

## The Benefit Of A Proper Workflow Design

Why is it important to keep the workflow design updated?

Should you really change the process design with each change request or added feature?

You will find out, how you benefit from a continuously refactoring as part of your Appway development work.

**My Intro:** My name is Saso Nikolov, computer scientist and coach for Appway. I worked 4 years for Appway as a product trainer and developer. I love the Appway process platform - really I do. Those of you, who attend one of my trainings know how much I do :) It is a great product, that allows you to create fast and complex process oriented solutions.

Single responsibility and reusable process designs will make your life easier in terms of developing with Appway. The beginning is always easy. The challenge comes with your best friend: The Change Request.

When we start mixing high level and low level activities and design patterns, it will become very hard to apply changes later. There is always the fear to break the solution. Not to mention, that we need too many variables, such as helper variables for gateways, script tasks or similar, on the process level.

Additional: most of the time we cannot reuse the process for similar approaches. This leads us to create more business objects and with this more maintenance work for the same code.

### What is the difference between high level process and low level process?

High level process contains, logical activities. Typically sub processes, gateways and events. The actual work is done within the low level processes. This allows us to use the low level processes to achieve a desired outcome.

I like to move the decision if specific tasks are executed or not, into the sub process. In the sub process you can use gateways to have different workflow approaches for the operation. This allows you to have a more static and much more simpler high level process design.

At the high process level design, you can basically follow the phase approach and just connect one activity (sub process) with the next one.

Example on the high level: “Gather client data”, “Review client data”, “Approve client”.

The low level process for “Gather client data” could have specific business logic to ask the right questions, to display additional screens and so on, depending on the client use case. e.g. legal entity or physical party.

But these steps are not needed on the higher level to understand the workflow. This approach allows to create the same process more than once, to have AB Testing or just to have other user experiences. But they are using all the same sub processes.

2 additional benefits from this approach:

1. Different teams are less disturbing each other
2. More performance if you want to spare helper variables to keep the value store smaller

On big projects it is not unusual to have few teams working on the same package or business objects. With Appway this becomes very fast very cumbersome. If we use more sub processes responsible for less things, than concurrent work is less often. This improves the performance of development.

If the goal is to reduce also the value store size for solutions, then often helper variables on processes with a lot of tasks are skipped, and the function to calculate the value typically held in the helper variable is executed more often. This will keep the value store smaller, but you pay the execution price and the solution suffers on performance. So using the helper variables within the low level processes limit their lifetime and enhance the performance of the solution by reducing function calls and engine-internal object creations.

High level process do not mean that it has to be the root process. You can have high level processes basically also much more deeper in the workflow. It is basically an concept to not mix execution activities with workflow design activities too much.

I say too much, because you should not over engineer and force a very strict approach. And at the same time, the more stricter you can be, the better it will be for you and later changes.

## Process helper Data Class

If you worked already with Appway you might experienced already that helper variables could be very useful. Helper variables stores values, that are not needed on the whole process, but at least on 2 activities within a process.

If we would add now a variable to solve this issue, then we would add a variable to the value store and pollute the process with variables that are not needed. We could try to move the variables into a sub process, and this is a good approach. But what if the activities are not next to each other? Then it will not solve your problem but increase the complexity.

That's where process helper data class could help. Yes, it will be added also to the value store. But the pollution is reduced and we can enhance the structure easily without changing the variable assignment. But if the value store is in issue, than we could repeatedly call the function to calculate the needed value. This is ugly, but in case of value store issues, a price we could pay.

The process helper data class has its shining staging moment, when it comes to reusability and single responsibility. Imaging you need to process scripts, then you can add it to a script task. But then it is not reusable.

Basically you have always the need for reusability. Even you have this code only once within your process design. But what if you want to test the code within this script task, then it would be great to have the code only in one place. A function of the process helper data class could be used. That way we reduce the complexity on the script task, and we can reuse the code, just by calling the function again.

Of course this function makes only sense, if the function does not belong directly to a data model. Otherwise it would be best to add this function into the data model.

## It simplify your process testing

If you decoupled your process that way, you can also test them on the interactive script editor. You can test it within a test process. Appway gives you the opportunity to start processes with code - Appway script.

This is great. If you decoupled the variables good enough, you just need to prepare a few variables to be passed to the process. Then you can run the process and check the variables for the result or generate even a test output. The test output could be a helper variable with the results, or even a PDF.

With Appway it is so easy to generate PDFs and you could have one PDF Output business object and a document prepared for this. To keep the same structure, I advise using a Data Class with the structure that will hold the testing results. It could even be the same, you are inspecting after a test run.

## Process helper Data Class for testing

It might be helpful for testing to create a process helper Data Class for testing. You can use logic within the process to understand if a test run is done. This would allow you to evaluate properties or functions to understand if you are in the test mode and which way to go. But yes, this requires much more work, and is not trivial.

## Can you test all processes that way?

**Downside: Screens.** For now you cannot run a process and auto submit Screen tasks on a dynamic way. The provided checkbox is static - for now :).

## What to do?

Try to keep the same level of complexity to your process design. Not only for testing, but also for reusability. It will make your process also easier to read. That way even new developers can catch the meaning and structure faster. New developers will break less code.

The testing part will be covered in another post, because it is a big topic and we have a lot of options to perform testing. So for that, it is not enough space in here. At the same time I want to use the single responsibility pattern also to my posts.

Great or great? You decide.

Saso von DigitalMove Consultants

## Appendix

Permalink:

<https://digitalmove.ch/knowhow/the-benefit-of-a-proper-workflow-design/>

About us:

We assist our clients in the banking and insurance industry to improve their development profitability. We do this by reducing their fixed operational costs, improving team performance, and expanding their solution quality.

Specialized in:

- Coaching your team during development on the Appway platform
- Consulting your team about the Appway platform
- Development with the Appway platform

Contact me:

Saso Nikolov  
english/deutsch

079 94 49 235

[s.nikolov@digitalmove.ch](mailto:s.nikolov@digitalmove.ch)

Contact us:

DigitalMove Consultants GmbH  
Bösch 23  
CH-6331 Hünenberg

+41 (0)44 461 14 28

[info@digitalmove.ch](mailto:info@digitalmove.ch)

[www.digitalmove.ch](http://www.digitalmove.ch)