

Website version 2, keep in PHP or move to .Net? [closed]

Asked 16 years, 1 month ago Modified 16 years ago

Viewed 702 times  Part of [PHP](#) Collective



9



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 6 years ago.

[Improve this question](#)

I have a website built in PHP 4 with a framework made by hand by me. The code is 3 years old and I am limited (well it requires a lot of effort to make changes).

I decided to do new version of this website. My knowledge has since increased, and now I know that a lot of frameworks exist and that **IOC** is there and **ORM** too... The problem is that in all my research, **I do not find something that satisfies me completely in PHP**. On the other hand, at my job (mostly in .Net winform 2.0) I see many good IOC and other solutions possible, and, I

think that .Net is more mature for developing software with best-practices.

I haven't found many good IOC for PHP. As for ORM, I have seen Propel, which looks nice, but haven't found a complete Framework too (Symfony documentation isn't up-to-date, contains lot of error; Zend looks too big I think...), etc.

Do you suggest that I simply try another framework and keep the website in PHP, or would it would be a good idea to switch to .Net?

Reasons of this question

- The system works but has begun to slow down (since it has more users). The database is normalized too much... etc.
- The code isn't bad - business logic is separated of the HTML - but the problem is that controllers are inside the view so it's harder and harder to make some layout changes.
- No unit tests have been done, and it's very "stressful" to make change.
- There's no logging in the system, and it's hard to get a snapshot to see if everything is working properly.

More information (added the 3 nov 2008)

- I know .Net 2.0 this is why my alternative is .Net and not RoR.
- I were suggesting an other language not for the speed issue but for the tool/framework available.
- It's slow because of the database normalisation, too many table that are great in theory but not in practice (we learn from our mistake!)
- It's an hobby, but I take it seriously cause I have over 25 000 users registered since the beginning (not all active).

Current suggestion from people over here

I think I'll keep what I have done and try to do some refactoring slowly. I am surprise that most of the vote have been to keep the current system but well :P this might be the simpler for me for the short term.

Thx to **nickf** who has spell check all my text.

PHP .net php frameworks

Share

Improve this question

Follow

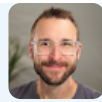
edited Jun 20, 2020 at 9:12



Community Bot

1 • 1

asked Nov 3, 2008 at 0:29



Patrick Desjardins

141k ● 89 ● 294 ● 346

Most other web languages make it difficult to *not* use their standard framework. PHP is the opposite - but that means it doesn't have a standard framework, and as a result, most are pretty pedestrian. – [staticsan](#) Nov 6, 2008 at 22:37

8 Answers

Sorted by:

Highest score (default)



15



If you made the whole framework yourself, I would suggest you just upgrade it for PHP 5 and go forward from there. Most PHP 4 code will "just work" in PHP 5; the exceptions are code that uses the new reserved words, and code that relies on the way PHP 4 differs from PHP 5 -- which means a few things in classes and references at the edge of PHP 4's capabilities.

Share Improve this answer

answered Nov 3, 2008 at 0:34

Follow



[staticsan](#)

30.5k ● 5 ● 63 ● 73

@daok, if you already have a working site, you can replace parts of it step by step with new/improved code. I'm doing just that, section by section from some old PHP4 to ZendFramework. New parts of the site get coded direct in PHP4/ZF. – [Alistar Bulman](#) Nov 3, 2008 at 18:08



If it is a hobby project and you feel a bit unhappy about the current state of the project I'd say: definitely try .net -

3



for the same reason I would suggest jsp or almost any other kind of feasible language/platform/runtime: the fun and the experience. Even if after some time you decide to revert to php you'll keep the "new perspective" of how things can be done. On the other hand it can be difficult to let go of the old code and methods. A complete rewrite is hard because at first you lose all the bug fixes, the little tweaks and tricks - back to square one. And usually you don't get the exact same result. But if you are going to build an (improved) version 2.0 anyway and want to stretch yourself ...go ahead, try something new. Even if you fail, at least you have something to blog about on your site :)

edit: It would also be possible to port only parts of the code or develop the new fetatures in .net - as long as can "bear" the two seperated codebases. And you might also want to look into <http://www.codeplex.com/Phalanger>, a .net compiler for php. (Haven't tried that yet.)

Share Improve this answer

edited Nov 6, 2008 at 14:10

Follow

answered Nov 3, 2008 at 1:35



VolkerK

96.1k ● 20 ● 167 ● 231

I know .Net (not 3.0 but 2.0 well), and I cannot fail because it's a website that has already a lot of people on it. Thx for the info +1. – [Patrick Desjardins](#) Nov 3, 2008 at 12:49



2

To respond to your extra information, moving from PHP 4 to PHP 5 is probably a golden time to rewrite your framework to address some of the problems it has.



You can also use the opportunity to increase layer separation, too. For instance, you say it has some slowdowns, possibly related to over-normalization. Well, if your framework is abstracting data access away enough, you should be able to add some caching (such as memcached) to that layer and the application doesn't need to know anything.

You could also use the same effort to add logging in a fairly easy way. If you use an object layer, you can add logging to the generic data storage and get a whole lot of logging practically for free.

Other slowdowns might be due to inadequate database indexing. And this is unrelated to your framework. If you are using MySQL, turn slow query logging on and run EXPLAIN over some of them. You should see which columns need indexing that aren't.

PHP in and of itself is not a slow language. Slowness will be elsewhere. :-)

Share Improve this answer

Follow

answered Nov 3, 2008 at 5:20



[staticsan](#)

30.5k ● 5 ● 63 ● 73

You are right, I need to work on the database, but still, I have other problem about the maintainability that need to be fixed.
+1 for the information! – [Patrick Desjardins](#) Nov 3, 2008 at 12:50



1

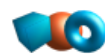


Depending on the amount of code, it might be a good idea to stay in PHP. It would basically be a complete rewrite if you decided to move to .Net. Because you would be using a completely different framework, with a completely different language, under a completely different development paradigm, there isn't a whole lot you could bring with you.

Share Improve this answer

answered Nov 3, 2008 at 1:09

Follow



[Kibbee](#)

66.1k ● 28 ● 144 ● 184

You are right that a lot of thing won't be portable. This is why I am not sure about what move to do. My concern with PHP is that I have developed with .Net since 3 years and about 5 years with PHP and I find .Net really stronger for maintainability. But I appreciate your advice, +1!
– [Patrick Desjardins](#) Nov 3, 2008 at 12:52



1



I strongly suggest ASP.NET MVC with LINQ to SQL or LINQ to Entities as ORM.

Share Improve this answer

answered Nov 3, 2008 at 2:05

Follow



[Andrei Rînea](#)

20.7k ● 18 ● 121 ● 169

I got some bad review with Entities, how is the learning curve of ASP.NET? – [Patrick Desjardins](#) Nov 3, 2008 at 12:52

Ouch... the learning curve is not quite encouraging but once you get past it things go smooth :) Regarding LINQ to Entities, I encountered bugs :(but they'll surely be removed. I am working now on a project and I'm using LINQ to SQL (not to entities) and it works like a charm :) – [Andrei Rînea](#) Jan 5, 2009 at 23:27



1



For a good PHP ORM, you should look at [Doctrine](#). It seems to be easier to install and to write with and have a better documentation. The way to write query may remind you of LINQ.

ABout [Zend Framework](#), it is big but it is not really relevant; all components are more or less independent. It is more a library than a monolithic framework like Cake PHP or ROR:

- You don't like [Zend_Db](#), You could use [ZF](#) + [Doctrine](#).

- You don't like [Zend_Controller](#) you could try [CodeIgniter](#) + [Doctrine](#) + [ZF](#)

You can use [Zend_Openid](#) without using [ZF](#)'s MVC solution.

You can also start to use Zend in your current application without rewriting everything.

Share Improve this answer

answered Nov 3, 2008 at 13:56

Follow



[Dinoboff](#)

2,732 ● 2 ● 26 ● 26



And why are you limiting yourself to these choices? Why not try Ruby on Rails? I has a built-in ORM!

0



Share Improve this answer

answered Nov 3, 2008 at 2:08

Follow



[Pablo Marambio](#)

1,568 ● 1 ● 17 ● 29



<flamebait> one of his concerns was that PHP was too slow... </flamebait> – [nickf](#) Nov 3, 2008 at 2:54

Haha. Well there's always Django Python? – [Ty](#). Mar 20, 2009 at 13:31



I would simply try to change section by section slowly to PHP5, try to keeping up with new framework slowly but

0

not rebuild the whole website (if it's a big website).



For you database issue, you might start on development environnement to de-normalise your database where you know it's slow.



Share Improve this answer

answered Nov 26, 2008 at 15:43

Follow



Mister Dev

10.4k ● 12 ● 43 ● 33
