ALLAN THRAEN | 1 years ago | PDF |

.NET Development   Optimizely (Episerver)   CMS   Website Improvements   Tips and Tricks   Tech Talk

CHRISTMAS COUNTDOWN: #3 NIHS - NOT INVENTED HERE SYNDROME IN REAL LIFE

One of the most common and dreaded diseases in web site development often go undiagnosed and untreated for a long time. But it really should be, cause the effects are scary. Yes, I'm talking about the Not-Invented-Here Syndrome

The 2023 Christmas Countdown: 12 Common Pitfalls in Optimizely CMS - and how to avoid them

This blogpost is part of the 2023 Christmas Countdown series where I each day for the last 12 days before Christmas go through my Top 12 list of the most common and dangerous pitfalls I typically see in Optimizely (EPiServer) CMS 12 Implementations. If you want to learn more and perhaps have your own site evaluated feel free to reach out to us here at CodeArt.
Here are links to all the posts in this series as they are published:
12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

"I should probably build my own [insert big side project here]" hands up - how often have you found yourself thinking that while in the middle of an implementation? More often than not is my guess, judging from all the greenfield development and all the lines of code I see in every single implementation I look at.

And I get it. I also love inventing stuff - in fact, if you scroll through my blog here you'll see plenty of evidence of it.
But think back to #8 on our top 12 list, Christmas Countdown: #8 Code maintenance is 90% of the work, and you might reconsider if it's the right time to re-invent the wheel. In fact, most of the time when we feel like building a huge side-project from scratch it's already a solved problem - either in the core CMS, in best practices, in a popular add-on or a 3rd party product.
And if you are lucky you can take that solution and either use it as-is, or perhaps extend it a bit so it fits your specific need.

Some of the things I often come across custom implementations of, that makes me wonder:

- Custom Commerce. Sometimes a complete bespoke commerce platform is built on top of the CMS instead of using EPiServer Commerce.
- Custom Rendering Tags. Perhaps rendering tags is a less known functionality of the CMS - it's not uncommon with inventions of new ways to control different renderings for the same content.
- Custom Image Converter. Nowadays ImageSharp or direct Cloudflare conversions are standard. But I still see a lot of other image converters.
- Custom Sitemap Generators
- Custom Rest API. I see this one very often on headless solutions - for some reasons almost no-one will use the Content Delivery API.

- Custom Forms. There are perfectly good Episerver forms - but no, let's invent something that does the same thing but in a much more complex manner, requiring loads of code

When I come across custom code like the above, try to figure out why. And it's often a bit of detective work. There can exist completely good and legit reasons like:

- Historical - a good solution didn't exist at the time.
- Legal/political reasons & requirements
- Cost of solution compared to 'home built' (although usually someone has forgotten to factor in the future maintenance cost)

But there are also a lot of bad reasons - which are all really just symptoms of the Not Invented Here Syndrome

- Not knowing that a standardized solution existed
- Evalutated and standardized solutions didn't meet the needs specifically. But remember they can usually be adjusted. Or how about forking the repo, improving the code and doing a PR - that way contributing to the world while still using somewhat standardized code.
- Agency lock-in (when the implementation agency wants to make sure the client doesn't switch vendor so they make sure to depent on code they own, not the client).
- Junior developers wanting to code rather and build from scratch.

NIHS is dangerous and can be incredibly expensive in maintenance in the long run.

.NET Development Optimizely (Episerver) CMS Website Improvements Tips and Tricks Tech Talk

- Custom Forms. There are perfectly good Episerver forms - but no, let's invent something that does the same thing but in a much more complex manner, requiring loads of code

When I come across custom code like the above, try to figure out why. And it's often a bit of detective work. There can exist completely good and legit reasons like:

- Historical - a good solution didn't exist at the time.
- Legal/political reasons & requirements
- Cost of solution compared to 'home built' (although usually someone has forgotten to factor in the future maintenance cost)

But there are also a lot of bad reasons - which are all really just symptoms of the Not Invented Here Syndrome

- Not knowing that a standardized solution existed
- Evalutated and standardized solutions didn't meet the needs specifically. But remember they can usually be adjusted. Or how about forking the repo, improving the code and doing a PR - that way contributing to the world while still using somewhat standardized code.
- Agency lock-in (when the implementation agency wants to make sure the client doesn't switch vendor so they make sure to depent on code they own, not the client).
- Junior developers wanting to code rather and build from scratch.