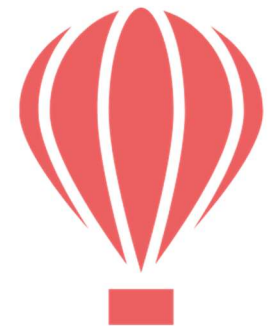


BPMN Essentials

A quick-reference guide



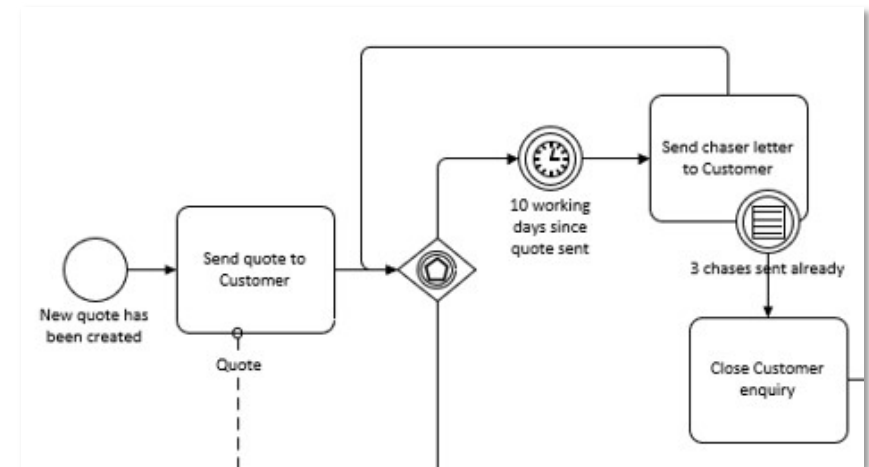
Better Business Analysis



What is BPMN?

BPMN is a framework and language for modelling business processes. It provides a way to categorise elements of a process and structure these into models that accurately describe the flow of activity.

The BPMN specification was formulated by the Object Management Group (OMG), and the current version (BPMN2.0) was published in 2011.





Why use BPMN?

There are many different methodologies for modelling business processes, and they all have advantages and drawbacks. BPMN is most helpful when:

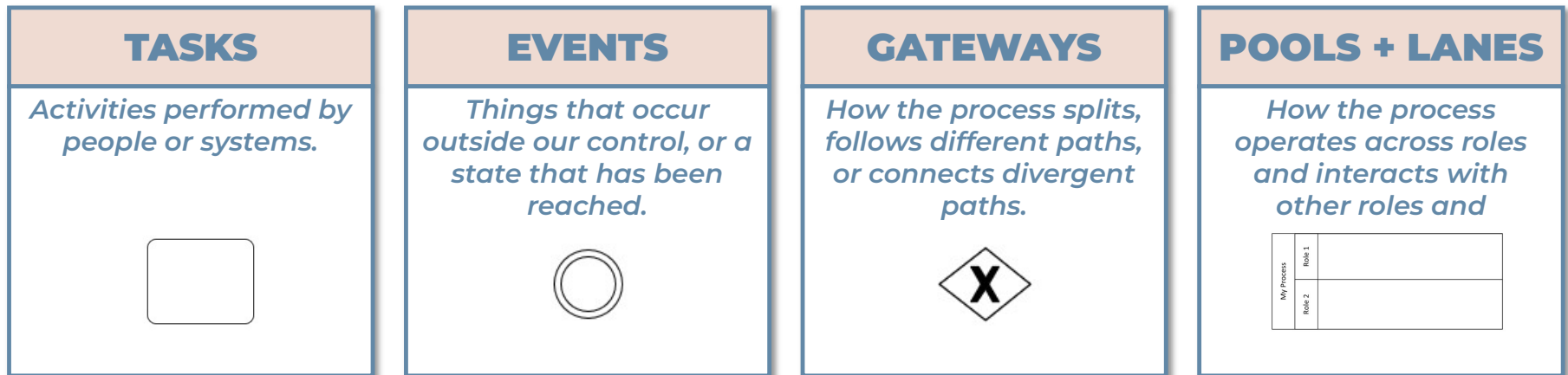
- Understanding the **precise order of steps** is essential, and ambiguity must be avoided.
- The process involves **complex logic** (e.g. where the steps followed vary based on business rules).
- The model will be used to **identify functional requirements** for technology to support the process.
- You need a **shared language** to describe processes (e.g. so models can be updated by multiple people).





Basic elements

BPMN uses a core set of objects to describe the components of a process. These essential elements are outlined below (and explained in more detail in the sections that follow).



For much more detail, check out the full specification here:
<https://www.omg.org/spec/BPMN/2.0/PDF/>



A note on BPMN logic

The action in BPMN models function like water pumped at high pressure through a series of pipes – represented by arrows or “Control Flows”. Components such as Tasks act like taps controlling the flow of the process, pausing or diverting it as required.

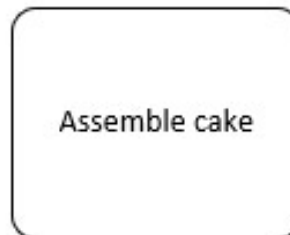
As soon as a Task is completed, a Gateway is passed, or an Event occurs, the process continues automatically and instantly to the next component. No time elapses, and no activity occurs within the arrows.

This rule can act as a prompt to ask insightful and revealing questions of stakeholders, challenging assumptions and tapping tacit knowledge about what really happens in the “gaps”.



Tasks

Tasks are used to show activities being performed by people or technology. They are the only place in a BPMN model where work is *actually* done. Tasks are represented by rounded rectangular boxes labelled with the activity performed (typically in noun-verb format).

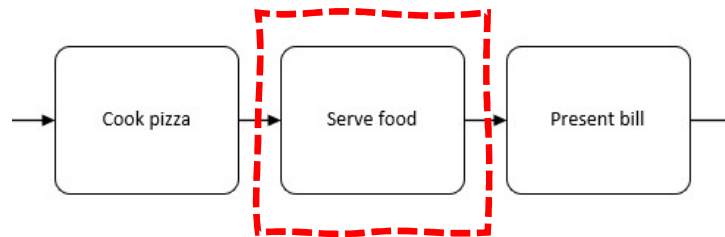


BPMN allows many additional symbols to be added indicating the type of activity performed (e.g. manual, automated etc.). Most do not change the logic of the process.

Tasks

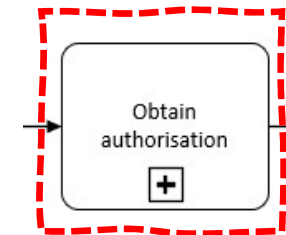


GENERAL TASK



- Describes activity performed by a person or system.
- Must contain an active verb for the activity performed, and a noun for the object of the activity.
- The label does not show the person or role performing the activity.

COLLAPSED SUB-PROCESS

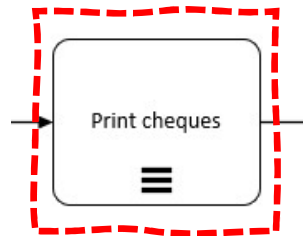


- Functions in the same way as a General Task.
- The task represents a larger collection of activities (that may be modelled elsewhere).
- The logic for starting and ending these activities should accurately map to the flow of this process.

Tasks



SEQUENTIAL LOOP TASK



- Indicates that the task repeats for each of the objects being handled.
- The process continues automatically when the task has been performed for each object.
- Useful for batch processing when items are handled in turn.

PARALLEL BATCH TASK



- Indicates that the task is performed multiple times simultaneously for each of the objects being handled.
- The process continues automatically when the task has finished.
- Useful for batch processing when many items are handled at the same time.

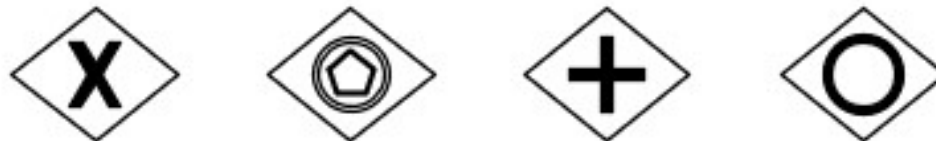


Gateways

Gateways are used to show how the process may follow different paths depending on certain conditions. They may also allow for multiple paths to be followed simultaneously.

Gateways are represented in BPMN by diamond shapes, and may contain symbols that indicate the Gateway's type.

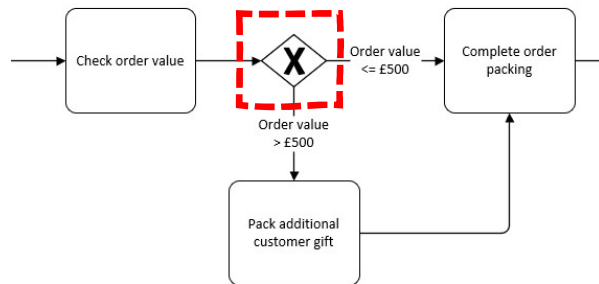
Paths exiting some Gateways are labelled to show the condition that must be met for that path to be taken.





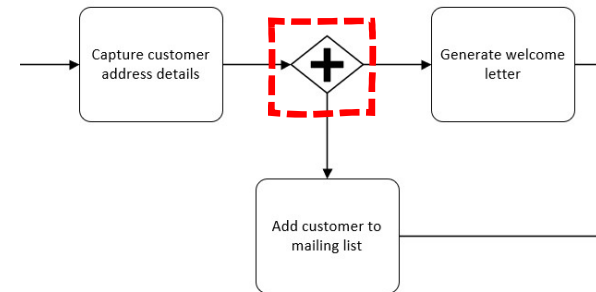
Gateways

EXCLUSIVE GATEWAY



- The process always follows one exit path.
- Describe conditions or guards on every exit path paths.
- Ensure at least one path will always be valid.
- Remember that no action takes place in the gateway – consider using a Task to describe decision-making step.

PARALLEL GATEWAY



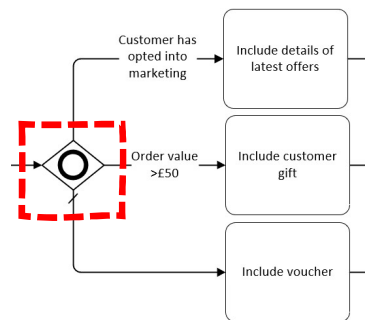
- The process follows every exit path automatically.
- Subsequent Tasks are triggered simultaneously.

Gateways



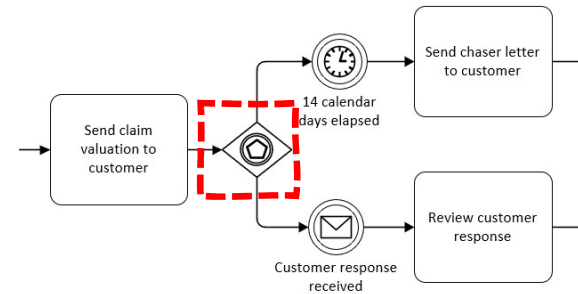
Better Business Analysis

INCLUSIVE GATEWAY



- The process automatically follows each exit path where the stated condition is met.
- You can include a “default” path (marked with a slash) to be followed where no other condition is met.

EVENT-BASED GATEWAY



- The process waits until one of the subsequent Events occurs. The process then follows that path only.
- Ensure all possible Events are included so the process cannot be left waiting indefinitely.
- Ideal for handling waits for responses and chasers!



Gateways as “joins”

Gateways can act as “joins” to combine two or more separate flows within a process. Joins may allow the process to continue as soon as one flow reaches them, or may force the process to wait until all upstream flows have reached this point.

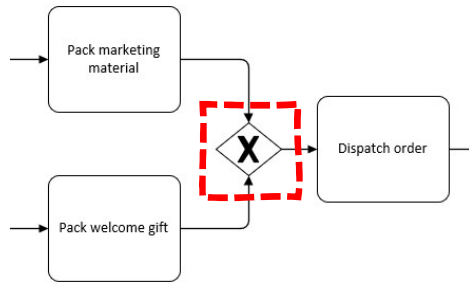
Like other Gateways, Joins are represented by diamond shapes, and may contain symbols that indicate the type of Join.



Joins

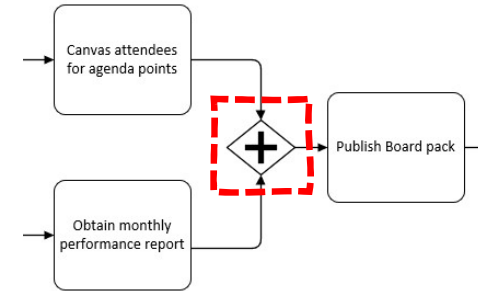


EXCLUSIVE JOIN



- The process waits until one of the preceding paths has reached this point, and then continues automatically.
- If another preceding path subsequently reaches this point, no additional action is taken.

PARALLEL JOIN

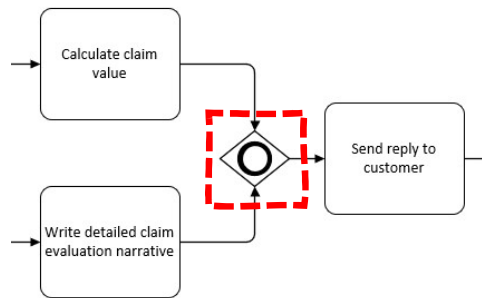


- The process waits until all preceding paths have reached this point, and then continues automatically.
- Must only have a one exit path, but this can be to another Gateway or Join if needed.

Joins



INCLUSIVE JOIN



- The process waits until all active paths reach this point.
- The process then continues automatically.
- Only one exit path leaves the Join, although this may connect to another Gateway or Join if needed.



Events

Events are used to describe things that happen outside of the control of the process. They can be used to describe the circumstances that trigger a process to begin, or specific events interrupting later steps in the process. An Event is also used to show that the flow of action has reached an end.

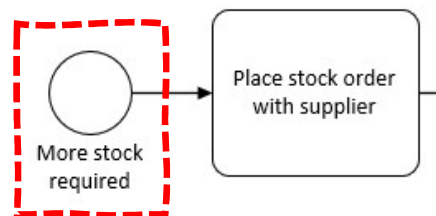
Events are represented by circle shapes, and may contain symbols that indicate the type of Event. The type of border around the circle also shows whether it is a Start, Intermediate, or End Event.



Events

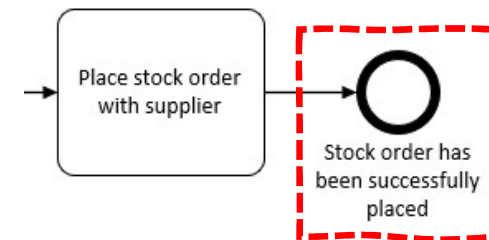


START EVENT



- A new instance of the process begins automatically whenever this Event occurs.
- Is labelled to describe the event that has occurred or state that has been reached.
- May have a specific type applied (time-based, message etc.) but does not require one.
- No activity is performed within the event itself.

END EVENT

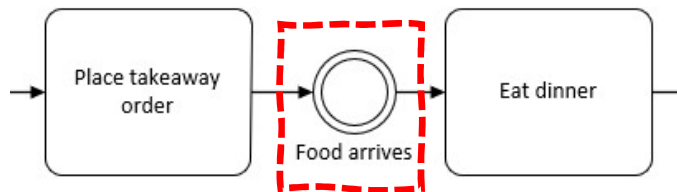


- Shows that the process has reached an end.
- Is labelled to describe the state that has been reached (this wording may match a Start Event of another process).
- No activity is performed within the event itself.



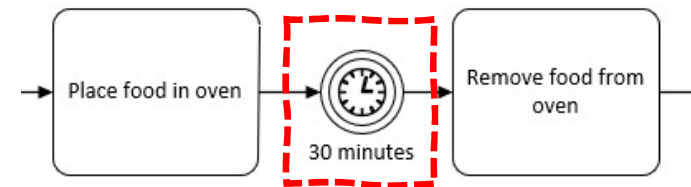
Events

INTERMEDIATE EVENT



- The process waits until this event occurs and then continues automatically.
- The label describes the state when the event has occurred.
- May have a specific type applied (time-based, message etc.) but does not require one.
- No activity is performed within the event itself.

TIME-BASED EVENT



- Functions in the same way as any other type of Event.
- Is used to represent a particular time/date being reached, or a certain amount of time having elapsed.



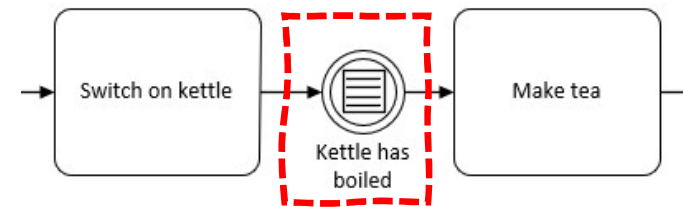
Events

MESSAGE EVENT



- Functions in the same way as any other type of Event.
- Is used to show that a direct communication of some kind has been received by the process (e.g. a letter, email, order, transfer of data).
- The label states that the communication or item has been received.

CONDITIONAL EVENT

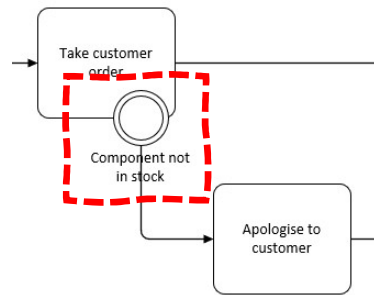


- Functions in the same way as any other type of Event.
- Is used to show that a specific condition or rule has been met.



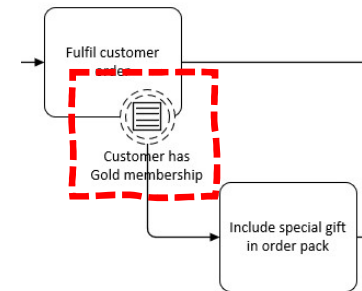
Boundary Events

INTERRUPTING BOUNDARY EVENT



- Placed on the edge of a Task.
- Can occur at any time while the Task is being performed.
- Stops the Task from being completed.
- The process proceeds down a new path.
- Label describes the event that occurred.
- Can use an Event type (e.g. Message, Timer etc.).

NON-INTERRUPTING BOUNDARY EVENT



- Functions like an Interrupting Boundary Event, but the Task still carries on to completion.
- An additional flow is therefore created within the process while the main flow carries on as normal.

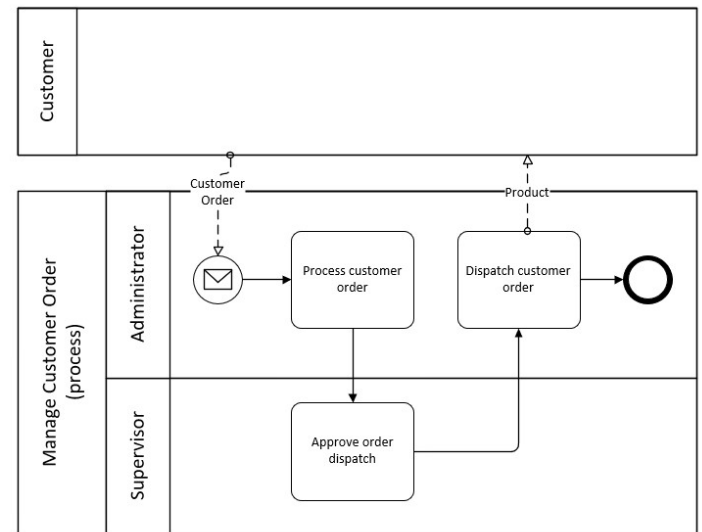


Pools, Lanes, and Messages

Pools are used to differentiate between connected processes or to show how roles outside of the main process interact with it.

Lanes are used within the Pool of the main process to represent the different roles (people or systems) that perform activities within it.

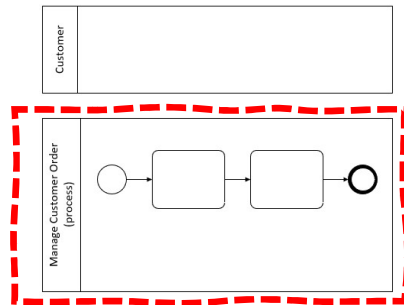
Message Flows represent the communications in and out of the process to other Pools (processes or external roles).





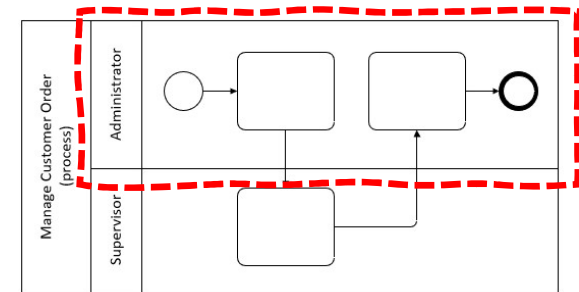
Pools, Lanes, and Messages

POOL



- Represents a process, role, or business area.
- Expanded Pools contain process detail (and are labelled with the name of the process) and may contain any number of Lanes.
- Collapsed Pools can represent other processes or actors we cannot control (e.g. customers, suppliers etc.).

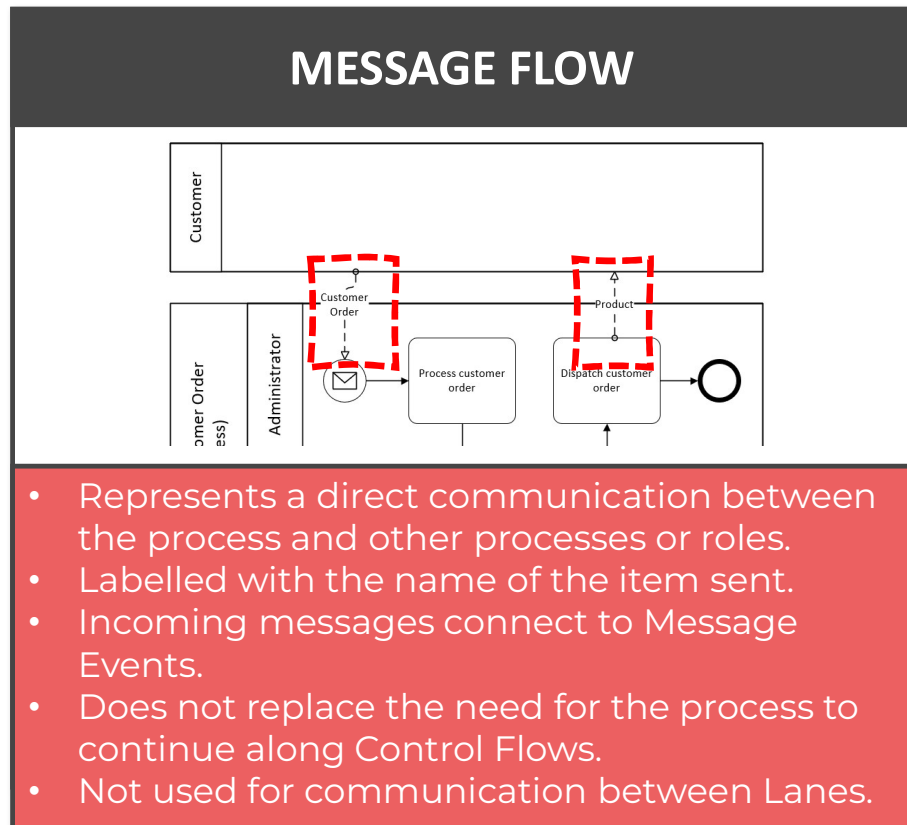
LANE



- Represents a named role within the process (although this may not map to a job title).
- Labelled with the name of the role.
- Pools can contain any number of Lanes.
- Only use Lanes where we can design the



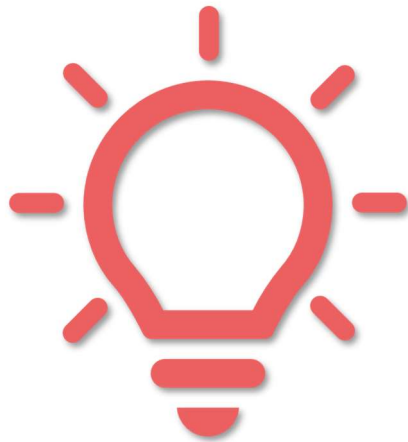
Pools, Lanes, and Messages





Hints and tips

Here are a few pointers to help you build accurate, accessible, and valuable models!



BPMN models are great for capturing detail but can quickly get convoluted and unwieldy. Consider breaking processes into smaller chunks - particularly if you are sharing your outputs with non-analysts!

BPMN offers many more symbols and objects within its full specification – these can be helpful when used well to tackle specific scenarios but risk making your models more cryptic and arcane!

Think about how your models will be used. Are they a communication tool, a governance artefact, or an input for further analysis? Being clear about purpose will help you find the right level of detail and accuracy.

Finding the right tool for constructing your models is important. Consider whether you need to link processes together, share models with non-analysts, or embed models into other artefacts.

Remember that work only gets done within Tasks – not in Gateways, Events, or on the arrows

Be wary of jumping between single instances of a process and processes that handle many things together (e.g. handling a job application vs filling a vacancy). It's generally cleaner to separate these!