

Best Practices in process modeling

Overview

The BPMN (Business Process Modeling and Notation) standard provides organizations with the capability of understanding their internal business processes in a graphical notation and the ability to communicate their procedures in a standard manner. However, the use of the standard doesn't make sure that processes are modeled in a clear and effective way; the way modelers interpret business conditions, and how they define its structure, is crucial to ensuring they are understood correctly.

This section provides process modelers some guidelines to build clear and effective models compliant with the BPMN standard.

BPMN modeling principles

When defining process diagrams you should take into account the following basic principles:

- [1. Keep a logical and clear sequence](#)
- [2. Use the BPMN standard](#)
- [3. Use strict labeling](#)
- [4. Simplify diagrams](#)

Apply process patterns

Do not reinvent the wheel. BPMN experts have worked on defining modeling patterns to different business situations. Use them to model the required business conditions while simplifying your diagrams.

For further information about modeling patterns please check the [BPMN Workflow patterns document](#).

Below you will find useful tips to follow these principles and aid the correct processes definition and communication.

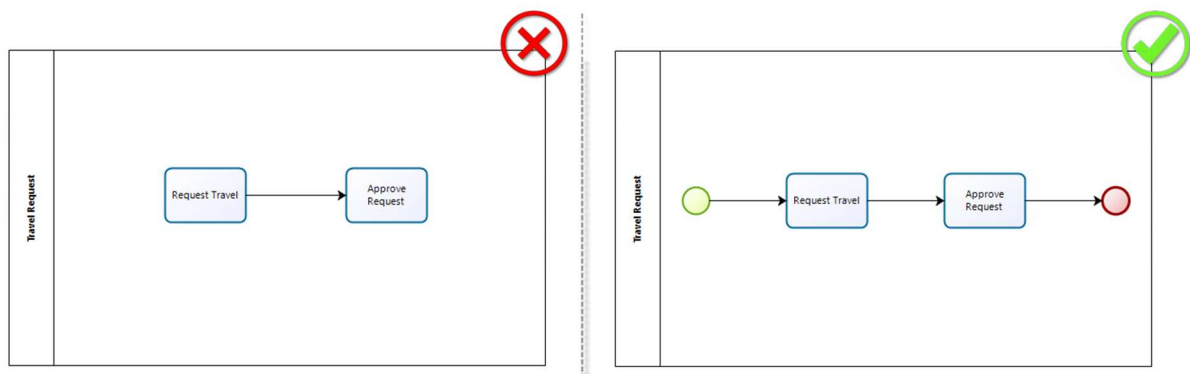
1. Keep a logical and clear sequence

This seems to be obvious but is one of the most common errors in process modeling. Diagrams can become unreadable and very confusing when the process logic is not explicit and clear. The following techniques will help you to maintain a logical and clear sequence in your models.

Define a clear beginning and end

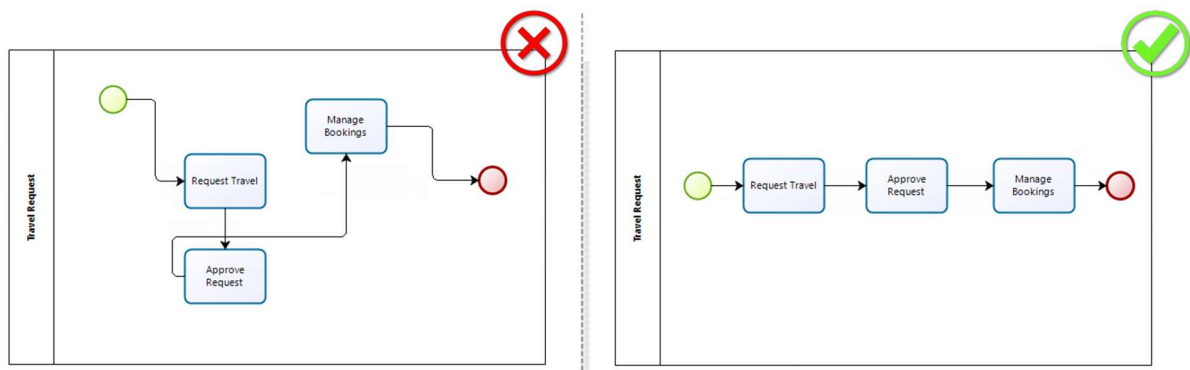
In BPMN, start and end events are optional. However, processes with implicit start and end events are undesirable and could lead to misinterpretations.

Use start and end events in each process and Sub-Process to represent its beginning and completion.



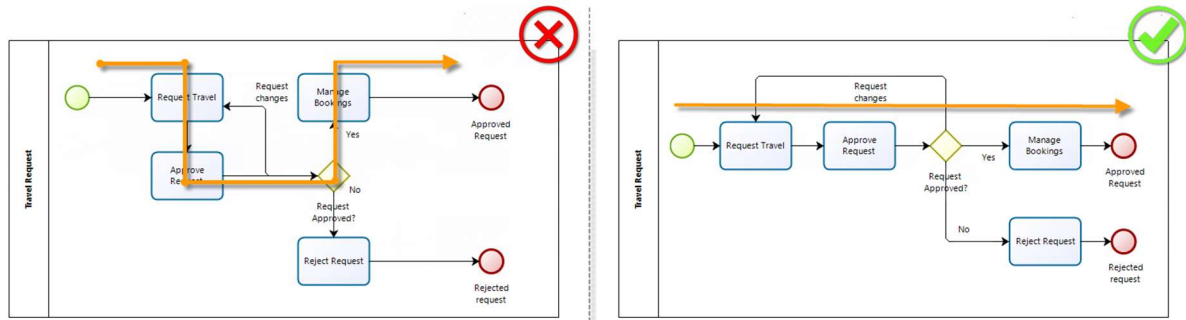
Follow a consistent direction of flow

Make the process logic visible in the diagram. Avoid crossed lines (connectors), maintain a time sequence and keep a consistent direction of flow. The diagram reading will be easier and its communication efficient.



Keep primary scenario clear

The "happy path" should be easily identified when reading a diagram. Diagram the happy path first and then the alternative flows.



Keep alternative scenarios clear

BPMN offers the necessary tools to represent exception handling logic explicitly in the diagram. Once the primary scenario is diagrammed, make use of the following elements to model alternative flows as required:

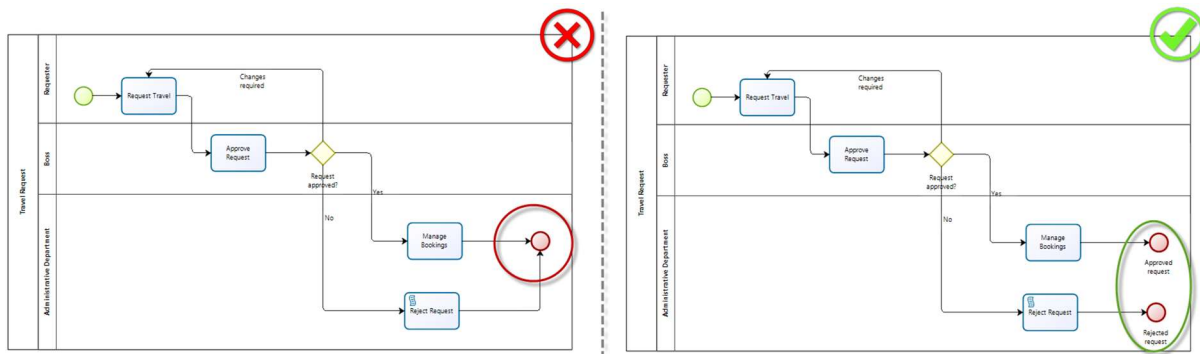
Use transactional processes

Transactional processes allow business scenarios with transactions. A set of activities must be successfully accomplished, otherwise compensation or cancellation flows are followed. For further information see [Sub-Processes types](https://help.bizagi.com/bpm-suite/en/index.html?best-practices-in-process-mode.htm)

Distinguish success and failure end states

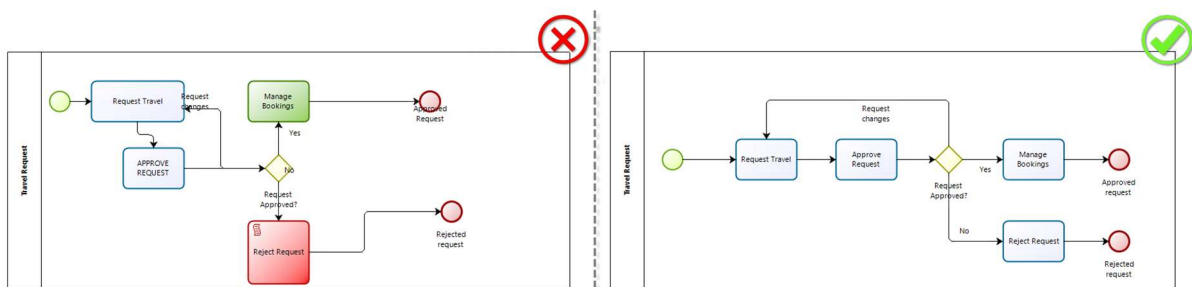
Use separate end events to identify when a process finished successfully and when it did not, for documentation and review purposes.

BPMN – Best Practices in process modeling



Keep a format standard

Keep a unique format along your diagrams and focus on a clean and friendly look and feel. Using different font sizes, colors, boxes sizes or overlapping labels might make the diagrams reading a challenge.



2. Use the BPMN standard

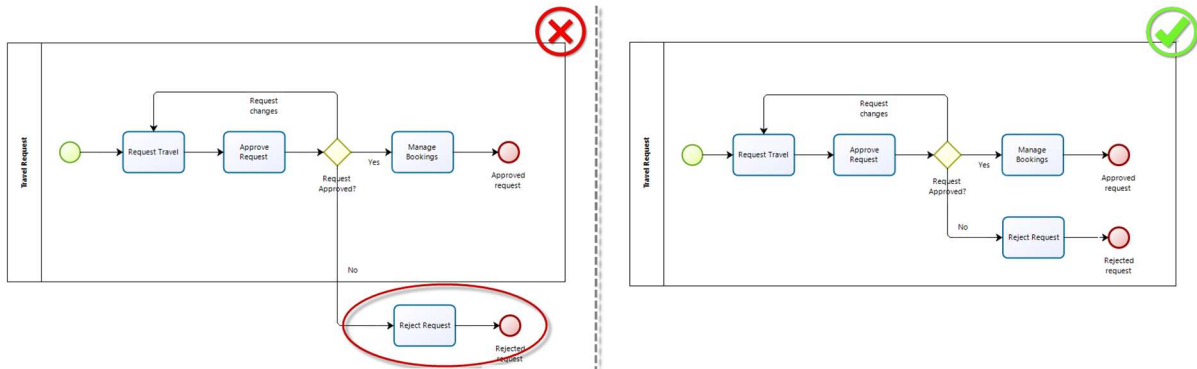
The BPMN standard defines the standards used to diagram business processes. However, following the BPMN guidelines is completely in your hands. Make sure your models comply with the standard to make sure its correct understanding.

Once the process logic has been defined, validate your diagrams making sure you properly use the different BPMN elements. The following aspect should be checked for each BPMN element:

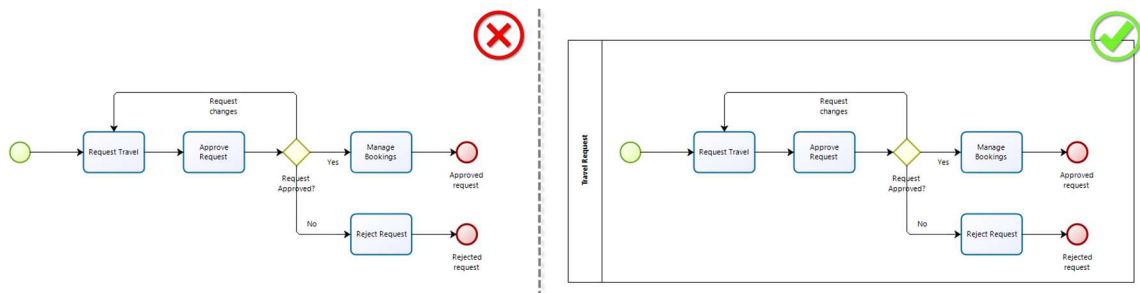
What to check in Pools

BPMN – Best Practices in process modeling

- Diagram processes completely within a Pool. Never diagram flows across Pool boundaries.

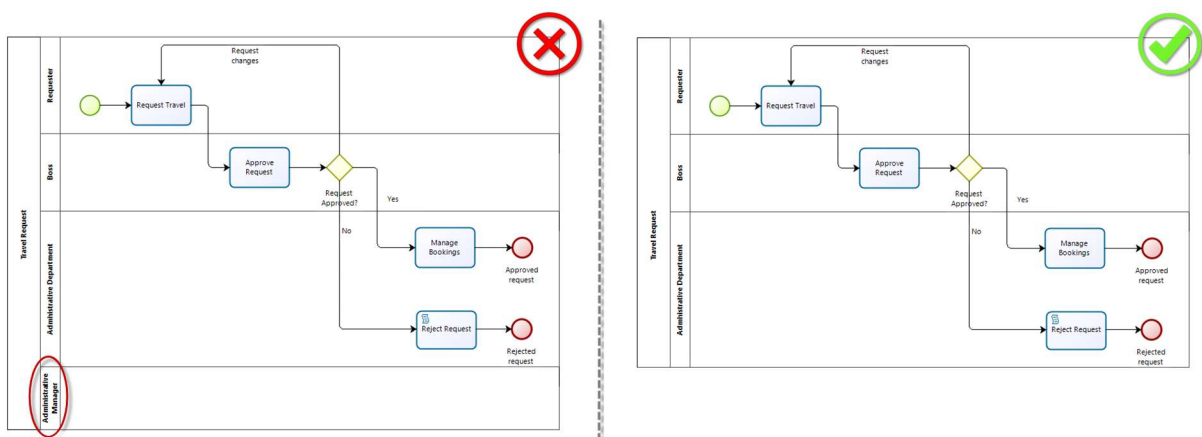


- Define as many pools as processes. There must be always at least one Pool.



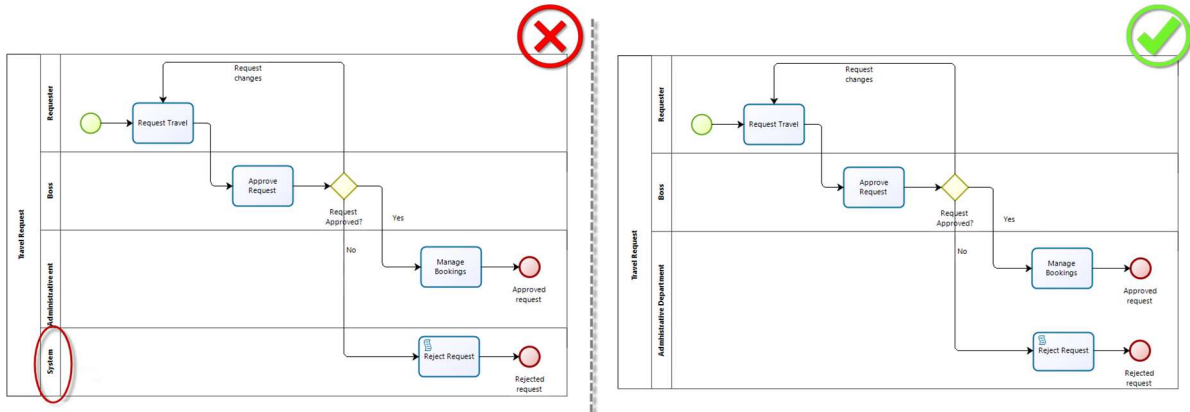
What to check in Lanes

- Create a lane only if at least one task or intermediate event is performed on it.

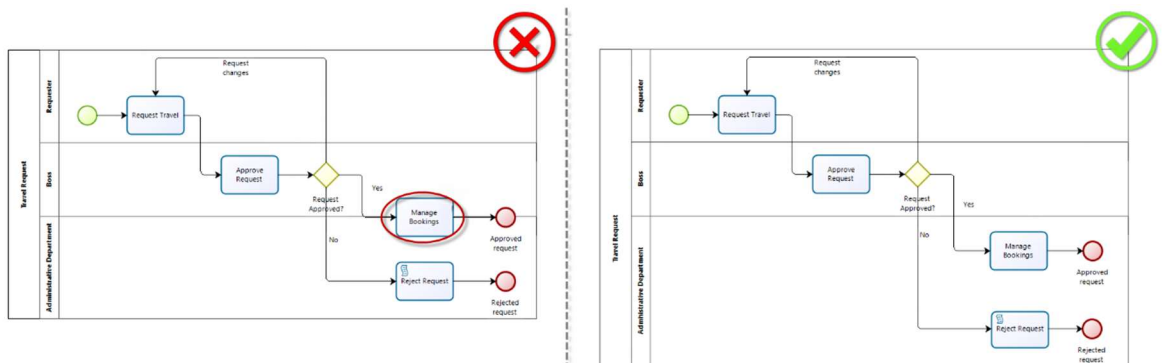


BPMN – Best Practices in process modeling

- Do not create lanes to represent the area or entity that carries out automatic tasks or gateways.



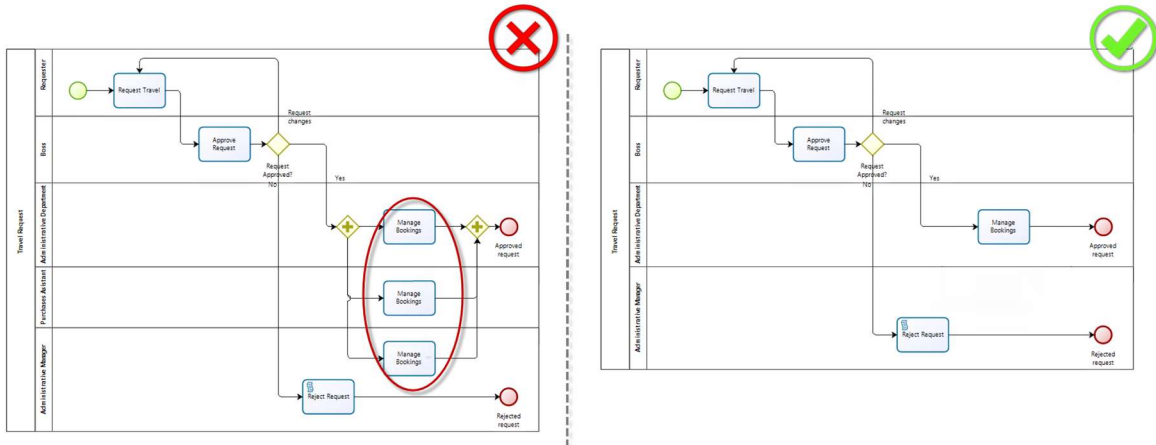
- Do not diagram tasks, gateways or events at the middle of two lanes.



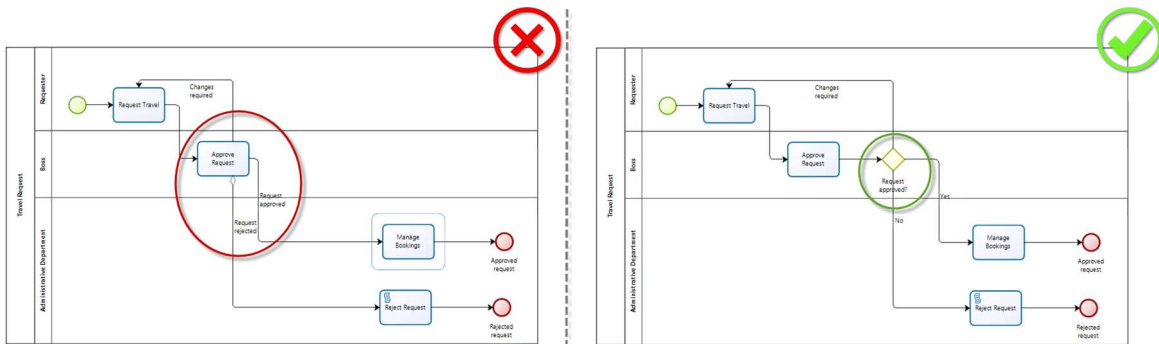
What to check in Activities

- Do not diagram multiple instances of the same task to represent multiple performers. Just diagram one task in one area. Define the multiple performers as Allocation Conditions in the documentation and [allocation rules](#).

BPMN – Best Practices in process modeling

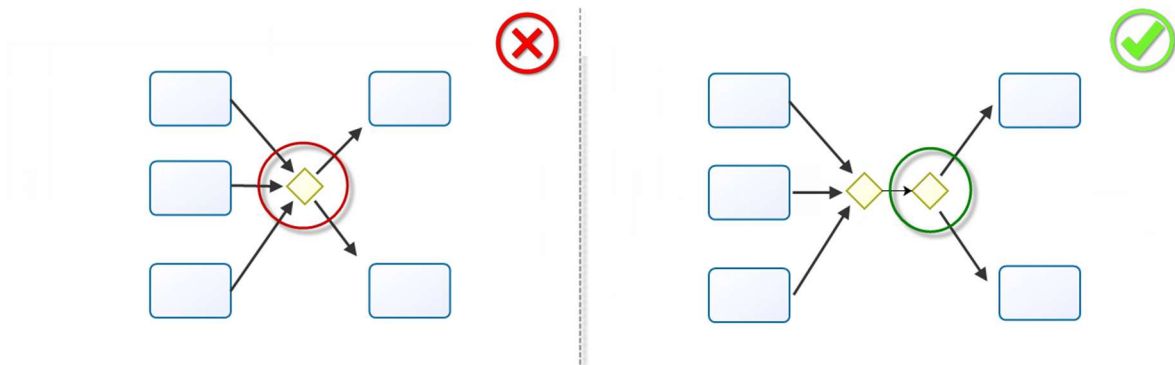


- Do not branch flows using tasks. Always use gateways to do so.



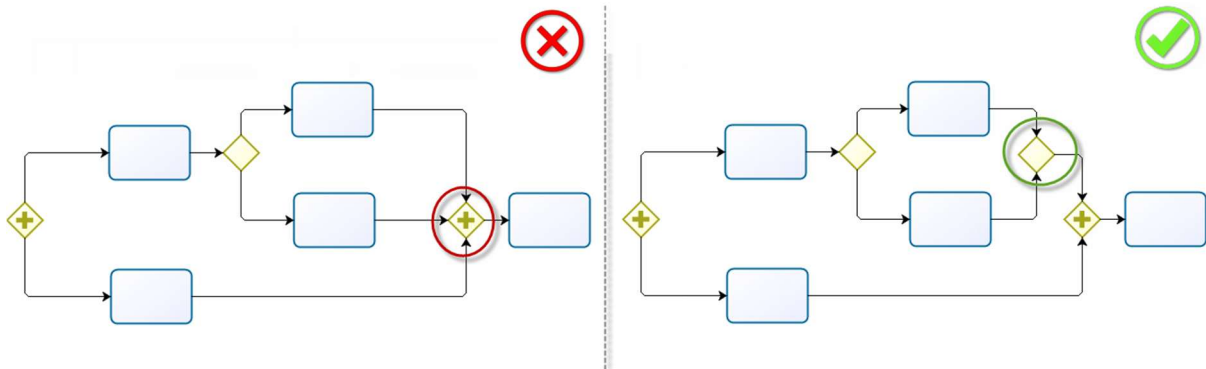
What to check in Gateways

- Do not use gateways to join and split at the same time. This will produce an error at Run-time.

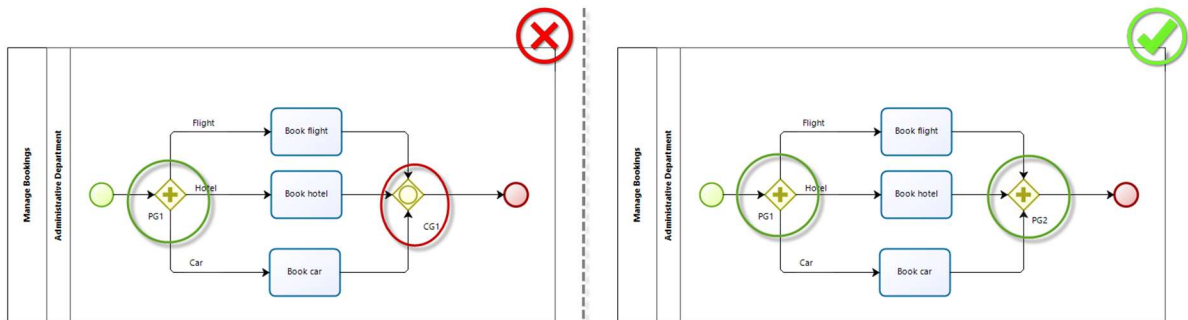


- Mandatory use of the Exclusive Gateway as a convergence element.

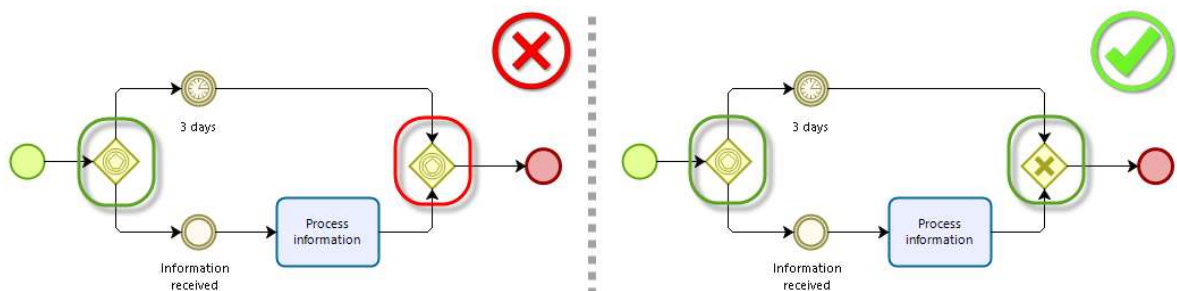
BPMN – Best Practices in process modeling



- Use the same type of Gateway used for splitting to join the flow.

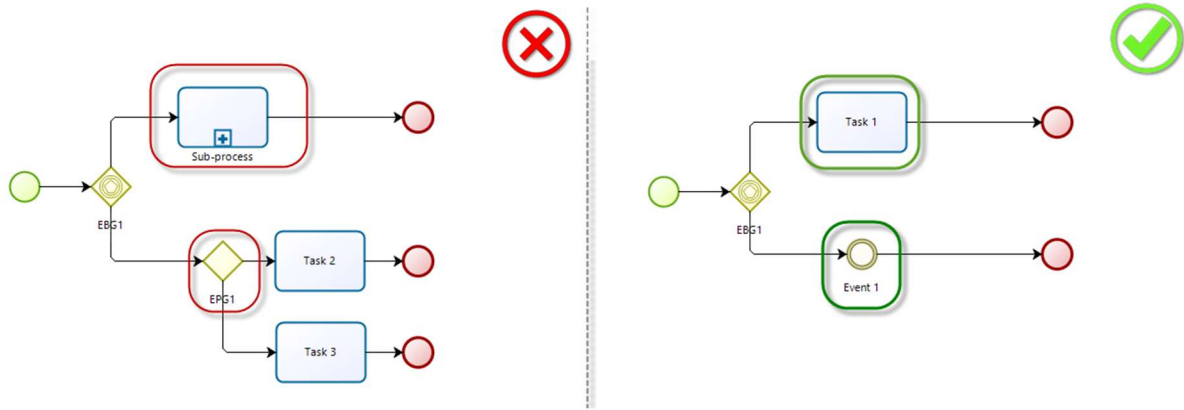


When using Event-based gateways, do not use an Event-based gateway for splitting to join the flow.



- Use only Events and/or Tasks after an Event-based gateway

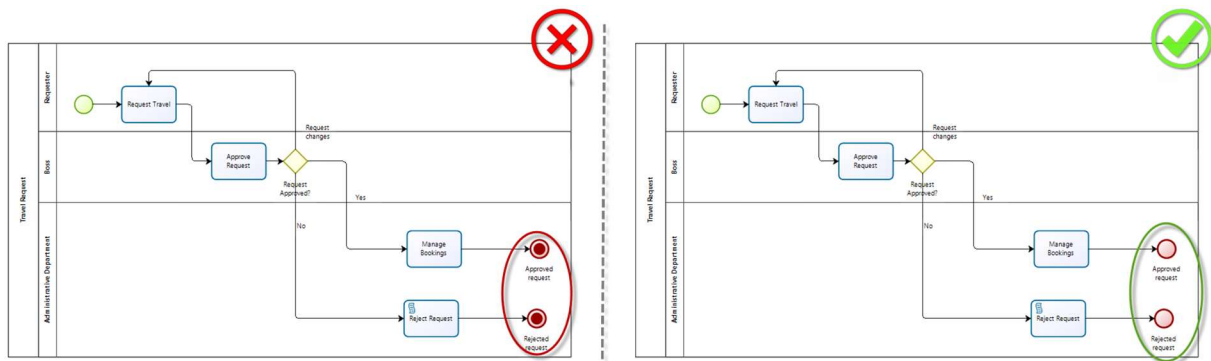
BPMN – Best Practices in process modeling



What to check in Events

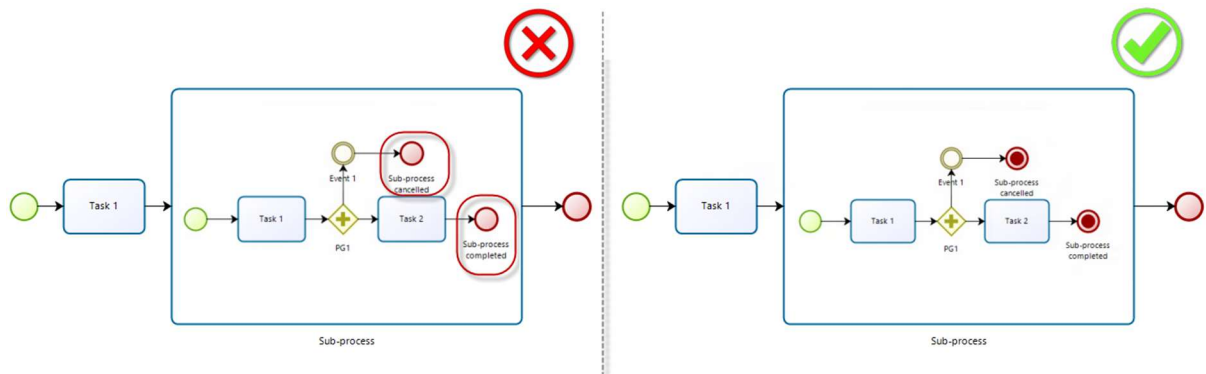
- Use terminate events only when this is strictly necessary. They are used to model situations where several alternative paths are enabled and the entire process has to be finished when one of them is completed.

This has an exception described in the next item.



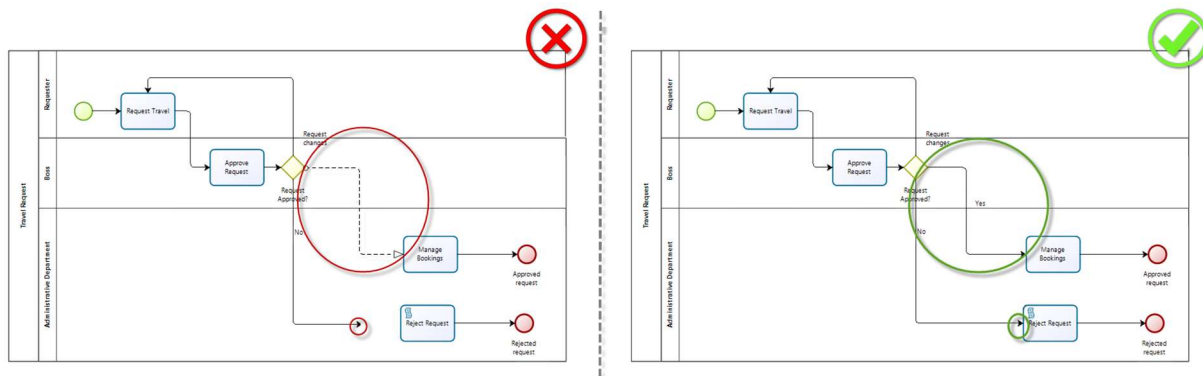
- Use Terminate End Events instead of End Events in Embedded Sub-Processes

BPMN – Best Practices in process modeling



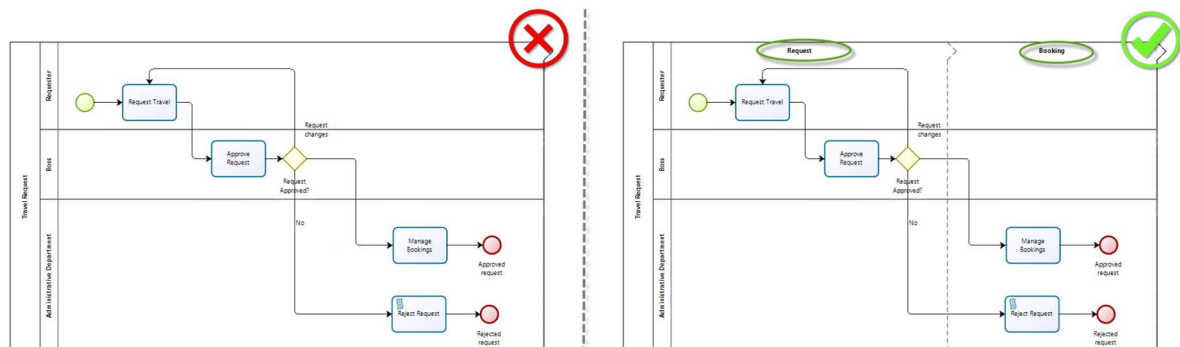
What to check in Connectors

- Use sequence flows to connect all the activities, events and gateways. Never use message flow to connect activities within the same pool or leave shapes unconnected.

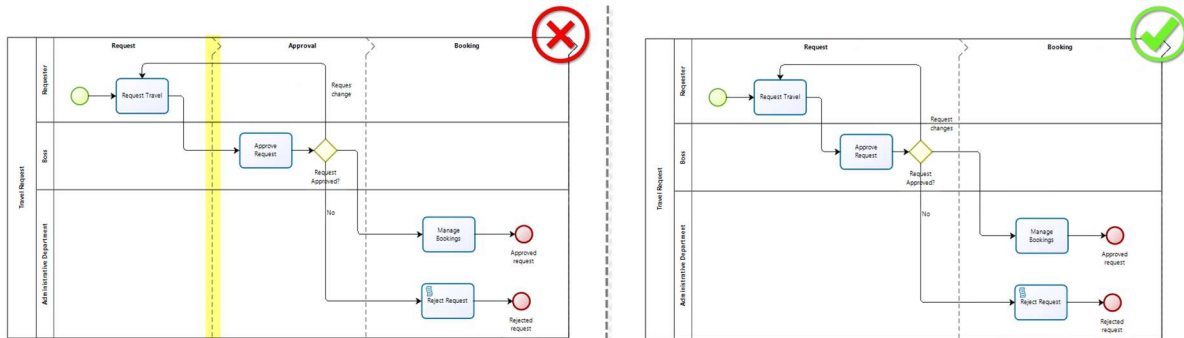


What to check in Milestones

- Always identify and define phases; these represent a period of time goal or transition in the process.



- Try to avoid come back or looping back across a milestone.



3. Use strict labeling

Correct labeling of the different elements of the diagrams is fundamental for an easy and correct understanding of processes.

When reviewing logs it is very useful to know how the process was executed. When any shape in the process is not named, Logs will display blank, making it difficult to understand. Here are some recommendations to help you do so:

Labeling Processes

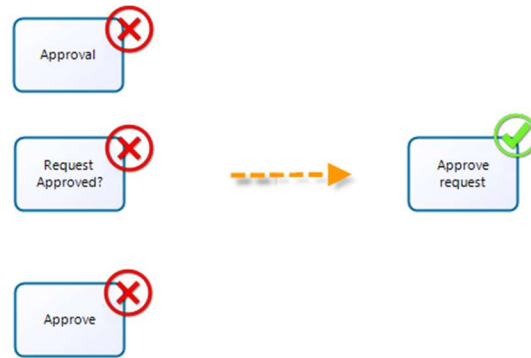
Processes labels should clearly describe their main purpose. Make sure that you do not use short names or abbreviations.

Prefix: using the process name create a prefix that will be used in all the components belonging to the process. For example, if the process name is: Consumer Requests, you may use the prefix CR_. The following prefixes are restricted for other purposes: “P_”, “M_”

Labeling Activities

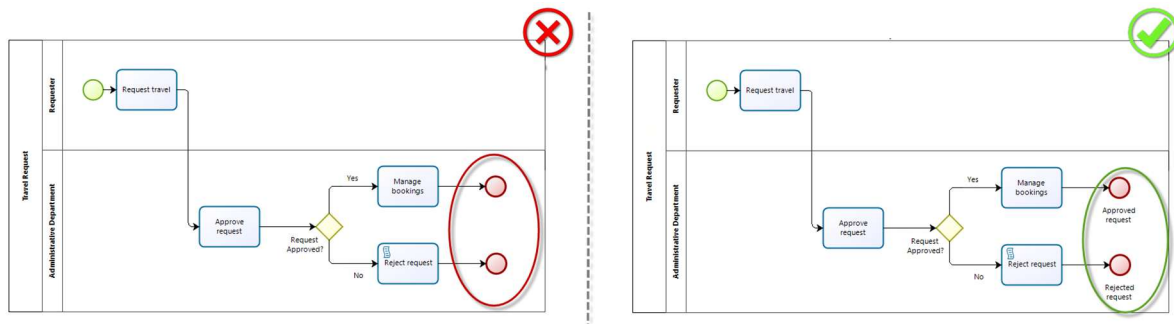
Give activities a label composed of **one verb**, and **one object**. This way readers can clearly understand the objective of a task. Also, make sure that you do not use short names or abbreviations.

BPMN – Best Practices in process modeling

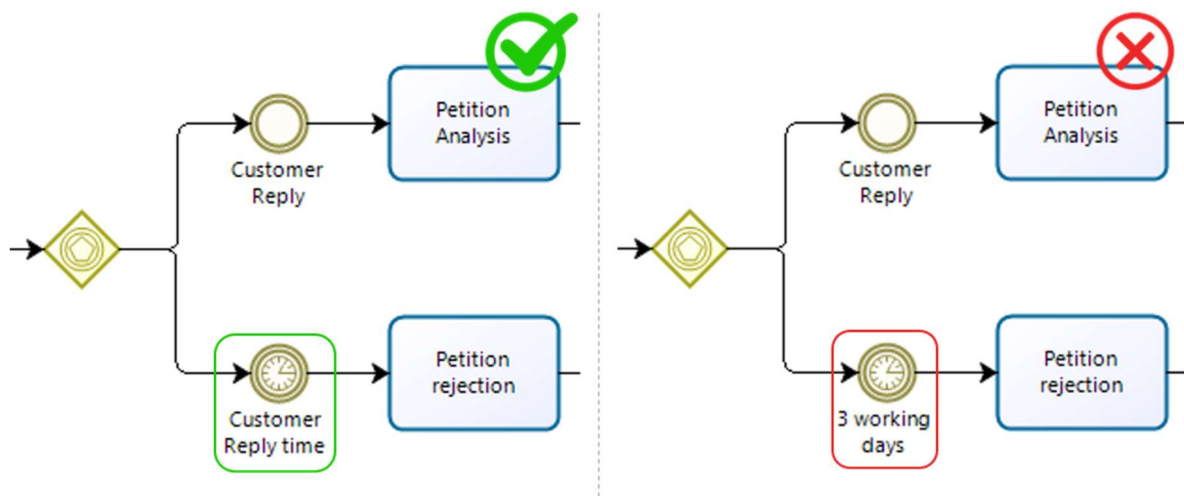


Labeling Events

Use labeling when multiple start and end events are used. Name them so the diagram can be self-explanatory and allow users to know how the process ends.



When Labeling timer events we recommend to not specify the amount of time of the event, instead, describe the expected event or the purpose of the given time.

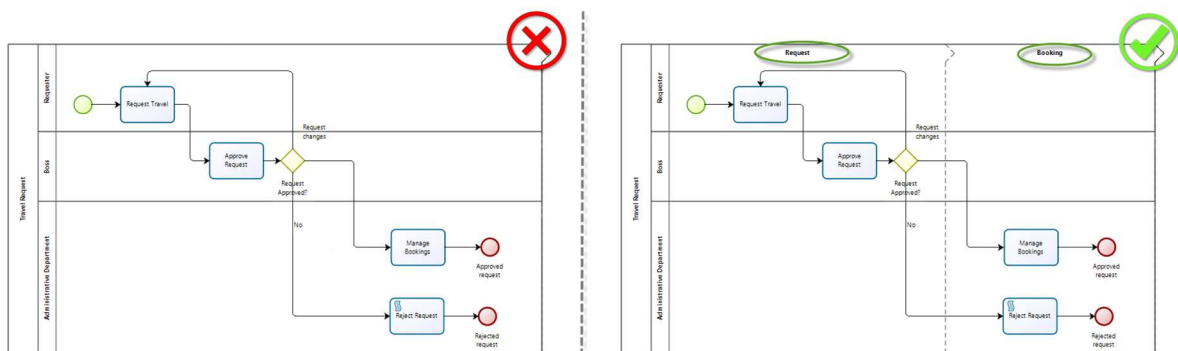


For the convergence and event based gateways we suggest to label them with the gateway type prefix and a serial number (e.g. the first convergence exclusive gateway should be EX01). The following table shows the gateways prefix:

Type	Label
Exclusive Gateway	EX
Parallel Gateway	P
Complex Gateway	C
Inclusive Gateway	I
Event Based Gateway	EV

Labeling Milestones

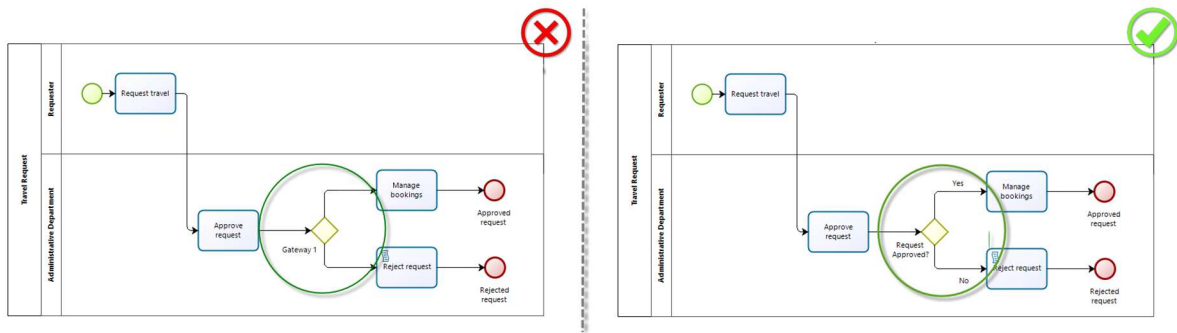
Milestones should be labeled with a noun making reference to a period of time (summer, maturity) or what happens in a period of time (creation, approval, delivery).



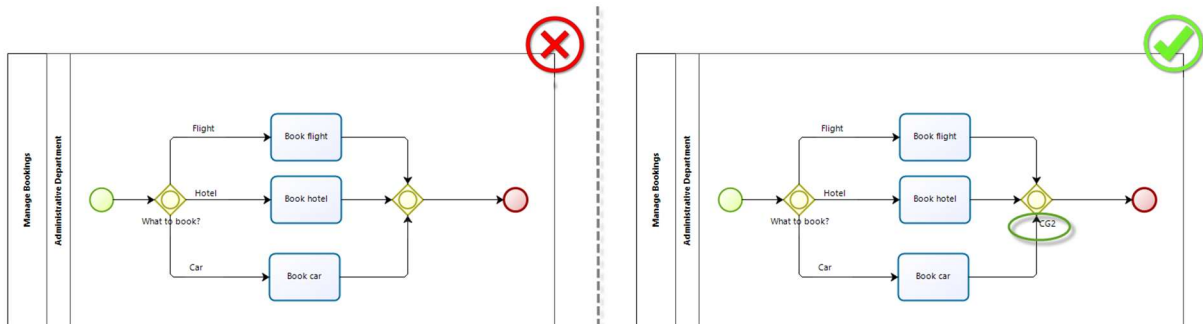
Labeling Gateways

Divergence gateways should have a clear name indicating the decision or condition evaluated when it applies. Use a name composed of one verb, one object, and a question mark to identify what is being evaluated. You can even use questions to clarify the decision involved.

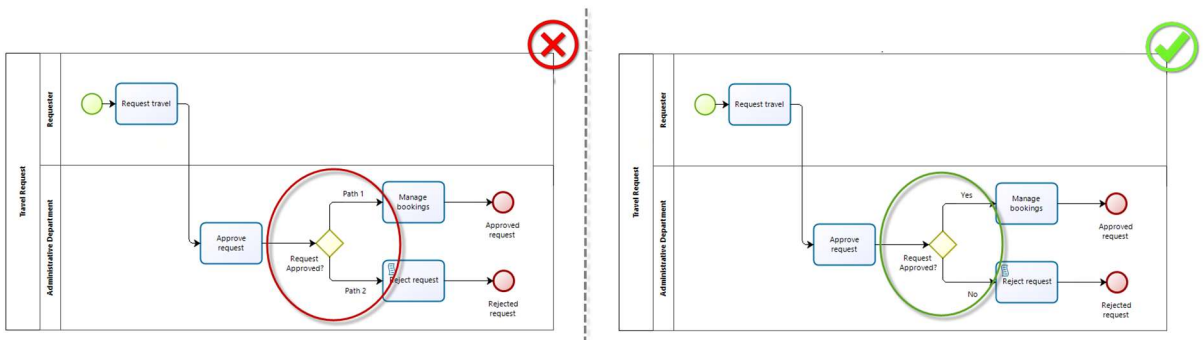
BPMN – Best Practices in process modeling



- If names do not apply for any gateway use abbreviations or numbers to differentiate them.



- Name transitions indicating the condition related



4. Simplify diagrams

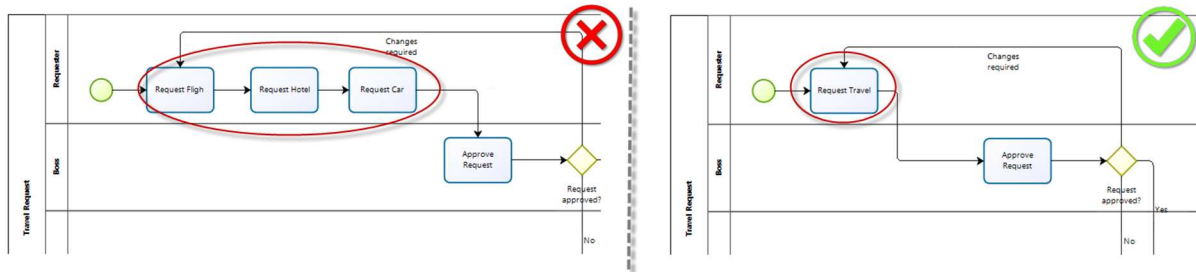
Large diagrams do not allow giving an end-to-end perspective to readers. They are difficult to read and clearly communicate the purpose of the process.

Defining the correct scope of tasks and level of detail of processes is key to reducing the overage of information. The following tips will help:

Reduce the number of redundant tasks

The level of detail in a process is sometimes a true challenge. In many cases you may face difficulties to define the scope of a single task. Take into account that:

- When diagramming it is useful to imagine that you are a final user. If a set of consecutive activities can be performed by the same person, at the same time then these activities could be integrated into a single activity.
- A set of consecutive activities in the same lane may indicate missing participant details, too much detail, or a misalignment in scope. Review these patterns to identify opportunities for activity integration.



Group activities

Use Sub-Processes to group activities with the same purpose. You can expand the Sub-Processes later to expose details of lower levels of hierarchy. A process will contain multiple pages, but internally the integrity of a single model is maintained.

Use embedded Sub-Processes when:

- A set of consecutive activities has an owner different from the main process owner (e.g. a *Purchase request* process is performed by the *Purchasing area* and the *Accounts payable* process is performed by the *Financial area*).
- A set of consecutive activities has a different goal from the main process one (e.g. a *Credit request* is focused on managing all the activities to approve a credit request).

and the *Verify applicant information* is focused on checking if the applicant is in the black list as well as the information submitted).

Use reusable Sub-Processes when:

- The Sub-Process needs to be invoked from different processes (e.g a *Verify applicant information* Sub-Process can be invoked from a *credit request* process or from a *Insurance request* process).

Use Start Forms

Use [Start forms](#) to launch a new case instance temporarily, and allow end users to confirm the process creation when they are certain of this action, or closing the form without confirming to avoid unnecessary case creation.

Do not use Anti-patterns

While simplifying your models take care of not using [Anti-patterns](#). These are modeling patterns that usually works in automation but are bad practices that are not recommended by Bizagi

Document minor details

Leave details to documentation. Do not include all the information on diagrams. Additional information should be documented as shape properties not as objects or text in the diagram.