# Webiny process builder

The goal of WPB is to create a UI where developers will be able to create processes that consist of forms, works flows and other elements and events, and combine them into a process with, one or more flows. These flows will be executed based on the defined conditions for each step.

Once a process is create, it can be compiled into a PHP code.

For now, the goal of WPB is to replace controllers inside modules.

## Steps

There are two types of steps:

1. **Process step**
   * Process steps can contain only elements.
   * Each process step can contain only one element.
   * It's a state in which a process can be.
   * One step can have one or more attached step that will be triggered when a condition is met.
2. **Event step**
   * Event steps can contain only events.
   * Events are also triggered based on the condition.
   * Events are triggered parallel to the process step.
   * You cannot attach a process step to an event step.
   * Event steps are always triggered before process steps

## Elements

Each step can have only one element inserted. The following elements should be available for process steps:

* Form
* Grid
* Entity (based on entity class we can get the form or a grid for that entity)
* Dialog box
* Message
* Redirect
* \*Workflow (nesting)

\*The workflow nesting element is actually a process inserted into another process   
(for more info visit: http://cdn.memegenerator.net/instances/400x/24205076.jpg).

When an element is inserted into a step, the step will automatically get 'callbacks' that this element provides and the conditions will get element rules.

## Events

Events are actions that are executed in the background. Following events should be made possible:

* AJAX request (POST & GET)
* Trigger PHP callback
* Trigger JS callback
* Send email
* Generate PDF?

All events return their result to the parent process step, which then provides that result to the condition for re-validation.

## Conditions

* Conditions define when the next step will be triggered.
* Conditions are threads that connect two steps.
* When a condition is met, the connected step is triggered.
* A step cannot exist without the condition, with the exception of a Root step.
* Conditions exist between both process and event steps.
* Rules that will be available in the condition are defined by the elements in the parent step and attached event steps.
* Every conditions has these callbacks:
  + Before transition
  + After transition