

Re: There has to be a better way to develop web applications.

Re: There has to be a better way to develop web applications.

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2005-07/msg04438.html>

- *From:* "KJ" <KJ@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 27 Jul 2005 22:11:01 -0700
-

Thank you cbwardsr!!

As I read through this thread I felt something was missing, was not sure if I was missing some point or concept, but it didn't feel like anyone was getting to the root. There were a couple of posts that eluded to it; but I feel that yours put into words what I felt the best.

It's not always a question of having to learn the latest new language or even platform. It has to do with "can I accomplish what I need to with the toolset that I have at my disposal." It's nice to know everything but not necessary. And if your learning something just because it is the latest I question why your doing it.

I've worked for a company that continues to always be a day late and a dollar short on the cutting edge of technology but for what they were trying to accomplish they had the right tools for what they needed. I tried to explain how OO methodologies would improve development cycles and patterns would lead to reuse, but the managers at the time did not see it. Even now one manager still feels VB is not a programmers language, I guess it has to be C++ and only C++ (not sure if he even knows what .NET is). But for developing business applications/tools, VB is way more productive language than any "programmer languages" that I've had the pleasure of learning (including Java). Looking back I understand why they didn't want to go down the OO path at the time. It was new and had a cost that they could not justify. But they're going down that road now and to them it's justified as they are pulling together their global business processes.

Anyways, I don't want to ramble and I hope I've stayed on topic. I just want to say that web development is as complex as you want to make it. If you don't want complex and can live with a tool such as Coldfusion, all the power to you. If you need to be on the cutting edge, for what ever reason, then understand that; yes it is a steep learning curve and yes you will need to learn every tom dick and harry language, standard, and platform so you can make solid choices on the path you take to developement. This is one thing that I think Rob is missing the point on. No one company has the ability to drive the web to a single development source.

Re: There has to be a better way to develop web applications.

Re: There has to be a better way to develop web applications.

BTW I do all my personal development with .NET (since I love the platform) and I work with (almost) everything else on the job (excluding .NET).

KJ

"cbwardsr" wrote:

- > I have been reading this thread with some interest but also noting that in
- > the context being discussed all of the arguments are on technology and not
- > on how to use the technology to increase productivity and to provide value to
- > our customers. Maybe that sounds strange to some of u, but as a MS Architect
- > I know this to be true and I've seen it on so many projects.
- >
- > It's is my experience, 25+ yrs, that most developers will continue to
- > re-invent the wheel for each project. Remember, 'I don't know why he did
- > that but I can do it much better'. The oft used word 'reuse' is generally
- > thought to be reusing business objects, but I suggest the benefits of reuse
- > are really felt in reusing code bases and coding patterns, they are
- > repeatable therefore prompting reuse.
- >
- > No and I mean no technology or application development will be productive
- > without interaction with the customer, understanding what they are asking for
- > and having them validate your design and implementation. The best thing a
- > developer can do to increase productivity, technology independent, is a good
- > design that is validated against use cases. I'm of the opinion that with
- > OOA/OOD you need to perform a 'standard' OOD then implement that design. It
- > will make the engineer more productive!!
- >
- > What does this have to do with the technology discussion we have here? Many
- > of the issues that have been discussed can be overcome with design goals. In
- > other words, the design will determine which technology and how it is to be
- > developed.
- >
- > Additionally, I have been developing, specifically, with MS technologies
- > since late 80s. At that time I came from the mainframe era and with the
- > coming of events, properties and object orientation I knew that a revolution
- > in development was on the horizon and still is with the advent of Object
- > oriented technologies. The speed with which development can occur has
- > increased many fold. In 2000 I designed and led the development of a 3M
- > project, for over 20K pensioners and their families paying out over 345M for
- > over two yrs. I did this with 5 developers and 4 SMEs; I would sure like to
- > see you do this with Java platform.
- >
- > Despite what some has said on this post, .Net platform creation was not
- > developed based on Java, remember Java is a language not a platform. The JVM
- > and Java application servers are the platform. If you want to argue the
- > benefits of one language vs another I don't think you do that in the context
- > of the platform. if you will look at the
- > distributed architecture for both platforms you will discover how different
- > these two platforms are.
- >

Re: There has to be a better way to develop web applications.

Re: There has to be a better way to develop web applications.

- > As was said, somewhere above, unless you get rid of W3, html, etc. aren't
- > going away and that is a good thing, no one company controls the standards.
- >
- > Learning curve; .Net supports multiple languages not because of portability,
- > as was mentioned, but for reduced training costs to move into object oriented
- > programming. Do you think it is more expensive to teach a cobol developer to
- > code C# or have the cobol developer use the .Net version of cobol? This makes
- > a better case for this platform rather than a platform that locks a developer
- > into one language, now that is control; anyway portability in Java is a
- > misnomer. Every platform you go to has a different JVM and different
- > application server, all of these require compiler changes to run on that
- > platform and some design changes
- >
- > Since I was using VB6 in an object oriented way my trip to dotnet wasn't a
- > huge learning curve, I already understood object orientation and how to use
- > it. Example, use callback in VB6 and use delegates with events in .Net to
- > have callback capability, Remoting replaced DCOM, is that a bad thing, no it
- > is not, it did improve on the way DCOM worked, primarily allowing remote
- > communication thru a firewall.
- >
- > As of this writing I can't think of very many things that I can't do with
- > .Net whether or not it is C# (or some other language), asp.net, etc. .Net
- > gives me access to windows APIs which means I can do anything the windows
- > server will allow me to do.
- > I'm sorry but imho, this thread is much to do about nothing; with the
- > exception that it allows the religious among us to proselytize about our
- > beliefs of which languages are better. Engineers should worry how do I apply
- > the technology rather than the state of the development world. It is like
- > abortion, everyone has their opinion so we never come to completion or
- > agreement.
- >
- > Regarding productivity in VS: I use two add-ins to add additional
- > functionality to VS that is useful and adds to VS' already awesome IDE. The
- > first is from JetBrains and is called ReSharper and their profiler, sorry but
- > I don't remember that name. the other product has to do with my architectural
- > duties (design) and it is visual-paradigm's SDE, software development
- > environment. It allows me to do a complete OOA/OOD and have it generate
- > code, even code with implementation in it. You can reverse engineer to keep
- > your model up to date and you can regenerate and not impact your
- > implementation. This capability gives a developer the capability to keep
- > technical specs up to date with code.
- >
- > Regarding the '90' languages on a resume. I'm one of those who has multiple
- > languages on his resume and it is not that I have used many languages but
- > demonstrates to the interviewer that I can go into many environments and pick
- > up and understand what is going on easier than those who haven't experienced
- > such varied environments. It is not a matter of numbers, maybe I have been
- > interviewing more than most of you, after all I'm a consultant, but this info
- > is very valuable in getting interviews.
- >
- > "Christopher Hansen" wrote:

Re: There has to be a better way to develop web applications.

Re: There has to be a better way to develop web applications.

>
>> The problem is multi-faceted. Disparate technology platforms, the hunger for
>> companies to have loyal followers(read: internal champions and customers),
>> the changing nature of technology in general.
>>
>> The problem with the concept of RAD is that, frankly, it's never happened.
>> By that I mean, RAD dictates that your engineers, developers, architects,
>> implementation and support people are already a well-oiled machine. If this
>> is the case, then you don't necessarily need to standardize on any single
>> vendor platform – it could be anything. But practically speaking, Microsoft
>> seems to have a major foothold in the server / web development marketplace
>> by virtue of the fact that IIS 6.0 comes bundled for free with WS03.
>>
>> Another observation in terms of the technological learning / use curve is
>> that RARELY does a program manager / client / project have a clear and
>> concise framework around what THEY want to achieve. How many times have I
>> heard "I'd like a Button that I can click...(that will essentially run my
>> super-complex business/role). That's certainly contributes to
>> cross-pollenating across 15 different technology iterations. IT professionals
>> use whatever is in the toolbox to get the job done. More often than not, by
>> design, the products (even freebies) are inter-related and dependent. This is
>> done to drive business. If you think tht this is all just R&D – think again.
>> We (sic.) are being led in a very specific direction, one bite at a time.
>> Companies are the ones that are forced to absorb the "hit"; the actual people
>> who suffer are the IT professionals who see the pervasive interpretation of
>> the business community at large of their roles as hemorrhaging expenses and
>> disfunctional agents of change. This is precisely why some 50% of the
>> business community was still using Windows NT 4.0 as of 2 years ago.
>>
>> Software comapnies (and hence developers) need to slow down and put out
>> something that has some serious meat to it – and stick with it. I mean a
>> finished product. Good examples of this busines model can be found with Adobe
>> and Macromedia – both of whom are th only companies I see as having pushed
>> finsihed products suites. You'll notice, as a result, that they don't have a
>> new version every 18 months. What they have currently rocks, needs little
>> tweakage, and it essentially a customizable, scalable platform (ColdFusion /
>> J2EE anyone??). This are highly under-rated technologies, imo.
>>
>> If the development community as a whole demands that their be business value
>> in requiring our time to absorb/use/champion the use of new technologies,
>> then it will require software manufacturers to make better products – and
>> hence, stop wasting our time (as in the original poster's commentary). If
>> we're all going the "geek" path of the newest, latest, greatest "toys" on the
>> block (e,g, software, tools, etc.) then there's no substantial reason for
>> software manufacturers to hold to the aforementioned model. I put the owness
>> back on us as developers and champions of our individual talents and tools.
>>
>>
>> "RCS" wrote:
>>
>>> I don't know if you're serious – but I'll respond anyhow.

Re: There has to be a better way to develop web applications.

Re: There has to be a better way to develop web applications.

>>>

>>> This is like anything else you'd do with programming – you have to take some
>>> time, to set up some templates/standards/etc – so that you can later "crank
>>> out" applications with ease. We needed to do the same thing with a sort of a
>>> tab/subtab layout – that we've used for the past 6+ years in ASP. We
>>> initially spent about 4 months making a tabbed-type layout that was
>>> extensible and integrated with our single-signon.. so that after that, it
>>> was a matter of making up a logo, changing the CSS classes, layout your
>>> tabs – and start development.

>>>

>>> So – these technologies aren't so much the enemy – as they are your friend –
>>> to help you do your job. And it only behooves you to know these. It doesn't
>>> matter if it were big technology called "Bob" with 8 sub-categories – or if
>>> it's called 8 different names. The things you mentioned below are the
>>> elements of what it takes to write applications.

>>>

>>> As for XML overhead – I think you have have needed to come through the "flat
>>> file" era of computing to see the advantage. For years (and sadly, still
>>> today) – companies pass text files (like comma-separated or tab-separated
>>> text files) back and forth. Think about this. How do you get that data into
>>> your database? Well – most would start by writing a parser – to turn those
>>> "hopefully it's the right format" file, into meaningful database fields.
>>> Most people write what I call a "write-once/use-once" parser. You write a
>>> parser specific to that file. You could obviously also spend time making a
>>> more flexible parser, but it needs to be robust. The core problem with a
>>> hand-written parser is that it not only needs to handle all the things you
>>> can think of... it also needs to handle the things you can't think of.
>>> Like – what if you get an end-of-file right in the middle of a record.. what
>>> if there is text in the presumed date/time field.. etc, etc, etc, etc. So
>>> for people who lived through that headache – XML is just staggering. Yes –
>>> there is a little bit of size overhead (not that it matters in these days of
>>> nearly unlimited RAM and disk space) – but there is one parser. The XML
>>> parser. Which, right off the bat – if it's not valid XML, it won't even
>>> load!! Then, you can apply a schema – and XML will check to make sure you
>>> have "exactly one" of these elements and one or more of these elements. So
>>> it's become an infinitely more efficient way to process foreign files – and
>>> files from different computing environments.

>>>

>>> HTML forms ARE bad. Especially in an ASP.NET environment. The main reason,
>>> is that it becomes very difficult to keep "user state". That is, "who is
>>> this user and exactly what are they doing". Which, on a single ASP page for
>>> example, is 100% controllable. If you have frames, it would take a
>>> significant chunk of work – to tell the navigation frame if the current
>>> frame is doing anything that should make the navigation change. Because –
>>> the navigation has absolutely-no-idea what is going on in the other frames.
>>> So it's a false sense of integration – it looks like the frames make up a
>>> whole website – but really, it's a couple of completely unrelated pages –
>>> that if you wanted to make them related – would take a significant effort.
>>> And if you are in a situation where you want to "post" to another page in
>>> ASP.NET – I wouldn't spend my time trying to make it work, I would spend my
>>> time thinking *why* I am trying to do that, and see if there is a more

Re: There has to be a better way to develop web applications.

Re: There has to be a better way to develop web applications.

>>> natural way to accomplish what I am trying to do.
>>>
>>> Lastly, you say it's a pipedream to use ASP.NET and the .NET Framework to
>>> write applications. First, the .NET Framework is what houses the logic for
>>> ASP.NET, XML, VB.NET, ADO.NET (are they still calling it that?) and
>>> FormAuthentication. But dare I say, you almost don't need to know HTML,
>>> because the design editors are so good now – and you can even drag–and–drop
>>> your database connection/command/etc and technically not do any coding. So
>>> if you are looking for the "good–old–days of point–n–click development" like
>>> MS–Access/VB3/VB6, they've made room for you in VS.NET. But if you want to
>>> embrace these new technologies and use them to leverage your development and
>>> become a more efficient developer, they've done that too.
>>>
>>> I've been in hip–deep in .NET since it came out and I have two consistent
>>> thoughts: 1) .NET is the biggest leap in software development and developer
>>> productivity that I've ever seen (see comment below) and 2) This is a great
>>> time to be a developer.
>>>
>>> You could argue the first one, that "Java has been around for years and
>>> never got the recognition!". Well, .NET is about the technology (and yes, C#
>>> is pretty much Java) – but it's also about how it's
>>> bundled/sold/trained/etc.. Because the Java people are so caught up in being
>>> open and not locking down developers – they became so open that they can't
>>> accomplish anything. So yes, Microsofts solution is a boxed/shrink–wrapped
>>> solution and they do force the developer to use one tool, etc... but it's
>>> for his own good, because forcing everyone to limit the tools they use,
>>> enables them to make forward progress.
>>>
>>> You can either run on one system, and become an advanced, proprietary
>>> system... or you can be an open system, and spend all your time NOT using
>>> any one thing that limits you. In the end, open–systems and forward–progress
>>> are mutually exclusive.
>>>
>>>
>>>>> I realize I'm learning web development and there is a STEEP learning
>>>>> curve,
>>>>> but so far I've had to learn:
>>>>>
>>>>> HTML
>>>>> XML
>>>>> JavaScript
>>>>> ASP.NET using VB.NET
>>>>> ..NET Framework
>>>>> ADO.NET
>>>>> SSL
>>>>> FormAuthentication
>>>>> (and probably a few more things)
>>>>>
>>>>> Now call me crazy, but this hog pog of languages & technologies is
>>>>> ridiculous!! The simplest of tasks become major R&D efforts (setting the
>>>>> enable state of a control on another ASPX page in a frame for example).

Re: There has to be a better way to develop web applications.

Re: There has to be a better way to develop web applications.

>>>> And
>>>> XML, OMG that has got to be the most ineffecient way to write out data
>>>> I've
>>>> ever seen --- the overhead is staggering!! So far the research I've seen
>>>> are
>>>> "frames are evil" --- great so freakin' helpful. This is just crazy, if
>>>> the
>>>> development community has to continue on in this bizarre environment of
>>>> languages and technology, then web application development is never going
>>>> to
>>>> mature and become cost effective for companies to exploit.
>>>>
>>>> This is NOT an efficient way to get work done --- just the cost to get
>>>> developers up to speed on all the technology can doom a project from the
>>>> start. The pipe dream of using ASP.NET with VB.NET and .NET framework
>>>> ONLY
>>>> for web development is just that --- a pipe dream, for only the simplest
>>>> of
>>>> applications could anyone get away with just those three
>>>> technology/tools.
>>>>
>>>> I just don't understand --- terms such as portability get tossed around,
>>>> but
>>>> the bottom line is, if you elect to use .NET Framework and ASP.NET your
>>>> bound to MS server OS. And, if this is all done in the name of
>>>> "portability" (at the cost of performance) how often are you folks moving
>>>> servers around and changing platforms?? If platforms are changing that
>>>> frequently, that begs the question why?! It's like building something
>>>> for
>>>> 5% that may need it while the majority don't --- so the majority have to
>>>> take
>>>> the penalty. There has got to be a better way?
>>>>
>>>> Rob.
>>>>
>>>>
>>>>
>>>
>>>
>>>
.

• **References:**

- ◆ **There has to be a better way to develop web applications.**
 ◇ From: Rob R. Ainscough
- ◆ **RE: There has to be a better way to develop web applications.**
 ◇ From: balrag
- ◆ **Re: There has to be a better way to develop web applications.**
 ◇ From: RCS

Re: There has to be a better way to develop web applications.

Re: There has to be a better way to develop web applications.

- ◆ **Re: There has to be a better way to develop web applications.**
 - ◇ From: Christopher Hansen
- ◆ **Re: There has to be a better way to develop web applications.**
 - ◇ From: cbwardsr

- Prev by Date: **Re: Deploying ASP.NET in a mixed NT4 & W2K domain**
- Next by Date: **Network debugging**
- Previous by thread: **Re: There has to be a better way to develop web applications.**
- Next by thread: **dynamic image (png) file**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**