

# Ethics of using a "fringe" language for your job? [closed]

Asked 15 years, 10 months ago   Modified 9 years, 9 months ago

Viewed 3k times



28



As it currently stands, this question is not a good fit for our Q&A format. We expect answers to be supported by facts, references, or expertise, but this question will likely solicit debate, arguments, polling, or extended discussion. If you feel that this question can be improved and possibly reopened, [visit the help center](#) for guidance.

Closed 12 years ago.

Just want to ask for some opinions here. How do you feel about using a language (and/or framework) that isn't widely used in your location to write software for a company? For instance, I live in an area dominated by .NET, with the occasional PHP job. Let's say that I'm learning Python and decide to use it to write software for my job (I'm a "Team of One" so I can pretty much use anything I want).

Now their software is written in a language that pretty much nobody in the area uses or knows; if I were to leave the company, they'd basically have nobody to maintain/add to it unless they retain on me as a

consultant. While that's really good for me, it seems a bit "crooked" - granted, that's how the business world works.

What are your thoughts?

I should mention that this is a very small company and I'm the only IT person, so I have full reign to choose our development platform. I'm not specifically using Python, but chose it as an example since my area is almost entirely .NET based; I don't care for .NET anymore though, which is why I don't want to consider using it. Also, the company is.. how shall we say... *extremely frugal* and wouldn't purchase the required resources for .NET (e.g. server licenses, SQL licenses, Visual Studio, components). I personally have an MSDN subscription but I can't use that for them.

Also FWIW there **are** people in the area who use the language I'm considering using (Ruby on Rails), but nowhere near as many people as .NET developers. It's not like I'm using something that only I know.

programming-languages

Share

edited Mar 1, 2015 at 22:30

Improve this question

Follow

community wiki

5 revs, 2 users 93%

Wayne M

- 
- 1 btw, Python is not a "fringe" language even on StackOverflow which dominated by .NET developers. – [jfs](#) Feb 4, 2009 at 15:08
- 

Yeah, personally, I wouldn't really worry about lack of developers if you're talking about Ruby on Rails or Python; those are mainstream languages and a company shouldn't have to scrape to find people who can write them. – [mqp](#) Feb 4, 2009 at 15:20

---

- 1 Okay - I was basing my calling it a "fringe" language on the amount of jobs I see advertising Rails/Python developers (zero) on the usual job boards. – [Wayne Molina](#) Feb 4, 2009 at 15:27
- 

@Wayne: It is not my intent to start a flame war, but it has been my observation that MS technologies (certifications in particular) tend to attract the lowest common denominator candidates. Perhaps they're always advertising for .NET jobs disproportionately because they can't get anyone good. – [rmeador](#) Feb 4, 2009 at 16:07

---

If you think picking a particular programming language is crooked then you have a a weird view. In reality, you're not the only RoR programmer but if it is rare in your area then it sounds like a plus for you. Maybe your problem is with advantages? Would your company give away its advantages? – [BobbyShafter](#) Feb 6, 2009 at 6:33

---

30 Answers

Sorted by:

Highest score (default)



27

You may think that this approach is good for you. But in fact all this does is paint you in to a corner. The best way to get promotion - within an organisation is to make yourself unnecessary in your current position. That might



seem like nonsense, but it is in fact true. Think of it like this, if it is essential to the company that you continue to maintain the python code you wrote for them, and they can't go to anyone else to get that skill, then they will continue to pay (maybe a little above market rates) to maintain that code.

If however, you write that code in .net where there is a plentiful supply in your area, then as the company expands and the code you've written proves successful, you will be able to hire people to maintain that code and you can move on to designing other systems. Or moving in to managing a team of .net coders - if that's your want.

Even if you want to leave, the best thing for your career is going to be to get the best possible reference. To do that, write them some code that is easy to maintain. Help them hire someone to replace you to maintain it. They will be grateful and recommend you as a consultant to their friends.

Code in something esoteric - for which there is little support in your area - and they will be saying to their friends on the golf course "no don't hire that guy, he wrote this system for us which does the job, but no one else can maintain it. We're stuck with him forever and now he's too busy to look after us properly!"

Do what's best for the business, not what might be of most interest to you - or appear that way on the face of it. You'll win out in the long run.

community wiki

[Sam Meldrum](#)

- 
- 1 +1 several times over for the "make yourself unnecessary" comment. There are way too many information hoarders out there and it truly helps no one. – [Austin Salonen](#) Feb 4, 2009 at 15:10
- 

Great advice but not really applicable since he is the entire IT staff at the company. What would he be promoted to?  
– [jmucchiello](#) Feb 4, 2009 at 16:25

---

@joe\_mucchiello - it's not just about current circumstances. Hopefully the firm grows and he becomes the manager of a new junior rather than the reverse. Or he gets a good reference for his next job. – [Sam Meldrum](#) Feb 4, 2009 at 16:29

---

Right, Sam, but if the IT department grows under him he can hire somebody who knows the language he's using, and he's not likely to get a bad reference for doing something nobody else quite understands. – [David Thornley](#) Feb 4, 2009 at 17:18

---

I disagree with the "make yourself unnecessary" comment. You've got to make yourself indispensable! Write simple code thats easy to maintain (so no fringe language) do it quickly at high quality and move on to the next project. Do this and you become indispensable and will get promoted!  
– [Stephen Doyle](#) Feb 4, 2009 at 23:58

---



18



I think that you're responsible to decide on the language that's best suited for the job. That includes an objective evaluation of the merits of the language and framework, it includes your own personal skill with the language (since you're the one doing the work) and it includes maintainability by others. Only you and your company can decide how much importance to place on each of those.

Share Improve this answer

answered Feb 4, 2009 at 14:39

Follow

community wiki  
[mqp](#)



8



For your own personal development, if your area is dominated by .net, why don't you want to get up to speed in that instead of Python?

From an ethical standpoint, I would not write something that could not easily be maintained by someone else.

Share Improve this answer

answered Feb 4, 2009 at 14:42

Follow

community wiki  
[HLGEM](#)



7



A lot of responses seem to be a poor fit for the question. We're not talking about using an unapproved language in an environment with existing standards. We're talking about a situation where the poster is the entire IT and development department for his company.

It's certainly important to keep in mind availability of talent, but Ruby is hardly a fringe language these days. In an environment where there's only one developer, productivity is also a very important consideration. Being able to build and maintain software quickly and easily without a large team requires tools with different characteristics than a large team might require.

I think what's more important than whether to use Ruby or (something else) is to try to pick something as general-purpose as practical and use it for everything unless there's a *really* good reason to use something else. If you go with Ruby, stick with Ruby for your utility scripts, cron jobs and that little GUI app the boss wanted to automatically SMS the intern when he takes more than five minutes to bring him his coffee.

Share Improve this answer

answered [Feb 4, 2009 at 22:06](#)

Follow

community wiki  
[Zak](#)

---



7



I think using python would be the right thing to do if it would meet the clients requirements, and save them money over the alternative. Whether or not there is a wide assortment of characters to work on the application down the road is irrelevant, unless they've specified this as a non-functional requirement.

As usual, using the best tool for the job at hand will serve you well.

Share Improve this answer

answered [Feb 4, 2009 at 14:41](#)

Follow

community wiki  
[AlexCuse](#)

---

Plus. python is really easy to learn. – [Seun Osewa](#) Feb 4, 2009 at 15:12

---

I'll second that... my consulting company will give a developer about 12 hours to come up to speed in Python, because that's all it takes for someone who knows any other language. It's gonna be more maintainable than .net, that's for sure. – [Jim Carroll](#) Feb 4, 2009 at 15:36

---

I agree that its' easy to learn, though I strongly disagree that it'd be more maintainable than .net, simply because it is python. Bad design is bad design. – [AlexCuse](#) Feb 4, 2009 at 15:41

---



It indeed is a bit crooked IF you use it only for that purpose.



6

However, if you use it because it IS the best solution, you're in the clear.



Also, they can just hire someone else who knows python.



My work ethics don't allow me to do something like this just to keep me in business.

Share Improve this answer

answered Feb 4, 2009 at 14:40

Follow

community wiki  
[Mostlyharmless](#)



3

My personal opinion is you should try where possible to respect the working practices of wherever you are - whether that's indentation style, naming convention, testing procedure or programming language.



If you feel strongly that a different language would be better suited for a certain task, then lobby to have it accepted (with the required re-training of others).



Purposefully leaving an app that no one else can maintain is very bad professional conduct, IMO.

Share Improve this answer

answered Feb 4, 2009 at 14:41

Follow

community wiki



2



We recently had a bad hire at my shop and he decided out of the blue he was going to use Perl instead of any version of .NET to do some simple reporting stuff (That could have just as easily been done in .NET). It was atrocious. I would suggest using the platform as specified and clearing any deviation with the people who run the joint...

Share Improve this answer

edited Feb 4, 2009 at 14:51

Follow

community wiki

2 revs, 2 users 67%

Jason Punyon

---

I've changed "PERL" -> "Perl". *But never write "PERL", because perl is not an acronym, apocryphal folklore and post-facto expansions notwithstanding.* [perldoc.perl.org/...](http://perldoc.perl.org/)  
– jfs Feb 4, 2009 at 14:54

---

@J.F. Sebastian - Thanks...I didn't know that the taxonomy was fraught with PERiL :) – [Restore The Data Dumps Again](#)  
Feb 4, 2009 at 15:10

---



2

Plenty of answers have touched on this, but here's my take based on production application support.

My company had a startup phase where code hustlers whipped up solutions in whatever the personal preference



or flavor of the week was. Bad for maintainability and supportability.



Making a change is ok, though, as long as it's consistent. If Python is going to pave the way to the future, then go for it. Don't forget that the legacy .NET and PHP code still needs to be supported until end of life. Building yourself a hodge podge of platforms and frameworks will just create more difficulty for you on the job and the company when you're no longer around.

Share Improve this answer

answered [Feb 4, 2009 at 15:32](#)

Follow

community wiki  
[spoulson](#)



If you feel in your heart you are acting dishonestly, then you probably are.

2

No one likes a dishonest person. That can't be good for your reputation.



Do your best to choose based on what is actually best, not what satisfies some underhanded motives.

Share Improve this answer

answered [Feb 4, 2009 at 17:04](#)

Follow

community wiki  
[Alex Baranosky](#)



1



It depends. I did some of what would normally just be a bash script, in Java instead at one place. Why? Because they're all Java programmers and frequently have interns/coops coming through that may or may not know anything else (and may or may not even be all that great with Java).



Other places though tend to have more experienced programmers and I expect that they'll be able to figure out another language without too much effort. So, I would go with what's "best" for the project.

Share Improve this answer

answered [Feb 4, 2009 at 14:42](#)

Follow

community wiki  
[Brian Knoblauch](#)



1



I agree with what mquander says above, but you may also have to be prepared to justify why you want to use this other language to your development manager. If he/she then agrees, perhaps the language could become more widely adopted within the company.



Share Improve this answer

answered [Feb 4, 2009 at 14:42](#)

Follow

community wiki

---

Well, this is a very small company so I *am* the development manager :) I get what you're saying, though.

– [Wayne Molina](#) Feb 4, 2009 at 14:48

---

Well then, if you've convinced yourself, then go ahead!

– [Ian Devlin](#) Feb 4, 2009 at 14:57

---



Think of it in terms of business benefit you bring to the company, now and in the mid-term.

1



If you can deliver something much faster using a different technology, and it still achieves the goals, I'd go for it - but I'd still let some other people know and respect the company's final decision. If however, it's purely for yourself, then I'd probably be a little more careful.



Share Improve this answer

answered [Feb 4, 2009 at 14:43](#)

Follow

community wiki

[Miles D](#)



I think it's a really bad idea. For you, it means there's no back up in case you want to have a day (or week) off. For them, there's no one else if you leave or are taking a day off. It's a well known ploy, and, honestly, might be reason to not keep you around.

1





However, this could also be a chance to introduce Python into the environment. You could teach others about it, and explain to management while it's a good third language to have at the group's disposal.

Share Improve this answer

answered [Feb 4, 2009 at 14:44](#)

Follow

community wiki  
[david.mchonechase](#)



**1**

I used to think that you should always pick the right language for the job at work. I'm reversing my opinion though.



The problem arises when some other guy picks a language you don't want to learn. I am concerned that I might be the guy who picks the language no one else wants to learn. Just because I think that Erlang might be the right choice for something doesn't mean that everyone else will want to learn Erlang or respect my decision for using Erlang.

Share Improve this answer

answered [Feb 4, 2009 at 14:51](#)

Follow

community wiki  
[Tim Stewart](#)

---

If it's *really* the right choice for a big system that it's your job to write or maintain, for some objective reasons that you can agree on, then perhaps it's simply your professional responsibility to learn Erlang? Not that people are really likely to actually agree on such a thing. – [mqp](#) Feb 4, 2009 at 14:55

---

One caution though... If you are selecting a language for a big system, you would be well served to get the language approved by your architects, managers, and perhaps even your CTO. If you can make the case to these folks that your language is the right one for the job you'll be in great shape.  
– [Tim Stewart](#) Feb 5, 2009 at 23:45

---



1

"if I were to leave the company, they'd basically have nobody to maintain/add to it unless they retain on me as a consultant."



Are you saying no one else can learn Python? I find that hard to believe.



New technology is often introduced in small projects by knowledgeable people and diffused through the organization because the small projects were successful.

Use Python. Be successful. Make your case based on your successes.

Share Improve this answer

answered Feb 4, 2009 at 14:57

Follow

community wiki

[S.Lott](#)



1



I had this same problem very often. Coincidentally, it was with those two languages you mention: .NET forced on me, when I preferred to use Python (among others). Could be the opposite, I don't judge.

I refrained to use Python, because of the reasons already mentioned in other answers. I did what I thought was best for the company. Using IronPython won't make your python code any more maintainable for an unexperienced Python programmer.

However, I left the company and now I work in something more in line with my tastes. I'm much happier. In this economy you may not have this option... but it will pass. Do the right thing.

Cheers.

Share Improve this answer

answered Feb 4, 2009 at 15:13

Follow

community wiki

[nachik](#)





1



There is a large difference between 'prototype' or 'one-shot' code and production code. For prototyping I use whatever works fastest, but I'm very clear about its status. Production code is written in one of the approved and supported environments.



Share Improve this answer

answered [Feb 4, 2009 at 15:38](#)



Follow

community wiki

[Stephan Eggermont](#)



1



The ethics is to use the best tool for the job. If there is a tool that takes you only 20% of the time to code vs other choices, and next to no maintenance, and easy to re-factor, you have a duty to pick that tool, assuming it's extensible as you may need in the business.



If you do a good job, hiring future people and training them in terms of *HOW* your workplace does business should be the practice of any growing business. They will be able to learn the code if they're the right person for the business.

In your case I'm not sure if you want to use Python, unless it has native .NET support to allow your .NET world to interact with it.

answered [Feb 4, 2009 at 15:44](#)

Share Improve this answer

Follow

community wiki

[Jas Panesar](#)

---

But besides coding time, he has to factor in other factors like maintainability. There is a balance that has to be maintained in that regard, no? – [Alex Baranosky](#) Feb 4, 2009 at 17:05

---

For sure. I find 80% of code is rarely touched again. If the specs are clearly written in terms of input/output it shouldn't be a big deal. We could be stuck with unmaintainable .NET code too. Employers who want it fast, cheap and right can get their way. Pay now or pay later, same thing?

– [Jas Panesar](#) Feb 4, 2009 at 17:16

---



1



Other posters have made some good points, but here's one I've not seen: Communicate the situation to management and let them decide. In other words, talk with your boss and tell him or her that there currently are more .NET developers in your area, so that if you're hit by a bus tomorrow it would be easier to find someone else to maintain your code; however, there are tools you need to do your job more efficiently and they cost money (and tell them how much). Alternatively, you could do this in Python or RoR (or whatever) and use free tools, but from what you know, there aren't currently that many people in the area who know those languages. I've used "currently" a couple times here because this may change over time.

Before having this conversation, it might be good to see if you can find user groups for the alternative technology in your area, and how large they are. You could also ask on listserves if there are people who know the alternatives in your area.

Of course, the boss may tell you to keep using .NET without any tools, but in that case it's their decision to shoot themselves in the foot. (And yours to decide if you want to find a new job.)

Share Improve this answer

answered [Feb 4, 2009 at 16:15](#)

Follow

community wiki  
[PTBNL](#)



Regarding the question as asked, I see nothing unethical about it, provided that:

**1**



1. It is a freely-available language. Although I am something of a FOSS partisan, that's not the point of this criterion. It needs to be freely-available (not necessarily FOSS) so that it doesn't impose costs on the company and so that others will have the opportunity to learn it if you ever need to be replaced (or if they want to compete with you for your job).



2. You are changing languages for solid reasons and not for the sake of creating vendor lock-in (or, if you prefer to think of it as such, "job security"). Ethics

aside, you really don't want to have a job where they hate you, but are stuck with you because you're the only one who can maintain the mess you've created anyhow.

In the particular case you've described, I would suggest that switching to RoR may be the more ethical choice, as it would be decidedly unethical (not to mention illegal) to use .NET if there are required resources which are for-pay only and your employer is too cheap frugal to purchase proper licenses for them.

Share Improve this answer

answered Feb 5, 2009 at 0:50

Follow

community wiki

[Dave Sherohman](#)



When in Rome... do as the Romans.

1



You might not be the one who as to maintain this code in the long term and not everyone wants to learn a "fringe language" to make bugfixes or enhancements.



I migrated some VBA stuff over to Perl for processing at a previous job and increased the efficiency by several orders of magnitude, but ultimately no-one else there was willing to learn Perl so I got stuck with that task longer than I wanted it.

Share Improve this answer

answered Feb 5, 2009 at 1:16

Follow

community wiki

[Chris Ballance](#)



1



I did that, it was Delphi in my case. I think Delphi was used often however when i was looking for a job .... i saw 3 delphi job offers in my whole life. i also saw more java/j2ee/php offers that i can remember. i think its bad idea, with the time i wasted in learning advance delphi programming i could get better with j2ee and start in better company and maybe make now more money.

If they cant find somebody to maintain the app you will always do it and when you quit they will have to re-write it. i think consultant thing is not used often.

Share Improve this answer

edited Feb 5, 2009 at 10:50

Follow

community wiki

[4 revs](#)

[lAdapter](#)

Three offers in your life? I did Delphi programming 10 years ago - haven't touched it since 1999 - and I've gotten email from three headhunters in the last week pimping Delphi jobs to me because they saw it on my resume.

– [Dave Sherohman](#) Feb 5, 2009 at 0:50

yes, in my whole life. maybe its because there are still a lot of delphi programmers and its only used for maintenance, so when someone quits/retires/dies they need somebody to replace him, but new projects dont use delphi. how is money in that delphi job? are they desperated? – [IAdapter](#) Feb 5, 2009 at 9:07

---



0



I used to be in the "use the best tool for the job" school, but I've changed my mind. It's not enough to just ask "how can I do this job the fastest." If you think you're the only one who will ever need to look at some code, there's a good chance you're mistaken. The total cost of introducing a new language into an environment is higher than you might imagine at first.

If you just need to produce a result, not a program, then you can use whatever you want. Say you need a report or you need to munge some files. If the output is really all that matters, say it's something you could have chosen to do by hand, you can practice using any language you want.

Share Improve this answer

answered [Feb 4, 2009 at 14:55](#)

Follow

community wiki  
[John D. Cook](#)



With the release of the MVC Framework I too have been in a similar ethical delema. Use WebForms or switch over

0



to MVC Framework for everything. The answer really is you have to do the right thing and use whatever the standard of the company is. If you deviate from the standard it creates a lot of problems for people.

Think how you would feel if you were dumped a project on VB6 when all you have been doing for years is .Net. So these are the two solutions I have come up with.

1. Use your fun languages for consulting contracts you do on the side. Make sure the client knows what you are doing and if they agree go for it.
2. Try and convince your current company to migrate over to this great new language you are working with.

If you follow these routes you will learn your language and not piss anyone off in the process.

Share Improve this answer

answered [Feb 4, 2009 at 14:56](#)

Follow

community wiki

[Al Katawazi](#)

---



0



Ruby on Rails is certainly not a *fringe* language. If the company is too cheap to pony up for the appropriate licensing for Microsoft's tools, then you would have no choice but to find an alternative. RoR certainly would be a reasonable choice and if helps move your career along as well, then it's win-win for both of you!

Share Improve this answer

answered [Feb 4, 2009 at 15:28](#)

Follow

community wiki

[Paul Lefebvre](#)



0



You can develop .NET adequately with free tools; cost is not a good reason to avoid that platform. Ruby on Rails is becoming reasonably mainstream for building data-driven internet websites. You haven't even told us if thats the sort of software you are building though.

There is really no way with the information that you have provided that anyone can give you a single correct answer.

If you are asking is it ethical to do your work in such a way that the company is dependent upon you, of course the answer is no. If you are asking is it ethical to develop in RoR then the answer is "we don't know" - but my opinion is that probably it would be fine if its the right tool for the job.



Don't underestimate the ability of someone else to support your work or replace you though - if you do your work reasonably well once the solution is in place any programmer worth their pay should be able to learn the platform well enough to maintain it. I've debugged, migrated and supported a few PHP applications for example without ever hardly learning the first thing about PHP. I'd be lost building a new PHP app from scratch and would never even try but its no problem to support one. I think the same would be true of the languages you mention as well - they've got the critical mass that means there is plenty of books and forums etc. Of course if its written badly enough in any language then it may be difficult to support regardless of anyone's skill in the language...

Share Improve this answer

answered [Feb 4, 2009 at 22:51](#)

Follow

community wiki

[Jeremy](#)

---

It would be for an e-commerce website and the accompanying back office ERP type system, as well as a custom order management system for creating quotes.

– [Wayne Molina](#) Feb 4, 2009 at 23:22

---

Rails is a pretty decent fit for that. As I mentioned and others did in the comments its not really such a fringe language - the only thing that worries me in that situation is the hosting side of it - if you are hosting with a 3rd party with decent rails support you are ok there though. – [Jeremy](#) Feb 5, 2009 at 0:38

---



0



So much discussion for such a clear-cut situation...

It's not up to you, it's up to them. If they're not technical enough to make the call, as it seems, then you have to make it for them *in good faith*. Anything less is dishonest, and I'm fairly sure that's not in your job description ;)



Share Improve this answer

answered [Feb 5, 2009 at 0:23](#)

Follow

community wiki

[jTresidder](#)

---



0



You've muddled the waters with all the wandering about in the thickets of personal motivations. The answer to that one is that your personal motivations are irrelevant unless and until you've formulated the business case for the possible decisions. If you've done that and the answer still isn't clear-cut, then sure, choosing the answer you like the best is one of the nice things about being in a position to make technical decisions in the first place.



As far as the actual question goes, to my mind if the most technically apt choice is also one that very few people work with, one of two things is happening: a) It's a good choice, and the number of people working with it is going to be exploding over the next 18-24 months (e.g. Django), or b) There's something wrong with my analysis. Technologies may be on the fringe because people are slow to adopt them, but that's generally not why they stay on the fringe.

If you find yourself thinking "I can't choose technology X, that'll make it easier for them to replace me!" you're in the wrong line of work. In almost any enterprise that's not actually failing, the IT guy who makes himself easy to replace tends to move up to harder and more interesting and more lucrative work.

Share Improve this answer

answered Feb 5, 2009 at 1:09

Follow



Robert Rossney

96.6k ● 24 ● 148 ● 218



0

I would not bring a new language/framework/whatever into the place unless they understood that's what I was doing, and that if I left/was fired/was hit by a bus, they'd have to find/train someone to work with it.



I have some experience in a contractor pulling in things just because he felt like it. In some cases they were the best tool for the job (in other cases they were not), but in all cases they were not the best tool for the team that had to maintain the code. In my case the contractor was a

serious jerk who didn't really give a darn about anyone else and I believe WAS trying to make himself harder to replace.

In your case, talk to your bosses. If they really don't want to spend the needed money on .NET framework tools/libs, then switching to something else may well BE the right thing to do for them, long term.

And, as someone who has spent his career walking into the middle projects that others have already started - thank you for thinking before you add a new tool to the mix.

[Share](#) [Improve this answer](#)

[edited Oct 19, 2012 at 3:02](#)

[Follow](#)

community wiki

[2 revs](#)

[Michael Kohne](#)

---

That's my thing - I know for a fact they won't spend the money on .NET tools (okay, they have 1 SQL Server license but it's on a two-processor server and they won't pay for a second one), so I'm left looking at free/open alternatives, the only one of which I kinda sorta know being Ruby on Rails

– [Wayne Molina](#) Feb 4, 2009 at 15:10

---