Journey from Monolithic to Modern CFML Development - Introduction

Journey from Monolithic to Modern CFML Development - Introduction (http://ciacfug.org/blog/journey-from-monolithic-to-modern-cfml-development-introduction)

Posted by Andy Peterson 14 Apr 2020 04:06 PM

Part 1 in a Series - Learning CommandBox, ContentBox, ColdBox, VS Code and GitHub - a little at a time and all at once.

Introduction

I am an executive at a state agency presiding over a team of bright young CFML developers, many of whom were hired to take over for a retiring workforce. It's hard to think that any CF developer is eligible for retirement but the language has indeed been around for decades now. I consider myself a bit younger than retirement age, but I've also been around for some time, and started coding in ColdFusion in the late 90's. And not only have I been around that long, but some of the code currently in production is as old as my career. In fact, my team is responsible for applications that have, at minimum, one million lines of ColdFusion code* - more than any other language we use.

One of the stranger opportunities that has emerged from the COVID crisis has been the added leverage needed to my push to adequately equip distributed development teams. Through a collection of circumstances, developers working in their separate local environments and updating code to a repository has finally taken root within our organization - something our private sector counerparts have learned long before. I still do consider our shop modern compared to the vast majority of state agencies I work with, but what we're doing now puts us on par with much of the private sector.

There are many technologies that have enabled this to happen for us - household names like the Microsoft and their ubiquitous VS Code (https://code.visualstudio.com/) and its enormous flexibility combined with virtually seamless integration with GitHub (https://github.com/) has made our lives easier. But I know we wouldn't be where we are today without Ortus Solutions (https://www.ortussolutions.com/)' box products: CommandBox (https://www.ortussolutions.com/products /commandbox), ColdBox (https://www.ortussolutions.com/products /coldbox), and ContentBox (https://www.ortussolutions.com/products /contentbox) to name a few, and I am pretty certain our CFML shop would have been swallowed up without their availability and my advocacy for their use. Ortus' prolific nature and passion for their products is unmatched in any space, and that kind of passion is contagious.

The beauty of our adopted new technology stack is that is accompanied by recent approvals by the exec team to use Axosoft (https://www.axosoft.com/) and Scrum for project management. Axosoft, as you may know, nicely integrates with GitHub as well.

So a couple of weeks ago I introduced a training manual of sorts for my team to move to a modern way of development. I'm pretty excited to share what I've learned, warts and all, in the next several blog posts. Because I was learning as I developed these instructions for my team, I faced many issues up front. While I wouldn't change anything about my journey I do hope to clear up any confusion you have about how all these pieces of the puzzle fit together.

So again, I feel that in some ways we're late to the party, but in other ways, we're right on time. I hope you'll stick around for more posts in the near future.

Tags: Axosoft (http://ciacfug.org/blog/category/axosoft), ColdBox (http://ciacfug.org/blog/category/coldbox), ColdFusion (http://ciacfug.org/blog/category/coldfusion), ContentBox (http://ciacfug.org/blog/category/contentbox), GitHub (http://ciacfug.org/blog/category/github), News (http://ciacfug.org/blog/category/news), VSCode (http://ciacfug.org/blog/category/vscode)

0 Comments (http://ciacfug.org/blog/journey-from-monolithic-to-modern-cfml-development-introduction#comments)