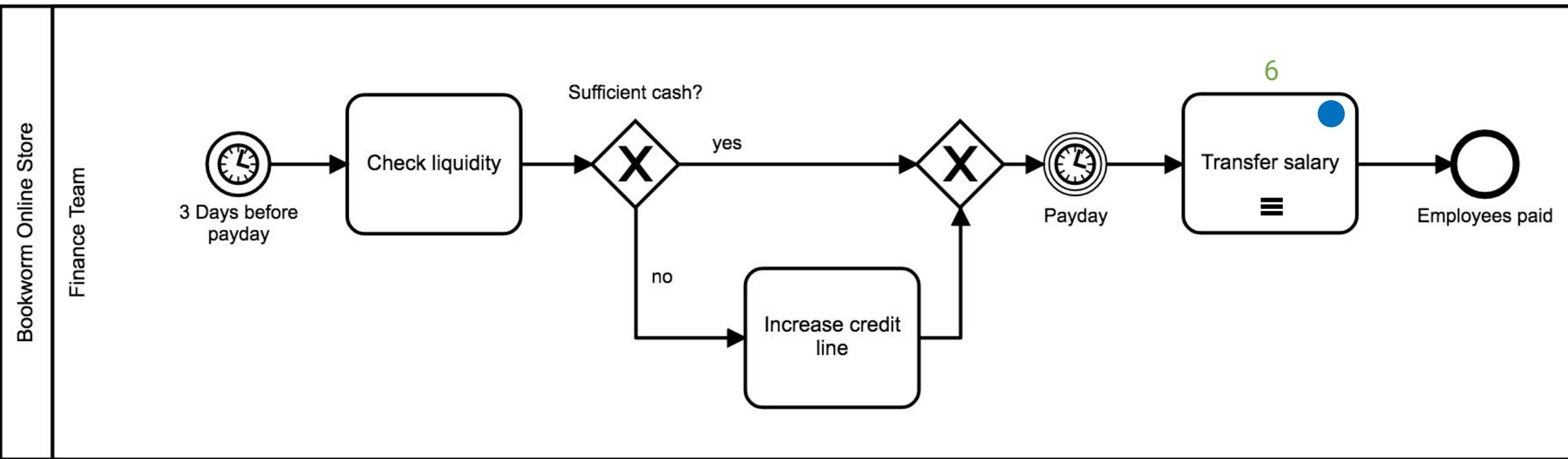


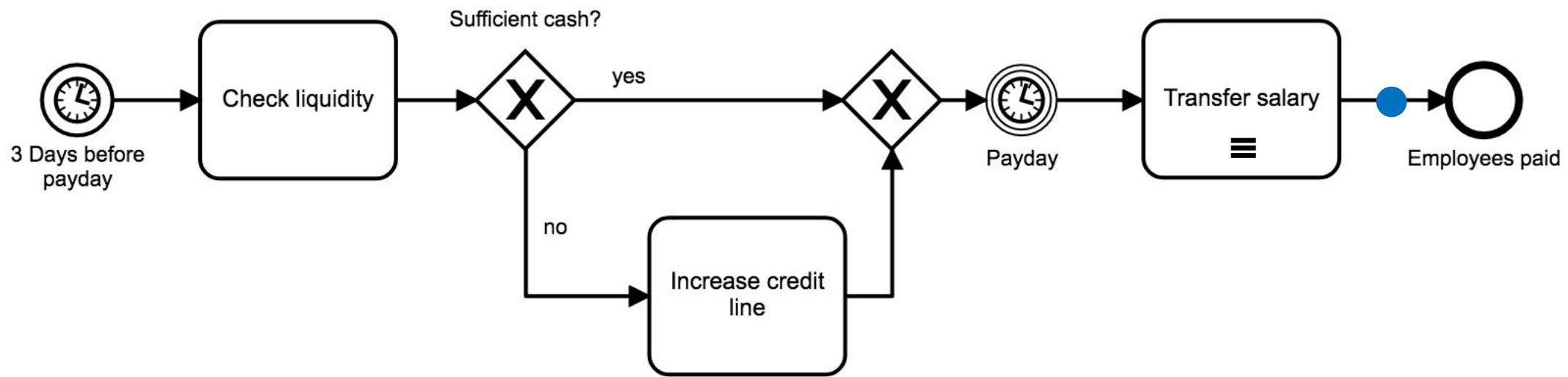


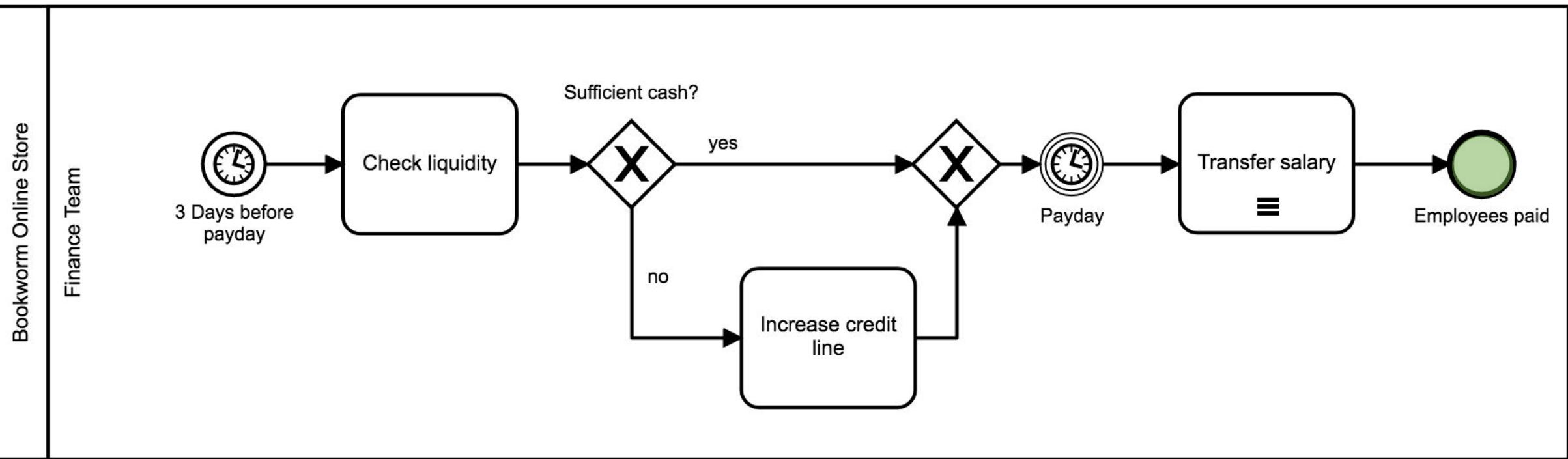














What is the difference between multi instance and loop?



Heat Panels

Ω

process camp



Heat Panels
 Ω



Salaries
 $=$



Heat Panels

Ω

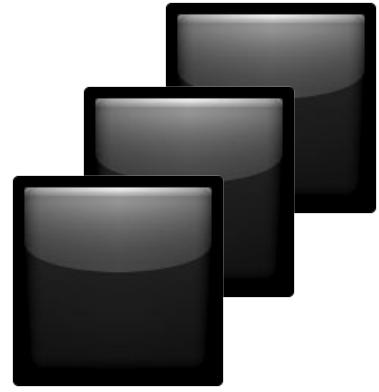
process camp



Heat Panels

Ω

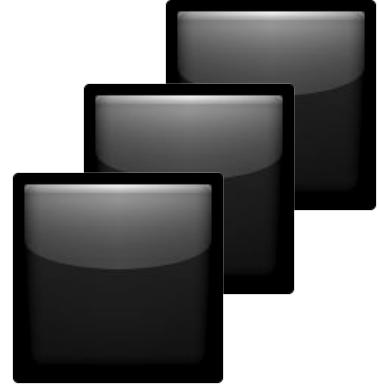
process camp



Heat Panels

Ω

process camp

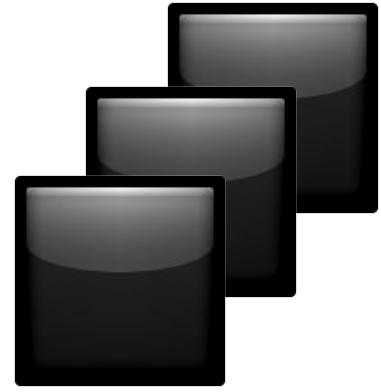


Heat Panels

Ω



Every sample is
exactly the same

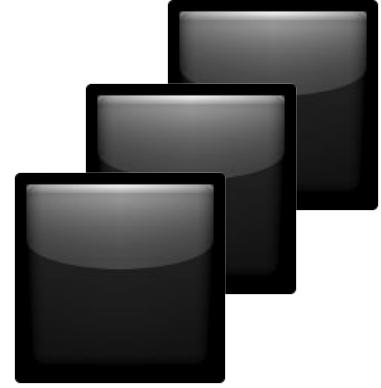


Heat Panels

Ω



The task is
performed many
times with **the
same input**

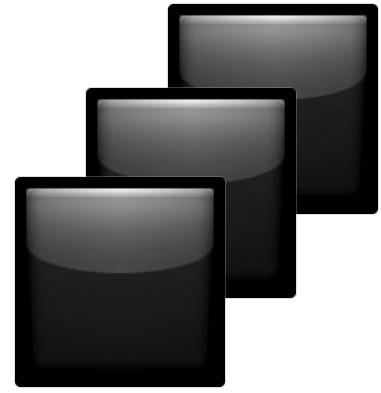


Heat Panels
 Ω



Salaries
 $=$

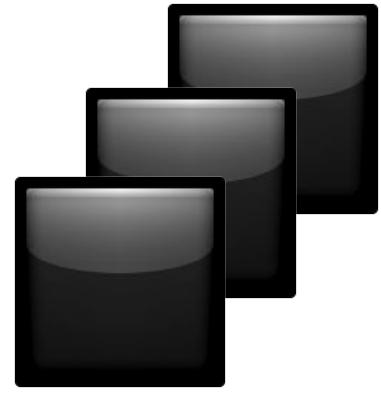
process camp



Heat Panels
 Ω



Salaries
≡



Heat Panels
 Ω



Salaries
 \equiv

process camp



Every salary is
different



Salaries



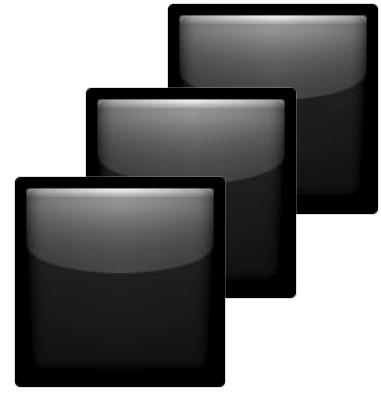


The task is
performed many
times with
different inputs



Salaries





Heat Panels
 Ω

Same input



Salaries
≡

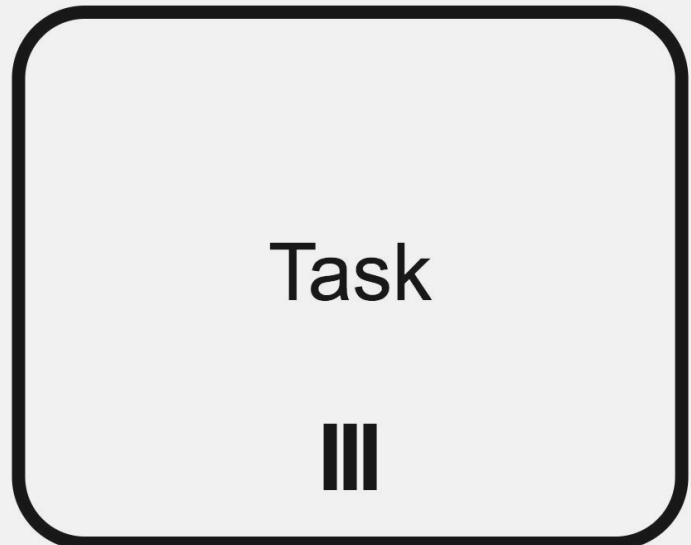
Different input

process camp

BPMN Theory

process camp

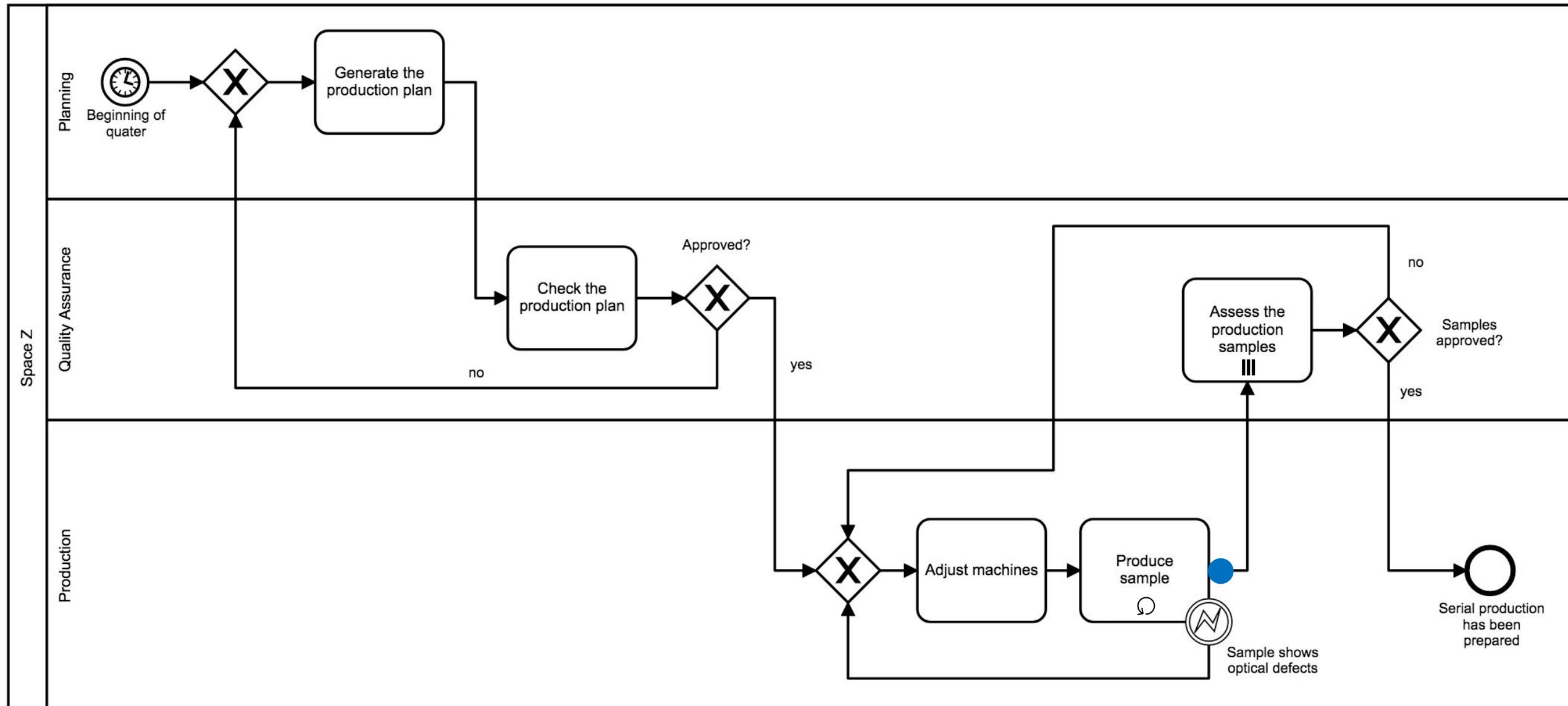
Parallel Multi Instance



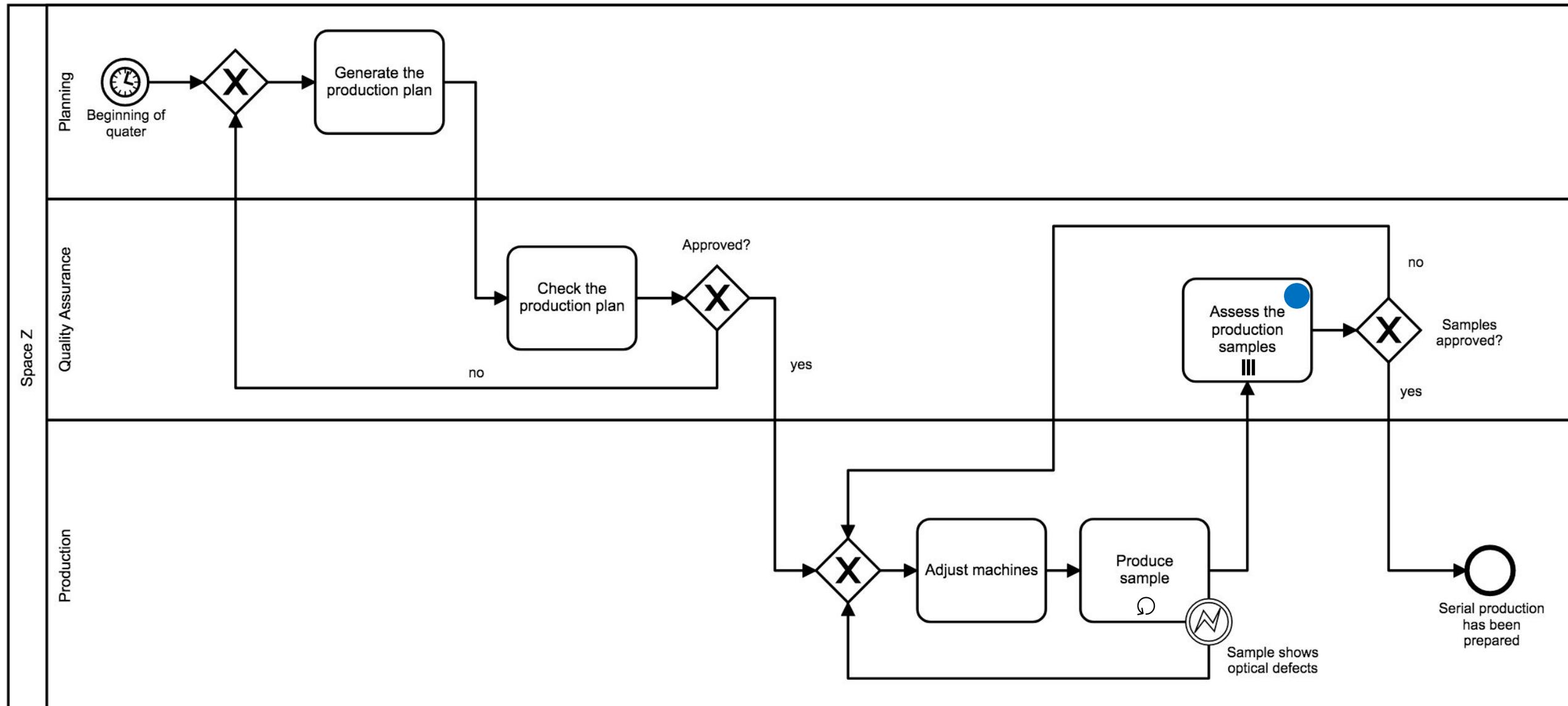
- Defines an activity that is executed in parallel
- The execution ends with a defined condition

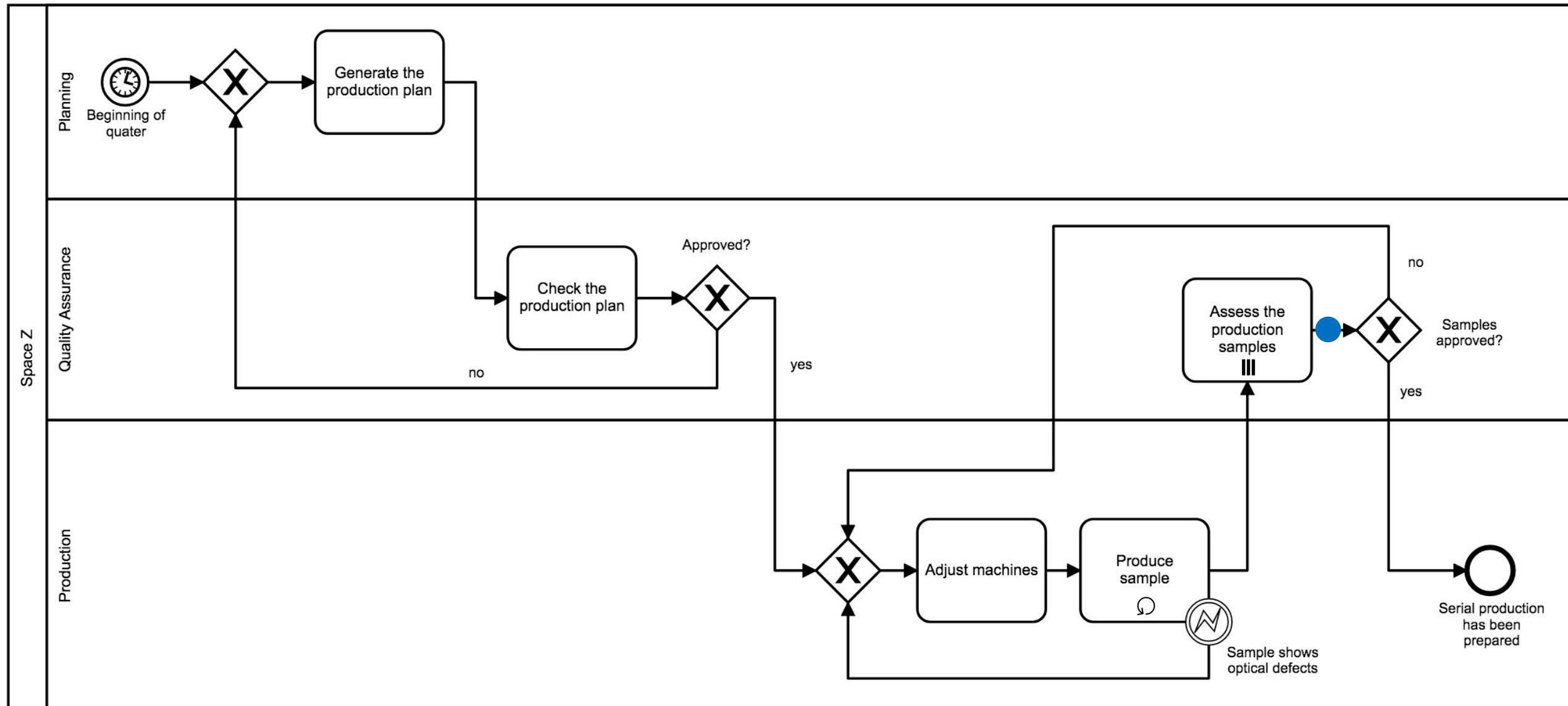


The Quality Assurance at Space Z is very
well equipped

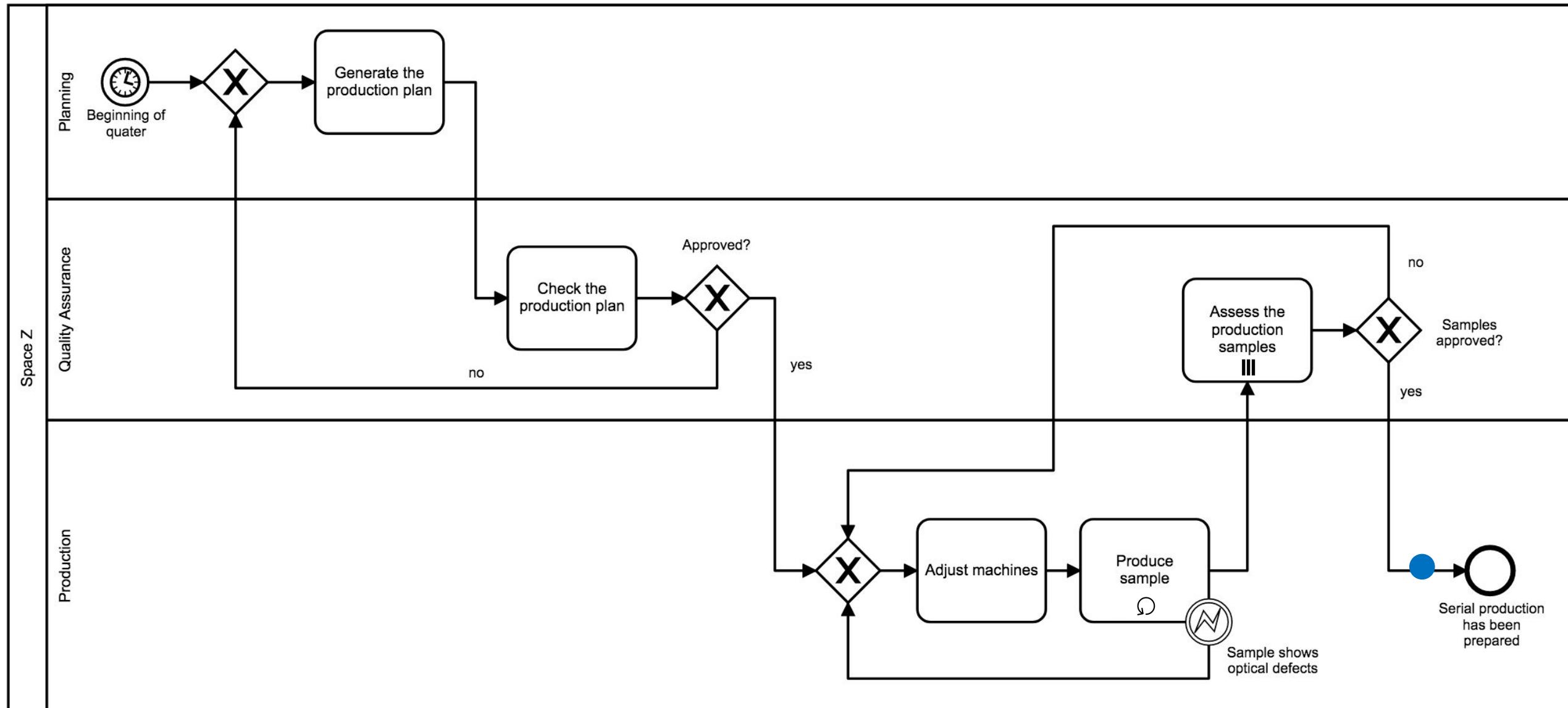


process camp

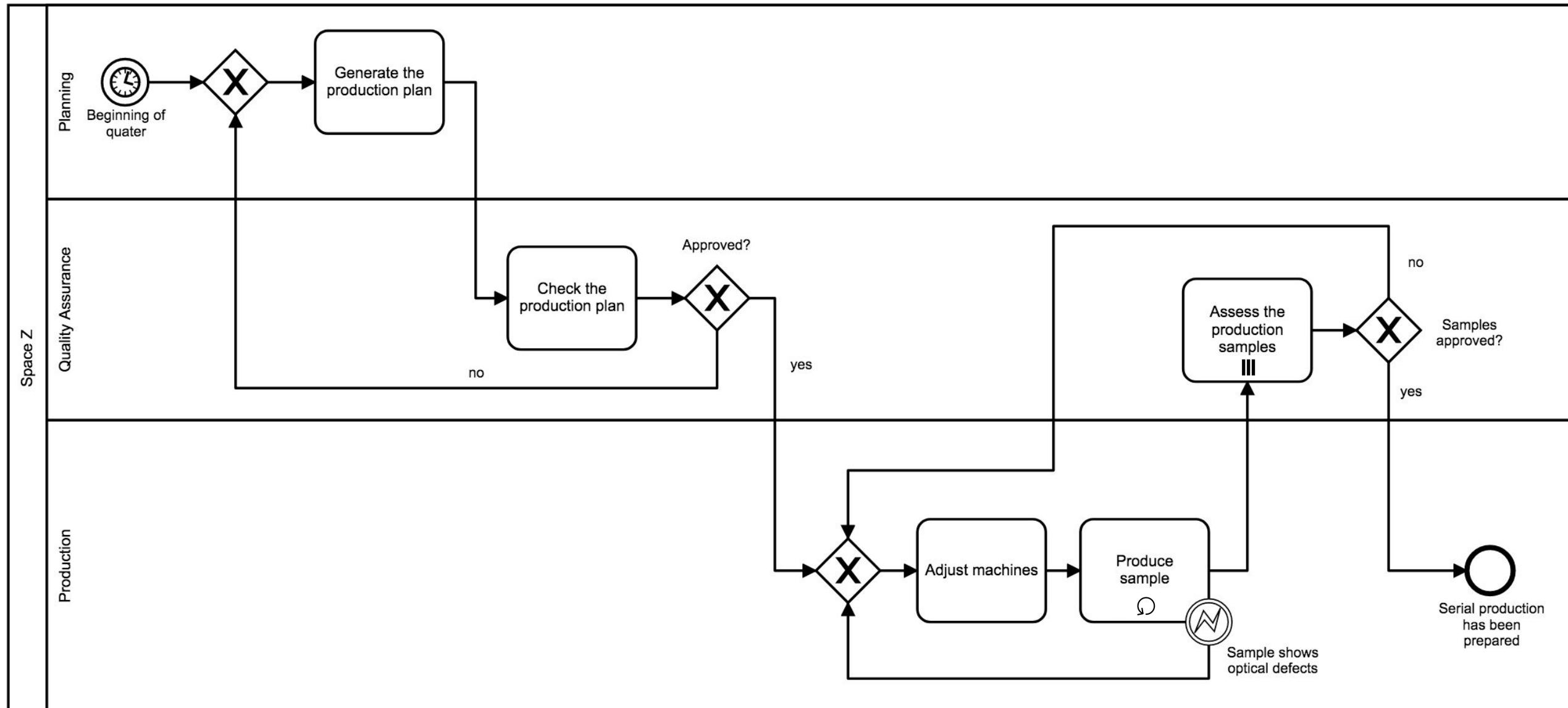




process camp



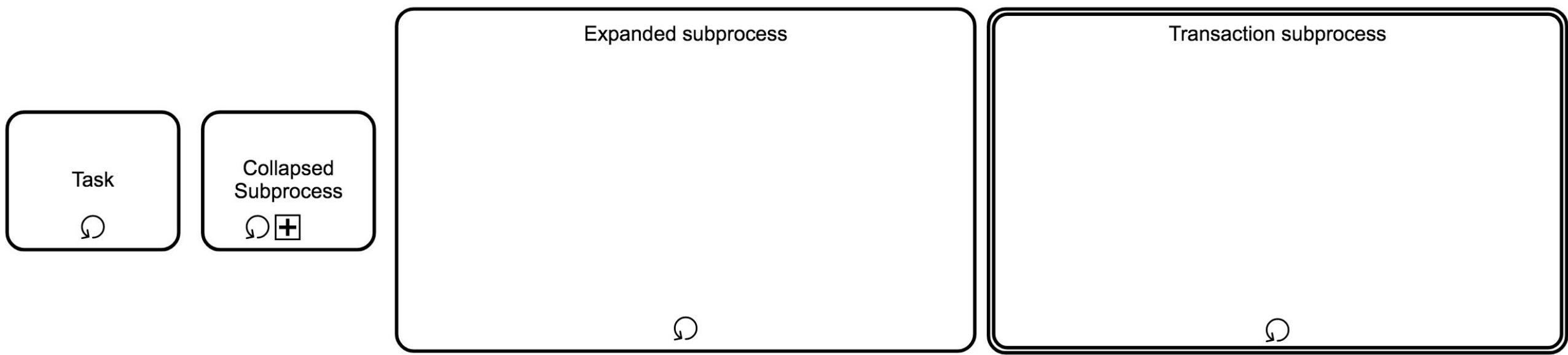
process camp



process camp



The Loop Types can be applied for
any kind of task and subprocess
(except the event subprocess)



process camp

BPMN Use Case





Tofisu

- Tofisu is a small and fast growing tofu retailer
- Lea, the business owner, recognized the trend early and is now in a good position to scale her business



Tofisu

- To grow the revenue, she hired five new sales representatives
- To organize this team and get it up to speed, efficient processes are indispensable



Tofisu

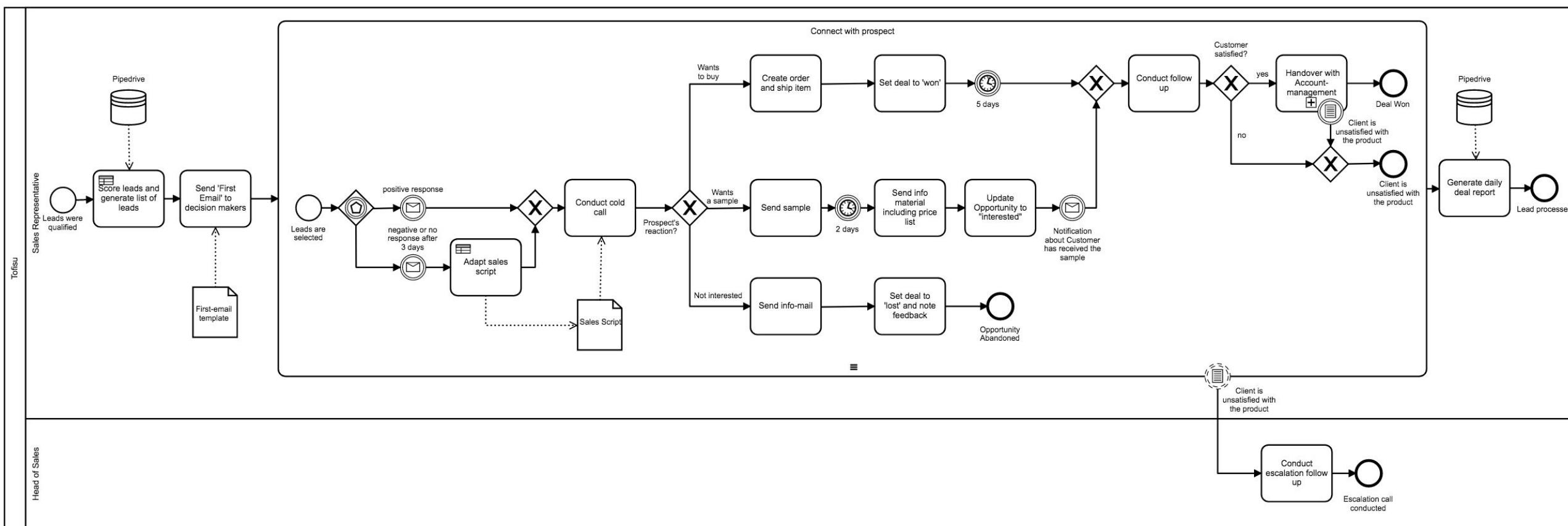
- Therefore Lea simply sketches the sales process so everybody knows how to do the job, even when she's not around

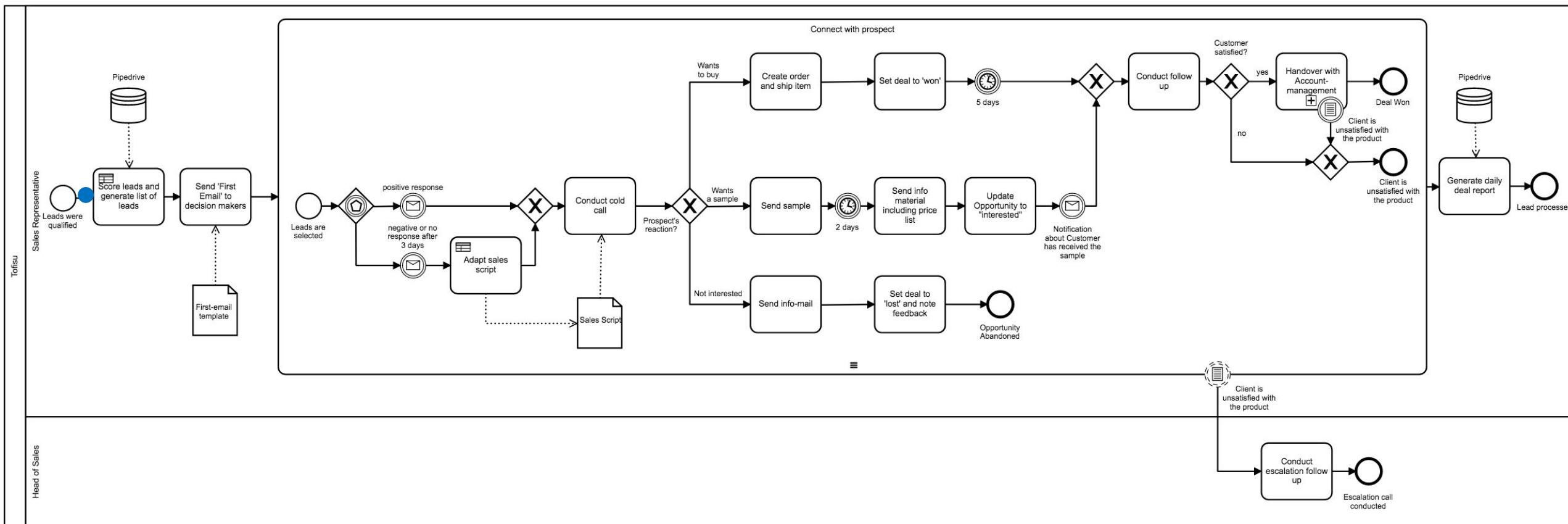


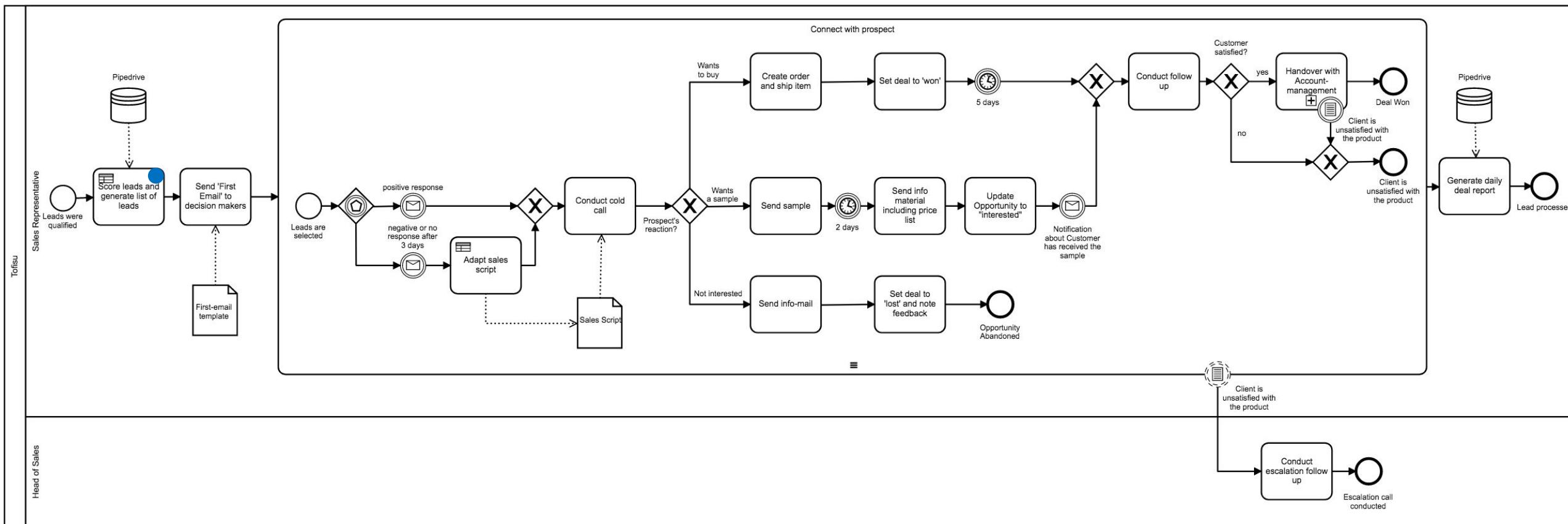
The process starts with a **lead**

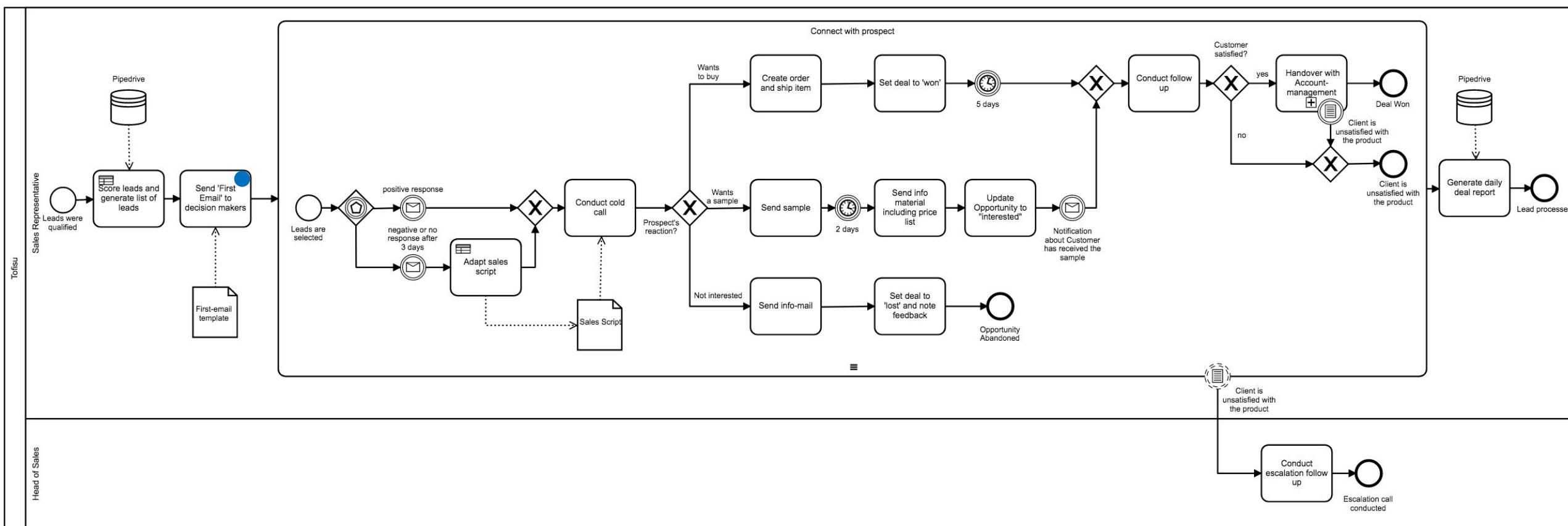


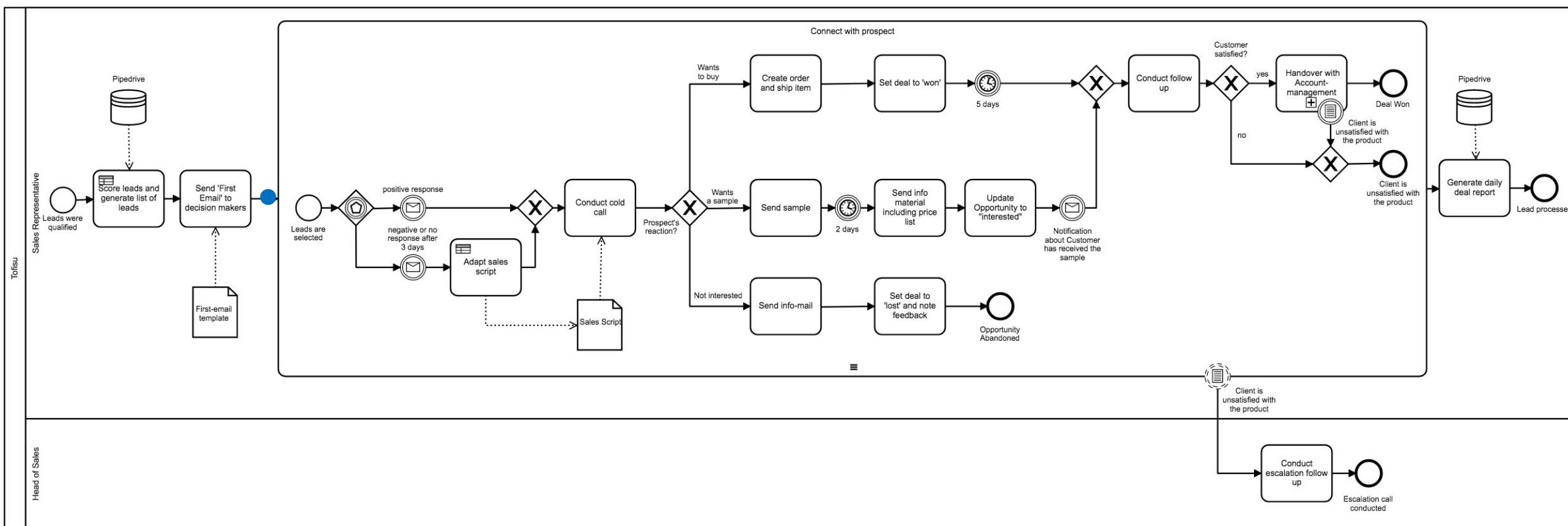
And ideally ends with a loyal customer





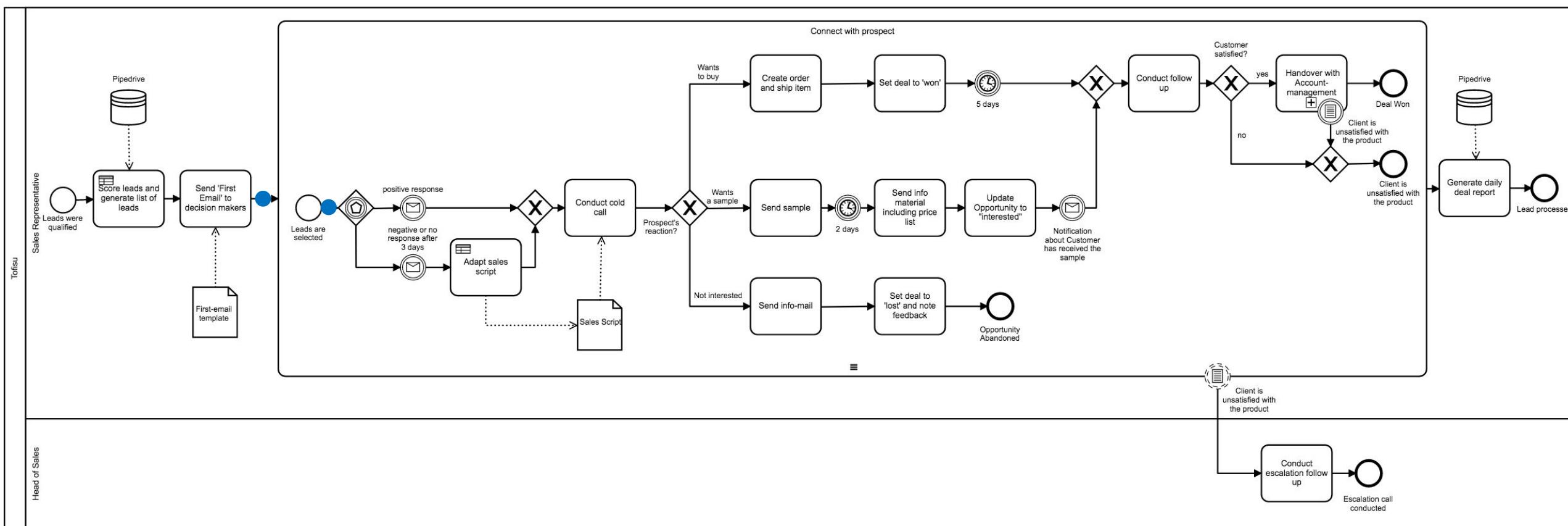


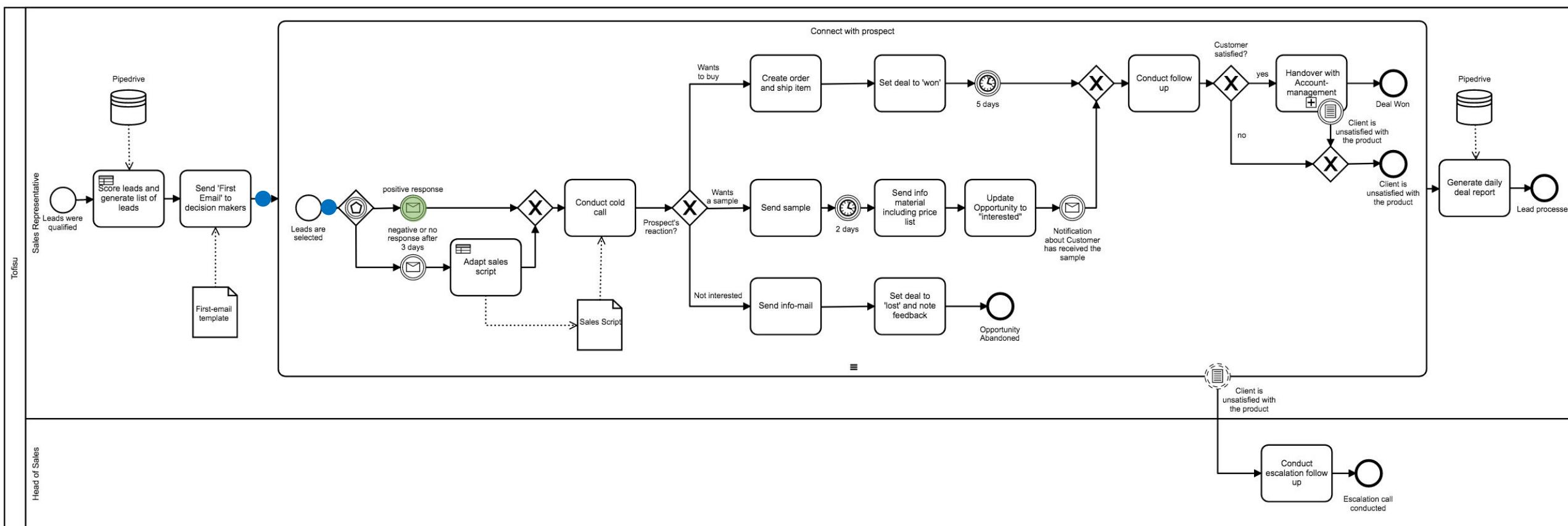


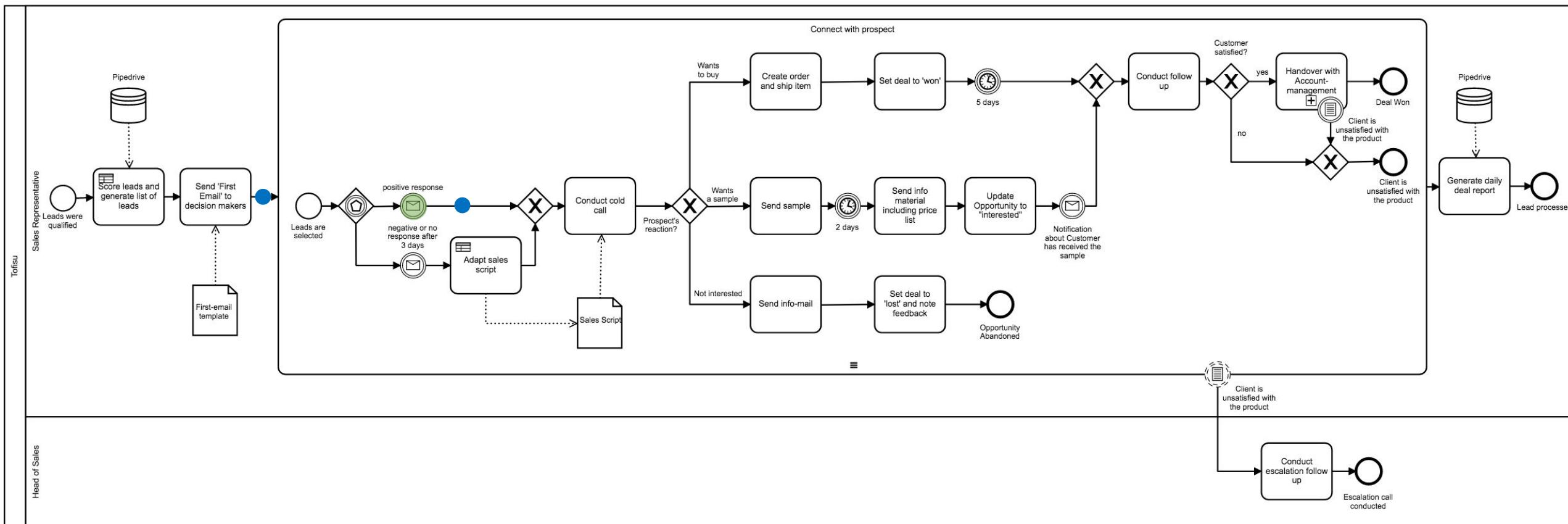


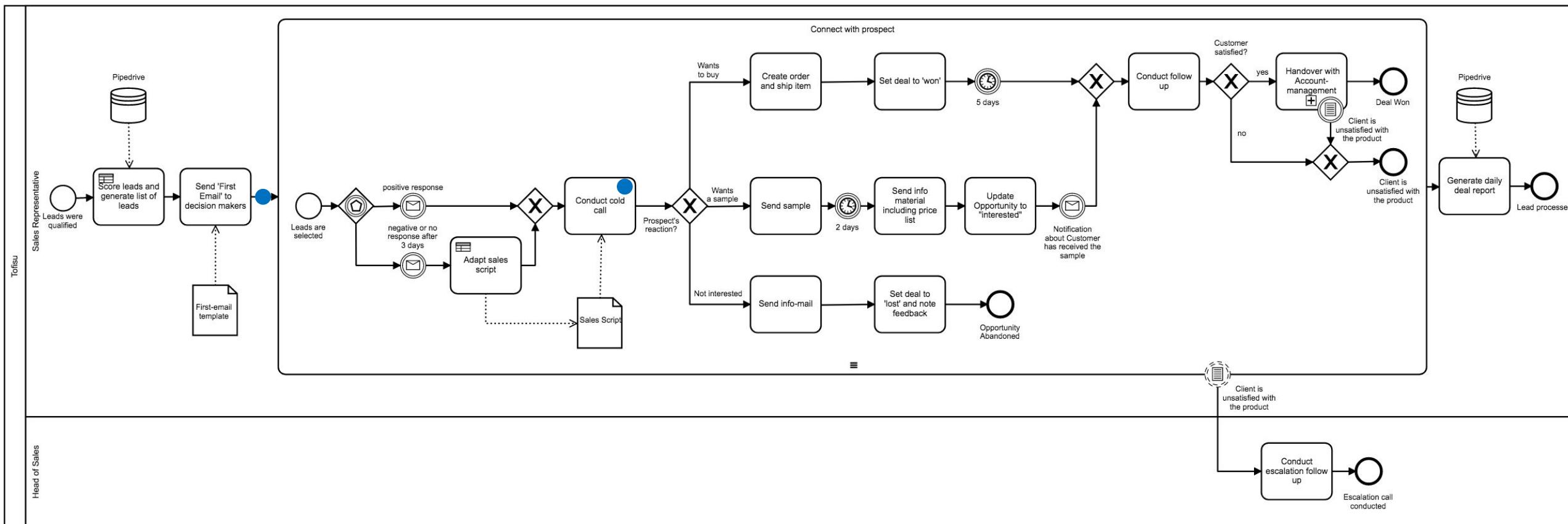


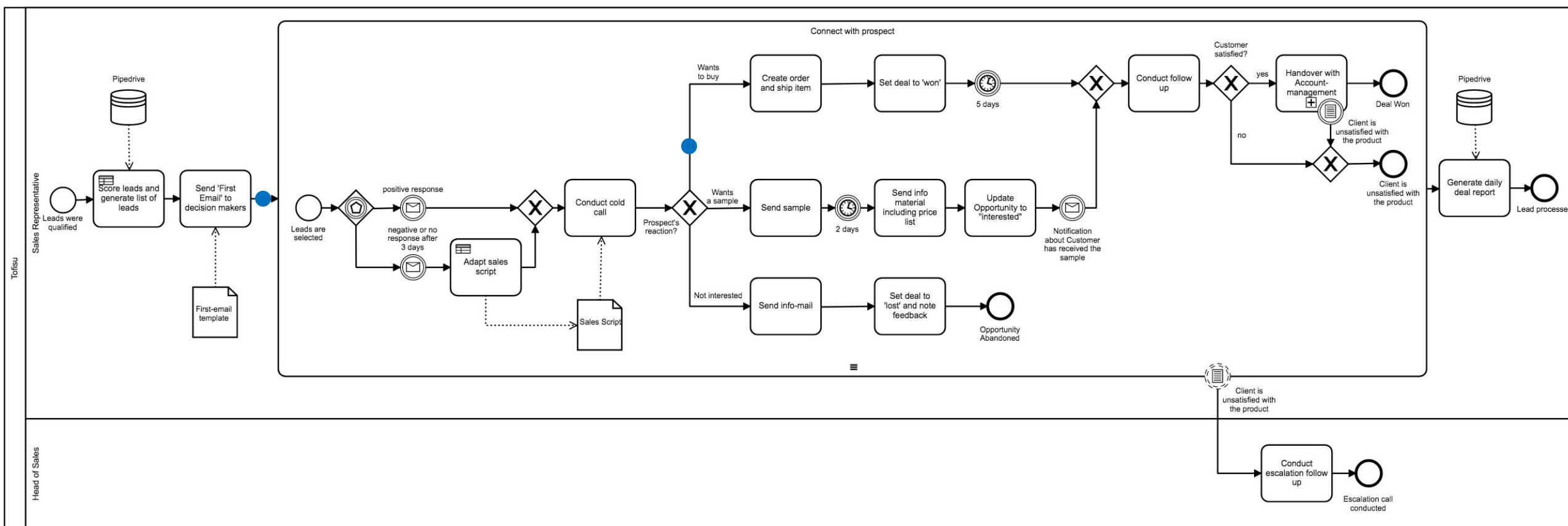
Let's start with the 1. Lead

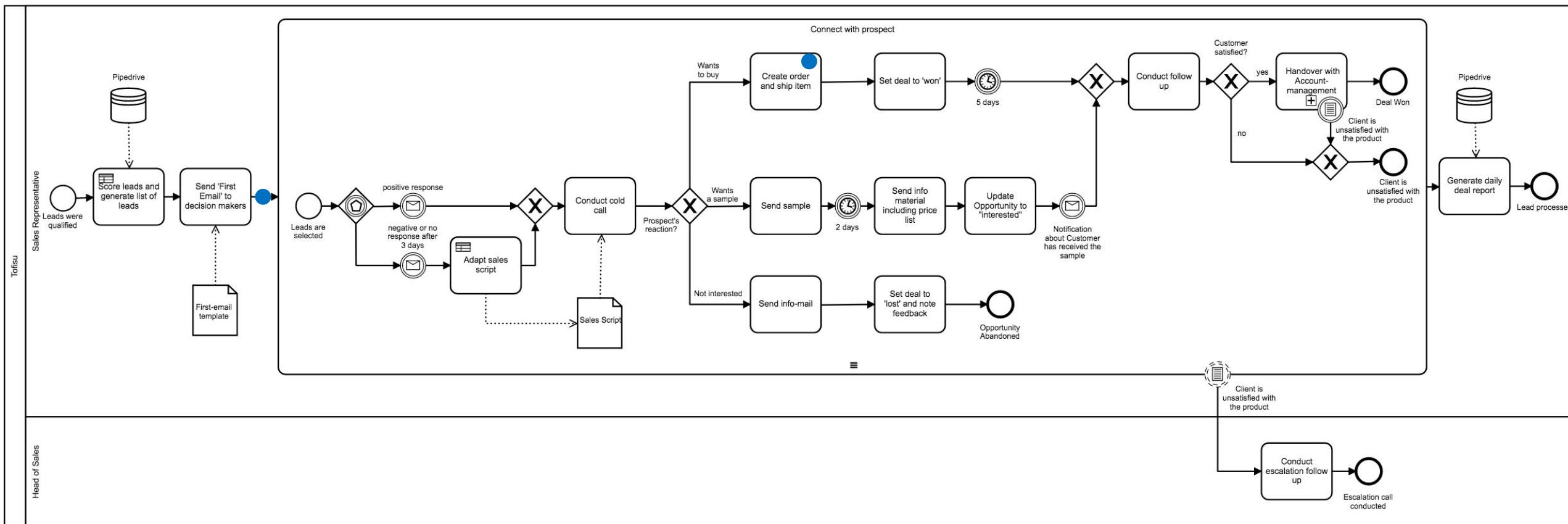


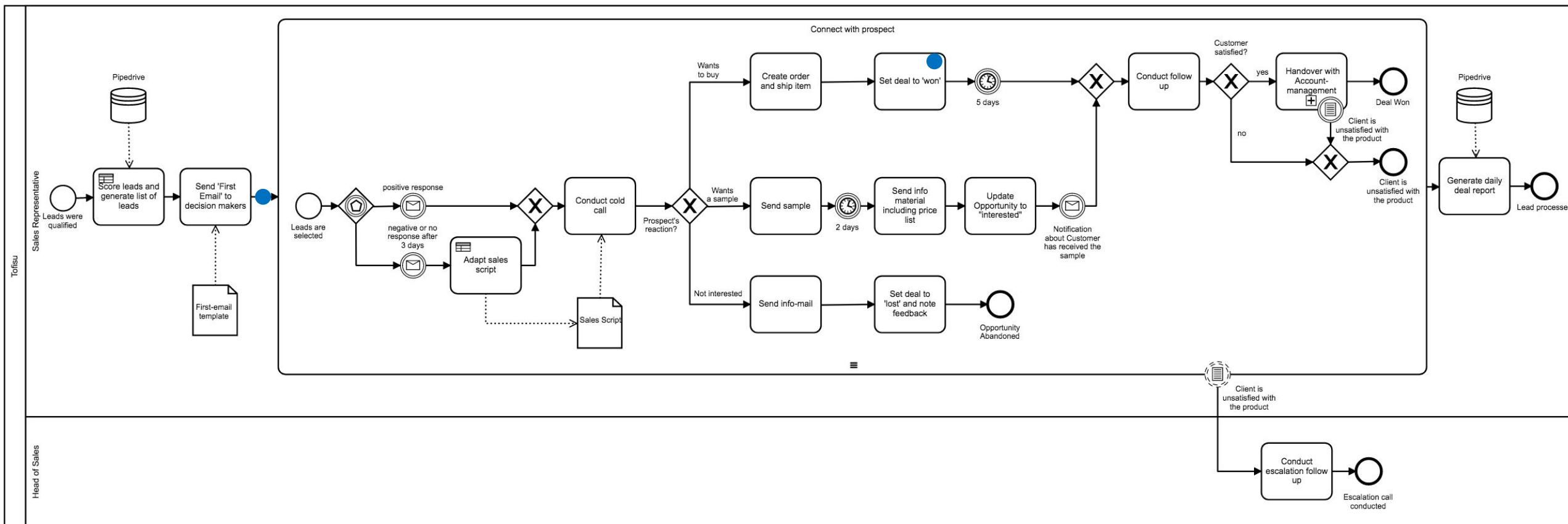


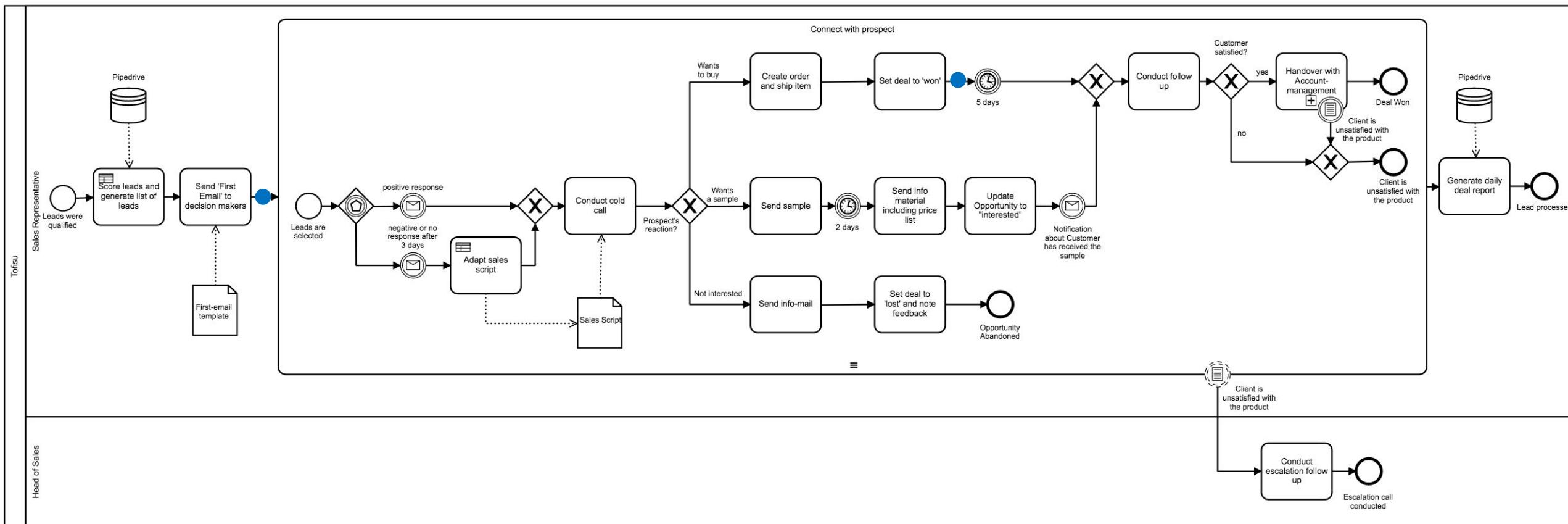


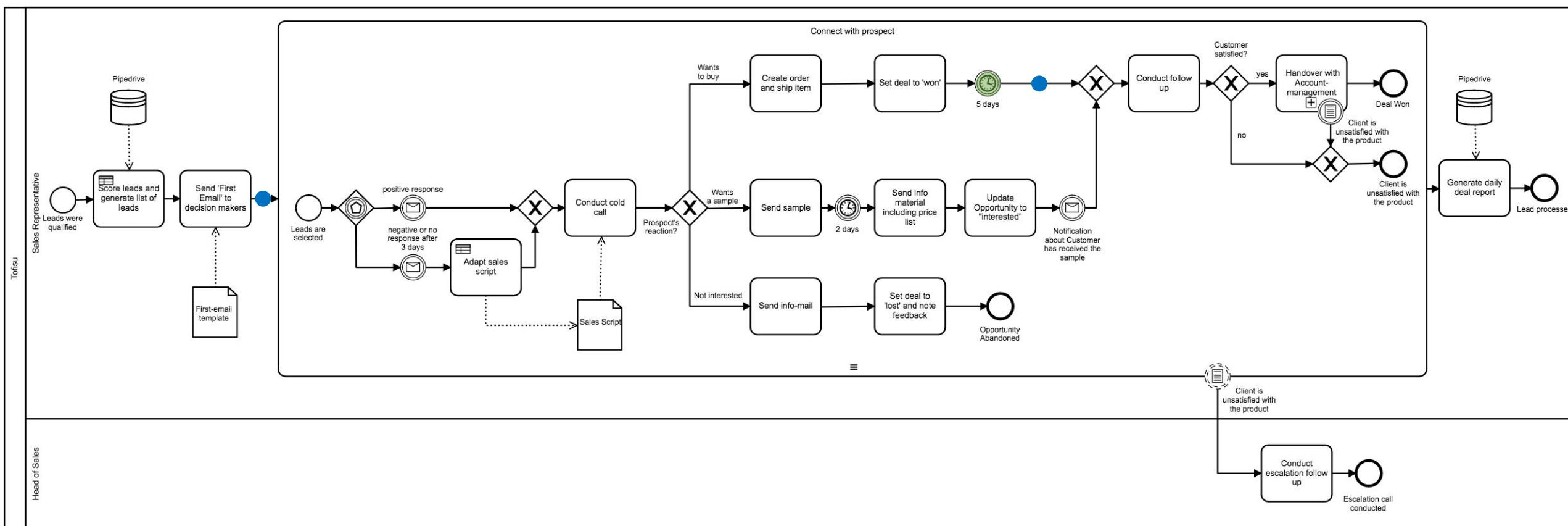


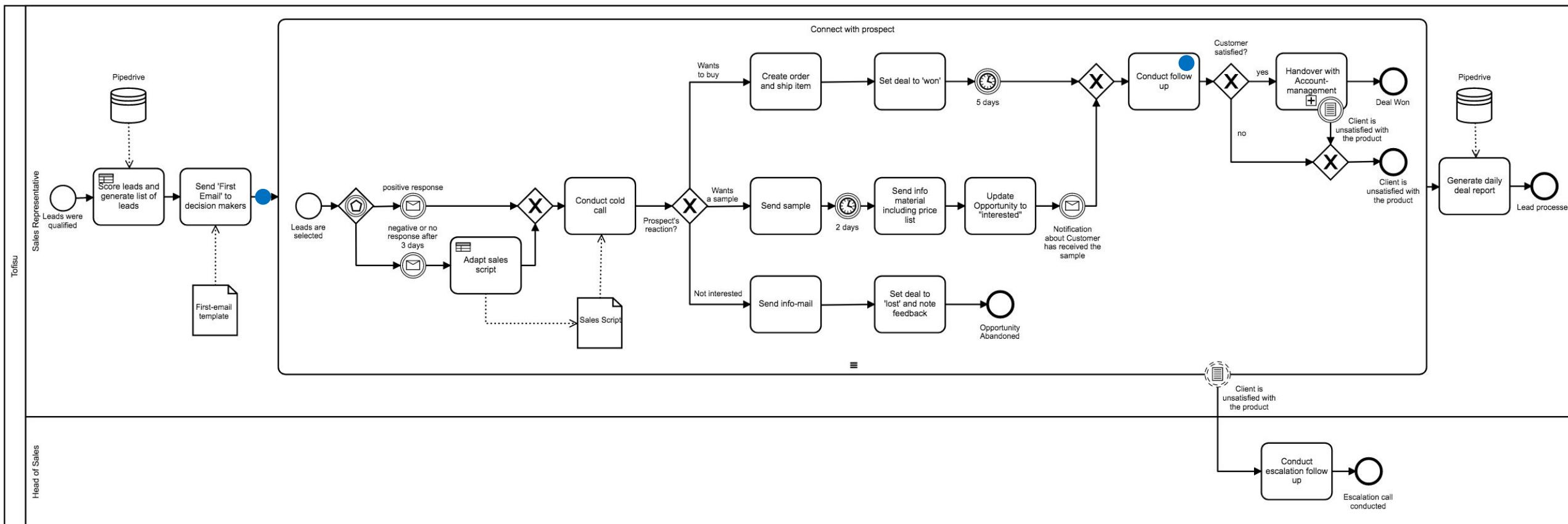


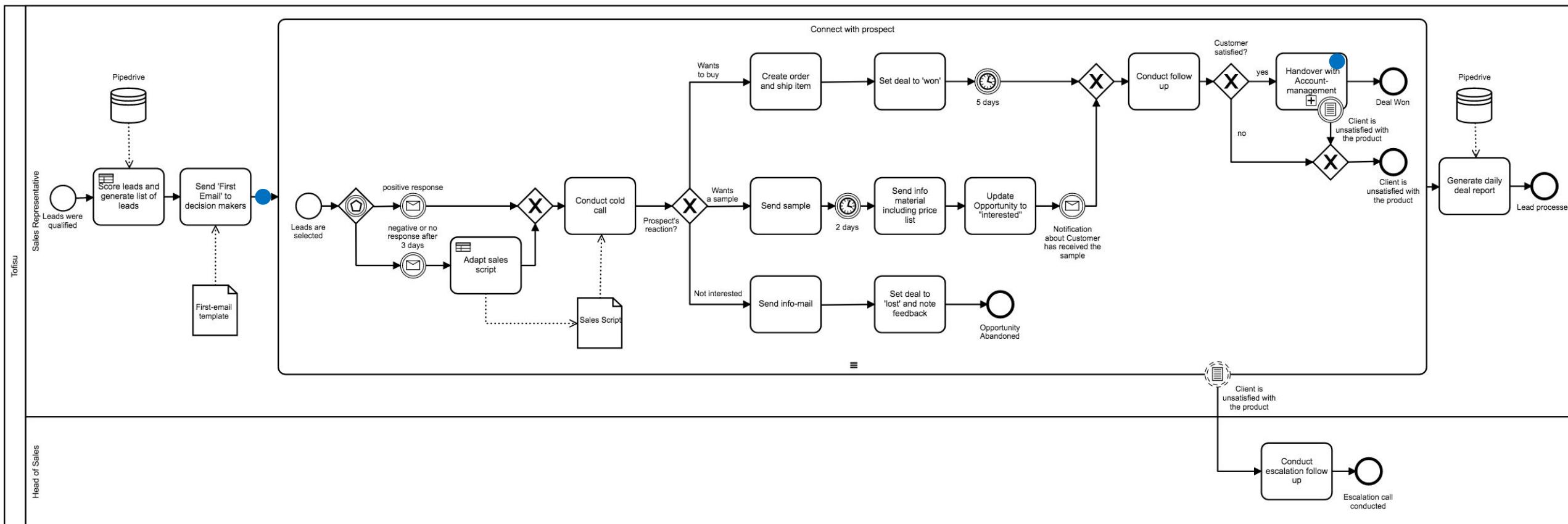


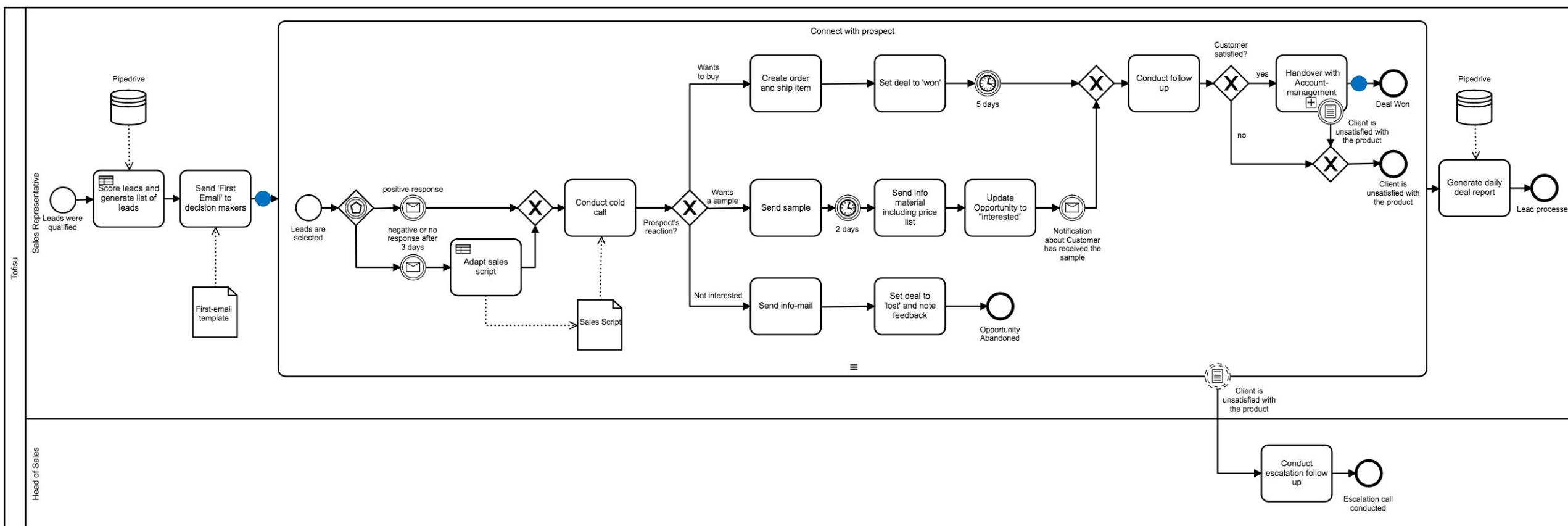


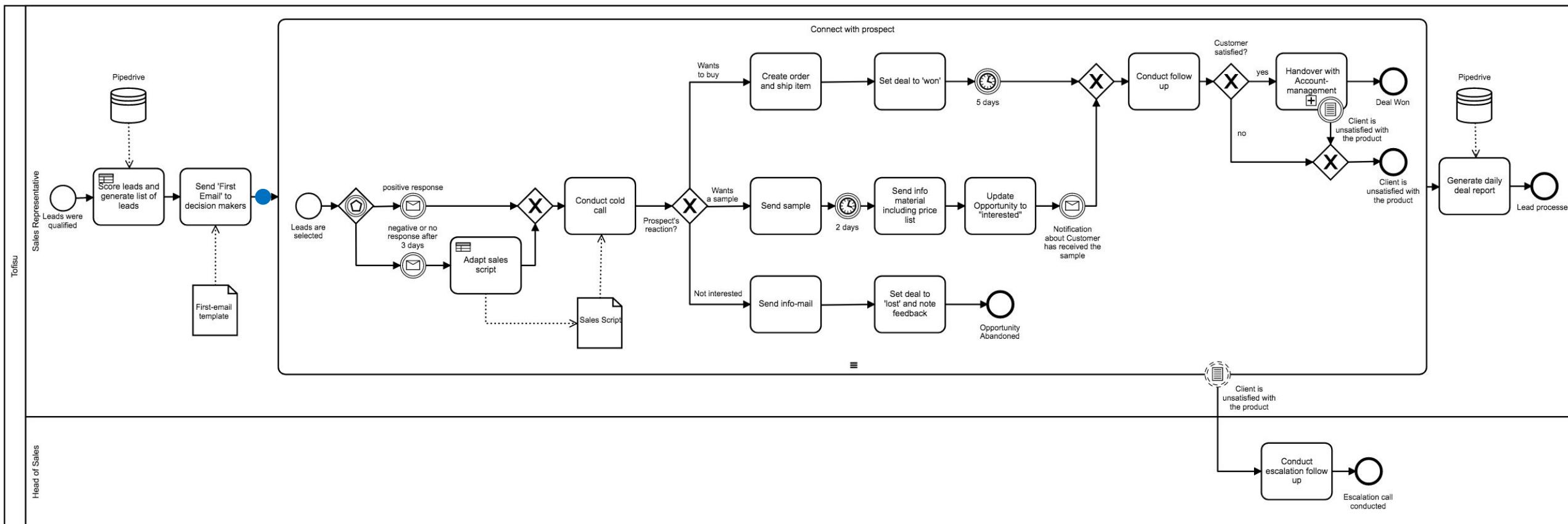


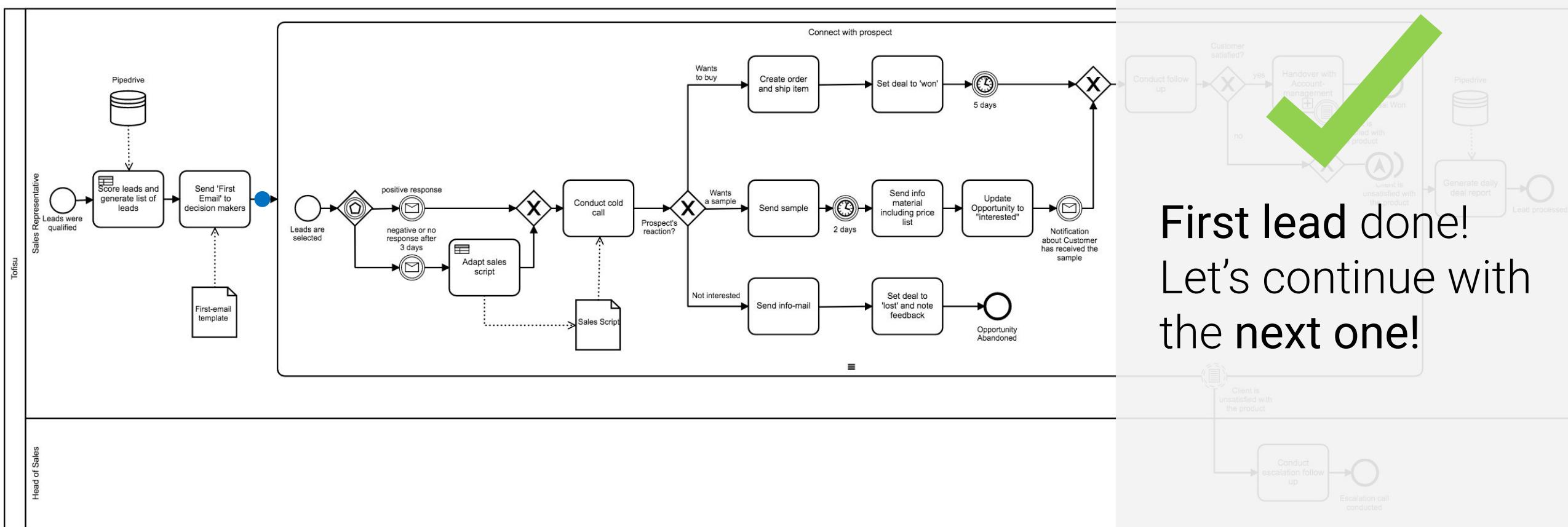






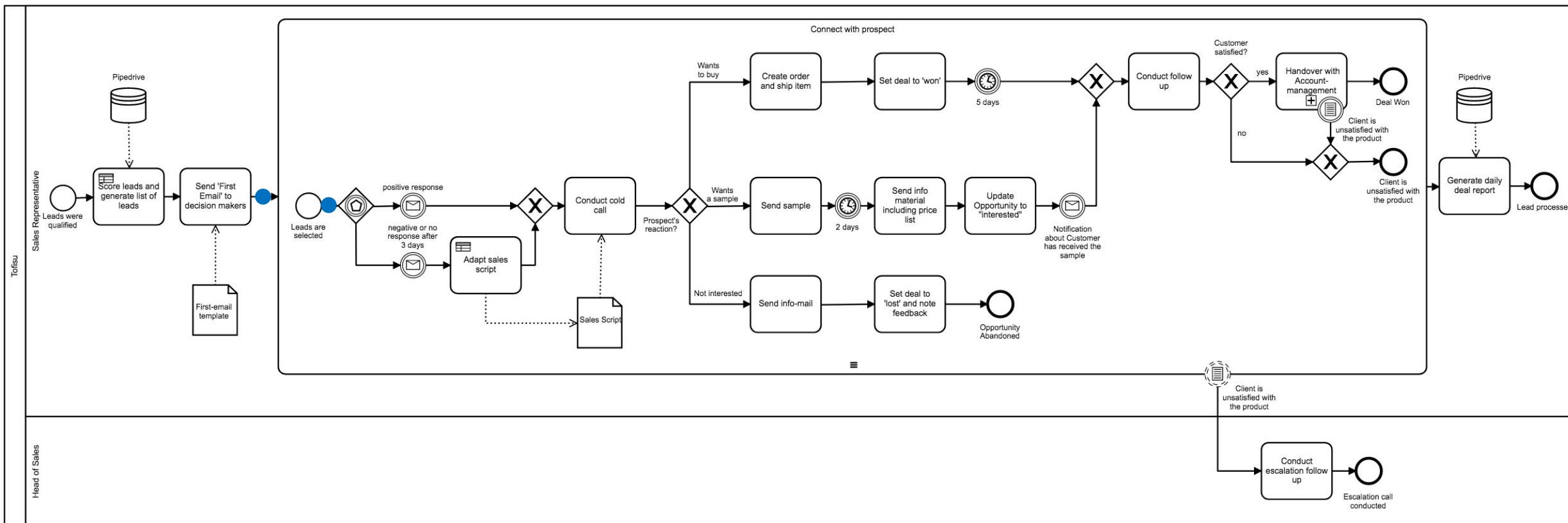




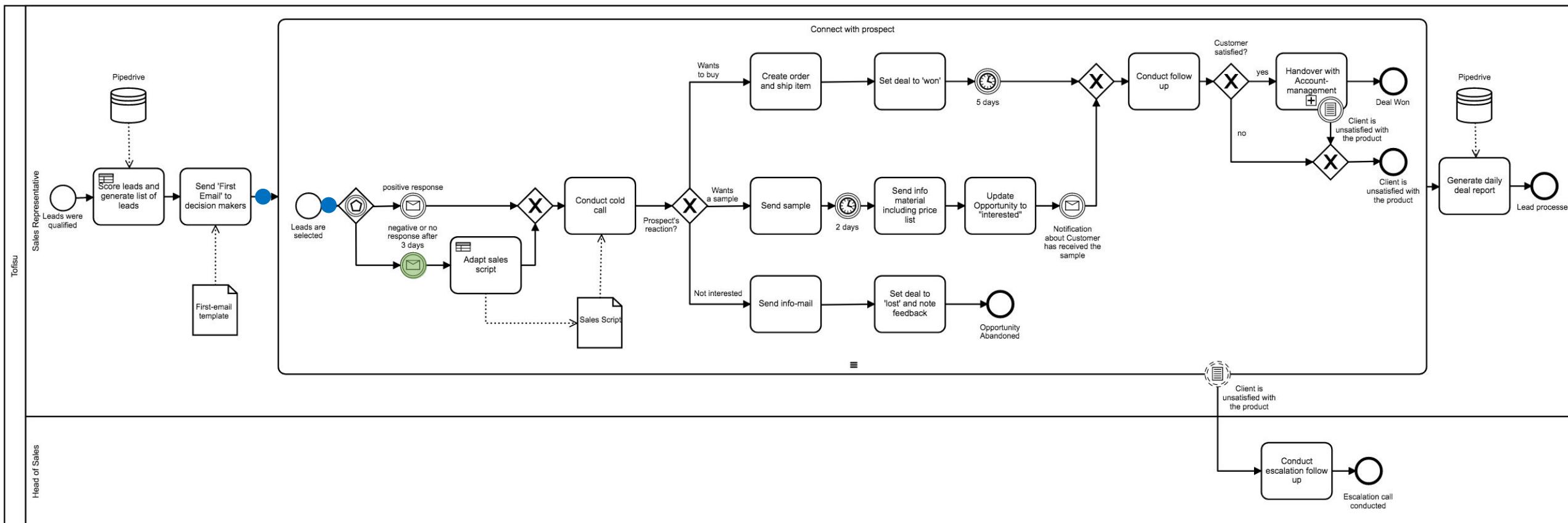


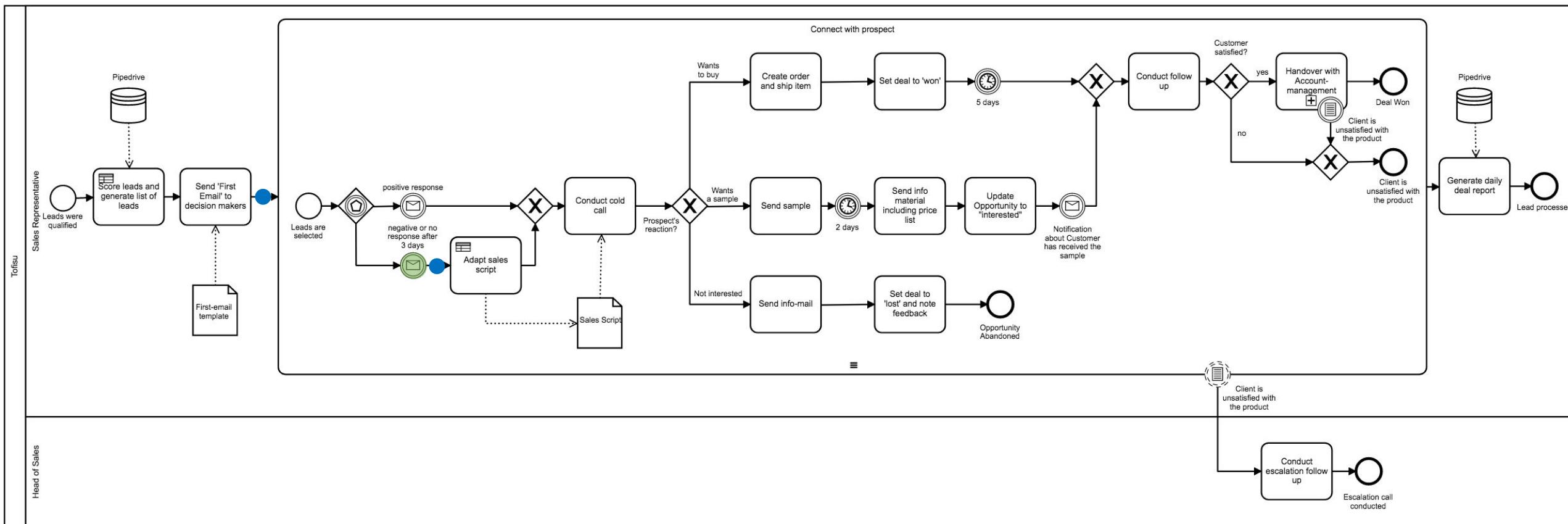
First lead done!
Let's continue with
the next one!

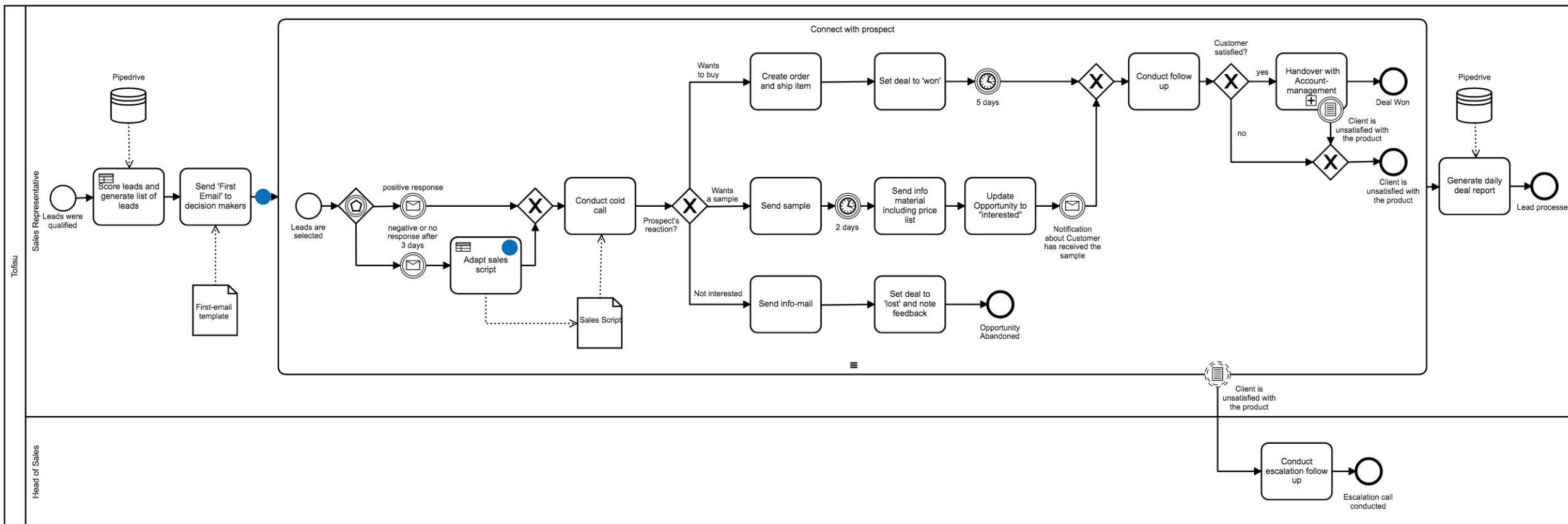
2. Lead

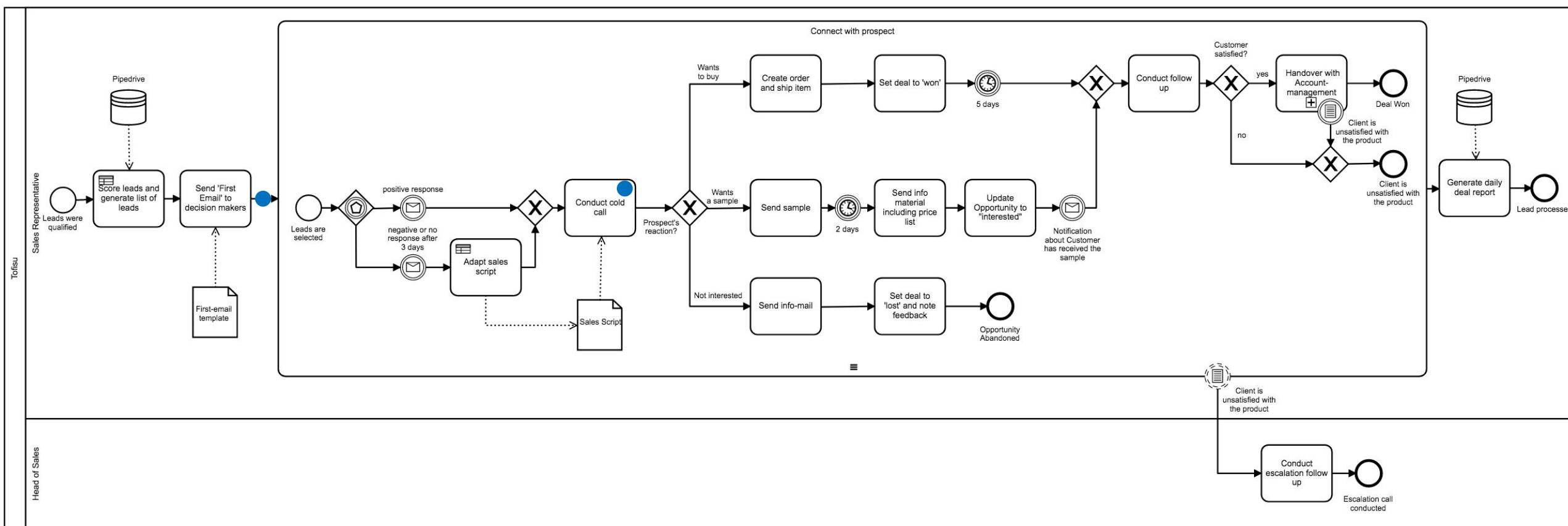


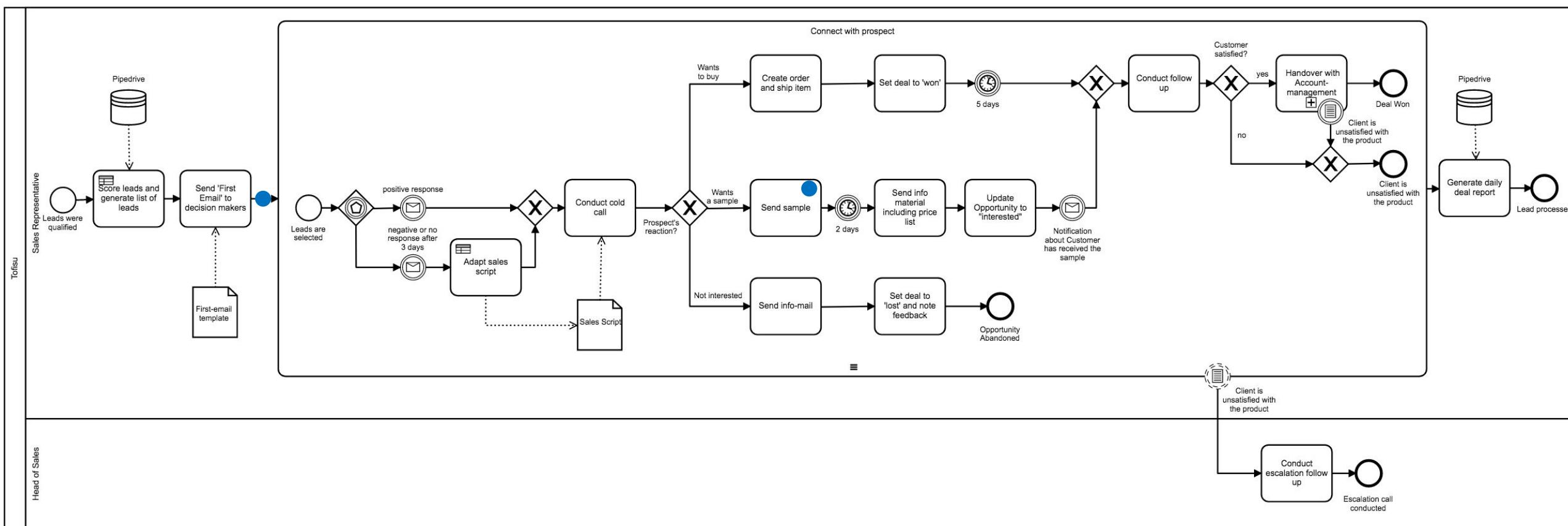
2. Lead

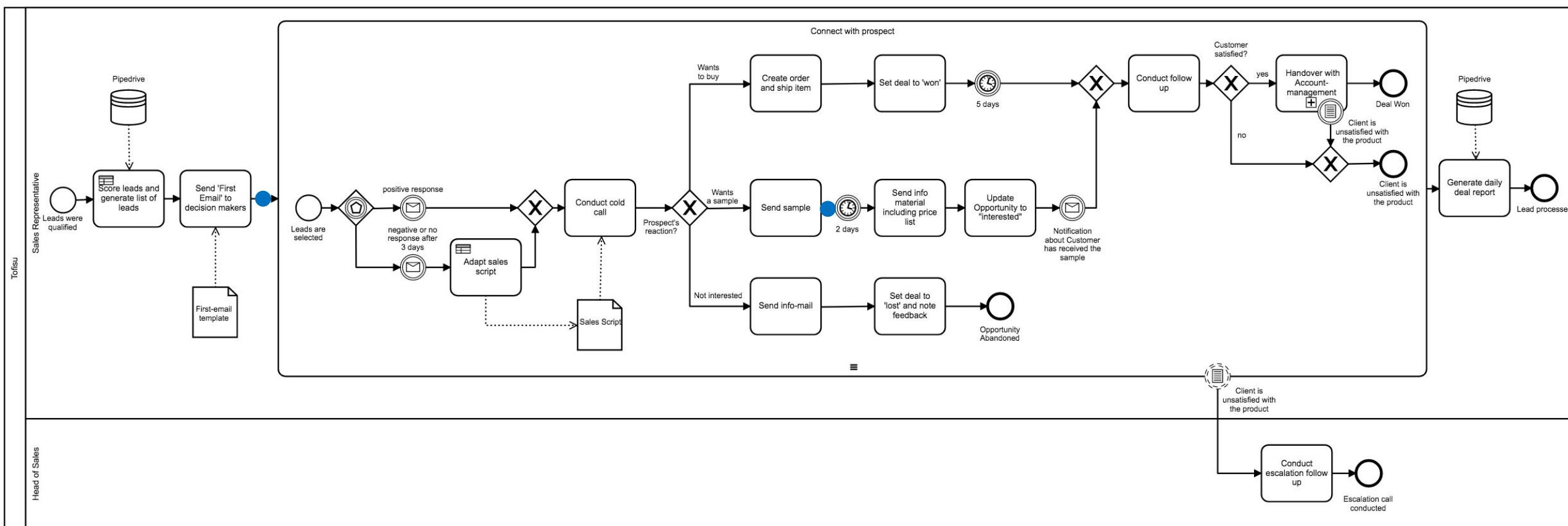


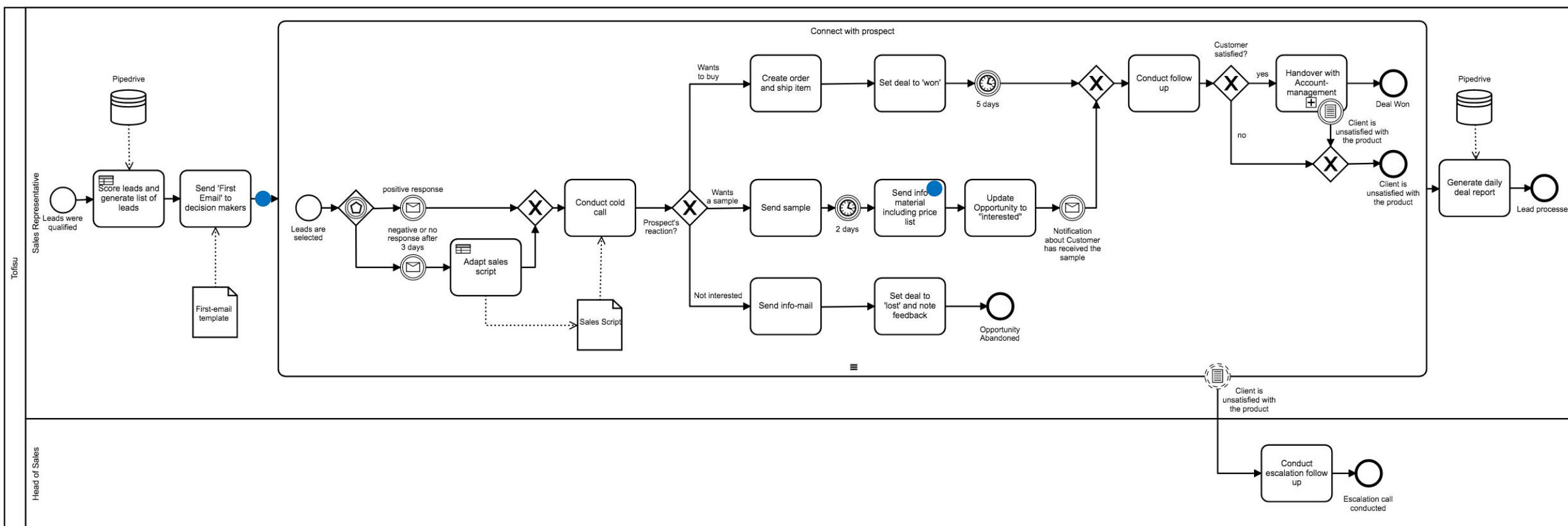


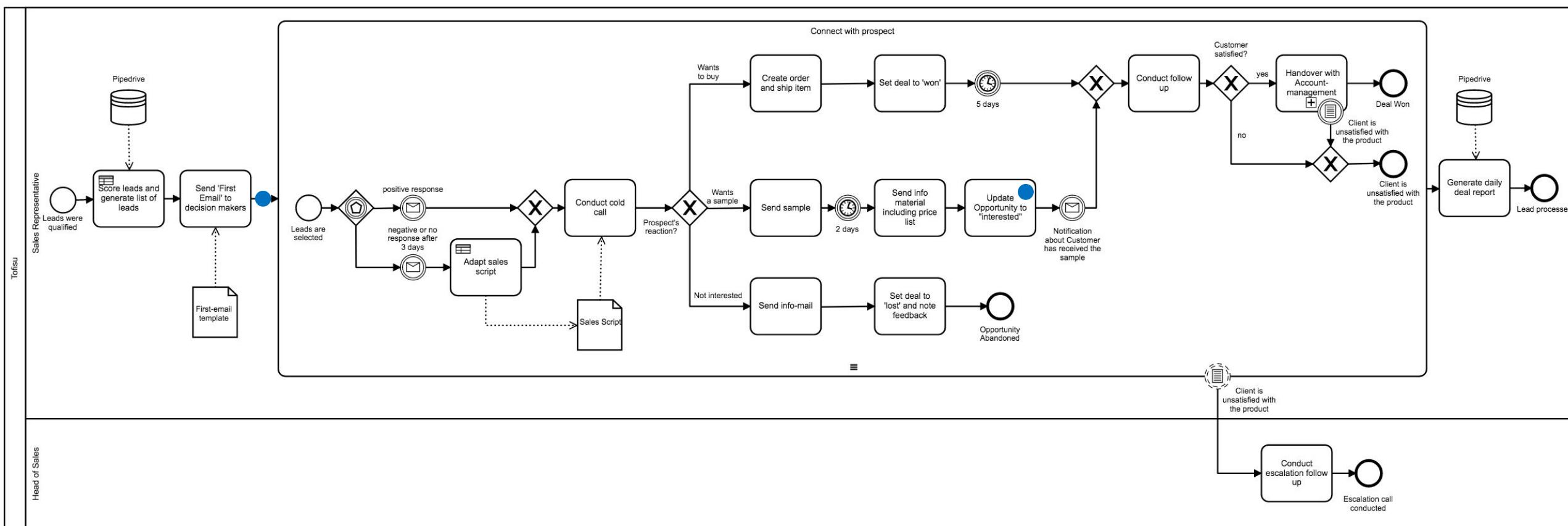


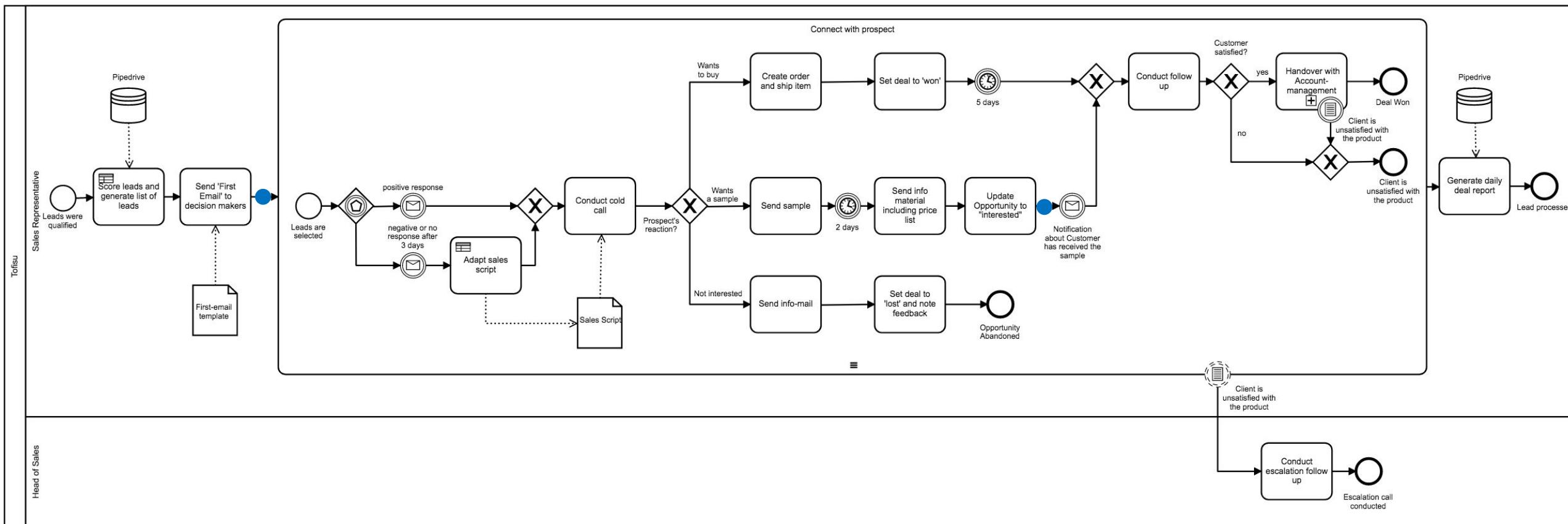


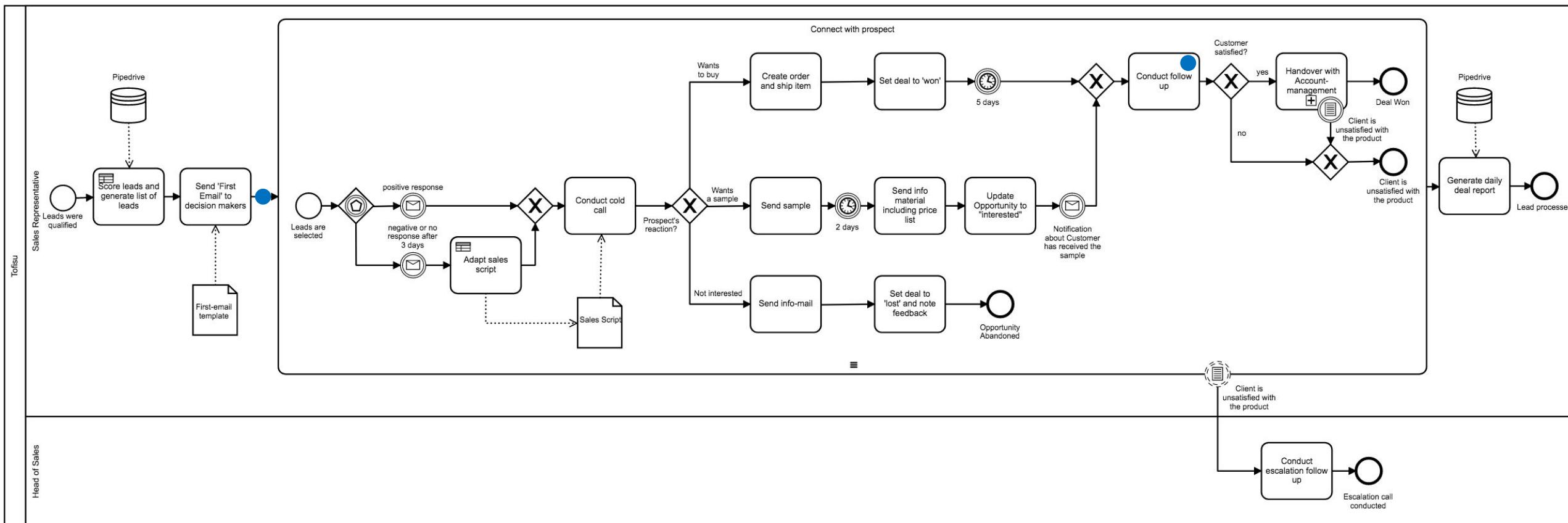


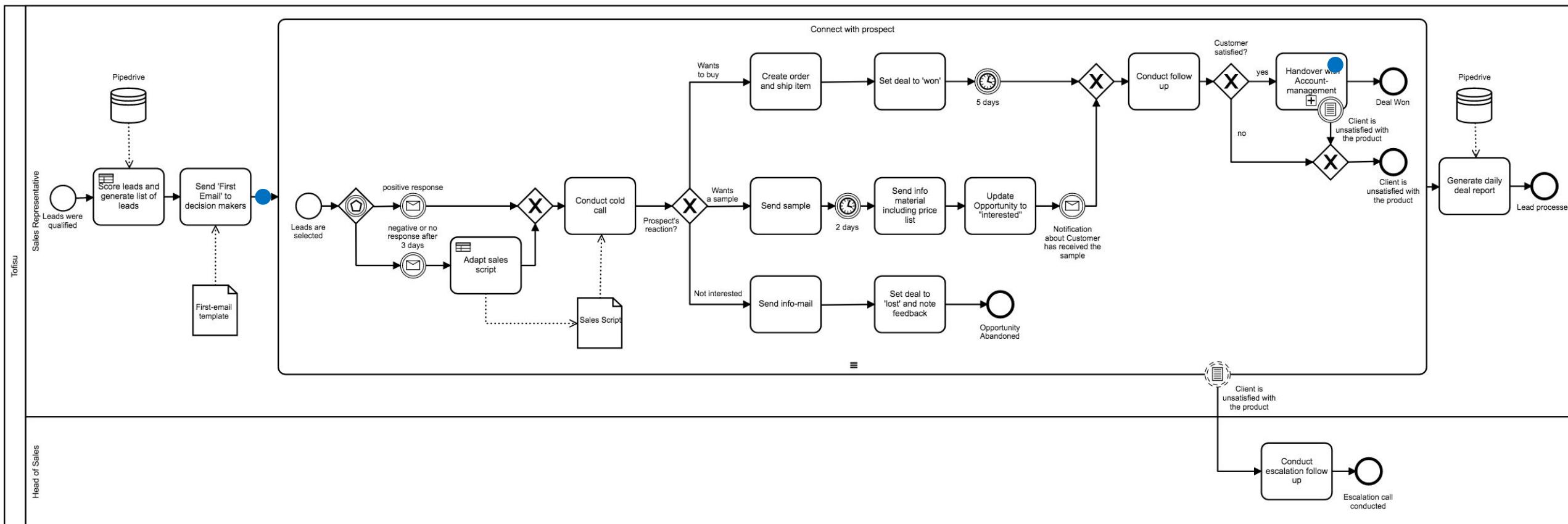


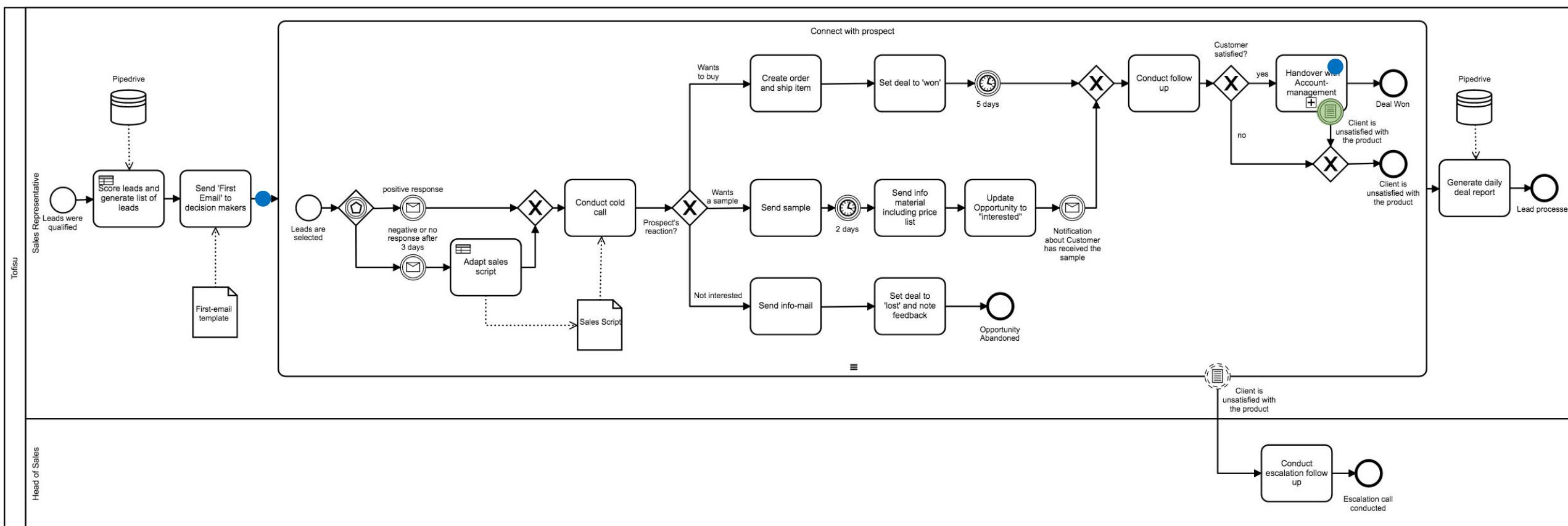


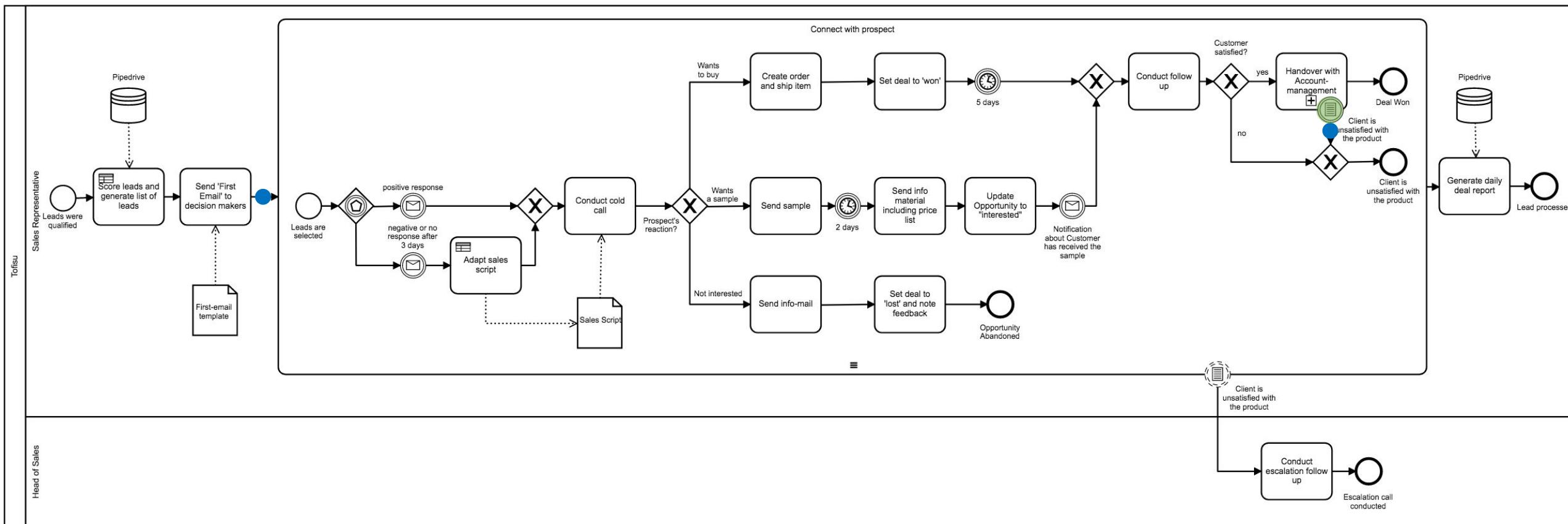


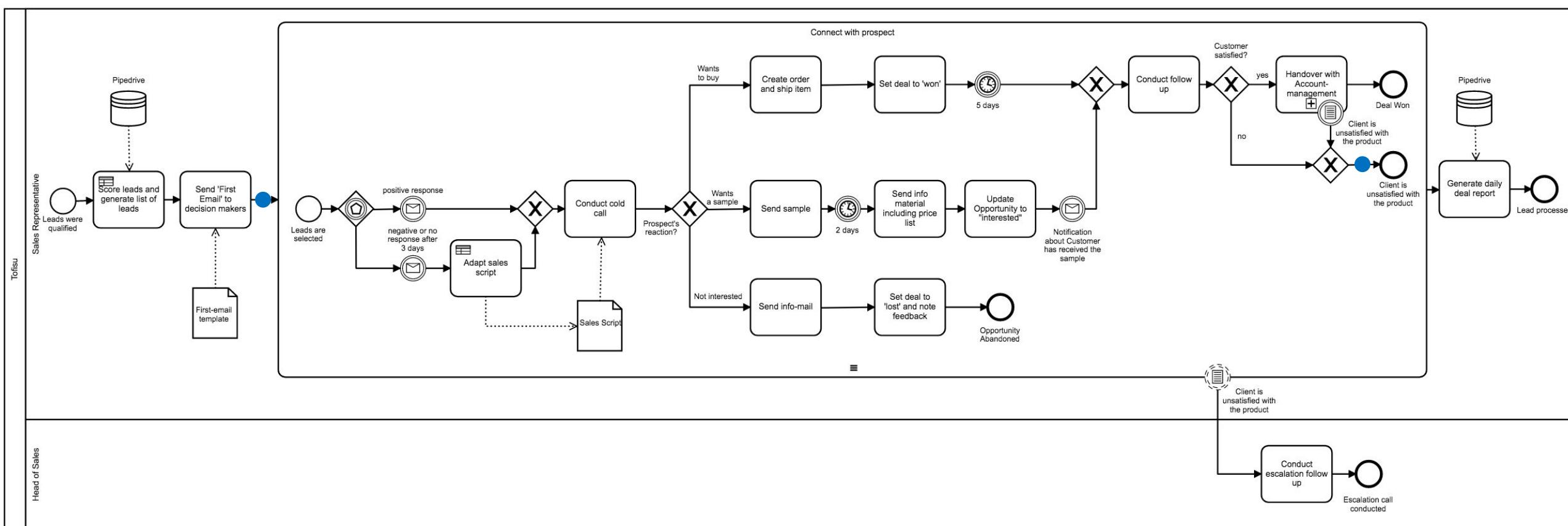


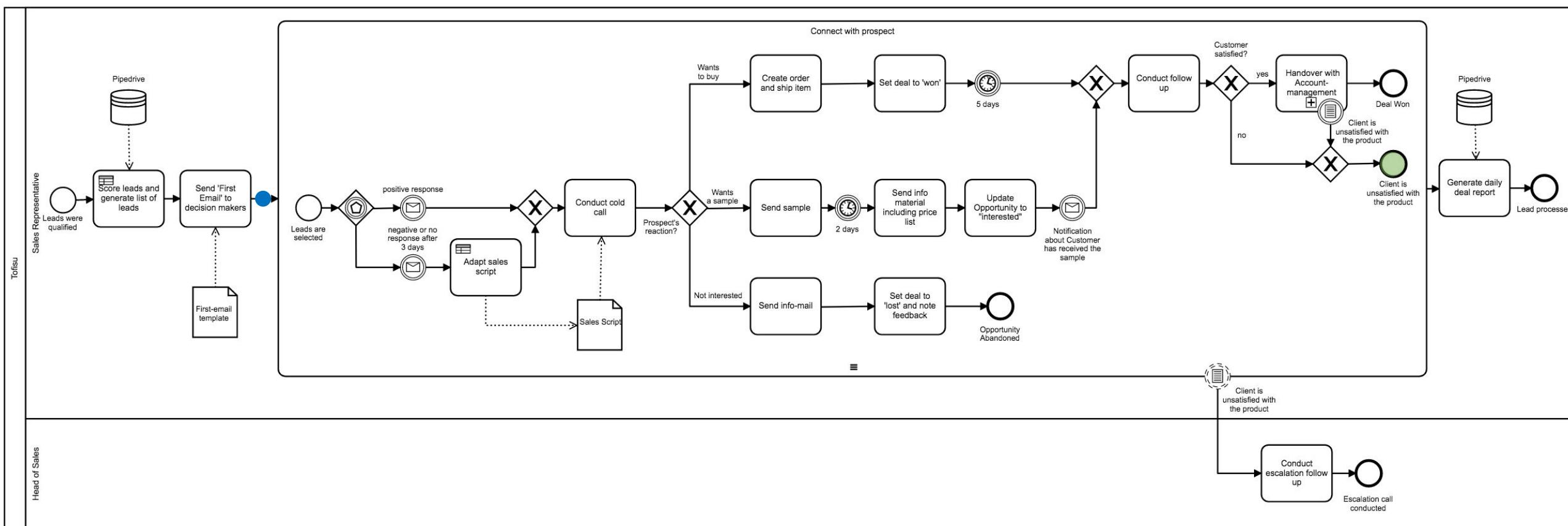


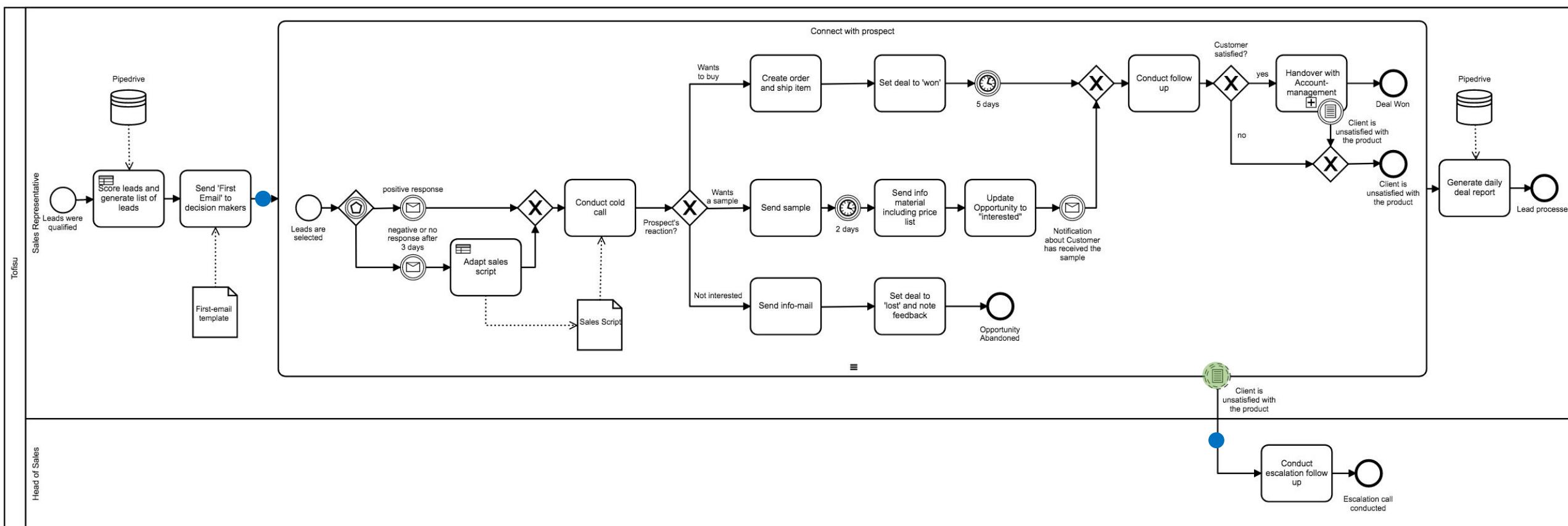


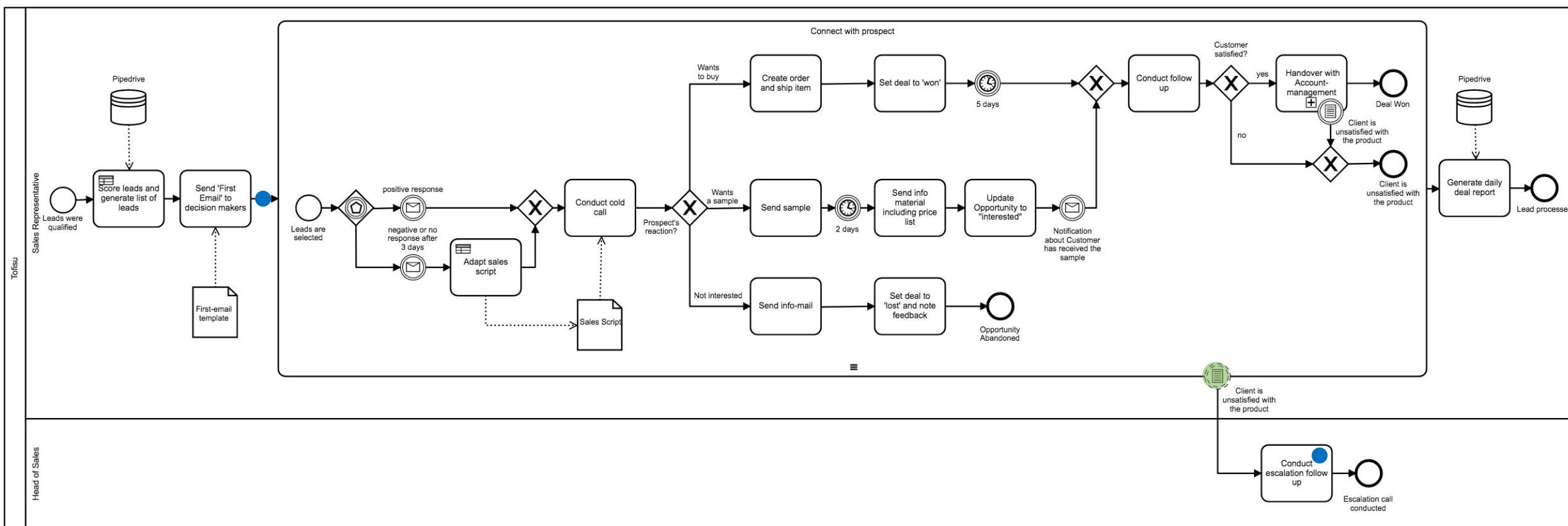


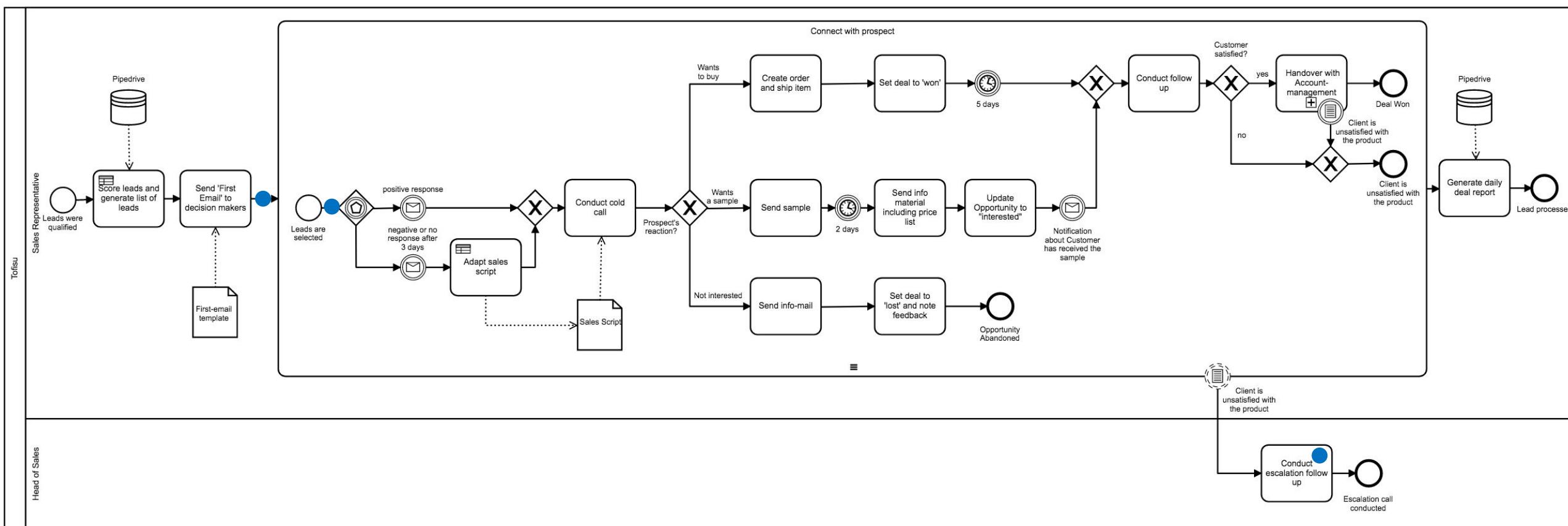


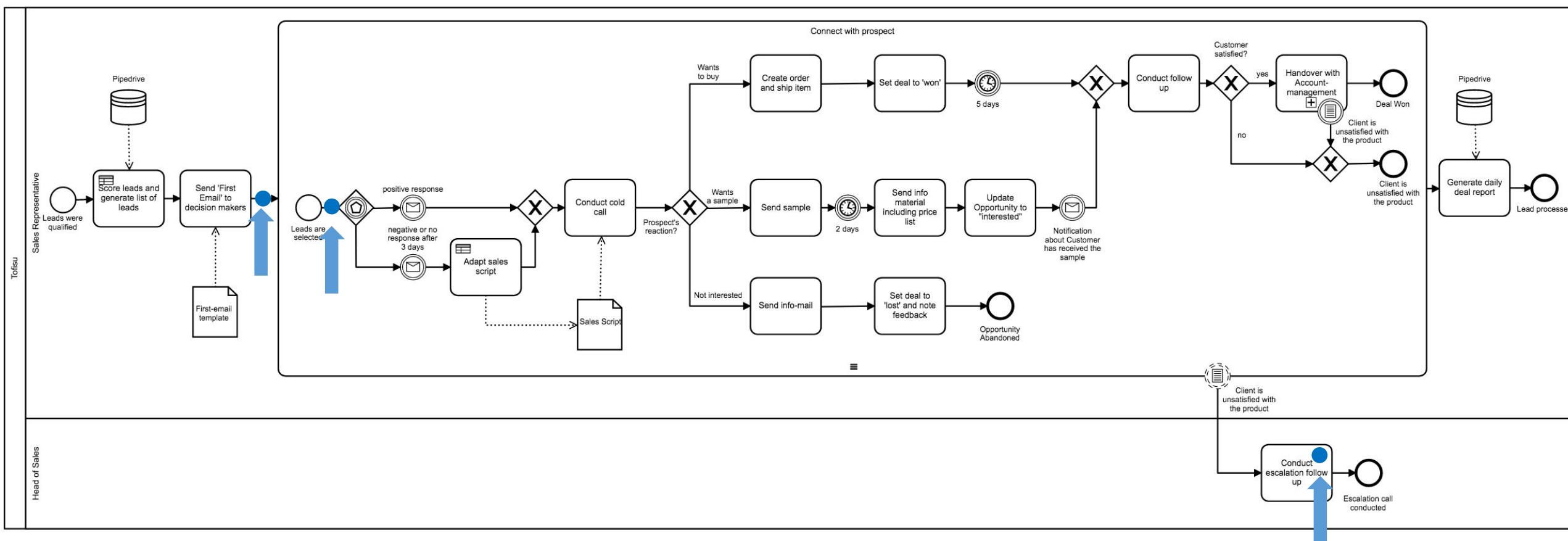


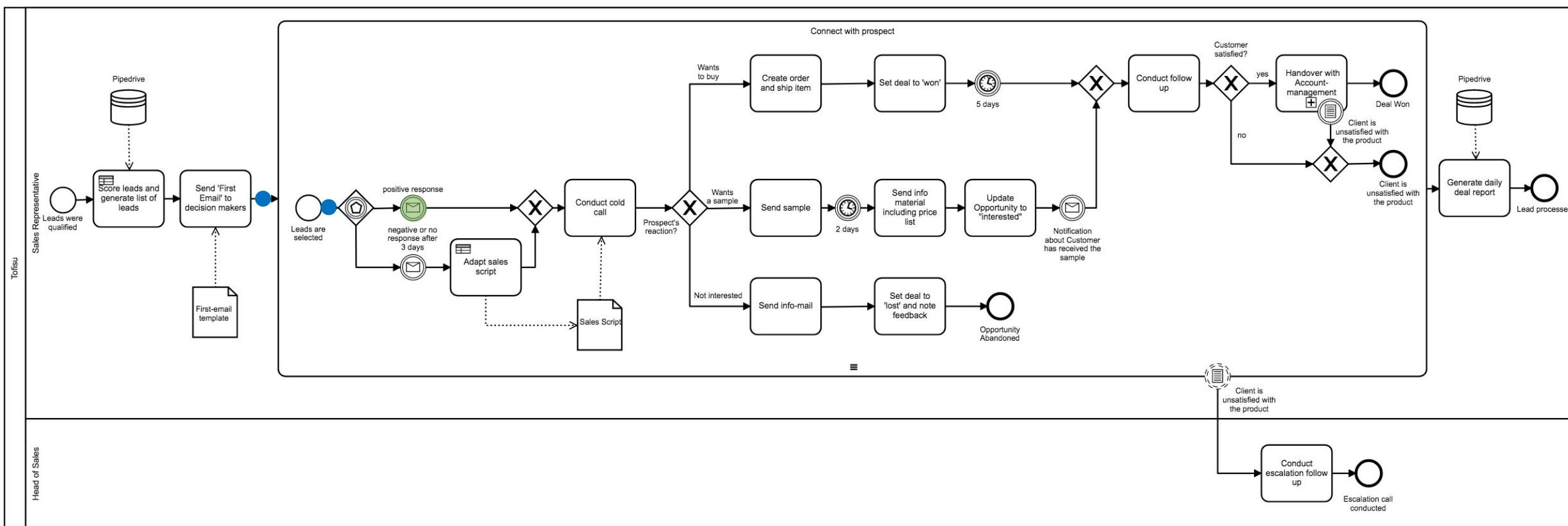


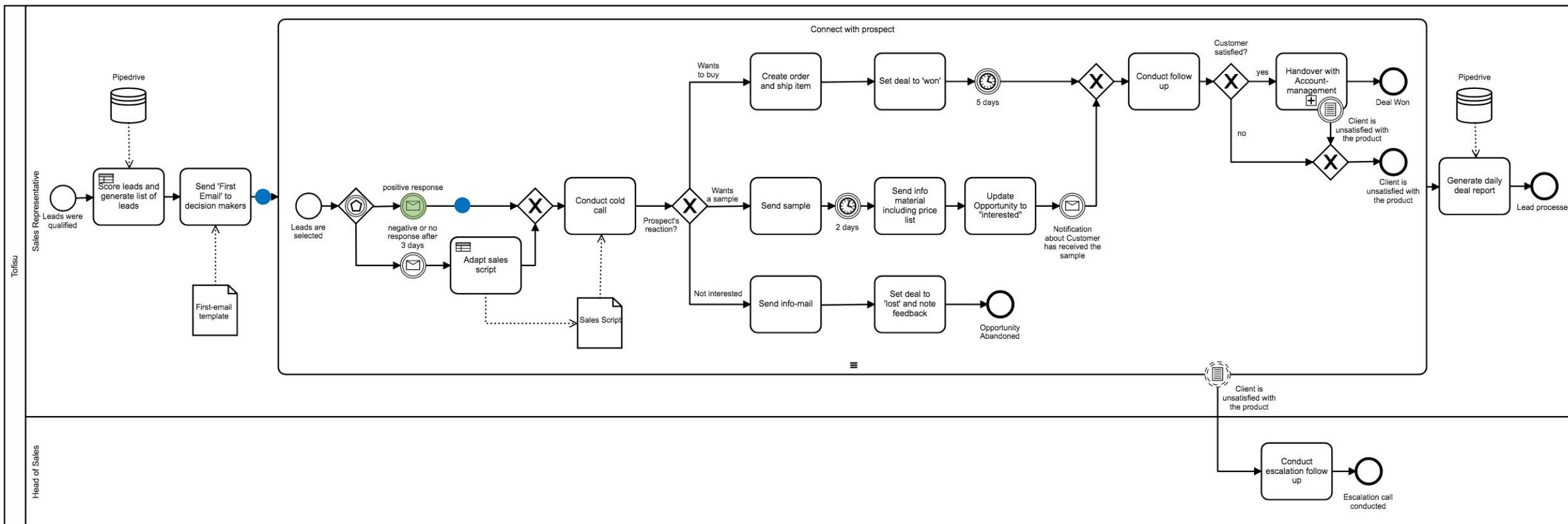


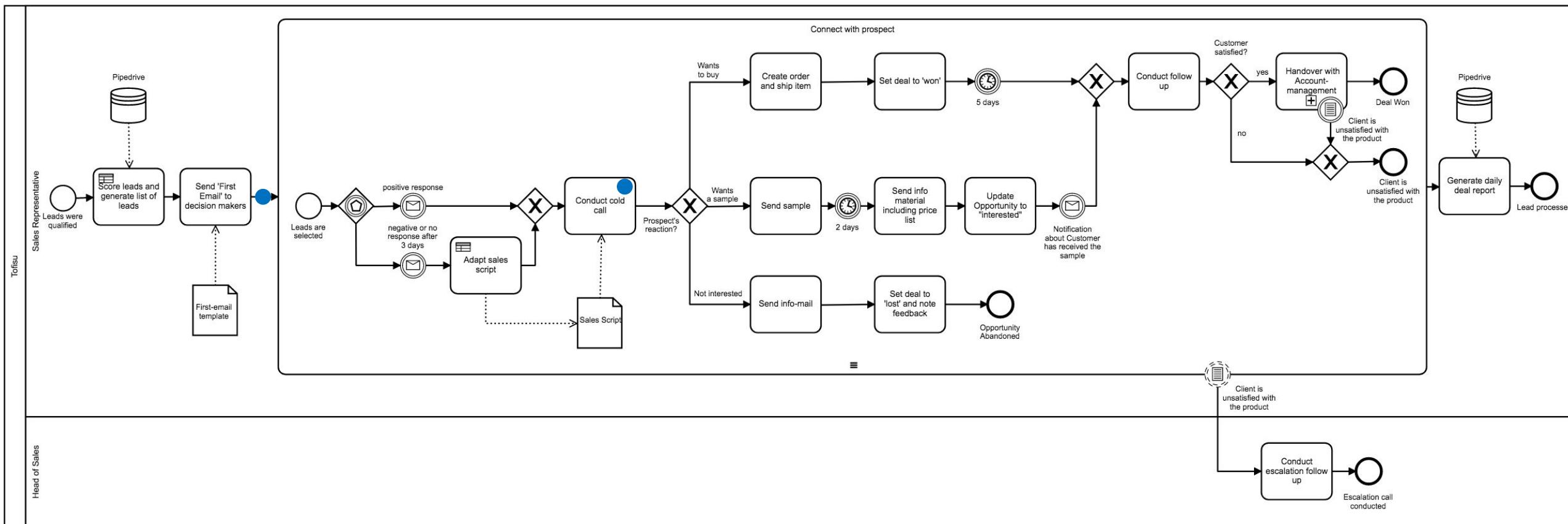


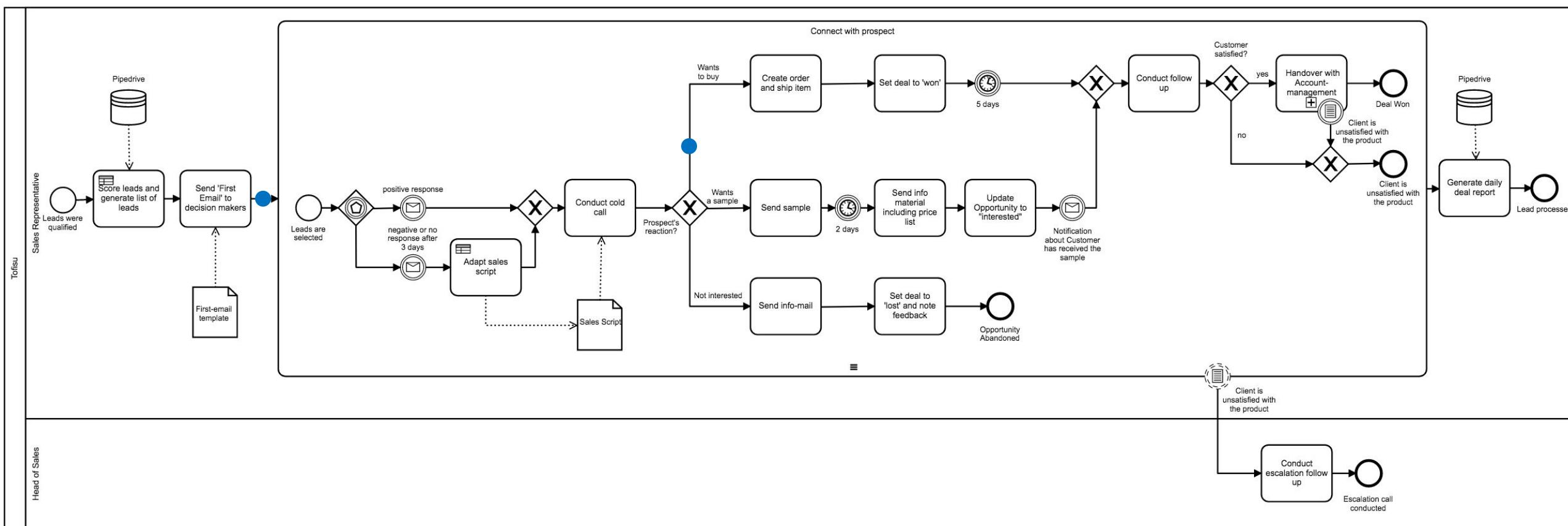


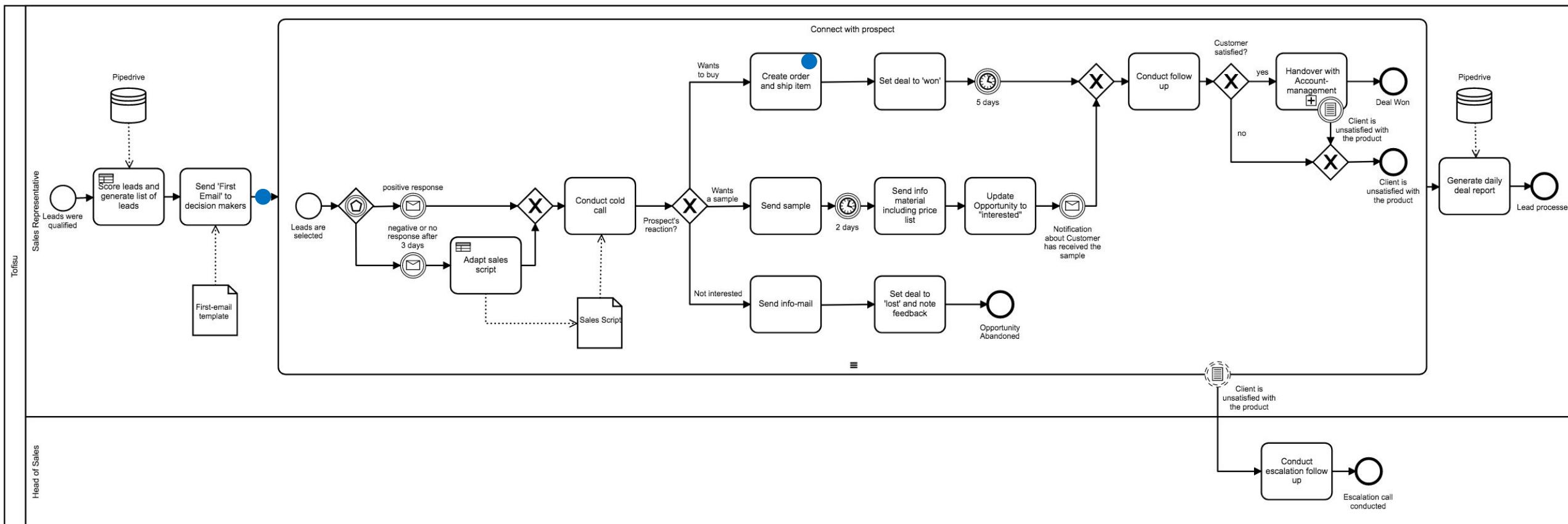


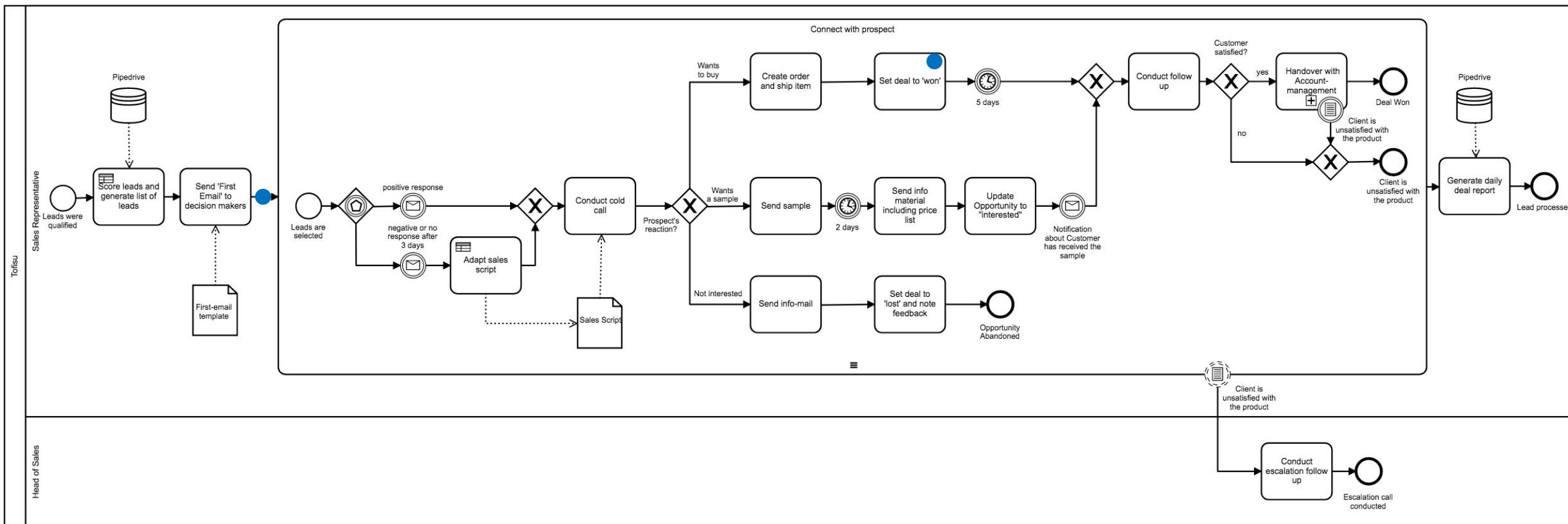


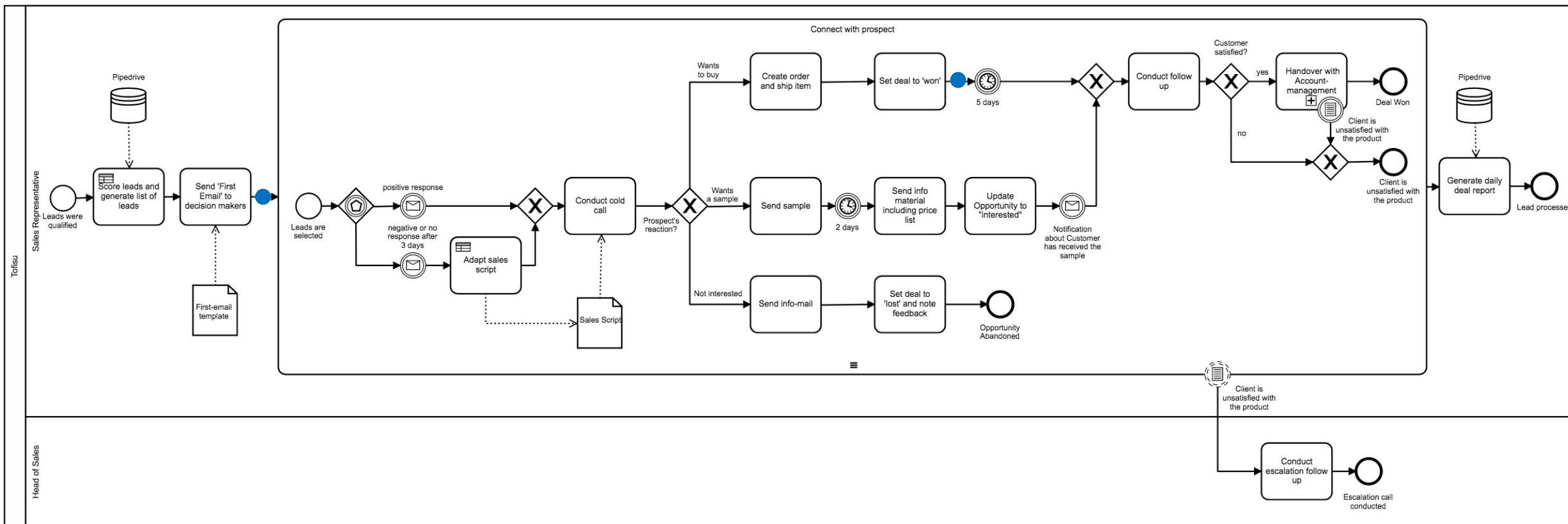


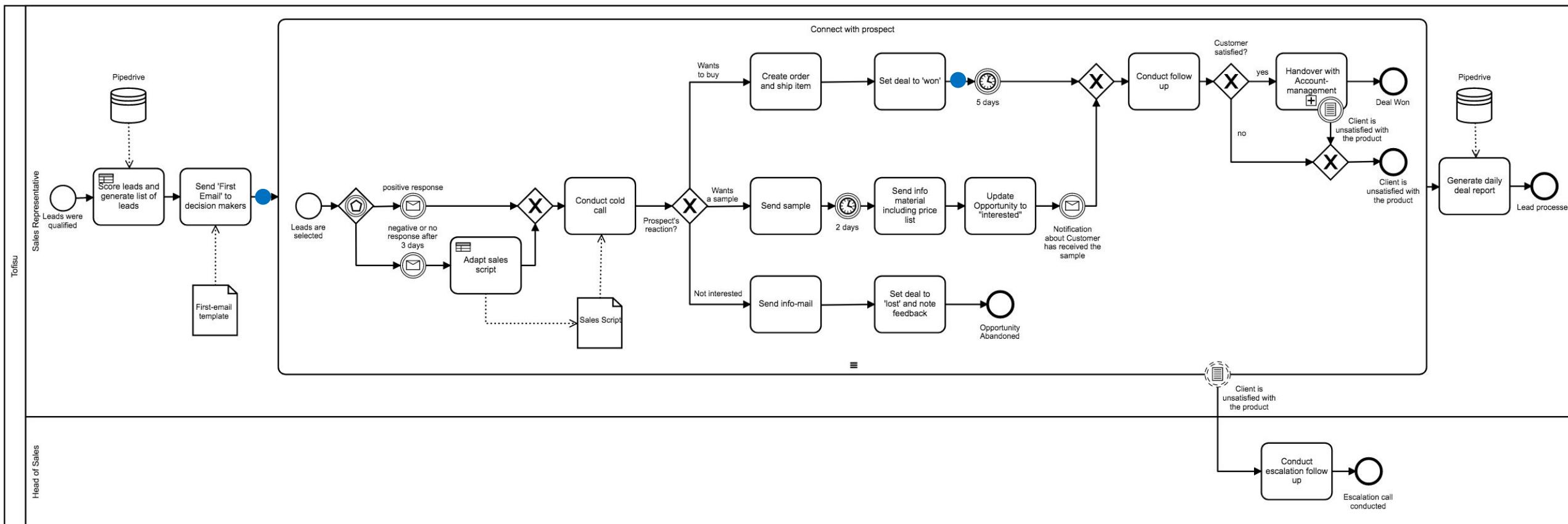


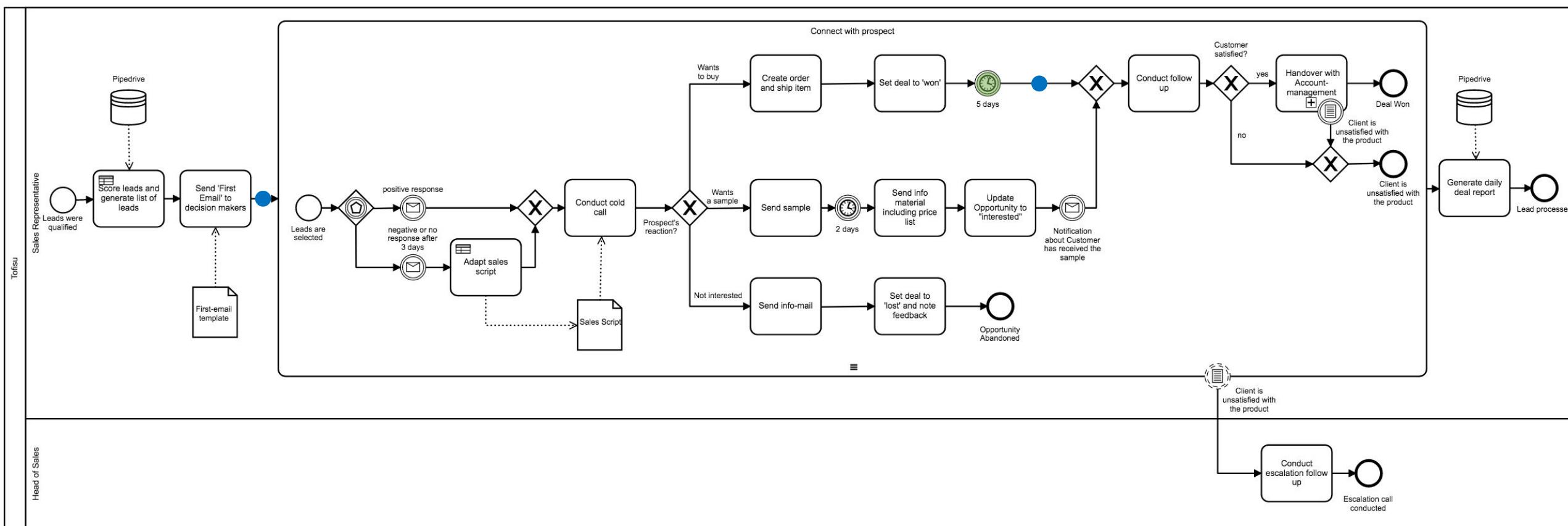


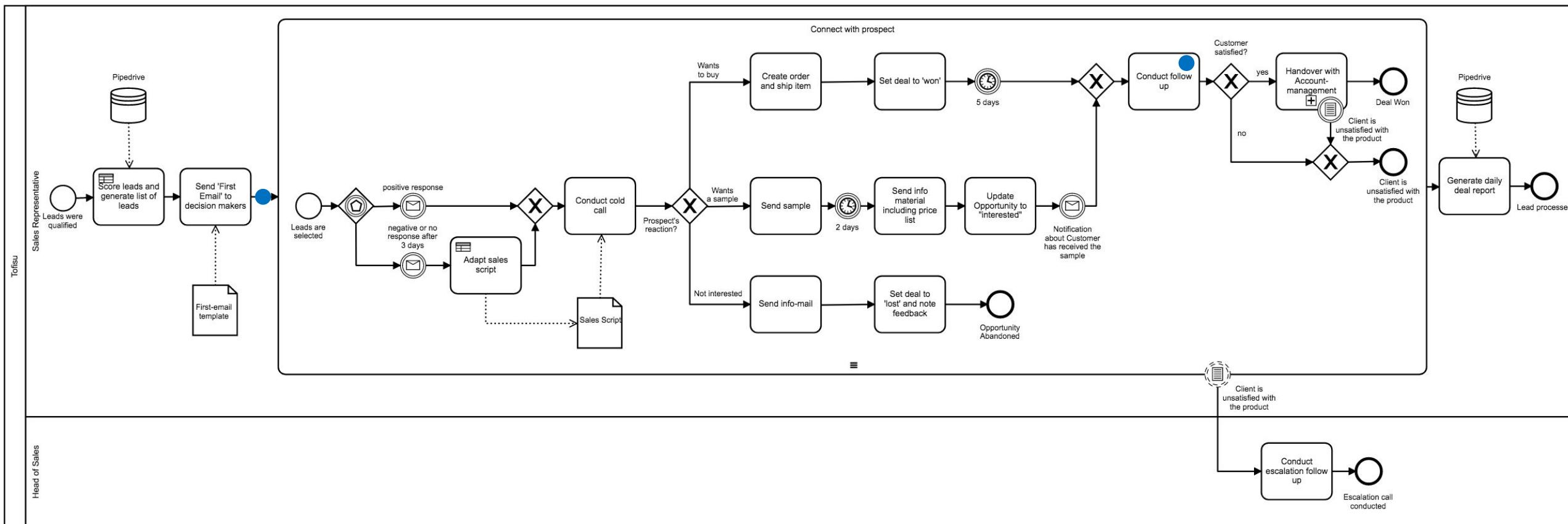


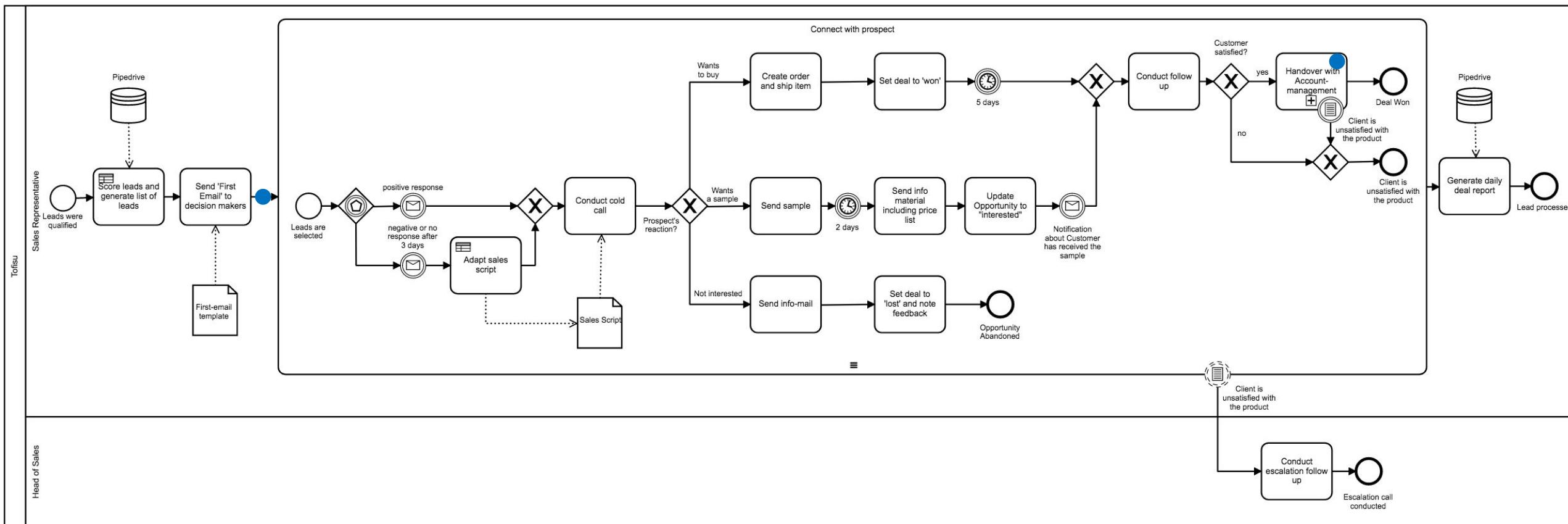


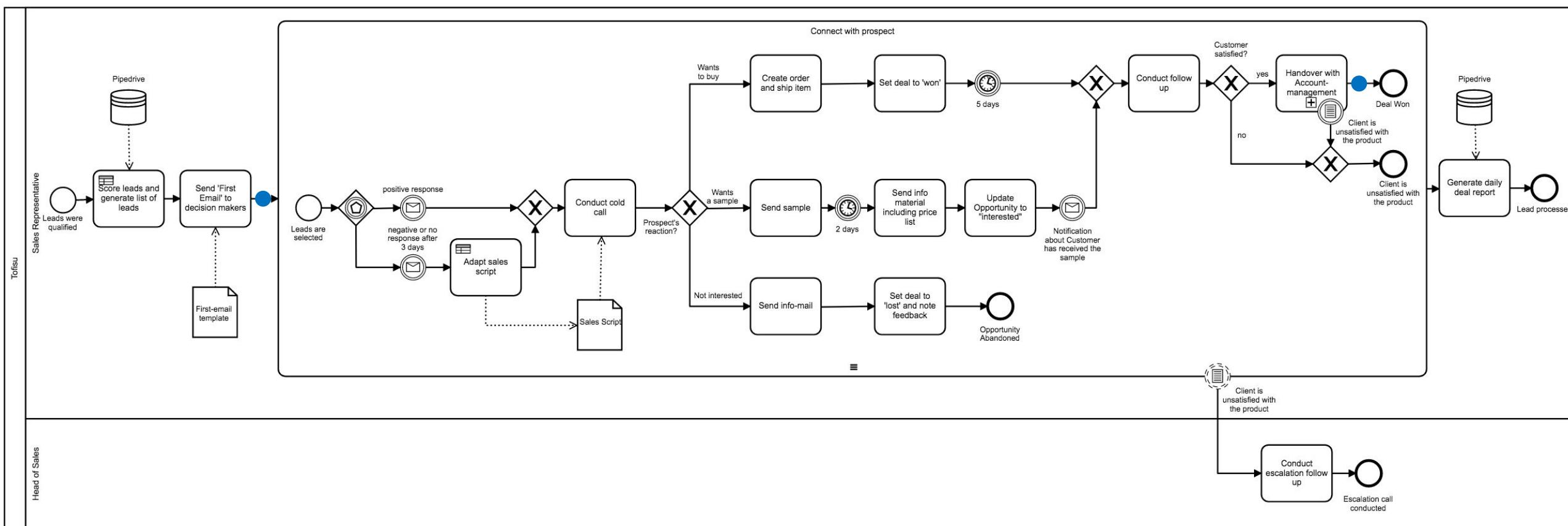


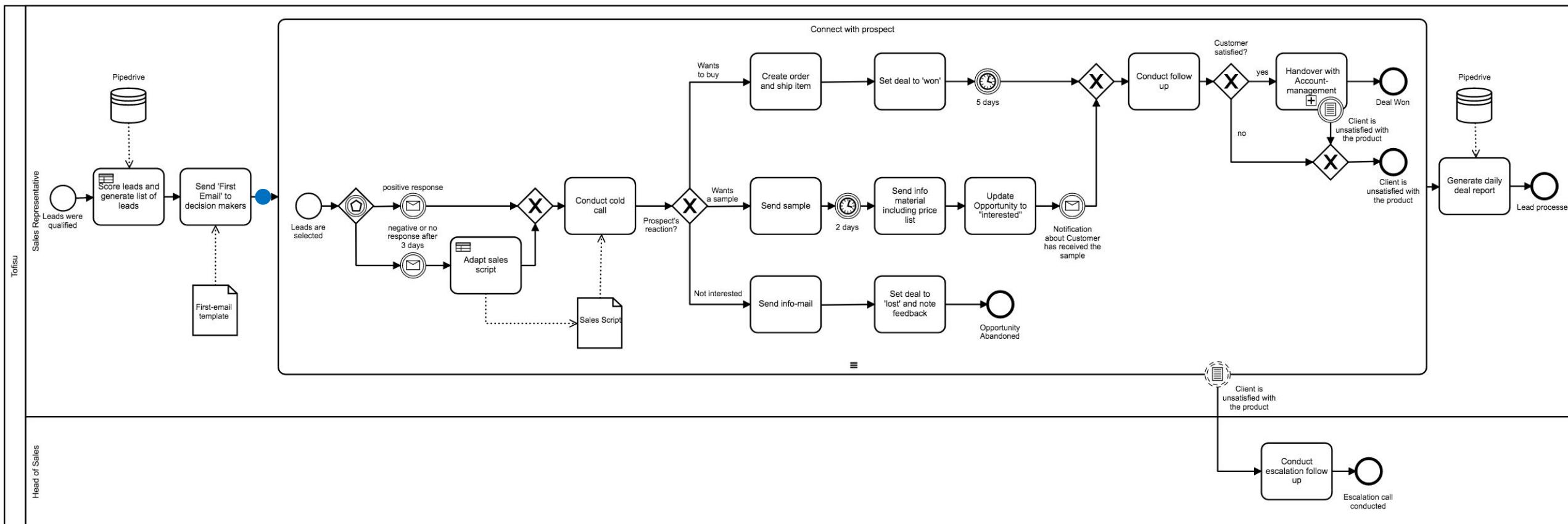


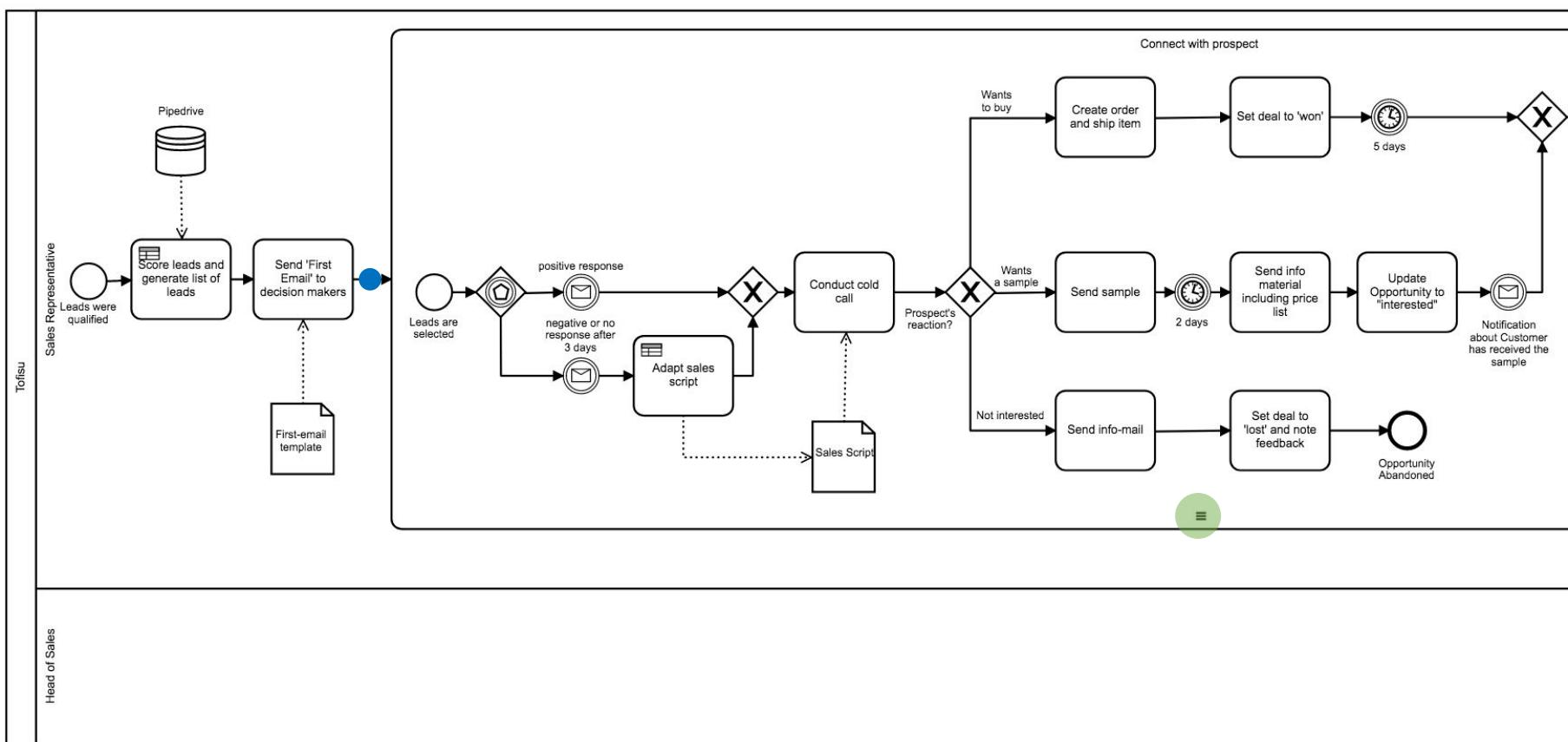




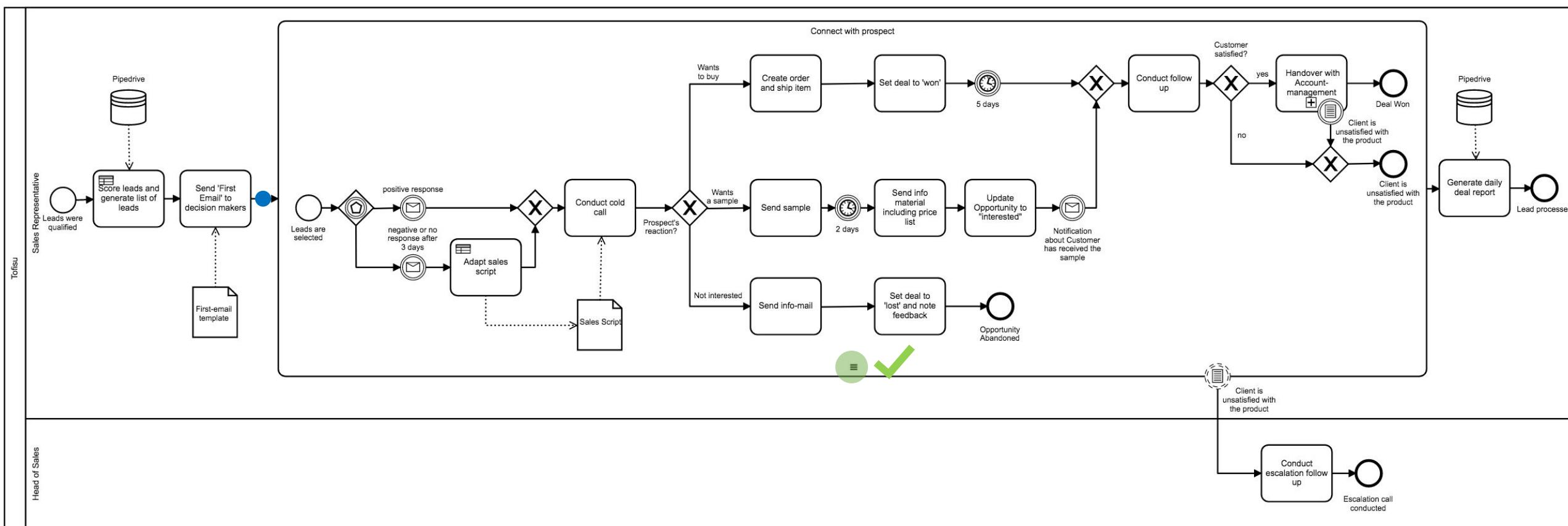


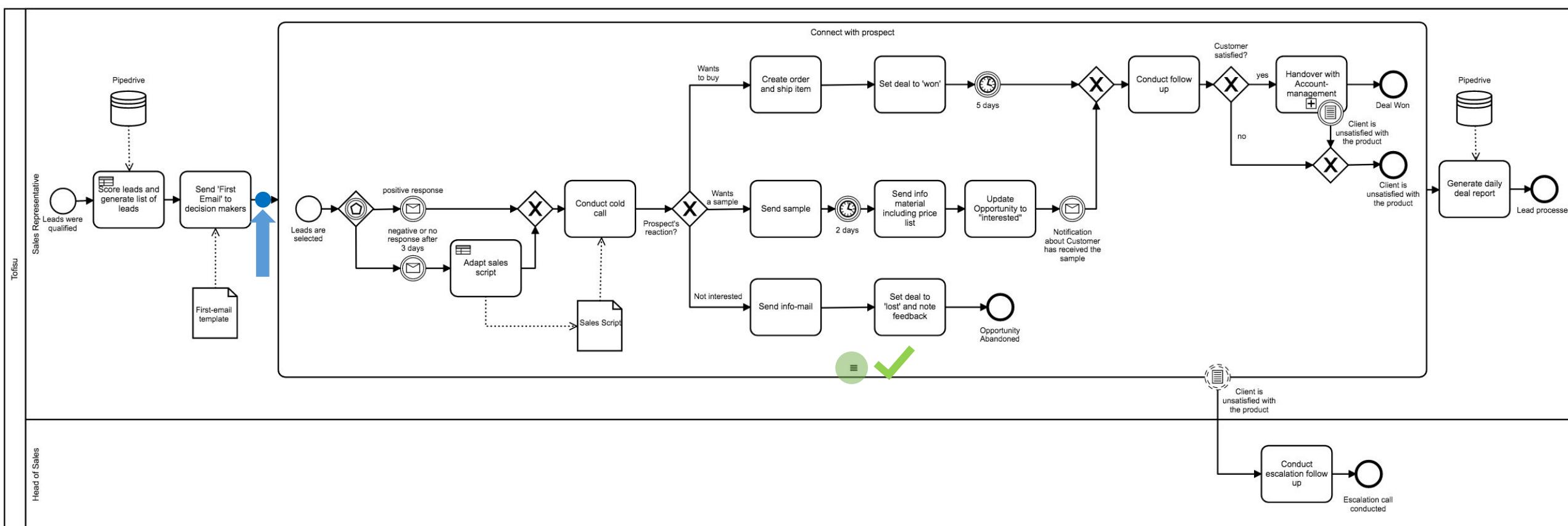


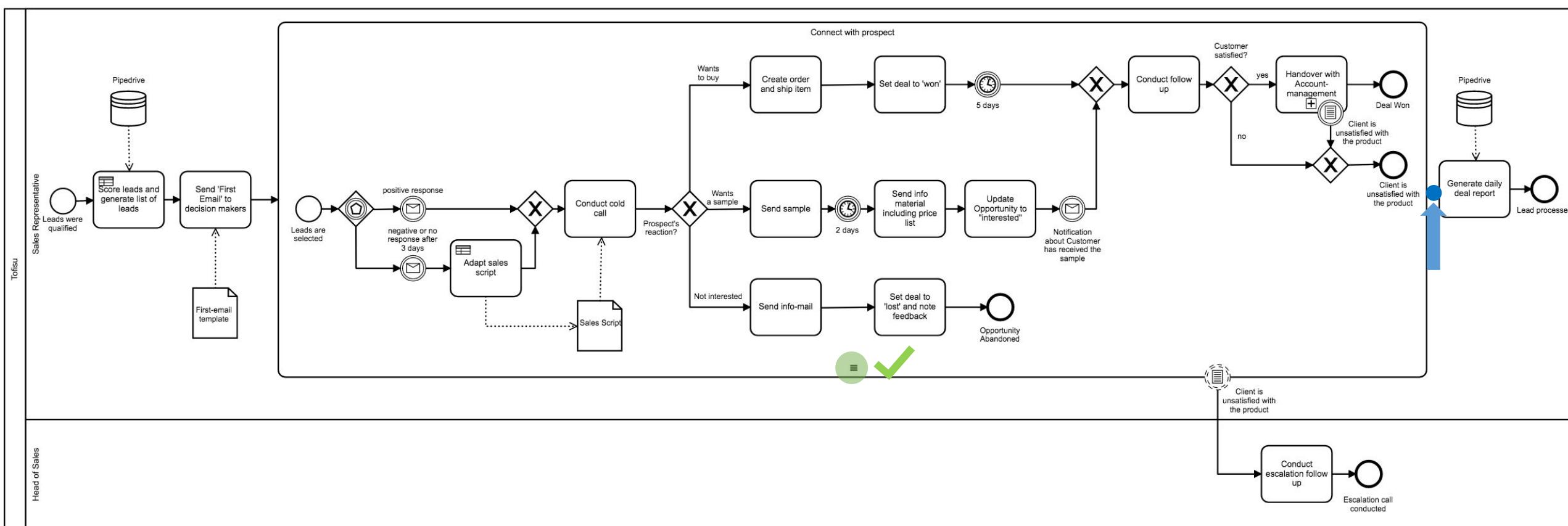


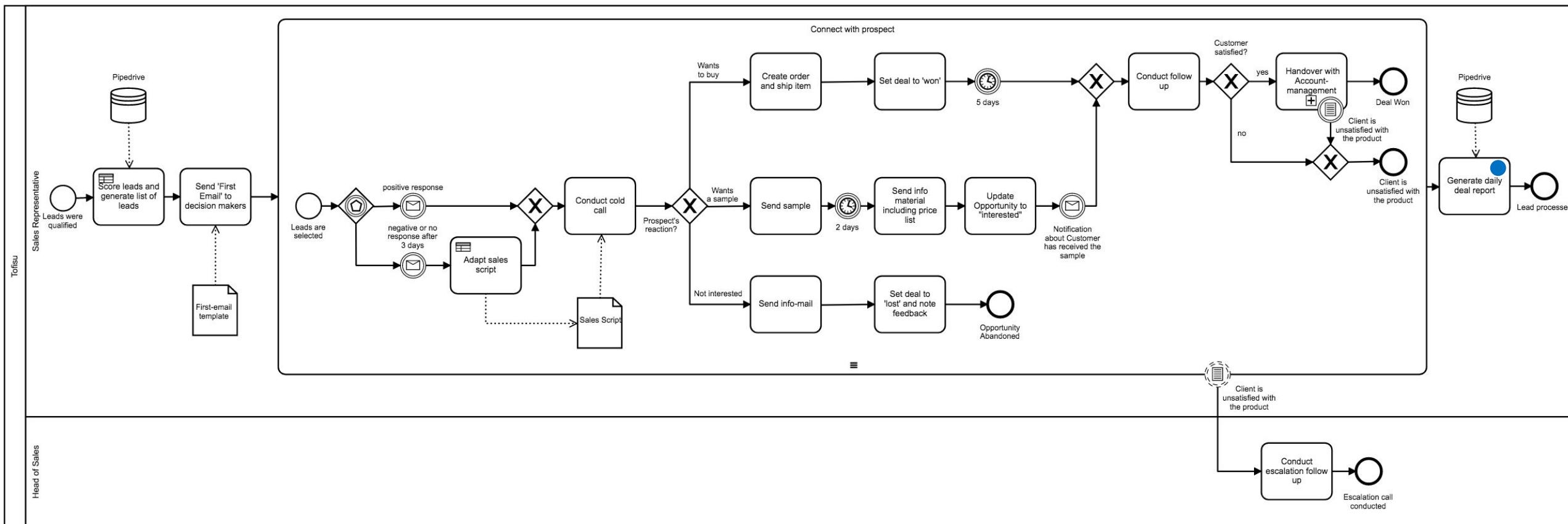


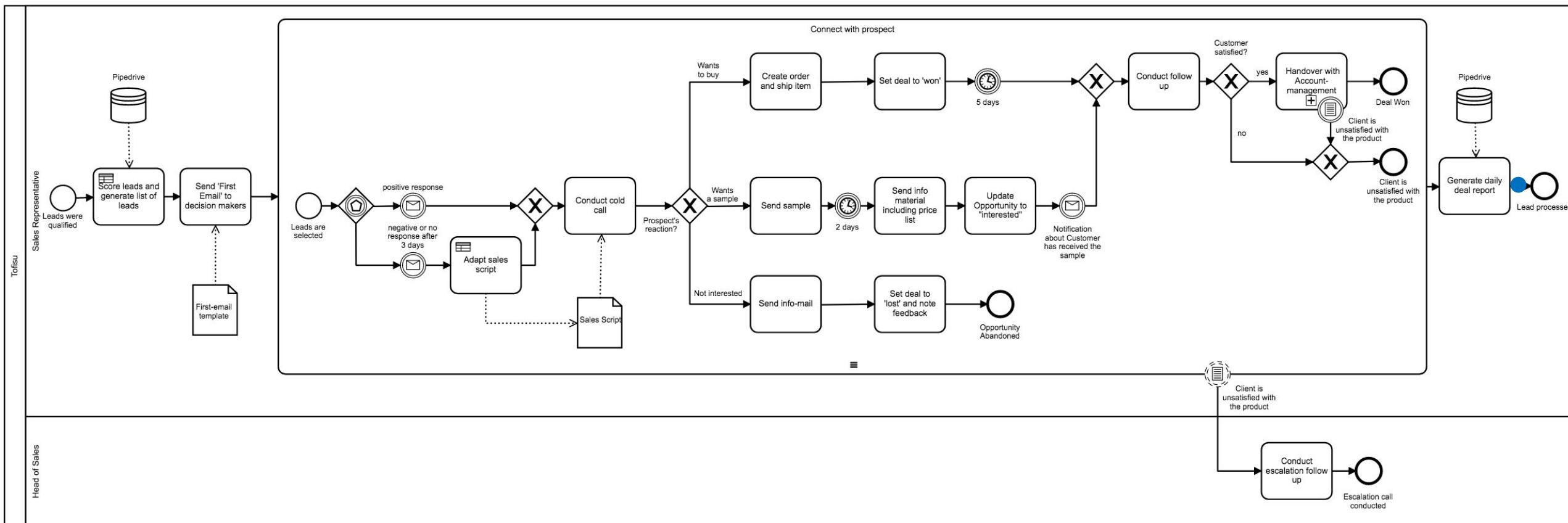
When all leads are processed, the multi instance subprocess ends

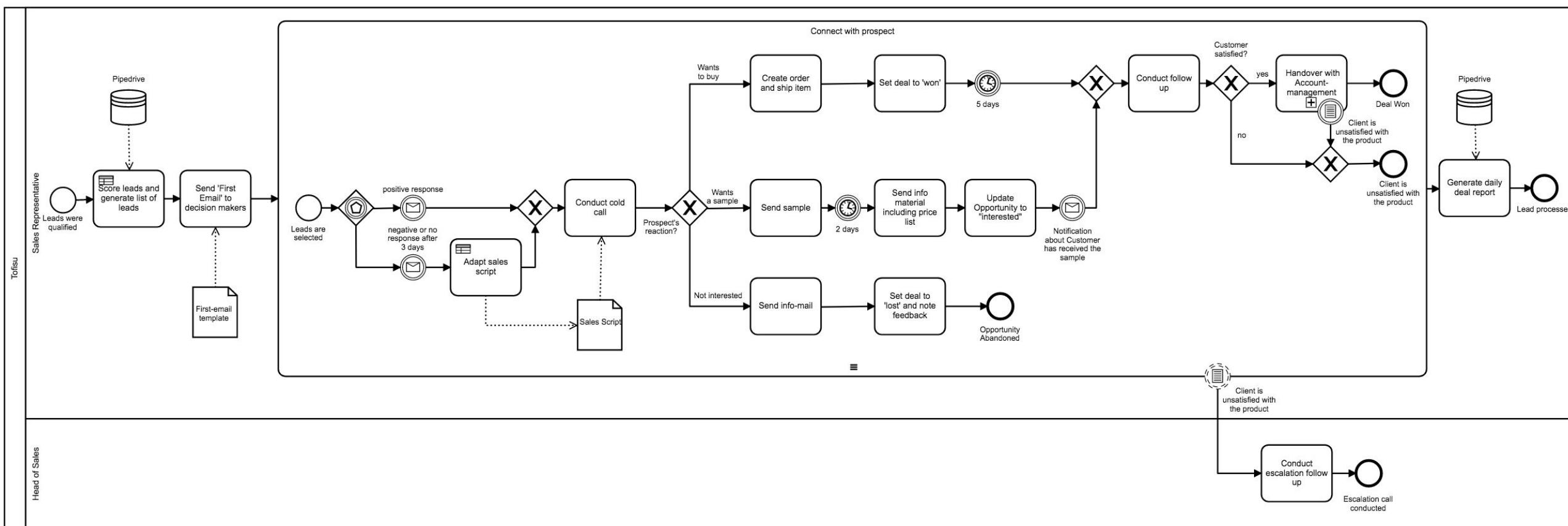






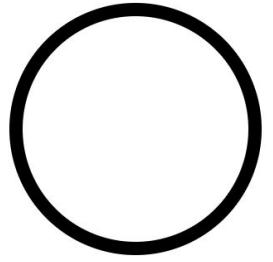




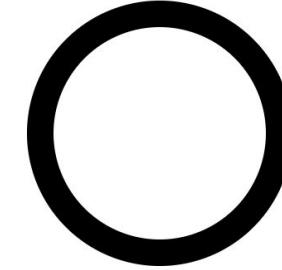




How to name processes?



Lead generated



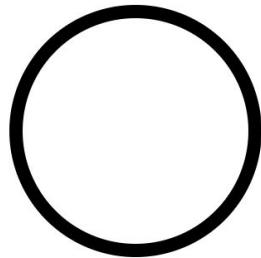
Loyal customer
won

process camp

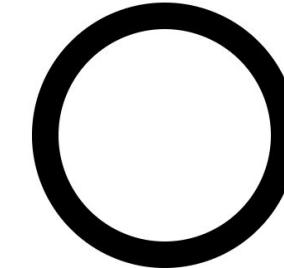
Lead

To

Loyal Customer



Lead generated



Loyal customer
won

Lead to Loyal Customer

process camp



Job to be done



Process Lead

process camp



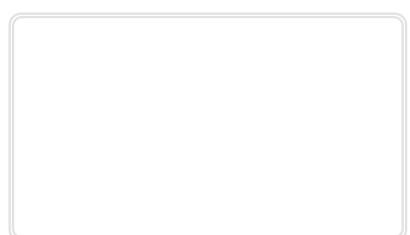
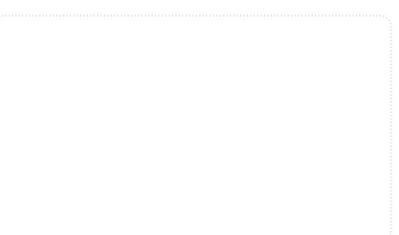
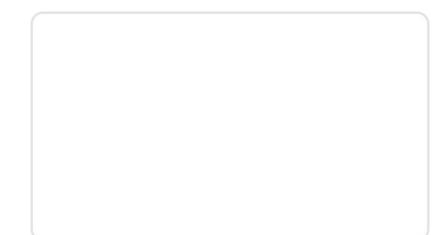
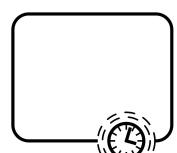
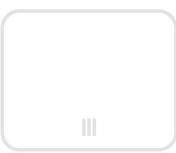
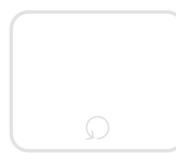
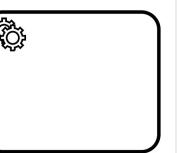
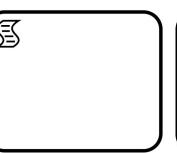
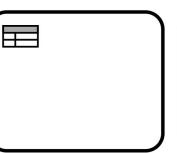
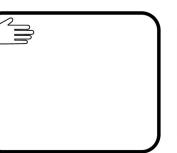
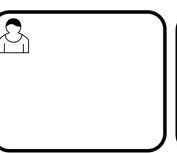
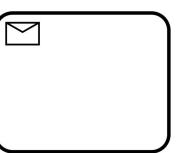
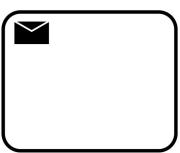
Best Practice Rule:

The process name should describe either:
The job to be done (e.g. 'Fulfil Order') or
The start to end (e.g. 'Order To Cash')

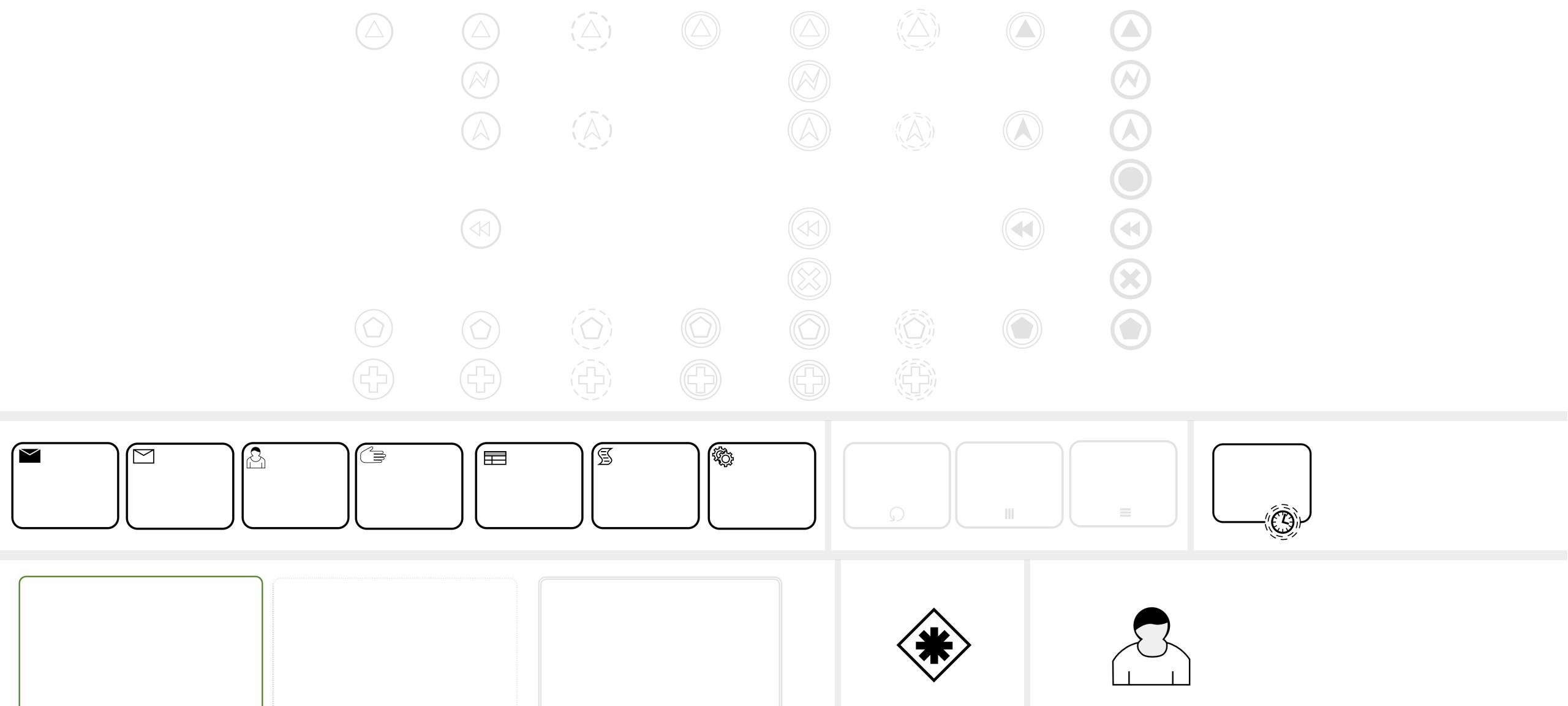


Congratulations!

process camp

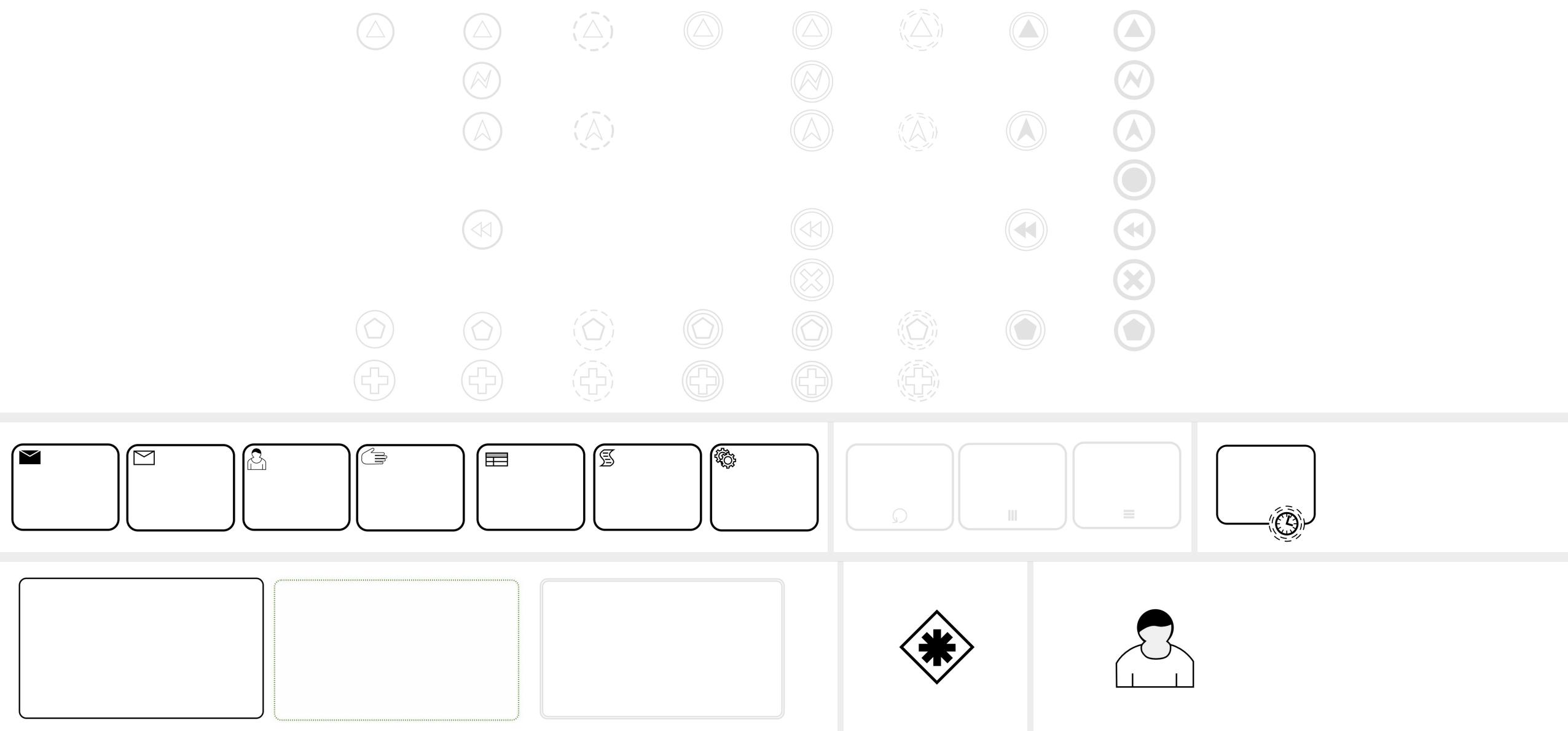


process camp



Expanded Subprocess

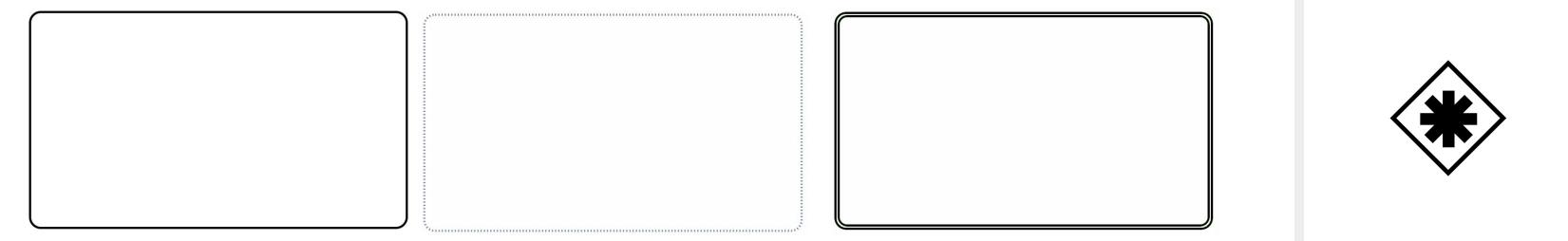
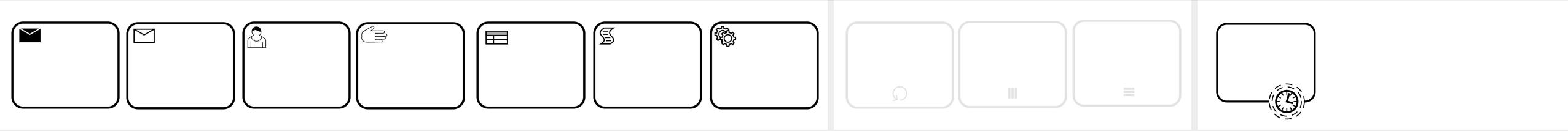
process camp



Expanded Event
Subprocess



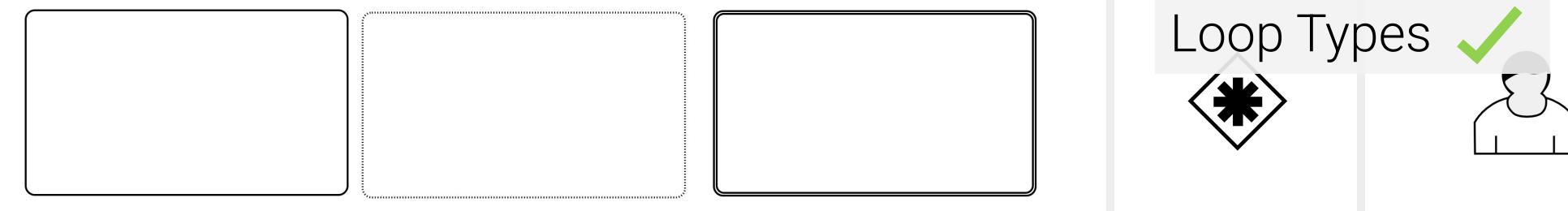
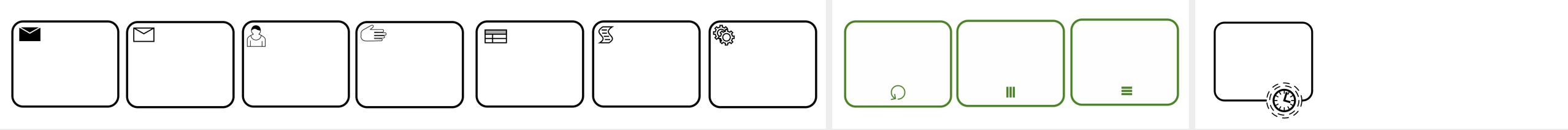
process camp



Transaction
Subprocess



process camp



process camp