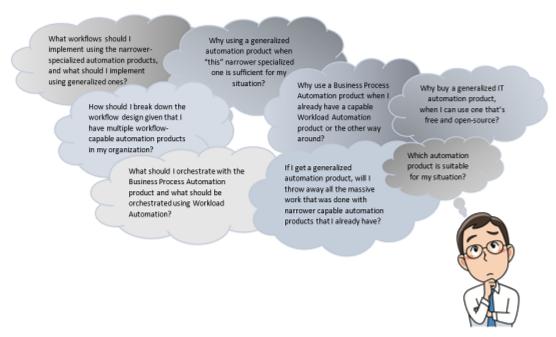# HOW TO SIMPLIFY COMPLEX ENTERPRISE WORKFLOWS IN THE DEVOPS WORLD

The multitude of workflow-capable automation products, with their wide spectrum of specialized/generalized, but overlapping capabilities, often causes confusion for potential buyers or workflow designers.

We can simplify the automation-product selection and implementation in a modern digital environment. Defining

workflows with dedicated automation tools is in some way like **writing software programs**, but at a higher level, usually in a visual and/or declarative approach ("-as-code" approach).

Besides grouping activities to achieve a modular divide-and-conquer functional decomposition that is easier to reuse, maintain, and grasp logically, some of the following should be considered when defining activity workflows:

- input data/information to be consumed
- actions to be carried
- logical decisions to be inferred before running one or other activities
- output data/information to be generated
- constraints to be satisfied
- exceptions to be handled

Given the similarities of workflow design and software development, it makes sense to refit some established software design principles, patterns, and practices from the software development domain into the workflow-oriented automation design.

The following workflow automation structural design principles, inspired by software engineering, will help in creating workflows that are easy to comprehend, reuse, progressively refine, and diagnose:

1. *Group activities in workflow units that reach only one major functional goal.*
2. *A higher-level workflow (more generalized) should call a lower-level workflow (more specialized) and not the other way around.*
3. *Segregate group of activities that are prone to change from the group of activities that have a pseudo-static change tendency.*
4. *Group activities in workflows that have the change propensity caused by the same reason.*

The first two principles are more intuitive and commonly applied, even if the second one might get often violated after the initial release of a workflow when engineers forget or don't understand its initial intent.

The third principle is important, as it helps to separate the group of activities in workflows that are stable from the ones that might need changes and retesting.

The most significant workflow design principle though is the fourth one. The benefits of this principle are the most valuable during the automation product selection but also towards achieving the goal that many organizations have of enhancing their enterprise-wide business agility.

Failing to adjust fast a procedure to eliminate an existing gap swings fast from a procedure that is repeatable, safe, and avoids human mistakes into a procedure that repeats fast and amplifies the negative side-effects of the given gaps.

If your organization is prone to change for various reasons—for example, adjusting to market changes, adopt the latest innovations, adopting DevOps, or going through an organizational transformation—it is probable that procedures, processes, activity workflows might also need to change in order to avoid gaps against the new goals. It means that you must be mindful during the workflow design and apply the principles mentioned above, select generalized automation products that are covering the change concerns that are relevant to your environment, and design your automation procedures in a way that recurrent changes are easy to apply.