

3 pitfalls Java developers fall for on an Appway Project

What is the difference between a good Java Developer an experienced Appway Developer? Which type brings you more benefits?

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

More and more companies decides to use Appway. So I am often asked what are the criterias to look for, for hiring a good Appway developer? Or at least the skills people should have to be able to make a fast start with Appway and being also immediately productive. I like that. Being productive from day 1 is not an issue, the issue is more the effectiveness. New Appway developers can start very fast. The learning curve is very steep, once you learned the business objects and their components.

But the danger comes from the easy to learn, very hard to master situation. Putting the wrong business objects together can harm your solution, your performance and with this this project. Maintenance nightmare and change requests monsters are often enough created.

Why is this? If Appway is easy to learn, why is it sometimes very difficult to get great performance and stability into Appway projects? The reason is, that knowledge comes with experience. And a lot of the developers just do not have enough experience with Appway.

What are the most common errors in Appway projects?

Even worse, often experienced Java developers are hurting the created solution more, than they help to build a great solution.

Common errors:

- Coded processes
- Too much logic on Screens
- Applied Java patterns to data classes without understanding all business objects
- Lack of understanding of web solutions - how they basically work and that they are NOT desktop applications
 - Memory usage
 - Performance optimization
 - Modelling the model

The following text are focusing on Java developers, just because I saw the most issues with their approach, that are perfect for desktop applications, but less great for Appway solutions.

1. Coded processes

Java developers tend to “code” processes. But Appway has a BPMN engine, and you should design processes. With the coding of processes you lose control over the workflow, in terms of stability, auditability, debugging and bug fixes.

A clean high level design not only offers you an easy to understand process model, but also increase the understanding why bugs happen. You can simply follow the workflow and fix the task.

You can spot coded processes easy by looking at the complexity already on the higher level. Not reusable workflows. A lot of script tasks with a lot of duplicated code. A lot of business logic implemented by using way to many complicated designs with many gateways and conditional connections.

The process should not mix too many high level activities like sub process tasks with low level activities like screens, gateways used for business logics, scripts and so. That way you make you process reusable. The single responsibility approach helps also on a visual approach to achieve better and clean implementations.

2. Too much logic on Screens

My very first screen to implement a very simply change request outside the academy was an eye opener. I started the screen editor and looked at a screen implementation that was basically orange :) Appway developer knows what I am talking about. In Appway the logical components within the screen editor are orange.

This screen was doing way too much magic. My task was simple, just to change an existing component a bit. I searched for a keyword and found that component. Quick change, save and reload of the screen - nothing. Uuuuh. I spent some time with changing, replacing and almost filing in a bug, till I realized a genius was duplicating the code base, and added his changes in the duplicated code. Surrounded the old code with an if-false statement. It was invisible for me, just because of the huge amount of components. I lost my last hair, many valuable minutes of lifetime. But I gained also something very important: the knowledge, that Appway Screens are not JSP forms and we should prevent this orange land at all costs.

Some of the logical components can execute code, but many developers forgot about the screen lifecycle model and wonder, why their approach is not working or why their solution is sooo slow. They are running loops of database operations 4 times and using them only to display some information.

Try to exclude the magic into a data class, process, rules returning includes.

3. Applied Java patterns

I often see great java developers trying to apply at all costs complicated java patterns to appway. Ignoring all the underlying Appway engine and implementation. Then they wonder, why is it so slow. Why not? No offense, but I saw often enough, that Java developers loves the language and focus strongly on the java pattern implementation details, than creating a solution to achieve the customer goals.

At the end you have a code monster shared teams can hardly handle or are not able to understand the whole pattern. So you will have very fast, less fast development, due everybody tries to understand how to add or change something without breaking the existing implementation.

Example: If you want to call an integration link, than do it. Do not create a monster of 20 business objects to call it and realizing it will not work for change requests. Appway is a great platform (I really love it), you can do so much so fast. But you need to respect how she works, how she needs to be treated. I know she is easy to learn, but hard to master. That's why I recommend continuously learning and refreshing of the Appway knowledge. Fresh out of the academy, you forgot already a lot of information. That's normal.

What to do?

Hire Appway Experts :) If you cannot, then look out for this pitfalls and speak about them right from the start. Add continuously learning to your developers. Add it to yourself too. Appway trainings are full of informations and people will forget a lot of them right after the training. Sometimes because, the topic happens once in year or less, or the information is not needed at the beginning of the project. Additional human beings just simply are not built to remember everything after one rehearsal, specially if the information is packed and pushed into a short period of time into our developer brains.

My best advice for quick wins: Go for single responsibility and strive for reusability. This will also makes your implementation more testable.

To go freely with Bruce Lee: "Our brain is still full of other coding knowledge and we need first to empty this, to fill in pure Appway development stuff in it." Have you ever tried to fill a strawberry milkshake into a glass still half full with diet coke. Sure you can drink it, but will you like it? Even if the first sips are ok, how will it taste if you reach the bottom of the glass. iiaaah :)

Great or great? You decide.

Saso von DigitalMove Consultants

About us:

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

Specialized:

- Coaching on the Appway platform
- Consulting for the Appway platform
- Development with Appway platform

Contact me:

Saso Nikolov
english/deutsch
079 94 49 235

s.nikolov@digitalmove.ch

Contact us:

DigitalMove Consultants GmbH
Weinberghöhe 10B
CH-6300 Zug

+41 (0)44 461 14 28

info@digitalmove.ch

www.digitalmove.ch