Which platform allows for more rapid development, ASP.NET webforms or MVC? [closed]

Asked 16 years, 1 month ago Modified 7 years, 4 months ago

Viewed 4k times

6

Closed. This question is opinion-based. It is not currently accepting answers.

Want to improve this question? Update the question so it can be answered with facts and citations by editing this post.

Closed 4 years ago.

Improve this question

Are there any obvious productivity gains in developing using webforms or MVC?

asp.net asp.net-mvc webforms

Share

Improve this question

Follow

edited Aug 16, 2017 at 8:56

Vadim Kotov

8,274 ● 8 ● 50 ● 63

asked Nov 8, 2008 at 13:19

helloWorld

1 When I see WebForms and MVC in a sentence together, I always wonder if MFC is meant instead or if Model-View-Controller is really meant. Even funnier is that, from the answers here, I still can't tell which it is. – orcmid Nov 8, 2008 at 18:29

11 Answers

Sorted by: Highest score (default)

9

When you are dealing with productivity of ASP.NET MVC vs. ASP.NET Web Forms, consider these two developers.

1. **Generic .NET Developer** who has extensive experience writing WinForms and WebForms applications.
2. **Generic Web Application Developer** who can write C# code, but knows HTML, CSS and jQuery/MooTools/etc like the back of their hand.

The **Generic .NET Developer** will be most efficient at writing code in the original **ASP.NET Web Forms**. This is because they know the event models very well (as they are mirrored between WinForms and WebForms) and will be able to quickly build a website using this mental model.

- MVC would have a large learning curve because it is a complete departure from his existing knowledge of

"How the web works". These are the stumbling blocks this developer will find with MVC:

- Having to learn HTML and CSS to a point where they understand what is going on and how things interact with each other.
- Learning javascript and how to call back to the server for ajax.
- Figure out how to maintain state, finding out cookies are not secure and that session has certain 'gotchas'.

The **Generic Web Application Developer** will be most efficient in **ASP.NET MVC** because the framework mirrors how they think of web applications - there are no events, no application state, etc. They already know how to write good HTML and style it using CSS, they know how to write an ajax call back to the server.

- Web Forms is not a good model for this developer because it is a departure from his existing knowledge of *"How the web works"*. These are the stumbling blocks this developer will find with ASP.NET Web Forms:
  - Time will have to be spent trying to figure out how the Page Lifecycle works.
  - ViewState will cause many problems, cause the page to bloat.
  - A lot of HttpRequest hacking usually will have to take place to fix all the bad code that comes out

of certain WebForm Controls.

- Learning how to deal with AJAX callbacks instead of just using their existing knowledge of how to call a URL from jQuery.

My main point is that different developers will have different levels of productivity based on their previous experience with web development. I personally prefer ASP.NET MVC, I can build something out very quickly and have it output exactly what I intend.

I am the #2 kind of developer, I have written web applications (most of them small) in Ruby on Rails, Django, ASP.NET Web Forms and ASP.NET MVC. The knowledge of web development between these 3 frameworks is almost interchangeable (with exception to having to know the hosting language - ruby vs. python vs. c#).

The most infuriating thing I find with ASP.NET Web Forms is that it tries to do too many things *for* the developer (things like application state, web events), the code output often does not play well with Web Standards, Web Forms 2.0 is tied directly to XHTML 1.0 Transitional. Anything that is beyond a quick and dirty DataGrid and Form to CRUD rows in a database usually causes more pain and suffering compared to writing it in ASP.NET MVC.

answered Feb 8, 2010 at 19:51

Astra
11.2k • 3 • 39 • 41

7

No. Personally, I find MVC works much more like my brain does with pretty clear separation of what goes where and why. I banged out a complete prototype commerce site with MVC in a few weeks that would've taken me, I'm sure, at least double that using webforms.

Share Improve this answer

Follow

edited Nov 8, 2008 at 13:35

Mitch Wheat
300k • 44 • 477 • 550

answered Nov 8, 2008 at 13:23

brmore
896 • 2 • 9 • 16

"MVC works much more like my brain does with pretty clear separation of what goes where and why" - I second that, I chunked out code wayy faster almost immediately after 'grogging' MVC. – chakrit Nov 8, 2008 at 13:50

7

With MVC, you never spend time trying to work out why ItemDataBound and ItemCommand aren't firing when you expect them to.

If you think that'll save you time, then MVC is probably for you.

answered Nov 8, 2008 at 13:48

Share Improve this answer

Follow

Dylan Beattie

54.1k ● 35 ● 135 ● 200

2 I hear you. WebForms seems quick when you can just drop a grid or control of some sort on a form and hook it up to a datasource. But when you are debugging events and chasing your tail it soon becomes painful. – Craig Nov 10, 2008 at 1:29

5

At this point, WebForms, is more mature. I wish, for example, that validation (client and server side) had better support in MVC. However, I fully expect that, over time, this will improve. The best thing about MVC from my perspective is that MVC makes my code much more testable. I was leaving a lot of code untested with WebForms simply because it was too difficult to test the codebehind. Now I can write unit tests for all my controller logic.

Even if it eventually turns out that MVC is somewhat slower (and I'm guessing that it won't be), enhanced testability would make it worth it. Eventually I will recoup that time, since there will be less going back and fixing buggy, untested code later.

Share Improve this answer

Follow

edited Nov 8, 2008 at 13:52

answered Nov 8, 2008 at 13:34

tvanfosson

532k ● 102 ● 699 ● 798

3

I find MVC a much more natural fit for web development. As the previous poster has noted, you'll undoubtedly save time by not having to deal with weird bugs thrown up by the web forms page execution lifecyle.

Also, the ability to write unit tests for your logic much more quickly (and with a lot less of a reliance on mocks!) is a great advantage.

Finally, I've found that it's a lot easier to write css and client script when using the MVC framework as you are able to specify your own IDs on HTML elements.

Share Improve this answer

Follow

answered Nov 10, 2008 at 0:47

user32326

437 ● 1 ● 6 ● 6

3

When I hear the term "rapid development", I find that people typically use it in the context of "how fast can I develop a new solution". There are a lot of factors that must be considered before a reasonable answer can be constructed.

- How familiar is the developer with the framework?
- How comprehensive is the design?
- How complex are the requirements?
- How stable is the framework?

I first came in contact with ASP.NET MVC shortly before 1.0 was released. Prior to that, I was very familiar and very comfortable with ASP.NET WebForms, but I was still frustrated with the development process as a whole. I was aware of the goals surrounding separation of concerns, and knew enough that clear separation wasn't possible with WebForms without swimming against the current created by the framework.

Since working with ASP.NET MVC, I found very quickly that I was missing the boat on a number of things. While WebForms allows you to build things "faster", you are basically held hostage by Microsoft's implementation rather than allowing you to more easily digest established standards (such as JavaScript, CSS and AJAX) in environments that don't involve Microsoft. Also, the creation of new tools, behaviors and functionality should not be motivated by sales and profitability, but because the technological demands of the developer community require it. I've been working on this current MVC project since April. I enjoy working with this framework very much, and would recommend it highly to anyone and everyone interested in moving away from WebForms. Learning the framework takes a bit of time, but once you understand it, you literally wonder why you couldn't have done things this way from the beginning.

You can build applications faster with WebForms, but, if you want professional looking websites, you will have to invest in component libraries that will still require you to learn those established standards in order to leverage

them properly. If my 11-year-old son expressed interest in learning web development tomorrow, given the choice of WebForms or MVC, my choice would be MVC. That being said, I would still direct him towards learning JavaScript, jQuery and AJAX before he even touched MVC, because understanding those frameworks makes understanding almost every other that much easier.

Personally, I am not an advocate of "rapid development". I have spent my career as a corporate developer, and view in-house development as an investment. I would prefer spending 20% more time in design and development instead of shorting the project by 20% just to meet an unrealistic deadline. Each dollar "saved" during initial development will easily cost you at least $1.50 due to maintenance costs, re-education, and architectural changes due to new requirements. But, not everyone thinks as I do, so...my simple answer would be MVC.

Share Improve this answer

Follow

answered Jan 31, 2010 at 23:45

Neil T.

3,319 ● 1 ● 27 ● 31

1

I think eventually MVC will be the clear winner for Web Developers. However ASP.Net will retain a very important role because there will always be .NET guys that consider themselves web developers who are really .NET winform guys. ASP.NET sans MVC lets you throw stuff together really quickly, but you can easily end up with horrible,

terrible, unholy HTML, CSS and javascript. MVC opens the door for wonderfulness but requires a more disciplined approach by developers.

Share Improve this answer

Follow

answered Nov 10, 2008 at 1:06

Tad Donaghe

6,588 ● 1 ● 31 ● 64

1 You should clarify you mean ASP.NET webforms. MVC is also built on the ASP.NET stack – Jace Rhea Oct 7, 2010 at 17:10

0

When it comes to the first 90% of development, WebForms is the clear winner. It's that last 10% that's the problem and where MVC has the advantage.

Share Improve this answer

Follow

answered Feb 1, 2010 at 0:10

Dan Diplo

25.3k ● 4 ● 70 ● 90

2 ..where the last 10% takes the other 90% of your time. – Jace Rhea Oct 7, 2010 at 17:06

0

It's very strange why so many people think that it's not possible in Web Froms to get clean HTML, use jquery (and jquery ajax), write css (?). That you must always to drag and drop controls from toolbar, when actually you can write it same way as you do in MVC.

Webforms page lifecycle, viewstate, databinding, updatepanels also require time to learn if you want to write good code. It seems most people doesn't spend time for this, that's why they say that WebForms "not so good".

While I'm mature Web Froms developer I'm trying to learn MVC (just for fun) and it's good enought to write complex websites. But as I see now, main problem with MVC is that you have to write a lot of code, really a lot of!!!

Share Improve this answer

Follow

answered Aug 5, 2011 at 5:27

Alexey Smolyakov

840 ● 7 ● 6

0

I've been using ASP.NET since the first beta was handed out at the Professional Developers Conference in Orlando in 2000. So obviously, I'm much more comfortable with WebForms, beause that's what I started with. Most all of the objections to WebForms can be overcome with sensible development mechanisms. But, even though WebForms provides an infrastructure for RAD on the web, I like MVC as well. MVC is still maturing, and I'm sure it will get even better. The bottom

line is "where are you" on the development learning timescale, and what previous experience do you have. You have to make that determination for yourself. If you're new, you have the luxury of investigating both.

Share Improve this answer

Follow

answered Nov 9, 2011 at 1:10

Peter Bromberg

1,496 ● 8 ● 11

-1

It depends on how you develop.

If you're someone who drags controls onto the designer surface, it would probably be hard to ever match that level of rapid development with MVC.

On the other hand, if you spend most of your time in source view, you'll probably find that MVC allows you to do so more efficiently.

Bringing AJAX into the picture, if you're someone who relies on the UpdatePanel, MVC would probably be unspeakably miserable for you.

(This doesn't begin to address the subjective issue of which type of development is more "correct". I don't believe that's the point of this question though.)

Share Improve this answer

Follow

answered Nov 10, 2008 at 1:01

Dave Ward

60.5k ● 14 ● 118 ● 134