Subject: Thank you!
Posted by Nebula on Thu, 18 Oct 2012 13:25:00 GMT
View Forum Message <> Reply to Message

Hope you don't mind me posting this here, couldn't think of a better public place.....

THANK YOU Mr M for writing such common-sense about so many programming issues that have left me scratching my head for years!

It has felt like 'they' are trying to make me feel like a lesser developer because I simply haven't 'got' the point of so many of the trendy buzzwords these days. Like you I design a database first, then the app to go with it - it's the data that's important. I managed for years without any fancy OO gubbins, and every time I tried to understand just what was so wonderful about it, I could see the point in some of it but it just didn't seem to offer anything exceptional.

I've been reading your breath-of-fresh-air opinions and it was a revelation that someone knowledgeable and experienced has rejected so many of the pointless contrivances that have left me feeling underwhelmed too! I haven't needed to work on web apps so far, and have been researching how it's done - which until now had me looking at the various 'rails' approaches. Just the ORM stuff alone had me dejected! When I found your bullet-pointed list of Techniques of Irrelevant Merit that included lambdas and closures, I actually whooped for joy and punched the air

Thanks for sharing your wealth of experience with us, and making me feel less of an outcast. I wasn't looking for 'validation' and an excuse to not learn new things, it's more like I've tried to see what all the fuss has been about and just couldn't see any fantastic new ways to enhance my life. So, thank you!

I'm still tempted to look for a completely Free (in both senses) Way, so I'm not sure I'll be a customer (and I'm a bit borderline in experience which means "it's not for you" but thanks for Telling It Like It Is) - but at least I've learnt a few useful things, and I could play with it all for nothing and perhaps convince my superiors that the commercial licence is good value - it looks that way compared to many alternatives, that's for sure. I don't object to paying for the benefit of all your work, it's a matter of persuading others.

Best wishes, and thanks again for the enjoyable reading - most uplifting!

Subject: Re: Thank you!
Posted by AJM on Thu, 18 Oct 2012 16:31:15 GMT
View Forum Message <> Reply to Message

Thanks for those kind words. It makes a change from the usual "your ideas are crap and you should be burned at the stake" nonsense that I get from so many people who think that my approach is too simple.

As for a commercial license, you don't need one if you create an application in-house that you use in-house. If you create a package that you then sell on to others then you will require a

commercial licence.

Subject: Re: Thank you!
Posted by Nebula on Fri, 19 Oct 2012 08:47:58 GMT
View Forum Message <> Reply to Message

Cheers.

I actually wish you'd collect all your writings together and produce a book, even just a PDF eBook. The Actual Reality of Software Development, as it were. Not the depressing approach I see all too much of, elsewhere - "I've been told this is the Right Way and I'll spout off about it on forums as if it were my own discovery, when in fact I don't really understand the issues".

I've seen it before, in so many other areas of life, my other hobbies are full of people who are convinced that they know best because they've been conditioned into believing it, rather than figure it out for themselves. They apply what was intended as Best Practice In Most Situations even when those Most Situations were only about 60% all All Situations, and blindly apply that dogma to the other 40% without questioning.

I had assumed that in Programming, full of the collected wisdom of years of applied thinking from people far more intelligent than myself, perhaps this time that dogma really had something going for it. So, if I found it hard to see how various techniques really would help me, somehow it was my fault for being thick. It's a revelation to see someone achieving great results by discarding things that don't really help, and producing something that works - without any of the downsides that The Better Way was supposed to solve.

You've given me a well needed boost of confidence; that if I produce something that works, was quick to develop, is clearly understood, documented, easily maintained, and has no greater chance of flaws than if it were produced using other techniques - why worry? The customer is happy, the devs are happy. If some idiot comes along later and turns his nose up at my coding, why the heck should I actually let it bother me? I don't assume that I'll ever be a ninja guru and know it all, but if the results are perfectly fine, that's all that matters. And it's certainly better than all those bazillions of other projects that get nowhere, and die halfway through

An eBook would be a great way to make your writings more accessible; help spread the word. I tend to do a lot of reading offline, and I'd need to do a 'wget' to scrape the website to download all your wisdom to read in peace, away from it all. I think in many instances that websites are old hat, and self-publishing downloadable documents is a better alternative. Or at least number the webpages with an index so we can be sure we've found it all

Just be careful that Your Way doesn't get too popular, or it will end up as dogma itself

Thanks again

Subject: Re: Thank you!

Posted by Nebula on Mon, 22 Oct 2012 10:04:05 GMT
View Forum Message <> Reply to Message

Quick question -
I found another site daring to question OO as 'oversold', supporting Table Oriented Programming instead (yay!)
http://www.geocities.com/tablizer/

- but I thought GeoCities was defunct? How come one user's geocities page lives on?

Subject: Re: Thank you!
Posted by jojohuahua on Sat, 17 Nov 2012 06:11:40 GMT
View Forum Message <> Reply to Message

I totally agree with what is said above!!

Subject: Re: Thank you!
Posted by Nebula on Sat, 17 Nov 2012 19:45:16 GMT
View Forum Message <> Reply to Message

Thanks.

I haven't been here for a while, I've been reading up on various Web App Frameworks, and it's starting to sink in - what the advantages are.
I'm no great designer, I haven't a hope in making a nice looking web app, so it's appealing to use something where all that is automatic and I get cool design for free.

Still hard to choose between Python(Django), RubyOnR, or good old PHP. Frankly I don't really need a general purpose interpreted scripting language, I can either write a batch/Powershell/AutoHotKey/bash script for simple operations, or leap into Visual Studio or Lazarus(FreePascal) for anything calling for a GUI.

So, while Python and Ruby are interesting, I just don't need them in my life. PHP was made for web apps, just seems like the right tool for the job - popular and capable despite the naysayers who call it an abomination.

It's irritating how many frameworks only offer mailing lists for support, and not forums - so that narrows the choice. And the Beginer's Guide documentation is usually appalling!

I've got as far as understanding the concepts of MVC, passing arrays back and forth between the classes, the command line stuff that sets up the required files and folders, 'scaffolding', and gives you an 'admin' app for free. But then it all goes a bit tricky as it's a LOT of studying to get to grips with how things are really achieved, and it's a real struggle to wade through documentation that isn't structured and grouped into logical categories in a way that would make learning easy.

So far I have three PHP frameworks on my shortlist, and it's hard to choose between them. I won't

say what they are, in case it's against forum sensibilities

Subject: Re: Thank you!
Posted by AJM on Sun, 18 Nov 2012 10:40:26 GMT
View Forum Message <> Reply to Message

You should realise that there is a difference between a front end "web site" (what vistors see) and a back end "web application" (everything). For example, an e-commerce website has only two functions - display products and accept sales - while the back end application has to do everything to support the front end, such product maintenance, customer maintenance, order maintenance, invoicing, inventory, shipping et cetera. The back office application provides the complete infrastructure while the front end website is just a small window into that infrastructure.

RADICORE was designed specifically for back end applications, but its use of the 3 Tier Architecture means that all the business logic and data access logic can be shared by any number of front end websites which then become thin clients to the back end server.

Subject: Re: Thank you!
Posted by Nebula on Mon, 19 Nov 2012 09:49:42 GMT
View Forum Message <> Reply to Message

Thanks,
yes I'm aware of that. In the last couple of months of part-time investigations, it seems that there are two approaches for web (even if limited to commercial in-house intranets) applications...

1) The 'Rails' type of web app framework, a page-by-page multiple URL HTML Get/Post kind of affair - where the dev effort is put into the server side, and one relies upon it generating HTML back to the browser such that the programmer doesn't have to put any scripted intelligence into the rendered page. The actual brains of the app lives in OOP class code on the server app, 'controller' code deciding what is required based upon the web address and param.s, calling 'model' code to deal with data, and then calling 'view' code to render the HTML to be shown to the user. Seems fair enough, although usually very badly explained in the documentation!

2) RIA (Rich Internet App) using a lot of Javascript in the HTML (obfuscated to make it hard to examine the code - but still vulnerable to reverse engineering and attack) to create a single page emulation of a traditional native app, using a supplied library of GUI controls.

The second seems to have another problem that validation efforts are duplicated; one has to sanitise user inputs on the front end, and again check on the server side that anything passed to it is sensible. I was looking at Qooxdoo briefly for that approach, but got hopelessly lost and decided that normal web frameworks might be better for this newb

Frankly I think it's still going to be some time before I'm comfortable with any approach, looking at PHP frameworks - even the best documentation has me lost and asking questions that simply aren't addressed... even in the first paragraph of the 'basics' introduction!