

Re: CoBOL moved to OO

Source: <http://coding.derkeiler.com/Archive/Cobol/comp.lang.cobol/2003-12/0978.html>

From: Peter E.C. Dashwood (*dashwood_at_externet.co.nz*)

Date: 12/24/03

Date: Thu, 25 Dec 2003 00:09:46 +1300

"Judson McClendon" <judmc@sunvaley0.com> wrote in message news:b15f52de7cc255400373dfa35846ecab@news.teranews.com...
> "*Peter E.C. Dashwood*" <dashwood@externet.co.nz> wrote:
> >
> > *But you are arguing with people who have no idea what they're talking about,*
> > *have never been involved in successful OO implementation, do not understand*
> > *the basic fundamentals, (witness Judson's post equating "iterative" to "procedural"), and are resisting to the last ditch anything that looks like*
> > *it might threaten their comfortable little world, predicated on "What we know", "What we have always done", and "Best practice (as we perceive to it)..."*.
>
> *You cannot be referring to me, because I have a completely different set of reasons for objecting to an "OO and nothing else" mentality. I have said before, and I reiterate, simply show me some statistical data that show an *across the board* improvement in development time, development costs, maintenance costs and training costs, and I will be happy to endorse an *across the board* move to OO. Without that, you argue simply from personal opinion, as you accuse others of doing.*

All (non-academic) argument stems from personal opinion and there is nothing wrong with that. I don't mind you having an opinion, I mind when you set it up as Holy Writ.

And, for the last time, I DON'T CARE whether you use OO or not! I care that you denigrate it from an uninformed position.

When you have actually implemented a true OO project, when you have implemented ANY project that isn't governed by the "Waterfall",

I really have no idea why you would do this, but I respect your right to do so.

To keep harping about how you will be persuaded by the statistics is just

sad. You won't be. Your mind is closed on it. I could show you sets of statistics regarding OO, based on my own experience, both with my own software development and on other commercial installations. It wouldn't make a jot of difference, any more than I care about your statistics for the "superiority" of Procedural Code. It is irrelevant to both of us, so there is no point in continuing this discussion.

> *Within 10 years after structured techniques were in use, I could have buried you under tons of such statistics in support of structured techniques. Structured techniques were clearly and decisively superior to previous techniques in every measurable way, except possibly in total lines of code. In fact, contrary to the trend we discuss, significant*
> *such data was available BEFORE the push toward structured. Now, after much longer trials, I have yet to see any such objective data to confirm a wholesale move toward OO. Perhaps I've been overlooking it, so kindly point me to it. :-)*
>

No, I absolutely agree with you on that. OO has been firmly rejected by the COBOL establishment. I find that sad, but I am enough of a Realist to recognize that such is the case.

However, If you look OUTSIDE COBOL, you find a completely different story. The REST of the World accepts OO as necessary and fundamental for the systems of tomorrow. So much so that the Operating systems, Network software, and Web Scripting languages are ALL OO. It is taken as read, and nobody outside of this forum would even bother to argue it whether it is superior or not. (I am probably one of the last, and I am heartily sick of it...<G>).

> *Please note that none of the above involves "understanding the basic concepts", "resisting to the last ditch", threatening any "comfortable little world", or any "supposed knowledge".*

Yes it does. It involves all of the above. But, like I said, I really don't care and this will be my last post on it.

> *If OO is as good as you say, then fine, show me the statistical data to back it up, based on data obtained from real world projects.*

Why? I DON'T CARE!!!

You have missed the whole point of my argument which is only incidentally about the "superiority" of OO. My "defense" of OO is based on an attack on it by someone who is unqualified to make such an attack. You have never learned it or used it, yet you can dismiss it due to lack of statistics regarding its superiority, or the fact that most COBOL installations don't use it, so it must be crap.

I believe the lack of uptake of OO by COBOL installations says more about the people running such installations than it does about Object Oriented Programming. I have run more than half a dozen successful pilot projects using OO on sites that were "COBOL shops". Not one of them decided to move to it, DESPITE the fact that their own programmers wanted to do so, DESPITE the fact that there were measurable productivity gains, and DESPITE the fact that all these projects were on-time and on-budget. (To be fair, they all used RAD type techniques for PM, as well as OO for programming, and it is very likely that the results were not just entirely because of OO.) In every case the decisions to maintain the Staus Quo were based on politics. (One Senior Manager admitted as much to me over a few beers.) I have no complaints; I was well paid for what I did, but I still feel sorry no real progress was made.

I don't hold you personally responsible for these problems, Judson, neither am I venting my frustration on you<G>. But I would sure like it a lot better if you had actually worked with OO before you start slagging it off.

> *Considering the truly vast*
> *expenditures in money and manpower that have been invested in a*
> *wholesale move toward OO, everyone should be constantly tripping*
> *over such data, because investing that kind of resources in anything*
> *less than clearly demonstrably superior is, as I have been saying,*
> *nuts. :-)*

I'm sure the rest of world will turn around one day and say: "Judson was right. We are all nuts..."

Fortunately I'll be dead by then.

Pete.